COMPUTATIONAL KINEMATICS

SOLID MECHANICS AND ITS APPLICATIONS
Volume 28

Aims and Scope of the Series

The fundamental questions arising in mechanics are: *Why?*, *How?*, and *How much?* The aim of this series is to provide lucid accounts written by authoritative researchers giving vision and insight in answering these questions on the subject of mechanics as it relates to solids.

The scope of the series covers the entire spectrum of solid mechanics. Thus it includes the foundation of mechanics; variational formulations; computational mechanics; statics, kinematics and dynamics of rigid and elastic bodies; vibrations of solids and structures; dynamical systems and chaos; the theories of elasticity, plasticity and viscoelasticity; composite materials; rods, beams, shells and membranes; structural control and stability; soils, rocks and geomechanics; fracture; tribology; experimental mechanics; biomechanics and machine design.

The median level of presentation is the first year graduate student. Some texts are monographs defining the current state of the field; others are accessible to final year undergraduates; but essentially the emphasis is on readability and clarity.

For a list of related mechanics titles, see final pages.

Computational Kinematics

edited by

JORGE ANGELES

*Department of Mechanical Engineering,
McGill University,
Montreal, Quebec, Canada*

GÜNTER HOMMEL
and
PETER KOVÁCS

*Department of Computer Science
Technical University of Berlin,
Berlin, Germany*

Springer-Science+Business Media, B.V.

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 978-90-481-4342-9 ISBN 978-94-015-8192-9 (eBook)
DOI 10.1007/978-94-015-8192-9

Printed on acid-free paper

Table of Contents

Preface

The aim of this book is to provide an account of the state of the art in *Computational Kinematics.* We understand here under this term .that branch of kinematics research involving intensive computations not only of the numerical type, but also of a symbolic nature.

Research in kinematics over the last decade has been remarkably oriented towards the computational aspects of kinematics problems. In fact, this work has been prompted by the need to answer fundamental questions such as the number of solutions, whether real or complex, that a given problem can admit. Problems of this kind occur frequently in the analysis and synthesis of kinematic chains, when finite displacements are considered. The associated models, that are derived from kinematic relations known as *closure equations*, lead to systems of nonlinear *algebraic* equations in the variables or parameters sought. What we mean by algebraic equations here is equations whereby the unknowns are numbers, as opposed to *differential* equations, where the unknowns are functions. The algebraic equations at hand can take on the form of multivariate polynomials or may involve trigonometric functions of unknown angles.

Because of the nonlinear nature of the underlying kinematic models, purely numerical methods turn out to be too restrictive, for they involve iterative procedures whose convergence cannot, in general, be guaranteed. Additionally, when these methods converge, they do so to only isolated solutions, and the question as to the number of solutions to expect still remains. These drawbacks have been overcome with the development of continuation techniques that are meant to produce all solutions to a given problem. While continuation techniques have provided solutions to a number of problems, they are still subjected to the uncertainties of iterative procedures. Hence, alternative approaches have been sought, that rely on modern software and hardware for symbolic computations. Commercial software of this kind is now very reliable and widespread; it has naturally found its way into kinematics research. In fact, current research in kinematics involves symbolic manipulations that were impossible to even imagine as recently as fifteen years ago, when the first symbolic manipulation packages started coming out of the computer science laboratories.

The book reports trends and progress attained in Computational Kinematics in a broad class of problems. In order to ease the task of the reader searching for information on particular topics, we have divided the book into six parts, namely, *i) kinematics algorithms*, whereby general kinematics problems are discussed in light of their solution algorithms; *ii) redundant manipulators*, which is self-descriptive; *iii) kinematic and dynamic control*, in which the link between kinematics and the disciplines of dynamics and control is highlighted; *iv) parallel manipulators*, in which an open problem is discussed, namely, the number of solutions of the associated direct kinematic problem; *v) motion planning*, touching upon computational geometry; and *vi) kinematics of mechanisms*, in which the main issue is the presence of closed kinematic chains, is discussed with regard to both analysis and synthesis.

The reader will find here a representative sample of the most modern techniques available nowadays for the solution of challenging kinematics problems. Thus, resultant methods based on dyalitic elimination are discussed critically, while Gröbner bases are proposed as a powerful alternative. Other, equally novel techniques, available only in conference proceedings of limited circulation, are included for the first time in book form.

In light of its contents, the book should be of interest to researchers, graduate students and practicing engineers working in kinematics or related problems. Especially, roboticists, biomechanicists, machine designers and computer scientists will find here a useful source of information comprising methods, algorithms and applications.

This book contains the Proceedings of the Workshop on Computational Kinematics, held at the *International Conference and Research Center for Computer Science* (IBFI), of Germany, from October 11 to October 15, 1993. IBFI is herewith given due acknowledgement for its financial and logistical support and encouragement. This support made it possible to bring together specialists of various disciplines working in the area. Among the participants, who met for one week at the Dagstuhl Castle of IBFI, we count engineers, computer scientists and biomechanicists, all of whom share a common interest, namely, Computational Kinematics. Prof. Dr. Reinhard Wilhelm, Scientific Director of IBFI, and his staff are especially acknowledged for their support. Dr. Nigel Hollingworth, of Kluwer Academic Publishers, is acknowledged for his encouragement and support in editing the book and publishing it in record time. The technical support of Mr. Kourosh Etemadi Zanganeh, a Research Assistant at the McGill Centre for Intelligent Machines, was decisive in bringing this book to completion.

Jorge Angeles, Günter Hommel and Peter Kovacs, Editors
Dagstuhl Castle, Germany

Part 1

Kinematics Algorithms

Computations in Kinematics

Bernard Roth
Department of Mechanical Engineering
Stanford University
Stanford, CA 94305
USA

Abstract

Several methods to solve sets of nonlinear equations are discussed. Then a modification of dialytic elimination is described, and two methods for obtaining new linearly independent equations are presented. Finally, one of the methods is applied to three quadratics and is shown to yield all of the solutions, without any extraneous roots.

Introduction

This paper deals with the numerical solution of sets of nonlinear equations which arise in the kinematics of mechanical systems. The most commonly used methods are iterative. All such methods require an initial guess at a solution, and, if the initial guess is not close to an acceptable solution, they tend to diverge, or converge very slowly, or converge to an unacceptable solution. Furthermore, nonlinear problems have more than one solution, and iterative methods produce only one of these — the solution "closest" to the initial guess. Once one solution is obtained, it is not any easier to obtain additional ones, and repetitive applications of iterative methods may not yield all solutions.

In order to overcome these difficulties, a technique known as the *bootstrap method* (Roth and Freudenstein 1963) was developed in the early 1960s and applied to kinematic synthesis of mechanisms, as well as to sets of nonlinear polynomial equations. This iterative procedure has been improved and is now known as the *homotopy* or *continuation method*. It is widely used in kinematics (for example, Wampler and Morgan (1990), and Raghavan (1993)). The advantage of the continuation method is that it incorporates a "good" initial guess and it yields all possible solutions. The disadvantages are that it is an iterative numerical procedure which (i) can have numerical difficulties, (ii) gives little or no information about how physical parameters influence the solutions and (iii) can require dealing with large numbers of unwanted solutions at infinity. This latter difficulty can be somewhat mitigated by the use of m-homogeneous coordinates (Raghavan 1993; Morgan and Sommese 1987).

J. Angeles et al. (eds.), Computational Kinematics, 3–14.

In this paper we will discuss non iterative methods which yield all the solutions to sets of nonlinear equations, and also partially overcome the disadvantages mentioned in the preceding paragraph. *Gröbner bases* and *elimination methods* are two classes of methods which potentially have the characteristics we are interested in. Gröbner bases seems to have first been published in 1964 under the name *standard bases*. They were then renamed and applied to various problems including the solution of multivariate polynomial sets (Buchberger 1985) and geometrical theorem proving. To-date this method seems inefficient for most of the types of problems we are interested in since it is prone to exploding intermediate results and computational time. However, some of the ideas behind the method are useful to us.

Currently, the elimination methods seem to be the most promising analytical techniques. The original ideas come from Cayley (1848). All elimination methods require forming equations known as eliminants or resultants. Summaries of various types of eliminants and resultants can be found in several books (Salmon 1885; Van der Waerden 1964). The elimination methods will, in theory, lead to a solution of any system of multivariate polynomial equations. In practice the method can only be applied to relatively simple equations — beyond these it explodes in complexity and introduces large numbers of extraneous solutions. There have been various attempts at improving these methods in order to make them practical computational tools, for example Arnon et al. (1984) and Canny (1987). Although there has been some success in certain problem areas, see for example Macaulay's resultant and the u-resultant (Lazard 1981), the basic methods remain computationally too costly to be applied to the types of multivariate polynomial problems which are common in kinematics.

One elimination method, known as *Sylvester's dialytic method* (Salmon 1885), has been used in kinematics to eliminate one or two unknowns from small sets of equations (Bottema and Roth 1990). Although this method has been known for a long time, its use has been limited because it is not practical for problems with more than two or three unknowns or equations of high degree. Recently we have been able to modify the dialytic method, in order to make it part of a more practical approach. In the following, first the basic dialytic method is described, then its limitations are pointed out with suggestions on how these may be overcome.

Dialytic elimination methods

There are six basic steps in using the dialytic elimination method to solve a nonlinear set of equations. Although the steps are easy to explain, the ideas behind Step 2 and Step 3 at first-sight seem strange, and even incorrect. The basic steps are:

1. Rewrite equations with one variable suppressed.
2. Define the remaining power products as new linear, homogeneous unknowns.

3. Obtain new linear equations so as to have as many linearly independent homogeneous equations as linear unknowns.
4. Set the determinant of the coefficient matrix to zero, and obtain a polynomial in the suppressed variable. (If one is interested in only numerical solutions, this step can be omitted if we calculate eigenvalues in Step 5.)
5. Determine the roots of the characteristic polynomial or the eigenvalues of the matrix. (This yields all possible values for the suppressed variable.)
6. Substitute (one of the roots or eigenvalues) for the suppressed variable and solve the linear system for the remaining unknowns. Repeat this for each value of the suppressed variable.

These steps can best be explained with a simple example. Consider the following two nonlinear equations:

$$axy^3+bx^3+cy^3+dx^2y+ex^2+f=0 \;;\; gx^4+hxy^3+ix^3+jx^2+kxy^2+\ell x+m=0. \qquad (1)$$

Here, a,b,c,d,e,f,g,h,i,j,k,ℓ and m are known coefficients, and x and y are the unknowns. Normally one considers these to be two fourth–degree algebraic equations in two unknowns. For elimination theory it is useful to take a different viewpoint. First we suppress one of the variables, say, x; i.e., we assume the value(s) of this variable are known and therefore it becomes part of the coefficients. In this example we obtain for Step 1:

$$(ax+c)y^3+(dx^2)y+(bx^3+ ex^2+f)=0 \;;\; (hx)y^3+(kx)y^2+(gx^4+ix^3+jx^2+\ell x+m)=0 \qquad (2)$$

The two equation now are in the form:

$$Ay^3 + By + C = 0 \;\;;\;\; Dy^3 + Ey^2 + F = 0\ , \qquad (3)$$

where the new coefficients, A,B,C,D,E and F, contain the suppressed variable x. No matter how many equations we have, the first step is the same: suppress one variable (or a single power product) and thereby reduce the number of explicit variables by one. (There are also cases where it is advantageous to simultaneously suppress several variables or power products.)

In Step 2, each power product is considered as a separate independent linear unknown. In order to illustrate this concept in a multivariable setting, it is instructive to revisit equation (1). In equations (1), if we consider the power products, we have nine unknowns, viz. x^4, xy^3, x^3, y^3, x^2y, x^2, xy, x, 1. It is important to note that we count the number 1 as a variable. The reasons for doing this are (i) it is convenient to always have homogeneous equations and (ii), as we shall see, it provides a rationale to discount trivial solutions. The coefficient of the "variable" 1 is the constant term. Taking this viewpoint is equivalent to rewriting equations (1) as the linear set:

$$aX_1 + bX_2 + cX_3 + dX_4 + eX_5 + fX_6 = 0 \; ; \; gX_7 + hX_1 + iX_2 + jX_5 + kX_8 + LX_9 + mX_6 = 0.$$

The result is that the equations are transformed in to linear equations. However, instead of simply x and y, we have seven additional variables. So far this change of variables seems rather trivial. However, it is here that the first important idea comes into play: in order to maintain the linear structure, we never explicitly employ any relationships between the new linear variables. So for example the fact that $X_9 \cdot X_9 = X_5$, $X_8 \cdot X_9 = X_4$, $X_9 \cdot X_5 = X_2$, etc. (i.e., $x \cdot x = x^2$, $xy \cdot x = x^2y$, $x \cdot x^2 = x^3$, etc.) is not used. Hence, we regard all nine variables as <u>independent</u> linear variables. The price we pay for this is that we now must introduce new equations in order to assure that the number of equations and unknowns are commensurate. Applying these ideas to the equations with the suppressed variable, (3), we obtain as a result of Step 2:

$$AY_1 + BY_2 + CY_3 = 0 \; ; \; DY_1 + EY_4 + FY_3 = 0. \tag{4}$$

Where Y_1, Y_2, Y_3 and Y_4 are linearly independent unknowns. Since we have two equations with four unknowns, we need additional equations. In Step 3, we obtain the additional equations. In our example this can be accomplished by multiplying equations (3) by y and then y^2. The results are four new equations (with only two new power products). Using the concept of Step 2, i.e., labeling every power product as an independent linear unknown, we can write these new equations as:

$$AY_5+BY_4+CY_2=0 \; ; \; DY_5+EY_1+FY_2=0 \; ; \; AY_6+BY_1+CY_4=0 \; ; \; DY_6+EY_5+FY_4=0. \tag{5}$$

This completes Step 3, since if we combine equations (5) and (4) we have a system of six homogeneous linear equations in six linear unknowns. This step contains the second major idea in the dialytic elimination method. Namely, new <u>linearly</u> independent equations can be obtained from the original set of equations simply by multiplying the original equations by one or more of the unknowns or power products of the unknowns. The key here is that even though the new equations are dependent on the original equations their dependence is not linear and hence they are linearly independent.

We are now ready for Step 4, in which we obtain a single polynomial equation in the suppressed variable. In our example, we rewrite equations (4) and (5) in a combined matrix form:

$$\begin{bmatrix} A & B & C & 0 & 0 & 0 \\ D & 0 & F & E & 0 & 0 \\ 0 & C & 0 & B & A & 0 \\ E & F & 0 & 0 & D & 0 \\ B & 0 & 0 & C & 0 & A \\ 0 & 0 & 0 & F & E & D \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{bmatrix} = 0 \ . \qquad (6)$$

Since we know that $Y_3 = 1$, it is clear that the trivial solution of $Y_i = 0$ $(i=1,2,...,6)$ is not admissible, and therefore the determinant of the coefficient matrix must be equal to zero. Expanding this determinant, and setting the result equal to zero, yields a polynomial equation in A,B,C,D,E,F. If we then substitute into this polynomial the expressions for the suppressed variable (viz.:$A=ax+c$, $B=dx^2$, $C=bx^3+ex^2+f$, $D=hx$, $E=kx$, $F=gx^4+ix^3+jx^2+\ell x+m$) we obtain a polynomial in the suppressed variable, x, and the original coefficients, a,b,c,d,e,f,g,h,i,j,k,ℓ,m. The term with the highest power in x is $a^3g^3x^{15}$ (it comes from A^3F^3); the resulting polynomial is of degree 15 in x.

The result of Step 4 will always be a polynomial in terms of the suppressed variable (or power product). For design studies and theoretical concepts there is a large advantage in obtaining the polynomial coefficients explicitly. However, in cases where we need only a numerical answer, obtaining the coefficients explicitly may add unnecessary computation time. Instead, we can threat (6) as an eigenvalue problem and determine the values of x in that way (Golub and Van Loan 1985; Manocha and Canny 1992). Step 4 can be omitted in such cases. There are problems for which the degree of the polynomial is too large for practical numerical computation. In such cases, an eigenvalue or alternative computation methods needs to be employed to deal with the matrix equation.

This brings us to Step 5. Here we obtain <u>all</u> the values of the suppressed variable (or power product) by using a root–finder routine for the roots of the suppressed–variable polynomial. In our example, we get 15 values of x as the roots of the 15th degree polynomial, or from an eigenvalue routine applied to (6). Usually we are only interested in real roots, so any complex or imaginary roots will reduce the number of actual solutions from the maximum possible value of 15.

Finally, in Step 6, we substitute for the suppressed variable in to the linear set of equations, and solve for the other original variables. In our example, we can use (6) to obtain y: substitute one root of x into (6), set $Y_3 = 1$, and then use any five of the resulting equations to solve for Y_2. Note that, because the system is linear, this will yield one y for each x. Thus the number and character (i.e., real or imaginary) of solutions are generally determined by the roots of the polynomial (or the eigenvalues). Exceptions to this occur only when the linear system, (6) in our example, has rank less than n–1 (where n×n is the size of the matrix). Hence, regardless of the size of the problem, if the rank the n×n is n–1 for each root, all the other variables are linear (single valued) functions of each root of the suppressed–variable polynomial.

Basic difficulty

In principle, the foregoing procedure will always work if enough new equations can be determined from the original equations to obtain an n×n system of linear homogeneous equations. Thus the crucial step is Step 3, where we obtain the additional equations. However there is a subtle and important practical caveat: It is not enough to obtain an n×n system, it is important that the value of n be as small as possible. If the procedure introduces extraneous roots, n is larger than its minimum value and the suppressed variable polynomial is of higher order than is necessary. In most problems with more than two variables, multiplying the equations by the variables create sets of linear equations where the values for n are much larger than they need to be.

In some problems we inherently have a very large value for n. It is not unheard of for nonlinear systems irt kinematics problems to have several hundred or even several thousand solutions. In such systems, it is extremely important not to introduce extraneous power products, since these increase the number n geometrically. To minimize n we need to, as far as possible, avoid introducing new power products and, if possible, eliminate solutions at infinity.

Even if we do not introduce extraneous roots, it may turn out that the value of n is so large that literal or even numerical calculation of the suppressed–variable polynomial becomes unfeasible. In such cases we need to resort to other means of dealing with the matrix in Step 4. The important idea here is that even if the entire polynomial cannot be obtained as a literal expression, it is still possible to determine properties of the physical system by studying the affect of the parameters on selected polynomial terms or the rank of the matrix. In this, we can utilize existing methods for dealing symbolically and numerically with matrices with polynomial entries, see for example Horowitz and Sahni (1975).

Minimum n

The theoretical minimum value for n is not a simple matter to determine. Even when it is known, it is not easy to obtain such a set of equations. In our example, we see that once we suppressed x, the resulting set of equations (3) had four power products. In our solution we used a 6×6 set of equations, and so our n was 6 not 4. The resulting polynomial was of degree 15. This seems to be a reasonable result, since we started with two fourth–degree polynomials. We know that two, non degenerate, fourth–degree polynomials have 16 common values. In this case because of the special nature of the fourth–degree terms one solution is at infinity regardless of the values of the coefficients, and that accounts for the degree of the suppressed variable polynomial being 15 rather than 16. Knowing what n should be is very useful. To–date the only two practical methods are to make use of the m–homogeneous Bezout count (Morgan and Sommese 1987), or to rely on physical insights into the nature of the solution.

A further complication is that the size of the nxn system is only indirectly related to the degree of the suppressed variable polynomial. For example, if in our problem we had suppressed y instead of x, then our equations (2) would become:

$$(b)x^3+(dy+e)x^2+(ay^3)x+(cy^3+f)=0 \;;\; (g)x^4+(i)x^3+(j)x^2+(hy^3+ky^2+L)x+m=0$$

Here we have five power products, and a balance between unknowns and equations occurs at n=7. (The additional five equations can be obtained by multiplying the first equation three times and the second twice by x.) If we then expand the 7×7 determinant, we will again get a 15th degree polynomial in the suppressed variable — this time y. So for this example we see that an n of 6 and an n of 7 yield suppressed–variable polynomials of the same degree.

Determining the theoretically smallest possible n is directly analogous to the problem of determining the set of variable groupings which give the smallest Bezout count (Morgan and Sommese 1987). This can be determined by exhaustive checking of all groupings (Raghavan 1993). A much greater difficulty occurs when the number of power products grows rapidly as we multiply equations by one or more variables. This effect is very pronounced when we have more than one non suppressed variable. For example if our original equations contained three, rather than two, unknowns, say:

$$axy^3 + bx^3z + cy^3z + dx^2y + ex^2z^2 + f\,z = 0$$
$$gx^4 + hxy^3 + ix^3 + jx^2 + kxy^2z + Lxz^3 + m = 0 \;.$$

Then after suppressing x we obtain:

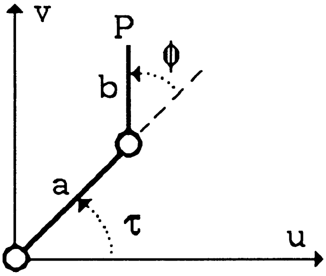
$$(c)y^3z + (ax)y^3 + (ex^2)z^2 + (dx^2)y + (bx^3+f)z = 0$$
$$(hx)y^3 + (Lx)z^3 + (kx)y^2z + (gx^4+ix^3+jx^2+m) = 0.$$

Here we have eight power products as opposed to the four we had in equation (3). If we multiply both equations by y we get two additional equations but now we have seven additional power products. Even with a third original equation without any new power products in it, it is clear that by the time we got to an equal number of equations and power products, the resulting value for n would be much larger than the eight power products we started with.

Minimum number of power products — first method

There appear to be ways to obtain new equations (for Step 3) without increasing the number of power products. One of the most promising is the method that was discovered in connection with the manipulator inverse kinematics problem (Raghavan and Roth 1990). The idea is rather simple: one obtains the additional equations by combining the original equations rather than multiplying them by a variable.

This can be illustrated with a simple example:
Consider the two links shown the Figure. The link of length a is connected to ground with a revolute joint and the link of length b is connected to link a, also with a revolute joint. The coordinates of the tip P are u,v as measured in the fixed coordinate system shown. Here we assume that the lengths a and b are known and the tip coordinates u,v are known, and we wish to determine the link angles τ and ϕ. From the geometry it follows: $u = a \cos\tau + b \cos(\tau+\phi)$; $v = a \sin\tau + b \sin(\tau+\phi)$



i.e., $u = a \cos\tau + b \cos\tau \cos\phi - b \sin\tau \sin\phi$
$v = a \sin\tau + b \sin\tau \cos\phi + b \cos\tau \sin\phi$.

These have seven power products: (1, $\cos\tau$, $\cos\tau \cos\phi$, $\sin\tau \sin\phi$, $\sin\tau$, $\sin\tau \cos\phi$, $\cos\tau \sin\phi$) and two equations. Getting new equations by multiplying by the power products further increases the number of power products. Instead we square each equation:

$$u^2 = a^2 \cos^2\tau + b^2 \cos^2\tau \cos^2\phi + b^2 \sin^2\tau \sin^2\phi + 2a\,b \cos^2\tau \cos\phi$$
$$-2a\,b \cos\tau \sin\tau \sin\phi - 2b^2 \cos\tau \cos\phi \sin\tau \sin\phi \qquad (7)$$
$$v^2 = a^2 \sin^2\tau + b^2 \sin^2\tau \cos^2\phi + b^2 \cos^2\tau \sin^2\phi + 2a\,b \sin^2\tau \cos\phi$$
$$+ 2a\,b \cos\tau \sin\tau \sin\phi - 2b^2 \sin\tau \cos\phi \cos\tau \sin\phi \;.$$

This gives us two new linearly independent equations but these equations contain ten new power products, so the situation seems to have deteriorated. However if we sum these two equations we get:

$$u^2 + v^2 = a^2 + b^2 + 2a\,b \cos\phi \;. \qquad (8)$$

This equation is exactly what we are looking for: it does not add any new power products to the original set! Now if we suppress all the τ terms we have from (7) and (8):

$$(b \cos\tau)\cos\phi - (b \sin\tau) \sin\phi + (a \cos\tau - u) = 0$$
$$(b \sin\tau)\cos\phi + (b \cos\tau) \sin\phi + (a \sin\tau - v) = 0 \qquad (9)$$
$$(2a\,b) \cos\phi + (a^2 + b^2 - u^2 - v^2) = 0 \;.$$

The set (9) can be viewed as three linear homogeneous equations in the three power products cosϕ, sinϕ, 1. Hence we need to equate the determinant of its coefficient matrix to zero. This yields a polynomial in terms of the suppressed variable τ (actually, it is a linear equation in terms of cosτ and sinτ), from which two values of τ follow. Then, with τ known, the first two equations in (9) are a linear set for cosϕ and sinϕ.

Clearly, in this example equation (8) could be solved directly for cosϕ, and this could be used to simplify the problem. We have not done this here, since our interest is in showing how (8) can be used in the general procedure. The point is that we are able to obtain this additional linearly independent equation by squaring and adding the original equations, and this new equation does not contain any additional power products.

This example touches upon the key to solving very complicated sets of equations. Namely: it is necessary to determine ways to manufacture new linearly independent equations from the original equations, and to do so in a manner which does not introduce new power products, or at worst introduces less power products than equations. In Raghavan and Roth (1990), we have shown how to produce eight new linearly independent equations from an original set of six by using the notions of vector products, and what is most important is that the power products in the eight new equations are the same as in the original six. With these equations we are able to obtain a minimum-degree, suppressed-variable polynomial — in this case the degree is sixteen. We are able to use these fourteen equations to determine the inverse kinematics of open chain manipulators with six revolute joints.

In Raghavan and Roth (1992), it is shown that exactly the same procedure can be used when some of the joints in the manipulator are prismatic, and in Raghavan and Roth (1993), it is shown that this method can be used for the inverse kinematics of single loop spatial linkages with lower pair joints. We conclude, therefore, that these operations are of general utility in solving mechanism kinematics problems.

Minimum number of power products — second method

Another method that sometimes yields new equations without increasing the power products relies upon the fact that the derivatives of the determinant of the Jacobian of a system of equations (written in terms of homogeneous coordinates) have the same zeros as the original system of equations (Salmon 1885). This method can be applied to the problem of finding the common intersection points of a system of three quadratic surfaces. This is an important practical problem which appears in connection with computer aided design and other aspects of kinematics (Morgan and Sarraga 1982). As far as the author knows, the solution presented here has not been previously published.

From Bezout's theorem we know that three quadratic surfaces have 2×2×2=8 common points. The goal here is to show how to use our methods to obtain these eight points directly, without any extraneous values. For the most general quadratics:

$$a_i x^2 + b_i y^2 + c_i z^2 + d_i xy + e_i xz + f_i yz + g_i x + h_i y + i_i z + j_i = 0,\ i=1,2,3 \qquad (10)$$

If we suppress z we have:

$$a_i x^2 + b_i y^2 + d_i xy + (e_i z+ g_i)x + (f_i z +h_i)y + (c_i z^2+i_i z+j_i) = 0,\ i=1,2,3 \qquad (11)$$

We have three equations and six power products, and so we need at least three more equations. We will obtain these by using the Jacobian of the system. First, however, we convert to homogeneous coordinates: substituting $x= X/W$, $y=Y/W$ and then multiplying by W^2. The result is:

$$a_i X^2+b_iY^2+d_i XY+(e_i z+g_i)XW+(f_i z+h_i)YW+(c_i z^2+i_i z+j_i)W^2=0,\ i=1,2,3 \qquad (12)$$

The Jacobian matrix, J, of (12), with respect to the homogeneous coordinates is: J=

$$\begin{bmatrix} 2a_1X+d_1Y+(e_1z+g_1)W & 2b_1Y+d_1X+(f_1z+h_1)W & (e_1z+g_1)X+(f_1z+h_1)Y+2(c_1z^2+i_1z+j_1)W \\ 2a_2X+d_2Y+(e_2z+g_2)W & 2b_2Y+d_2X+(f_2z+h_2)W & (e_2z+g_2)X+(f_2z+h_2)Y+2(c_2z^2+i_2z+j_2)W \\ 2a_3X+d_3Y+(e_3z+g_3)W & 2b_3Y+d_3X+(f_3z+h_3)W & (e_3z+g_3)X+(f_3z+h_3)Y+2(c_3z^2+i_3z+j_3)W \end{bmatrix}$$

If we form the determinant of this matrix, a cubic polynomial follows:

$$\begin{aligned} |J| = {} & A(z)\, X^3 + B(z)\, X^2 Y + C(z^2)\, X^2 W + D(z)\, X Y^2 + E(z^3)\, X W^2 \\ & + F(z^2)\, X Y W + G(z)\, Y^3 + H(z^2)\, Y^2 W + I(z^3)\, Y W^2 + J(z^4)\, W \qquad (13) \end{aligned}$$

The coefficients A,B,C,...,J are functions of $a_i,b_i,c_i,\ldots j_i$ (i=1,2,3) and z. The parentheses are used to indicate that these coefficients are functions of z, and the power of z indicates the highest degree that it appears in that coefficient. Now, we take the derivatives of this equation with respect to the homogeneous coordinates:

$$\begin{aligned} |J|_X &=3A(z)X^2 + 2B(z)\, X Y + 2C(z^2)\, X W + D(z)\, Y^2 + E(z^3)\, W^2 + F(z^2)\, Y W \\ |J|_Y &=B(z)X^2+2D(z)XY+F(z^2)XW+3G(z)Y^2+2H(z^2)YW+I(z^3)W^2 \qquad (14) \\ |J|_\omega &=C(z^2)X^2+2E(z^3)XW+ F(z^2)\, X Y + H(z^2)\, Y^2 + 2I(z^3)\, Y W + 3J(z^4)\, W^2 \end{aligned}$$

If we set equations (14) equal to zero we obtain three new equations which have the same zeros as the original set of equations, as given in (12). The main thing to notice is that the power products in (14) are identical to those in (12). So now we have achieved our ideal goal, we have the same number of equations as power products, and we have accomplished this without introducing any new power products. Rewriting equations (12) and (14) we obtain:

$$\begin{bmatrix} a_1 & b_1 & d_1 & (e_1z+g_1) & (f_1z+h_1) & (c_1z^2+i_1z+j_1) \\ a_2 & b_2 & d_2 & (e_2z+g_2) & (f_2z+h_2) & (c_2z^2+i_2z+j_2) \\ a_3 & b_3 & d_3 & (e_3z+g_3) & (f_3z+h_3) & (c_3z^2+i_3z+j_3) \\ 3A(z) & D(z) & 2B(z) & 2C(z^2) & F(z^2) & E(z^3) \\ B(z) & 3G(z) & 2D(z) & F(z^2) & 2H(z^2) & I(z^3) \\ C(z^2) & H(z^2) & F(z^2) & 2E(z^3) & 2I(z^3) & 3J(z^4) \end{bmatrix} \begin{bmatrix} X^2 \\ Y^2 \\ XY \\ XW \\ YW \\ W^2 \end{bmatrix} = 0 \quad (15)$$

What remains now is to set the determinant of the coefficient matrix equal to zero (or to determine the eigenvalues of this matrix). If we expand this determinant, we obtain an eight-degree polynomial in z. The coefficients of this polynomial are functions of the coefficients of the original three equations, so the roots can be readily determined if the surfaces are known.

For each real root, we can then substitute for z into (15) and solve the linear system for x and y. The easiest way to do this is to set W = 1, in which case the first five equations of (15) can be rewritten as:

$$\begin{bmatrix} a_1 & b_1 & d_1 & (e_1z+g_1) & (f_1z+h_1) \\ a_2 & b_2 & d_2 & (e_2z+g_2) & (f_2z+h_2) \\ a_3 & b_3 & d_3 & (e_3z+g_3) & (f_3z+h_3) \\ 3A(z) & D(z) & 2B(z) & 2C(z^2) & F(z^2) \\ B(z) & 3G(z) & 2D(z) & F(z^2) & 2H(z^2) \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \\ x \\ y \end{bmatrix} = - \begin{bmatrix} (c_1z^2+i_1z+j_1) \\ (c_2z^2+i_2z+j_2) \\ (c_3z^2+i_3z+j_3) \\ E(z^3) \\ I(z^3) \end{bmatrix}$$

From this, it is obvious that in general we get one value of x and y for each z. Furthermore, since the polynomial in z can be obtained with explicit literal coefficients, we have a way of studying all the singular and special cases. (An alternative way of identifying singular cases is to determine when the rank of equation (15) is less than five.) Since, in general, we have at most eight real intersection points, this method gives us all the intersection points without any extraneous roots.

Summary

A basic six-step elimination method has been detailed, and two methods for obtaining new linearly independent equations have been described. Finally, for three quadratics it is shown how to find (i) all of the solutions, and (ii) an analytical means to determine the affect of the system parameters on the number and character of the solutions.

Acknowledgment

The financial support of the NSF is gratefully acknowledged.

Bibliography

Arnon, D., Collins, G. and McCallum, S., 1984, "Cylindrical Algebraic Decomposition 1: The Basic Algorithm," *SIAM, J. of Computation*, v.13, 4.

Bottema, O. and Roth, B., 1990, *Theoretical Kinematics,* reprinted Dover Publications, NY, 558p.

Buchberger, B., 1985, "Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory," Chapter 6 in *Multidimensional System Theory,* ed. N. K. Bose. D. Reidel Publishing Co.

Canny, J. F., 1987, *The Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA.

Cayley, A., 1848, "On the Theory of Elimination," *Cambridge and Dublin Mathematics Journal,* v.3.

Golub, G. and Van Loan, C., 1985, *Matrix Computations*, The Johns Hopkins University Press.

Horowitz, E. and Sahni, S., 1975, "On Computing the Exact Determinant of Matrices with Polynomial Entries," *J. of the ACM*, v.22, 1, pp.38–50.

Lazard, D., 1981, "Resolution des systemes d'equations algebriques," *Theoretical Computer Science*, v. 15, pp.77–110.

Manocha, D. and Canny, J. F., 1992, "Real Time Inverse Kinematics for General 6R Manipulators," *Proceedings, 1992 IEEE International Conf. on Robotics and Automation*, v.1, pp.383–389.

Morgan, A. P. and Sarraga, R. F., 1982, "A method for computing three surface intersection points in GMSOLID," ASME, 82–DET–41.

Morgan, A. P. and Sommese, A. J., 1987, "A homotopy for solving general polynomial systems that respect m–homogeneous structures," *Appl. Math. Comput.*, v.24, pp.101–113.

Raghavan, M., 1993, "The Stewart Platform of General Geometry Has 40 Configurations," *J. of Mech. Design, Trans. ASME*, v.115, 2, pp.277-282.

Raghavan, M. and Roth, B., 1990, "Kinematic Analysis of the 6R Manipulator of General Geometry," *Robotics Research, The Fifth Inter. Symp.*, eds. H. Miura and S. Arimoto, MIT Press, pp.263–270.

Raghavan, M. and Roth, B., 1992, "A General Solution of the Inverse Kinematics of All Series Chains," *Proceedings Eight CISM–IFToMM Symposium on Theory and Practice of Robots and Manipulators,* (*RoManSy–90*), Cracow, Poland, pp.24–32.

Raghavan, M. and Roth, B., 1993, "Inverse Kinematics of the General 6R Manipulator and Related Linkages," *J. of Mech. Design, Trans. ASME*, v.115, 3.

Roth, B. and Freudenstein, F., 1963, "Synthesis of Path–Generating Mechanisms by Numerical Methods," *J. of Engineering for Industry, Trans. ASME*, v.85, pp.298–307.

Salmon, G., *Higher Algebra*, 5th Edition (1885), reprinted Chelsea Publishing Co., NY, 1964, 376p.

Van der Waerden, B., 1964, *Modern Algebra*, vol.2, Frederick Unger Publishing Co.

Wampler, C. W., Morgan, A. P., and Sommese, A. J., 1990, "Numerical Continuation Methods for Solving Polynomial Systems Arising in Kinematics," *J. of Mech. Design, Trans. ASME*, v.112, pp.59–68.

Reducing the Inverse Kinematics of Manipulators to the Solution of a Generalized Eigenproblem

Masoud Ghazvini
Institute of Robotics, ETH Zurich
CH-8092 Zurich, Switzerland
e-mail: ghazvini@ifr.ethz.ch   FAX: +41 (1) 252 02 76

ABSTRACT. A new method is presented which reduces the full determination of the inverse kinematics of manipulators with revolute and prismatic joints to the solution of a generalized eigenproblem $\mathbf{G}\cdot\rho+\lambda\,\mathbf{H}\cdot\rho=\mathbf{0}$. Related single-loop mechanisms can also be solved. The eigenproblem-method is numerically stable, because eigensystems can be computed without previous determination of the characteristic polynomial. Furthermore, a compact and efficient formulation of the basic equations is shown. Numerical results are reported at the end.

1. INTRODUCTION

In the study of robot manipulators there are two basic problems. The first, known as the forward kinematics, requires as input the geometry and the joint variables of the manipulator. Output are the position and orientation of the endeffector. This task is easily solved and has a unique solution. The second problem is just the reverse: given the position and orientation of the endeffector, compute all the joint variables. The problem is highly nonlinear, and multiple solutions exist. Because the equations are nonlinear, there are closed-form solutions only for some special manipulators. Other manipulators can merely be solved with numerical and iterative methods. This paper focus on manipulators of the last kind, however, the method can also be applied to other mechanisms.

A first general method to obtain one inverse kinematic solution can be traced back to (Uicker, Denavit, Hartenberg 1964). They described the problem as an overconstrained system of nonlinear equations. Starting at an appropriate initial value, they solved the equations with a least-square-method.

The first complete solution, i.e. with all the configurations, originates from (Tsai, Morgan 1985). The problem has been formulated as a system of eight second-degree equations in eight unknowns. The iterative numerical solution is done with a continuation method. The method starts with 256 initial solutions. Most of them are eliminated during the solution process. Their experiments suggest that there are only 16 solutions for the general 6R-manipulator.

A full and theoretically correct solution to the inverse kinematics of closed-loop 7R-mechanisms was given by (Lee, Liang 1988). They derived a 16th degree polynomial in the half tangens of one joint variable and mapped the 16 roots to the 16 configurations. Unfortunately, the roots are very sensitive to the unavoidable roundoff errors in the polynomial coefficients (Wilkinson 1969).

The generalization of the polynomial-method to any open-loop manipulator and related

J. Angeles et al. (eds.), Computational Kinematics, 15–26.

single-loop mechanism can be found in (Raghaven, Roth 90) and (Lee 90). Polynomials of degree n=2,4,8,16 were derived.

The basic new idea in this work is to reduce the inverse kinematics of nonredundant manipulators with revolute and prismatic joints and related single-loop mechanisms to a *generalized eigenproblem*

$$\mathbf{G}\cdot\mathbf{p}+\lambda\,\mathbf{H}\cdot\mathbf{p}=\mathbf{0} \qquad \mathbf{G},\mathbf{H}\in\mathbb{R}^{n\times n};\ \mathbf{p}\in\mathbb{C}^{n};\ \lambda\in\mathbb{C};\ n\in\{2,4,8,16\} \qquad (1)$$

The eigenvalues and right eigenvectors will be computed with stable standard numerical methods. Therefore the *eigenproblem-method* combines the advantages of determining all the configurations and delivering numerically accurate results.

The method will particularily be used for manipulators with general geometry and without closed form solutions. However, the eigenproblem-method is not restricted to such manipulators, which is expressed in (1) by $n\in\{2,4,8,16\}$. By virtue of the fact that any lower-pair joint can be modelled by a combination of prismatic and revolute joints, manipulators with such joints may also be solved with this method.

The sequel starts with an introduction of used notation and terms. Next, an efficient formulation of the extension of the closure equations for a 6R-manipulator is worked out, followed by a discussion of the obtained basic equations and their special properties. Afterwards the generalized eigenproblem for the inverse kinematics problem is derived. The issue of enhancing the method to prismatic joints is taken up in a separate section. Finally, the numerical properties of the eigenproblem-method are shown by an example.

2. FUNDAMENTALS

This paragraph is intended to be a short introduction to used terms and notation. For more information it is referred to textbooks like (Paul 81) or (Craig 89).

A *manipulator* (arm) can be modelled to consist of rigid-bodies, or *links*, that are connected by *revolute (R)* or *prismatic (P) joints*. Furthermore, we will make the restriction to manipulators with rigid-bodies connected in series. Therefore each body has at least one and at most two neighbours. The link at one end of the chain is fixed to a nonmovable *base*, the link at the other end is free and is called the *endeffector* (hand).

Because of its complexity we begin with the nonredundant 6R (Revolute) manipulator and extend later to easier solvable manipulators.

For the purpose of formal description of the 6R manipulator, coordinate systems, or *frames*, are attached to the 6 joints and to the endeffector. *Homogenous transformation matrices* $\mathbf{A}_i$ refer the position and orientation of the (i+1)th frame to the position and orientation of the ith frame.

$$\mathbf{A}_i:=\left[\begin{array}{c|c}\mathbf{C}_i & \mathbf{C}_{z,i}\cdot\mathbf{t}_i\\ \hline \mathbf{0} & 1\end{array}\right]\in\mathbb{R}^{4\times4} \qquad \mathbf{A}_i^{-1}:=\left[\begin{array}{c|c}\mathbf{C}_i^{T} & -\mathbf{C}_{x,i}^{T}\cdot\mathbf{t}_i\\ \hline \mathbf{0} & 1\end{array}\right]\in\mathbb{R}^{4\times4} \qquad (2)$$

The rotational parts $\mathbf{C}_i\in\mathbb{R}^{3\times3}$ and $\mathbf{C}_i^T$ are defined as

$$\begin{array}{l}\mathbf{C}_i:=\mathbf{C}_{z,i}\cdot\mathbf{C}_{x,i}\\ \\ \mathbf{C}_i^T:=\mathbf{C}_{x,i}^T\cdot\mathbf{C}_{z,i}^T\end{array} \quad \textit{with}\quad \mathbf{C}_{z,i}:=\mathbf{Rot}(z,\theta_i)=\begin{bmatrix}c_i & -s_i & 0\\ s_i & c_i & 0\\ 0 & 0 & 1\end{bmatrix},\quad \mathbf{C}_{x,i}:=\mathbf{Rot}(x,\alpha_i)=\begin{bmatrix}1 & 0 & 0\\ 0 & \lambda_i & -\mu_i\\ 0 & \mu_i & \lambda_i\end{bmatrix} \qquad (3a)$$

$$c_i:=\cos\theta_i,\quad s_i:=\sin\theta_i \qquad\qquad \lambda_i:=\cos\alpha_i,\quad \mu_i:=\sin\alpha_i$$

and the translational parts $\mathbf{C}_{z,i}\cdot\mathbf{t}_i$ and $-\mathbf{C}_{x,i}^T\cdot\mathbf{t}_i$ with

$$\mathbf{t}_i:=\begin{bmatrix}a_i & 0 & d_i\end{bmatrix}^T \qquad (3b)$$

The matrices $\mathbf{C}_i$, $\mathbf{C}_{z,i}$, $\mathbf{C}_{x,i}$ are orthogonal with a determinant equal to 1. This property will be used. The Denavit-Hartenberg-parameters a_i, α_i, d_i (i=1..6) are constant geometric quantities. The 6R manipulator can now be described by the matrix closure equation

$$\mathbf{A}_{\mathbf{hand}} = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3 \cdot \mathbf{A}_4 \cdot \mathbf{A}_5 \cdot \mathbf{A}_6 \qquad (4)$$

$\mathbf{A}_{\mathbf{hand}}$ defines position and orientation of the endeffector with respect to the base frame. Using $\mathbf{A}_7 = \mathbf{A}_{\mathbf{hand}}^{-1}$ the open-loop 6R manipulator can be related to a closed-loop 7R-mechanism, and vice versa. Therefore the matrix closure equation (4) can also be stated as

$$\mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3 \cdot \mathbf{A}_4 \cdot \mathbf{A}_5 \cdot \mathbf{A}_6 \cdot \mathbf{A}_7 = \mathbf{I} \qquad (5)$$

I defines the identity matrix of appropriate dimensions. The interpretation of an open-loop manipulator as a single-loop mechanism is preferred. $\mathbf{A}_7$ will be defined as

$$\mathbf{A}_{\mathbf{hand}} = \left[\begin{array}{c|c} \mathbf{C}_{\mathbf{hand}} & \mathbf{t}_{\mathbf{hand}} \\ \hline \mathbf{0} & 1 \end{array}\right] = \left[\begin{array}{ccc|c} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \quad \mathbf{A}_7 = \left[\begin{array}{c|c} \mathbf{C}_{z,7} \cdot \mathbf{C}_{x,7} & \mathbf{C}_{z,7} \cdot \mathbf{t}_7 \\ \hline \mathbf{0} & 1 \end{array}\right], \quad \begin{array}{l} \mathbf{C}_{x,7} = \mathbf{I} \\ \mathbf{C}_{z,7} = \mathbf{C}_{\mathbf{hand}}^{\mathrm{T}} \\ \mathbf{t}_7 \;\; = -\mathbf{t}_{\mathbf{hand}} \end{array} \qquad (6)$$

Figure 1 represents equations (4) and (5) as a directed graph. The inversion $\mathbf{A}_i^{-1}$ of the homogenous transformations $\mathbf{A}_i$ is simply the inversion of the i$^{\text{th}}$ arrow.

The goal of inverse kinematics is, given the endeffector description by $\mathbf{A}_{\mathbf{hand}}$ and given the equations (4) or (5), to determine the unknown joint variables θ_i (i=1..6). The problem is highly nonlinear and has up to 16 solutions for the 6R manipulator.

The easier and unique problem, the forward kinematics, needs, given the joint variables θ_i (i=1..6) and equations (4) or (5), the evaluation of $\mathbf{A}_{\mathbf{hand}}$.

3. Solving the Inverse Kinematics of the General 6R Manipulator

The eigenproblem-method is described by the following steps:
[1] Define 14 basic equations: 2 scalar and 4 vector equations.
[2] Replace the vectors in the equations by their matrix representations.
[3] Factor out all the unknowns.
[4] Enlarge the system of equations by 6 more equations.
[5] Eliminate θ_4, and θ_5.
[6] Derive the generalized eigenproblem.
[7] Take $\mathbf{A}_{\mathbf{hand}}$ as input, solve the generalized eigenproblem $\Rightarrow \theta_3^{(k)}, \theta_2^{(k)}, \theta_1^{(k)}$.
[8] Compute the other unknowns $\theta_4^{(k)}$, $\theta_5^{(k)}$, $\theta_6^{(k)}$.

Step 1: The 14 Basic Equations

To solve the inverse kinematics, at most 6 joint variables must be found. For manipulators with special geometric properties there are closed-form solutions, which can be derived by algebraic manipulations of equations (4) or (5) (Paul, Zhang 1986). Most of the industrial robots have special geometries and therefore closed-form solutions.
For manipulators like the general 6R manipulator such solutions don't exist. In these cases the matrix closure equation (5) has to be enhanced. First (5) may be written as

$$\mathbf{A}_3 \cdot \mathbf{A}_4 \cdot \mathbf{A}_5 = \mathbf{A}_2^{-1} \cdot \mathbf{A}_1^{-1} \cdot \mathbf{A}_7^{-1} \cdot \mathbf{A}_6^{-1} \qquad (7)$$

Geometrically this corresponds to cut the chain of links at two different joints (Woernle 88). For instance, the chain could be cut at joint 3 and joint 6. Figure 2 verifies that there are two different subchains to get from joint 3 to joint 6. Clockwise you obtain

$A_3\cdot A_4\cdot A_5$ (the *A*-chain), counterclockwise $A_2^{-1}\cdot A_1^{-1}\cdot A_7^{-1}\cdot A_6^{-1}$ (the *B*-chain).

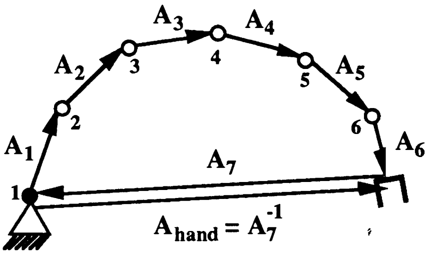


Fig. 1. 6R manipulator with its homogenous transformations A_i

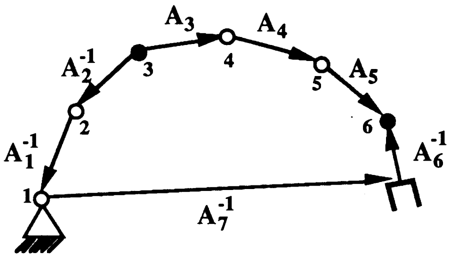


Fig. 2. Geometric interpretation of eqn. $A_3\cdot A_4\cdot A_5 = A_2^{-1}\cdot A_1^{-1}\cdot A_7^{-1}\cdot A_6^{-1}$

Multiplying the matrix equation (7) by $\mathbf{e}_3$ and $\mathbf{e}_4$ on both sides produces

$$\mathbf{z}|_A := A_3A_4A_5\mathbf{e}_3 = A_2^{-1}A_1^{-1}A_7^{-1}A_6^{-1}\mathbf{e}_3 =: \mathbf{z}|_B \tag{8}$$

$$\mathbf{p}|_A := A_3A_4A_5\mathbf{e}_4 = A_2^{-1}A_1^{-1}A_7^{-1}A_6^{-1}\mathbf{e}_4 =: \mathbf{p}|_B \tag{9}$$

Some remark on notation. $\mathbf{p}$ alone specifies the whole vector-equation $\mathbf{p}|_A=\mathbf{p}|_B$. $\mathbf{p}_i$ defines the i$^{\text{th}}$ equation. The same is true for future equations of the form $.....|_A=.....|_B$.
$\mathbf{p}|_A$ and $\mathbf{p}|_B$ are the vectors starting at the origin of the 3rd frame and ending at the origin of the 6th frame and written in components of the 3rd frame. $\mathbf{z}|_A$ and $\mathbf{z}|_B$ describe the z-axis of the 6th frame written in components of the 3rd frame.
The $\mathbf{p}_4$- and $\mathbf{z}_4$-equations define the trivial equations 1=1 und 0=0 and are ignored.
More equations are got by applying scalar- and vector-products to the vectors $\mathbf{p}|_A$, $\mathbf{p}|_B$ and $\mathbf{z}|_A$, $\mathbf{z}|_B$ The nontrivial results are the two scalar and one vector-equations

$$\left.(\mathbf{p}^T\cdot\mathbf{p})\right|_A=\left.(\mathbf{p}^T\cdot\mathbf{p})\right|_B \quad (10) \qquad \left.(\mathbf{p}^T\cdot\mathbf{z})\right|_A=\left.(\mathbf{p}^T\cdot\mathbf{z})\right|_B \quad (11) \qquad \left.(\mathbf{p}\times\mathbf{z})\right|_A=\left.(\mathbf{p}\times\mathbf{z})\right|_B \quad (12)$$

A last relation is obtained by applying scalar- and vector-product as well as addition and multiplication to the scalar $\left.(\mathbf{p}^T\cdot\mathbf{z})\right|_A$, $\left.(\mathbf{p}^T\cdot\mathbf{z})\right|_B$ and the vectors $\mathbf{p}|_A$, $\mathbf{p}|_B$, $\left.(\mathbf{p}\times\mathbf{z})\right|_A$, $\left.(\mathbf{p}\times\mathbf{z})\right|_B$:

$$\left.\left\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}\cdot(\mathbf{p}^T\cdot\mathbf{z})\right\}\right|_A = \left.\left\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}\cdot(\mathbf{p}^T\cdot\mathbf{z})\right\}\right|_B \tag{13}$$

The vector-equation (13) has been published by (Lee, Liang 1988) in the form

$$\left.\left\{(\mathbf{p}^T\cdot\mathbf{p})\cdot\mathbf{z}-2(\mathbf{p}^T\cdot\mathbf{z})\cdot\mathbf{p}\right\}\right|_A = \left.\left\{(\mathbf{p}^T\cdot\mathbf{p})\cdot\mathbf{z}-2(\mathbf{p}^T\cdot\mathbf{z})\cdot\mathbf{p}\right\}\right|_B \tag{14}$$

Using the identity for double vector products,

$$\mathbf{a}\times(\mathbf{b}\times\mathbf{z}) \equiv (\mathbf{a}^T\cdot\mathbf{z})\cdot\mathbf{b}-(\mathbf{a}^T\cdot\mathbf{b})\cdot\mathbf{z} \qquad \forall\mathbf{a},\mathbf{b},\mathbf{z}\in\mathbb{R}^3 \tag{15}$$

it can be shown that the equations are equivalent. But for our purpose equation (13) can be handled easier. At the moment, no more equations have been found which also hold the condition to be linear in the terms $\cos\theta_i$ and $\sin\theta_i$.
The basic equations in a compound form are therefore (see also Lee, Liang 1988)

$$\boxed{\begin{aligned} \mathbf{z}|_A := A_3A_4A_5\mathbf{e}_3 &= A_2^{-1}A_1^{-1}A_7^{-1}A_6^{-1}\mathbf{e}_3 =: \mathbf{z}|_B \quad (16)\\ \mathbf{p}|_A := A_3A_4A_5\mathbf{e}_4 &= A_2^{-1}A_1^{-1}A_7^{-1}A_6^{-1}\mathbf{e}_4 =: \mathbf{p}|_B \quad (17)\end{aligned}}$$

$$(\mathbf{p}^T\mathbf{p})\big|_A = (\mathbf{p}^T\mathbf{p})\big|_B \qquad (18)$$
$$(\mathbf{p}^T\mathbf{z})\big|_A = (\mathbf{p}^T\mathbf{z})\big|_B \qquad (19)$$
$$(\mathbf{p}\times\mathbf{z})\big|_A = (\mathbf{p}\times\mathbf{z})\big|_B \qquad (20)$$
$$\left\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})\right\}\big|_A = \left\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})\right\}\big|_B \qquad (21)$$

STEP 2: REPLACING THE VECTORS BY THEIR ORTHOGONAL MATRIXDEFINITIONS

For the evaluation of the equations (16-21), first, the formal definitions (2) for the homogenous matrices $\mathbf{A}_i$ should be applied and not their component definitions (3). Using (2) and the relations (22-24), it is easy to simplify the terms in the basic equations (16-21).

$\forall \mathbf{Q},\mathbf{R}\in\mathbb{R}^{3\times3}$: orthogonal $\wedge$ $\det\mathbf{Q}=\det\mathbf{R}=1$, $\forall \mathbf{a},\mathbf{b},\mathbf{z}\in\mathbb{R}^3$:

$$(\mathbf{Qa})^T\cdot(\mathbf{Qb})=\mathbf{a}^T\mathbf{b} \qquad (22)$$
$$(\mathbf{Qa})\times(\mathbf{Qb})=\mathbf{Q}(\mathbf{a}\times\mathbf{b}) \qquad (23)$$

$$\begin{aligned}&(\mathbf{Qb})\times[\mathbf{a}\times(\mathbf{QRz})]+(\mathbf{Qb})\cdot\left[\mathbf{a}^T\cdot(\mathbf{QRz})\right] \qquad (24)\\ &+\mathbf{a}\times[(\mathbf{Qb})\times(\mathbf{QRz})]+\mathbf{a}\cdot\left[(\mathbf{Qb})^T\cdot(\mathbf{QRz})\right]\\ &=2\left\{\mathbf{a}\times\{\mathbf{Q}\cdot[\mathbf{b}\times\mathbf{Rz}]\}+\mathbf{a}\cdot\left[\mathbf{b}^T\cdot\mathbf{Rz}\right]\right\}\end{aligned}$$

On the other hand, begining with the component definitions (3) for the $\mathbf{A}_i$-matrices, trigonometric identities such as

$$\cos^2\beta+\sin^2\beta=1$$

frequently occur and must be eliminated from the equations. Especially the simplification of the vector-equation $\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})$ gets very expensive in terms of computer time (some days on a SUN 4-390 with Mathematica™) and computer memory.

The relations (22-23) express the fact that the scalar- and vector-products are invariant with respect to orthogonal transformations. The last relation will only be used for the evaluation of $\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})$ and reveals the symmetry of this equation. (24) is easily proved by application of (22-23) and the identity for double vector products (15). This way the orthogonal properties of parts of the homogenous matrices $\mathbf{A}_i$ are used.

The fully worked out equations in orthogonal matrix form are found in Appendix A.

DISCUSSION OF THE EQUATIONS

An analysis of the basic equations (16-21) in orthogonal matrix form reveals some interesting properties:

a) The endeffector-description $\mathbf{A}_{\mathbf{hand}}$ appears only on the right hand side.

b) All the equations are independent of θ_6.

c) $\mathbf{z}_3$, $\mathbf{p}_3$, $\mathbf{p}^T\cdot\mathbf{z}$, $\mathbf{p}^T\cdot\mathbf{p}$, $(\mathbf{p}\times\mathbf{z})_3$ and $\left\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})\right\}_3$ are independent of θ_3.

d) $\theta_1,\theta_2,\theta_4,\theta_5$ appear only as the linear variables $\cos\theta_i$ and $\sin\theta_i$ with $i\in\{1,2,4,5\}$, i.e. never as $\cos^2\theta_i$, $\sin^2\theta_i$ or $\cos\theta_i\cdot\sin\theta_i$.

e) $\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}^T\mathbf{p}$, $\mathbf{p}^T\mathbf{z}$, $(\mathbf{p}\times\mathbf{z})_1$, $(\mathbf{p}\times\mathbf{z})_2$, $\left\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})\right\}_1$, $\left\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})\right\}_2$ are linear in the variable $\tan(\theta_3/2)$.

The proof of these properties is very easy, as long the basic equations in orthogonal matrix form (Appendix A) are used, and is worked out in Appendix B. Other proofs may be found in (Raghaven, Roth 1990) and (Lee 1990).

STEP 3: FACTORING OUT THE UNKNOWNS

In order to prepare the later elimination of some unknowns out of the basic equations, all the unknown terms must be factored out.
The angle θ_3 will be replaced by its half tangens substitution

$$\cos\theta_i = \frac{1-x_i^2}{1+x_i^2}\ ,\quad \sin\theta_i = \frac{2x_i}{1+x_i^2}\ ,\quad x_i := \tan\frac{\theta_i}{2} \qquad (25)$$

respectively better by the matrix-equivâlent

$$\mathbf{X}_3^- \cdot \mathbf{C}_{z,3} = \mathbf{X}_3^+ \qquad \text{where } \mathbf{X}_i^- = \begin{bmatrix} -x_i & 1 & 0 \\ 1 & x_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{X}_i^+ = \begin{bmatrix} x_i & 1 & 0 \\ 1 & -x_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (26)$$

The right hand side terms $\cos\theta_1$, $\sin\theta_1$, $\cos\theta_2$ and $\sin\theta_2$ are also replaced by their half tangens substitutions. The basic equations now can be formulated as

$$\hat{\mathbf{U}}\cdot\left[\begin{array}{c} x_3\cdot\hat{\xi} \\ \hline \hat{\xi} \end{array}\right]\cdot(1+x_1^2)(1+x_2^2) = \hat{\mathbf{V}}\cdot\left[\begin{array}{c} x_3\cdot\hat{\rho} \\ \hline \hat{\rho} \end{array}\right] \qquad \hat{\mathbf{U}},\hat{\mathbf{V}}\in\mathbb{R}^{14\times18};\ \hat{\xi},\hat{\rho}\in\mathbb{C}^9 \qquad (27)$$

$$\hat{\xi} = [s_4s_5,\ s_4c_5,\ c_4s_5,\ c_4c_5,\ s_4,\ c_4,\ s_5,\ c_5,\ 1]^{\mathrm{T}},\ \hat{\rho} = [x_1^2x_2^2,\ x_1^2x_2,\ x_1^2,\ x_1x_2^2,\ x_1x_2,\ x_1,\ x_2^2,\ x_2,\ 1]^{\mathrm{T}}$$

For the determination of all the elements in the matrices $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ without formula-manipulation-programs, it is usefull to apply the following matrixdefinitions:

$$\begin{array}{lll} \mathbf{C}_{z,i} = \mathbf{D}_u + s_i\mathbf{D}_s + c_i\mathbf{D}_c & [i=4,5] & \mathbf{D}_u = \begin{bmatrix} 0&0&0 \\ 0&0&0 \\ 0&0&1 \end{bmatrix},\ \mathbf{D}_s = \begin{bmatrix} 0&-1&0 \\ 1&0&0 \\ 0&0&0 \end{bmatrix},\ \mathbf{D}_c = \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&0 \end{bmatrix} \\ \mathbf{X}_i^+ = \mathbf{D}_p + x_i\mathbf{D}_x;\ \mathbf{X}_i^- = \mathbf{D}_p - x_i\mathbf{D}_x & [i=3] & \\ \mathbf{C}_{z,i} = \dfrac{\{\mathbf{I} + x_i^2(\mathbf{D}_u - \mathbf{D}_c) + 2x_i\mathbf{D}_s\}}{1+x_i^2} & [i=1,2] & \mathbf{D}_p = \begin{bmatrix} 0&1&0 \\ 1&0&0 \\ 0&0&1 \end{bmatrix},\ \mathbf{D}_x = \begin{bmatrix} 1&0&0 \\ 0&-1&0 \\ 0&0&0 \end{bmatrix} \end{array} \qquad (28)$$

After multiplication with $(1+x_1^2)(1+x_2^2)$ to cancel denominators, the elements of the matrices $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ are easily extracted.

STEP 4: 6 MORE EQUATIONS

Multiplying the 6 scalar equations $\mathbf{p}_3$, $\mathbf{z}_3$, $\mathbf{p}^{\mathrm{T}}\mathbf{z}$, $\mathbf{p}^{\mathrm{T}}\mathbf{p}$, $(\mathbf{p}\times\mathbf{z})_3$, $\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}\cdot(\mathbf{p}^{\mathrm{T}}\cdot\mathbf{z})\}_3$, which are independant of θ_3, with $x_3 := \tan(\theta_3/2)$, and then adding one of the other 5 equations, results in 6 new equations:

$$\begin{array}{llll} [x_3\mathbf{z}_3 + \mathbf{p}_3]\big|_A = [x_3\mathbf{z}_3 + \mathbf{p}_3]\big|_B & (29a) & [x_3\mathbf{p}^{\mathrm{T}}\mathbf{z} + \mathbf{p}^{\mathrm{T}}\mathbf{p}]\big|_A = [x_3\mathbf{p}^{\mathrm{T}}\mathbf{z} + \mathbf{p}^{\mathrm{T}}\mathbf{p}]\big|_B & (29d) \\ [x_3\mathbf{p}_3 + \mathbf{z}_3]\big|_A = [x_3\mathbf{p}_3 + \mathbf{z}_3]\big|_B & (29b) & [x_3(\mathbf{p}\times\mathbf{z})_3 + \mathbf{z}_3]\big|_A = [x_3(\mathbf{p}\times\mathbf{z})_3 + \mathbf{z}_3]\big|_B & (29e) \\ [x_3\mathbf{p}^{\mathrm{T}}\mathbf{p} + \mathbf{p}^{\mathrm{T}}\mathbf{z}]\big|_A = [x_3\mathbf{p}^{\mathrm{T}}\mathbf{p} + \mathbf{p}^{\mathrm{T}}\mathbf{z}]\big|_B & (29c) & & \end{array}$$

$$\left[x_3\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^{\mathrm{T}}\mathbf{z})\}_3 + \mathbf{p}_3\right]\Big|_A = \left[x_3\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^{\mathrm{T}}\mathbf{z})\}_3 + \mathbf{p}_3\right]\Big|_B \qquad (29f)$$

The addition of original equations is necessary to get later a full rank eigenproblem.
An inspection of the combined system of equations (27,29) shows that the terms $\{x_3x_1^2x_2^2,\ x_3x_1^2,\ x_3x_2^2,\ x_3,\ x_1^2x_2^2,\ x_1^2,\ x_2^2,\ 1\}$ are found on both sides of the equations. Therefore they are collected on the right hand side. Thereby the vector $\hat{\xi}$ is shortened by the elements $\{x_3, 1\}$ and the 20 equations (27,29) can be written as

$$\mathbf{U}\cdot\xi\cdot(1+x_1^2)(1+x_2^2)=\mathbf{V}\cdot\begin{bmatrix}x_3\cdot\hat{\rho}\\ \hat{\rho}\end{bmatrix}\qquad \mathbf{U}\in\mathbb{R}^{20\times16},\ \mathbf{V}\in\mathbb{R}^{20\times18},\ \xi\in\mathbb{C}^{16},\ \hat{\rho}\in\mathbb{C}^{9}\qquad(30)$$

$$\xi=\left[x_3s_4s_5,x_3s_4c_5,x_3c_4s_5,x_3c_4c_5,x_3s_4,x_3c_4,x_3s_5,x_3c_5,\quad s_4s_5,s_4c_5,c_4s_5,c_4c_5,s_4,c_4,s_5,c_5\right]^{\mathrm{T}}$$

$$\hat{\rho}=\left[x_1^2x_2^2,\ x_1^2x_2,\ x_1^2,\quad x_1x_2^2,\ x_1x_2,\ x_1,\quad x_2^2,\ x_2,\ 1\right]^{\mathrm{T}}$$

STEP 5: ELIMINATION OF θ_4 AND θ_5

There are 16 different terms, which contain θ_4 and θ_5, and they only appear in the vector ξ. 16 linear independant equations out of the 20 equations (30) are used to eliminate the vector $\xi\cdot(1+x_1^2)(1+x_2^2)$ in the other 4 equations: steps (31) through (33):

$$\begin{bmatrix}\mathbf{U}_{16}\\ \mathbf{U}_4\end{bmatrix}\cdot\xi\cdot(1+x_1^2)(1+x_2^2)=\begin{bmatrix}\mathbf{V}_{16}\\ \mathbf{V}_4\end{bmatrix}\cdot\begin{bmatrix}x_3\cdot\hat{\rho}\\ \hat{\rho}\end{bmatrix}\qquad \begin{matrix}\mathbf{U}_4\in\mathbb{R}^{4\times16};\ \mathbf{U}_{16}\in\mathbb{R}^{16\times16}\\ \mathbf{V}_4\in\mathbb{R}^{4\times18};\ \mathbf{V}_{16}\in\mathbb{R}^{16\times18}\end{matrix}\qquad(31)$$

$$\xi\cdot(1+x_1^2)(1+x_2^2)=\mathbf{U}_{16}^{-1}\mathbf{V}_{16}\cdot\begin{bmatrix}x_3\cdot\hat{\rho}\\ \hat{\rho}\end{bmatrix}\qquad(32)$$

$$\mathbf{U}_4\cdot\mathbf{U}_{16}^{-1}\cdot\mathbf{V}_{16}\cdot\begin{bmatrix}x_3\cdot\hat{\rho}\\ \hat{\rho}\end{bmatrix}=\mathbf{V}_4\cdot\begin{bmatrix}x_3\cdot\hat{\rho}\\ \hat{\rho}\end{bmatrix}\qquad(33)$$

STEP 6: 12 MORE EQUATIONS AND THE GENERALIZED EIGENPROBLEM

Now we can derive the generalized eigenproblem. Equations (33) can also be written as

$$\mathbf{U}_4\cdot\mathbf{U}_{16}^{-1}\cdot\left[{}^{x}\mathbf{V}_{16},\ {}^{1}\mathbf{V}_{16}\right]\cdot\begin{bmatrix}x_3\hat{\rho}\\ \hat{\rho}\end{bmatrix}=\left[{}^{x}\mathbf{V}_4,\ {}^{1}\mathbf{V}_4\right]\cdot\begin{bmatrix}x_3\hat{\rho}\\ \hat{\rho}\end{bmatrix}\qquad \begin{matrix}{}^{x}\mathbf{V}_{16},\ {}^{1}\mathbf{V}_{16}\in\mathbb{R}^{16\times9}\\ {}^{x}\mathbf{V}_4,\ {}^{1}\mathbf{V}_4\in\mathbb{R}^{4\times9}\end{matrix}$$

Expanding this yields

$$\underbrace{\{(\mathbf{U}_4\mathbf{U}_{16}^{-1})\cdot{}^{1}\mathbf{V}_{16}-{}^{1}\mathbf{V}_4\}}_{=:\hat{\mathbf{G}}}\cdot\hat{\rho}+x_3\underbrace{\{(\mathbf{U}_4\mathbf{U}_{16}^{-1})\cdot{}^{x}\mathbf{V}_{16}-{}^{x}\mathbf{V}_4\}}_{=:\hat{\mathbf{H}}}\cdot\hat{\rho}=0\qquad \begin{matrix}\hat{\mathbf{G}},\hat{\mathbf{H}}\in\mathbb{R}^{4\times9}\\ \hat{\rho}\in\mathbb{C}^9;\ x_3\in\mathbb{C}\end{matrix}\qquad(34)$$

The essential trick is now to multiply these 4 multivariate polynomial (!) equations by x_1x_2, x_2 and x_1 respectively to get 12 more independant equations. Using for the equations (34) the column-representation

$$\underbrace{[\mathbf{abcdefghj}]}_{\hat{\mathbf{G}}}\cdot\hat{\rho}+x_3\underbrace{[\mathbf{klmnpqrst}]}_{\hat{\mathbf{H}}}\cdot\hat{\rho}=0\qquad \begin{matrix}\mathbf{a,b,c,d,e,f,g,h,j,k,l,m,n,p,q,r,s,t}\in\mathbb{R}^4\\ \hat{\rho}\in\mathbb{C}^9\end{matrix}\qquad(35)$$

$$\hat{\rho}=\left[x_1^2x_2^2,\ x_1^2x_2,\ x_1^2,\quad x_1x_2^2,\ x_1x_2,\ x_1,\quad x_2^2,\ x_2,\ 1\right]^{\mathrm{T}}$$

then the 16 final equations have the form

$$\underbrace{\left[\begin{array}{cccc:cccc:cccc:cccc}\mathbf{a}&\mathbf{b}&\mathbf{c}&\mathbf{0}&\mathbf{d}&\mathbf{e}&\mathbf{f}&\mathbf{0}&\mathbf{g}&\mathbf{h}&\mathbf{j}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \hdashline\mathbf{0}&\mathbf{a}&\mathbf{b}&\mathbf{c}&\mathbf{0}&\mathbf{d}&\mathbf{e}&\mathbf{f}&\mathbf{0}&\mathbf{g}&\mathbf{h}&\mathbf{j}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \hdashline\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{a}&\mathbf{b}&\mathbf{c}&\mathbf{0}&\mathbf{d}&\mathbf{e}&\mathbf{f}&\mathbf{0}&\mathbf{g}&\mathbf{h}&\mathbf{j}&\mathbf{0}\\ \hdashline\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{a}&\mathbf{b}&\mathbf{c}&\mathbf{0}&\mathbf{d}&\mathbf{e}&\mathbf{f}&\mathbf{0}&\mathbf{g}&\mathbf{h}&\mathbf{j}\end{array}\right]}_{=:\mathbf{G}}\cdot\rho+x_3\underbrace{\left[\begin{array}{cccc:cccc:cccc:cccc}\mathbf{k}&\mathbf{l}&\mathbf{m}&\mathbf{0}&\mathbf{n}&\mathbf{p}&\mathbf{q}&\mathbf{0}&\mathbf{r}&\mathbf{s}&\mathbf{t}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \hdashline\mathbf{0}&\mathbf{k}&\mathbf{l}&\mathbf{m}&\mathbf{0}&\mathbf{n}&\mathbf{p}&\mathbf{q}&\mathbf{0}&\mathbf{r}&\mathbf{s}&\mathbf{t}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \hdashline\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{k}&\mathbf{l}&\mathbf{m}&\mathbf{0}&\mathbf{n}&\mathbf{p}&\mathbf{q}&\mathbf{0}&\mathbf{r}&\mathbf{s}&\mathbf{t}&\mathbf{0}\\ \hdashline\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{k}&\mathbf{l}&\mathbf{m}&\mathbf{0}&\mathbf{n}&\mathbf{p}&\mathbf{q}&\mathbf{0}&\mathbf{r}&\mathbf{s}&\mathbf{t}\end{array}\right]}_{=:\mathbf{H}}\cdot\rho=0\qquad(36)$$

$$\rho=\left[x_1^3x_2^3,x_1^3x_2^2,x_1^3x_2,x_1^3,\ \vdots\ x_1^2x_2^3,x_1^2x_2^2,x_1^2x_2,x_1^2,\ \vdots\ x_1x_2^3,x_1x_2^2,x_1x_2,x_1,\ \vdots\ x_2^3,x_2^2,x_2,1\right]^T\in\mathbb{C}^{16}$$

But that's nothing else than the *Generalized Eigenproblem*

$$\boxed{\mathbf{G}\cdot\rho+x_3\,\mathbf{H}\cdot\rho=\mathbf{0} \qquad \mathbf{G},\mathbf{H}\in\mathbb{R}^{16\times16};\ \rho\in\mathbb{C}^{16};\ x_3\in\mathbb{C}} \tag{37}$$

In case the matrix **H** is nonsingular, the generalized eigenproblem may be reduced to the *Standard Eigenproblem*

$$\mathbf{H}^{-1}\cdot\mathbf{G}\cdot\rho+x_3\,\rho=\mathbf{0} \tag{38}$$

But if the matrix **H** is ill-conditioned, the original generalized eigenproblem (37) should be used (Golub, Van Loan 1989, p. 395). Otherwise the results won't be accurate.
Generating new equations by multiplication of existing equations with already appearing terms, is a standard technique to solve multivariate polynomial systems of equations (Cox, Little, O'Shea 1991).

Step 7: Solving the Eigenproblem

The eigenproblem can be solved by standard numerical methods, as for instance the QR-algorithm for the standard eigenproblem or the QZ-algorithm for the generalized eigenproblem (Golub, Van Loan 1989).
An alternative approach to determine the generalized eigenvalues would be to calculate the roots of the characteristic polynomial det(G+x_3 H)=0. Unfortunately, the accuracy of the roots is very sensitive to the unavoidable roundoff errors in the polynomial coefficients (Wilkinson 1969). But, because polynomials and their roots have a unique mapping, the polynomials appearing in (Raghaven, Roth 1990) and (Lee 1990) are identical to the characteristic polynomials det(G+x_3 H)=0 respectively det(G+x_1 H)=0.

The generalized eigenproblem $\mathbf{G}\cdot\rho+x_3\,\mathbf{H}\cdot\rho=\mathbf{0}$ has always 16 eigenvalues $x_3^{(k)}\in\mathbb{C}$ and eigenvectors $\rho^{(k)}\in\mathbb{C}^{16}$. Whereas the eigenvalues $x_3^{(k)}$ are uniquely defined, the eigenvectors $\rho^{(k)}$ of different eigenvalues $x_3^{(k)}$ are only uniquely defined with respect to the direction, but neither with respect to sign nor length. In our context, this problem is fortunately solved by noting in (36) that the last element $\rho_{16}^{(k)}$ in the kth eigenvector $\rho^{(k)}$ should equal to 1. The following statement reflects this

$$\rho^{(k)}:=\rho^{(k)}/\rho_{16}^{(k)} \qquad k=1..16 \tag{39}$$

Determination of $\theta_3^{(k)},\theta_2^{(k)},\theta_1^{(k)}$

The eigenvectors $\rho^{(k)}$ contain the elements $x_2^{(k)}$ and $x_1^{(k)}$. Therefore the joint variables $\theta_3^{(k)},\theta_2^{(k)},\theta_1^{(k)}$ can be determined by applying the relation $x_i=\tan(\theta_i/2)$ to each eigenvalue $x_3^{(k)}$ and eigenvector $\rho^{(k)}$:

$$\theta_i^{(k)}=2\arctan x_i^{(k)} \qquad i=1,2,3;\ k=1..16 \tag{40}$$

Step 8: Computation of $\theta_4^{(k)},\theta_5^{(k)},\theta_6^{(k)}$

The terms $c_4^{(k)}:=\cos\theta_4^{(k)}$, $s_4^{(k)}:=\sin\theta_4^{(k)}$, $c_5^{(k)}:=\cos\theta_5^{(k)}$ and $s_5^{(k)}:=\sin\theta_5^{(k)}$ appear in the vector ξ. Substituting the numerical values into the 4 appropriate equations in (32) we get

$$\theta_i^{(k)}=\mathrm{Arctan}(c_i^{(k)},s_i^{(k)}) \qquad i=4,5;\ k=1..16 \tag{41}$$

And last, the joint variables $\theta_6^{(k)}$ can be determined out of $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6\mathbf{A}_7=\mathbf{I}$.

Discussion of the Solutions

Each solution with a real eigenvalue represents a realizable manipulator configuration. In case all the solutions are imaginary, the endeffector cannot reach the specified pose $\mathbf{A}_{\mathbf{hand}}$. As soon as two joint variable solutions $\theta_j^{(l)}$ and $\theta_j^{(k)}$ have the same real value, the manipulator is at a singular position (Lee 1990, p. 95).

4. Prismatic Joints, Special Manipulators

Only prismatic and revolute joints will be considered. Other lower-pair joints may be modelled by a combination of these two joint types.

The eigenproblem-method can be extended to prismatic joints using the same arguments as can be found in (Raghaven, Roth 1990) for the polynomial method. The reason are the substantially common basic equations (16-21).

Replacing in a general 6R manipulator the i^{th} revolute (R) joint by a prismatic (P) joint does not change the dimension of the corresponding generalized eigenproblem, since the $\cos\theta_i$ and $\sin\theta_i$ are simply replaced by d_i and d_i^2. On the other hand, if there are more than one prismatic joint, the dimension n of the generalized eigenproblem reduces to n=8, 4 or even 2. In this context the following properties are noteworthy

f) $\mathbf{z}$ doesn't contain any d_i.

g) $\mathbf{p}$, $\mathbf{p}^T\mathbf{z}$ und $\mathbf{p}\times\mathbf{z}$ contain d_i, but neither d_i^2 nor d_id_k.

h) $\mathbf{p}^T\mathbf{p}$ and $\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}(\mathbf{p}^T\mathbf{z})$ contain d_i, d_i^2 and d_id_k, but neither $d_i^2d_k$ nor $d_i^2d_k^2$.

Therefore the equation system (30) consists of less unknown terms than in case of a general 6R manipulator. As a consequence, less equations are needed to eliminate unknowns and derive a generalized eigenproblem. Appendix C shows the application of these ideas to a RPRRPR-manipulator.

Manipulators with special geometry can also be reduced to a variant of equations (31) with less unknown terms, and thus to an eigenproblem with a dimension smaller than 16.

5. Example: General 6R-Manipulator

The following computations have been done with 15-digit arithmetic.

Given a general 6R manipulator, its Denavit-Hartenberg-parameters and the joint variables $\theta_1,\theta_2,\theta_3$, $\theta_4,\theta_5,\theta_6$ of **table 1** (Raghaven, Roth 90):

link	a_i	α_i [deg]	d_i	θ_i [deg]
1	0.8	20	0.9	14
2	1.2	31	3.7	29.7
3	0.33	45	1.0	-45
4	1.8	81	0.5	71
5	0.6	12	2.1	-63
6	2.2	100	0.63	10

Table 1

	solution k=15	solution k=16
$\theta_1^{(k)}$	13.1097107766116	14.0000000000008
$\theta_2^{(k)}$	50.9925511934656	29.7000000000001
$\theta_3^{(k)}$	-72.0441108063809	-45.0000000000015
$\theta_4^{(k)}$	72.0649090215457	70.9999999999993
$\theta_5^{(k)}$	-7.19625925238062	-62.9999999999977
$\theta_6^{(k)}$	-37.8522931900531	10.0000000000018
Error	$1.83047*10^{-13}$	$1.63307*10^{-13}$

Table 2

Solving the forward kinematics we obtain position and orientation of the endeffector:

$$\mathbf{A}_{\mathbf{hand}}=\begin{bmatrix} 0.354937475307970 & 0.461639573991742 & -0.812962663562557 & 6.82151837150213 \\ 0.876709605247149 & 0.137616185817978 & 0.460914366741046 & 1.4614670400283 \\ 0.324653132880913 & -0.876327957516839 & -0.355878707125017 & 5.36950521368663 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the eigenproblem-method (standard eigenproblem variant) to solve again the inverse kinematics yields for $\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$ the only two real solutions of **table 2**. The error of the solution has been determined by solving the forward kinematics for each inverse kinematics solution. Then the matrix-2-norm has been applied to the difference of the exact and the approximated value for $\mathbf{A}_{\mathbf{hand}}$ (Golub, Van Loan p. 58).

6. Conclusions and Future Work

This paper presented a new method to solve the inverse kinematics of nonredundant manipulators with prismatic and revolute joints and related single-loop mechanisms. The key idea is to reduce the inverse kinematics problem to a generalized eigenproblem $\mathbf{G}\cdot\mathbf{p}+\lambda\,\mathbf{H}\cdot\mathbf{p}=\mathbf{0}$ where $\mathbf{G},\mathbf{H}\in\mathbb{R}^{n\times n}$, $\mathbf{p}\in\mathbb{C}^n$, $\lambda\in\mathbb{C}$, $n\in\{2,4,8,16\}$. The *eigenproblem-method* combines the advantages to determine all the configurations and to deliver numerically accurate results. It has been shown, by using simple orthogonal-matrix-relations, that the necessary basic equations may be derived without using formula-manipulation-programs.

Future work will be done on the optimization and on the automatization of the eigenproblem-method with a partial implementation on a parallel processor system.

7. References

Cox, D., Little, J., O'Shea, D. 1991. *Ideals, varieties, and algorithms*. New York - Berlin - Heidelberg: Springer.

Craig, J.J. 1989. *Introduction to Robotics: Mechanics and Control*. 2nd ed. Reading, MA:.Addison-Wesley.

Denavit, J., Hartenberg, R.S. 1955. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *ASME Journal of Applied Mechanics* 22(2):215-221.

Golub, G.H., Van Loan, C.F. 1989. *Matrix Computations*. 2nd ed. Baltimore - London: The John Hopkings University Press, pp. 331-408.

Lee, H.Y. 1990. *Ein Verfahren zur vollständigen Lösung der Rückwärtstransformation für Industrieroboter mit allgemeiner Geometrie*. Ph.D. Thesis, Universität Gesamthochschule Duisburg.

Lee, H.Y., Liang, C.G. 1988. Displacement Analysis of the General Spatial 7-Link 7R Mechanism. *Mechanism and Machine Theory 23(3):219-226.*

Raghaven, M., Roth, B. 1990. Inverse Kinematics of the General 6R Manipulator and Related Linkages. Mechanism synthesis and analysis: *1990 ASME Design Technical Conferences - 21st Biennial Mechanisms Conference*, Chicago, Illinois, September 16-19. pp. 59-65.

Paul, R.P., Zhang, H. 1986. Computationally Efficient Kinematics for Manipulators with Spherical Wrists Based on the Homogenous Transformation Representation. *The International Journal of Robotics Research* 5(2):32-44.

Paul, R.P. 1981. *Robot Manipulators: Mathematics, Programming, and Control*. Cambridge, MA: MIT Press.

Tsai, L.W., Morgan, A. 1985. Solving the Kinematics of the Most General Six- and Five-Degree-of-Freedom Manipulators by Continuation Methods. *ASME Journal of Mechanisms, Transmissions and Automation in Design* 107:189-200.

Uicker, J.J., Denavit, J., Hartenberg, R.S. 1964. An Iterative Method for the Displacement Analysis of Spatial Mechanisms. *ASME J. of Applied Mechanics* 31:309-314.

Wilkinson, J.H. 1969. *Rundungsfehler*. Berlin: Springer.

Woernle, C. 1988. *Ein systematisches Verfahren zur Aufstellung der geometrischen Schliessbedingungen in kinematischen Schleifen mit Anwendung bei der Rückwärtstransformation für Industrieroboter*. Ph.D. Thesis, Universität Stuttgart.

Appendix A: The Basic equations in orthogonal-matrix-form

With substitutions (42), the definitions (2), and the application of the orthogonal-matrix-

$$S_1 := C_{x,7} \cdot C_{z,1} \equiv C_{z,1}, \qquad S_i := C_{x,i-1} \cdot C_{z,i} \quad \text{for } i=2..7 \qquad (42)$$

relations (22-24) to the basic equations (16-21), the following equations in orthogonal matrix form are obtained

$$z|_A = C_{z,3}S_4S_5S_6e_3 = C_{x,2}^T S_2^T S_1^T S_7^T e_3 = z|_B$$

$$p|_A = \begin{pmatrix} C_{z,3}t_3 \\ +C_{z,3}S_4t_4 \\ +C_{z,3}S_4S_5t_5 \end{pmatrix} = \begin{pmatrix} -C_{x,2}^T t_2 \\ -C_{x,2}^T S_2^T t_1 \\ -C_{x,2}^T S_2^T S_1^T t_7 \\ -C_{x,2}^T S_2^T S_1^T S_7^T t_6 \end{pmatrix} = p|_B$$

$$\left(p^T p\right)\Big|_A = \begin{pmatrix} t_3^T t_3 \\ +2t_3^T S_4 t_4 \\ +2t_3^T S_4 S_5 t_5 \\ \\ + \quad t_4^T t_4 \\ + \quad 2t_4^T S_5 t_5 \\ \\ + \quad t_5^T t_5 \\ \\ \end{pmatrix} = \begin{pmatrix} t_2^T t_2 \quad + \\ 2t_2^T S_2^T t_1 \quad + \\ 2t_2^T S_2^T S_1^T t_7 \quad + \\ 2t_2^T S_2^T S_1^T S_7^T t_6 + \\ t_1^T t_1 \quad + \\ 2t_1^T S_1^T t_7 \quad + \\ 2t_1^T S_1^T S_7^T t_6 + \\ t_7^T t_7 \quad + \\ 2t_7^T S_7^T t_6 + \\ t_6^T t_6 + \end{pmatrix} = \left(p^T p\right)\Big|_B$$

$$\left(p^T z\right)\Big|_A = \begin{pmatrix} t_3^T S_4 S_5 S_6 e_3 \\ + \; t_4^T S_5 S_6 e_3 \\ + \quad t_5^T S_6 e_3 \\ \\ \end{pmatrix} = \begin{pmatrix} -t_2^T S_2^T S_1^T S_7^T e_3 \\ -t_1^T S_1^T S_7^T e_3 \\ -t_7^T S_7^T e_3 \\ -t_6^T e_3 \end{pmatrix} = \left(p^T z\right)\Big|_B$$

$$(p \times z)\big|_A = \begin{pmatrix} C_{z,3}(t_3 \times S_4S_5S_6e_3) \\ +C_{z,3}S_4(t_4 \times S_5S_6e_3) \\ +C_{z,3}S_4S_5(t_5 \times S_6e_3) \end{pmatrix} = \begin{pmatrix} C_{z,3}(t_3 \times S_4S_5S_6e_3) \\ +C_{z,3}S_4(t_4 \times S_5S_6e_3) \\ +C_{z,3}S_4S_5(t_5 \times S_6e_3) \end{pmatrix} = (p \times z)\big|_B$$

$$\left\{p\times(p\times z)+p\cdot\left(p^T\cdot z\right)\right\}\Big|_A =$$

$$\begin{pmatrix} C_{z,3}\{t_3\times(t_3\times S_4S_5S_6e_3)+t_3(t_3^T S_4S_5S_6e_3)\}+ \\ 2C_{z,3}\{t_3\times S_4(t_4\times S_5S_6e_3)+t_3 \quad (t_4^T S_5S_6e_3)\}+ \\ 2C_{z,3}\{t_3\times S_4S_5(t_5\times S_6e_3)+t_3 \quad (t_5^T S_6e_3)\}+ \\ \\ C_{z,3}S_4\{t_4\times(t_4\times S_5S_6e_3)+ \quad t_4(t_4^T S_5S_6e_3)\}+ \\ 2C_{z,3}S_4\{t_4\times S_5(t_5\times S_6e_3)+ \quad t_4 \quad (t_5^T S_6e_3)\}+ \\ \\ C_{z,3}S_4S_5\{t_5\times(t_5\times S_6e_3)+ \quad t_5(t_5^T S_6e_3)\} \\ \\ \end{pmatrix} = \begin{pmatrix} C_{x,2}^T\{t_2\times(t_2\times S_2^T S_1^T S_7^T e_3)+t_2(t_2^T S_2^T S_1^T S_7^T e_3)\}+ \\ 2C_{x,2}^T\{t_2\times S_2^T(t_1\times S_1^T S_7^T e_3)+t_2 \quad (t_1^T S_1^T S_7^T e_3)\}+ \\ 2C_{x,2}^T\{t_2\times S_2^T S_1^T(t_7\times S_7^T e_3)+t_2 \quad (t_7^T S_7^T e_3)\}+ \\ 2C_{x,2}^T\{t_2\times S_2^T S_1^T S_7^T(t_6\times e_3)+t_2 \quad (t_6^T e_3)\}+ \\ C_{x,2}^T S_2^T\{t_1\times(t_1\times S_1^T S_7^T e_3)+ \quad t_1(t_1^T S_1^T S_7^T e_3)\}+ \\ 2C_{x,2}^T S_2^T\{t_1\times S_1^T(t_7\times S_7^T e_3)+ \quad t_1 \quad (t_7^T S_7^T e_3)\}+ \\ 2C_{x,2}^T S_2^T\{t_1\times S_1^T S_7^T(t_6\times e_3)+ \quad t_1 \quad (t_6^T e_3)\}+ \\ C_{x,2}^T S_2^T S_1^T\{t_7\times(t_7\times S_7^T e_3)+ \quad t_7(t_7^T S_7^T e_3)\}+ \\ 2C_{x,2}^T S_2^T S_1^T\{t_7\times S_7^T(t_6\times e_3)+ \quad t_7 \quad (t_6^T e_3)\}+ \\ C_{x,2}^T S_2^T S_1^T S_7^T\{t_6\times(t_6\times e_3)+ \quad t_6(t_6^T e_3)\} \end{pmatrix}$$

$$=\left\{p\times(p\times z)+p\cdot\left(p^T\cdot z\right)\right\}\Big|_B$$

The regular structure of the equations $\mathbf{p}^T\mathbf{p}$, $\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}\cdot(\mathbf{p}^T\cdot\mathbf{z})$ should be noted, especially the $\mathbf{t}_i$-terms. E.g. the structure is the same you get by expanding $(a+b+c)^2$:

$$(a+b+c)^2=(a^2+2ab+2ac)+(b^2+2bc)+(c^2)$$

APPENDIX B: PROOF OF PROPOSITIONS a) - e)

The easiest way to prove the propositions a) through e) is to use the basic equations in orthogonal-matrix-form (see Appendix A).

Proof a): $\mathbf{S}_7$ and $\mathbf{t}_7$ appear only on the right hand side.

Proof b): θ_6 always appears as $\mathbf{S}_6\cdot\mathbf{e}_3$. But using the substitutions (42), this can be reduced to $\mathbf{S}_6\cdot\mathbf{e}_3=\mathbf{C}_{x,5}\cdot\mathbf{C}_{z,6}\cdot\mathbf{e}_3=\mathbf{C}_{x,5}\cdot\mathbf{e}_3$, which proves the independance of θ_6.

Proof c): In none of the equations $\mathbf{p}^T\mathbf{z}$ and $\mathbf{p}^T\mathbf{p}$ the orthogonal matrix $\mathbf{C}_{z,3}$ can be found. Thus the equations are independant of θ_3. In the equations $\mathbf{z}_3$, $\mathbf{p}_3$, $(\mathbf{p}\times\mathbf{z})_3$, $\{\mathbf{p}\times(\mathbf{p}\times\mathbf{z})+\mathbf{p}\cdot(\mathbf{p}^T\cdot\mathbf{z})\}_3$ the variable θ_3 always appears as $\mathbf{e}_3^T\cdot\mathbf{C}_{z,3}$. But it can easily be shown that equation $\mathbf{e}_3^T\cdot\mathbf{C}_{z,3}\equiv\mathbf{e}_3^T$ always holds.

Proof d): Each orthogonal matrix $\mathbf{S}_i$ and $\mathbf{C}_{z,3}$ appears at most once in each multiple matrixproduct.

Proof e): This property is a direct consequence of (26).

APPENDIX C: RPRRPR-MANIPULATOR

This section shows how the inverse kinematics of a general RPRRRPR-manipulator can be reduced to a generalized eigenproblem with dimension 8. It is known that this manipulator has at most 8 solutions to the inverse kinematic problem.

Only the equations (16,17,19,20), i.e. $\mathbf{z}$, $\mathbf{p}$, $\mathbf{p}^T\mathbf{z}$, $\mathbf{p}\times\mathbf{z}$, are used. The kinematical chain will be cut as before at the joints 3 and 6. The analogous system of equations of (30) can be obtained as

$$\boxed{\mathbf{U}\cdot\xi\cdot(1+x_1^2)=\mathbf{V}\cdot\begin{bmatrix}x_3\cdot\hat{\rho}\\ \hat{\rho}\end{bmatrix}\qquad \mathbf{U}\in\mathbb{R}^{14\times6},\ \mathbf{V}\in\mathbb{R}^{14\times16},\ \xi\in\mathbb{C}^6,\ \hat{\rho}\in\mathbb{C}^8} \tag{43}$$

$$\xi=\left[x_3s_4d_5,\ \ x_3c_4d_5,\ \ x_3d_5,\ \ s_4d_5,\ \ c_4d_5,\ \ d_5\right]^T,\ \hat{\rho}=\left[s_4,\ c_4,\ \ x_1^2d_2,\ x_1^2,\ \ x_1d_2,\ x_1,\ \ d_2,1\right]^T$$

Using 6 equations out of the 14 in (43), the unknown vector $\xi\cdot(1+x_1^2)$ may be stated as

$$\xi\cdot(1+x_1^2)=\mathbf{U}_6^{-1}\mathbf{V}_6\cdot\begin{bmatrix}x_3\cdot\hat{\rho}\\ \hat{\rho}\end{bmatrix}\quad\text{with}\ \ \mathbf{U}=\begin{bmatrix}\mathbf{U}_6\\ \mathbf{U}_8\end{bmatrix},\ \mathbf{V}=\begin{bmatrix}\mathbf{V}_6\\ \mathbf{V}_8\end{bmatrix}\ \begin{matrix}\mathbf{U}_8\in\mathbb{R}^{8\times6},\mathbf{U}_6\in\mathbb{R}^{6\times6}\\ \mathbf{V}_8\in\mathbb{R}^{8\times8},\ \mathbf{V}_6\in\mathbb{R}^{6\times8}\end{matrix} \tag{44}$$

and then eliminated out of the remaining 8 equations. One more reformulation yields

$$\underbrace{\{(\mathbf{U}_8\mathbf{U}_6^{-1})\cdot{}^1\mathbf{V}_6-{}^1\mathbf{V}_8\}}_{=:\hat{\mathbf{G}}}\cdot\hat{\rho}+x_3\underbrace{\{(\mathbf{U}_8\mathbf{U}_6^{-1})\cdot{}^x\mathbf{V}_6-{}^x\mathbf{V}_8\}}_{=:\hat{\mathbf{H}}}\cdot\hat{\rho}=0\quad \hat{\mathbf{G}},\hat{\mathbf{H}}\in\mathbb{R}^{8\times8};\ \hat{\rho}\in\mathbb{C}^8;\ x_3\in\mathbb{C} \tag{45}$$

where $\qquad \mathbf{V}_6=\left[{}^x\mathbf{V}_6,{}^1\mathbf{V}_6\right],\ \mathbf{V}_8=\left[{}^x\mathbf{V}_8,{}^1\mathbf{V}_8\right]\qquad {}^x\mathbf{V}_6,\ {}^1\mathbf{V}_6\in\mathbb{R}^{6\times8},\ \ {}^x\mathbf{V}_8,\ {}^1\mathbf{V}_8\in\mathbb{R}^{8\times8}$

With $\mathbf{G}:=\hat{\mathbf{G}}$, $\mathbf{H}:=\hat{\mathbf{H}}$, $\rho:=\hat{\rho}$ this is already the generalized eigenproblem with dimension 8 for the RPRRRPR-manipulator.

$$\boxed{\mathbf{G}\cdot\rho+x_3\,\mathbf{H}\cdot\rho=\mathbf{0}\qquad\qquad \mathbf{G},\mathbf{H}\in\mathbb{R}^{8\times8};\ \rho\in\mathbb{C}^8;\ x_3\in\mathbb{C}} \tag{46}$$

In this case the steps analogous to (35,36) are unnecessary.

On the Tangent-Half-Angle Substitution

Peter Kovács and Günter Hommel

Institut für Technische Informatik
Technische Universität Berlin, D-10587 Berlin, Germany
e-mail: kovacs@cs.tu-berlin.de

Abstract. The tangent-half-angle substitution is commonly used to convert "goniometric" equations in the sine and cosine of a certain variable θ into polynomial equations in a new variable $x = \tan(\theta/2)$. This facilitates the solution of goniometric equations and systems of equations. Elementary problems concerning the special case $\tan(\pm\pi/2)$ and the introduction of trivial extraneous roots into systems of equations are well known and can be handled. The article shows that nontrivial extraneous roots may be generated when a tangent-half-angle substitution for some variable is used to derive a characteristic equation for another variable from a goniometric system of equations. An effective method is presented to decide before the symbolic solution of a system whether a tangent-half-angle substitution produces such extraneous roots. The results can also be used to detect relevant simple subclasses of manipulators in a given superclass when no general symbolic solution for the superclass is explicitly known.

As an application, it is proven that the Raghavan-Roth algorithm for the symbolic solution of the inverse kinematics problem never generates these nontrivial extraneous roots.

1 Introduction

The solution of equations and systems of equations containing trigonometric functions in one or several variables is more difficult than the solution of ordinary polynomial equation(system)s. Therefore, methods for converting such equations into polynomial equations are desirable. This article investigates problems of a special conversion method for a special class of goniometric functions, the so-called *sine-cosine polynomials* or short *SC-polynomials*. An SC-polynomial $g(\theta)$ in some variable θ is a function of the type

$$g(\theta) \equiv \sum_{i+j=0}^{m} g_{ij}\, s^i c^j \quad \text{where } s \text{ and } c \text{ stand for } \sin(\theta) \text{ and } \cos(\theta) \text{ and } m \in \mathbf{N}. \tag{1}$$

The coefficients g_{ij} can be arbitrary functions in other variables; in particular, they can be SC-polynomials in other variables.

The most familiar conversion method for SC-polynomials is the so-called *tangent-half-angle substitution.* It is based on the elementary trigonometric identities

$$\sin(\theta) = \frac{2\tan(\theta/2)}{1+\tan^2(\theta/2)} \quad \text{and} \quad \cos(\theta) = \frac{1-\tan^2(\theta/2)}{1+\tan^2(\theta/2)} \quad \text{for } \theta \neq k\,\pi,\ k \in \mathbf{Z}\ . \tag{2}$$

By application of (2), every SC-polynomial $g(\theta)$ can be converted into a rational polynomial $\bar{g}(x)$ in a new variable $x = \tan(\theta/2)$. When $\bar{g}(x)$ is collected over the common denominator, common factors

J. Angeles et al. (eds.), Computational Kinematics, 27–39.

are cancelled and the maximal factor $(1 + x^2)^k$ in the denominator is eliminated, one obtains an ordinary polynomial $\bar{g}(x)$ in x. $\bar{g}(x)$ is called the *TS-form* ("TS" stands for tangent-half-angle substitution) of $g(\theta)$ concerning θ. To find the roots $\theta^{(i)}$ of an SC-polynomial $g(\theta)$ it is sufficient to determine the roots $x^{(i)}$ of the corresponding TS-form $\bar{g}(x)$ because $\theta^{(i)} = 2 \arctan(x^{(i)})$ for $\theta^{(i)} \neq \pm^{\pi}/_{2}$.

The conversion into the TS-form poses two well known, elementary problems. One results from the fact that $\tan(^{\theta}/_{2})$ is undefined for $\theta = \pm\pi$. This causes problems in reconstructing the roots $\theta = \pm\pi$ from $\bar{g}(x)$. Moreover, it can become difficult to reconstruct other roots, occuring in conjunction with the root $\theta = \pm\pi$; see (Lipkin & Duffy, 1985) for a detailed discussion of this problem. In (Kovács, 1991), the *homogeneous tangent-half-angle susbstitution* was introduced which solves these problems in an elegant, unified way but complicates subsequent calculations with the converted equations.

The actual reason for a conversion to TS-form is not so much the solution of single SC-polynomials but the symbolic solution of *systems* of equations of SC-polynomials ("SC-equations"). The second problem is related to this type of systems. The major step in a symbolic solution is to find a so-called (*univariate*) *characteristic equation* for a variable θ, i. e. to eliminate all other variables from the system until an equation is obtained that contains only the desired variable θ. To find a characteristic equation for θ, it may be helpful to bring SC-equations into TS-form with respect to θ. It is obvious that the elimination of the denominator $(1 + x^2)^k$ during the conversion may prevent the cancellation of factors $(1 + x^2)$ in later steps of the elimination process. Therefore, extraneous roots $x = \pm\iota$ with $\iota = \sqrt{-1}$, may appear in the final characteristic equation for x. This phenomenon, well known in kinematics, is discussed for example by Raghavan and Roth (1991). Obviously, extraneous factors $(1 + x^2)$ can be detected and eliminated easily from characteristic equations.

This article investigates another problem of the tangent-half-angle substitution which is closely related to the last one. We show that extraneous roots "generated" by a conversion to TS-form with respect to θ_i can induce extraneous roots also in the characteristic equations $f(\theta_j) = 0$ for variables θ_j with $i \neq j$. Note that the extraneous roots are generated in $f(\theta_j)$ although their "generator", the variable x_i, is eliminated during the solution of the system of equations and does not appear in $f(\theta_j)$. In contrast to the second problem mentioned above, this kind of extraneous roots cannot be detected and eliminated easily anymore. Simultaneous conversions into TS-form for variables θ_i, θ_j, ... can generate a set of different extraneous roots in the characteristic equation for θ_k.

The results presented in this article are the following:

- An effective method is developed to decide "in advance", before a symbolic solution of the system is known, whether a conversion into TS-form with respect to some variable generates extraneous roots in a given system of SC-equations ("SC-system").
- The method can serve as an *indicator* for certain relevant simplifications during the solution of SC-systems. They can be detected easily *before* the system is solved. The simplifications are independent from the conversion into TS-form - they apply to unconverted systems.
- The technique permits to identify "in advance" subclasses of a manipulator class that possess simpler solutions than the general class. Manipulators with simple solutions are relevant for industrial applications.

2 Extraneous roots and extraneous factors

From now on the problem is discussed from a kinematical viewpoint; see (Paul, 1981) for a comprehensive introduction to kinematics. SC-polynomials play a major role in this field. In the sequel we distinguish in particular between *revolute* variables θ and *prismatic* variables d. If a variable is revolute, all equations in the system of kinematics equations are SC-equations with respect to θ. For a prismatic variable d, all equations are ordinary polynomial equations with respect to d.

In kinematics, we are not so much interested in isolated extraneous roots that occur only at certain effector poses. Their occurence after a conversion to TS-form is a normal, unavoidable phenomenon that does not affect the solution of kinematical systems of equations; see (Kovács, 1993). Much more important are extraneous factors. A (*global*) *extraneous factor* in a characteristic equation $f(q) = 0$ for some variable q, which may be revolute or prismatic, is a factor $p(q)$ of $f(q)$ such that all solutions of $p(q) = 0$ are extraneous roots. Note that extraneous factors yield extraneous roots for any given effector pose, reachable or not. The distinction between extraneous roots and factors is apparently meaningless if all parameters in the system of kinematics equations are set to numerical values. The method presented here detects both: isolated extraneous roots and extraneous factors caused by a conversion into TS-form.

2.1 An example

To examine the problem in detail, a kinematical example is investigated. Let

$$(\theta_1, 0, 0, 0)\ (0, d_2, 0, {}^{\pi}/_2)\ (\theta_3, d_3, a_3, \alpha_3)\ (\theta_4, d_4, a_4, 0)\ (\theta_5, 0, a_5, \alpha_5)\ (\theta_6, 0, 0, 0) \tag{3}$$

be the Denavit-Hartenberg specification (see (Paul, 1981)) of a manipulator class with *constant* d_3 and θ_4, i. e. the third and fourth joint *variable* are θ_3 and d_4. As usual, $\sin(\theta_i)$, $\cos(\theta_i)$, $\sin(\alpha_j)$ and $\cos(\alpha_j)$ are abbreviated by s_i, c_i, μ_j and λ_j respectively. The *pose parameters*, i. e. the elements of the homogeneous 4 × 4 effector matrix $\boldsymbol{T}$ specifying the effector position, are denoted by t_{ij}. Let

$$X = (\lambda_3\, d_3 + d_4),\quad Y = (c_4\, a_3 + a_4 + d_3\, \mu_3\, s_4),\quad Z = (c_4\, d_3\, \mu_3 - a_3\, s_4). \tag{4}$$

The following equations can be derived from the system of kinematics equations of (3)

$$\begin{aligned} 0 = \;& t_{14} + t_{13}\, (\mu_5\, (c_5\, Z - s_5\, Y) - \lambda_5\, X) + s_6\, \big(t_{11}\, (\lambda_5\, (s_5\, Y - c_5\, Z) - \mu_5\, X) + \\ & t_{12}\, (a_5 + c_5\, Y + s_5\, Z)\big) + c_6\, \big(t_{12}\, (\lambda_5\, (s_5\, Y - c_5\, Z) - \mu_5\, X) - t_{11}\, (a_5 + c_5\, Y + s_5\, Z)\big) \end{aligned} \tag{5}$$

$$\begin{aligned} 0 = \;& t_{24} + t_{23}\, (\mu_5\, (c_5\, Z - s_5\, Y) - \lambda_5\, X) + s_6\, \big(t_{21}\, (\lambda_5\, (s_5\, Y - c_5\, Z) - \mu_5\, X) + \\ & t_{22}\, (a_5 + c_5\, Y + s_5\, Z)\big) + c_6\, \big(t_{22}\, (\lambda_5\, (s_5\, Y - c_5\, Z) - \mu_5\, X) - t_{21}\, (a_5 + c_5\, Y + s_5\, Z)\big) \end{aligned} \tag{6}$$

$$\begin{aligned} 0 = \;& (\lambda_3\, \mu_5 + \lambda_5\, \mu_3\, (c_5\, c_4 - s_5\, s_4))\, (s_6\, t_{31} + c_6\, t_{32}) + \mu_3\, (c_4\, s_5 + c_5\, s_4)\, (c_6\, t_{31} - s_6\, t_{32}) + \\ & (\lambda_3\, \lambda_5 + \mu_3\, \mu_5\, (-c_5\, c_4 + s_5\, s_4))\, t_{33} \end{aligned} \tag{7}$$

Equations (5) and (6) are the first and second "positional equation" ("element-equation (1,4) and (2,4)") of the homogeneous 4 × 4 kinematic matrix equation

$$A_1 * A_2 = \boldsymbol{T} * A_6^{-1} * A_5^{-1} * A_4^{-1} * A_3^{-1} \tag{8}$$

where the A_i are the familiar arm matrices; see (Paul, 1981). (7) is element-equation (3,3) of (8). Equations (5), (6) and (7) constitute a system of three equations in three variables d_4, θ_5 and θ_6. For a first, detailed inspection of the problem, we simplify the manipulator class drastically and set $d_3 = a_3 = \alpha_3 = \theta_4 = a_4 = 0$, $a_5 = 1$ and $\alpha_5 = \text{atan2}(3, 4)$, i. e. $\mu_5 = {}^3/_5$ and $\lambda_5 = {}^4/_5$. In this case, θ_5 vanishes from (5) and (6), yielding a simple system of two equations in two unknowns d_4 and θ_6. To avoid any distraction and to simplify considerations further, all pose parameters are set to (arbitrary) numerical values

$$T_0 = \begin{bmatrix} -1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

As a notational simplification we omit the indices of the variables. This transforms (5) and (6) into

$$\left\{\begin{array}{l} {}^3/_5\ s\ d + c - 5 = 0 \\ s - {}^3/_5\ c\ d + 2 = 0 \end{array}\right\} \qquad \begin{array}{r}(10)\\(11)\end{array}$$

Obviously, no direct simplification of this system is possible. Solution of (10) for d and subsequent substitution into (11) gives

$$-5c + c^2 + 2s + s^2 = 0 \quad \text{for } s \neq 0. \qquad (12)$$

The last equation can be simplified and yields a characteristic equation of degree 2 after conversion to TS-form.

