Springer Tracts in Advanced Robotics 96

Gianluca Antonelli

Underwater Robots

Third Edition

star

Springer

Springer Tracts in Advanced Robotics

Volume 96

For further volumes:
http://www.springer.com/series/5208

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices of EURON (European Robotics Research Network)

Gianluca Antonelli

Underwater Robots

Third Edition

Springer

Gianluca Antonelli
Dipartimento di Ingegneria Elettrica e
dell'Informazione
Università di Cassino e Lazio Meridionale
Cassino
Italy

ISSN 1610-7438 ISSN 1610-742X (electronic)
ISBN 978-3-319-02876-7 ISBN 978-3-319-02877-4 (eBook)
DOI 10.1007/978-3-319-02877-4
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013953199

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

A “bish–bish” Andrea e “demone” Giustina

E la locomotiva sembrava fosse un mostro strano che l'uomo dominava con il pensiero e con la mano…

Francesco Guccini, La locomotiva, 1972

Foreword

Robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into human environments and vigorously engaged in its new challenges. Interacting with, assisting, serving, and exploring with humans, the future robots will increasingly touch people and their lives.

Beyond its impact on physical robots, the body of knowledge robotics has produced reveals a much wider range of applications reaching across diverse research areas and scientific disciplines, such as biomechanics, haptics, neurosciences, virtual simulation, animation, surgery, and sensor networks among others. In return, the challenges of the new emerging areas are proving an abundant source of stimulation and insights for the field of robotics. It is indeed at the intersection of disciplines that the most striking advances happen.

The *Springer Tracts in Advanced Robotics (STAR)* is devoted to bringing to the research community the latest advances in the robotics field on the basis of their significance and quality. Through a wide and timely dissemination of critical research developments in robotics, our objective with this series is to promote more exchanges and collaborations among the researchers in the community and contribute to further advancements in this rapidly growing field.

The volume by Gianluca Antonelli is the Third Edition of a successful monograph, which was one of the first volumes to be published in the series and a bestseller for the two previous editions throughout the years. Being focused on an important class of robotic systems, namely underwater vehicle-manipulator systems, this volume expands the manipulation aspects which are at the basis of the work. Further to updating the state-of-the-art in the field, the educational value of the contents are enhanced by resorting to a platform-independent simulation software tool.

A well-assessed blend of theoretical and experimental results, this volume confirms as a classic in our STAR series!

Naples, Italy, July 2013 Bruno Siciliano

Preface to the Third Edition

Whenever possible, I attend the IEEE ICRA (International Conference on Robotics and Automation) conference where I usually meet Thomas, a Springer Senior Editor, who updates me on the sellings of this monograph. Despite the very niche topic, underwater manipulation, sellings are still interesting so that a Third Edition is worth.

Ten years after the First Edition, I decided to work on this project by reconsidering some of the material. First of all, I have withdrawn one Chapter, the one devoted to multi vehicles coordination, since it was somehow orthogonal to the remaining part of the book. It dealt with underwater and control but was not specific to manipulation. It was also too short to cover in a decent way all the challenges of this control problem.

I have obviously updated most references, mainly adding then deleting existing ones, and clarified some of the modeling parts based on the feedback from the readers. Some details on the way the Jacobians are computed, as well as all the Jacobians used in the kinematic control Chapter, have been streamlined and explicitly reported.

The First Edition of this monograph was based on my Ph.D. work, done during the 1997–1999, and most of the material was of *research* kind. As a matter of fact, at the time of publication some of the journal papers were still in their reviewing process. After ten years, it is interesting to notice how part of the material lost his innovative halo to acquire an educational one. For this reason, Chapters containing experiments physically performed in the end of last century have been kept. On this trail is also the decision to give the simulation tool and write a small reference Chapter to it together with the dynamic parameters of several models used in the testing of the algorithms. For the online version, color and hyper references have been included too. The reader will not find in this monograph any details on the sensorial or communication systems, moreover, if he is interested in vehicle control alone, i.e., without manipulator, I would suggest one of the Fossen's books.

Cassino, Italy, July 2013 Gianluca Antonelli

Preface to the Second Edition

The purpose of this Second Edition is to add material not covered in the First Edition as well as streamline and improve the previous material.

The organization of the book has been substantially modified, an introductory Chapter containing the state of the art has been considered; the modeling Chapter is substantially unmodified. In Chapter 3, the problem of controlling a 6-Degrees-Of-Freedoms (DOFs) Autonomous Underwater Vehicle (AUV) is investigated. Chapter 4 is a new Chapter devoted at a survey of fault detection/tolerant strategies for ROVs/AUVs, it is mainly based on the Chapter published in [6]. The following Chapter (Chapter 5) reports experimental results obtained with the vehicle ODIN. The following 3 Chapters, from Chapter 6 to Chapter 8 are devoted at presenting kinematic, dynamic, and interaction control strategies for Underwater Vehicle Manipulator Systems (UVMSs); new material has been added; thanks also to several colleagues who provided me with valuable material, I warmly thank all of them. The content of Chapter 9[1] is new in this Second Edition and reports preliminary results on the emerging topic of coordinated control of platoon of AUVs. Finally, the bibliography has been updated.

The reader might be interested in knowing what she/he will not find in this book. Since the core of the book is the coordinated control of manipulators mounted on underwater vehicles, control of non-holonomic vehicles is not dealt with; this is an important topic also in view of the large number of existing *torpedo*-like vehicles. Another important aspect concerns the sensorial apparatus, both from the technological point of view and from the algorithmic aspect; most of the AUVs are equipped with redundant sensorial systems required both for localization/navigation purposes and for fault detection/tolerant capabilities. Actuation is mainly obtained by means of thrusters; those are still object of research for the modeling characteristics and might be the object of improvement in terms of dynamic response.

Cassino, Italy, January 2006 Gianluca Antonelli

[1] Chapter removed in the third edition

Preface to the First Edition

Underwater Robotics have known in the last few years an increasing interest from research and industry. Currently, the use of manned underwater robotics systems to accomplish missions as sea bottom and pipeline survey, cable maintenance, offshore structures' monitoring and maintenance, collect/release of biological surveys is common. The strong limit of the use of manned vehicles is the enormous cost and risk in working in such a hostile environment. The aim of the research is to progressively make it possible to perform such missions in a completely autonomous way.

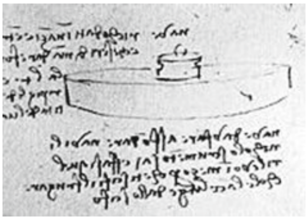
This objective is challenging from the technological as well as from the theoretical aspects since it implies a wide range of technical and research topics. Sending an autonomous vehicle in an unknown and unstructured environment, with limited online communication, requires some on board *intelligence* and the ability of the vehicle to react in a reliable way to unexpected situations. Techniques as artificial intelligence, neural network, discrete events, fuzzy logic can be useful in this *high* level mission control. The sensory system of the vehicle must deal with a noisy and an unstructured environment; moreover, technologies as GPS are not applicable due to the impossibility to underwater electromagnetic transmission; vision based systems are not fully reliable due to the generally poor visibility. The actuating system is usually composed of thrusters and control surfaces; all of them have a non-linear dynamics and are strongly affected by the hydrodynamic effects.

In this framework, the use of a manipulator mounted on an autonomous vehicle plays an important role. From the control point of view, underwater robotics is much more challenging with respect to ground robotics since the former deal with unstructured environments, mobile base, significant external disturbance, low bandwidth of sensory and actuating systems, difficulty in the estimation of the dynamic parameters, and highly non-linear dynamics.

Referring to Autonomous Underwater Vehicles (AUVs), i.e., untethered, unmanned vehicles to be used mainly in survey missions, [304, 331] present the state of the art of several existing AUVs and their control architecture. Currently, there are more than 46 AUV models [331], among others: *ABE* of the Woods Hole Oceanographic Institution (MA, USA), *MARIUS* developed under the Marine

Science and Technology Programme of the IV framework of European Commission (Lisbon, Portugal), *ODIN* designed at the Autonomous Systems Laboratory of the University of Hawaii (Honolulu, HI, USA), *OTTER* from the Monterey Bay Aquarium and Stanford University (CA, USA), *Phoenix* and *ARIES* belonging to the Naval Postgraduate School (Monterey, CA, USA), *Twin Burgers* developed at the University of Tokyo (Tokyo, Japan), *Theseus* belonging to ISE Research Ltd (Canada). Reference [78] shows the control architecture of *VORTEX*, a vehicle developed by Inria and Ifremer (France), and *OTTER*. Focusing on the low level motion control of AUVs, most of the proposed control schemes take into account the uncertainty in the model by resorting to an adaptive strategy [72, 77, 112, 113, 123, 324] or a robust approach [75, 79, 130, 196, 270, 319, 320]. In [130], an estimation of the dynamic parameters of the vehicle NPS AUV *Phoenix* is also provided. An overview of control techniques for AUVs is reported in [108].

As a curiosity, in the Figure below, there is a draw of one of the first *manned* underwater vehicles. It was found in the *Codice Atlantico* (Codex Atlanticus), written by Leonardo Da Vinci between 1480 and 1518, together with the development of some diver's devices. Legends say that Leonardo worked on the idea of an underwater military machine that he further destroyed by himself the results judged too dangerous. Maybe the first idea of an underwater machine is from Aristotle; following the legend he built a machine: *skaphe andros* (boat-man) that allowed Alexander the Great to stay in deep for at least half a day during the war of Tiro in 325 b.C. This is unrealistic, of course, also considering that the Archimedes's law was still to become a reality (around 250 b.C.).



Draw of the manned underwater vehicle developed by Leonardo Da Vinci

The current technology in control of underwater manipulation is limited to the use of a master/slave approach in which a skilled operator has to move a master manipulator that works as *joystick* for the slave manipulator that is performing the task [45, 298]. The limitations of such a technique are evident: the operator must be well trained, underwater communication is hard and a significant delay in the control is experienced. Moreover, if the task has to be performed in deep waters, a manned underwater vehicle close to the unmanned vehicle with the manipulator needs to be considered to overcome the communication problems thus leading to enormous cost increasing. Few research centers are equipped with an autonomous Underwater Vehicle-Manipulator System. Among the others:

- *ODIN* and *OTTER* can be provided with a one-/two-link manipulator to study the interaction of the manipulator and the vehicle in order to execute automatic retrieval tasks [307];
- on *VORTEX* a seven-link manipulator (*PA10*) can be mounted with a large inertia with respect to the vehicle that implies a strong interaction between them;
- *SAUVIM*, a semi-autonomous vehicle with an *Ansaldo* seven-link manipulator is under development at the Autonomous Systems Laboratory of the University of Hawaii; this vehicle, in the final version, will be able to operate at the depth of 4,000 m.
- *AMADEUS*, an acronym for Advanced MAnipulation for DEep Underwater Sampling, funded by the European Commission, that involved the Heriot-Watt University (UK), the Università di Genova (Italy), CNR Istituto Automazione Navale, (Italy), the Universitat de Barcelona (Spain), and the Institute of Marine Biology of Crete (Greece). The project focused on the co-ordinated control of two teleoperated underwater *Ansaldo* seven-link manipulators and the development of an underwater hand equipped with a slip sensor.

Focusing on the motion control of UVMSs, [45, 146] present a telemanipulated arm; in [182] an *intelligent* underwater manipulator prototype is experimentally validated; [52, 53, 54] present some simulation results on a Composite Dynamics approach for *VORTEX/PA10*; [83] evaluates the dynamic coupling for a specific UVMS; adaptive approaches are presented in [107, 186, 187]. Reference [204] reports some interesting experiments of coordinated control. Very few papers investigated the redundancy resolution of UVMSs by applying inverse kinematics algorithm with different secondary tasks [21, 24, 23, 259, 260].

This book deals with the main control aspects in underwater manipulation tasks and dynamic control of AUVs. First, the mathematical model is discussed; the aspects with significant impact on the control strategy will be remarked. In Chap. 6, kinematic control for underwater manipulation is presented. Kinematic control plays a significant role in unstructured robotics where off-line trajectory planning is not a reliable approach; moreover, the vehicle-manipulator system is often kinematically redundant with respect to the most common tasks and redundancy resolution algorithms can then be applied to exploit such characteristic. Dynamic control is then discussed in Chap. 7; several motion control schemes are analyzed and presented in this book. Some experimental results with the autonomous vehicle *ODIN* (without manipulator) are presented, moreover some theoretical results on adaptive control of AUVs are discussed. In Chap. 8, the interaction with the environment is detailed. Such kind of operation is critical in underwater manipulation for several reasons that do not allow direct implementation of the force control strategies developed for ground robotics. Finally, after having developed some conclusions, a simulation tool for multi-body systems is presented. This software package, developed for testing the control strategies studied along the book, has been designed according to modular requirements that make it possible to generate generic robotic systems in any desired environment.

Napoli, August 2002 Gianluca Antonelli

Acknowledgments

The contributions of a quite large number of people were determinant for the realization of this monograph.

Stefano Chiaverini was my Co-tutor during my Ph.D. at the Università degli Studi di Napoli Federico II, he is now Full Professor at the Università degli Studi di Cassino e Lazio Meridionale. Since the beginning of the doctoral experience up to present most of the research that I have done is shared with him. As a matter of fact, I consider myself as a Co-author of this monograph. Furthermore, Stefano was, somehow, also *responsible* of my decision to join the academic career.

Lorenzo Sciavicco developed a productive, and friendly, research environment in Napoli that was important for my professional growing. Bruno Siciliano, my Tutor both in the Master and Ph.D. thesis, to whom goes my warmest acknowledgments.

Fabrizio Caccavale, currently at the Università della Basilicata, Giuseppe Fusco currently at the Università degli Studi di Cassino e Lazio Meridionale, Tarun Podder, currently at the Case Western Reserve University, Nilanjan Sarkar, currently at the Vanderbilt University, Luigi Villani currently at the Università degli Studi di Napoli Federico II, Michael West, currently at the Georgia Tech; all of them are Co-authors of the *wet* papers embedded in this monograph and deserve a lot of credit for this work.

During the Ph.D., I have been Visiting Researcher at the Autonomous Systems Laboratory of the University of Hawaii where I carried out some experiments on dynamic control of autonomous underwater vehicles and worked on the interaction control Chapter. I would like to acknowledge Nilanjan Sarkar and Junku Yuh, my guests during the staying.

For this Third Edition several colleagues provide me with their illustrative material, I would like to thank Massimo Caccia, Giuseppe Casalino, Wan Kyun Chung, Jonghui Han, Tim McLain, Antonio Pascoal, Pere Ridao, Pedro Sanz, Hanumant Singh, Daniel Stilwell, Gianmarco Veruggio and Junku Yuh.

My mother, my father, Marco, Fabrizio, Giustina, Andrea and Ettore, they all tolerated my *engineeringness.*

Contents

Notations

AUV	Autonomous Underwater Vehicle
CLIK	Closed Loop Inverse Kinematics
DH	Denavit-Hartenberg
DOF	Degree Of Freedom
EKF	Extended Kalman Filter
FD	Fault Detection
FIS	Fuzzy Inference System
FTC	Fault Tolerant Controller
IK	Inverse Kinematics
KF	Kalman Filter
PID	Proportional Integral Derivative
ROV	Remotely Operated Vehicle
TCM	Thruster Control Matrix
UUV	Unmanned Underwater Vehicle
UVMS	Underwater Vehicle-Manipulator System
$\sum_{i,} O - \boldsymbol{xyz}$	Inertial frame (see Fig. 2.1)
$\sum_{b,} O_b - \boldsymbol{x}_b\boldsymbol{y}_b\boldsymbol{z}_b$	Body(vehicle)-fixed frame (see Fig. 2.1)
$\mathbb{R}, \mathbb{N}$	Real, Natural numbers
$\boldsymbol{\eta}_1 = [x\,y\,z]^{\mathrm{T}} \in \mathbb{R}^3$	Body(vehicle) position coordinates in the inertial frame (see Fig. 2.1)
$\boldsymbol{\eta}_2 = [\phi\,\theta\,\psi]^{\mathrm{T}} \in \mathbb{R}^3$	Body(vehicle) Euler-angle coordinates in the inertial frame (see Fig. 2.1)
$\mathcal{Q} = \{\varepsilon \in \mathbb{R}^3, \eta \in \mathbb{R}\}$	Quaternion expressing the body(vehicle) orientation with respect to the inertial frame defined in Sect. 2.2.2
$\boldsymbol{\eta} = [\boldsymbol{\eta}_1^{\mathrm{T}}\,\boldsymbol{\eta}_2^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$	Body(vehicle) position/orientation defined in Eq. (2.17)
$\boldsymbol{\eta}_q = [\boldsymbol{\eta}_1^{\mathrm{T}}\,\varepsilon^{\mathrm{T}}\,\eta]^{\mathrm{T}} \in \mathbb{R}^7$	Body(vehicle) position/orientation with the orientation expressed by quaternions defined in Eq. (2.22)

$\boldsymbol{\nu}_1 = [u\, v\, w]^{\mathrm{T}} \in \mathbb{R}^3$ Vector representing the linear velocity of the origin of the body(vehicle)-fixed frame with respect to the origin of the inertial frame expressed in the body(vehicle)-fixed frame (see Fig. 2.1)

$\boldsymbol{\nu}_2 = [p\, q\, r]^{\mathrm{T}} \in \mathbb{R}^3$ Vector representing the angular velocity of the body(vehicle)-fixed frame with respect to the inertial frame expressed in the body(vehicle)-fixed frame (see Fig. 2.1)

$\boldsymbol{\nu} = \left[\boldsymbol{\nu}_1^{\mathrm{T}}\, \boldsymbol{\nu}_2^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^6$ Vector representing the linear/angular velocity in the body(vehicle)-fixed frame defined in Eq. (2.18)

$R_{\alpha}^{\beta} \in \mathbb{R}^{3\times 3}$ Rotation matrix expressing the transformation from frame α to frame β

$J_{k,o}(\boldsymbol{\eta_2}) \in \mathbb{R}^{3\times 3}$ Jacobian matrix defined in Eq. (2.2)

$J_{k,oq}(\mathcal{Q}) \in \mathbb{R}^{4\times 3}$ Jacobian matrix defined in Eq. (2.10)

$J_e(\boldsymbol{\eta_2}) \in \mathbb{R}^{6\times 6}$ Jacobian matrix defined in Eq. (2.19)

$J_{e,q}(\mathcal{Q}) \in \mathbb{R}^{7\times 6}$ Jacobian matrix defined in Eq. (2.23)

$\tau_1 = [X\, Y\, Z]^{\mathrm{T}} \in \mathbb{R}^3$ Vector representing the resultant forces acting on the rigid body(vehicle) expressed in the body(vehicle)-fixed frame defined in Eq. (2.38)

$\tau_2 = [K\, M\, N]^{\mathrm{T}} \in \mathbb{R}^3$ Vector representing the resultant moment acting on the rigid body(vehicle) expressed in the body(vehicle)-fixed frame to the pole O_b defined in Eq. (2.39)

$\tau_v = \left[\tau_1^{\mathrm{T}}\, \tau_2^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^6$ Generalized forces: forces and moments acting on the vehicle defined in Eq. (2.42)

$\tau_v^{\star} \in \mathbb{R}^6$ Generalized forces in the earth-fixed-frame-based model defined in (2.56)

n Degrees of freedom of the manipulator

$q \in \mathbb{R}^n$ Joint positions defined in Sect. 2.9

$\boldsymbol{\eta}_{ee1} = [x_E\, y_E\, z_E]^{\mathrm{T}} \in \mathbb{R}^3$ Position of the end effector in the inertial frame defined in Eq. (2.61) (denoted with $x = [x_E\, y_E\, z_E]^{\mathrm{T}}$ in the interaction control sections)

$\boldsymbol{\eta}_{ee2} = [\phi_E\, \theta_E\, \psi_E]^{\mathrm{T}} \in \mathbb{R}^3$ Orientation of the end effector in the inertial frame expressed by Euler angles defined in Eq. (2.61)

$\boldsymbol{\zeta} = \left[\boldsymbol{\nu}_1^{T}\, \boldsymbol{\nu}_2^{\mathrm{T}}\, \dot{q}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{6+n}$ *System* velocity defined in Eq. (2.62)

$\boldsymbol{\nu}_{ee1} \in \mathbb{R}^3$ End-effector linear velocity with respect to the inertial frame expressed in the end-effector frame defined in Eq. (2.69)

$\boldsymbol{\nu}_{ee2} \in \mathbb{R}^3$ End-effector angular velocity with respect to the inertial frame expressed in the end-effector frame defined in Eq. (2.70)

$J_k\left(R_B^I\right) \in \mathbb{R}^{(6+n)\times(6+n)}$ Jacobian matrix defined in Eq. (2.63)

$J\left(R_B^I, q\right) \in \mathbb{R}^{6\times(6+n)}$ Jacobian matrix defined in Eq. (2.73)

$\tau_q \in \mathbb{R}^n$ Joint torques defined in Eq. (2.75)

$\tau = \left[\tau_v^{\mathrm{T}}\ \tau_{\boldsymbol{q}}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{6+n}$	Generalized forces: vehicle forces and moments and joint torques defined in Eq. (2.76)
$\boldsymbol{u} \in \mathbb{R}^p$	Control inputs, $\tau = Bu$ (see (2.78))
$\boldsymbol{\Phi} \in R^{(6+n)\times n_\theta}$	UVMS regressor defined in Eq. (2.79)
$\boldsymbol{\theta} \in R^{n_\theta}$	Vector of the dynamic parameters of the UVMS regressor defined in Eq. (2.79)
$\boldsymbol{\Phi}_v \in R^{6\times n_{\theta,v}}$	Vehicle regressor defined in Eq. (2.57)
$\boldsymbol{\theta}_v \in R^{n_{\theta,v}}$	Vector of the dynamic parameters of the vehicle regressor defined in (2.57)
$\boldsymbol{h}_i^i = \left[\boldsymbol{f}_i^{i\mathrm{T}}\ \boldsymbol{\mu}_i^{i\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^6$	Forces and moments exerted by body $i-1$ on body i (see Fig. 2.7)
$\boldsymbol{h}_e = \left[\boldsymbol{f}_e^{\mathrm{T}}\ \boldsymbol{\mu}_e^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^6$	Forces and moments at the end effector (see Fig. 2.8)
$t \in \mathbb{R}$	Time
$\lambda_{\min(\max)}(X)$	Smallest(largest) eigenvalue of matrix $\boldsymbol{X}$
$\mathrm{diag}\{x_1, \ldots, x_n\}$	Diagonal matrix filled with x_i in the i row, i column and zero in any other place
$\mathrm{blockdiag}\{X_1, \ldots, X_n\}$	Block diagonal matrix filled with matrices $X_1, \ldots, X_n$ in the main diagonal and zero in any other place
$\mathcal{R}(X)$	Range of matrix $\boldsymbol{X}$
$\dot{x}$	Time derivative of the variable x
$\lVert x \rVert$	2-norm of the vector $\boldsymbol{x}$
$\hat{x}(\hat{X})$	Estimate of the vector $\boldsymbol{x}$ (matrix $\boldsymbol{X}$)
x_d	Desired value of the variable x
$\tilde{x}$	Error variable defined as $\tilde{x} = x_d - x$
$x^{\mathrm{T}}(X^{\mathrm{T}})$	Transpose of the vector $\boldsymbol{x}$ (matrix $\boldsymbol{X}$)
x_i	i th element of the vector $\boldsymbol{x}$
$X_{i,j}$	Element at row i, column j of the matrix $\boldsymbol{X}$
$X^{\dagger}$	Moore–Penrose inversion (pseudoinversion) of matrix $\boldsymbol{X}$ If $\boldsymbol{X}$ is low rectangular it is $X^{\dagger} = X^{\mathrm{T}}(XX^{\mathrm{T}})^{-1}$ If X is high rectangular it is $X^{\dagger} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$
$\boldsymbol{I}_r$	$(r \times r)$ identity matrix
$\boldsymbol{O}_{r_1\times r_2}$	$(r_1 \times r_2)$ null matrix
$\boldsymbol{S}(\cdot) \in \mathbb{R}^{3\times 3}$	Matrix performing the cross product between two (3×1) vectors defined in (2.6)
ρ^3	Water density
μ	Fluid dynamic viscosity
R_n	Reynolds number
$\boldsymbol{g}^I$	Gravity acceleration expressed in the inertial frame

Chapter 1
Introduction

It is during the fifteenth century that the systematic design of underwater vehicles starts engaging the scientists of the era. Figure 1.1 reports a draw by Roberto Valturio, an Italian historian lived from 1405 to 1475.

One of the most famous scientists devoting his efforts in designing an underwater vehicle is Leonardo Da Vinci. A draw has been found in the *Codice Atlantico* (Codex Atlanticus), written between 1480 and 1518, together with the development of some diver's devices (see Figs. 1.2 and 1.3 where the corresponding page of the Codex is reported).

Legends say that Leonardo worked on the idea of an underwater military machine and that he further destroyed by himself the results judged too dangerous. In Fig. 1.4 the graphical reconstruction of a vehicle with floating capabilities to hide part of it.

Maybe the first idea of an underwater machine is from Aristotle; following the legend he built a machine: *skaphe andros* (boat-man) that allowed Alexander the Great (Alexander III of Macedon, 356–323 B.C.) to stay in deep for at least half a day during the war of Tiro in 325 B.C. This is unrealistic, of course, also considering that the Archimedes's law was still to become a reality (around 250 B.C.).

In August, the 4th, 2005, in the Pacific sea, in front of the Kamchatka, at a depth of 200 m, a Russian manned submarine, the *AS-28*, got stacked into the cables of a underwater radar; at that moment, seven men were in the vehicle. One day later a British Remotely Operated Vehicle (ROV), *Scorpio*, was there and, after another day of operations, it was possible to cut the cables thus allowing the submarine to surface safely. In addition than exceptional operations like the one mentioned, underwater robots can be used to accomplish missions such as sea bottom and pipeline survey, cable maintenance, off-shore structures' monitoring and maintenance, collect/release of biological surveys. Currently, most of the operations mentioned above are achieved via manned underwater vehicles or remotely operated vehicles; in case of manipulation tasks, moreover, those are performed resorting to remotely operated master-slave systems. The strong limit of the use of manned vehicles is the enormous cost and risk in working in such an hostile environment; the daily operating cost is larger than 8,000 € [1]. In may, 2009, the Air France flight 447 flying between

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_1,

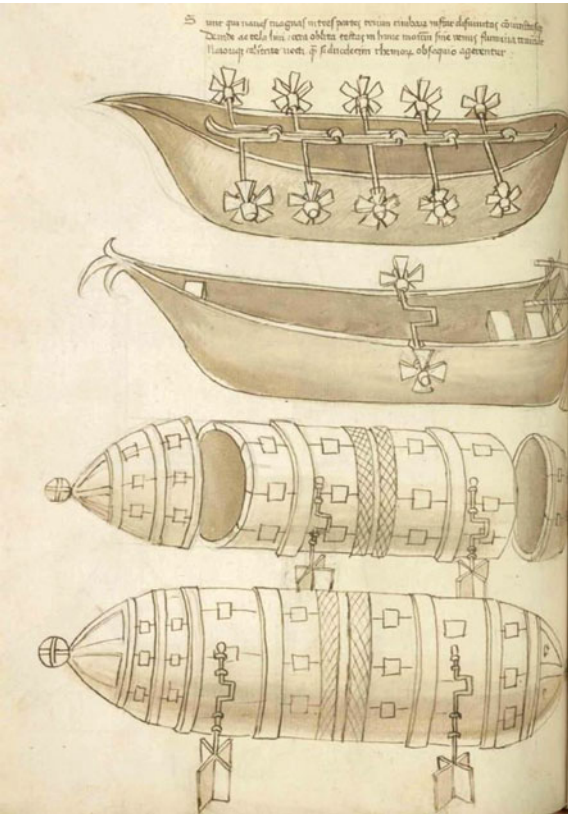


Fig. 1.1 Underwater vehicle drawn by Roberto Valturio, fifteenth century

Paris and Rio de Janeiro crashed causing the killing of all the more than 200 people onboard. The costs to recover the black box, achieved resorting both to AUVs and ROVs, is not official but estimated in several millions of euros. The aim of the research is to progressively make it possible to perform such missions in a completely autonomous way.

This objective is challenging from the technological as well as from the theoretical aspects since it implies a wide range of technical and research topics. Sending an autonomous vehicle in an unknown and unstructured environment, with limited on-line communication, requires some on board *intelligence* and the ability of the vehicle to react in a reliable way to unexpected situations. The sensory system of the vehicle must deal with a noisy and unstructured environment; moreover, technologies as GNSS (Global Navigation Satellite System) are not applicable due to the impossibility to underwater electromagnetic transmission at GNSS specific frequencies; vision based systems are not fully reliable due to the generally poor visibility. The actuating system is usually composed of thrusters and control surfaces, both of them have a non-linear dynamics and are strongly affected by the hydrodynamic effects.

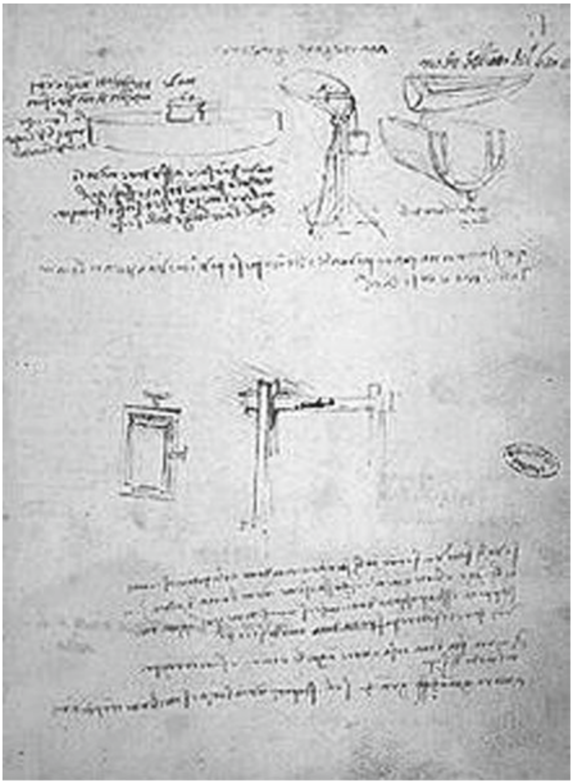


Fig. 1.2 Page of the *Codice Atlantico* (around 1500) containing the draw of the manned underwater vehicle developed by Leonardo Da Vinci

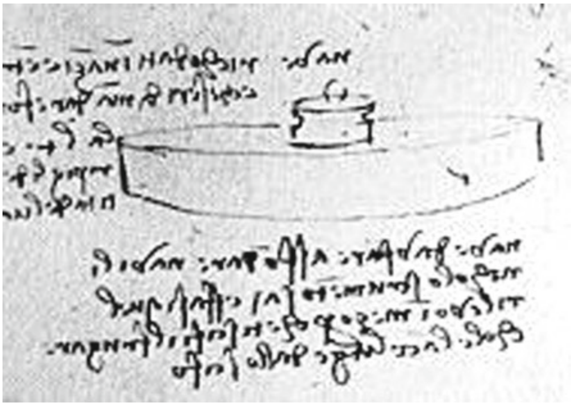


Fig. 1.3 Particular of the page of the *Codice Atlantico* containing the draw of the manned underwater vehicle developed by Leonardo Da Vinci

The book of Fossen [2] is one of the first books dedicated to control problems of marine systems, the case of surface vehicles, in fact, is also taken into account. The same author presents, in [3], an updated and extended version of the topics developed in the first book and in [4], an handbook on marine craft hydrodynamics and control.

Fig. 1.4 Graphical reconstruction of a vehicle with floating capabilities to hide part of it, original from Leonardo Da Vinci

Some very interesting talks about state of the art and direction of the underwater robotics were discussed by, e.g., Yuh in [5, 6], Yuh and West in [7], Ura in [8]. At the best of our knowledge this is the sole book dedicated to control problems of underwater robotic systems with particular regard with respect to the manipulation; this is an emerging topic in which experimental results are in their embryonic phase.

In this chapter an overview of control problem in underwater robotics is presented; some of these aspects will be further analyzed along this book.

1.1 Underwater Vehicles

The therm Remotely Operated Vehicle (ROV) denotes an underwater vehicle physically linked, via the tether, to an operator that can be on a submarine or on a surface ship. The tether is in charge of giving power to the vehicle as well as closing the manned control loop. Autonomous Underwater Vehicles (AUVs), on the other side, are supposed to be completely autonomous, thus relying to onboard power system and *intelligence*. These two types of underwater vehicles share some control problems, in this case one has to refer to them as Unmanned Underwater Vehicles (UUVs). In case of missions that require interaction with the environment, the vehicle can be equipped with one or more manipulators; in this case the system is usually called Underwater Vehicle-Manipulator System (UVMS).

Currently, there are about more than 100 prototypes in the laboratories all over the world, see e.g., [7]. Among the others: *r2D4* developed at URA laboratory of the University of Tokyo (Tokyo, Japan), *ABE* of the Deep Submerge Laboratory of the Woods Hole Oceanographic Institution (Massachusetts, USA), *Odissey IId* belonging to the AUV Laboratory of the Massachusetts Institute of Technology (Massachusetts, USA), *ODIN III* designed at the Autonomous Systems Laboratory of the University of Hawaii (Hawaii, USA), *Phoenix* and *ARIES*, torpedo-like vehicles developed at the Naval Postgraduate School (California, USA), *Girona500* belonging to the University of Girona (Girona, Spain).

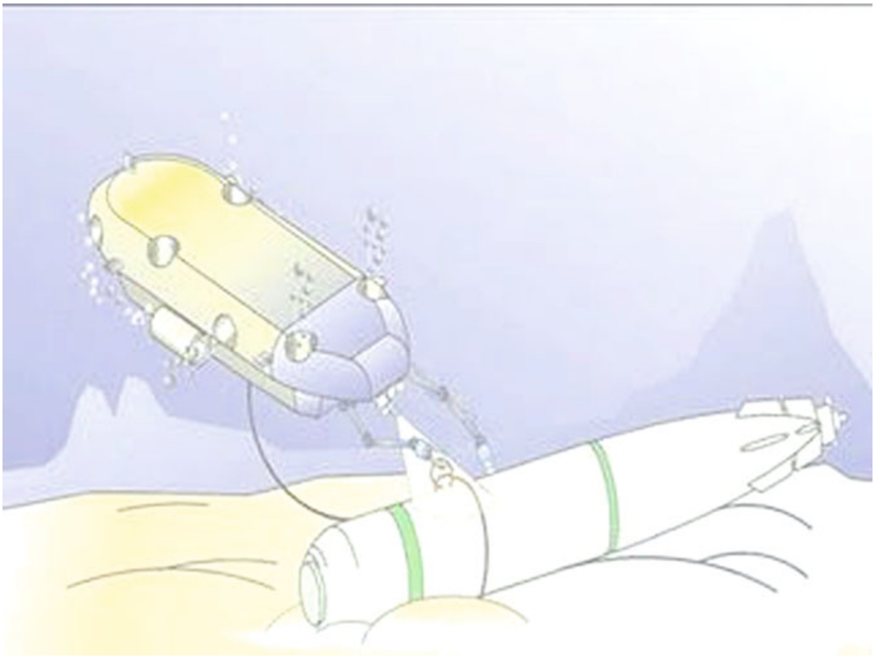


Fig. 1.5 Sketch of the underwater vehicle-manipulator system SAUVIM, currently under development at the Autonomous Systems Laboratory of the University of Hawaii (courtesy of J. Yuh)

Currently, a few but increasing number of companies sell AUVs; among the others: Bluefin Corporations developed, in collaboration with MIT, different AUVs, such as Bluefin 21, for deep operations up to 4,500 m; C&C technologies designed Hugin 3000, able to run autonomously for up to 50 h; the Canadian ISE Research Ltd developed several AUVs such as, e.g, Explorer or Theseus; Hafmynd, in Iceland, designed a very small AUV named Gavia; the Danish Maridan developed the Maridan 600 vehicle. Reference [9] provides the perspective of Chevron on the long-term goal for AUV development with a possible metrics on achieved results in specific missions and a following road-map.

The UVMSs are still under development; several laboratories built some manipulation devices on underwater structures but few of them can be considered as capable of autonomous manipulation. *SAUVIM* (see Fig. 1.5), a semi-autonomous vehicle with an *Ansaldo* 7-link manipulator has been developed at the Autonomous Systems Laboratory of the University of Hawaii [10]; this vehicle, in the final version, will be able to operate at the depth of 4,000 m.

AMADEUS, an acronym for Advanced MAnipulation for DEep Underwater Sampling, funded by the European Commission, that involved the Heriot-Watt University (UK), the Università di Genova (Italy), the National Research Council-ISSIA (Italy), the Universitat de Barcelona (Spain), the Institute of Marine Biology of Crete (Greece). The project focused on the coordinated control of two tele-operated underwater *Ansaldo* 7-link manipulators and the development of an underwater hand equipped with a slip sensor; Fig. 1.6 shows a wet test in a pool.

The French company Cybernétix (http://www.cybernetix.fr) sells hydraulic manipulators mounted on ROVs that can be remotely operated by means of a joystick or in a master-slave configuration. Figure 1.7 shows the draw of Jaguar, a proof of concept UVMS developed and the Woods Hole Oceanographic Institution.

Figure 1.8 reports the Girona500 I-AUV [11], this UVMS was built during *TRIDENT* [12, 13], an European project with underwater intervention goals involving the University Jaume I (Spain), the University of Girona (Spain), the University of Baleares (Spain), the University of Bologna (Italy), the University of Genova (Italy), the Technical University of Lisbon (Portugal), the Heriot-Watt University (Scotland) and the Italian SME Graaltech. The final UVMS built during the project is shown in Fig. 1.9 [14].

It is also of interest to mention a new concept design of underwater manipulation. To minimize the control problems given by the physical interaction between the vehicle and the manipulator, in [15] the proposal to have a small ROV connected to the AUV is made. A mock-up system has also been designed.

1.2 Sensorial Systems

The AUVs need to operate in an unstructured hazardous environment; one of the major problems with underwater robotics is in the localization task due to the absence of a single, proprioceptive sensor that measures the vehicle position and the impossibility to use the GNSS under the water. The use of redundant multi-sensor system, thus, is common in order to perform sensor fusion tasks and give fault detection and tolerance capabilities to the vehicle.

To give an idea of the sensors used in underwater robotics, the following lists some the main:

- Compass. A gyrocompass can provide an estimate of geodetic north accurate to a fraction of a degree. Magnetic compasses can provide estimates of magnetic north with an accuracy of less than 1°. Tables or models can be used to convert betweeen the two;
- Gyroscope. The term gyroscope denotes any instrument measuring inertial angular rotation;
- Inertial measurement unit (IMU). An IMU provides information about the vehicle's linear acceleration and angular velocity. These measurements are combined to form estimates of the vehicle's attitude including an estimate of geodetic (true) north from the most complex units;
- Depth sensor. Measuring the water pressure gives the vehicle's depth. These estimates are generally reliable and accurate;
- Altitude and forward-looking sonar. These are used to detect the presence of obstacles and distance from the seafloor.

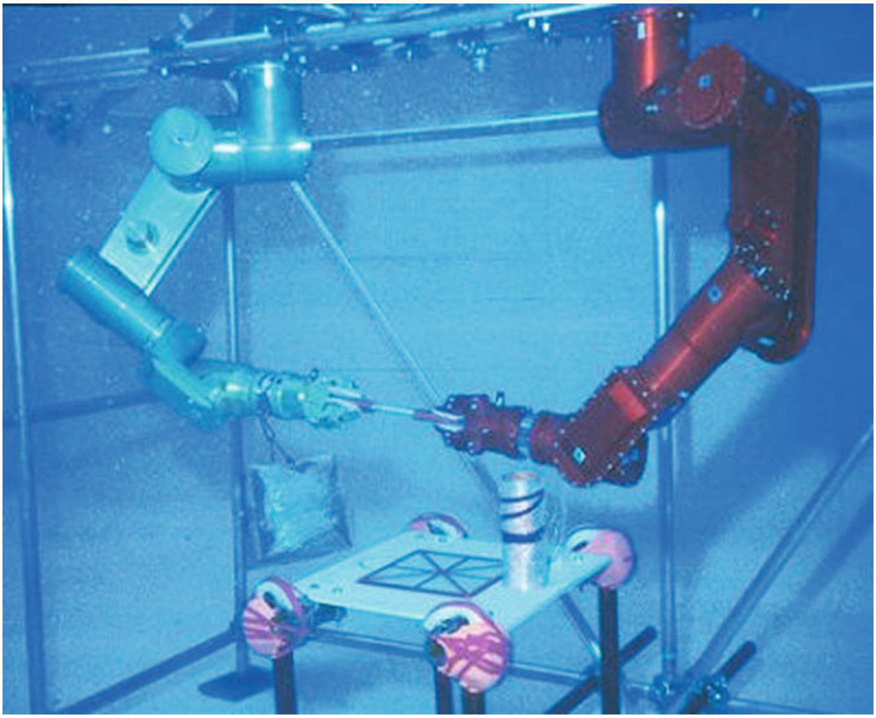
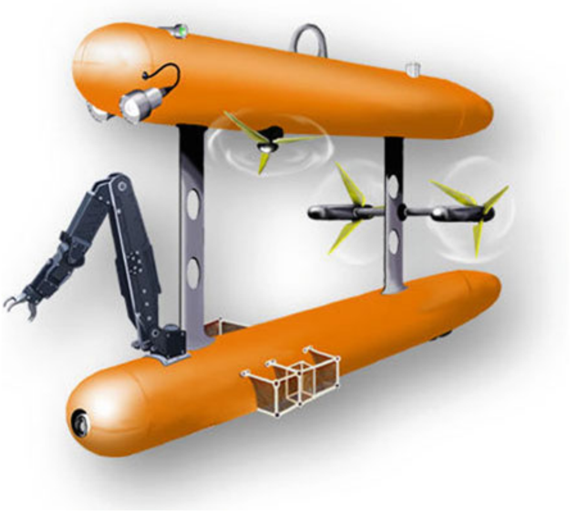


Fig. 1.6 Coordinated control of two seven-link Ansaldo manipulators during a wet test in a pool (courtesy of G. Casalino, Genoa Robotics and Automation Laboratory, Università di Genova and G. Veruggio, National Research Council-ISSIA, Italy)

Fig. 1.7 Proof of concept for a Woods Hole Oceanographic Institution vehicle equipped with a manipulator, named Jaguar (courtesy of H. Singh)



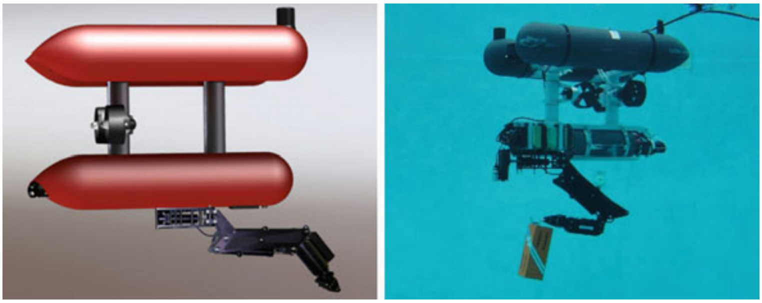


Fig. 1.8 Girona500 equipped with a CSIP arm developed at the University of Girona (Courtesy of Pere Ridao, Centre d'Investigaciò en Robótica Submarina (CIRS), Computer Vision and Robotics Research Team (VICOROB), University of Girona)

Table 1.1 JHUROV instrumentations

Measured variable	Sensor	Precision	Update rate (Hz)
3DOF-vehicle position	SHARP acoustic transponder	0.5 cm	10
Depth	Foxboro/ICT model n. 15	2.5 cm	20
Heading	Litton LN200 IMU Gyro	0.01°	20
Roll and pitch	KVH ADGC	0.1°	10
Heading	KVH ADGC	1°	10

Table 1.2 ODIN III sensors update

Measured variable	Sensor	Update rate (Hz)
xy vehicle position	8 sonars	3
Depth	Pressure sensor	30
Roll, pitch and yaw	IMU	30

- Doppler velocity log (DVL). By processing reflected acoustic energy from the seafloor and the water column from three or more beams, estimates of vehicle velocity relative to the seafloor and relative water motion can be obtained.
- Global Navigation Satellyte System (GNSS). This is used to localize the vehicle while on the surface;
- Acoustic positioning. A variety of schemes exist for determining vehicle position using acoustics (see Sect. 1.4);
- Vision systems. Cameras can be used to obtain estimates of relative, and in some cases absolute, motion and used to perform tasks such as visual tracking of pipelines, station keeping, visual servoing or image mosaicking.

As an example, Table 1.1 reports some data of the instrumentations of the ROV developed at the John Hopkins University [16] and Table 1.2 some data of the AUV ODIN III [1, 17].

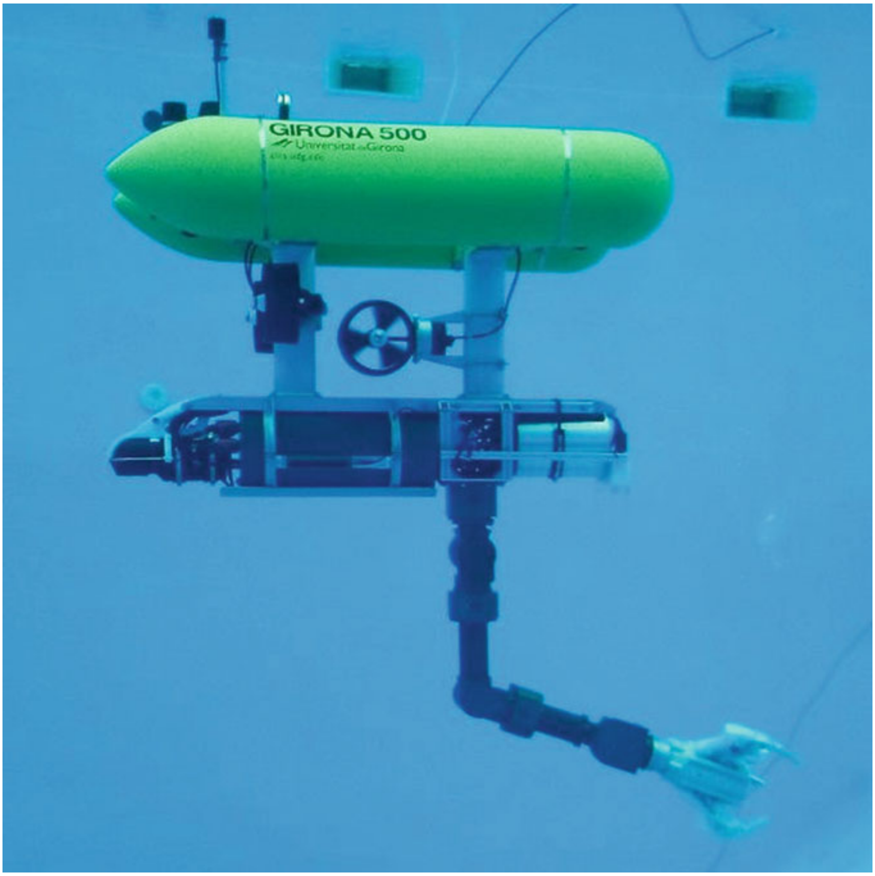


Fig. 1.9 Final UVMS developed during the TRIDENT project (courtesy of Pere Ridao, Pino Casalino and Pedro Sanz)

Reference [18] shows some data fusion results with a redundant sensorial system mounted on the AUV Oberon, while Ref. [19] reviews advances in navigation technology.

1.3 Actuation

Underwater vehicles are usually controlled by thrusters and/or control surfaces. Control surfaces, such as rudders and sterns, are common in cruise vehicles; those are torpedo-shaped and usually used in cable/pipeline inspection. The main configuration is not changed in the last century, there is a main thruster and at least one rudder and one stern, in Fig. 1.10 it is reported the underwater manned vehicle named SLC (*Siluro a Lenta Corsa*, Slow Running Torpedo), or *maiale* (pig), used in

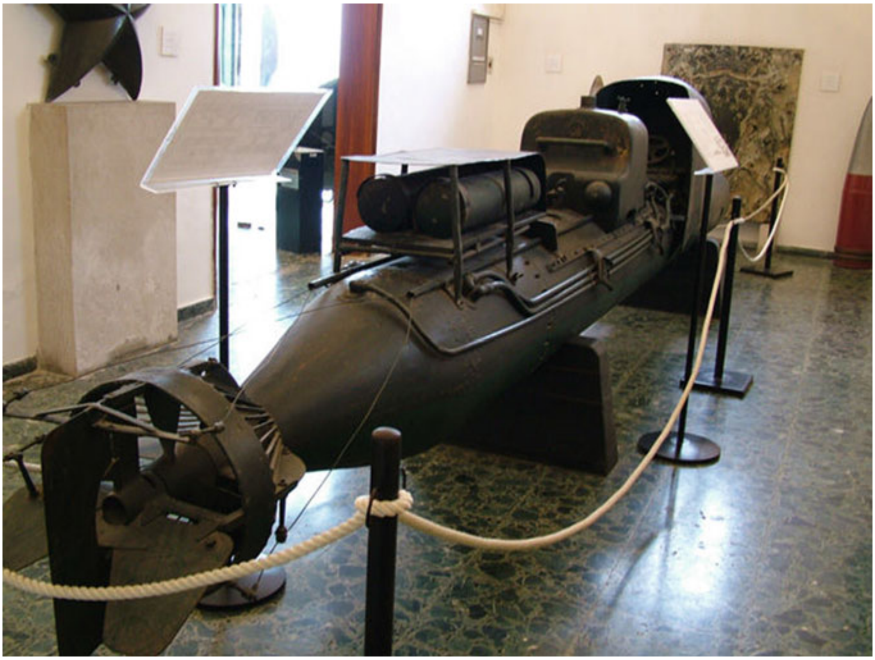


Fig. 1.10 *Pig*, manned vehicle used by the Royal Italian Navy during the Second World War in the Mediterranean Sea. The thruster and the group rudder/stern can be observed in the bottom left angle of the photo

the second world war by the *Regia Marina Italiana* (Royal Italian Navy). Since the force/moment provided by the control surfaces is function of the velocity and it is null in hovering, they are not useful to manipulation missions in which, due to the manipulator interaction, full control of the vehicle is required.

The relationship between the force/moment acting on the vehicle and the control input of the thrusters is highly nonlinear. It is function of some structural variables such as: the density of the water; the tunnel cross-sectional area; the tunnel length; the volumetric flowrate between input-output of the thrusters and the propeller diameter. The state of the dynamic system describing the thrusters is constituted by the propeller revolution, the speed of the fluid going into the propeller and the input torque.

A detailed theoretical and experimental analysis of thrusters' behavior can be found in [20–25]. In [3] a chapter is dedicated to modelling and control of marine thrusters. Roughly speaking, thrusters are the main cause of limit cycle in vehicle positioning and bandwidth constraint. In [26] the thruster model is explicitly taken into account in the control law. Reference [27] presents experimental results on the performance of model-based control law for AUVs in presence of model mismatching and thrusters' saturation.

1.4 Localization

The position and attitude of a free floating vehicle is not measurable by the use of a single, internal sensor. This poses the problem of estimating the AUV's position. As detailed above, several sensors are normally mounted on an AUV in order to implement sensor fusion algorithms and obtain an estimation more reliable than by using a single sensor.

A possible approach for AUV localization is to rely on dead reckoning techniques, i.e., the estimation of the position by properly merging and integrating measurements obtained with inertial and velocity sensors. Dead reckoning suffers from numerical drift due to the integration of sensor noise, as well as sensor bias and drift, and may be prone to the presence of external currents and model uncertainties. The variance of the estimation error grows with the distance travelled, dead reackoning, thus, it is used for short dives.

Several approaches are based on trilateration-based algorithms, however, localization devices such as Global Navigation Satellite Systems (GNSSs) are ineffective underwater due to the attenuation of electromagnetic radiations. Most of the vehicles are equipped as well with GNSSs devices to reset the error to zero during temporary surfaces.

By resorting to the same concept of trilateration, but changing technology, under the water acoustic signals are commonly used. Among the commercially available solutions, Long, Short, and Ultra-Short Baseline systems have found widespread use. In Long BaseLine (LBL) acoustic localization systems, a set of transponders is installed at fixed, known underwater positions. While submerged, the AUV computes its distance to each of the transponders by interrogating them and measuring the round-trip travel time of the acoustic waves emitted. In Short BaseLine (SBL) systems, three or more transducers are mounted on a surface vessel, at relative distances on the order of tens of meters. In this case, it is up to the surface segment to request replies from a transponder installed on-board the submerged AUV. In Ultra-short BaseLine (USBL) systems, a set of transducers is assembled in a compact stand-alone device installed on board a support ship; the estimation of the AUV transponder position is done by measuring the relative phases among the signals arriving at the transducers in response to queries by the surface segment. In the latter two systems, there is a need to transmit back to the AUV its estimated position. Acoustic underwater positioning is commercially mature and several companies offer a variety of products such as, Kongsberg, EvoLogics, Tritech and other, see, e.g., the USBL in Fig. 1.11.

Acoustic devices have been recently used also to estimate the relative positions between two AUVs by resorting to dead-reckoning, depth and distance measurements. This localization problem, commonly known in the literature as *single beacon localization* or *range-only localization*, has received increasing attention in the very last years as pointed out by a number of recently available papers of which [28–31] are representative examples.

In [32] the concept of exchanging mutual information between one moving vehicle and one fixed sea floor station is introduced. In few words, the vehicle, by resorting

Fig. 1.11 Underwater acoustic USBL system (courtesy of Tritech International Ltd)

Fig. 1.12 USBL Underwater Acoustic USBL System, device for tracking and communication in shallow waters, providing data transfer rates nomially up to 31.2 kbit/s over a 1,000 m range (courtesy of EvoLogics GmbH)

to acustical devices able to measure its relative position with respect to the station, further *fused* with the relative bearing received from the station, is able to compute its position without expensive INS or time-consuming calibration methods.

Additional sensors, such as, e.g., video-cameras may be used to obtain relative measurements and further fuse in a filter designed for state estimate of the dynamic system.

A recent survey on techniques for underwater acoustic sensor networks, sharing most of the mathematics with AUV localization, is provided in [33].

1.4.1 Communication Systems

Communication of the vehicle with the surface or with other vehicles is mandatory even for *autonomous* missions.

The *fuzzy* propagation of electromagnetic radiations prevent the use of them both for positioning and communication, the more diffused communication technology is based on acustic propagation. The performance of acustic modems are not as efficient as their aerial conuterparts. Three main factors affect the acustic propagation: a low speed of sound ($\approx 1{,}500$ m/s), the presence of time-varying multiple paths, an attenuation increasing with the frequency. Finally, the channel capacity is function of the distance and is limited [34]. For such a challenging communication medium, efficient communication protocols need to be properly designed [35].

The market is currently offering devices that merge communication and localization, based on USBL, capabilities for short distances and shall water environments such as the intrusment shown in Fig. 1.12, manufactured by EvoLogics.

1.5 AUVs' Control

Control of AUVs' is challenging, in fact, even though this problem is kinematically similar to the widely studied one of controlling a free-floating rigid body in a six-dimensional space, the underwater environment makes the dynamics to be faced quite different. An overview of the main control techniques for AUVs can be found, e.g., in [2, 3], experimental comparison in [36].

A main difference in control of underwater vehicles is related to the type of actuation; cruise vehicles, in fact, are usually actuated by means of one thruster and several control surfaces; they are under-actuated and mainly controlled in the surge, sway and heave directions. On the other hand, if a vehicle is conceived for manipulation tasks it is required that it is actuated in all the DOFs even at very low velocities; 6 or more thrusters are then designed.

An example of cruise vehicle is ARIEL, belonging to the Naval Postgraduate School; a detailed description and its command and control subsystems is provided in [37]. In [38], the control system of the NPS AUV II is given together with experimental results. Control laws for cruise vehicles are usually designed at a nominal velocity since the vehicle is designed for exploration or cable tracking missions, see, e.g., [39] for a pipeline tracking with Twin-Burger 2. The homing operation needs specific algorithms, [40] presents experimental results performed with the vehicle Odyssey IIb, of an homing system based on an electromagnetic guidance rather than an acoustic signal. Topic also address in [41] or [42].

Research efforts have been devoted at controlling fully actuated underwater vehicle, in particular at very low velocity or performing a station keeping task. This topic will be discussed in Chap. 3, some experimental results is given in Chap. 5.

Identification of the dynamic parameters of underwater robotic structures is a very challenging task due to the model characteristics, i.e., non-linear and coupled dynamics, difficulty in obtaining effective data; the interested reader can refer to [16, 43–48].

An overview of control architectures, specifically designed for AUVs is provided in [49]. Recent examples are [11, 14, 50].