Next, we investigate the solution in the case of a tangent-half-angle substitution. The substitution according to (2) yields for (10) and (11):

$$\frac{-4 + {}^6/_5 dx - 6x^2}{1 + x^2} = 0\,. \qquad (13)$$

$$\frac{2 - {}^3/_5 d + 2x + (2 + {}^3/_5 d)x^2}{1 + x^2} = 0 \qquad (14)$$

No simplification is possible in neither of the equations. An extraneous root is generated when the denominators $(1+x^2)$ are cleared. Solution of the numerator of (13) for d and substitution into (14) yields for $x \neq 0$

$$-2 + 2x + x^2 + 2x^3 + 3x^4 \equiv (1+x^2)(-2 + 2x + 3x^2) = 0\,. \qquad (15)$$

(15) contains an extraneous factor and the equation also reveals its origin: it is cancelled, if the denominator $(1+x^2)$ of (14) is not eliminated. This side-effect of a coversion to TS-form is well known and it is easy to determine the correct characteristic equation in such a case: We only have to investigate if[1] $x = \pm\iota$ is a solution of the characteristic equation and if so, divide the equation by the maximum power of $(1+x^2)$. According to (2), $x = \pm\iota$ corresponds to a solution θ at complex infinity.

A more difficult problem arises if a characteristic equation for another variable is derived after conversion to TS-form. Let $\mathfrak{F}$ be the system of equations consisting of the numerators of (13) and (14). If $\mathfrak{F}$ is solved for d instead of θ we get

$$-17500 - 6075d^2 + 81d^4 \equiv (9d^2 - 700) * (9d^2 + 25) = 0\,. \qquad (16)$$

The factor $(9d^2 - 700)$ yields the correct solutions, corresponding to the two proper solutions of (10) and (11). The solutions $d = \pm{}^5/_3\iota$ of $(9d^2 + 25) = 0$ are generated by the "original" extraneous roots $x = \pm\iota$ of $\mathfrak{F}$. This becomes obvious by setting $d = \pm{}^5/_3\iota$ in $\mathfrak{F}$ because the only solutions of the resulting two systems in one variable are $x = \pm\iota$. We say that x (or more precisely: the extraneous factor $(1 + x^2)$ in (15)) is the *generator* and $(9d^2 + 25)$ in (16) is its *conjugate* extraneous factor. Note that the origin of the two solutions $d = \pm{}^5/_3\iota$ and the fact that they are extraneous cannot be derived from the characteristic equation (16) alone. The conjugate extraneous factor $(9d^2 + 25)$ in (16) does not appear in a particular, clearly recognizable form like its generator $(1 + x^2)$ in equation (15). In general, generators are always of the form $(1 + x^2)$ but the form of their conjugate extraneous factors is not unique. The latter is determined by the particular system of equations. Thus, generators can be identified directly but not their conjugate extraneous factors. To detect conjugate extraneous roots in the general case, one has to inspect all solution sets (in kinematics: all *joint configurations*) of the system of equations.

[1] As mentioned before, ι stands for $\sqrt{-1} \in \mathbb{C}$.

All observed phenomena can obviously occur as well if the system contains no prismatic but only revolute variables.

It is evident that the symbolic solution of systems of equations usually becomes far more difficult in the presence of extraneous factors since intermediate and resulting equations are more complicated and of larger degree than the correct ones. Moreover, it is more difficult to solve the resulting characteristic equations if conjugate extraneous factors are not detected and eliminated because they are of larger degree. This is particularily important for time critical applications like in robot control units.

The above problems can even increase. If complicated systems of multivariate SC-equations are to be solved it may be helpful to carry out simultaneous conversions to TS-form with respect to several variables, i. e. the converted system may contain different new variables $x_i = \tan(\theta_i/2)$. According to the preceding considerations, a characteristic equation for some variable q_j may now contain different extraneous factors which are difficult to identify. Each of these factors corresponds to a solution $x_i = \pm\iota$ for some x_i. The identification and elimination of these extraneous factors can require considerable effort, especially in the case of large systems with several formal, i. e. nonnumerical, parameters. Section 4 gives an example of such a combination of extraneous factors.

Consequently, it is desirable to decide in advance, before a system is solved symbolically, if a conversion to TS-form introduces extraneous roots. Before an effective method for the early detection of extraneous roots is developed we briefly investigate some attempts to circumvent the problem. First, it may seem that the homogeneous tangent substitution mentioned above can prevent the generation of extraneous roots but is proven in (Kovâcs, 1991) that this is never the case. Theoretically, the concept of *ideal-quotients* can be used to eliminate extraneous factors from an ideal but unfortunately this does not provide a practical method of solving the problem.

2.2 Real extraneous roots

Equation (16) seems to indicate that conjugate extraneous roots must be complex. This is not true. The system of equations

$$\left\{\begin{array}{c} 10\, s_2\, s_1 - 4\, c_1\, d + 7 = 0 \\ 4\, s_1\, d + 5\, c_1 - 1 = 0 \\ s_2^2 - {}^5/_2\, s_2\, c_1 - s_1\, d = 0 \end{array}\right\} \tag{17}$$

is a counter example. When it is converted to TS-form with respect to θ_1, extraneous roots are generated. The resulting univariate characteristic equation for s_2 is

$$(2\, s_2 - 1)^2 * (-294 + 49\, s_2 + 674\, s_2^2 - 100\, s_2^3 - 196\, s_2^4 + 16\, s_2^5 + 16\, s_2^6) = 0 \tag{18}$$

and $s_2 = {}^1/_2$ is the extraneous root conjugated to x_1. Thus, extraneous roots in characteristic equations can even be real.

2.3 Euler substitution

Actually, the problem is not specifically related to the tangent-half-angle substitution. It occurs whenever a conversion of a system of equations into polynomial form requires the elimination of non-trivial common denominators. As an example, the *Euler substitution* is investigated which is based on the familiar identities

$$\sin(\theta) = \iota * \frac{1 - y^2}{2\, y}, \qquad \cos(\theta) = \frac{y^2 + 1}{2\, y} \qquad \text{with } y = e^{\iota\theta}. \tag{19}$$

Conversion of (10) and (11) and subsequent elimination of common denominators yields

$$\left\{\begin{matrix} 5\,\iota\ -3\,d+20\,y-5\,\iota\ y^2-3\,d\,y^2=0 \\ 5+3\,\iota\ d-50\,y+5\,y^2-3\,\iota\ d\,y^2=0 \end{matrix}\right\}. \tag{20}$$

The resulting univariate characteristic equation for y is

$$(21-20\,\iota\,)\,y+(-10+4\,\iota\,)\,y^2+29\,y^3=0 \tag{21}$$

and obviously, $y = 0$ is an extraneous root. The univariate characteristic equation for d is

$$-3500\,\iota\ -2100\,d-45\,\iota\,d^2+27\,d^3\equiv(9d^2-700)*(3\,d-5\,\iota)=0 \tag{22}$$

and $d = {}^5/_3\iota$ is the corresponding extraneous root. Thus, a similar effect as in the case of the conversion to TS-form is encountered. However, the Euler substitution is apparently better suited for practical purposes: The elementary "original" extraneous factor (y) is linear instead of the quadratic factor $(1 + x^2)$ and is of simpler form! Moreover, the conjugate extraneous roots in other characteristic equations are also linear as demonstrated by (22). (Kovács, 1993) contains details about Euler substitutions.

3 Control systems

In this section a method is developed to decide whether a conversion to TS-form generates extraneous roots in a given system of SC-equations.

3.1 Definition and basic properties

Let $\mathfrak{T}$ be a system of SC-equations in variables θ, q_1, q_2, ... of the type

$$\Sigma_{i+j=0}^{m}\ g_{ij}(q_1, q_2, ...)\ s^i\,c^j\ =\ 0\ \ ,\quad m\in\mathbf{N} \tag{23}$$

with coefficients $g_{ij}(q_1, q_2, ...)$ that are functions in the remaining variables q_1, q_2, ... The task is to determine if a conversion to TS-form with respect to θ produces a factor $(1 + x^2)$ in the resulting characteristic equation for x. Such a factor appears if and only if the two values $x = \pm\iota$ lead to correct solutions of the system for all values of the formal parameters contained in the system. That means: For $x = \pm\iota$ and for any set of real or complex values for the formal parameters there exists a set of values for the remaining variables q_1, q_2, ... such that the resulting converted system of equations is satisfied. In the case of kinematic systems of equations, $x = \pm\iota$ must lead to correct solutions for arbitrary real or complex values of the pose parameters t_{ij} and of the formal (nonnumeric) Denavit-Hartenberg parameters (short: *DH-parameters*).

Before continuing we introduce a normalization for SC-polynomials that was already proposed by Lipkin and Duffy (1985). This normalization proves to be crucial for the subsequent investigation. If all occurences of s^2 in an SC-polynomial $g(\theta)$ of type (1) or (23) respectively are replaced by $(1 - c^2)$, one obtains an equivalent SC-polynomial $h(\theta)$ where s appears only linearly

$$h(\theta)\equiv\Sigma_{j=0}^{n}h_{0j}\,c^j\ +s*\Sigma_{j=0}^{n-1}h_{1j}\,c^j \qquad \text{with } n\in\mathbf{N},\, n\leq m \tag{24}$$

and with corresponding coefficients h_{ij} and $h_{0n} \neq 0$ or $h_{1,n\text{-}1} \neq 0$. (24) is called the *c-s-normalform* of $g(\theta)$. All SC-polynomials are considered to be in c-s-normalform from now on.

The conversion of (24) to TS-form yields[1]

[1] Indices are separated by comma when necessary.

$$\left(\sum_{j=1}^{n}(h_{0j}\,(1\text{-}x^2) + h_{1,j\text{-}1}\,(2x))\,(1\text{-}x^2)^{j\text{-}1}\,(1\text{+}x^2)^{n\text{-}j}\right) + h_{00}\,(1\text{+}x^2)^n = 0. \qquad (25)$$

To investigate if the system produces the factor $(1 + x^2)$ we must set x first to $\imath$ and subsequently to $-\imath$ and inspect the resulting two systems. Each of both substitutions eliminates all but the leading two summands (with $j = n$) on the left-hand side of (25), yielding

$$h_{0n}\,(1 - \imath^2)^n + h_{1,n\text{-}1}\,(\pm 2\imath)\,(1 - \imath^2)^{n\text{-}1} = 0\,. \qquad (26)$$

Thus, after dividing the resulting equations by the common factor 2^n one obtains

$$h_{0n} + \imath\, h_{1,n\text{-}1} = 0 \qquad \text{for } x = \imath \qquad (27)$$

$$h_{0n} - \imath\, h_{1,n\text{-}1} = 0 \qquad \text{for } x = -\imath \qquad (28)$$

Note that the two coefficients are usually functions $h_{ij}(q_1, q_2, ...)$ containing the remaining variables. Obviously, the two equations are considerably simpler than the original equation (25). The prior conversion to c-s-normalform has contributed significantly to this simplification. The two equations (27) and (28) are called the *positive* and *negative control equation* of (25) *with respect to* θ. The set of positive control equations of all equations of $\mathfrak{F}$ constitutes the *positive control system with respect to* θ. In the analogous way, the *negative control system* is formed. A solution of a control system is a set of values for the "remaining" variables $q_1, q_2, ...$ such that the control system is satisfied. If a control system of a system of kinematics equations is solvable for *all* real and complex effector poses we say that it is *globally solvable*. The solutions of the control system must of course be the conjugate extraneous roots with respect to x; section 3.2 gives an example.

If both control systems of a system of (kinematics) equations with respect to a variable θ are globally solvable, $x = \pm\imath$ must be a solution of the converted system for all effector poses. Therefore, the characteristic equation for x must contain an extraneous factor $(1 + x^2)$ and the characteristic equations for the other variables contain conjugate extraneous roots.

If one of the control systems is not globally solvable, then x cannot generate an extraneous *factor*. However, if a conversion of $\mathfrak{F}$ to TS-form was done also for some other variable $\theta_i \neq \theta$, the univariate characteristic equation for x may contain conjugate extraneous factors which are generated by the conversion with respect to θ_i.

Nonglobal solutions of a control system identify effector poses where the total number of real and complex joint configurations decreases. Every neighbourhood of such an effector pose contains correct solutions θ that are arbitrarily close to complex infinity (without proof).

If all equations in the original system have real coefficients the (global) solvability of one of the two control systems always implies the solvability of the other; a proof of this statement is contained in (Kovács, 1993). Systems of kinematics equations always have real coefficients and thus it is sufficient to check only the positive control system.

3.2 The general case of the initial example

As an example, the general, unspecialized system of equations (5), (6) and (7) is investigated. Its positive control system with respect to θ_6 is

$$(\imath\, t_{11} + t_{12})\left(\imath\, a_5 - \mu_5 X + c_5\,(\imath\, Y - \lambda_5 Z) + s_5\,(\lambda_5 Y + \imath\, Z)\right) = 0 \qquad (29)$$

$$(\imath\, t_{21} + t_{22})\left(\imath\, a_5 - \mu_5 X + c_5\,(\imath\, Y - \lambda_5 Z) + s_5\,(\lambda_5 Y + \imath\, Z)\right) = 0 \qquad (30)$$

$$(\imath\, t_{31} + t_{32})\left(\lambda_3\, \mu_5 + \mu_3\,(c_5\,(\lambda_5\, c_4 - \imath\, s_4) - s_5\,(\imath\, c_4 + \lambda_5\, s_4))\right) = 0 \qquad (31)$$

This system is globally solvable: The second factor of (31) determines θ_5 and for any value of θ_5, the right factor in (30) vanishes by appropriate choice of d_4 in X. The right factor in (29) is identical to the corresponding factor in (30) and vanishes simultaneously. Consequently, a conversion to

TS-form generates an extraneous factor $(1 + x_6^2)$ in the system. If the parameters are set to the values of the example in section 2.1, (29) or (30) yield exactly the conjugate extraneous roots of (16).

3.3 Extraneous roots under equivalent transformations

It is an important fact that the generation of extraneous roots can be prevented in some cases by a *preceding* equivalent transformation of the system. This is demonstrated by the trivial example

$$t_{11}\, d\, c^2 + t_{11}\, s + t_{22} = 0 \tag{32}$$

$$t_{33}\, d\, c^2 + t_{22}\, c + d = 0 \tag{33}$$

The system ((32) and (33)) leads to a globally solvable control system $\{t_{11}\, d = 0,\ t_{33}\, d = 0\}$. If this system is converted to TS-form, the resulting characteristic equations contain an extraneous factor. The generation of extraneous roots can be prevented as follows. Prior to the conversion to TS-form the algebraic combination $t_{33} * (32) - t_{11} * (33)$ of the original equations yields

$$t_{33}\, t_{11}\, s - t_{11}\, t_{22}\, c + t_{33}\, t_{22} - t_{11}\, d = 0\,. \tag{34}$$

One of the two original equations, e. g. (32), may be replaced by (34) without changing the solution set of the system. If the conversion to TS-form is done *after* this transformation the control system is not globally solvable since the control equation corresponding to (34) is $t_{33} t_{11}\, \iota - t_{11} t_{22} = 0$! The reverse transformation of ((33) and (34)) into ((32) and (33)) demonstrates that equivalent transformations of the original system can introduce additional extraneous roots into the converted system! It is obvious that a conversion to c-s-normalform never has this effect.

4 Essential pythagorean simplifications

After having investigated methods to identify extraneous roots we inspect the unconverted system in case that a conversion generates extraneous roots. First, single equations are investigated. A tangent-half-angle substitution of (12) yields

$$\begin{array}{ccc} -5c + 2s & + & c^2 + s^2 \\ \downarrow & & \downarrow \\ \dfrac{-5 + 4\,x + 5\,x^2}{1 + x^2} & + & \dfrac{(1 + x^2)^2}{(1 + x^2)^2} \end{array} \tag{35}$$

When (35) is collected over the common denominator $(1 + x^2)^2$ (i. e. without cancellation in the second term!), the extraneous factor appears in the numerator. The resulting numerator equals (15) multiplied by 2. (35) indicates that the occurence of an extraneous factor $(1 + x^2)$ in the numerator (TS-form!) corresponds to the applicability of the *pythagorean formula* $s^2 + c^2 = 1$ in the original SC-polynomial. We investigate the general case. Let

$$g(\theta) \equiv (s^2 + c^2)^{\kappa}\, b_1(\theta) + b_2(\theta)\,, \qquad \kappa \in \mathbf{N} \tag{36}$$

with some SC-polynomials $b_1(\theta)$ and $b_2(\theta)$ in c-s-normalform. Apparently, $g(\theta)$ is not in c-s-normalform. The equivalent transformation of g into $b_1 + b_2$ is called a *pythagorean simplification.* We speak of an *essential* pythagorean simplification if $\deg(b_1) + 4\,\kappa > \deg(b_2)$, where the degree $\deg(g)$ of an SC-polynomial $g(\theta)$ is defined as the total number of roots of $g(\theta)$ counted with multiplicity[1]. Essential pythagorean simplifications are exactly those that eliminate all m+1 *leading* co-

[1] With some practically irrelevant exceptions, $\deg(g)$ equals the ordinary polynomial degree of the TS-form of g with respect to x. For $h(\theta)$ of (24) it can be shown that $\deg(g) = 2n$.

efficients g_{ij} in (1), i. e. the coefficients with $i + j = m$. If such a simplification is possible for certain parameter values only the degree of the equation apparently decreases for these values.

Let $\bar{b}_1$ and $\bar{b}_2$ be the TS-forms of b_1 and b_2. A tangent-half-angle substitution for the right side of (35) yields

$$\frac{(1 + x^2)^{2\kappa}\ \bar{b}_1(x)}{(1 + x^2)^{\nu}} + \frac{\bar{b}_2(x)}{(1 + x^2)^{\mu}} \qquad \text{with } \nu, \mu \in \mathbf{N} \tag{37}$$

Since b_1 and b_2 are in c-s-normalform, (25) shows that $\bar{b}_1$ and $\bar{b}_2$ cannot contain a factor $(1 + x^2)$. When (37) is collected over its common denominator without cancellation of common factors the numerator contains a factor $(1 + x^2)$ if and only if $(2\nu =)\ \deg(b_1) + 4\ \kappa > \deg(b_2)\ (= 2\mu)$. Consequently, an extraneous factor $(1 + x^2)$ appears in the numerator of (37) if and only if the original SC-polynomial $g(\theta)$ permits an essential pythagorean simplification. The extraneous factor can obviously be cancelled in (37), i. e. it does not appear in the TS-form of $g(\theta)$.

This has an important practical consequence for the solution of SC-systems: the appearance of the extraneous factor $(1 + x^2)$ in a characteristic equation indicates an essential pythagorean simplification at some time during the solution process of the *un*converted system. Thus, any symbolic solution of the system must contain one characteristic equation whose degree decreases[1]. For manipulators, all prior results show that there is at most one equation of degree greater two in a symbolic solution. Consequently, the degree of this "complicated" equation must drop due to an essential pythagorean simplification. Usually, the *univariate* characteristic equation is affected.

Note that it is impossible to obtain this information with methods that yield (symbolic) solutions only in case that all formal parameters are set to numerical values.

The above observation is helpful for practical mass examinations of manipulator classes. Assume that a manipulator class $\mathcal{M}$ is investigated and no general symbolic solution for the class is known but some system of equations is found from which characteristic equations can be derived. The concept of control systems permits the identification of subclasses of $\mathcal{M}$ for which the degree of the "complicated" characteristic equation decreases. These are the simple subclasses, i. e. the industrially relevant manipulators among the given class. The information is obtained without having to calculate the general symbolic solution of the - perhaps very difficult - system of equations. It is sufficient to determine the values of the DH-parameters for which the control system with respect to some variable θ is globally solvable.

As an example we investigate the system ((29), (30), (31)) again. As seen before, there must be an essential pythagorean simplification due to the extraneous factor generated by x_6. Now, simplifications due to θ_5 or x_5 respectively are investigated. The positive control system with respect to θ_5 is

$$(\iota Y - Z)\ \ (t_{11}\,(\lambda_5\, s_6 + \iota\, c_6) + t_{12}\,(\lambda_5\, c_6 - \iota\, s_6) - \mu_5\, t_{13}) = 0 \tag{38}$$

$$(\iota Y - Z)\ \ (t_{21}\,(\lambda_5\, s_6 + \iota\, c_6) + t_{22}\,(\lambda_5\, c_6 - \iota\, s_6) - \mu_5\, t_{23}) = 0 \tag{39}$$

$$\mu_3\,(c_4 + -\iota\, s_4)\,(\lambda_5\,(s_6\, t_{31} + c_6\, t_{32}) + \iota\,(c_6\, t_{31} - s_6\, t_{32}) - \mu_5\, t_{33}) = 0 \tag{40}$$

First we note that the variable d_4 is not contained in the control system. The system is globally solvable only if $(\iota\, Y - Z) = 0$ because the factors of (38) and (39) containing pose parameters cannot vanish identically for arbitrary $\boldsymbol{T}$. On the other hand, if $(\iota\, Y - Z) = 0$ then there is always some θ_6 satisfying (40) and thus the system is globally solvable. For real DH-parameters, $(\iota\, Y - Z) = 0$ is equivalent to $Y = Z = 0$. Except from trivial solutions, the last condition is satisfied if and only if

[1] A symbolic solution always consist of a *sequence* of characteristic equations such that the j-th equation determines the solutions of some variable q_{i_j}.

$a_3 = -c_4\, a_4$ and $a_4 = -\, d_3\, \mu_3 / s_4$. Consequently, an essential pythagorean simplification must be possible during the derivation of the resulting characteristic equations. Thus we have identified a simple(r) subclass of the original class having a smaller number of configurations.

Actually, the global solvability of the control system ((38), (39), (40)) for $Y = Z = 0$ just reflects the fact that the variable θ_5 vanishes from (5) and (6) in this case. The resulting system of two equations can be solved easily.

The control system ((29), (30), (31)) with respect to θ_6 was globally solvable without restrictions. Thus, $Y = Z = 0$ is an example of a case where two essential pythagorean simplifications combine, i. e. a simultaneous conversion to TS-form with respect to θ_5 and θ_6 generates two extraneous factors, one by θ_5 and the other by θ_6.

Obviously, a drop in the number of solutions of a system of SC-equations must not necessarily be caused by an essential pythagorean simplification. It can occur as well in the way it does in ordinary polynomial systems. These "ordinary" cases can be investigated with the aid of the familiar *homogenization* of the system. Information about a drop in the number of solutions is obtained by setting the *homogeneous extension* to zero. It must also be mentioned that control systems do not indicate all pythagorean simplifications. The corresponding statement only holds for control *equations*, but not for *systems*. A detailed analysis of the remaining cases is contained in (Kovács, 1993).

5 Application to the Raghavan-Roth algorithm

As a relevant example the universal algorithm for the solution of the inverse kinematics problem by Raghavan and Roth (1991) (short: *RR-algorithm*) is investigated. In the following, we use the notation of Raghavan and Roth and refer to the formulae in their paper by preceding formula-numbers with "RR", e. g. (RR-17). The RR-algorithm reduces the solution of the inverse kinematics problem to the solution of a system of 6 quadratic equations in three variables θ_3, θ_4 and θ_5. This system is solved via a conversion to TS-form for all three variables and a subsequent dialytic elimination of x_4 and x_5. Raghavan and Roth prove that a conversion with respect to θ_3 always generates an extraneous factor $(1+x_3^2)^4$ in the characteristic equation for x_3. For this reason they do not eliminate denominators $(1+x_3^2)$ in all six equations but leave two equations in rational form, i. e. in the form $\overline{f}(x_3) = 0$ of section 1. They prove that the extraneous factor $(1+x_3^2)^4$ is cancelled in this way during the dyalitic elimination.

In the light of the preceding sections it is natural to ask whether the conversion with respect to θ_4 (or θ_5) generates nontrivial conjugate extraneous factors in the characteristic equation for x_3. Since the RR-algorithm produces at most a 16-th degree polynomial in x_3 it is obvious that conjugate extraneous factors cannot appear in the case of manipulators with 16-th degree characteristic equations. Thus, the question is whether special manipulator geometries with lower degree characteristic equations exist where the algorithm produces nontrivial extraneous factors.

The naïve approach to this problem would try to proceed as follows: Calculate the symbolic solution for the completely general case where all DH- and pose parameters are *formal* (unspecialized) parameters, inspect the characteristic equations for x_4 and x_5 (after having determined x_3) and try to determine DH-parameters such that one of these equations contains solutions $x_j = \pm\iota$ for all effector poses. The last step is difficult enough, but the naïve approach already fails because of the first step. It is virtually impossible to explicitly obtain the characteristic equations for completely general formal parameters with the RR-algorithm (or in any other known way) due to the extreme complexity of the necessary calculations in this case. Thus, the concepts developed above are the only means to investigate the problem.

The system of the original six equations (RR-17) consists of two extremely complex equations (equations A and B of appendix 2 in[1] (Raghavan & Roth, 1991)) and four comparatively simple equations (RR-26). The control system of the original six equations (RR-17) still is very complex. It contains two remaining variables θ_3 and θ_5. No pair of control equations contain a common factor which is free of pose parameters as in (29) and (30). Consequently, two of the six control equations determine values of θ_3 and θ_5 and the other four equations must hold for *all* these values. We assume first that A and B determine θ_3 and θ_5. This is the most difficult case. The four remaining equations (RR-26) are obtained from a system of six quadratic equations in θ_1, θ_3, θ_4 and θ_5 by using two of the six equations to eliminate θ_1. We investigate this system in detail. The equations are denoted $l_3, p_3, \tilde{\boldsymbol{p}} \cdot \tilde{\boldsymbol{p}}, \tilde{\boldsymbol{p}} \cdot \tilde{\boldsymbol{l}}, (\tilde{\boldsymbol{p}} \times \tilde{\boldsymbol{l}})_3, ((\tilde{\boldsymbol{p}} \cdot \tilde{\boldsymbol{p}})\,\tilde{\boldsymbol{l}} - (2\tilde{\boldsymbol{p}} \cdot \tilde{\boldsymbol{l}})\,\tilde{\boldsymbol{p}}\,)_3$ in the original article. For the sake of simplicity we denote these equations by E_1 to E_6 in the sequence given above. The left- and right-hand sides of E_i are L_i and R_i, i. e. E_i is of the form $L_i = R_i$. Without loss of generality it can be assumed that $d_1 = d_6 = a_6 = \alpha_6 = 0$ because the corresponding constant elementary transformations of the kinematic matrix equation can always be "multiplied into the effector matrix T", yielding a constant, linear, invertible transformation of T. All six equations E_i are of the form

$$u_{01}{}^{(i)}\, c_4 + u_{10}{}^{(i)}\, s_4 + u_{00}{}^{(i)} = v_{01}{}^{(i)}\, c_1 + v_{10}{}^{(i)}\, s_1\ , \quad i \in \{1, \ldots 6\} \tag{41}$$

where $u_{01}{}^{(i)}$ and $u_{10}{}^{(i)}$ are functions of θ_3, θ_5 and of the Denavit-Hartenberg- but not of the pose parameters. The coefficients $u_{00}{}^{(i)}$ depend on all variables and parameters including the pose parameters. $v_{01}{}^{(i)}$ and $v_{10}{}^{(i)}$ are variable-free expressions in a_1, λ_1, μ_1 and the pose parameters. The RR-algorithm leaves open which equations to use for the elimination of θ_1. We select E_1 and E_2, i. e. l_3 and p_3 for this purpose because their right sides have a simple structure. The solution of E_1 and E_2 for θ_1 yields

$$s_1 = \frac{L_1\, t_{23} - L_2\, t_{24}}{\mu_1\,(t_{14}\, t_{23} - t_{13}\, t_{24})}\ , \quad c_1 = \frac{L_1\, t_{13} - L_2\, t_{14}}{\mu_1\,(t_{14}\, t_{23} - t_{13}\, t_{24})} \quad \text{for } \mu_1 \neq 0 \tag{42}$$

When all occurences of s_1 and c_1 in $R_3, \ldots R_6$ are replaced by (42) one obtains four new equations of the form

$$L_i = w_1{}^{(i)}\, L_1 + w_2{}^{(i)}\, L_2 \text{ for } i \in \{3, \ldots, 6\} \tag{43}$$

with variable-free expressions $w_1{}^{(i)}$ and $w_2{}^{(i)}$ containing a_1, λ_1, μ_1 and the pose parameters. These are the four equations of (RR-26). According to (41) the control equation of (43) is

$$u_{01}{}^{(i)} + u_{10}{}^{(i)}\, \iota = w_1{}^{(i)}\, L_1' + w_2{}^{(i)}\, L_2'\ , \quad i \in \{3, \ldots, 6\} \tag{44}$$

with $L_1' = (u_{01}{}^{(1)} + u_{10}{}^{(1)}\, \iota)$ and $L_2' = (u_{01}{}^{(2)} + u_{10}{}^{(2)}\, \iota)$. L_1' and L_2' and the left-hand side of (44) are free of pose parameters. To be globally solvable the right side must also be free of pose parameters. The only possibility to achieve this is to select DH-parameters such that $L_1' = L_2' = 0$. To find appropriate sets of parameters we investigate L_1' and L_2'. Let $r = (c_3\, \lambda_3\, \mu_2 + \lambda_2\, \mu_3)$ and let

$$U = (\iota\, s_5,\ \iota\, c_5,\ s_5,\ c_5,\ \iota,\ 1) \tag{45}$$

$$V_1 = (-s_3\, a_5\, \lambda_4\, \mu_2,\ r\, a_5,\ r\, a_5\, \lambda_4,\ s_3\, a_5\, \mu_2,\ r\, a_4 + s_3\, d_5\, \mu_2\, \mu_4,\ s_3\, a_4\, \mu_2 - r\, d_5\, \mu_4) \tag{46}$$

$$V_2 = (r\, \mu_5,\ s_3\, \lambda_4\, \mu_2\, \mu_5,\ s_3\, \mu_2\, \mu_5,\ -r\, \lambda_4\, \mu_5,\ s_3\, \lambda_5\, \mu_2\, \mu_4,\ -r\, \lambda_5\, \mu_4)\,. \tag{47}$$

The V_i terms consist of the first six elements of the rows corresponding to l_3 and p_3 of the matrix $\boldsymbol{P}$ in (RR-16). In accordance with (RR-13) and (RR-14), $L_1' = V_1 * U$ and $L_2' = V_2 * U$. To be identical zero, all components of V_1 and V_2 must be zero. Figure 1 depicts *all* different possibilities to achieve this. A complete directed path in the figure defines a sequence of conditions yielding $L_1' = L_2' = 0$. The label (i, j) indicates that the following condition(s) or alternatives, respectively,

[1] These two equations contain minor printing errors.

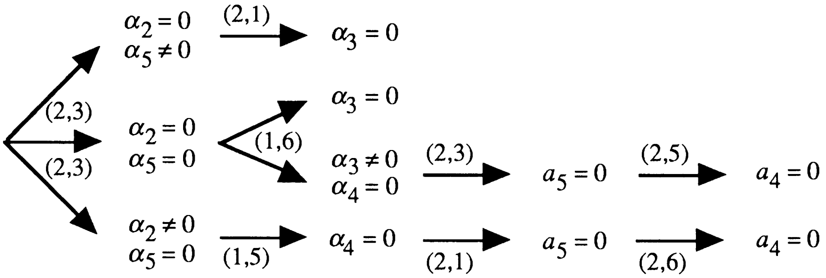


Figure 1

originate from the j-th component of V_i. We restrict considerations to non-degenerated manipulators. This rules out both paths with $\alpha_4 = 0$ and the only remaining case is $\alpha_2 = \alpha_3 = 0$. This implies $a_2 \neq 0 \neq a_3$ because the class would be degenerated otherwise. The control equations of E_3 and E_4 can also be written in the form $V_i * U = 0$ for $i \in \{3, 4\}$. Now, a similar investigation as above leads directly, without any alternative, to the conditions $\alpha_4 = a_4 = a_5 = \alpha_5 = 0$ (components (3,2), (4,1), (4,6) and (3,5) of V_3 and V_4). The resulting manipulator class is degenerated.

Even in the degenerated case extraneous factors can be prevented completely. If we eliminate θ_1 not from E_1 and E_2 as in (42) but from E_3 and E_4 a similar investigation shows that the resulting system of six equations is free of extraneous factors for this manipulator class.

It only remains to inspect the second case of the initial alternative, i. e. we assume now that at most one of the equations A and B is used for the determination of θ_3 and θ_5. In this case the control equation of the unused equation is globally solvable only if four complicated expressions that are free of pose parameters (the left-hand sides of the control equations of l_1, l_2, p_1, p_2 in (Raghavan & Roth, 1991)) are constantly zero. This very restrictive case can be ruled out with the same methods as above. Thus we have shown that a coversion to TS-form with respect to θ_4 of the system (RR-17) does not generate extraneous factors. Due to the structure of the original system (RR-17) it can be expected that x_5 cannot generate extraneous roots either. This was not checked explicitly yet.

6 Conclusions

We have investigated effects of a tangent-half-angle substituion on the solution of systems of SC-equations. It was demonstrated that a tangent-half-angle substituion and subsequent elimination of common denominators ("TS-form") can introduce nontrivial extraneous roots into SC-systems. A simple kinematical example showed how these extraneous roots can combine in resulting characteristic equations. An elementary but effective technique was presented that permits the determination of extraneous roots which are generated by a conversion to TS-form. This information can be obtained prior to the solution of the original SC-system. The concept of control systems can be used to determine certain simple, industrially relevant subclasses of a given manipulator class without knowing a symbolic solution for this class.

The results were applied to the Raghavan-Roth algorithm and it was proven that the generation of nontrivial extraneous roots by x_4 can be prevented in all cases. The example demonstrated how the use of control systems reduces the investigation of extraneous factors and essential pythagorean simplifications in very complicated systems of SC-equations to remarkable simplicity. An important means to reduce the complexity further was to delay the evaluation of the concrete equations as long as possible and to inspect only their abstract structure instead; see equations (41), (43) and (44).

References

Kovâcs, P., 1991, *Neue algebraische Verfahren für fortgeschrittene Kinematik-Expertensysteme*, Ph. D. Thesis, Techn. Univ. Berlin.

Kovâcs, P., 1993, *Rechnergestützte symbolische Roboterkinematik*, Vieweg Verlag, Wiesbaden.

Lipkin, H., Duffy, J., 1985, *A Vector Analysis of Robot Manipulators*. In: Recent advances in Robotics (Eds. Beni, G., Hackwood, S. J.), Wiley, New York, pp. 175 - 242

Paul, R. P., 1981, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge.

Raghavan, M., Roth, B., 1990, *Kinematic Analysis of the 6R Manipulator of General Geometry*. Proc. 5th Intl. Symp. Robotics Research, MIT Press, Cambridge, S. 263 - 269.

RESULTANT METHODS FOR THE INVERSE KINEMATICS PROBLEM

JÜRGEN WEISS
Institut für Informatik I, Universität Bonn *and*
GMD, Institut I1, Postfach 1316, 53731 St. Augustin, Germany
weiss@cartan.gmd.de

Abstract. We closely reconsider the mathematical tools needed for the solution of the inverse kinematics problem for $6R$ series manipulators by resultant methods. We discuss the reduction of the original problem, the homogenization of the reduced equations, and several approaches for the application of resultant methods.

1. Introduction

In recent years the application of resultant methods (dyalitic elimination) to the inverse kinematics problem of 6-degrees-of-freedom robots with 6 revolute joints gave interesting results. In particular for robots of general geometry Lee & Liang (1988a, 1988b) and Raghavan & Roth (1990) demonstrated that there is a univariate trigonometric polynomial of degree 16 in one of the joint angles, which determines all solutions of the inverse kinematics problem.

Failures of the methods when applied to some robots of special geometry make it necessary to reconsider their mathematical foundations, and to make an in depth analysis of their limitations and their range of applicability.

While we develop the material we will closely follow the paper of Raghavan & Roth (1990). Their paper contains two main results. From the closure equations they derive a system of 6 equations multilinear in three angles. By resultant methods they determine a univariate polynomial of degree 16 in one angle from this system.

First we show that the first 4 of the 6 equations given by Raghavan & Roth are uniquely extensible to a system equivalent to the closure equations (in mathematical terms they generate an elimination ideal).

Resultant methods intrinsically work with homogeneous equations only. Thus we discuss the relations between affine systems of equations and homogeneous systems. In particular we apply these considerations to trigonometric polynomials. In this context we reconsider the homogeneous substitution by the tangent of the half angle. This substitution allows us to eliminate the algebraic dependencies between sine and cosine.

The central part of the article will be a review of resultant methods and their limitations.

In the last section we apply our theory to the inverse kinematics problem. We consider robots of general and special geometry. We give examples which demonstrate some of the

J. Angeles et al. (eds.), Computational Kinematics, 41–52.

difficulties encountered. In particular we discuss an approach for handling systems with an infinite number of solutions at infinity. Finally we state some open problems.

2. Notation

We use the notation of Tsai & Morgan (1985) and Raghavan & Roth (1990) throughout this article. We describe the links of an $6R$ series manipulator by homogeneous 4×4 transformation matrices A_i relating the coordinate system of the $(i+1)^{\mathrm{st}}$ link to that of the i^{th} link.

$$A_i = \mathrm{Rot}_z(\theta_i)\,\mathrm{Trans}_z(d_i)\,\mathrm{Trans}_x(a_i)\,\mathrm{Rot}_x(\alpha_i) \tag{2.1}$$

with the translation and rotation matrices defined by

$$\begin{aligned}
\mathrm{Rot}_z(\theta_i) &= \begin{pmatrix} c_i & -s_i & 0 & 0 \\ s_i & c_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \mathrm{Trans}_z(d_i) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
\mathrm{Trans}_x(a_i) &= \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \mathrm{Rot}_x(\alpha_i) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \lambda_i & -\mu_i & 0 \\ 0 & \mu_i & \lambda_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned} \tag{2.2}$$

where

$$c_i = \cos(\theta_i) \qquad s_i = \sin(\theta_i) \qquad \lambda_i = \cos(\alpha_i) \qquad \mu_i = \sin(\alpha_i). \tag{2.3}$$

The closure equation for the $6R$ manipulator is the matrix equation

$$A_1A_2A_3A_4A_5A_6 = A_H = \begin{pmatrix} l_x & m_x & n_x & p_x \\ l_y & m_y & n_y & p_y \\ l_z & m_z & n_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.4}$$

The inverse kinematics problem consists in solving these 12 scalar equations for the variables $\theta_1, \ldots, \theta_6$.

3. Reduction to 4 equations in 3 variables

From these 12 equations (2.4) in $\theta_1, \ldots, \theta_6$, which constitute the inverse kinematics problem, we will construct a system of 4 equations in $\theta_3, \theta_4, \theta_5$, which may be uniquely extended to a system equivalent to the closure equations. This is to say that every solution of the inverse kinematics problem solves these 4 equations. And each solution $(\theta_3, \theta_4, \theta_5)$ of these 4 equations may be uniquely extended to a solution of the original system. Furthermore, we get two additional equations of the same form, such that for robots of general geometry these 6 equations are linearly independent. Naturally, these two additional equations, though linearly independent, are algebraically dependent on the first 4 equations. The 6 equations were already given by Raghavan & Roth (1990). The proof that the first 4 of them may be uniquely extended to a system equivalent to the closure equations seems to be new.

Splitting up the matrix A_2 we may equivalently write the closure equation (2.4) in the following form:

$$\mathrm{Trans}_z(d_2)\mathrm{Trans}_x(a_2)\mathrm{Rot}_x(\alpha_2)A_3A_4A_5 = \mathrm{Rot}_z^{-1}(\theta_2)A_1^{-1}A_HA_6^{-1} \tag{3.1}$$

Let $\vec{l}$ be the third column of these matrix equations, $\vec{p}$ the fourth:

$$\vec{l}_l = \vec{l}_r \qquad \vec{p}_l = \vec{p}_r \tag{3.2}$$

Both vector equations are independent of θ_6. Kovács & Hommel (1990) proved that this system of 6 scalar equations in the 5 variables $\theta_1, \ldots, \theta_5$ may be uniquely extended to a system equivalent to the closure equations.

The next step is the elimination of θ_2. We may hope that invariants under the rotation group built from $\vec{l}$ and $\vec{p}$ do not depend on some angles at all or have a polynomial dependence on c_i and s_i of low degree. The invariants are scalar and cross products of our vectors. We consider $\vec{l}$ and $\vec{p}$ of the first degree, $\vec{p}\cdot\vec{p}$, $\vec{p}\cdot\vec{l}$ and $\vec{p}\times\vec{l}$ of the second degree and the difference $(\vec{p}\cdot\vec{p})\vec{l} - 2(\vec{p}\cdot\vec{l})\vec{p}$ of the third degree in $\vec{p}$ and $\vec{l}$. The system of 6 equations formed by the scalar equations and the third components of the vector equations, that is

$$l_3 \quad p_3 \quad \vec{p}\cdot\vec{p} \quad \vec{p}\cdot\vec{l} \quad (\vec{p}\times\vec{l})_3 \quad ((\vec{p}\cdot\vec{p})\vec{l} - 2(\vec{p}\cdot\vec{l})\vec{p})_3, \tag{3.3}$$

is free of θ_2 and does not have higher degree with respect to the other variables than the equations $\vec{l}$ and $\vec{p}$. In particular the equations are linear with respect to c_1 and s_1 (Raghavan & Roth 1990). The 5 equations l_3, p_3, $\vec{p}\cdot\vec{p}$, $\vec{p}\cdot\vec{l}$, $(\vec{p}\times\vec{l})_3$ may be uniquely extended to a system equivalent to the vector equations $\vec{l}$ and $\vec{p}$ (Kovács & Hommel 1990), provided $\vec{p}\cdot\vec{p} - p_3^2 \neq 0$ or $\vec{l}\cdot\vec{l} - l_3^2 \neq 0$. The second condition is equivalent to $l_3 \neq \pm 1$. Thus the last equation $((\vec{p}\cdot\vec{p})\vec{l} - 2(\vec{p}\cdot\vec{l})\vec{p})_3$ of the system (3.3) is algebraically dependent on the first 5 equations.

The right-hand sides of some of these 6 equations (3.3) are

$$p_3: \qquad \mu_1(ps_1 - qc_1) + \lambda_1(r - d_1) \tag{3.4}$$

$$n_3: \qquad \mu_1(us_1 - vc_1) + \lambda_1 w \tag{3.5}$$

$$\vec{p}\cdot\vec{p}: \qquad -2a_1(qs_1 + pc_1) + p^2 + q^2 + r^2 + a_1^1 + d_1^2 - 2rd_1 \tag{3.6}$$

$$\vec{p}\cdot\vec{l}; \qquad -a_1(vs_1 + uc_1) + pu + qv + rw - d_1 w. \tag{3.7}$$

The constants p, q, r, and u, v, w depend on the geometry of the robot and the hand matrix only (Raghavan & Roth 1990). Given these 6 equations (3.3) linear in c_1 and s_1 we may eliminate c_1 and s_1 by Gaussian elimination. This gives 4 equations in $\theta_3, \theta_4, \theta_5$:

$$f_i(\theta_3, \theta_4, \theta_5) = 0, \qquad i = 1, \ldots, 4 \tag{3.8}$$

Given a solution to the equations (3.8) and provided that $\mu_1 \neq 0$ or $a_1 \neq 0$ and $pu - qv \neq 0$, we are able to determine c_1 and s_1 uniquely by linear equations. But we have to check that the condition $c_1^2 + s_1^2 = 1$ is satisfied. To prove that this is always the case, we observe that there is the following algebraic relation between the 6 equations (3.3).

$$(\vec{p}\cdot\vec{l})^2 + (\vec{p}\times\vec{l})^2 + \left[(\vec{p}\cdot\vec{p})\vec{l} - 2(\vec{p}\cdot\vec{l})\vec{p}\right]_3 l_3 - ((\vec{p}\cdot\vec{p}) - p_3^2) = 0 \tag{3.9}$$

The left-hand sides of the equations (3.3) satisfy this relation identically in $\theta_3, \theta_4, \theta_5$. The right-hand sides (assuming no relation between c_1 and s_1) give

$$(c_1^2 + s_1^2 - 1)\left(\mu_1^2(p^2 + q^2 + (r - d_1)^2 - (up + vq + w(r - d_1)^2) + a_1^2(u^2 + v^2)\right) = 0 \tag{3.10}$$

For most positions and orientations of the hand matrix, the second factor is not 0. So the first factor must be 0, giving the needed relation.

We have shown that the 4 equations (3.8) may be uniquely extended to a system equivalent to the inverse kinematics problem. Any solution of these 4 equations may be uniquely extended to a solution of the inverse kinematics problem. And this gives all solutions. Because the sines and cosines of the remaining variables $\theta_1, \theta_2, \theta_6$ are determined by linear equations, we further note, that any real solution of the 4 equations will extend to a real solution of the inverse kinematics problem.

These 4 equations will be multilinear in the variable sets $S_i = \{c_i, s_i\}$, $i = 3, 4, 5$ as shown by Raghavan & Roth (1990). Thus they will have a total degree of 3. From the 8 unused components of the vector equations we may derive 2 additional equations of the same form.

$$f_i(\theta_3, \theta_4, \theta_5) = 0, \qquad i = 5, 6 \tag{3.11}$$

These are necessarily algebraically dependent on the equations (3.8). But for a robot of general geometry the six polynomials are linearly independent.

4. Homogenization

Solutions of the equations (3.8) and (3.11) may be calculated by several methods. (Multivariate) resultant methods may be applied to homogeneous polynomials only. So it seems necessary to recall some results on the connection between affine and homogeneous ideals and the connection between the solutions of affine and the corresponding homogeneous equations. Proofs and further background can be found in Macaulay (1916), Zariski & Samuel (1958), Renschuch (1976), and Möller & Mora (1984).

First we define the operator ^{h} which maps polynomials to homogeneous polynomials and the inverse operator ^{a}. Let K be a field.

$$^h : K[x_1, \ldots, x_n] \to K[x_0, \ldots, x_n] \qquad {}^hf = x_0^{\deg(f)} f(x_1/x_0, \ldots, x_n/x_0) \tag{4.1}$$

$$^a : K[x_0, \ldots, x_n] \to K[x_1, \ldots, x_n] \qquad {}^aF = F(1, x_1, \ldots, x_n) \tag{4.2}$$

We note the following properties:

- $^{ah}f = f$
- if F is homogeneous and $F = x_0^t G, G \notin (x_0)$, that is, x_0 does not divide G, then $^aF =^a G$ and $^{ha}F = G$.

Next we extend these operators to ideals. For the ideal $I \subseteq K[x_1, \ldots, x_n]$ define $^hI = (^hf \mid f \in I)$. Note that the set $\{^hf \mid f \in I\}$ is not an ideal. If the ideal I is given by a basis, that is $I = (f_1, \ldots, f_r)$, we define $^*I = (^hf_1, \ldots, ^hf_r)$. $^*I$ depends on the basis chosen. Given an *affine* system of polynomial equations $f_i = 0$, $i = 1, \ldots, r$, we will call $^hf_i = 0$, $i = 1, \ldots, r$ the *homogenized* system of equations. This relation corresponds to the relation between I and $^*I$. We note some properties:

- The operator ^{h} maps the polynomials of I of degree less than or equal to d bijectively onto the polynomials of ^hI of degree d.
- $f \in I \implies \exists\, t$ such that $x_0^t\, {}^hf \in {}^*I$

- ${}^*I \subseteq {}^hI$
- $F \in {}^*I \implies {}^aF \in I$
- ${}^hI : (x_0) = {}^hI$
- ${}^*I : (x_0) = {}^*I \implies {}^*I = {}^hI$
- A basis $(f_1, \dots, f_r)$ of I is an H-basis for I, iff $\forall f \in I, \exists g_i, i = 1, \dots, r$ such that $f = \sum g_i f_i$ with $\deg(f) = \max_{1 \leq i \leq r}(\deg(g_i f_i))$. Let I be given by an H-basis, then ${}^*I = {}^hI$.

The zero set $\mathcal{V}$ of an ideal $I \subseteq K[x_1, \dots, x_n]$ is

$$\mathcal{V}(I) = \{(a_1, \dots, a_n) \in K^n \mid f(a_1, \dots, a_n) = 0, \ \forall f \in I\}. \tag{4.3}$$

For homogeneous polynomials $F(a_0, \dots, a_n) = 0$ implies $F(\lambda a_0, \dots, \lambda a_n) = 0$, $\lambda \in K$. For homogeneous ideals J we identify these points with each other and count a complete solution ray as one point in projective space, that is $V(J) \subseteq \mathcal{P}_K^n$. If the zero set $\mathcal{V}(I)$ consists of a finite number of points only, there is a bijection between these points and points of $\mathcal{V}({}^hI)$. The inclusion ${}^*I \subseteq {}^hI$ implies the reverse inclusion for the zero sets of the ideals. The zero set of ${}^*I$ may be bigger than the zero set of I; that is, it may contain additional points, which are called points at *infinity*. Depending on the basis given, the zero set $\mathcal{V}({}^*I)$ may have no, finitely many, or infinitely many additional points at infinity.

In the context of affine or homogeneous systems of equations, the zero sets $\mathcal{V}(I)$ and $\mathcal{V}({}^*I)$ of the ideals I and ${}^*I$ generated by the polynomials in the equations correspond to the solutions of the equations. The additional points at infinity in $\mathcal{V}({}^*I)$ correspond to solutions at infinity of the homogenized equations.

Given an affine ideal I by its basis $(f_1, \dots, f_r)$, we would like to work with the homogeneous ideal ${}^hI$. But this requires the construction of an H-basis of I, which is a difficult task. It will be important to find as many linearly independent elements of low degree in the ideal I as possible. We must certainly avoid the case of infinitely many solutions at infinity, because this will completely break resultant methods.

5. Trigonometric Ideals

We will apply the considerations of the last section to trigonometric polynomials and trigonometric ideals. Let us assume that the c_i, s_i, $i = 1, \dots, n$ are indeterminates. Let $T = (c_i^2 + s_i^2 - 1, \ i = 1, \dots, n)$. We will call an ideal $I \subseteq K[c_1, s_1, \dots, c_n, s_n]$ a *trigonometric* ideal if $I \subseteq T$. Equivalently we may work with $I/T \subseteq K[c_1, s_1, \dots, c_n, s_n]/T$.

We homogenize trigonometric polynomials in the following way:

$$\begin{aligned} {}^{ht} &: K[c_1, s_1, \dots, c_n, s_n] \to K[c_1, s_1, z_1, \dots, c_n, s_n, z_n] \\ f(c_1, s_1, \dots, c_n, s_n) &\mapsto z_1^{\deg_{\{c_1, s_1\}}(f)} \cdots z_n^{\deg_{\{c_n, s_n\}}(f)} f(c_1/z_1, s_1/z_1, \dots, c_n/z_n, s_n/z_n) \end{aligned} \tag{5.1}$$

The well known homogeneous substitution by the tangent of the half angle is the ring homomorphism $t_{1/2} : K[c_1, s_1, z_1, \dots, c_n, s_n, z_n] \to K[u_1, v_1, \dots, u_n, v_n]$, defined by the images of the generators

$$c_i \mapsto u_i^2 - v_i^2 \qquad s_i \mapsto 2u_i v_i \qquad z_i \mapsto u_i^2 + v_i^2. \tag{5.2}$$

The kernel of the homomorphism is ${}^{ht}T$. The image is generated by $(u_i^2, u_i v_i, v_i^2,\ i = 1, \ldots, n)$. That is, the image is the tensor product $\otimes_{i=1}^{n} \oplus_{d=0}^{\infty} K[u_i, v_i]_{2d}$. $K[u_i, v_i]_d$ denotes the K-module of homogeneous polynomials in u_i and v_i of degree d.

The image of $t_{1/2}$ is isomorphic to $K[c_1, s_1, z_1, \ldots, c_n, s_n, z_n]/{}^{ht}T$, the ring of homogenized trigonometric polynomials in n angles, by the Isomorphism Theorem. We define the inverse mapping $\otimes_{i=1}^{n} \oplus_{d=0}^{\infty} K[u_i, v_i]_{2d} \rightarrow K[c_1, s_1, z_1, \ldots, c_n, s_n, z_n]/{}^{ht}T$ by the images of $u_i^2, u_i v_i, v_i^2$

$$u_i^2 \mapsto (z_i + c_i)/2 \qquad v_i^2 \mapsto (z_i - c_i)/2 \qquad u_i v_i \mapsto s_i/2. \tag{5.3}$$

This definition of the inverse map immediately results in a simple algorithm for the conversion of trigonometric polynomials in the form $f(u, v)$ to the standard form $f(c, s)$. For example write the polynomial

$$f = au^4 + bu^3 v + cu^2 v^2 + duv^3 + ev^4 \tag{5.4}$$

in the form

$$f = a(u^2)^2 + b(u^2)(uv) + c(u^2)(v^2) + d(uv)(v^2) + e(v^2)^2. \tag{5.5}$$

Use the substitutions (5.3) and set z to 1.

Example 5.1. *The system of two linear trigonometric equations*

$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \begin{pmatrix} c \\ s \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = 0 \tag{5.6}$$

may be solved in the following ways:

- Gröbner basis method*: Add the equation $c^2 + s^2 - 1 = 0$ to the scalar equations given and calculate a Gröbner basis with respect to a lexicographic variable order.*
- direct method*: Assume that the coefficient matrix of c and s is invertible. Write*

  $$\begin{pmatrix} c \\ s \end{pmatrix} + \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}^{-1} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = 0 \tag{5.7}$$

  This allows us to calculate the solutions, provided $c^2 + s^2 - 1 = 0$, that is

  $$(b_1 d_2 - b_2 d_1)^2 + (a_1 d_2 - a_2 d_1)^2 = (a_1 b_2 - a_2 b_1)^2 \tag{5.8}$$

- resultant method*: Map the equations to*

  $$(a_i + d_i)u^2 + 2b_i uv + (d_i - a_i)v^2 = 0, \qquad i = 1, 2, \tag{5.9}$$

  and calculate the resultant of these equations. The vanishing of the resultant is necessary and sufficient for the existence of solutions. The condition is the same as the condition (5.8).

It is obvious that the Gröbner basis and the resultant approaches are valid even for nonlinear systems of equations.

6. Resultant Methods

In this section we will develop a criterion for the *non-existence* of solutions of polynomial equation systems. That this is a reasonable approach will become clear as we proceed.

Consider the ideal $I \subseteq K[x_1, \ldots, x_n]$, which is generated by $f_1, \ldots, f_r$.

$$I = \{f \mid f = \sum_{i=1}^{r} g_i f_i,\ g_i \in K[x_1, \ldots, x_n]\} \tag{6.1}$$

Now a special version of Hilbert's zeros theorem states that $1 \in I \Leftrightarrow \mathcal{V}(I) = \emptyset$. So if it is possible to write $1 = \sum_{i=1}^{r} g_i f_i$ with appropriate $g_i \in K[x_1, \ldots, x_n]$, the equations

$$f_i = 0, \qquad i = 1, \ldots, r \tag{6.2}$$

do not have any solutions. Let $d_i = \deg(f_i)$. Given $D \in \mathbb{N}$, consider the set S_D of polynomials $f \in I$ of the form $f = \sum_{i=1}^{r} g_i f_i$ with $\deg(g_i) \leq D - d_i$. If 1 is in the set S_D, 1 is in I and $\mathcal{V}(I)$ is empty.

We may view the set S_D as the image of the map

$$\Phi_D : K[x_1, \ldots, x_n]^r \to K[x_1, \ldots, x_n] \qquad (g_1, \ldots, g_r) \mapsto \sum_{i=1}^{r} g_i f_i. \tag{6.3}$$

This map is linear in the coefficients of the g_i. The image S_D is a subspace of the K-vector space with a basis which consists of all monomials of degree $\leq D$.

We have thus reduced our nonlinear problem to a linear algebraic problem. If 1 is in the image of the linear map Φ_D, the equations (6.2) do not have solutions. We will further simplify our criterion. If the map Φ_D is onto, 1 is certainly in the image, and there are no solutions.

The surjectivity of the linear map Φ_D described by an $n \times m$ matrix Φ may be checked by the following methods: Assume that $n = m$; then Φ is onto if and only if $\det(\Phi) \neq 0$. If the dimensions do not agree, the surjectivity may be verified by a method suggested by Cayley (1848). For a modern treatment see Gelfand *et al.* (1992). Another method is the calculation of the gcd of the maximal minors of Φ (Macaulay (1902, 1916, 1921)). The equivalence of the methods is shown in Kapranov *et al.* (1992).

We have shown that the surjectivity of Φ_D is sufficient, that our equations do not have solutions. But it is not necessary.

What else is needed for a necessary condition? The first problem is the degree bound D. Given d and $1 \notin S_d$, maybe $1 \in S_{d+1}$? Let us assume that the f_i are ordered in such a way that $d_1 \geq d_2 \geq \ldots \geq d_r$. If $r < n+1$ set $d_i = 1$ for $i = r+1, \ldots, n+1$. Let $D' = \sum_{i=1}^{n+1} d_i - n$. Under appropriate conditions, in particular that the homogenized system has only a finite number of solutions, Lazard (1981) shows, that $1 \notin S_{D'}$ implies $1 \notin S_d$, $d \geq D'$. For our applications this *a priori* bound will prove to be too high to be useful.

The second problem is that we test for surjectivity and not if 1 is in the image of the map Φ_D. If the map is not onto, we cannot be sure which monomial(s) is(are) not in the image. A closer inspection shows, that the absence of an arbitrary monomial in the image is a criterion for a solution of the homogenized equations. Thus, resultant methods give criteria for the existence or nonexistence of solutions for systems of homogeneous equations only. When applied to affine systems the methods treat solutions at infinity on equal footing with ordinary solutions.

We will use these considerations in the following context. Given a system of affine equations

$$f_i(x_1,\dots,x_n), \qquad i=1,\dots,r \tag{6.4}$$

with a finite number of solutions $(a_{1i},\dots,a_{ni})$, $i=1,\dots,l$. If we specialize the indeterminate x_1 to $c \in K$, the system

$$f_i(c,x_2,\dots,x_n), \qquad i=1,\dots,r \tag{6.5}$$

has solutions if and only if $c \in \{a_{1i} \mid i=1,\dots,l\}$. We may consider x_1 as a parameter and determine if the resulting system has no solutions. A sufficient condition, that no solutions exist is the surjectivity of the linear map Φ_D, which is now parametrized by x_1.

The importance of the surjectivity tests described above is that they still work, when the linear map is parametrized. Both tests give a *polynomial* R in the parameter x_1. The map as a function of x_1 is onto if and only if $R(x_1) \neq 0$.

For all specializations of the parameter c with $R(c) \neq 0$, the map is surjective and there are no solutions. Only if $R(c)=0$ there may be solutions of our system (6.4). We may factor R into irreducible components over K. All roots of an irreducible factor are algebraically equivalent. Each irreducible factor of R corresponds to

- solutions of the affine system (6.4),
- solutions at infinity, that is, solutions of the homogenized system not present in the affine system,
- or to an artefact, because the degree D, up to which we considered polynomials in our ideal, was too low.

If the homogenized system has an infinite number of solutions at infinity, R will be identically 0. And the methods discussed so far will fail.

7. Inverse Kinematics Problem

In section 3 we reduced the inverse kinematics problem to a system of four equations (3.8) in $\theta_3, \theta_4, \theta_5$. Our aim is to apply the resultant methods developed in the last section. This will necessitate the homogenization of the equations. The two additional equations (3.11), which in general are linearly independent will help us to avoid some zeros at infinity.

We treat the angle θ_3 as a parameter. Thus the result of our resultant methods will be a trigonometric polynomial in θ_3.

We call robots without algebraic relations between the Denavit-Hartenberg parameters robots of general geometry. Note, that this definition is different from the one given in Mavroidis & Roth (1992). If there are algebraic relations, we call the robot a robot of special geometry. In particular, a robot with rational values for some of the parameters is a robot of special geometry, because there will be linear equations for these parameters. For example if $d_3=5$, there will be the equation $d_3-5=0$.

7.1. Robots of General Geometry. For robots of general geometry the 6 equations (3.8) and (3.11) are linearly independent. They are still linearly independent when viewed as polynomials in θ_4, θ_5 with coefficients which are trigonometric polynomials in θ_3. The equations are linear in the 9 monomials c_4c_5, c_4s_5, s_4c_5, s_4s_5, c_4, s_4, c_5, s_5, and 1.

Let us homogenize these equations with one additional variable z. Multiply each of the six equations by c_4, s_4 and z. Using the relation $s^2 = z^2 - c^2$ we get 18 equations in the 15 monomials $c_4^2c_5$, $c_4^2s_5$, $c_4s_4c_5$, $c_4s_4s_5$, c_4^2z, c_4s_4z, c_4c_5z, c_4s_5z, s_4c_5z, s_4s_5z, c_4z^2, s_4z^2, c_5z^2, s_5z^2, and z^3.

It is not correct to choose 15 equations which are linearly independent out of these 18, because the determinant of this system will have extraneous factors. But we may apply Cayley's method to the 18 equations. The result is a trigonometric polynomial in θ_3 of degree 16.

Raghavan & Roth (1990) homogenize the angles θ_4 and θ_5 separately and apply the ring homomorphism $t_{1/2}$. Then they set v_4 and v_5 to 1. In principle this may generate singularities not present in the original equations. In practice, because they use the resulting equations as the input to a resultant method, they implicitly continue to use the homogeneous form of the equations anyway. In the following we will not set v_4 and v_5 to 1. The result of the transformations is a system of 6 equations in the 9 monomials $u_4^2u_5^2$, $u_4^2u_5v_5$, $u_4^2v_5^2$, $u_4v_4u_5^2$, $u_4v_4u_5v_5$, $u_4v_4v_5^2$, $v_4^2u_5^2$, $v_4^2u_5v_5$, and $v_4^2v_5^2$.

Multiplying these equations by u_4 and v_4 gives 12 equations in the 12 monomials $u_4^3u_5^2$, $u_4^3u_5v_5$, $u_4^3v_5^2$, $u_4^2v_4u_5^2$, $u_4^2v_4u_5v_5$, $u_4^2v_4v_5^2$, $u_4v_4^2u_5^2$, $u_4v_4^2u_5v_5$, $u_4v_4^2v_5^2$, $v_4^3u_5^2$, $v_4^3u_5v_5$, and $v_4^3v_5^2$. These 12 equations are linearly independent for robots of general geometry. The determinant of these equations is a trigonometric polynomial in θ_3 of degree 16.

7.2. Robots of Special Geometry. For robots of special geometry the methods just described may not work. Equations linearly independent for robots of general geometry may become linearly dependent.

Systems with an equation in one angle or systems with two equations in two angles only should be handled separately. In these cases there usually exist some additional polynomials of low degree in the ideal linearly independent of the ones given. If we homogenize the equations, this will lead to solutions at infinity. In the worst case there may be an infinite number of solutions at infinity, rendering our resultant methods useless. Besides, a direct treatment of such simple cases is much more efficient.

If we still want to use the second method, that is homogenizing the angular variables separately, we must consider the case that an equation does not depend on an angle at all, say θ_i. So the homogenized form of this equation is of degree 0 in the homogeneous variables u_i and v_i. Multiplying the equation by u_i^2, u_iv_i, and v_i^2, we get 3 equations homogeneous of degree 2, which we all need to consider. It is not sufficient to multiply the original equation by $u_i^2 + v_i^2$ only, as this may add solutions at infinity.

7.3. Examples.

EXAMPLE 7.1. *Let us consider a robot given by the Denavit-Hartenberg parameters*

$$(\theta_1, 0, a_1, \alpha_1), (\theta_2, d_2, 0, \alpha_2), (\theta_3, d_3, a_3, \alpha_3), (\theta_4, d_4, 0, \pi/2), (\theta_5, 0, 0, \pi/2), (\theta_6, 0, 0, 0) \quad (7.1)$$

and $a_1 = 5, \lambda_1 = 3/5, \mu_1 = 4/5, d_2 = 3, \lambda_2 = 5/13, \mu_2 = 12/13, d_3 = 7, a_3 = 11, \lambda_3 = 4/5, \mu_3 = 3/5, d_4 = 13.$

The six equations (3.8) and (3.11) considered as polynomials in θ_4, θ_5 contain the four monomials c_4s_5, s_4s_5, c_5, and 1 only. If we homogenize each angular variable separately, the resulting system of equations does not have a finite number of solutions. If we homogenize with one variable z only, we may directly apply the resultant methods to the six equations in the 4 monomials c_4s_5, s_4s_5, c_5z, and z^2. The result is a fourth degree univariate polynomial in θ_3 without extraneous factors. If we had taken the first four equations (3.8) only, which are sufficient to eliminate the four monomials, the result for θ_3 would have developed an additional factor of degree 4, which does not correspond to any solutions of the affine system.

EXAMPLE 7.2. *This example demonstrates, that for some robots it is not sufficient to consider polynomials of degree 3 in u_4 and v_4 and of degree 2 in u_5 and v_5.*

$$(\theta_1, 0, 0, \alpha_1), (\theta_2, d_2, a_2, \alpha_2), (\theta_3, d_3, 0, \pi/2), (\theta_4, 0, 0, \pi/2), (\theta_5, d_5, a_5, \alpha_5), (\theta_6, 0, 0, 0) \quad (7.2)$$

with $\lambda_1 = 3/5, \mu_1 = 4/5, d_2 = 5, a_2 = 3, \lambda_2 = 5/13, \mu_2 = 12/13, d_3 = 7, d_5 = 11, a_5 = 13, \lambda_5 = 4/5, \mu_5 = 3/5$.

From the six equations (3.8) and (3.11) only 5 are linearly independent in $\theta_3, \theta_4, \theta_5$. The methods of Raghavan & Roth (1990) lead to 10 linearly independent equations in 12 monomials. So the resultant methods are not applicable at these degrees. If we multiply the 5 linearly independent equations by u_4u_5, u_4v_5, v_4u_5, and v_4v_5 instead of u_4 and v_4 only we will get 20 equations in 16 monomials. These give a polynomial in θ_3 of degree 8 without extraneous factors. This solution cannot be obtained by considering polynomials of lower degree only. This effect may even occur when the original 6 equations are linearly independent.

EXAMPLE 7.3. *Here we give an example, where our resultant methods fail completely.*

$$(\theta_1, 0, a_1, \alpha_1), (\theta_2, d_2, 0, \alpha_2), (\theta_3, 0, a_3, \alpha_3), (\theta_4, d_4, 0, \alpha_4), (\theta_5, 0, 0, \alpha_5), (\theta_6, 0, 0, 0) \quad (7.3)$$

with $a_1 = 3, \lambda_1 = 3/5, \mu_1 = 4/5, d_2 = 7, \lambda_2 = 5/13, \mu_2 = 12/13, a_3 = 7, \lambda_3 = 4/5, \mu_3 = 3/5, d_4 = 5, \lambda_4 = 8/17, \mu_4 = 15/17, \lambda_5 = 12/13, \mu_5 = 5/13$.

From the six equations (3.8) and (3.11) only 4 are linearly independent in θ_4, θ_5. Only the monomial s_5 is missing in the equations, when compared to the general case. The homogenized system always has an infinite number of solutions at infinity. A close inspection of the ideal by Gröbner basis methods reveals that we have not identified all polynomials of low degree, which are linearly independent. So our resultant methods fail.

7.4. An Infinite Number of Solutions at Infinity. As demonstrated by Example 7.3, resultant methods may fail due to an infinite number of solutions at infinity. This indicates, that there are polynomials of low degree in our ideal, which we were not able to identify. This problem is unrelated to the problem addressed in Example 7.2, though both problems manifest themselves by almost the same symptoms. In both cases we do not have enough linearly independent equations to eliminate all monomials.

How could we hope to handle systems with an infinite number of solutions in our framework? If there is only a finite number of solutions, we find the minimal degree at which we can apply the resultant methods by the condition that we need as many linearly independent equations as we have monomials (see Example 7.2). If the equations have an infinite number of solutions, we do not have as many linearly independent equations as we have monomials for any degree. Thus, we may not constructively determine this bound. Unfortunately, the *a priori* bound given in Section 6 is too high to be useful.

If we knew an appropriate bound D, we could examine our parametrized linear map Φ_D. Because there is an infinite number of solutions, the map will not be onto. But in a matrix representation of the map the ranks of *all* minors of maximal rank will drop for some specializations of the parameter. Basically this seems to be the idea of Canny (1990).

A similar approach is suggested in Mavroidis & Roth (1992). But their methods exhibit two problems. They use the degree bound obtained from robots of general geometry. In general it seems not to be sufficient to look at one minor of maximal rank only.

7.5. Open Problems. At the end of section 6 we have shown that resultant methods may give extraneous factors which correspond to solutions at infinity or even to artefacts of the method. In practice it seems that factors corresponding to artefacts of the method occur very rarely if they occur at all. There should be a theoretical reason for this.

On theoretical grounds this question seems to be closely related to the degree bound for systems with an infinite number of solutions discussed in the last subsection.

If more equations than monomials are given, there may be circumstances when a judicious choice of equations may prove it unnecessary to apply Cayley's method at all, reducing the problem to a single determinant.

If no general answers can be obtained, constructive criteria would be helpful. Integrated into implementations of the algorithms they could prevent the application of solution methods inappropriate for a specific robot.

8. Conclusions

Resultant methods are an important tool for solving the inverse kinematics problem. They map the problems of systems of nonlinear equations to linear algebraic problems. This allows us to use fast and well understood algorithms for handling matrices and determinants. The general drawbacks of resultant algorithms may be mostly avoided by the proper use of specific knowledge about the ideals involved. Therefore, they seem to be better suited for the inverse kinematics problem than Gröbner basis techniques.

9. Acknowledgements

This work is supported by the Deutsche Forschungsgemeinschaft under grant Kr 393/4-1.

References

J. Canny (1990), Generalised characteristic polynomials. *Journal of Symbolic Computation* 9, 241–250.

A. Cayley (1848), On the theory of elimination. *Cambridge Dublin Mathematical Journal* 3, 116–120. reprinted in "Collected Papers", Vol. 1, No. 59, pp. 370–374, Cambridge University Press, London/New York, 1889.

I. M. Gelfand, M. M. Kapranov, and A.V. Zelevinsky (1992), Hyperdeterminants. *Advances in Mathematics* **96**, 226–263.

M. M. Kapranov, B. Sturmfels, and A. V. Zelevinsky (1992), Chow polytopes and general resultants. *Duke Mathematical Journal* **67**, 189–218.

P. Kovács and G. Hommel (1990), Reduced equation systems for the inverse kinematics problem. Forschungsbericht 1990–20, Technische Universität Berlin, Fachbereich 20.

D. Lazard (1981), Resolution des systèmes d'èquations algebriques. *Theoretical Computer Science* **15**, 77–110.

H. Y. Lee and C. G. Liang (1988a), A new vector theory for the analysis of spatial mechanisms. *Mechanism and Machine Theory* **23**, 209–217.

H. Y. Lee and C. G. Liang (1988b), Displacement analysis of the general spatial 7-link 7R mechanisms. *Mechanism and Machine Theory* **23**, 219–226.

F. S. Macaulay (1902), Some formulæ in elimination. *Proceedings of the London Mathematical Society* **35**, 3–27.

F. S. Macaulay (1916), *The Algebraic Theory of Modular Systems.* Cambridge University Press, Cambridge.

F. S. Macaulay (1921), Note on the resultant of a number of polynomials of the same degree. *Proceedings of the London Mathematical Society* **21**, 14–21.

C. Mavroidis and B. Roth (1992), Structural parameters which reduce the number of manipulator configurations. In *Robotics and Spatial Mechanisms, and Mechanical Systems. Proc. ASME 22nd Biennial Mechanisms Conference*, ed. G. Kinzel et al., DE-vol. 45, 359–366

H. M. Möller and F. Mora (1984), Upper and lower bounds for the degree of groebner bases. In *Eurosam 84*, ed. John Fitch, vol. 174 of *Lecture Notes in Computer Science*, Berlin-Heidelberg-New York, Springer-Verlag, 172–183.

M. Raghavan and B. Roth (1990), Kinematic analysis of the 6R manipulator of general geometry. In *Proc. of* 5^{th} *International Symposium of Robotics Research*, ed. H. Miuara and S. Arimoto. MIT Press, 263–269.

B. Renschuch (1976), *Elementare und praktische Idealtheorie.* VEB Deutscher Verlag der Wissenschaften, Berlin.

L-W. Tsai and A. Morgan (1985), Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods. *Transactions of the ASME, Journal of Mechanisms, Transmissions, and Automation in Design* **107**, 189–200.

O. Zariski and P. Samuel (1958), *Commutative Algebra*, vol. I/II. D. van Nostrand Company, New York, London, Toronto.

Part 2

Redundant Manipulators

REDUNDANCY RESOLUTION FOR AN EIGHT-AXIS MANIPULATOR

H. Heiß, BMW, ET-203, 80788 München, Germany

Abstract A method allowing closed form solutions of the joint variables (inverse kinematic problem) has been developed for a man-like kinematic structure. Moreover, the procedure takes into account the available joint range, thus opening up new ways towards intelligent optimization strategies.

Keywords man-like kinematic structure; redundant robots; explicit backwards solution; mechanical variables control; computer aided simulation; automation.

INTRODUCTION

Robots with 7 or 8 degrees of freedoms, which require an efficient control are currently under development, or even in use [Karlen, 1989; Karlen, 1990].

Moreover, there is great potential not only for increasing the efficiency of manual work operations, but also for optimizing the design of workshop places under ergonomic aspects in order to reduce the burden on the workers. It is for this reason that the Institut für Montageautomatisierung (ifm) has developed a simulation system for the 3-dimensional graphic planning of both the layout of the work cells and the sequence of manual operations. This system, which is called COSIMAN (COmputer aided SImulation of MANual assembly), is implemented on a microVAX II from DEC and draws on the capabilities of the graphic terminal PS390 from Evans&Sutherland.

In order to enable the assembly planner to simulate the sequence of manual operations of a human worker, it was necessary to design a kinematic model of man. After modelling in a CAD system, the different geometrical elements of the model were linked together, defining 40 joints (see Fig. 1). Each single joint could be moved using the dials of the PS390. The kinematic model described in Table 1 and Fig. 2 is the arm part of this man model without the three gripper axes, which simulate the fingers of a hand.

This paper thus deals with a kinematic model with 8 rotational joints, arranged as follows:
rotational joint – spherical joint – rotational joint – spherical joint

J. Angeles et al. (eds.), Computational Kinematics, 55–66.

DESCRIPTION OF THE PROBLEM AND THE METHOD OF SOLUTION

Prior to the continuous simulation of the series of manual operations by the robot or worker each individual position adopted by the man model had to be taught to the robot by the assembly planner (similar to the "Teach-In" of robots). Up until now this procedure has always required a lot of practice because the planner can only change the values of the different joints using the dials of the PS390. For example the planner has to change and coordinate the values of up to eight joints of the hand-arm-system, in order to move one of the man's hands to a new position. It is thus very difficult and time-consuming – in some cases nearly impossible – to place, for example, an object grasped by one of the man's hands in a position where one of its planes is coplanar to another object's plane. It is particularly these placement functions, however, which are very often necessary when simulating the assembly of several components. To reduce the time needed for the planning, therefore, it was necessary to simplify the method of teaching new positions. This meant realizing the possibility of moving the man model's hand both in the direction of the axes of a system of coordinates, and additionally by adopting elementary placement functions (like point-to-point, plane-to-plane and other CAD-functionality).

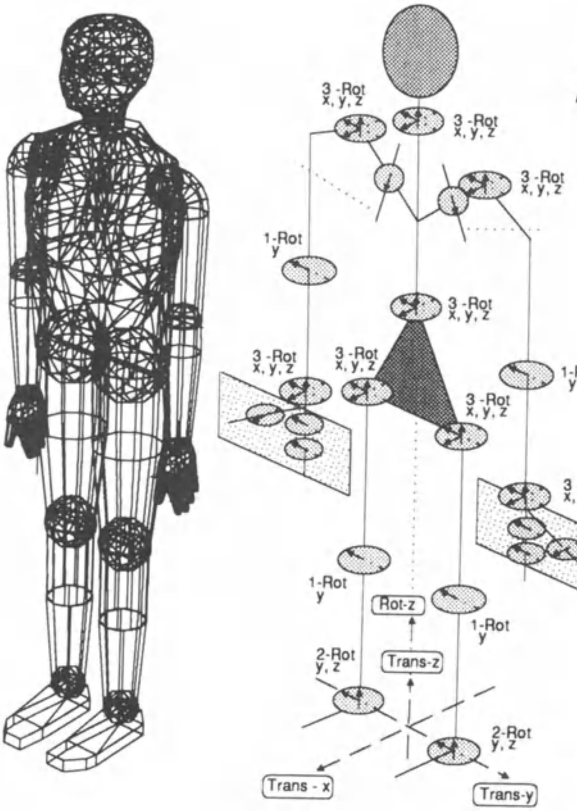


Fig. 1: Volume model of a human worker and its affiliated kinematical structure

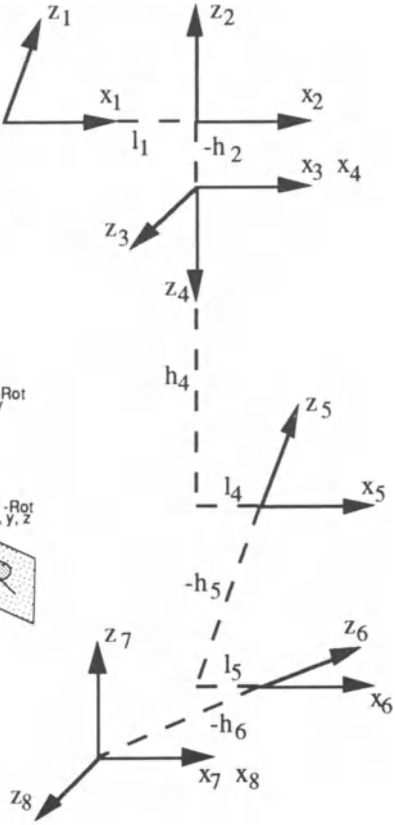


Fig. 2: D-H-coordinate systems of the kinematic model of the arm

In both cases the basic requirement was to develop a backwards solution for the eight rotational joints of the man model's hand-arm-system. This meant calculating the values for these axes at a given position and orientation of the hand. Because of the on-line visualisation in the simulation system COSIMAN, it was necessary for the solution to be computed as quickly as possible. The same is true for a robot control which processes cartesian coordinates. It is for this reason that we do not employ differential methods (using velocity information) to solve the inverse kinematics, but an explicit solution of the eight joint variables [Heiß, 1986a]: θ_i = f(<values of aim pose>)

TABLE 1 The Denavit-Hartenberg description of the model

θ_i (Rotation about z)	d_i (Translation along z)	a_i (Translation along x)	α_i (Rotation about x)
θ_1 (variable)	0	a_1	α_1
θ_2 (variable)	d_2	0	90°
θ_3 (variable)	0	0	90°
θ_4 (variable)	d_4	a_4	α_4
θ_5 (variable)	d_5	a_5	α_5
θ_6 (variable)	d_6	0	90°
θ_7 (variable)	0	0	90°
θ_8 (variable)	0	0	0

Another reason for an explicit solution formula is that we want to hold all joint values within their mechanical range. As a universal movement in space has six degrees of freedom, two of the eight joint variables can be defined by the user. For the remaining six joint variables it is necessary not only to find a closed solution formula, but also to take into account the mechanical – that means in this case also the anthropometrical – joint range. A result of this is that the range of the two user-defined variables will be further reduced so that the other joint values – computed by the solution formula – are within their mechanical range.

PREPARATIONS

Basics of the robot kinematic

The D-H-Parameters shown in Table 1 lead to the basic kinematic equation

$A_{1,2}\bullet A_{2,3}\bullet A_{3,4}\bullet A_{4,5}\bullet A_{5,6}\bullet A_{6,7}\bullet A_{7,8}\bullet A_{8,9} = T$, in which the parameters d_1, d_8, a_8, α_8 and the tool transformation TR are integrated in the given aim pose [Heiß, 1987] and denoted by T. $A_{i,i+1}=Z_i\bullet X_{i+1}$ is the well-known "Denavit-Hartenberg matrix", as described, for example, by [Heiß, 1987]. Application of the method of the characteristic joint pair [Woernle, 1988] and modification of the basic equation by inverting the outer rotation-translation-matrices Z_i and X_{i+1} yield suitable single equations for solving the joint variables [Heiß, 1986c]. Because our kinematic structure has only rotational joints, we have attempted to find an equation "a*sin(θ)+b*cos(θ)=c" which can be solved as follows:

$$\theta = \mathrm{ATAN2}(c\ ,\ \pm\sqrt{a^2+b^2-c^2}) + \mathrm{ATAN2}(-b\ ,\ a).$$

ATAN2(0,0) is a special case, indicating that the variable θ vanishes from the equation, thus rendering the equation correct for all θ-values, or never in the case of $c \neq 0$. We can make a distinction between mathematical attainability and mechanical joint range [Heiß, 1986b]. Unattainability is expressed by insoluble equations, an effect which can be put down to the existence of a negative argument under a square root in the formula (see $a^2+b^2-c^2$). Result values θ out of the joint range, however, cannot be seen in the structure of the formula.

Definition of the term "free variable"

Because it has eight degrees of freedom, the given aim pose of the robot/man model can be achieved by varying two joint variables in an application specific way. These variables are called "free variables". For the above described kinematical structure one of them is interpreted as a distance related degree of freedom and will be represented by θ_1 or θ_5. This free variable has an influence on all other joint variables. Due to the special kinematic with two spherical joints, the other free variable affects only the variables of the two spherical joints. This degree of freedom rotates about the axis through the two spherical joints (see Fig. 3) and is expressed by one of the joint variables θ_2, (θ_3,) θ_4, θ_6 (, θ_7) or θ_8.

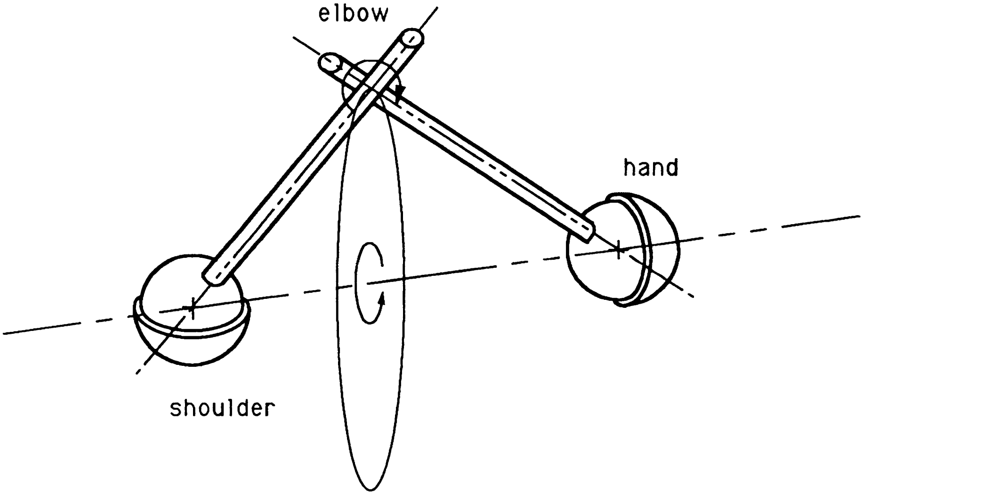


Fig. 3: Elbow rotation

SOLUTIONS FOR θ_1 TO θ_5

Use of the appropriate kinematic equation (I) provides us with the possibility of defining the two free variables and of solving the variables θ_1, θ_2, θ_3, θ_4 and θ_5.

$$A_{3,4} \bullet A_{4,5} \bullet A_{5,6} \bullet A_{6,7} \bullet Z_7 = (A_{1,2} \bullet A_{2,3})^{-1} \bullet T \bullet (X_8 \bullet A_{8,9})^{-1} \qquad (I)$$

Deducing θ_1 and θ_5 from distance equation

I1.4 denotes the single equation at position 1.4 (first row, fourth column) of the matrix equation (I) and the distance equation is defined by $I1.4^2+I2.4^2+I3.4^2$. This distance equation gives us $a_5*\sin(\theta_5) + b_5*\cos(\theta_5) + k_{15} = a_1*\sin(\theta_1) + b_1*\cos(\theta_1)$ (1)

Further details of the principles of this method can be found in [Heiß, 1986c].

θ_1 as free variable Defining $c_5:=a_1*\sin(\theta_1) + b_1*\cos(\theta_1) - k_{15}$,

$$\theta_5 = \text{ATAN2}(c_5 , \pm\sqrt{a_5^2+b_5^2-c_5^2}) + \text{ATAN2}(-b_5 , a_5) \qquad (2)$$

θ_5 as free variable Defining $c_1:=a_5*\sin(\theta_5) + b_5*\cos(\theta_5) + k_{15}$,

$$\theta_1 = \text{ATAN2}(c_1 , \pm\sqrt{a_1^2+b_1^2-c_1^2}) + \text{ATAN2}(-b_1 , a_1) \qquad (3)$$

Choosing θ_1 or θ_5 as a free variable and fixing its value as described in a later section ("Choice of the free variables") provides us with a user-defined and a deduced (from formula) value for θ_1 and θ_5. These values are the basis for the further computations.

Deducing θ_2 and θ_4 from the "single position equation"

With help of the position method using I3.4 [Heiß, 1986c] we get

$$a_2*\sin(\theta_2) + b_2*\cos(\theta_2) = a_4*\sin(\theta_4) + b_4*\cos(\theta_4) \qquad (4)$$

With $c_2:=a_4*\sin(\theta_4) + b_4*\cos(\theta_4) \Rightarrow c_2 = \sqrt{a_4^2+b_4^2}*\sin(\theta_4-\varphi)$

and $c_4:=a_2*\sin(\theta_2) + b_2*\cos(\theta_2) \Rightarrow c_4 = \sqrt{a_2^2+b_2^2}*\sin(\theta_2-\psi)$ follows

$$\theta_2 = \text{ATAN2}(c_2 , \pm\sqrt{a_2^2+b_2^2-c_2^2}) + \text{ATAN2}(-b_2 , a_2) \qquad (5)$$

or $$\theta_4 = \text{ATAN2}(c_4 , \pm\sqrt{a_4^2+b_4^2-c_4^2}) + \text{ATAN2}(-b_4 , a_4) \qquad (6)$$

As discussed below, the smaller amplitude $\sqrt{a_i^2+b_i^2}$ defines the free variable.

Unique solution of θ_3

From I1.4 and I2.4 we have I1.4: $b_3*\sin(\theta_3) - a_3*\cos(\theta_3) = e_3$ (7)

I2.4: $a_3*\sin(\theta_3) + b_3*\cos(\theta_3) = c_3$ (8)

and therefore $\theta_3 = \text{ATAN2}(a_3*c_3+b_3*e_3,b_3*c_3-a_3*e_3)$ always exists [Heiß, 1985]. (9)

SOLUTIONS FOR θ_6, θ_7 AND θ_8

In order to solve the joint variables θ_6, θ_7 and d_8 we use the orientation method [Heiß, 1986c] and choose the kinematic equation

$$A_{7,8}\bullet A_{8,9} = A_{6,7}^{-1}\bullet(A_{1,2}\bullet A_{2,3}\bullet A_{3,4}\bullet A_{4,5}\bullet A_{5,6})^{-1}\bullet T \qquad (II)$$

With $K:=(A_{1,2}\bullet A_{2,3}\bullet A_{3,4}\bullet A_{4,5}\bullet A_{5,6})^{-1}\bullet T$ and $Konf\theta_6 \in \{-1,1\}$ we get the formulae:
when $K_{13}=0$ and $K_{23}=0$

then θ_6 = an arbitrary value; (10)

$\theta_7 = ATAN2(0 , -K_{33})$; (11)

$\theta_8 = ATAN2(K_{11}*\sin(\theta_6)-K_{21}*\cos(\theta_6) , K_{12}*\sin(\theta_6)-K_{22}*\cos(\theta_6))$ (12)

According to the later described algorithm for defining the permissible joint range, θ_6 has to be restricted with respect to θ_8 in order to avoid a θ_8-value out of the permissible range.

otherwise $\theta_6 = ATAN2(Konf\theta_6*K_{23} , Konf\theta_6*K_{13})$ (13)

$\theta_7 = ATAN2(Konf\theta_6*\sqrt{K_{13}^2+K_{23}^2} , -K_{33})$ (14)

$\theta_8 = ATAN2(-Konf\theta_6*K_{32} , Konf\theta_6*K_{31})$ (15)

OBSERVATIONS ON THE JOINT RANGE

Basic considerations

Viewing the joints from a mathematical perspective means that rotational joints with a range of 360° and translational joints with an unlimited range have to be assumed. Based on this assumption we can identify a variable of the spherical joints which – moving over its full range of 360° – does not produce insoluble equations for the other variables. This can be interpreted geometrically as the 360°-rotation of the "elbow" around the axis through the two spherical joints (see Fig. 3). The variable depends on the given aim pose and cannot therefore be determined in advance, but has to be integrated into the solution algorithm.

The mechanical joint range is even more important than the mathematical view, as a result of which the joint range of the free variable has to be connected with the accompanying values of the other joint variables. If the dependent joint variables run out of their range, the range D_{free} of the free variable has to be restricted in such a way, that all the dependent joint variables lie within their mechanical range.
That means: $f_{backsol}(D_{free}) \subseteq MR_{dependent}$
For this purpose it is necessary to find "inverse" functions f_i: $\theta_i \rightarrow \theta_{free}$ and project the mechanical range MR_i of θ_i to the definition range of θ_{free}:

$$D_{free} = \cap f_i(MR_i) \cap MR_{free}$$

Then it is possible to find for each value from D_{free} a backwards solution variant, which projects the result value in the permissible mechanical range of the calculated joint.

Algorithm to define the permissible joint range of the both free variables

Defining the "inverse" functions

The distance equation (1) and the equation (4) are both suitable for the generation of solution formulae (function) and computations of range limits (inverse function). The third inverse function, necessary for the range definition of the free variable $\theta_{free} \in \{\theta_2, \theta_4\}$ by θ_3, will also be derived from the kinematic equation I.

In order to simplify the computation process, the whole kinematic equation will be inverted using the kinematic symmetry between the two spherical joints.

$$(A_{3,4}\bullet A_{4,5}\bullet A_{5,6}\bullet A_{6,7}\bullet Z_7)^{-1} = X_8\bullet A_{8,9}\bullet T^{-1}\bullet A_{1,2}\bullet A_{2,3} \qquad \text{(III)}$$

This allows interchange between the two spherical joints and enables the variables θ_6, θ_7 and θ_8 to be dealt with in an analogous way, like θ_2, θ_3 and θ_4. Therefore the variable $\theta_{pseudofree}$ is defined as an element of $\{\theta_6, \theta_8\}$. Because there is no degree of freedom for the (pseudo-)free variable from θ_6, θ_7 and θ_8, this variable has to be determined by the kinematic equation and the second free variable has to take into consideration the permissible joint range of the pseudo-free variable. The advantage of this procedure is that the limits of θ_6, θ_7 and θ_8 can be applied to the pseudo-free variable [Kiener, 1991], meaning that it is only necessary to connect the pseudo-free variable with the free variable by a further inverse function. This function arises from

$$A_{3,4}\bullet Z_4 = A_{2,3}^{-1}\bullet A_{1,2}^{-1}\bullet T\bullet(X_5\bullet A_{5,6}\bullet A_{6,7}\bullet A_{7,8}\bullet A_{8,9})^{-1} \qquad \text{(IV)}$$

Joint range computation

The inverse functions f_i from (1) and (4) can be described by $^{\pm}F_{free}^{-1}\bullet F_{calc}$: $D_{calc} \rightarrow \Re$.

$$F_{calc}(\theta_{calc})=c_{free}(\theta_{calc}) \qquad ^{\pm}F_{free}^{-1}(x)=ATAN2(x,\pm\sqrt{a_{free}^2+b_{free}^2-x^2})+ATAN2(-b_{free},a_{free})$$

To obtain efficient computation of the set $f_i(MR_i)$ we use the property of monotony. Dividing D_{calc} into the two sets $^{+}B_{calc}$ and $^{-}B_{calc}$, we can prove the following statement:

$^{+}B_{calc}$ is the set of mathematical solutions of θ_{calc} from an equation with "$+\sqrt{\ldots}$"

$^{-}B_{calc}$ is the set of mathematical solutions of θ_{calc} from an equation with "$-\sqrt{\ldots}$"

a) F_{calc}: $^{+/-}B_{calc} \rightarrow \Re$ is monotone

b) $^{+/-}F_{free}^{-1}$ is monotone for all soluble values $x \in F_{calc}(^{+/-}B_{calc})$

Because we do this computation in order to restrict the free variable to the mechanical range of θ_{calc}, it is not important, that there are some values θ_{calc} with $\theta_{calc} \notin {}^{+/-}B_{calc}$ which cannot be attained from θ_{free}. Based on this restriction to $^{+/-}B_{calc}$, the requirement "$D_{free} \subseteq MR_{free} \cap \{\theta_{free} \mid \theta_{calc}=f_{backsol}(\theta_{free})$ is soluble$\}$" implicitly holds true for D_{free} and does not have to be specifically guaranteed.

Proof:

Differentiating $^{+/-}F_{free}^{-1}(x)$ results in $^{+/-}F'_{free}{}^{-1}(x) = 1 / (\pm\sqrt{a_{free}^2+b_{free}^2-x^2})$ and this confirms statement b).

Dealing with a) we need knowledge about $^{+/-}B_{calc}$. Later in this paragraph you find from (24) and (25)

$$^{+}B_i \subseteq [ATAN2(-b_i,a_i)-\pi/2\ ,\ ATAN2(-b_i,a_i)+\pi/2] = [ATAN2(-a_i,-b_i)\ ,\ ATAN2(a_i,b_i)]$$

$$^{-}B_i \subseteq [ATAN2(-b_i,a_i)+\pi/2,\ ATAN2(-b_i,a_i)+3*\pi/2] = [ATAN2(a_i,b_i)\ ,\ ATAN2(-a_i,-b_i)]$$

Differentiating $F_{calc}(\theta_{calc})$ gives us

$$F'_{calc}(\theta_{calc}) = a_{calc}*\cos(\theta_{calc}) - b_{calc}*\sin(\theta_{calc})$$
$$= \sqrt{a_{calc}^2+b_{calc}^2} *\sin(\theta_{calc}-\varphi), \quad \varphi=ATAN2(-a_{calc},-b_{calc})$$

Therefore $F_{calc}(\theta_{calc})$ is monotone for $\theta_{calc}\in [ATAN2(-a_{calc},-b_{calc}),ATAN2(a_{calc},b_{calc})]$
and $\theta_{calc}\in [ATAN2(a_{calc},b_{calc}),ATAN2(-a_{calc},-b_{calc})]$
and consequently statement a) is proved true.

${}^{++}D_{calc\to free}$ is a set of θ_{free}-values, calculated from ${}^{+}B_{calc}\cap MR_{calc}$ by means of ${}^{+}F_{free}^{-1}$
${}^{-+}D_{calc\to free}$ is a set of θ_{free}-values, calculated from ${}^{-}B_{calc}\cap MR_{calc}$ by means of ${}^{+}F_{free}^{-1}$
${}^{+-}D_{calc\to free}$ is a set of θ_{free}-values, calculated from ${}^{+}B_{calc}\cap MR_{calc}$ by means of ${}^{-}F_{free}^{-1}$
${}^{--}D_{calc\to free}$ is a set of θ_{free}-values, calculated from ${}^{-}B_{calc}\cap MR_{calc}$ by means of ${}^{-}F_{free}^{-1}$

Thus: $D_{free} = ({}^{++}D_{calc\to free} \cup {}^{+-}D_{calc\to free} \cup {}^{-+}D_{calc\to free} \cup {}^{--}D_{calc\to free}) \cap MR_{free}$

θ_1 and θ_5 will be considered in isolation of the other variables due to the fact that there is another second free variable to guarantee the solubility of the kinematic system. Only in the case of a degeneration of the permissible range of the second free variable does the determination of θ_1 or θ_5 and of an appropriate value become necessary.

The second free variable θ_2 or θ_4, however, has to take into account the variable θ_3. We got the value of θ_3 from equation (9): $\theta_3 = ATAN2(a_3*c_3+b_3*e_3 , b_3*c_3-a_3*e_3)$
But there is a more suitable equation for the range computation; [Kiener, 91] has shown, that from equation "$-e_3*\sin(\theta_3)-c_3*\cos(\theta_3)=-b_3$" (when $\theta_{free}=\theta_2$) (16)
or from equation (8) (when $\theta_{free}=\theta_4$) θ_3 can be computed. (17)
The main advantage is that from this equation "$f_{free}(\theta_{free})*\sin(\theta_3) + k*\cos(\theta_3) = m$" the inverse function f_3: $\theta_3 \to \theta_{free}$ can be derived and the ± ambiguity of B_{calc} is strictly connected to the ± solution of θ_3.

Thus the permissible range D_{free} of the second free variable is defined by

$$D_{+B\to free} = ({}^{++}D_{calc\to free} \cup {}^{-+}D_{calc\to free}) \cap ({}^{++}D_{3\to free} \cup {}^{-+}D_{3\to free})$$

$$D_{-B\to free} = ({}^{+-}D_{calc\to free} \cup {}^{--}D_{calc\to free}) \cap ({}^{+-}D_{3\to free} \cup {}^{--}D_{3\to free})$$

$$D_{free} = (D_{+B\to free} \cup D_{-B\to free}) \cap MR_{free}$$

Now we have to deal with the computation of $^{\cdots}D_{3\rightarrow free}$. Again the inverse function f_3 will be transformed into $F_{free}^{-1} \bullet F_3$: $\theta_3 \rightarrow d_{free}$

$F_3(\theta_3) = (m\text{-}k*\cos(\theta_3))/\sin(\theta_3)$

$F_{free}^{-1}(x) = \text{ATAN2}(\ x\ , \pm\sqrt{a_{free}^2+b_{free}^2\text{-}x^2}\ ) + \text{ATAN2}(\ \text{-}a_{free}\ , \text{-}b_{free}\ )$

$F_{free}^{-1}(x)$ is confirmed as monotone by statement b) and only the following statement has to be proved:

c) F_3: $]\text{-}\pi,\pi[- \{0\} \rightarrow \Re$ is strictly monotone

$F'_3(\theta_3) = (k\text{-}m*\cos(\theta_3))/\sin^2(\theta_3)$ and when $k^2 \geq m^2$ the statement c) holds true; the subdivision into ${}^{-}S_3=[\text{-}\pi,0]$ and ${}^{+}S_3=[0,\pi]$ is congruent with ${}^{+}B_i$ and ${}^{-}B_i$.

I1.4: $b_3*\sin(\theta_3) - a_3*\cos(\theta_3) = e_3$

I2.4: $a_3*\sin(\theta_3) + b_3*\cos(\theta_3) = c_3$

I3.4: $a_4*\sin(\theta_4) + b_4*\cos(\theta_4) = a_2*\sin(\theta_2) + b_2*\cos(\theta_2)$

The distance equation "$1.4^2+2.4^2+3.4^2$" from (I) gives us
with $a_3=\text{-}b_4*\sin(\theta_4)+a_4*\cos(\theta_4)$ and $e_3=b_2*\sin(\theta_2)\text{-}a_2*\cos(\theta_2)$
the equivalence "$a_4^2+b_4^2+b_3^2 = a_2^2+b_2^2+c_3^2$".

When $\theta_{free}=\theta_2$,
then $k=\text{-}c_3$; $m=\text{-}b_3$; $k^2\text{-}m^2=c_3^2\text{-}b_3^2=a_4^2+b_4^2\text{-}(a_2^2+b_2^2)$ and
with the condition "$a_i^2+b_i^2 \geq a_{free}^2+b_{free}^2$" for the second free variable
$k^2\text{-}m^2 \geq 0$ is proved true. (18)

When $\theta_{free}=\theta_4$,
then $k=b_3$; $m=c_3$; $k^2\text{-}m^2=b_3^2\text{-}c_3^2=a_2^2+b_2^2\text{-}(a_4^2+b_4^2) \geq 0$ (19)
according to the condition "$a_i^2+b_i^2 \geq a_{free}^2+b_{free}^2$".

- Calculation of the limits of ${}^{+/-}B_i$

There are two types of equation which are responsible for ${}^{+/-}B_i$:

$\theta_i = \text{ATAN2}(c_i,\pm\sqrt{a_i^2+b_i^2\text{-}c_i^2})+\text{ATAN2}(\text{-}b_i,a_i)$ with $c_i:=a_j*\sin(\theta_j)+b_j*\cos(\theta_j)+k$ (20)

and

$f_{free}(\theta_{free})*\sin(\theta_i)+k*\cos(\theta_i)=m$ (21)

For equation (20) it holds true

$$[UG,OG] =[-\sqrt{a_j^2+b_j^2} +k , +\sqrt{a_j^2+b_j^2} +k] \cap [-\sqrt{a_i^2+b_i^2} , +\sqrt{a_i^2+b_i^2}]$$

$$\begin{aligned} {}^{+}B_i = [&ATAN2(UG , +\sqrt{a_i^2+b_i^2-UG^2})+ATAN2(-b_i , a_i) , \\ &ATAN2(OG , +\sqrt{a_i^2+b_i^2-OG^2})+ATAN2(-b_i , a_i)] \end{aligned} \qquad (22)$$

$$\begin{aligned} {}^{-}B_i = [&ATAN2(OG , -\sqrt{a_i^2+b_i^2-OG^2})+ATAN2(-b_i , a_i) , \\ &ATAN2(UG , -\sqrt{a_i^2+b_i^2-UG^2})+ATAN2(-b_i , a_i)] \end{aligned} \qquad (23)$$

(20) leads to $^{+}B_i \subseteq [ATAN2(-b_i,a_i)-\pi/2 , ATAN2(-b_i,a_i)+\pi/2]$ (24)

$^{-}B_i \subseteq [ATAN2(-b_i,a_i)+\pi/2 , ATAN2(-b_i,a_i)+3*\pi/2]$ (25)

For (21) we define: $^{\pm}f_i^{-1} \bullet f_{free} : \theta_{free} \rightarrow \theta_i$

$$f_{free}(\theta_{free}) = -b_{free}*\sin(\theta_{free}) + a_{free}*\cos(\theta_{free}) \ (=x)$$

$$^{\pm}f_i^{-1}(x) = ATAN2(m , \pm\sqrt{x^2+k^2-m^2}) + ATAN2(-k , x)$$

$$= ATAN2(x*m\mp k*\sqrt{x^2+k^2-m^2} , k*m\pm x*\sqrt{x^2+k^2-m^2})$$

$k^2 \geq m^2$ holds true (see (18) and (19)) and thus $^{\pm}f_i^{-1}(x)$ cannot fail; in dependence on the sign of k $^{\pm}f_i^{-1}(x)$ is monotone. From this it follows for $^{+/-}B_i$:

$$^{+}B_i = [^{+}f_i^{-1}(-sign(k)*\sqrt{a_{free}^2+b_{free}^2}) , ^{+}f_3^{-1}(sign(k)*\sqrt{a_{free}^2+b_{free}^2})] \qquad (26)$$

$$^{-}B_i = [^{-}f_i^{-1}(-sign(k)*\sqrt{a_{free}^2+b_{free}^2}) , ^{-}f_3^{-1}(sign(k)*\sqrt{a_{free}^2+b_{free}^2})] \qquad (27)$$

- Connecting the pseudo-free variable to the second free variable

In order to connect the range of the pseudo-free variable with the second free variable we use equation IV and get – abbreviating to $L:=A_{1,2}^{-1} \bullet T \bullet (X_5 \bullet A_{5,6} \bullet A_{6,7} \bullet A_{7,8} \bullet A_{8,9})^{-1}$ – the inverse functions

$$\theta_{free} = \theta_2 = \Phi(\theta_{pseudofree}) = ATAN2(L_{23} , L_{13}) \{+\pi\} \quad \text{or} \qquad (28)$$

$$\theta_{free} = \theta_4 = \Phi(\theta_{pseudofree}) = ATAN2(-L_{32} , L_{31}) \{+\pi\} \qquad (29)$$

Inside L the backwards solutions of θ_7 and θ_{calc} ($F_7(\theta_{pseudofree})$, $F_{6/8}(\theta_{pseudofree})$), based on $\theta_{pseudofree}$, have to be applied and therefore we must distinguish between the $^{+}$L- and $^{-}$L-variant and the accompanying range of $\theta_{pseudofree}$.

Until now we have not succeeded in proving the theorem "$f(\theta)=ATAN2(\ F1(\theta)\ ,\ F2(\theta)\ )$ is monotone in $[-\pi\ ,\ \pi]$", but all our simulations have shown this result; so we use this theorem to simplify the last step of the range computation.

The backwards solution of $\theta_{pseudofree}$ from θ_{free} contains two ambiguities and this results in four intervals S+H+, S+H–, S–H+, S–H– of $\theta_{pseudofree}$ instead of two (see ${}^{+/-}B$).

We have already reduced $MR_{pseudofree}$ to $D_{pseudofree}$, but with $\sin(\theta+2\pi)=\sin(\theta)$ and $\cos(\theta+2\pi)=\cos(\theta)$ the theorem shows that all intervals $S\cdots H\cdots$ have an extent of 2π and therefore there is no need for the intersection "$S\cdots H\cdots\cap D_{+/-B\rightarrow pseudofree}$", as done in the former section.

The interval $D_{pseudofree\rightarrow free}$ arises from the union of four sets:

$${}^{++}D_{pseudofree\rightarrow free} = {}^{+}\Phi(D_{+B\rightarrow pseudofree}\cap MR_{pseudofree})\text{ with }{}^{+}L$$

$${}^{+-}D_{pseudofree\rightarrow free} = {}^{+}\Phi(D_{-B\rightarrow pseudofree}\cap MR_{pseudofree})\text{ with }{}^{-}L$$

$${}^{-+}D_{pseudofree\rightarrow free} = {}^{-}\Phi(D_{+B\rightarrow pseudofree}\cap MR_{pseudofree})\text{ with }{}^{+}L\ (={}^{++}D_{pseudofree\rightarrow free}+\pi)$$

$${}^{--}D_{pseudofree\rightarrow free} = {}^{-}\Phi(D_{-B\rightarrow pseudofree}\cap MR_{pseudofree})\text{ with }{}^{-}L\ (={}^{+-}D_{pseudofree\rightarrow free}+\pi)$$

$$D_{pseudofree\rightarrow free} = {}^{++}D_{pseudofree\rightarrow free}\cup{}^{+-}D_{pseudofree\rightarrow free}\cup{}^{-+}D_{pseudofree\rightarrow free}\cup{}^{--}D_{pseudofree\rightarrow free}$$

The final form of D_{free} looks like

$$D_{free} = (\ (D_{+B\rightarrow free}\cap D_{pseudofree\rightarrow free})\cup(D_{-B\rightarrow free}\cap D_{pseudofree\rightarrow free}))\cap MR_{free}$$

$$\Rightarrow\quad D_{free} = (D_{+B\rightarrow free}\cup D_{-B\rightarrow free})\cap D_{pseudofree\rightarrow free}\cap MR_{free}$$

Choice of the free variables

The described algorithm will be applied to both θ_1 and θ_5. We can then choose this variable as a free variable whose actual value lies within the definition range D_i or very near to it. A user-specific decision has to be made when both actual variables values lie within D_i.