1.5.1 Multiple Autonomous Underwater Vehicles

An emerging topic is also constituted by control of platoon of marine vehicles, both surface and underwater. Control of multiple robots is a huge topic, receiving increasing attention in the last years in all domains of robotics. An introduction to the more general topic of interconnected dynamic systems is represented by [51–56].

Designing of control algorithms for platoon of AUVs can not neglect the characteristics briefly listed in this Introduction, namely the nonholonomy, the hovering difficulties for underactuated models, the communication and localization issues.

The use of several AUVs to achieve fulfillment of a task might be of benefit in several situations: explorations of areas, de-mining [57], as in un the first Gulf's war, or interaction with the environment in which the team of AUVs may, e.g., push an object that one single AUV would not have the power to move.

Considering surface marine vehicles, in [58] proposes a leader-follower formation control algorithm; experiments with one scale vehicle following a simulated leader are provided. Reference [59] presents a naval minesweeper platoon in which a supervisor vehicle is in charge of tasking in real-time the remaining vehicles.

Reference [60–62] reports interesting experimental results on the use of an underwater glider fleet for adaptive ocean sampling. The sampling, thus, is not achieved by resorting to deterministic, pre-programmed, paths but is made adaptive in order to extract the maximum possible information during one mission. The formation control is obtained using virtual bodies and artificial potential techniques [63]; the Authors separate the problem into small area coverage (5 km) and synoptic area coverage for larger scale (5–100 km). For the former, experimental result performed with 3 gliders in 2003 are reported; the experiments were successful thus demonstrating the reliability of the approach in a real environment under the effect, e.g, of the ocean current.

In [64, 65], an effective decentralized control technique for platoons is proposed and simulated for underwater vehicles. Remarkably, the approach requires a limited amount of inter-vehicle communication that is independent from the platoon dimension; moreover, control of the platoon formation is achieved through definition of a suitable (global) task function without requiring the assignment of desired motion trajectories to the single vehicles. In Fig. 1.13, one of the AUVs developed at the Virginia Tech is shown, these vehicles will be very small in size and cheap with most of the components custom-engineered [66].

Reference [67] presents a behavior-based intelligent control architecture that computes discrete control actions. The control architecture separates the sensing aspect, called *perceptor* in the paper, from the control action, called *response controller* and it focuses on the latter implemented by means of a set of discrete event models. The design of a sampling mission for AUVs is also discussed in the paper.

Some interesting projects have recently been funded by the European Commission, under the FP7 funding programme. Among the others:

- MORPH: Marine robotic system of self-organizing, logically linked physical nodes (http://www.morph-project.eu);
- CO^{3}AUVs: Cooperative Cognitive Control for Autonomous Underwater Vehicles (http://www.Co3-AUVs.eu);
- GREX: Coordination and Control of Cooperating Heterogeneous Unmanned Systems in Uncertain Environments (http://www.grex-project.eu);

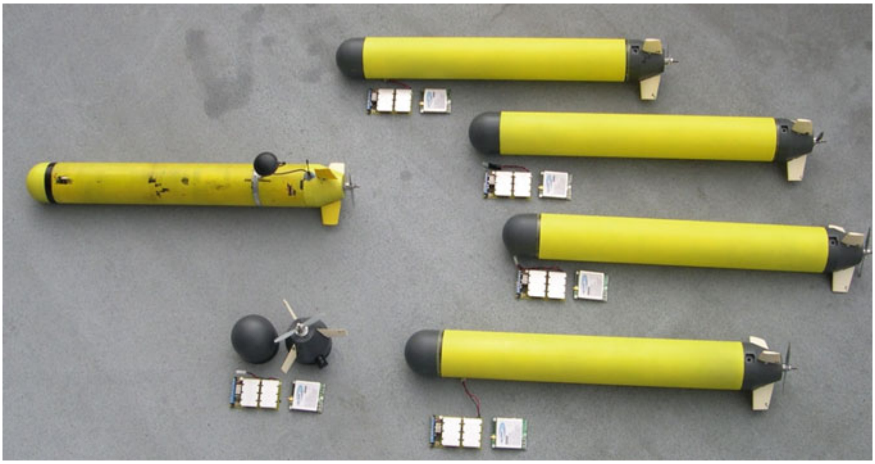


Fig. 1.13 Platoon of AUVs developed at the Virginia Tech for research purposes (courtesy of D. Stilwell)

1.5.2 Fault Detection/Tolerance for UUVs

ROVS and AUVs are complex systems engaged in missions in un-structured, unsafe environments for which the degree of autonomy becomes a crucial issue. In this sense, the capability to detect and tolerate faults is a key to successfully terminate the mission or recuperate the vehicle. An overview of fault detection and fault tolerance algorithms, specifically designed for UUVs is presented in Chap. 4.

In Fig. 1.14, the vehicle Romeo operating over thermal vents in the Milos Island, Aegean Sea, Greece, is shown; this vehicle has been built at the Robot-Lab, National Research Council (CNR-ISSIA), Genova, Italy (http://www.robotlab.ian.ge.cnr.it). This vehicle has been object of several experimental studies on fault detection/tolerance algorithms.

1.6 UVMS' Coordinated Control

The use of Autonomous Underwater Vehicles (AUVs) equipped with a manipulator (UVMS) to perform complex underwater tasks give rise to challenging control problems involving nonlinear, coupled, and high-dimensional systems. Currently, the state of the art is represented by tele-operated master/slave architectures; few research centers are equipped with autonomous systems [7, 68, 69].

In the recent years, one interesting project, named TRIDENT (Marine Robots and Dexterous Manipulation for Enabling Autonomous Underwater Multipurpose

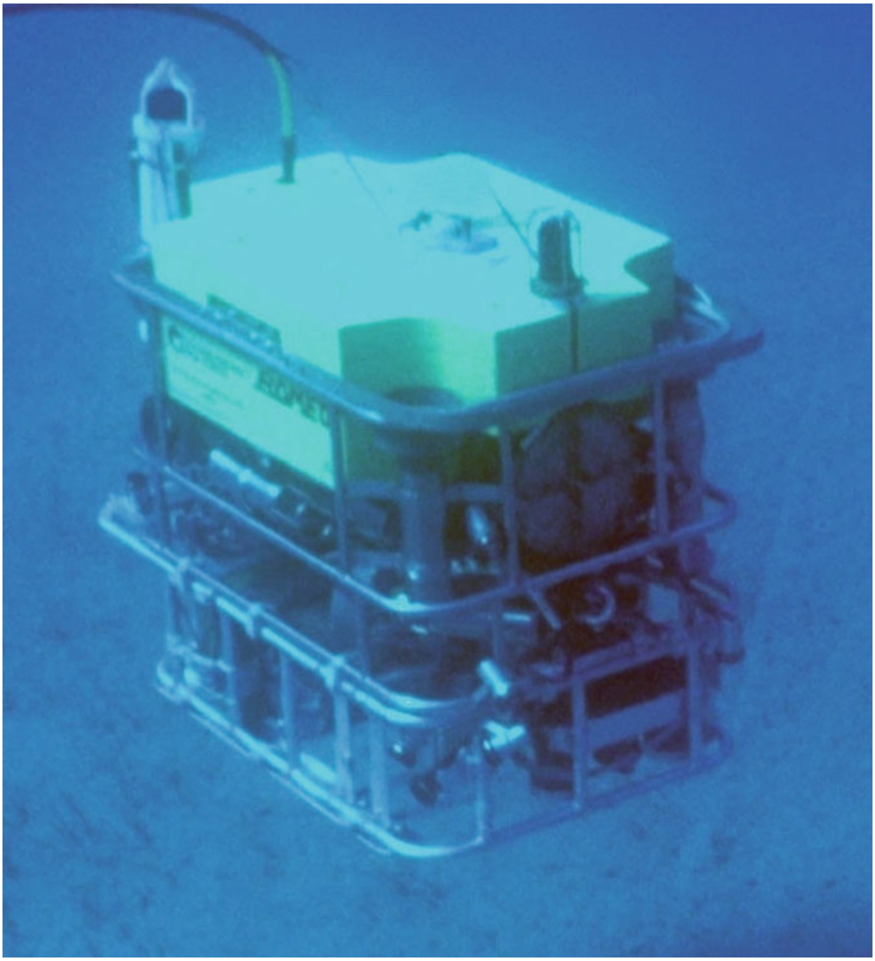


Fig. 1.14 Romeo operating over thermal vents in the Milos Island, Aegean Sea, Greece, during the final demo of the EC-funded project ARAMIS (courtesy of M. Caccia, National Research Council-ISSIA, Italy)

Intervention Missions http://www.irs.uji.es/trident) focused also on underwater manipulation [11–14, 70].

The *core* of this monograph is dedicated to this topic, in Chap. 6 the kinematic control will be discussed, Chap. 7 presents dynamic control laws for UVMSs and Chap. 8 shows some interaction control schemes.

1.7 Future Perspectives

Marine robotics is receiving further interest in the last years. To name few applications: monitoring, sampling, defence, energy production and transportation, archaeology. Missions may be implemented by resorting to single surface or underwater vehicles, gliders, vehicle equipped with manipulators or team of coordinated vehicles. For long duration or deep missions the presence of the man in the loop is discouraged for several reasons, the use of autonomous underwater robots appears to be the main road.

For single vehicles, both surface or underwater, the current technology allows to safely run long duration and distances missions. Underwater manipulation is currently performed via master-slave architecture, i.e., with the man in the loop. This needs for a high-enough communication frequence to be set-up between vehicle and operator thus preventing deep applications.

Recently, additional experimental works on UVMS has been performed within the TRIDENT project [12, 13] with a full-size vehicle and a 7-DOFs arm. The results are interesting and encouraging.

The perception aspect, not covered in this monograph, is also a very active and stimulating topic of research for the marine environment. Under the water each of the sensor we are used to on the surface suffers from one or more drawbacks. It is required to implement a proper sensor fusion strategy to localize the vehicle, the localize the target with respect to the vehicle, to correctly *contextualize* the UVMS within the operating scenario. As an example, underwater vision is affected by light absorbion, turbidity and particulate; a-priori calibration of the vision-based algorithm, thus, appear to be not possible.

Control, in all its aspects, also need to be adaptive with respect to, e.g., the current or the payload. Due to the unstructured characteristic of the underwater environment it needs to be *reactive*. UVMSs will benefit from the overall progresses of advanced robotics, a tight collaboration among the various communities working on autonomous robotics might be fruitful.

References

1. S. Zhao, J. Yuh, Experimental study on advanced underwater robot control. IEEE Trans. Robot. **21**(4), 695–703 (2005)
2. T. Fossen, *Guidance and Control of Ocean Vehicles* (Wiley, Chichester, 1994)
3. T. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles* (Marine Cybernetics, Trondheim, 2002)
4. T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control* (Wiley, Chichester, 2011)
5. J. Yuh, Development in underwater robotics, in *Proceedings of IEEE International Conference on Robotics and Automation*, vol.2, pp. 1862–1867, 1995
6. J. Yuh, Exploring the mysterious underwater world with robots, in *6th IFAC Conference on Manoeuvring and Control of Marine Craft*, Girona, 2003

7. J. Yuh, M. West, Underwater robotics. Adv. Robot. **15**(5), 609–639 (2001)
8. T. Ura, Steps to intelligent AUVs, in *6th IFAC Conference on Manoeuvring and Control of Marine Craft*, Girona, 2003
9. B. Gilmour, G. Niccum, T. O'Donnell, Field resident AUV systems-schevron's long-term goal for AUV development, in *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*, 2012, pp. 1–5
10. G. Marani, S.K. Choi, J. Yuh, Underwater autonomous manipulation for intervention missions AUVs. Ocean Eng. **36**(1), 15–23 (2009)
11. D. Ribas, N. Palomeras, P. Ridao, M. Carreras, A. Mallios, Girona 500 auv: from survey to intervention. IEEE/ASME Trans. Mechatron. **17**(1), 46–53 (2012)
12. P.J. Sanz, P. Ridao, G. Oliver, C. Melchiorri, G. Casalino, C. Silvestre, Y. Petillot, A. Turetta, TRIDENT: a framework for autonomous underwater intervention missions with dexterous manipulation capabilities, in *Proceedings of the 7th IFAC Symposium on Intelligent Autonomous Vehicles IAV-2010*, 2010
13. P.J. Sanz, P. Ridao, G. Oliver, C. Insaurralde, G. Casalino, C. Silvestre, C. Melchiorri, A. Turetta, TRIDENT: recent improvements about intervention missions, in *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV2012)*, 2012
14. M. Prats, D. Ribas, N. Palomeras, J.C. García, V. Nannen, S. Wirth, J.J. Fernández, J.P. Beltrán, R. Campos, P. Ridao et al., Reconfigurable AUV for intervention missions: a case study on underwater object recovery. Intel. Serv. Robot. **5**(1), 19–31 (2012)
15. S.C. Yu, J. Yuh, J. Kim, Armless underwater manipulation using a small deployable agent vehicle connected by a smart cable. Ocean Eng. **70**, 149–159 (2013)
16. D.A. Smallwood, L.L. Whitcomb, Adaptive identification of dynamically positioned underwater robotic vehicles. IEEE Trans. Control Syst. Technol. **11**(4), 505–515 (2003)
17. H.T. Choi, A. Hanai, S.K. Choi, J. Yuh, Development of an underwater robot, ODIN-III, in *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 1, 2003, pp. 836–841
18. S. Majumder, S. Scheding, H.F. Durrant-Whyte, Multisensor data fusion for underwater navigation. Robot. Auton. Syst. **35**(2), 97–108 (2001)
19. J.C. Kinsey, R.M. Eustice, L.L. Whitcomb, A survey of underwater vehicle navigation: recent advances and new challenges, in *IFAC Conference of Manoeuvering and Control of Marine Craft*, 2006
20. R. Bachmayer, L.L. Whitcomb, M.A. Grosenbaugh, An accurate four-quadrant nonlinear dynamical model for marine thrusters: theory and experimental validation. IEEE J. Oceanic Eng. **25**(1), 146–159 (2000)
21. A. Healey, S.M. Rock, S. Cody, D. Miles, J.P. Brown, Toward an improved understanding of thruster dynamics for underwater vehicles. IEEE J. Oceanic Eng. **20**(4), 354–361 (1995)
22. J. Kim, W.K. Chung, Accurate and practical thruster modeling for underwater vehicles. Ocean Eng. **33**(5), 566–586 (2006)
23. C.L. Tsukamoto, W. Lee, J. Yuh, S.K. Choi, J. Lorentz, Comparison study on advanced thruster control of underwater robots, in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1845–1850, 1997
24. L.L. Whitcomb, D. Yoerger, Development, comparison, and preliminary experimental validation of nonlinear dynamic thruster models. IEEE J. Oceanic Eng. **24**(4), 481–494 (1999)
25. D. Yoerger, J.G. Cooke, J.J. Slotine, The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. IEEE J. Oceanic Eng. **15**(3), 167–178 (1990)
26. T.H. Koh, M.W.S. Lau, E. Low, G. Seet, S. Swei, P.L. Cheng, A study of the control of an underactuated underwater robotic vehicle, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, IEEE, Lausanne, CH, 2001, pp. 2049–2054
27. D.A. Smallwood, L.L. Whitcomb, The effect of model accuracy and thruster saturation on tracking performance of model based controllers for underwater robotic vehicles: experimental results, in *Proceedings of IEEE International Conference on Robotics and Automation, ICRA'02*, vol. 2, IEEE, Washington, DC, 2002, pp. 1081–1087

28. M.B. Larsen, Synthetic long baseline navigation of underwater vehicles, in *Proceedings MTS/IEEE Conference Oceans 2000*, 2000, pp. 2043–2050
29. A.S. Gadre, D.J. Stilwell, Underwater navigation in the presence of unknown currents based on range measurements from a single location, in *Proceedings 2005 American Control Conference*, Portland, OR, June 2005, pp. 565–661
30. A. Bahr, J.J. Leonard, M.F. Fallon, Cooperative localization for autonomous underwater vehicles. Int. J. Robot. Res. **28**, 714–728 (2009)
31. S.E. Webster, R.M. Eustice, H. Singh, L.L. Whitcomb. Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles, in *Proceeding of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 2053–2060
32. T. Maki, T. Matsuda, T. Sakamaki, T. Ura, J. Kojima, Navigation method for underwater vehicles based on mutual acoustical positioning with a single seafloor station. IEEE J. Oceanic Eng. **38**(1), 167–177 (2013)
33. M. Erol-Kantarci, H. Mouftah, S. Oktug, A survey of architectures and localization techniques for underwater acoustic sensor networks. IEEE Commun. Surv. Tutor. **13**(3), 487–502 (2011)
34. M. Stojanovic, J. Preisig, Underwater acoustic communication channels: propagation models and statistical characterization. IEEE Commun. Mag. **47**(1), 84–89 (2009)
35. D. Pompili, I. Akyildiz, Overview of networking protocols for underwater wireless communications. IEEE Commun. Mag. **47**(1), 97–102 (2009)
36. D.A. Smallwood, L.L. Whitcomb, Model-based dynamic positioning of underwater robotic vehicles: theory and experiment. IEEE J. Oceanic Eng. **29**(1), 169–186 (2004)
37. D. Marco, A. Healey, Command, control, and navigation experimental results with the NPS ARIES AUV. IEEE J. Oceanic Eng. **26**(4), 466–476 (2001)
38. A. Healey, D. Lienard, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J. Oceanic Eng. **18**(3), 327–339 (1993)
39. A. Balasuriya, T. Ura, Autonomous target tracking by Twin-Burger 2, in *Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000 (IROS 2000)*, IEEE, vol. 2, 2000, pp. 849–854
40. M.D. Feezor, S.F. Yates, P. Blankinship, J. Bellingham, Autonomous underwater vehicle homing/docking via electromagnetic guidance. IEEE J. Oceanic Eng. **26**(4), 515–521 (2001)
41. J.Y. Park, B.H. Jun, P.M. Lee, J. Oh, Experiments on vision guided docking of an autonomous underwater vehicle using one camera. Ocean Eng. **36**(1), 48–61 (2009)
42. S. Gulati, K. Richmond, C. Flesher, A. Murarka, B.P. Hogan, G. Kuhlmann, M. Sridharan, W.C. Stone, P.T. Doran, Toward autonomous scientific exploration of ice-covered lakesField experiments with the ENDURANCE AUV in an Antarctic Dry Valley, in *Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2010, pp. 308–315
43. A. Alessandri, M. Caccia, G. Indiveri, G. Veruggio, Application of LS and EKF techniques to the identification of underwater vehicles, in *Proceedings of the 1998 IEEE International Conference on Control Applications*, vol. 2, IEEE, Trieste, 1998, pp. 1084–1088
44. M. Caccia, G. Indiveri, G. Veruggio, Modeling and identification of open-frame variable configuration unmanned underwater vehicles. IEEE J. Oceanic Eng. **25**(2), 227–240 (2000)
45. S. Eiani-Cherif, G. Lebret, M. Perrier, Identification and control of a submarine vehicle, in *Proceedings of the 5th IFAC Symposium on Robot, Control*, 1997, pp. 327–332
46. P. Ridao, A. Tiano, A. El-Fakdi, M. Carreras, A. Zirilli, On the identification of non-linear models of unmanned underwater vehicles. Control Eng. Pract. **12**(12), 1483–1499 (2004)
47. C.J. McFarland, L.L. Whitcomb, Comparative experimental evaluation of a new adaptive identifier for underwater vehicles, in *Proceedings of ICRA'13 IEEE International Conference on Robotics and Automation, 2013 (ICRA'13)*, IEEE, 2013, pp. 4599–4605
48. S.C. Martin, L.L. Whitcomb, Preliminary experiments in comparative experimental identification of six degree-of-freedom coupled dynamic plant models for underwater robot vehicles, in

Proceedings of IEEE International Conference on Robotics and Automation, 2013 (ICRA'13), IEEE, 2013, pp. 2947–2954

49. K. Valavanis, D. Gracanin, M. Matijasevic, R. Kolluru, G.A. Demetriou, Control architectures for autonomous underwater vehicles. IEEE Control Syst. **17**(6), 48–64 (1997)
50. N. Palomeras, A. El-Fakdi, M. Carreras, P. Ridao, COLA2: a control architecture for AUVs. IEEE J. Oceanic Eng. **37**(4), 695–716 (2012)
51. V. Kumar, D. Rus, S. Sukhatme, in *Networked Robots*, ed. by B. Siciliano, O. Khatib. Springer Handbook of Robotics (Springer, Heidelberg, 2008), pp. 943–958
52. Y.U. Cao, A.S. Fukunaga, A.B. Kahng, in *Cooperative Mobile Robotics: Antecedents and Directions*, ed. by R.C. Arkin, G.A. Bekey. Robot Colonies. Special Issue of Autonomous Robots, vol. 4 (Kluwer Academic Publisher, Boston, 1997), pp. 7–27
53. L.E. Parker, Distributed intelligence: overview of the field and its application in multi-robot systems. J. Phys. Agents **2**(1), 5 (2008)
54. L.E. Parker, in *Multiple Mobile Robot Systems*, ed. by B. Siciliano, O. Khatib. Springer Handbook of Robotics (Springer, Heidelberg, 2008), pp. 921–941
55. V. Gazi, K. Passino, *Swarm Stability and Optimization* (Springer, Heidelberg, 2010)
56. G. Antonelli, Interconnected dynamic systems. An overview on distributed control. IEEE Control Syst. Mag. **33**(1), 76–88 (2013)
57. D. Cecchi, A. Caiti, S. Fioravanti, F. Baralli, E. Bovio, Target detection using multiple autonomous underwater vehicles, in *Proceedings IARP International Workshop on Underwater Robotics*, Genova, I, 2005, pp. 161–168
58. I. Ihle, R. Skjetne, T. Fossen, Nonlinear formation control of marine craft with experimental results, in *43rd IEEE Conference on Decision and Control, 2004, CDC*, vol. 1, IEEE, 2004, pp. 680–685
59. A. Healey, Application of formation control for multi-vehicle robotic minesweeping, in *Proceedings of the 40th IEEE Conference on Decision and Control, 2001*, vol. 2, IEEE, 2001, pp. 1497–1502
60. E. Fiorelli, N. Leonard, P. Bhatta, D. Paley, R. Bachmayer, D. Fratantoni, Multi-AUV control and adaptive sampling in Monterey Bay. IEEE J. Oceanic Eng. **31**(4), 935–948 (2006)
61. P. Bhatta, E. Fiorelli, F. Lekien, N. Leonard, D. Paley, F. Zhang, R. Bachmayer, R. Davis, D. Fratantoni, R. Sepulchre, Coordination of an underwater glider fleet for adaptive ocean sampling, in *Proceedings IARP International Workshop on Underwater Robotics*, Genova, I, 2005, pp. 61–69
62. N. Leonard, D. Paley, R. Davis, D. Fratantoni, F. Lekien, F. Zhang, Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. J. Field Robot. **27**(6), 718–740 (2010)
63. N. Leonard, E. Fiorelli, Virtual leaders, artificial potentials and coordinated control of groups, in *Proceedings of the 40th IEEE Conference on Decision and Control, 2001*, vol. 3, IEEE, 2001, pp. 2968–2973
64. D. Stilwell, Decentralized control synthesis for a platoon of autonomous vehicles, in *Proceedings of IEEE International Conference on Robotics and Automation, 2002, ICRA'02*, vol. 1, IEEE, 2002, pp. 744–749
65. D. Stilwell, B. Bishop, Platoons of underwater vehicles. IEEE Control Syst. **20**(6), 45–52 (2000)
66. A. Gadre, J. Mach, D. Stilwell, C.E. Wick, Design of a prototype miniature autonomous underwater vehicle, in *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003 (IROS 2003)*, vol. 1, IEEE, 2003, pp. 842–846
67. R. Kumar, J. Stover, A behavior-based intelligent control architecture with application to coordination of multiple underwater vehicles. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **30**(6), 767–784 (2000)
68. D. Lane, G. Bartolini, G. Cannata, G. Casalino, J.B.C. Davies, G. Veruggio, M. Canals, C. Smith, Advanced manipulation for deep underwater sampling: the AMADEUS research project, in *Proceedings of the 1998 IEEE International Conference on Control Applications, 1998*, vol. 2, IEEE, 1998, pp. 1068–1073

69. G. Marani, J. Yuh, S.K. Choi, Autonomous manipulation for an intervention AUV. IEE Control Eng. Ser. **69**, 217 (2006)

70. J. Fernández, M. Prats, P. Sanz, J. C. García Sánchez, R. Marin, M. Robinson, D. Ribas, P. Ridao, Grasping for the Seabed: Developing a New Underwater Robot Arm for Shallow-Water Intervention. IEEE Robot. Autom. Mag. (2013). doi:10.1109/MRA.2013.2248307

Chapter 2
Modelling of Underwater Robots

> *"We have Einstein's space, de Sitter's spaces, expanding universes, contract- ing universes, vibrating universes, mysterious universes. In fact the pure mathematician may create universes just by writing down an equation, and indeed, if he is an individualist he can have an universe of his own".*
>
> J.J. Thomson, around 1919.

2.1 Introduction

In this chapter the mathematical model of UVMSs is derived. Modeling of rigid bodies moving in a fluid or underwater manipulators has been studied in literature by, among others, [1–12], where a deeper discussion of specific aspects can be found. In [13], the model of two UVMSs holding the same rigid object is derived. A short introduction to underwater vehicles, without manipulators, thus, is given by [14], while deep discussion may be found in [15–17].

2.2 Rigid Body's Kinematics

A rigid body is completely described by its position and orientation with respect to a reference frame Σ_i, $O - \boldsymbol{xyz}$ that it is supposed to be earth-fixed and inertial. Let define $\boldsymbol{\eta}_1 \in \mathbb{R}^3$ as

$$\boldsymbol{\eta}_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_2,

the vector of the body position coordinates in a earth-fixed reference frame. The vector $\dot{\boldsymbol{\eta}}_1$ is the corresponding time derivative (expressed in the earth-fixed frame). If one defines

$$\boldsymbol{\nu}_1 = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

as the linear velocity of the origin of the body-fixed frame Σ_b, $O_b - \boldsymbol{x}_b\boldsymbol{y}_b\boldsymbol{z}_b$ with respect to the origin of the earth-fixed frame expressed in the body-fixed frame (from now on: body-fixed linear velocity) the following relation between the defined linear velocities holds:

$$\boldsymbol{\nu}_1 = \boldsymbol{R}_I^B \dot{\boldsymbol{\eta}}_1 \,, \tag{2.1}$$

where $\boldsymbol{R}_I^B$ is the rotation matrix expressing the transformation from the inertial frame to the body-fixed frame.

In the following, two different attitude representations will be introduced: Euler angles and Euler parameters or quaternion. In marine terminology is common the use of Euler angles while several control strategies use the quaternion in order to avoid the representation singularities that might arise by the use of Euler angles.

2.2.1 Attitude Representation by Euler Angles

Let define $\boldsymbol{\eta}_2 \in \mathbb{R}^3$ as

$$\boldsymbol{\eta}_2 = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

the vector of body Euler-angle coordinates in a earth-fixed reference frame. In the nautical field those are commonly named roll, pitch and yaw angles and corresponds to the elementary rotation around x, y and z in fixed frame [18]. The vector $\dot{\boldsymbol{\eta}}_2$ is the corresponding time derivative (expressed in the inertial frame). Let define

$$\boldsymbol{\nu}_2 = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

as the angular velocity of the body-fixed frame with respect to the earth-fixed frame expressed in the body-fixed frame (from now on: body-fixed angular velocity). The vector $\dot{\boldsymbol{\eta}}_2$ does not have a physical interpretation and it is related to the body-fixed angular velocity by a proper Jacobian matrix:

$$\boldsymbol{\nu}_2 = \boldsymbol{J}_{k,o}(\boldsymbol{\eta}_2)\dot{\boldsymbol{\eta}}_2 \,. \tag{2.2}$$

The matrix $\boldsymbol{J}_{k,o} \in \mathbb{R}^{3\times 3}$ can be expressed in terms of Euler angles as:

$$\boldsymbol{J}_{k,o}(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix}, \tag{2.3}$$

where c_α and s_α are short notations for $\cos(\alpha)$ and $\sin(\alpha)$, respectively. Matrix $\boldsymbol{J}_{k,o}(\boldsymbol{\eta}_2)$ is not invertible for every value of $\boldsymbol{\eta}_2$. In detail, it is

$$\boldsymbol{J}_{k,o}^{-1}(\boldsymbol{\eta}_2) = \frac{1}{c_\theta} \begin{bmatrix} 1 & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -c_\theta s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}, \tag{2.4}$$

that it is singular for $\theta = (2l+1)\frac{\pi}{2}$ rad, with $l \in \mathbb{N}$, i.e., for a pitch angle of $\pm\frac{\pi}{2}$ rad.