As shown earlier there is $a*\sin(\theta)+b*\cos(\theta)=r*\sin(\theta-\varphi)$, which means that – viewing joints G2 and G4 from a mathematical perspective –, the second free variable has a range of 360°, when choosing the variable with the smaller amplitude r as the free variable. Besides we have to choose the second free variable under this condition because the statement about the monotony of θ_3 is based on it.

More details on the computation of $f_i({}^{\pm}B_i\cap MR_i)$ and the optimization of the free variable θ_1/θ_5 in the case of degeneration of the other free variable can be found in [Kiener, 1991].

CONCLUSION

The proposed algorithm provides explicit, closed formulae for solving the inverse kinematic problem of a man-like structure with eight rotational joints, and it takes into account the available range of the joints. As a result of this it becomes possible to stay close to the current joint values and to implement further optimization strategies to define the values of the free variables adequately from the permitted range.

The algorithm was implemented on a microVAX II (DEC) and needs about 60 msec computation time (without optional optimization of the free variable θ_1/θ_5).

Based on this work it was shown that also kinematical structures with more than six joints can also be handled efficiently and profitably by explicit methods.

REFERENCES

Heiß, H. (1985). Die explizite Lösung der kinematischen Gleichung für eine Klasse von Industrierobotern. *Bericht 18504 der TU München.*

Heiß, H. (1986a). Explizite und inkrementelle Rückwärtsrechnung bei Industrierobotern. *VDI-Berichte*, **598**, 175-185.

Heiß, H. (1986b). Konstruktionskriterien und Lösungsverfahren für Industrieroboter mit explizit lösbarer kinematischer Gleichung. *Robotersysteme*, **2**, 129-137.

Heiß, H. (1986c). Homogeneous and dual matrices for treating the kinematic problem of robots, *Symp. on Theory of Robots*, 13-17.

Heiß, H. (1987). Theorie und Anwendung der Koordinatentransformation bei Roboterkinematiken. *Informatik - Forschung und Entwicklung*, **2**, 19-33.

Karlen, J.P. and others (1989). Reflexive posture control for kinematically redundant manipulators. *ISA ROBEXS*, 1-7.

Karlen, J.P. and others (1990). A dual-arm dexterous manipulator system with anthropomorphic kinematics", *IEEE Int. Conf. on Robotics and Automation*, 368-373.

Kiener, G. (1991) "Rückwärtsrechnung für ein Hand-Arm-System mit acht Gelenken", *Technical University Munich, Department of Computer Science*, diploma thesis.

Woernle, C. (1988). Ein systematisches Verfahren zur Aufstellung der geometrischen Schließbedingungen in kinematischen Schleifen mit Anwendung bei der Rücktransformation für Industrieroboter. *Fortschrittsberichte VDI Reihe 18*, **59**.

A Mixed Numeric and Symbolic Approach to Redundant Manipulators[1]

Michael Kauschke
Institute for Robotics and Computer Control
Technical University of Braunschweig
D-38114 Braunschweig F.R.G.

Abstract
This paper presents a method for generating locally optimized link variable solution for redundant serial link robots. It combines closed form solutions for the inverse kinematic problem of nonredundant subchains and a numeric approach solving the local optimizing problem. The result is an efficient method for low degree of redundancy and complex optimizing tasks.

1 Introduction

Most methods dealing with redundant robots reduce the highly nonlinear mathematical descriptions to a locally linearized representation. This allows the use of well known methods from linear algebra for solving the inverse kinematic problem (IKP) plus additional tasks. Unfortunately it is difficult to transfer simplifications resulting from a specialized kinematic structure to these approaches. Therefore a closed form solution of the IKP, which relys on specialization, often gives better results in the case of nonredundant manipulators. The solution can efficiently be generated by the algorithms given by R. Paul in 1981. Attempts to build closed form solutions, which are able to handle additional tasks for redundant manipulators, lead to impractical results with respect to computational complexity. Even a simple task, like the limitation of a link variable range, produces exhaustive expressions. Nevertheless combining numerical and symbolic approaches lead to efficient solutions as will be shown in the following.

[1]This work has been supported by the German Forschungsgemeinschaft DFG.

J. Angeles et al. (eds.), Computational Kinematics, 67–74.

2 The Closed Form Solution

The direct kinematic equations for a manipulator with n links can be written in the following form:

$$\vec{\mathrm{f}}(\vec{\mathrm{q}}_n, \vec{\mathrm{d}}) = \vec{\mathrm{x}}_k$$

$\vec{\mathrm{q}}_n$ is the n-dimensional link variable vector, $\vec{\mathrm{x}}_k$ contains the position of the end effector in k-dimensional Cartesian space. Vector $\vec{\mathrm{d}}$ describes the kinematic structure of the robot, often given as Denavit-Hartenberg parameters (omitted in the following). When dealing with globally nondegenerated structures, it is often possible to generate a closed form inversion of $\vec{\mathrm{f}}$ for $n = k$:

$$\vec{\mathrm{f}}^{-1}(\vec{\mathrm{x}}_k) = \vec{\mathrm{i}}(\vec{\mathrm{x}}_k, h) = \vec{\mathrm{q}}_n \quad \wedge \quad k = n$$

h is used to select a unique solution of the finite set of solutions for $\vec{\mathrm{f}}^{-1}$. The corresponding scalar equations of $\vec{\mathrm{i}}(\vec{\mathrm{x}}_k)$ can be denoted in a triangular form:

$$\begin{aligned} i_{n1} &= q_{n1} = g_1(\vec{\mathrm{x}}, h) \\ i_{n2} &= q_{n2} = g_2(\vec{\mathrm{x}}, q_{n1}, h) \\ i_{nn} &= q_{nn} = g_n(\vec{\mathrm{x}}, q_{n1}, \ldots, q_{n\, n-1,h}) \end{aligned} \tag{1}$$

This can be extended to redundant robots by building a subset $\vec{\mathrm{q}}_m$ out of $\vec{\mathrm{q}}_n$ with $m = k$ link variables, which allows the detemination of a closed form solution for this subset as shown by Schrake in 1990.

$$\begin{aligned} \vec{\mathrm{f}}(\vec{\mathrm{q}}_n) = \vec{\mathrm{f}}(\vec{\mathrm{q}}_m, \vec{\mathrm{q}}_r) &= \vec{\mathrm{x}}_k \\ \vec{\mathrm{i}}(\vec{\mathrm{x}}_k, \vec{\mathrm{q}}_r, h) &= \vec{\mathrm{q}}_n \end{aligned} \tag{2}$$

$\vec{\mathrm{q}}_r$ contains all link variables not in $\vec{\mathrm{q}}_m$. The solution set $\mathbb{Q}_r$ for $\vec{\mathrm{q}}_r$ is $\mathbb{R}^r$ excluding those elements which do not lead to a valid solution for the inverse kinematic:

$$\mathbb{Q}_r = \left\{ \vec{\mathrm{q}}_r \in \mathbb{R}^r \,|\, \vec{\mathrm{f}}(\vec{\mathrm{q}}_m, \vec{\mathrm{q}}_r) = \vec{\mathrm{x}}_k \wedge \vec{\mathrm{q}}_m \in \mathbb{Q}_m \right\}$$

To be able to use all elements of $\mathbb{Q}_r$ it is necessary that $\vec{\mathrm{i}}$ describes the complete solution set of the IKP. This is a natural property of symbolic solutions. Note, that equation (2) delivers a stable solution for the positioning task independent of $\vec{\mathrm{q}}_r$ as long as the manipulator is not in a singularity with respect to $\vec{\mathrm{q}}_m$.

3 Integrating Additional Task Constraints

To get a finite set of solutions for $\vec{\mathrm{q}}_n$ additional task constraints have to be introduced. These can be divided into two groups:

1. Tasks, which depend directly on the link variables, like obstacle avoidance, joint range limitations, etc.
2. Tasks, which are described by equations including differential expressions of the link variables. Examples are energy minimization, torque optimization, etc.

The second group implies that a differential term for the inverse kinematic has to be generated:

$$\frac{d\vec{\mathrm{i}}(\vec{\mathrm{x}}_k, \vec{\mathrm{q}}_r, h)}{d\,\vec{\mathrm{q}}_r} \tag{3}$$

Having a triangular form of $\vec{\mathrm{i}}$ like in (1), this can be obtained by applying the chain rule recursively to the equation system. Unfortunately the results are far too complex to be of any practical use. So, this work is focused on tasks of the first kind. They can be described by a cost function $t(\vec{\mathrm{q}}_n)$ to be minimized. Using equation (2), one gets

$$t(\vec{\mathrm{q}}_n) = t(\vec{\mathrm{i}}(\vec{\mathrm{x}}_k, \vec{\mathrm{q}}_r, h), \vec{\mathrm{q}}_{\mathrm{r}}) = t'(\vec{\mathrm{x}}_k, h, \vec{\mathrm{q}}_r) \tag{4}$$

In the case of tasks with a given trajectory $\vec{\mathrm{x}}_k(t)$ the optimizing problem advantageously can be considered in the configuration space $\mathbb{Q}_r$. According to (4) one has to solve the following differential equation for every point of the trajectory:

$$\begin{aligned} 0 &= \frac{dt'(\vec{\mathrm{x}}_k, \vec{\mathrm{q}}_r)}{d\vec{\mathrm{q}}_r} = \frac{d\,t(\vec{\mathrm{i}}(\vec{\mathrm{x}}_k, \vec{\mathrm{q}}_r, h), \vec{\mathrm{q}}_r)}{d\,\vec{\mathrm{q}}_r} \\ &= \frac{d\,t(\vec{\mathrm{q}}_m, \vec{\mathrm{q}}_r)}{\vec{\mathrm{q}}_n} \frac{d\,\vec{\mathrm{i}}(\vec{\mathrm{x}}_k, \vec{\mathrm{q}}_r, h)}{d\,\vec{\mathrm{q}}_r} + \frac{d\,\vec{\mathrm{i}}(\vec{\mathrm{x}}_k, \vec{\mathrm{q}}_r, h)}{d\,\vec{\mathrm{q}}_r} \end{aligned} \tag{5}$$

The second term of the sum is the same as in (3); so the solution of (5) cannot be determined independently of the optimizing task with reasonable efforts.
Nevertheless, it is possible to build a local model of t' which is easy to produce and easy to differentiate:

$$l(\vec{\mathrm{q}}_r) = \sum_{i=1}^{r} \left(a_i q_i + \sum_{j=i}^{r} b_{ij} q_i q_j \right) + d \tag{6}$$

This quadratic polynomial has $n_c = r + \frac{(r+1)r}{2} + 1$ coefficients to be determined by solving a linear equation system generated by n_c test points of $t'(\vec{\mathrm{q}}_r)$. The minimum of $l(\vec{\mathrm{q}}_r)$ can be found by the following set of differential equations which are linear in $\vec{\mathrm{q}}_r$:

$$0 = \frac{d\,l(\vec{\mathrm{q}}_r)}{d\,q_i} = a_i + \sum_{j=i}^{r} b_{ij} q_j + b_{ii} q_i \quad \wedge \quad i = 1, \ldots, r \tag{7}$$

Using a local model of the modified cost function t', there is no need to differentiate t'. This means that one can use nonanalytic functions combining many different task constraints. Nevertheless, a function $t' \in \mathbb{C}^2$ supports a stable iteration. The resulting computational complexity of this optimizing strategy is determined as follows:

Computational Step	Costs
Calculation of the test points for t'	$O(r^2)$
Determining coefficients of (6)	$O(n_c^3) = O(r^5)$
Calculating $d\,l(\vec{\mathrm{q}}_r)/d\,\vec{\mathrm{q}}_r = 0$	$O(n_c^3) = O(r^5)$

4 Handling Singularities of the Inverse Kinematic

The differential movement of the robot in Cartesian coordinates can be derived from a differential movement in link variable space:

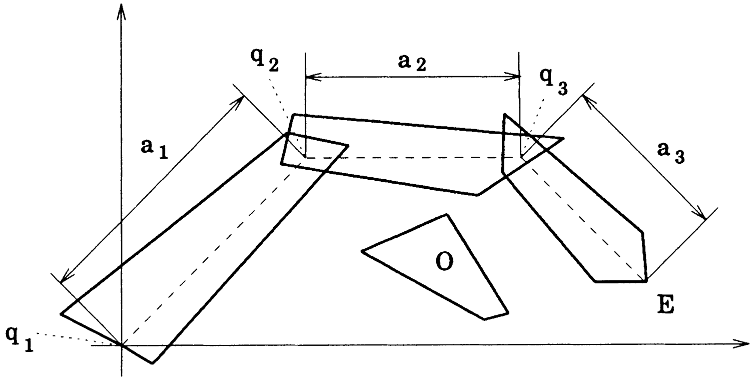
$$d\vec{\mathrm{x}}_k = \frac{d\vec{\mathrm{f}}(\vec{\mathrm{q}}_n)}{d\vec{\mathrm{q}}_n}\, d\vec{\mathrm{q}}_n = \begin{bmatrix} | & & | \\ \frac{d\vec{\mathrm{f}}}{dq_1}(\vec{\mathrm{q}}_n) & \cdots & \frac{d\vec{\mathrm{f}}}{dq_n}(\vec{\mathrm{q}}_n) \\ | & & | \end{bmatrix} d\vec{\mathrm{q}}_n \tag{8}$$

The manipulator is in a nonsingular position if the set $\mathbb{I}$ of all k-dimensional link variable vectors corresponding to k linear independent columns contains at least one element. An inverse kinematic (IK) $\vec{\mathrm{i}}(\vec{\mathrm{q}}_r, h) = \vec{\mathrm{q}}_m$ becomes singular if $\vec{\mathrm{q}}_m \notin \mathbb{I}$. So, it is necessary to build an IK $\vec{\mathrm{i}}(\vec{\mathrm{q}}_r, h)$ for each possible k-dimensional subvector of $\vec{\mathrm{q}}_n$ to cover all cases of nonsingular positions of the whole manipulator with $|\mathbb{I}| = 1$. This gives a total number of $\binom{k}{n}$ different IKs, which can be generated in advance by the aid of automatic IK generators like SKIP (H. Rieseler 1990) or INKAS (L. Herrera-Bendezu 1988). Having a complete set of IKs at hand, it is possible to generate a stable iteration for all nonsingular positions of the complete manipulator by choosing an adequate IK. As all IKs describe the complete solution set for the corresponding subchain, switching between nonsingular IKs can be done in a continious way by choosing the corresponding pairs of selection variables h.

In cases where the complete set of IKs cannot be built, the cost function t' has to be modified by an additional term expressing the distance to a singularity of the actually used IK to prevent the iteration process to select such configurations.

5 An Example

The end effector E of a 2D-manipulator consisting of 3 rotational links represented by arbitrary polygons follows a given trajectory $\vec{e}(t)$ simultaneously avoiding collision with some obstacle O:



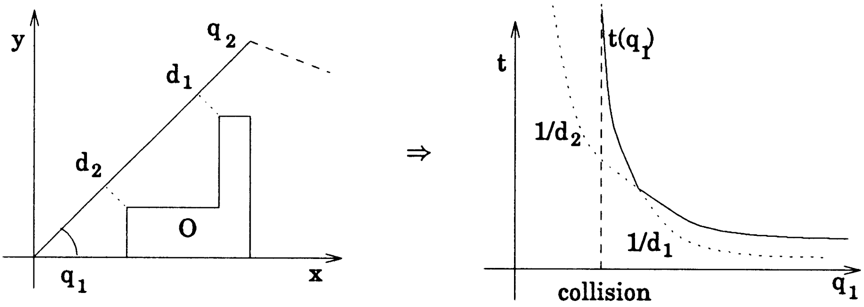
Using the described strategy one has to build 3 different IKs $\vec{i}_1(\vec{x}(t), q_1)$, $\vec{i}_2(\vec{x}(t), q_2)$, $\vec{i}_3(\vec{x}(t), q_3)$. The IK $\vec{i}_1$ solving q_2, q_3 has the following form:

$$\begin{aligned}
p_x &= e_x(t) - a_1 \cos(q_1)\\
p_y &= e_y(t) - a_1 \sin(q_1)\\
sq &= (a_2 + a_3)^2 - px^2 - py^2) * ((a_2 - a_3)^2 - px^2 - py^2)\\
q_2 &= \operatorname{atan2}(p_x^2 p_y + p_y^3 + a_2^2 p_y + p_x\sqrt{sq} - a_3^2 p_y,\\
&\quad a_2^2 p_x + p_x^3 + p_x p_y^2 - p_x a_3^2 - \sqrt{sq} p_y) - q_1\\
q_3 &= \operatorname{atan2}(p_y - a_2 \sin(q_1 + q_2), p_x - a_2 * \cos(q_1 + q_2)) - q_2 - q_1
\end{aligned}$$

The other IKs can be built in an analogous way. To avoid collision one has to introduce a cost function t describing a potential field. As there is no need to differentiate t, this field can be defined in a simple manner:

$$t(\vec{q}) = 1/ \min_{i \in \{1,2,3\}} (mindist(P_i, O)) \tag{9}$$

$mindist(P_i, O)$ gives the shortest distance between the polygon attached to link i and the obstacle O. The resulting potential field is not a C^2-curve in configuration space:



In contrast to global differentiable definitions of t this approach allows the use of heuristic strategies for finding the minimal distance to an obstacle in more complex environments. The local model l and its minimum T_0 are calculated once for an instant t_{n+1} using sample points T_0, T_1, T_2 derived from the last iteration n for the free link variable of the IK in use:

$$T_{0(n+1)} = quadmin(T_{0n}, T_{1n}, T_{2n}) \tag{10}$$

$$T_{1(n+1)} = \frac{T_{0(n+1)} + T_{0n}}{2} \tag{11}$$

$$T_{2(n+1)} = \frac{T_{0n} + T_{1n}}{2} \tag{12}$$

The function *quadmin* realises the optimization.

The end effector of the given manipulator is moving along a straight line from the space point $(5, 5)$ to $(5, -5)$ avoiding collision with obstacle points. In the first example (Fig. 1) the potential field generates a continuous run of the local minimum for all instants of time giving a continious movement of the manipulator.

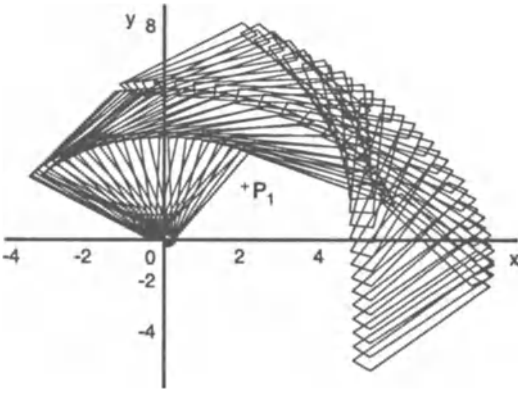


Figure 1: A task leading to a continious movement

The second example (Fig. 2) shows a potential field which has two local minima to the left and right auf P_2. The resulting jump of the trajectory in configuration space can be seen in Fig. 3. As the task optimization and the tracking of a given trajectory are decoupled by the inverse kinematic, a simple limitation of the joint velocity can deliver a continious solution for q_n.

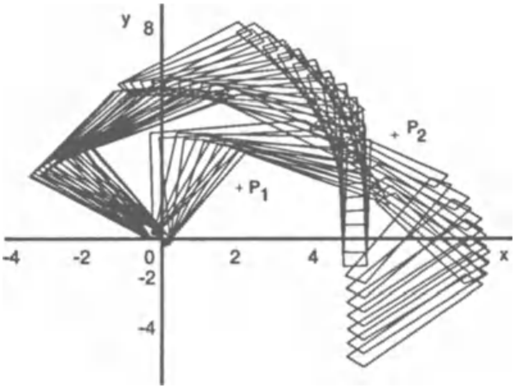


Figure 2: An noncontinious movement produced by a second obstacle

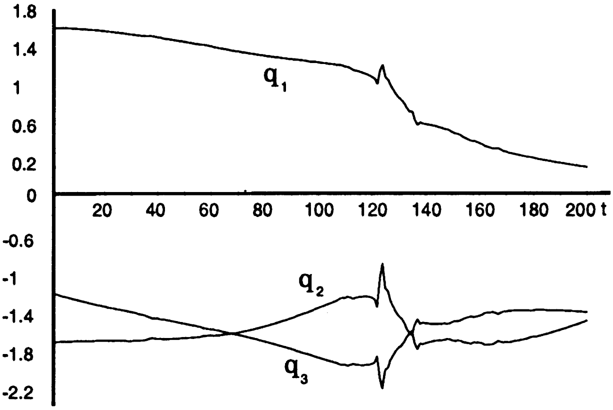


Figure 3: The plot of $\vec{q}_n = \vec{i}(\vec{e}(t))$ belonging to the second example

6 Conclusions

The use of closed form solutions for the inverse kinematics of subchains of redundant robots leads to a reduction in the dimension of the configuration space in which the local optimization is performed numerically according to some task constraint. This approach has the following advantages:

- High speed for a low degrees of redundancy.
- Nonanalytic task functions can be integrated.
- A stable trajectory for the end effector is guaranteed. The distance between two interpolation points of the trajectory can be tailored to the precision needed for the optimizing task.

However, there are some drawbacks:

- High degree of redundancy results in unacceptable high computational effort.
- Known properties of the task constraints can not be integrated in the local model.
- There is no simple criterion for the stability of the iteration.

Future work will concentrate on the development of better local models and task constraint functions for obstacle avoidance.

References

Herrera-Bendezu L. G. , Mu E. and Cain J. T. 1988. Symbolic Computation of Robot Manipulator Kinematics. In *Proc. IEEE International Conference on Robotics and Automation*, pages 993–998.

Paul, R. P. 1981. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press.

Rieseler H., Schrake H., and Wahl F. M. 1991. Symbolic Computation of Closed Form Solutions with Prototype Equations. In *Proc. 2^{nd} International Workshop on Advances in Robot Kinematics.* Springer Verlag.

Schrake H. , Rieseler H. , and Wahl F. M. 1990. Symbolic Kinematics Inversion of Redundant Robots. In *Proc. of the 4^{th} International Symposium on Foundations of Robotics*, Institut für Automatisierung (Berlin) und Institut für Mechanik (Chemnitz).

COMPUTATIONAL CONSIDERATIONS ON KINEMATICS INVERSION OF MULTI-LINK REDUNDANT ROBOT MANIPULATORS

Jadran Lenarčič
The "Jožef Stefan" Institute, Ljubljana, Slovenia

ABSTRACT - The paper critically evaluates the utilisation of pseudo-inverse-based methods for the kinematic inversion of hyper-redundant multi-link robot manipulators. The validity of these methods is questioned from the viewpoint of the computational efficiency specified in terms of arithmetic operations per iteration step. Even though the pseudo-inverse-based methods provide better convergence, less computation time is needed by steepest descent methods especially in continuous path control and in applications where very few iterations are needed or good initial estimations are provided.

1. INTRODUCTION

A common feature in the utilisation of multi-link hyper-redundant manipulators is an infinity of possible motions that these mechanisms can make. There is a need to decide which particular motion should be executed in order to satisfy the task constraints and simultaneously optimise a given set of criteria. From the kinematics viewpoint, the advantages that multi-link manipulators introduce in the task are their high flexibility and versatility. These mechanisms have the ability to solve a given task with a minimum effort with respect to various kinematic and dynamic criteria, and in the same time, to move in very complex environments avoiding obstacles, as reported by Maciejewski and Klein (1985), Galicki (1992), Colbaugh et al. (1989), and undesired or ill-conditioned configurations reported by Baker and Wampler (1988), Nakamura and Hanafusa (1986), and Shamir (1990). The large number of degrees of freedom permits the grasping of

J. Angeles et al. (eds.), Computational Kinematics, 75–84.

objects with the body of the manipulator, presented by Pettinato and Stephanou (1989), and Kerr et al. (1992), introducing the objects in tubes of different shapes, presented by Chiacchio et al. (1991), or gaining advantages of kinematic singularities with the goal to compensate weak actuators, presented by Kieffer and Lenarčič (1992). With a large number of rigid links they can approximate a continuous morphology of snakes or tentacles, reported by Chirikjian and Burdick (1991).

Despite their potential advantages, practical utilisation of multi-link hyper-redundant manipulators is still far away. This is primarily because of several technical and technological problems in the mechanical design, control and programming, which call for more effective mathematical treatment. Numerical complexity of the existing methods for redundancy resolution usually increases with the number of degrees of freedom. Hence, real-time computation for robot manipulators possessing one hundred or more degrees of freedom is still unrealisable in practice or requires extreme computer capacity. The closed-form solutions to the inverse kinematics problem for redundant manipulators are rather exclusive and exist only for special manipulator structures. These were investiagted by Chang (1987). For a general manipulator, there are only iterative numerical solutions that can be computationally very expensive. However, the numerical approach is essential for the development of a general-purpose computer-aided robot design and control.

2. POSITION AND ORIENTATION OF THE END EFFECTOR

As is well known, the kinematics model of a serial robot manipulator is specified as a set of independent algebraic equations (Lenarčič, 1993)

$$\mathbf{p} - \mathbf{p}_0(\theta) = \mathbf{0}\,, \tag{1}$$

where $\mathbf{p}$ is an m-dimensional column vector that expresses the desired end effector position and orientation of the robot mechanism, θ is an n-dimensional column vector of joint (generalised) coordinates, and $\mathbf{p}_0(\theta)$ is a trigonometric vector function of joint coordinates representing the actual end effector position and orientation that is also referred to as the vector of Cartesian (task) coordinates. The Jacobian form of (1) is obtained by differentiation with respect to time

$$\dot{\mathbf{p}} - \mathbf{J}(\theta)\dot{\theta} = \mathbf{0}\,, \tag{2}$$

where

$$\mathbf{J} = \left[\partial \mathbf{p}_\theta / \partial \theta_1, \ldots, \partial \mathbf{p}_\theta / \partial \theta_n\right] . \qquad (3)$$

J is the $m \times n$ Jacobian matrix, and $\dot{\mathbf{p}}$ and $\dot{\theta}$ are Cartesian and joint velocities of the manipulator, respectively. The inverse kinematics problem is referred to as the calculation of joint coordinates for a given combination of Cartesian coordinates based on their relationship specified by (1,2,3).

In general, the inverse kinematics problem of a robot manipulator is the problem of finding joint coordinates or their velocities that will produce a given end effector motion. The solution to this problem can be used in the real-time control of robots to determine joint motions that correspond to a desired Cartesian path of the end effector. For serial robot manipulators, the inverse kinematics problem is often difficult to solve, since the joint coordinates appear in (1,2,3) as arguments of trigonometric functions. This problem may not always have a real solution, and the closed-form solution can be specified only for the simple, usually non redundant, structures of robot mechanisms. For more complex mechanisms, time-consuming numerical procedures must be used. The solutions are multiple and there is a need to decide which combination of values is more suitable in each particular case. Providing a robot mechanism with some redundant degrees of freedom enhances its motion capabilities considerably. Kinematically redundant robots provide the means for solving sophisticated motion tasks, but require complex approaches in both mechanical design and control. The main feature of redundancy is that there is an infinity of possible motions which these mechanisms can make, all of which satisfy the task constraints. Mathematically, a robot manipulator is redundant with respect to the prescribed task, when (1) is underdetermined and the dimension of θ is greater than the dimension of $\mathbf{p}$; $n > m$. It results in a non square Jacobian matrix (3) which is not directly invertible.

3. THE MOORE-PENROSE GENERALISED INVERSE

A widely used approach of finding joint coordinates θ for a given combination of Cartesian coordinates $\mathbf{p}$ is to use the pseudo inverse. Also termed the Moore-Penrose generalised inverse, it was proposed by many authors in similar variations of iterative numerical schemes (Klein and Huang, 1983; Benhabib et al., 1985; Lovass-Nagy and Schilling, 1987; Nakamura et al., 1987; Nenchev, 1989). The Moore-Penrose generalised inverse **M** of the Jacobian matrix **J** is given by

$$\mathbf{M} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}. \tag{4}$$

Its size is $n \times m$ and solves (2) as follows

$$\dot{\theta} = \mathbf{M}\dot{\mathbf{p}}\,, \tag{5}$$

This is a particular solution. We can verify that from the infinity of solutions the Moore-Penrose generalised inverse chooses the one that minimises the Euclidian norm of joint velocities. These are very known results in mathematics related to underdetermined systems of equations where the number of unknowns is greater than the number of constraints. The formulation (5) forms the background of very elegant numerical procedures which by integration of (5) can calculate joint coordinates of a redundant manipulator for given end effector positions and orientations. One is given by

$$\theta^{i+1} = \theta^i + \mathbf{M}(\theta^i)(\mathbf{p} - \mathbf{p}_\theta(\theta^i))\,, \tag{6}$$

where $i = 1,2,...$is the number of iterations, $\mathbf{p}$ is a desired end effector pose, $\mathbf{p}_\theta(\theta^i)$ is the actual end effector pose, $\mathbf{M}(\theta^i)$ is the corresponding Moore-Penrose generalised inverse as given in (4), and θ^0 is a known initial estimation. In order to control the error in Cartesian coordinates of the end effector we must additionally compute

$$\varepsilon(\theta^i) = (\mathbf{p} - \mathbf{p}_\theta(\theta^i))^T(\mathbf{p} - \mathbf{p}_\theta(\theta^i)) \tag{7}$$

for all $i = 1,2,...$ The procedure is stopped when ε is less than the desired accuracy ε_0.

There are, however, some disadvantages that must be taken into account in the utilisation of the pseudo-inverse-based methods. These are entirely numerical procedures and is very difficult to find any analytical result. They have intrinsic inaccuracy and accumulate error that becomes larger as the velocity increases. To overcome these difficulties some authors developed analytical or semi analytical methods to resolve redundancy. However, no symbolic solution can be developed for a general redundant manipulator unless certain conditions are met by the manipulator structure. Some attempts were reported by Stanišič and Pennock (1985), Varma and Huang (1992), Duffy and Crane (1980), and Chaware and Amarnath (1987). More general approaches were presented by Ghosal and Roth (1988), and Chang (1987). The calculation schemes based on the Moore-Penrose generalised inverse are procedures of local optimisation (Nenchev, 1989). They minimise a weighted Euclidian norm of joint velocities at every point or any given moment. It is very hard to know a priori how large the minimum is. Recently, promising results were obtained in the global optimisation with integral-type criteria (Nedungadi and Kazerounian, 1989; Kazerounian amd Wang, 1988). Generalised inverse methods yield a non conservative motion. The particular problem of concern is the non-repeatability of periodic motions resulting from drift associated with pseudo inverse approach. The drift problem was described Klein and Huang (1983). It has become the major criticism of the methods based on pseudo inverses and, consequently, one of the most investigated subjects in redundancy resolution. We can refer here to the work presented by Klein and Kee (1989), and more recently by Luo and Ahmad (1992), and Bay (1992).

From the viewpoint of adopting the pseudo inverse techniques for the inverse kinematics of multi-link robot manipulators, the main difficulty is associated to their numerical complexity. According to (6,7), the calculation of the vector of joint coordinates in each iteration $i = 1,2,...$ contains $\mathrm{N}_{\otimes}^{MPI}$ multiplications and divisions and $\mathrm{N}_{\oplus}^{MPI}$ additions and subtractions is proportional to a third order polynomial of n and m

$$\begin{aligned} \mathrm{N}_{\otimes}^{MPI} &= n(m^2+m)+m^3+m^2+m \\ \mathrm{N}_{\oplus}^{MPI} &= n(m^2+m)+m^3+m-1 \end{aligned} \quad , \tag{8}$$

where the number of arithmetic operations is taken from regular matrix multiplication rules and matrix inversion by performing LU decomposition. It can be observed that for a high number of degrees of freedom, for example $n = 100$, and for $m = 6$ that corresponds to the number of position and orientation constraints, the number of multiplications $\mathrm{N}_{\otimes}^{MPI}$ = 4458, and additions $\mathrm{N}_{\oplus}^{MPI}$ = 4421 in each iteration step $i = 1,2,...$ The number of iterations that find a solution depends on the accuracy of the first estimation and on the desired accuracy of the result. We can also observe in (8) that the number of multiplications and additions increases with the number of degrees of freedom multiplied by a quadratic polynomial of the number of the task constraints.

4. RAPIDLY CONVERGENT DESCENT METHOD

We can minimise the number of arithmetic operations by decreasing the task constraints m or, as it is shown in this section, by simply redefining the optimisation criterion. Searching for a solution of (1) we can consider a local optimisation problem of minimising the difference between the desired Cartesian coordinates $\mathbf{p}$ and the actual Cartesian coordinates $\mathbf{p}_0(\theta)$. An adequate criterion would be a quadratic function of the form

$$f(\theta)=\frac{1}{2}\varepsilon(\theta)\,, \tag{9}$$

where f is the objective function given as a norm $\varepsilon(\theta)$ specified in (7). The criterion function f (θ) can be interpreted as the quadratic distance between the desired and the actual end effector position and orientation. Assume that f is differentiable with respect to θ. Its gradient is then given, in accordance to the definition of the Jacobian matrix **J** (3), by

$$\mathbf{g}(\theta)=\mathbf{J}^T(\mathbf{p}-\mathbf{p}_\theta(\theta))\,, \tag{10}$$

where **g** is *n*-dimensional column vector. We are concerned here. with the general problem of finding an unrestricted local minimum of the function f(θ) of several variables. We suppose that the function of interest can be calculated at all points. It is also convenient to group functions into two main classes according to whether the gradient vector **g** is defined analytically at each point or must be estimated from the difference of values of f. The method described in this paper is applicable to the case of the analytically defined gradient.

The steepest descent method introduced almost fifty years ago and later modified by many authors, for example by Powell (1962), is still one of the most attractive and efficient approaches for non linear minimisation problems. The aim is to find the sequence of vectors of joint coordinates θ that minimises the objective function f(θ). Applied to our case, this method can be expressed as

$$\theta^{i+1}=\theta^i+\alpha^i\mathbf{g}(\theta^i)\,, \tag{11}$$

where i = *1,2,...* is the number of iterations, α^i is the iteration step size, and θ^0 is a known initial estimation. The gradient **g** is minimised in each iteration depending on the step size α^i until the error $\varepsilon(\theta^i)$ is less than the desired accuracy ε_0. α^i can be kept constant or can be changed within the procedure in order to guarantee the convergence, as well as to minimise the number of iterations. The reader is referred to the work of Powell (1962) for more extensive details. Comparing (11) with (6) and in relation to the definition of the Moore-Penrose generalised inverse (4), a suitable choice of the step size α^i, i = *1,2,...* can be

$$\alpha^i=\frac{\varepsilon(\theta^i)}{\mathbf{g}(\theta^i)^T\mathbf{g}(\theta^i)}\,. \tag{12}$$

For functions of several variables, there are two very useful conditions that enable very fast and stable convergence. One is to have the Jacobian matrix **J** whose columns depend only on a very small number of variables, and the other is to have quantitatively similar dependency on each of the variables. The resulting numerical procedure has a quadratic convergence and finds the nearest local minimum of a function of several variables very efficiently, especially when the initial estimation is sufficiently close to the solution. From the computational viewpoint, the main difference between the steepest descent method and the method that utilises the Moore-Penrose generalised inverse is in the number of arithmetic operations needed in each iteration. For every $i = 1,2,\ldots$ the steepest descent method as presented in (11,12) contains

$$\begin{aligned} \mathrm{N}_{\otimes}^{SD} &= n(m+1)+2m+1 \\ \mathrm{N}_{\oplus}^{SD} &= n(m+1)+2m-1 \end{aligned} \tag{13}$$

arithmetic operations. When $n >> m$

$$\frac{\mathrm{N}_{\otimes}^{SD}}{\mathrm{N}_{\otimes}^{MPI}} \approx \frac{\mathrm{N}_{\oplus}^{SD}}{\mathrm{N}_{\oplus}^{MPI}} \approx \frac{1}{m}\,. \tag{14}$$

However, to properly compare the mentioned approaches we must also take into account the number of iterations needed to obtain a desired accuracy, as well as the number of operations necessary to compute the Jacobian matrix in each iteration step.

5. NUMERICAL EXAMPLE: *N*-R PLANAR MECHANISM

In order to validate these theoretical findings a planar mechanism that possesses equal links and parallel revolute joints was used. The Moore-Penrose generalised inverse method (6) and the descent method (11) were implemented and compared on a personal computer IBM 425SX (486-processor). The inverse kinematics problem was solved for a given end effector position, $m = 2$, and for a series of examples of mechanisms with different number of degrees of freedom, $n = 20, 50, 100, 500$. The computation time per iteration as a function of the number of degrees of freedom is as follows

n	20	50	100	500
MPI (ms)	2.860	6.570	12.91	64.59
SD (ms)	1.530	3.790	7.630	39.14
SD/MPI	0.535	0.577	0.591	0.606

The ratio SD/MPI is in favour of the descent method and approximately corresponds to the prediction in (14) - the difference arises from the computation of the end effector position and its derivatives that are not included in (14). The generalised inverse method showed better convergence. However, extremely fast convergence was obtained also with the descent method when the joint coordinates of the mechanism were specified in terms of the absolute angles between the links and the reference coordinate frame - instead of relative joint angles that are measured between neighbouring links. A comparison of the convergence specified in terms of the square root of the error (7) for a typical case (n = 100) is as follows

iteration	MPI	SD
1	0.737188	0.737188
2	0.341060	0.293930
3	0.009324	0.013108
4	0.000005	0.000810
5	0.000000	0.000037

6. CONCLUSIONS

This work reports a critical evaluation of the pseudo inverse approach in the kinematic inversion of multi-link hyper-redundant robot manipulators from the viewpoint of computational efficiency. If the inverse kinematics problem is defined as an optimisation problem of minimising the quadratic distance between the desired and the actual end effector position, the pseudo inverse approach can be converted to a more simple procedure of finding local minimum of a function of several variables by using the steepest descent approach. The iteration step of this procedure is approximately m-times less expensive (where m is the number of task coordinates). However, the main point of concern in the implementation of the steepest descent method is related to its convergence. It was shown that very fast convergence, comparable to the convergence of the pseudo-inverse-based method, can be obtained when the coordinates are expressed by the absolute joint angles. This, on the other hand, can incorporate some difficulties in the algorithm. For instance, if there are limits applied to the joint angles, the transformation between absolute and relative joint angles must be performed in each iteration step. In the presented case of a planar n-R mechanism the transformation is trivial, but for a general-type spatial mechanism might be more complex. Another difficulty of the utilisation of absolute angles in the iterative procedure is when, in an initial configuration, two or more absolute angles are equal. They remain equal in the whole procedure, as well as in the final configuration, and may, therefore, disturb the convergence. In conclusion, the steepest descent method is more efficient in applications where the number of iterations is small. An example is in continuos path control where the neighbouring points of the end effector trajectory are relatively close and thus good initial estimations are provided.

7. REFERENCES

Maciejewski, A.A, and Klein, C.A. 1985. Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments, *Int. J. Robotics Research*, 4(4): 109-117

Galicki, M. 1992. Optimal Planning of a Collision-free Trajectory of Redundant Manipulators, *Int. J. Robotics Research*, 11(6): 549-559

Colbaugh, R., Seraji, H., Glass, K.L. 1989. Obstacle Avoidance for Redundant Robots Using Configuration Control, *J. Robotic Syst.*, 6(6):721-744 (1989)

Baker, D.R., Wampler, C.W. 1988. On the Inverse Kinematics of Redundant Manipulators, *Int. J. Robotics Research*, 7(2): 3-21

Nakamura, Y., Hanafusa, H. 1986. Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control, *Trans. ASME, J. Dynamic Syst., Measurement, and Control*, 108: 163-171

Shamir, T. 1990. The Singularities of Redundant Robot Arms, *J. Robotics Research*, 9(1): 113-121

Pettinato, J.S., Stephanou, H.E. 1989. Manipulability and Stability of a Tentacle Based Robot Manipulator, *IEEE Conf. on Robotics and Automation*, Arizona: 458-463

Kerr, D.R., Griffits, M., Sanger, D.J., Duffy, J. 1992. Redundant Grasps, Redundant Manipulators, and Their Dual Relationships", *J. Robotic Syst.*, 9(7): 973-1000

Chiacchio, P.,Chiaverini, S., Sciavicco, L., Siciliano, B. 1991. Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy, *J. Robotics Research.*, 10(4): 410-425

Kieffer, J., Lenarčič, J. 1992. On the Exploitation of Mechanical Advantage Near Robot Singularities, *Int. Workshop on Advances in Robot Kinematics*, Ferrara, Italy

Chirikjian, G.S., Burdick, J.W. 1991. Kinematics of Hyper-Redundant Manipulators, *Advances in Robot Kinematics*, Springer-Verlag, Wien

Chang, P.H. 1987. A Closed-Form Solution for Inverse Kinematics of Robot Manipulators with Redundancy, *IEEE J. Robotics and Automation*, RA-3(5): 393-403

Lenarčič, J. 1993. Motion of Robot Mechanisms: Robot Kinematics, *Encyclopedia of Microcomputers* - Vol. 11, Marcel Dekker, New York

Klein, C.A., Huang, C.H. 1983. Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators, *IEEE Trans. on Syst., Man, and Cybern.*, SMC-13(3): 245-250

Benhabib, B., Goldenberg, A.A., Fenton, R.G. 1985 A Solution to the Inverse Kinematics of Redundant Manipulators, *J. Robotics Systems*, 2(4): 373-385

Lovass-Nagy, L., Schilling, R.J. 1987. Control of Kinematically Redundant Robots Using {1}-Inverses, *IEEE Trans. on Syst., Man, and Cybern.*, 17(4): 644-649

Nakamura, Y., Hanafusa, H., Yoshikawa, T. 1987. Task-Priority Based Redundancy Control of Robot Manipulators, *Int. J. Robotics Research*, 6(2): 3-15

Nenchev, D.N. 1989. Redundancy Resolution through Local Optimisation: A Review, *Int. J. Robotic Systems,* 6(6): 769-798

Stanišić, M:M., Pennock, G.R. 1985. A Nondegenerate Kinematic Solution of Seven-Jointed Robot Manipulator, *Int. J. Robotics Research*, 4(2): 10-20

Varma, H., Huang, M.Z. 1992. Analytic Minimum-Norm Solution for Rate Coordination in Redundant Manipulators, *J. Robotic Systems* 9(8): 1001-1021

Duffy, J., Crane, C., A Displacement Analysis of the General 7-Link, 7R Mechanism, *Mech. Mach. Theory*, 15(3): 153-169

Chaware, P.R., Amarnath, C. 1987. A Combined Approach for the Solution of the Inverse Kinematic Problem for Certain Special 7R Manipulators, *Mech. Mach. Theory*, 22 (2): 147-151

Ghosal, A., Roth, B. 1988. A New Approach for Kinematic Resolution of Redundancy, *Int. J. Robotics Research*, 7(2): 22-35

Nedungadi, A., Kazerounian, K. 1989. A Local Solution with Global Characteristics for the Joint Torque Optimization of a Redundant Manipulator, *J. Robotic Systems* 6(5): 631-654

Kazerounian, K.,Wang, Z. 1988. Global versus Local Optimization in Redundancy Resolution of Robotic Manipulators, *Int. J. Robotics Research*, 7(5): 3-13

Klein, C.A., Kee, K.B. 1989. The Nature of Drift in Pseudoinverse Control of Kinematically Redundant Manipulators, *IEEE J. Robotics and Automation*, 5(2): 231-234

Luo, S., Ahmad, S. 1992. Predicting Drift Motion for Kinematically Redundant Robots, *IEEE Trans. Syst., Man, and Cybern.*, 22(4): 717-728

Bay, J.S. 1992. Geometry and Prediction of Drift-Free Trajectories for Redundant Machines Under Pseudoinverse Control, *Int. J. Robotics Research*, 11(1): 41-53

Powell, M.J.D. 1962. An Iterative Method for Finding Stationary Values of a Function of Several Variables, *The Computer J.*, 5: 147

On Finding the Set of Inverse Kinematic Solutions for Redundant Manipulators

Enric Celaya and Carme Torras
Institut de Cibernètica (UPC - CSIC)
Diagonal 647, 08028-Barcelona, SPAIN

Abstract

The inverse kinematic solution of a manipulator with ρ redundant d.o.f.'s can be seen as the configuration space of a closed kinematic loop with mobility $M = \rho$. This set can be described by means of the fasible ranges of values that each variable can take. It is possible, for all planar and spherical loops, obtaining such ranges without explicitly finding the algebraic expression of the solution. The form presented by such ranges permits inferring topological properties of the solution space as a whole.

1 Introduction

A serial manipulator with n degrees of freedom (d.o.f.) can be described by an open kinematic chain with r revolute pairs and p prismatic pairs, with $n = r + p$. A manipulator configuration is determined by an n-dimensional vector $\mathbf{q} = (q_1, \ldots, q_n)$, where q_i is the joint coordinate corresponding to joint i. The forward kinematic function f associates each joint configuration vector $\mathbf{q}$ to an end-effector pose (in general, location and orientation), $\mathbf{x}$:

$$\mathbf{x} = f(\mathbf{q}). \tag{1}$$

The manipulator joint space, or configuration space, is an n-dimensional manifold with the structure $T^r \times R^p$, where T^r is an r-dimensional torus. The set of all possible poses $\mathbf{x}$ reachable by the end-effector, that we will call end-effector workspace, is a subset of $SE(3) = SO(3) \times R^3$, whose dimension, $m \leq 6$, depends on the manipulator's morfology and link-parameters. Thus, for example, in general planar, spherical and regional manipulators $m = 3$, and in spatial manipulators $m = 6$. Since the end-effector workspace cannot have a higher dimensionality than the configuration space, we must have $m \leq n$. We say that a manipulator is redundant if $n > m$, and the difference $\rho = n - m$ is the *redundancy* of the manipulator. In the case of $m = n$

J. Angeles et al. (eds.), Computational Kinematics, 85–94.

the manipulator is non-redundant, and for any given pose $\mathbf{x}$ there are in general a finite number of joint vectors $\mathbf{q}$ satisfying (1).

For a redundant manipulator, the different dimensionality between the joint space and the end-effector workspace implies that the forward kinematic function must define a many-to-one mapping where an infinite number of configurations $\mathbf{q}$ are mapped onto the same pose $\mathbf{x}$.

Of major interest in Kinematics is the study of the inverse kinematic function

$$\mathbf{q} = f^{-1}(\mathbf{x}), \tag{2}$$

that computes the set of configuration vectors that place the end-effector at a given pose.

The solution to the inverse kinematic function for a regular value $\mathbf{x}$ is one or more self-motion manifolds of dimension ρ (Burdick 1989).

The solution set can be mathematically characterized by a set of non-linear equations involving the joint variables. By using appropriate algebraic methods, as for example, those described in Buchberger (1989), it is possible, in principle, to derive a generalized input-output equation for any variable, i.e., a single equation involving only the desired variable and a set of ρ other variables that play the role of parameters. The parameterization of the solution set obtained in this way is not a "proper" one, in the sense that it is not clear beforehand what vectors of parameter values will provide an actual solution (the input-output equations may produce imaginary values). This is so since, in general, for a given pose of the end-effector, each joint variable has a restricted set of feasible values. In order to characterize the solution set, it is necessary to provide, besides the $n - \rho$ generalized input-output equations, the feasible ranges for the variables treated as parameters. In fact, this is not yet enough, since in general, the feasible set of values for a given parameter depends on the values already assigned to other parameters. A solution to this problem is to determine parameter ranges "on-line": before assigning a value to a parameter, its feasible range compatible with the already assigned ones has to be computed.

Note that if there is an effective way of computing compatible ranges for variables, solution vectors $\mathbf{q}$ may be obtained without any need of deriving input-output equations: Just extending the process to all variables in the loop (and not just to those taken as parameters) will provide the desired solution vector $\mathbf{q}$. This is adequate for those kinds of problems in which one or more particular solutions, and not the functional relationships between variables, are required. Furthermore, the solution may be directed by the user in the process of fixing variables into their ranges. Additional constraints, as externally imposed joint limitations, may be accounted for in a straightforward way.

The remaining of this paper is organized as follows: In Section 2 each inverse kinematic problem is associated with a kinematic loop, Sections 3 and 4 are devoted to the determination of feasible ranges for variables in planar and spherical loops, respectively, and Section 5 shows some consequences for the topology of the solution derived from the properties of the ranges.

2 Kinematic Loop Associated to an Inverse Kinematic Problem

When the end-effector of a manipulator is fixed at a given pose $\mathbf{x}$, the resulting kinematic chain becomes a loop with an extra imaginary link attaching the end-effector to the base link. The set of inverse kinematic solutions of the manipulator for this pose coincides with the configuration space of the corresponding loop. The redundance ρ of the manipulator corresponds to the mobility M of the loop, which, according to the well-known Kutzbach-Grübler formula (Duffy 1980), is:

$$M = \sum_{i=1}^{n} f_i - d,$$

where f_i is the number of d.o.f. of joint i, and d is, roughly speaking, the dimension of the loop's workspace. In our case $f_i = 1$ and $d = m$, which yields

$$M = n - m = \rho.$$

This formula fails to provide the correct mobility value in those loops obtained from a non-regular point of the end-effector workspace. In this case, the dimension of the workspace locally decreases to some value $m' < m$, and the actual mobility M' increases:

$$M' = n - m' > M.$$

Our purpose is the determination of the feasible ranges for variables in kinematic loops without explicit derivation of generalized input-output equations. We will provide explicit ranges for variables in arbitrary planar and spherical loops. The approach we take is broadly the same in both cases: First, *loop closure conditions* are derived for each kind of loop. Such closure conditions take the form of inequalities involving all link parameters in the loop, and they establish necessary and suficient conditions for the loop to be possible. To find the feasible range for a variable q_i in a given loop we consider a new loop obtained by formally substituting the two links sharing joint i by a single link whose parameters are a function of q_i.[1] The closure condition for this new loop is interpreted as a condition on q_i. The feasible range for variable q_i in the original loop is the set of values for which this condition is satisfied.

[1] In some cases, due to the conventions used in the link-parameters definition, the parameters affected may correspond to some other link.

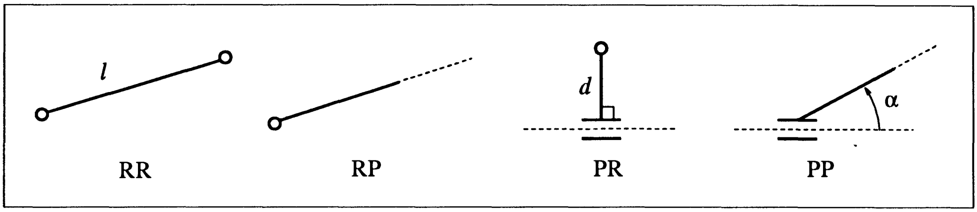


Figure 1: Representation of planar links.

3 Planar Loops

Each link in a planar loop can be characterized by no more than a single magnitude whose meaning depends on the type of joints it connects. Figure 1 shows the representation used for each type of link, and the parameters by means of which they are characterized.

According to these conventions, a planar loop may be represented in two alternative forms, depending on what direction is taken to define the loop. The difference comes from the uneven treatment given to RP and PR links. Figure 2 shows a loop represented in non-standard form, and its two possible normalized representations.

From simple geometric considerations, loop closure conditions to be satisfied by the links' parameters can be derived (see Celaya (1992) for details). Depending on the number of P pairs existing in the loop, different closure conditions hold:

1. **Loops with no prismatic pairs:**

A planar loop with n R pairs is characterized by n parameters l_i. The necessary and suficient closure condition can be expressed as:

$$2l_M \leq \sum_{i=1}^{n} l_i, \tag{3}$$

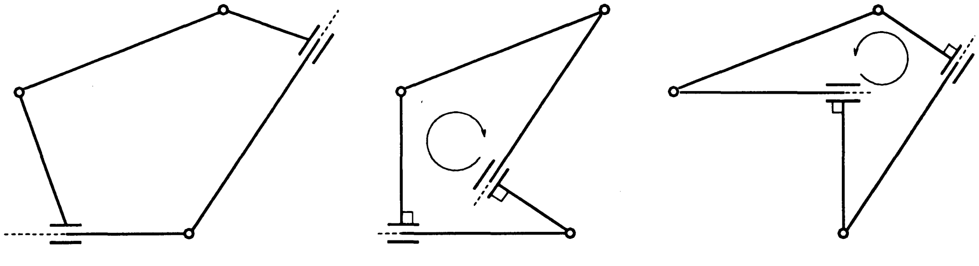


Figure 2: A planar loop and its two possible representations.

Kind of loop	Variable	Ranges for which there is a solution
nR	θ_k	$\frac{\delta^2-l_{k-1}^2-l_k^2}{2l_{k-1}l_k} \leq \cos\theta_k \leq \frac{\sigma^2-l_{k-1}^2-l_k^2}{2l_{k-1}l_k}$ $\sigma = \sum_{i\notin\{k,k-1\}} l_i$ $\delta = \max(0,\ 2l_{M'} - \sigma)$ $l_{M'} = \max(l_i,\ i \notin \{k, k-1\})$
nR-P	θ_1	$\frac{-\sum_{i=2}^{n-1} l_i-d}{l_1} \leq \cos\theta_1 \leq \frac{\sum_{i=2}^{n-1} l_i-d}{l_1}$
	θ_n	$\frac{d-\sum_{i=1}^{n-2} l_i}{l_{n-1}} \leq \sin\theta_n \leq \frac{d+\sum_{i=1}^{n-2} l_i}{l_{n-1}}$
	θ_k $(k \neq 1, n)$	$\frac{\delta^2-l_k^2-l_{k-1}^2}{2l_k l_{k-1}} \leq \cos\theta_k$ $\delta = \max(0,\ d - \sum_{i\notin\{k,k-1\}} l_i)$
	x	$\delta^2 - d^2 \leq x^2 \leq (\sum_{i=1}^{n-1} l_i)^2 - d^2$ $\delta = \max(0,\ 2l_M - \sum_{i=1}^{n-1} l_i)$ $l_M = \max(l_i,\ i \in \{1, \ldots, n-1\})$
nR-P-P	θ_k	No restriction if there are two or more R pairs
	x_k	$\mid x_k \sin\alpha \mid \leq \sum_{i=1}^{n-1} l_i$
n_1R-P-n_2R-P	θ_k	No restriction
	x_k	$\delta^2 - d_k^2 \leq x_k^2$ $\delta = \max(0,\ d_l - \sum l_i)$
...P...P...P	θ_k	No restriction if there are two or more R pairs
	x_k	No restriction

Table 1: Ranges for variables in planar loops.

where $l_M = \max(l_i, i \in \{1,\ldots,n\})$. An equivalent form to express this condition, that is more appropriate for the purpose of finding ranges for variables is:

$$2l_{M'} - \sum_{i\neq k} l_i \leq l_k \leq \sum_{i\neq k} l_i, \tag{4}$$

where $l_{M'} = \max(l_i, i \neq k)$.

2. **Loops with one prismatic pair:**

If the loop has n R pairs, the loop will have one PR link with parameter d, one RP link, which is characterized by no parameter, and $(n-1)$ RR links with parameters l_i. In this case, the closure condition is:

$$d \leq \sum_{i=1}^{n-1} l_i. \tag{5}$$

3. **Loops with two or more prismatic pairs:**

In this case there is no closure condition to be satisfied, provided there is at least one R pair. In the particular case of a loop with n P pairs and no R pairs, the loop is formed by n PP links with parameters α_i, and the closure condition is:

$$\cos(\sum_{i=1}^{n} \alpha_i) = 1 \tag{6}$$

Using these closure conditions, and following the general procedure described above, ranges for variables in any planar loop can be obtained (Celaya 1992). The feasible ranges for all possible cases are summarized in Table 1.

4 Spherical Loops

A spherical kinematic loop may be represented by an articulated spherical polygon in which sides correspond to links, and vertices correspond to revolute pairs. A link of length α in a spherical loop is kinematically equivalent to a link of length $2\pi - \alpha$ (Chiang 1988), thus the sides of the polygon may always be taken of length $\leq \pi$.

If, in a polygon with sides $\alpha_i \leq \pi$, we substitute one vertex p_i by its antipodal $\overline{p}_i$, sides α_{i-1} and α_i are replaced by $\pi - \alpha_{i-1}$ and $\pi - \alpha_i$, respectively. The different polygons obtained through these substitutions are called *supplementary polygons* following Chiang (1988), and they may be used to represent the same spherical loop. With an appropriate selection of the vertices or their antipodals, it is always possible to obtain a supplementary polygon with at most one side $> \pi/2$ representing the same spherical loop as the original polygon (Fig. 3).

The closure condition for a spherical loop with n links represented by a spherical polygon with sides $\alpha_i \in [0, \pi/2]$ for $i \neq k$, and $\alpha_k \in [0, \pi]$ is (see Celaya and Torras

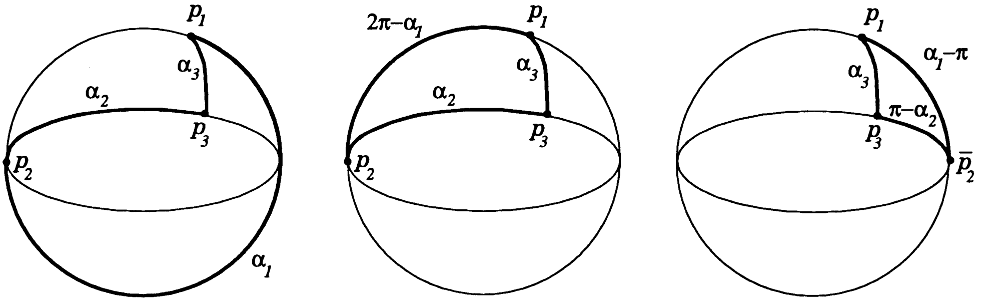


Figure 3: Supplementary spherical polygons.

(1990) or Celaya (1992) for a derivation):

$$2\alpha_M \leq \sum_{i=1}^{n} \alpha_i, \tag{7}$$

where $\alpha_M = \max(\alpha_1, \ldots, \alpha_n)$. This closure condition can be rewritten as:

$$2\alpha_{M'} - \sum_{i \neq k} \alpha_i \leq \alpha_k \leq \sum_{i \neq k} \alpha_i, \tag{8}$$

where $\alpha_{M'} = \max(\alpha_i, i \neq k)$.

Replacing sides α_{k-1} and α_k by a variable side $\overline{\alpha}$ (Figure 4) we have, using the cosine law for spherical triangles:

$$\cos \overline{\alpha} = \cos \alpha_{k-1} \cos \alpha_k - \sin \alpha_{k-1} \sin \alpha_k \cos \theta_k. \tag{9}$$

The closure condition for the new loop yields the following constraints for θ_k:

$$\frac{\cos \delta - \cos \alpha_{k-1} \cos \alpha_k}{-2 \sin \alpha_{k-1} \sin \alpha_k} \leq \cos \theta_k \leq \frac{\cos \sigma - \cos \alpha_{k-1} \cos \alpha_k}{-2 \sin \alpha_{k-1} \sin \alpha_k}, \tag{10}$$

where $\delta = \max(0, 2\alpha_M - \sum_{i \notin \{k-1,k\}} \alpha_i)$, with $\alpha_M = \max(\alpha_i,\ i \notin \{k-1, k\})$, and $\sigma = \min(\pi, \sum_{i \notin \{k-1,k\}} \alpha_i)$.

5 Properties of the Solution Sets

An examination of the conditions obtained for the different variables in all planar and spherical loops (Table 1 and eq. (10), respectively) reveals that in all cases, angular variables θ_k appear in sine or cosine functions, and translational variables x_k in absolute value or in squared form. The fact that the inverses of these functions are

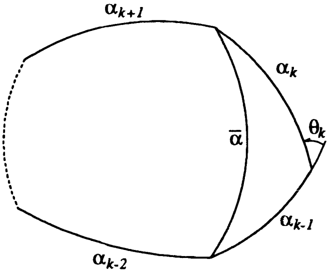


Figure 4: Finding ranges for θ_k in an spherical loop.

double-valued implies that the sets of feasible values for the variables are symmetric. More precisely, the set of permissible values S_i for any variable q_i must take one of these forms:

1. The empty set (unfeasible loop).
2. A single point or interval, that eventually may be the full range $[0, 2\pi]$ for rotational variables, or $(-\infty, \infty)$ for translational variables.
3. Two points or two intervals S_i^- and S_i^+ of the same length.

It is useful considering the mapping M_{ij} that for each value of variable q_i in a given loop gives the set of compatible values for q_j. Figure 5 shows the graphical representation of M_{ij} for a particular spherical loop. The image of each value a of q_i is obtained as the set of feasible values for q_j in the loop with one less link that results when q_i is fixed to the value a. The projection of the shaded region on the horizontal (resp. vertical) axis is the set S_i (resp. S_j) of allowed values for q_i (resp. q_j) in the original loop. Of course, the same sets S_i and S_j can be obtained directly by using the corresponding conditions for variables q_i and q_j in the original loop.

The above-mentioned properties of symmetry holding for S_i must hold also for the image $M_{ij}(a)$ of each value a of q_i, since this is the set of feasible values for variable q_j in a different loop. Note also that the intervals' extremes for θ_j vary continuously with the parameters of the links, and hence with the value of θ_i. In general we can derive the following property:

Prop. 1 *If the mapping M_{ij} has two disconnected regions, their projections on θ_i, S_i^+ and S_i^-, either coincide or form two non-intersecting intervals.*

This is so since a partial overlapping of S_i^+ and S_i^- would imply an assymetry at the extreme points of the overlapped regions. A direct consequence of Prop. 1 is:

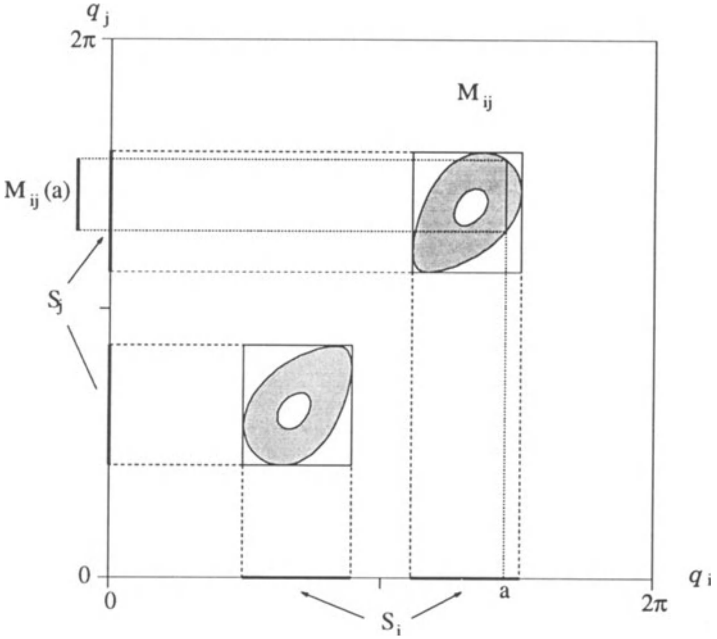


Figure 5: Set of compatible values for variables q_i and q_j in a given spherical loop.

Prop. 2 *A revolute joint is a crank (is able to perform complete turns) if and only if its feasible range is the complete interval* $[0, 2\pi]$.

This can be seen as a generalization of the Grashof condition (Hunt 1978) for planar and spherical loops with more than four links, and for planar loops with both R and P pairs.

Clearly, the existence of a variable with two disjoint feasible intervals implies that the configuration space of the loop has two disconnected parts, i.e., the loop has two modes of assembly. As a consequence of Prop. 1, the converse is also true:

Prop. 3 *The solution space is formed by a single connected manifold if and only if the feasible range for each variable is a single interval.*

The only circumstance in which Prop. 3 is not justified by Prop. 1 is the hypothetical case of a loop having two different modes of assembly, in which all variables have full mobility, thus avoiding the partial overlapping problem. We conjecture that such a situation is not possible.

6 Conclusions

Expressions giving the sets of feasible values for variables in all planar and spherical loops have been provided in explicit form. This results can be applied to finding inverse kinematic solutions of redundant planar and spherical manipulators. A simple examination of the feasible sets for variables permits deriving global properties of the solution set, as the number of self-motion manifolds.

This formulation based in ranges for variables lends itself to considering problems with multiple kinematic loops, an issue that has been investigated in Celaya and Torras (1992). Some extensions of these procedures to spatial kinematic chains can be found in Celaya (1992).

7 Acknowledgements

This work has been partially supported by the ESPRIT III Basic Research Action Program of the EC under contract No. 6546 (project PRoMotion).

References

Buchberger, B. 1989. Applications of Gröbner Bases in non-linear Computational Geometry. In *Geometric Reasoning*, edited by D. Kapur and L. Mundy. MIT Press. Cambridge, Massachusetts. pp. 413-446.

Burdick, J.W. 1989. On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. *Proc. of IEEE Conf. on Robotics and Automation*, pp 264-270.

Celaya, E., and Torras, C. 1990. Finding object configurations that satisfy spatial relationships. *Proc. of the ECAI'90*. Stockholm. pp. 147-152.

Celaya, E. 1992. Geometric reasoning for the determination of the position of objects linked by spatial relationships. Ph.D. dissertation, Universitat Politècnica de Catalunya.

Celaya, E., and Torras, C. 1992. Solving multiloop linkages with limited-range joints. *Mechanism and Machine Theory* (to appear).

Chiang, C.H. 1988. *Kinematics of spherical mechanisms.* Cambridge University Press. Cambridge.

Duffy, J. 1980. *Analysis of mechanisms and robot manipulators.* Edward Arnold. London.

Hunt, K.H. 1978. *Kinematic geometry of mechanisms.* Clarendon Press. Oxford.

The Self-Motion Manifolds of the N-bar Mechanism

Federico Thomas
Institut de Cibernètica (UPC – CSIC)
Diagonal 647, 2 planta
08028 Barcelona. SPAIN

1 Introduction

This paper investigates the global sets of solutions for single loop inverse kinematic problems containing only independent rotational and translational degrees of freedom, providing a rational and compact method to obtain these sets. This is a sequel of the work presented in Thomas (1992).

A forward kinematic function, **F**, is defined as a non-linear vector function which relates a set of n joint coordinates, Ω, of a closed kinematic chain so that

$$\mathbf{F}(\Omega) = \mathbf{I}, \tag{1}$$

where **I** is the identity displacement.

One of the primary problems of practical interest in kinematics is determining the inverse kinematic function, $\mathbf{F}^{-1}$, which computes the sets of joint coordinates that keep the mechanism closed. The problem is especially hard to solve when dealing with redundant kinematic chains, i.e. kinematic chains with extra degrees of freedom (d.o.f.). This paper essentially deals with the problem of obtaining these solution sets, keeping in mind that, for redundant mechanisms, there is an infinite number of joint coordinates that satisfy (1).

Kinematics of interconnected rigid bodies may lead to very complex computations and it is important to perform these computations in the most compact form and to search for their most rational organization. This goal motivates a great deal of research on the fundamental operations and the algebraic structures underlying kinematic methods. Nevertheless, no general satisfactory solution, convenient for practical use, has been found for the general positional inverse kinematic problem. This problem is highly complicated because of its non-linearity, the non-uniqueness of the solution, and the existence of singularities. This is why the redundant manipulator literature has focused on the linearized first order instantaneous kinematic relation between joint velocities, that is

$$\mathbf{J}(\Omega)\dot{\Omega} = 0 \tag{2}$$

J. Angeles et al. (eds.), Computational Kinematics, 95–106.
© 1993 *Kluwer Academic Publishers.*

where $\mathbf{J}(\Omega) = d\mathbf{F}(\Omega)/d\Omega$ is the Jacobian of the kinematic chain. In this literature, the inverse solution of (2) is often referred to as the inverse kinematic solution, rather than that of (1). Thus, given the position and velocity states, the set of joint coordinates can be obtained either by directly solving positional equations (1), or by solving the first-order differential equations (2). Since (2) at a particular position is linear, numerical solutions to the inverse velocity problem are relatively easier than that of the inverse position problem. Nevertheless, practical applications, including most industrial robot coordination algorithms, avoid numerical inversion of the Jacobian by using analytical inverses developed on an ad hoc basis.

In general, ad hoc methods make use of simplifying heuristic arguments.These simplifying arguments have made ad hoc analytical inverses dominate many of the practical applications in this area.

The first foundation for a unified theory of analysis of spatial mechanisms was provided in Duffy (1975). Nevertheless, the algebra used prevents any intuitive interpretation of the results, making any further insight difficult. The present work presents an alternative foundation for a unified theory of analysis of spatial mechanisms based on simple results.

This paper is structured as follows. Section 2 states the inverse positional kinematic problem in terms of 3 × 3 dual-number matrices. Section 3 is devoted to finding the expressions of the components of rotation and translation of a kinematic equation used throughout this paper. Section 4 investigates the sets of solutions of the rotations equation. It is shown that this problem is equivalent to that of finding a global solution to the positional kinematic problem of the orthogonal spherical mechanisms. Section 5 deals with the problem of obtaining those points of the sets of solutions of the rotations equation that also satisfy the equation of translations. Section 6 deals with the problem of parameterizing the sets of solutions of the rotations equations. Section 7 presents two examples whose analysis is carried out using the developed methodology and, finally, Section 8 provides a brief summary of the main points in this paper.

2 Stating the problem

The general inverse kinematic problem will be formulated in terms of 3 × 3 dual matrices (see, for example, Duffy 1975, or Veldkamp 1976). The dual matrix formulation is preferred here to the popular homogeneous coordinate form because, for analytical manipulation, the formulation obtained for the separation of the rotational and translational parts is essential herein.

A kinematic chain is defined as a set of n links in series. The proportions of link i will be specified by a constant dual angle $\hat{\phi}_i = \phi_i + \varepsilon d_i$ between the two adjacent joint axes. The dual operator is defined by $\varepsilon^m = 0$ where m is any integer greater than one. The parameters ϕ_i and d_i are referred to as the twist angle and the length of link i, respectively.

Neighboring links have a common joint axis between them. One parameter of interconnection is the distance along this common axis from one link to the next. Thus, the displacement linking the reference frame of element i with that of element $i + 1$ can be expressed as:

$$\mathbf{Rx}(\alpha_i + \varepsilon x_i)\ \mathbf{Ry}(\beta_i + \varepsilon y_i), \tag{3}$$

or, in other words,

$$\mathbf{Rx}(\alpha_i + \varepsilon x_i)\ \mathbf{X}\ \mathbf{Rx}(\beta_i + \pi + \varepsilon y_i)\ \mathbf{X}\ \mathbf{Rx}(\pi). \tag{4}$$

where

$$\mathbf{X} = \mathbf{Rz}(\pi/2) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

Thus, any kinematic chain can be described kinematically by giving the values of four quantities for each link. Two describe the link itself, and the other two describe the link's connection to a neighboring link (Craig 1986, page 64). In what follows, we will consider the four quantities as variables. This way we can model any kinematic chain by constraining the appropriate rotational or translational d.o.f.

If we define

$$\mathbf{B}(\hat{\phi}_i) = \mathbf{Rx}(\hat{\phi}_i)\ \mathbf{X} = \left(\mathbf{Rx}(\phi_i) + d_i\varepsilon \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin\phi_i & -\cos\phi_i \\ 0 & \cos\phi_i & -\sin\phi_i \end{pmatrix}\right)\mathbf{X}, \tag{6}$$

the kinematic equation for a closed kinematic chain is an equation of the form:

$$\mathbf{F}(\hat{\varPhi}) = \prod_{i=1}^{n} \mathbf{B}(\hat{\phi}_i) = \mathbf{I}. \tag{7}$$

This equation can be interpreted as the loop equation of the n-bar mechanism (Thomas 1992), where $\hat{\varPhi} = (\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_n) = (\phi_1 + \varepsilon d_1, \phi_2 + \varepsilon d_2, \ldots, \phi_n + \varepsilon d_n)$ is called the *vector of displacements*; $\varPhi = (\phi_1, \phi_2, \ldots \phi_n)$, the *vector of rotations*; and $D = (d_1, d_2, \ldots d_n)$, the *vector of translations.*

3 Equating real and dual parts

Some proposed symbolic methods for solving simple inverse kinematic problems can be factored into a solution for the rotational component and a solution for the translational component (thomas and Torras 1988). Notice that the rotation component can be extracted directly from the original equation by simply removing all the translations, but not the translational component. The approach taken here also follows this sequencing strategy. Under the dual number formalism, both components can be obtained by equating the real and dual parts of (7), respectively.

Equating the real parts in (7), we have:

$$\mathbf{F}(\varPhi) \;=\; \prod_{i=1}^{n} \mathbf{B}(\phi_i) \;=\; \mathbf{I} \tag{8}$$

which will be called the *equation of rotations.*

The partial derivatives of $\mathbf{F}(\Phi)$ with respect to ϕ_i can be expressed as:

$$\frac{\partial \mathbf{F}(\Phi)}{\partial \phi_i} = \mathbf{A}_i \; \mathbf{Q} \; \mathbf{A}_i^t \tag{9}$$

where

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{A}_i = \begin{cases} \mathbf{I}, & i = 1 \\ \prod_{j=1}^{i-1} \mathbf{B}(\phi_j), & i > 1 \end{cases} \tag{10}$$

Thus, we define $\nabla \mathbf{F}(\Phi)$ as

$$\nabla \mathbf{F}(\Phi) = (\mathbf{A}_1 \mathbf{Q} \mathbf{A}_1^t, \; \mathbf{A}_2 \mathbf{Q} \mathbf{A}_2^t, \ldots, \mathbf{A}_n \mathbf{Q} \mathbf{A}_n^t). \tag{11}$$

On the other hand, the linearization of equation (8) around a solution $\Phi_0 = (\phi_1^0, \ldots, \phi_n^0)$ can be expressed as:

$$\sum_{i=1}^{n} \Delta\phi_i \; \mathbf{A}_i \mathbf{Q} \mathbf{A}_i^t = 0 \tag{12}$$

where $\Delta\phi_i = (\phi_i - \phi_i^0)$. In other words,

$$\nabla \mathbf{F}(\Phi_0)(\Phi - \Phi_0)^t = \nabla \mathbf{F}(\Phi_0) \Delta\Phi^t = 0 \tag{13}$$

This is called the *equation of approximation*, which agrees with the result obtained in Uicker (1964). Actually, equation (12) is the first order approximation of $\mathbf{F}(\Phi) = \mathbf{I}$ around Φ_0, which defines a *hyperplane of approximation.*

Now, equating the dual parts in (7) and taking into account that $\varepsilon^2 = 0$, it can be easily shown that

$$\nabla \mathbf{F}(\Phi) D^t = 0 \tag{14}$$

which will be called the *equation of translations.*

Note that the *superposition principle* applies for the translational solutions, that is, if for a given vector of rotations we have two vectors of translation that satisfy equation (14), say D_1 and D_2, then any vector of translations which can be expressed as $D_3 = \mu_1 D_1 + \mu_2 D_2$, $\forall \mu_1, \mu_2 \in \Re$ also satisfies it.

Considering that there are three translational degrees of freedom in space, it should be expected that there be three equations representing this information. Actually, if

$$\mathbf{A}_i = (\mathbf{n}_i \; \mathbf{o}_i \; \mathbf{a}_i) = \begin{pmatrix} \mathbf{n}_{ix} & \mathbf{o}_{ix} & \mathbf{a}_{ix} \\ \mathbf{n}_{iy} & \mathbf{o}_{iy} & \mathbf{a}_{iy} \\ \mathbf{n}_{iz} & \mathbf{o}_{iz} & \mathbf{a}_{iz} \end{pmatrix}, \tag{15}$$

then it can be proved that

$$\mathbf{A}_i \mathbf{Q} \mathbf{A}_i^t = \begin{pmatrix} 0 & -\mathbf{n}_{iz} & \mathbf{n}_{iy} \\ \mathbf{n}_{iz} & 0 & -\mathbf{n}_{ix} \\ -\mathbf{n}_{iy} & \mathbf{n}_{ix} & 0 \end{pmatrix}. \tag{16}$$

Then, if we define the vectors $\mathbf{A}_i\widehat{\mathbf{Q}\mathbf{A}}_i^t = (\mathbf{n}_{ix}, \mathbf{n}_{iy}, \mathbf{n}_{iz})^t$, and we express the equation of approximation and the equation of translations in terms of them, we get the three equations we were expecting. Actually, a deeper insight reveals that vector $\mathbf{A}_i\widehat{\mathbf{Q}\mathbf{A}}_i^t$ in $\Re^3$ is a unit vector pointing in the positive direction of translation d_i with respect to the first bar of the n-bar mechanism. Thus, the equation of translations can also be expressed as:

$$\begin{pmatrix} \mathbf{n}_{1x} & \mathbf{n}_{2x} & \dots & \mathbf{n}_{nx} \\ \mathbf{n}_{1y} & \mathbf{n}_{2y} & \dots & \mathbf{n}_{ny} \\ \mathbf{n}_{1z} & \mathbf{n}_{2z} & \dots & \mathbf{n}_{nz} \end{pmatrix} D^t = \mathbf{J}(\Phi)D^t = 0. \tag{17}$$

Likewise, for the equation of approximation. It can be checked that $\mathbf{J}(\Phi)$ is the $3 \times n$ Jacobian matrix of the spatial transformation $\mathbf{F}(\phi)$, that is:

$$\mathbf{J}(\Phi) = \frac{\mathrm{d}\mathbf{F}(\Phi)}{\mathrm{d}\Phi}. \tag{18}$$

This formulation leads to the following remark.

Remark I. The solution of the translations equation of the n-bar mechanism (13) is the Jacobian null space of the corresponding rotations equation (8).

If the following condition is satisfied

$$\max_{\Phi} \ (\text{rank } \ \mathbf{J}(\Phi)) = 3, \tag{19}$$

it is said that the degree of redundancy of the closed spherical kinematic chain is $r = n - 3$. If

$$\text{rank } \ \mathbf{J}(\Phi) = 2, \tag{20}$$

for some Φ, then we say that the closed kinematic chain is in a singular state (all the n bars are on a plane). In others words, the vectors $\mathbf{A}_i\widehat{\mathbf{Q}\mathbf{A}}_i^t$, i.e. the columns of the Jacobian matrix, define a subspace of $\Re^3$ which coincides with $\Re^3$ iff the mechanism is not in a singular state.

Since all the axes of rotations of an n-bar mechanism can never be arranged so that they all keep aligned, rank $(\mathbf{J}(\Phi))$ cannot be lower than two.

Let $N(\mathbf{J})$ be the null space of the linear mapping $\mathbf{J}$. Any element of this subspace is mapped into the zero vector. If the Jacobian is of full rank, the dimension of the null space, $\dim(N(\mathbf{J}))$, is the same as r, the degrees of redundancy. When the Jacobian is degenerate, its rank decreases, and the dimension of the null space increases by the same amount. The sum of the two is always equal to n, that is:

$$\text{rank}(\mathbf{J}) + \dim(N(\mathbf{J})) = n. \tag{21}$$

The solution to equation (17) involves the same number of arbitrary parameters as the dimension of the null space.

4 Self-motion sets and singularities

Equation (8) corresponds to the loop equation of a orthogonal spherical mechanism. The configuration space, C, for a spherical mechanism is a product space formed by the n-fold product of the individual variables of rotation, that is, $C = S^1 \times S^1 \times \ldots \times S^1 = T^n$, where T^n is an n-torus, which is a compact n-dimensional manifold. The space of pointing directions in three-space is two-dimensional and can be represented by the set of unit vectors in three-space or, equivalently, by the surface of a sphere. In contrast, the configuration space is an n-torus. The mismatch between the topological properties of a sphere and a torus prevents the construction of a singularity-free inverse kinematic mapping (Baker and Wampler 1988).

For non-redundant kinematic chains, there is a finite set of configurations that solve the inverse kinematic problem. For redundant ones, there is an infinite number of configurations, which can be grouped into regions of the configuration space with the structure of algebraic sets of dimension r.

The term "self-motion sets" was coined by Burdick in Burdick (1989) motivated by the fact that any trajectory inside these sets corresponds to a continuous motion of the elements of the closed kinematic chain which does not require to open it.

Formally, let a redundant inverse kinematic solution of a rotation equation be denoted as the union of disjoint r-dimensional algebraic sets

$$\mathbf{F}(\Phi) = \mathbf{I} \;\Rightarrow\; \Phi = \bigcup_i S_i, \tag{22}$$

where S_i is the i^{th} r-dimensional connected algebraic set and $S_i \cap S_j = \emptyset$ when $i \neq j$.

Self-motion sets can be seen as smooth hypersurfaces of dimension r intersecting themselves. The stratification of these sets leads to several manifolds of dimension r, which will be denoted M_i, connected through manifolds of lower dimension. The former will be called *self-motion manifolds*, and the latter just *singularities*. Note that, motivated by the presence of these singularities, the definition of self-motion manifolds is a rather different definition from the one introduced in Burdick (1989).

Bounds on the number of self-motions sets are also discussed in Burdick (1989), where it is shown that a redundant kinematic chain can have no more self-motions sets than the maximum number of inverse kinematic solutions of a non-redundant kinematic chain of the same class.

In Yoshikawa (1986), Thomas and Torras (1988), or Wampler (1989) it is shown that a discrete closed-form solution exists for spherical mechanisms with up to three d.o.f. For $n = 3$, there are two discrete solutions. For $n > 3$, a spherical kinematic chain becomes redundant. Thus, regardless of the number of d.o.f., a spherical redundant mechanism can have at most two distinct self-motion sets.

Self-motion manifolds can be parameterized, at least locally, by a set of r independent parameters, say $\Psi = \{\psi_1, \ldots, \psi_r\}$, so that distinct self-motions can be generated by continuously sweeping ψ_i through their range.

As already noted in Duffy (1975), equation (8) has a straightforward geometric interpretation as an n-sided spherical polygon. Consider a unit radius sphere centered at the coordinate origin. As a result of applying successive rotations, the $z-$axis will describe on the surface of the sphere a spherical polygon with sides (*arcs of great circles*) of length ϕ_i, and exterior angles $\pi/2$. Alternatively, the y-axis will describe a spherical polygon with sides of length $\pi/2$ and exterior angles ϕ_i. These two polygons are considered duals. All theorems from spherical trigonometry are thus applicable; in particular, the sine, cosine, and sine-cosine laws, which are the the three basic laws for spherical triangles. By triangulating the thus obtained n-sided spherical polygon, the global solutions to equation (8) will be obtained in terms of r independent parameters. This is not the only possible parametrization. Actually, we can take any r variables as parameters. We will discuss in detail this latter parameterization in section 6, but, for the moment, let us assume that a set of expressions of the ϕ_i, $i = 1 \ldots n$, parameterized in terms of $n-3$ independent parameters, is obtained, so that they yield one solution for every choice of the parameters.

5 Satisfaying the equation of translations

The set of all tangent vectors to a manifold M at a fixed point Φ_0 is called the tangent space to the manifold M at point Φ_0. This set is denoted $T_{\Phi_0}(M)$, which can be endowed with the structure of a linear space. The tangent space to M_i at some $\Phi_0 = \Phi(\Psi_0)$ in terms of the independent parameters is:

$$T_{\Phi_0}(M_i) = \frac{dM_i(\Phi_0)}{d\Psi} = \frac{d\Phi(\Psi_0)}{d\Psi} \in \Re^{n\times(n-3)} \tag{23}$$

where $d\Psi \in \Re^{n-3}$. Then, by applying the chain rule,

$$\frac{d\mathbf{F}(\Phi_0)}{d\Psi} = \frac{d\mathbf{F}(\Phi(\Psi_0))}{d\Phi}\frac{d\Phi(\Psi_0)}{d\Psi} = \mathbf{J}(\Phi_0)\frac{d\Phi(\Psi_0)}{d\Psi} = \mathbf{J}(\Phi_0)T_{\Phi_0}(M_i) \in \Re^{3\times(n-3)}. \tag{24}$$

However, when $\mathbf{F}$ is restricted to a self-motion manifold, $\mathbf{F}$ must equal the identity displacement, $d\mathbf{F}(\Phi_0)/d\Psi$ must be zero, and therefore the tangent space of $M_i(\Psi)$ at $\Psi = \Psi_0$, must be in the null space of the Jacobian evaluated at this point. Thus, since the null space of the Jacobian is the solution of the equation of rotations, the tangent space of $M_i(\Psi)$ is part of the solution of the translations equation.

If $\Phi(\Psi_0)$ is a point on the self-motion manifold, then

$$\operatorname{rank}\left(\frac{d\Phi(\Psi_0)}{d\Psi}\right) = N\left(\mathbf{J}\left(\Phi(\Psi_0)\right)\right) = r. \tag{25}$$

If $\Phi(\Psi_0)$ is a point of a singularity, then the dimension of $N(\mathbf{J}(\Phi(\Psi_0)))$ is greater than r and $d\Phi(\Psi_0)/d\Psi$ is not strictly defined. This leads to the following remark:

Remark II. For the n-bar mechanism, the vector of translations that satisfies the equation of translations at $\Phi(\Psi_0)$, restricted to a self-motion manifold of the equation of rotations, is any vector in the subspace of dimension $n-3$ defined by $d\Phi(\Psi_0)/d\Psi$.

The set of all tangent vectors $T_{\Phi_0}(M_i)$ to a manifold at a point Φ_0 is a linear space of the same dimension as that of M_i. The whole set of tangent vectors to a manifold M_i is denoted $T(M_i)$ and called the *tangent bundle* of M_i. Then, if the solution of the rotations equation can be represented by the self-motion manifolds M_i, the solution of the translations equation is the tangent bundle of M_i, that is $T(M_i)$. Hence,

$$D = \lambda_1 \frac{\partial\Phi(\Psi_0)}{\partial\psi_1} + \ldots + \lambda_r \frac{\partial\Phi(\Psi_0)}{\partial\psi_r}, \qquad \forall\lambda_1, \ldots, \lambda_r \in \Re. \tag{26}$$

Following differential geometry nomenclature, the numbers $(\lambda_1, \ldots, \lambda_r)$ are called coordinates of the tangent vector D to M_i at point $\Phi(\Psi_o)$ in a local coordinate system $(\psi_1, \ldots, \psi_r)$ on the surface M_i, and equation (26) is called the *tensor law of coordinate transformation for the tangent vector D.*

At a point belonging to a singularity, the Jacobian becomes degenerate, that is, the dimension of its null space is greater than r. Hence, the associated spherical mechanism becomes planar (all axis of rotation lie on a plane), and, since an n-bar mechanism can degenerate to planar only when n is even, it is easy to infere that there are no singularities when n is odd.

Given a point on a singularity, there is a set of arbitrarily close points on the self-motion manifolds for which we know the allowed translations. Then, by applying the superposition principle for the translational component, the solution at a point of the singularities can be expressed as a linear combination of the solution for each of these points.

6 Local parameterizations

In this section, we concentrate ourselves in obtaining simple local parameterizations, or *local charts*, useful for local analysis of the self-motion manifolds.

Since the rank of the Jacobian matrix is maximal at every point $\Phi_0 \in M_c$ (i.e. rank of $J(\Phi_0) = 3$), M_c is a r-dimensional smooth manifold of class C^∞, and r coordinates of the surrounding space T^n can be taken as local coordinates in the neighborhood of each point $\Phi_0 \in M_c$. This is, in fact, the implicit function theorem formulated in convenient terms whose proof can be found in any text book of differential geometry. In what follows we will study this simple parameterization.

Let us take r consecutive variables in the chain as parameters. Without loss of generality, let $\{\phi_1, \phi_2, \ldots, \phi_r\}$ be the set of parameters. Hence, the equation of rotations can be expressed as:

$$\mathbf{Rx}(\phi_{r+1}) \ \mathbf{X} \ \mathbf{Rx}(\phi_{r+2}) \ \mathbf{X} \ \mathbf{Rx}(\phi_{r+3}) \ = \ \mathbf{A}, \tag{27}$$

which has always solution for any proper orthogonal matrix $\mathbf{A}$ encompassing all the parameters. In general, this equation has the following two discrete solutions:

$$\begin{aligned} \phi_{r+1} &= \operatorname{atan2}(\pm a_{31}, \mp a_{21}) \\ \phi_{r+2} &= \operatorname{atan2}(\pm\sqrt{1 - a_{11}^2}, a_{11}) \\ \phi_{r+3} &= \operatorname{atan2}(\mp a_{13}, \mp a_{12}) \end{aligned} \tag{28}$$

where a_{ij} denotes the element (i, j) of $\mathbf{A}$. One solution is obtained by taking the upper row of signs, and the other, the lower one.

When $a_{11} = \pm 1$, there appear infinite solutions. Those points of the self-motion manifolds where this happens are called singularities of the parametrization, and it can be easily shown that they correspond to those situations in which the axes of rotation of bars $r + 1$, $r + 2$, and $r + 3$ lie on the same plane.

Using this parameterization, two different sets of values for the parameters might lead to the same point on the self-motion manifold. Therefore, this mapping is not inyective but, by constraining its domain, we can get a local chart (this will become clear in the second example of the next section). Let Ψ_0 be a point of the space of parameters that maps onto Φ_o. This mapping is regular because at least one of the minors of order 3 × 3 of the Jacobian matrix is different from zero in $\varPsi_0$. Since this minor is continuous, it is different from zero in a neighborhood of $\varPsi_0$. Therefore, we conclude that there will be an open subset in the space of parameters containing $\varPsi_0$ in which the mapping in inyective. The highest the value of n, the *smaller* the size of this set. This is why, as $n \longrightarrow \infty$ (that is, the n-bar mechanism behaves like a rubber band), this parametrization becomes useless.

Thus, taking the proper r consecutive rotational variables as parameters, we always obtain a local parameterization of the self-motion manifold. This leads to the following remark:

Remark III. The set of n local charts consisting of r consecutive variables used as parameters constitutes an atlas of the self-motion manifolds.

Obviously, certain points are provided with several local coordinate systems.

7 Examples

This section contains two examples. The first one is a simple global analysis, including the stratification of the resulting self-motion sets and the analysis of the singularities. The second one is about a local analysis using the parameterization described in the last section, including the situation in which it is singular.

7.1 Global analysis of the 4-bar mechanism

The kinematic equation to solve for the 4-bar mechanism is:

$$\prod_{i=1}^{4} \mathbf{Rx}(\phi_i + \varepsilon d_i)\ \mathbf{X} = \mathbf{I}. \tag{29}$$

It is a well-known result that a rotation can be resolved into three successive rotations about perpendicular axes of rotation. In doing so, we get

$$\left.\begin{array}{l} \phi_1 = \pm\phi_3 \\ \phi_2 = \pm\phi_4 \end{array}\right\} \tag{30}$$

The stratification of this algebraic set leads to 4 manifolds of dimension 1 and 4 of dimension 0. As shown in the following diagram, the strata of dimension 1 are connected through those of dimension 0.

$$\begin{tikzcd}
\left.\begin{array}{c} \phi_1 - \phi_3 = 0 \\ \phi_2 = \phi_4 = 0 \end{array}\right\} & \varPhi_0 = (0,0,0,0) \arrow[l] \arrow[r] & \left.\begin{array}{c} \phi_2 - \phi_4 = 0 \\ \phi_1 = \phi_3 = 0 \end{array}\right\} \\
\varPhi_1 = (\pi,0,\pi,0) \arrow[u] \arrow[d] & & \varPhi_2 = (0,\pi,0,\pi) \arrow[u] \arrow[d] \\
\left.\begin{array}{c} \phi_2 + \phi_4 = 0 \\ \phi_1 = \phi_3 = \pi \end{array}\right\} & \varPhi_3 = (\pi,\pi,\pi,\pi) \arrow[l] \arrow[r] & \left.\begin{array}{c} \phi_1 + \phi_3 = 0 \\ \phi_2 = \phi_4 = \pi \end{array}\right\}
\end{tikzcd} \tag{31}$$

The tangent spaces of these strata are constant. For example, the tangent space of the upper left stratum on diagram (31) is $(1,0,1,0)$. Thus, $D_1 = \lambda_1(1,0,1,0)$, $lambda_1 \in \Re$, is the only possible solution of the corresponding equation of translations for any point on this stratum. For the other strata, the solutions are: $D_2 = \lambda_2(0,1,0,1)$, for the upper right stratum; $D_3 = \lambda_3(1,0,-1,0)$, for the lower left stratum; and $D_4 = \lambda_4(0,1,0,-1)$ for the lower right stratum.

Let us just see what happens at the singularity $\Phi_0 = (0,0,0,0)$. In this point the Jacobian is:

$$\mathbf{J}(\Phi_0) = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{32}$$

which is clearly singular, as expected. Since its rank is two, the 4-bar mechanism is planar, also as expected.

By applying the superposition principle at this singular point (i.e. the solution to the translations equation at this point can be expressed as a linear combination of those at the non-singular points on their boundaries), we get:

$$D_5 = \lambda_1(0,1,0,1) + \lambda_2(1,0,1,0), \quad \forall \lambda_1, \lambda_2 \in \Re, \tag{33}$$

which is clearly the solution sought.

7.2 Local analysis of the 5-bar mechanism

The equation to solve for a 5-bar mechanism is:

$$\prod_{i=1}^{5} \mathbf{Rx}(\phi_i + \epsilon d_i)\mathbf{X} = \mathbf{I}. \tag{34}$$

A solution to its rotational component is $\Phi_0 = (0, -\frac{\pi}{4}, \frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{4})$. At this point the Jacobian is:

$$J(\Phi(\Psi_0)) = \begin{pmatrix} 1 & 0 & -0.707 & -0.707 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -0.707 & 0.707 & 0 \end{pmatrix} \tag{35}$$

This point is located on a self-motion manifold of dimension 2 embedded in T^5. Now, let us assume that we want to know the allowed translational d.o.f. for this particular set of rotational values. To this end, we can obtain the tangent space around Φ_0 using a finite differences technique as follows.

We can take $\psi_1 = \phi_1$ and $\psi_2 = \phi_2$ as parameters of the self-motion manifold. This choice is obviously arbitrary. Note that given a set of values for the parameters, there are two solutions for the other rotational d.o.f., but choosing the closest points to Φ_0, we get:

$$\Phi_1(\psi_1 = \Delta, \psi_2 = -\frac{\pi}{4}) = (0.001, -0.78539, 1.5715034, -1.570089, 0.78539) \tag{36}$$

$$\Phi_2(\psi_1 = 0, \psi_2 = -\frac{\pi}{4} + \Delta) = (0, -0.78439, 1.57079, -1.57079, 0.7863) \tag{37}$$

with $\Delta = 0.001$. Thus,

$$\frac{\partial M_0}{\partial \psi_1} \simeq \frac{\Phi_1 - \Phi_0}{\Delta} = (1, 0, 0.7071, 0.7071, 0) \tag{38}$$

and

$$\frac{\partial M_0}{\partial \psi_2} \simeq \frac{\Phi_2 - \Phi_0}{\Delta} = (0, 1, 0, 0, 1). \tag{39}$$

Since

$$D(\Phi = \Phi_0) = \lambda_1 \frac{\partial M_0}{\partial \psi_1} + \lambda_2 \frac{\partial M_0}{\partial \psi_2}, \qquad \lambda_1, \lambda_2 \in \Re, \tag{40}$$

finally we get,

$$\begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{pmatrix} = \lambda_1 \begin{pmatrix} 1 \\ 0 \\ 0.7071 \\ 0.7071 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{41}$$

Thus, $d_2 = d_5$, and $d_3 = d_4 = 0.7071 \cdot d_1$, which certainly corresponds to solution.

It is important to realize that, if we would take ϕ_3 and ϕ_4 as parameters, we would be in a singular point of this particular parametrization.

8 Conclusions

We have proved that the information required for the analysis of any single loop spatial mechanism is essentially stored in the self-motion sets of the orthogonal spherical mechanism. Thus, it has been shown the great relevance of deepening on the structure of the self-motion manifolds of the orthogonal redundant spherical mechanisms, and how a thorough understanding of them is very helpful in the study of spatial mechanisms.

Any kinematic loop equation can be modeled as the loop equation derived from an n-bar mechanism by taking as many bars as needed and constraining some of the resulting d.o.f. It has been shown that the set of angle solutions of a n-bar mechanism can be obtained by computing the tangent space of the self-motion manifolds of the orthogonal redundant spherical mechanism.

Acknowledgements

This research has been partially supported by the ESPRIT III Basic Research Actions Program of the EC under contract No. 6546 (project PROMotion).

References

Baker, D.R., and Wampler, C.W. 1988. On the Inverse Kinematics of Redundant Manipulators, *The International Journal of Robotics Research*, Vol. 7, No. 2, pp. 3-21, Mach/April.

Burdick, J.W. 1989. On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds, *1989 IEEE Int. Conf. on Robotics and Automation*, pp. 264-270.

Craig, J.J. 1986. *Introduction to Robotics. Mechanics and Control*, Addison-Wesley Publish. Comp.

Duffy, J., and Rooney, J. 1975. A Foundation for a Unified Theory of Analysis of Spatial Mechanisms, *Transactions of the ASME, Journal of Engineering for Industry*, November, pp. 1159-1165.

Thomas, F. 1992. On the N-bar Mechanism, or How to Find Global Solutions to Single Loop Spatial Kinematic Chains, *1992 International Conference on Robotics and Automation*, Nice, France, May.

Thomas, F., and Torras, C. 1988. A Group Theoretic Approach to the Computation of Symbolic Part Relations, *IEEE Journal of Robotics and Automation*, vol. 4, no. 6, December.

Uicker, J.J., Denavit, J., and Hartenberg, R.S. 1964. An Iterative Method for the Displacement Analysis of Spatial Mechanisms, *Transactions of the ASME, Journal of Applied Mechanics*, June, pp. 309-314.

Veldkamp, G.R. 1976. On the Use of Dual Numbers, Vectors and Matrices in Instantaneous, Spatial Kinematics, *Mechanisms and Machine Theory*, vol. 11, pp. 141-156.

Yoshikawa, T. 1986. Manipulability of Robotic Mechanisms, *Robotics Research – The Second International Symposium*, edited by H. Hanafusa and H. Inoue, Cambridge, Mass.: MIT Press, pp. 439-446.

Wampler, C.W., 1989. Inverse Kinematic Functions for Redundant Spherical Wrists, *IEEE Trans. on Robotics ans Automation*, vol. 5, no. 1, pp. 106-111, February.

Part 3

Kinematic and Dynamic Control

3.1 T.H. Connolly and F. Pfeiffer
Feedforward Torque Computations with the Aid of Maple V

3.2 C. Woernle
Nonlinear Control of Constrained Redundant Manipulators

3.3 M. Shoham and V. Brodsky
Analysis of Mechanisms by the Dual Inertia Operator

Feedforward Torque Computations with the Aid of Maple V

Thomas H. Connolly and Friedrich Pfeiffer
Lehrstuhl B für Mechanik
Technische Universität München
80333 München, Germany

Abstract

Using the symbolic manipulation software package Maple V, the kinematic and dynamic representations needed to design a feedforward torque controller are derived. The derived kinematic representations include the inverse kinematics, the Jacobian and its time-derivative, which are needed to convert a task space trajectory to the joint level. Two Newton-Euler based recursive multibody dynamics algorithms are also programmed symbolically; one for the inverse dynamics to design the feedforward torque controller and one for the forward dynamics to validate the controller. The symbolic computations are demonstrated on a six degree of freedom PUMA 560 manipulator.

1 Introduction

Kinematic and dynamic representations of robotic manipulators are regularly necessary for their control. For multilink manipulators these representations are algebraically complex, making hand calculation impractical. The development of symbolic manipulation software such as MACSYMA, Mathematica and Maple V has made the derivation of complex kinematic and dynamic systems less time-consuming and less error prone.

Engineers have made use of commercially available symbolic manipulation software and have also written problem specific software for deriving kinematic and dynamic expressions of mechanical systems. Manocha and Canny (1992) developed an algorithm for solving the inverse kinematics of a general 6R manipulator with the aid of symbolic computation. While optimizing the design of a redundant manipulator, Mayorga et al. (1992) used symbolic manipulation software to develop a Jacobian based objective function for their minimization procedure. Custom symbolic manipulation software has also been written to automatically generate the forward and inverse kinematics of a manipulator from its Denavit-Hartenberg parameters (Herrera-Bendezu et al. 1988; Rieseler and Wahl 1990). The equations of motion

J. Angeles et al. (eds.), Computational Kinematics, 109–117.

for manipulators have also been written making use of symbolic manipulation. Lieh (1991) made use of Maple for developing the equations of motion for an elastic manipulator. Armstrong et al. (1986) developed an explicit dynamic model of a PUMA 560 manipulator using the software EMDEG. Symbolic computation software has also been used in control systems design. Van Essen and De Jager (1993) used Maple V to analyze and design nonlinear control systems, and to check for stability of nonlinear systems Rothfuß et al. (1993) used MACSYMA to assist in the extensive analytical and numerical calculations.

Computed feedforward torque controllers have been demonstrated to provide good tracking performance (Leahy 1988; Tarn et al. 1993) in robotic control systems. The structure of such a controller is shown in Equation (1).

$$\tau = M(\Theta_d)\ddot{\Theta}_d + V(\Theta_d, \dot{\Theta}_d) + G(\Theta_d) + K_p(\Theta - \Theta_d) + K_d(\dot{\Theta} - \dot{\Theta}_d) \tag{1}$$

where the first three terms make up the feedforward part of the control law and are computed off-line and the last two terms are the PD controller part calculated online. Solving for the feedforward terms involves computing the inverse dynamics for a preplanned path. The path (position,velocity and acceleration profiles), however, is often described in task (Cartesian) space and hence needs to be converted into joint space using the manipulator's inverse kinematics, Jacobian and its time-derivative as shown in Equations (2-4)

$$\begin{aligned} \Theta_d &= \Gamma(x_d) && (2) \\ \dot{\Theta}_d &= J^{-1}(\Theta)\dot{X}_d && (3) \\ \ddot{\Theta}_d &= J^{-1}(\Theta)\left(\ddot{X}_d - \dot{J}(\Theta, \dot{\Theta})\dot{\Theta}\right) && (4) \end{aligned}$$

where Γ represents the inverse kinematics. For a multilink manipulator these derivations as well as deriving the inverse dynamics model are time consuming and prone to errors, therefore, the use of symbolic manipulation software will greatly aid in their development.

In Section 2.2 the use of Maple V is demonstrated in developing the inverse kinematics, the Jacobian and its time-derivative for a PUMA 560 manipulator. The application of Maple V to the development of the inverse dynamics model is discussed in Section 3. The appendix contains some of the results.

2 Kinematics

The coordinate frames used for developing the kinematic relationships are those used by Unimation, Inc. and are shown in Figure 1. The modified Denavit-Hartenberg parameters for this coordinate frame convention can be found in Armstrong (1986). After writing the corresponding Denavit-Hartenberg matrix for all six links, the displacement of a point in the sixth coordinate frame can be written in the base coordinate frame using the closure equation of the manipulator

$${}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T \tag{5}$$

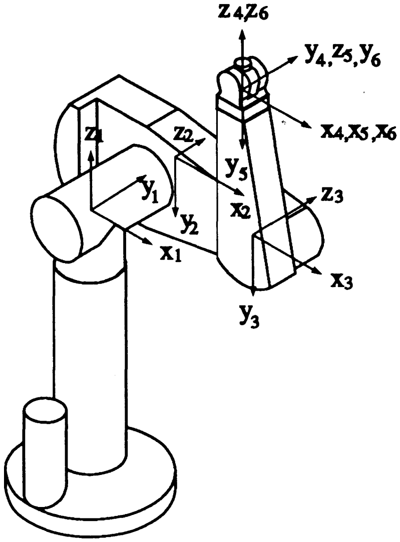


Figure 1: PUMA 560 Manipulator in the Zero Position

Algebraic manipulation of this equation will be the basis of the inverse kinematics solution.

2.1 Inverse Kinematics

The procedure used to solve for the inverse kinematics is the algebraic technique described in Paul (1981). If ${}^{0}_{6}T$ is specified, then succesive premultiplication of Equation (5) by the inverse of the coordinate transformations between the links $({}^{i}_{i+1}T^{-1})$ will yield expressions isolating each joint variable.

Using Maple V, this procedure is greatly simplified and leaves little work to be done by hand. To show the ease at which this technique can be implemented the case for calclulating the inverse kinematics of joint angle 5 is shown.

Equation (5) can be rewritten as follows

$$[T01T12T23T34]^{-1}\,T06 = T45T56$$

In Maple V the two sides of the equation can be written as

$$\begin{aligned} \text{SIDE1} &:= \text{multiply(inverse(multiply(T01,T12,T23,T34),T06));} \\ \text{SIDE2} &:= \text{multiply(T45,T56);} \end{aligned}$$

Examination of the resulting 12 nonlinear equations shows that the (1,3) and (3,3) elements of SIDE2 isolate $\sin(\theta_5)$ and $\cos(\theta_5)$ respectively in terms of known link parameters and already determined joint angles. Joint angle 5 can then be solved as

```
    t5 := arctan(SIDE1[1,3],SIDE1[3,3]);
```

2.2 Jacobian and its Time-Derivative

The linear velocity partial derivative terms of the Jacobian can be found by differentiating the translational elements in Equation (5) with respect to time. This can be performed using Maple V by applying the "diff" function to each of the three translational elements. For example,

```
    v[i] := diff(T06[1,4],t);
```

will perform chain rule differentiation on the x component of the translational part of ${}^{0}_{6}T$ with respect to t. The elements of the Jacobian coming from the linear velocity partial derivative terms are then extracted using the "coeff" function. The linear acceleration partial derivative terms, which are needed for determining the time-derivative of the Jacobian, are found similarly by differentiating each of the three linear velocity components with respect to time.

The angular velocity terms for the Jacobian come from the angular velocity equation for the sixth link, which is found recursively using the following equation.

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{\ \ i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1} \qquad (6)$$

Such recursive equations, which are often used in multibody system analysis, can easily be implemented using Maple V. For the six links of a PUMA 560, Equation (6) was implemented as follows

```
omega0 := vector([0, 0, 0]);
omega1 := add(multiply(inverse(minor(T01,4,4)),omega0),tdot1);


omega6 := add(multiply(inverse(minor(T56,4,4)),omega5),tdot6);
```

where "tdot" is the relative angular velocity between coordinate frames. The elements of the Jacobian are again extracted using the "coeff" function.

The angular acceleration terms used for the time-derivative of the Jacobian come from the angular acceleration equation for the sixth link, which is found recursively using

$$ {}^{i+1}\alpha_{i+1} = {}^{i+1}_{\ \ i}R\,{}^{i}\alpha_i + {}^{i+1}_{\ \ i}R\,{}^{i}\omega \times \dot{\theta}_{i+1} + \ddot{\theta}_{i+1} $$

This was written in Maple V as

```
alpha0 := vector([0, 0, 0]);
alpha1 := add(add(multiply(inverse(minor(T01,4,4)),alpha0),\
          crossprod(multiply(inverse(minor(T01,4,4)),omega0),tdot1)),tddot1);
```

```
alpha6 := add(add(multiply(inverse(minor(T56,4,4)),alpha5),\
          crossprod(multiply(inverse(minor(T56,4,4)),omega5),tdot6)),tddot6);
```

where "tddot" is the relative angular acceleration between coordinate frames.

The expressions for the time-derivative for the Jacobian are given in the Appendix.

3 Inverse Dynamics Model

The feedforward joint torques can be computed from the desired joint angle positions, velocities and accelerations (2-4) using the manipulator's inverse dynamics equations. The recursive multibody dynamics algorithm chosen for solving the inverse dynamics is the Newton-Euler based method which can be found in Craig (1989). This algorithm can be easily implemented in Maple V using similar linear algebra statements demonstrated in Section 2.2. The PUMA 560 parameters used in the recursive equations are from Armstrong (1986).

The resulting expressions for the joint torques, even after applying the trigonometric simplifications built into Maple V, were pages of expressions containing manipulator parameters and trigonometric functions. Verifying these results with a hand derivation would be impractical, therefore, a second recursive multibody dynamics algorithm was programmed to check these expressions.

The second recursive multibody dynamics algorithm, which is also based on a Newton-Euler formulation, is that of Roberson and Schwertassek (1988). This method solves for the forward dynamics of the manipulator recursively without inversion of the mass matrix. The resulting expressions for the joint accelerations can be compared to the first method's results by solving the first method for the manipulator's joint accelerations.

Since the resulting expressions of both methods were too complex to check term for term, the model of the PUMA 560 was built up one link at a time so that application errors of the algorithms could be found early from less complicated expressions. After correcting these errors, the complete PUMA 560 models were compared against each other with numerical examples and agreement was achieved. The required vectors and matrices used for the recursive inverse dynamics algorithm are given in the Appendix.

4 Conclusions

In this paper we have shown how symbolic manipulation can be used in writing kinematic and dynamic expressions for a multilink manipulator. The use of symbolic computation reduces immensely the amount of hand calculation and simplification needed when deriving these expressions. The tree structure associated with most manipulators allows for easy application of programs such as Maple V to the recursive equations used to write the kinematics and dynamics of the systems. Symbolic computation has also been demonstrated to be a useful tool on understanding how a particular algorithm should be implemented. Although the results from symbolic computation for the inverse dynamics could not provide a compact closed form solution, symbolic computation software did provide a convenient means to implement and check the recursive multibody dynamics algorithm for future programming in a lower level computer language.

Acknowledgements

This study was supported by German BMFT Grant 413-5839-01 IN 104 D/7.

References

Armstrong, B., Khatib, O. and Burdick, J. 1986. The explicit dynamic model and inertial parameters of the PUMA 560 arm. *Proc. of IEEE Int. Conf. on Robotics and Automation*,.vol. 1, pp. 510–518.

Craig, J. 1989. *Introduction to Robotics.* New York: Addison-Wesley.

Herrera-Bendezu, L.G., Mu, E. and Cain, J.T. 1988. Symbolic computation of robotic manipulator kinematics. *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 993–998.

Leahy Jr., M.B. 1988. Dynamics based control of vertically articulated manipulators. *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1048–1053.

Lieh, J. 1991. An alternative method to formulate closed-form dynamics for elastic manipulators using symbolic process. *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2369–74.

Manocha, D. and Canny, J.F. 1989. Real time inverse kinematics for general manipulators. *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 383–389.

Mayorga, R.V. Ressa, B. and Wong, A.K.C. 1992. A kinematic design optimization of robot manipulators. *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 396–401.

Paul, R.P. 1981. *Robot Manipulators: Mathematics, Programming and Control.* Cambridge, MA: The MIT Press.

Rieseler, H. and Wahl, F.M. 1990. Fast symbolic computation of the inverse kinematics of robots. *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 462–467.

Roberson, R.E. and Schwertassek, R. 1988. *Dynamics of Multibody Systems.* Berlin: Springer-Verlag.

Rothfuß, R. Schaffner, J. and Zeitz, M. 1993. Computer-aided stability analysis of nonlinear systems by means of a computer algebra system. *Proc. of the 2nd European Control Conf.*, vol. 4, pp. 2096–2102.

Tarn, T.J., Bejczy, A.K., Marth, G.T. and Ramadorai, A.K. 1993. Performance comparison of four manipulator servo schemes. *IEEE Control Systems*, vol. 13(1), pp. 22–29.

Van Essen, H. and ·De Jager, B. 1993. Analysis and design of nonlinear control systems with the symbolic computation system MAPLE. *Proc. of the 2nd European Control Conf.*, vol. 4, pp. 2081–2085.

Appendix

The time-derivative of the Jacobian in the coordinate frame system shown in Figure 1 is given below. The angular acceleration terms of the time-derivative of the Jacobian have been expressed in frame 6. To express these terms in the base frame, as were the linear acceleration terms, the lower 3 × 6 submatrix of the time-derivative of the Jacobian should be premultiplied by ${}^{0}_{6}R$, which is the rotational submatrix of the closure equation.

$$\dot{J} = \begin{bmatrix} \frac{\partial a_x}{\dot{\partial\theta_1}} & \frac{\partial a_x}{\dot{\partial\theta_2}} & \frac{\partial a_x}{\dot{\partial\theta_3}} & \frac{\partial a_x}{\dot{\partial\theta_4}} & \frac{\partial a_x}{\dot{\partial\theta_5}} & \frac{\partial a_x}{\dot{\partial\theta_6}} \\ \frac{\partial a_y}{\dot{\partial\theta_1}} & \frac{\partial a_y}{\dot{\partial\theta_2}} & \frac{\partial a_y}{\dot{\partial\theta_3}} & \frac{\partial a_y}{\dot{\partial\theta_4}} & \frac{\partial a_y}{\dot{\partial\theta_5}} & \frac{\partial a_y}{\dot{\partial\theta_6}} \\ \frac{\partial a_z}{\dot{\partial\theta_1}} & \frac{\partial a_z}{\dot{\partial\theta_2}} & \frac{\partial a_z}{\dot{\partial\theta_3}} & \frac{\partial a_z}{\dot{\partial\theta_4}} & \frac{\partial a_z}{\dot{\partial\theta_5}} & \frac{\partial a_z}{\dot{\partial\theta_6}} \\ \frac{\partial \alpha_x}{\dot{\partial\theta_1}} & \frac{\partial \alpha_x}{\dot{\partial\theta_2}} & \frac{\partial \alpha_x}{\dot{\partial\theta_3}} & \frac{\partial \alpha_x}{\dot{\partial\theta_4}} & \frac{\partial \alpha_x}{\dot{\partial\theta_5}} & \frac{\partial \alpha_x}{\dot{\partial\theta_6}} \\ \frac{\partial \alpha_y}{\dot{\partial\theta_1}} & \frac{\partial \alpha_y}{\dot{\partial\theta_2}} & \frac{\partial \alpha_y}{\dot{\partial\theta_3}} & \frac{\partial \alpha_y}{\dot{\partial\theta_4}} & \frac{\partial \alpha_y}{\dot{\partial\theta_5}} & \frac{\partial \alpha_y}{\dot{\partial\theta_6}} \\ \frac{\partial \alpha_z}{\dot{\partial\theta_1}} & \frac{\partial \alpha_z}{\dot{\partial\theta_2}} & \frac{\partial \alpha_z}{\dot{\partial\theta_3}} & \frac{\partial \alpha_z}{\dot{\partial\theta_4}} & \frac{\partial \alpha_z}{\dot{\partial\theta_5}} & \frac{\partial \alpha_z}{\dot{\partial\theta_6}} \end{bmatrix}$$

$$\frac{\partial a_x}{\partial \theta_1} = ((a_4\,c23 - d_4\,s23 - a_3\,c2)\,c1 + (d_2 - d_3)\,s1)\,\dot{\theta}_1$$

$$\frac{\partial a_x}{\partial \theta_2} = -2\,(a_4\,s23 + d_4\,c23 - a_3\,s2)\,s1\,\dot{\theta}_1 + (a_4\,c23 - d_4\,s23 - a_3\,c2)\,c1\,\dot{\theta}_2 + 2\,(a_4\,c23 - d_4\,s23)\,c1\,\dot{\theta}_3$$

$$\frac{\partial a_x}{\partial \theta_3} = -2\,(a_4\,s23 + d_4\,c23)\,s1\,\dot{\theta}_1 + (a_4\,c23 - d_4\,s23)\,c1\,\dot{\theta}_3$$

$$\frac{\partial a_x}{\partial \theta_4} = 0$$

$$\frac{\partial a_x}{\partial \theta_5} = 0$$

$$\frac{\partial a_x}{\partial \theta_6} = 0$$

$$\frac{\partial a_y}{\partial \theta_1} = ((a_4\,c23 - d_4\,s23 - a_3\,c2)\,s1 + (d_3 - d_2)\,c1)\,\dot{\theta}_1$$

$$\frac{\partial a_y}{\partial \theta_2} = 2\,(d_4\,c23 + a_4\,s23 - a_3\,s2)\,c\acute{1}\,\dot{\theta}_1 + (a_4\,c23 - d_4\,s23 - a_3\,c2)\,s1\,\dot{\theta}_2 + 2\,(a_4\,c23 - d_4\,s23)\,s1\,\dot{\theta}_3$$

$$\frac{\partial a_y}{\partial \theta_3} = 2\,(d_4\,c23 + a_4\,s23)\,c1\,\dot{\theta}_1 + (a_4\,c23 - d_4\,s23)\,s1\,\dot{\theta}_3$$

$$\frac{\partial a_y}{\partial \theta_4} = 0$$

$$\frac{\partial a_y}{\partial \theta_5} = 0$$

$$\frac{\partial a_y}{\partial \theta_6} = 0$$

$$\frac{\partial a_z}{\partial \theta_1} = 0$$

$$\frac{\partial a_z}{\partial \theta_2} = (a_3\,s2 - d_4\,c23 - a_4\,s23)\,\dot{\theta}_2 - 2\,(d_4\,c23 + a_4\,s23)\,\dot{\theta}_3$$

$$\frac{\partial a_z}{\partial \theta_3} = -(d_4\,c23 + a_4\,s23)\,\dot{\theta}_3$$

$$\frac{\partial a_z}{\partial \theta_4} = 0$$

$$\frac{\partial a_z}{\partial \theta_5} = 0$$

$$\frac{\partial a_z}{\partial \theta_6} = 0$$

$$\frac{\partial \alpha_x}{\partial \theta_1} = 0$$

$$\frac{\partial \alpha_x}{\partial \theta_2} = (s23s5c6 + c23(s4s6 - c4c5c6))\dot{\theta}_1 - (s4s6 - c4c5c6)\dot{\theta}_4 - s4s5c6\dot{\theta}_5 + (c4c6 - s4c5s6)\dot{\theta}_6$$

$$\frac{\partial \alpha_x}{\partial \theta_3} = (s23s5c6 + c23(s4s6 - c4c5c6))\dot{\theta}_1 - (s4s6 - c4c5c6)\dot{\theta}_4 - s4s5c6\dot{\theta}_5 + (c4c6 - s4c5s6)\dot{\theta}_6$$

$$\frac{\partial \alpha_x}{\partial \theta_4} = s23(c4s6 + s4c5c6)\dot{\theta}_1 - c5c6\dot{\theta}_5 + s5s6\dot{\theta}_6$$

$$\frac{\partial \alpha_x}{\partial \theta_5} = c6(s23c4s5 - c23c5)\dot{\theta}_1 + c6\dot{\theta}_6$$

$$\frac{\partial \alpha_x}{\partial \theta_6} = (s23s4c6 + s6(s23c4c5 + c23s5))\dot{\theta}_1$$

$$\frac{\partial \alpha_y}{\partial \theta_1} = 0$$

$$\frac{\partial \alpha_y}{\partial \theta_2} = (c23(c4c5s6 + s4c6) - s23s5s6)\dot{\theta}_1 - (s4c6 + c4c5s6)\dot{\theta}_4 + s4s5s6\dot{\theta}_5 - (c4s6 + s4c5c6)\dot{\theta}_6$$

$$\frac{\partial \alpha_y}{\partial \theta_3} = (c23(c4c5s6 + s4c6) - s23s5s6)\dot{\theta}_1 - (s4c6 + c4c5s6)\dot{\theta}_4 + s4s5s6\dot{\theta}_5 - (c4s6 + s4c5c6)\dot{\theta}_6$$

$$\frac{\partial \alpha_y}{\partial \theta_4} = s23(c4c6 - s4c5s6)\dot{\theta}_1 + c5s6\dot{\theta}_5 + s5c6\dot{\theta}_6$$

$$\frac{\partial \alpha_y}{\partial \theta_5} = (c23c5s6 - s23c4s5s6)\dot{\theta}_1 - s6\dot{\theta}_6$$

$$\frac{\partial \alpha_y}{\partial \theta_6} = (s23(c4c5c6 - s4s6) + c23s5c6)\dot{\theta}_1$$

$$\frac{\partial \alpha_z}{\partial \theta_1} = 0$$

$$\frac{\partial \alpha_z}{\partial \theta_2} = -(s23c5 + c4s5c23)\dot{\theta}_1 + s4c5\dot{\theta}_5 + c4s5\dot{\theta}_6$$

$$\frac{\partial \alpha_z}{\partial \theta_3} = -(s23c5 + c4s5c23)\dot{\theta}_1 + s4c5\dot{\theta}_5 + c4s5\dot{\theta}_6$$

$$\frac{\partial \alpha_z}{\partial \theta_4} = s23s4s5\dot{\theta}_1 - s5\dot{\theta}_5$$

$$\frac{\partial \alpha_z}{\partial \theta_5} = -(c23s5 + s23c4c5)\dot{\theta}_1$$

$$\frac{\partial \alpha_z}{\partial \theta_6} = 0$$

Link No. (i)	${}^{i+1}_{i}R$	$\dot{\theta}_{i+1}$	${}^{i}P_{i+1}$(m)	$\ddot{\theta}_{i+1}$	${}^{i+1}P_{C_{i+1}}$(m)	m_{i+1} (kg)	${}^{i+1}I_{C_{i+1}}$(kg·m^2)
0	$\begin{bmatrix} c1 & s1 & 0 \\ -s1 & c1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$[\,0 \;\; 0 \;\; \dot{\theta}_1\,]$	$[\,0 \;\; 0 \;\; 0\,]$	$[\,0 \;\; 0 \;\; \ddot{\theta}_1\,]$	$[\,0 \;\; 0 \;\; 0\,]$	——	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1.49 \end{bmatrix}$
1	$\begin{bmatrix} c2 & 0 & -s2 \\ -s2 & 0 & -c2 \\ 0 & 1 & 0 \end{bmatrix}$	$[\,0 \;\; 0 \;\; \dot{\theta}_2\,]$	$[\,0 \;\; .2435 \;\; 0\,]$	$[\,0 \;\; 0 \;\; \ddot{\theta}_2\,]$	$[\,.068 \;\; .006 \;\; -.016\,]$	17.40	$\begin{bmatrix} .130 & 0 & 0 \\ 0 & .524 & 0 \\ 0 & 0 & 5.249 \end{bmatrix}$
2	$\begin{bmatrix} c3 & s3 & 0 \\ -s3 & c3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$[\,0 \;\; 0 \;\; \dot{\theta}_3\,]$	$[\,.4318 \;\; 0 \;\; -.0934\,]$	$[\,0 \;\; 0 \;\; \ddot{\theta}_3\,]$	$[\,0 \;\; -.070 \;\; .014\,]$	4.80	$\begin{bmatrix} .066 & 0 & 0 \\ 0 & .0125 & 0 \\ 0 & 0 & .916 \end{bmatrix}$
3	$\begin{bmatrix} c4 & 0 & s4 \\ -s4 & 0 & c4 \\ 0 & -1 & 0 \end{bmatrix}$	$[\,0 \;\; 0 \;\; \dot{\theta}_4\,]$	$[\,-.0203 \;\; -.4331 \;\; 0\,]$	$[\,0 \;\; 0 \;\; \ddot{\theta}_4\,]$	$[\,0 \;\; 0 \;\; -.019\,]$	.82	$\begin{bmatrix} .0018 & 0 & 0 \\ 0 & .0018 & 0 \\ 0 & 0 & .2013 \end{bmatrix}$
4	$\begin{bmatrix} c5 & 0 & -s5 \\ -s5 & 0 & -c5 \\ 0 & 1 & 0 \end{bmatrix}$	$[\,0 \;\; 0 \;\; \dot{\theta}_5\,]$	$[\,0 \;\; 0 \;\; 0\,]$	$[\,0 \;\; 0 \;\; \ddot{\theta}_5\,]$	$[\,0 \;\; 0 \;\; 0\,]$	.34	$\begin{bmatrix} .0003 & 0 & 0 \\ 0 & .0003 & 0 \\ 0 & 0 & .1794 \end{bmatrix}$
5	$\begin{bmatrix} c6 & 0 & s6 \\ -s6 & 0 & c6 \\ 0 & -1 & 0 \end{bmatrix}$	$[\,0 \;\; 0 \;\; \dot{\theta}_6\,]$	$[\,0 \;\; 0 \;\; 0\,]$	$[\,0 \;\; 0 \;\; \ddot{\theta}_6\,]$	$[\,0 \;\; 0 \;\; .032\,]$	.09	$\begin{bmatrix} .00015 & 0 & 0 \\ 0 & .00015 & 0 \\ 0 & 0 & .19304 \end{bmatrix}$

Vectors and Matrices Needed for Recursive Inverse Dynamics Model

Nonlinear Control of Constrained Redundant Manipulators

Christoph Woernle
Institute A of Mechanics, University of Stuttgart,
Germany

Abstract. The dynamic hybrid position and contact force control of open-chain manipulators with redundant degrees of freedom by means of exact input-output linearization is investigated. The dynamics of the manipulator is transferred into a decoupled linear dynamics in the subspace of the control variables and a still nonlinear dynamics in the subspace of the redundant degrees of freedom. By projecting artificial potential and damping forces into the subspace of the redundant degrees of freedom, it is possible e. g. to realize repeatable motions in the joint space for cyclic trajectories or to avoid collisions with obstacles in the workspace.

1 Introduction

Kinematically redundant manipulators have more independently controllable degrees of freedom than control variables specified by the particular task. This provides increased flexibility for the execution of complex tasks. Typical variables for dynamic control are the position coordinates of the endeffector or independent position coordinates together with contact forces in case of material contacts between manipulator and environment. The considered control task is to make all control variables track prescribed time functions and to accomplish desirable subtasks using the redundant degrees of freedom such as avoiding collisions with obstacles in the workspace. Because of the nonlinearities of the mechanical system, the method of exact input-output linearization is applied. By means of a nonlinear state feedback the dynamics of the manipulator is transferred into a linear, decoupled dynamics in the subspace of the control variables and a still nonlinear dynamics in the subspace of the redundant degrees of freedom that is non-observable from the control variables.

The dynamics of the manipulator in the subspace of its redundant degrees of freedom is then prescribed by nonlinear artificial potential and damping forces that generate asymptotically stable equilibrium positions in this subspace. Potential forces, that depend only on the joint coordinates, are applied to realize repeatability in

J. Angeles et al. (eds.), Computational Kinematics, 119–128.

the joint motions for cyclic trajectories. Collision avoidance is achieved by potential forces that are defined over the cartesian workspace and push the manipulator away from the obstacles. The described procedure can be applied to unconstrained manipulators, to manipulators with material contacts at the endeffector, as described e. g. by KHATIB (1987) or YOSHIKAWA (1990), and also to manipulators with multiple contacts at the endeffector and/or other limbs.

2 Dynamics of a Constrained Manipulator

2.1 Equations of Motion

Assuming a rigid-body model of an open-chain manipulator with f degrees of freedom and f control forces $\boldsymbol{u} \in \mathbb{R}^f$ acting directly along the f independent joint coordinates $\boldsymbol{q} \in \mathbb{R}^f$, the equations of motion are

$$M(q)\,\ddot{q} = g(q,\dot{q}) + u \,. \tag{1}$$

Here, $\boldsymbol{M} \in \mathbb{R}^{f,f}$ is the symmetric, positive-definite inertia-matrix, while the vector $\boldsymbol{g} \in \mathbb{R}^f$ summarizes the generalized centrifugal forces as well as all applied forces except for the control forces $\boldsymbol{u}$.

During execution of a task, the manipulator may be constrained by material contacts of the endeffector or other limbs with the environment. It is assumed that these material contacts lead to $d < f$ independent constraints on the joint coordinates $\boldsymbol{q}$:

$$\varphi_{Fi}(q) = 0 \,, \quad i = 1,\ldots,d, \qquad \text{or} \qquad \varphi_F(q) = 0 \,. \tag{2}$$

If the functions $\varphi_F(\boldsymbol{q})$ are differentiable at least twice with respect to time, the total second-order time derivative of Eq. (2) is

$$\begin{aligned} &\ddot{\varphi}_F \equiv \Phi_F(q)\,\ddot{q} \;+\; \bar{\ddot{\varphi}}_F(q,\dot{q}) = 0 \\ &\text{with} \quad \Phi_F(q) = \frac{\partial\varphi_F(q)}{\partial q} \in \mathbb{R}^{d,f} \,, \qquad \bar{\ddot{\varphi}}_F(q,\dot{q}) = \frac{d\,\Phi_F(q)}{dt}\,\dot{q} \in \mathbb{R}^d \,. \end{aligned} \tag{3}$$

The $d < f$ constraints (2) reduce the number of degrees of freedom of the manipulator from f to $f - d$. In the equations of motion (1) constraint forces $\Phi_F^{\mathrm{T}}\,\lambda$ are to be added, whereby $\lambda \in \mathbb{R}^d$ are independent coordinates of the constraint forces:

$$M(q)\,\ddot{q} = g(q,\dot{q}) + \Phi_F^{\mathrm{T}}(q)\,\lambda + u \,. \tag{4}$$

The differential-algebraic set of equations (2) and (4) then describes the dynamics of the constrained manipulator. Hereby, the constraints (2) must be fulfilled on position, velocity and acceleration level. If the positions $\boldsymbol{q}$ and the velocities $\dot{\boldsymbol{q}}$, which have to be consistent with the constraints (2) and their first-order time derivatives, as well as the control inputs $\boldsymbol{u}$ are given, Eqs. (3) and (4) together represent a set of $f + d$ linear equations to determine uniquely the f unknown accelerations $\ddot{\boldsymbol{q}}$ and the d unknown contact force coordinates $\boldsymbol{\lambda}$:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{\Phi}_F^{\mathrm{T}} \\ \boldsymbol{\Phi}_F & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} + \boldsymbol{u} \\ -\bar{\ddot{\boldsymbol{\varphi}}}_F \end{bmatrix} . \tag{5}$$

2.2 Control Variables

The objective of the considered control is to make altogether m control variables $\boldsymbol{y}(t)$ track prescribed functions $\hat{\boldsymbol{y}}(t)$. For the constrained manipulator described by (2) and (4), the m control variables may include m_F force outputs $\boldsymbol{y}_F$ as well as m_P position outputs $\boldsymbol{y}_P$. In this work, all d coordinates $\boldsymbol{\lambda}$ of the constraint forces are defined as the $m_F = d$ force outputs:

$$y_{Fi} = \lambda_i \,, \quad i = 1, \ldots, m_F \,, \qquad \text{or} \qquad \boldsymbol{y}_F = \boldsymbol{\lambda} \,. \tag{6}$$

If only a subset of $\boldsymbol{\lambda}$ is chosen as control variables, the manipulator is *statically redundant.* This case is not considered here, although the following derivations can be easily extended to include it.

With $f - m_F$ degrees of freedom of the constrained manipulator, $m_P \leq f - m_F$ position outputs $\boldsymbol{y}_P$ can be defined as functions of the position variables $\boldsymbol{q}$:

$$y_{Pi} = \varphi_{Pi}(\boldsymbol{q}) \,, \quad i = 1, \ldots, m_P \,, \qquad \text{or} \qquad \boldsymbol{y}_P = \boldsymbol{\varphi}_P(\boldsymbol{q}) \,. \tag{7}$$

In the same manner as the constraint functions $\boldsymbol{\varphi}_F(\boldsymbol{q})$ of Eq. (2), they are assumed to be at least twice differentiable with respect to time:

$$\begin{aligned} &\ddot{\boldsymbol{y}}_P = \boldsymbol{\Phi}_P(\boldsymbol{q}) \, \ddot{\boldsymbol{\varphi}} \, + \, \bar{\ddot{\boldsymbol{\varphi}}}_P(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ &\text{with} \qquad \boldsymbol{\Phi}_P(\boldsymbol{q}) = \frac{\partial \boldsymbol{\varphi}_P(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{m_P, f} \,, \qquad \bar{\ddot{\boldsymbol{\varphi}}}_P(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{d\boldsymbol{\Phi}_P(\boldsymbol{q})}{dt} \, \dot{\boldsymbol{q}} \in \mathbb{R}^{m_P} \,. \end{aligned} \tag{8}$$

The kinematical control variables $\boldsymbol{y}_P$ must be chosen within the admissible motion space of the constrained manipulator. This condition is fulfilled, if the m_F functions $\boldsymbol{\varphi}_F(\boldsymbol{q})$ of Eq. (2) and the m_P functions $\boldsymbol{\varphi}_P(\boldsymbol{q})$ of Eq. (7) are independent, or equivalently, the $m_F + m_P$ row vectors of the functional matrices $\boldsymbol{\Phi}_F$ and $\boldsymbol{\Phi}_P$ are linearly independent.

For $m = m_P + m_F < f$, the manipulator is *kinematically redundant* with $f - m$ redundant degrees of freedom that are not specified by control variables of type (7). Basically, it can be always made non-redundant by introducing $f - m$ additional independent control variables of type (7). The practical problem is, however, that desired values for these additional outputs are not prescribed by the manipulator task. The redundant degrees of freedom will be therefore controlled in a different manner.

3 Input-Output Linearization

3.1 Linear Input-Output Equation

For the derivation of the input-output linearization, the so-called partial relative degree δ_i of the i-th output y_i is important. It is exactly the number of times one has to differentiate y_i totally with respect to time in order to have at least one component of the input vector $\boldsymbol{u}$ explicitly appearing, refer to ISIDORI (1989). By this, a set of equations is obtained that is linear in the δ_ith-order derivatives of y_i and the inputs $\boldsymbol{u}$. The solution of this "linear input-output equation" with respect to $\boldsymbol{u}$ yields the "inverse system" necessary for input-output-linearization and -decoupling.

The linear input-output equation for the manipulator is obtained by inserting the accelerations $\ddot{\boldsymbol{q}}$ and the contact forces $\boldsymbol{\lambda}$ resulting from Eq. (5) into the control output equations (6) and (8). It can be brought into the form

$$\begin{bmatrix} \boldsymbol{y}_F \\ \ddot{\boldsymbol{y}}_P \end{bmatrix} = \underbrace{\boldsymbol{F}\left(\begin{bmatrix} \boldsymbol{\Phi}_F \\ \boldsymbol{\Phi}_P \end{bmatrix} \boldsymbol{M}^{-1}\boldsymbol{g} + \begin{bmatrix} \bar{\ddot{\boldsymbol{\varphi}}}_F \\ \bar{\ddot{\boldsymbol{\varphi}}}_P \end{bmatrix}\right)}_{\boldsymbol{d}(\boldsymbol{q},\dot{\boldsymbol{q}})} + \underbrace{\boldsymbol{F}\begin{bmatrix} \boldsymbol{\Phi}_F \\ \boldsymbol{\Phi}_P \end{bmatrix} \boldsymbol{M}^{-1}}_{\boldsymbol{D}(\boldsymbol{q})} \boldsymbol{u} \tag{9}$$

with

$$\boldsymbol{F} = \begin{bmatrix} -\left(\boldsymbol{\Phi}_F \boldsymbol{M}^{-1} \boldsymbol{\Phi}_F^{\mathrm{T}}\right)^{-1} & \boldsymbol{0} \\ -\left(\boldsymbol{\Phi}_P \boldsymbol{M}^{-1} \boldsymbol{\Phi}_F^{\mathrm{T}}\right)\left(\boldsymbol{\Phi}_F \boldsymbol{M}^{-1} \boldsymbol{\Phi}_F^{\mathrm{T}}\right)^{-1} & \boldsymbol{I} \end{bmatrix}.$$

The matrix $\boldsymbol{D} \in \mathbb{R}^{m,f}$ here has full row rank $\mathrm{r}(\boldsymbol{D}) = m \leq f$ and is the *decoupling matrix* of the system, ISIDORI (1989). This means that the m outputs $\boldsymbol{y}$ can be independently controlled by the f inputs $\boldsymbol{u}$. Thus, the relative degrees are $\delta_{Pi} = 2$ for all position outputs $\boldsymbol{y}_P$ and $\delta_{Fi} = 0$ for all force outputs $\boldsymbol{y}_F$.

3.2 General Decoupling Feedback

To obtain the linearizing state-feedback, the input-output equation (9) is solved with respect to $\boldsymbol{u}$. The control outputs $\boldsymbol{y}_F \in \mathbb{R}^{m_F}$ and $\ddot{\boldsymbol{y}}_P \in \mathbb{R}^{m_P}$ are hereby replaced by new inputs $\boldsymbol{w}_F \in \mathbb{R}^{m_F}$ and $\boldsymbol{w}_P \in \mathbb{R}^{m_P}$ of the decoupling feedback. Since in the considered redundant case the decoupling matrix $\boldsymbol{D}$ is nonsquare, the solution of Eq. (9) is not unique. It can be generally expressed by

$$\boldsymbol{u} = \boldsymbol{D}^{-}\begin{bmatrix} \boldsymbol{w}_F \\ \boldsymbol{w}_P \end{bmatrix} - \boldsymbol{D}^{-}\boldsymbol{d} + (\boldsymbol{I} - \boldsymbol{D}^{-}\boldsymbol{D})\boldsymbol{w}^{*}. \tag{10}$$

The matrix $\boldsymbol{D}^{-} \in \mathbb{R}^{f,m}$ is an arbitrary generalized right-inverse of $\boldsymbol{D}$ with the property $\boldsymbol{D}\boldsymbol{D}^{-} = \boldsymbol{I}$. The arbitrary vector $\boldsymbol{w}^{*} \in \mathbb{R}^{f}$ is projected into the nullspace of $\boldsymbol{D}$.

A particular generalized inverse $\boldsymbol{D}^-$ will be determined in section 4.2. The cancellation of the nonlinearities of the mechanical system by the feedback (10) is shown by inserting Eq. (10) into the input-output equation (9). If the matrices $\boldsymbol{D}$ and $\boldsymbol{d}$ in Eqs. (9) and (10) coincide, a linear and decoupled system is obtained consisting of m_F proportional and m_P double-integrating input-output channels:

$$\boldsymbol{y}_F = \boldsymbol{w}_F \qquad \text{or} \qquad y_{Fi} = w_{Fi}\,, \quad i = 1,\ldots,m_F\,, \tag{11}$$

$$\ddot{\boldsymbol{y}}_P = \boldsymbol{w}_P \qquad \text{or} \qquad \ddot{y}_{Pi} = w_{Pi}\,, \quad i = 1,\ldots,m_P\,. \tag{12}$$

These input-ouput channels and the redundant degrees of freedom can now be separately controlled.

4 Control of the Decoupled System

4.1 Control of the Output Variables

To obtain asymptotically stable tracking control for the system outputs, i. e.

$$\lim_{t\to\infty}(\hat{\boldsymbol{y}}_F(t) - \boldsymbol{y}_F(t)) = \boldsymbol{0}\,, \qquad \lim_{t\to\infty}(\hat{\boldsymbol{y}}_P(t) - \boldsymbol{y}_P(t)) = \boldsymbol{0}\,,$$

(desired values marked by " ^ ") each input-output channel (11) for the contact forces is separately controlled by an integral feedback

$$w_{Fi}(t) = \alpha_{Fi}\int e_{Fi}\,dt\,, \quad i = 1,\ldots,m_F\,, \tag{13}$$

of the force error $e_{Fi} = \hat{y}_{Fi} - y_{Fi}$, while each input-output channel (12) for the positions is separately controlled by a feedback

$$w_{Pi}(t) = \ddot{\hat{y}}_{Pi} + \alpha_{Di}\,\dot{e}_{Pi} + \alpha_{Pi}\,e_{Pi} + \alpha_{Ii}\int e_{Pi}\,dt\,, \quad i = 1,\ldots,m_P\,, \tag{14}$$

of the position error $e_{Pi} = \hat{y}_{Pi} - y_{Pi}$. The feedback gains α_{Ii}, α_{Pi}, α_{Di}, and α_{Fi} are determined by pole-placement, see e. g. ISIDORI(1989) or YOSHIKAWA (1990).

4.2 Control of the Redundant Degrees of Freedom

Independently from the control of the decoupled input-output channels, the dynamics of the manipulator in the subspace of its $f - m$ redundant degrees of freedom is inherently determined by a particular inverse system (10). This dynamics is now prescribed in such a way that desired configurations of the manipulator are asymptotically stable equilibrium positions. The manipulator then tends to move towards these configurations. In terms of the joint coordinates $\boldsymbol{q}$, this artificial dynamics is stated as simple as possible in the form

$$\boldsymbol{Q}\,\ddot{\boldsymbol{q}}_u = \boldsymbol{p}, \tag{15}$$

where $\boldsymbol{Q}$ is regarded as a diagonal, positive-definite inertia-matrix and $\boldsymbol{p}$ as a vector of generalized applied forces in the joint space. The asymptotic motion of the kinematical chain towards desired configurations must be realized by the applied forces $\boldsymbol{p}$, while the time behaviour of this motion is additionally influenced by the masses $\boldsymbol{Q}$. Eq. (15) only holds, if the joint accelerations $\ddot{\boldsymbol{q}}_u$ are unconstrained, as indicated by the index "u". However, the actual accelerations $\ddot{\boldsymbol{q}}$ must be consistent with the m_F constraints (3) and the m_P output equations (8). With the replacement $\ddot{\boldsymbol{y}}_P \to \boldsymbol{w}_P$, the latter are additional constraints on $\ddot{\boldsymbol{q}}$. Eqs. (3) and (8) together are a set of $m = m_F + m_P$ constraint equations for $\ddot{\boldsymbol{q}}$:

$$\underbrace{\begin{bmatrix} \boldsymbol{\Phi}_F \\ \boldsymbol{\Phi}_P \end{bmatrix}}_{\boldsymbol{\Phi} \in \mathbb{R}^{m,f}} \ddot{\boldsymbol{q}} = -\underbrace{\begin{bmatrix} \bar{\ddot{\boldsymbol{\varphi}}}_F \\ \bar{\ddot{\boldsymbol{\varphi}}}_P \end{bmatrix}}_{\bar{\ddot{\boldsymbol{\varphi}}}} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{V}} \boldsymbol{w}_P . \tag{16}$$

Because of these constraints, the "unconstrained" accelerations $\ddot{\boldsymbol{q}}_u = \boldsymbol{Q}^{-1}\boldsymbol{p}$ from Eq. (15) is different from the actual accelerations $\ddot{\boldsymbol{q}}$. The determination of the actual $\ddot{\boldsymbol{q}}$ can be formulated as an optimization problem by means of the *principle of least constraint* of GAUSS, refer e. g. to PARS (1968). It defines the "constraint" Z as the sum of the mass-weighted squares of the differences between the accelerations $\ddot{\boldsymbol{q}}_u$ of an unconstrained and $\ddot{\boldsymbol{q}}$ of a constrained mechanical system, and requires that Z gets minimal with respect to $\ddot{\boldsymbol{q}}$, i. e.

$$Z(\ddot{\boldsymbol{q}}) = \frac{1}{2}(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_u)^{\mathrm{T}}\, \boldsymbol{Q}\, (\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_u) \overset{!}{=} \min_{\ddot{q}}$$

or, with $\ddot{\boldsymbol{q}}_u = \boldsymbol{Q}^{-1}\boldsymbol{p}$,

$$Z(\ddot{\boldsymbol{q}}) = \frac{1}{2}\ddot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{Q}\,\ddot{\boldsymbol{q}} - \boldsymbol{p}^{\mathrm{T}}\ddot{\boldsymbol{q}}\ \left(+\boldsymbol{p}^{\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{p}\right) \overset{!}{=} \min_{\ddot{q}} . \tag{17}$$

This optimization problem is constrained by Eq. (16). To solve it, the extended objective function

$$\bar{Z}(\ddot{\boldsymbol{q}}) = \frac{1}{2}\ddot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{Q}\,\ddot{\boldsymbol{q}} - \boldsymbol{p}^{\mathrm{T}}\ddot{\boldsymbol{q}} + \boldsymbol{\mu}^{\mathrm{T}}\left(\boldsymbol{\Phi}\,\ddot{\boldsymbol{q}} + \bar{\ddot{\boldsymbol{\varphi}}} - \boldsymbol{V}\,\boldsymbol{w}_P\right) \overset{!}{=} \min_{\ddot{q}} \tag{18}$$

with LAGRANGE-multipliers $\boldsymbol{\mu} \in \mathbb{R}^m$ is minimized. The necessary minimum condition gives in combination with the constraint equations (16) the linear set of equations

$$\begin{matrix} \dfrac{\partial \bar{Z}}{\partial \ddot{\boldsymbol{q}}^{\mathrm{T}}} = 0 & \to \\ (16) & \to \end{matrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{\Phi}^{\mathrm{T}} \\ \boldsymbol{\Phi} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p} \\ -\bar{\ddot{\boldsymbol{\varphi}}} + \boldsymbol{V}\,\boldsymbol{w}_P \end{bmatrix} . \tag{19}$$

These are differential-algebraic equations of motion for the dynamics of the manipulator within its redundant degrees of freedom. Elimination of $\boldsymbol{\mu}$ yields the solution

$$\ddot{\boldsymbol{q}} = \boldsymbol{\Phi}^+_{(Q)}(\boldsymbol{V}\,\boldsymbol{w}_P - \bar{\ddot{\boldsymbol{\varphi}}}) \;+\; (\boldsymbol{I} - \boldsymbol{\Phi}^+_{(Q)}\boldsymbol{\Phi})\boldsymbol{Q}^{-1}\boldsymbol{p} \tag{20}$$

with

$$\boldsymbol{\Phi}^+_{(Q)} = \boldsymbol{Q}^{-1}\,\boldsymbol{\Phi}^T\,(\boldsymbol{\Phi}\,\boldsymbol{Q}^{-1}\,\boldsymbol{\Phi}^T)^{-1}\,.$$

Here, $\boldsymbol{\Phi}^+_{(Q)} \in \mathbb{R}^{f,m}$ is the pseudoinverse of $\boldsymbol{\Phi}$ weighted by $\boldsymbol{Q}$. It has the property $\boldsymbol{\Phi}\,\boldsymbol{\Phi}^+_{(Q)} = \boldsymbol{I}$. Inserting (20) into (4) yields together with (6) the particular inverse system, refer to Eq. (10):

$$\boldsymbol{u} = \underbrace{\left[-\boldsymbol{\Phi}_F^{\mathrm{T}} \mid \boldsymbol{M}\boldsymbol{\Phi}^+_{(Q)}\boldsymbol{V}\right]}_{\boldsymbol{D}^-}\begin{bmatrix}\boldsymbol{w}_F\\ \boldsymbol{w}_P\end{bmatrix} - \boldsymbol{M}\,\boldsymbol{\Phi}^+_{(Q)}\,\bar{\ddot{\boldsymbol{\varphi}}} - \boldsymbol{g} + \boldsymbol{M}\,(\boldsymbol{I} - \boldsymbol{\Phi}^+_{(Q)}\boldsymbol{\Phi})\,\boldsymbol{Q}^{-1}\boldsymbol{p}\,. \tag{21}$$

If the corresponding matrices of the input-output equation (9) of the mechanical system and the inverse system (21) coincide, the decoupled input-output channels (11) and (12) are obtained, as shown by inserting (21) into (9). The feedback (21) transfers the dynamics of the mechanical system into the linear input-output dynamics (11), (12) and the nonlinear "internal" dynamics (19) that is non-observable from the outputs $\boldsymbol{y}_P$ and $\boldsymbol{y}_F$. A block diagram of the overall control structure is shown in Fig. 1.

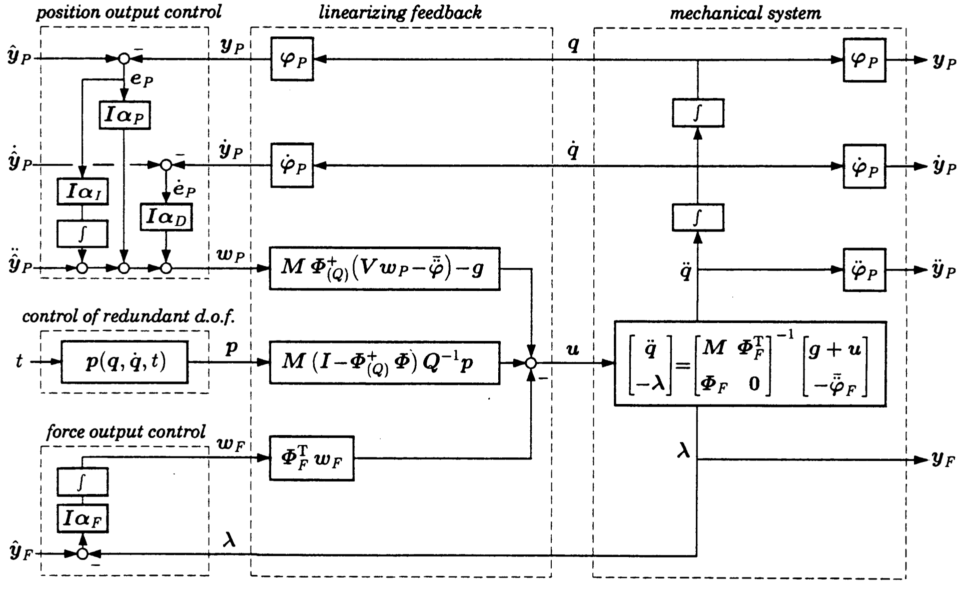


Fig. 1: Nonlinear Control Structure

5 Examples

The desired configurations are now formulated as equilibrium positions of the constrained dynamics (19) of the manipulator in the subspace of its redundant degrees of freedom. This can be achieved by an artificial potential energy field $U(\boldsymbol{q})$ that is minimal in desired configurations of the manipulator. The control forces $\boldsymbol{p}$ then are conservative forces $\boldsymbol{p}_c(\boldsymbol{q})$ together with dissipative forces $\boldsymbol{p}_d(\boldsymbol{q},\dot{\boldsymbol{q}})$ for asymptotic stabilization:

$$\boldsymbol{p}(\boldsymbol{q},\dot{\boldsymbol{q}}) = -\left(\frac{\partial U(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^{\mathrm{T}} - \boldsymbol{p}_d(\boldsymbol{q},\dot{\boldsymbol{q}}) \quad \text{with e. g.} \quad \boldsymbol{p}_d = \begin{bmatrix} k_1\dot{q}_1 \\ \vdots \\ k_f\dot{q}_f \end{bmatrix}, \quad k_i = const\,. \tag{22}$$

The choice $\boldsymbol{Q} = \boldsymbol{I}$ for the artificial mass matrix turned out to be reasonable for manipulators with revolute joints. Compared to e. g. $\boldsymbol{Q} = \boldsymbol{M}(\boldsymbol{q})$, these decoupled "masses" avoid undesired dynamic couplings between the coordinates that make the dynamic transition to the equilibrium position less predictable, see also HOLLERBACH and SUH (1987). Examples are given in the following.

5.1 Repeatable Motions

If there are no other restrictions, it is usually desired to reach the initial joint positions at the end of a closed-loop trajectory of the position outputs $\boldsymbol{y}_P$ in order to be able to carry out repeated tasks in the same manner. This can be realized by artificial springs in the joints that are not tensioned in the initial configuration $\boldsymbol{q}_0$:

$$U(\boldsymbol{q}) = \frac{1}{2}\sum_{i=1}^{f} c_i\,(q_i - q_{0,i})^2\,, \qquad c_i = const\,. \tag{23}$$

A repeatable motion of a planar manipulator with three redundant degrees of freedom tracking a circular trajectory $\hat{\boldsymbol{y}}_P(t)$ (no contact constraints) is shown in Fig. 2.

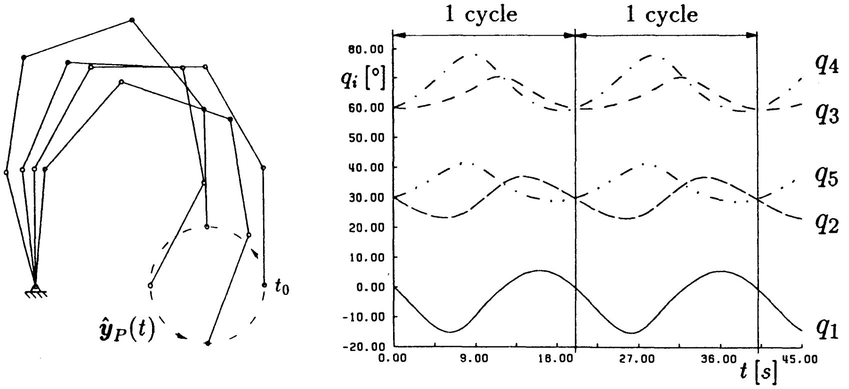


Fig. 2: Repeatable motion in joint space ($f = 5$, $m_P = 2$, $m_F = 0$)

5.2 Hybrid Position/Force Control

The motion of a planar manipulator with one redundant d.o.f. along a wall with prescribed contact force under a potential of type (23) is shown in Fig. 3a. The contact force $\lambda(t)$ converges asymptotically towards its desired value $\hat{\lambda}$, Fig. 3b.

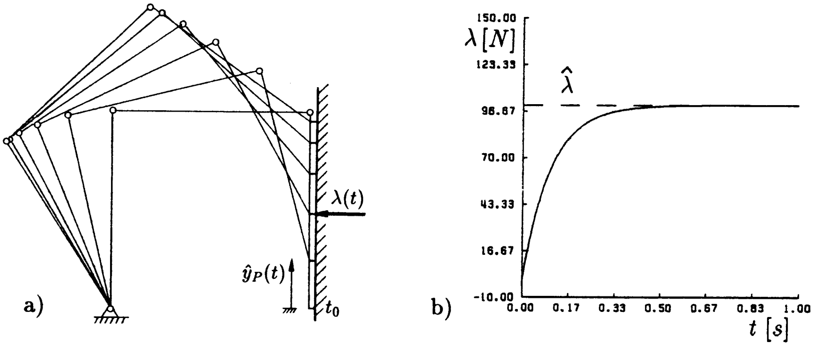


Fig. 3: Hybrid position-/force control ($f = 3$, $m_P = 1$, $m_F = 1$)

5.3 Collision avoidance

A main advantage of kinematically redundant robots is their ability to avoid collisions with obstacles in the workspace while tracking a desired trajectory. This can be achieved by artificial potential forces repulsing the manipulator arm from the obstacle. To a discrete point P_i on the manipulator with the absolute position vector $\boldsymbol{r}_i$ and the shortest distance $d(\boldsymbol{r}_i)$ from the obstacle a potential $U(\boldsymbol{r}_i)$ is assigned, refer to Khatib (1986):

$$U(\boldsymbol{r}_i) = \begin{cases} \frac{1}{2}\kappa\left(\frac{1}{d(\boldsymbol{r}_i)} - \frac{1}{d_0}\right)^2 & \text{for} \quad d(\boldsymbol{r}_i) \le d_0\,, \\ 0 & \text{for} \quad d(\boldsymbol{r}_i) > d_0\,. \end{cases} \tag{24}$$

Here, κ is a scaling factor and d_0 denotes the distance of influence of the obstacle. The artificial repulsing force on P_i derived from $U(\boldsymbol{r}_i)$ for $d(\boldsymbol{r}_i) \le d_0$ is:

$$\boldsymbol{f}_i(\boldsymbol{r}_i) = -\left(\frac{\partial U(\boldsymbol{r})}{\partial \boldsymbol{r}}\right)^{\mathrm{T}}_{\boldsymbol{r}=\boldsymbol{r}_i} = \kappa\left(\frac{1}{d(\boldsymbol{r}_i)} - \frac{1}{d_0}\right)\frac{1}{(d(\boldsymbol{r}_i))^2}\left(\frac{\partial d(\boldsymbol{r})}{\partial \boldsymbol{r}}\right)^{\mathrm{T}}_{\boldsymbol{r}=\boldsymbol{r}_i}\,. \tag{25}$$

To get the control force $\boldsymbol{p}$ needed in Eq. (21), the forces $\boldsymbol{f}_i$ on all n_{Pi} discrete points P_i are projected into the joint space and there summarized:

$$\boldsymbol{p} = \sum_{i=1}^{n_{Pi}} \boldsymbol{p}_i \quad \text{with} \quad \boldsymbol{p}_i = \left(\frac{\partial \boldsymbol{r}_i(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^{\mathrm{T}} \boldsymbol{f}_i\,. \tag{26}$$

Two examples for trajectories of a planar five-link manipulator with this collision avoiding strategy are shown Fig. 4.

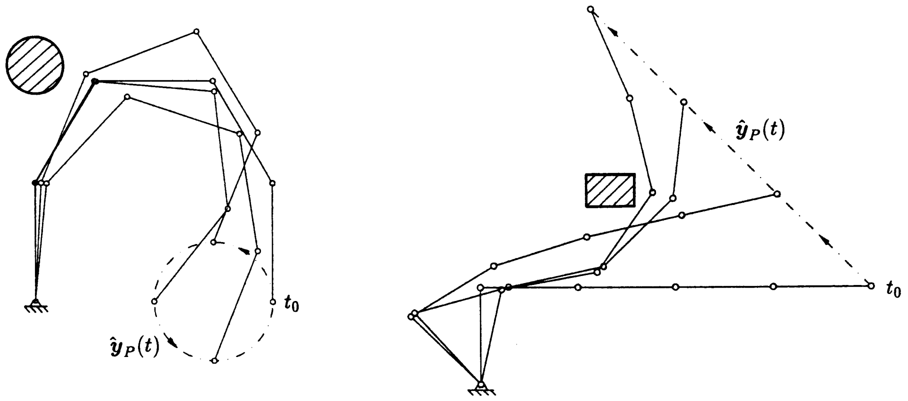


Fig. 4: Collision avoidance ($f = 5$, $m_P = 2$, $m_F = 0$)

6 Conclusion

The dynamics of a constrained redundant manipulator in the subspace of its redundant degrees of freedom is prescribed by artificial applied forces that generate asymptotically stable equilibrium positions in this subspace. Control goals such as repeatability in the joint motions or collision avoidance can be realized. It is a "local" optimization procedure instantaneously minimizing the "constraint" on the motion in the subspace of the redundant degrees of freedom. Further improvements can be expected by application of "global" dynamical optimization schemes covering whole trajectories, that are, however, more complex. Other future investigations concern control of combined kinematically and statically redundant manipulators where the control inputs needed to realize a desired motion can be minimized, if no explicit control of the contact forces is desired.

References

HOLLERBACH, M. J. and SUH, K. C., 1987, "Redundancy Resolution of Manipulators through Torque Optimization," *IEEE J. Rob. Autom.*, Vol. 3, pp. 308–316.

ISIDORI, A., 1989, *Nonlinear Control Systems*, Springer, Berlin.

KHATIB, O., 1986, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robotics Research*, Vol. 5, pp. 90–98.

KHATIB, O., 1987, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE J. Rob. Autom.*, Vol. 3, pp. 43–53.

PARS, L.A., 1968, *A Treatise on Analytical Dynamics*, Heinemann, London.

YOSHIKAWA, T., 1990, *Foundations of Robotics*, MIT Press, Cambridge.

ANALYSIS OF MECHANISMS BY THE DUAL INERTIA OPERATOR

M. Shoham and V. Brodsky
Department of Mechanical Engineering,
Technion - Israel Institute of Technology, Haifa 32000, Israel

ABSTRACT

The application of dual numbers to kinematics is based on the principle of transference that extends vector algebra to dual vector (motor) algebra. No such direct extension exists however, for dynamics. Inertia binor is used to obtain the dual momentum, from which the dual equations of motion are derived. This derivation raises the dual dynamic equations to six dimensions, and in fact, it does not act on the dual vector as a whole, but rather on its real and dual parts as two distinct real vectors.

In this investigation, the dual inertia operator is introduced. This gives the mass a dual property which has the inverse sense of Clifford's dual unit, namely, it reduces a motor to a rotor proportional to the vector part of the former, allowing direct relation of dual force to dual acceleration. As a result, the same equation of momentum which holds for linear motion, holds also for angular motion if dual force, dual acceleration, and dual inertia, replace their real counterparts.

This approach was implemented in a symbolic computer program. By adding dual number algebra, the program is able to handle dual quantities. Furthermore, applying the dual inertia rules, the dual equations of motion are obtained by replacing real with dual quantities as it is illustrated in the example of a three-degrees-of-freedom robot.

1. INTRODUCTION

Dual numbers were invented by Clifford in the nineteenth century [Clifford, 1873], and were generalized for kinematic applications by Kotelnikov's principle of transference [1895] that extends the algebra of vectors to that of dual-vectors (motors). This principle states that when dual-numbers replace real ones, the relations of vector algebra for kinematics of a rigid body with a fixed point, hold for motor algebra of a free body (see also [Hsia and Yang, 1981; Martinez and Duffy, 1993; Rooney, 1975]). But it was about one century latter before they were applied to mechanisms analysis [Dimentberg, 1965; Yang and Freudenstein, 1964; Yang, 1966].

In the 80's, as robot manipulators became popular, several investigators applied the dual-numbers to their kinematic analysis [Gu and Luh, 1987; Lee and Soni, 1979; McCarthy, 1986; Pennock and Yang, 1985; Pradeep, Yoder, and Mukundan, 1989]. Direct kinematics of robot manipulators, both position and orientation, can therefore, be obtained by consecutively multiplying 3X3 dual orthogonal matrices, one for each degree of freedom, instead of 4X4 real ones. It was shown [Gu and Luh, 1987] that the same multiplication also yields the Jacobian matrix in an explicit form.

J. Angeles et al. (eds.), Computational Kinematics, 129–138.

Only a few investigations over the past thirty years aimed, however, at the application of dual numbers to mechanisms dynamics [Dimentberg, 1965; Lee and Sony, 1979; Luh and Gu, 1984; Yang, 1967; Yang, 1967; Yanzhu, 1988]. In order to understand the difficulties involving the use of dual numbers in dynamics, it is best to quote from Dimentberg's book [1965] "... It will be appropriate to note here that the attempt to apply the transfer principle to dynamics no longer produces such simple relationship as can be obtained for kinematics and statics.... because the complex operator linking the kinematic and force screws cannot be obtained from the corresponding affine operator linking the angular velocity vector with the moment by substitution of complex for real quantities. As a result many dynamic and static problems must be solved on the basis of general screw theory with the screw expressed by means of six Plucker coordinates. [page 61]".

Also, on page 137, he remarked that even though the three-dimensional dual-kinematic equations are expanded into six real ones, there is still a discrepancy in the derivation of the dynamic equations " ...Then the dynamic equations decompose into two groups of three equation each. But those three equations that express the relation of the principal part of screw *U*, i.e., the angular-velocity vector, to the moment of the external forces will be not the principal part of the equations, but rather their moment part. The corresponding vector equation will be not the principal, but the moment part of the screw equation. Thus, the differential equations for the principal part of the kinematic screw are not the principal part of the basic differential equations, but, to the contrary, are their moment part.". The same problem obviously, arose in Yanzhu's [1988] investigation, where he reversed the order of the velocity screw in order to derive the Newton-Euler equations in a screw matrix form.

Furthermore, Yang [1966] showed that the acceleration of a rigid body is not a dual vector, as it does not follow the relevant transformation rule. Nevertheless, the acceleration can be composed in the regular manner of a dual vector into real and dual parts, a quantity Yang called a pseudo-dual vector [Yang, 1967].

In order to overcome the problem of dual number representation of a rigid body dynamics, Kotelnikov (according to Dimentberg) introduced the inertia binor concept. The binor is a combination of two dual matrices which, when applied to a dual vector transforms it into a new dual vector, according to the rule:

$$\hat{\mathbf{r}}' = (\mathbf{A})\,\hat{\mathbf{r}} = \hat{\mathbf{A}}\,\mathbf{r} + \hat{\mathbf{A}}^{+}\,\mathbf{r}_o \qquad (1)$$

where $\mathbf{r}$ and $\mathbf{r}_0$ are, respectively, the real and dual parts of a dual vector $\hat{\mathbf{r}}$, and $\hat{\mathbf{A}}$ and $\hat{\mathbf{A}}^{+}$ are dual matrices, generally different. The symbol ^ defines a pair of a real and dual quantities.

The inertia binor about a body-fixed coordinate system is defined as:

$$(\mathbf{T}) = (\hat{\mathbf{T}},\hat{\mathbf{T}}^{+}) = \left(\begin{bmatrix} \varepsilon I_{xx} & S_z + \varepsilon I_{xy} & -S_y + \varepsilon I_{xz} \\ -S_z + \varepsilon I_{xy} & \varepsilon I_{yy} & S_x + \varepsilon I_{yz} \\ S_y + \varepsilon I_{xz} & -S_x + \varepsilon I_{yz} & \varepsilon I_{zz} \end{bmatrix}, \begin{bmatrix} m & -\varepsilon S_z & \varepsilon S_y \\ \varepsilon S_z & m & -\varepsilon S_x \\ -\varepsilon S_y & \varepsilon S_x & m \end{bmatrix} \right) \qquad (2)$$

where ε - the dual unit

m, I_{ij} - body mass, and moments of inertia matrix element, respectively

S_i - body moments about axis i

Multiplying the above equation by the dual velocity, we obtain the dual momentum:

$$\hat{\mathbf{p}} = (\mathbf{T})\,\hat{\mathbf{u}} = \hat{\mathbf{T}}\,\omega + \hat{\mathbf{T}}^{+}\,\mathbf{v} \tag{3}$$

where $\hat{\mathbf{u}} = \omega + \varepsilon\mathbf{v}$ - is the kinematic screw of the body, in accordance with Dimentberg's definition. Thus we obtain the dynamic equation in dual form:

$$\frac{d\hat{\mathbf{p}}}{dt} = \frac{d}{dt}\,[(\mathbf{T})\,\hat{\mathbf{u}}] = \hat{\mathbf{f}} \tag{4}$$

where $\hat{\mathbf{f}}$ is a dual force (wrench).

Examining the binor operation and its effect on a screw, Dimentberg observed that replacement of the real quantities of any real operator with dual ones does not yield a binor. Hence, the screw formulation for dynamics is not a straight generalization of a vector formula into the dual-number domain. In other words, the principle of transference does not extend to rigid body dynamics.

It has to be noted that even though dual matrices were applied in earlier investigations to the Lagrangian dynamics of rigid bodies (see for example [Luh and Gu, 1984]), their use was limited there to the kinematic part, and after separation of the dual expressions into linear and angular parts, the dynamic terms were calculated in the traditional way.

Dual numbers were applied to dynamics of a rigid body in almost complete dual number form by Yang [1967, 1969]. The dual momentum about the body's center of mass is utilized in his study as follows:

$$\begin{gathered} \hat{\mathbf{h}} = m\mathbf{v} + \varepsilon\mathbf{h} \\ \mathbf{h} = \mathbf{I}\,\omega \end{gathered} \tag{5}$$

where $\mathbf{v}$ - is the velocity of the center of mass with respect to the inertial frame;

$\mathbf{I}$ - is the body moment of inertia about its center of mass.

If dual number algebra is applied, then the expression for the momentum is obtained through an inertia binor multiplication (Eq. 3). But such an operation expands the equation's dimensionality to six, and actually operates on two separate sets of three-dimensional real vectors. For dual dynamic analysis, the six-dimensional real vector is in turn, combined into a three-dimensional dual one in reverse order.

In order to obtain a general formulation of the dynamics of a rigid body in dual form, and to utilize its advantage of compactness and efficiency, we propose to consider the inertia as being of a dual nature and replace the binor operator with a much simpler dual inertia operator. Under the new representation, the spaces of wrenches and their integrals (i.e. dual forces and dual momenta) can directly be connected with those of twist derivatives and twists (i.e. dual accelerations and dual velocities), respectively.

2. FORMULATION OF DUAL MASS OPERATOR

Let us begin our dynamic analysis with the motion of a particle. The use of dual numbers and dual vectors in representing particle motion is obviously formal, since they refer to motion of rigid bodies about lines not of points. Hence, formally, a particle velocity in space is described by a dual vector as follows (this is a pure vector; hence, in its dual representation, is a pure dual number) :

$$\hat{\mathbf{u}} = \varepsilon\mathbf{v} \tag{6}$$

and its acceleration is,

$$\hat{\mathbf{a}} = \varepsilon\mathbf{a} = \frac{d}{dt}\varepsilon\mathbf{u} = \frac{d}{dt}\hat{\mathbf{u}}. \tag{7}$$

When the particle of mass m moves with acceleration, there is a force which acts on it in accordance with Newton's law,

$$\mathbf{f} = m\mathbf{a} \tag{8}$$

In terms of the dual numbers as defined by Clifford, the force is the rotor part of a motor, hence a real vector,

$$\hat{\mathbf{f}} = \mathbf{f} \tag{9}$$

whereas the linear acceleration is the dual part of a motor, hence a pure dual vector,

$$\hat{\mathbf{a}} = \varepsilon\mathbf{a} \tag{10}$$

In order to satisfy Eq. (8) it is necessary to include with the mass a dual operator in order to transform the pure dual vector into a real one. This transformation can be achieved if the dual symbol ε in the acceleration is eliminated. We propose to add such an operator to the mass and denote it formally by $d/d\varepsilon$, even though it has no derivative sense.

This at first sight artificial operation, is justified if one considers it in the light of Clifford's definitions of a motor and the dual symbol ε. The operator $d/d\varepsilon$ applied to any motor, changes it into a rotor parallel to its vector axis and proportional to the vector part of it. We can easily see that $d/d\varepsilon$ operating on a motor twice, also reduces it to zero. Accordingly, Newton's law in dual vectors reads

$$\hat{\mathbf{f}} = m\frac{d}{d\varepsilon}\hat{\mathbf{a}} = m\frac{d}{d\varepsilon}(\varepsilon\mathbf{a}) = m\,\mathbf{a} \tag{11}$$

Considering dynamical relations such as Newton's law or the expression for momentum, we can conclude that with respect to dual-vectors the mass has a special property, that it acts on the vector part of a motor (pseudo-motor) and transforms it into the rotor part of a pseudo-motor (motor). Therefore, we can combine the mass and its dual operator into the dual mass operator $\hat{m}$ as follows

$$\hat{m} = m\frac{d}{d\varepsilon} \tag{12}$$

Using this operator we can rewrite the equation of motion of a particle

$$\hat{\mathbf{f}} = \hat{m}\hat{\mathbf{a}}$$

$$\hat{\mathbf{h}}_l = \hat{m}\hat{\mathbf{v}} \tag{13}$$

where $\mathbf{h}_l$ is the linear momentum of a particle.

In accordance with the above formulation one can obtain the moment of momentum of a rigid body as the sum (integral) of moments of momentum of all elementary particles which constitute the body. Carrying out this derivation the dual moment of momentum, $\mathbf{h}_c$, becomes:

$$\hat{\mathbf{h}}_c = \varepsilon\mathbf{I}\omega \tag{14}$$

where $\mathbf{I}$ - is the body inertia matrix.

Comparing this equation with the analogous real one it is seen that in order to satisfy the dual relation we must multiply $\mathbf{I}$ by ε. Note that ω is a rotor, or in other words the real part of the velocity motor. Accordingly we can rewrite (14) in the form:

$$\hat{\mathbf{h}}_c = (\varepsilon \mathbf{I})(\omega + \varepsilon v) = \varepsilon \mathbf{I}\hat{v} \tag{15}$$

Note that the linear momentum of a rigid body is still a real vector:

$$\hat{\mathbf{h}}_l = \hat{m}\hat{v} = m\frac{d}{d\varepsilon}(\omega + \varepsilon v) = mv \tag{16}$$

it seems to be natural to combine momentum and moment of momentum into a new quantity - dual momentum, given by the expression:

$$\hat{\mathbf{h}} = \hat{\mathbf{h}}_l + \hat{\mathbf{h}}_c \tag{17}$$

After substitution (13) and (15) in (17) we finally obtain,

$$\hat{\mathbf{h}} = \hat{m}\hat{v} + \varepsilon \mathbf{I}\hat{v} = (m\frac{d}{d\varepsilon} + \varepsilon \mathbf{I})(\omega + \varepsilon v) = mv + \varepsilon \mathbf{I}\omega \tag{18}$$

We can expand the mass into a diagonal matrix, if a unified notation is desired. Then we define dual inertia as follows:

$$\hat{\mathbf{M}} = \hat{m} + \hat{\mathbf{I}} = m\frac{d}{d\varepsilon} + \varepsilon \mathbf{I} = \begin{bmatrix} m\frac{d}{d\varepsilon} + \varepsilon I_{xx} & \varepsilon I_{xy} & \varepsilon I_{xz} \\ \varepsilon I_{yx} & m\frac{d}{d\varepsilon} + \varepsilon I_{yy} & \varepsilon I_{yz} \\ \varepsilon I_{zx} & \varepsilon I_{zy} & m\frac{d}{d\varepsilon} + \varepsilon I_{zz} \end{bmatrix} \tag{19}$$

At this point we can summarize the derivation of dual momentum by the following compact form:

$$\hat{\mathbf{h}} = \hat{\mathbf{M}}\hat{v} \tag{20}$$

To derive the dual force acting on the body, we differentiate equation (20) with respect to time in the moving body-fixed coordinate system:

$$\hat{\mathbf{f}} = \frac{d}{dt}\hat{\mathbf{h}} = \hat{\mathbf{M}}\frac{d\hat{v}}{dt} + \omega \times \hat{\mathbf{M}}\hat{v} \tag{21}$$

It is seen from Eqs. (20) and (21) that both dual momentum and dual force are obtained without using the inertia binor, and without artificially interchanging the dual with the real part. In fact, Eq. (20) and its time derivative might be considered as an extension of the principle of transference to dynamics.

3. DYNAMIC ANALYSIS OF MECHANISMS

In this section we use the dual inertia operator to obtain the equations of motion of a mechanism. We propose to base the sought equations on the virtual-work and D'Alembert principles [Shoham and Srivatsan,1992] since almost all components of this equation are parts of the Jacobian matrix which is explicitly obtained through the dual-direct-kinematics expression. This equation has the form:

$$\mathbf{n} = \mathbf{J}^T\mathbf{f} + \sum_{i=1}^{L}\mathbf{J}_i^T\left[\begin{bmatrix} \mathbf{m}_i & 0 \\ 0 & \mathbf{I}_i \end{bmatrix}\left[\begin{matrix} \mathbf{g} \\ 0 \end{matrix} + \mathbf{J}_i\ddot{\theta} + \dot{\mathbf{J}}_i\dot{\theta}\right] + \right.$$
$$\left.\begin{bmatrix} \Omega_i & 0 \\ 0 & \Omega_i \end{bmatrix}\left[\begin{bmatrix} \mathbf{m}_i & 0 \\ 0 & \mathbf{I}_i \end{bmatrix}\mathbf{J}_i\dot{\theta}\right]\right] \tag{22}$$

where: $\mathbf{f} = [f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]$ - vector of external force and moment
$\mathbf{n}$ - vector of joint moment or force
$\mathbf{J}$ - Jacobian matrix for the end-effector
$\mathbf{J}_i$ - Jacobian matrix for center of mass of the i-th link
m_i - mass of the i-th link
L - number of links
$\mathbf{h}_i$ - moment of momentum of the i-th link about its center of mass
$\mathbf{g}$ - gravitation vector

and

$$\mathbf{m}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \qquad \Omega_i = \begin{bmatrix} 0 & -\omega_{iz} & \omega_{iy} \\ \omega_{iz} & 0 & -\omega_{ix} \\ -\omega_{iy} & \omega_{ix} & 0 \end{bmatrix}$$

We now extend this equation to dual formulation. The vector of generalized coordinates in dual form is given by:

$$\hat{\theta} = (\hat{q}_1, \hat{q}_2, \ldots \hat{q}_L)^T \tag{23}$$

where $\hat{q}_i = \theta_i$ - for a revolute joint

$\hat{q}_i = \varepsilon d_i$ - for a prismatic joint

$\hat{q}_i = \theta_i + \varepsilon d_i$ - for a cylindrical joint.

Using the dual inertia operator to relate dual force and acceleration (19) we, finally, obtain:

$$\hat{\mathbf{n}} = \hat{\mathbf{J}}^T \hat{\mathbf{f}} + \sum_{i=1}^{L} \hat{\mathbf{J}}_i^T \hat{\mathbf{M}}\left[\varepsilon \mathbf{g} + \hat{\mathbf{J}}_i \ddot{\hat{\theta}} + \dot{\hat{\mathbf{J}}}_i \dot{\hat{\theta}}\right] + \Omega \hat{\mathbf{M}} \hat{\mathbf{J}}_i \dot{\hat{\theta}}\Big] \tag{24}$$

where: $\hat{\mathbf{f}} = (f_x + \varepsilon\tau_x, f_y + \varepsilon\tau_y, f_z + \varepsilon\tau_z)^T$ - dual force applied at the end-effector

$\hat{\mathbf{M}} = \hat{\mathbf{m}} + \hat{\mathbf{I}} = \mathbf{m}\frac{d}{d\varepsilon} + \varepsilon \mathbf{I}$ - dual 3x3 inertia operator

$\hat{\mathbf{J}}_i$, $\hat{\mathbf{J}}$ - dual Jacobians, obtained in accordance with a known algorithm, e.g. [Gu and Luh, 1987].

One can see that the 6 x L and 6 x 6 matrices of equation (22) have been reduced to 3 x L and 3 x 3 matrices in (24). The advantage of the dual-form equation (24), apart from its compactness, is reduction of the number of generalized coordinates when the manipulator has cylindrical joints (see below).

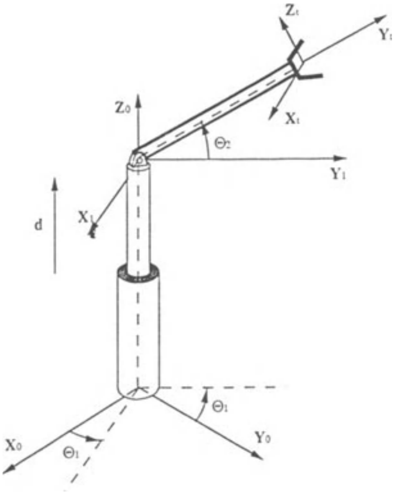
4. COMPUTER PROGRAM FOR SYMBOLIC DERIVATION OF A MECHANISM DUAL-DYNAMIC EQUATIONS

A symbolic computer program based on the Mathematica software package, was developed to obtain the symbolic representation of a mechanism dynamic equations.

Utilizing the dual inertia operator, derivation of the dual dynamic equation is direct extension of the real one. Since the same mathematical formulas hold also for dual vectors, it is possible to define the algebraic rules of dual numbers, and then the development environment becomes "dually" oriented where numbers can be pure real, pure dual, or a combination of both (this is similar to the program's ability to handle complex numbers). Once these rules are defined, all build-in mathematical rules, e.g., vector, scalar, and matrix multiplication, as well as special dual rules such as function of dual numbers, become applicable. As a result, it was relatively simple to expand the Mathematica program to deal with dual numbers, which facilitates the derivation of a mechanism symbolic dynamic equations once kinematics and inertia parameters are given.

5. ILLUSTRATIVE EXAMPLE

As an example, consider a three-degrees-of-freedom robot, shown below. First the dual transformation matrices are derived and then from their corresponding columns, the dual Jacobian is constituted. The dual inertia operator is then applied to obtain the joint's dual forces.



Transformation from the tool coordinate system to coordinate system 1 is given by:

$$\hat{A}_T^1 = \begin{bmatrix} 1 & \varepsilon l\, s_2 & -\varepsilon l\, c_2 \\ 0 & c_2 & s_2 \\ \varepsilon l & -s_2 & c_2 \end{bmatrix} \tag{25}$$

and complete dual transformation from the tool to the world coordinate system by means of:

$$\hat{A}_T^0 = \begin{bmatrix} c_1 - \varepsilon(l\, s_1 s_2 + d s_1) & s_1 + \varepsilon(d c_1 + l\, c_1 s_2) & -\varepsilon l\, c_1 \\ -s_1 c_2 + \varepsilon d s_2 c_1 & c_1 s_2 - \varepsilon d c_2 s_1 & s_2 \\ -s_1 s_2 + \varepsilon(l\, c_1 + d c_1 s_2) & -c_1 s_2 + \varepsilon(l\, s_1 - d s_1 s_2) & c_2 \end{bmatrix} \tag{26}$$

where s_i and c_i denote $\sin\theta_i$ and $\cos\theta_i$, respectively.

Because of the cylindrical pair, we consider the manipulator as having only two dual degrees of freedom (instead of three real ones)

$$\hat{q}_1 = \theta_1 + \varepsilon d$$
$$\hat{q}_2 = \theta_2 \tag{27}$$

The first rotation and translation is about the z-axis in frame 0 (see Fig. 1) and the second about the x-axis in frame 1, hence, the third column of (44) is the first column of the Jacobian matrix, and the first column of (43) is the second column of the latter; we then obtain:

$$\hat{\mathbf{J}} = \begin{bmatrix} -\varepsilon l\, c_2 & 1 \\ s_2 & 0 \\ c_2 & \varepsilon l \end{bmatrix} \tag{28}$$

Let **r** be the distance from the origin of coordinate system 1 to the center of mass of link 1, and let the link inertia matrix, in a link-attached coordinate system at the center of mass, be:

$$\mathbf{I} = \begin{bmatrix} I & 0 & 0 \\ 0 & I_0 & 0 \\ 0 & 0 & I \end{bmatrix} \tag{29}$$

The Jacobian matrix $\hat{\mathbf{J}}_1$ for the center of mass is obtained by replacing r for l..

The dynamic equation (24) requires knowledge of the angular velocity matrix, which is also a part of the Jacobian matrix, given by:

$$\Omega_1 = \begin{bmatrix} 0 & -c_2\dot{\theta}_1 & s_2\dot{\theta}_1 \\ c_2\dot{\theta}_1 & 0 & -\dot{\theta}_2 \\ -s_2\dot{\theta}_1 & \dot{\theta}_2 & 0 \end{bmatrix} \tag{30}$$

Substituting Eqs. (28,29,30) in Eq.(24), the dual forces are obtained:

$$\hat{\mathbf{n}} = \begin{bmatrix} -\varepsilon l\, c_2 & s_2 & c_2 \\ 1 & 0 & \varepsilon l \end{bmatrix} \begin{bmatrix} f_x \qquad {}_{x} \\ f_y + \varepsilon\tau_y \\ f_z + \varepsilon\tau_z \end{bmatrix} + \begin{bmatrix} -\varepsilon r c_2 & s_2 & c_2 \\ 1 & 0 & \varepsilon r \end{bmatrix}$$

$$\{\hat{\mathbf{M}} \left(\begin{bmatrix} 0 & -\varepsilon r c_2 & 1 \\ -g s_2 \varepsilon \;+ & s_2 & 0 \\ -g c_2 \varepsilon & c_2 & \varepsilon r \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 + \varepsilon \ddot{d} \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} \varepsilon r s_2 \dot{\theta}_2 & 0 \\ c_2 \dot{\theta}_2 & 0 \\ -s_2 \dot{\theta}_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 + \varepsilon \dot{d} \\ \dot{\theta}_2 \end{bmatrix} \right) + \tag{31}$$

$$\begin{bmatrix} 0 & -c_2\dot{\theta}_1 & s_2\dot{\theta}_1 \\ c_2\dot{\theta}_1 & 0 & -\dot{\theta}_2 \\ -s_2\dot{\theta}_1 & \dot{\theta}_2 & 0 \end{bmatrix} \hat{\mathbf{M}} \begin{bmatrix} -\varepsilon r c_2 & 1 \\ s_2 & 0 \\ c_2 & \varepsilon r \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 + \varepsilon \dot{d} \\ \dot{\theta}_2 \end{bmatrix} \}$$

where $\hat{\mathbf{M}}$ is as per (19).

After simple calculations, we obtain the vector of dual forces in the directions of the corresponding joint axes, or more precisely, the corresponding dual degrees of freedom

$$\hat{\mathbf{n}} = \begin{bmatrix} f_y s_2 + f_z c_2 + m\ddot{d} - mg + mrc_2\ddot{\theta}_2 - mrs_2\dot{\theta}_2^2 + \\ \varepsilon[\tau_y s_2 + \tau_z c_2 - f_x l\, c_2 + mr^2 c_2^2 \ddot{\theta}_1 - 2mr^2 c_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + \\ (I_0 s_2^2 + Ic_2^2)\ddot{\theta}_1 + 2(I_0 - I)c_2 s_2 \dot{\theta}_1 \dot{\theta}_2] \\ \\ f_x - mrc_2\ddot{\theta}_1 + 2mrs_2\dot{\theta}_1\dot{\theta}_2 + \\ \varepsilon[\tau_x + f_z l + mrc_2\ddot{d} + mr^2\ddot{\theta}_2 + mr^2 c_2 s_2 \dot{\theta}_1^2 - mgrc_2 + \\ I\ddot{\theta}_2 + (I - I_0)c_2 s_2 \dot{\theta}_1^2] \end{bmatrix} \tag{32}$$

This simple derivation of the inverse dynamics of a three degrees-of-freedom robot, yields even more than is needed. The bottom term in (34) contains not only the moment about the second joint but also the force along it, which is not specifically needed for control purposes but is useful in robot synthesis and design.

6. CONCLUSIONS

As the principle of transference was not extended to dynamics, dual number representation of the dynamics of a rigid body was unobtainable merely through replacement of real with dual numbers. Rather, the inertia binor was utilized to derive the dual momentum, which actually operates on two separate three-dimensional real vectors. By introducing the dual inertia operator and development from the basic particle equation, the equation of motion of a rigid body is obtained in a complete dual three-dimensional form. This operator may be considered as the inverse of Clifford's dual number, reducing a motor to a rotor proportional to the vector part of the former, and thus relating the dual force acting on a particle to its dual acceleration. There is no need for the inertia binor, and at no point are the equations expanded to six dimensions. The dual force is obtained merely by manipulation of dual-generalized displacements (dual angles), dual velocities, and dual transformation matrices.

A mechanism equation of motion is written in such a manner that most of its kinematic elements are parts of the Jacobian matrix. This lend itself to dual representation since the dual Jacobian is explicitly obtained from the dual-direct kinematics of the mechanism. The application of the dual inertia operator to the analysis of mechanisms is illustrated by an example of a three-degrees-of-freedom robot. Apart from the compact equation obtained by the dual representation, it has to be noted that since this particular robot contains a cylindrical joint which counts in dual terms as one degree of freedom, the dimensionality is further reduced to two dual-degrees-of-freedom.

A symbolic computer program was developed to implement the derivation of the dual dynamic equations. By adding such rules as dual number algebra and dual inertia operator, the build-in mathematical library was adopted to operate on dual-numbers. This provides a useful and convenient tool which enables the derivation the dual dynamic equation merely by replacing real with dual numbers.

REFERENCES

Clifford, W.K., "Preliminary Sketch of Bi-Quaternions," *Proc. London Mathematic Society*, vol. 4, pp. 381-395, 1873.

Dimentberg, F.M., *The Screw Calculus and its Applications in Mechanics*, (Izdat. "Nauka", Moscow, USSR, 1965) English translation: AD680993 (Clearinghouse for Federal and Scientific Technical Information).

Hsia, L.M., and Yang, A.T., "On the Principle of Transference In Three-Dimensional Kinematics," *Trans. ASME, J. of Mechanical Design,* Vol. 103, July 1981, pp. 652-656.

Gu, Y.L., and Luh, J.Y.S., "Dual-Number Transformation and Its Applications to Robotics," *IEEE J. of Robotics and Automation,* Vol. RA-3, No. 6, December, 1987, pp. 615-623.

Lee, I.P.J., and Soni, A.H., "Dynamic Analysis of Spatial Mechanisms Using Dual Successive Screw Method and D'Alembert's Principle," *J. of Mechanical Design,* Vol. 101, October, 1979, pp. 569-581.

Luh, J.Y.S., and Gu ,Y.L., "Lagrangian Formulation of Robot Dynamics with Dual - Number Transformation for Computational Simplification", *Proc. 1984 Conf. Information Sciences and systems* , Princeton, NJ, March, 1984, pp. 680- 684.

Martinez, J.M.R., and Duffy, J., "The Principle of Transference: History, Statement and Proof," *Mechanisms and Machine Theory,* Vol. 28, No. 1, 1993, pp. 165-177.

McCarthy, J.M., "Dual Orthogonal Matrices in Manipulator Kinematics," *Int. J. of Robotics Research,* Vol. 5, No. 2, Summer, 1986, pp. 45-51.

Pennock, G.R., and Yang, A.T., "Application of Dual-Number Matrices to the Inverse Kinematics Problem of Robot Manipulators," *ASME J. of Mechanisms, Transmissions, and Automation in Design,* Vol. 107, June 1985, pp. 201-208.

Pradeep, A.K., Yoder, P.J., and Mukundan, R., " On the Use of Dual-Matrix Exponentials in Robotic Kinematics," *The Int. J. of Robotics Research,* Vol. 8, No. 5, Oct. 1989, pp. 57-66.

Rooney, J., "on the Principle of Transference," in *Inst. Mechanical Engineering Proc. 4-th World Congress of IFToMM 5,* Newcastle-upon-Tyne, England, Sept. 1975, pp. 1089-1094.

Shoham, M. and Srivatsan, R., "Automation of Surface Finishing Processes", *Robotics & Computer-Integrated Manufacturing,* Vol. 9, No. 3, 1992, pp. 219- 226.

Yang, A.T., and Freudenstein F., "Application of Dual-Number Quaternion Algebra to the Analysis of Spatial Mechanisms," *Trans. ASME, J. of Applied Mechanics,* Series E, Vol. 86, June 1964, pp. 300-308.

Yang, A.T., "Acceleration Analysis of Spatial Four-Link Mechanisms", *Trans. ASME, J. of Engineering for Industry,* Vol. 88, 1966, pp. 296-300.

Yang, A.T., "Application of Dual Quaternions to the Study of Gyrodynamics,"*Trans. ASME, J. of Engineering for Industry,* Series B, Vol. 89, Feb. 1967, pp. 137-143.

Yang, A.T., "Analysis of an Offset Unsymmetric Gyroscope With Oblique Rotor Using (3 X 3) Matrices With Dual-Number Elements," *Trans. ASME, J. of Engineering for Industry,* Series B, Vol. 91, No. 3, Aug. 1969, pp. 535-542.

Yanzhu, L., "Screw-Matrix Method in Dynamics of Multibody Systems," *Acta Mechanica Sinica,* Vol. 4, No. 2, May, 1988, Science press, Beijing, China, Allerton Press, Inc., New York, U.S.A..

Part 4

Parallel Manipulators

Direct Kinematics in Analytical Form of a General Geometry 5-4 Fully-Parallel Manipulator

Carlo Innocenti
Dipartimento di ingegneria delle costruzioni meccaniche
Università di Bologna - Viale Risorgimento, 2 - 40136 Bologna, Italy
Fax (+39)-51-6443446 - e-mail:MECCAPP8@INGBO1.CINECA.IT

Vincenzo Parenti-Castelli
Istituto di Ingegneria - Facoltà di Ingegneria
Università di Ferrara - Via Scandiana, 21- 44100 Ferrara, Italy
Tel. (+39)-532-65150 - Fax (+39)-532-740983

Abstract. This paper presents the analytical form solution of the direct position analysis for a fully-parallel manipulator that features a base and a platform connected by six adjustable-length legs whose extremities meet the base and the platform respectively at four and five points. When the leg lengths are given, the manipulator becomes a statically determined structure that can be assembled in different configurations. The direct position analysis aims at solving all possible configurations. In the paper, the analysis is first reduced to the solution of a three non-linear equation system in three unknowns, then two unwanted unknowns are eliminated thus obtaining a final 24th order polynomial equation in only one unknown. The twenty-four roots of the equation provide as many configurations of the 5-4 structure in the complex field. Numerical examples support the new theoretical findings.

1 Introduction

Recently the interest of many researchers has been focused on parallel mechanisms. These, indeed, offer appealing performances in many fields, ranging from automobile to aerospatial applications, in addition to robotics. Outstanding characteristics are a favorable payload to manipulator weight ratio and high structural stiffness, which allow for great precision of end effector positioning. Parallel manipulators are closed-chain mechanisms with one or more loops where only a certain number of pairs are actively controlled. Fully parallel mechanisms, in particular, feature two rigid bodies, base and platform, connected by six adjustable-length legs whose extremities meet the base and platform at single or multiple spherical kinematic pairs. The six variable-length legs provide the platform with six degrees of freedom with respect to the base. When a set of leg lengths is given, the mechanism becomes a statically determined structure.

J. Angeles et al. (eds.), Computational Kinematics, 141–152.

Position analysis of fully-parallel manipulators involves a direct and an inverse problem. The inverse problem is trivial for it asks for the leg lengths when position and orientation of the platform are given with respect to the base. On the contrary, the direct position analysis (DPA), which calls for the position and orientation of the platform when the leg lengths are given, is a difficult problem since non-linear equations are involved and many solutions are possible. Numerical methods prove difficulty to find all solutions and do not guarantee finding all of them, whereas analytical solutions would give more insight into the mechanism behavior and make it possible to improve the control strategies for platform motion. Although some general ideas are worthy of consideration when tackling the solution of a new mechanism in analytical form (Innocenti and Parenti-Castelli, 1992), a general, successful procedure has not yet been found. Moreover, for the time being, general approaches would lead to a system of closure equations too cumbersome to deal with even using the most powerful symbolic computational packages. As a consequence, every mechanism, or at least a class of mechanisms, must be solved by a customized strategy. In this perspective analytically solving a mechanism may represent a further step toward the solution of more difficult cases.

Basic ideas for solving the DPA in analytical form can be summarized as follows. First, the position and orientation of the platform with respect to the base are parameterized by as small number of parameters as possible, so that a closure system with a reduced number of equations and unknowns can be written. Second, by a suitable elimination procedure a final polynomial equation in only one unknown is sought. When the whole process succeeds, the roots of the final polynomial equation (which is solved numerically when the order is greater than four) can be found and, by substitution, all solutions can be found. Consequently the order of the polynomial equation represents the number of solutions of the DPA in the complex field.

The DPA of numerous fully-parallel manipulators has been solved in analytical form and presented in the literature. Much work has been done by Duffy, Song, Merlet, Waldron and their coworkers (Lin et al., 1992; Lin et al., 1990; Merlet, 1990; Nanua et al., 1990; Zhang and Song, 1992); further references can be found in (Innocenti and Parenti-Castelli, 1992). Nevertheless other mechanisms still remain unsolved. Well known among these is the most general - and difficult - case, the 6-6 fully-parallel manipulator also referred to as the 6-6 generalized Stewart platform. For special geometries such as, for example, base and platform planar or symmetrically shaped, the DPA in analytical form can be more workable (Merlet, 1990; Lin et al., 1992; Zang and Song, 1992).

In this paper the analytical form solution of the DPA of the fully-parallel manipulator shown in Fig. 1, is presented. Such an arrangement is called the 5-4 fully-parallel manipulator because of the number of connection points on the base and the platform. When the leg lengths are frozen the manipulator becomes the structure shown in Fig. 2, which is called the 5-4 structure. Different leg arrangements are possible while maintaining the 5-4 pattern. These cases are shown in (Lin et al., 1992), where their DPA in analytical form is also presented when the base and platform are both planar. The mechanism studied in the present paper is more general than that considered in (Lin et al., 1992) since it allows for both base and platform with general geometry, i.e., spherical pair centers both on the base and on the platform do not necessarily belong to a plane.

The direct position analysis of the 5-4 structure here considered is carried out in two steps. Position and orientation of the platform with respect to the base are first parameterized by three angles, a closure system of three equations in these angles being obtained. Then, by a suitable procedure, two unwanted unknowns are eliminated thus obtaining a final 24th-order polynomial equation in only one unknown, which is representative of the structure assembly. Hence, from each root of the polynomial equation, and via a back substitution, all sets of three angles can be determined. Thus, twenty-four closures of the 5-4 structure are possible in the complex field.

Finally, a numerical example supports the new theoretical result.

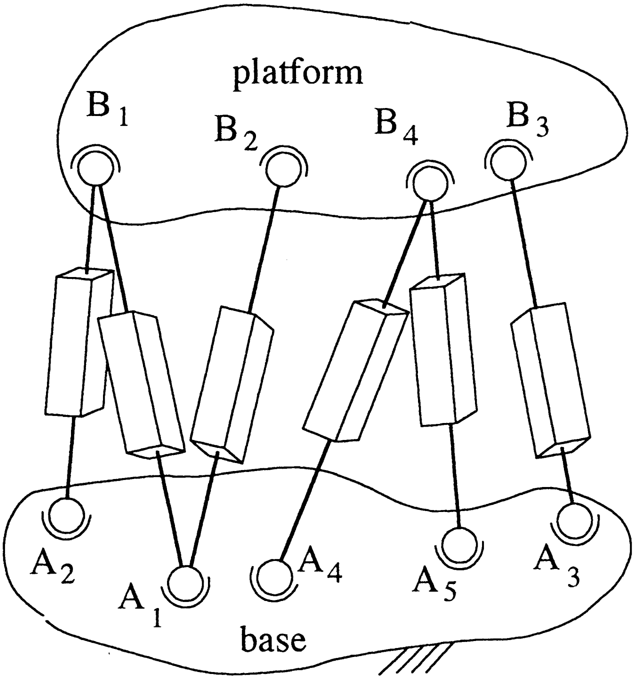


Fig. 1 Schematic of a 5-4 fully-parallel manipulator

2 Direct kinematics

2.1 Kinematic model

Inspection of the 5-4 structure (see Fig. 2) shows that for a given set of leg lengths the three triangles $\tau_1 \equiv A_1B_1A_2$, $\tau_2 \equiv A_1B_2B_1$ and $\tau_3 \equiv A_4A_5B_4$ can be considered as rigid bodies. If the platform is momentarily disconnected from the legs at points B_1, B_2, and B_3 and the distance between points B_1 and B_2 is maintained, the triangles τ_i, i=1,2,3, can rotate about the axes A_1A_2, A_1B_1, and A_4A_5 respectively, and their angular positions with respect to each other or the base can be defined by three angles θ_1, θ_2, and θ_3. Thus the positions of points B_1, B_2, and B_4 with respect to the base is uniquely defined by these angles. In particular θ_1 defines B_1, θ_1 and θ_2 define B_2, and θ_3 defines B_4. Points B_1, B_2, and B_4 also belong to the platform, and their mutual distances are known quantities. By lying down that corresponding mutual distances evaluated with respect to the base and to the platform are the same, two equations can be written. One involves θ_1 and θ_3, when considering distance B_1B_4, and the other θ_1, θ_2, and θ_3, when considering distance B_2B_4.

A third equation can be obtained as follows. Angles θ_i, i=1,2,3, uniquely define the positions of the three points B_1, B_2 and B_4, and consequently parameterize the position and orientation of the platform with respect to the base. Thus the position of point B_3 of the platform can also be defined

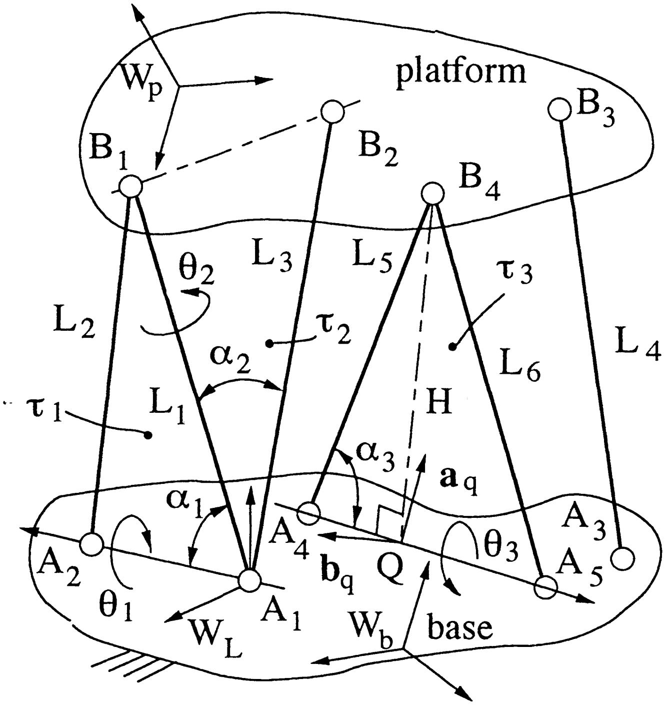


Fig. 2 Kinematic model of the 5-4 structure

with respect to the base as a function of the three angles θ_i, i=1,2,3. Point A_3 belongs to the base, then the distance between points A_3 and B_3 can be written as a function of the three angles θ_i, i=1,2,3. Considering that points A_3 and B_3 also belong to a leg whose length, L_4, is known, a constraint equation can be written. Then a third equation in the unknowns θ_i, i=1,2,3, is obtained.

2.2 Closure equations

With reference to Fig. 2, W_b represents the reference system where the base points A_i, (i=1,...,5), are given. Then a second reference system, W_L, fixed to the base is chosen with origin in A_1, z axis from A_1 to A_2, and x axis parallel to the plane x-y of W_b. The x axis of W_L has a positive component along the x axis of W_b (when vector $(A_1\text{-}A_2)$ is parallel to the x axis of W_b, then the x axis of W_L is chosen parallel to the y axis of W_b). Moreover, an arbitrary coordinate system W_p fixed to the platform is chosen.

Since triangles τ_1, τ_2 and τ_3 can be considered as rigid bodies, angles α_1, α_2, and α_3 can be evaluated in terms of their sines and cosines as follows:

$$u_1 = \frac{L_1^2 + (A_2\text{-}A_1)_b^2 - L_2^2}{2\ L_1 \surd\, [(A_2\text{-}A_1)_b^2]} \quad ; \qquad v_1 = + \surd\, (1\text{-}u_1^2) \qquad (1)$$

$$u_2 = \frac{L_1{}^2 + L_3{}^2 - (B_2\text{-}B_1)_p{}^2}{2\, L_1L_3} \quad ; \qquad v_2 = + \sqrt{(1\text{-}u_2{}^2)} \tag{2}$$

$$u_3 = \frac{L_5{}^2 + (A_5\text{-}A_4)_b{}^2\text{-}\ L_6{}^2}{2\, L_5\sqrt{[(A_5\text{-}A_4)_b{}^2]}} \quad ; \qquad v_3 = + \sqrt{(1\text{-}u_3{}^2)} \tag{3}$$

where u_j=cosα_j; v_j=sinα_j, j=1,2,3, and L_i, i=1,2,...,6, are leg lengths. A vector subscript points to the coordinate system where vector components are to be evaluated.

The distance H of point B_4 from the line A_4A_5 is equal to $L_5 \cdot v_3$. The projection Q of point B_4 on line A_4A_5 is defined by:

$$(Q - A_4)_b = \frac{L_5 \cdot u_3}{\sqrt{[(A_5 - A_4)_b{}^2]}} \cdot (A_5 - A_4) \tag{4}$$

Two unitary vectors $\mathbf{a}_q$ and $\mathbf{b}_q$ are chosen so that vectors ($\mathbf{a}_q$, $\mathbf{b}_q$, $(A_5 - A_4)$) represent a right hand reference system; vector $\mathbf{a}_q$ is parallel to the x,y plane of W_b and has a positive component along the x axis of W_b (if $(A_5 - A_4)$ is parallel to the x axis of W_b, vector $\mathbf{a}_q$ is parallel to the y axis of W_b). Position vectors of points B_1, B_2 and B_4 in reference system W_L can be obtain as follows:

$$(B_1 - A_1)_L = L_1 \cdot \begin{bmatrix} v_1s_1 \\ -v_1c_1 \\ u_1 \end{bmatrix} \tag{5}$$

$$(B_2 - A_1)_L = L_3 \cdot \begin{bmatrix} v_2c_1s_2 + u_1v_2s_1c_2 + v_1u_2s_1 \\ v_2s_1s_2 - u_1v_2c_1c_2 - v_1u_2c_1 \\ -v_1v_2c_2 + u_1u_2 \end{bmatrix} \tag{6}$$

$$(B_4 - Q)_L = H \cdot (\mathbf{a}_q \cdot c_3 + \mathbf{b}_q \cdot s_3) \tag{7}$$

where c_j=cosθ_j and s_j=sinθ_j (j=1,2,3).

The first closure equation can now be written by setting the distance between points B_4 and B_1 be equal when measured in W_L and in W_p. The equation can be written as follows:

$$(B_4\text{-}B_1)_L{}^2 = (B_4\text{-}B_1)_p{}^2 \tag{8}$$

The following position holds:

$$[(B_4\text{-}Q)_L + (Q\text{-}A_1)_L - (B_1\text{-}A_1)_L]^2 - (B_4\text{-}B_1)_p{}^2 = 0 \tag{9}$$

which leads to:

$$[H^2 + (Q\text{-}A_1)_L{}^2 + L_1{}^2 - (B_4\text{-}B_1)_p{}^2]/2$$

$$+ (B_4\text{-}Q)_L \cdot (Q\text{-}A_1)_L - (B_4\text{-}Q)_L \cdot (B_1\text{-}A_1)_L - (Q\text{-}A_1)_L \cdot (B_1\text{-}A_1)_L = 0 \tag{10}$$

Equation (10), taking into account relations (5) and (7), can be written as:

$$\sum_{i,j=0,2} a_{ij}\cdot s_1{}^{p(i)}\cdot c_1{}^{q(i)}\cdot s_3{}^{p(j)}\cdot c_3{}^{q(j)} = 0 \tag{11}$$

where a_{ij} are coefficients which depend only on the geometry of the 5-4 structure, p(i) = (i mod 2) means the remainder of the division of integer i by 2, and q(i) = [i - p(i)]/2. Equation (11) represents a sine-cosine linear equation in the two unknowns θ_1 and θ_3.

A second closure equation can be written by setting the distance between points B_2 and B_4 be equal when measured in W_L and W_p. The equation is:

$$(B_4-B_2)_L{}^2 = (B_4-B_2)_p{}^2 \tag{12}$$

that can be written as:

$$[(B_4-Q)_L+ (Q-A_1)_L- (B_2-A_1)_L]^2 - (B_4-B_2)_p{}^2 = 0 \tag{13}$$

and, subsequently, as:

$$[H^2 + (Q-A_1)_L{}^2 + L_3{}^2 - (B_4-B_2)_p{}^2]/2 + (B_4-Q)_L\cdot(Q-A_1)_L$$

$$- (B_4-Q)_L\cdot(B_2-A_1)_L - (Q-A_1)_L\cdot(B_2-A_1)_L= 0 \tag{14}$$

Equation (14), taking into account relations (6) and (7), can be written as:

$$\sum_{i,j,k=0,2} m_{ijk}\cdot s_1{}^{p(i)}\cdot c_1{}^{q(i)}\cdot s_2{}^{p(j)}\cdot c_2{}^{q(j)}\cdot s_3{}^{p(k)}\cdot c_3{}^{q(k)} = 0 \tag{15}$$

Here m_{ijk} are coefficients that depend only on the geometry of the 5-4 structure. Equation (15) represents a sine-cosine linear equation in the three unknowns θ_1, θ_2, and θ_3.

Before developing the third closure equation, the position of point B_3 is revalued in W_p as follows. Since all points B_i, i=1,2,3,4, belong to the same rigid body, the following relation holds:

$$(B_3-B_1)_p = u\cdot(B_2-B_1)_p + v\cdot(B_4-B_1)_p + w\cdot(B_2-B_1)_p \times (B_4-B_1)_p \tag{16}$$

where u, v, and w are constant scalar quantities to be determined.

Vector equation (16) represents three linear scalar equations in the unknowns u, v and w. Equation (16), after rearrangement, leads to:

$$\begin{bmatrix} (B_2-B_1)_p{}^2 & (B_2-B_1)_p\cdot(B_4-B_1)_p \\ (B_2-B_1)_p\cdot(B_4-B_1)_p & (B_4-B_1)_p{}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (B_2-B_1)_p\cdot(B_3-B_1)_p \\ (B_4-B_1)_p\cdot(B_3-B_1)_p \end{bmatrix} \tag{17}$$

$$w\cdot[(B_2-B_1)\times(B_4-B_1)]^2 = (B_3-B_1)\cdot (B_2-B_1)\times(B_4-B_1) \tag{18}$$

The two linear equations (17) provide u and v, while equation (18) provides w.

By imposing the distance between points A_3 and B_3 measured in W_L equal to leg length L_4, the third closure equation can be obtained:

$$(B_3\text{-}A_3)_L{}^2 = L_4{}^2 \tag{19}$$

which can be written as:

$$[(B_3\text{-}B_1)_L + (B_1\text{-}A_1)_L - (A_3\text{-}A_1)_L\,]^2 - L_4{}^2 = 0 \tag{20}$$

By adopting the following positions:

$$(B_1\text{-}A_1)_L = \mathbf{a};\quad (B_2\text{-}A_1)_L = \mathbf{b};\quad (B_4\text{-}Q)_L = \mathbf{c};\quad (Q\text{-}A_1)_L = \mathbf{d};\quad (A_3\text{-}A_1)_L = \mathbf{e} \tag{21}$$

vector $(B_3\text{-}B_1)_L$ becomes:

$$(B_3\text{-}B_1)_L = u(\mathbf{b}\text{-}\mathbf{a}) + v(\mathbf{c}+\mathbf{d}\text{-}\mathbf{a}) + w(\mathbf{b}\text{-}\mathbf{a}) \times (\mathbf{c}+\mathbf{d}\text{-}\mathbf{a}) \tag{23}$$

and the closure equation (20) can be written in the following form:

$$\begin{aligned}&[(B_3\text{-}B_1)_p{}^2 + L_1{}^2 + (A_3\text{-}A_1)^2 - L_4{}^2]/2 + v(\mathbf{a}\cdot\mathbf{c}) + v(\mathbf{a}\cdot\mathbf{d}) + u(\mathbf{a}\cdot\mathbf{b}) - (u + v)\mathbf{a}^2 + \\ &+ w(\mathbf{a}\times\mathbf{b})\cdot\mathbf{c} + w(\mathbf{a}\times\mathbf{b})\cdot\mathbf{d} - v(\mathbf{c}\cdot\mathbf{e}) - v(\mathbf{d}\cdot\mathbf{e}) - u(\mathbf{b}\cdot\mathbf{e}) + (u+v)(\mathbf{a}\cdot\mathbf{e}) + \\ &- w(\mathbf{b}\times\mathbf{c})\cdot\mathbf{e} + w(\mathbf{a}\times\mathbf{c})\cdot\mathbf{e} - w(\mathbf{d}\times\mathbf{e})\cdot\mathbf{b} + w(\mathbf{d}\times\mathbf{e})\cdot\mathbf{a} - w(\mathbf{a}\times\mathbf{b})\cdot\mathbf{e} - \mathbf{a}\cdot\mathbf{e} = 0\end{aligned} \tag{24}$$

Moreover, by considering that

$$\mathbf{a}\cdot\mathbf{b} = L_1\cdot L_3\cdot u_2 \tag{25}$$

and

$$\mathbf{a}\times\mathbf{b} = L_1\cdot L_3 \begin{bmatrix} -u_1v_2s_1s_2 + v_2c_1c_2 \\ u_1v_2c_1s_2 + v_2s_1c_2 \\ v_1v_2s_2 \end{bmatrix} \tag{26}$$

the third closure equation (19) can be written as follows:

$$\sum_{i,j,k=0,2} n_{ijk}\cdot s_1{}^{p(i)}\cdot c_1{}^{q(i)}\cdot s_2{}^{p(j)}\cdot c_2{}^{q(j)}\cdot s_3{}^{p(k)}\cdot c_3{}^{q(k)} = 0 \tag{27}$$

where n_{ijk} are coefficients which depend only on the geometry of the 5-4 structure. Equation (27) represents a sine-cosine linear equation in the three unknowns θ_1, θ_2, and θ_3.

2.3 Elimination of θ_2 and θ_3

To solve the direct position analysis in analytical form, two unknowns must be eliminated from the three closure equations (11), (15), and (27). Since equation (11) does not contain the unknown θ_2, the dialytic elimination of θ_2 between equations (15) and (27) could be performed as a first step then, in a second step, either of the unknowns θ_1 or θ_3 could be dialytically eliminated between equation (11) and the resultant of equations (15) and (27). This procedure is correct but cumbersome, so, in the following a single-step elimination procedure proposed in (Lin et al., 1992; Innocenti, 1993) will be used.

The closure equations are first transformed into algebraic equations by substituting for sine and cosine of θ_2 and θ_3 the well-known expressions:

$$c_i = (1 - t_i^2)/(1 + t_i^2); \qquad s_i = 2 \cdot t_i/(1 + t_i^2) \tag{28}$$

where $t_i = \tan(\theta_i/2)$. After substitution, equations (11), (15) and (27) can be respectively written in compact form as follows:

$$\sum_{j=0,2} R_j \cdot t_3^j = 0 \tag{29.1}$$

$$\sum_{j,k=0,2} W_{jk} \cdot t_2^j \cdot t_3^k = 0 \tag{29.2}$$

$$\sum_{j,k=0,2} Z_{jk} \cdot t_2^j \cdot t_3^k = 0 \tag{29.3}$$

where:

$$R_j = \sum_{i=0,2} r_{ij} \cdot s_1^{p(i)} \cdot c_1^{q(i)} \qquad (j=0,1,2) \tag{30.1}$$

$$W_{jk} = \sum_{i=0,2} w_{ijk} \cdot s_1^{p(i)} \cdot c_1^{q(i)} \qquad (j,k=0,1,2) \tag{30.2}$$

$$Z_{jk} = \sum_{i=0,2} z_{ijk} \cdot s_1^{p(i)} \cdot c_1^{q(i)} \qquad (j,k=0,1,2) \tag{30.3}$$

and coefficients r_{ij}, w_{ijk}, z_{ijk}, (i,j,k=0,1,2), resulting from substitution of positions (28) into equations (11), (15), and (27), are functions of quantities a_{ij}, m_{ijk}, and n_{ijk}; therefore they depend on the geometry of the 5-4 structure only.