The following script `J_ko_rpy.m` is available in SIMURV4.0 to compute the Jacobian in (2.3):

```
 function J_ko = J_ko_rpy(rpy)
%
% Jacobian to transform derivative of the
%   Euler angles to body-fixed angular velocity
%
% function J_ko = J_ko_rpy(rpy)
%
% input:
%       rpy     dim 3x1     roll-pitch-yaw angles
%
% output:
%       J_ko    dim 3x3     Jacobian matrix
```

The rotation matrix $\boldsymbol{R}_I^B$, needed in (2.1) to transform the linear velocities, is expressed in terms of Euler angles by the following:

$$\boldsymbol{R}_I^B(\boldsymbol{\eta}_2) = \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ -s_\psi c_\phi + c_\psi s_\theta s_\phi & c_\psi c_\phi + s_\psi s_\theta s_\phi & s_\phi c_\theta \\ s_\psi s_\phi + c_\psi s_\theta c_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi & c_\phi c_\theta \end{bmatrix}. \tag{2.5}$$

Table 2.1 shows the common notation used for marine vehicles according to the SNAME notation [19], Fig. 2.1 shows the defined frames and the elementary motions.

The following script `Rpy2Rot.m` is available in SIMURV4.0 to compute the rotation matrix from roll-pitch-yaw angles:

Table 2.1 Common notation for marine vehicle's motion

		Forces and moments	$\boldsymbol{\nu}_1, \boldsymbol{\nu}_2$	$\boldsymbol{\eta}_1, \boldsymbol{\eta}_2$
Motion in the x-direction	Surge	X	u	x
Motion in the y-direction	Sway	Y	v	y
Motion in the z-direction	Heave	Z	w	z
Rotation about the x-axis	Roll	K	p	ϕ
Rotation about the y-axis	Pitch	M	q	θ
Rotation about the z-axis	Yaw	N	r	ψ

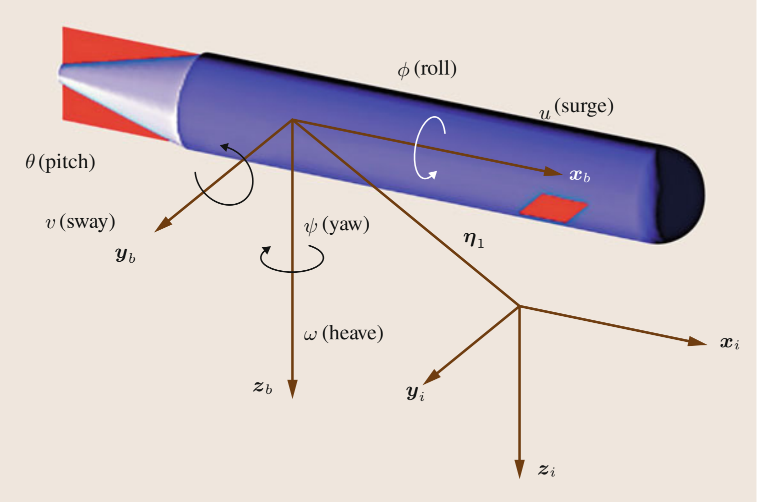


Fig. 2.1 Frames and elementary vehicle's motion

```
function R = Rpy2Rot(rpy)
%
% Rotation matrix from body-fixed frame to inertial
   frame in roll-pitch-yaw angles
%
% function R = Rpy2Rot(rpy)
%
% input:
%        rpy   dim 3x1      roll-pitch-yaw angles
%
% output:
%        R        dim 3x3       Rotation matrix
```

2.2.2 Attitude Representation by Quaternion

To overcome the possible occurrence of representation singularities it might be convenient to resort to non-minimal attitude representations. One possible choice is given by the quaternion. The term *quaternion* was introduced by Hamilton in 1840, 70 years after the introduction of a four-parameter rigid-body attitude representation by Euler. In the following, a short introduction to quaternion is given.

By defining the mutual orientation between two frames of common origin in terms of the rotation matrix

$$\boldsymbol{R_k}(\delta) = \cos\delta \boldsymbol{I}_3 + (1 - \cos\delta)\boldsymbol{k}\boldsymbol{k}^{\mathrm{T}} - \sin\delta \boldsymbol{S}(\boldsymbol{k}) ,$$

where δ is the angle and $\boldsymbol{k} \in \mathbb{R}^3$ is the unit vector of the axis expressing the rotation needed to align the two frames, $\boldsymbol{I}_3$ is the (3×3) identity matrix, $\boldsymbol{S}(\boldsymbol{x})$ is the matrix operator performing the cross product between two (3×1) vectors

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} , \tag{2.6}$$

the unit quaternion is defined as

$$\mathcal{Q} = \{\boldsymbol{\varepsilon}, \eta\}$$

with

$$\boldsymbol{\varepsilon} = \boldsymbol{k}\sin\frac{\delta}{2} ,$$
$$\eta = \cos\frac{\delta}{2} ,$$

where $\eta \geq 0$ for $\delta \in [-\pi, \pi]$ rad. This restriction is necessary for uniqueness of the quaternion associated to a given matrix, in that the two quaternion $\{\boldsymbol{\varepsilon}, \eta\}$ and $\{-\boldsymbol{\varepsilon}, -\eta\}$ represent the same orientation, i.e., the same rotation matrix. Notice that the order between the scalar and vector part in the quaternion is arbitrary, it is required to obviously take it into account in the corresponding equations.

The unit quaternion satisfies the condition

$$\eta^2 + \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = 1 . \tag{2.7}$$

The relationship between $\boldsymbol{\nu}_2$ and the time derivative of the quaternion is given by the *quaternion propagation equations*

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\eta\boldsymbol{\nu}_2 + \frac{1}{2}\boldsymbol{S}(\boldsymbol{\varepsilon})\boldsymbol{\nu}_2 \,, \tag{2.8}$$

$$\dot{\eta} = -\frac{1}{2}\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\nu}_2 \,, \tag{2.9}$$

that can be rearranged in the form:

$$\begin{bmatrix} \dot{\boldsymbol{\varepsilon}} \\ \dot{\eta} \end{bmatrix} = \boldsymbol{J}_{k,oq}(\mathcal{Q})\boldsymbol{\nu}_2 = \frac{1}{2}\begin{bmatrix} \eta & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & \eta & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & \eta \\ -\varepsilon_1 & -\varepsilon_2 & -\varepsilon_3 \end{bmatrix}\boldsymbol{\nu}_2 \,. \tag{2.10}$$

The matrix $\boldsymbol{J}_{k,oq}(\mathcal{Q})$ satisfies:

$$\boldsymbol{J}_{k,oq}^{\mathrm{T}}\boldsymbol{J}_{k,oq} = \frac{1}{4}\boldsymbol{I}_3 \,,$$

that allows to invert the mapping (2.10) yielding:

$$\boldsymbol{\nu}_2 = 4\boldsymbol{J}_{k,oq}^{\mathrm{T}}\begin{bmatrix} \dot{\boldsymbol{\varepsilon}} \\ \dot{\eta} \end{bmatrix} .$$

For completeness the rotation matrix $\boldsymbol{R}_I^B$, needed to compute (2.1), in terms of quaternion is given:

$$\boldsymbol{R}_I^B(\mathcal{Q}) = \begin{bmatrix} 1-2(\varepsilon_2^2+\varepsilon_3^2) & 2(\varepsilon_1\varepsilon_2+\varepsilon_3\eta) & 2(\varepsilon_1\varepsilon_3-\varepsilon_2\eta) \\ 2(\varepsilon_1\varepsilon_2-\varepsilon_3\eta) & 1-2(\varepsilon_1^2+\varepsilon_3^2) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\eta) \\ 2(\varepsilon_1\varepsilon_3+\varepsilon_2\eta) & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\eta) & 1-2(\varepsilon_1^2+\varepsilon_2^2) \end{bmatrix} . \tag{2.11}$$

The following script `Quat2Rot.m` is available in SIMURV4.0 to compute the rotation matrix from quaternions:

```
function R = Quat2Rot(e)
%
% Rotation matrix from body-fixed frame to inertial
   frame in quaternions
%
% function R = Quat2Rot(e)
%
% input:
%       e      dim 4x1     quaternion
%
% output:
%       R      dim 3x3      Rotation matrix
```

2.2.3 Attitude Error Representation

Let now define $\boldsymbol{R}_B^I \in \mathbb{R}^{3\times 3}$ as the rotation matrix from the body-fixed frame to the earth-fixed frame, which is also described by the quaternion $\mathcal{Q}$, and $\boldsymbol{R}_d^I \in \mathbb{R}^{3\times 3}$ the rotation matrix from the frame expressing the desired vehicle orientation to the earth-fixed frame, which is also described by the quaternion $\mathcal{Q}_d = \{\boldsymbol{\varepsilon}_d, \eta_d\}$. One possible choice for the rotation matrix necessary to align the two frames is

$$\widetilde{\boldsymbol{R}} = \boldsymbol{R}_B^{I^{\mathrm{T}}} \boldsymbol{R}_d^I = \boldsymbol{R}_I^B \boldsymbol{R}_d^I ,$$

where $\boldsymbol{R}_I^B = \boldsymbol{R}_B^{I\,\mathrm{T}}$. The quaternion $\widetilde{\mathcal{Q}} = \{\tilde{\boldsymbol{\varepsilon}}, \tilde{\eta}\}$ associated with $\widetilde{\boldsymbol{R}}$ can be obtained directly from $\widetilde{\boldsymbol{R}}$ or computed by composition (quaternion product): $\widetilde{\mathcal{Q}} = \mathcal{Q}^{-1} * \mathcal{Q}_d$, where $\mathcal{Q}^{-1} = \{-\boldsymbol{\varepsilon}, \eta\}$:

$$\tilde{\boldsymbol{\varepsilon}} = \eta \boldsymbol{\varepsilon}_d - \eta_d \boldsymbol{\varepsilon} + \boldsymbol{S}(\boldsymbol{\varepsilon}_d)\boldsymbol{\varepsilon}, \tag{2.12}$$

$$\tilde{\eta} = \eta \eta_d + \boldsymbol{\varepsilon}^{\mathrm{T}} \boldsymbol{\varepsilon}_d . \tag{2.13}$$

Since the quaternion associated with $\widetilde{\boldsymbol{R}} = \boldsymbol{I}_3$ (i.e. representing two aligned frames) is $\widetilde{\mathcal{Q}} = \{\boldsymbol{0}, 1\}$, it is sufficient to represent the attitude error as $\tilde{\boldsymbol{\varepsilon}}$.

The quaternion propagation equations can be rewritten also in terms of the error variables:

$$\dot{\tilde{\boldsymbol{\varepsilon}}} = \frac{1}{2}\tilde{\eta}\tilde{\boldsymbol{\nu}}_2 + \frac{1}{2}\boldsymbol{S}(\tilde{\boldsymbol{\varepsilon}})\tilde{\boldsymbol{\nu}}_2 , \tag{2.14}$$

$$\dot{\tilde{\eta}} = -\frac{1}{2}\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\tilde{\boldsymbol{\nu}}_2 , \tag{2.15}$$

where $\tilde{\boldsymbol{\nu}}_2 = \boldsymbol{\nu}_{2,d} - \boldsymbol{\nu}_2$ is the angular velocity error expressed in body-fixed frame. Defining

$$\boldsymbol{z} = \begin{bmatrix} \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\eta} \end{bmatrix} ,$$

the relations in (2.14)–(2.15) can be rewritten in the form:

$$\dot{\boldsymbol{z}} = \frac{1}{2}\begin{bmatrix} \tilde{\eta}\boldsymbol{I}_3 + \boldsymbol{S}(\tilde{\boldsymbol{\varepsilon}}) \\ -\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}} \end{bmatrix} \tilde{\boldsymbol{\nu}}_2 = \boldsymbol{J}_{k,oq}(\boldsymbol{z})\tilde{\boldsymbol{\nu}}_2 \tag{2.16}$$

The equations above are given in terms of the body-fixed angular velocity. In fact, they will be used in the control laws of Chap. 7. The generic expression of the propagation equations is the following:

$$\dot{\tilde{\boldsymbol{\varepsilon}}}_{ba}^{a} = \frac{1}{2}\boldsymbol{E}(\widetilde{\mathcal{Q}}_{ba})\tilde{\boldsymbol{\omega}}_{ba}^{a} ,$$

$$\dot{\tilde{\eta}}_{ba} = -\frac{1}{2}\tilde{\boldsymbol{\varepsilon}}_{ba}^{a^{\mathrm{T}}}\tilde{\boldsymbol{\omega}}_{ba}^{a} ,$$

with

$$\boldsymbol{E}(\widetilde{\mathcal{Q}}_{ba}) = \widetilde{\eta}_{ba}\boldsymbol{I}_3 - \boldsymbol{S}(\widetilde{\boldsymbol{\varepsilon}}^a_{ba}).$$

where $\widetilde{\mathcal{Q}}_{ba} = \{\widetilde{\boldsymbol{\varepsilon}}^a_{ba}, \widetilde{\eta}_{ba}\}$ is the quaternion associated to $\boldsymbol{R}^a_b = \boldsymbol{R}^{\mathrm{T}}_a\boldsymbol{R}_b$ and the angular velocity $\widetilde{\boldsymbol{\omega}}^a_{ba} = \boldsymbol{R}^{\mathrm{T}}_a(\boldsymbol{\omega}_b - \boldsymbol{\omega}_a)$ of the frame Σ_b relative to the frame Σ_a, expressed in the frame Σ_a.

A survey on rigid-body attitude control may be found in [20].

Quaternion from Rotation Matrix

It can be useful to recall the procedure needed to extract the quaternion from the rotation matrix [15, 21].

Given a generic rotation matrix $\boldsymbol{R}$:

1. compute the trace of $\boldsymbol{R}$ according to:

$$R_{4,4} = \mathrm{tr}(\boldsymbol{R}) = \sum_{j=1}^{3} R_{j,j}$$

2. compute the index i according to:

$$R_{i,i} = \max(R_{1,1}, R_{2,2}, R_{3,3}, R_{4,4})$$

3. define the scalar c_i as:

$$|c_i| = \sqrt{1 + 2R_{i,i} - R_{4,4}}$$

in which the sign can be plus or minus.

4. compute the other three values of c by knowing the following relationships:

$$\begin{aligned} c_4c_1 &= R_{3,2} - R_{2,3} \\ c_4c_2 &= R_{1,3} - R_{3,1} \\ c_4c_3 &= R_{2,1} - R_{1,2} \\ c_2c_3 &= R_{3,2} + R_{2,3} \\ c_3c_1 &= R_{1,3} + R_{3,1} \\ c_1c_2 &= R_{2,1} + R_{1,2} \end{aligned}$$

simply dividing the equations in which c_i is involved by c_i itself.

5. compute the quaternion $\mathcal{Q}$ by the following:

$$\left[\boldsymbol{\varepsilon}\ \eta\right]^{\mathrm{T}} = \frac{1}{2}\left[c_1\ c_2\ c_3\ c_4\right]^{\mathrm{T}}.$$

The following script `Rot2Quat.m` is available in SIMURV4.0 to extract the quaternions from the rotation:

```
function e = Rot2Quat(R)
%
% From rotation matrix to quaternions
%
% function e = Rot2Quat(R)
%
% input:
%       R        dim 3x3      rotation matrix
%
% output:
%       e        dim 4x1       quaternion
```

Quaternion from Euler Angles

The transformation from Euler angles to quaternion is always possible, i.e., it is not affected by the occurrence of representation singularities [15]. This implies that the use of quaternion to control underwater vehicles is compatible with the common use of Euler angles to express the desired trajectory of the vehicle.

The algorithm consists in computing the rotation matrix expressed in Euler angles by (2.5) and using the procedure described in the previous subsection to extract the corresponding quaternion. In SIMURV4.0 this is done in the file `Rpy2Quat.m`.

2.2.4 Six-DOFs Kinematics

It is useful to collect the kinematic equations in 6-dimensional matrix forms. Let us define the vector $\boldsymbol{\eta} \in \mathbb{R}^6$ as

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} \tag{2.17}$$

and the vector $\boldsymbol{\nu} \in \mathbb{R}^6$ as

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{bmatrix}, \tag{2.18}$$

and by defining the matrix $\boldsymbol{J}_e(\boldsymbol{R}_B^I) \in \mathbb{R}^{6\times 6}$

$$\boldsymbol{J}_e(\boldsymbol{R}_B^I) = \begin{bmatrix} \boldsymbol{R}_I^B & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{O}_{3\times 3} & \boldsymbol{J}_{k,o} \end{bmatrix}, \tag{2.19}$$

where the rotation matrix $\boldsymbol{R}_I^B$ given in (2.5) and $\boldsymbol{J}_{k,o}$ is given in (2.3), it is

$$\boldsymbol{\nu} = \boldsymbol{J}_e(\boldsymbol{R}_B^I)\dot{\boldsymbol{\eta}}. \tag{2.20}$$

The inverse mapping, given the block-diagonal structure of $\boldsymbol{J}_e$, is given by:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_e^{-1}(\boldsymbol{R}_B^I)\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{R}_B^I & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{O}_{3\times 3} & \boldsymbol{J}_{k,o}^{-1} \end{bmatrix} \boldsymbol{\nu}\,, \tag{2.21}$$

where $\boldsymbol{J}_{k,o}^{-1}$ is given in (2.4).

The following script `J_e.m` tautologically computes $\boldsymbol{J}_e$:

```
function J = J_e(rpy)
%
% Jacobian to transform derivative of the
% vehicle position + Euler angles
% to body-fixed linear and angular velocities
%
% function J = J_e(rpy)
%
% input:
%       rpy     dim 3x1     roll-pitch-yaw angles
%
% output:
%       J    dim 6x6     Jacobian matrix
```

On the other hand, it is possible to represent the orientation by means of quaternions. Let us define the vector $\boldsymbol{\eta}_q \in \mathbb{R}^7$ as

$$\boldsymbol{\eta}_q = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\varepsilon} \\ \eta \end{bmatrix} \tag{2.22}$$

and the matrix $\boldsymbol{J}_{e,q}(\boldsymbol{R}_B^I) \in \mathbb{R}^{6\times 7}$

$$\boldsymbol{J}_{e,q}(\boldsymbol{R}_B^I) = \begin{bmatrix} \boldsymbol{R}_I^B & \boldsymbol{O}_{3\times 4} \\ \boldsymbol{O}_{3\times 3} & 4\boldsymbol{J}_{k,oq}^{\mathrm{T}} \end{bmatrix}\,, \tag{2.23}$$

where $\boldsymbol{J}_{k,oq}$ is given in (2.10); it is

$$\boldsymbol{\nu} = \boldsymbol{J}_{e,q}(\boldsymbol{R}_B^I)\dot{\boldsymbol{\eta}}_e\,. \tag{2.24}$$

The inverse mapping is given by:

$$\dot{\boldsymbol{\eta}}_e = \begin{bmatrix} \boldsymbol{R}_B^I & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{O}_{4\times 3} & \boldsymbol{J}_{k,oq} \end{bmatrix} \boldsymbol{\nu}\,. \tag{2.25}$$

2.3 Rigid Body's Dynamics

Several approaches can be considered when deriving the equations of motion of a rigid body. In the following, the Newton-Euler formulation will be briefly summarized.

The motion of a generic system of material particles subject to external forces can be described by resorting to the *fundamental principles of dynamics* (Newton's laws of motion). Those relate the resultant force and moment to the time derivative of the linear and angular momentum.

Let ρ be the density of a particle of volume dV of a rigid body $\mathcal{B}$, ρdV is the corresponding mass denoted by the position vector $\boldsymbol{p}$ in an inertial frame $O - \boldsymbol{xyz}$. Let also $V_{\mathcal{B}}$ be the body volume and

$$m = \int_{V_{\mathcal{B}}} \rho dV$$

be the total mass. The *center of mass* of $\mathcal{B}$ is defined as

$$\boldsymbol{p}_C = \frac{1}{m} \int_{V_{\mathcal{B}}} \boldsymbol{p} \rho dV.$$

The *linear momentum* of the body $\mathcal{B}$ is defined as the vector

$$\boldsymbol{l} = \int_{V_{\mathcal{B}}} \dot{\boldsymbol{p}} \rho dV = m \dot{\boldsymbol{p}}_C.$$

For a system with constant mass, the Newton's law of motion for the linear part

$$\boldsymbol{f} = \dot{\boldsymbol{l}} = m \frac{d}{dt} \dot{\boldsymbol{p}}_C \tag{2.26}$$

can be rewritten simply by the *Newton's equations of motion*:

$$\boldsymbol{f} = m \ddot{\boldsymbol{p}}_C \tag{2.27}$$

where $\boldsymbol{f}$ is the resultant of the external forces.

Let us define the *Inertia tensor* of the body $\mathcal{B}$ relative to the pole O:

$$\boldsymbol{I}_O = \int_{V_{\mathcal{B}}} \boldsymbol{S}^{\mathrm{T}}(\boldsymbol{p}) \boldsymbol{S}(\boldsymbol{p}) \rho dV,$$

where $\boldsymbol{S}$ is the skew-symmetric operator defined in (2.6). The matrix $\boldsymbol{I}_O$ is symmetric and positive definite. The positive diagonal elements I_{Oxx}, I_{Oyy}, I_{Ozz} are the *inertia moments* with respect to the three coordinate axes of the reference frame. The off diagonal elements are the *products of inertia*.

The relationship between the inertia tensor in two different frames $\boldsymbol{I}_O$ and $\boldsymbol{I}'_O$, related by a rotation matrix $\boldsymbol{R}$, with the same pole O, is the following:

$$\boldsymbol{I}_O = \boldsymbol{R}\boldsymbol{I}'_O\boldsymbol{R}^{\mathrm{T}}.$$

The change of pole is related by the *Steiner's theorem*:

$$\boldsymbol{I}_O = \boldsymbol{I}_C + m\boldsymbol{S}^{\mathrm{T}}(\boldsymbol{p}_C)\boldsymbol{S}(\boldsymbol{p}_C),$$

where $\boldsymbol{I}_C$ is the inertial tensor relative to the center of mass, when expressed in a frame parallel to the frame in which $\boldsymbol{I}_O$ is defined.

Notice that O can be either a fixed or moving pole. In case of a fixed pole the elements of the inertia tensor are function of time. A suitable choice of the pole might be a point fixed to the rigid body in a way to obtain a constant inertia tensor. Moreover, since the inertia tensor is symmetric positive definite is always possible to find a frame in which the matrix attains a diagonal form, this frame is called *principal frame*, also, if the pole coincides with the center of mass, it is called *central frame*. This is true also if the body does not have a significant geometric symmetry.

Let Ω be any point in space and $\boldsymbol{p}_\Omega$ the corresponding position vector. Ω can be either moving or fixed with respect to the reference frame. The *angular momentum* of the body $\mathcal{B}$ relative to the pole Ω is defined as the vector:

$$\boldsymbol{k}_\Omega = \int_{V_\mathcal{B}} \dot{\boldsymbol{p}} \times \left(\boldsymbol{p}_\Omega - \boldsymbol{p}\right) \rho dV. \tag{2.28}$$

Taking into account the definition of center of mass, (2.28) can be rewritten in the form:

$$\boldsymbol{k}_\Omega = \boldsymbol{I}_C\boldsymbol{\omega} + m\dot{\boldsymbol{p}}_C \times \left(\boldsymbol{p}_\Omega - \boldsymbol{p}_C\right), \tag{2.29}$$

where $\boldsymbol{\omega}$ is the angular velocity.

The resultant moment $\boldsymbol{\mu}_\Omega$ with respect to the pole Ω of a rigid body subject to n external forces $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ is:

$$\boldsymbol{\mu}_\Omega = \sum_{i=1}^{n} \boldsymbol{f}_i \times \left(\boldsymbol{p}_\Omega - \boldsymbol{p}_i\right).$$

In case of a system with constant mass and rigid body, the angular part of the Newton's law of motion

$$\boldsymbol{\mu}_\Omega = \dot{\boldsymbol{k}}_\Omega$$

yields the *Euler equations of motion*:

$$\boldsymbol{\mu}_\Omega = \boldsymbol{I}_\Omega\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \left(\boldsymbol{I}_\Omega\boldsymbol{\omega}\right). \tag{2.30}$$

The right-hand side of the Newton and Euler equations of motion, (2.27) and (2.30), are defined inertial forces and inertial moments, respectively.

2.3.1 Rigid Body's Dynamics in Matrix Form

To derive the equations of motion in matrix form it is useful to refer the quantities to a body-fixed frame $O_b - \boldsymbol{x}_b\boldsymbol{y}_b\boldsymbol{z}_b$ using the body-fixed linear and angular velocities that has been introduced in Sect. 2.2.

The following relationships hold:

$$\boldsymbol{p}_\Omega - \boldsymbol{p}_C = \boldsymbol{R}_B^I \boldsymbol{r}_C^b \tag{2.31}$$

$$\dot{\boldsymbol{R}}_B^I \cdot = \boldsymbol{\omega} \times (\boldsymbol{R}_B^I \cdot) \tag{2.32}$$

$$\boldsymbol{R}_I^B \left(\boldsymbol{\omega} \times \boldsymbol{R}_B^I \cdot\right) = \boldsymbol{\nu}_2 \times \cdot \tag{2.33}$$

$$\boldsymbol{\omega} = \boldsymbol{R}_B^I \boldsymbol{\nu}_2 \tag{2.34}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{R}_B^I \dot{\boldsymbol{\nu}}_2 \tag{2.35}$$

$$\dot{\boldsymbol{p}}_C = \boldsymbol{R}_B^I (\boldsymbol{\nu}_1 + \boldsymbol{\nu}_2 \times \boldsymbol{r}_C^b) \tag{2.36}$$

$$\boldsymbol{I}_C = \boldsymbol{R}_B^I \boldsymbol{I}_C^b \boldsymbol{R}_I^B \tag{2.37}$$

where, according to the (2.31), $\boldsymbol{r}_C^b$ is the vector position from the origin of the body-fixed frame to the center of mass expressed in the body-fixed frame ($\dot{\boldsymbol{r}}_C^b = \boldsymbol{0}$ for a rigid body).

Equation (2.26) can be rewritten in terms of the linear body-fixed velocities as

$$\begin{aligned} \boldsymbol{f} &= m\frac{d}{dt}\left[\boldsymbol{R}_B^I \left(\boldsymbol{\nu}_1 + \boldsymbol{\nu}_2 \times \boldsymbol{r}_C^b\right)\right] \\ &= m\boldsymbol{R}_B^I \left(\dot{\boldsymbol{\nu}}_1 + \dot{\boldsymbol{\nu}}_2 \times \boldsymbol{r}_C^b + \boldsymbol{\nu}_2 \times \dot{\boldsymbol{r}}_C^b\right) + m\boldsymbol{\omega} \times \boldsymbol{R}_B^I (\boldsymbol{\nu}_1 + \boldsymbol{\nu}_2 \times \boldsymbol{r}_C^b), \end{aligned}$$

Premultiplying by $\boldsymbol{R}_I^B$ and defining as

$$\boldsymbol{\tau}_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{2.38}$$

the resultant forces acting on the rigid body expressed in a body-fixed frame, and as

$$\boldsymbol{\tau}_2 = \begin{bmatrix} K \\ M \\ N \end{bmatrix}, \tag{2.39}$$

the corresponding resultant moment to the pole O_b, one obtains:

$$\boldsymbol{\tau}_1 = m\dot{\boldsymbol{v}}_1 + m\dot{\boldsymbol{v}}_2 \times \boldsymbol{r}_C^b + m\boldsymbol{v}_2 \times \boldsymbol{v}_1 + m\boldsymbol{v}_2 \times (\boldsymbol{v}_2 \times \boldsymbol{r}_C^b).$$

Equation (2.29) is written in an inertial frame. It is possible to rewrite the angular momentum in terms of the body-fixed velocities:

$$\boldsymbol{k}_\Omega = \boldsymbol{R}_B^I \left(\boldsymbol{I}_C^b \boldsymbol{v}_2 + m\boldsymbol{v}_1 \times \boldsymbol{r}_C^b \right). \tag{2.40}$$

Derivating (2.40) one obtains:

$$\boldsymbol{\tau}_2^I = \boldsymbol{\omega} \times \boldsymbol{R}_B^I \left(\boldsymbol{I}_C^b \boldsymbol{v}_2 + m\boldsymbol{v}_1 \times \boldsymbol{r}_C^b \right) + \boldsymbol{R}_B^I \left(\boldsymbol{I}_C^b \dot{\boldsymbol{v}}_2 + m\dot{\boldsymbol{v}}_1 \times \boldsymbol{r}_C^b \right),$$

that, using the relations above, can be written in the form:

$$\boldsymbol{\tau}_2 = \boldsymbol{I}_C^b \dot{\boldsymbol{v}}_2 + \boldsymbol{v}_2 \times (\boldsymbol{I}_C^b \boldsymbol{v}_2) + m\boldsymbol{v}_2 \times (\boldsymbol{v}_1 \times \boldsymbol{r}_C^b) + m\dot{\boldsymbol{v}}_1 \times \boldsymbol{r}_C^b.$$

It is now possible to rewrite the Newton-Euler equations of motion of a rigid body moving in the space. It is:

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{v}} + \boldsymbol{C}_{RB}(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau}_v, \tag{2.41}$$

where

$$\boldsymbol{\tau}_v = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{bmatrix}. \tag{2.42}$$

The matrix $\boldsymbol{M}_{RB}$ is constant, symmetric and positive definite, i.e., $\dot{\boldsymbol{M}}_{RB} = \boldsymbol{O}$, $\boldsymbol{M}_{RB} = \boldsymbol{M}_{RB}^{\mathrm{T}} > \boldsymbol{O}$. Its unique parametrization is in the form:

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_3 & -m\boldsymbol{S}(\boldsymbol{r}_C^b) \\ m\boldsymbol{S}(\boldsymbol{r}_C^b) & \boldsymbol{I}_{O_b} \end{bmatrix},$$

where $\boldsymbol{I}_3$ is the (3×3) identity matrix, and $\boldsymbol{I}_{O_b}$ is the inertia tensor expressed in the body-fixed frame.