Multiplying equations (29) by suitable factors, sixteen equations are obtained that can globally be regarded as a homogeneous linear system in sixteen unknowns. In particular, the first eight equations are obtained by multiplying equation (29.1) by factors:

$$1,\ t_3,\ t_2,\ t_2 \cdot t_3,\ t_2^2,\ t_2^2 \cdot t_3,\ t_2^3,\ t_2^3 \cdot t_3 \tag{31}$$

and the remaining eight equations, grouped in fours, are obtained by multiplying equations (29.2) and (29.3) by factors:

$$1,\ t_2,\ t_3,\ t_2 \cdot t_3 \tag{32}$$

The sixteen unknowns of the homogeneous linear system are represented, apart from an arbitrary scaling factor, by the following sixteen quantities:

$$1,\ t_2,\ t_2^2,\ t_2^3,\ t_3,\ t_2\cdot t_3,\ t_2^2\cdot t_3,\ t_2^3\cdot t_3,\ t_3^2, t_2\cdot t_3^2,\ t_2^2\cdot t_3^2, t_2^3\cdot t_3^2,\ t_3^3,\ t_2\cdot t_3^3,\ t_2^2\cdot t_3^3,\ t_2^3\cdot t_3^3 \tag{33}$$

In order to have non trivial solutions for the above-mentioned homogeneous linear system, the determinant of the coefficient matrix:

$$M = \begin{bmatrix}
R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 & 0 & 0 & 0 \\
0 & R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 & 0 & 0 \\
0 & 0 & R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 & 0 \\
0 & 0 & 0 & R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & R_0 & 0 & 0 & 0 & R_1 & 0 & 0 & 0 & R_2 \\
W_{00} & W_{10} & W_{20} & 0 & W_{01} & W_{11} & W_{21} & 0 & W_{02} & W_{12} & W_{22} & 0 & 0 & 0 & 0 & 0 \\
0 & W_{00} & W_{10} & W_{20} & 0 & W_{01} & W_{11} & W_{21} & 0 & W_{02} & W_{12} & W_{22} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & W_{00} & W_{10} & W_{20} & 0 & W_{01} & W_{11} & W_{21} & 0 & W_{02} & W_{12} & W_{22} & 0 \\
0 & 0 & 0 & 0 & 0 & W_{00} & W_{10} & W_{20} & 0 & W_{01} & W_{11} & W_{21} & 0 & W_{02} & W_{12} & W_{22} \\
Z_{00} & Z_{10} & Z_{20} & 0 & Z_{01} & Z_{11} & Z_{21} & 0 & Z_{02} & Z_{12} & Z_{22} & 0 & 0 & 0 & 0 & 0 \\
0 & Z_{00} & Z_{10} & Z_{20} & 0 & Z_{01} & Z_{11} & Z_{21} & 0 & Z_{02} & Z_{12} & Z_{22} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & Z_{00} & Z_{10} & Z_{20} & 0 & Z_{01} & Z_{11} & Z_{21} & 0 & Z_{02} & Z_{12} & Z_{22} & 0 \\
0 & 0 & 0 & 0 & 0 & Z_{00} & Z_{10} & Z_{20} & 0 & Z_{01} & Z_{11} & Z_{21} & 0 & Z_{02} & Z_{12} & Z_{22}
\end{bmatrix} \tag{34}$$

must vanish. Then:

$$\det M = 0 \tag{35}$$

represents the necessary and sufficient condition under which a couple of values t_2 and t_3 simultaneously satisfy equations (29). Equation (35) is the result of the elimination of unknowns t_2 and t_3 from equations (29).

When expressions (30) are substituted for the elements of matrix M, by direct computation it results that equation (35) can be written as:

$$\sum_{i=0,24} b_i\cdot s_1^{p(i)}\cdot c_1^{q(i)} = 0 \tag{36}$$

where coefficients b_i, i=(0,1,...,24) depend only on the geometry of the 5-4 structure.

If positions (28) for i=1 are substituted for sine and cosine in equation (36), it follows that:

$$\sum_{i=0,24} h_i\cdot t_1^i = 0 \tag{37}$$

that represents an algebraic equation in the unknown t_1 of order twenty-four. Coefficients h_i, i=(0,1,...,24), are known quantities that depend only on the geometry of the 5-4 structure. Equation (37) is the final result of the elimination of unknowns θ_2 and θ_3 from closure equations (11), (15), and (27). It provides 24 solutions for t_1 in the complex field.

2.4 Back substitution

For a given geometry of the 5-4 structure, equation (37) can be solved for t_1. Every root t_{1r}, r=1,2,...,24, provides the values s_{1r} and c_{1r} for s_1 and c_1 by means of relations (28), and the solution θ_{1r} for angle θ_1, can be determined. For every value θ_{1r} of θ_1, all elements of matrix M can now be computed by equations (30). Since condition (35) is satisfied, the homogeneous linear system of 16 equations, whose coefficient matrix is M, provides a non-trivial solution for the 16 unknowns (33). In particular, solutions t_{2r} and t_{3r} for the unknowns t_2 and t_3 are then determined. Actually, they are the second and fifth quantities of (33) when the first one is made unitary. Again, relations (28) provide the sine and cosine of θ_{2r}, and θ_{3r}, thus angles θ_{2r}, and θ_{3r} can be determined. Since every (θ_1, θ_2, θ_3) triplet defines a configuration of the 5-4 structure, it follows that for the 5-4 structure 24 configurations are possible in the complex field.

3 Case study

The direct position analysis of a 5-4 fully-parallel manipulator is reported in this section. With reference to Fig. 2, the geometry of the 5-4 structure is defined by the following data, with arbitrary length unit. Position vectors of points A_i, i=1,2,3,4,5 in base reference W_b are in the order (4,-2,1), (1,5,2), (-3,-4,-1), (-2,3,-2), (6,1,0). Position vectors of points B_j, j=1,2,3,4, in platform reference system W_p are in the order (5,4,4), (-2,1,3), (2,3,-3), (3,-6,5). The leg lengths L_i, i=1,2,...,6, are in the order 6.78, 4.58, 7.00, 8.83, 12.44, 9.11.

The results of the DPA for all twenty-four configurations are shown in Table 1 in terms of coordinates of points B_j, (j=1,...,4), in W_b. All solutions have been verified by inverse position analysis.

Table 1. Coordinates x, y, z in reference system W_b of points B_1, B_2, B_3, and B_4 for every configuration (only one out of two complex conjugate configurations is reported.)

#	x	y	z
1	(5.01956785, 0.00000000)	(4.01336765, 0.00000000)	(3.96113000, 0.00000000)
	(-1.99075338, 0.00000000)	(1.03099903, 0.00000000)	(2.98088840, 0.00000000)
	(2.01638037, 0.00000000)	(2.97675411, 0.00000000)	(-3.03217374, 0.00000000)
	(2.98487373, 0.00000000)	(-5.97269309, 0.00000000)	(5.02818701, 0.00000000)
2	(1.56385449, 0.00000000)	(3.42139699, 0.00000000)	(-2.26221546, 0.00000000)
	(-0.66318696, 0.00000000)	(-3.73995867, 0.00000000)	(-3.92211654, 0.00000000)
	(-3.96435898, 0.00000000)	(2.00334207, 0.00000000)	(-7.40303020, 0.00000000)
	(7.04394220, 0.00000000)	(-2.92105316, 0.00000000)	(-8.15644695, 0.00000000)
3	(1.34235715, 0.00000000)	(3.32454892, 0.00000000)	(-2.24877103, 0.00000000)
	(8.90648553, 0.00000000)	(2.47133853, 0.00000000)	(-1.22115543, 0.00000000)
	(4.18038607, 0.00000000)	(-1.17425139, 0.00000000)	(3.29256343, 0.00000000)
	(7.19944446, 0.00000000)	(-2.47706914, 0.00000000)	(-8.33447198, 0.00000000)
4	(1.07154018, 0.00000000)	(3.20326692, 0.00000000)	(-2.21224788, 0.00000000)
	(2.23885959, 0.00000000)	(3.16889458, 0.00000000)	(5.37960195, 0.00000000)
	(5.36038824, 0.00000000)	(-2.18647962, 0.00000000)	(1.18722481, 0.00000000)
	(7.52038695, 0.00000000)	(9.63042102, 0.00000000)	(2.48924820, 0.00000000)

Table 1 (continued)

5	(4.12514321, 0.00000000)	(4.38024067, 0.00000000)	(-1.29025504, 0.00000000)
	(10.48426203, 0.00000000)	(0.13678564, 0.00000000)	(-0.54547502, 0.00000000)
	(4.36483712, 0.00000000)	(-1.75614555, 0.00000000)	(3.32356236, 0.00000000)
	(7.25736521, 0.00000000)	(-2.30617224, 0.00000000)	(-8.39525806, 0.00000000)
6	(-1.55641752, 0.00000000)	(1.75861745, 0.00000000)	(0.01642529, 0.00000000)
	(-0.89947156, 0.00000000)	(-5.41394980, 0.00000000)	(-2.65241362, 0.00000000)
	(3.60620617, 0.00000000)	(-0.08909495, 0.00000000)	(-5.36254074, 0.00000000)
	(2.97078677, 0.00000000)	(-5.77947829, 0.00000000)	(5.27774965, 0.00000000)
7	(0.54566594, 0.00000000)	(2.95820529, 0.00000000)	(-2.07443922, 0.00000000)
	(4.55959244, 0.00000000)	(-2.30543616, 0.00000000)	(-5.97090848, 0.00000000)
	(-1.95842254, 0.00000000)	(0.10100381, 0.00000000)	(-8.75021188, 0.00000000)
	(8.92113465, 0.00000000)	(5.21431480, 0.00000000)	(-7.52984881, 0.00000000)
8	(0.56763720, 0.00000000)	(2.96871231, 0.00000000)	(-2.08207456, 0.00000000)
	(6.29086862, 0.00000000)	(4.35022969, 0.00000000)	(2.85108182, 0.00000000)
	(5.65633200, 0.00000000)	(-2.46183588, 0.00000000)	(-0.18093500, 0.00000000)
	(8.95569086, 0.00000000)	(8.29404568, 0.00000000)	(-4.58834275, 0.00000000)
9-10	(-1.92028430, 0.18943905)	(1.49683659, 0.24240132)	(0.75729097, -1.12849211)
	(1.45872664, -0.25139656)	(-0.43820406, 1.69220030)	(7.57862564, -0.49884871)
	(-5.98729698, -0.24049775)	(-1.00465406, 1.82755079)	(7.00432275, -0.77365509)
	(3.30724075, -0.21262471)	(-7.52505868, 0.38368057)	(2.18635332, 1.23417939)
11-12	(-0.06343771, -0.57884262)	(2.66729611, -0.29637671)	(-1.86538595, 0.33810911)
	(-3.63041762, 4.16989024)	(-7.99650246, -1.31890178)	(-2.76840029, -6.34465673)
	(-6.15192074, -0.54193989)	(2.15174433, -0.04240203)	(-6.52760258, 0.26183235)
	(6.55312948, 4.05666233)	(-8.58941136, 9.07731357)	(-11.86155430, -7.14933576)
13-14	(7.67181424, -1.46468268)	(5.13877186, -1.32490296)	(4.04003969, 4.88027270)
	(6.98775527, 0.17434734)	(3.53910508, 4.45047803)	(-5.61224324, 3.80696410)
	(0.63796195, -0.89506311)	(4.74222925, -0.25596295)	(0.51643720, 3.62289472)
	(10.61173565, -1.57841677)	(14.45294747, 4.66538743)	(-5.05362014, 10.97905453)
15-16	(-1.82549262, 2.76557083)	(0.84319449, 0.75301065)	(5.61716071, 3.02563797)
	(7.74142362, 5.18134230)	(0.77760801, 2.42857510)	(9.35536116, -3.12748014)
	(5.66011502, -0.02349371)	(-1.92088666, 1.34785212)	(0.68588930, -1.54154787)
	(6.79425093, 1.30698055)	(9.45433979, 1.91922319)	(5.21771109, -3.30869901)
17-18	(8.74940158, 0.49424291)	(5.96400427, -0.68652897)	(1.49617485, 6.28843151)
	(16.02750547, 10.73728234)	(-25.98402681, -41.70915767)	(44.21193364, -26.13852662)
	(24.84567831, 5.01017039)	(-65.89968503, 12.77368796)	(-10.77660035, -66.60553213)
	(15.95816486, -3.87429192)	(29.51312050, 12.54384233)	(-11.37916396, 28.04101000)
19-20	(8.82557583, -0.15927829)	(5.93503772, 0.83988708)	(1.92746342, -6.35704446)
	(-0.01314890, 7.19717413)	(22.51665587, -26.73504293)	(-27.11910421, -24.33709031)
	(12.66712640, 10.06368418)	(-7.97367778, -41.02338271)	(-41.77698994, 7.86430575)
	(2.36458704, -0.98513816)	(-9.52248577, 2.92310312)	(3.95954107, 6.86365577)
21-22	(-1.17427028, 3.24700238)	(1.02198897, 1.02342008)	(6.31926640, 2.57706658)
	(1.73421878, -1.49225442)	(4.08999128, -0.33950430)	(-2.05053113, 0.43059511)
	(-6.38191737, -0.30647878)	(5.20033726, -0.83228020)	(0.49669658, 4.42359038)
	(7.00513791, 1.29875706)	(9.76341220, 1.71225658)	(4.68323557, -3.48277167)
23-24	(1.83065526, -4.38396402)	(2.08793107, -1.83727455)	(7.87244824, -0.29097019)
	(-4.51615060, -0.51171999)	(-2.22521488, -1.57618618)	(2.00832076, -4.67397404)
	(0.98369071, 0.26794119)	(3.89546324, -0.13749610)	(-1.04687959, -0.38824148)
	(3.32108424, 0.25601072)	(-7.59858147, -0.42695331)	(2.05745658, -1.45099620)

4 Conclusions

This paper presented the direct position analysis in analytical form of a general-geometry 5-4 fully-parallel manipulator. By considering the 5-4 structure, which derives from the manipulator when inputs are given, an original kinematic model has been reported which reduces the closure equations to a three-equation system in three unknowns. By a single-step elimination procedure two unknowns are eliminated and a final 24th-order polynomial equation in only one unknown has been found. Since introduction of extraneous roots is avoided, twenty-four configurations are possible for the 5-4 manipulator in the complex field. Numerical results validated the new theoretical findings.

Acknowledgement

The financial support of MURST and CNR is gratefully acknowledged.

References

Innocenti C., 1993, "Analytical-Form Position Analysis of the 7-Link Assur Kinematic Chain with Four Serially-Connected Ternary Links," accepted for *ASME Journal of Mechanical Design.*

Innocenti, C., Parenti-Castelli, V., 1991, "Direct Kinematics of the Reverse Stewart Platform Mechanism," *IFAC International Symposium on Robot Control, SYROCO'91*, September 16-18, Vienna, Austria.

Innocenti, C., Parenti-Castelli, V., 1992, "Basic Ideas and Recent Techniques for the Analytical Form Solution of the Direct Position Analysis of Fully-Parallel Mechanism," *International Journal of Laboratory Robotics and Automation*, Vol. 4, pp.107-113.

Lin, W., Crane, C.D., and Duffy, J., 1992, "Closed-Form Forward Displacement Analyses of the 4-5 In-Parallel Platforms," *22nd ASME Biennial Mechanisms Conference*, Scottsdale, Arizona, DE-Vol.45, September 13-16, pp. 521-527.

Lin, W., Duffy, J., and Griffis, M., 1990, "Forward Displacement Analyses of the 4-4 Stewart Platforms," *Proceedings of the 21st ASME Mechanisms Conference*, Chicago, IL, September 16-19, pp. 263-269.

Merlet, J. P., 1990, *Les Robots Paralleles*, Hermes, Paris.

Nanua, P., Waldron, K. J., and Murthy, V., 1990, "Direct Kinematic Solution of a Stewart Platform," *IEEE Transaction on Robotics and Automation*, 6(4), pp. 438-444.

Zhang, C., Song, S., 1992, "Forward Position Analysis of Nearly General Stewart Platforms," *22nd ASME Biennial Mechanisms Conference*, Scottsdale, Arizona, DE-Vol.45, September 13-16, pp. 81-87.

THE KINEMATICS OF 3-DOF PLANAR AND SPHERICAL DOUBLE-TRIANGULAR PARALLEL MANIPULATORS

H. R. MOHAMMADI DANIALI, P. J. ZSOMBOR-MURRAY and J. ANGELES
Department of Mechanical Engineering & McGill Research Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada.
e-mail: hamid@mcrcim.mcgill.ca, paul@mcrcim.mcgill.ca and angeles@mcrcim.mcgill.ca

Abstract. Two double-triangular mechanisms are introduced here. These are planar and spherical three-degree-of-freedom mechanisms that consist of two triangles moving with respect to each other. Moreover, each side of the moving triangle intersects one corresponding side of the fixed one at a given point defined over this side. The direct kinematic analysis of the mechanisms leads to a quadratic equation for the planar and a polynomial of 16th degree for the spherical mechanism. Numerical examples are included that admit two real solutions for the former and four real solutions for the latter, among which only two positive values are acceptable. All solutions, both real and complex, are listed.

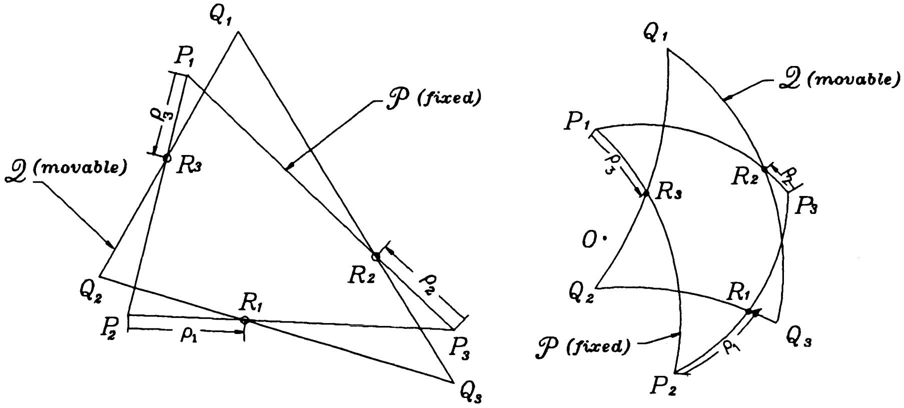
Key words: Direct Kinematics – Double-Triangular – Manipulator – Parallel – Planar – Spherical – 3-dof

1. Introduction

Parallel manipulators are known to offer some advantages over conventional serial manipulators when high accuracy, superior structural stiffness and low inertia are required. The best known spatial parallel manipulators are probably those of the platform type (Stewart, 1965). The kinematics of several planar parallel manipulators was investigated by Gosselin and Angeles (1990), Hunt (1983) and Gosselin and Sefrioui (1991). Spherical parallel manipulators constitute an important type in that they find applications in robotic wrists and other devices used to orient rigid bodies. The kinematics of a few spherical parallel manipulators was investigated by Gosselin and Angeles (1989, 1990), Craver (1989) and Gosselin et al. (1992a, 1992b).

In this paper we introduce first a new class of parallel manipulators in two versions, planar and spherical, as shown in Fig. 1. Similar to the double-tetrahedron mechanism, which was investigated by Tarnai and Makai (1988, 1989a, 1989b) and Zsombor-Murray and Hyder (1992), these two manipulators consist of two bounded rigid bodies whose bounding edges are in contact. The geometric model of a planar 3-dof double-triangular (DT) device consists of two triangles. These triangles move with respect to each other such that each side of the moving triangle intersects a corresponding side of the fixed triangle at a designated point defined over that fixed side. Moreover, the intersection point should lie within the physical boundaries of the respective edges in contact. For a given set of intersection points defined over the sides of the fixed triangle, we determine the position and orientation of

J. Angeles et al. (eds.), Computational Kinematics, 153–164.



a— Planar Manipulator

b— Spherical Manipulator

Fig. 1. Planar and Spherical Double-Triangular Parallel Manipulators

the moving triangle, a problem that we call *direct kinematics* (DK) of the device at hand. We show that the DK of this device leads to a quadratic equation. The two real solutions, for one numerical example, are included here.

On the other hand, the geometric model of a spherical 3-dof DT device consists of two spherical triangles. These triangles move with respect to each other so that each arc of the fixed triangle intersects one arc of the moving triangle at a designated point exactly as in the planar case. For a given set of intersection points defined over the sides of the fixed triangle, we determine the orientation of the moving triangle. We show that the DK of this device leads to a polynomial of 16th degree in the tangent of one half of one of the arcs sought. We include all real and complex solutions of this problem for one numerical example.

2. Planar DT Parallel Manipulator

Consider two triangles, $\mathcal{P}$ and $\mathcal{Q}$, with vertices $P_1P_2P_3$ and $Q_1Q_2Q_3$, respectively. Triangle $\mathcal{P}$ is designated the *fixed triangle* (FT), while $\mathcal{Q}$ is the *movable triangle* (MT), such that P_2P_3 intersects Q_2Q_3 at point R_1, P_3P_1 intersects Q_3Q_1 at R_2 and P_1P_2 intersects Q_1Q_2 at R_3. Moreover, R_i, for $i = 1, 2, 3$, cannot lie outside its corresponding vertices. Thus, feasible or admissible motions maintain R_i within edges $Q_{i+1}Q_{i-1}$ and $P_{i+1}P_{i-1}$, for $i = 1, 2, 3$, the sum and the difference in the foregoing subscripts being understood as *modulo 3*.

The motion of triangle $\mathcal{Q}$ can thus be described through changes in the edge length parameters, ρ_i, which locate R_i along a side of $\mathcal{P}$, measured from P_{i+1}, as shown in Fig. 1a, for $i = 1, 2, 3$. The non-negative displacements

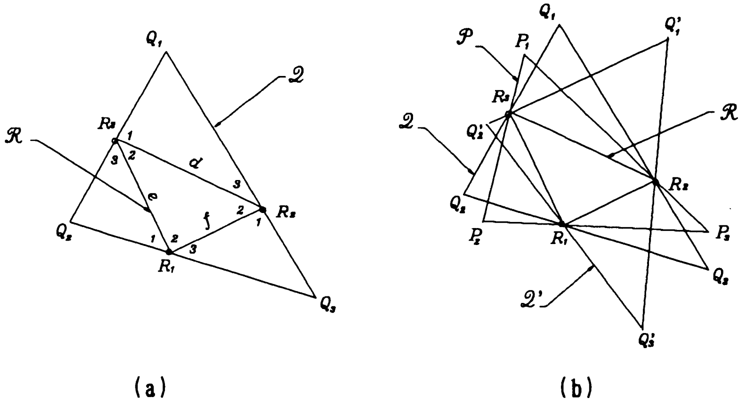


Fig. 2. a, Triangles $\mathcal{Q}$ and $\mathcal{R}$; b, Solutions of the Example Problem

ρ_i are assumed to be produced by actuators, and hence, they are termed the *actuator coordinates.* The coordinates of the moving triangle $\mathcal{Q}$, in turn, are the set of variables used to define its pose. Note that the Cartesian coordinates of the three vertices of $\mathcal{Q}$ can be used to define this pose.

2.1. Direct Kinematics

The direct kinematic problem of the manipulator described above is the subject of this subsection. This problem may be formulated as: *Given the actuator coordinates ρ_i, for $i = 1, 2, 3$, find the Cartesian coordinates of the vertices of triangle $\mathcal{Q}$.*

We solve this problem by *kinematic inversion*, i.e., by fixing the MT $\mathcal{Q}$ and letting the FT $\mathcal{P}$ to accommodate itself to the constraints imposed. To this end, we define points R_i at given distances ρ_i, for $i = 1, 2, 3$, on the edges of $\mathcal{P}$, thereby defining a triangle $R_1R_2R_3$, henceforth termed triangle $\mathcal{R}$, that is fixed to $\mathcal{P}$. Next, we let d, e and f be the lengths of the sides of this triangle. The problem now consists of finding the set of all possible positions of triangle $\mathcal{R}$ for which vertex R_i lies within the side $Q_{i+1}Q_{i-1}$, for $i = 1, 2, 3$, as shown in Fig. 2a. By carrying $\mathcal{R}$ back into its fixed configuration, while attaching $\mathcal{Q}$ rigidly to it, we determine the set of possible configurations of the MT for the given values of actuator coordinates.

In Fig. 2a we note that each vertex R_i is common to three angles labeled with numbers 1, 2 and 3. We will denote these angles by a subscripted capital letter. The subscript indicates one of the three angles common to that vertex, while the capital letter corresponds to the lower-case label of the opposite side of the triangle $R_1R_2R_3$. We thus have at vertices R_1, R_2 and R_3 the angles D_i, E_i and F_i, for $i = 1, 2, 3$.

Considering triangle $Q_1R_3R_2$, the law of sines for triangles yields

$$Q_1R_2 = a_1 \sin(F_1) \tag{1}$$

where

$$a_1 = \frac{d}{\sin(Q_1)}$$

Similarly, for triangle $Q_3R_2R_1$ we have

$$Q_3R_2 = a_2 \sin(D_3) \tag{2}$$

where

$$a_2 = \frac{f}{\sin(Q_3)}$$

Adding eq.(1) to eq.(2) gives

$$a_1 \sin(F_1) + a_2 \sin(D_3) = b \tag{3}$$

where

$$b = \overline{Q_1Q_3}$$

From triangle $Q_2R_1R_3$, we have

$$D_1 = \pi - F_3 - Q_2 \tag{4}$$

but

$$F_3 = \pi - F_1 - F_2 \tag{5}$$

Substitution of F_3 from eq.(5) into eq.(4) yields

$$D_1 = F_1 + F_2 - Q_2 \tag{6}$$

Again, we have

$$D_3 = \pi - D_1 - D_2 \tag{7}$$

Substitution of D_1 from eq.(6) into eq.(7) yields in turn

$$D_3 = G - F_1 \tag{8}$$

where

$$G = \pi - D_2 - F_2 + Q_2$$

Substituting the expression for $\sin(D_3)$ from eq.(8) into eq.(3), we obtain

$$b_1 \sin(F_1) + b_2 \cos(F_1) = b \tag{9}$$

where

$$b_1 = a_1 - a_2 \cos(G), \qquad b_2 = a_2 \sin(G)$$

In the above equations, we substitute now the equivalent expressions for cosines and sines given below:

$$\cos(F_1) = \frac{1-T^2}{1+T^2}, \qquad \sin(F_1) = \frac{2T}{1+T^2}$$

where

$T = \tan(F_1/2)$

Upon simplification, eq.(9) leads to

$$c_1T^2 + c_2T + c_3 = 0 \tag{10}$$

where

$c_1 = -b_2 - b, \qquad c_2 = 2b_1, \qquad c_3 = b_2 - b$

Solving eq.(10) for T gives

$$T = \frac{-b_1 \pm \sqrt{b_1^2 + b_2^2 - b^2}}{-(b_2 + b)} \tag{11}$$

The above expression leads us to the result below:

Theorem 1: *Given two triangles $\mathcal{R}$ and $\mathcal{Q}$, we can inscribe $\mathcal{R}$ in $\mathcal{Q}$ in at most two poses such that vertex R_i is located on the edges $Q_{i+1}Q_{i-1}$ of triangle $\mathcal{Q}$, for $i = 1, 2, 3$.*

2.2. Example

Consider the following sides assigned to the triangles $\mathcal{P}$ and $\mathcal{Q}$:

$Q_1Q_2 = 0.4\text{m} \qquad Q_2Q_3 = 0.5\text{m} \quad Q_3Q_1 = 0.6\text{m}$

$P_1P_2 = 0.29065\text{m} \quad P_2P_3 = 0.5\text{m} \quad P_3P_1 = 0.47875\text{m}$

Choose three points, R_1, R_2 and R_3, located by three actuator coordinates specified as $\rho_1 = 0.2$m, $\rho_2 = 0.14161$m and $\rho_3 = 0.03064$m. These values produce the lengths d, e and f given below:

$d = 0.33166\text{m}, \qquad e = 0.26458\text{m}, \qquad f = 0.2\text{m}$

The two roots of eq.(11) are:

$T_1 = 1.0788\ , \qquad T_2 = 0.4447$

i.e., $(F_1)_1 = 94.34°$, $(F_1)_2 = 48°$. Equations(1–8) are used to compute the other parameters, which leads to two poses of the triangle, Fig. 2b. The two triangles $\mathcal{Q}$ and $\mathcal{Q}'$ represent the two solutions which correspond to the assembly modes of the manipulator.

3. Spherical DT Parallel Manipulator

Consider an unit sphere with centre at O and a spherical triangle $P_1P_2P_3$, referred to as $\mathcal{P}$, on its surface. Moreover, a second spherical triangle, labeled $Q_1Q_2Q_3$, likewise referred to as $\mathcal{Q}$, is defined. Furthermore, the side P_2P_3 of $\mathcal{P}$, arbitrarily regarded as the FT, intersects the arc Q_2Q_3 of $\mathcal{Q}$, regarded as the MT, at point R_1. We denote by R_2 and R_3 the other intersection points, that are defined correspondingly. Moreover R_i, for $i = 1,2,3$, cannot lie outside its corresponding vertices. Thus, feasible or admissible motions maintain R_i within edges $Q_{i+1}Q_{i-1}$ and $P_{i+1}P_{i-1}$, for $i = 1,2,3$.

Thus, the motion of triangle $\mathcal{Q}$ can be described through the arc lengths ρ_i of Fig. 1b, or *actuator coordinates*, for $i = 1,2,3$. Likewise, the Cartesian coordinates of the moving triangle $\mathcal{Q}$ are the set of variables defining its orientation. Note that the Cartesian coordinates of the three vertices of $\mathcal{Q}$ can be determined once its orientation is given.

3.1. Direct Kinematics

Similar to the direct kinematic problem of the planar mechanism, the same problem, as pertaining to the spherical mechanism, may be formulated as: *Given the actuator coordinates ρ_i, for $i =$ '$1,2,3$, find the Cartesian coordinates of the vertices of triangle $\mathcal{Q}$.*

Likewise, we solve this problem by *kinematic inversion*, i.e., by fixing the MT $\mathcal{Q}$ and letting the FT $\mathcal{P}$ to accommodate itself to the constraints imposed. To this end, we define points R_i at given arc lengths ρ_i, for $i = 1,2,3$, on the edges of $\mathcal{P}$, thereby defining a triangle $R_1R_2R_3$, henceforth termed triangle $\mathcal{R}$, that is fixed to $\mathcal{P}$. Next, we let d, e and f be the sides of this triangle. The problem now consists of finding the set of all possible orientations of triangle $\mathcal{R}$ for which vertex R_i lies within the side $Q_{i+1}Q_{i-1}$, for $i = 1,2,3$, as shown in Fig. 3a. By carrying $\mathcal{R}$ back into its fixed configuration, while attaching $\mathcal{Q}$ rigidly to it, we determine the set of possible configurations of the MT for the given values of actuator coordinates.

In Fig. 3a we note that each vertex R_i is common to three spherical angles labeled with numbers 1, 2 and 3. Similar to the planar mechanism, we call them the spherical angles D_i, E_i and F_i, for $i = 1,2,3$.

We introduce now the definitions below:

$$s \equiv (d+e+f)/2, \quad k \equiv \sqrt{\frac{\sin(s-d)\,\sin(s-e)\,\sin(s-f)}{\sin(s)}} \tag{12}$$

From spherical trigonometry we have

$$D_2 = 2\arctan(\frac{k}{\sin(s-d)}) \tag{13}$$

$$E_2 = 2\arctan(\frac{k}{\sin(s-e)}) \tag{14}$$

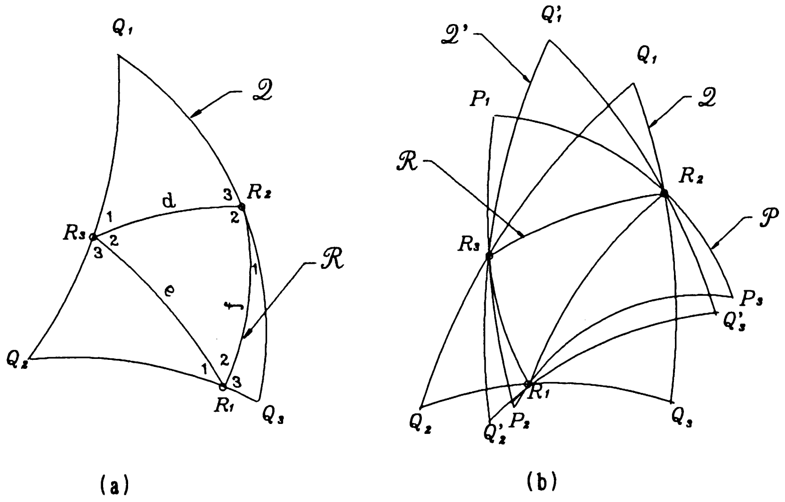


Fig. 3. a, Triangles $\mathcal{Q}$ and $\mathcal{R}$; b, Solutions of the Example Problem

$$F_2 = 2\arctan(\frac{k}{\sin(s-f)}) \tag{15}$$

Consider now the spherical triangle $Q_1R_3R_2$. Using the law of cosines for spherical triangles we have

$$\cos Q_1 = -\cos F_1 \cos E_3 + \sin F_1 \sin E_3 \cos d \tag{16}$$

Similarly, for the spherical triangles $Q_2R_1R_3$ and $Q_3R_2R_1$ we have

$$\cos Q_2 = -\cos D_1 \cos F_3 + \sin D_1 \sin F_3 \cos e \tag{17}$$

$$\cos Q_3 = -\cos E_1 \cos D_3 + \sin E_1 \sin D_3 \cos f \tag{18}$$

However,

$$D_3 = \pi - D_1 + D_2 \tag{19}$$

$$E_3 = \pi - E_1 + E_2 \tag{20}$$

$$F_3 = \pi - F_1 + F_2 \tag{21}$$

Substitution of the expressions for $\cos E_3$ and $\sin E_3$ from eq.(20) into eq.(16), we obtain

$$a_{11}c_1c_2 + a_{12}c_1s_2 + a_{13}s_1s_2 + a_{14}s_1c_2 + a_{15} = 0 \tag{22}$$

where

$$a_{11} = \cos E_2 \qquad a_{12} = -\sin E_2$$

$$\begin{aligned} a_{13} &= \cos d \cos E_2 & a_{14} &= \cos d \sin E_2 \\ a_{15} &= -\cos Q_1 & c_1 &= \cos F_1 \\ s_1 &= \sin F_1 & c_2 &= \cos E_1 \\ s_2 &= \sin E_1 \end{aligned}$$

Similarly, substitution of eq.(21) into eq.(17) yields:

$$a_{21}c_3c_1 + a_{22}c_3s_1 + a_{23}s_3s_1 + a_{24}s_3c_1 + a_{25} = 0 \tag{23}$$

where

$$\begin{aligned} a_{21} &= \cos F_2 & a_{22} &= -\sin F_2 \\ a_{23} &= \cos e \cos F_2 & a_{24} &= \cos e \sin F_2 \\ a_{25} &= -\cos Q_2 & c_3 &= \cos D_1 \\ s_3 &= \sin D_1 \end{aligned}$$

Likewise, substitution of eq.(19) into eq.(18) yields:

$$a_{31}c_2c_3 + a_{32}c_2s_3 + a_{33}s_2s_3 + a_{34}s_2c_3 + a_{35} = 0 \tag{24}$$

where

$$\begin{aligned} a_{31} &= \cos D_2 & a_{32} &= -\sin D_2 \\ a_{33} &= \cos f \cos D_2 & a_{34} &= \cos f \sin D_2 \\ a_{35} &= -\cos Q_3 \end{aligned}$$

Equations (22–24) must be solved simultaneously to determine the values of angles D_1, E_1 and F_1. In the above equations, we substitute now the equivalent expressions for cosines and sines given below:

$$c_i = \frac{1 - x_i^2}{1 + x_i^2}\,, \qquad s_i = \frac{2x_i}{1 + x_i^2}$$

where x_i, for $i = 1, 2, 3$, are tangents of one half of the angles F_1, E_1 and D_1, respectively.

Upon simplification, eqs.(22–24) lead to

$$d_1x_2^2 + d_2x_2 + d_3 = 0 \tag{25}$$
$$d_4x_2^2 + d_5x_2 + d_6 = 0 \tag{26}$$
$$d_7x_3^2 + d_8x_3 + d_9 = 0 \tag{27}$$

where

$$
\begin{aligned}
d_1 &= (a_{11}+a_{15})x_1^2 - 2a_{14}x_1 + (a_{15}-a_{11})\\
d_2 &= -2a_{12}x_1^2 + 4a_{13}x_1 + 2a_{12}\\
d_3 &= (a_{15}-a_{11})x_1^2 + 2a_{14}x_1 + (a_{15}-a_{11})\\
d_4 &= (a_{31}+a_{35})x_3^2 - 2a_{34}x_3 + (a_{35}-a_{31})\\
d_5 &= -2a_{32}x_3^2 + 4a_{33}x_3 + 2a_{32}\\
d_6 &= (a_{35}-a_{31})x_3^2 + 2a_{34}x_3 + (a_{35}-a_{31})\\
d_7 &= (a_{21}+a_{25})x_1^2 - 2a_{24}x_1 + (a_{25}-a_{21})\\
d_8 &= -2a_{22}x_1^2 + 4a_{23}x_1 + 2a_{22}\\
d_9 &= (a_{25}-a_{21})x_1^2 + 2a_{24}x_1 + (a_{25}-a_{21})
\end{aligned}
$$

We now eliminate x_2 from eqs.(25) and (26), using Bezout's method (Salmon, 1964), the resulting equation thus containing only x_1 and x_3, namely,

$$\det\begin{bmatrix}\Delta_{11} & \Delta_{12}\\ \Delta_{21} & \Delta_{11}\end{bmatrix} = 0 \tag{28}$$

where quantities Δ_{11}, Δ_{12} and Δ_{21} are defined below:

$$\Delta_{11} \equiv \det\begin{bmatrix}d_1 & d_3\\ d_4 & d_6\end{bmatrix}, \quad \Delta_{12} \equiv \det\begin{bmatrix}d_5 & d_2\\ d_4 & d_1\end{bmatrix}, \quad \Delta_{21} \equiv \det\begin{bmatrix}d_2 & d_3\\ d_5 & d_6\end{bmatrix}$$

After expansion and simplification, eq.(28) reduces to

$$A_1x_3^4 + A_2x_3^3 + A_3x_3^2 + A_4x_3 + A_5 = 0 \tag{29}$$

where

$$A_i = \sum_{p=0}^{4} A_{ip}x_1^p \tag{30}$$

and the coefficients A_{ip} are constants and depend only on the data. Detailed expressions for A_{ip} are not given here because these expansions would be too large to serve any useful purpose.

Now, x_2 is eliminated from eqs. (25) and (26), while x_3 is eliminated likewise from eqs. (27) and (29), thereby obtaining one single equation in x_1, namely,

$$\det\begin{bmatrix}d_{11} & d_{12} & A_4d_7 & A_5d_7\\ d_{21} & d_{22} & A_4d_8 + A_5d_7 & A_5d_8\\ d_7 & d_8 & d_9 & 0\\ 0 & d_7 & d_8 & d_9\end{bmatrix} = 0 \tag{31}$$

where

$$\begin{aligned} d_{11} &= A_2 d_7 - A_1 d_8 & d_{12} &= A_3 d_7 - A_1 d_9 \\ d_{21} &= A_3 d_7 - A_1 d_9 & d_{22} &= A_3 d_8 - A_2 d_9 + A_4 d_7 \end{aligned}$$

The foregoing determinant is now expanded and simplified, which then leads to

$$\sum_{i=0}^{16} k_i x_1^i = 0 \tag{32}$$

where k_i are constants and depend only on kinematic parameters, and are related by

$$k_i = (-1)^i k_{16-i}, \quad i = 1, \ldots, 7 \tag{33}$$

As with coefficients A_{ip} of eq.(30), detailed expressions for k_i are not given here because these expansions would be too large to serve any useful purpose. What is important to point out here is that the above equation admits 16 solutions, whether real or complex, among which we are interested only in the real positive solutions. The real negative solutions lead to the same configurations of the positive ones with the exception that the sides of the triangle $\mathcal{R}$, d, e and f, are replaced by another triangle with the same vertices $R_1 R_2 R_3$, but different sides, namely, $2\pi - d, 2\pi - e$ and $2\pi - f$. Then, the negative solutions are discarded. The upper bound for the number of real positive solutions of a polynomial is given by Descartes theorem, namely,

The number of real positive solutions of a polynomial is given by the number of change of signs of the coefficients $k_0, k_1, \cdots, k_n$ minus $2m$, where $m \geq 0$.

The maximum number of changes of sign in the foregoing polynomial is eight. Therefore, the problem leads to a maximum of eight real positive solutions and, as a result, triangle $\mathcal{Q}$ of Fig. 1b admits up to eight different orientations, for the specified values of ρ_1, ρ_2 and ρ_3.

3.2. Example

Consider the spherical triangles $\mathcal{P}$ and $\mathcal{Q}$ given as:

$$\begin{aligned} Q_1 Q_2 &= 60^\circ & Q_2 Q_3 &= 70^\circ & Q_3 Q_1 &= 50^\circ \\ P_1 P_2 &= 70^\circ & P_2 P_3 &= 58.6^\circ & P_3 P_1 &= 81.5^\circ \end{aligned}$$

and three points, R_1, R_2 and R_3, located by the three values $\rho_1 = 10^\circ$, $\rho_2 = 49.5^\circ$ and $\rho_3 = 40^\circ$. These values correspond to the angles D_2, E_2 and F_2 given below:

$$D_2 = 43.4745^\circ, \quad E_2 = 37.9120^\circ, \quad F_2 = 106.7287^\circ$$

TABLE I
The sixteen solutions of the example problem

NO.	x_1	D_1 Deg.	E_1 Deg.	F_1 Deg.
1	-3.52853659	$180° + (D_1)_{13}$	$180° + (E_1)_{13}$	$180° + (F_1)_{13}$
2	-1.81493883	$180° + (D_1)_{16}$	$180° + (E_1)_{16}$	$180° + (F_1)_{16}$
3	$-0.7122360 - \jmath 0.9461246$	-	-	-
4	$-0.7122360 + \jmath 0.9461246$	-	-	-
5	$-0.4987636 - \jmath 1.6448662I$	-	-	-
6	$-0.4987636 + \jmath 1.6448662$	-	-	-
7	$-0.0110361 - \jmath 1.7618928$	-	-	-
8	$-0.0110361 + \jmath 1.7618928$	-	-	-
9	$0.00355500 - \jmath 0.5675491$	-	-	-
10	$0.0035550 + \jmath 0.5675491$	-	-	-
11	$0.1688234 - \jmath 0.5567607$	-	-	-
12	$0.1688234 + \jmath 0.5567607$	-	-	-
13	0.28340360	31.64584216	76.17273858	42.53021089
14	$0.5078577 - \jmath 0.6746313$	-	-	-
15	$0.5078577 + \jmath 0.6746313$	-	-	-
16	0.55098275	57.70801252	99.32576667	64.91849185

Equation (32) is solved for x_1. The solutions are shown in Table 1. For this particular problem, we were able to find two real positive solutions. These solutions, which are depicted in Fig. 3b, correspond to the assembly modes of the manipulator.

4. Conclusions

In this paper planar and spherical double-triangular parallel manipulators with three degrees of freedom were introduced. The direct kinematics of the two manipulators have been formulated. We showed that the direct kinematics of the planar mechanism leads to a quadratic polynomial in the tangent of one half of a representative angle. The direct kinematics of the spherical mechanism leads, in turn, to a 16th-order polynomial in the tangent of one half of a representative angle. This result implies that, for a set of three given actuator displacements, the foregoing polynomial admits up to 16 different solutions, among which only eight would be real positive. Since only the real positive solutions are acceptable, the moving triangle Q admits up to eight different orientations. However, in tests we ran, we found that the maximum number of geometrically distinct solutions was only two.

Acknowledgements

The research work reported here was made possible under NSERC (Natural Sciences and Engineering Research Council, of Canada) Grants A4532, A4219, STRGP 205, and EQP00-92729. Support from IRIS, the Institute for Robotics and Intelligent Systems, a network of Canadian centres of excellence, is also acknowledged. The first author was funded by the Ministry of Culture and Higher Education of the Islamic Republic of Iran and the third author completed his contribution while on leave at the Institute B for Mechanics of the Technical University of Munich, under an Alexander von Humboldt Research Award.

References

Craver, W.M.: 1989, *Structural Analysis and Design of a Three-Degree-of-Freedom Robotic Shoulder*, M. Eng. Thesis, The University of Texas at Austin: Austin.

Hunt, K. H.: 1983, 'Structural Kinematics of In-Parallel-Actuated Robot Arms', *ASME J. of Mechanisms, Transmissions, and Automation in Design* **vol. 105, no. 4**, pp .705–712.

Gosselin, C. and Angeles, J.: 1989, 'The Optimum Kinematic Design of Spherical Three-Degree-of-Freedom Parallel Manipulator', *ASME J. of Mechanisms, Transmissions, and Automation in Design* **vol. 111, no. 2**, pp. 202–207.

Gosselin, C. and Angeles, J.: 1990, 'Kinematic Inversion of Parallel Manipulator in the Presence of Incompletely Specified Tasks', *ASME Journal of Mechanical Design* **vol. 112**, pp. 494–500.

Gosselin, C. M. and Sefrioui, J.: 1991, 'Polynomial Solutions for the Direct Kinematic Problem of Planar Three-Degree-of-Freedom Parallel Manipulators', *Proc. of the Fifth Int. Conf. On Advanced Robotics, Pisa*, pp. 1124–1129.

Gosselin, C. M., Sefrioui, J. and Richard, M. J.: 1992a, 'On the Direct Kinematics of General Spherical Three-Degree-of-Freedom Parallel Manipulators', *Proc. of the ASME Mechanisms Conference, Phoenix*, pp. 7–11.

Gosselin, C. M., Sefrioui, J. and Richard, M. J.: 1992b, ' On the Direct Kinematics of a Class of Spherical Three-Degree-of-Freedom Parallel Manipulators', *Proc. of the ASME Mechanisms Conference, Phoenix*, pp. 13–19.

Nanua, P., Waldron, K. J. and Murthy, V.: 1990, 'Direct Kinematic Solution of a Stewart Platform', *IEEE Trans. on Robotics and Automation* **vol. 6, no. 3**, pp. 438–444.

Salmon, G.: 1964, *Lessons Introductory to the Modern Higher Algebra (5th ed.)*, New York: Chelsea.

Stewart, D.: 1965, 'A Platform with Six Degrees of Freedom', *Proc. of the Institution of Mechanical Engineers* **vol. 180, no. 5**, pp. 371–378.

Tarnai, T., and Makai, E.: 1988, 'Physically Inadmissible Motions of a Pair of Tetrahedra', *Proc. of 3rd Int. Conf. Eng. Graph. and Desc. Geom., Vienna* **vol. 2**, pp. 264–271.

Tarnai, T., and Makai, E.: 1989a, 'A Movable Pair of Tetrahedra', *Proc. of R. Soc. Lon.* **A432**, pp. 419–442.

Tarnai, T., and Makai, E.: 1989b, 'Kinematical Indeterminacy of a Pair of Tetrahedral Frames', *Acta Technica Acad. Sci. Hung.* **vol. 102**, pp. 123–145.

Zsombor-Murray, P. J. and Hyder, A.: 1992, 'An Equilateral Tetrahedral Mechanism', *J. of Robotics and Autonomous Systems* **vol. 9**, pp. 227–236.

THE SEMIGRAPHICAL SOLUTION OF THE DIRECT KINEMATICS OF GENERAL PLATFORM-TYPE PARALLEL MANIPULATORS

K. E. ZANGANEH and J. ANGELES
McGill Research Centre for Intelligent Machines &
Department of Mechanical Engineering, McGill University
817 Sherbrooke Street West, Montreal, Canada H3A 2K6
e-mail: etemadi@mcrcim.mcgill.ca and angeles@mcrcim.mcgill.ca

Abstract. A semigraphical method is presented for computing all real direct kinematic solutions of platform-type parallel manipulators with general geometries. The direct kinematic problem is reduced to basically two bivariate equations in the sines and cosines of two unknown angles. One equation is derived by solving an overdetermined system of equations that can be perturbed by different multiples of the least-square error involved in the solutions. Upon perturbing this equation by two different multiples, two distinct equations are obtained. The first bivariate equation and each of these two equations define three contours in the plane of the two angles involved, the intersections of these contours providing all real solutions. The method is used to find all real direct kinematic solutions of a general parallel manipulator of the platform type.

Key words: Direct kinematics – Parallel manipulator – Semigraphical – Symbolic computations

1. Introduction

Platform-type parallel manipulators are multibody mechanical systems comprising a movable platform, henceforth abbreviated as MP, connected to a base platform (BP) by several kinematic subchains leading to an architecture with multiple kinematic loops. The direct kinematic problem of these manipulators is, in general, more challenging than the inverse problem, because of the nonlinearities involved. Hence, all methods presented in the literature for the direct kinematic analysis of general platform-type parallel manipulators are essentially numerical (Parenti-Castelli, 1992). Among these methods, only few aim at finding all possible solutions (Innocenti and Parenti-Castelli, 1992; Raghavan, 1991; Lazard, 1992; Angeles and Zanganeh, 1992).

In this paper we introduce a comprehensive semigraphical method for solving the direct kinematics of general platform manipulators, as compared to the procedure previously presented in (Angeles and Zanganeh, 1992). Within the proposed solution procedure, we derive two basic bivariate equations in the sines and cosines of two unknown angles. One equation is derived by solving an overdetermined system of equations that can be perturbed by different multiples of the least-square error involved in the solutions. Upon perturbing this equation by two different multiples, two distinct equations are obtained. The first bivariate equation and each of the latter two equa-

J. Angeles et al. (eds.), Computational Kinematics, 165–173.

tions define three contours in the plane of the unknown angles, the intersections of the three contours providing all real solutions of the problem.

Figure 1 depicts the general platform manipulator under study. It comprises a kinematic chain with six legs connecting a mobile platform to a base platform. The legs at each end are joined to the platforms by spherical joints, the centers of the latter being, in general, non-coplanar. Moreover, each leg is driven by a prismatic actuator.

The direct kinematic problem (DKP) of the manipulator under study is defined as: *Given the lengths of the six legs, determine the corresponding MP pose, i.e., the orientation and position of the MP.*

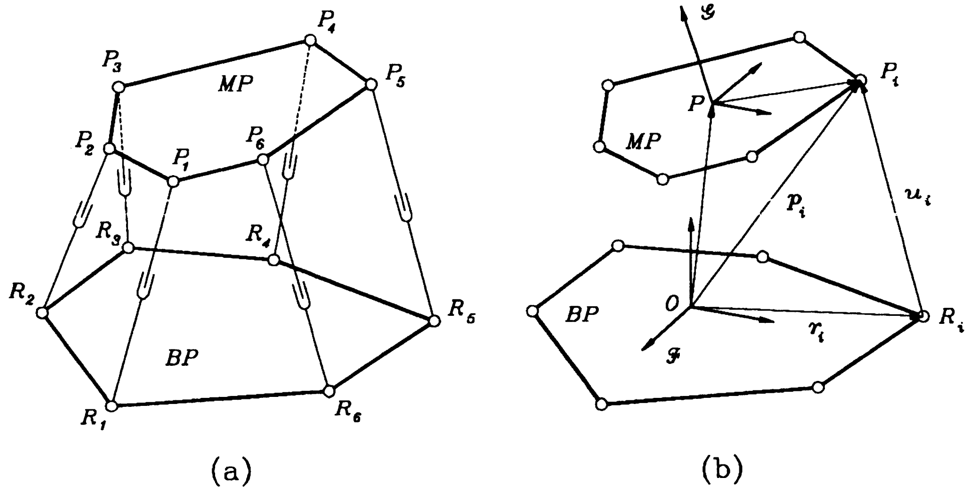


Fig. 1. General platform-type parallel manipulator

2. Kinematic Model

Given the general platform manipulator of Fig. 1, the attachment points of the ith leg on the BP and MP are denoted by R_i and P_i, respectively. Further, we fix a coordinate frame $\mathcal{F}$ to the BP at point O, and a coordinate frame $\mathcal{G}$ to the MP at point P. Moreover, the MP is moved from the home configuration (HC) to the current configuration (CC) with the aid of the six actuators. Hence, if at the HC the vector directed from point P to P_i is denoted by $\boldsymbol{\sigma}_i$, and at the CC by $\boldsymbol{\pi}_i$, then we can write

$$\boldsymbol{\pi}_i = \boldsymbol{Q}\boldsymbol{\sigma}_i; \qquad i = 1, \cdots, 6 \qquad (1)$$

where $\boldsymbol{Q}$ is the rotation matrix relating the orientation of the MP at the HC with that at the CC. Moreover, at the CC we denote the position vectors of the vertices P_i and R_i by $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$, respectively, while vector $\boldsymbol{u}_i$ is defined as directed from R_i to P_i, its magnitude being q_i, for $i = 1, \cdots, 6$, and $\boldsymbol{p}$ is the position vector of P, as shown in Figure 1. Referring to this figure, the geometrical relations below can be readily derived:

$$\|\boldsymbol{u}_i\|^2 = \|\boldsymbol{p}_i - \boldsymbol{r}_i\|^2 \,; \qquad i = 1, \cdots, 6 \tag{2}$$

where $\| \cdot \|$ denotes the Euclidean norm of $(\cdot)$. After expanding the right-hand-side of eq.(2), we obtain

$$\frac{1}{2}\boldsymbol{p}^2 + (\boldsymbol{Q}\boldsymbol{\sigma}_i - \boldsymbol{r}_i)^T\boldsymbol{p} - \boldsymbol{r}_i^T\boldsymbol{Q}\boldsymbol{\sigma}_i + s_i = 0 \,; \qquad i = 1, \cdots, 6 \tag{3}$$

where $s_i \equiv \frac{1}{2}(\boldsymbol{\sigma}_i^2 + \boldsymbol{r}_i^2 - \boldsymbol{u}_i^2)$ and $(\cdot)^T$ represents the transpose of $(\cdot)$. The DKP of the general platform manipulator under study consists of finding all real solutions to the system of six equations in eq.(3).

3. The Contour Equations

For completeness, we recall a part of the elimination procedure introduced first in (Angeles and Zanganeh, 1992). At the outset, we eliminate $\boldsymbol{p}$ by rewriting eq.(3) in linear homogeneous form, namely,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \tag{4}$$

where the 5-dimensional vector $\boldsymbol{x}$ is defined as $\boldsymbol{x} \equiv [\boldsymbol{p}^2 \quad \boldsymbol{p} \quad 1]^T$ and $\boldsymbol{A}$ is the 6×5 matrix shown below:

$$\boldsymbol{A} \equiv \begin{bmatrix} 1/2 & (\boldsymbol{Q}\boldsymbol{\sigma}_1 - \boldsymbol{r}_1)^T & -\boldsymbol{r}_1^T\boldsymbol{Q}\boldsymbol{\sigma}_1 + s_1 \\ \vdots & \vdots & \vdots \\ 1/2 & (\boldsymbol{Q}\boldsymbol{\sigma}_6 - \boldsymbol{r}_6)^T & -\boldsymbol{r}_6^T\boldsymbol{Q}\boldsymbol{\sigma}_6 + s_6 \end{bmatrix} \tag{5}$$

Now, for eq.(4) to admit a nontrivial solution, all determinants of the 5×5 submatrices of $\boldsymbol{A}$ should vanish. This results in a set of six nonlinear equations in $\boldsymbol{Q}$. However, only five of them are independent. If we denote the determinant of the ith submatrix by Δ_i, then

$$\Delta_i \equiv \sum_{\substack{j=1 \\ j \neq i}}^{6} (-1)^{i+j}\eta_j\Delta_{ij} = 0 \,; \qquad i = 1, \cdots, 6 \tag{6}$$

where $\eta_j \equiv \boldsymbol{r}_j^T\boldsymbol{Q}\boldsymbol{\sigma}_j - s_j$, $\Delta_{ij} \equiv \det(\boldsymbol{A}_{ij})$ and $\boldsymbol{A}_{ij}$ is the 4×4 submatrix of $\boldsymbol{A}$ obtained by deleting the ith and jth rows and the last column of $\boldsymbol{A}$. The expansion of Δ_{ij} yields in turn an expression of the form

$$\Delta_{ij} = \frac{1}{2} \sum_{\substack{k=1 \\ k \neq i,\, k \neq j}}^{6} (-1)^{i+j+k}\Delta_{ijk} \tag{7}$$

where $\Delta_{ijk} \equiv \det(\boldsymbol{A}_{ijk})$ and the 3×3 matrix $\boldsymbol{A}_{ijk}$ is also a submatrix of $\boldsymbol{A}$, namely, that obtained by deleting the first and last columns together with the ith, jth and kth rows of $\boldsymbol{A}$. Using the above notation, some symmetry properties can be exploited among the foregoing subdeterminants that ease their computations. Moreover, every rotation matrix $\boldsymbol{Q}$ can be written in the form

$$\boldsymbol{Q} = [\boldsymbol{m} \quad \boldsymbol{n} \quad \boldsymbol{m} \times \boldsymbol{n}] \tag{8}$$

subjected to the conditions

$$\boldsymbol{m}^T\boldsymbol{n} = 0 \tag{9a}$$

$$\boldsymbol{m}^T\boldsymbol{m} = 1 \tag{9b}$$

$$\boldsymbol{n}^T\boldsymbol{n} = 1 \tag{9c}$$

where, $\boldsymbol{m} \equiv [m_1,\ m_2,\ m_3]^T$ and $\boldsymbol{n} \equiv [n_1,\ n_2,\ n_3]^T$. Now, in order to derive the bivariate equations of the contours, we have to eliminate four of the unknowns among the six unknown components of $\boldsymbol{m}$ and $\boldsymbol{n}$. To do so, we first use eq.(9a) to eliminate one of the components of $\boldsymbol{m}$. Without loss of generality, we can eliminate the third component, i.e., m_3, provided that $n_3 \neq 0$. Thus, we have:

$$m_3 = -\frac{1}{n_3}(n_1 m_1 + n_2 m_2) \tag{10}$$

Upon substitution of the above equation into eq.(9b) and multiplying the equation thus resulting by n_3^2 to clear denominators, we obtain:

$$(n_1^2 + n_3^2)m_1^2 + (n_2^2 + n_3^2)m_2^2 + 2n_1 n_2 m_1 m_2 - n_3^2 = 0 \tag{11}$$

Moreover, the same substitution in eq.(6) yields a set of six nonlinear equations free of m_3, only four of which are independent. These equations can be cast in the form

$$\boldsymbol{N}'\boldsymbol{y}' = \boldsymbol{0} \tag{12}$$

where all entries of the 6×6 matrix $\boldsymbol{N}'$ are functions of vector $\boldsymbol{n}$ only, while $\boldsymbol{y}' \equiv [\,m_1^2,\ m_2^2,\ m_1 m_2,\ m_1,\ m_2,\ 1\,]^T$. Furthermore, from eq.(11) we can derive an expression for m_1^2, i.e.,

$$m_1^2 = \frac{1}{n_1^2 + n_3^2}[(n_2^2 + n_3^2)m_2^2 + 2n_1 n_2 m_1 m_2 - n_3^2] \tag{13}$$

Now, we substitute eq.(13) into eq.(12) and then pick up four independent equations among the six equations thus obtained. This yields a system of four homogeneous equations in the form

$$\boldsymbol{N}\boldsymbol{y} = \boldsymbol{0} \tag{14}$$

where all entries of the 4×5 matrix $\boldsymbol{N}$ are functions of vector $\boldsymbol{n}$ only, while $\boldsymbol{y} \equiv [\, m_2^2, m_1 m_2, m_1, m_2, 1\,]^T$. Moreover, $\boldsymbol{N}$ is of full rank, and hence, a 5-dimensional nonzero vector $\boldsymbol{\zeta}$ exists that spans the nullspace of $\boldsymbol{N}$. Let $\boldsymbol{\zeta}$ have the components below:

$$\boldsymbol{\zeta} \equiv [\, \zeta_1 \quad \zeta_2 \quad \dots \quad \zeta_5 \,]^T \tag{15}$$

Comparing the components of $\boldsymbol{\zeta}$ with those of $\boldsymbol{y}$ defined above, we notice that ζ_5 should be different from zero, for the fifth component of $\boldsymbol{y}$ is. Hence, we can normalize $\boldsymbol{\zeta}$ by dividing all its components by nonzero ζ_5. Moreover, the 3rd and 4th components of $\boldsymbol{y}$ are nothing but m_1 and m_2, and so, the latter are equal to the 3rd and 4th components of the normalized $\boldsymbol{\zeta}$, namely,

$$m_1 = \zeta_3/\zeta_5, \qquad m_2 = \zeta_4/\zeta_5 \tag{16}$$

Moreover, if eq.(16) is substituted into eq.(10) we obtain an expression for m_3 in terms of $\boldsymbol{n}$ only, i.e.,

$$m_3 = -\frac{1}{n_3\zeta_5}(n_1\zeta_3 + n_2\zeta_4) \tag{17}$$

Since $\boldsymbol{m}$ is of unit magnitude, we can use eqs.(16 & 17) in order to derive

$$\zeta_3^2 + \zeta_4^2 - \zeta_5^2 + \frac{1}{n_3^2}(n_1\zeta_3 + n_2\zeta_4)^2 = 0 \tag{18}$$

We now express the components of vector $\boldsymbol{n}$ in spherical coordinates in order to reduce the number of unknowns to only two, i.e.,

$$\boldsymbol{n} \equiv [\, \sin\theta_1 \cos\theta_2 \quad \sin\theta_1 \sin\theta_2 \quad \cos\theta_1 \,]^T \tag{19}$$

Upon substitution of eq.(19) into eqs.(18), a bivariate equations in θ_1 and θ_2 is derived, namely,

$$f_1(\theta_1, \theta_2) \equiv \zeta_3^2 + \zeta_4^2 - \zeta_5^2 + \frac{1}{n_3^2}(n_1\zeta_3 + n_2\zeta_4)^2 = 0 \tag{20}$$

Function f_1 defines a contour C_1 in the θ_1-θ_2 plane, which represents the locus of all possible solutions, actual and spurious. However, the actual solutions of the problem are only those which satisfy the equation:

$$p^2 - \boldsymbol{p}^T\boldsymbol{p} = 0 \tag{21}$$

If we now substitute eqs.(16, 17 & 19) into eq.(4), all entries of the coefficient matrix $\boldsymbol{A}$ become functions of θ_1 and θ_2 only. The equation thus resulting can be written in the form

$$\boldsymbol{B}\tilde{\boldsymbol{x}} = \boldsymbol{b} \tag{22}$$

where $\tilde{\boldsymbol{x}} \equiv [\boldsymbol{p}^2, \boldsymbol{p}]^T$ and the entries of the 6×4 matrix $\boldsymbol{B}$ and the components of the 6-dimensional vector $\boldsymbol{b}$ are functions of θ_1 and θ_2 only. Hence, we can solve eq.(22) for $\boldsymbol{p}^2$ and $\boldsymbol{p}$ in terms of the two angles involved. However, this is an overdetermined system of six equations in four unknowns. The least-square solution to this system can readily be found by applying Householder reflections (Golub and Van Loan, 1989) to both sides, thereby deriving

$$\begin{bmatrix} \boldsymbol{R} \\ \mathbf{O} \end{bmatrix} \tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} \tag{23}$$

where $\boldsymbol{R}$ is a 4×4 upper triangular matrix, $\mathbf{O}$ is the 2×4 zero matrix and $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are 4- and 2-dimensional vectors, respectively. Hence, the solution is obtained by back substitution into $\boldsymbol{R}\tilde{\boldsymbol{x}} = \boldsymbol{b}_1$, which yields four relations for $\boldsymbol{p}^2$ and the components of $\boldsymbol{p}$ in terms of θ_1 and θ_2. On the other hand, the least-square error to eq.(22) is equal to $||\boldsymbol{b}_2||$. Thus, the foregoing four relations, when substituted into eq.(21) and added with a multiple of $\boldsymbol{b}_2^T\boldsymbol{b}_2$, result into a single equation in θ_1 and θ_2, namely,

$$f_{i+1}(\theta_1, \theta_2) \equiv \boldsymbol{p}^2 - \boldsymbol{p}^T\boldsymbol{p} + \alpha_i \boldsymbol{b}_2^T\boldsymbol{b}_2 = 0\,; \qquad i = 1, 2, \ldots \tag{24}$$

where α_i is a nonzero constant that is meant to perturb the equation. For two distinct values of α_i, i.e., α_1 and α_2, we obtain two equations that, in turn, lead to functions f_2 and f_3, respectively. These functions define two more contours, C_2 and C_3, in the θ_1-θ_2 plane. Thus, the intersection points of the three contours C_1, C_2 and C_3, when superimposed, yield all real solutions of the problem.

4. Numerical Example

By means of the above procedure, the contour intersections and the position and orientation of a general platform-type parallel manipulator were obtained, for given values of the actuator variables $\{q_i\}_1^6$. The parameters of this manipulator are given as

$$\boldsymbol{r}_1 = \begin{bmatrix} 0.06503501 \\ -1.42475422 \\ 1.93172294 \end{bmatrix}, \quad \boldsymbol{r}_2 = \begin{bmatrix} -0.50508753 \\ -0.0289200 \\ -0.04608834 \end{bmatrix}, \quad \boldsymbol{r}_3 = \begin{bmatrix} -1.09491488 \\ 0.22313340 \\ 1.93280714 \end{bmatrix},$$

$$\boldsymbol{r}_4 = \begin{bmatrix} 0.00048050 \\ 0.32929144 \\ -0.05044410 \end{bmatrix}, \quad \boldsymbol{r}_5 = \begin{bmatrix} 0.75279702 \\ 0.81007514 \\ 1.82952667 \end{bmatrix}, \quad \boldsymbol{r}_6 = \begin{bmatrix} 0.11011463 \\ -0.32291556 \\ 0.01200174 \end{bmatrix}$$

$$\boldsymbol{\sigma}_1 = \begin{bmatrix} 0.01346965 \\ -1.65009147 \\ 0.77441816 \end{bmatrix}, \quad \boldsymbol{\sigma}_2 = \begin{bmatrix} -1.04206133 \\ -0.59772236 \\ 0.02532155 \end{bmatrix}, \quad \boldsymbol{\sigma}_3 = \begin{bmatrix} -1.73287180 \\ 1.51458094 \\ 0.95305524 \end{bmatrix},$$

$$\boldsymbol{\sigma}_4 = \begin{bmatrix} -0.00177893 \\ 1.22775426 \\ -0.00914577 \end{bmatrix}, \quad \boldsymbol{\sigma}_5 = \begin{bmatrix} 5.07444276 \\ 0.77126528 \\ 0.90334291 \end{bmatrix}, \quad \boldsymbol{\sigma}_6 = \begin{bmatrix} 1.11289374 \\ -0.62965332 \\ 0.00374341 \end{bmatrix}$$

$$q_1 = 0.26321981, \qquad q_2 = 1.52167226, \qquad q_3 = 1.49376759,$$
$$q_4 = 1.53376882, \qquad q_5 = 4.25152610, \qquad q_6 = 1.53758131$$

For the values of $\alpha_1 = -1$ and $\alpha_2 = -3$, the superimposed contours are shown in Figures 2, in which $\mathcal{C}_1$ is shown with dashed line. Moreover, in Fig. 3 we magnified the region where the intersection points exist. From these figures two solutions are found as recorded in Table 1.

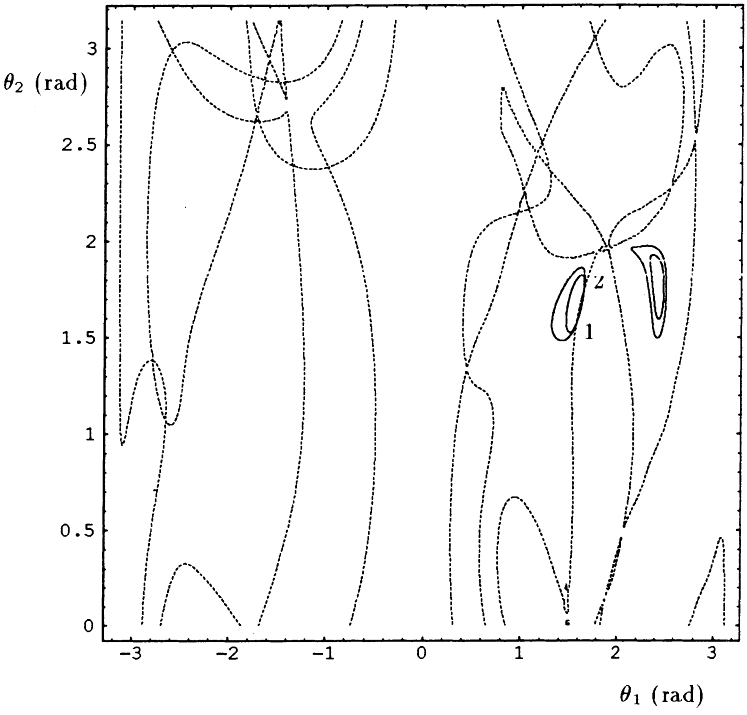


Fig. 2. $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$ contours of the numerical example

It should be noted that the contours have been plotted for all values of θ_2 between zero and π. We have done this because, for each pair of (θ_1, θ_2) values in Table 1, we can always find a complementary pair (θ_1', θ_2') that satisfies the following identity:

$$\boldsymbol{n} \equiv \begin{bmatrix} \sin\theta_1 \cos\theta_2 \\ \sin\theta_1 \sin\theta_2 \\ \cos\theta_1 \end{bmatrix} = \begin{bmatrix} \sin\theta_1' \cos\theta_2' \\ \sin\theta_1' \sin\theta_2' \\ \cos\theta_1' \end{bmatrix}$$

where, $\theta_1' \equiv -\theta_1$ and $\theta_2' \equiv \theta_2 - \pi$.

TABLE I
Direct kinematic solutions

point	1	2
θ_1 (rad)	1.570796	1.643131
θ_2 (rad)	1.570796	1.742817
m_1	1.0	0.983827
m_2	0.0	0.165975
m_3	0.0	-0.067350
p_x	-0.0790339	-0.034122
p_y	0.0005794	0.158842
p_z	1.198686	1.152051

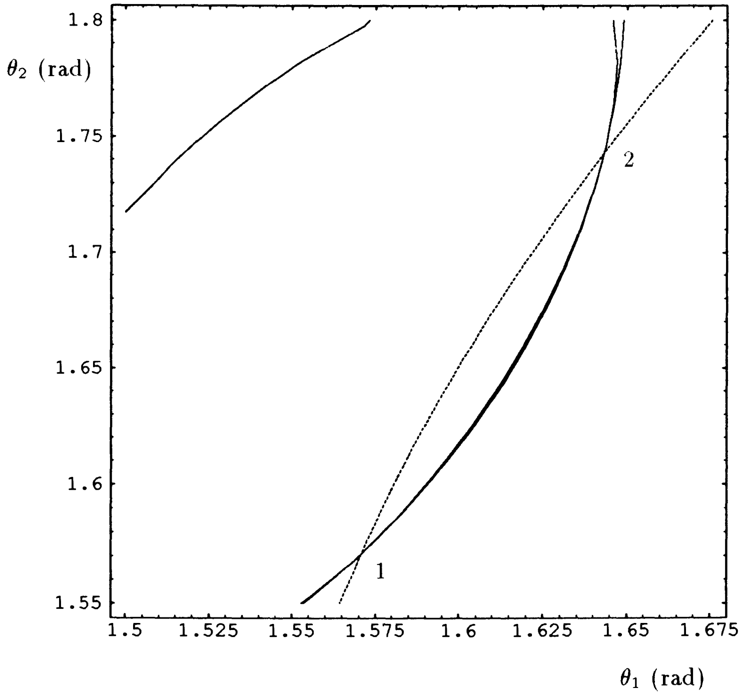


Fig. 3. Magnifcation of the contours of Fig. 2

The architecture of the manipulator of this example was chosen so as to yield an isotropic Jacobian matrix. Isotropic Jacobian matrices with dimensionally inhomogeneous entries, as in the case at hand, were defined using the concept of characteristic length in (Angeles et al., 1992).

5. Conclusions

In this paper, a mathematical model for the direct kinematic problem of platform manipulators with general geometries was derived. Then, the equations were reduced to two bivariate equations. By using the first equation and perturbing the second equation, we found three contours in the plane of the two unknown angles involved. The superimposed plots of contours yield all real solutions. A numerical example was included that admits two real solutions.

Acknowledgements

The research work reported here was made possible under NSERC (Natural Sciences and Engineering Research Council of Canada) Grants A4532, STRGP 205 and EQP00-92729. Support from IRIS, the Institute for Robotics and Intelligent Systems, a network of Canadian centers of excellence, is also acknowledged.

References

Angeles, J., Ranjbaran, F., and Patel, R. V.: 1992, 'On the Design of the Kinematic Structure of Seven-Axis Redundant Manipulators for Maximum Conditioning', *Proc. IEEE Int. Conf. on Robotics and Automation*, Nice, May 12–14, pp. 494–499.

Angeles, J. and Zanganeh, K. E.: 1992, 'The Semigraphical Solution of the Direct Kinematics of General Platform Manipulators', *Proc. 4th Int. Symp. on Robotics and Manufacturing*, Santa Fe, Nov. 11–13, pp. 45–52.

Golub, G. H. and Van Loan, C.: 1989, *Matrix Computations*, Johns Hopkins: Baltimore.

Innocenti, C., and Parenti-Castelli, V.: 1993, 'Forward Kinematics of the general 6-6 Fully-Parallel Mechanism: An Exhaustive Numerical Approach Via a Mono-Dimensional-Search Algorithm', to appear in *ASME, J. Mechanical Design.*

Lazard, D.: 1992, 'Stewart Platforms and Gröbner Basis', *Proc. 3rd Int. Workshop on Advances in Robot Kinematics*, Ferrara, Sept. 7–9, pp. 136–142.

Parenti-Castelli, V.: 1992, 'Recent Techniques for Direct Position Analysis of the Generalized Stewart Platform Mechanism', *Proc. 3rd Int. Workshop on Advances in Robot Kinematics*, Ferrara, Sept. 7–9, pp. 129–135.

Raghavan, M.: 1991, 'The Stewart Platform of General Geometry Has 40 Configurations' in Gabriele, G. A., ed., *Advances in Design Automation*, ASME Press: New York, pp. 397–402.

ON THE REPRESENTATION OF RIGID-BODY MOTIONS AND ITS APPLICATION TO GENERALIZED PLATFORM MANIPULATORS*

D. LAZARD**
LITP, Institut Blaise Pascal, boite 168,
4 place Jussieu, 75252 Paris Cedex 05, France

Abstract. Different ways for representing rigid-body motions (direct isometries) in a computer are presented. It appears that the choice between them may have a dramatic effect on the difficulty of a computation or of a proof.

As an application, a computational proof is given of the fact that the direct kinematic problem for a generalized Stewart platform has at most 40 complex solutions.

1. Introduction

A *generalized Stewart platform* is a body with 6 legs. The legs are fixed on the body and on the ground, which is not necessary plane. The geometry of the platform is given, as well as the points where the legs are connected with the platform and with the ground. The position of the platform is commanded by modifying the lengths of the legs.

The problem studied in this paper consists in computing the position of the platform from its geometry and the lengths of the legs (direct kinematic).

If an initial position of the platform is defined, this problem is equivalent to compute a *rigid-body motion* (i.e. a direct isometry) satisfying some constraints, i.e. to solve an equation, the unknown of which is a rigid-body motion.

The variety of rigid-body motions has dimension 6; there are many ways for representing it with scalar variables submitted to some constraints. It appears that the complexity of solving an equation with a rigid-body motion as unknown heavily depends on the representation chosen.

In Section 2, we describe some representations that have been tried, and we discuss their interest for solving the problem, which is the main result of this paper: Prove that the number of complex solutions of the direct problem for the generalized Stewart platform is at most 40 or infinite.

We first proved this result for the case of planar platforms (Lazard, 1992). Then Ronga and Vust proved by hand the general case, using intersection theory, blowing-up and Chern classes. Having heard the result by J.J. Risler, we figured out their proof before reading their paper, which led us to the proof given below, that is conceptually very simple, but needs machine com-

* Work supported by EEC projects POSSO and PROMOTION.
** E-mail: lazard@litp.ibp.fr

J. Angeles et al. (eds.), Computational Kinematics, 175–181.

putations. Ronga and Vust (1992) were, in turn, influenced by this and devised a much simpler proof.

At the same time, B. Mourrain devised a different proof of the same result, as yet to appear.

As rigid-body motions appear in many applied geometrical problems, especially in robotics, we hope that the considerations developed in this paper will be useful in a much more general context than the one of generalized Stewart platforms, which is for us a kind of paradigm for problems of spatial geometry.

2. How to Represent Rigid-Body Motions?

A rigid-body motion is the product of a translation by a rotation. Many authors use Euler angles for representing rotations. For this reason, angles and trigonometric functions appear in many papers related to generalized Stewart platforms. Angles are convenient if floating point computations are used, but are very difficult to handle in symbolic or exact computations. For this reason, we do not consider here the representations that involve angles explicitly.

A standard way for eliminating angles consists in using the formulas

$$\sin(\theta) = \frac{2t}{1+t^2}; \quad \cos(\theta) = \frac{1-t^2}{1+t^2} \tag{1}$$

As three angles are needed for a spatial rotation, such a transformation leads to complicated formulas of a rather high degree. For this reason, we do not consider further this approach.

2.1. Representation by Images of Points

In (Lazard, 1992), we have represented a rigid-body motion by the images of three points of the platform (nine coordinates), submitted to the constraint that their three respective distances are given. The other points of the platform are defined by their coordinates in the local coordinate system defined by the first three points. For the direct kinematic problem we are considering, this leads, when the platform is planar, to a system of nine equations, six of them being quadratic, the remaining ones being linear.

Thus, in this case, the variety of rigid-body motions (of dimension six) is represented as imbedded in $\mathbf{R}^9$. Bezout's theorem, applied to the system of equations to be solved, gives a bound of 64 for the number of solutions. A study of the "solutions at infinity" (i.e. the solutions obtained in homogenizing the equations and looking at the solutions for which the homogenizing variable is 0) shows that they are of dimension 0 and of degree 24. Thus, in the planar case, the number of solutions is at most 40, if they form a set of dimension 0.

If the platform is not planar, for representing a point of it in the coordinate system defined by three points, we need to use vector products of the vectors defined by these three points. After some reductions, this leads to a system with eight quadratic equations and one cubic one. It is clearly much more difficult to solve. It appears also that the solutions at infinity now form an algebraic set of positive dimension. It follows that Bezout's theorem does not give a sharp bound. *This is the reason for which the sharp bound of 40 was so difficult to prove.*

Another representation, also introduced in (Lazard, 1992) consists in defining a rigid-body motion by the images of four points subjected to six constraints (the values of their respective distances). This representation codes not only rigid-body motions, but also skew isometries (product of a rigid-body motion by a reflection). For the Stewart platform problem, this leads to ten quadratic and two linear equations. The best general bound obtained in (Lazard, 1992) follows from this representation.

2.2. Quaternions

In previous representations, rigid-body motions do not really appear explicitly. We consider now representations where rigid-body motions appear as explicit objects.

The representation by quaternions has the advantage to be rational, as the expressions appearing in eq.(1) are, and to introduce less denominators than if the identities of eq.(1) were used.

Let us recall that the quaternions are a skew field which is a real vector field of dimension 4 with $(1, i, j, k)$ as a basis. The multiplication table may easily be deduced from

$$i^2 = j^2 = k^2 = -1 \quad ijk = -kji = -1.$$

The *pure quaternions* are those that are in the subspace generated by (i, j, k). A quaternion q acts on the vector space of pure quaternions by $v \rightarrow qvq^{-1}$. This action is a rotation of this real vector space that defines an isomorphism between the real projective space of dimension 3 (quaternions up to the multiplication by a real number) and the space of rotations.

A rigid-body motion is the product of a rotation represented by a quaternion $r = r_0 + r_1 i + r_2 j + r_3 k$ and a translation of vector $t = t_1 i + t_2 j + t_3 k$. The image produced by this motion of a point of coordinates x_1, x_2, x_3 has, for coordinates, the sum of a t_i and of a fraction with numerator homogeneous of degree 2 in r_0, r_1, r_2, r_3 and denominator $r_0^2 + r_1^2 + r_2^2 + r_3^2$.

Using this, the direct kinematic problem for generalized Stewart platform leads to an equation of degree 2 in the t_i and five equations that are homogeneous of degree 2 in the r_i and linear in the t_i.

There are several ways for applying Bezout's theorem to such a system. The standard ones give a bound of 486. The multi-homogeneous Bezout theorem (Shafarevich, 1974) gives a better bound of 160.

Thus, this approach leads to systems of higher degree, but with only six equations. The fact that the equations are homogeneous with respect to a part of the variables may be useful in some situations.

2.3. Representation with orthogonal matrices

The most obvious representation of the rotations consists in using orthogonal matrices, i.e. matrices such that $R \cdot R^t = I$ and $\det(R) = 1$. This makes the rotations a subvariety of $\mathbf{R}^9$, of dimension 3. This means that rigid-body motions are represented by twelve parameters submitted to six conditions (in fact, seven). As computational problems grow exponentially with the number of variables, we have preferred the preceding representation.

Let us examine further the equations that arise here.

The condition $R \cdot R^t = I$ means that the three row vectors of the matrix R have norm 1 and that the three dot products between rows are zero. These relations and $\det(R) = 1$ generate many other conditions, among them are the similar conditions on columns ($R \cdot R^t = I$) and $R = \text{cofactor}(R)$, where cofactor(R) is, as usually, defined by $\det(R) \cdot [\text{cofactor}(R)]^t = R^{-1}$.

It appears that the 21 equations $R \cdot R^t = I, R \cdot R^t = I$ and $R = \text{cofactor}(R)$ give a Gröbner basis of the ideal generated by $R \cdot R^t = I$ and $\det(R) = 1$, for the degree ordering. This basis is neither minimal nor reduced, but a reduced basis is easily obtained by replacing the equations for the norms of the first column and the first row by their common normal form with respect to the other equations. The resulting Gröbner basis appears as a subset of the Gröbner basis given below.

Using this Gröbner basis, the Hilbert function is easy to compute, showing that the rotations are a variety of dimension 3 and degree 8 in $\mathbf{R}^9$.

With this representation, the distance between a point A (represented by its position vector) and the image of a point M by the rotation R followed by the translation of vector T is given by the norm of $R.M + T - A$.

For the Stewart platform problem the data are the coordinates of six pairs of points M and A such that the first M and the first A may be chosen at the origin, as well as the six lengths of the legs. This leads to the equations

$$\begin{aligned} \|T\| &= l_1^2 \\ T^t \cdot R \cdot M_i - A_i^t \cdot R \cdot M_i - A_i^t \cdot T &= \text{constant} \qquad \text{for } i = 2..6, \end{aligned} \tag{2}$$

where the unknowns are the coefficients of the square matrix R and the column matrix T. This means that we intersect the variety of degree 8 of the rigid motions by six quadratic hypersurfaces. But the vector space generated by the quadratic parts of the five last equations is of dimension at most 3 (the dimension of the space generated by the M_i). Thus, two of these equations reduce to linear ones, and Bezout's theorem gives a bound of $8 * 2^3 = 64$ for our problem.

3. Bounding the Number of Solutions for the Generalized Stewart Platform Problem

In the above considerations, we have viewed a rigid-body motion as a rotation of matrix R followed by a translation of vector T. It may also be viewed as another translation U followed by the same rotation T. The two representations are related by formulas

$$T = R \cdot U \qquad U = R^t \cdot T, \tag{3}$$

where the nonlinear part of preceding equations appears.

This leads us to introduce a new representation of rigid-body motions as a subvariety of $\mathbf{R}^{15}$, where the coordinates are the coefficients of the matrix R and of both vectors T and U. The equations of this variety are those of the rotations and eq.(3). Surprisingly, the Gröbner basis of this ideal is very simple and rather easy to compute (note that the twenty polynomials not depending on t_i and u_i, which appear in the middle, are a Gröbner basis for the variety of rotations:

$$\begin{array}{l}
r_{2,1}r_{2,2}u_1 + r_{3,1}r_{3,2}u_1 - r_{1,1}r_{2,2}u_2 - r_{1,1}r_{3,2}u_3 - r_{1,1}t_2, \\
r_{2,2}^2u_1 + r_{3,2}^2u_1 - r_{1,2}r_{2,2}u_2 - r_{1,2}r_{3,2}u_3 - r_{1,2}t_2 - u_1, \\
r_{2,1}r_{2,3}u_1 + r_{3,1}r_{3,3}u_1 - r_{1,1}r_{2,3}u_2 - r_{1,1}r_{3,3}u_3 - r_{1,1}t_3, \\
r_{2,2}r_{2,3}u_1 + r_{3,2}r_{3,3}u_1 - r_{1,2}r_{2,3}u_2 - r_{1,2}r_{3,3}u_3 - r_{1,2}t_3, \\
r_{2,3}^2u_1 + r_{3,3}^2u_1 - r_{1,3}r_{2,3}u_2 - r_{1,3}r_{3,3}u_3 - r_{1,3}t_3 - u_1,
\end{array}$$

$$\begin{array}{l}
r_{1,1}^2 - r_{2,2}^2 - r_{2,3}^2 - r_{3,2}^2 - r_{3,3}^2 + 1, \\
r_{1,1}r_{1,2} + r_{2,1}r_{2,2} + r_{3,1}r_{3,2}, \\
r_{1,2}^2 + r_{2,2}^2 + r_{3,2}^2 - 1, \\
r_{1,1}r_{1,3} + r_{2,1}r_{2,3} + r_{3,1}r_{3,3}, \\
r_{1,2}r_{1,3} + r_{2,2}r_{2,3} + r_{3,2}r_{3,3}, \\
r_{1,3}^2 + r_{2,3}^2 + r_{3,3}^2 - 1, \\
r_{1,1}r_{2,1} + r_{1,2}r_{2,2} + r_{1,3}r_{2,3}, \\
r_{1,2}r_{2,1} - r_{1,1}r_{2,2} + r_{3,3}, \\
r_{1,3}r_{2,1} - r_{1,1}r_{2,3} - r_{3,2}, \\
r_{2,1}^2 + r_{2,2}^2 + r_{2,3}^2 - 1, \\
r_{1,3}r_{2,2} - r_{1,2}r_{2,3} + r_{3,1}, \\
r_{1,1}r_{3,1} + r_{1,2}r_{3,2} + r_{1,3}r_{3,3}, \\
r_{1,2}r_{3,1} - r_{1,1}r_{3,2} - r_{2,3}, \\
r_{1,3}r_{3,1} - r_{1,1}r_{3,3} + r_{2,2}, \\
r_{2,1}r_{3,1} + r_{2,2}r_{3,2} + r_{2,3}r_{3,3}, \\
r_{2,2}r_{3,1} - r_{2,1}r_{3,2} + r_{1,3}, \\
r_{2,3}r_{3,1} - r_{2,1}r_{3,3} - r_{1,2}, \\
r_{3,1}^2 + r_{3,2}^2 + r_{3,3}^2 - 1,
\end{array}$$

$$
\begin{aligned}
&r_{1,3}r_{3,2} - r_{1,2}r_{3,3} - r_{2,1},\\
&r_{2,3}r_{3,2} - r_{2,2}r_{3,3} + r_{1,1},\\
\\
&r_{1,1}t_1 + r_{1,2}t_2 + r_{1,3}t_3 + u_1,\\
&r_{1,2}t_1 - r_{1,1}t_2 - r_{3,3}u_2 + r_{2,3}u_3,\\
&r_{1,3}t_1 - r_{1,1}t_3 + r_{3,2}u_2 - r_{2,2}u_3,\\
&r_{2,1}t_1 + r_{2,2}t_2 + r_{2,3}t_3 + u_2,\\
&r_{2,2}t_1 - r_{2,1}t_2 + r_{3,3}u_1 - r_{1,3}u_3,\\
&r_{2,3}t_1 - r_{2,1}t_3 - r_{3,2}u_1 + r_{1,2}u_3,\\
&r_{3,1}t_1 + r_{3,2}t_2 + r_{3,3}t_3 + u_3,\\
&r_{3,2}t_1 - r_{3,1}t_2 - r_{2,3}u_1 + r_{1,3}u_2,\\
&r_{3,3}t_1 - r_{3,1}t_3 + r_{2,2}u_1 - r_{1,2}u_2,\\
&t_1^2 + t_2^2 + t_3^2 - u_1^2 - u_2^2 - u_3^2,\\
&r_{1,3}t_2 - r_{1,2}t_3 - r_{3,1}u_2 + r_{2,1}u_3,\\
&r_{2,3}t_2 - r_{2,2}t_3 + r_{3,1}u_1 - r_{1,1}u_3,\\
&r_{3,3}t_2 - r_{3,2}t_3 - r_{2,1}u_1 + r_{1,1}u_2,\\
&r_{1,1}u_1 + r_{2,1}u_2 + r_{3,1}u_3 + t_1,\\
&r_{1,2}u_1 + r_{2,2}u_2 + r_{3,2}u_3 + t_2,\\
&r_{1,3}u_1 + r_{2,3}u_2 + r_{3,3}u_3 + t_3
\end{aligned}
$$

This Gröbner basis computation took 3h30' with Maple (Char *et al.*, 1985, 1988), 4' with Axiom (Jenks and Sutor, 1992) and less than 3" with GB, an experimental software written by J.C. Faugère (on SUN–Sparc 2 station or, for Axiom, IBM–RISC 6000). Computing the Hilbert function of this ideal, for example with Macaulay (Bayer and Stillman, 1983, 1990), it is easy to see that it is of dimension 6, as expected, and of degree 20.

Now, by replacing$T^t \cdot R$ by U^t in equations (2), they become linear, except for the first one, which is unchanged and remains of degree 2. Thus, Bezout's theorem asserts that, if the set of solutions is of dimension 0, it has a degree of $2 \cdot 20 = 40$.

Clearly, for special configurations, some of these 40 points may be at infinity, and the number of actual solutions may be smaller. We have also to remark that the version of Bezout's theorem given by Heintz (1983) bounds also the number of isolated solutions, even if there are components of positive dimension at infinity. Thus, we have proven:

THEOREM 1. *If the direct kinematic problem for generalized Stewart platform has a finite number of complex solutions, this number is at most 40.*

4. Conclusions

We have shown that the choice of the equations for describing rigid-body motions may have a dramatic effect on the difficulty of a problem.

Clearly, for the generalized Stewart platform, the main problem is to effectively compute the solutions. We have the hope that the Gröbner basis method could be an efficient and robust way for this purpose. Unfortunately, floating point computations are not convenient, because of the number of equality tests which are needed. Thus, it would be better to compute with exact integers or rational numbers.

This is possible, but difficult because of the size of the integers that appear during the computation, and even in the final output. We have done some experiments with randomly chosen configurations. For a planar platform modelled as in subsection 2.1, the Gröbner basis is computed in 8 minutes by GB on a Sparc 10 *without the final verification that all S-polynomials reduce to 0.* Another example (not planar) has been run using the modelling of Section 3, starting from the Gröbner basis given there. Despite the fact that the equations which are added to this Gröbner basis are one very simple quadratic one and five linear ones, the computation took 4 hours (without the final verification) and the result needs 5Mbytes for storing the integers of several thousand of digits which appear in the result.

Thus, the best way for modelling this problem in order to efficiently compute the position of the platform requires further study. Clearly, execution depends on the solution algorithm that is used, and the best choice may by completely different for floating point algorithms and for exact computations like Gröbner bases.

References

Bayer, D. and Stillman, M.: 1983, 1990, *Macaulay: A system for computation in algebraic geometry and commutative algebra.*, Manual, source and object code for Unix and Macintosh computers available via anonymous ftp from `zariski.harvard.edu`.

Char, B.W. *et al.*: 1985, 1988, *MAPLE Reference Manual.*, Watcom Publ.: Waterloo, Canada.

Heintz, J.: 1983, 'Definability and Fast Quantifier Elimination in Algebraic Closed Fields' *Theoretical Computer Science* **24**, 239–277.

Jenks, R.D. and Sutor, R.S.: 1992, *AXIOM, The Scientific Computation System*, Springer-Verlag: New York.

Lazard, D.; 1992, 'Stewart Platforms and Gröbner Basis', in Parenti-Castelli, V. and Lenarčič, J., eds, *3rd International Workshop on Advances in Robot Kinematic (3ARK, Ferrare, Italy)*, V. Parenti-Castelli: Ferrare, 136–142.

Merlet, J.P.: 1990, *Les robots parallèles*, Hermès: Paris.

Ronga F. and Vust, Th.: 1992, *Stewart platforms without computer?*, Preprint, Université de Genève.

Shafarevich, I.R.: 1974, *Basic Algebraic Geometry*, Springer: Berlin.

ALGEBRAIC-GEOMETRY TOOLS FOR THE STUDY OF KİNEMATICS OF PARALLEL MANIPULATORS

JEAN-PIERRE MERLET
INRIA Sophia Antipolis, BP 93, 06902 Sophia-Antipolis Cedex, France

Abstract. Manipulation of algebraic equations arise frequently in kinematic problems. But in many of these problems it is not necessary to solve the algebraic equations to establish interesting results as sometimes only the number of real solutions is important. Fortunately many theorems in algebraic geometry, some of them being not well known, may give some insight on this point. We present some of these theorems and show how they can be applied to demonstrate interesting results in the field of kinematic problems for parallel manipulators.

Key words: Parallel manipulators – Kinematics – Algebraic geometry

1. Introduction

Systems of algebraic equations play an important role in kinematics problems as most of these problems can be stated as solving such a system. For some kinematics problems it is not necessary to *solve* the equations but it is more important to determine:

- the maximum number of real roots of the system
- the number of real roots in a given interval

We will present some tools which can be used for these purposes *without* computing the roots of the equations and study their application for some kinematics problems related to parallel manipulators.

2. Bezout's theorem

This theorem is one of the most interesting in algebraic geometry. An extensive study of Bezout's theorem can be found in (Walker, 1950). We give here a simplified version of this theorem:

The intersection of m algebraic equations in n unknowns $(m \geq n)$ of degree n_1, $n_2, \ldots, n_m$ is constituted of at most $\prod_{i=1}^{i=m} n_i$ points

In the case of planar algebraic curves a version of Bezout's theorem can be stated as:

two curves of order m, n with no common components have exactly mn intersection points.

J. Angeles et al. (eds.), Computational Kinematics, 183–194.

3. Circularity

This notion and its application to kinematic problems has been discussed in detail in (Hunt, 1978).

3.1. Planar case

Bezout's theorem may seem to be rather strange in some cases. Let us consider two circles described by algebraic equation of order 2. It is well known that they will have at most two real intersection points....
Let a circle of radius r, with center at coordinates (a, b), be defined by the equation:

$$(x-a)^2+(y-b)^2-r^2=0$$

By expanding this equation we get:

$$x^2-2\,xa+a^2+y^2-2\,yb+b^2-r^2=0$$

The terms of this equation are not homogeneous, i.e. their order with respect to the variables x, y are 2, 1 or 0. Let us rewrite this equation with a new unknown w:

$$(\frac{x}{w}-a)^2+(\frac{y}{w}-b)^2-r^2=0$$

where w is simply a scaling factor. The previous equation is now homogeneous as it can be written as:

$$(x-aw)^2+(y-bw)^2-r^2w^2=0$$

or

$$x^2-2\,xaw+a^2w^2+y^2-2\,ybw+b^2w^2-r^2w^2=0$$

for which the order of each term with respect to the variables x, y, w is now 2. The system of unknowns x, y, w is called a planar homogeneous system of coordinates.

If $w=0$ the circle is infinitely enlarged and every point is at infinity. The line $w=0$ is called the *line at infinity* and this line crosses the circle in two points defined by:

$$x^2+y^2=0 \tag{1}$$

i.e. at the points $\mathcal{S}_1$, $\mathcal{S}_2$

$$\mathcal{S}_1\left\{\begin{array}{l} w=0 \\ x=iy \end{array}\right. \qquad \mathcal{S}_2\left\{\begin{array}{l} w=0 \\ x=-iy \end{array}\right.$$

These two imaginary points are called the *the imaginary circular points* and equation (1) defines the *imaginary circle*

As the parameters a, b, r do not appear in the definition of the imaginary circular points, they belong to any circle. Therefore, they belong too to the intersection of two circles. Accordingly, two circles with two real intersection points have also in common the two imaginary circular points, i.e. a total of four intersection points, in accordance with Bezout's theorem.

If a planar curve has the points S_1, S_2 as double, triple.. points it will be said that this curve has a *circularity* of 2,3,... Therefore a circle has circularity 1.

3.2. Application of circularity

Let us consider the following planar parallel manipulator described in figure 1. The triangular plate BDE is connected to the three fixed points

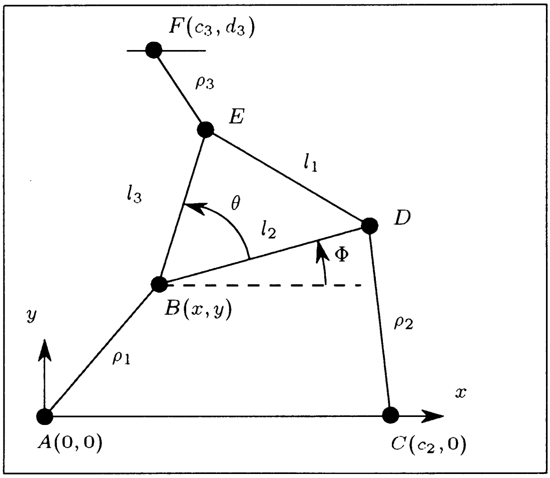


Fig. 1. A planar parallel manipulator.

A, C, F by three links with revolute joints at each extremity. A linear actuator in each link enables to change the link length and it may be shown that by controlling these lengths the posture of the triangular plate can be adjusted at will. Indeed let us assume that we have fixed the position and orientation of the triangular plate BDE in some reference frame. Therefore, the positions of its vertices are also known in this frame. The link lengths corresponding to the given posture are simply the distances between the points AB, CD, EF and we have solved the *inverse kinematic problem* of this manipulator. Formally let us define the reference frame such that: $A : (0,0)$, $C : (c_2, 0)$, $F : (c_3, d_3)$ and we define the posture of the triangular plate by the coordinates (x, y) in the reference frame of point B and its orientation by the angle Φ between the line BD and the x axis. The link lengths ρ can be computed as:

$$\rho_1^2 = x^2 + y^2 \tag{2}$$

$$\rho_2^2 = (x + l_2 \cos\Phi - c_2)^2 + (y + l_2 \sin\Phi)^2 \tag{3}$$

$$\rho_3^2 = (x + l_3 \cos(\Phi + \theta) - c_3)^2 + (y + l_3 \sin(\Phi + \theta) - d_3)^2 \tag{4}$$

Let $T = \tan(\Phi/2)$. We have:

$$\sin(\Phi) = \frac{2T}{1+T^2} \qquad \cos(\Phi) = \frac{1-T^2}{1+T^2} \tag{5}$$

The previous equations can be written now as:

$$x^2 + y^2 - \rho_1^2 = 0 \tag{6}$$

$$x^2 + x^2T^2 + y^2T^2 + y^2 + a_1x + a_2xT^2 + a_3T^2 + a_4yT^2 + a_5 = 0 \tag{7}$$

$$x^2 + x^2T^2 + y^2T^2 + y^2 + b_1x + b_2xT^2 + b_3T^2 + b_4yT^2 + b_5 = 0 \tag{8}$$

The orders of these equations are 2, 4, 4. Suppose now that the lengths of the links are fixed and that we want to determine the position and orientation of the triangular plate i.e. solve the *direct kinematic problem.* We have therefore to solve the previous system of algebraic equations. Using Bezout's theorem we deduce that this system will have at most 32 (2x4x4) solutions, either real or complex. We will show now thac in fact a smaller upper-bound of the number of real solutions can be established. Let us consider another mechanism, as described in figure 2.

This mechanism is called a *four-bar mechanism.* Many authors (Hunt, 1978) have shown that point C of this mechanism describes a sixth order curve, a *sextic* with a circularity of 3 (which is the maximum for the circularity of a sextic).

Now let us consider the four-bar mechanism $ABEDC$ in the mechanism of figure 1. E lies on the sextic of the four-bar mechanism but also belongs to the circle centered in F, of radius FE for a valid solution of the direct kinematic problem. E is therefore the intersection point of two algebraic curves of order 2 and 6 and there will be at most 12 intersection points according to Bezout's theorem.

But the intersection will contain the two circular imaginary points $\mathcal{S}_1$, $\mathcal{S}_2$ as triple points. Therefore there will be at most 6 *real* intersection points and therefore this is an upper bound for the number of postures of the direct kinematic problem.

This has been confirmed in (Gosselin, 1990) who has shown that the system of equations (2, 3, 4) can be reduced to a 6th-order polynomial in one variable. Indeed, let us subtract equation (2) from equations (3), (4). We get a linear system of two equations in the two unknowns x, y, which can be solved, the result being substituted into equation 2. The only unknown

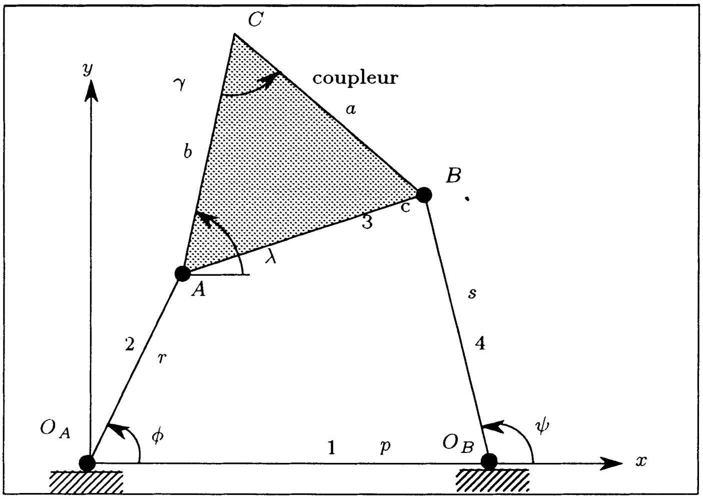


Fig. 2. A four-bar mechanism.

in this equation is now Φ. Using the substitution described by equation (5) the remaining equation becomes a 6th-order polynomial in T:

$$a_6T^6 + a_5T^5 + a_4T^4 + a_3T^3 + a_2T^2 + a_1T + a_0 = 0 \tag{9}$$

3.3. Spatial case

Let us consider now the intersection of two spheres i.e. two surfaces of degree 2 which, according to Bezout's theorem, must intersect along a curve of order 4.

But it is known that the intersection of two spheres is a circle of degree 2. We must therefore find a conic at infinity which explains the missing degree. We rewrite the equation of the sphere in homogeneous coordinates:

$$(x - aw)^2 + (y - bw)^2 + (z - cw)^2 - r^2w^2 = 0$$

The plane $w = 0$ is called the plane at infinity and the intersection of the sphere and this plane is found as:

$$x^2 + y^2 + z^2 = 0 \tag{10}$$

As none of the parameters a, b, r appear in this equation, this curve of order 2 belongs to all the spheres and, therefore, to the intersection of any two spheres. Equation (10) defines an imaginary cone whose intersection with the plane at infinity is the *imaginary spherical circle* which belongs to all the spheres.

The imaginary cone intersects the plane $z = 0$ along two imaginary lines defined by $x = \pm iy$ and, therefore, the circular imaginary points belong to the cone. As a consequence there cannot be more than 2 real intersection points between a sphere and a circle.

The circularity of a surface is then defined as the order of multiplicity with which it contains the imaginary spherical circle. A sphere has therefore a circularity 1. For example, it may be shown that a general torus (fourth-order surface) has a circularity 2 (maximal circularity).

3.4. Application to a kinematic problem

Let us consider the spatial mechanism described in figure 3. A triangular plate $B_1B_2B_3$ is connected to three fixed points A_1, A_2, A_3 by three links which have a rotoid joint at point A and a ball-and-socket joint at point B. We assume that the lengths of links A_1B_1, A_2B_2, A_3B_3 are fixed and we want to determine the possible locations of the triangular plate $B_1B_2B_3$ i.e. solve the direct kinematic problem for this mechanism.

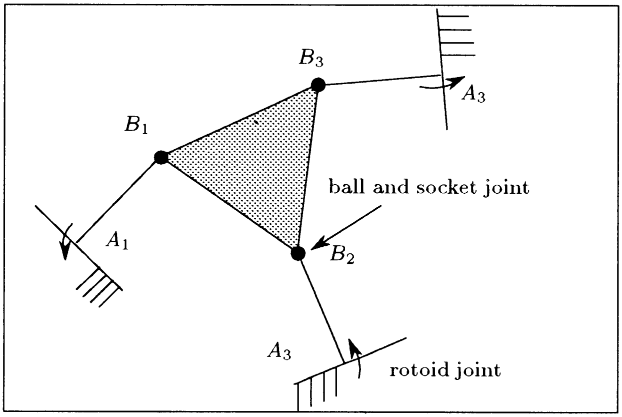


Fig. 3. A general spatial mechanism.

We will consider the spatial mechanism obtained when we dissociate one of the B_i. We get the mechanism described in figure 4 which is known under the name $RSSR$.

We use now one of Cayley's theorem (Hunt, 1978):

A line with two points C, D lying on two algebraic curves of degree n_c, n_d and circularities p_c, p_d describes a ruled surface of degree $2n_c(n_d - p_d) + 2n_d(p_c - n_c)$

For the $RSSR$ mechanism we have two points lying on two circles i.e. $n_c = n_d = 2, p_c = p_d = 1$. The order of the surface is therefore 8 and B lies on

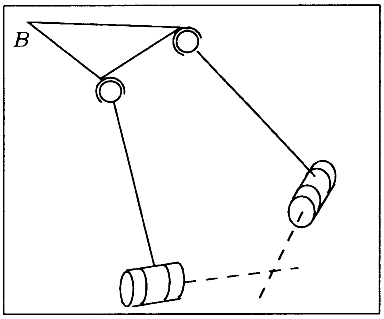


Fig. 4. The *RSSR* mechanism

a surface of order 16 (as point B can freely rotate around the line joining the centers of the ball-and-socket joints). It may be shown that the circularity of this surface is 8 (Merlet, 1989). For a valid posture of the mechanism described in figure 3, point B_1 belongs to such a surface but also to the circle centered in A_1 whose radius is the link length . According to Bezout's theorem there are 32 intersection points (either complex or real) between the surface and the circle and, according to the circularity of the surface and the circle, 16 points among these 32 intersection points are the circular imaginary points. Therefore, there are at most 16 real intersection points and, consequently, the direct kinematic problem has at most 16 solutions.

Now let us consider a parallel manipulator (figure 5). In this manipulator the 6 legs have their extremities connected to the plates by ball-and-socket joints. Their lengths can be modified in order to control the position and orientation of the upper plate.

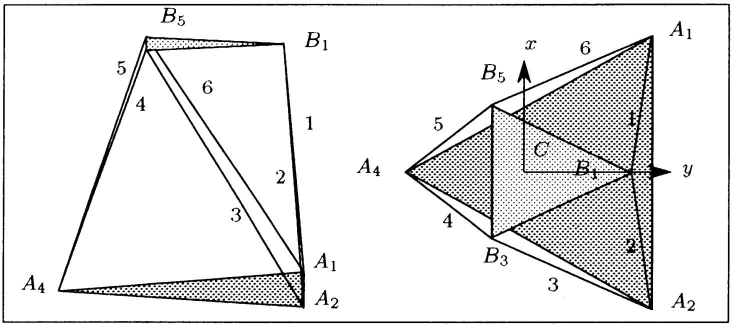


Fig. 5. A parallel manipulator.

Suppose that the legs have a known fixed length. The direct kinematics problem consists in determining the maximum number of postures of the upper plate for these leg lengths. Let us consider two legs with a common point B on the upper plate. As the leg lengths are fixed, B will lie on a circle whose center and radius can be computed from the leg lengths and the position of the joint centers on the fixed plate. We can therefore substitute the two legs by a virtual leg whose only possible motion is a rotation around a fixed axis.

This can be done for any pair of legs sharing a common point on the upper plate and, therefore, the parallel manipulator upper plate will have the same possible postures as the mechanism described in figure 3: its maximum number of postures will be 16.

A more tedious way to demonstrate this result is to combine the algebraic equations describing the inverse kinematic problem to get a polynomial in one variable, whose order shall be 16 or less. A sixteenth-order polynomial has been first found in (Charentus and Renaud, 1989) and later by many authors, for example (Dedieu 1990) (who give additional results about the convexity of the solution), (Griffis, 1989), (Innocenti, 1990). Using this result, an example of manipulator with 16 possible postures for the end-effector has been presented by (Merlet, 1989) and (Dedieu, 1990). In the former reference it has been shown that this result can be extended to many others manipulators as soon as they have a triangular end-effector.

4. Number of real roots of a polynomial in a given interval

The systems of algebraic equations arising in some kinematic problems can be reduced to the analysis of a polynomial in only one variable which shall furthermore lie in a given interval. Therefore, it is of interest to consider a polynomial in one variable and to determine the number of its real roots in a given interval.

4.1. Sturm's method

An excellent and practical introduction to this method can be found in (Mineur, 1966).

Principle

Let $f(x)$ be a polynomial of degree n in x

$$f_0(x) = \sum_{i=0}^{i=n} a_n x^n = 0$$

We consider the first derivative of this polynomial with respect to x:

$$f_1(x) = f_0'(x)$$

We denote by $\mathrm{Rem}(f_{i-1}(x), f_i(x))$ the remainder of the Euclidian division of $f_{i-1}(x)$ by $f_i(x)$

We build a sequence of functions by:

$$f_{i+1} = -\mathrm{Rem}(f_{i-1}(x), f_i(x)) \quad i \in [1, n-1]$$

The last function of this sequence does not depend upon x. Let $[x_1, x_2]$ be the interval in which we are looking for the real roots of $f_0(x) = 0$.

Sturm's theorem

The number of real roots of the equation $f_0(x) = 0$ in the interval $[x_1, x_2]$ is obtained as the number of sign changes in the sequence $f_i(x_1), f_{i+1}(x_1), i \in [0, n-1]$ minus the number of sign changes in the sequence $f_i(x_2), f_{i+1}(x_2), i \in [0, n-1]$.

4.2. Application example

We consider a particular case of the planar parallel mechanism described in a previous section (figure 6). The equations giving the links lengths for a given posture of the end-effector are:

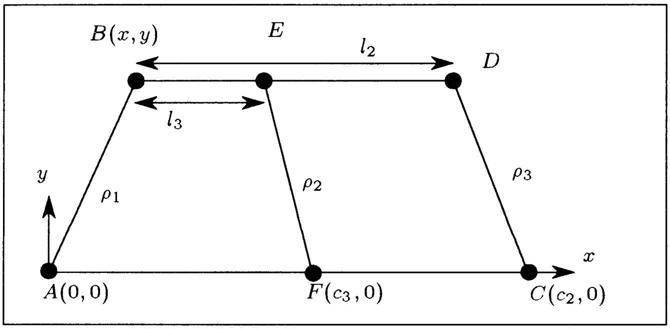


Fig. 6. A special case of planar parallel manipulator.

$$\begin{aligned}
\rho_1^2 &= x^2 + y^2 \\
\rho_2^2 &= (x + l_2 \cos\Phi - c_2)^2 + (y + l_2 \sin\Phi)^2 \\
\rho_3^2 &= (x + l_3 \cos\Phi - c_3)^2 + (y + l_3 \sin\Phi)^2
\end{aligned}$$

By manipulating these equations in a similar manner as in section 3.2, they can be reduced to a polynomial in one variable:

$$f_0(T) = a_3T^3 + a_2T^2 + a_1T + a_0 = 0 \tag{11}$$

with $T = \cos \Phi$. Therefore there can be up to 3 real roots for this polynomial and, as each root defines two values for Φ, we may think that an upper bound of the number of postures for the direct kinematic problem is 6. We will show now that, in fact, there will be at most 4 solutions to this problem.

To solve the direct kinematic problem we have to find the real roots of the polynomial, but we are looking only for the roots in the interval [-1,1]. It may be shown that $f_0(1), f_0(-1)$ are always strictly positive (their symbolic values are squares)

If we build the Sturm sequence, we get four functions f_0, f_1, f_2, f_3, where f_3 is a constant. We are looking for sequences such that the number of real roots in the interval [-1,1] is maximal. This number will be a maximum in four cases:

	f_0	f_1	f_2	f_3	number of sign changes
$x = -1$	+	-	+	+	2
$x = 1$	+	+	+	+	0

	f_0	f_1	f_2	f_3	number of sign changes
$x = -1$	+	+	-	+	2
$x = 1$	+	+	+	+	0

	f_0	f_1	f_2	f_3	number of sign changes
$x = -1$	+	-	+	-	3
$x = 1$	+	+	+	-	1

	f_0	f_1	f_2	f_3	number of sign changes
$x = -1$	+	-	+	-	3
$x = 1$	+	-	-	-	1

In all cases the number of roots will be at most 2 and, therefore, the direct kinematic problem will have up to 4 solution.

5. Huat's Theorem

Let a polynomial equation of degree n in x:

$$f_0(x) = \sum_{i=0}^{i=n} a_n x^n = 0$$

with real coefficients.

Theorem: *If the roots of $f_0(x)$ are all real, the square of every non extremal coefficients is necessarily greater than the product of its neighboring coefficients*

$$a_k^2 > a_{k-1}a_{k+1} \quad \forall k \in [1, n-1]$$

In fact, Huat's theorem is a result of Newton inequalities (Hardy, 1967), which states that, if $f_0(x)$ has only real roots, then:

$$k(n-k)a_k^2 \geq (k+1)(n-k+1)a_{k-1}a_{k+1} \;\; \forall\, k \in [1, n-1]$$

5.1. Application in kinematics

Let us consider the planar parallel manipulator described in figure 1. We have seen in a previous section that, for a fixed set of link lengths, there can be a maximum of 6 different postures for the triangular mobile plate. We are considering now a robot with a given geometry and are looking for a set of link lengths such that the direct kinematic problem has effectively 6 solutions i.e. 6 postures of the end-effector can be found.

To solve this problem we may choose randomly three link lengths, compute the coefficients of the 6th order polynomial (9) and then solve the polynomial until we find a set of link lengths such that all the 6 roots of the polynomial are real. Although this method has worked in practice (an example of solution is given in (Merlet, 1989)), the computation time may be huge.

A faster way is to choose randomly only two of the three link lengths, say ρ_1, ρ_2 and then compute the 7 coefficients a_i of the forward kinematics polynomial (9) as functions of the unknown link length ρ_3.

Then we compute the square of all the non extremal coefficients minus the product of their neighbor coefficients i.e. $a_1^2 - a_0a_2, a_2^2 - a_1a_3, a_3^2 - a_2a_4, a_4^2 - a_5a_3, a_5^2 - a_4a_6$, which happen to be all fourth-order polynomials in ρ_3.

The roots of these 5 polynomials P_j are computed and are used to determine for each polynomial the intervals of ρ_3 such that the polynomial is positive.

If the intersection $I_\cap$ of these intervals is empty, then there is no value of ρ_3 such that the direct kinematic problem will have 6 solutions for the link lengths ρ_1, ρ_2.

On the contrary, if the intersection is not empty, the possible solutions for ρ_3 will lie in the interval $I_\cap$. Therefore, random values for ρ_3 can be tested but only in $I_\cap$.

Such an algorithm has been implemented using the symbolic computation program MAPLE.

6. Conclusion

Dealing with algebraic equations is the "essence" of kinematic problems, but many of these problems can be solved in an elegant way without determining the roots of these equations. By using basic theorems of algebraic geometry we have shown that many powerful results can be established in the field of parallel manipulator kinematic problems. These results have been established in most cases by dealing either with the geometrical aspect of the problem or with manipulations on the symbolic values of the coefficient of the algebraic equations that arise during the solution of the problem. Therefore, we have avoided to use numerical procedures in which numerical errors may introduce spurious results. Unfortunately many of these algebraic geometry theorems are not well known and are missing in many textbooks.

References

Charentus S. and Renaud M.: 1989, 'Calcul du modèle géométrique direct de la plate-forme de Stewart', Research Report 89260, LAAS, Toulouse, France

Dedieu J-P and Norton G.H.: 1990, 'Stewart varieties: a direct algebraic method for Stewart platforms', *SigSam*, 24(4)

Gosselin C.: 1990, 'Solution polynomiale au problème de la cinématique directe des manipulateurs parallèles plans à 3 degrés de liberté', *Mechanism and Machine Theory*, 27(2):107–119

Griffis M. and Duffy J.: 1989, 'A forward displacement analysis of a class of Stewart platform', *J. of Robotics Systems*, 6(6):703–720,

H. Mineur: 1966, *Technique de calcul numérique.* Dunod

Hardy , Littlewood , and Polya: 1967, *Inequalities.* Cambridge University Press

Hunt K.H.: 1978, *Kinematic geometry of mechanisms.* Clarendon Press

Innocenti C. and Parenti-Castelli V.: 1990, 'Direct position analysis of the Stewart platform mechanism', *Mechanism and Machine Theory*, 25(6):611–621,

Merlet J-P.: 1989, 'Manipulateurs parallèles, 4eme partie : mode d'assemblage et cinématique directe sous forme polynomiale', Research Report 1135, INRIA

Walker R.J.: 1950, *Algebraic curves.* Springer-Verlag, New-York

Part 5

Motion Planning

SINGULARITY CONTROL FOR SIMPLE MANIPULATORS USING "PATH ENERGY"

J. E. LLOYD
McGill Research Center for Intelligent Machines &
Department of Electrical Engineering, McGill University
3480 University Street, Montreal, Canada H3A 2A7
e-mail: lloyd@mcrcim.mcgill.ca

Abstract. A new method is presented for controlling the trajectories of straight-line Cartesian paths near the kinematic singularities of simple manipulators. Conventional approachs to this problem, which typically employ a pseudo-inverse of the manipulator Jacobian, result in path deviations and have difficulty controlling joint accelerations. The more global approach described here uses a "potential function" in the region of the singularity to reduce the path velocity in such a way that both the joint velocities and accelerations remain bounded without incurring any deviation from the desired path. For cases such as the elbow or shoulder singularities of the PUMA manipulator, the results are very good and the necessary computations are simple enough to be done on-line.

1. Introduction

It is well known that most serial chain robot manipulators possess regions in their workspace where the mapping from Cartesian to joint coordinates becomes ill-conditioned along one or more degrees of freedom. Such regions correspond to *singularities* of the manipulator. The execution of a Cartesian trajectory in such a region may produce arbitrarily large accelerations or velocities in one or more of the manipulator's joints. Singularities are associated with a loss of rank in the manipulator Jacobian $\mathbf{J}$ which maps joint velocities $\dot{\mathbf{\Theta}}$ into Cartesian velocities $\mathbf{v}$. The inverse of the Jacobian (or a pseudo-inverse in the case of a redundant manipulator) can be used to determine the joint velocities required to effect a particular Cartesian velocity, according to

$$\dot{\mathbf{\Theta}} = \mathbf{J}^{-1}\mathbf{v}. \tag{1}$$

Clearly, in regions where the Jacobian is ill-conditioned, some of the resulting joint velocities may become very large.

Since singularities restrict the Cartesian workspace of the manipulator, and because Cartesian trajectories can easily blunder into them, particularly when controlled on-line by sensors or operator inputs, techniques for managing singularities are of importance.

Most approaches to singularity control involve trying to condition the relationship in (1), either by using some form of pseudo-inversion technique (Chiaverini, et al., 1991; Maciejewski and Klein, 1989; Wampler, 1988) or by directly eliminating degenerate degrees of freedom from the Jacobian (Aboaf and Paul, 1987). An overview is given in (Chiaverini et al., 1990).

J. Angeles et al. (eds.), Computational Kinematics, 197–206.

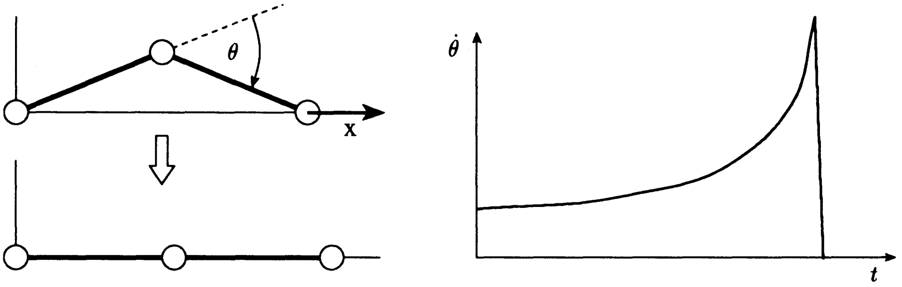


Fig. 1. A two-link manipulator being driven to its outer boundary singularity (left). The corresponding velocity profile (for discretely sampled values of θ) is shown at the right.

A common limitation of these approaches is that they control only the joint velocities (as a consequence of using only equation 1). The effect of the singularity, however, extends to higher derivatives. In particular, the joint accelerations must be controlled as well, as has been discussed briefly in (Maciejewski and Klein, 1989).

As an example of this, consider Figure 1, which shows a two-link planar manipulator being driven at constant speed along the x axis into the boundary singularity at $\theta = 0$. The velocity $\dot{\theta}$ rises rapidly as the singularity is approached, but when the singularity is reached, the velocity falls immediately to zero. The problem in this case is not only that $\dot{\theta}$ becomes large, but that it is also discontinuous at the singularity. The practical effect of this is that, upon reaching the workspace boundary, θ will overshoot, causing the manipulator to reverse direction and possibly inducing controller instabilities. This problem has been noted in (Deo and Walker, 1992).

Unfortunately, acceleration constraints complicate the singularity problem considerably. Simply "clipping" the acceleration can introduce considerable wander and overshoot into the computed path, since this fails to ensure that the integral of the resulting velocity profile matches the displacement to the target.

In this paper, we introduce a somewhat different approach to singularity control, in which we handle velocity and acceleration constraints by simply slowing down the trajectory execution *without* deviating from the desired path. This capability is particularly useful because it opens the way for making explicit time/accuracy tradeoffs near singularities.

The method is intended for simple manipulators for which closed form inverse kinematic solutions are available. It implicitly makes use of the observations by (Nielson et al., 1990; Pohl and Lipkin, 1991) that well-behaved motion along the degenerate degrees of freedom associated with a singularity is often possible, particularly when the trajectory is sampled in discrete time, providing the desired Cartesian path displacements are scaled appropriately.

2. Problem Description

We will limit our consideration to straight-line Cartesian trajectories with constant orientation. Such a path can be described in terms of a single parameter s by

$$\mathbf{p}(s) = \mathbf{p}_1 + s\,\hat{\mathbf{p}}_d, \tag{2}$$

where $\mathbf{p}_1$ is the initial point and $\hat{\mathbf{p}}_d$ is a unit vector in the path's direction.

Timing for the trajectory is established by making s a function of time t. In areas of the workspace where the Jacobian is well-conditioned, it is generally possible to keep the joint velocities $\dot{\mathbf{\Theta}}$ and accelerations $\ddot{\mathbf{\Theta}}$ within bounds by placing constant bounds on $\dot{s}$ and $\ddot{s}$:

$$|\dot{s}| \leq M_v, \quad |\ddot{s}| \leq M_a. \tag{3}$$

Such bounds are also frequently part of the task specification.

Close to singularities, these constraints will not be sufficient, and so the path timing $s(t)$ must be further constrained. If we were only interested in limiting the joint velocities, then it would be enough to simply scale $\dot{s}$ to match the corresponding rise in $\dot{\mathbf{\Theta}}$. However, we must also limit the joint accelerations, while at the same time preserving the bounds on $\dot{s}$ and $\ddot{s}$ given in (3) so as to maintain the behavior of joints which are not directly affected by the singularity.

3. The Path Energy Approach

It is often the case for simple manipulators that one joint *dominates* the singularity, such that if the derivatives for that joint are controlled, the derivates of any other joints associated with the singularity will be controlled as well.

In such cases, we should be able to control the singularity by "stretching" the velocity timing for the dominant joint and then back-solving for the appropriate path parameter timing $s(t)$. We have found that an easy way to do this is to use the concept of "path energy", which is defined simply as $\dot{s}^2/2$.

With path energy so defined, a potential energy function $U(s)$ can be established to control the path execution. Figure 2 shows one such function $U_0(s)$ defined by two line segments of slopes $\pm M_a$ descending from 0 to a third line segment of constant value $-M_v^2/2$. If the path energy is initialized to zero at $s = 0$, then letting the path parameter s follow this potential will

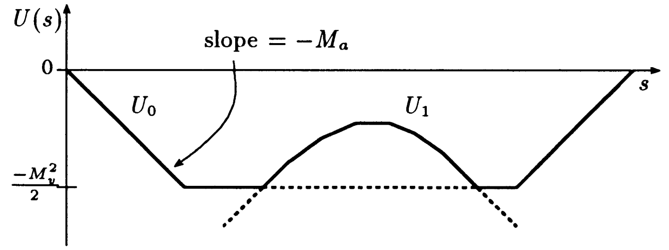


Fig. 2. Potential energy curves for controlling the path parameter s.

generate a simple trapezoidal profile for $\dot{s}$ that satisfies the constraints in (3).

"Slowing down" the velocity profile in certain parts of the path can be accomplished by adding extra potential functions $U_i(s)$ that serve as "hills" over which the path parameter must climb, such as U_1 in Figure 2. The net $U(s)$ can be computed from these by

$$U(s) = \max_i U_i(s). \tag{4}$$

One useful aspect of the energy function approach is that it allows easy accommodation of the constraints in (3). The maximum operation in (4) takes care of the velocity, and acceleration bounds are satisfied by simply ensuring that $|dU_i(s)/ds| \leq M_a$, which follows from the fundamental relationship

$$\ddot{s} = -dU(s)/ds.$$

Using this technique, the slowing down required in the region of a singularity can be accomplished using an appropriate potential function U_s. How can this function be generated?

Let the dominant angle for a singularity be given by θ, and let $I = [s_a, s_b]$ define a path interval where $\dot{\theta}$ or $\ddot{\theta}$ exceed limits under the default path timing. Let θ_a, $\dot{\theta}_a$, θ_b, and $\dot{\theta}_b$ denote θ and $\dot{\theta}$ at s_a and s_b. From the manipulator forward kinematics, θ can be determined as a function $\theta = f(s)$ of the path parameter. Connect $\theta_a, \dot{\theta}_a$ and $\theta_b, \dot{\theta}_b$ with a smooth timed curve $\theta = q(t)$ that satisfies the joint velocity and acceleration constraints. Then sample $q(t)$ at various values of t and determine the corresponding values for s and $\dot{s}$ from

$$s = f^{-1}(q(t)), \quad \dot{s} = \frac{df^{-1}}{d\theta}(q(t))\dot{\theta}.$$

These values are used to construct a piecewise linear approximation to the required potential curve U_s.

To ensure that s remains monotonically increasing, $q(t)$ must be constructed so that $\text{sgn}(\dot{q}) = \text{sgn}(df/ds)$ within I. If I contains points s_z where $df/ds = 0$, then knot points where $\theta = f(s_z)$ and $\dot{\theta} = 0$ must be added to $q(t)$. The inversion of f can be performed piecewise between these knot points.

The rate at which $q(t)$ must be sampled when producing points for the potential has only been studied experimentally thus far. We have found that a rate equal to the nominal trajectory sampling rate (about 30 milliseconds) works well, and that this can be reduced farther from the singularity.

4. Elbow and Shoulder Singularities of the PUMA Manipulator

To compute a controlling potential $U_s(s)$, we need to be able to relate the path parameter s to the dominant angle for the singularity. In this section we do this for the elbow and shoulder singularities of the PUMA robot. Because the PUMA is wrist-partioned, such paths imply a straight-line path for the center of the wrist, which is itself a function of only the first three joints.

4.1. Review of the PUMA kinematics

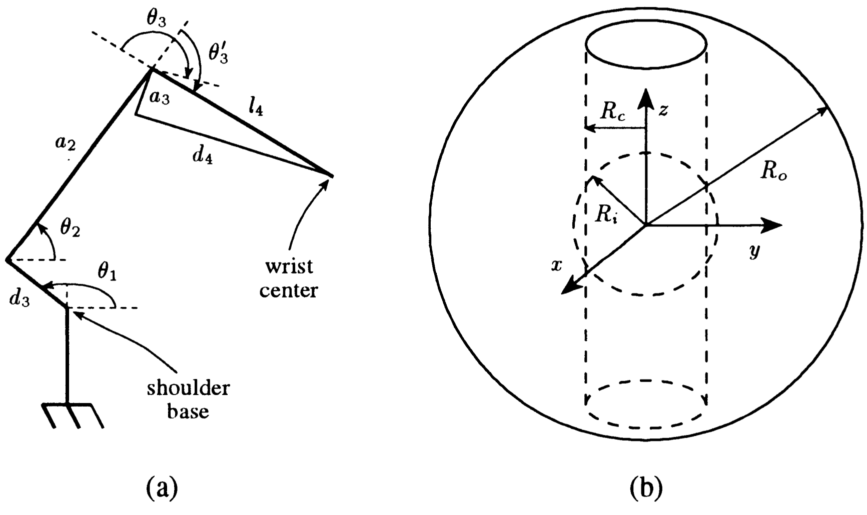


Fig. 3. (a) Kinematic structure of the first three links of the PUMA manipulator. (b) The workspace boundaries for the center of the wrist.

Results similar to those found in (Elgazzar, 1985) are given here. Figure 3(a) shows the kinematic structure of the first three joints. Note that the canonical offsets d_4 and a_3 can be combined into a single offset $l_4 = \sqrt{d_4^2 + a_3^2}$ so that joints two and three comprise a two-link revolute sub-manipulator, with link lengths a_2 and l_4. Redefining θ_3 by $\theta_3' = -\theta_3 + \pi/2 - \tan^{-1}(a_3/d_4)$ makes it correspond to the more standard definition of the elbow joint for a two-link manipulator.

With the origin taken to be the base frame for joint one, the forward kinematics gives the position of the wrist center $(p_x, p_y, p_z)^T$ as a function of the joint angles $(\theta_1, \theta_2, \theta_3)^T$:

$$\begin{aligned} r_1 &= d_4 S_{23} + a_3 C_{23} + a_2 C_2, \\ p_x &= C_1 r_1 - S_1 d_3, \\ p_y &= S_1 r_1 + C_1 d_3, \\ p_z &= d_4 C_{23} - a_3 S_{23} - a_2 S_2, \end{aligned} \tag{5}$$

where S_i, C_i, S_{ij}, and C_{ij} denote $\sin\theta_i$, $\cos\theta_i$, $\sin(\theta_i + \theta_j)$, and $\cos(\theta_i + \theta_j)$.

It is easier to state the inverse kinematics using θ_3' in place of θ_3:

$$r_1 = k_s \sqrt{p_x^2 + p_y^2 - d_3^2}, \tag{6}$$

$$\theta_1 = \tan^{-1}(p_y/p_x) - \tan^{-1}(d_3/r_1),$$

$$\theta_3' = k_e \cos^{-1}\left(\frac{r_1^2 + p_z^2 - a_2^2 - l_4^2}{2a_2 l_4}\right), \tag{7}$$

$$\theta_2 = \tan^{-1}\left(\frac{l_4 \sin\theta_3'}{a_2 + l_4 \cos\theta_3'}\right) - \tan^{-1}\left(\frac{p_z}{r_1}\right).$$

The variables k_s and k_e are assigned values of either 1 or -1 to resolve solution ambiguities and select the robot's "configuration". Similar variables, with slightly different definitions, are described in (Elgazzar, 1985).

The elbow and shoulder singularities occur at the boundaries of the workspace reachable by the wrist center (ignoring joint limits). These boundaries are geometrically quite simple (Zhang, 1991), consisting of an outer sphere of radius $R_o = \sqrt{(a_2 + l_4)^2 + d_3^2}$ and an inner sphere of radius $R_i = \sqrt{(a_2 - l_4)^2 + d_3^2}$, both centered at the origin, and a cylinder of radius $R_c = d_3$ centered on the z (or θ_1) axis (Figure 3(b)).

4.2. The elbow singularity

The elbow singularity occurs when $\theta_3' = 0$. In this situation the arm is fully outstretched and is touching the outer sphere of the workspace boundary*.

The dominant angle for the elbow singularity is θ_3', which we will now relate to the path parameter s.

It will be convenient to replace s in (2) by $u = s + \mathbf{p}_1 \cdot \hat{\mathbf{p}}_d$, so that

$$\mathbf{p}(u) = \mathbf{p}_0 + u\mathbf{p}_d, \quad \text{with} \quad \mathbf{p}_0 = \mathbf{p}_1 - (\mathbf{p}_1 \cdot \hat{\mathbf{p}}_d)\hat{\mathbf{p}}_d. \tag{8}$$

* There is a companion singularity at $\theta_3' = \pi$, which corresponds to the inner sphere of the workspace boundary. This singularity is slightly more complicated to handle and will not be discussed here. As a matter of practicality, it lies outside the range for joint 3.

$\mathbf{p}_0$ gives the point of closest approach to the origin, which implies that $\mathbf{p}_0 \cdot \hat{\mathbf{p}}_d = 0$ and $||\mathbf{p}||^2 = p_0^2 + u^2$ (where $p_0 \equiv ||\mathbf{p}_0||$). Noting that $||\mathbf{p}||^2 = p_x^2 + p_y^2 + p_z^2$, applying (7) and (6), we get

$$\theta_3' = k_e \cos^{-1}\left(\frac{p_0^2 + u^2 - d_3^2 - a_2^2 - l_4^2}{2a_2 l_4}\right), \tag{9}$$

which can be differentiated to yield

$$\dot{\theta}_3' = -\frac{u}{a_2 l_4 \sin\theta_3'}\,\dot{u}. \tag{10}$$

Now we need to solve for u and $\dot{u}$. It is easy to invert (9) to get

$$u = \pm\sqrt{2a_2 l_4 \cos\theta_3' - p_0^2 + d_3^2 + a_2^2 + l_4^2}.$$

To determine the sign, apply the sgn function to both sides of (10) to get

$$\operatorname{sgn}(\dot{\theta}_3') = -\frac{\operatorname{sgn}(u)\operatorname{sgn}(\dot{u})}{\operatorname{sgn}(\theta_3')}.$$

From (9) we have that $\operatorname{sgn}(\theta_3') = k_e$, and if the path is directed such that s (and consequently u) is increasing, then $\operatorname{sgn}(\dot{u}) = 1$ and so $\operatorname{sgn}(u) = -k_e \operatorname{sgn}(\dot{\theta}_3')$. Lastly, inverting (10) for $\dot{u}$, we obtain

$$u = -k_e \operatorname{sgn}(\dot{\theta}_3')\sqrt{2a_2 l_4 \cos\theta_3' - p_0^2 + d_3^2 + a_2^2 + l_4^2},$$

$$\dot{u} = -\frac{a_2 l_4 \sin\theta_3'}{u}\,\dot{\theta}_3'. \tag{11}$$

4.3. The shoulder singularity

The shoulder singularity occurs when the argument of the square-root in equation (6) equals zero. The set of points for which this is true corresponds to the inner workspace cylinder of radius $R_c = d_3$. The shoulder singularity is dominated by θ_1.

Since θ_1 is a function of p_x and p_y only, the problem can be treated as a two-dimensional one in the x-y plane. From this perspective, the cylinder becomes a circle of radius d_3 centered on the origin (Figure 4).

The projection of the path (2) into the x-y plane is also a straight-line path that can be expressed as

$$\mathbf{q}(s) = \mathbf{q}_1 + s\,Q\,\hat{\mathbf{q}}_d,$$

where $\mathbf{q}_1$ is the initial point, $\mathbf{q}_d$ is a unit vector in the path direction, and $Q = \hat{\mathbf{q}}_d \cdot \hat{\mathbf{p}}_d$.

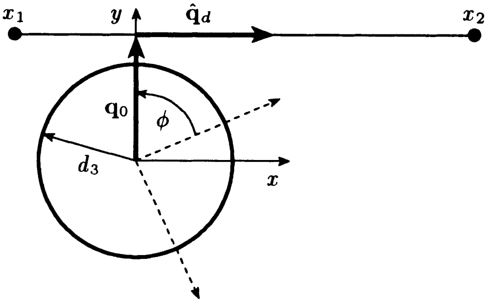


Fig. 4. Path projected into the x-y plane, with the axes rotated so that the path is parallel to the x axis, while x_1 and x_2 denote the path endpoints.

Again, it will be convenient to replace s by $u = Qs + \mathbf{q}_1 \cdot \hat{\mathbf{q}}_d$, so that

$$\mathbf{q}(u) = \mathbf{q}_0 + u\hat{\mathbf{q}}_d, \quad \text{with} \quad \mathbf{q}_0 = \mathbf{q}_1 - (\mathbf{q}_1 \cdot \hat{\mathbf{q}}_d)\hat{\mathbf{q}}_d. \tag{12}$$

(Note that if Q is very small, the corresponding displacement of θ_1 will also be small and so, this case is not of concern).

Now, without loss of generality, rotate the x and y axes by the angle $\phi = \tan^{-1}(q_{0x}/q_{0y})$, so that $\mathbf{q}_0$ is parallel to the y axis and $\hat{\mathbf{q}}_d$ is parallel to the x axis (Figure 4). $\mathbf{q}$ now becomes

$$\mathbf{q} = \begin{pmatrix} q_d\, u \\ q_0 \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix}, \tag{13}$$

where $q_0 = \|\mathbf{q}_0\|$ and $q_d = \pm 1$, depending on the direction of $\mathbf{q}_d$ along the x axis. θ_1 is also transformed according to $\theta_1' = \theta_1 - \phi$, although for notational simplicity we will denote θ_1' simply as θ_1. Combining (13) with the forward kinematics (5) and differentiating yields

$$u = \frac{C_1 q_0 - d_3}{q_d\, S_1}, \quad \dot{u} = \frac{-q_0 + C_1 d_3}{q_d\, S_1^2}\dot{\theta}_1. \tag{14}$$

5. Implementation and Experimental Results

Our implementation of this method for the PUMA works in this way: First, the path is clipped to lie entirely within the manipulator workspace. This is easy to do using methods described in (Zhang, 1991). Second, the controlling potentials are constructed roughly as follows: let θ be the dominant angle for a singularity. A trapezoidal velocity profile is constructed to connect the values of θ at the path endpoints (computed from the forward kinematics). This is sampled at time intervals of about 30 milliseconds, and the corresponding values of θ are used to compute s and $\dot{s}$, and hence $U_s(s)$, from

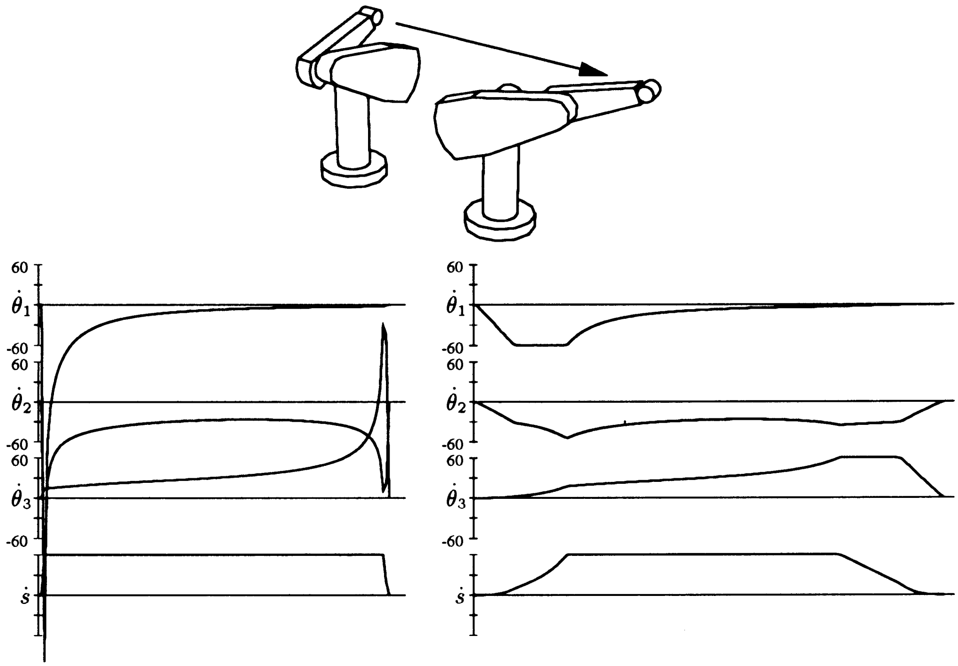


Fig. 5. Velocity profiles for $\dot\theta_1$, $\dot\theta_2$, $\dot\theta_3$, (deg/s) and $\dot s$ (m/s) for a straight-line motion (pictured at the top) from a shoulder-singular position at $(d_3, 0, 300.0)^T$ to an elbow singular position at $(R_o, 0, 300.0)^T$. The graphs at the left show the velocities without singularity control; note the large spikes. The graphs at the right show the velocities as controlled by the method described in the paper.

the relations (11) or (14). In the case of the elbow singularity, if the path contains the point $u = 0$ (see 8), then the corresponding value of θ'_3 must be computed and added to the profile as a knot point with $\dot\theta'_3 = 0$ (since this corresponds to a place where $df/ds = 0$, as described in section 3). Lastly, the net potential function $U(s)$ is integrated to produce the output trajectory. Integration is relatively simple because $U(s)$ is composed of line segments.

In practice, we can omit computing many of the points in a controlling potential for locations far from the singularity. In these places, the controlling potential will generally lie greatly *below* the default path energy. The sampling rate used to build the potential can also be reduced farther from the singularity. With these optimizations, calculation of the controlling potentials typically requires about 20 points for the elbow singularity and 35 points for the shoulder singularity. The total compute time is about four milliseconds on a Silicon Graphics Indigo workstation with an IP12 CPU, permitting all computations to be done on-line. The results have been tested and shown to work well both in simulation and on a physical robot. A particular demonstration is shown in Figure 5.

6. Conclusion and Future Work

A "path energy" method has been developed for computing straight-line Cartesian trajectories near certain singularities of simple robot manipulators. The resulting trajectory can be calculated quickly, and satisfies both joint velocity and acceleration constraints without deviating from the path.

Further investigation should study extending the method to curved trajectories, handling multiple singularities, and dealing with on-line changes in the path direction. Also, as currently implemented, the controlling potentials are computed prior to execution of the path. While this can be done quickly (in several milliseconds), it would be more efficient to compute the potentials concurrently with the path execution, by previewing the trajectory ahead of the current path point.

Acknowledgements

Support for this work was provided by a grant from the Institute for Robotics and Intelligent Systems (IRIS) of Canada's Centers of Excellence Program (NCE) under the project C-2: Simulation, Control and Planning in Robotics. Funding from NSERC, the Natural Sciences and Engineering Research Council of Canada, is also acknowledged.

References

Aboaf, E.W., and Paul, R.P. 1987. Living with the Singularity of Robot Wrists. *IEEE Conference on Robotics and Automation*, Raleigh, North Carolina, pp. 1713-1717.

Chiaverini, S., Egeland, O., and Kanestrom, R.K. 1991. Achieving User-defined Accuracy with Damped Least-Squares Inverse Kinematics. *Fifth International Conference on Advanced Robotics*, Pisa, Italy, pp. 672-677.

Chiaverini, S., Sciavicco, L., and Siciliano, B. 1990. Control of robotics systems through singularities. *International Workshop on Nonlinear and Adaptive Control: Issues in Robotics*, (Springer Verlag, Lecture Notes in Control and Information Sciences, 162), Grenoble, France, pp. 285-295.

Deo, A.S., and Walker, I.D. 1992. Robot Subtask Performance with Singularity Robustness using Optimal Damped Least-Squares. *IEEE Conference on Robotics and Automation*, Nice, France, pp. 434-441.

Elgazzar, S. 1985. Efficient Kinematic Transformations for the PUMA 560 Robot. *IEEE Journal of Robotics and Automation*, RA-1(3): 142-151.

Maciejewski, A.A., and Klein, C.A. 1989. The Singular Value Decomposition: Computation and Applications to Robotics. *International Journal of Robotics Research*, December 1989, pp. 8(6): 63-79.

Nielsen, L., Canudas de Wit, C., and Hagander, P. 1990. Controllability Issues of Robots near Singular Configurations. *Advances in Robot Kinematics, 2nd International Workshop*, Linz, Austria, pp. 283-290.

Pohl. E.D., and Lipkin, H. 1991. A new method of robotic motion control near singularities. *Fifth International Conference on Advanced Robotics*, Pisa, Italy, pp. 405-410.

Wampler, C.W., and Leifer, L.J. 1988. Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 110(31): 31-38.

Zhang, H., 1991. Feasibility Analysis of Displacement Trajectories for Robot Manipulators with a Spherical Wrist. *IEEE Conference on Robotics and Automation*, Sacramento, California, pp. 1252-1257.

AN INVESTIGATION OF PATH TRACKING SINGULARITIES FOR PLANAR 2R MANIPULATORS

J. KIEFFER AND B. O'LOGHLIN
Engineering Department, The Faculties,
The Australian National University
Canberra, Australia

Abstract. The planar 2R manipulator is used as a vehicle for investigating the problem of tracking end-effector paths that force the manipulator into a singular configuration. Results show that isolated points, turning points, nodes, cusps, hypernodes, and hypercusps can arise in the locus of inverse kinematic solutions depending the end-point path's degree of contact with the workspace boundary. Methods for determining smooth local representations of each type of path-tracking singularity are developed based on low-order analysis. These representations provide complete low-order information on all families of trajectories that track the path at the singularity.

1. Introduction

Kinematic singularities of serial manipulators are of interest for a number of reasons. Primarily for the nuisance they present to robot control, but also due to their fundamental relation to workspace boundaries, closed-form inverse kinematic solutions, and the possibility of using them to gain mechanical advantage. A fairly comprehensive review of related literature is given in Kieffer (1993).

The problem of tracking end-effector paths that force the manipulator into a singularity configuration was investigated by Kieffer (1992) and (1993) for general six-degree-of-freedom serial manipulators. Results showed that the locus of inverse kinematic solutions can take the form of three types of curve singularities: isolated points, turning points, and simple nodes. In addition, general algorithms were developed to determine smooth local models for each of these cases, but these results were not exhaustive. They extended only to those singularities that can be unambiguously defined by the first three terms in a Taylor series expansion of the matrix equation of closure.

In this paper we take advantage of the much simpler equations associated with the planar 2R manipulator to derive an exhaustive classification of path following singularities for any smooth path passing through a point on the outer workspace boundary. In addition, we determine smooth local models for each case that provide low-order joint rate relations as well. The results show that the topology of local models is closely related to the degree of contact between the endpoint path and the workspace boundary. Section 2 presents the problem formulation. Sections 3 and 4 develop the analytic solutions. Section 5 presents results and their physical implications.

J. Angeles et al. (eds.), Computational Kinematics, 207–216.

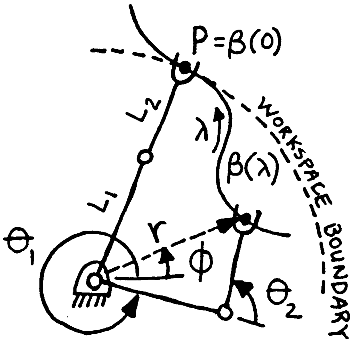


Fig. 1. 2R manipulator with path $\beta(\lambda)$ forcing it into a singular configuration.

2. Problem Formulation

Figure 1 depicts the robotic path tracking problem that we wish to address. We assume that a smooth endpoint path β is given that includes a point P on the outer workspace boundary that forces the manipulator into an *outstretched* singular configuration. The problem is to determine the locus of inverse kinematic solutions (θ_1, θ_2) that maintain the path in the neighborhood of P, as well as their differentials and possible time rates of change. The endpoint path β may intersect the workspace boundary at P or have any degree of tangency with it.

To simplify explanations in this short paper we choose *not* to consider *folded* singularities associated with the *inner* workspace boundary even though the approach can be easily modified to include them with only minor differences in detail. The approach is not so easily extended to include the special singularity that occurs when the workspace void shrinks to a point for 2R manipulators with equal link lengths.

Without loss of generality we choose to represent β in polar coordinates (r,ϕ) as a parametric function of a path parameter λ, i.e., $\beta(\lambda) = [r(\lambda), \phi(\lambda)]$, with $\lambda = 0$ corresponding to point P. To exclude reversals along the path, we also require the parameterization to be regular, i.e.,

$$(r')^2 + (r\phi')^2 > 0 \tag{1}$$

The degree of contact between the endpoint path and the workspace boundary at point P can be determined from the first nonzero coefficient in the Taylor series expansion of $r(\lambda)$ about $\lambda = 0$. Without loss of generality let $\beta(\lambda)$ be represented in a Taylor series as follows.

$$r(\lambda) = (L_1 + L_2) + \frac{r^{(k)}}{k!}\lambda^k + H.O.T \qquad (k \geq 1) \tag{2}$$

$$\phi(\lambda) = \phi^{(0)} + \phi^{(1)}\lambda + H.O.T. \tag{3}$$

Here $r^{(k)}$ denotes r's first nonzero derivative, $\left.\frac{d^k r}{d\lambda^k}\right|_0 \neq 0$. The degree of contact between the endpoint path and the workspace boundary is equal to k: e.g., $k = 1$ implies 1-point contact (intersection), $k = 2$ implies 2-point contact (simple tangency), etc. Requirement (1) for a regular parameterization implies that if $k > 1$, then $\phi^{(1)} \neq 0$.

The following equations relate the endpoint coordinates (r, ϕ) to the joint coordinates (θ_1, θ_2). Equation (4) can be derived using the law of cosines. Equations (5) and (6) follow from simple trigonometry.

$$r^2 = L_1^2 + L_2^2 + 2L_1L_2\cos\theta_2 \tag{4}$$

$$r\cos(\phi - \theta_1) = L_1 + L_2\cos\theta_2 \tag{5}$$

$$r\sin(\phi - \theta_1) = L_2\sin\theta_2 \tag{6}$$

Recalling that r and ϕ are parametric functions of the path parameter λ, our problem is to solve equations (4)-(6) for local relations between λ, θ_1, and θ_2 in the neighborhood of the singular solution, $(\lambda, \theta_1, \theta_2) = (0, \phi^{(0)}, 0)$. Because equation (4) does not involve θ_1, we can solve this problem in two steps: first analyze equation (4) to determine local relations between λ and θ_2, then analyze equations (5) and (6) to locally determine θ_1.

In the neighborhood of any outer workspace boundary singularity, equations (5) and (6) provide the following unique and continuously-differentiable solution for θ_1, considering that $\theta_2 \approx 0$ has been previously determined.

$$\theta_1 = \phi - \arcsin\left(\frac{L_2\sin\theta_2}{r}\right) \tag{7}$$

With this in mind, we replace equations (5) and (6) with equation (7), but defer its use to section 4 where θ_1 will be determined. For the moment we will concentrate on determining the local relation between θ_2 and λ by analyzing the following form of equation (4) that is obtained by substituting Taylor series expansion (2) into equation (4).

$$\cos\theta_2 = 1 + a\frac{r^k}{k!}\lambda^k + H.O.T. \qquad (k \geq 1) \tag{8}$$

$$where : a = \frac{L_1 + L_2}{L_1L_2} \tag{9}$$

3. Local Relations between λ and θ_2

Our objective is now to determine the locus of solutions to (8) in the neighborhood of the singular solution $(\lambda, \theta_2) = (0, 0)$. In addition to determining the relation between finite displacements in θ_2, and λ, we want to determine the relations between differentials in θ_2, and λ that track the path near $(\lambda, \theta_2) = (0, 0)$.

We will see that a variety of forms can arise, including isolated solutions, turning points, nodes, and cusps. The following subsections explore the difficulties of identifying these path-following singularities and representing them with low-order models that are smooth and differentiable.

3.1. Direct Approach

It is straightforward to solve equation (8) for θ_2 and to represent $\theta_2(\lambda)$ by the following double-valued function.

$$\theta_2 = \pm \arccos(1 + a\frac{r^{(k)}}{k!}\lambda^k + H.0.T.) \tag{10}$$

We can classify solutions (10) into three cases based on the fact that the argument of the arccos function must be less than 1 to obtain a physically-meaningful real value for θ_2.

(a) If k is even and $r^{(k)} > 0$, the solution $(\lambda, \theta_2) = (0, 0)$ will be locally isolated.

(b) If k is even and $r^{(k)} < 0$, both functions (10) will be continuous, but not necessarily smooth at $\lambda = 0$. It will be shown that these cases correspond to nodes or hypernodes of degree k/2.

(c) If k is odd, each function (10) will be undefined for either positive or negative values of λ. We will show that these cases correspond to turning points, cusps, and hypercusps for $k = 1$, $k = 3$, and odd $k > 3$, respectively.

These three cases are simple to explain based on the geometry of contact between the endpoint path $\beta(\lambda)$ and the workspace boundary. Figure 2 illustrates a representative example for each case. For case (a), the endpoint path only touches the workspace boundary without entering the workspace. For case (b) the entire endpoint path lies within the workspace, except points that contact the workspace boundary. For case (c) half of the endpoint path lies within the workspace and the other half lies outside. The observations made in the previous paragraph then follow considering that two, one, and zero kinematic solutions exist for the workspace interior, boundary, and exterior, respectively. The two kinematic solutions for the workspace interior correspond to elbow-right and elbow-left configurations of the manipulator.

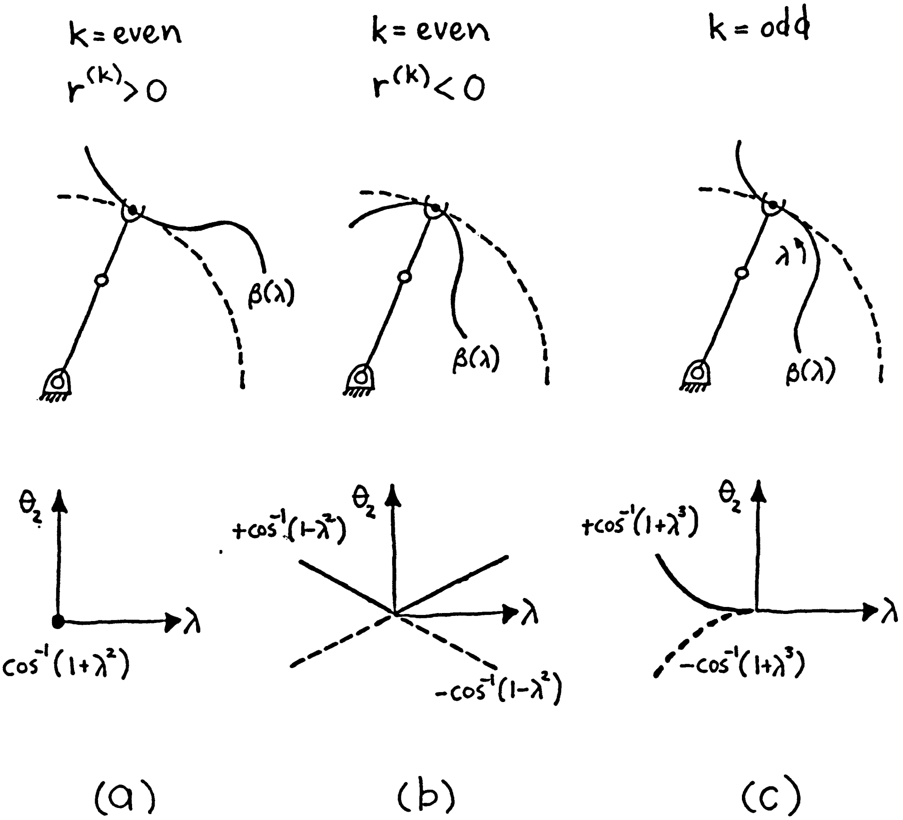


Fig. 2. Representative examples for the three cases.

The plots of Fig. 2 also highlight two difficulties in determining relations between differentials at the singularity. Firstly, for case (b) examples, like the one shown in Fig. 2(b), both functions (10) are continuous, but not smooth at the singularity. Nevertheless it appears that alternate sides of the two functions join smoothly, implying that this is only a problem of representation. In subsection 3.2 we show that such loci of solutions can be represented by two smooth branches that cross at $(\lambda, \theta_2) = (0,0)$ to form a node or hypernode.

The second difficulty relates to case (c) examples such as the cusp shown in Fig. 2(c). Here the two functions (10) terminate at $\lambda = 0$ as a consequence of the path leaving the workspace. Although the robot cannot follow the path out of the workspace, it is possible for the manipulator to change configurations (elbow-left vs. elbow-right) while reversing the direction of path traversal at $\lambda = 0$ and never leaving the path. However, because (10) represents the locus of solutions discontinuously, by two functions that join at $\lambda = 0$, the constraints on performing such a maneuver are unclear. In subsection 3.3 we show how turning points, cusps, and hypercusps can be represented as a smooth parametric function of an introduced parameter s at the singularity. In doing so we clarify the constraints on performing such maneuvers.

3.2. Low-Order Approach for Case (b) Singularities

If we are only interested in the most local representation of solutions near $(\lambda, \theta_2) = (0, 0)$, we can replace equation (8) with the following local equivalent obtained by replacing $\cos\theta_2$ with its Taylor series expansion about $\theta_2 = 0$ and ignoring all but the lowest order terms in θ_2 and λ.

$$\theta_2^2 = -2a\frac{r^{(k)}}{k!}\lambda^k \tag{11}$$

Solving (11) for $\theta_2(\lambda)$ provides the following double-valued function which shares the same local properties as (10).

$$\theta_2(\lambda) = \pm\sqrt{-2a\frac{r^{(k)}}{k!}\lambda^k} \tag{12}$$

In particular, the need for a positive argument to the square root leads to the same classification of singularity types (a)-(c) and the same observations made in subsection 3.1. Furthermore, functions (12) have the same local discontinuities as functions (10) and present the same difficulties in determining relations between differentials. Thus the low-order approximation (12) seems to duplicate the characteristics of functions (10) without offering any advantages.

Nevertheless, we can gain advantage from the low-order approach by considering that, for case (b) singularities (recall $r^{(k)} < 0$, and $k = even$), it is not necessary to represent the locus of solutions in the form of (12). We can instead represent solutions to (11) as follows.

$$\theta_2(\lambda) = \pm\sqrt{-2a\frac{r^{(k)}}{k!}}\lambda^{k/2} \tag{13}$$

Note that (13) does not follow as a simplification of (12) because proper extraction of λ from under that radical would require that λ in (13) be replaced by $|\lambda|$. Nevertheless (13), provides a correct solution that is easy to verify by substitution into (11).

The advantage of representation (13) over (10) is that, for case (b) singularities only, both branches are now represented by smooth and differentiable functions of λ. For the case (b) example shown in Fig. 2(b), the two branches of (13) are linear functions that cross at $\lambda = 0$ to form a simple node. Together, they locally form the same loci of solutions, as before but now in a branchwise-differentiable manner. Similar results hold for case (b) singularities of higher order.

The disadvantage of representation (13) is that it is only a locally-valid approximation to the exact locus of solutions to (8). The following conjecture attempts to quantifying the accuracy of that approximation.

CONJECTURE 1. *For case (b) singularities, functions (13) provide a low-order approximation to the loci of solutions to (8) in the neighborhood of the singularity that are correct up to the orders in λ shown. This means the first k/2 derivatives of functions (13) evaluated at $\lambda = 0$ will agree with those of the exact locus of solutions. Namely:*

$$\left.\frac{d^n\theta_2}{d\lambda^n}\right|_0 = \begin{cases} 0 & if\ n < \frac{k}{2} \\ \pm\frac{k}{2}!\sqrt{\frac{-2ar^{(k)}}{k!}} & if\ n = \frac{k}{2} \end{cases} \tag{14}$$

3.3. Local Parametric Approach for Case (c) Singularities

Recall that for case (c) singularities, the double-branched solutions (10) or (12) join at in a turning point, cusp, or hypercusp at $\lambda = 0$. This implies that the manipulator can pass from one solution branch to the other at $\lambda = 0$, but because the locus of solutions is represented discontinuously, by two functions that join at $\lambda = 0$, the constraints on performing such a maneuver are unclear. In this section we overcome this problem by introducing an auxiliary parameter s to interrelate θ_2 and λ.

The locus of solutions to (11) for case (c) singularities can be represented by the following parametric functions of s.

$$\lambda(s) = -sign(r^{(k)})s^2 \qquad k = odd \tag{15}$$

$$\theta_2(s) = \sqrt{\left|2a\frac{r^{(k)}}{k!}\right|}s^k \qquad k = odd \tag{16}$$

Functions (15) and (16) are not unique because there are many other equally-valid parameterizations of the same curve that differ in speed of parameterization. This parameterization was chosen because it is minimum-order in s and its correctness is easy to verify by substitution into (11).

The advantage of this parametric representation over the double-valued function (10) or (12) is that both branches are now combined into a single smooth and differentiable function of s that implicitly provides the constraints for changing solution branches (elbow-right vs. elbow-left) at the singularity while tracking the endpoint path.

The following conjecture attempts to quantify the accuracy of this low-order representation.

CONJECTURE 2. *For case (c) singularities, parametric functions (15) and (16) provide low-order approximations to the loci of solutions to (8) in the neighborhood of the singularity that are correct up to the orders in s shown.*

4. Local Determination of θ_1

We now consider that local relations between λ and θ_2 have been determined by functions (13) for case (b) singularities, or parametric functions (15) and (16) for case (c) singularities. The problem is now to extend these local models to include θ_1. No such problem exists for case (a) isolated solutions because equation (7) provides a unique value for θ_1 given that θ_2 and λ have been uniquely determined.

Local determination of $\theta_1(\lambda)$ for case (b) singularities can be based on a Taylor series expansion of (7) in terms of λ about $\lambda = 0$. Considering that $r = r(\lambda)$, $\phi = \phi(\lambda)$, and $\theta_2 = \theta_2(\lambda)$ we obtain the following expression.

$$\begin{aligned}\theta_1(\lambda) &= \phi + \left[\phi' - \frac{L_2}{r}\theta_2'\right]\lambda \\ &\quad + \left[\frac{1}{2}\phi'' - \frac{1}{2}\frac{L_2}{r}\theta_2'' + \frac{L_2}{r^2}\theta_2' r'\right]\lambda^2 + H.O.T. \end{aligned}\tag{17}$$

Evaluating all quantities at $\lambda = 0$ using equations (2), (3) and (13) and retaining only the lowest order terms in λ, we obtain the following results that apply to case (b) singularities only.

$$\theta_1(\lambda) = \begin{cases} \phi^{(0)} + \left[\phi^{(1)} \mp b\sqrt{a|r^{(2)}|}\right]\lambda & k = 2 \\ \phi^{(0)} + \phi^{(1)}\lambda & \textit{even } k > 2. \end{cases}\tag{18}$$

$$\textit{Here } a \textit{ is given by } (9) \textit{ and } b = \frac{L_2}{L_1 + L_2}\tag{19}$$

Local determination of $\theta_1(s)$ for case (c) singularities can be based on a Taylor series expansion of (7) in terms of s about the singularity where $s = 0$. Considering that $r = r(\lambda(s))$, $\phi = \phi(\lambda(s))$, and $\theta_2 = \theta_2(s)$ we obtain the following expression.

$$\begin{aligned}\theta_1(s) &= \phi + \left[\phi'\lambda' - \frac{L_2}{r}\theta_2'\right]s \\ &\quad + \left[\frac{1}{2}\phi'\lambda'' + \frac{1}{2}\phi''(\lambda')^2 - \frac{1}{2}\frac{L_2}{r}\theta_2'' + \frac{L_2}{r^2}\theta_2' r'\lambda'\right]s^2 + H.O.T. \end{aligned}\tag{20}$$

Evaluating all quantities at the singularity using equations (2), (3), (15), and (16) and retaining only the lowest order terms in s, we obtain the following results which apply to case (c) singularities only.

$$\theta_1(s) = \begin{cases} \phi^{(0)} - b\sqrt{2a|r^{(1)}|}\ s & k = 1 \\ \phi^{(0)} - \left[\phi^{(1)} sgn(r^{(k)})\right] s^2 & odd\ k > 1. \end{cases} \tag{21}$$

It is worth noting that the regularity condition (1) guarantees that $\phi^{(1)} \neq 0$ for low-order solutions (18) and (21) if $k > 1$. The following conjecture attempts to quantify the accuracy of these low-order representations.

CONJECTURE 3. *For case (b) and (c) singularities, respectively, functions (18) and (21) provide low-order approximations to the loci of solutions to (7) in the neighborhood of the singularity that are correct up to the orders in λ or s shown.*

5. Results and Physical Interpretation

Recall that our objective is to determine the locus of inverse kinematic solutions (θ_1, θ_2) that track a smooth path $\beta(\lambda)$ in the neighborhood of a point P on the outer workspace boundary. In addition we want to determine the relations between joint rates for real motions that track the path at P.

Our results so far have been to classify all singularities of this path-tracking problem into three cases (a),(b), and (c) based on the degree of contact, k, between the path and the workspace boundary, using the sign of $r^{(k)} = \frac{d^k r}{d\lambda^k}\Big|_0 \neq 0$, to distinguish between cases (a) and (b). For each case, we have also developed appropriate low-order representations of the locus of inverse kinematic solutions at the singularity that are smooth and locally accurate (assuming conjectures 1-3 are correct). Our final objective is now to show how these results can be interpreted.

The most obvious results are the dramatic differences between the three cases: case (a) singularities allow no motion whatsoever, case (b) singularities allows complete tracking of the path using either one of two smooth branches of inverse kinematic solutions as well as the possibility of switching between these branches at the singularity, and case (c) singularities allow path tracking on the workspace side of point P using either one of two inverse kinematic solution branches that join at point P.

More subtle results apply to the differences between members of each case, excluding case (a), whose members are all simply isolated points. For case (b) members, we make the following observations.

(i) all case (b) members represent nodes or hypernodes composed of two smooth branches that contact each other with degree of contact equal to $\frac{k}{2}$ —- e.g., members $k = 2$, $k = 4$, and $k = 6$ correspond to simple nodes, tachnodes, and flecnodes respectively.

(ii) if $\frac{k}{2}$ is odd (even) each smooth solution branch results in (does not result in) a change in the manipulator configuration (elbow-right vs. elbow-left) as the singularity is passed.

(iii) switching between smooth solution branches at the singularity implies a discontinuity at the level of the $\frac{k}{2}th$ derivative.

With respect to case (c) members, we make the following observations.

(i) member $k = 1$ represents a turning point and higher-order members represent cusps or hypercusps —- e.g., $k = 3$ corresponds to a simple cusp and $k = 5$ corresponds to a ramphoid cusp.

(ii) joint rates become unbounded for member $k = 1$ (turning point) as the singularity is approached with unit speed in the path parameter λ. However this does not occur for higher-order members (cusps and hypercusps).

Finally we make a general but powerful observation that can be applied to either case (b) or case (c) singularities: the constraints and freedoms in timing joint trajectories that track the path (in particular allowable joint velocities, accelerations, etc. at the singularity) can be determined by considering the free parameter, λ (case b), or s (case c) in the appropriate low-order model to be an arbitrary function of time. Chain rule evaluation of the derivatives of the model with respect to time will then provide information about allowable joint rates at the singularity. Care must be taken, however, not to infer too much from the low-order models. In general they will only define the first non-zero derivatives for θ_1, θ_2, and λ

References

Kieffer J., 1992, "Manipulator Inverse Kinematics for Untimed End-Effector Trajectories with Ordinary Singularities," *International Journal of Robotics Research*, Vol.11 No.3, pp. 225-237.

Kieffer J., 1993, "Differential Analysis of Bifurcations and Isolated Singularities for Robots and Mechanisms," *IEEE Transactions on Robotics and Automation*, (to appear).

ROBOT MOTIONS WITH TRAJECTORY INTERPOLATION AND OVERCORRECTION

H. Heiß, BMW, ET-203, 80788 München, Germany

Abstract This paper makes proposals for trajectory interpolation in joint space and in cartesian space regarding as well orientation as position and enlarging the application bandwidth; furthermore it shows new methods for connecting path segments in a smooth manner, which take into account not only velocity but also acceleration values and improve the facility of robot programming by user-defined starting and end points of the smoothing interval.

Keywords robot motion computation; trajectory interpolation; interpolation of position and orientation; path smoothing; overcorrection.

PRESENT SITUATION AND OBJECTIVES

From the point of view of the user two areas stand out as worthy of improvement in the field of robot motions with trajectory interpolation and overcorrection:

1) performance of the robot control in path smoothing and trajectory interpolation

2) clarity of these two functions in the robot control manual

Apart from such obvious things like the failure of overcorrection when changing the tool or base values, the first point is very difficult for the user to check. This is especially the case if there is a lack of documentation, when very costly measurements have to be taken to establish the principles inside the robot control. Such a task is not the responsibility of the user, nor is it relevant for the user to know what future developments are being worked on in the labs of universities and producers of robot controls; in practice only the functions of an available robot control count. This paper is based on the author's experience of currently available robot controls, whereby the overriding impression is one of a performance deficiency. Whether this deficiency is based on slow processors, or on non-observance of special conditions, or on inadequate design and implementation inside the control remains an unanswered question for the user; moreover it is a question which is largely irrelevant.

J. Angeles et al. (eds.), Computational Kinematics, 217–228.

Taking user wants and needs with regard to robot controls and their manuals as a basis, this paper will present a framework which consists of the following functions and their description:

- equivalent use of interpolation of orientation and position
- circular interpolation up to 360 degrees
- clear and simple handling of parts and integration of external axes into the programming of robots by extending the interpolation concept over several areas
- mathematical base for dealing with the path smoothing calculations
- maintaining a smooth position, velocity and acceleration curve
- user-determined starting point and length of the smoothing interval and ability to make contact with all the path definition points
- joint interpolated motion with integrated overcorrection

TRAJECTORY INTERPOLATION IN JOINT SPACE

Joint calculation

Cartesian target values (including sensor correction and other modifications) have to transform into joint values; for this purpose the following main equations are used:

robot base equation: base•robot kinematics•tool = aim

joint base equation: robot kinematics = base^{-1}•aim•tool^{-1}

In order to simplify the joint calculation further modifications of the joint base equation can be made [Heiß, 1986]. This calculation results in a suitable set of joint values.

(time coordinated) point-to-point movement

Velocity profile with linear acceleration and deceleration ramp

The joint movements of a robot designed to overcome the joint difference between two targets are initiated from within the robot control and are limited by the maximum velocity and the maximum acceleration of the single joint. In most cases the velocity profile of a joint movement looks like a trapezium and is defined by the constant maximum acceleration during the acceleration and deceleration phases and by a constant velocity between them. It is essentially the overall motion time T and the appropriate velocity of each single joint which have to be calculated.

Start-stop mode with smooth acceleration behaviour

The above-mentioned method gives rise to two disadvantages:

1) peaks in the acceleration

2) The movement can only be defined stage-by-stage due to the trapezium form

A smooth movement taking into account both position, and velocity and acceleration, in the starting and aim pose can be defined by a fifth degree polynomial:

$$p(t) = k5*h^5+k4*h^4+k3*h^3+k2*h^2+k1*h+k0 \quad \text{with} \quad h=t/T \qquad 0 \le t \le T$$

$$k0 = p1, \quad k1 = T*v1, \quad k2 = T^2*a1/2,$$

$$k3 = 10*(p2-p1)-T*(6*v1+4*v2)+T^2*(a2-3*a1)/2$$

$$k4 = -15*(p2-p1)+T*(8*v1+7*v2)-T^2*(a2-3*a1/2)$$

$$k5 = 6*(p2-p1)-T*3*(v1+v2)+T^2*(a2-a1)/2$$

p1, v1, a1, p2, v2 and a2 are the position, velocity and acceleration values in the starting and aim pose.
As the start-stop mode has to be used, it results in the following: $v1 = v2 = a1 = a2 = 0$.

As in the case of trapezium shaped velocity, the overall motion time has to be calculated according to its dependence on the upper limit of the joint velocity and joint acceleration values: the extreme acceleration value at $h=\frac{3\pm\sqrt{3}}{6}$ with $\mp\frac{10*(p2-p1)}{T^2*\sqrt{3}} \Rightarrow \sqrt{\frac{10*|p2-p1|}{A_i*\sqrt{3}}} \le T$
The extreme velocity values lie at h=0 and h=1, per definition always with a value equal to 0, and at h=0.5 with a value of 1.875*(p2-p1)/T; this results in $\frac{1.875*(p2-p1)}{maxG_i} \le T$

(time coordinated) point-to-point movement with acceleration-oriented path smoothing

Because the polynomial p(t) is an overcorrection polynomial and because no path forms have to be adhered to for point-to-point movements, it appears reasonable to integrate the overcorrection into the trajectory interpolation. For this way all joint positions can be attained, although the velocity and the acceleration do not always decrease to zero at the end of a motion segment.

$vmax_i = 1.875*(p2-p1)/T$ is the extreme velocity value of the current segment,
$vmaxf_i = 1.875*(p3-p2)/Tf$ is the value of the following segment.
With regard to acceleration, the second extreme value of the current segment at $h=(3+\sqrt{3})/6$ $amax_i=\frac{-10*(p2-p1)}{T^2*\sqrt{3}}$ corresponds to the first value of the next segment $amaxf_i=\frac{10*(p3-p2)}{Tf^2*\sqrt{3}}$.

For v2 and a2 of the current segment (and the analogous values v1 and a1 of the following segment) this results in:

$$v2 = \begin{cases} 0 & \text{if } signum(vmax_i) \neq signum(vmaxf_i) \\ \min\{vmax_i, vmaxf_i\} & \text{if } vmax_i \ge 0 \text{ and } vmaxf_i \ge 0 \\ \max\{vmax_i, vmaxf_i\} & \text{if } vmax_i < 0 \text{ and } vmaxf_i < 0 \end{cases}$$

$$a2 = \begin{cases} 0 & \text{if } signum(amax_i) \neq signum(amaxf_i) \\ \min\{amax_i, amaxf_i\} & \text{if } amax_i \ge 0 \text{ and } amaxf_i \ge 0 \\ \max\{amax_i, amaxf_i\} & \text{if } amax_i < 0 \text{ and } amaxf_i < 0 \end{cases}$$

TRAJECTORY INTERPOLATION IN CARTESIAN SPACE

Cartesian trajectory interpolation between starting and aim should not carried out with the two robot flange points "robot kinematics$_{start}$" = base$_{start}^{-1}$•start•tool$_{start}^{-1}$ and "robot kinematics$_{aim}$" = base$_{aim}^{-1}$•aim•tool$_{aim}^{-1}$; instead, it should be dependent on the start and the aim in order to maintain the given shape and not to swing out.

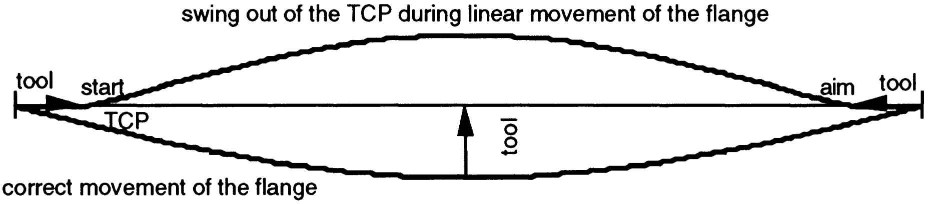

swing out of the TCP during linear movement of the flange
tool
start
TCP
tool
aim
tool
correct movement of the flange

Figure 1: Deviation during flange interpolation

Trajectory interpolation therefore requires knowledge about the last aim pose, even in case of overcorrection.

interpolation curve IC(t) = f(start, aim)   robot kinematics(t) = base^{-1}•IC(t)•tool^{-1}

In the draft version of DIN 66312 part 2 methods of interpolation will be described as defined by producers of robot control, institutes or users.

Three areas of interpolation

If a change in the base or tool values is permitted during the robot movement for the purpose of modelling external axes or part handling, then the interpolation has to be applied to all three areas of change. If this is not the case, the robot will leave the given path shape or move jerkily.

Interpolation of aim defines the movement of the tool by the robot (tool handling), interpolation of tool values allows definition of a path on a part, and interpolation of changed base values integrates external axes into the programming of the robot. In the event that the robot is moved by an external axis, the definition of base is trivial; otherwise, if a tool is connected to the external axis, the base values have to be calculated as follows:

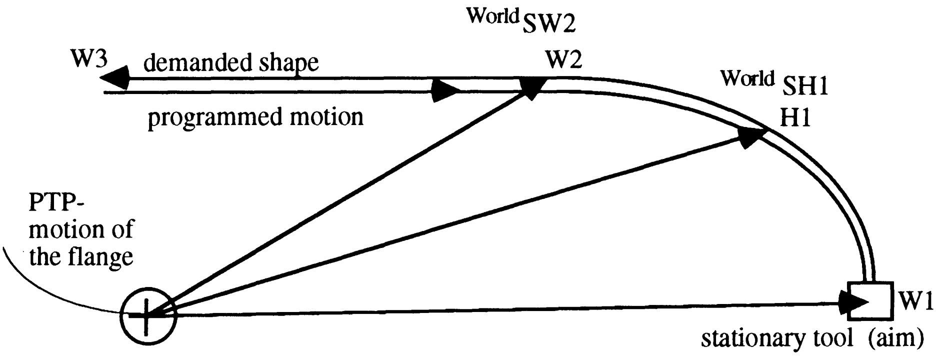
$$\text{base} = {}^{\text{extAxissystem}}\text{axisdrive}^{-1} \bullet {}^{\text{Worldsystem}}\text{extAxissystem}^{-1} \bullet {}^{\text{Worldsystem}}\text{Robotbase}$$

base(t) = f$_b$(base$_{start}$, base$_{aim}$)   interpolation curve IC(t) = f(start, aim)

tool(t) = f$_t$(tool$_{start}$, tool$_{aim}$)   robot kinematics(t) = base(t)$^{-1}$•IC(t)•tool(t)$^{-1}$

In order to interpolate base and tool the robot control has to have data on the last base and tool values; moreover a third auxiliary position is needed for circular interpolation. "base" uses the world system as its reference system, "tool" the flange; the motion time T should be given, or possible to calculate.

The diagram below provides an example of the advantages of interpolating in three different areas:



The task is: to work on the part moved by robot in a circle from W1 via H1 to W2 and linearly from W2 to W3.

Transforming the task into normal robot programming terms, we obtain the following:

WKZ = W1; move PTP aim; activate (stationary) tool;

WKZ = W2; move CIRC aim via $^{World}SH1$;

WKZ = W3; move LIN aim;

This procedure gives rise to the following problems:

A) Extensive calculation by the robot programmer is necessary before the point SH1 can be transformed into world coordinates.

B) Activating the new WKZ during running interpolation calculations (LIN or CIRC) will create a jerk in joints.

C) Smoothing between two path segments and pre-planning of the desired path shape requires modifications within the robot control
(methods to deal with the different WKZ values or strictly sequential operations in start-stop mode; it should be possible to calculate the flange pose F from "move PTP aim" with aim•$W1^{-1}$ at the end of PTP-motion and then to compute $^{World}SW2$ with F•W2. The same is true for $^{World}SH1$, for which only the relative transformation "Flange–H1" is known, but which needs global values (see A)).

In order to overcome these problems, interpolation should be carried out separately in the three areas "base", "aim" and "tool". Moreover, the task-oriented view remains valid, and the range of the available path shapes increases considerably due to these 3 interpolations.

Interpolation in position space

Linear

$x(t)=x_{start}+(x_{aim} - x_{start})*t/T$; $y(t)=y_{start}+(y_{aim} - y_{start})*t/T$; $z(t)=z_{start}+(z_{aim} - z_{start})*t/T$

Circular

The three circle points are a (start of the circle segment), b (auxiliary point for circle definition) and c (aim of the circle segment); m is the centre of the circle and M the centre system in world coordinates with an x-axis=(a-m)/||a-m|| and z-axis=perpendicular to the circle plane. Hence, the following must be valid: ${}^{M}a_z = {}^{M}b_z = {}^{M}c_z = 0$

radius r = ||a-m||, α is the extent of the circle segment, and circular interpolation results in

${}^{M}x(t) = r*\cos(\alpha*t/T)$ ${}^{M}y(t) = r*\sin(\alpha*t/T)$ ${}^{M}z(t) = 0$ $0 \leq t \leq T$

The time function of the circular interpolation is:

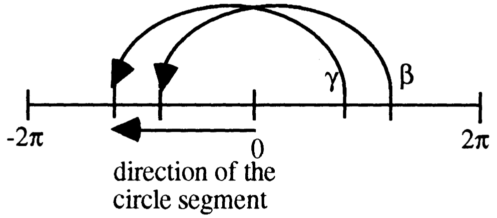
$x(t) = M_{11}*r*\cos(\alpha*t/T) + M_{12}*r*\sin(\alpha*t/T) + M_{14}$

$y(t) = M_{21}*r*\cos(\alpha*t/T) + M_{22}*r*\sin(\alpha*t/T) + M_{24}$

$z(t) = M_{31}*r*\cos(\alpha*t/T) + M_{32}*r*\sin(\alpha*t/T) + M_{34}$ $0 \leq t \leq T$

As a result it becomes clear that path definition using discrete point is more complex and time-consuming than using time functions.
In most current robot controls the extent of the circle has an upper limit of 180°. This limitation can be overcome by dealing with each different case separately.

${}^{M}b = M^{-1}\bullet b$; ${}^{M}c = M^{-1}\bullet c$;
$\beta = ATAN2({}^{M}b_y , {}^{M}b_x) \in]-2\pi, 2\pi[$, $\gamma = ATAN2({}^{M}c_y , {}^{M}c_x) \in]-2\pi, 2\pi[$,



```
if $\gamma=0$,
then $\alpha:=2\pi$
else    if signum($\beta$)*signum($\gamma$)>0
        then    if $\gamma>0$
                then    if $\beta<\gamma$, then $\alpha:=\gamma$    else $\alpha:=\gamma-2\pi$
                else    if $\beta>\gamma$, then $\alpha:=\gamma$    else $\alpha:=\gamma+2\pi$
        else    if $\gamma>0$
                then    if $\beta+2\pi<\gamma$, then $\alpha:=\gamma$    else $\alpha:=\gamma-2\pi$
                else    if $\beta-2\pi>\gamma$, then $\alpha:=\gamma$    else $\alpha:=\gamma+2\pi$
```

Interpolation in orientation space

A is the orientation at the beginning of the interpolation phase, E the orientation at the end and Ü is the 3x3-matrix defined by $Ü = A^{-1} \bullet E$, namely a "transition matrix".

Two-Angle-Interpolation

Expressed in Eulerian notation (α, β, γ), Ü is $Rot(z,\alpha) \bullet Rot(x,\beta) \bullet Rot(z,\gamma)$ with

$$\alpha = ATAN2(-Ü_{13}, Ü_{23}) \{+\pi\}; \qquad \text{if } Ü_{13} = Ü_{23} = 0, \text{ then } \alpha = 0.$$

$$\beta = ATAN2(-\sqrt{1-Ü_{33}^2}, Ü_{33}) \{*-1\}$$

$$\gamma = ATAN2(-Ü_{31}, -Ü_{32}) \{+\pi\};$$

$$\text{if } Ü_{13} = Ü_{23} = 0, \text{ then } \gamma = ATAN2(Ü_{21}, Ü_{11})*Ü_{33}$$

Linear interpolation takes place in the case of angles β and γ; in order to avoid the jerk "$A \rightarrow A \bullet Rot(z,\alpha)$" caused by α at the beginning of the interpolation process, the interpolation procedure $A \bullet Rot(z,\alpha) \bullet Rot(x,\beta*t/T) \bullet Rot(z,\gamma*t/T)$ is modified according to [Paul, 81]:

$$InterpolationOr(t) = A \bullet Rot(z,\alpha) \bullet Rot(x,\beta*t/T) \bullet Rot(z,-\alpha) \bullet Rot(z,(\alpha+\gamma)*t/T) \quad \alpha+\gamma \in]-\pi, \pi]$$

A result of this is that the problem of ambiguity of (α, β, γ) is solved; there is a difference between interpolation in $(\alpha, \beta*t/T, \gamma*t/T)$ and in $(\alpha+\pi, -\beta*t/T, (\gamma+\pi)*t/T)$, but not in $(\alpha\{+\pi\}, \pm\beta*t/T, (\alpha+\gamma\{+2\pi\})*t/T)$
Many other interpolation procedures are conceivable, but their practical use seems questionable.

Fusion of the three interpolation areas

The orientation interpolation function and the position interpolation function are collected in all three interpolation areas together in a homogeneous matrix H; the joint base equation has to be calculated according to "robot kinematics(t) = $Hbase(t)^{-1} \bullet HIC(t) \bullet Htool(t)^{-1}$".

Suitable data structure

The calculation of overcorrection cannot take into account conditions such as orthogonality or normality, which means that redundant data structures, e.g. homogeneous matrix, dual matrix [Heiß, 86b] or quaternion, cannot be applied.

Non-redundant structures are the basis of the calculation:

- n-dimensional joint related time function (resulting from interpolation in joint space)
- six-dimensional cartesian time function (from interpolation in cartesian space)

An example of the difficulties arising from the use of redundant structures is:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \ddot{U} = \mathrm{Rot}(z,90°)$$

Linear overcorrection (without consideration of velocity and acceleration) results in the following orientation process:

$$\begin{bmatrix} 1\text{-}1{*}t/T & 0\text{-}1{*}t/T & 0 \\ 1{*}t/T & 1\text{-}1{*}t/T & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

t=T/2 results in $\begin{bmatrix} 0{,}5 & -0{,}5 & 0 \\ 0{,}5 & 0{,}5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, and this violates the unit restriction.

Description of pose with joint coordinates

The problem of potential redundancy resulting from more than six active joints is already solved in the process of inverse kinematic computation, meaning that the functions $g_1(t)$, ..., $g_n(t)$ are not subject to any restrictions and can be used without any problem for the calculation of overcorrection.

Description of pose by a sixtuple (x(t), y(t), z(t), α(t), β(t), γ(t))

x(t), y(t) and z(t) result directly from the fourth column of the homogeneous matrix "robot kinematics(t)". Using the 3x3 orientation matrix Or(t), which is part of "robot kinematics", (α, β, γ) can be calculated in Eulerian notation as $Or(t)=Rot(z,\alpha)\bullet Rot(x,\beta)\bullet Rot(z,\gamma)$:

$\alpha(t) = \mathrm{ATAN2}(-Or_{13}(t)\,,\, Or_{23}(t)\,)\ \{+\pi\}$;   if $Or_{13}(t) = Or_{23}(t) = 0$, then $\alpha(t)$ should be continued continuously with $\arctan(\lim\limits_{t \to to} \frac{-Or_{13}(t)}{Or_{23}(t)}) + (1 - \lim\limits_{t \to to}(signum(Or_{23}(t)))) * \frac{\pi}{2}$

$\beta(t) = \mathrm{ATAN2}(-\sqrt{1\text{-}Or_{33}^{2}(t)}\,,\, Or_{33}(t))\ \{*\text{-}1\}$

$\gamma(t) = \mathrm{ATAN2}(-Or_{31}(t)\,,\, -Or_{32}(t)\,)\ \{+\pi\}$;

if $Or_{13}(t) = Or_{23}(t) = 0$, then $\gamma(t) = (\mathrm{ATAN2}(Or_{21}(t),Or_{11}(t)) - \alpha(t)){*}Or_{33}(t)$

Due to this computation with two result values, care has to be taken to ensure that the function values of $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ at the beginning of the second trajectory segment are close to the values of $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ at the end of the first trajectory in order to avoid extensive movements during the overcorrection phase.

Overcorrection by means of a polynomial

Degree and coefficients of a universal overcorrection polynomial

With respect to position, velocity and acceleration of the accompanying trajectory segments at the beginning and end of the overcorrection phase, there are six conditions which must be fulfilled, and thus a five degree polynomial results:

$$p(t) = k5*h^5+k4*h^4+k3*h^3+k2*h^2+k1*h+k0 \qquad \text{with } h=t/T \qquad 0 \leq t \leq T$$

With p1, v1, a1, p2, v2 and a2 as the position, velocity and acceleration values of the two neighbouring segments at the start and end of the overcorrection phase, the coefficients of p(t) can be defined as follows:

$$k0 = p1, \quad k1 = T*v1, \quad k2 = T^2*a1/2,$$
$$k3 = 10*(p2-p1)-T*(6*v1+4*v2)+T^2*(a2-3*a1)/2,$$
$$k4 = -15*(p2-p1)+T*(8*v1+7*v2)-T^2*(a2-3*a1/2),$$
$$k5 = 6*(p2-p1)-T*3*(v1+v2)+T^2*(a2-a1)/2$$

The overcorrection polynomial p(t) is universal, meaning that there are no restrictions on the shape of the two trajectory segments, which should be connected. Should there be a symmetrical overcorrection between two linear trajectory segments, then a1=a2=0 and p2=p1+v1*T/2+v2*T/2 and the polynomial decreases to degree four:

$$k0 = p1, \quad k1 = T*v1, \quad k2 = 0,$$
$$k3 = 10*(p2-p1)-T*(6*v1+4*v2),$$
$$k4 = -15*(p2-p1)+T*(8*v1+7*v2)$$

However, these savings in computation time and effort do not justify being treated as a special case.

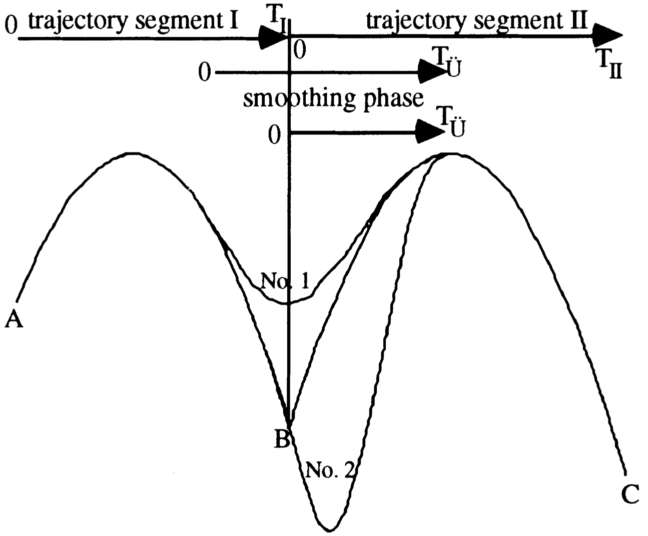
The idea of three interpolation areas makes a wide variety of trajectory shapes possible, but it prevents an A•D(r)•W description of the trajectory shape based on linear interpolation. This idea, using the sixtuple $D(r)=(x(t),y(t),z(t),\alpha(t),\beta(t),\gamma(t))$ to describe the trajectory shape was mentioned in [Paul, 81] and is a relative method. It refers to the starting pose A and is applicable only for linear interpolation, not for circular movements. Moreover neither tool changes nor part handling are possible with this method. As a result, it is not suitable for a universal approach.

Application of the overcorrection polynomial

The polynomial p(t) has to be calculated for each joint function $g_i(t)$ or each cartesian function x(t), y(t), z(t), $\alpha(t)$, $\beta(t)$, $\gamma(t)$, and thus for each interpolation mode there must be a function in the robot control which provides the matrices H(t), $\dot{H}(t)$ and $\ddot{H}(t)$ in relation to time t and the values of starting and aim (in the case of linear interpolation), or a, b and c (in the case of circular interpolation) or Ü (in the case of orientation interpolation).

The starting and end points of the smoothing phase – as well as the length – can be chosen arbitrarily. The best way to establish these is by using time based functions since position, velocity and acceleration functions can be easily calculated using this form of parameterisation.

Two examples of overcorrection curves:



The wide scale of application of the overcorrection technique enables producers of robot controls, to offer many functions related to trajectory smoothing, e.g.:

- start of the overcorrection procedure only when the required pose is attained; this results in an exact movement through all the required poses.
- use of distance values to define the beginning and end of the overcorrection phase
- constant length of the overcorrection phase

The movements at the beginning and end of the trajectory can be integrated just as easily into the method: at the beginning the overcorrection starts with both the first trajectory segment and with p1=start, v1=0 and a1=0; the length of the adaptation phase must be defined by the user; at the end of the trajectory deceleration begins under user control and finishes at the scheduled time with p2=end, v2=0 and a2=0.

It is very difficult for the user to obtain a spatial image of the trajectory. Therefore, overcorrection should be reduced to joint coordinates, meaning that neither inverse kinematic computation nor transformation into sixtuples will be necessary during overcorrection. Only at the start and end of the overcorrection phase the pose, velocity and acceleration have to be determined (in form of 4x4 matrices) and calculated in joint values, joint veloci-

ties and joint accelerations.
The differential of the joint related overcorrection polynomial can be described in a formal manner, thus reducing the time and effort needed for calculation.

VARIABLE OVERRIDE FACTOR AND SHAPE CONSTANCE

An important aspect of trajectory planning is to obtain the same trajectory shape under different override values.

The following statement holds true for time interpolated trajectory functions F(t) and their velocity function v(t) and acceleration function a(t):

The override value ov $\in$]0,...] is a factor of the time parameter t. The trajectory shape F(t*ov) is the same shape, but the motion time will, of course, vary; velocity behaviour ov*v(t*ov) changes as expected by the factor ov, and analogous statements are valid for acceleration $ov^2 * a(t*ov)$.

From this it becomes clear that despite the use of discrete poses to define the trajectory, continuous functions (preferably based on time parameters) have to be utilized inside the robot control to fulfil the requirements.

The question of shape behaviour during overcorrection is more interesting and a little more complex. Working on the basis that users want the same overcorrection shape in spite of different motion time, velocity and acceleration, it is important that shape constance is ensured. This can be achieved by the following steps inside the robot control:

Overcorrection phase starts at $t=ts_{old}/ov$ and lasts T/ov. The overcorrection polynomial p(t) has the form $p(t) = k5*h^5+k4*h^4+k3*h^3+k2*h^2+k1*h+k0$
with the coefficients calculated originally and a new parameter $h=t*ov/T \quad 0 \leq t \leq T/ov$

The form of the overcorrection trajectory thus remains the same.
Furthermore, $\frac{dp}{dt}(t) = \dot{p}(t)*ov/T$ shows with $\frac{dp}{dt}(0)=v1*ov$, $\frac{dp}{dt}(1)=v2*ov$, $\frac{dp}{dt^2}(0)=a1*ov^2$ and $\frac{dp}{dt^2}(1)=a2*ov^2$ that the correct velocity and acceleration values exist at the connection points between trajectory segments and overcorrection path.

CONCLUSION

The above-described advance from discrete description forms to time functions as well as the integration of velocity and acceleration functions open the way to a unified, but nevertheless variable overcorrection method.

Even if acceleration is ignored in order to cut calculation time, the method still remains valid, the polynomial simply reducing to degree three with new coefficients:

k2 = 3*(p2-p1)-T*(2*v1+v2)   k3 = -2*(p2-p1)+T*(v1+v2)

A procedure was developed for PTP movements which combines overcorrection with time-controlled, steady motion behaviour.

The idea of three interpolation areas "basis", "aim" and "tool" increases the range of available path shapes considerably and simplifies programming substantially. In order to simplify robot programming for the user, care was taken to establish a definition of circular interpolation which can deal with circles less than 360°.

Literatur

Heiß, H. (1986a). Konstruktionskriterien und Lösungsverfahren für Industrieroboter mit explizit lösbarer kinematischer Gleichung. *Robotersysteme*, **2**, 129-137.

Heiß, H. (1986b). Homogeneous and dual matrices for treating the kinematic problem of robots, *Symp. on Theory of Robots*, 13-17.

Paul, R. P. (1981). Robot manipulators: mathematics, programming, and control. *MIT Press*.

COMPUTATIONAL GEOMETRY AND MOTION APPROXIMATION

Q. J. Ge
Assistant Professor
Department of Mechanical Engineering
State University of New York at Stony Brook
New York, NY 11767, U.S.A.

B. Ravani
Professor
Department of Mechanical and Aeronautical Engineering
University of California at Davis
Davis, CA 95616, U.S.A.

ABSTRACT

This paper develops a geometric construction algorithm for designing a second order geometrically (G^2) continuous motion. It combines results in kinematics with the notion of geometric continuity from the field of Computer Aided Geometric Design and develops geometric conditions for piecing two motion segments smoothly. A complete algorithm is presented for constructing a G^2 continuous piecewise Bézier type motion. The results have applications in mechanical systems animation, computer vision, robot trajectory planning and key framing in computer graphics.

INTRODUCTION

This paper deals with design and modeling of motions of a three dimensional object for Computer Aided Animation. Motion approximation involves finding a smooth motion of an object that approximates a given set of configurations[1] (or key configurations) of the object. If the motion allows the object to pass through the key configurations, it is said that the motion interpolates these key configurations. The purpose of this paper is to develop a geometric construction algorithm for synthesizing or designing a smooth desired motion by adjusting the key configurations. Such a motion approximation method can be used in mechanical systems animation.

The traditional approach for computer animation of three dimensional objects has separated interpolations of translations and rotations, see Reeves (1981), Shoemake (1985), Duff(1986), and Pletinckx (1989). Recently, Ge and Ravani (1993a) extended the work of Shoemake, who used unit quaternions for animating rotations, and developed an analytical method for designing complete motion interpolants (including both translations and rotations), which properly took into account the geometry of the underlying space. They used a kinematic mapping (see Ravani and Roth 1984) to establish a geometric foundation for studying motion interpolation and approximation problems. They made the mapping orientable to capture topological considerations

[1] positions and orientations

J. Angeles et al. (eds.), Computational Kinematics, 229–238.

and used the mapping to transform the problem of motion interpolation into that of interpolating points in a special projective three-space called the *image space* of the mapping. In this manner, a piecewise parametric motion was represented by curve segments in the image space. Continuity conditions for piecing motion segments then correspond to continuity conditions for the corresponding curve segments in the image space. They then developed a method for generating a piecewise cubic Hermite type motion interpolant such that its image curve has curvature and torsion continuities. Furthermore, Ge and Ravani (1993b) developed deCasteljau-like geometric construction algorithms for generating motions based on repeated screw motion interpolation.

The present paper extends the work of Ge and Ravani (1993b) and seeks to develop smooth *composite Bézier type motions* that can be used to design or model more complex trajectories of a rigid body. The problem of achieving second order geometric continuity or G^2 continuity in piecing motion segments together is studied geometrically, taking advantage of the notion of geometric continuity in Computer Aided Geometric Design (CAGD). This results in a geometric algorithm for constructing G^2 continuous spline motions that has similarities to the Farin (1993) construction of G^2 spline curves.

The organization of the paper is as follows. First a review of the orientable image space is given as a geometric representation of oriented screw displacements. This is followed by a brief discussion on screw motion interpolants and Bézier type motions. Section 2 addresses geometric continuity of motion segments. Section 3 develops a new geometric construction method for designing G^2 continuous motions. The material in section 2 and 3 are the new contributions presented in this paper.

1 THE ORIENTABLE IMAGE SPACE

The orientable image space (see Ge and Ravani, 1993a and 1993b) is a mathematical space (Ravani and Roth, 1984) each point of which represents an oriented screw displacement. An oriented screw displacement in physical space (denoted as P) is a rotation about and a translation along a directed line in P called the (directed) screw axis. Two screw displacements are considered to be "oppositely oriented" if their screw axes occupy the same position in P but with opposite sense of direction. They may be called the "forward" screw displacement and the "backward" screw displacement, respectively.

A general displacement is geometrically equivalent to a pair of two oppositely oriented screw displacements. They can be represented by two sets of oppositely signed *dual Euler parameters* $\hat{\mathbf{X}} = (\hat{X}_1, \hat{X}_2, \hat{X}_3, \hat{X}_4)$ and $-\hat{\mathbf{X}} = (-\hat{X}_1, -\hat{X}_2, -\hat{X}_3, -\hat{X}_4)$ with $\hat{X}_i = X_i + \epsilon X_i^0$ $(i = 1, 2, 3, 4)$ where X_i are the Euler parameters of rotation and X_i^0 are defined in terms of the vector of translation $\mathbf{d} = (d_1, d_2, d_3)$ as

$$\begin{bmatrix} X_1^0 \\ X_2^0 \\ X_3^0 \\ X_4^0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -d_3 & d_2 & d_1 \\ d_3 & 0 & -d_1 & d_2 \\ -d_2 & d_1 & 0 & d_3 \\ -d_1 & -d_2 & -d_3 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}. \tag{1}$$

The symbol ϵ denotes the dual-number unit with the property $\epsilon^2 = 0$. Details on dual numbers and the Euler parameters can be found in Bottema and Roth (1979).

Only six of the eight components of the dual Euler parmaters are independent, for they satisfy the relations

$$\begin{aligned} &X_1^2 + X_2^2 + X_3^2 + X_4^2 = 1, \\ &X_1X_1^0 + X_2X_2^0 + X_3X_3^0 + X_4X_4^0 = 0. \end{aligned} \tag{2}$$

Therefore, the dual Euler parameters $\hat{\mathbf{X}}' = (\hat{X}_1, \hat{X}_2, \hat{X}_3, \hat{X}_4)$ may be used as a set of *signed* homogeneous dual-number coordinates to define a geometric mapping of oriented screw displacements into oriented points in an orientable projective space with three dual dimensions. This space, denoted by Σ, is called the *orientable image space* of spatial displacements. Let $\hat{w}$ denote a nonpure dual number $w + \epsilon w^0$ where $w \neq 0$. By signed homogeneous coordinates, we mean that the coordinates $\hat{\mathbf{X}}$ and $\hat{w}\hat{\mathbf{X}}$ represent one and the same point in the image space if $w > 0$; and they represent two oppositely oriented (or *antipodal*) points if $w < 0$. In this way a general displacement, which can be achieved either by a forward or a backward screw displacement, can be represented by either one of the two corresponding oppositely oriented image points.

Specialists in projective geometry will notice that geometry of the orientable image space Σ is equivalent to geometry of a unit hypersphere (denoted by H^3) in a space of four dual dimensions, with oriented points, oriented lines, and oriented planes in Σ corresponding to points, oriented great circles, and oriented great spheres on H^3, respectively. This oriented version of spherical geometry is also termed *doubly elliptic geometry.*

The distance between two points, $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$, in Σ is a dual angle, $\hat{\phi} = \phi + \epsilon h$, which is obtained from:

$$\cos\hat{\phi} = \hat{\mathbf{X}} \cdot \hat{\mathbf{Y}} = \hat{X}_1\hat{Y}_1 + \hat{X}_2\hat{Y}_2 + \hat{X}_3\hat{Y}_3 + \hat{X}_4\hat{Y}_4. \tag{3}$$

The dual angular distance $\hat{\phi}$ is uniquely defined, provided that ϕ is restricted to the range $[0, \pi]$. When $\phi < \pi/2$, the two points $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ are said to be *similarly oriented.*

Since the image space Σ has three *dual* dimensions, a general line in Σ is a one-dual-dimensional line or a *twofold line.* Kinematically, a twofold line in Σ is the mapping of a two-degree-of-freedom screw motion which consists of two independent simple motions, a rotation and a translation. If the translation is made dependent on the rotation, then the resulting motion becomes a one-degree-of-freedom screw motion which maps into a special line, called a *unifold line*, in Σ. Of special value is the unifold line-segment that corresponds to a constant-speed screw motion (see Ge and Ravani 1993b):

$$\hat{\mathbf{L}}(\hat{\mathbf{b}}_0, \hat{\mathbf{b}}_1; t) = \frac{\sin((1-t)\hat{\phi})}{\sin\hat{\phi}}\hat{\mathbf{b}}_0 + \frac{\sin(t\hat{\phi})}{\sin\hat{\phi}}\hat{\mathbf{b}}_1, \quad t \in [0,1] \tag{4}$$

where $\hat{\mathbf{b}}_0$ and $\hat{\mathbf{b}}_1$ denote two similarly oriented image points that represent two configurations of an object in physical space and $\hat{\phi} = \phi + \epsilon h$ is the dual distance from $\hat{\mathbf{b}}_0$ to $\hat{\mathbf{b}}_1$. This unifold linear interpolation was used by Ge and Ravani (1993b) to construct special image curves, called the *Bézier type image curves*[2], that have Bézier type end-

[2] This type of curves are not of the Bernstein form, for they are not algebraic curves and do not possess the subdivision property, see Ge and Ravani (1993b).

point interpolation properties. The corresponding one-degree-of-freedom motions are termed *Bézier type motions.*

2 GEOMETRIC CONTINUITY

The smooth joining of two one-degree-of-freedom motion segments corresponds to the smooth joining of two unifold curve segments in the image space Σ. The simplest way to define local smoothness for an image curve is to require the curve to be n-times differentiable with respect to its current parametrization. Kinematically, this corresponds to the continuity of kinematic instantaneous invariants at the junction point. The geometric instantaneous properties of a motion are related to the differential properties of the corresponding image curve (McCarthy and Ravani, 1986). This section describes smoothness conditions in terms of the tangent directions and curvature of the image curve. From kinematics point of view, this corresponds to the continuity of the second order geometric instantaneous invariants of two motion segments. From CAGD point of view, this work represents an extension of the concept of geometric continuity of the second order, denoted as G^2, form Euclidean three-space to the image space. The study of geometric continuity and its application to curve design can be found in many CAGD literature, see for example, Barskey and DeRose (1989), Boehm (1987), and Farin (1993).

2.1 General Unifold Image Curves

G^2 continuity for unifold curves in Σ is defined in the same way as G^2 continuity for curves in Euclidean three-space. A unifold image curve is G^2 continuous if it is two-times differentiable with respect to arc length but not necessarily two-times differentiable with respect to its current parametrization.

Let $\hat{\mathbf{X}}_-(t_-)$ and $\hat{\mathbf{X}}_+(t_+)$ ($t_-, t_+ \in [0,1]$) denote the left segment and the right segment, respectively. They can be thought of as a composite curve with a dual arc length parametrization $\hat{s}(t)$ defined over $[\hat{s}_-, \hat{s}_+]$. The junction point is $\hat{\mathbf{X}}(\hat{s}_0) = \hat{\mathbf{X}}_-(1) = \hat{\mathbf{X}}_+(0)$. The joining of the two segments is G^1 continuous at $\hat{\mathbf{X}}(\hat{s}_0)$ if

$$\left.\frac{d\hat{\mathbf{X}}}{d\hat{s}}\right|_{\hat{s}=\hat{s}_0} = \frac{\dot{\hat{\mathbf{X}}}_+(0)}{\hat{v}_+(0)} = \frac{\dot{\hat{\mathbf{X}}}_-(1)}{\hat{v}_-(1)}, \tag{5}$$

where

$$\dot{\hat{\mathbf{X}}}_+(0) = \frac{d\hat{\mathbf{X}}_+(0)}{dt_+}, \quad \dot{\hat{\mathbf{X}}}_-(1) = \frac{d\hat{\mathbf{X}}_-(1)}{dt_-}, \quad \hat{v}_+ = \frac{d\hat{s}(t_+)}{dt_+}, \quad \hat{v}_- = \frac{d\hat{s}(t_-)}{dt_-}.$$

Let $\hat{\alpha} = \hat{v}_+(0)/\hat{v}_-(1) = \alpha + \epsilon\alpha^0$ ($\alpha > 0$), then Eq.(5) leads to $\dot{\hat{\mathbf{X}}}_+(0) = \hat{\alpha}\dot{\hat{\mathbf{X}}}_+(0)$, which is equivalent to the continuity of the tangent line-segments at the junction point:

$$\hat{\mathbf{X}}_+(0) \wedge \dot{\hat{\mathbf{X}}}_+(0) = \hat{\alpha}\hat{\mathbf{X}}_-(1) \wedge \dot{\hat{\mathbf{X}}}_-(1), \tag{6}$$

where the symbol "$\wedge$" denotes the vector wedge product which generalizes the three-dimensional vector cross product to higher dimensions (see Flanders 1967, Ge and Ravani 1993a).

Similarly, the requirement that the second order derivative of $\hat{\mathbf{X}}(\hat{s})$ be continuous at $\hat{s} = \hat{s}_0$ leads to the G^2 continuity condition:

$$\hat{\mathbf{X}}_+(0) \wedge \dot{\hat{\mathbf{X}}}_+(0) \wedge \ddot{\hat{\mathbf{X}}}_+(0) = \hat{\alpha}^3 \hat{\mathbf{X}}_-(1) \wedge \dot{\hat{\mathbf{X}}}_-(1) \wedge \ddot{\hat{\mathbf{X}}}_-(1). \tag{7}$$

This implies the continuity of curvature and osculating plane at the junction point.

2.2 Bézier Type Image Curves

We now specialize G^2 continuity conditions to composite Bézier type image curves. Let $\hat{\mathbf{b}}_{-3}$, $\hat{\mathbf{b}}_{-2}$, $\hat{\mathbf{b}}_{-1}$, $\hat{\mathbf{b}}_0$ and $\hat{\mathbf{b}}_0$, $\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$, $\hat{\mathbf{b}}_3$ be the control points of two adjacent Bézier curve segments which are denoted by $\hat{\mathbf{X}}_-(t_-)$ and $\hat{\mathbf{X}}_+(t_+)$, respectively. For the segment $\hat{\mathbf{X}}_+(t_+)$ on the right, the tangent and curvature properties at the junction point are given by

$$\hat{\mathbf{X}}_+(0) \wedge \dot{\hat{\mathbf{X}}}_+(0) = 3\frac{\hat{\phi}_0}{\sin\hat{\phi}_0}\hat{\mathbf{b}}_0 \wedge \hat{\mathbf{b}}_1, \tag{8}$$

$$\hat{\mathbf{X}}_+(0) \wedge \dot{\hat{\mathbf{X}}}_+(0) \wedge \ddot{\hat{\mathbf{X}}}_+(0) = 18\frac{\hat{\phi}_0^2\hat{\phi}_1}{\sin^2\hat{\phi}_0 \sin\hat{\phi}_1}\hat{\mathbf{b}}_0 \wedge \hat{\mathbf{b}}_1 \wedge \hat{\mathbf{b}}_2, \tag{9}$$

where $\hat{\phi}_0$, $\hat{\phi}_1$ are the dual angular distances between $\hat{\mathbf{b}}_0$, $\hat{\mathbf{b}}_1$ and $\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$, respectively.

It is interesting to note that, in the limiting case when $\phi_i \to 0$ $(i = 0, 1)$, the tangent and curvature properties of thê Bézier type image curves approach to those of Bernstein-Bézier cubics in Euclidean three-space, $3\hat{\mathbf{b}}_0 \wedge \hat{\mathbf{b}}_1$ and $18\hat{\mathbf{b}}_0 \wedge \hat{\mathbf{b}}_1 \wedge \hat{\mathbf{b}}_2$, respectively.

Similarly, for the segment $\hat{\mathbf{X}}_-(t_-)$ on the left, we have

$$\hat{\mathbf{X}}_-(1) \wedge \dot{\hat{\mathbf{X}}}_-(1) = 3\frac{\hat{\phi}_{-1}}{\sin\hat{\phi}_{-1}}\hat{\mathbf{b}}_{-1} \wedge \hat{\mathbf{b}}_0, \tag{10}$$

$$\hat{\mathbf{X}}_-(1) \wedge \dot{\hat{\mathbf{X}}}_-(1) \wedge \ddot{\hat{\mathbf{X}}}_-(1) = 18\frac{\hat{\phi}_{-1}^2\hat{\phi}_{-2}}{\sin^2\hat{\phi}_{-1} \sin\hat{\phi}_{-2}}\hat{\mathbf{b}}_{-2} \wedge \hat{\mathbf{b}}_{-1} \wedge \hat{\mathbf{b}}_0, \tag{11}$$

where $\hat{\phi}_{-2}$, $\hat{\phi}_{-1}$ are the dual angles between $\hat{\mathbf{b}}_{-2}$, $\hat{\mathbf{b}}_{-1}$ and $\hat{\mathbf{b}}_{-1}$, $\hat{\mathbf{b}}_0$, respectively.

In view of (6), (8), and (10), the two segments are G^1 continuous at $\hat{\mathbf{b}}_0$ if

$$\frac{\hat{\phi}_0}{\sin\hat{\phi}_0}\hat{\mathbf{b}}_0 \wedge \hat{\mathbf{b}}_1 = \hat{\alpha}_{0,-1}\frac{\hat{\phi}_{-1}}{\sin\hat{\phi}_{-1}}\hat{\mathbf{b}}_{-1} \wedge \hat{\mathbf{b}}_0. \tag{12}$$

Substitute the ratio of dual speed $\hat{\alpha}_{0,-1} = \hat{\phi}_0/\hat{\phi}_{-1}$ into (12) to obtain

$$\hat{\mathbf{b}}_0 \wedge \hat{\mathbf{b}}_1 = \frac{\sin\hat{\phi}_0}{\sin\hat{\phi}_{-1}}\hat{\mathbf{b}}_{-1} \wedge \hat{\mathbf{b}}_0. \tag{13}$$

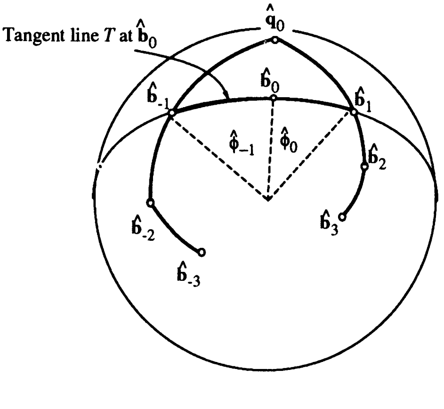


Figure 1: The curvature conditions.

In view of (7), (9), and (11), the two Bézier segments are G^2 continuous at $\hat{\mathbf{b}}_0$ if in addition

$$\frac{\hat{\phi}_0^2\hat{\phi}_1}{\sin^2\hat{\phi}_0\sin\hat{\phi}_1}\hat{\mathbf{b}}_0\wedge\hat{\mathbf{b}}_1\wedge\hat{\mathbf{b}}_2=\hat{\alpha}_{0,-1}^3\frac{\hat{\phi}_{-1}^2\hat{\phi}_{-2}}{\sin^2\hat{\phi}_{-1}\sin\hat{\phi}_{-2}}\hat{\mathbf{b}}_{-2}\wedge\hat{\mathbf{b}}_{-1}\wedge\hat{\mathbf{b}}_0.$$

After the substitution of $\hat{\alpha}_{0,-1}$ we obtain

$$\hat{\mathbf{b}}_0\wedge\hat{\mathbf{b}}_1\wedge\hat{\mathbf{b}}_2=\frac{\hat{\phi}_0\hat{\phi}_{-2}\sin^2\hat{\phi}_0\sin\hat{\phi}_1}{\hat{\phi}_1\hat{\phi}_{-1}\sin^2\hat{\phi}_{-1}\sin\hat{\phi}_{-2}}\hat{\mathbf{b}}_{-2}\wedge\hat{\mathbf{b}}_{-1}\wedge\hat{\mathbf{b}}_0. \tag{14}$$

Combine (13) and (14) to obtain

$$\hat{\mathbf{b}}_{-1}\wedge\hat{\mathbf{b}}_1\wedge\hat{\mathbf{b}}_2=\frac{\hat{\phi}_0\hat{\phi}_{-2}\sin\hat{\phi}_0\sin\hat{\phi}_1}{\hat{\phi}_1\hat{\phi}_{-1}\sin\hat{\phi}_{-1}\sin\hat{\phi}_{-2}}\hat{\mathbf{b}}_{-2}\wedge\hat{\mathbf{b}}_{-1}\wedge\hat{\mathbf{b}}_1. \tag{15}$$

Eq.(13) indicates that the three neighboring control points, $\hat{\mathbf{b}}_{-1}$, $\hat{\mathbf{b}}_0$ and $\hat{\mathbf{b}}_1$, are collinear and they define the common tangent line T at $\hat{\mathbf{b}}_0$. Eq.(14) indicates that the five control points, $\hat{\mathbf{b}}_{-2}$, $\hat{\mathbf{b}}_{-1}$, $\hat{\mathbf{b}}_0$, $\hat{\mathbf{b}}_1$, and $\hat{\mathbf{b}}_2$, are coplanar and they define the common osculating plane at $\hat{\mathbf{b}}_0$. Therefore the line defined by $\hat{\mathbf{b}}_{-2}$, $\hat{\mathbf{b}}_{-1}$ and the line defined by $\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$ are coplanar and meet in a point $\hat{\mathbf{q}}_0$. This suggests the use of points such as $\hat{\mathbf{q}}_0$ to construct the control points for a G^2 continuous composite Bézier type image curve (Figure 1 and 2).