On the other hand, it does not exist a unique parametrization of the matrix $\boldsymbol{C}_{RB}$, representing the Coriolis and centripetal terms. It can be demonstrated that the matrix $\boldsymbol{C}_{RB}$ can always be parameterized such that it is skew-symmetrical, i.e.,

$$\boldsymbol{C}_{RB}(\boldsymbol{v}) = -\boldsymbol{C}_{RB}^{\mathrm{T}}(\boldsymbol{v}) \qquad \forall \boldsymbol{v} \in \mathbb{R}^6,$$

explicit expressions for $\boldsymbol{C}_{RB}$ can be found, e.g., in [15].

Notice that (2.41) can be greatly simplified if the origin of the body-fixed frame is chosen coincident with the central frame, i.e., $\boldsymbol{r}_C^b = \boldsymbol{0}$ and $\boldsymbol{I}_{O_b}$ is a diagonal matrix.

2.4 Hydrodynamic Effects

In this Section the major hydrodynamic effects on a rigid body moving in a fluid will be briefly discussed.

The theory of fluidodynamics is rather complex and it is difficult to develop a reliable model for most of the hydrodynamic effects. A rigorous analysis for incompressible fluids would need to resort to the Navier-Stokes equations (distributed fluid-flow). However, in this book modeling of the hydrodynamic effects in a context of automatic control is considered. In literature, it is well known that kinematic and dynamic coupling between vehicle and manipulator can not be neglected [6, 8, 22], while most of the hydrodynamic effects have no significant influence in the range of the operative velocities.

2.4.1 Added Mass and Inertia

When a rigid body is moving in a fluid, the additional inertia of the fluid surrounding the body, that is accelerated by the movement of the body, has to be considered. This effect can be neglected in industrial robotics since the density of the air is much lighter than the density of a moving mechanical system. In underwater applications, however, the density of the water, $\rho \approx 1000\,\text{kg/m}^3$, is comparable with the density of the vehicles. In particular, at 0°, the density of the fresh water is $1002.68\,\text{kg/m}^3$; for sea water with 3.5 % of salinity it is $\rho = 1028.48\,\text{kg/m}^3$.

The fluid surrounding the body is accelerated with the body itself, a force is then necessary to achieve this acceleration; the fluid exerts a reaction force which is equal in magnitude and opposite in direction. This reaction force is the added mass contribution. The added mass is not a quantity of fluid to add to the system such that it has an increased mass. Different properties hold with respect to the (6×6) inertia matrix of a rigid body due to the fact that the added mass is function of the body's surface geometry. As an example, the inertia matrix is not necessarily positive definite.

The hydrodynamic force along $\boldsymbol{x}_b$ due to the linear acceleration in the $\boldsymbol{x}_b$-direction is defined as:

$$X_A = -X_{\dot{u}}\dot{u} \qquad \text{where} \qquad X_{\dot{u}} = \frac{\partial X}{\partial \dot{u}},$$

where the symbol ∂ denotes the partial derivative. In the same way it is possible to define all the remaining 35 elements that relate the 6 force/moment components $\begin{bmatrix} X \ Y \ Z \ K \ M \ N \end{bmatrix}^{\text{T}}$ to the 6 linear/angular acceleration $\begin{bmatrix} \dot{u} \ \dot{v} \ \dot{w} \ \dot{p} \ \dot{q} \ \dot{r} \end{bmatrix}^{\text{T}}$. These elements can be grouped in the Added Mass matrix $\boldsymbol{M}_A \in \mathbb{R}^{6\times 6}$. Usually, all the elements of the matrix are different from zero.

There is no specific property of the matrix $\boldsymbol{M}_A$. For certain frequencies and specific bodies, such as catamarans, negative diagonal elements have been documented [15].

However, for completely submerged bodies it can be considered $\boldsymbol{M}_A > \boldsymbol{O}$. Moreover, if the fluid is ideal, the body's velocity is low, there are no currents or waves and frequency independence it holds [23]:

$$\boldsymbol{M}_A = \boldsymbol{M}_A^{\mathrm{T}} > \boldsymbol{O}. \tag{2.43}$$

The added mass has also an *added* Coriolis and centripetal contribution. It can be demonstrated that the matrix expression can always be parameterized such that:

$$\boldsymbol{C}_A(\boldsymbol{\nu}) = -\boldsymbol{C}_A^{\mathrm{T}}(\boldsymbol{\nu}) \qquad \forall \boldsymbol{\nu} \in \mathbb{R}^6.$$

If the body is completely submerged in the water, the velocity is low and it has three planes of symmetry as common for underwater vehicles, the following structure of matrices $\boldsymbol{M}_A$ and $\boldsymbol{C}_A$ can therefore be considered:

$$\boldsymbol{M}_A = -\mathrm{diag}\left\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\right\},$$
$$\boldsymbol{C}_A = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix}.$$

The added mass coefficients can be theoretically derived exploiting the geometry of the rigid body and, eventually, its symmetry [15], by applying the strip theory. For a cylindrical rigid body of mass $\overline{m}$, length $\overline{L}$, with circular section of radius $\overline{r}$, the following added mass coefficients can be derived [15]:

$$\begin{aligned} X_{\dot{u}} &= -0.1\overline{m} \\ Y_{\dot{v}} &= -\pi\rho\overline{r}^2\overline{L} \\ Z_{\dot{w}} &= -\pi\rho\overline{r}^2\overline{L} \\ K_{\dot{p}} &= 0 \\ M_{\dot{q}} &= -\frac{1}{12}\pi\rho\overline{r}^2\overline{L}^3 \\ N_{\dot{r}} &= -\frac{1}{12}\pi\rho\overline{r}^2\overline{L}^3. \end{aligned}$$

Notice that, despite (2.43), in this case it is $\boldsymbol{M}_A \geq \boldsymbol{O}$. This result is due to the geometrical approach to the derivation of $\boldsymbol{M}_A$. As a matter of fact, if a sphere submerged in a fluid is considered, it can be observed that a pure rotational motion of the sphere does not involve any fluid movement, i.e., it is not necessary to add an inertia term due to the fluid. This small discrepancy is just an example of the difficulty in representing with a closed set of equations a distributed phenomenon as fluid movement.

In [24] the added mass coefficients for an ellipsoid are derived.

In [25], and in the Appendix, the coefficients for the experimental vehicle NPS AUV Phoenix are reported. These coefficients have been experimentally derived and, since the vehicle can work at a maximum depth of few meters, i.e., it is not submerged in an unbounded fluid, the structure of $\boldsymbol{M}_A$ is not diagonal. To give an order of magnitude of the added mass terms, the vehicle has a mass of about 5000 kg, the term $X_{\dot{u}} \approx -500$ kg.

A detailed theoretical and experimental discussion on the added mass effect of a cylinder moving in a fluid can be found in [8] where it is shown that the added mass matrix is state-dependent and its coefficients are function of the distance traveled by the cylinder.

2.4.2 Damping Effects

The viscosity of the fluid also causes the presence of dissipative drag and lift forces on the body.

A common simplification is to consider only linear and quadratic damping terms and group these terms in a matrix $\boldsymbol{D}_{RB}$ such that:

$$\boldsymbol{D}_{RB}(\boldsymbol{\nu}) > \boldsymbol{O} \qquad \forall \boldsymbol{\nu} \in \mathbb{R}^6.$$

The coefficients of this matrix are also considered to be constant. For a completely submerged body, the following further assumption can be made:

$$\begin{aligned}\boldsymbol{D}_{RB}(\boldsymbol{\nu}) = &-\mathrm{diag}\left\{X_u, Y_v, Z_w, K_p, M_q, N_r\right\}\\ &-\mathrm{diag}\left\{X_{u|u|}\left|u\right|, Y_{v|v|}\left|v\right|, Z_{w|w|}\left|w\right|, K_{p|p|}\left|p\right|, M_{q|q|}\left|q\right|, N_{r|r|}\left|r\right|\right\}.\end{aligned}$$

Assuming a diagonal structure for the damping matrix implies neglecting the coupling dissipative terms.

The detailed analysis of the dissipative forces is beyond the scope of this work. In the following, only the nature of these forces will be briefly discussed. Introductory analysis of this phenomenon can be found in [3, 10, 15, 24, 26], while in depth discussion in [27, 28].

The viscous effects can be considered as the sum of two forces, the *drag* and the *lift* forces. The former are parallel to the relative velocity between the body and the fluid, while the latter are normal to it. Both drag and lift forces are supposed to act on the center of mass of the body. In order to solve the distributed flow problem, an integral over the entire surface is required to compute the net force/moment acting on the body. Moreover, the model of drag and lift forces is not known and, also for some widely accepted models, the coefficients are not known and variables.

For a sphere moving in a fluid, the drag force can be modeled as [3]:

$$F_{drag} = 0.5\rho U^2 S C_d(R_n),$$

Table 2.2 Lift and drag coefficient for a cylinder

Reynolds number	Regime motion	C_d	C_l
$R_n < 2 \cdot 10^5$	Subcritical flow	1	$3 \div 0.6$
$2 \cdot 10^5 < R_n < 5 \cdot 10^5$	Critical flow	$1 \div 0.4$	0.6
$5 \cdot 10^5 < R_n < 3 \cdot 10^5$	Transcritical flow	0.4	0.6

where ρ is the fluid density, U is the velocity of the sphere, S is the frontal area of the sphere, C_d is the adimensional drag coefficients and R_n is the Reynolds number. For a generic body, S is the projection of the frontal area along the flow direction. The drag coefficient is then dependent on the Reynolds number, i.e., on the laminar/turbulent fluid motion:

$$R_n = \frac{\rho |U| D}{\mu}$$

where D is the characteristic dimension of the body perpendicular to the direction of U and μ is the dynamic viscosity of the fluid. In Table 2.2 the drag coefficients in function of the Reynolds number for a cylinder are reported [10]. The drag coefficients can be considered as the sum of two physical effects: a frictional contribution of the surface whose normal is perpendicular to the flow velocity, and a pressure contribution of the surface whose normal is parallel to the flow velocity.

The lift forces are perpendicular to the flow direction. For an hydrofoil they can be modeled as [3]:

$$F_{lift} = 0.5\rho U^2 S C_l(R_n, \alpha),$$

where C_l is the adimensional lift coefficient. It can be recognized that it also depends on the angle of attack α. In Table 2.2 the lift coefficients in function of the Reynolds number for a cylinder are reported [10].

Vortex induced forces are an oscillatory effect that affects both drag and lift directions. They are caused by the vortex generated by the body that separates the fluid flow. They then cause a periodic *disturbance* that can be the cause of oscillations in cables and some underwater structures. For underwater vehicles it is reasonable to assume that the vortex induced forces are negligible, this, also in view of the adoption of small design surfaces that can reduce this effect. For underwater manipulators with cylindrical links this effects might be experienced.

2.4.3 Current Effects

Control of marine vehicles cannot neglect the effects of specific disturbances such as waves, wind and ocean current. In this book wind and waves phenomena will not be discussed since the attention is focused to autonomous vehicles performing a motion or manipulation task in an underwater environment. However, if this task has to be achieved in very shallow waters, those effects can not be neglected.

Ocean currents are mainly caused by: tidal movement; the atmospheric wind system over the sea earth's surface; the heat exchange at the sea surface; the salinity changes and the Coriolis force due to the earth rotation. Currents can be very different due to local climatic and/or geographic characteristics; as an example, in the fjords, the tidal effect can cause currents of up to 3 m/s [15].

The effect of a small current has to be considered also in structured environments such as a pool. In this case, the refresh of the water is strong enough to affect the vehicle dynamics [29].

Let us assume that the ocean current, expressed in the inertial frame, $\boldsymbol{\nu}_c^I$ is constant and irrotational, i.e.,

$$\boldsymbol{\nu}_c^I = \begin{bmatrix} \nu_{c,x} \\ \nu_{c,y} \\ \nu_{c,z} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and $\dot{\boldsymbol{\nu}}_c^I = \boldsymbol{0}$; its effects can be added to the dynamic of a rigid body moving in a fluid simply considering the *relative* velocity in body-fixed frame

$$\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{R}_I^B \boldsymbol{\nu}_c^I \tag{2.44}$$

in the derivation of the Coriolis and centripetal terms and the damping terms.

A simplified modeling of the current effect can be obtained by assuming the current irrotational and constant in the earth-fixed frame, its effect on the vehicle, thus, can be modeled as a constant disturbance in the earth-fixed frame that is further projected onto the vehicle-fixed frame. To this purpose, let define as $\boldsymbol{\theta}_{v,C} \in \mathbb{R}^6$ the vector of constant parameters contributing to the earth-fixed generalized forces due to the current; then, the vehicle-fixed current disturbance can be modelled as

$$\boldsymbol{\tau}_{v,C} = \boldsymbol{\Phi}_{v,C}(\boldsymbol{R}_B^I)\boldsymbol{\theta}_{v,C}, \tag{2.45}$$

where the (6×6) regressor matrix simply expresses the force/moment coordinate transformation between the two frames and it is given by

$$\boldsymbol{\Phi}_{v,C}(\boldsymbol{R}_B^I) = \begin{bmatrix} \boldsymbol{R}_I^B & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{R}_I^B \end{bmatrix}. \tag{2.46}$$

Notice that in [30, 31] compensation of the ocean current effects is obtained through a quaternion-based velocity/force mapping instead. Moreover, in some papers [10, 29, 31, 32], the effect of the current is simply modeled as a time-varying, vehicle-fixed, disturbance $\boldsymbol{\tau}_{v,C}$ that would lead to the trivial regressor

$$\boldsymbol{\Phi}'_{v,C} = \boldsymbol{I}_6. \tag{2.47}$$

2.5 Gravity and Buoyancy

"Ses deux mains s'accrochaient à mon cou; elles ne se seraient pas accrochées plus furieusement dans un naufrage. Et je ne comprenais pas si elle voulait que je la sauve, ou bien que je me noie avec elle".

Raymond Radiguet, "Le diable au corps" 1923.

When a rigid body is submerged in a fluid under the effect of the gravity two more forces have to be considered: the gravitational force and the buoyancy. The latter is the only hydrostatic effect, i.e., it is not function of a relative movement between body and fluid.

Let us define as

$$\boldsymbol{g}^I = \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} \text{m/s}^2$$

the acceleration of gravity, ∇ the volume of the body and m its mass. The submerged weight of the body is defined as $W = m \left\| \boldsymbol{g}^I \right\|$ while its buoyancy $B = \rho \nabla \left\| \boldsymbol{g}^I \right\|$.

The gravity force, acting in the center of mass $\boldsymbol{r}_C^B$ is represented in body-fixed frame by:

$$\boldsymbol{f}_G(\boldsymbol{R}_I^B) = \boldsymbol{R}_I^B \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix},$$

while the buoyancy force, acting in the center of buoyancy $\boldsymbol{r}_B^B$ is represented in body-fixed frame by:

$$\boldsymbol{f}_B(\boldsymbol{R}_I^B) = -\boldsymbol{R}_I^B \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix}.$$

The (6×1) vector of force/moment due to gravity and buoyancy in body-fixed frame, included in the left hand-side of the equations of motion, is represented by:

$$\boldsymbol{g}_{RB}(\boldsymbol{R}_I^B) = -\begin{bmatrix} \boldsymbol{f}_G(\boldsymbol{R}_I^B) + \boldsymbol{f}_B(\boldsymbol{R}_I^B) \\ \boldsymbol{r}_G^B \times \boldsymbol{f}_G(\boldsymbol{R}_I^B) + \boldsymbol{r}_B^B \times \boldsymbol{f}_B(\boldsymbol{R}_I^B) \end{bmatrix}.$$

In the following, the symbol $\boldsymbol{r}_G^B = \begin{bmatrix} x_G \; y_G \; z_G \end{bmatrix}^{\mathrm{T}}$ (with $\boldsymbol{r}_G^B = \boldsymbol{r}_C^B$) will be used for the center of gravity.

The expression of $\boldsymbol{g}_{RB}$ in terms of Euler angles is represented by:

$$\boldsymbol{g}_{RB}(\boldsymbol{\eta}_2) = \begin{bmatrix} (W-B)s_\theta \\ -(W-B)c_\theta s_\phi \\ -(W-B)c_\theta c_\phi \\ -(y_G W - y_B B)c_\theta c_\phi + (z_G W - z_B B)c_\theta s_\phi \\ (z_G W - z_B B)s_\theta + (x_G W - x_B B)c_\theta c_\phi \\ -(x_G W - x_B B)c_\theta s_\phi - (y_G W - y_B B)s_\theta \end{bmatrix}, \tag{2.48}$$

while in terms of quaternion is represented by:

$$\boldsymbol{g}_{RB}(\mathcal{Q}) = \begin{bmatrix} 2(\eta\varepsilon_2 - \varepsilon_1\varepsilon_3)(W-B) \\ -2(\eta\varepsilon_1 + \varepsilon_2\varepsilon_3)(W-B) \\ (-\eta^2 + \varepsilon_1^2 + \varepsilon_2^2 - \varepsilon_3^2)(W-B) \\ (-\eta^2 + \varepsilon_1^2 + \varepsilon_2^2 - \varepsilon_3^2)(y_G W - y_B B) + 2(\eta\varepsilon_1 + \varepsilon_2\varepsilon_3)(z_G W - z_B B) \\ -(-\eta^2 + \varepsilon_1^2 + \varepsilon_2^2 - \varepsilon_3^2)(x_G W - x_B B) + 2(\eta\varepsilon_2 - \varepsilon_1\varepsilon_3)(z_G W - z_B B) \\ -2(\eta\varepsilon_1 + \varepsilon_2\varepsilon_3)(x_G W - x_B B) - 2(\eta\varepsilon_2 - \varepsilon_1\varepsilon_3)(y_G W - y_B B) \end{bmatrix}.$$

By looking at (2.48), it can be recognized that the difference between gravity and buoyancy $(W - B)$ only affects the linear force acting on the vehicle; it is also clear that the restoring linear force is constant in the earth-fixed frame. On the other hand, the two vectors of the first moment of inertia $W\boldsymbol{r}_G^B$ and $B\boldsymbol{r}_B^B$ affect the moment acting on the vehicle and are constant in the vehicle-fixed frame. In summary, the expression of the restoring vector is linear with respect to the vector of four constant parameters

$$\boldsymbol{\theta}_{v,R} = \left[W-B \; x_G W - x_B B \; y_G W - y_B B \; z_G W - z_B B \right]^{\mathrm{T}} \tag{2.49}$$

through the (6×4) regressor

$$\boldsymbol{\Phi}_{v,R}(\boldsymbol{R}_B^I) = \begin{bmatrix} \boldsymbol{R}_I^B \boldsymbol{z} & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{S}\left(\boldsymbol{R}_I^B \boldsymbol{z}\right) \end{bmatrix}, \tag{2.50}$$

i.e.,

$$\boldsymbol{g}_{RB}(\boldsymbol{R}_B^I) = \boldsymbol{\Phi}_{v,R}(\boldsymbol{R}_B^I)\boldsymbol{\theta}_{v,R}.$$

In (2.50) $\boldsymbol{S}(\cdot)$ is the operator performing the cross product. Notice that, alternatively to (2.48), the restoring vector can be written in terms of quaternions; however, this would lead again to the regressor (2.50) and to the vector of dynamic parameters (2.49).

2.6 Thrusters' Dynamics

Underwater vehicles are usually controlled by thrusters (Fig. 2.2) and/or control surfaces.

Control surfaces, such as rudders and sterns, are common in cruise vehicles; those are torpedo-shaped and usually used in cable/pipeline inspection. Since the

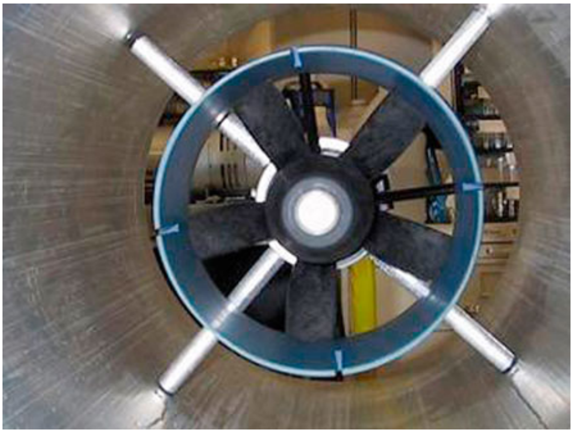


Fig. 2.2 Thruster of SAUVIM (courtesy of J. Yuh, Autonomous Systems Laboratory, University of Hawaii)

force/moment provided by the control surfaces is function of the velocity and it is null in hovering, they are not useful to manipulation missions in which, due to the manipulator interaction, full control of the vehicle is required.

The relationship between the force/moment acting on the vehicle $\boldsymbol{\tau}_v \in \mathbb{R}^6$ and the control input of the thrusters $\boldsymbol{u}_v \in \mathbb{R}^{p_v}$ is highly nonlinear. It is function of some structural variables such as: the density of the water; the tunnel cross-sectional area; the tunnel length; the volumetric flowrate between input-output of the thrusters and the propeller diameter. The state of the dynamic system describing the thrusters is constituted by the propeller revolution, the speed of the fluid going into the propeller and the input torque.

A detailed theoretical and experimental analysis of thrusters' behavior can be found in [24, 33–39]. Roughly speaking, thrusters are the main cause of limit cycle in vehicle positioning and bandwidth constraint.

A common simplification is to consider a linear relationship between $\boldsymbol{\tau}_v$ and $\boldsymbol{u}_v$:

$$\boldsymbol{\tau}_v = \boldsymbol{B}_v \boldsymbol{u}_v, \tag{2.51}$$

where $\boldsymbol{B}_v \in \mathbb{R}^{6 \times p_v}$ is a known constant matrix known as the Thruster Control Matrix (TCM). Along the book, the matrix $\boldsymbol{B}_v$ will be considered square or low rectangular, i.e., $p_v \geq 6$. This means full control of force/moments of the vehicle.

As an example, ODIN has the following TCM:

$$\boldsymbol{B}_v = \begin{bmatrix} * & * & * & * & 0 & 0 & 0 & 0 \\ * & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * \\ * & * & * & * & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.52}$$

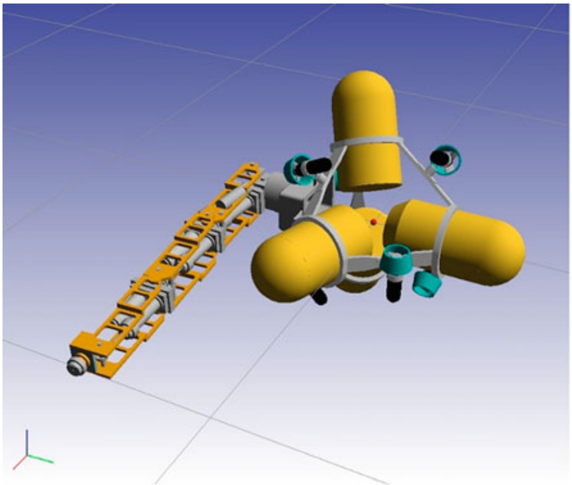


Fig. 2.3 Underwater vehicle-manipulator system PETASUS (courtesy of Wan Kyun Chung and Jonghui Han, Pohang University of Science and Technology)

where $*$ means a non-zero constant factor depending on the thruster allocation. Different TCM can be observed as in, e.g., the vehicle Phantom S3 manufactured by Deep Ocean Engineering that has four thrusters:

$$\boldsymbol{B}_v = \begin{bmatrix} * & * & 0 & 0 \\ 0 & 0 & * & * \\ 0 & 0 & * & * \\ 0 & 0 & * & * \\ * & * & 0 & 0 \\ * & * & 0 & 0 \end{bmatrix} \tag{2.53}$$

in which it can be recognized that not all the directions are independently actuated. The vehicle PETASUS [40] exhibits an interesting TCM, too as visible in Fig. 2.3.

On the other hand, if the vehicle is controlled by thrusters, each of which is locally fed back, the effects of the nonlinearities discussed above is very limited and a linear input-output relation between desired force/moment and thruster's torque is experienced. This is the case, e.g., of ODIN [31, 41, 42] where the experimental results show that the linear approximation is reliable.

2.7 Underwater Vehicles' Dynamics in Matrix Form

By taking into account the inertial generalized forces, the hydrodynamic effects, the gravity and buoyancy contribution and the thrusters' presence, it is possible to write the equations of motion of an underwater vehicle in matrix form:

$$\boldsymbol{M}_v \dot{\boldsymbol{\nu}} + \boldsymbol{C}_v(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}_{RB}(\boldsymbol{R}_B^I) = \boldsymbol{B}_v \boldsymbol{u}_v, \tag{2.54}$$

where $\boldsymbol{M}_v = \boldsymbol{M}_{RB} + \boldsymbol{M}_A$ and $\boldsymbol{C}_v = \boldsymbol{C}_{RB} + \boldsymbol{C}_A$ include also the added mass terms. Taking into account the current, a possible, approximated, model is given by:

$$\boldsymbol{M}_v \dot{\boldsymbol{v}} + \boldsymbol{C}_v(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}_{RB}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}_{RB}(\boldsymbol{R}_B^I) = \boldsymbol{\tau}_v - \boldsymbol{\tau}_{v,C}. \tag{2.55}$$

The following properties hold:

- the inertia matrix is symmetric and positive definite, i.e., $\boldsymbol{M}_v = \boldsymbol{M}_v^{\mathrm{T}} > \boldsymbol{O}$;
- the damping matrix is positive definite, i.e., $\boldsymbol{D}_{RB}(\boldsymbol{v}) > \boldsymbol{O}$;
- the matrix $\boldsymbol{C}_v(\boldsymbol{v})$ is skew-symmetric, i.e., $\boldsymbol{C}_v(\boldsymbol{v}) = -\boldsymbol{C}_v^{\mathrm{T}}(\boldsymbol{v}), \forall\, \boldsymbol{v} \in \mathbb{R}^6$.