Essential to the construction algorithm is the ability to determine the dual angles $\hat{\phi}_{3i}$ and $\hat{\phi}_{3i+1}$ that specify the location of the Bézier junction point $\hat{\mathbf{b}}_{3i}$ on the line segment joining $\hat{\mathbf{b}}_{3i-1}$ to $\hat{\mathbf{b}}_{3i+1}$ such that G^2 condition at $\hat{\mathbf{b}}_{3i}$ are satisfied (see Figure 2). In view of Eq.(15), the G^2 condition is given by:

$$\hat{\mathbf{b}}_{3i-1}\wedge\hat{\mathbf{b}}_{3i+1}\wedge\hat{\mathbf{b}}_{3i+2}=\frac{\hat{\phi}_{3i}\hat{\phi}_{3i-2}\sin\hat{\phi}_{3i}\sin\hat{\phi}_{3i+1}}{\hat{\phi}_{3i+1}\hat{\phi}_{3i-1}\sin\hat{\phi}_{3i-2}\sin\hat{\phi}_{3i-1}}\hat{\mathbf{b}}_{3i-2}\wedge\hat{\mathbf{b}}_{3i-1}\wedge\hat{\mathbf{b}}_{3i+1}.$$

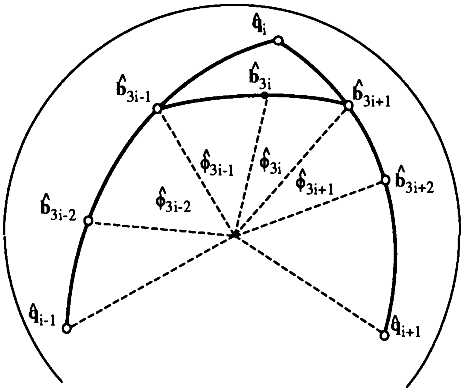


Figure 2: Construction of a G^2 continuous piecewise motion of Bézier type.

This leads to

$$\hat{\phi}_{3i}\sin\hat{\phi}_{3i} = \hat{\sigma}_i\hat{\phi}_{3i-1}\sin\hat{\phi}_{3i-1}, \tag{16}$$

where $\hat{\sigma}_i = \sigma_i + \epsilon\sigma_i^0$ is a dual number given by

$$\hat{\sigma}_i = \frac{\hat{A}_i\hat{\phi}_{3i-2}\sin\hat{\phi}_{3i+1}}{\hat{B}_i\sin\hat{\phi}_{3i-2}\hat{\phi}_{3i+1}} \tag{17}$$

and $\hat{A}_i$, $\hat{B}_i$ are the magnitudes of the trivectors $\hat{\mathbf{b}}_{3i-1}\wedge\hat{\mathbf{b}}_{3i+1}\wedge\hat{\mathbf{b}}_{3i+2}$ and $\hat{\mathbf{b}}_{3i-2}\wedge\hat{\mathbf{b}}_{3i-1}\wedge\hat{\mathbf{b}}_{3i+1}$, respectively. Eq.(16) and the relation

$$\hat{\phi}_{3i-1} + \hat{\phi}_{3i} = \hat{\psi}_i, \tag{18}$$

where $\hat{\psi}_i = \arccos(\hat{\mathbf{b}}_{3i-1}\cdot\hat{\mathbf{b}}_{3i+1})$, are the two equations needed to solve for the dual angles $\hat{\phi}_{3i-1}$, $\hat{\phi}_{3i}$. After eliminating ϕ_{3i}, the real parts of (16) and (18) combines into the following equation:

$$\frac{\cos\phi_{3i-1}}{\sin\phi_{3i-1}} = \frac{\cos\psi_i}{\sin\psi_i} + \frac{\sigma_i\phi_{3i-1}}{(\psi_i - \phi_{3i-1})\sin\psi_i},$$

from which the angle ϕ_{3i-1} can be solved using the Newton-Raphson method. From the dual parts of (16) and (18), one can obtain $\phi_{3i-1}^0 = \psi_i^0 - \phi_{3i}^0$ and

$$\phi_{3i}^0 = \frac{\sigma_i^0\phi_{3i}\phi_{3i-1}\sin\psi_i + \sigma_i\psi_i^0(\phi_{3i}\sin\psi_i + \phi_{3i}\phi_{3i-1}\cos\psi_i) + \sigma_i^2\psi_i^0\phi_{3i-1}^2}{\phi_{3i}^2 + \sigma_i(\psi_i\sin\psi_i + 2\phi_{3i}\phi_{3i-1}\cos\psi_i) + \sigma_i^2\phi_{3i-1}^2}.$$

3 DESIGNING A G^2 CONTINUOUS MOTION

This section presents a construction algorithm for designing a G^2 continuous piecewise Bézier type motion that approximates m key configurations. The orientation is specified by the angle (θ) and the axis ($\mathbf{s}$) of rotation, and the translation is specified by a vector $\mathbf{d}$. The algorithm proceeds as follows:

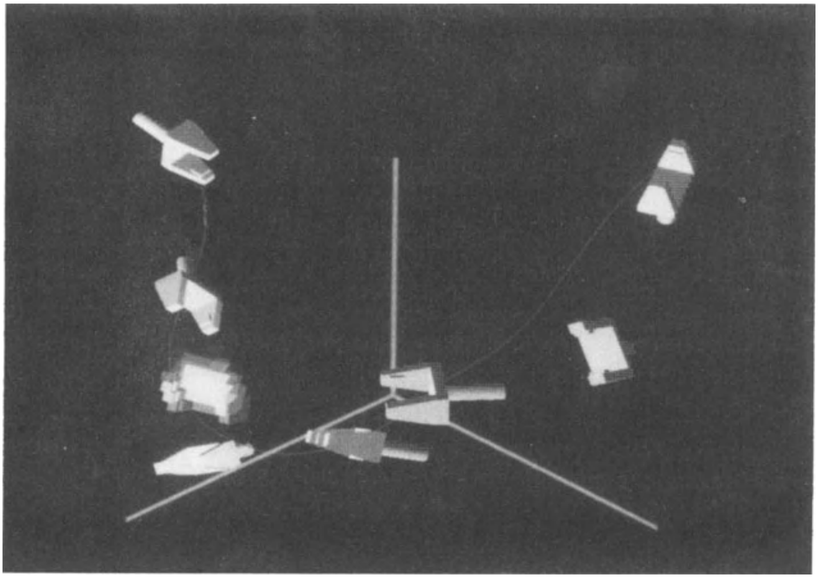


Figure 3: A set of seven given configurations.

1. Compute the dual Euler parameters $\hat{\mathbf{q}} = \mathbf{q} + \epsilon\mathbf{q}^0$ for all m configurations, where

$$\mathbf{q} = \left(\frac{s_x}{|\mathbf{s}|}\sin\frac{\theta}{2}, \frac{s_y}{|\mathbf{s}|}\sin\frac{\theta}{2}, \frac{s_z}{|\mathbf{s}|}\sin\frac{\theta}{2}, \cos\frac{\theta}{2}\right),$$

   and $|\mathbf{s}| = (s_x^2 + s_y^2 + s_z^2)^{1/2}$. The dual part $\mathbf{q}^0$ is obtained using (1).

2. Generate the control points $\hat{\mathbf{b}}_j$, where $j = 0, 1, 2, \cdots, 3(m-3)$, of a piecewise Bézier polygon:

   (a) Let $\hat{\mathbf{b}}_0 = \hat{\mathbf{q}}_0$, $\hat{\mathbf{b}}_1 = \hat{\mathbf{q}}_1$ and use (4) to compute $\hat{\mathbf{b}}_2 = \hat{\mathbf{L}}(\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2; \lambda_1)$ where $\lambda_1 = 1/2$ and can be adjusted in the range of $(0, 1)$ for fine tuning.

   (b) Let $\hat{\mathbf{b}}_{3(m-3)-1} = \hat{\mathbf{q}}_{m-2}$, $\hat{\mathbf{b}}_{3(m-3)} = \hat{\mathbf{q}}_{m-1}$ and $\hat{\mathbf{b}}_{3(m-3)-2} = \hat{\mathbf{L}}(\hat{\mathbf{q}}_{m-3}, \hat{\mathbf{q}}_{m-4}; \lambda_{m-3})$ where $\lambda_{m-3} = 1/2$ and can be adjusted in the range of $(0, 1)$.

   (c) For $i = 1, 2, 3, \cdots, (m-5)$, compute the in-between control points, $\hat{\mathbf{b}}_{3i+1} = \hat{\mathbf{L}}(\hat{\mathbf{q}}_{i+1}, \hat{\mathbf{q}}_{i+2}; \lambda_{i+1})$ and $\hat{\mathbf{b}}_{3i+2} = \hat{\mathbf{L}}(\hat{\mathbf{q}}_i, \hat{\mathbf{q}}_{i+1}; \lambda'_{i+1})$ where $\lambda_{i+1} = 1/3$ and $\lambda'_{i+1} = 2/3$ and they can be adjusted in the range of $0 < \lambda_{i+1} < \lambda'_{i+1} < 1$.

   (d) Generate $\hat{\mathbf{b}}_{3i}$ $(i = 1, 2, 3, \cdots, (m-4))$ such that both G^1 and G^2 continuity conditions are satisfied. This is achieved by the following:

      i. Compute $\hat{A}_i = \left|\hat{\mathbf{b}}_{3i-1} \wedge \hat{\mathbf{b}}_{3i+1} \wedge \hat{\mathbf{b}}_{3i+2}\right|$ and $\hat{B}_i = \left|\hat{\mathbf{b}}_{3i-2} \wedge \hat{\mathbf{b}}_{3i-1} \wedge \hat{\mathbf{b}}_{3i+1}\right|$.

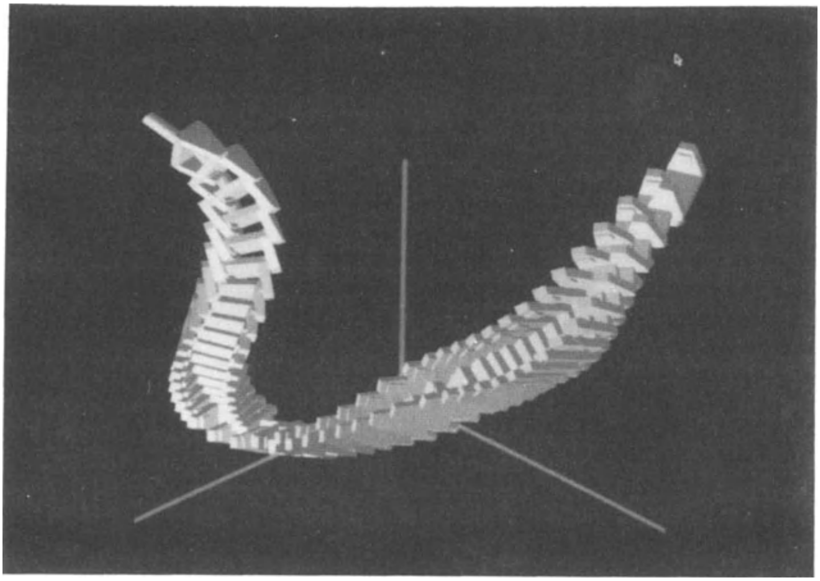


Figure 4: A G^2 spline motion which approximates a set of seven configurations.

ii. Compute $\hat{\phi}_{3i-2} = \arccos(\hat{\mathbf{b}}_{3i-2} \cdot \hat{\mathbf{b}}_{3i-1})$, $\hat{\phi}_{3i+1} = \arccos(\hat{\mathbf{b}}_{3i+1} \cdot \hat{\mathbf{b}}_{3i+2})$, and $\hat{\psi}_i = \arccos(\hat{\mathbf{b}}_{3i-1} \cdot \hat{\mathbf{b}}_{3i+1})$.

iii. Compute $\hat{\sigma}_i$ using (17).

iv. Obtain $\hat{\phi}_{3i-1}$ and $\hat{\phi}_{3i}$ by solving (16) and (18).

v. Compute $\hat{\mathbf{b}}_{3i} = (\sin\hat{\phi}_{3i}\hat{\mathbf{b}}_{3i-1} + \sin\hat{\phi}_{3i-1}\hat{\mathbf{b}}_{3i+1})/\sin\hat{\psi}_i$.

3. Select a knot sequence or the timing of the motion and generate each Bézier curve segment (see Ge and Ravani 1993b).

4. Convert each image point $\hat{\mathbf{q}} = \mathbf{q}+\epsilon\mathbf{q}^0$ into a rotation matrix $[R]$ and a translation vector $\mathbf{d}$ for computer graphic animation, where

$$[R] = \begin{bmatrix} q_1^2-q_2^2-q_3^2+q_4^2 & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_2q_4) \\ 2(q_1q_2+q_3q_4) & -q_1^2+q_2^2-q_3^2+q_4^2 & 2(q_2q_3-q_1q_4) \\ 2(q_1q_3-q_2q_4) & 2(q_2q_3+q_1q_4) & -q_1^2-q_2^2+q_3^2+q_4^2 \end{bmatrix},$$

and $\mathbf{d}$ can be obtained by inverting (1).

For designing closed loop motions, the steps 2.(*a*) and 2.(*b*) in the above construction are omitted. Figure 3 shows the set of seven configurations together with several configurations belonging to the resulting G^2 motion. Figure 4 shows the entire G^2 motion that approximates the given configurations.

It is interesting to point out that in the special case when all given configurations are of the same orientation, the aforementioned construction algorithm reduces to the Farin construction for designing G^2 Euclidean splines (Farin, 1993).

CONCLUSIONS

This paper has presented a geometric construction method that is most suitable for interactive design of a G^2 continuous motion, manipulating the key configurations until a desired motion is obtained. It has applications in motion animation, kinematics and CAD/CAM. The resulting motion, however, does not pass through the input configurations except the first and the last one. An inverse design algorithm may be developed to obtain the unknown control points, $\hat{\mathbf{q}}_0, \cdots, \hat{\mathbf{q}}_{m-1}$, such that the G^2 piecewise Bézier type image curve $\hat{\mathbf{b}}(t)$ passes through the given input points, $\hat{\mathbf{b}}_j$, $i = 0, 1, 2, \cdots, 3(m-3)$. This is left as future work.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation Research Initiation Award MSS-9211490 to the State University of New York at Stony Brook, and in part by National Science Foundation grant DMC-8796348 to the University of California at Davis.

REFERENCES

Barsky, B. A. and DeRose, T. D., 1989, Geometric continuity of parametric curves: three equivalent characterizations. *IEEE Computer Graphics and Applications*, 9(6):60-68.

Boehm, W., 1987, Smooth curves and surfaces. *Geometric Modeling: Algorithms and New Trends*, edited by G. Farin, SIAM, Philadelphia, PA, pp. 175-184.

Bottema, O. and B. Roth, 1979, *Theoretical Kinematics.* North Holland Publ., Amsterdam, 558pp. Pages 51-60, 150-152, 521-523.

Duff, T., 1986, Quaternion splines for animating orientation. Technical report, AT&T Bell Laboratories.

Farin, G., 1993, *Curves and Surfaces for Computer Aided Geometric Design.* 3rd ed., Academic Press, San Diego, 334pp.

Flanders, H. 1963, *Differential Forms with Application to the Physical Sciences.* Academic Press, New York, 205pp.

Ge, Q.J., and B. Ravani, 1993a, Computer aided geometric design of motion interpolants, *ASME J. of Mechanical Design*, in press.

Ge, Q.J., and B. Ravani, 1993b, Geometric construction of Bézier Type Motions, *ASME J. of Mechanical Design*, in press.

McCarthy, J.M., and B. Ravani, 1986, Differential kinematics of spherical and spatial motions using kinematic mapping. *Trans. ASME J. of Appl. Mech.*, 53:15-22.

Pletinckx, D., 1989, Quaternion calculus as a basic tool in computer graphics. *The Visual Computer*, 5:2-13.

Ravani, B., and B. Roth, 1984, Mappings of spatial kinematics. *Trans. ASME J. of Mech., Transmissions., and Auto. in Design.* 106(3):341-347.

Reeves, W., 1981, Inbetweening for computer animation utilizing moving point constraints. *ACM Siggraph*, 15(3):263-269.

Shoemake, K., 1985, Animating rotation with quaternion curves. *ACM Siggraph*, 19(3):245-254.

Part 6

Kinematics of Mechanisms

Forward Kinematics of a 3-DOF Variable-Geometry-Truss Manipulator

Roger B. Hertz and Peter C. Hughes
University of Toronto Institute for Aerospace Studies
4925 Dufferin St., Downsview
Ontario, Canada M3H 5T6

Abstract *The forward kinematics of an octahedral type variable-geometry-truss manipulator is presented. The manipulator is comprised of two stacked octahedral trusses. The intersection of the octahedra consist of 3 linear actuators, which are used to control the position of the moving plane of the manipulator relative to the base, giving the mechanism a 3-DOF capability. A kinematical model of the manipulator is presented which includes important features of non-equal octahedral geometry and inter-hinge displacements. Vectorial equations are formulated to reduce the problem into two systems of nonlinear equations, each of which has three unknowns. Each system of equations is shown to have 8 reflected solution pairs through the formulation of polynomials in one variable, giving the manipulator a maximum of 256 unique configurations for one set of actuator lengths. Numerical examples confirm the validity of the results.*

Introduction

The use of articulated truss mechanisms in robotic applications has been examined by many researchers since 1984: Miura et al. (1984), Rhodes and Mikulas (1985), Sincarsin and Hughes (1987), Hughes et al. (1991) and Chirikjian and Burdick (1991). These *variable-geometry-truss* (VGT) mechanisms possess a high degree of parallelism and have favorable stiffness properties, making them promising for both space and industrial robotic tasks. The stacked octahedral form of these mechanisms is commonly used in robotic applications, due to its excellent stiffness and the simplicity of its hinge design (Sincarsin and Hughes 1987). In this configuration, the intersection of the octahedral trusses is made up of three linear actuators, as shown in Figure 1.

The forward kinematics of the double octahedral VGT has been addressed by several researchers. The problem is to find the relative displacement and orientation between the base and moving planes of the mechanism, given the three actuator lengths. Miura et al. (1984) reduced the problem to solving three nonlinear equations for the *petal angles* of the lower and upper octahedra. Reinholtz and Gokhale (1988), Tidwell et al. (1990) and Arun et al. (1992) used the same method, solving the kinematics of a

J. Angeles et al. (eds.), Computational Kinematics, 241–250.

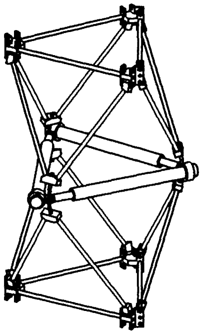


Figure 1: A 3-DOF Variable-Geometry-Truss Manipulator

single octahedron. Arun et al. (1992), however, used polynomial continuation methods to identify a maximum of 16 possible solutions to the nonlinear petal equations.

Similarities between the VGT and the Steward Platform are evident, since they both share octahedral geometries. This commonality extends to the forward kinematic analysis. Griffis and Duffy (1989) and Innocenti and Parenti-Castelli (1990) have demonstrated that the forward kinematics problem can be formulated as three nonlinear equations, to which a maximum of 16 solutions exist. The solutions consist of 8 solution pairs reflected about the fixed base of the octahedron (Griffis and Duffy 1989).

In this paper, the forward kinematics problem is formulated for the double octahedral VGT explicitly; as with Miura et al. (1984), the problem is addressed for the mechanism as a whole. Previous work is extended by considering several important kinematic features. The first considers the more general case of non-equal octahedron geometry, which allows the analysis of a *tapered VGT* — a mechanism well suited to robotic applications. The second feature is the inclusion of kinematically correct joint offsets, a factor crucial to accurate control of VGT manipulators. The vector relationships are formulated to reduce the kinematics to a pair of nonlinear petal angle equations. The petal angle equations are then decoupled though polynomial elimination: each equation is shown to have a maximum of 8 reflected solution pairs, giving the VGT a maximum of 256 possible kinematic assemblies with one set of actuator lengths. In a numerical example, a tapered VGT is examined with a set of actuator lengths that give the manipulator a total of 64 possible assemblies.

Kinematic Model

The kinematic model for the double octahedral VGT is shown in Figure 2. Pertinent dimensions are the actuator lengths, l_1, l_2 and l_3; fixed member lengths, l_{t1}, l_{t2}, l_{t3} and l_{t4}; and hinge nodal offsets, Δ_1, Δ_2, Δ_3 and Δ_4. The actuator lengths are l_1, l_2 and l_3. The reference frames $\mathbf{F}_a$ and $\mathbf{F}_c$ are each attached to the centroid of the base and moving triangular planes, respectively. The following assumptions are made for the kinematic

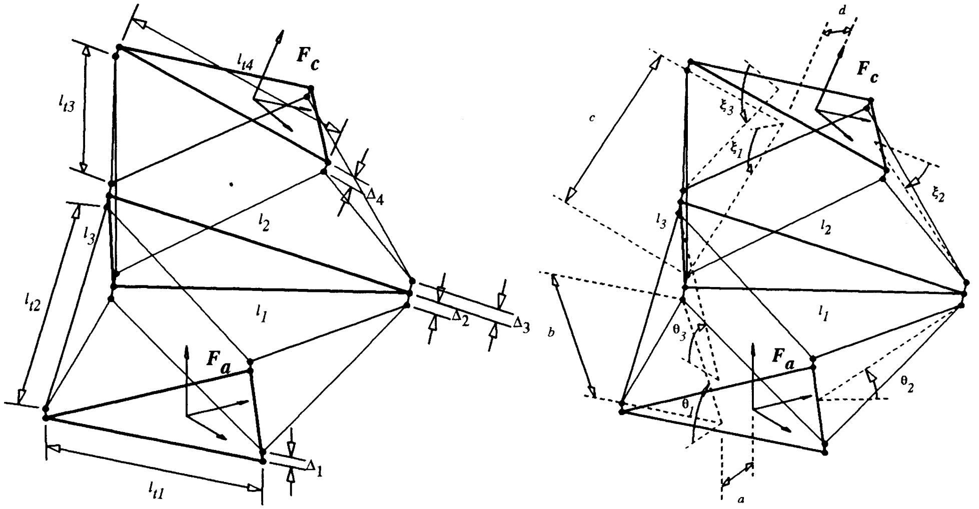


Figure 2: **Kinematic model for the double octahedral VGT: a) length parameter definitions, b) petal angle definitions**

model:

- The base and moving planes consist of equilateral triangles of lengths l_{t1} and l_{t4}, respectively.
- The lower and upper petals are isosceles triangles, with side lengths of l_{t2} and l_{t3}, respectively.
- The joint offsets are perpendicular to their respective planes.

It is convenient in the forward kinematic analysis to define the following petal angles and associated lengths. The lower petal angles are denoted as $\{\theta_1, \theta_2, \theta_3\}$, while the upper petal angles are $\{\xi_1, \xi_2, \xi_3\}$. The associated lengths are given by

$$a = \frac{\sqrt{3}}{6} l_{t1}, \quad b = \frac{1}{2}\sqrt{4\,l_{t2}^{\;2} - l_{t1}^{\;2}}, \quad c = \frac{1}{2}\sqrt{4\,l_{t3}^{\;2} - l_{t4}^{\;2}}, \quad d = \frac{\sqrt{3}}{6} l_{t4}.$$

Vectorial Kinematics

In order to solve for the relative displacement between the fixed and moving planes, two reference frames $\mathbf{F}_a$ and $\mathbf{F}_c$ are defined for each the base and moving plane. For each reference frame, four sets of direction vectors are defined: one perpendicular to the plane, and three lying in the plane, perpendicular to each side of the equilateral triangle. As depicted in Figure 3, the displacement vector can be expressed as follows:

$$\underset{\rightarrow}{\boldsymbol{d}} = \underset{\rightarrow}{\boldsymbol{u}}_1 - \underset{\rightarrow}{\boldsymbol{v}}_1 \tag{1}$$

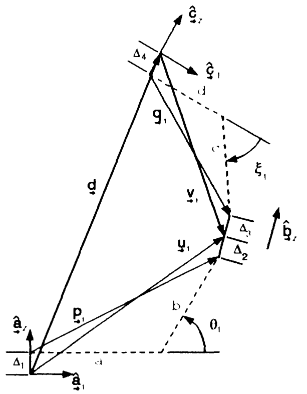


Figure 3: **Vector relationships for the VGT**

where

$$\underset{\to}{\boldsymbol{u}}_1 = \Delta_1 \underset{\to}{\hat{\boldsymbol{a}}}_z + \underset{\to}{\boldsymbol{p}}_1 + \Delta_1 \underset{\to}{\hat{\boldsymbol{b}}}_z, \qquad \underset{\to}{\boldsymbol{v}}_1 = -\Delta_4 \underset{\to}{\hat{\boldsymbol{c}}}_z + \underset{\to}{\boldsymbol{q}}_1 - \Delta_3 \underset{\to}{\hat{\boldsymbol{b}}}_z \tag{2}$$

with the intermediate vectors for the octahedra defined as

$$\underset{\to}{\boldsymbol{p}}_1 = (a + b\cos\theta_1)\underset{\to}{\hat{\boldsymbol{a}}}_1 + b\sin\theta_1 \underset{\to}{\hat{\boldsymbol{a}}}_z, \qquad \underset{\to}{\boldsymbol{q}}_1 = (d + c\cos\xi_1)\underset{\to}{\hat{\boldsymbol{c}}}_1 - c\sin\xi_1 \underset{\to}{\hat{\boldsymbol{c}}}_z. \tag{3}$$

Formulation of the Petal Equations

To solve for the petal equations for the VGT, the closure equations are written for both the lower and upper portions of the mechanism. The vectors for the entire VGT are obtained by expressing (2) and (3) in a more general form:

$$\underset{\to}{\boldsymbol{u}}_n = \Delta_1 \underset{\to}{\hat{\boldsymbol{a}}}_z + \underset{\to}{\boldsymbol{p}}_n + \Delta_1 \underset{\to}{\hat{\boldsymbol{b}}}_z, \qquad \underset{\to}{\boldsymbol{v}}_n = -\Delta_4 \underset{\to}{\hat{\boldsymbol{c}}}_z + \underset{\to}{\boldsymbol{q}}_n - \Delta_3 \underset{\to}{\hat{\boldsymbol{b}}}_z, \qquad n = \{1, 2, 3\} \tag{4}$$

where now

$$\underset{\to}{\boldsymbol{p}}_n = (a + b\cos\theta_n)\underset{\to}{\hat{\boldsymbol{a}}}_n + b\sin\theta_n \underset{\to}{\hat{\boldsymbol{a}}}_z, \qquad \underset{\to}{\boldsymbol{q}}_n = (d + c\cos\xi_n)\underset{\to}{\hat{\boldsymbol{c}}}_n - c\sin\xi_n \underset{\to}{\hat{\boldsymbol{c}}}_z. \tag{5}$$

The closure equations for the mechanism can now be written in terms of the actuator vectors

$$\underset{\to}{\boldsymbol{l}}_r = \underset{\to}{\boldsymbol{u}}_s - \underset{\to}{\boldsymbol{u}}_r, \qquad \underset{\to}{\boldsymbol{l}}_r = \underset{\to}{\boldsymbol{v}}_s - \underset{\to}{\boldsymbol{v}}_r \tag{6}$$

where

$$r \triangleq \{1, 2, 3\}, \qquad s \triangleq \operatorname{mod}(r, 3) + 1$$

and r and s are termed *companion indices* from Innocenti and Parenti-Castelli (1990). The function mod(n, m) is the abbreviated form of n *modulo* m, the remainder on the division of n by m. The substitution of (4) into (6) yields a set of equations in terms of $\boldsymbol{p}_n$ and $\boldsymbol{q}_n$

$$\underrightarrow{\boldsymbol{l}}_r \;=\; \underrightarrow{\boldsymbol{p}}_s - \underrightarrow{\boldsymbol{p}}_r, \qquad \underrightarrow{\boldsymbol{l}}_r \;=\; \underrightarrow{\boldsymbol{q}}_s - \underrightarrow{\boldsymbol{q}}_r. \tag{7}$$

The subsequent squaring of the actuator vectors results in

$$l_r^{\,2} = \underrightarrow{\boldsymbol{p}}_r \cdot \underrightarrow{\boldsymbol{p}}_r - 2\,\underrightarrow{\boldsymbol{p}}_r \cdot \underrightarrow{\boldsymbol{p}}_s + \underrightarrow{\boldsymbol{p}}_s \cdot \underrightarrow{\boldsymbol{p}}_s, \qquad l_r^{\,2} = \underrightarrow{\boldsymbol{q}}_r \cdot \underrightarrow{\boldsymbol{q}}_r - 2\,\underrightarrow{\boldsymbol{q}}_r \cdot \underrightarrow{\boldsymbol{q}}_s + \underrightarrow{\boldsymbol{q}}_s \cdot \underrightarrow{\boldsymbol{q}}_s. \tag{8}$$

By substituting (5) into (8), the following petal angle equations are obtained:

$$l_r^{\,2} = D\ (\mathrm{c}\theta_r + \mathrm{c}\theta_s) + E\ \mathrm{c}\theta_r\mathrm{c}\theta_s + F\ \mathrm{s}\theta_r\mathrm{s}\theta_s + G \tag{9}$$

$$l_r^{\,2} = H\ (\mathrm{c}\xi_r + \mathrm{c}\xi_s) + I\ \mathrm{c}\xi_r\mathrm{c}\xi_s + J\ \mathrm{s}\xi_r\mathrm{s}\xi_s + K \tag{10}$$

where

$$D = 3ab, \quad E = b^2, \quad F = -2b^2, \quad G = 3a^2 + 2b^2,$$
$$H = 3cd, \quad I = c^2, \quad F = -2c^2, \quad G = 2c^2 + 3d^2,$$

c $=$ cos and s $=$ sin, noting the following identities for the unit normals:

$$\underrightarrow{\hat{\boldsymbol{a}}}_r \cdot \underrightarrow{\hat{\boldsymbol{a}}}_s \equiv -1/2, \quad \underrightarrow{\hat{\boldsymbol{a}}}_r \cdot \underrightarrow{\hat{\boldsymbol{a}}}_z \equiv 0, \quad \underrightarrow{\hat{\boldsymbol{c}}}_r \cdot \underrightarrow{\hat{\boldsymbol{c}}}_s \equiv -1/2, \quad \underrightarrow{\hat{\boldsymbol{c}}}_r \cdot \underrightarrow{\hat{\boldsymbol{c}}}_z \equiv 0.$$

Trivial Solution

The trivial solution to (9) and (10) is of particular interest because it represents the case when the VGT is in a *stowed* state. If this is applied as a constraint to the design of a VGT, both the lower and upper bays will approach the stowed state simultaneously. For equal actuator lengths, the stowed constraint is given by

$$2\,l_r \;=\; l_{t1} + \sqrt{3(4l_{t2}^{\,2} - l_{t1}^{\,2})} \;=\; l_{t4} + \sqrt{3(4l_{t3}^{\,2} - l_{t4}^{\,2})} \tag{11}$$

The ratio $t_r = l_{t1}/l_{t4}$ is termed the *taper ratio.* Given the fixed member lengths l_{t1} and l_{t2}, and the taper ratio, for example, l_{t3} may be solved from (11).

Indirect Solution

Each of the three equations represented in (9) and (10) are nonlinear with respect to the petal angles. An indirect solution method like Newton-Raphson's can be employed to iterate to the unknown angles given a suitable initial guess. Since an analytical derivative is readily available, the iterative method is computationally efficient with successive calls and updates to the initial guess. The solution is somewhat sensitive to the initial guess, however, especially in geometries close to singularities. This leaves open the possibility that the solver will iterate to an unwanted solution, which warrants a closer investigation to the solution of petal angle equations.

Direct Solution

A direct solution to (9) and (10) may be obtained by expressing each of the equations in terms of a single variable. We will consider only (9) in detail here, since the solution of (10) follows directly.

The decoupling of (9) involves the elimination of two of the petal angles, resulting in a high degree polynomial expression in a single petal angle. To facilitate this, we substitute the following half-tangent trigonometric identities into (9):

$$c\theta_n = \frac{1 - \mathrm{t}_n^{\,2}}{1 + \mathrm{t}_n^{\,2}}, \qquad s\theta_n = \frac{2\,\mathrm{t}_n}{1 + \mathrm{t}_n^{\,2}} \tag{12}$$

where $\mathrm{t}_n = \tan(\theta_n/2)$. With expansion of the companion indices, the petal equations become

$$\begin{aligned} a_1 + b_1\mathrm{t}_1^{\,2} + c_t\mathrm{t}_1\mathrm{t}_2 + b_1\mathrm{t}_2^{\,2} + e_1\mathrm{t}_1^{\,2}\mathrm{t}_2^{\,2} &= 0 && (13)\\ a_2 + b_2\mathrm{t}_2^{\,2} + c_t\mathrm{t}_2\mathrm{t}_3 + b_2\mathrm{t}_3^{\,2} + e_2\mathrm{t}_2^{\,2}\mathrm{t}_3^{\,2} &= 0 && (14)\\ a_3 + b_3\mathrm{t}_3^{\,2} + c_t\mathrm{t}_3\mathrm{t}_1 + b_3\mathrm{t}_1^{\,2} + e_3\mathrm{t}_3^{\,2}\mathrm{t}_1^{\,2} &= 0 && (15) \end{aligned}$$

where

$$a_n = 2D + E + G - l_n^{\,2}, \quad b_n = -E + G - l_n^{\,2}, \quad c_t = 4F, \quad e_n = -2D + E + G - l_n^{\,2}.$$

An immediate observation on the system of equations in (13)-(15) is that it has a Bezout number of 16 (Arun et al. 1992). This indicates that the number of solutions the petal angle system is 16, but does not give us information as to the existence of reflected solutions. In order to do so, successive polynomial elimination must be employed as in Griffis and Duffy (1989) and Innocenti and Parenti-Castelli (1990). By this, the t_2 term is eliminated from (13) and (14) to form a new polynomial in t_1 and t_3. This new polynomial is then used with (15) to isolate t_1. To do so, the system is written as

$$\begin{aligned} j\,\mathrm{t}_3^{\,2} + k\,\mathrm{t}_3 + l &= 0 && (16)\\ m\,\mathrm{t}_3^{\,3} + n\,\mathrm{t}_3 + \phi &= 0 && (17)\\ u\,\mathrm{t}_2^{\,2} + v\,\mathrm{t}_2 + w &= 0 && (18) \end{aligned}$$

where

$$\begin{aligned} j &= b_2 + e_2\mathrm{t}_2^{\,2}, & k &= c_t\mathrm{t}_2, & l &= a_2 + b_2\mathrm{t}_2^{\,2},\\ m &= b_3 + e_3\mathrm{t}_1^{\,2}, & n &= c_t\mathrm{t}_1, & \phi &= a_3 + b_3\mathrm{t}_1^{\,2},\\ u &= b_1 + e_1\mathrm{t}_1^{\,2}, & v &= c_t\mathrm{t}_1, & w &= a_1 + b_1\mathrm{t}_1^{\,2}. \end{aligned}$$

Since the polynomials in (16) and (17) share solutions, their resultant is equal to zero (Kendig 1976). This is given by

$$(lm - j\phi)^2 - (km - jn)(ln - k\phi) = 0$$

which when expanded and grouped in terms of t_2 yields:

$$p\,\mathrm{t}_2{}^4 + q\,\mathrm{t}_2{}^3 + r\,\mathrm{t}_2{}^2 + s\,\mathrm{t}_2 + t \;=\; 0 \tag{19}$$

where

$$p = p_4\mathrm{t}_1{}^4 + p_2\mathrm{t}_1{}^2 + p_0, \qquad q = q_3\mathrm{t}_1{}^3 + q_1\mathrm{t}_1, \qquad r = r_4\mathrm{t}_1{}^4 + r_2\mathrm{t}_1{}^2 + r_0,$$

$$s = s_3\mathrm{t}_1{}^3 + s_1\mathrm{t}_1, \qquad t = t_4\mathrm{t}_1{}^4 + t_2\mathrm{t}_1{}^2 + t_0.$$

The coefficients for the above terms are found by additional grouping of t_1. The resultant of (18) and (19) is again equal to zero, leading to the following 8th order polynomial in terms of $\mathrm{t}_1{}^2$:

$$\sum_{i=0}^{8} x_i\,\mathrm{t}_1{}^{2i} = 0 \tag{20}$$

where the x_i coefficients are expressed in terms of a_n, b_n, c_t, and e_n terms in (13)-(15). Given each solution for t_1, a solution for t_2 can be extracted from (18) and (19), from which a solution for t_3 can be found from (16) and (17). Once the half-tangent values are determined, the solution for the petal angles follows from $\theta_n = 2\,\tan^{-1}\mathrm{t}_n$, for $n = \{1, 2, 3\}$. The 8th order polynomial in (20) confirms the existence of 16 roots, and indicates they that occur in reflected pairs. The number of real solutions will depend, however, on the input parameters in the problem.

The implication of these results is that both the lower and upper bays have 8 reflected solutions. This has a significant impact on the total number of solutions for the VGT, especially due to the fact that the geometry of the octahedra may be unequal. The VGT has a maximum of 256 real kinematic solutions, given one set of actuator lengths. Note that the inclusion of the offsets in the kinematical model precludes the existence of 128 reflected solutions.

Scalar Kinematics

Following the notation convention developed in Hughes (1986), the scalar equivalents of the vectorial kinematics can be written in matrix form. The vectors are defined in terms of their local basis vectors, or reference frames, as follows:

$$\underset{\rightarrow}{\boldsymbol{d}} = \underset{\rightarrow}{\mathbf{F}}{}_a^T \boldsymbol{d} \tag{21}$$

where

$$\underset{\rightarrow}{\mathbf{F}}{}_a = \begin{bmatrix} \underset{\rightarrow}{\hat{\boldsymbol{a}}}_x & \underset{\rightarrow}{\hat{\boldsymbol{a}}}_y & \underset{\rightarrow}{\hat{\boldsymbol{a}}}_z \end{bmatrix}^T, \qquad \boldsymbol{d} = \begin{bmatrix} d_x & d_y & d_z \end{bmatrix}^T$$

Other vector equivalents used in the analysis are summarized by

$$\begin{bmatrix} \underset{\rightarrow}{\boldsymbol{u}}_n & \underset{\rightarrow}{\boldsymbol{p}}_n & \underset{\rightarrow}{\hat{\boldsymbol{a}}}_n & \underset{\rightarrow}{\hat{\boldsymbol{b}}}_x & \underset{\rightarrow}{\hat{\boldsymbol{b}}}_y & \underset{\rightarrow}{\hat{\boldsymbol{b}}}_z & \underset{\rightarrow}{\boldsymbol{l}}_n \end{bmatrix} \;=\; \underset{\rightarrow}{\mathbf{F}}{}_a^T \begin{bmatrix} \boldsymbol{u}_n & \boldsymbol{p}_n & \boldsymbol{a}_n & \boldsymbol{b}_x & \boldsymbol{b}_y & \boldsymbol{b}_z & \boldsymbol{l}_n \end{bmatrix} \tag{22}$$

$$\begin{bmatrix} \underset{\rightarrow}{\boldsymbol{v}}_n & \underset{\rightarrow}{\boldsymbol{q}}_n & \underset{\rightarrow}{\hat{\boldsymbol{c}}}_n & \underset{\rightarrow}{\hat{\boldsymbol{b}}}_x & \underset{\rightarrow}{\hat{\boldsymbol{b}}}_y & \underset{\rightarrow}{\hat{\boldsymbol{b}}}_z & \underset{\rightarrow}{\boldsymbol{l}}_n \end{bmatrix} \;=\; \underset{\rightarrow}{\mathbf{F}}{}_c^T \begin{bmatrix} \boldsymbol{v}_n & \boldsymbol{q}_n & \boldsymbol{c}_n & \boldsymbol{b}_{xc} & \boldsymbol{b}_{y_c} & \boldsymbol{b}_{zc} & \boldsymbol{l}_{nc} \end{bmatrix} \tag{23}$$

With these definitions, the vector equation in (1) becomes

$$\boldsymbol{d} = \boldsymbol{u}_1 - \boldsymbol{C}_{ac}\boldsymbol{v}_1 \tag{24}$$

where

$$\boldsymbol{u}_1 = \Delta_1 \mathbf{1}_z + \boldsymbol{p}_1 + \Delta_1 \boldsymbol{b}_z, \qquad \boldsymbol{v}_1 = -\Delta_4 \mathbf{1}_z + \boldsymbol{q}_1 - \Delta_3 \boldsymbol{b}_{zc} \tag{25}$$

$$\boldsymbol{p}_1 = (a + b\cos\theta_1)\boldsymbol{a}_1 + b\sin\theta_1 \mathbf{1}_z, \qquad \boldsymbol{q}_1 = (d + c\cos\xi_1)\boldsymbol{c}_1 - c\sin\xi_1 \mathbf{1}_z \tag{26}$$

In (25) and (26), $\mathbf{1}_z$ denotes the unit column matrix $[0\ 0\ 1]^T$, while the unit normals are given by

$$\boldsymbol{b}_z = \frac{\boldsymbol{l}_1^{\times}\boldsymbol{l}_2}{||\boldsymbol{l}_1^{\times}\boldsymbol{l}_2||}, \qquad \boldsymbol{b}_{zc} = \frac{\boldsymbol{l}_{1c}^{\times}\boldsymbol{l}_{2c}}{||\boldsymbol{l}_{1c}^{\times}\boldsymbol{l}_{2c}||} \tag{27}$$

where $\boldsymbol{l}_1^{\times}$ denotes the cross product operation on the 3×1 matrix $\boldsymbol{l}_1$. The 3×3 rotation matrix $\boldsymbol{C}_{ac}$ in (24) is defined as $\underset{\rightarrow}{\mathrm{F}}_a \cdot \underset{\rightarrow}{\mathrm{F}}_c^T$. This matrix can found from the product of rotation matrices from each octahedron

$$\boldsymbol{C}_{ac} = \boldsymbol{C}_{ab}\boldsymbol{C}_{cb}^T \tag{28}$$

where

$$\boldsymbol{C}_{ab} = \begin{bmatrix} \boldsymbol{b}_x & \boldsymbol{b}_y & \boldsymbol{b}_z \end{bmatrix}, \qquad \boldsymbol{C}_{cb} = \begin{bmatrix} \boldsymbol{b}_{xc} & \boldsymbol{b}_{y_c} & \boldsymbol{b}_{zc} \end{bmatrix},$$

$$\boldsymbol{b}_x = \frac{\boldsymbol{l}_1}{||\boldsymbol{l}_1||}, \qquad \boldsymbol{b}_y = \boldsymbol{b}_z^{\times}\boldsymbol{b}_x, \qquad \boldsymbol{b}_{xc} = \frac{\boldsymbol{l}_{1c}}{||\boldsymbol{l}_{1c}||}, \qquad \boldsymbol{b}_{y_c} = \boldsymbol{b}_{zc}^{\times}\boldsymbol{b}_{xc}.$$

Special Case: Symmetric Octahedra

The result in (24) can be applied to the special case where the octahedra are symmetric, namely $a = d$, $b = c$, $\Delta_1 = \Delta_4$, and $\Delta_2 = \Delta_3$. With equal geometry on each octahedron, the petal equation in (9) need only be solved once. The general displacement solution in (24) reduces to

$$\boldsymbol{d} = 2\, \boldsymbol{u}_1{}^T \boldsymbol{b}_z\, \boldsymbol{b}_z \tag{29}$$

which confirms the assertion by Miura et al. (1984) that the displacement vector is perpendicular the actuator plane.

Numerical Example

A tapered VGT was considered in the following numerical example. The lower two member lengths l_{t1} and l_{t2} were both set to 0.75m. A taper ratio (t_r) of 2 was selected, resulting in $l_{t3} = 0.7806$m and $l_{t4} = 0.375$m from the stowed constraint in (11). Joint offsets were set to 5cm, and the actuator lengths were selected as follows: $l_1 = 1.3$m, $l_2 = 1.2$m, and $l_3 = 1.1$m. The direct solution to the lower and upper petal equations yielded eight real solutions each, giving a total of 64 possible assemblies of the manipulator. The petal angles are given in the following table, and six selected assemblies are depicted in Figure 4. (The $n{:}m$ solution denotes the n-th lower and m-th upper petal angle solution.)

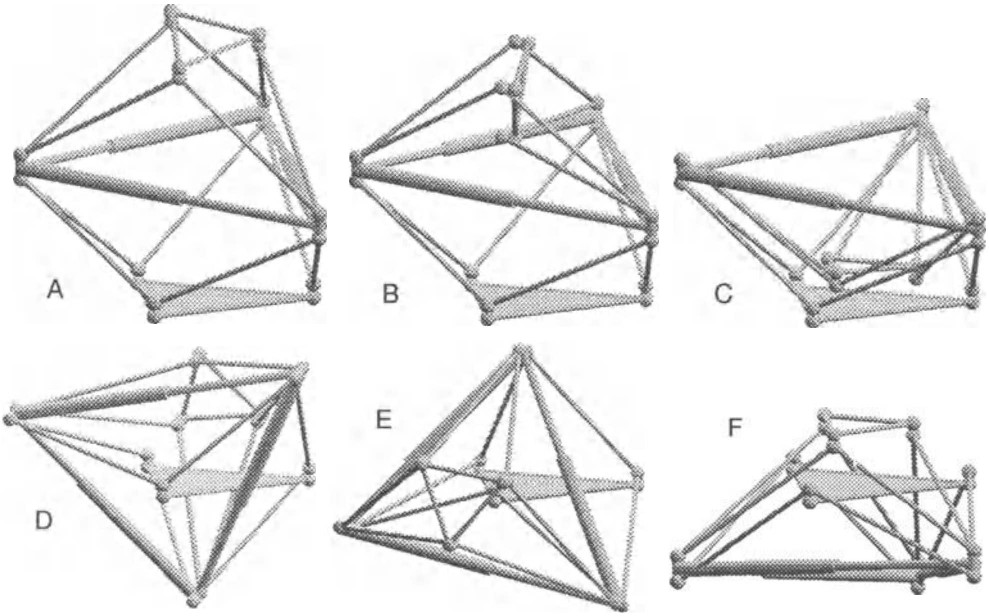


Figure 4: Tapered VGT, selected real solutions: a) 2:2, b) 2:4, c) 2:6, d) 1:2, e) 5:2, f) 6:2

Lower Octahedron Real Solutions (deg)				Upper Octahedron Real Solutions (deg)			
No.	θ_1	θ_2	θ_3	No.	ξ_1	ξ_2	ξ_3
1	20.846	48.821	−121.380	1	23.835	40.756	−133.786
2	41.633	27.824	58.309	2	38.184	26.161	54.152
3	76.706	−108.913	24.680	3	56.657	−130.150	35.968
4	107.556	−83.429	−4.210	4	131.751	−59.625	-21.706
5	−20.846	−48.821	121.380	5	−23.835	−40.756	133.7861
6	−41.633	−27.825	−58.309	6	−38.184	−26.161	−54.152
7	−76.706	108.913	−24.680	7	−56.657	130.150	−35.968
8	−107.556	83.429	4.210	8	−131.751	59.625	21.706

Conclusions

An analysis of the forward kinematics of a general octahedral variable-geometry-truss has been presented. The kinematic model featured two items of importance for robotics considerations: non-equal octahedral geometry and joint offsets. The petal angle equations for the manipulator were formulated and analyzed. Both the indirect (iterative) and direct solutions were examined for the petal angles. It was found that there exists a maximum of 256 kinematic assemblies given one set of actuator lengths. Numerical results illustrated a tapered VGT with a total of 64 solutions.

The authors are pleased to acknowledge research support from The Institute for Space and Terrestrial Science and The Natural Sciences and Engineering Research Council of Canada, and would like to thank Chris Hay and Vince Pugliese for their help with producing the figures.

References

Arun, V., Reinholtz, C. F. and Watson, L. T. 1992. Application of new homotopy continuation techniques to variable geometry trusses. *Journal of Mechanical Design - Transactions of the ASME. 114*(3):422-427.

Chirikjian, G. S. and Burdick, J. W. 1991. Parallel formulation of the inverse kinematics of modular hyper-redundant manipulators. *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*:708-713.

Griffis, M. and Duffy, J. 1989. A forward displacement analysis of a class of stewart platforms. *Journal of Robotic Systems. 6*(6):703-720.

Hughes, P. C. 1986. *Spacecraft Attitude Dynamics.* New York: John Wiley & Sons.

Hughes, P. C. Sincarsin, W. G. and Carroll, K. A. 1991. Trussarm - a variable-geometry-truss manipulator. *Journal of Intell. Mater. Syst. and Struct. 3*:148-160.

Innocenti, C. and Parenti-Castelli, V. 1990. Direct position analysis of the stewart platform mechanism. *Mechanism and Machine Theory. 25*(6):611-621.

Kendig, K. 1976. *Elementary Algebraic Geometry.* New York: Springer-Verlag.

Miura, K., Furuya, H. , and Suzuki, K. 1984. Variable geometry truss and its application to deployable truss and space crane arm. *35th Congress of the International Astronautical Federation* [IAF-84-394], Lausanne, Switzerland.

Reinholtz, C. F. and Gokhale, D. 1988. Design and analysis of variable geometry truss robots. Presented at the 10th Applied Mechanisms Conference.

Rhodes, M. D. and Mikulas, M. M. 1985. Deployable controllable geometry truss beam. NASA TM-86366.

Sincarsin, W. G. and Hughes, P. C. 1987. Trussarm: candidate geometries. Dynacon Enterprises Ltd. Report 28-611/0401.

Tidwell, P. H., Reinholtz, C. F., Robertshaw, H. H. and Horner, C. G. 1990. Kinematic analysis of general adaptive trusses. *Proceedings of the First Joint U.S./Japan Conference on Adaptive Structures*:772-791.

ANALYTICAL DETERMINATION OF THE INTERSECTIONS OF TWO COUPLER-POINT CURVES GENERATED BY TWO FOUR-BAR LINKAGES

Carlo Innocenti
Dipartimento di ingegneria delle costruzioni meccaniche
Università di Bologna - Viale Risorgimento, 2 - 40136 Bologna - Italy
Fax: +39-51-6443446 - e-mail: MECCAPP8@INGBO1.CINECA.IT

Abstract. The paper presents the analytical determination of the intersections of two coupler-point curves generated by two distinct four-bar linkages. After devising a suitable set of three compatibility equations, and performing algebraic elimination, a final polynomial equation of eighteenth order with only one unknown is obtained whose roots represent the sought-for intersections. As a result, two coupler-point curves cross each other, in the complex field, at eighteen points. The contribution is aimed at further delineating the analytical properties of four-bar linkage coupler-point curves. A case study is finally reported.

INTRODUCTION

Despite its structural simplicity, the planar four-bar linkage has been, and still is, a topical subject in kinematics. The reason can be ascribed to the fact that such a linkage, other than being very common in practice, is not so intricate as to prevent some kind of dimensional synthesis to be solved even in analytical terms.

As is known, dimensional synthesis of mechanisms represents one of the more demanding issues in kinematics. That is why, in most cases, it can only be solved numerically via optimization techniques. For the four-bar linkage, instead, kinematicians early began to develop analytical methodologies for tackling some canonical problems of dimensional synthesis. Such methodologies entailed deep knowledge of a number of geometrical loci intimately related to the kinematic features of the four- bar linkage. The properties of these loci were extensively investigated and classified, as can be traced out in every good treatise of theoretical kinematics (Hartenberg and Denavit, 1964; Hunt, 1978; Bottema and Roth, 1979; Sandor and Erdman, 1984). Moreover, those very properties have now and then been exploited in more recent studies that, focusing on new problems of synthesis (Blechschmidt and Uicker, 1986; Ting and Wang, 1991), try to solve them by more sophisticated means than the "brute force" of numerical optimization.

Among the above mentioned loci, a prominent role is played by the coupler-point curve, i.e., the trajectory a given point belonging to the coupler traces on a plane fixed to the frame. It is well known that, in terms of x-y cartesian coordinates, the coupler-point curve is

J. Angeles et al. (eds.), Computational Kinematics, 251–262.

represented by a polynomial equation of the sixth order, i.e., a sextic (see, for instance, Hunt, 1978). Moreover, the coupler-point curve is tricircular, which means that it passes through each of the two circular points at infinity three times. Additional properties of the coupler-point curve - such as freedom, class, deficiency, and number of double points - can be found in any of the above referred kinematics treatises.

The present paper is aimed at enriching the gamut of known properties concerning the coupler-point curve of the four-bar linkage by assessing the number of points where two coupler-point curves - ascribable to two distinct four-bar linkages - intersect each other.

In a previous paper (Innocenti, 1993a) the number of intersections was assessed at eighteen as a side result of the position analysis of a 7-link Assur kinematic chain. This paper corroborates that estimate by the development of a specific procedure. First, a set of three equations representing the necessary and sufficient conditions so that two coupler-point curves intersect, is devised. Then, by a single-step elimination scheme, two out of the three unknowns are dropped. As a result, a final polynomial equation of eighteenth order, free of extraneous solutions, is obtained whose eighteen roots represent, in the complex field, as many intersections of the coupler-point curves.

Finally, a case study shows application of the new procedure.

THE KINEMATIC MODEL

In this section, a set of three transcendental equations is contrived as representing the necessary and sufficient conditions so that two coupler-point curves intersect each other.

With reference to Fig. 1, two four-bar linkages $A_1B_1P_1Q_1$ and $A_2B_2P_2Q_2$ are given in

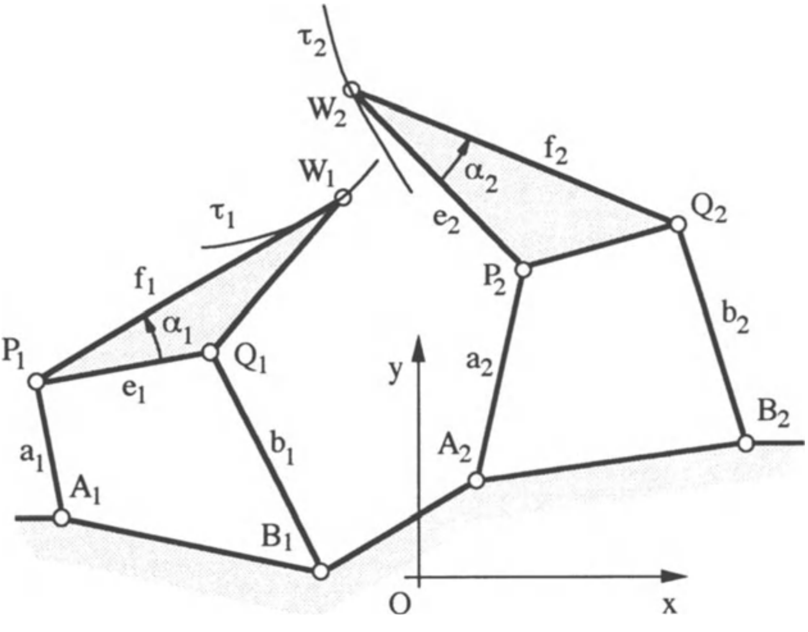


Fig. 1 - Two four-bar linkages describing two coupler-point curves.

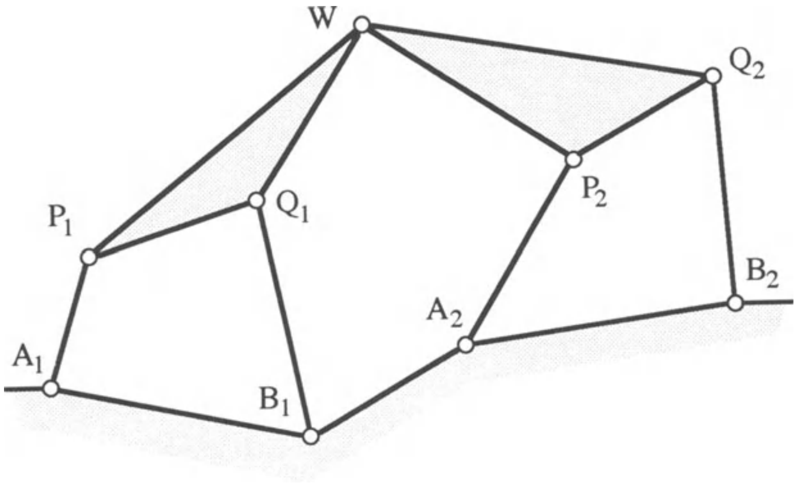


Fig. 2 - Two four-bar linkages joined at a coupler-point.

terms of lengths a_j, b_j, e_j, f_j, angles α_j, and coordinates of points A_j and B_j (j=1,2) in an arbitrary reference system Oxy fixed to the frame (note that, due to subsequent convenience, the description of the geometry of the couplers is not consistent). Points W_1 and W_2 generate the trajectories τ_1 and τ_2 whose intersections are to be determined.

Figure 2 represents the four-bar linkages with points W_1 and W_2 superimposed at W, a generic intersection of τ_1 and τ_2. It can easily be verified that the problem of finding all intersections of τ_1 and τ_2 is equivalent to determining all assembly configurations of the 7-link structure represented in Fig. 2, where links P_jQ_jW (j=1,2) are to be considered as joined at W by a revolute pair.

The assemblage of the 7-link structure of Fig. 2 can in turn be tackled by temporarily removing binary links B_1Q_1, A_2P_2, and B_2Q_2 (see Fig. 3) and looking for the configurations of the resulting three-degree-of-freedom serial open chain that prove able to place points Q_1, P_2, and Q_2 at distances respectively equal to b_1, a_2, and b_2 from points B_1, A_2, and B_2. Hence the compatibility equations for assembly of the structure represented in Fig. 2 are:

$$[(Q_1-P_1) + (P_1-A_1) - (B_1-A_1)]^2 = b_1^2 \tag{1a}$$

$$[(P_2-W) + (W-P_1) + (P_1-A_1) - (A_2-A_1)]^2 = a_2^2 \tag{1b}$$

$$[(Q_2-W) + (W-P_1) + (P_1-A_1) - (B_2-A_1)]^2 = b_2^2 \tag{1c}$$

where vectors are denoted as point differences, and the square of a vector stands for the scalar product of the vector by itself.

In order to parametrize the generic configuration of the serial chain, angles θ_0, θ_1, and θ_2 are introduced (see Fig. 3), representing respectively the orientation of links A_1P_1, P_1Q_1W,

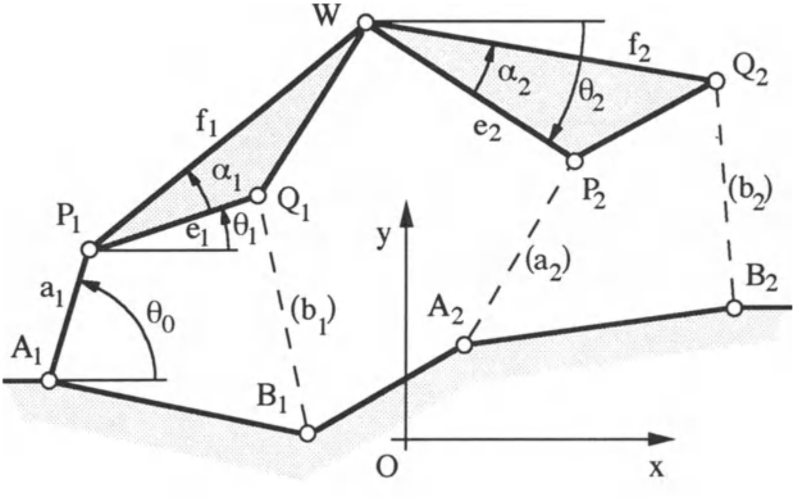


Fig. 3 - The three-degree-of-freedom serial open chain.

and P_2Q_2W with respect to the positive x-axis of reference system Oxy. Angles θ_0, θ_1, and θ_2 are positive if a counterclockwise rotation is needed to superimpose the positive x-axis of Oxy on vectors $(P_1\text{-}A_1)$, $(Q_1\text{-}P_1)$, and $(P_2\text{-}W)$ respectively.

Holding the above-stated conventions, the variable vectors appearing on the left-hand sides of equations (1) admit of the following expressions when referred to Oxy:

$$(P_1\text{-}A_1) = a_1 \cdot (c_0,\ s_0)^T \tag{2a}$$

$$(Q_1\text{-}P_1) = e_1 \cdot (c_1,\ s_1)^T \tag{2b}$$

$$(P_2\text{-}W) = e_2 \cdot (c_2,\ s_2)^T \tag{2c}$$

$$(W\text{-}P_1) = f_1 \cdot (u_1 \cdot c_1 - v_1 \cdot s_1,\ v_1 \cdot c_1 + u_1 \cdot s_1)^T \tag{2d}$$

$$(Q_2\text{-}W) = f_2 \cdot (u_2 \cdot c_2 - v_2 \cdot s_2,\ v_2 \cdot c_2 + u_2 \cdot s_2)^T \tag{2e}$$

where T is the transposition operator, and

$$c_j = \cos \theta_j; \qquad s_j = \sin \theta_j \qquad (j=0,1,2) \tag{3a}$$

$$u_j = \cos \alpha_j; \qquad v_j = \sin \alpha_j \qquad (j=1,2) \tag{3b}$$

By substituting relations (2) and (3) into equations (1), the following conditions can be obtained:

$$\sum_{i,j=0,2} l_{ij} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot s_1^{p(j)} \cdot c_1^{q(j)} = 0 \tag{4a}$$

$$\sum_{i,j,k=0,2} m_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot s_1^{p(j)} \cdot c_1^{q(j)} \cdot s_2^{p(k)} \cdot c_2^{q(k)} = 0 \tag{4b}$$

$$\sum_{i,j,k=0,2} n_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot s_1^{p(j)} \cdot c_1^{q(j)} \cdot s_2^{p(k)} \cdot c_2^{q(k)} = 0 \tag{4c}$$

where angles θ_j ($j=0,1,2$) are to be regarded as unknowns. In expressions (4), the following positions have been adopted:

$$p(i) = (i \bmod 2); \qquad q(i) = [i - p(i)]/2 \tag{5}$$

where (i mod 2) means the remainder of the division of integer i by 2. Moreover, since coefficients l_{ij}, m_{ijk}, n_{ijk} ($i,j,k=0,1,2$) can be computed directly basing on the given geometry of the four-bar linkages, they are to be considered as known. Their expressions are here omitted for the sake of conciseness.

Transcendental equations (4) represent a set of necessary and sufficient conditions for intersection of coupler-point curves τ_1 and τ_2.

SOLUTION PROCEDURE

In order to find all solutions of equation set (4), two out of the three unknowns, say θ_1 and θ_2, are eliminated by the single step procedure which was independently conceived by Lin et al. (1992) and Innocenti (1993b). Since such a procedure is algebraic, equations (4) must be first transformed so that their dependence on θ_1 and θ_2 becomes algebraic.

Substitution of θ_1 and θ_2

The well-known identities

$$c_j = (1 - t_j^2) / (1 + t_j^2); \qquad s_j = 2 \cdot t_j / (1 + t_j^2) \tag{6}$$

where $t_j = \tan(\theta_j/2)$, are substituted for $j=1,2$ throughout equations (4). After rationalization, the following expressions are obtained:

$$\sum_{i,j=0,2} L_{ij} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot t_1^j = 0 \tag{7a}$$

$$\sum_{i,j,k=0,2} M_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot t_1^j \cdot t_2^k = 0 \tag{7b}$$

$$\sum_{i,j,k=0,2} N_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot t_1^j \cdot t_2^k = 0 \tag{7c}$$

Coefficients L_{ij}, M_{ijk}, N_{ijk} (i,j,k=0,1,2) of equations (7) depend on coefficients l_{ij}, m_{ijk}, n_{ijk} (i,j,k=0,1,2) of equations (4), hence they too can be considered as known quantities.

Equation set (7) might not be equivalent to equation set (4), because rationalization has been necessary in order to attain a polynomial form in t_1 and t_2. Since rationalization required multiplication by either or both quantities $(1+t_1^2)$ and $(1+t_2^2)$, it is necessary to verify whether those factors vanish for some of the solutions of equation set (7). Should this be the case, those solutions must be considered as extraneous, and discarded.

Actually, equation set (7) does contain a couple of extraneous solutions. The remainder of the present subsection is devoted to their identification. It can be easily proved that extraneous roots arise only when the following two conditions are simultaneously satisfied:

$$1 + t_1^2 = 0 \tag{8a}$$

$$1 + t_2^2 = 0 \tag{8b}$$

Holding these conditions, and after rationalizing, only the terms of equation (4a) that contain unknown θ_1, and only the terms of equations (4b) and (4c) that contain both θ_1 and θ_2 do persist. Hence the special forms of equations (7) under conditions (8) are

$$\sum_{i=0,2;\, j=1,2} l_{ij} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot (2 \cdot t_1)^{p(j)} \cdot 2^{q(j)} = 0 \tag{9a}$$

$$\sum_{i=0,2;\, j,k=1,2} m_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot (2 \cdot t_1)^{p(j)} \cdot 2^{q(j)} \cdot (2 \cdot t_2)^{p(k)} \cdot 2^{q(k)} = 0 \tag{9b}$$

$$\sum_{i=0,2;\, j,k=1,2} n_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \cdot (2 \cdot t_1)^{p(j)} \cdot 2^{q(j)} \cdot (2 \cdot t_2)^{p(k)} \cdot 2^{q(k)} = 0 \tag{9c}$$

where the identity $(1 - t_j^2)=2$, i=1,2, (see equations (8)) has been taken into account. Now the actual expressions of the coefficients involved in equations (9) in terms of the geometry of the four-bar linkages are considered, thus obtaining from (9):

$$a_1 \cdot (c_0 + s_0 \cdot t_1) - [(B_1 - A_1)_x + (B_1 - A_1)_y \cdot t_1] = 0 \tag{10a}$$

$$f_1 \cdot e_2 \cdot [(u_1 - v_1 \cdot t_1) + (v_1 + u_1 \cdot t_1) \cdot t_2] = 0 \tag{10b}$$

$$f_1 \cdot f_2 \cdot [(u_1 - v_1 \cdot t_1) \cdot (u_2 - v_2 \cdot t_2) + (v_1 + u_1 \cdot t_1) \cdot (v_2 + u_2 \cdot t_2)] = 0 \tag{10c}$$

In equation (10a), the subscript x or y beside a vector means selection of the x or y component of that vector.

If $(f_1 \cdot e_2 \cdot f_2) \neq 0$, which is by far the most frequent case, equations (10b) and (10c) are satisfied only when $t_1 = t_2 = \sqrt{(-1)}$ or $t_1 = t_2 = -\sqrt{(-1)}$. Correspondingly, equation (10a) provides the extraneous solutions in terms of angle θ_0:

$$a_1 \cdot [c_0 + s_0 \cdot \sqrt{(-1)}] - [(B_1\text{-}A_1)_x + (B_1\text{-}A_1)_y \cdot t_1 \cdot \sqrt{(-1)}] = 0 \tag{11a}$$

$$a_1 \cdot [c_0 - s_0 \cdot \sqrt{(-1)}] - [(B_1\text{-}A_1)_x - (B_1\text{-}A_1)_y \cdot t_1 \cdot \sqrt{(-1)}] = 0 \tag{11b}$$

By multiplying side by side equations (11), one can obtain:

$$2 \cdot a_1 \cdot [(B_1\text{-}A_1)_x \cdot c_0 + (B_1\text{-}A_1)_y \cdot s_0] - a_1^2 - (B_1\text{-}A_1)^2 = 0 \tag{12}$$

or, in terms of $t_0 = \tan(\theta_0/2)$:

$$[a_1^2 + (B_1\text{-}A_1)^2 + 2 \cdot a_1 \cdot (B_1\text{-}A_1)_x] \cdot t_0^2 - 4 \cdot a_1 \cdot (B_1\text{-}A_1)_y \cdot t_0 + \\ [a_1^2 + (B_1\text{-}A_1)^2 - 2 \cdot a_1 \cdot (B_1\text{-}A_1)_x] = 0 \tag{13}$$

Equation (13) provides all extraneous solutions of equation set (7) in terms of t_0 values.

Elimination of t_1 and t_2

Unknowns t_1 and t_2 are now eliminated from equation set (7) by a single-step procedure that, as shown in (Innocenti, 1993b), does not of itself introduce extraneous solutions. Since equation set (7) does contain a couple of extraneous solutions (see equation (13)), what can be expected as a result of elimination is an equation in the only unknown θ_0 that contains, besides the right solutions, the couple of extraneous roots provided by equation (13).

According to the procedure explained in (Innocenti, 1993b), the eliminant of equations (7) can be computed as the determinant of the following 16x16 matrix D:

$$D = \begin{bmatrix}
E_0 & E_1 & E_2 & `0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & E_0 & E_1 & E_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & E_0 & E_1 & E_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & E_0 & E_1 & E_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_0 & E_1 & E_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_0 & E_1 & E_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_0 & E_1 & E_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_0 & E_1 & E_2 \\
F_{00} & F_{10} & F_{20} & 0 & F_{01} & F_{11} & F_{21} & 0 & F_{02} & F_{12} & F_{22} & 0 & 0 & 0 & 0 & 0 \\
0 & F_{00} & F_{10} & F_{20} & 0 & F_{01} & F_{11} & F_{21} & 0 & F_{02} & F_{12} & F_{22} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & F_{00} & F_{10} & F_{20} & 0 & F_{01} & F_{11} & F_{21} & 0 & F_{02} & F_{12} & F_{22} & 0 \\
0 & 0 & 0 & 0 & 0 & F_{00} & F_{10} & F_{20} & 0 & F_{01} & F_{11} & F_{21} & 0 & F_{02} & F_{12} & F_{22} \\
G_{00} & G_{10} & G_{20} & 0 & G_{01} & G_{11} & G_{21} & 0 & G_{02} & G_{12} & G_{22} & 0 & 0 & 0 & 0 & 0 \\
0 & G_{00} & G_{10} & G_{20} & 0 & G_{01} & G_{11} & G_{21} & 0 & G_{02} & G_{12} & G_{22} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & G_{00} & G_{10} & G_{20} & 0 & G_{01} & G_{11} & G_{21} & 0 & G_{02} & G_{12} & G_{22} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{00} & G_{10} & G_{20} & 0 & G_{01} & G_{11} & G_{21} & 0 & G_{02} & G_{12} & G_{22}
\end{bmatrix} \tag{14}$$

where:

$$E_j = \sum_{i=0,2} L_{ij} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \qquad (j=0,1,2) \qquad (15a)$$

$$F_{jk} = \sum_{i=0,2} M_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \qquad (j,k=0,1,2) \qquad (15b)$$

$$G_{jk} = \sum_{i=0,2} N_{ijk} \cdot s_0^{p(i)} \cdot c_0^{q(i)} \qquad (j,k=0,1,2) \qquad (15c)$$

As a consequence, the following equation

$$\det D = 0 \qquad (16)$$

represents the result of eliminating unknowns t_1 and t_2 from the equation set (7).

The Final Equation

If coefficients L_{ij}, M_{ijk}, and N_{ijk} $(i,j,k=0,1,2)$ were arbitrary quantities, one would expect the substitution of relations (15) and (14) into equation (16) to lead to an equation containing, at worst, terms so involved as $(s_0 \cdot c_0^{15})$ and (c_0^{16}). However, the above-mentioned coefficients, likewise coefficients of equations (4) from which they derive, are not completely independent. Accordingly, some reduction in complexity might be possible.

Actually, direct computation shows that expansion of equation (16) results in the following condition:

$$\sum_{i=0,20} H_i \cdot s_0^{p(i)} \cdot c_0^{q(i)} = 0 \qquad (17)$$

where coefficients H_i $(i=0,..,20)$ are functions of the geometry of the couple of four-bar linkages only.

If positions (6) for $j=0$ are inserted into equation (17), one can obtain:

$$\sum_{i=0,20} h_i \cdot t_0^i = 0 \qquad (18)$$

where coefficients h_i $(i=0,..,20)$ are functions of coefficients H_i only.

Equation (18) still contains the couple of extraneous roots implicitly defined by equation (13). If the left-hand side of equation (18) is divided by the left-hand side of equation (13), the following final polynomial equation is obtained

$$\sum_{i=0,18} r_i \cdot t_0^i = 0 \qquad (19)$$

which is free from extraneous roots, and whose coefficients r_i $(i=0,..,18)$ depend on the geometry of the four-bar linkages only. Equation (19) provides eighteen roots for t_0 in the complex field.

Back Substitution

For every root t_{0i} ($i=1,..,18$) of equation (19), the corresponding value θ_{0i} for θ_0 can be computed. Furthermore, matrix D expressed by relation (14) can be considered (Innocenti, 1993b) as the coefficient matrix of a homogeneous linear system, which is surely endowed with a nontrivial solution vector (see equation (16)). The ratios of the second and fifth components of the solution vector to the first component provide the values t_{1i} and t_{2i} for t_1 and t_2 respectively. Hence also values θ_{1i} and θ_{2i} for θ_1 and θ_2 are determined.

As a consequence, for every root of equation (19) an assembly configuration of the structure represented in Fig. 2 is found, and also the position of point W is determined (see equations (2a) and (2d)). This proves that the number of intersections of two coupler-point curves generated by two distinct four-bar linkages amounts - in the complex field - to eighteen.

NUMERICAL EXAMPLE

With reference to Fig. 1, two four-bar planar linkages are considered as having the following dimensions (lengths are given in arbitrary length unit, while angles are measured in radians):

$A_1 =$	(0., 0.)	$A_2 =$	(0.55, 1.2)
$B_1 =$	(2., 0.5)	$B_2 =$	(-1.342, 0.41)
$a_1 =$	1.975	$a_2 =$	2.19
$b_1 =$	2.11	$b_2 =$	2.108
$e_1 =$	1.93	$e_2 =$	2.33
$f_1 =$	3.355	$f_2 =$	2.577
$\alpha_1 =$	0.775	$\alpha_2 =$	0.824

Figure 4 represents the resulting coupler-point curves in reference system Oxy. Both curves are unicursal because the four-bar linkages are of non-Grashof type (Hunt, 1978). Due to the particular geometry here considered, all eighteen intersections are real. The coordinates in Oxy of each intersection point W are listed in Table 1, together with the corresponding locations of points P_1, Q_1, P_2, and Q_2. The numeration of intersection points in Fig. 4 and Table 1 is consistent.

REMARKS

Coupler-point curves of planar four-bar linkages are sextics and - according to Bezout's theorem - should intersect at 6x6=36 points in the complex field (including points at infinity, and counting each intersection with its own multiplicity). However, the foregoing proves that the actual number of finite intersections is eighteen, which means that as many as eighteen intersections go to infinity.

This result is not self-evident: as a matter of fact, any coupler-point curve satisfies three times each of the two circular points at infinity (Hunt, 1978), which implies that two coupler-point curves should cross each other no less than three times at the above-mentioned circular points at infinity. This would account for an upper bound of 36-2x3=30 finite intersections. Actually, what happens is that two coupler-point curves intersect nine times at each of the circular points at infinity, which accounts for a set of eighteen finite intersections.

A final remark relates the number of intersections of two coupler-point curves generated by distinct points belonging to the same coupler: it has been verified that such a number amounts to fourteen.

CONCLUSIONS

The determination of all intersections of two coupler-point curves traced out by two distinct planar four-bar linkages, is presented. The analysis, performed in analytical form by an original procedure, results in an eighteenth-order polynomial equation whose eighteen roots represent, in the complex field, as many intersections of the considered coupler-point curves.

The contribution widens the set of known properties pertaining to the sextic generated by a coupler-point of a four-bar linkage.

A numerical example is finally reported.

Acknowledgments

The financial support of MURST by funds 60% is gratefully acknowledged.

REFERENCES

Blechschmidt, J.L., Uicker, J.J., Jr., 1986, "Linkage Synthesis Using Algebraic Curves," *ASME Journal of Mechanisms, Transmissions, and Automation in Design,* Vol. 108, pp. 543-548.

Bottema, O., Roth, B., 1979, *"Theoretical Kinematics,"* North-Holland, Amsterdam.

Hartenberg, R.S., Denavit, J., 1964, *"Kinematic Synthesis of Linkages,"* McGraw-Hill, New York.

Hunt, K.J., 1978, *"Kinematic Geometry of Mechanisms,"* Clarendon Press, Oxford.

Innocenti, C., 1993a, "Polynomial Solution to the Position Analysis of the 7-Link Assur Kinematic Chain with One Quaternary Link," submitted to *Mechanism and Machine Theory.*

Innocenti, C., 1993b, "Analytical-Form Position Analysis of the 7-Link Assur Kinematic Chain with Four Serially-Connected Ternary Links," accepted for publication on the *ASME Journal of Mechanical Design.*

Lin, W., Crane, C.D., Duffy, J., 1992, "Closed-Form Forward Displacement Analyses of the 4-5 In-Parallel Platforms," *Proceedings of the 22nd Biennial ASME Mechanisms Conference,* Scottsdale, AZ, Sept. 13-16, pp. 521-527.

Sandor, G.N., Erdman, A.G., 1984, *"Advanced Mechanism Design: Analysis and Synthesis,"* Vol. 2, N.J., Prentice-Hall.

Ting, K.L., Wang, S.C., 1991, "Fourth and Fifth Order Double Burmester Points and the Highest Attainable Order of Straight Lines," *ASME Journal of Mechanical Design,* Vol. 113, pp. 213-219.

Table 1. Coordinates of points W, P_1, Q_1, P_2, and Q_2 in reference system Oxy.

#	W	P_1	Q_1	P_2	Q_2
1	0.0478676553	-1.6140878759	0.2420559439	-1.3888757233	0.4572798983
	4.0525669477	1.1381323863	1.6669330296	2.2182636837	1.5082967939
2	0.8771218818	1.8144582660	2.7259091679	-1.0514144729	0.4894895677
	4.0013795043	0.7799783336	2.4812006158	2.6938445990	1.4537001310
3	-1.3828396355	-1.4048246504	-0.0453662532	-1.0132217743	0.7620877697
	1.9667303213	-1.3881976451	-0.0182440449	-0.3337658506	0.5383692232
4	-1.1582848156	-1.5478672410	-0.0464407875	-1.1138727999	0.7659184793
	2.1056289050	-1.2266751829	-0.0139845361	-0.2239477890	0.3914612703
5	-1.1495624053	1.8676185210	1.2181941599	-1.1170522652	0.7658584728
	2.1095489961	0.6423597589	2.4598162231	-0.2202241883	0.3855733989
6	0.0150379424	-1.7843471452	0.0949546355	-1.3358845015	0.5409497129
	1.9850501614	-0.8465992354	-0.4071395478	0.0866538512	-0.5377153469
7	0.7146932960	-1.1195635300	0.7650278989	-1.3965218095	-0.0713808243
	1.1821635745	-1.6270225882	-1.2108313504	0.1964299501	-1.2720198900
8	0.8630183558	-0.9796124843	0.9062124056	0.4934126806	-1.2819014831
	1.0887706695	-1.7149298471	-1.3043637932	3.3892687992	2.5171431295
9	2.1930193947	-1.0590448260	-0.0544676885	1.6322786938	-0.0638628423
	0.8423642463	1.6670480067	0.0191023841	3.1038836175	2.0863142325
10	2.3309129758	0.3933607366	2.2921380309	1.7191731426	0.0464769228
	0.8035318489	-1.9354307869	-1.5896782936	3.0517921489	1.9961260463
11	2.8820163276	0.8912629152	2.7964396061	1.7972245607	0.3933008239
	0.9380788864	-1.7624628836	-1.4539150324	3.0001474649	1.6068270764
12	-1.2811287219	1.8678395196	0.1077276564	-1.5568441591	-3.3661668449
	-0.5158772023	0.6417168605	1.4334909628	1.7977521971	0.9985682492
13	-1.1679011257	1.8644131900	0.0403201540	1.1590290771	0.6773727841
	-0.7840468570	0.6516045250	1.2821476212	-0.9036120325	1.0148120028
14	-0.4293086474	1.8215937150	-0.1048913858	-1.4720601199	-2.9039506444
	-1.7246979100	0.7631653409	0.6467387270	0.3589453816	-1.0056179514
15	1.7534922640	0.8710271242	-0.0691982465	2.6761911666	0.7101472470
	-1.4643118056	1.7725509157	0.0870610011	0.6752037319	0.8920328583
16	-1.4134014700	1.7418507384	-0.0139044644	-1.4273858001	-3.3150539081
	-2.0712578782	-0.9309033275	-0.1295147402	0.2587001554	-0.3321066471
17	-1.2146084112	1.8163109388	-0.0083932843	-1.3504971847	-3.2046628963
	-2.2142483560	-0.7756542875	-0.1468820725	0.1117856595	-0.5769908483
18	1.7005820711	1.9542752904	0.5034272019	1.5789898562	-0.2793735287
	-3.0600297231	0.2853648352	-0.9874037313	-0.7332045613	-1.4105738059

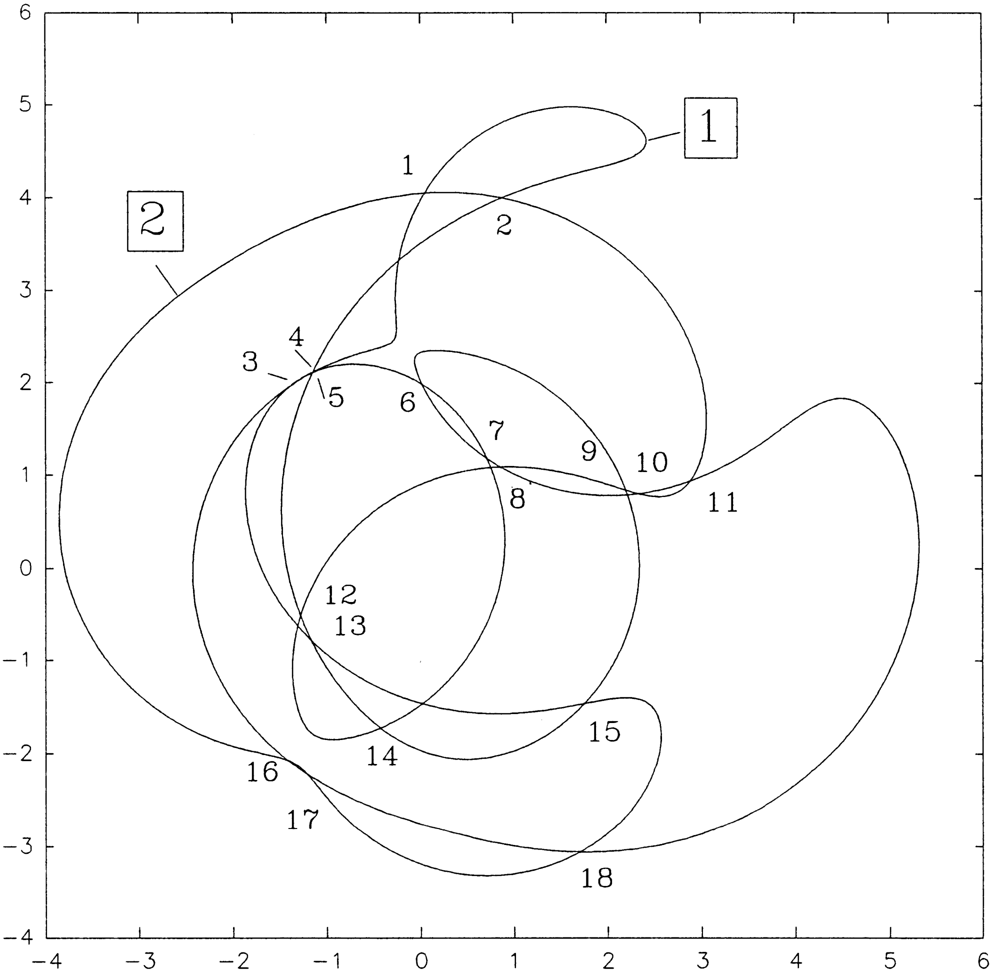


Fig. 4 - Two coupler-point curves intersecting at eighteen real points.

ON CLOSED FORM SOLUTIONS OF MULTIPLE-LOOP MECHANISMS

Andrés Kecskeméthy, Dr. Ing.
Fachgebiet Mechatronik
Universität — GH — Duisburg
Lotharstraße 1, D-47058 Duisburg, Germany
email: kecs@mechatronik.uni-duisburg.de

Abstract

Discussed in this paper is a method for the automatic generation of symbolic equations for multiple-loop mechanisms whose kinematics can be solved in closed form. The method is based on geometric and topological properties of the system which are invariant to coordinate transformations. It focuses on the treatment of the individual multibody loops as transmission elements which encompass the solution of the local nonlinear constraint equations and which can be assembled by linear equations to yield general mechanisms. The global processing is obtained as a combination of algorithms for generating the local kinematics of the individual loops, detecting a suitable set of independent loops, and finding an optimal "solution flow" in the resulting kinematical block-diagram which represents the order in which the equations are to be solved. An example processed by the current implementation of the method with the symbolic-computation language *Mathematica* illustrates the basic ideas and the scope of the approach.

1 Introduction

Symbolical solutions of kinematics have several advantages, among which are high efficiency, determinism in computation time and high precision. Such solutions exist for many technical applications, such as vehicle dynamics, robotics and mechanism theory, including the spatial case (Hiller et al. 1986).

For single-loop mechanisms, which imply also the problem of inverse kinematics in robotics, there has been strong research in this direction, yielding a great spectrum of solution methods, from fully-automatic schemes for picking out a suitable set of equations out of a large set of equations obtained by re-ordering the algebraic equations of closure (Mehner 1990) to pre-processed schemes in which particular solutions for the most general cases of robot configurations are parameterized (Paul and Zhang 1986, Lloyd and Hayward 1988). Also, some authors have described the general geometric criteria which must be fulfilled for the existence of closed-form solutions in single-loop systems (Heiß 1987, Woernle 1988). These criteria are of great importance, because they enable the experienced kinematician to find by simple inspection particular solutions which are generally more efficient then those generated by purely algebraic computation schemes (Hunt 1986). Recently, a method was developed for combining the geometric criteria with the algebraic procedure,

J. Angeles et al. (eds.), Computational Kinematics, 263–274.

yielding a very compact algorithm suited for detecting and generating closed-form solutions of general single-loop spatial mechanisms (Kecskeméthy and Hiller 1992). A similar method was also proposed by Fanghella 1988.

For the general spatial multiple-loop case, there has not been must research along these lines. As a consequence, it is still customary to apply iterative methods (e. g. Uicker et al. 1964) when multiple-loops arise, even though in many industrial applications, especially in robotics and mechanism design, closed-form solutions exist. The idea of the present paper is to dissect a general multiple-loop mechanism into a system of coupled single-loop mechanisms, as explained in Hiller and Kecskeméthy 1989. Then, it is possible to apply the well-known algorithms for solving single-loop mechanisms to the individual loops, and the overall kinematics are obtained by simple concatenation of these solutions.

Section **2** discusses the method adopted here for the generation of symbolical equations of a single loop, while Section **3** addresses the problem of selecting a suitable set of mulibody loops, to which this algorithm for single-loop mechanisms can be applied. Then, one only needs to state the coupling equations between the loops and to specify orientations for the edges of the resulting block-diagram, called the *solution flow*, as described in Section **4** and Section **5**, respectively. This yields a set of explicitly resolvable equations for the cases where closed-form solutions exist.

2 Recognizing Closed-Form Solutions in a Single Loop

A single multibody loop can be modeled as a sequence of homogeneous transformations A_i , $i = 1, \ldots, n$, between reference frames $\mathcal{K}_{i-1}$ and $\mathcal{K}_i$, respectively, such that $\mathcal{K}_n \equiv \mathcal{K}_0$ (Fig. 1), i. e. such that the closure condition

$$\mathrm{A}_1 \mathrm{A}_2 \cdots \mathrm{A}_n = \boldsymbol{I}_4 \, . \tag{1}$$

is fulfilled.

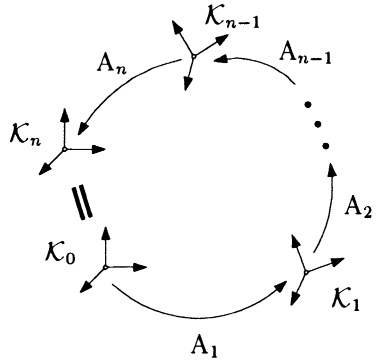


Figure 1: Basic structure of a loop

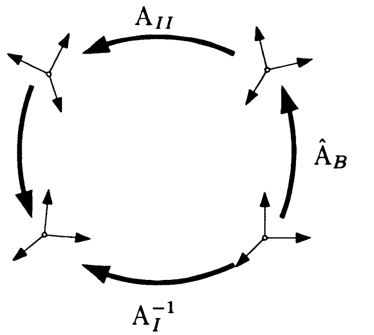


Figure 2: Grouping of Transformations

The homogeneous transformation matrices A_i are constant transformations for rigid links and variable transformations for joints, and have the structure

$$\mathrm{A}_i = \begin{bmatrix} {}^{i-1}\boldsymbol{R}_i & {}^{i-1}_{i-1}\underline{r}_i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} & r_1 \\ \rho_{21} & \rho_{22} & \rho_{23} & r_2 \\ \rho_{31} & \rho_{32} & \rho_{33} & r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} , \tag{2}$$

where ${}^{i-1}\boldsymbol{R}_i$ is the orthogonal 3×3 matrix of rotation transforming vector components from $\mathcal{K}_i$ to $\mathcal{K}_{i-1}$, and ${}^{i-1}_{i-1}\underline{r}_i$ is the radius vector from the origin $\mathcal{O}_{i-1}$ of $\mathcal{K}_{i-1}$ to the origin $\mathcal{O}_i$ of $\mathcal{K}_i$ in the decomposition with respect to $\mathcal{K}_{i-1}$ (the indices i and $i-1$ have been left out in coefficient-wise notation for better clarity). Thus, from the closure condition twelve non-trivial scalar equations result, of which of course six are dependent because of the orthogonality condition of the rotational part.

For an optimal statement of closure conditions, one tries to find one equation which contains as few unknowns as possible, then a next one containing one unknown more, and so on, until there is a set of equations having as much of a triangular shape as possible for determining the six unknowns of the loop. One way of achieving this is to state all possible alternative forms of the closure condition

$$\mathrm{A}_{i_{j+1}} \cdots \mathrm{A}_{i_k} = \mathrm{A}_{i_j}^{-1} \cdots \mathrm{A}_{i_1}^{-1}\, \mathrm{A}_{i_n}^{-1} \cdots \mathrm{A}_{i_{k+1}}^{-1} \tag{3}$$

where $1 < j < k < n$ and $i_1, \ldots, i_n$ is a cyclic permutation of $1, \ldots, n$, and extract from the resulting set of $12n^2$ equations the ones which are most simple. However, because the number of terms in some of the coefficients of the resulting matrix can grow up to $2^{(n-1)}$, such an approach would not be feasible already for modestly sized loops involving e. g. 10 transformations.

An alternative approach is to group the transformations into four particular sequences A_I , A_{II} , $\hat{\mathrm{A}}_A$ and $\hat{\mathrm{A}}_B$, where $\hat{\mathrm{A}}_A$ and $\hat{\mathrm{A}}_B$ are characterized by the fact that they are the longest possible sequences of transformation leaving one of the geometric elements *point*, *line*, *plane* or *direction* invariant (Fig. 2). Let $\underline{\xi}_A$ and $\underline{\xi}_B$ denote two such invariant geometric elements. Then, any measurement taken between these two elements will not depend on the transformation sequences $\hat{\mathrm{A}}_A$ and $\hat{\mathrm{A}}_B$ or the unknown variables contained therein, which are thus eliminated. To see this, let the closure condition be re-stated as

$$\hat{\mathrm{A}}_B\, \mathrm{A}_{II}\, \hat{\mathrm{A}}_A = \mathrm{A}_I^{-1} \quad , \tag{4}$$

and let the measurement be represented as a projection operator $\pi(\underline{\xi}_B, \underline{\xi}_A; \mathrm{A})$ where A is the homogeneous transformation describing the motion of the reference system holding $\underline{\xi}_A$ with respect to the reference system holding $\underline{\xi}_B$. Then, after applying the projection operator to both sides of Eq. (4), one obtains the scalar equation

$$\pi(\underline{\xi}_B, \underline{\xi}_A; \mathrm{A}_{II}) = \pi(\underline{\xi}_B, \underline{\xi}_A; \mathrm{A}_I^{-1}) \tag{5}$$

which does not depend on the unknowns contained in $\hat{\mathrm{A}}_A$ and $\hat{\mathrm{A}}_B$.

The recognition of appropriate sequences $\hat{\mathrm{A}}_A$ and $\hat{\mathrm{A}}_B$ is easy, because transformations leaving any geometric elements invariant conform groups denominated the *isotropic subgroups* (Olver 1986) of rigid-body motion. Thus, one can first decompose the transformations within the loop into elementary transformations, then put down in a matrix which geometric elements are invariant with respect to the individual transformations and finally search for the longest sequence of contiguous entries to find the largest transformation sequences leaving one geometric element invariant, together with this element. By applying this scheme repeatedly, one obtains expressions for the remaining unknows in the loop, until no unknowns remain.

The approach is explained in more detail in Kecskeméthy and Hiller 1992. An implementation exists (in Mathematica) which can also handle simple overconstrained cases as e. g. planar and spherical mechanisms or cardan mechanisms.

3 Decomposition of Multiple-Loop Systems into Suitable Sets of Loops

In connected multiple-loop systems, there exist $n_L = n_{\mathcal{B}} - n_G$ independent loops, where $n_{\mathcal{B}}$ is the number of bodies (excluding the inertial body) and n_G is the number of *binary* joints. The number $f_{\mathcal{G}_i}$ of degrees of freedom of the joints plays no role, but it is important that in this context a joint connects exactly two bodies. Thus in the case of *multiple joints* connecting $n_{\mathcal{B}}(\mathcal{G}_i)$ bodies, these must be replaced by $n_{\mathcal{B}}(\mathcal{G}_i)\,(\,n_{\mathcal{B}}(\mathcal{G}_i) - 1\,)/2$ binary joints for the above formula to apply.

While the number of independent loops is unique, this is not the case for the set of loops itself, as there are infinitely many ways of choosing a suitable set of n_L loops where no loop results as a combination of the other loops. It is obvious that depending on the choice of loops the kinematics will be more or less simple to solve. Thus, an important task of the processing will be to determine a set of loops which is advantageous for the resolution of the kinematics. A quite heuristic criterion for this purpose is that the loops should be kept as small as possible, where "small" refers to the number of joint coordinates within the loop: as will be shown in Section **4**, the sum of degrees of freedom over all loops minus the number of coupling conditions between the loops is an invariant which represents the number of degrees of freedom of the system. Thus if a small loop is replaced by a larger loop, the number of coupling conditions between this loop and the others will increase, and the overall resolution of equations becomes more complicated.

The determination of the smallest set of loops is possible using some results from graph theory. To this end, a graph is built where the vertices correspond to the bodies and the edges correspond to binary joints. A cycle is a set of edges and vertices such that each vertex is connected to an even number of edges. Note that a cycle can embrace more than one multibody loop. We will exclude these cases implicitly. The sum of two cycles C and D is defined as their symmetric difference $C + D = (C \cup D) - (C \cap D)$. The set of cycles in a graph is closed under the operation of addition. A set of cycles is independent if no cycle in this set results from the addition of other cycles in this set. A set of cycles is a basis if any cycle which is not an element of this set can be generated by the sum of some of the cycles in this set.

A special kind of cycle basis is the fundamental cycle set. In a fundamental cycle set, it is possible to find one edge for each cycle such that removal of all of these edges leads to a connected tree, and re-insertion of each of these edges closes exactly one of these cycles, which is the one to which the edge was associated (Horton 1987). This remaining graph is termed the spanning tree of the graph, and the set of removed edges the corresponding cotree (Andrews and Kesavan 1975). Clearly, a cycle basis in which there exists one cycle whose every edge is also part of some other cycle can not be a fundamental cycle basis, because removal of any of the edges of the internal cycle will also open the surrounding cycles. Thus fundamental cycle sets are special cases of cycle bases, and the set of smallest cycles, the *minimal cycle basis*, needs not always to correspond to a fundamental cycle set.

While polynomial-time algorithms for the determination of minimal cycle bases exist (Hubicka and Syslo 1975, Horton 1987), the problem of determining a fundamental set of minimal cycles is solvable only by trying all possibilities. In the present modeling no requirements have been made that the multibody loops form a fundamental set. Thus it is possible to use the minimal cycle-basis algorithms for determining a suitable set of loops. The algorithm described next for this purpose is a combination of the methods described in Horton 1987 and Hiller and Anantharaman 1989. Its computational effort is roughly $O\,(\,n_{\mathcal{B}}^3\,)$.

The multibody system is mapped to a graph where the edges are weighted according to the number of degrees of freedom of the joint. Multiple joints are replaced by their corresponding complete graphs, which result by supplying an edge for every two bodies connected by the joints. These edges are marked especially, allowing to replace them after the determination of loops again by a multiple joint. The algorithm works with a *path matrix* Σ having elements σ_{ij} expressing the length of the shortest path from vertex V_i to vertex V_j, and the *successor matrix* Γ having elements γ_{ij} which denote the successor of the vertex V_i on the path from V_i to V_j.

Initially, only the entries related to the existing edges are set in matrices Σ and Γ, while all other elements are set to "infinity" and "0", respectively; also, vertices and edges are marked as "unvisited", vertices also as "unfinished"; finally, the vertex corresponding to the root is marked as "visited".

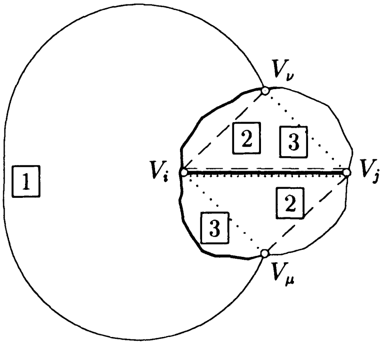


Figure 3: Extraction of shortest paths

The algorithm now works through the set of all vertices marked as "visited" but also as "unfinished". For each vertex V_i with these properties, each edge starting at V_i and marked "unvisited" is regarded; the edge is marked "visited" and now its end vertex V_j is considered, whereby two situations can occur:

(A) The vertex V_j is marked "unvisited". Then, set $\sigma_{\nu j} = \sigma_{\nu i}+\sigma_{ij}$ and $\gamma_{\nu j} = \gamma_{\nu i}$ for all vertices V_ν for which $\sigma_{\nu i} \neq$ infinity, i.e. for those vertices V_ν for which a connection to V_i already exists; mark the vertex V_j as "visited", and, if no "unvisited" edges starting at V_j remain, also as "finished".

(B) The vertex V_j is "visited". Then, there exist already paths between V_i and V_j and the edge connecting V_i and V_j is a *chord* closing one or more loops. Choose the loop resulting from the shortest existing path between V_i and V_j (this is the one whose length is stored in σ_{ij} and which can be reconstructed by following the vertices $i_1 = \gamma_{ij}$, $i_2 = \gamma_{i_1 j}$, ...). This loop is retained for the minimal cycle basis. Additionally, it must be verified if the edge connecting V_i and V_j may shorten existing paths. For this purpose, all already visited vertices V_ν and V_μ are checked out, comparing (Fig. 3) $\boxed{1}$ the existing path length $\sigma_{\nu\mu}$ with $\boxed{2}$ the length of the path from V_ν through V_i and V_j to V_μ, and $\boxed{3}$ the length of the path from V_ν through V_j and V_i to V_μ. From these, the shortest path is retained.

Note that the overall computational effort of the algorithms is determined by the comparision of path lengths in step (B).

4 Determination of Loop-Coupling Conditions

When multiple loops occur, one can take each loop by turn and investigate its local kinematics independently of the rest of the system. Assume that the local kinematics of such a loop has already been solved. Then, six of the joint coordinates will be known as functions of the other f_L joint coordinates, the same holding also for their respective first and second time derivatives. This yields the model of an individual multibody loop as a nonlinear transmission element denominated "kinematical transformer" which maps f_L input variables $\underline{q}$ to 6 output joint coordinates $\underline{\beta}^{\text{exi}}$ (Fig. 4).

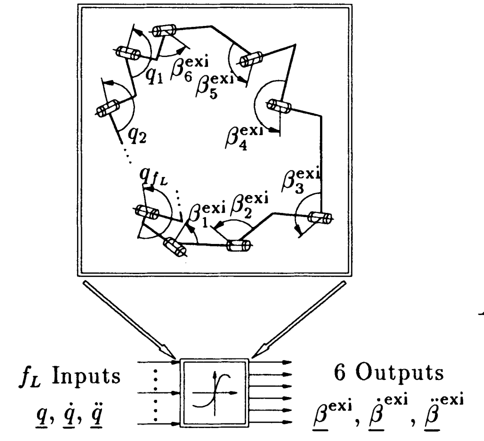


Figure 4: "*Kinematical transformer*"

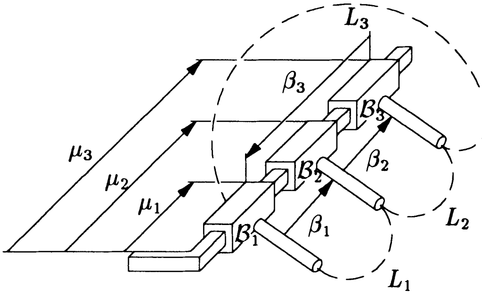


Figure 5: Multiple joint

The joint variables within a loop are denoted by $\beta_1 , \dots , \beta_{n_\beta(L)}$ when it is not clear which of the coordinates to use as inputs and which as outputs. It is clear that by modeling each loop independently of the others a redundant set of joint coordinates is introduced and that a dependency between the joint coordinates defined in the different loops will result. This dependency can be formulated quite simply if one regards only elementary joints (i. e. prismatic (P), revolute (R) or screw (H) joints) as joints where a coupling can occur. This pre-assumption is not very severe because, as it is known, all technical joints can be decomposed into a sequence of elementary joints. However, it will be allowed that multiple joints $\mathcal{G}$ connecting $n_\mathcal{B}(\mathcal{G}) \geq 2$ bodies $\mathcal{B}_i$ occur (Fig. 5).

Now, assume that the joint is part of $n_L(\mathcal{G})$ independent loops L_i. Let μ_i denote the position of body $\mathcal{B}_i$ in the natural coordinate system of the joint with respect to a given reference point, and β_k a relative joint coordinate defined in loop L_k describing the relative position of bodies $\mathcal{B}_{i(k)}$ and $\mathcal{B}_{j(k)}$. Each loop connected to the joint introduces a linear equation defining a relative joint coordinate as the difference of the relative position of two of the bodies connected by the joint:

$$\mu_{i(k)} - \mu_{j(k)} = \beta_k + \alpha_k \quad , \quad k = 1 , \dots , n_L(\mathcal{G}) \quad . \tag{6}$$

After collecting all of these equations, and supplementing them with a equation for the definition of the reference point within the joint, e. g.

$$\mu_1 = 0 \quad , \tag{7}$$

one obtains a system of $n_L(\mathcal{G}) + 1$ linear equations

$$P\,\underline{\mu} = \underline{\hat{\beta}} + \underline{\hat{\alpha}} \tag{8}$$

with

$$\underline{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{n_B(\mathcal{G})} \end{bmatrix} \quad , \quad \underline{\hat{\beta}} = \begin{bmatrix} 0 \\ \beta_1 \\ \vdots \\ \beta_{n_L(\mathcal{G})} \end{bmatrix} \quad , \quad \underline{\hat{\alpha}} = \begin{bmatrix} 0 \\ \alpha_1 \\ \vdots \\ \alpha_{n_L(\mathcal{G})} \end{bmatrix} \tag{9}$$

defining $\underline{\mu}$ as linear function of $\underline{\hat{\beta}}$. Note that P is a $(n_L(\mathcal{G}) + 1) \times n_B(\mathcal{G})$ matrix, where, in general $(n_L(\mathcal{G}) + 1) \neq n_B(\mathcal{G})$. Thus, in order for Eq. (8) to have a solution, it must be ensured that the vector $\underline{\hat{\beta}} + \underline{\hat{\alpha}}$ lies in the range of matrix P. This condition can be expressed with the help of the orthogonal complement B of P (Halmos 1987), defined as

$$B^{\mathrm{T}} P = 0 \quad . \tag{10}$$

It follows

$$B^{\mathrm{T}}\,(\underline{\hat{\beta}} + \underline{\hat{\alpha}}) = 0 \quad . \tag{11}$$

The number of independent columns of matrix B and thus the number of linear equations which the joint coordinates $\underline{\hat{\beta}}$ must fulfill is equal to $n_L(\mathcal{G}) + 1 - r$, where r is the rank of P. This rank is equal to $n_B(\mathcal{G})$ minus the increase of the number of connected components resulting from the original system when the joint is removed. Thus, for a 2-connected graph, i. e. one in which at least two joints have to be removed in order for the mechanism to fall into two parts (which is the normal case in mechanism analysis), the rank of P is $n_B(\mathcal{G})$ and the number of coupling conditions at a joint can be determined by the formula:

$$p(\mathcal{G}_i) = n_L(\mathcal{G}_i) - n_B(\mathcal{G}_i) + 1 \quad . \tag{12}$$

It is interesting to note that matrix P, after removing the first line, can be interpreted as the incidence matrix of a graph whose nodes represent the coordinates μ_i and the edges the variables β_k directed from $\mu_{j(k)}$ to $\mu_{i(k)}$. Then, matrix B^{T} defined by Eq. (10) corresponds just to the cycle matrix of this graph. Thus one can obtain the coupling conditions using the same algorithms as described in Section **3**. Note also that from this property it follows directly that the elements of matrix B are only $+1$, -1, 0, so that the coupling conditions are just signed sums of joints variables.

It can be shown that the counting rule (12) yields enough coupling conditions between the independently modeled loops so as to allow to assemble them correctly to general systems (Kecskeméthy 1993). Indeed, the following identity holds:

$$f = \sum_{j=1}^{n_L} f_{L_j} - \sum_{i=1}^{n_{\hat{G}}} p(\mathcal{G}_i) \quad , \tag{13}$$

where f are the global degrees of freedom of the mechanism, f_{L_j} is the local degree of freedom of loop L_j, and $n_{\hat{G}}$ denotes the number of joints, including muliple joints. Thus, dissecting a general multibody system into individual loops and then formulating the coupling conditions at the joints yields a system of equations which is equivalent to the traditional methods of multibody kinematic analysis. Eq. (12) plays a role similar to the GRÜBLER-KUTZBACH formula of spatial kinematics.

The conditions of linear coupling derived for elementary joints can be extended to other joints as well, but these joints must fulfill two conditions: (1) it must be ensured that the result of two joint transformations applied in sequence is again a transformation which can be described by the same joint, and (2) the composition of two transformations must be commutative. Property (1) implies that the transformations associated with the joint must be subgroups of rigid-body motion, while property (2) states that these subgroups must be Abelian. Table 1 shows joints of different degrees of freedom $f_{\mathcal{G}}$ possessing these properties. Besides the elementary joints, these are the cylindrical (C), planar translational (2P) and spatial translational (3P) joints.

$f_{\mathcal{G}}$	1	2	3	4–6
type	R, P, H	C, 2 P	3 P	none

Table 1: Joints yielding linear coupling conditions

5 Selection of an Appropriate Solution Flow

After having established the independent loops and their couplings, for the remaining problem of determining an appropriate ordering of the global equations one can imagine the individual loops as kinematical transformers which are connected by summing junctions representing the linear coupling equations, and the task is now to find orientations for the edges connecting the loops such that a block diagram results having the following properties (Hiller and Kecskeméthy 1989):

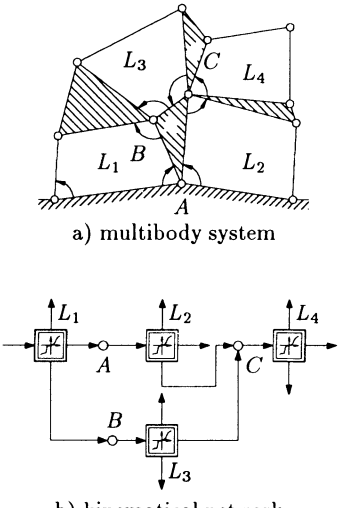
1. The number of external inputs is equal to the number of degrees of freedom of the system.
2. The number of inputs for each multibody loop L_i is equal to the local degree of freedom f_{L_i} of the loop.
3. Each summing junction has exactly one output.
4. There are no closed circuits.
5. The local kinematics of the transformers are recursively solvable.

It is obvious that the complete system of equations is then recursively solvable.

Surprisingly, for many technical applications conditions (1) through (5) can indeed be fulfilled. These systems are termed *recursively solvable systems*. Systems for which not all conditions can be fulfilled are called *non-recursively solvable systems*. The most common reason for the appearance of a non-recursively solvable system is that conditions (1) through (4) can not be accomplished. The cases for which condition (5) is violated (e.g. for the general case of a 7R-mechanism) do not occur so often in practice and shall not be regarded here.

A very simple method for finding the appropriate orientation of the edges for the case of *recursively solvalbe* systems is to start at the *sinks* of the system, i. e. where all edges connected to an element can be oriented *into* the element. Then, after finding such an element and orienting the edges, one removes both the element and the oriented edges and looks for the next sink, and so on. Clearly, the number of allowed input edges for a kinematical transformer is equal to its internal degree of freedom, while the number of allowed input edges for a summing junction equals the number of connections minus one. Also, there will exist branching nodes which have exactly one input. It may happen that after carrying out this algorithm some summing junctions or branching nodes remain which do not have enough inputs: this problem is fixed easily by reversing the direction of an appropriate number of edges pointing out of the element, re-orienting edges only once in order to avoid dead-locks. The resulting oriented block-diagram is denominated the "solution flow".

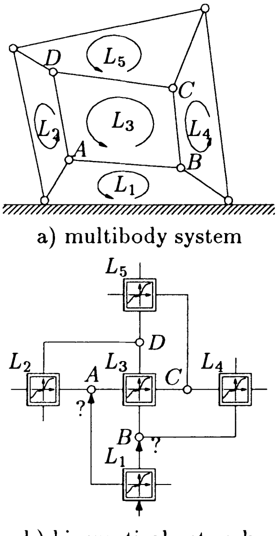
As an example of a recursively solvable system, a planar mechanism consisting of four interconnected planar four-bar loops is considered (Fig. 6). The redundant set of relative coordinates includes for each loop four variables. Three of these can be solved as functions of the fourth in closed form, yielding corresponding kinematical transformers. There are three linear assembly equations occuring in the joints A, B, C. A sequence of elements for which unoriented edges can be oriented as described above is: L_4, C, L_3, L_2, B, A, L_1. This sequence yields a "solution flow" which obviously is recursive. Thus the constraint equations of this system are solvable in *closed form.*



a) multibody system

b) kinematical network

Figure 6: Recursively solvable system



a) multibody system

b) kinematical network

Figure 7: Non-recursively solvable system

An example of a non-recursively solvable system is shown in Fig. 7. The planar mechanism consists of five independent multibody loops which are again four-bar mechanisms. There are four linear assembly equations at the joints A, B, C, D. From the corresponding block diagram it is clear that the algorithm described above can not start, because there is no element which has an allowable number of inputs greater than the number of connections. This situation changes when the summing junction D is removed and an additional input $\tilde{q}$ is provided. In this case the system is recursively solvable. The original system results after re-inserting the summing junction D, yielding an implicit equation for the determination of the function $\tilde{q}(q)$.

6 Example

The theory described above was implemented using the symbolic-computation language Mathematica (Wolfram 1991). As a consequence, it is possible to generate symbolically the kinematical transmission equations for recursively solvable multiple-loop spatial mechanisms in an automatic fashion. This was applied to the model of a mixing unit of the swash plate of a helicopter (Fig. 8). Fig. 9 shows the corresponding kinematical skeleton, from which it is seen that the system consists of $n_{\mathcal{B}} = 14$ moving bodies, numbered $\mathcal{B}_1, \ldots, \mathcal{B}_{14}$ and $n_G = 20$ joints, numbered $\mathcal{G}_1, \ldots, \mathcal{G}_{20}$ of which 6 are revolute (R), 6 are spherical (S), 7 are cardanic (T), and 1 is prismatic (P). Thus, $n_L = n_G - n_{\mathcal{B}} = 6$ independent loops are expected. Note that there is one multiple joint consisting of joints $\mathcal{G}_{15}$ and $\mathcal{G}_{20}$. The overall degree of freedom is $f = 3$.

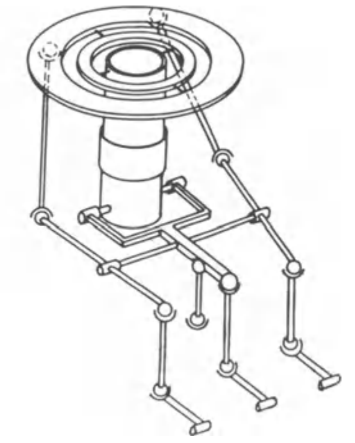


Figure 8: Swash-plate of a helicopter

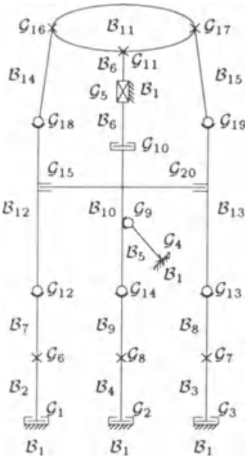


Figure 9: Kinematical skeleton

The first step is to apply the algorithm for finding the smallest loops to the system. After selecting body $\mathcal{B}_0$ as the root of the graph, the following cycle basis results (loops are indicated by lists of indices of bodies):

$$L_1 = \{11,15,13,10,6\} \quad , \quad L_2 = \{11,14,12,10,6\} \quad , \quad L_3 = \{1,3,8,13,10,6\}$$

$$L_4 = \{1,2,7,12,10,6\} \quad , \quad L_5 = \{1,5,10,6\} \quad , \quad L_6 = \{1,4,9,10,6\}$$

These loops are rendered in Fig. 10. Note that all loops contain bodies $\mathcal{B}_{10}$ and $\mathcal{B}_6$. This is because the revolute joint between them is weighted by just 1, while the cardanic and spherical joints are weighted by 2 and 3, respectively, so all loops try to pass through this joint. Note also that the loops have the local degree of freedom $f_{L_1} = f_{L_2} = f_{L_3} = f_{L_4} = 3$, $f_{L_5} = 1$ and $f_{L_6} = 2$.

After defining the loops, the coupling conditions can be determined quite easily: the multiple joint $\mathcal{G}_{15}/\mathcal{G}_{20}$ connects 3 bodies and 4 loops, so $p(\mathcal{G}_{15/20}) = 2$, joint

$\mathcal{G}_{10}$ connects 2 bodies and 6 loops, so $p(\mathcal{G}_{10}) = 5$, joint $\mathcal{G}_5$ connects 2 bodies and 4 loops, so $p(\mathcal{G}_5) = 3$, and joint $\mathcal{G}_{11}$ connects 2 bodies and 2 loops by two degrees of freedom, so $p(\mathcal{G}_5) = 2 \times 1 = 2$. All other joints produce no coupling between the loops. Thus the resulting degree of freedom of the system is indeed $f = 4 \times 3 + 1 + 2 - (2 + 5 + 3 + 2) = 3$. The result of coupling the loops is shown in Fig. 11. The edges are oriented according to the optimal sequence of resolution, which corresponds in this case to a closed-form solution.

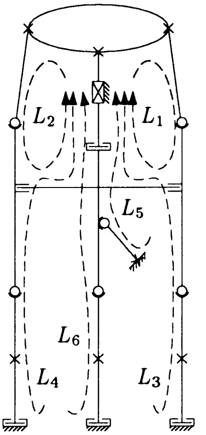


Figure 10: Independent Loops

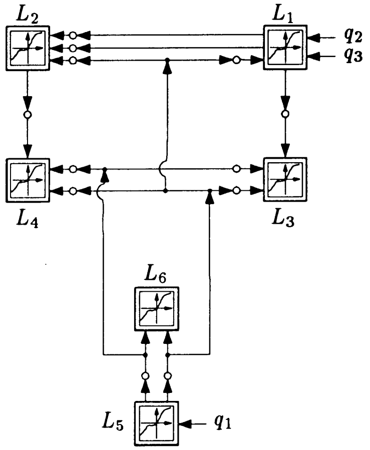


Figure 11: Block-Diagram

7 Conclusions

The method described in this paper is suited for obtaining symbolical expressions for the kinematics of multiple-loop spatial mechanisms for which closed-form solutions exist. The method is based on geometric and topological properties of the system, thus being independent of coordinate representations. Its main advantage is that the well-known algorithms for the solution of single-loop mechanisms can be fully integrated into the general procedure for treating multiple-loop systems. By the proposed algorithm for finding a minimal cycle basis, the loops to be processed as well as the number of coupling conditions between the loops can be kept to a minimum. This results in compact symbolical solutions. The method can be also used for detecting mechanisms having closed-form solutions, as well as the number of degrees of freedom in spatial mechanisms. This has been verified by an implementation based on the symbolical-computation language *Mathematica*. Also, the method is being currently extended to cope with the velocity and the acceleration problems.

References

Andrews, G. C. and Kesavan, H. K. (1975). The vector network model: A new approach to vector dynamics. *Mechanism and Machine Theory*, 10:509–519.

Fanghella, P. (1988). Kinematics of spatial linkages by group algebra: A structure-based approach. *Mechanism and Machine Theory*, 23(3):171–183.

Halmos, P. R. (1987). *Finite-Dimensional Vector Spaces.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, Heidelberg, Berlin.

Heiß, H. (1987). Theorie und Anwendung der Koordinatentransformation bei Roboterkinematiken. *Informatik Forschung und Entwicklung*, 2:19–33.

Hiller, M. and Anantharaman, M. (1989). Systematische Strukturierung der Bindungsgleichungen mehrschleifiger Mechanismen. *Zeitschrift für angewandte Mathematik und Mechanik*, 69:T303–T305.

Hiller, M. and Kecskeméthy, A. (1989). Equations of motion of complex multibody systems using kinematical differentials. *Transactions of the Canadian Society of Mechanical Engineers*, 13(4):113–121.

Hiller, M., Kecskeméthy, A., and Woernle, C. (1986). A loop-based kinematical analysis of spatial mechanisms. ASME-Paper 86-DET-184, New York.

Horton, J. (1987). A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal of Computing*, 16(2):358–366.

Hubicka, E. and Sysło, M. M. (1975). Minimal bases of cycles of a graph. In *Proc. 2nd Czechoslovak Symposium on Graph Theory*, pages 283–293, Prag. Academia.

Hunt, K. (1986). The particular or the general? (Some examples from robot kinematics). *Mechanism and Machine Theory*, 21(6):481–487.

Kecskeméthy, A. (1993). *Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen.* PhD thesis, Universität - GH - Duisburg.

Kecskeméthy, A. and Hiller, M. (1992). Automatic closed-form kinematics-solutions for recursive single-loop chains. In *Flexible Mechanisms, Dynamics, and Analysis, Proc. of the 22nd Biennal ASME-Mechanisms Conference, Scottsdale (USA)*, pages 387–393.

Lloyd, J. and Hayward, V. (1988). Kinematics of common industrial robots. *Robotics*, 4:169–191.

Mehner, F. (1990). Automatische Generierung von Rücktransformationen für nichtredundante Roboter. *Robotersysteme*, 6.

Olver, P. J. (1986). *Applications of Lie Groups to Differential Equations.* Graduate Texts in Mathematics 107. Springer-Verlag.

Paul, R. P. and Zhang, H. (1986). Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation. *The International Journal of Robotics Research*, 5(2):32–44.

Uicker, J., Denavit, J., and Hartenberg, R. (1964). An iterative method for the displacement analysis of spatial mechanisms. *Transactions of the ASME, Journal of Applied Mechanics*, pages 309–314.

Woernle, C. (1988). *Ein systematisches Verfahren zur Aufstellung der geometrischen Schließbedingungen in kinematischen Schleifen mit Anwendung bei der Rückwärtstransformation für Industrieroboter.* Fortschrittberichte VDI Reihe 18 Nr. 59. VDI Verlag, Düsseldorf.

Wolfram, S. (1991). *Mathematica. A System for Doing Mathematics by Computer.* Addison-Wesley Publishing Company, second edition.

A MODULAR METHOD FOR COMPUTATIONAL KINEMATICS

Pietro Fanghella and Carlo Galletti
University of Genoa, Istituto di Meccanica Applicata alle Macchine
Via Opera Pia 15 A - 16145 Genoa, ITALY

Abstract

A modular method for symbolic kinematic modelling of multiloop mechanisms is outlined. For a given mechanism, the method identifies automatically a list of modules for which closure equations can be generated and solved hierarchically. Closed-form solutions can be obtained in many cases of practical interest.

1 Introduction

The problem of kinematic analysis of a rigid-body mechanism, for which the body positions are determined by a set of known holonomic time-independent constraints and driving constraints, involves:
1) choosing a set of variables from which the positions of the mechanism bodies can be easily derived;
2) writing a set of closure equations that are equivalent to the mechanism constraints;
3) finding and executing algorithms for solving the closure equations.

Many choices can be made in mechanism analysis; therefore, for a given mechanism, various sets of closure equations (different in number and type) are obtained according to these choices. The solutions of such sets of equations can be more or less cumbersome and computationally efficient, depending on the equation structures.

There exist two opposite ways of performing kinematic modelling and solving rigid-body mechanisms: general global modelling and modular modelling.

A general global method (for instance, Haug, 1989) implements the following strategy:
1) for each body, a complete set of generalized coordinates is chosen;
2) equations with fixed, predefined structures are written for each constraint (pair, driver or composite constraint) in the mechanism;
3) numerical methods (e.g., Newton-Raphson) are utilized for iterative solution of the equations, using trial values of the generalized coordinates (in this phase, the independence of the equations is also checked by first order approaches).

No check is performed to apply specific modelling-and-solution techniques to different classes of mechanisms; the only exception consists in allowing the user to distinguish between planar and spatial mechanisms. However, for mechanisms made up of planar and spatial subassemblies, a fully spatial modelling is adopted.

A modular method (extensive Refs. in Galletti, 1986; Fanghella and Galletti, 1990) uses a different strategy:
0) subsets of bodies with special structural properties (modules) are recognized;
1) global or relative coordinates are defined for the bodies of each subset;

J. Angeles et al. (eds.), Computational Kinematics, 275–284.

2) the closure equations of each module are stated;
3) closed-form or numerical solutions are obtained for each module and propagated to different modules.

The most important characteristics of a modular method is that the closure equations can be written and solved separately for each module, in a hierarchical order defined by the mechanism structure.

A modular method is a special case of the well known analysis methods based on independent loops. In fact, a proper choice of independent loops leads naturally to mechanism modularization and equation uncoupling. Vice versa, equations obtained by a generic choice of independent loops are usually coupled. Such equations can be reduced, through a symbolic manipulation, to those obtained directly by a modular method, but, in general, this simplification is difficult, even when one uses a symbolic mathematical package.

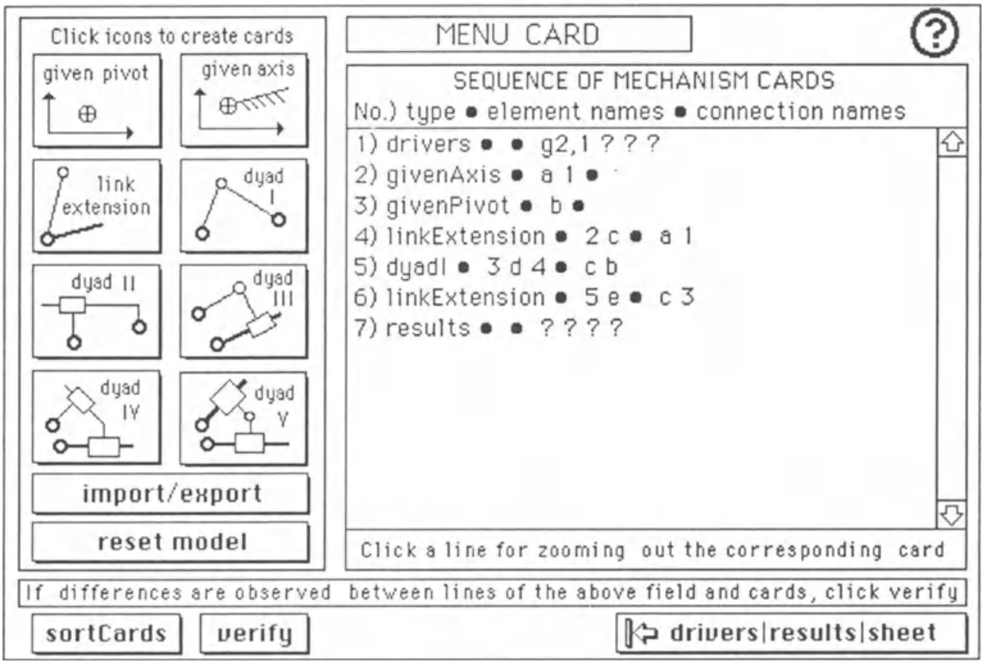
In the following, we show that a computer-aided implementation of step 0) of a modular method requires a heavy processing of the information describing the mechanism structure (i.e., topology, geometry, and driver location). Structural properties obtained by this step can be successfully used also to build closure equations with "particular" structures that simplify their solutions (for a discussion of "particular" and "general" methods, see Hunt, 1986). Moreover, as a special case of loop-based methods, a modular method benefits from using relative coordinates to define the body positions. In this way, it is possible to develop an optimal approach to the closure of each module in order to obtain a minimum set of independent decoupled equations. As a result, in many interesting cases, closed-form solutions can be obtained.

A rigorous modular method can be derived in a systematic way. By contrast, limited heuristic approaches to find subsets of closure equations in echelon form are available, which lead to such concepts as "weakly or strongly coupled" loops.

Computer-aided implementations of both modular and non-modular methods are feasible. The non-modular ones are the most widely used (Schielen, 1990). In general, they yield numerical solutions to multibody kinematics; therefore, they require that the user provide an initial, well-approximated and constraint-compatible estimate of the position of each body. This is a difficult task for the user; so, in numerical solutions, convergence problems may arise from poor initial estimates.

Computer-aided implementations of modular methods for planar linkages were first presented in the international literature in the early '70s (Bona et al., 1973, Brat and Lederer, 1973), and were further developed in the following decades (Rossi et al., 1981); various references can be found in Galletti, 1986. Program designers stated that computer programs require, as input, a description of mechanism modules. Therefore, the program user must find off line, by himself, the modules that make up the mechanism he wants to analyze; he must include this information in the program. This operation has proved somewhat easy for mechanism designers, who generally agree this way to input a mechanism description; moreover, as shown by Giannotti and Galletti (1993), very simple interfaces and learning supports can be provided to users (Fig. 1). This procedure is made possible by the small number of modules useful in planar mechanism analysis and by the fact that designers often perform mechanism synthesis by combining elementary modules. Figure 1 shows an icon-based interface where a small set of modules, corresponding to the 5 arrangements of 2 links and 3 pairs (dyads), is presented, together with few more basic elements (e.g., pivots fixed to the frame).

More difficult problems are involved in spatial mechanisms: open and closed loops can coexist in a mechanism; many different modules must be used even in basic industrial applications; modules with different motion characteristics (families) can be simultaneously present in a mechanism; and, in general, it is difficult for users to identify the modules in multiloop chains, like those used in robots.



SEQUENCE OF MECHANISM CARDS

No.) type • element names • connection names

1) drivers • • g2,1 ? ? ?
2) givenAxis • a 1 •
3) givenPivot • b •
4) linkExtension • 2 c • a 1
5) dyadI • 3 d 4 • c b
6) linkExtension • 5 e • c 3
7) results • • ? ? ? ?

Click a line for zooming out the corresponding card

If differences are observed between lines of the above field and cards, click verify

Figure 1 An interactive interface for modular modelling of planar mechanisms

Nevertheless, it is possible to implement a modular approach that, in many cases of practical interest, yields symbolic closed-form closure equations. Such an approach requires only data on the mechanism structure considered: the shapes of the links, the types and connections of the pairs, and the driver locations. No global position information and no numerical preassembling are needed to model a mechanism. Starting from structural information, one can generate a symbolic kinematic model, independently of actual driver values, which are instead necessary to perform a complete kinematic analysis.

Closed-form, automated solutions to the inverse problem of serial manipulators have previously been proposed by some researchers (Andrez et al., 1985; Herrera-Bendezu et al., 1988; Halperin, 1991; Rieseler et al., 1991). Our work addresses the more complex case of multiloop chains, which are important to model actuation chains and transmissions in robot mechanisms, and also multiple cooperating arms. The proposed approach aims to develop a software precompiler that transforms the description of a robot into an explicit, closed-form solution of the closure equations.

2 Towards an automatic modular method for spatial mechanisms

A module is a set of constrained rigid bodies that verifies the following assumptions:

- the relative body positions are determined only by the main body dimensions and by constraints among the bodies (both fixed and driving constraints are considered here);
- if one or more bodies are taken from the given set, no modules exist among the remaining bodies.

Therefore, a module is a kinematically determined chain that does not contain any simpler determined kinematic chain. Clearly, this definition is an extension of the concepts of Assur's and Input groups. In this work, only single-loop modules (i.e., single-loop de-

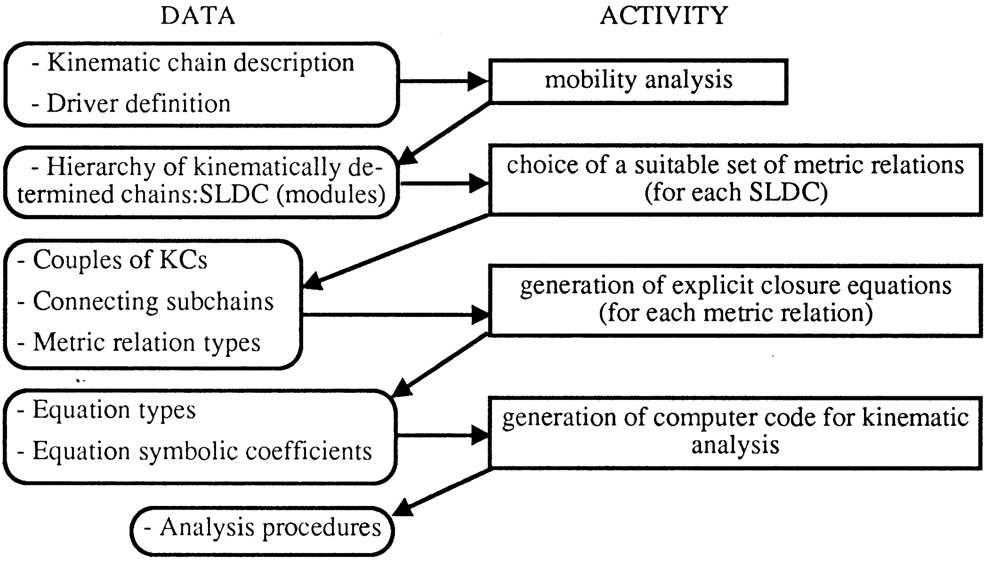

DATA
ACTIVITY
- Kinematic chain description
- Driver definition
mobility analysis
- Hierarchy of kinematically determined chains:SLDC (modules)
choice of a suitable set of metric relations (for each SLDC)
- Couples of KCs
- Connecting subchains
- Metric relation types
generation of explicit closure equations (for each metric relation)
- Equation types
- Equation symbolic coefficients
generation of computer code for kinematic analysis
- Analysis procedures

Figure 2 Scheme of computer-aided modular modelling of mechanisms

termined chains, SLDCs) are considered; then, the use of a modular method yields a set of independent kinematic loops that are uncoupled and hierarchically organized. Not all interesting spatial mechanisms can be modelled in this way, nor can all combinations of drivers for a given mechanism be considered. However, a notable amount of interesting spatial mechanisms (e.g., multiloop industrial robots) can be solved in both direct and inverse kinematic problems. The authors are studying extensions to more general cases.

Figure 2 shows the scheme of a process for modelling mechanisms by recognition of their modules. A first step (Section 4), based on mobility analysis, recognizes modules (SLDCs) one at a time, and allocates them according to computable hierarchy. Note that modules are identified sequentially, one after the other, so the process of finding modules is developed in time. A second step (Section 5) states what metric relations are to be used to form a set of independent closure equations for each SLDC. Then, explicit, symbolic closure equations can be built by inserting pair and driver transformations in the metric relations. The symbolic equations for each SLDC can be stated separately and are described by their structures, by the symbolic expressions for the equation coefficients, and by explicit solution algorithms. They are solved in sequential order dependent on the hierarchical organization of the SLDCs.

3 Definitions

In order to implement the above steps, in a systematic way, we have developed original concepts, algorithms, and formal definitions of useful entities. Displacement group theory, as proposed by Herve' (1978), is used both to recognize SLDCs and to obtain closure equations for each of them. Formal definitions are given in a paper by Fanghella and Galletti (1992): simplified definitions are reported here for the reader's convenience.

Displacement group: it is a subgroup of the Euclidean group of rigid transformations.

Set of Invariant Properties, IP: it is the set of geometric properties that do not change under all transformations of a group; IP strictly depends on the form of the transformations of a group. IP can be uniquely defined by means of a small number of representative geometric elements (points and directions).

A spherical pair corresponds to a displacement group with its center point as IP.

Kinematic Constraint, KC: this entity models a holonomic constraint between any two bodies i and j (KC_{ij}) in a mechanism, and is defined by the following information:

- the group containing all possible relative displacements between i and j;
- the set of actual invariant properties (IP) of the group, given in reference systems fixed to the links i and j;
- the connectivity, i.e., the number of degrees of freedom of the relative displacement.

Kinematic constraints can be related by composition and intersection operations. Fanghella and Galletti (1991) give a complete implementation of KC operations for single-loop chains: tables, rules and algorithms provide the tools for automatic computation.

Set of Determined Bodies, DB: two possibilities exist:

- a set of links connected by drivers;
- the links of an SLDC; moreover, if a link of the SLDC belongs to another DB, also the links of such a DB are included.

The KCs between any couple of links of a DB are obtained by explicit algorithms. It is possible to show that the connectivities of all KCs in a DB are equal to zero.

DBs are built sequentially, one at a time, during module recognition; at a generic stage, the existing DBs contain all recognized modules. In other words, a DB contains one or more kinematically determined chains.

Single-Loop Generalized Chain, SLGC: given a set of n serial open chains, the i-th chain starts from body s_i and ends with body e_i; bodies e_i and s_{i+1}, i=1 . . . n, belong to the same DB. The one-loop closed chain $s_1 \ldots e_1 \, s_2 \ldots e_2 \ldots s_n \ldots e_n \, s_1$ is an SLGC. It represents one or more linear sequences of bodies connected to one or more sets of modules already recognized.

Single-Loop Determined Chain, SLDC: it is an SLGC with zero mobility, i.e., a module.

Figure 3 shows an example of a single-loop generalized chain (SLGC): here two linear sequences of bodies (one starting from pair A and ending at pair E, and the other starting from pair B and ending at pair F) are connected by three different DBs in order to form a closed loop. A mobility analysis of such a closed chain can be performed by considering the actual pairs of the chain (A, B, C...E, F, G) and the kinematic constraints, KC_{uv} and KC_{st}, between the DB links. According to definitions, the connectivities of KC_{uv} and KC_{st} are equal to zero. If the mobility of the SLGC is zero, the following properties hold:

- the SLGC is a single-loop determined chain (SLDC);

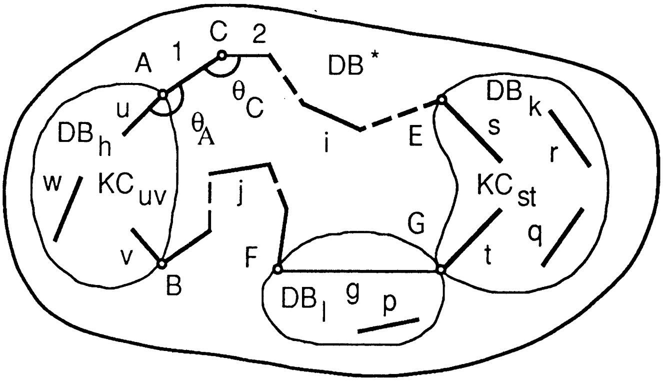


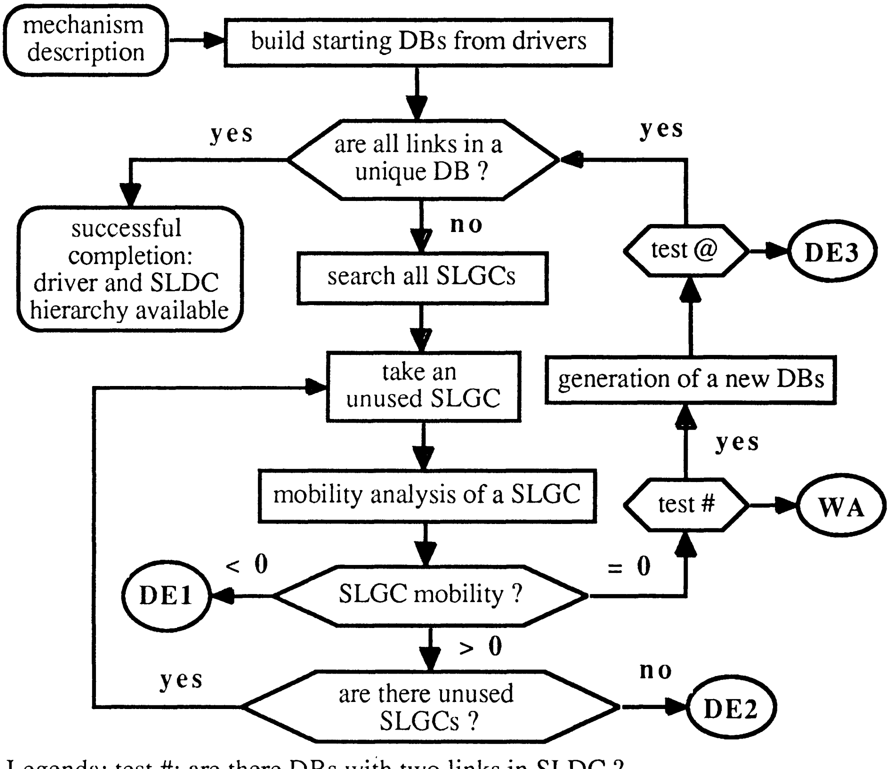
Figure 3 A single loop generalized chain (SLGC) and various sets of determined bodies (DB)

- the KCs between all couples of links of the SLDC can be computed; they depend only on the links and pairs of the SLDC and on the KCs of the DBs;
- the relative positions of all links and the values of the pair variables of the SLDC are determined by the relative positions of the links of each DB of the SLDC;
- the mobility of any DB and the KCs between its links are not affected by the SLDC;
- the set of all links of the SLDC and of all DBs of the SLDC defines a new DB*.

Therefore, if it is possible to find a set of SLDCs that contains all links and pairs of a mechanism, this is a complete set of independent kinematic loops. Besides, since, by definition, each SLDC can be analyzed separately, following a hierarchical order, the sets of closure equations for different loops can be solved autonomously. In this way, it is possible to obtain a systematic approach to generate a symbolic modular model of a multiloop mechanism.

4 Flow diagram and algorithms for SLDC recognition

The flow diagram of the process of finding modules is shown in Fig. 4.



Legenda: test #: are there DBs with two links in SLDC ?
test @: has the new DB more than one boundary link ?
WA: warning: links of this SLDC cannot have relative displacements.
DE1: data error: overconstrained robot mechanism.
DE2: data error: the mechanism cannot be modelled by SLDCs.
DE3: data error: no. of driver variables less than no. of mechanism d. o. f.

Figure 4 Flow diagram of the process of finding modules

The most important actions are:

a) Building initial DBs from driving constraints. This action makes it possible to initialize the search for determined loops. Since the relative positions of all bodies connected by driving constraints are determined, each set of such bodies is used to build a DB. Algorithms are presented in Fanghella and Galletti (1992).

b) Searching for all SLGCs. This action is performed whenever a new determined chain must be singled out from a set of DBs connected by various link chains. For such a set of DBs and for the chains connecting them, no explicit information is available to establish if any SLDC exists. Therefore, a search approach is used: all SLGCs are found and analyzed (action c) in order of increasing number of links. SLGCs with open-loop connectivities ≥ 6 are always mobile; hence they are not considered, except for some special cases involving spherical and planar KCs, which yield one degree of passive internal mobility.

c) Performing a mobility analysis of an SLGC. Mobility analysis of a single-loop chain establishes if the chain is overconstrained, has zero mobility, or is mobile, and provides the KCs between all couples of links. Explicit algorithms are given in Fanghella and Galletti (1991). If the chain has zero-mobility, it is an SLDC.

If, after iterating actions b) and c), no SLDC is found, the considered mechanism cannot be modelled by a hierarchy of SLDCs.

d) Creating a new DB by merging an SLDC and all DBs in the SLDC. This action allows one to obtain a new set of DBs, for which the search process can be repeated.

The flow diagram of Fig. 4 has been implemented by a software package for the recognition of independent single loops.

5 Closure equations for an SLDC

Various papers present approaches similar to that discussed in this section: for instance, that proposed by Litvin (1975) and by Hiller and Woerlne (1987).

If we consider a generic SLDC (Fig. 5), the system of its closure equations can be expressed as:

$$\mathbf{f}\,(\,T_{uv}, T_{st}, \ldots ; T_1, T_2, \ldots ; \theta_A, \theta_B, \ldots\,) = \mathbf{0} \qquad (1)$$

where:

- T_{uv}, T_{st}, . . are the transformations between the links of the DBs of the SLDC;
- T_1, T_2, . . are the constant transformations defining the geometries of the bodies of the SLDC;
- θ_A, θ_B, . . are the unknown pair variables of the SLDC.

The structure of equations (1) can be obtained in the following way.

For a single-loop kinematic chain, any two kinematic constraints, λ and μ, can be considered. It is possible to write a set of metric relations, **MR**, that relate the invariant properties of λ and μ (e.g., distance between points, angle between axes, etc.) and to express them through paths π and δ. Since such relations are reference-independent scalar functions of the pair variables and linkage dimensions in the two paths, they can be equated, thus obtaining the independent closure equations

$$\mathbf{MR}(\pi) = \mathbf{MR}(\delta) \qquad (2)$$

which are free from the pair variables of the kinematic constraints λ and μ. Fanghella and Galletti (1989) provide tables for metric relations and give the general rules for selecting constraints in order to obtain a number of independent closures (2) equal to the number of pair variables appearing in the constraints, so these unknowns are computable.

It is interesting to note that the number, the form, the geometric meaning, and the use of metric relations depends on the displacement group of the SLDC.

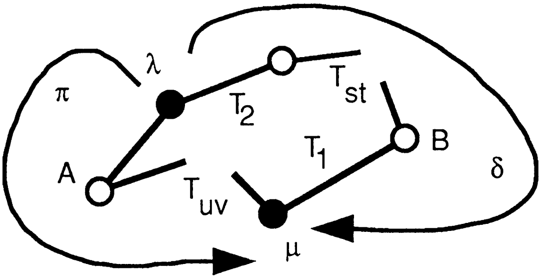


Figure 5 Paths for metric relations in a single-loop kinematic chain

For instance, let us consider an SLDC containing two revolutes, a and b, which are assumed to be constraints λ and μ. Figure 6 presents the geometric relations between the IPs of two revolutes, using points A_a and A_b, and unit vectors v_a and v_b.

Column 1 gives the group of the SLDC. The relations are given in two columns:

- column 2 lists the metric relations that can be used to obtain closure equations (2); such equations contain, as unknowns, the variables of the pairs in paths π and δ;
- column 3 gives the offset relations that provide equations similar to (2) but independent of all the pair variables in the SLDC. These equations must be satisfied by the dimensions of the bodies of the SLDC in order to allow mechanism assembly. If the chain is spatial and the revolute axes are skew, four metric relations exist. In any other case, a smaller number of metric relations exists. For instance, only one relation exists between the revolutes of a planar chain, and can be used to set the system (2). Moreover, one offset condition states that the sum of displacements normal to the motion plane must be zero. Note that this information is useful to verify the correctness of a given mechanism.

For a given SLDC, many combinations of λ and μ exist: for instance, the kinematic pairs of the chain and any composition of them can be used. Among all these alternatives, the one giving the minimum number of closure equations and as many unknowns is the most useful one. Fanghella (1993) discusses this topic in more detail, and defines the cases where all closure equations are written in triangular form and can be solved one at a time.

chain group	metric relations	offset relations
D	$v_a \cdot v_b$ $(A_a-A_b) \cdot v_a \times v_b$ $(A_a-A_b) \cdot v_a$ $(A_a-A_b) \cdot v_b$	-
RP3	$(A_a-A_b) \cdot v_a$ $[(A_a-A_b) \times v_a]^2$	-
F	$[(A_a-A_b) \times v_a]^2$	$(A_a-A_b) \cdot v_a$
S (center in A_{ab})	$v_a \cdot v_b$	$(A_a-A_{ab}) \cdot v_a$ $(A_a-A_{ab}) \cdot v_b$
C	$(A_a-A_b) \cdot v_a$	-
R	-	$(A_a-A_b) \cdot v_a$

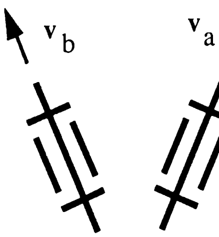


Figure 6 Geometric relations between two revolutes a and b

Finally, a symbolic solution using $T_{uv,}$ T_{st}, . . . as solution parameters can be obtained for the system (1). The actual numerical values of these parameters are computed whenever a kinematic analysis is required for given numerical values of the variables of the driving constraints.

6 Example

An example of a 6 degrees of freedom robot mechanism is shown in Fig. 7. This mechanism can be analyzed as a sequence of single-loop determined chains by the approach outlined in this paper, both for direct and inverse kinematic problems.

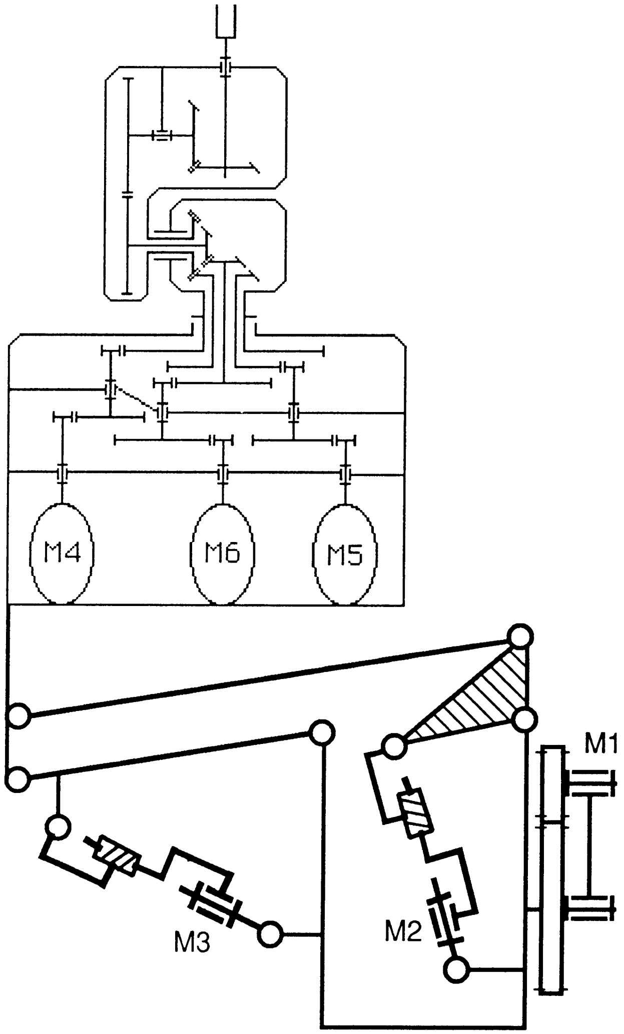


Figure 7 A complete mechanism for a 6 degrees of freedom robot (arm and elbow)

References

Andrez, E., Losco, L., Andre, P., and Taillard J.P., 1985, "Generation Automatique et Simplification des Equations Litterales des Systemes Mecaniques Articules", *Mech. Mach. Theory*, Vol. 20, pp. 199-208

Bona, C., Galletti, C., and Lucifredi, A., 1973 "Computer-Aided Automatic Design", *Mech. Mach. Theory*, Vol. 8, pp. 437-456

Brat,, V., and Leder, P., 1973, "KIDYAN: Computer-Aided Kinematic and Dynamic Analysis of Planar Mechanisms", *Mech. Mach. Theory*, Vol. 8, pp. 457-467

Fanghella, P., 1993, "Systematic Kinematics of Robot Arms", Technical Report IMAGE L5-93-PF, Univ. Genova, Genoa, I

Fanghella, P., and Galletti, C., 1989,. "Particular or General Methods in Robot Kinematics?", *Mech. Mach. Theory*, Vol. 24, pp. 383-394

Fanghella, P., and Galletti, C., 1990, "Kinematics of Robot Mechanisms With Closed Actuating Loops", *Int. Jl of Robotics Research*, Vol. 6, pp. 19-24

Fanghella, P., and Galletti, C., 1991,"Mobility Analysis of Single-loop Kinematic Chains: an Algorithmic Approach Based on Displacement Groups", Technical Report IMAGE L5-92-PFCG, Univ. Genova, Genoa, I

Fanghella, P., and Galletti, C., 1992, "Hierarchical Generation of Independent Loops for Symbolic Kinematics of Robot Mechanisms", *3rd Int. Workshop Advances in Robot Kinematics*, J. Lenarcic and V .Parenti-Castelli, ed., Felloni, Ferrara, I, pp. 96-103

Galletti, C., 1986, "A Note on Modular Approaches to Planar Linkage Kinematic Analysis", *Mech. Mach. Theory*, Vol. 21, pp. 385-391

Giannotti, E., Galletti, C., 1993, "A Hypertext Environment for Simulation of Multibody Mechanical Systems", *Proc. Hypermedia in Vaasa 93*, M. Linna and P. Ruotsala eds., Vaasa Inst. Technology, Vaasa, SF, pp. 96-102

Halperin, D., 1991, "Automatic Kinematic Modelling of Robot Manipulators and Symbolic Generation of their Inverse Kinematics Solutions", *Advances in Robot Kinematics,* S. Stifter and J. Lenarcic ed., Springer-Verlag, Berlin, FRG, pp. 310-317

Haug, E.J., 1989, *Computer Aided Kinematics and Dynamics of Mechanical Systems,* Allyn & Bacon, Boston, MA

Herrera-Bendezu, L., Mu, E., and Cain, J., 1988, "Symbolic Computation of Robot Manipulator Kinematics", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 993-998

Herve', J., 1978, "Analyse Structurelle des Mecanismes par Groupe des Deplacements", *Mech. Mach. Theory*, Vol .13, pp. 437-450

Hiller, M., and Woerlne, C., 1987, "A Systematic Approach for Solving the Inverse Kinematic Problem of Robot Manipulators", *Proc. 7th Conf. Th. Mach. Mech.*, Sevilla, E, pp. 215-221

Hunt, K. H., 1986, "The particular or the General? Some Examples from Robot Kinematics", *Mech. Mach. Theory*, Vol. 21, pp. 481- 487

Litvin, F.L., 1975. Simplification of the matrix method of linkage analysis by division into unclosed kinematic chains", *Mech. Mach. Theory*, Vol .10, pp. 315-326

Rieseler, H., Schrake, H., and Wahl, F, 1991, "Symbolic Computation of Closed Form Solutions with Prototype Equations", *Advances in Robot Kinematics,* S. Stifter and J. Lenarcic eds., Springer-Verlag, Berlin, FRG, pp. 343-351

Rossi, A., Fanghella, P., and Giannotti, E., 1981, "An Interactive Computer Package for Planar Mechanism Analysis", *Proc. II Int. Conf. Eng. Software*, R. Adey ed., CML, Southampton, UK, pp. 193-203

Schielen, W., 1990, *Multibody Systems Handbook*, Springer-Verlag, Berlin, FRG

Acknowledgement

This work has been partially funded by MURST 40%

SYNTHESIS FOR RIGID BODY GUIDANCE USING POLYNOMIALS

Rojas Salgado, Angel A.; Torres Navarro, J. Isidro
Facultad de Ingeniería. Universidad Nacional Autónoma de México
Cd. Universitaria, A.P. 70-256, 04510 México D.F. MEXICO.
FAX NR.(52)(5)548-0950, (52)(5)548-3044

ABSTRACT

A method for the exact synthesis for rigid-body guiding mechanisms through polynomials with dimensional constraints is presented. This method produces a broad solution spectrum that can be visually examined. The designer can choose a deffect-free mechanism within a restricted area. The search along perpendicular axes by means of polynomials yields a thorough set of solutions. An example is presented for illustration purposes.

INTRODUCTION

It is known that the maximum number of configurations that can be specified in the exact synthesis for rigid body guidance is five (Bottema and Roth, 1979). Each configuration consists of the location of a certain point, and the orientation of a straight line fixed to the body. The exact synthesis for five positions offers the designer a limited number of solutions, since from the synthesis equation, a unique fourth-degree polynomial is obtained, which yields a maximum of four dyads, which, when combined, give place to six mechanisms at most (Rojas, 1988).

When reducing a configuration, one is free to choose a parameter, such that it is possible to solve problems with certain restrictions, so that the mechanism be bifurcation-free (Angeles and Rojas, 1983), or so that the central points B be within a certain restricted zone.

The treatment discussed in this article allows one to obtain a set of solutions by means of a cubic polynomial. The polynomial is obtained in the central-point complementary coordinate, using one of the coordinates of the central-point. The coordinates of the circular-point depend on the complementary coordinate. With the values of the coordinates, the set of Burmester point pairs (BPP), which represent the dyads, is obtained. Hence, by specifying a value or a set of values for a

J. Angeles et al. (eds.), Computational Kinematics, 285–293.

coordinate of the central-point, solutions can be determined either along a particular line, or within a definite zone. When plotting the set of coordinates chosen along perpendicular axes, the Burmester curves that represent the central and circular-point loci are obtained (Sandor and Erdman, 1984). This makes possible a visualization of the solution curves without loss of accuracy.

SYNTHESIS EQUATIONS

The derivation of the synthesis equations is based upon the invariability of the lenghts of the planar mechanism (4R). Both the driving link AB and the driven link A*B* are shown in Fig. 1. Point A (or A*) is known as a circular point. It describes a circle when link AB (or A*B*) is turned. The Appendix includes the synthesis equations that are obtained from the constancy of length AB (or A*B*), which, when using the specified configurations, leads to a cubic polynomial depending on one coordinate of the central-point, b_x (equation A.7) or b_y(equation A.13). As is well known, the polynomial has either one or three real roots. In the latter case three solution sets are obtained for the coordinates of the circular point, eqs. (A.8) or (A.14).

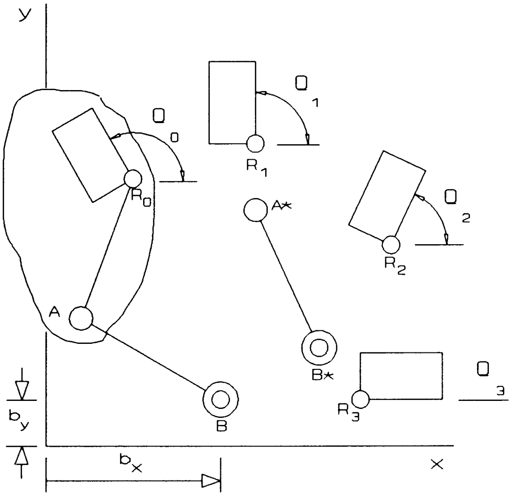


Fig. 1 Driving dyad

In each case, one has a solution set of the form

$$\{a_x,a_y,b_k\}_j = f(R_i,\ \theta_i,\ b_s)_j \tag{1}$$

where $i=0,...,3$ $k=x,y$ $s=y,x$

j depends on the number of real roots of the polynomial

θ_i angle that line rigid-body makes with the x-axis

$R_i(x_i,\ y_i)$ precision point

$A(a_x,\ a_y)$ circular point of the dyad

$B(b_x,\ b_y)$ central point of the dyad

Moreover, the solutions in terms of b_x or b_y happen to complement each other, and hence, it is important to search along the two coordinates axes.

SOLUTION FOR A RESTRICTED AREA

Let b_s denote either b_x or b_y. To begin with, b_s takes the boundary values in order to verify that there exists a solution within the restricted area. If such solution exists, b_s will take any value within the interval, thus yielding three solution curves at most. In Figs. 3 and 4, the solutions are presented within the design region, using b_x and b_y, which, when combined, form a better defined curve (Fig. 5). The latter figure represents Burmester's curves. By choosing some points (BPP), such as in Fig. 6, one has a representation of the links of the mechanism, which eases the visualization of the solution process. In principle, there can be built as many mechanisms as desired, so as to enable the designer to make the best choice, or to undertake more accurate calculations in a specified zone. Therefore, a more interactive design way for the synthesis is achieved, which is similar to those methods based on Burmester curves (Sandor and Erdman, 1984), but yielding a high accuracy.

The process can be summarized in the following stages:

1) The data (precision points) are given as
   $r_i(x_i,y_i)$ and θ_i for $i=0,..,3$ (2)
2) The calculations to obtain the polynomial (A.7 or A.13) are done. The boundary values must be verified first. If a solution exists, the method proceeds.
3) The roots x_1 and x_2 are calculated (A.8 or A.14).
4) The analysis is done, and the values which do not meet the restrictions are ignored.
5) Steps (2) through (4) are done again, this once with the other coordinate.
6) The values are plotted in order to obtain the curves which represent the solution.
7) The solution bars are drawn within the permitted zone.
8) Test bars are chosen, and their kinematic characteristics are analyzed (mechanical advantage, branching and order problems).

By means of this process, one can easily find out whether the solution is acceptable or not.

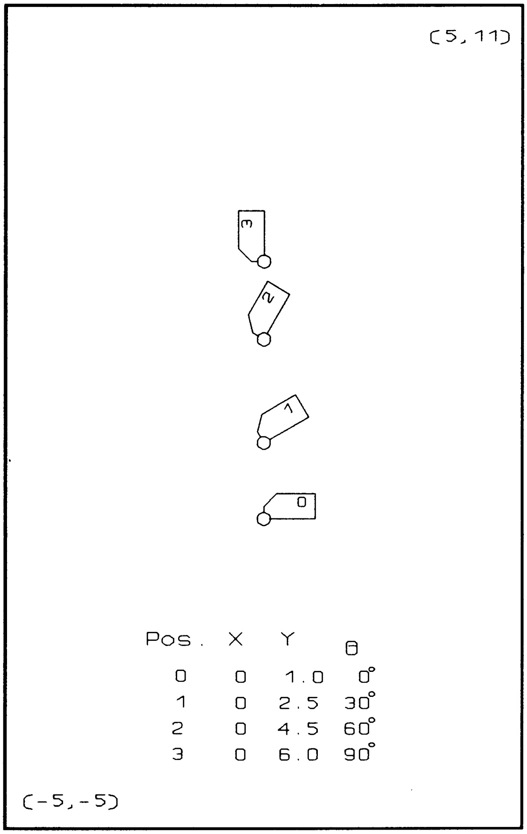
EXAMPLE

It is desired to guide a tool from a starting position (position 0) through other three given positions. Figure 2 contains the information that corresponds to the precision points, angles and limits imposed to the zone within which the solution mechanism must be found.

Using the data, the coordinates of the central point are calculated, giving b_x, thus obtaining b_y (fig. 3), then using b_y and obtaining b_x (fig. 4). The results are superimposed (fig. 5). The bars are drawn within the permitted zone (fig. 6). Afterwards, the representative bars are selected (fig. 7) and analyzed, in order to obtain the solution mechanism that yields an adequate kinematic performance.

Figure 8 shows the solution mechanism for the above example. As can be observed in figure 8, this solution is the result of the lines Nr. 2 and 7 in figure 7. The mechanism can perform the work cycle adequately.

In this application, MATLAB was used to do the calculations, and the drawings were made with AutoCAD.



Pos.	X	Y	θ
0	0	1.0	0°
1	0	2.5	30°
2	0	4.5	60°
3	0	6.0	90°

Fig 2 Positions for the RB and design zone

CONCLUSIONS

The method outlined here represents an effective way to solve the exact synthesis problem for rigid-body guidance, whose advantages are the computational accuracy and the graphical representation of the solutions. It is a valuable tool in kinematic design. This work can be used to develop software that helps to automatically analyze mechamisms, and that allows to identify certain desired characteristics.

ACKNOWLEDGEMENTS

This work, which is part of the IN301698 Project, was supported by the Dirección General de Asuntos del Personal Académico, and by the Department of Mechanical Engineering at UNAM.

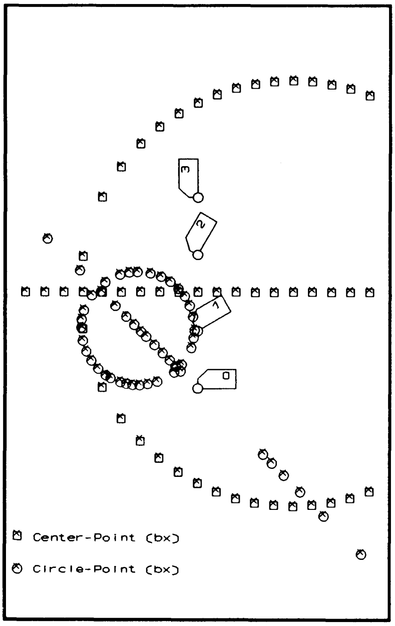


Fig 3 Central points for bx

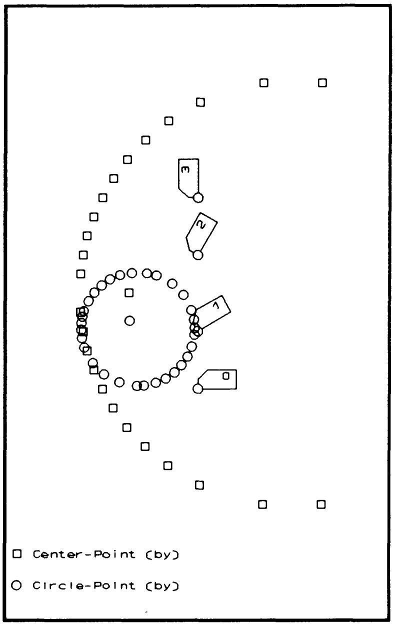


Fig 4 Central points for by

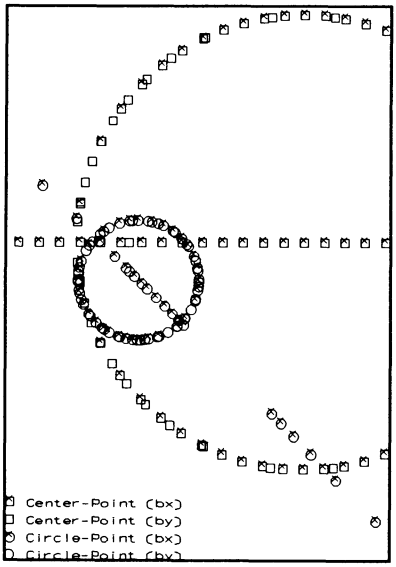


Fig 5 Solution points for bx and by

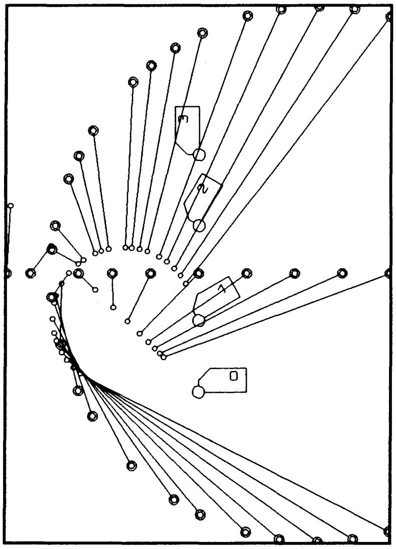


Fig 6 Generated bars

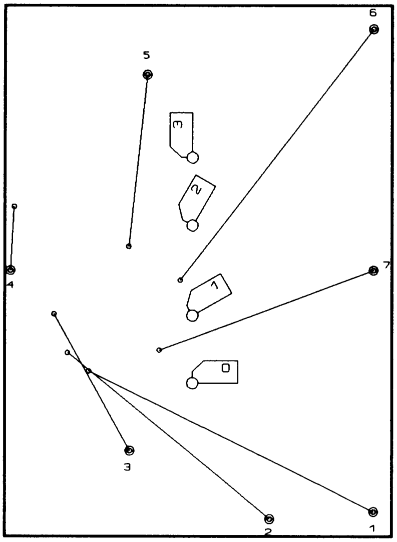
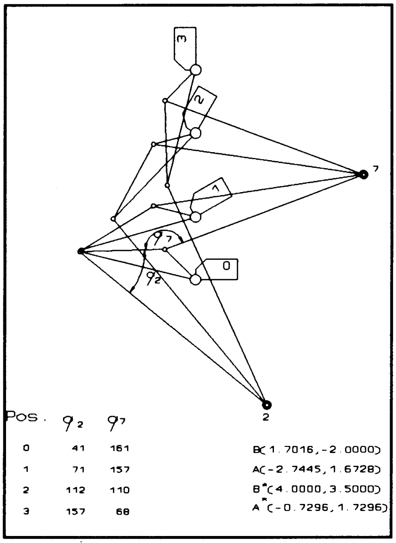


Fig 7 Selected links for assemble



Pos.	φ_2	φ_7
0	41	161
1	71	157
2	112	110
3	157	68

B(1.7016,-2.0000)
A(-2.7445,1.6728)
B*(4.0000,3.5000)
A*(-0.7296,1.7296)

Fig 8 Solution mechanism

REFERENCES

Angeles J., Rojas A., "An Optimisation Approach to the Branching Problem of Plane Linkage Synthesis", Proceedings of the VI World Congress on Theory of Machines and Mechanisms, New Delhi, India, 1983, pp. 120-123.

Bottema O., Roth B. Theoretical Kinematics. North Holland Publishing Co., Amsterdam, 1979, pp. 249-255.

Rojas A. A., "Síntesis exacta de conducción de cuerpo rígido mediante polinomios", Proceedings of the XIV Congress of Academia Nacional de Ingeniería, Guanajuato, México, 1988, pp 44-48.

Sandor G. N., Erdman A. G., Advanced Mechanism Design: Analysis and Synthesis. Vol 2, Prentice Hall, Inc., USA. 1984.

APPENDIX

The solution of the synthesis function for the 4R mechanism for a rigid- body guidance is based on the constancy of the lengths of the driving link (AB), and driven link (A*B*) (Angeles and Rojas, 1983). In figure A.1 the link AB and the rigid-body are shown. Both are referred to an origin O, for the reference position and for a new position i. Given that the point B is a central point (it works as a pivot

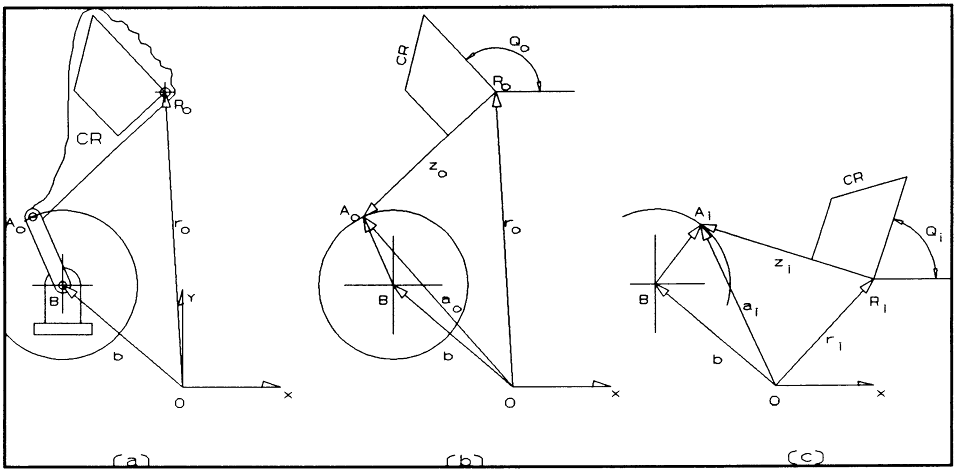


Fig A.1 Positions for the dyad and the RB.

to link AB). Vector **b** is invariant; additionaly, vectors BA and **z** are constants in magnitude. The rigid body has an angular displacement given by $\boldsymbol{\theta}_i$-$\boldsymbol{\theta}_0$.

From the above discussion, the synthesis function, for configuration i and for bs=bx as a parameter, becomes

$$f_j(x)= a_{j1}a_x+a_{j2}a_y+a_{j3}b_y+ g'_j\ a_xb_y+ h'_j\ a_yb_y+ k'_j = 0 \quad \text{for } j=1,,3$$

The necessary values to evaluate the coefficients of the solution polynomials, are the following:

A.1 Coordinates of the points: (x_i,y_i); for i=0,,3

A.2 Angular displacement: $\boldsymbol{\beta}_i= \boldsymbol{\theta}_i-\boldsymbol{\theta}_0$ for i=1,,3

A.3 Define vectors **g', h', k'**:

$$g'_j = -\sin\beta_j$$

$$h'_j = 1-\cos\beta_j$$

$$k'_j = 0.5(x^2_0+y^2_0+x^2_j+y^2_j)+ [(x_j-b_s)y_0 -x_0y_j]\sin\beta_j+$$

$$- [(x_j-b_s)x_0+ y_0y_j]\cos\beta_j- b_sx_j \quad \text{for } j=1,,3$$

A.4 Define matrix **A** via its entries a_{jk}, as

$a_{j1} = b_s\text{-}x_0+(x_j\text{-}b_s)\cos ß_j+ y_j\sin ß_j$

$a_{j2} =\text{-}y_0+ (b_s\text{-}x_j)\sin ß_j+ y_j\cos ß_j$

$a_{j3} = x_0\sin ß_j+ y_0\cos ß_j\text{-} y_j$ for j=1„3

A.5 Vectors **g,h,k** are calculated from equations

Ag = -g'
Ah = -h'
Ak = -k'

A.6 Resulting system:

$\mathbf{x} = \mathbf{g}\ a_x b_y + \mathbf{h}\ a_y b_y + \mathbf{k}$

where $\mathbf{x} = [\ a_x, a_y, b_y]^T = [x_1, x_2, x_3]^T$

A.7 The cubic polynomial is derived as

$\alpha_3 x_3^3+\alpha_2 x_3^2+\alpha_1 x_3+\alpha_0=0$

where: $\alpha_3 = \det(M_{33})$

$\alpha_2 =\text{-}(\det(\mathbf{C})+ g_1 + h_2)$

$\alpha_1 = 1 + \det(M_{22}) + \det(M_{11})$

$\alpha_0 =\text{-}k_3$
$\mathbf{C} = [\mathbf{g}, \mathbf{h}, \mathbf{k}]^T$

M_{ij}= ij minor of matrix **C**

g_i = i element of vector **g**; for i=1„3

A.8 Other values are:

$x_1= (h_1x_3\text{-}k_3h_1+k_1h_3)/[(g_3h_1\text{-}g_1h_3)x_3+h_3]$

$x_2= (g_2x_3\text{-}g_2k_3+g_3k_2)/[(h_3g_2\text{-}g_3h_2)x_3+g_3]$

The synthesis function for configuration i and for bs=by as parameter becomes

$f_j(x)= a_{j1}a_x+a_{j2}a_y+a_{j3}b_y+ g'_j\ a_yb_x+ h'_j\ a_xb_x+ k'_j = 0$ for j=1„3

A.9 Elements of the vectors **g', h', k'**, are:

$g'_j = \text{sinß}_j$

$h'_j = 1\text{-cosß}_j$

$k'_j = 0.5(x^2_0+y^2_0+x^2_j+y^2_j) - [(y_j-b_s)x_0 - y_0x_j]\text{sinß}_j+$

$- [(y_j-b_s)y_0+ x_0x_j]\text{cosß}_j - b_sy_j$ for j=1„3

A.10 Coefficients of the matrix **A**:

$a_{j1} = -x_0+(y_j-b_s)\text{sinß}_j+ x_j\text{cosß}_j$
$a_{j2} = -y_0+b_s+(y_j-b_s)\text{cosß}_j- x_j\text{senß}_j$
$a_{j3} = x_0\text{cosß}_j- y_0\text{sinß}_j- x_j$ for j=1„3

A.11 Vectors **g,h,k** are calculated as in A.5

A.12 Resulting system:

$\mathbf{x} = \mathbf{g}\ a_yb_x + \mathbf{h}\ a_xb_x + \mathbf{k}$

where $\mathbf{x} = [\ a_x, a_y, b_y]^T = [x_1, x_2, x_3]^T$

A.13 The cubic polynomial is obtained as

$$\alpha_3x_3^3+\alpha_2x_3^2+\alpha_1x_3+\alpha_0=0$$

where: $\alpha_3 = \det(M_{33})$

$\alpha_2 = -(\det(\mathbf{C})+ g_2 + h_1)$

$\alpha_1 = -[1 + \det(M_{12}) + \det(M_{21})]$

$\alpha_0 = k_3$
$\mathbf{C} = [\mathbf{g}, \mathbf{h}, \mathbf{k}]^T$
M_{ij}= minor ij of matrix **C**
g_i = element i of vector **g**; for i=1„3

A.14 Other values are:

$x_1= (g_1x_3+g_3k_1-g_1k_3)/[(h_3g_1-h_1g_3)x_3+g_3]$

$x_2= (h_2x_3-h_2k_3+h_3k_2)/[(g_3h_2-h_3g_2)x_3+h_3]$

Designing Mechanisms for Workspace Fit

F.C. Park, A.P. Murray, and J.M. McCarthy
Mechanical and Aerospace Engineering
University of California, Irvine
Irvine, CA 92717

Abstract

In this paper we examine the problem of designing a mechanism such that its tool frame comes closest to reaching a set of desired goal frames. We regard SE(3), the Euclidean group of rigid-body motions, as a Lie group, and demonstrate that a left-invariant Riemannian metric on SE(3) provides not only a solution that is invariant with respect to choice of inertial frame, but also a natural means of regarding SE(3) as a metric space. To illustrate our methodology this metric is applied to the design and positioning of certain planar and spherical mechanisms.

1 Introduction

In many mechanism synthesis problems the design requirements are often specified by a set of goal frames that indicate the desired positions and orientations of some workpiece. A mechanism is then sought whose tool frame can reach this set or, when this is not possible, comes closest to reaching these goal frames. Consider, for example, two cooperating manipulators holding a common workpiece; the dual objectives are to position the manipulator bases, and determine the kinematic parameters and workpiece grasp points such that the resulting closed chain comes closest to reaching all the goal frames.

Assuming that an inertial reference frame and length scale for physical space has been chosen, each goal frame can be assigned an element of the Euclidean group SE(3) (also known in the robotics literature as the *homogeneous transformations*, or rigid-body motions). If $\mathcal{M}$ denotes the joint-space manifold, the workspace fitting problem can now be characterized as selecting a forward kinematic map $f : \mathcal{M} \rightarrow \mathrm{SE}(3)$ such that the minimum distance from each goal frame to the workspace[1] of f is minimized. In short, workspace fitting can essentially be viewed as an approximation problem in $\mathcal{C}^0(\mathcal{M}, \mathrm{SE}(3))$, the space of continuous maps from $\mathcal{M}$ to SE(3).

[1] By workspace we mean the set of all positions *and* orientations achievable by the mechanism's tool frame – mathematically this is just the image of f.

J. Angeles et al. (eds.), Computational Kinematics, 295–306.

The first step towards obtaining a mathematical solution to this problem is to define a metric, in the sense of a distance measure, on SE(3). Clearly any number of arbitrary metrics can be defined, but in order for the metric to be physically meaningful it must be ***invariant with respect to the choice of inertial frame***; that is, the distance between any two frames should not depend on the location of the inertial frame of reference. This requirement can be expressed mathematically as follows: if X_1, X_2 are elements of SE(3) representing two frames, and $d(\cdot,\cdot)$ is the distance metric on SE(3), then $d(X_1, X_2) = d(TX_1, TX_2)$ for any $T \in$ SE(3). This must hold because under a change of the inertial frame by a transformation T, the frames represented by X_1 and X_2 are transformed according to $(X_1, X_2) \mapsto (TX_1, TX_2)$.

In this paper we obtain solutions to the class of mechanism design problems described above that are inertial frame-invariant, by applying a left-invariant *Riemannian metric* (to be distinguished from a distance metric) on SE(3). Specifically, we show how this Riemannian metric can be used to construct a natural distance metric on SE(3) with the desired features; the resulting metric provides a useful and mathematically rigorous tool for treating a wide range of mechanism design problems. Our goal, which is motivated in part by the work of Ravani (1982) and Bodduluri (1990), is to optimize the kinematic parameters so that the resulting workspace is a best fit to the set of goal frames.

The paper is organized as follows: In Section 2 we discuss the geometry of SE(3) as a Lie group, and derive an explicit formula for the distance metric on SE(3) that is induced from the left-invariant Riemannian metric. In Section 3 we illustrate our design methodology, by applying this distance metric to a class of planar and spherical mechanism design problems.

2 The Geometry of SE(3)

2.1 Canonical Coordinates

For our purposes it is sufficient to think of SE(3) as consisting of matrices of the form $\begin{bmatrix} \Theta & b \\ 0 & 1 \end{bmatrix}$, where $\Theta \in$ SO(3) and $b \in \Re^3$. SE(3) has the structure of both a differentiable manifold and an algebraic group, and is an example of a *Lie group.*[2] Elements of SE(3) will alternatively be denoted by the pair (Θ, b), with group multiplication understood to be $(\Theta_1, b_1) \cdot (\Theta_2, b_2) = (\Theta_1\Theta_2, \Theta_1 b_2 + b_1)$. SO(3) and $\Re^3$ are also Lie groups under matrix multiplication and vector addition, respectively, as is their Cartesian product $\Re^3 \times$ SO(3). This latter product space should not be confused with SE(3), as group multiplication is defined differently in each case. Some other well-known examples of matrix Lie groups include Gl(n), the general linear group of $n \times n$ nonsingular matrices, and Sl(n), the special linear group of $n \times n$ nonsingular matrices with unit determinant.

[2] See, e.g., Boothby [1975] for a background on Lie groups and Riemannian geometry.

Let p be a point on a matrix Lie group $\mathbf{G}$, and $X(t)$ a smooth curve on $\mathbf{G}$ defined over some open interval of 0 such that $X(0) = p$. The derivative $\dot{X}(0)$ is said to be a *tangent vector* to $\mathbf{G}$ at p; the set of all tangent vectors at p, denoted $\mathrm{T}_p\mathbf{G}$, forms a vector space, called the *tangent space to* $\mathbf{G}$ *at* p. The tangent space at the identity $p = I$ is given a special name, called the *Lie algebra* of $\mathbf{G}$, and denoted by a lower-case $\mathbf{g}$. On SO(3) it is easily seen that the Lie algebra so(3) consists of the 3×3 skew-symmetric matrices: if $\Theta(t)$ is a curve on SO(3) such that $\Theta(0) = I$, then differentiating both sides of $\Theta(t)\Theta^T(t) = I$, it follows that $\dot{\Theta}(0) + \dot{\Theta}^T(0) = 0$, so that elements of so(3) are matrices of the form

$$\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \triangleq [\omega]$$

where $\omega \in \Re^3$. Where no confusion arises an element $[\omega] \in \mathrm{so}(3)$ will also be written as $\omega \in \Re^3$. Similarly, the Lie algebra of SE(3), denoted se(3), consists of the 4×4 matrices of the form $\begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix}$ where $[\omega] \in \mathrm{so}(3)$ and $v \in \Re^3$. Elements of se(3) will alternatively be represented as (ω, v). Observe that so(3) and se(3) are vector spaces that can be identified with $\Re^3$ and $\Re^6$, respectively.

Defined on each Lie algebra is the exponential mapping into the corresponding Lie group. On matrix groups the exponential mapping corresponds to the usual matrix exponential, i.e., if A is an element of the Lie algebra, then $\exp A = I + A + A^2/2! + \ldots$ is an element of the Lie group. On so(3) and se(3) the exponential mapping is *onto*, i.e., for every $\Theta \in \mathrm{SO}(3)$ (resp., $X \in \mathrm{SE}(3)$) there exists a $[\omega] \in \mathrm{so}(3)$ (resp., $x \in \mathrm{se}(3)$) such that $\exp[\omega] = \Theta$ (resp., $\exp x = X$). The following explicit formulas for the exponential mappings on so(3) and se(3) are well-known:

Lemma 1 *Given* $[\omega] \in \mathrm{so}(3)$,

$$\exp[\omega] = I + \frac{\sin\|\omega\|}{\|\omega\|} \cdot [\omega] + \frac{1 - \cos\|\omega\|}{\|\omega\|^2} \cdot [\omega]^2$$

where $\|\omega\|$ *is the standard Euclidean norm.*

Lemma 2 *Let* $(\omega, v) \in \mathrm{se}(3)$. *Then*

$$\exp\begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \exp[\omega] & Av \\ 0 & 1 \end{bmatrix}$$

where

$$A = I + \frac{1 - \cos\|\omega\|}{\|\omega\|^2} \cdot [\omega] + \frac{\|\omega\| - \sin\|\omega\|}{\|\omega\|^3} \cdot [\omega]^2$$

Note that if A is an element of some Lie algebra, the set $\{e^{At} | \, t \in \Re\}$ itself forms a group, in this case a subgroup of the Lie group. Such groups are called *one-parameter subgroups* of a Lie group. We shall see shortly that the one-parameter subgroups play an important role in defining distance metrics on SO(3) and SE(3).

It is well known that the exponential map is a homeomorphism[3] over a neighborhood of the identity element of a Lie group. On SO(3) this neighborhood includes essentially the entire group except for a subset of measure zero. Hence, over this neighborhood the inverse of the exponential, or *logarithm*, is a well-defined continuous mapping. The following formula for the logarithm on SO(3) is also well-known:

Lemma 3 *Given* $\Theta \in \mathrm{SO}(3)$ *such that* $\mathrm{Tr}(\Theta) \neq -1$. *Then*

$$\log \Theta = \frac{\phi}{2 \sin \phi}(\Theta - \Theta^T)$$

where ϕ *satisfies* $1 + 2\cos\phi = \mathrm{Tr}(\Theta)$, $|\phi| < \pi$. *Furthermore,* $\|\log\Theta\|^2 = \phi^2$.

Remark 1 Lemmas 1 and 3 suggest the standard visualization of SO(3) as a ball of radius π, centered at the origin with the antipodal points identified; a point ω in the ball represents a rotation by an angle $\|\omega\|$ about the line passing from the origin through ω. Similarly, for any $\Theta \in \mathrm{SO}(3)$, the point $\log\Theta$ in the solid ball represents the rotation Θ. Thus, the image of the logarithm mapping on SO(3) is $\{\omega \in \Re^3 \mid \|\omega\| < \pi\}$. The rotations whose traces equal -1 have a rotation angle of π, and their logarithms are points on the boundary of the solid ball. In this case $\log\Theta$ can have two possible values: if $\hat{\omega}$ is a unit length eigenvector of Θ associated with the eigenvalue 1, then a simple calculation shows that $\log\Theta = \pm\pi[\hat{\omega}]$. This coordinate representation for SO(3) will be referred to as the *canonical coordinates.*

Remark 2 ¿From the previous lemmas and the above remark it can be seen that on SO(3) the preimage of the exponential mapping is multiple-valued: for any $\Theta \in \mathrm{SO}(3)$ such that $[\omega] = \log\Theta$, $\|\omega\| \leq \pi$, it follows that $\Theta = e^{[\omega]+2\pi n[\hat{\omega}]}$ for any integer n, where $\hat{\omega} = \omega/\|\omega\|$. This is akin to the situation in the complex plane, where if $e^{i\phi}$ is a point on the unit circle for some $0 \leq \phi \leq 2\pi$, then $e^{i(\phi+2\pi n)}$ corresponds to the same point for any integer n.

2.2 Riemannian Metrics on SO(3) and SE(3)

A Riemannian metric on a differentiable manifold is a smooth assignment of an inner product to the tangent space at each point on the manifold. Riemannian metrics are the means by which familiar Euclidean concepts like lengths, angles, and volumes can be extended to abstract differentiable manifolds. Recall, for example, that

[3] A *homeomorphism* is a continuous 1-1 mapping whose inverse is also continuous.

the length of a space curve $(x(t), y(t), z(t))$ in $\Re^3$, where $a \leq t \leq b$, is given by $\int_a^b \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}\; dt$. Similarly, the length of a curve $p(t)$ lying on a manifold $\mathcal{M}$, where t is defined over the range $a \leq t \leq b$, can be defined in terms of a Riemannian metric $\langle \cdot, \cdot \rangle$ as

$$L = \int_a^b \langle \dot{p}(t), \dot{p}(t) \rangle_p^{\frac{1}{2}}\; dt$$

where $\langle \cdot, \cdot \rangle_p$ denotes the inner product on $T_p\mathcal{M}$, the tangent space to $\mathcal{M}$ at p. In cases where there is no confusion the subscript p will be omitted from the Riemannian metric symbol.

We now consider the representation of velocities of rigid-body motions. Let $X(t) = (\Theta(t), p(t))$ be a curve in SE(3) describing the trajectory of a rigid body relative to an inertial frame. The tangent vector $\dot{X}(t)$ can then be identified with an element of se(3) in one of two ways: it is easily verified that both $\dot{X}X^{-1} = (\dot{\Theta}\Theta^{-1}, \dot{p} - \dot{\Theta}\Theta^{-1}p)$ and $X^{-1}\dot{X} = (\Theta^{-1}\dot{\Theta}, \Theta^{-1}\dot{p})$ are elements of se(3). The latter is referred to as the *body-fixed velocity* representation of $\dot{X}$, since $\Theta^{-1}\dot{\Theta}$ and $\Theta^{-1}\dot{p}$ are the angular and translational velocities of the rigid body relative to its body-fixed frame, respectively. By similar argument $\dot{X}X^{-1}$ is known as the *inertial velocity* representation of $\dot{X}$. One subtle yet important difference in the interpretation of the inertial velocity representation, however, is that while $\dot{\Theta}\Theta^{-1}$ is indeed the angular velocity of the rigid body relative to the inertial frame, the translational velocity relative to the inertial frame is not $\dot{p} - \dot{\Theta}\Theta^{-1}p$, but simply $\dot{p}$.

Since any tangent vector on SE(3) can be expressed as an element of se(3) by either the inertial or body-fixed velocity representation, it follows that any inner product on se(3) will define two special classes of Riemannian metrics on SE(3). Specifically, let $X_1(t)$ and $X_2(t)$ denote two smooth curves in SE(3) passing through p at $t = 0$. Denote their body-fixed velocities at p by $X_1^{-1}\dot{X}_1 = (\omega_{1b}, v_{1b})$ and $X_2^{-1}\dot{X}_2 = (\omega_{2b}, v_{2b})$, respectively. Now, define an inner product on se(3) by the symmetric positive-definite quadratic form

$$Q = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}$$

so that $\langle \dot{X}_1(0), \dot{X}_2(0) \rangle_p = \omega_{1b}^T A \omega_{2b} + \omega_{1b}^T B^T v_{2b} + v_{1b}^T B \omega_{2b} + v_{1b}^T C v_{2b}$. A Riemannian metric generated in this way is said to be *left-invariant*; if instead we had used the inertial velocity representation $\dot{X}_1 X_1^{-1}$ and $\dot{X}_2 X_2^{-1}$ with the quadratic form Q, the metric would then be called *right-invariant.*

The physical significance of the left- and right-invariant Riemannian metrics is that they are invariant with respect to translations of the inertial and body-fixed reference frames, respectively. That is, under a change of inertial frame the trajectory of a rigid body $X(t)$ is transformed according to $X(t) \mapsto TX(t)$ for some constant $T \in \mathrm{SE}(3)$. Since $(TX)^{-1}(T\dot{X}) = X^{-1}\dot{X}$ it follows that under the left-invariant metric $\langle \dot{X}(t), \dot{X}(t) \rangle = \langle T\dot{X}(t), T\dot{X}(t) \rangle$ for any $T \in \mathrm{SE}(3)$. Similarly, under the right-invariant metric the quantity $\langle \dot{X}(t), \dot{X}(t) \rangle$ is preserved under any transformation of the form $X(t) \mapsto X(t)T$ (which amounts to a change of body-fixed frame). Note that the left- and right-invariant Riemannian metrics can be generalized in the obvious way to general matrix Lie groups.

If the left- and right-invariant metrics defined by Q are identical, the metric is then said to be *bi-invariant.* On SO(3) it is well-known that a family of bi-invariant metrics exists, of the form $Q = cI$, where c is a positive scale factor and I is a 3×3 identity matrix. The metric with $c = 1$ will be referred to as the *standard bi-invariant metric* on SO(3). It is also well-known that SE(3) has no bi-invariant metric (see, e.g., Loncaric [1985], Park [1991]). Since we only require left-invariance (i.e., inertial frame-invariance) for our mechanism problems we choose a left-invariant metric on SE(3) defined by $A = cI$, $C = dI$, and $B = 0$ in the quadratic form Q above; here c and d denote positive scalar constants that act as scale factors for orientation and position, respectively. For this reason we shall refer to this metric as the *scale-dependent left-invariant metric on* SE(3). With this choice of Q the metric restricted to SO(3) is bi-invariant, and also ensures the isotropy of $\Re^3$. Thus, if $X(t)$ is a curve in SE(3) and $X^{-1}\dot{X} = (\omega_b, v_b)$ is its body coordinate velocity, the left-invariant metric applied to the tangent vector of X is

$$\langle \dot{X}, \dot{X} \rangle = c\,\omega_b^T \omega_b + d\,v_b^T v_b \tag{1}$$

Note that angular and rectilinear velocities are being combined in this sum, so that the choice of c and d determines the relative emphases on position and orientation. Since there is no natural choice for these scales, one that is suitable for the application at hand must be chosen. This is an important point worth emphasizing: any scheme for defining distances on SE(3) ultimately depends on a choice of length and angle scales, because of the fundamental geometry of SE(3). In our later examples we suggest some ways of choosing scales that are appropriate for the given class of tasks.

2.3 SO(3) and SE(3) as Metric Spaces

The length of a curve on a Riemannian manifold $\mathcal{M}$ can be defined in terms of the Riemannian metric as follows. Given a curve $X : [a, b] \rightarrow \mathcal{M}$, with Riemannian metric $\langle \cdot, \cdot \rangle$, its length is defined as

$$L = \int_a^b \langle \dot{X}(t), \dot{X}(t) \rangle^{\frac{1}{2}}\, dt$$

By this definition the length of the curve is invariant under reparametrizations of $[a, b]$; L can therefore be regarded as the arc-length of the curve. $\mathcal{M}$ then admits the structure of a metric space, in which the distance between two points X_1 and X_2 is defined as the infimum of the lengths of piecewise-differentiable curves from X_1 to X_2. Unfortunately the square root in the integrand complicates the Euler-Lagrange equations for the length functional. It is more convenient to consider the *energy* functional

$$E = \int_a^b \langle \dot{X}(t), \dot{X}(t) \rangle \, dt$$

Critical points of E are known as *geodesics*, and it is well-known that the geodesics are also critical points of L. While E is clearly dependent upon the parametrization of the curve, it turns out that the geodesics are automatically parametrized proportional to arc-length – said another way, if $X(t)$ is a geodesic, then $\langle \dot{X}(t), \dot{X}(t) \rangle$ is constant along the entire length of $X(t)$. In short, the minimum-length curve between two points on a Riemannian manifold (i.e., the curve minimizing L) corresponds to the minimum-length geodesic[4] (i.e., the curve minimizing E) between these two points. For this reason the minimum-length curves are also referred to as the *minimal geodesics.*

To regard SO(3) and SE(3) as metric spaces the corresponding minimal geodesics on these Lie groups need to be determined. The geodesics on SO(3) (with respect to the bi-invariant metric) are given by left and right translations of the one-parameter subgroups $e^{[\omega]t}$, where $[\omega] \in$ so(3) and $t \in \Re$ (see, e.g., Boothby [1975]). While in general there may exist more than one minimal geodesic between two points on an arbitrary Riemannian manifold (consider, for example, two antipodal points on the sphere), on SO(3) the minimal geodesics are especially easy to identify in terms of the canonical coordinates. Specifically, the inverse images of the one-parameter subgroups correspond to straight lines in $\Re^3$ passing through the origin – this is at once immediate from the solid ball analogy of SO(3). Moreover, the length of the minimal geodesic is especially simple to compute:

Theorem 1 *Let $\Theta_1, \Theta_2 \in$ SO(3). Then the distance $L = d(\Theta_1, \Theta_2)$ induced by the standard bi-invariant metric on* SO(3) *is*

$$d(\Theta_1, \Theta_2) = \| \log(\Theta_1^{-1}\Theta_2) \|$$

where $\| \cdot \|$ denotes the standard Euclidean norm.

Proof: SO(3), being a compact simply-connected Lie group, has a bi-invariant Riemannian metric corresponding to the negative of its Killing form; it is well-known that the geodesics on Lie groups with bi-invariant metric are formed by left- and right-translations of the one-parameter subgroups. Let $\langle \cdot, \cdot \rangle$ denote the standard bi-invariant Riemannian metric on SO(3). Then given a curve $\Theta : [0,1] \rightarrow$ SO(3), with $\Theta(0) = \Theta_1$ and $\Theta(1) = \Theta_2$,

$$\begin{aligned} \langle \dot{\Theta}, \dot{\Theta} \rangle^{\frac{1}{2}} &= \| \Theta^{-1} \dot{\Theta} \| \\ &= \sqrt{\frac{1}{2} \mathrm{Tr} \left((\Theta^{-1}\dot{\Theta})(\Theta^{-1}\dot{\Theta})^T \right)} \end{aligned}$$

[4] There is another technical condition that the manifold must be *geodesically complete* in order for this to be true; since the manifolds considered in this paper all satisfy this condition, we shall not dwell on this point any further.

The minimal geodesic from Θ_1 to Θ_2 is then $\Theta(t) = \Theta_1 e^{[\omega]t}$, where $[\omega] = \log(\Theta_1^{-1}\Theta_2)$. Its length is therefore $d(\Theta_1, \Theta_2) = \int_0^1 \langle \dot{\Theta}, \dot{\Theta} \rangle^{\frac{1}{2}}\, dt = \sqrt{\frac{1}{2}\mathrm{Tr}([\omega][\omega]^T)} = \|\omega\|$ as claimed. □

Remark 3 A simple calculation shows that this distance measure is invariant with respect to both left and right translations; that is, $d(T\Theta_1, T\Theta_2) = d(\Theta_1 T, \Theta_2 T) = d(\Theta_1, \Theta_2)$ for any $T \in \mathrm{SO}(3)$. Also, from the solid ball analogy of SO(3) there exist two minimal geodesics when $\mathrm{Tr}(\Theta_1^{-1}\Theta_2) = -1$, given in canonical coordinates by the two line segments from the origin to the two antipodal points representing $\log(\Theta_1^{-1}\Theta_2)$. Clearly the lengths of the two geodesics are identical, so that the distance formula is valid for either value of $\log(\Theta_1^{-1}\Theta_2)$.

The situation in SE(3) is unfortunately more complicated, primarily because the one-parameter subgroups e^{At}, $A \in \mathrm{se}(3)$ and $t \in \Re$, are no longer geodesics on SE(3). However, the geodesics can be obtained from the geodesics on the product space $\Re^3 \times \mathrm{SO}(3)$ as follows. To determine the minimal geodesics (with respect to the scale-dependent left-invariant Riemannian metric) between two points $(\Theta_1, b_1), (\Theta_2, b_2) \in \mathrm{SE}(3)$, we first determine the minimal geodesic between Θ_1 and Θ_2 on SO(3), denoted $\Theta^*(t)$, and the minimal geodesic in $\Re^3$ between b_1 and b_2, denoted $b^*(t)$; note that $b^*(t)$ is simply the straight line connecting b_1 and b_2, and $\Theta^*(t)$ is found with respect to the usual bi-invariant metric as before. The minimal geodesic on SE(3) is then $(\Theta^*(t), b^*(t))$, and the geodesic distance on SE(3) is given by the following simple formula:

Theorem 2 *Let $X_1 = (\Theta_1, b_1)$ and $X_2 = (\Theta_2, b_2)$ be two points in* SE(3). *Then the distance $L = d(X_1, X_2)$ induced by the scale-dependent left-invariant metric on* SE(3) *is*

$$d(X_1, X_2) = \sqrt{c\|\log(\Theta_1^{-1}\Theta_2)\|^2 + d\|b_2 - b_1\|^2}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Remark 4 Because the Riemannian metric is only left-invariant, the above distance measure on SE(3) is invariant with respect to only left-translations: $d(X_1, X_2) = d(TX_1, TX_2)$ for any $T \in \mathrm{SE}(3)$, so that in particular the distance between X_1 and X_2 is the same as the distance between I and $X_1^{-1}X_2$. In general, however, $d(X_1, X_2) \neq d(X_1 T, X_2 T)$ for $T \in \mathrm{SE}(3)$. Of course, use of the right-invariant Riemannian metric would lead to a distance measure that was invariant with respect to right rather than left translations. However, since our applications require inertial-frame invariance we shall adhere to the left-invariant distance measure proposed.

Example 1 We compute the distance between two points in SE(3) using the left-invariant distance metric above. Let $X_1 = (\Theta_1, b_1)$ and $X_2 = (\Theta_2, b_2)$ be two points in SE(3), and let $\Omega = \log(\Theta_1^{-1}\Theta_2)$. Then the minimal geodesic from X_1 to X_2 is given by the curve $X : [0,1] \rightarrow \mathrm{SE}(3)$, $X(t) = (\Theta_1 \exp \Omega t, (1-t)b_1 + tb_2)$. If $\langle \cdot, \cdot \rangle$ denotes the scale-dependent left-invariant metric on SE(3) with $c = d = 1$, then the distance between X_1 and X_2 is

$$\begin{aligned} d(X_1, X_2) &= \int_0^1 \langle \dot{X}(t), \dot{X}(t) \rangle^{\frac{1}{2}} dt \\ &= \int_0^1 \| X^{-1} \dot{X} \| \, dt \\ &= \sqrt{\|\Omega\|^2 + \|b_2 - b_1\|^2} \end{aligned}$$

where $\| \cdot \|$ denotes the Euclidean norm.

3 Applications to Mechanism Design

The workspace fitting problem of mechanism design can now be stated as follows. Let $\mathcal{M}$ denote the joint-space manifold of the mechanism in question, and $f : \mathcal{M} \to \mathrm{SE}(3)$ its forward kinematic map. Suppose further that f is a function of the joint variable vector x and a vector of kinematic parameters A: $f = f(x, A)$. Given a set of desired goal frames $\{X_1, X_2, \ldots, X_p\} \subseteq \mathrm{SE}(3)$, the general form of the workspace fitting problem can now be stated as finding the kinematic parameters A and the location of the base and tool frames, $B, T \in \mathrm{SE}(3)$, that minimize

$$J(A, B, T) = \sum_{i=1}^{p} \min_x d^2(B \cdot f(x, A) \cdot T, X_i)$$

We consider three numerical examples of the workspace fitting problem to illustrate our design methodology.

Example 2 We determine the optimal base location for a $1R$ planar mechanism with unit link length, given 4 goal frames in the plane. The position and orientation of the goal frames relative to the inertial frame are marked in the table below as (x_d, y_d, θ_d). Distances are measured with $c = d = 1$ in our scale-dependent distance measure. The location of the base frame that minimizes the distance between the workspace and the set of desired frames is determined numerically to be $(x, y) = (0.108, 0.110)$. The workspace points closest to the goal frames are given by (x_a, y_a, θ_a) below:

Frame	x_d	y_d	θ_d	x_a	y_a	θ_a
1	1.0	0.0	0°	1.10	0.04	−2.6°
2	1.0	1.0	45°	0.82	0.81	45.8°
3	0.0	1.0	90°	0.06	1.11	94.0°
4	1.0	0.5	30°	1.01	0.55	27.6°

To demonstrate the dependence of the solution on choice of length scales, we now consider two goal frames, and vary the position and orientation scale factors c and d. The goal frames are

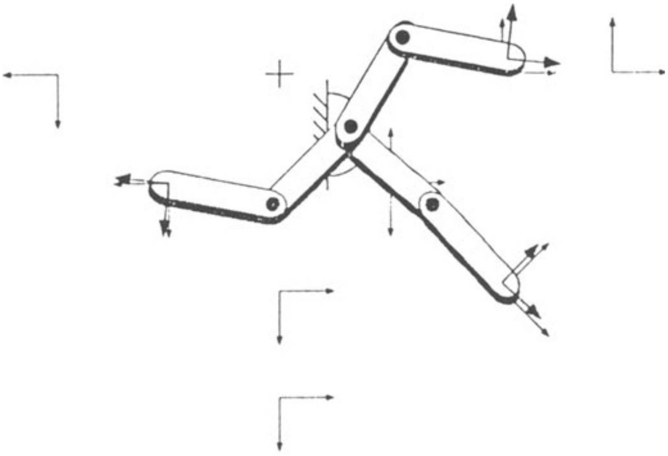


Figure 1: Finding the optimal base location for a $2R$ planar chain.

Frame	x_d	y_d	θ_d
1	0.0	1.0	0°
2	0.0	-1.0	0°

With $c = 0$ and $d = 1$ the optimal base location for the unit link length $1R$ mechanism is the origin, as expected - the mechanism is able to reach both goal positions $(0, 1)$ and $(0, -1)$, with orientation errors ignored since $c = 0$. As $d/c \rightarrow 0$ the base location asymptotically approaches $(-1, 0)$ along the x-axis, with numerical difficulties encountered as d/c becomes very small. Of course, when $d = 0$ and $c > 0$ no unique solution exists; any base location will result in a zero total distance, since the mechanism can always achieve the desired goal frame orientations. The translating base location $(0, 0) \rightarrow (-1, 0)$ along the x-axis agrees with our intuition, since we are increasingly emphasizing orientation accuracy over position accuracy as $d/c \rightarrow 0$.

Example 3 Now consider the problem of finding the optimal base location of a $2R$ planar open chain, given 9 goal frames in the plane. Suppose each link is of unit length, and that the scale parameters are set to $c = d = 1$. The nine desired goal frames of the end-effector are given by (x_d, y_d, θ_d) in the table below. The base location that minimizes the distance between the workspace and the desired frames is determined numerically to be $(x, y) = (0.583, 0.584)$. Figure 1 shows the optimal placement for the planar dyad relative to the goal frames. The three depicted configurations of the dyad are the locations where the distance is minimized between the desired and achieved end-effector frames for frames 2, 6 and 9.

Example 4 In the final example we determine the optimal joint-axis direction for the base link of a spherical dyad, given seven goal frames. The twist angles of the dyad are 90 degrees, and the seven desired goal frames are specified by their *x*-*y*-*z* Euler angles $(\theta_d, \phi_d, \psi_d)$ in the table below. The optimal direction for the base joint axis (measured in the inertial frame) is numerically determined to be $(-0.113, 0.966, 0.231)$. The points in the workspace closest to the seven goal frames are given by $(\theta_a, \phi_a, \psi_a)$ below:

Frame	θ_d	ϕ_d	ψ_d	θ_a	ϕ_a	ψ_a
1	90°	0°	0°	92.78°	−3.149°	−1.175°
2	130°	10°	−65°	126.4°	−0.176°	−59.99°
3	90°	0°	90°	100.4°	9.721°	90.42°
4	125°	30°	−10°	128.3°	25.74°	−8.527°
5	145°	25°	−15°	142.5°	29.61°	−18.96°
6	105°	35°	55°	99.08°	34.41°	55.92°
7	110°	5°	0°	106.1°	8.982°	0.035°

Distances in this example are measured only in SO(3), which, recall, has a natural (bi-invariant) Riemannian metric. Therefore, for purely spherical mechanisms the solutions are independent of the scale parameter *c*.

The previous three examples demonstrate that the distance metric can be applied successfully to mechanism design problems to produce physically appealing results. While the choice of length scale obviously affects certain design problems, it should be emphasized that this length scale dependence is not a consequence of the design algorithm, but rather an intrinsic geometric feature of SE(3). Engineering considerations can typically determine the appropriate length scale for a given problem. One possible method of treating position and orientation equally is to choose the scales *c* and *d* such that the position and orientation volumes of the resulting workspace are more or less equal. Specifically, the Cartesian positions of the goal frames can be used to obtain an estimate of the Cartesian workspace volume. Since the volume of SO(3) with respect to the standard bi-invariant metric is $8\pi^2$ (corresponding to setting $c = 1$, see Park [1991]), the length scale *d* can be chosen such that the Cartesian workspace volume is approximately $8\pi^2$.

4 Conclusion

In this paper we have formulated, using ideas from Riemannian geometry, a distance metric on SE(3) that is invariant with respect to choice of inertial frame. This metric preserves the isotropy of physical space, and also reflects the fact that no natural length scale for physical space exists. Moreover, when the metric is restricted to SO(3) it has the added feature of being invariant with respect to both the inertial and body-fixed frames. We argue that this metric is not only physically correct but also easy to compute. Application of this metric to mechanism design problems involving best workspace fit have been presented.

Because the fundamental geometric structure of SE(3) does not admit a natural length or angle scale, some ways have been suggested of choosing these scales appropriate for a given mechanism design problem. Of potentially greater benefit would be the ability to prescribe tolerances for the workspace: given a set of goal frames *and* a set of maximum allowable position and orientation errors for each goal frame, the design system automatically generates a mechanism that conforms to these specified tolerances. At a more fundamental level, we believe our results are useful for any kinematic application in which the notion of SE(3) as a metric space is required.

References

Bodduluri, R.M.C. 1990. Design and planned movement of multi-degree-of-freedom spatial mechanisms. Ph.D. thesis. Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA.

Boothby, W. 1975. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* New York: Academic Press.

Cheeger, J., and Ebin, D. G. 1975. *Comparison Theorems in Riemannian Geometry.* Amsterdam: North-Holland.

Chevalley, C. 1946. *Theory of Lie Groups.* Princeton, NJ: Princeton University Press.

Gallot, S., Hulin, D., and Lafontaine, J. 1990. *Riemannian Geometry.* Berlin: Springer-Verlag.

Loncaric, J. 1985. Geometric analysis of compliant mechanisms in robotics. Ph.D. thesis. Harvard University, Cambridge, MA.

Ravani, B. 1982. Kinematic mapping as applied to motion approximation and mechanism synthesis. Ph.D. thesis. Department of Mechanical Engineering, Stanford University, Stanford, CA.

Park, F. C. 1991. The optimal kinematic design of mechanisms. Ph.D. thesis. Harvard University, Cambridge, MA.

Tchoń, K., and Dulęba, I., 1990. A kinematic metric for robot manipulators. *Instytut Cybernetyki Technicznej* preprint 7-90, Politechnika Wroclawska, Wroclaw, Poland.

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

Aims and Scope of the Series

The fundamental questions arising in mechanics are: *Why?, How?,* and *How much?* The aim of this series is to provide lucid accounts written by authoritative researchers giving vision and insight in answering these questions on the subject of mechanics as it relates to solids. The scope of the series covers the entire spectrum of solid mechanics. Thus it includes the foundation of mechanics; variational formulations; computational mechanics; statics, kinematics and dynamics of rigid and elastic bodies; vibrations of solids and structures; dynamical systems and chaos; the theories of elasticity, plasticity and viscoelasticity; composite materials; rods, beams, shells and membranes; structural control and stability; soils, rocks and geomechanics; fracture; tribology; experimental mechanics; biomechanics and machine design.

1. R.T. Haftka, Z. Gürdal and M.P. Kamat: *Elements of Structural Optimization.* 2nd rev.ed., 1990 ISBN 0-7923-0608-2
2. J.J. Kalker: *Three-Dimensional Elastic Bodies in Rolling Contact.* 1990 ISBN 0-7923-0712-7
3. P. Karasudhi: *Foundations of Solid Mechanics.* 1991 ISBN 0-7923-0772-0
4. N. Kikuchi: *Computational Methods in Contact Mechanics.* (forthcoming) ISBN 0-7923-0773-9
5. *Not published.*
6. J.F. Doyle: *Static and Dynamic Analysis of Structures.* With an Emphasis on Mechanics and Computer Matrix Methods. 1991 ISBN 0-7923-1124-8; Pb 0-7923-1208-2
7. O.O. Ochoa and J.N. Reddy: *Finite Element Analysis of Composite Laminates.* ISBN 0-7923-1125-6
8. M.H. Aliabadi and D.P. Rooke: *Numerical Fracture Mechanics.* ISBN 0-7923-1175-2
9. J. Angeles and C.S. López-Cajún: *Optimization of Cam Mechanisms.* 1991 ISBN 0-7923-1355-0
10. D.E. Grierson, A. Franchi and P. Riva: *Progress in Structural Engineering.* 1991 ISBN 0-7923-1396-8
11. R.T. Haftka and Z. Gürdal: *Elements of Structural Optimization.* 3rd rev. and exp. ed. 1992 ISBN 0-7923-1504-9; Pb 0-7923-1505-7
12. J.R. Barber: *Elasticity.* 1992 ISBN 0-7923-1609-6; Pb 0-7923-1610-X
13. H.S. Tzou and G.L. Anderson (eds.): *Intelligent Structural Systems.* 1992 ISBN 0-7923-1920-6
14. E.E. Gdoutos: *Fracture Mechanics.* An Introduction. 1993 ISBN 0-7923-1932-X
15. J.P. Ward: *Solid Mechanics.* An Introduction. 1992 ISBN 0-7923-1949-4
16. M. Farshad: *Design and Analysis of Shell Structures.* 1992 ISBN 0-7923-1950-8
17. H.S. Tzou and T. Fukuda (eds.): *Precision Sensors, Actuators and Systems.* 1992 ISBN 0-7923-2015-8
18. J.R. Vinson: *The Behavior of Shells Composed of Isotropic and Composite Materials.* 1993 ISBN 0-7923-2113-8

Kluwer Academic Publishers – Dordrecht / Boston / London

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

19. H.S. Tzou: *Piezoelectric Shells.* Distributed Sensing and Control of Continua. 1993 ISBN 0-7923-2186-3
20. W. Schiehlen: *Advanced Multibody System Dynamics.* Simulation and Software Tools. 1993 ISBN 0-7923-2192-8
21. C.-W. Lee: *Vibration Analysis of Rotors.* 1993 ISBN 0-7923-2300-9
22. D.R. Smith: *An Introduction to Continuum Mechanics.* 1993 ISBN 0-7923-2454-4
23. G.M.L. Gladwell: *Inverse Problems in Scattering.* An Introduction. 1993 ISBN 0-7923-2478-1
24. G. Prathap: *The Finite Element Method in Structural Mechanics.* 1993 ISBN 0-7923-2492-7
25. J. Herskovits (ed.): *Structural Optimization '93.* 1993 ISBN 0-7923-2510-9
26. M.A. González-Palacios and J. Angeles: *Cam Synthesis.* 1993 ISBN 0-7923-2536-2

Kluwer Academic Publishers – Dordrecht / Boston / London

Mechanics

FLUID MECHANICS AND ITS APPLICATIONS

Series Editor: R. Moreau

Aims and Scope of the Series

The purpose of this series is to focus on subjects in which fluid mechanics plays a fundamental role. As well as the more traditional applications of aeronautics, hydraulics, heat and mass transfer etc., books will be published dealing with topics which are currently in a state of rapid development, such as turbulence, suspensions and multiphase fluids, super and hypersonic flows and numerical modelling techniques. It is a widely held view that it is the interdisciplinary subjects that will receive intense scientific attention, bringing them to the forefront of technological advancement. Fluids have the ability to transport matter and its properties as well as transmit force, therefore fluid mechanics is a subject that is particularly open to cross fertilisation with other sciences and disciplines of engineering. The subject of fluid mechanics will be highly relevant in domains such as chemical, metallurgical, biological and ecological engineering. This series is particularly open to such new multidisciplinary domains.

1. M. Lesieur: *Turbulence in Fluids.* 2nd rev. ed., 1990 ISBN 0-7923-0645-7
2. O. Métais and M. Lesieur (eds.): *Turbulence and Coherent Structures.* 1991 ISBN 0-7923-0646-5
3. R. Moreau: *Magnetohydrodynamics.* 1990 ISBN 0-7923-0937-5
4. E. Coustols (ed.): *Turbulence Control by Passive Means.* 1990 ISBN 0-7923-1020-9
5. A.A. Borissov (ed.): *Dynamic Structure of Detonation in Gaseous and Dispersed Media.* 1991 ISBN 0-7923-1340-2
6. K.-S. Choi (ed.): *Recent Developments in Turbulence Management.* 1991 ISBN 0-7923-1477-8
7. E.P. Evans and B. Coulbeck (eds.): *Pipeline Systems.* 1992 ISBN 0-7923-1668-1
8. B. Nau (ed.): *Fluid Sealing.* 1992 ISBN 0-7923-1669-X
9. T.K.S. Murthy (ed.): *Computational Methods in Hypersonic Aerodynamics.* 1992 ISBN 0-7923-1673-8
10. R. King (ed.): *Fluid Mechanics of Mixing.* Modelling, Operations and Experimental Techniques. 1992 ISBN 0-7923-1720-3
11. Z. Han and X. Yin: *Shock Dynamics.* 1993 ISBN 0-7923-1746-7
12. L. Svarovsky and M.T. Thew (eds.): *Hydroclones.* Analysis and Applications. 1992 ISBN 0-7923-1876-5
13. A. Lichtarowicz (ed.): *Jet Cutting Technology.* 1992 ISBN 0-7923-1979-6
14. F.T.M. Nieuwstadt (ed.): *Flow Visualization and Image Analysis.* 1993 ISBN 0-7923-1994-X
15. A.J. Saul (ed.): *Floods and Flood Management.* 1992 ISBN 0-7923-2078-6
16. D.E. Ashpis, T.B. Gatski and R. Hirsh (eds.): *Instabilities and Turbulence in Engineering Flows.* 1993 ISBN 0-7923-2161-8
17. R.S. Azad: *The Atmospheric Boundary Layer for Engineers.* 1993 ISBN 0-7923-2187-1
18. F.T.M. Nieuwstadt (ed.): *Advances in Turbulence IV.* 1993 ISBN 0-7923-2282-7
19. K.K. Prasad (ed.): *Further Developments in Turbulence Management.* 1993 ISBN 0-7923-2291-6
20. Y.A. Tatarchenko: *Shaped Crystal Growth.* 1993 ISBN 0-7923-2419-6

Kluwer Academic Publishers – Dordrecht / Boston / London