It is possible to rewrite the dynamic model (2.54) in terms of earth-fixed coordinates; in this case, the state variables are the (6×1) vectors $\boldsymbol{\eta}$, $\dot{\boldsymbol{\eta}}$ and $\ddot{\boldsymbol{\eta}}$. The equations of motion are then obtained, through the kinematic relations (2.1)–(2.2) as

$$\boldsymbol{M}_v^{\star}(\boldsymbol{R}_B^I)\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_v^{\star}(\boldsymbol{R}_B^I, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \boldsymbol{D}_{RB}^{\star}(\boldsymbol{R}_B^I, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \boldsymbol{g}_{RB}^{\star}(\boldsymbol{R}_B^I) = \boldsymbol{\tau}_v^{\star}, \tag{2.56}$$

where [15]

$$\begin{aligned}
\boldsymbol{M}_v^{\star} &= \boldsymbol{J}_e^{-\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{M}_v\boldsymbol{J}_e^{-1}(\boldsymbol{R}_B^I)\\
\boldsymbol{C}_v^{\star} &= \boldsymbol{J}_e^{-\mathrm{T}}(\boldsymbol{R}_B^I)\left(\boldsymbol{C}_v(\boldsymbol{v}) - \boldsymbol{M}_v\boldsymbol{J}_e^{-1}(\boldsymbol{R}_B^I)\dot{\boldsymbol{J}}(\boldsymbol{R}_B^I)\right)\boldsymbol{J}_e^{-1}(\boldsymbol{R}_B^I)\\
\boldsymbol{D}_{RB}^{\star} &= \boldsymbol{J}_e^{-\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{D}_{RB}(\boldsymbol{v})\boldsymbol{J}_e^{-1}(\boldsymbol{R}_B^I)\\
\boldsymbol{g}_{RB}^{\star} &= \boldsymbol{J}_e^{-\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{g}_{RB}(\boldsymbol{R}_B^I)\\
\boldsymbol{\tau}_v^{\star} &= \boldsymbol{J}_e^{-\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{\tau}_v.
\end{aligned}$$

Again, the current can be taken into account by resorting to the relative velocity or, introducing an approximation, considering the following equations of motion:

$$\boldsymbol{M}_v^{\star}(\boldsymbol{R}_B^I)\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_v^{\star}(\boldsymbol{R}_B^I, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \boldsymbol{D}_{RB}^{\star}(\boldsymbol{R}_B^I, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \boldsymbol{g}_{RB}^{\star}(\boldsymbol{R}_B^I) = \boldsymbol{\tau}_v^{\star} - \boldsymbol{\tau}_{v,C}^{\star},$$

where $\boldsymbol{\tau}_{v,C}^{\star} \in \mathbb{R}^6$ is the disturbance introduced by the current. It is worth noticing that the earth-fixed and the body-fixed models with the introduction of the current as a simple external disturbance implies different dynamic properties. In particular, this is true if, in case of the design of a control action, the disturbance is considered as constant or slowly varying.

2.7.1 Linearity in the Parameters

Relation (2.54) can be written by exploiting the linearity in the parameters property. It must be noted that, while this property is proved for rigid bodies moving in the space [18], for underwater rigid bodies it depends on a suitable representations of the hydrodynamics terms. With a vector of parameters $\boldsymbol{\theta}_v$ of proper dimension it is possible to write the following:

$$\boldsymbol{\Phi}_v(\boldsymbol{R}_B^I, \boldsymbol{\nu}, \dot{\boldsymbol{\nu}})\boldsymbol{\theta}_v = \boldsymbol{\tau}_v. \tag{2.57}$$

The inclusion of the ocean current is straightforward by using the relative velocity as shown in Sect. 2.4.3. However, it might be useful to consider also the regressor form of the two approximations given by considering the current as an external disturbance. In particular, it is of interest to isolate the contribution of the restoring forces and current effects, those are the sole terms giving a non-null contribution to the dynamic with the vehicle still and for this reason will be defined as *persistent dynamic terms*.

Starting from the Eq. (2.55) let first consider the current as an external disturbance $\boldsymbol{\tau}_{v,C}$ constant in the body-fixed frame, it is possible to write:

$$\boldsymbol{M}_v\dot{\boldsymbol{\nu}} + \boldsymbol{C}_v(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\Phi}_{v,R}(\boldsymbol{R}_B^I)\boldsymbol{\theta}_{v,R} + \boldsymbol{\Phi}'_{v,C}\boldsymbol{\theta}_{v,C} = \boldsymbol{\tau}_v$$

that can be rewritten as:

$$\boxed{\boldsymbol{M}_v\dot{\boldsymbol{\nu}} + \boldsymbol{C}_v(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\Phi}_{v,P'}(\boldsymbol{R}_B^I)\boldsymbol{\theta}_{v,P} = \boldsymbol{\tau}_v} \tag{2.58}$$

with the use of the (6 × 10) regressor:

$$\boldsymbol{\Phi}_{v,P'}(\boldsymbol{R}_B^I) = \begin{bmatrix} \boldsymbol{R}_I^B\boldsymbol{z} & \boldsymbol{O}_{3\times 3} & \boldsymbol{I}_3 & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{S}\left(\boldsymbol{R}_I^B\boldsymbol{z}\right) & \boldsymbol{O}_{3\times 3} & \boldsymbol{I}_3 \end{bmatrix}.$$

On the other side the current can be modeled as constant in the earth-fixed frame and, merged again with the restoring forces contribution, gives the following

$$\boxed{\boldsymbol{M}_v\dot{\boldsymbol{\nu}} + \boldsymbol{C}_v(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\Phi}_{v,P}(\boldsymbol{R}_B^I)\boldsymbol{\theta}_{v,P} = \boldsymbol{\tau}_v} \tag{2.59}$$

with the use of the (6 × 9) regressor:

$$\boldsymbol{\Phi}_{v,P}(\boldsymbol{R}_B^I) = \begin{bmatrix} \boldsymbol{O}_{3\times 3} & \boldsymbol{R}_I^B & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{S}\left(\boldsymbol{R}_I^B\boldsymbol{z}\right) & \boldsymbol{O}_{3\times 3} & \boldsymbol{R}_I^B \end{bmatrix}.$$

It is worth noticing that the two regressors have different dimensions. In order to extrapolate the minimum number of independent parameters, i.e., the number of columns of the regressor, it is possible to resort to the numerical method proposed by Gautier [43] based on the Singular Value Decomposition.

Model (2.59) can by rewritten in a sole regressor of proper dimension yielding:

$$\boldsymbol{\Phi}_{v,T}(\boldsymbol{R}_B^I, \boldsymbol{\nu}, \dot{\boldsymbol{\nu}})\boldsymbol{\theta}_{v,T} = \boldsymbol{\tau}_v. \tag{2.60}$$

2.8 Common AUV Designs

Depending on the purpose a vehicle is conceived for, several, different designs have been proposed along the last decades. Vehicles aimed at surving, exploring, monitoring needs to operate for almost of their life-time at a velocity different from zero, a low-drag profile is thus necessary and one main thruster with control surfaces preferred with respect to a thruster-only actuation. When the vehicle works for most of its time at low velocity, control surfaces are not usefull any more neither it is crucial to exhibits a torpedo-like profile, in such a case a *box* is built with easy access to the components and several thrusters to control most of its DOFs. In long duration mission the energy becomes the main constraint to satisfies, specific design has been implemented to achieved the so-called gliders [44], i.e., vehicles that exploits the gravity-buoyancy forces to achieve a slow, sawtooth-like movement. Recent models achieves 6 months or 5000 km missions. Figure 2.4 reports samples for the cited designs. In addition, body shaped as oblate or tear or vehiucles conceived according to a biomimetic inspiration such as fish, jellyfish or crawler can be find.

In the following, we will assume that the the vehicle acts as a fully-actuated base for the manipulator.

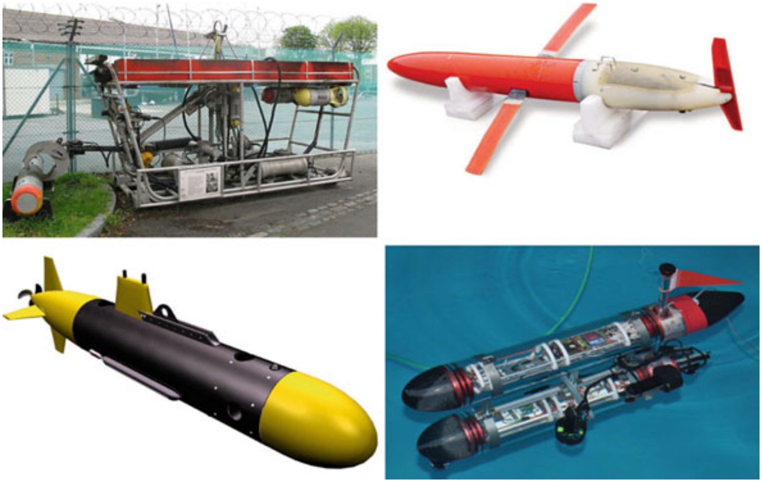


Fig. 2.4 Common designs for AUVs. *Top-left* A Royal Navy ROV (Cutlet) first used in the 1950s to retrieve practice torpedoes and mines. *Top-right* Spray glider (courtesy of BlueFin Robotics). *Bottom-left* SeaCat (courtesy of Atlas Elektronik). *Bottom-right* Medusa (Courtesy of Institute for Systems and Robotics/Instituto Superior Técnico). Not all those designs are suitable for intervention purposes

2.9 Kinematics of Manipulators with Mobile Base

In robotics, *direct* kinematics is the process of computing the end-effector position orientation by knowing the n joint positions $\boldsymbol{q} \in \mathbb{R}^n$. For floating base manipulators the use of the vehicle configuration is also required.

Let us define the position of the end effector in the inertial frame, $\boldsymbol{\eta}_{ee1} \in \mathbb{R}^3$; this is a function of the system configuration, i.e., $\boldsymbol{\eta}_{ee1}(\boldsymbol{\eta}_1, \boldsymbol{R}_I^B, \boldsymbol{q})$. Let us further define $\boldsymbol{\eta}_{ee2} \in \mathbb{R}^3$ as the orientation of the end effector in the inertial frame expressed by Euler angles: also $\boldsymbol{\eta}_{ee2}$ is a function of the system configuration, i.e., $\boldsymbol{\eta}_{ee2}(\boldsymbol{R}_I^B, \boldsymbol{q})$. The relation between the end-effector posture

$$\boldsymbol{\eta}_{ee} = \begin{bmatrix} \boldsymbol{\eta}_{ee1} \\ \boldsymbol{\eta}_{ee2} \end{bmatrix} \in \mathbb{R}^6$$

and the system configuration can be expressed by the following nonlinear equation:

$$\boldsymbol{\eta}_{ee} = \boldsymbol{k}(\boldsymbol{\eta}, \boldsymbol{q}). \tag{2.61}$$

In the following, few details about the function $\boldsymbol{k}$ are given. The end-effector configuration is usually obtained by first placing a frame attached to each degrees of freedom of the articulated structure. By concatenating the various rotations and translations the end-effector configuration is computed. The robotics community developed some systematic procedures to obtain a common procedure to place the frames and compute $\boldsymbol{\eta}_{ee}$. One well known procedure is known as the Denavit-Hartenberg (DH) convention [18] that, for a n-link manipulator, gives the *rules* to place $n + 1$ frames, i.e., one for each link plus a *zero*, inertial, frame. Following the procedure allows to represent the transformation between two consecutive frames according to 4 parameters, known as Denavit-Hartenberg parameters, of course…, and to build *systematically* the homogeneous transformation matrices between two consecutive frames (please refer still to [18] for definitions and further details on those concepts). By the DH convention the definition of the joint position is univoque and it will be defined as $\boldsymbol{q} \in \mathbb{R}^n$ where n is the number of joints.

In a manipulator with floating base the zero frame following the Denavit-Hartenberg procedure is attached to the vehicle, and not interial of course, in a position that depends on the manipulator itself. From the considerations made in the previous sections, however, it should be clear that the vehicle-fixed frame is commonly placed coincident with the center of mass. We took the decision to not move the vehicle-fixed frame from it *natural* position and to have two frames both attached to the vehicle.

Figure 2.5 represents a sketch of an Underwater Vehicle-Manipulator System with relevant frames attached as discussed.

The following script `DirectKinematics.m` is available in SIMURV4.0 to compute the direct kinematics:

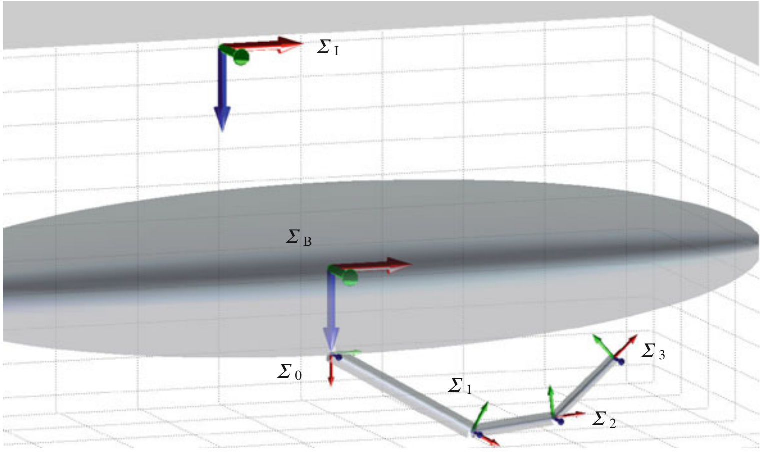


Fig. 2.5 Sketch of an Underwater Vehicle-Manipulator System with relevant frames for $n = 3$, notice that two frames Σ_B and Σ_0 are both attached to the vehicle

```
function T = DirectKinematics(eta,DH,T_B_0)
%
% Computes the homogeneous transformation matrix
%   from intertial frame to end-effector
%
% function T = DirectKinematics(eta,DH,T_B_0)
%
% input:
%        eta     dim 6x1     vehicle position/
   orientation
%        DH      dim nx4     Denavit-Hartenberg table
%        T_B_0   dim 4x4     Homogeneous transformation
%                 matrix from vehicle to zero frame
%
% output:
%        T       dim 4x4     Homogeneous transformation
%             matrix from inertial to end-effector
```

In `Homogeneous_dh.m` the homogeneous transformation connecting to consecutive links is performed.

2.10 Differential Kinematics of Manipulators with Mobile Base

Let define $\boldsymbol{\zeta} \in \mathbb{R}^{6+n}$ as

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \dot{\boldsymbol{q}} \end{bmatrix} \tag{2.62}$$

where $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ is the time derivative of the joint positions, i.e., the joint velocities. It is useful to rewrite the relationship between body-fixed and earth-fixed velocities given in Eqs. (2.1)–(2.2) in a more compact form:

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_I^B & \boldsymbol{O}_{3\times 3} & \boldsymbol{O}_{3\times n} \\ \boldsymbol{O}_{3\times 3} & \boldsymbol{J}_{k,o}(\boldsymbol{R}_I^B) & \boldsymbol{O}_{3\times n} \\ \boldsymbol{O}_{n\times 3} & \boldsymbol{O}_{n\times 3} & \boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \\ \dot{\boldsymbol{q}} \end{bmatrix} = \boldsymbol{J}_k \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \\ \dot{\boldsymbol{q}} \end{bmatrix}, \tag{2.63}$$

where $\boldsymbol{O}_{n_1\times n_2}$ is the null $(n_1 \times n_2)$ matrix and the matrix $\boldsymbol{J}_{k,o}(\boldsymbol{R}_I^B)$ in terms of Euler angles has been introduced in (2.3).

Knowing $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$, $\dot{\boldsymbol{\nu}}_1$, $\dot{\boldsymbol{\nu}}_2$, (vehicle linear and angular velocities and acceleration in body fixed frame), $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, (joint velocities and acceleration) it is possible to calculate, for every link, the following variables:

$\boldsymbol{\omega}_i^i$, angular velocity of the frame i,
$\dot{\boldsymbol{\omega}}_i^i$, angular acceleration of the frame i,
$\boldsymbol{v}_i^i$, linear velocity of the origin of the frame i,
$\boldsymbol{v}_{ic}^i$, linear velocity of the center of mass of link i,
$\boldsymbol{a}_i^i$, linear acceleration of the origin of frame i,

by resorting to the following relationships:

$$\boldsymbol{\omega}_i^i = \boldsymbol{R}_{i-1}^i \left(\boldsymbol{\omega}_{i-1}^{i-1} + \dot{q}_i z_{i-1}\right) \tag{2.64}$$

$$\dot{\boldsymbol{\omega}}_i^i = \boldsymbol{R}_{i-1}^i \left(\dot{\boldsymbol{\omega}}_{i-1}^{i-1} + \boldsymbol{\omega}_{i-1}^{i-1} \times \dot{q}_i z_{i-1} + \ddot{q}_i z_{i-1}\right) \tag{2.65}$$

$$\boldsymbol{v}_i^i = \boldsymbol{R}_{i-1}^i \boldsymbol{v}_{i-1}^{i-1} + \boldsymbol{\omega}_i^i \times \boldsymbol{r}_{i-1,i}^i \tag{2.66}$$

$$\boldsymbol{v}_{ic}^i = \boldsymbol{R}_{i-1}^i \boldsymbol{v}_{i-1}^{i-1} + \boldsymbol{\omega}_i^i \times \boldsymbol{r}_{i-1,c}^i \tag{2.67}$$

$$\boldsymbol{a}_i^i = \boldsymbol{R}_{i-1}^i \boldsymbol{a}_{i-1}^{i-1} + \dot{\boldsymbol{\omega}}_i^i \times \boldsymbol{r}_{i-1,i}^i + \boldsymbol{\omega}_i^i \times (\boldsymbol{\omega}_i^i \times \boldsymbol{r}_{i-1,i}^i) \tag{2.68}$$

where z_i is the versor of frame i, $\boldsymbol{r}_{i-1,i}^i$ is the constant vector from the origin of frame $i-1$ toward the origin of frame i expressed in frame i.

The vectors $\dot{\boldsymbol{\eta}}_{ee1}$ and $\dot{\boldsymbol{\eta}}_{ee2}$, time derivative of $\boldsymbol{\eta}_{ee1}$ and $\boldsymbol{\eta}_{ee2}$, are related to the body-fixed velocities $\boldsymbol{\nu}_{ee} = \begin{bmatrix} \boldsymbol{\nu}_{ee1} & \boldsymbol{\nu}_{ee2} \end{bmatrix}^{\mathrm{T}}$ via relations analogous to (2.1) and (2.2), i.e.,

$$\boldsymbol{\nu}_{ee1} = \boldsymbol{R}_I^n \dot{\boldsymbol{\eta}}_{ee1} \tag{2.69}$$

$$\boldsymbol{\nu}_{ee2} = \boldsymbol{J}_{k,o}(\boldsymbol{\eta}_{ee2}) \dot{\boldsymbol{\eta}}_{ee2} \tag{2.70}$$

where $\boldsymbol{R}_I^n$ is the rotation matrix from the inertial frame to the end-effector frame (i.e., frame n) and $\boldsymbol{J}_{k,o}$ is the matrix defined as in (2.3) with the use of the Euler angles of the end-effector frame. If the end-effector orientation is expressed via quaternion the relation between end-effector angular velocity and time derivative of the quaternion can be easily obtained by the *quaternion propagation equation* (2.10).

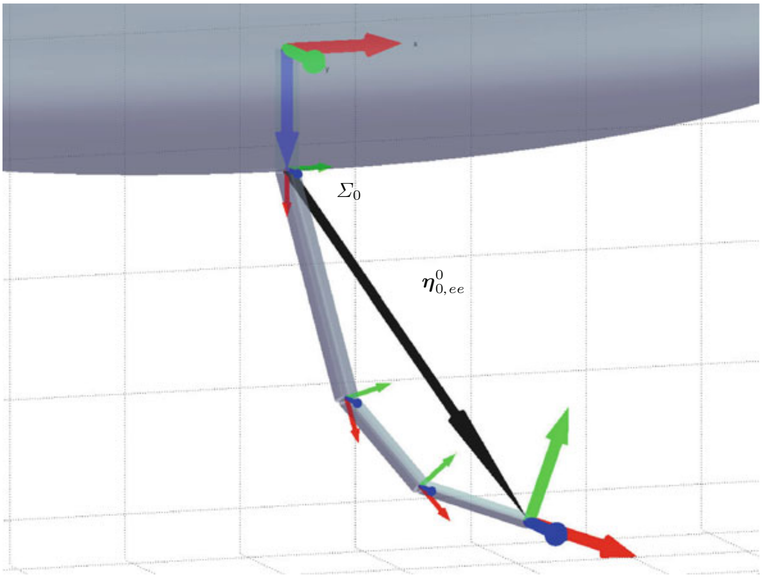


Fig. 2.6 Vector connecting the origin of the zero frame with the end-effector frame expressed in the zero frame

The end-effector linear velocity (expressed in the inertial frame) is related to the body-fixed system velocity by a suitable Jacobian matrix, i.e.,

$$\dot{\boldsymbol{\eta}}_{ee1} = \boldsymbol{J}_{pos}(\boldsymbol{R}^I_B, \boldsymbol{q})\boldsymbol{\zeta}\,. \tag{2.71}$$

The relation above may be computed starting from (2.61), rewritten as

$$\boldsymbol{\eta}_{ee1} = \boldsymbol{\eta}_0 + \boldsymbol{R}^I_0 \boldsymbol{\eta}^0_{0,ee}$$

where $\boldsymbol{\eta}_0 \in \mathbb{R}^3$ is the position of the frame 0 and $\boldsymbol{\eta}^0_{0,ee} \in \mathbb{R}^3$ is the vector connecting the origin of the zero frame with the end-effector frame expressed in the zero frame as shown in Fig. 2.6.

By differentation and using the equations of velocity composition we obtain:

$$\dot{\boldsymbol{\eta}}_{ee1} = \dot{\boldsymbol{\eta}}_0 - \boldsymbol{S}(\boldsymbol{R}^I_0 \boldsymbol{\eta}^0_{0,ee})\boldsymbol{\omega}_0 + \boldsymbol{R}^I_0 \dot{\boldsymbol{\eta}}^0_{0,ee}$$

where the term $\dot{\boldsymbol{\eta}}^0_{0,ee}$ is appealing since it represents the *standard* positional geometric Jacobian $\boldsymbol{J}^0_{pos,man} \in \mathbb{R}^{3\times n}$ computed resorting to the equations that can be found, e.g., in [18] obtained in agreement with the DH convention $\dot{\boldsymbol{\eta}}^0_{0,ee} = \boldsymbol{J}^0_{pos,man}\dot{\boldsymbol{q}}$. Let us define

$$\boldsymbol{J}_{pos,man} = \boldsymbol{R}^I_0 \boldsymbol{J}^0_{pos,man} \in \mathbb{R}^{3\times n}$$

and rewrite $\dot{\boldsymbol{\eta}}_{ee1}$:

$$\dot{\boldsymbol{\eta}}_{ee1} = \dot{\boldsymbol{\eta}}_0 - \boldsymbol{S}(\boldsymbol{R}_0^I \boldsymbol{\eta}_{0,ee}^0)\boldsymbol{\omega}_0 + \boldsymbol{J}_{pos,man}\dot{\boldsymbol{q}}.$$

By further observing that $\boldsymbol{\omega}_0 = \boldsymbol{R}_0^I \boldsymbol{\nu}_2$ and

$$\dot{\boldsymbol{\eta}}_0 = \boldsymbol{R}_B^I \boldsymbol{\nu}_1 - \boldsymbol{S}(\boldsymbol{R}_B^I \boldsymbol{r}_{B0}^B)\boldsymbol{R}_B^I \boldsymbol{\nu}_2$$

we have finally rewritten $\dot{\boldsymbol{\eta}}_0$ in terms of $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$ and $\dot{\boldsymbol{q}}$ by means of the configuration dependent positional Jacobian:

$$\dot{\boldsymbol{\eta}}_{ee1} = \boldsymbol{J}_{pos}(\boldsymbol{R}_B^I, \boldsymbol{q})\boldsymbol{\zeta}$$

where

$$\boldsymbol{J}_{pos}(\boldsymbol{R}_B^I, \boldsymbol{q}) = \left[\boldsymbol{R}_B^I \; -\Big(\boldsymbol{S}(\boldsymbol{R}_B^I \boldsymbol{r}_{B0}^B) + \boldsymbol{S}(\boldsymbol{R}_0^I \boldsymbol{\eta}_{0,ee}^0)\Big)\boldsymbol{R}_B^I \; \boldsymbol{J}_{pos,man}\right] \in \mathbb{R}^{3\times 6+n}.$$

For the orientation we start from the angular velocity of the end effector expressed in the inertial frame written with respect to the angular velocities of the vehicle (frame zero in this case) and the angular velocity of the manipulator with respect to the zero frame:

$$\boldsymbol{\omega}_{ee} = \boldsymbol{\omega}_0 + \boldsymbol{R}_0^I \boldsymbol{\omega}_{0,ee}^0$$

that can be rewritten as

$$\boldsymbol{\omega}_{ee} = \boldsymbol{R}_B^I \boldsymbol{\nu}_2 + \boldsymbol{R}_0^I \boldsymbol{J}_{or,man}^0 \dot{\boldsymbol{q}}$$

where, again, the *standard*, DH-compliant, orientation Jacobian $\boldsymbol{J}_{or,man}^0 \in \mathbb{R}^{3\times n}$ appears. By defining:

$$\boldsymbol{J}_{or,man} = \boldsymbol{R}_0^I \boldsymbol{J}_{or,man}^0 \in \mathbb{R}^{3\times n}$$

we finally obtain

$$\boldsymbol{\omega}_{ee} = \boldsymbol{J}_{or}(\boldsymbol{R}_B^I, \boldsymbol{q})\boldsymbol{\zeta}$$

where

$$\boldsymbol{J}_{or}(\boldsymbol{R}_B^I, \boldsymbol{q}) = \left[\boldsymbol{O}_{3\times 3} \; \boldsymbol{R}_B^I \; \boldsymbol{J}_{or,man}\right] \in \mathbb{R}^{3\times 6+n}. \tag{2.72}$$

Finally, the following compact expression is achieved:

$$\dot{\boldsymbol{x}}_E = \begin{bmatrix} \dot{\boldsymbol{\eta}}_{ee1} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{R}_B^I, \boldsymbol{q})\boldsymbol{\zeta} \tag{2.73}$$

where

$$\boldsymbol{J}(\boldsymbol{R}_B^I, \boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{pos}(\boldsymbol{R}_B^I, \boldsymbol{q}) \\ \boldsymbol{J}_{or}(\boldsymbol{R}_B^I, \boldsymbol{q}) \end{bmatrix} \in \mathbb{R}^{6\times 6+n}.$$

The use of the end-effector velocities $\dot{\boldsymbol{\eta}}_{ee1}$ and $\boldsymbol{\omega}_{ee}$ instead of $\boldsymbol{\nu}_{ee1}$ and $\boldsymbol{\nu}_{ee2}$ defined in (2.69)–(2.70) is due to the use of the former in the inverse kinematics algorithms.

The following script `J_man.m` is available in SIMURV4.0 to compute the manipulator Jacobian with respect to the frame 0:

```
function J = J_man(DH)
%
% Computes the geometric Jacobian for a manipulator
% with rotational-only joints with respect to
% the zero(base) frame
%
% function J = J_man(DH)
%
% input:
%       DH      dim nx4     Denavit-Hartenberg table
%
% output:
%       J       dim 6xn     Jacobian
```

The following script `Jacobian.m` is for the Jacobian in (2.73):

```
function J = Jacobian(eta,DH,T_B_0)
%
% Computes the Jacobian from zita to
% end-effector linear and angular velocities
% expressed in the inertial frame
%
% function J = Jacobian(eta,DH,T_B_0)
%
% input:
%       eta     dim 6x1    vehicle position/
   orientation
%       DH      dim nx4    Denavit-Hartenberg table
%       T_B_0   dim 4x4    Homogeneous transformation
%               matrix from vehicle to zero frame
%
% output:
%       J       dim 6x(6+n) Jacobian
```

2.11 Dynamics of Underwater Vehicle-Manipulator Systems

By knowing the forces acting on a body moving in a fluid it is possible to easily obtain the dynamics of a serial chain of rigid bodies moving in a fluid.

The inertial forces and moments acting on the generic body are represented by:

$$\begin{aligned}
\boldsymbol{F}_i^i &= \boldsymbol{M}_i[\boldsymbol{a}_i^i + \dot{\boldsymbol{\omega}}_i^i \times \boldsymbol{r}_{i,c}^i + \boldsymbol{\omega}_i^i \times (\boldsymbol{\omega}_i^i \times \boldsymbol{r}_{i,c}^i)] \\
\boldsymbol{T}_i^i &= \boldsymbol{I}_i^i \dot{\boldsymbol{\omega}}_i^i + \boldsymbol{\omega}_i^i \times (\boldsymbol{I}_i^i \boldsymbol{\omega}_i^i),
\end{aligned}$$

where $\boldsymbol{M}_i$ is the (3×3) mass matrix comprehensive of the added mass, $\boldsymbol{I}_i^i$ is the (3×3) inertia matrix plus added inertia with respect to the center of mass, $\boldsymbol{r}_{i,c}^i$ is the vector from the origin of frame i toward the center of mass of link i expressed in frame i.

Let us define $\boldsymbol{d}_i^i$ the drag and lift forces acting on the center of mass of link i, $\boldsymbol{r}_{i-1,i}^i$ the vector from the origin of frame $i-1$ to the origin of frame i expressed in frame i, $\boldsymbol{r}_{i-1,c}^i$ the vector from the origin of frame $i-1$ to the center of mass of link i expressed in frame i and $\boldsymbol{r}_{i-1,b}^i$ the vector from the origin of frame $i-1$ to the center of buoyancy of link i expressed in frame i,

$$\boldsymbol{g}^i = \boldsymbol{R}_I^i \boldsymbol{g}^I = \boldsymbol{R}_I^i \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} \quad \text{m/s}^2 .$$

The total forces and moments acting on the generic body of the serial chain are given by (Fig. 2.7):

$$\begin{aligned} \boldsymbol{f}_i^i &= \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1} + \boldsymbol{F}_i^i - m_i \boldsymbol{g}^i + \rho \nabla_i \boldsymbol{g}^i + \boldsymbol{p}_i \\ \boldsymbol{\mu}_i^i &= \boldsymbol{R}_{i+1}^i \boldsymbol{\mu}_{i+1}^{i+1} + \boldsymbol{R}_{i+1}^i \boldsymbol{r}_{i-1,i}^{i+1} \times \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1} + \boldsymbol{r}_{i-1,c}^i \times \boldsymbol{F}_i^i + \boldsymbol{T}_i^i \\ &\quad + \boldsymbol{r}_{i-1,c}^i \times (-m_i \boldsymbol{g}^i + \boldsymbol{d}_i) + \boldsymbol{r}_{i-1,b}^i \times \rho \nabla_i \boldsymbol{g}^i \end{aligned}$$

The torque acting on joint i is finally given by:

$$\tau_{q,i} = \boldsymbol{\mu}_i^{i\mathrm{T}} \boldsymbol{z}_{i-1}^i + f_{di} \mathrm{sign}(\dot{q}_i) + f_{vi} \dot{q}_i \qquad (2.74)$$

with f_{di} and f_{vi} the motor dry and viscous friction coefficients.

Let us define as $\boldsymbol{\tau}_q \in \mathbb{R}^n$

$$\boldsymbol{\tau}_q = \begin{bmatrix} \tau_{q,1} \\ \vdots \\ \tau_{q,n} \end{bmatrix} \qquad (2.75)$$

the vector of joint torques and $\boldsymbol{\tau} \in \mathbb{R}^{6+n}$

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_v \\ \boldsymbol{\tau}_q \end{bmatrix} \qquad (2.76)$$

the vector of force/moment acting on the vehicle as well as joint torques. It is possible to write the equations of motions of an UVMS in a matrix form:

$$\boxed{\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) = \boldsymbol{\tau}} \qquad (2.77)$$

where $\boldsymbol{M} \in \mathbb{R}^{(6+n)\times(6+n)}$ is the inertia matrix including added mass terms, $\boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} \in \mathbb{R}^{6+n}$ is the vector of Coriolis and centripetal terms, $\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} \in \mathbb{R}^{6+n}$ is the vector of dissipative effects, $\boldsymbol{g}(\boldsymbol{q},\boldsymbol{R}_I^B) \in \mathbb{R}^{6+n}$ is the vector of gravity and buoyancy effects. The relationship between the generalized forces $\boldsymbol{\tau}$ and the control input is given by:

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_v \\ \boldsymbol{\tau}_q \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_v & \boldsymbol{O}_{6\times n} \\ \boldsymbol{O}_{n\times 6} & \boldsymbol{I}_n \end{bmatrix} \boldsymbol{u} = \boldsymbol{B}\boldsymbol{u}, \tag{2.78}$$

where $\boldsymbol{u} \in \mathbb{R}^{p_v+n}$ is the vector of the control input. Notice that, while for the vehicle a generic number $p_v \geq 6$ of control inputs is assumed, for the manipulator it is supposed that n joint motors are available.

The following scripts is available in SIMURV4.0 for the Inverse Dynamics

```
function tau = InverseDynamics(eta2,DH,zita,dzita,
               PARAM)
%
% Computes the inverse dynamics
%
% function tau = InverseDynamics(eta2,DH,zita,dzita,
%                  PARAM)
%
% input:
%       eta2    dim 3x1      vehicle orientation
%       DH      dim nx4      Denavit-Hartenberg table
%                            (include joint pos)
%       zita    dim 6+nx1    system velocities
%       dzita   dim 6+nx1    system accelerations
%       PARAM   struct       parameters for the
   dynamic simulation
%
% output:
%       tau     dim 6+nx1    generalized forces
```

The numerical simulation, however, needs to compute the Direct Dynamics, i.e., the acceleration knowin the current state and the input generalized forces. The corresponding algorithm is illustrated in Chap. 9.

It can be proven that:

- The inertia matrix $\boldsymbol{M}$ of the system is symmetric and positive definite:

$$\boldsymbol{M} = \boldsymbol{M}^{\mathrm{T}} > \boldsymbol{O}$$

moreover, it satisfies the inequality

$$\lambda_{\min}(\boldsymbol{M}) \leq \|\boldsymbol{M}\| \leq \lambda_{\max}(\boldsymbol{M}),$$

where $\lambda_{\min}(\boldsymbol{M})$ ($\lambda_{\max}(\boldsymbol{M})$) is the minimum (maximum) eigenvalue of $\boldsymbol{M}$.

- For a suitable choice of the parametrization of $\boldsymbol{C}$ and if all the single bodies of the system are symmetric, $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew-symmetric [45]

$$\boldsymbol{\zeta}^{\mathrm{T}}\left(\dot{\boldsymbol{M}}-2\boldsymbol{C}\right)\boldsymbol{\zeta}=0$$

which implies

$$\dot{\boldsymbol{M}}=\boldsymbol{C}+\boldsymbol{C}^{\mathrm{T}}$$

moreover, the inequality

$$\|\boldsymbol{C}(\boldsymbol{a},\boldsymbol{b})\boldsymbol{c}\|\leq C_M\,\|\boldsymbol{b}\|\,\|\boldsymbol{c}\|$$

and the equality

$$\boldsymbol{C}(\boldsymbol{a},\alpha_1\boldsymbol{b}+\alpha_2\boldsymbol{c})=\alpha_1\boldsymbol{C}(\boldsymbol{a},\boldsymbol{b})+\alpha_2\boldsymbol{C}(\boldsymbol{a},\boldsymbol{c})$$

hold.

- The matrix $\boldsymbol{D}$ is positive definite

  $$\boldsymbol{D}>\boldsymbol{O}$$

  and satisfies

  $$\|\boldsymbol{D}(\boldsymbol{q},\boldsymbol{a})-\boldsymbol{D}(\boldsymbol{q},\boldsymbol{b})\|\leq D_M\,\|\boldsymbol{a}-\boldsymbol{b}\|\,.$$

In [10], it can be found the mathematical model written with respect to the earth-fixed-frame-based vehicle position and the manipulator end-effector. However, it must be noted that, in that case, a 6-dimensional manipulator is considered in order to have square Jacobian to work with; moreover, kinematic singularities need to be avoided.

Reference [5] reports some interesting dynamic considerations about the interaction between the vehicle and the manipulator. The analysis performed allows to divide the dynamics in separate meaningful terms.

2.11.1 Linearity in the Parameters

UVMS have a property that is common to most mechanical systems, e.g., serial chain manipulators: linearity in the dynamic parameters. Using a suitable mathematical model for the hydrodynamic forces, (2.77) can be rewritten in a matrix form that exploits this property:

$$\boldsymbol{\Phi}(\boldsymbol{q},\boldsymbol{R}_B^I,\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\boldsymbol{\theta}=\boldsymbol{\tau} \tag{2.79}$$

with $\boldsymbol{\Phi}\in R^{(6+n)\times n_\theta}$, being n_θ the total number of parameters. Notice that n_θ depends on the model used for the hydrodynamic generalized forces and joint friction terms. For a single rigid body the number of dynamic parameter $n_{\theta,v}$ is a number greater than 100 [15]. For an UVMS it is $n_\theta=(n+1)\cdot n_{\theta,v}$, that gives an idea of the complexity of such systems.

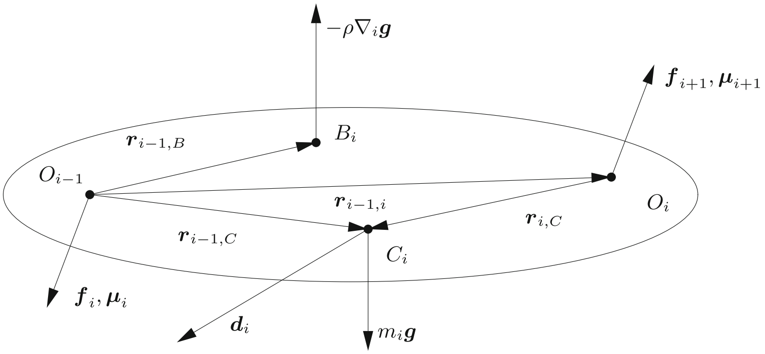


Fig. 2.7 Force/moment acting on link *i*

Differently from ground fixed manipulators, in this case the number of parameters can not be reduced because, due to the six degrees of freedom (DOFs) of the sole vehicle, all the dynamic parameters provide an individual contribution to the motion.

2.12 Contact with the Environment

If the end effector of a robotic system is in contact with the environment, the force/moment at the tip of the manipulator acts on the whole system according to the Equation [18]

$$\boxed{\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) = \boldsymbol{\tau} + \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q}, \boldsymbol{R}_B^I)\boldsymbol{h}_e,} \tag{2.80}$$

where $\boldsymbol{J}$ is the Jacobian matrix defined in (2.73) and the vector $\boldsymbol{h}_e \in \mathbb{R}^6$ is defined as

$$\boldsymbol{h}_e = \begin{bmatrix} \boldsymbol{f}_e \\ \boldsymbol{\mu}_e \end{bmatrix}$$

i.e., the vector of force/moments at the end effector expressed in the inertial frame. If it is assumed that only linear forces act on the end effector equation (2.80) becomes

$$\boxed{\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) = \boldsymbol{\tau} + \boldsymbol{J}_{pos}^{\mathrm{T}}(\boldsymbol{q}, \boldsymbol{R}_B^I)\boldsymbol{f}_e} \tag{2.81}$$

Contact between the manipulator and the environment is usually difficult to model. In the following the simple model constituted by a frictionless and elastically com-

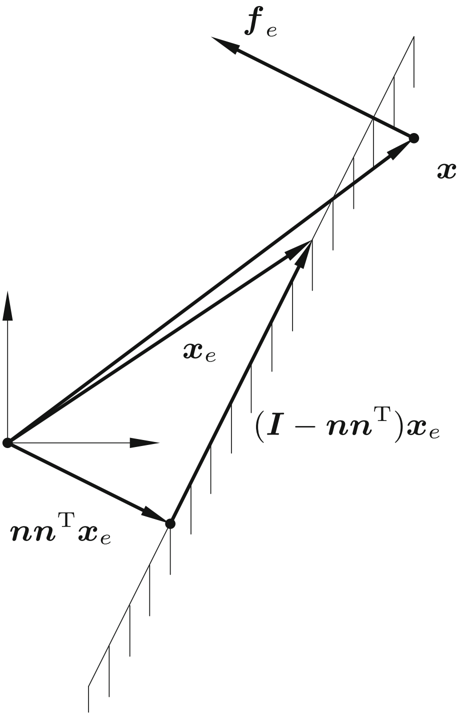

$\boldsymbol{f}_e$
$\boldsymbol{x}$
$\boldsymbol{x}_e$
$(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\boldsymbol{x}_e$
$\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{x}_e$

Fig. 2.8 Planar view of the chosen model for the contact force

pliant plane will be considered. The force at the end effector is then related to the deformation of the environment by the following simplified model [18] (see Fig. 2.8)

$$\boldsymbol{f}_e = \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}_e), \tag{2.82}$$

where $\boldsymbol{x}$ is the position of the end effector expressed in the inertial frame, $\boldsymbol{x}_e$ characterizes the constant position of the unperturbed environment expressed in the inertial frame and

$$\boldsymbol{K} = k\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}, \tag{2.83}$$

with $k > 0$, is the stiffness matrix being $\boldsymbol{n}$ the vector normal to the plane.

In our case it is $\boldsymbol{x} = \boldsymbol{\eta}_{ee1}$; however, in the force control chapter, the notation $\boldsymbol{x}$ will be maintained.

2.13 Identification

Identification of the dynamic parameters of underwater robotic structures is a very challenging task. The mathematical model shares its main characteristics with the model of a ground-fixed industrial manipulator, e.g., it is non-linear and coupled. In case of underwater structure, however, the hydrodynamic terms are approximation of the physical effects. The actuation system of the vehicle is achieved mainly by the thrusters the models of which are still object of research. Finally, accurate measurement of the whole configuration is not easy. For these reasons, while from the mathematical aspect the problem is not new, from the practical point of view it is very difficult to set-up a systematic and reliable identification procedure for UVMSs. At the best of our knowledge, there is no significant results in the identification of full UVMSs model. Few experimental results, moreover, concern the sole vehicle; often driven in few DOFs.

Since most of the fault detection algorithms rely on the accuracy of the mathematical model, this is also the reason why in this domain too, there are few experimental results (see Chap. 4).

In [46] the hydrodynamic damping terms of the vehicle Roby 2 developed at the Naval Automation Institute, National Research Council, Italy, (now CNR-ISSIA) have been experimentally estimated and further used to develop fault detection/tolerance strategies. The vehicle is stable in roll and pitch, hence, considering a constant depth, the sole planar model is identified.

In [47] some sea trials have been set-up in order to estimate the hydrodynamic derivatives of an 1/3-scale PAP-104 mine countermeasures ROV. The paper assumes that the added mass is already known, moreover, the identification concerns the planar motion for the 6-DOFs model. The position of the vehicle is measured by means of a redundant acoustic system; all the measurements are fused in an EKF in order to obtain the optimum state estimation. Experimental results are given.

The work [48] reports some experimental results on the single DOF models for the ROV developed at the John Hopkins University (JHUROV). First, the mathematical model is written so as to underline the Input-to-State-Stability; then a stable, on-line, adaptive identification technique is derived. The latter method is compared with a classic, off-line, Least-Squares approach. The approximation required by the proposed technique is that the equations of motion are decoupled, diagonal, there is no tether disturbance and the added mass is constant. Interesting experimental results are reported. The same set-up has also been used in [49] to develop an adaptive identification algorithm, prove its local stability and implement comparative experimental tests, and in [50] to experimental compare six different models and identification techniques for the full-DOF vehicle.

The interested reader can refer also to [3, 51–53].

References

1. K.R. Goheen, Modeling methods for underwater robotic vehicle dynamics. J. Robot. Syst. **8**(3), 295–317 (1991)
2. I. Schjølberg, Modeling and Control of Underwater Robotic Systems, Ph.D. thesis, Doktor ingeniør degree, Norwegian University of Science and Technology, Trondheim, Norway, 1996
3. G. Indiveri, Modelling and identification of underwater robotic systems. Comput. Sci. (1998)

4. K. Ioi, K. Itoh, Modelling and simulation of an underwater manipulator. Adv. Robot. **4**(4), 303–317 (1989)
5. J. Kim, W.K. Chung, J. Yuh, Dynamic analysis and two-time scale control for underwater vehicle-manipulator systems, in *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003 (IROS 2003)*, vol. 1, IEEE, 2003, pp. 577–582
6. K.N. Leabourne, S.M. Rock, M.J. Lee, Model development of an underwater manipulator for coordinated arm-vehicle control, in *OCEANS'98 Conference Proceedings*, vol. 2, IEEE, 1998, pp. 941–946
7. B. Lévesque, M.J. Richard, Dynamic analysis of a manipulator in a fluid environment. Int. J. Robot. Res. **13**(3), 221–231 (1994)
8. T.W. McLain, S.M. Rock, Development and experimental validation of an underwater manipulator hydrodynamic model. Int. J. Robot. Res. **17**(7), 748–759 (1998)
9. M.J. Richard, B. Levesque, Stochastic dynamical modelling of an open-chain manipulator in a fluid environment. Mech. Mach. Theory **31**(5), 561–572 (1996)
10. I. Schjølberg, T. Fossen, Modelling and control of underwater vehicle-manipulator systems, in *Proceedings of 3rd Conference on Marine Craft maneuvering and control*, Southampton, UK, 1994, pp. 45–57
11. T.J. Tarn, G.A. Shoults, S.P. Yang, A dynamic model of an underwater vehicle with a robotic manipulator using Kane's method. Auton. Robot. **3**(2), 269–283 (1996)
12. T.J. Tarn, S.P. Yang, Modeling and control for underwater robotic manipulators—an example, in *Proceedings of 1997 IEEE International Conference on Robotics and Automation, 1997*, vol. 3, IEEE, Albuquerque, NM, 1997, pp. 2166–2171
13. T. Padir, A.J. Koivo, Modeling of two underwater vehicles with manipulators on-board, in *IEEE International Conference on Systems, Man and Cybernetics, 2003*, vol. 2, IEEE, 2003, pp. 1359–1364
14. G. Antonelli, T. Fossen, D. Yoerger, in *Underwater Robotics*, ed. by B. Siciliano, O. Khatib. Springer Handbook of Robotics (Springer, Heidelberg, 2008), pp. 987–1008
15. T. Fossen, *Guidance and Control of Ocean Vehicles* (Wiley, Chichester, 1994)
16. T. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles* (Marine Cybernetics, Trondheim, Norway, 2002)
17. T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control* (Wiley, New York, 2011)
18. B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer, New York, 2009)
19. SNAME, Nomenclature for treating the motion of a submerged body through a fluid. Technical and Research Bulletin, 1950, pp. 1–5
20. N.A. Chaturvedi, A.K. Sanyal, N.H. McClamroch, Rigid-body attitude control. IEEE Control Syst. Mag. **31**(3), 30–51 (2011)
21. S.W. Shepperd, Quaternion from rotation matrix. J. Guid. Control **1**, 223 (1978)
22. T.W. McLain, S.M. Rock, M.J. Lee, Experiments in the coordinated control of an underwater arm/vehicle system. Auton. Robot. **3**(2), 213–232 (1996)
23. J.N. Newman, *Marine Hydrodynamics* (MIT Press, Cambridge, 1977)
24. E. Olguin Diaz, Modélisation et Commande d'un Système Véhicule/Manipulateur Sous-Marin (in French). Ph.D. thesis, Docteur de l'Institut National Polytechnique de Grenoble, Grenoble, France, 1999
25. A. Healey, D. Lienard, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J. Oceanic Eng. **18**(3), 327–339 (1993)
26. S. McMillan, D.E. Orin, R.B. McGhee, Efficient dynamic simulation of an unmanned underwater vehicle with a manipulator, in *Proceedings of 1994 IEEE International Conference on Robotics and Automation, 1994*, IEEE, San Diego, CA, 1994, pp. 1133–1140
27. T. Sarpkaya, M. Isaacson, *Mechanics of Wave Forces on Offshore Structures* (Van Nostrand Reinhold Company, New York, 1981)
28. B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation* (Wiley, New York, 1992)

29. G. Antonelli, S. Chiaverini, N. Sarkar, M. West, Adaptive control of an autonomous underwater vehicle. Experimental results on ODIN, in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, Nov 1999, pp. 64–69
30. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, On the use of integral control actions for autonomous underwater vehicles, in *2001 European Control Conference*, Porto, Sept 2001
31. G. Antonelli, S. Chiaverini, N. Sarkar, M. West, Adaptive control of an autonomous underwater vehicle: experimental results on ODIN. IEEE Trans. Control Syst. Technol. **9**(5), 756–765 (2001)
32. T. Fossen, J.G. Balchen et al., The NEROV autonomous underwater vehicle, in *Proceedings of Conference Oceans 91*, Citeseer, Honolulu, HI, 1991
33. R. Bachmayer, L.L. Whitcomb, M.A. Grosenbaugh, An accurate four-quadrant nonlinear dynamical model for marine thrusters: theory and experimental validation. IEEE J. Oceanic Eng. **25**(1), 146–159 (2000)
34. A. Healey, S.M. Rock, S. Cody, D. Miles, J.P. Brown, Toward an improved understanding of thruster dynamics for underwater vehicles. IEEE J. Oceanic Eng. **20**(4), 354–361 (1995)
35. J. Kim, W.K. Chung, Accurate and practical thruster modeling for underwater vehicles. Ocean Eng. **33**(5), 566–586 (2006)
36. T.H. Koh, M.W.S. Lau, E. Low, G. Seet, S. Swei, P.L. Cheng, A study of the control of an underactuated underwater robotic vehicle, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002*, vol. 2, IEEE, Lausanne, CH, 2002, pp. 2049–2054
37. D.A. Smallwood, L.L. Whitcomb, The effect of model accuracy and thruster saturation on tracking performance of model based controllers for underwater robotic vehicles: Experimental results, in *Proceedings of IEEE International Conference on Robotics and Automation, 2002, ICRA'02*, vol. 2, IEEE, Washington, DC, 2002, pp. 1081–1087
38. L.L. Whitcomb, D. Yoerger, Development, comparison, and preliminary experimental validation of nonlinear dynamic thruster models. IEEE J. Oceanic Eng. **24**(4), 481–494 (1999)
39. D. Yoerger, J.G. Cooke, J.J. Slotine, The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. IEEE J. Oceanic Eng. **15**(3), 167–178 (1990)
40. J. Jang, J. Han, Y. Choi, W.K. Chung, Development of a small underwater vehicle-manipulator system for tasks in shallow water, in *International Conference on Advanced Robotics*, 2007
41. S.K. Choi, J. Yuh, Experimental study on a learning control system with bound estimation for underwater robots. Auton. Robot. **3**(2), 187–194 (1996)
42. J. Nie, J. Yuh, E. Kardash, T. Fossen, On-board sensor-based adaptive control of small UUVS in very shallow water. Int. J. Adapt. Control Signal Process. **14**(4), 441–452 (2000)
43. M. Gautier, W. Khalil, Direct calculation of minimum set of inertial parameters of serial robots. IEEE Trans. Robot. Autom. **6**, 368–373 (1990)
44. D. Webb, P. Simonetti, C. Jones, SLOCUM: an underwater glider propelled by environmental energy. IEEE J. Oceanic Eng. **26**(4), 447–452 (2001)
45. C. Canudas de Wit, E. Olguin Diaz, M. Perrier, Robust nonlinear control of an underwater vehicle/manipulator system with composite dynamics, in *Proceedings of 1998 IEEE International Conference on Robotics and Automation, 1998*, IEEE, Leuven, Belgium, 1998, pp. 452–457
46. A. Alessandri, M. Caccia, G. Indiveri, G. Veruggio, Application of LS and EKF techniques to the identification of underwater vehicles, in *Proceedings of the 1998 IEEE International Conference on Control Applications, 1998*, vol. 2, IEEE, Trieste, 1998, pp. 1084–1088
47. J. Pereira, A. Duncan, System identification of underwater vehicles, in *Proceedings of the 2000 International Symposium on Underwater Technology, 2000, UT 00*, IEEE, Tokyo, JP, 2000, pp. 419–424
48. D.A. Smallwood, L.L. Whitcomb, Adaptive identification of dynamically positioned underwater robotic vehicles. IEEE Trans. Control Syst. Technol. **11**(4), 505–515 (2003)
49. C.J. McFarland, L.L. Whitcomb, Comparative experimental evaluation of a new adaptive identifier for underwater vehicles, in *Proceedings of IEEE International Conference on Robotics and Automation, 2013, ICRA'13*, IEEE, 2013, pp. 4599–4605

50. S.C. Martin, L.L. Whitcomb, Preliminary experiments in comparative experimental identification of six degree-of-freedom coupled dynamic plant models for underwater robot vehicles, in *Proceedings of IEEE International Conference on Robotics and Automation, 2013, ICRA'13*, IEEE, 2013, pp. 2947–2954
51. M. Caccia, G. Indiveri, G. Veruggio, Modeling and identification of open-frame variable configuration unmanned underwater vehicles. IEEE J. Oceanic Eng. **25**(2), 227–240 (2000)
52. D. Smallwood, L. Whitcomb, Adaptive identification of dynamically positioned underwater robotic vehicles. IEEE Trans. Control Syst. Technol. **11**(4), 505–515 (2003)
53. W.L. Chan, T. Kang, Simultaneous determination of drag coefficient and added mass. IEEE J. Oceanic Eng. **36**(3), 422–430 (2011)

Chapter 3
Dynamic Control of 6-DOF AUVs

3.1 Introduction

Control of UVMSs require full-DOF control of the vehicle, cruise vehicles with rudder and stern are not suitable to hold a manipulator arm for their incapacity to counteract the interaction forces with the arm itself. For this reason the following chapter restricts the discussion to the problem of controlling an underwater vehicle in 6-DOFs.

To effectively compensate the hydrodynamic effects several adaptive (integral) control laws have been proposed in the literature (see, e.g., [1–5]). In [6], a number of adaptive control actions are proposed, where the presence of an external disturbance is taken into account and its counteraction is obtained by means of a switching term; simulation on the simplified three-DOF horizontal model of NEROV are given. In [7], a body-fixed-frame based adaptive control law is developed. In [8], an adaptive control law based on Euler angle representation of the orientation has been proposed for the control of an AUV; planar simulations are provided to show the effectiveness of the proposed approach. Reference [9] proposes a self-adaptive neuro-fuzzy inference system that makes use of a 5-layer-structured neural network to improve the function approximation. In [10] a fuzzy membership function based-neural network is proposed; the control's membership functions derivation is achieved by a back propagation network.

Six-DOF experimental results are not common in the literature [11]. References [6, 12–14] describe 6-DOF control laws in which the orientation is described by the use of quaternions. The papers [12, 13, 15–17] report 6-DOF experimental results on the underwater vehicle ODIN (Omni-Directional Intelligent Navigator).

An experimental work is given in [18, 19] by the use of the Johns Hopkins University ROV on a single DOF. Different simple control laws are tested on the vehicle in presence of model mismatching and thruster saturation and their performance is evaluated. The work [20] gives an interesting 6-DOF experimental comparison among PID, model-based with and without exact linearization in a

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_3,

variety of operative conditions, by varying the control gains, implementing adaptive versions, by intentionally wrongly compensating for the terms, etc. The conclusions represent an important witness for control practitioners. In [21] an experimental comparison for 6-DOF control of the Johns Hopkins University ROV under model and non-model based approaches is performed to evaluate how it is important to compensate for the dynamics especially in the coupled maneuvers.

Among the other hydrodynamic effects acting on a rigid body moving in a fluid, the restoring generalized forces (gravity plus buoyancy) and the ocean current are of major concern in designing a motion control law for underwater vehicles, since they are responsible of steady-state position and orientation errors. However, while the restoring generalized forces are usually dealt with in the framework of adaptive dynamic compensations, only few papers take into account the effect of the ocean current. The works [12, 13, 22] consider a 6-DOF control problem in which the ocean current is compensated in vehicle-fixed coordinates; since the current effects are modeled as an external disturbance acting on the vehicle, there is no need for additional sensors. In [23–25] a different approach is proposed for the three-DOF surge control of the vehicle Phoenix: the current, or more generally, the sea wave, is modeled by an Auto-Regressive dynamic model and an extended Kalman filter is designed to estimate the relative velocity between vehicle and water; the estimated relative velocity is then used by a sliding-mode controller to drive the vehicle. In this case, additional sensors besides those typically available on-board are required. The controller developed in [23] has been also used for the NPS ARIES AUV [26]. Reference [27] reports an algorithm for the underwater navigation of a torpedo-like vehicle in presence of unknown current where the current itself is estimated by resorting to a range measurement from a single location.

The importance of the restoring forces can be appreciated in large-dimension vehicles, where small displacement of the centers of gravity and buoyancy generates large required thrusts to compensate for [28].

A common feature of all the adaptive control laws proposed in the literature is that they are designed starting from dynamic models written either in the earth-fixed frame or in the vehicle-fixed frame. Nevertheless, some hydrodynamic effects are seen as constant in the earth-fixed frame (e.g., the restoring linear force) while some others are constant in the vehicle-fixed frame (e.g., the restoring moment).

This chapter extends the content of [29], where a comparison among the controllers developed in [7, 14, 22, 30–32] is shown. The controllers have been designed for 6-DOFs control of AUVs and they do not need the measurement of the ocean current (when it is taken into account). The analysis will mainly concern the controllers capacity to compensate for the persistent dynamic effects, e.g., the restoring forces and the ocean current. For each controller a reduced version is derived and eventually modified so as to achieve null steady state error under modeling uncertainty and presence of ocean current with respect to a minimal number of parameters. It is worth noticing that the reduced controller is not given by the authors of the corresponding paper; this has to be taken into account while observing the simulation results. The reduced controller will be developed in order to achieve a PD action plus the adaptive/integral compensation of the persistent effects, i.e., it can be

Table 3.1 Labels of the discussed controllers

Label	Authors	Section	Frame
A	Fjellstad and Fossen	3.2	Earth
B	Yuh et al.	3.3	Earth
C	Fjellstad and Fossen	3.4	Vehicle
D	Fossen and Balchen	3.5	Earth/Vehicle
E	Antonelli et al.	3.6	Earth/Vehicle
F	Sun and Cheah	3.7	Earth/Vehicle

considered as the equivalent of an adaptive PD+gravity compensation for industrial manipulator. In other words, the velocity and acceleration based dynamic terms of the model will not be compensated for. Numerical simulations using the model of ODIN [32], have been run to verify the theoretical results.

For easy of readings, Table 3.1 reports the label associated with each controller.

3.2 Earth-Fixed-Frame-Based, Model-Based Controller

In 1994, Fjellstad and Fossen [14] propose an earth-fixed-frame-based, model-based controller that makes use of the 4-parameter unit quaternion (Euler parameter) to reach a singularity-free representation of the attitude. The controller is obtained by extending the results obtained in [33] for robot manipulators.

By defining

$$\tilde{\boldsymbol{p}} = \boldsymbol{p}_d - \boldsymbol{p} \tag{3.1}$$

where $\boldsymbol{p} = \left[\boldsymbol{\eta}_1^{\mathrm{T}}\ \mathcal{Q}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^7$ is the quaternion-based position/attitude vector of the vehicle and $\boldsymbol{p}_d = \left[\boldsymbol{\eta}_{1,d}^{\mathrm{T}}\ \mathcal{Q}_d^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^7$ is its desired value. The following (7×1) vector can be further defined:

$$\boldsymbol{s} = \boldsymbol{K}_D \dot{\tilde{\boldsymbol{p}}} + \boldsymbol{K}_P \tilde{\boldsymbol{p}} + \boldsymbol{K}_I \int_0^t \tilde{\boldsymbol{p}}(\tau) d\tau = \dot{\boldsymbol{p}}_r - \boldsymbol{K}_D \dot{\boldsymbol{p}} \tag{3.2}$$

that implies the vector $\dot{\boldsymbol{p}}_r \in \mathbb{R}^7$ defined as

$$\dot{\boldsymbol{p}}_r = \boldsymbol{K}_D \dot{\boldsymbol{p}}_d + \boldsymbol{K}_P \tilde{\boldsymbol{p}} + \boldsymbol{K}_I \int_0^t \tilde{\boldsymbol{p}}(\tau) d\tau \tag{3.3}$$

where $\boldsymbol{K}_D$, $\boldsymbol{K}_P$ and $\boldsymbol{K}_I$ are (7×7) positive definite matrices of gains.

The following control law is proposed

$$\boxed{\boldsymbol{\tau}_v^\star = \boldsymbol{M}_v^\star \ddot{\boldsymbol{p}}_r + \boldsymbol{C}_v^\star \dot{\boldsymbol{p}}_r + \boldsymbol{D}_{RB}^\star \dot{\boldsymbol{p}}_r + \boldsymbol{g}_{RB}^\star + \boldsymbol{\Lambda} \boldsymbol{s},} \tag{3.4}$$

where $\boldsymbol{\Lambda}$ is a (7×7) positive definite matrix of gains. Notice that the above control law refers to a quaternion-based dynamic model in earth-fixed coordinates which can be obtained from (2.56) by using the matrix $\boldsymbol{J}_{k,oq} \in \mathbb{R}^{4\times 3}$ instead of the matrix $\boldsymbol{J}_{k,o} \in \mathbb{R}^{3\times 3}$ in the construction of the Jacobian $\boldsymbol{J}_e(\boldsymbol{R}_B^I)$. Also notice that, with respect to the model detailed in Sect. 2.7 the dimension of $\boldsymbol{J}_e(\boldsymbol{R}_B^I)$ are different.

Let now consider the positive semi-definite function:

$$V = \frac{1}{2} \boldsymbol{s}^{\mathrm{T}} \boldsymbol{M}_v^\star \boldsymbol{s} > 0, \qquad \forall \boldsymbol{s} \neq \alpha \begin{bmatrix} \boldsymbol{0} \\ \mathcal{Q} \end{bmatrix}, \qquad \alpha \in \mathbb{R} \tag{3.5}$$

after straightforward calculation, its time derivative is given by:

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}} \left[\boldsymbol{\Lambda} + \boldsymbol{D}_{RB}^\star \right] \boldsymbol{s} < 0, \qquad \forall \boldsymbol{s} \neq \boldsymbol{0} \tag{3.6}$$

Since V is only positive semi-definite and the system is non-autonomous the stability can not be derived by applying the Lyapunov's theorem. By further assuming that $\dot{\boldsymbol{p}}_r$ is twice differentiable, then $\ddot{V}$ is bounded and $\dot{V}$ is uniformly continuous. Hence, application of the Barbălat's Lemma allows to prove global convergence of $\boldsymbol{s} \to \boldsymbol{0}$ as $t \to \infty$. Due to the definition of the vector $\boldsymbol{s}$, its convergence to zero also implies convergence of $\tilde{\boldsymbol{p}}$ to the null value.

In case of perfect knowledge of the dynamic model, moreover, the convergence of the error to zero can be demonstrated even for $\boldsymbol{K}_I = \boldsymbol{O}$, i.e., without integral action.

Compensation of the persistent effects. If a reduced version of the controller is implemented, e.g., by neglecting the model-based terms in (3.4), the restoring moment is not compensated efficiently. In fact, let consider a vehicle in the two static postures shown in Fig. 3.1 and let suppose that the vehicle, starting from the left configuration, is driven to the right configuration and, after a while, back to the left configuration. In the left configuration the integral action in (3.3) does not give any contribution to the control moment as expected, because the vectors of gravity and buoyancy are aligned. Furthermore, in the right configuration the integral action will compensate exactly at the steady state for the moment generated by the misalignment between gravity and buoyancy. When the vehicle is driven back to the left configuration, a null steady-state compensation error is possible after the integral action is discharged; this poses a severe limitation to the control bandwidth that can be achieved. A similar argument holds in the typical practical situation in which the compensation implemented through the vector $\boldsymbol{g}_{RB}^\star$ is not exact.

On the other hand, since the error variables are defined in the earth-fixed frame, the controller is appropriate to counteract the current effect. This point will be clarified in next subsections, when discussing the drawbacks of the controllers **C** and **D** with respect to the current compensation.

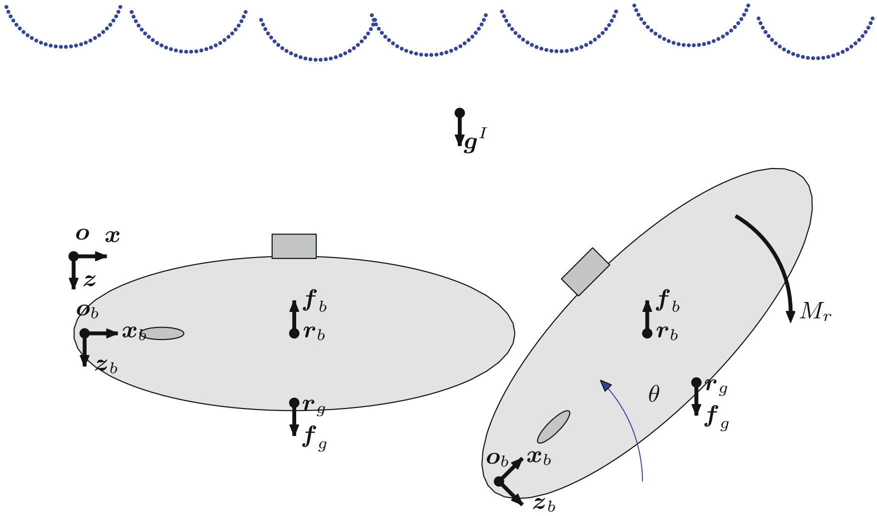


Fig. 3.1 Planar view parallel to the xz earth-fixed frame of two different configurations and corresponding restoring forces and moments: $\boldsymbol{r}_G$ ($\boldsymbol{r}_B$) is the center of gravity (buoyancy), $\boldsymbol{f}_G$ ($\boldsymbol{f}_B$) is the gravity (buoyancy) force, and M_R is the y component of the restoring moment

Finally, an adaptive version of this controller is not straightforward. In fact, since the dynamic model (2.56) does not depend on the absolute vehicle position, a steady null linear velocity of the vehicle with a non-null position error would *not excite* a corrective adaptive control action. As a result, null position error at rest cannot be guaranteed in presence of ocean current. From the theoretical point of view, this drawback can be avoided by defining the velocity error using the current measurement. However, from the practical point of view, this approach cannot achieve fine positioning of the vehicle since local vortices can make current measurement too noisy.

Reduced controller. According to the motivation given in the introduction of this section, the following reduced version of the control law is considered

$$\boldsymbol{\tau}_v^\star = \hat{\boldsymbol{g}}_{RB}^\star(\boldsymbol{R}_B^I) + \dot{\boldsymbol{p}}_r, \tag{3.7}$$

where $\hat{\boldsymbol{g}}_{RB}^\star(\boldsymbol{R}_B^I)$ is a model-based estimate of the restoring generalized force acting on the vehicle. It might be useful to substituting the definition of $\dot{\boldsymbol{p}}_r$ yielding

$$\boldsymbol{\tau}_v^\star = \hat{\boldsymbol{g}}_{RB}^\star(\boldsymbol{R}_B^I) + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{p}}}_d + \boldsymbol{K}_P\tilde{\boldsymbol{p}} + \boldsymbol{K}_I\int_0^t \tilde{\boldsymbol{p}}(\tau)d\tau \tag{3.8}$$

that has a clear interpretation.

3.3 Earth-Fixed-Frame-Based, Non-Model-Based Controller

In 1999, Yuh proposes an earth-fixed-frame-based, non-model-based controller [32]. The control input is given by:

$$\boxed{\boldsymbol{\tau}_v^\star = \boldsymbol{K}_1\ddot{\boldsymbol{\eta}}_d + \boldsymbol{K}_2\dot{\boldsymbol{\eta}} + \boldsymbol{K}_3 + \boldsymbol{K}_4\dot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{K}_5\tilde{\boldsymbol{\eta}} = \sum_{i=1}^{5}\boldsymbol{K}_i\boldsymbol{\phi}_i\,,} \tag{3.9}$$

where $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_d - \boldsymbol{\eta}$, the gains $\boldsymbol{K}_i \in \mathbb{R}^{6\times6}$ are computed as

$$\boldsymbol{K}_i = \frac{\hat{\gamma}_i \boldsymbol{s}_e \boldsymbol{\phi}_i^{\mathrm{T}}}{\|\boldsymbol{s}_e\|\,\|\boldsymbol{\phi}_i\|} \qquad i = 1, \ldots, 5,$$

where

$$\boldsymbol{s}_e = \dot{\tilde{\boldsymbol{\eta}}} + \sigma\tilde{\boldsymbol{\eta}} \qquad \text{with } \sigma > 0,$$

and the factors $\hat{\gamma}_i$'s are updated by

$$\dot{\hat{\gamma}}_i = f_i\,\|\boldsymbol{s}_e\|\,\|\boldsymbol{\phi}_i\| \qquad \text{with } f_i > 0 \qquad i = 1, \ldots, 5.$$

Please notice that $\boldsymbol{\phi}_3 = \boldsymbol{k} \in \mathbb{R}^6$, i.e., a positive constant vector.

Experimental results in 6-DOFs on the use of (3.9) are reported in [16, 17]; these have proven the effectiveness of this controller starting from the surface (with null initial gains) where a smooth version of the controller has been implemented.

The stability analysis can be performed using Lyapunov-like arguments starting from the function

$$V = \frac{1}{2}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}\tilde{\boldsymbol{\eta}} + \frac{1}{2}\sum_{i=1}^{5}\frac{1}{f_i}\left(\gamma_i - \hat{\gamma}_i\right)^2$$

that, differentiated with respect to the time yields a negative semi-definite scalar function; details can be found in [17].

Recently, in [17], Zhao and Yuh propose a model-based version of this control law. The eventual knowledge of the vehicle dynamics is exploited by implementing a *disturbance observer* in charge of partially compensate for the system dynamics. Interesting experimental results with the vehicle ODIN are reported where the tuning of the gains has been achieved on the single DOFs independently. This has been made possible in view of the specific shape of the vehicle, close to a sphere, that reduces the coupling effects and the difference among the directions. It is worth noticing that the considerations below are valid also for this version of the controller.

Compensation of the persistent effects. Since the tracking errors are defined in the earth-fixed frame, similar considerations to those developed for the law **A** can be done with respect to compensation of both restoring generalized forces and ocean current effects.

Reduced controller. Being the sole non-model-based law, the controller presented in [32] is characterized by an error behavior much different from that obtained by the other considered controllers and a fair comparison is made difficult. For this reason, a reduced controller has not been retrieved.

3.4 Vehicle-Fixed-Frame-Based, Model-Based Controller

Several controllers have been proposed in literature which are based on vehicle-fixed-frame error variables. Among them, the controller proposed in 1994 by Fjellstad and Fossen [7]; full DOFs experimental results have been reported by Antonelli et al. in [12, 13].

Let consider the vehicle-fixed variables:

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{R}_I^B \tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{\varepsilon}} \end{bmatrix}$$
$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu}_d - \boldsymbol{\nu},$$

where $\tilde{\boldsymbol{\eta}}_1 = \boldsymbol{\eta}_{1,d} - \boldsymbol{\eta}_1$, being $\boldsymbol{\eta}_{1,d}$ the desired position, and $\tilde{\boldsymbol{\varepsilon}}$ is the quaternion based attitude error,

$$\boldsymbol{s}_v = \tilde{\boldsymbol{\nu}} + \boldsymbol{\Lambda} \tilde{\boldsymbol{y}}, \tag{3.10}$$

with $\boldsymbol{\Lambda} = \text{blockdiag}\{\lambda_p \boldsymbol{I}_3, \lambda_o \boldsymbol{I}_3\}$, $\boldsymbol{\Lambda} > \boldsymbol{O}$.

$$\boldsymbol{\nu}_a = \boldsymbol{\nu}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{y}}. \tag{3.11}$$

Reminding the vehicle regressor $\boldsymbol{\Phi}_v \in R^{6 \times n_{\theta,v}}$ defined in (2.57) and the corresponding vector of dynamic parameters $\boldsymbol{\theta}_v \in R^{n_{\theta,v}}$, the control law is given by:

$$\boxed{\boldsymbol{\tau}_v = \boldsymbol{\Phi}_v(\boldsymbol{R}_B^I, \boldsymbol{\nu}, \boldsymbol{\nu}_a, \dot{\boldsymbol{\nu}}_a)\hat{\boldsymbol{\theta}}_v + \boldsymbol{K}_D \boldsymbol{s}_v,} \tag{3.12}$$

where $\boldsymbol{K}_D$ is a (6×6) positive definite matrix. The parameter estimate $\hat{\boldsymbol{\theta}}_v$ is updated by

$$\boxed{\dot{\hat{\boldsymbol{\theta}}}_v = \boldsymbol{K}_\theta^{-1} \boldsymbol{\Phi}_v^{\mathrm{T}}(\boldsymbol{R}_B^I, \boldsymbol{\nu}, \boldsymbol{\nu}_a, \dot{\boldsymbol{\nu}}_a)\boldsymbol{s}_v,} \tag{3.13}$$

where $\boldsymbol{K}_\theta$ is a suitable positive definite matrix of appropriate dimension.

The stability analysis is achieved by considering the following Lyapunov candidate function:

$$V = \frac{1}{2}\,\boldsymbol{s}_v^{\mathrm{T}}\boldsymbol{M}_v\boldsymbol{s}_v + \frac{1}{2}\,\tilde{\boldsymbol{\theta}}_v^{\mathrm{T}}\boldsymbol{K}_\theta\tilde{\boldsymbol{\theta}}_v > 0, \qquad \forall \boldsymbol{s} \neq \boldsymbol{0},\; \tilde{\boldsymbol{\theta}}_v \neq \boldsymbol{0} \tag{3.14}$$

the time derivative of which, by applying the proposed control law, is given by

$$\dot{V} = -\boldsymbol{s}_v^{\mathrm{T}}\left[\boldsymbol{K}_D + \boldsymbol{D}_{RB}\right]\boldsymbol{s}_v \tag{3.15}$$

It is now possible to prove the system stability in a Lyapunov-Like sense using the Barbălat's Lemma. Since

- V is lower bounded
- $\dot{V}(\boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v) \leq 0$
- $\dot{V}(\boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v)$ is uniformly continuous

then

- $\dot{V}(\boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v) \rightarrow 0$ as $t \rightarrow \infty$.

Thus $\boldsymbol{s}_v \rightarrow \boldsymbol{0}$ as $t \rightarrow \infty$. In view of the definition of $\boldsymbol{s}_v$, this implies that $\tilde{\boldsymbol{\nu}} \rightarrow \boldsymbol{0}$ as $t \rightarrow \infty$; in addition, due to the properties of the quaternion, it results that $\tilde{\eta} \rightarrow 1$ as $t \rightarrow \infty$. However, as usual in adaptive control schemes, it is not possible to prove asymptotic stability of the whole state since $\tilde{\boldsymbol{\theta}}_v$ is only guaranteed to be bounded.

Notice that, in [7], further discussion is developed by considering different choices for the matrix $\boldsymbol{\Lambda}$.

In 1997 [34] Conte and Serrani develop a Lyapunov-based control for AUVs. The designed controller, by introducing a representation of the model uncertainties, is made robust using Lyapunov techniques. It is worth noticing that this approach does not take into account explicitly for an adaptive/integral action to compensate for the current effect. The position error is represented within a vehicle-fixed representation with a feedback term similar to the vector $\boldsymbol{s}_v$ used in this Section.

Compensation of the persistent effects. Following the same reasoning as previously done for the law **A**, it can be deducted that also in this case the adaptive compensation cannot guarantee null position error at rest in the presence of ocean current. In fact, the dynamic model (2.54) shows that a steady null linear velocity of the vehicle with a non-null position error does not excite a corrective adaptive control action. Again, ocean current measurement cannot overcome this problem in practice.

To achieve a null position error in presence of ocean current, an integral action on body-fixed-frame error variables was considered in [12, 13]. However, this integral action has the following drawback: let suppose that the vehicle is at rest in the left configuration of Fig. 3.2 in presence of a water current aligned to the earth-fixed $\boldsymbol{y}$ axis; the control action builds the current compensation term which, in this particular configuration, turns out to be parallel to the $\boldsymbol{y}_b$ axis. If the vehicle is now *quickly* rotated to right configuration, the built compensation term rotates together

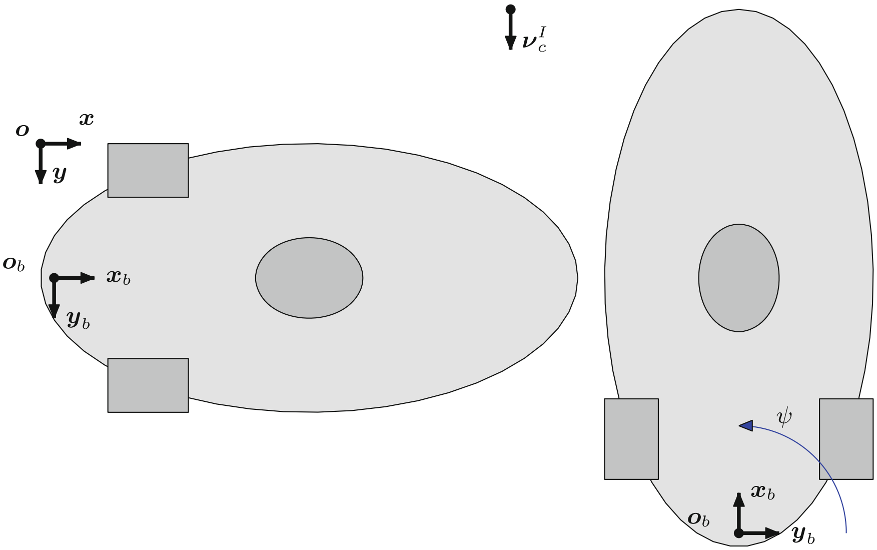


Fig. 3.2 Planar view parallel to the xy earth-fixed frame of two different configurations under the constant current $\boldsymbol{\nu}_c^I$

with the vehicle keeping its alignment to the $\boldsymbol{y}_b$ axis; however, this vehicle-fixed axis has now become parallel to the $\boldsymbol{x}$ axis of the earth-fixed frame. Therefore, the built compensation term acts as a disturbance until the integral action has re-built proper current compensation for the right configuration. It is clear at this point that this drawback does not arise for the controller **E** and for the controllers **A** and **B**, since they build ocean current compensation in an earth-fixed frame.

Reduced controller. Similarly to the other cases, a reduced form of the controller has been derived. An integral action on body-fixed-frame error variables has also been considered as in [13] to counteract the current effects, i.e.,

$$\boldsymbol{\tau}_v = \boldsymbol{\Phi}_{v,P'}(\boldsymbol{R}_B^I)\hat{\boldsymbol{\theta}}_v + \boldsymbol{K}_D\boldsymbol{s}_v \tag{3.16}$$

$$\dot{\hat{\boldsymbol{\theta}}}_v = \boldsymbol{K}_\theta^{-1}\boldsymbol{\Phi}_{v,P'}^{\mathrm{T}}\boldsymbol{s}_v . \tag{3.17}$$

3.5 Model-Based Controller Plus Current Compensation

Fossen and Balchen, in 1991 [22], propose a control law that explicitly takes into account the ocean current without need for the current measurement.

Being:

$$\boldsymbol{s}_e = \dot{\tilde{\boldsymbol{\eta}}} + \sigma\tilde{\boldsymbol{\eta}} \qquad \text{with } \sigma > 0\,, \tag{3.18}$$

the controller is given by:

$$\boxed{\boldsymbol{\tau}_v = \boldsymbol{\Phi}_v(\boldsymbol{R}^I_B, \boldsymbol{\nu}, \boldsymbol{\nu}_a, \dot{\boldsymbol{\nu}}_a)\hat{\boldsymbol{\theta}}_v + \hat{\boldsymbol{v}} + \boldsymbol{J}^{\mathrm{T}}_e(\boldsymbol{R}^I_B)\boldsymbol{K}_D\boldsymbol{s}_e} \tag{3.19}$$

where $\boldsymbol{K}_D \in \mathbb{R}^{6\times 6}$, is a positive definite matrix of gains. Notice that the PD action is similar to the transpose of the Jacobian approach developed for industrial manipulators [35]. The current compensation $\hat{\boldsymbol{v}}$ is updated by the following

$$\boxed{\dot{\hat{\boldsymbol{v}}} = \boldsymbol{W}^{-1}\boldsymbol{J}^{-1}_e(\boldsymbol{R}^I_B)\boldsymbol{s}_e} \tag{3.20}$$

with $\boldsymbol{W} > 0$ and the dynamic parameters are updated by

$$\boxed{\dot{\hat{\boldsymbol{\theta}}}_v = \boldsymbol{K}^{-1}_\theta\boldsymbol{\Phi}^{\mathrm{T}}_v(\boldsymbol{R}^I_B, \boldsymbol{\nu}, \boldsymbol{\nu}_a, \dot{\boldsymbol{\nu}}_a)\boldsymbol{J}^{-1}_e(\boldsymbol{R}^I_B)\boldsymbol{s}_e\,,} \tag{3.21}$$

where, again, $\boldsymbol{K}_\theta$, is a positive definite matrices of gains of appropriate dimensions.

It is worth noticing that with this control the model considers the current as an additive disturbance, constant in the body-fixed frame (see Sect. 2.4.3).

The stability analysis is developed by defining as Lyapunov candidate function

$$V = \boldsymbol{s}^{\mathrm{T}}_e\boldsymbol{M}^\star_v\boldsymbol{s}_e + \frac{1}{2}\,\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_v\boldsymbol{K}_\theta\tilde{\boldsymbol{\theta}}_v + \frac{1}{2}\,\tilde{\boldsymbol{v}}^{\mathrm{T}}\boldsymbol{W}\tilde{\boldsymbol{v}} \tag{3.22}$$

that is positive definite $\forall \boldsymbol{s}_e \neq \boldsymbol{0},\ \tilde{\boldsymbol{\theta}}_v \neq \boldsymbol{0},\ \tilde{\boldsymbol{v}} \neq \boldsymbol{0}$.

By applying the proposed control law, and following the guidelines in [22], it is possible to demonstrate that

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}}_e\left[\boldsymbol{K}_D + \boldsymbol{D}^\star_{RB}\right]\boldsymbol{s}_e \leq 0 \tag{3.23}$$

It is now possible to prove again the system stability in a Lyapunov-Like sense using the Barbălat’s Lemma. Since

- V is lower bounded
- $\dot{V}(\boldsymbol{s}_e, \tilde{\boldsymbol{\theta}}_v, \tilde{\boldsymbol{v}}) \leq 0$
- $\dot{V}(\boldsymbol{s}_e, \tilde{\boldsymbol{\theta}}_v, \tilde{\boldsymbol{v}})$ is uniformly continuous

then

- $\dot{V}(\boldsymbol{s}_e, \tilde{\boldsymbol{\theta}}_v, \tilde{\boldsymbol{v}}) \to 0$ as $t \to \infty$.

Thus $\boldsymbol{s}_e \to \mathbf{0}$ as $t \to \infty$. In view of the definition of $\boldsymbol{s}_e$, this implies that $\tilde{\boldsymbol{\eta}} \to \mathbf{0}$ as $t \to \infty$. However, the vectors $\tilde{\boldsymbol{\theta}}_v$ and $\tilde{\boldsymbol{v}}$ are only guaranteed to be bounded.

Compensation of the persistent effects. Since the adaptive compensation term $\boldsymbol{\Phi}_v(\boldsymbol{R}_B^I, \boldsymbol{\nu}, \boldsymbol{\nu}_a, \dot{\boldsymbol{\nu}}_a)\hat{\boldsymbol{\theta}}_v$ operates in vehicle-fixed coordinates, the control law is suited for effective compensation of the restoring moment.

On the other hand, despite the error vector $\boldsymbol{s}_e$ is based on earth-fixed quantities, the current-compensation term $\hat{\boldsymbol{v}}$ is built through the integral action (3.20) which works in vehicle-fixed coordinates; this implies that, as for counteraction of the ocean current, this control law suffers from the same drawback as that discussed for the law **C**.

Reduced controller. The model-based compensation has been reduced to the restoring generalized force alone:

$$\boldsymbol{\tau}_v = \boldsymbol{\Phi}_{v,R}(\boldsymbol{R}_B^I)\hat{\boldsymbol{\theta}}_{v,R} + \hat{\boldsymbol{v}} + \boldsymbol{J}_e^{\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{K}_D\boldsymbol{s}_e \tag{3.24}$$

$$\dot{\hat{\boldsymbol{v}}} = \boldsymbol{W}^{-1}\boldsymbol{J}_e^{-1}(\boldsymbol{R}_B^I)\boldsymbol{s}_e \tag{3.25}$$

$$\dot{\hat{\boldsymbol{\theta}}}_{v,R} = \boldsymbol{K}_\theta^{-1}\boldsymbol{\Phi}_{v,R}^{\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{J}_e^{-1}(\boldsymbol{R}_B^I)\boldsymbol{s}_e, \tag{3.26}$$

3.6 Mixed Earth/Vehicle-Fixed-Frame-Based, Model-Based Controller

In 2001, Antonelli, Caccavale, Chiaverini and Fusco [30, 36] propose and adaptive tracking control law that takes into account the different nature of the hydrodynamic effects acting on the AUV (ROV); this is achieved by suitably building each dynamic compensation action in a proper (either inertial or vehicle-fixed) reference frame. In fact, since adaptive or integral control laws asymptotically achieve compensation of the *constant* disturbance terms, it is convenient to build the compensation action in a reference frame with respect to which the disturbance term itself is seen as much as possible as constant. The analysis has been extended in [37, 38].

Let consider the vehicle-fixed variables:

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{R}_I^B\tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{\varepsilon}} \end{bmatrix}$$

$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu}_d - \boldsymbol{\nu},$$

where $\tilde{\boldsymbol{\eta}}_1 = \boldsymbol{\eta}_{1,d} - \boldsymbol{\eta}_1$, being $\boldsymbol{\eta}_{1,d}$ the desired position, and $\tilde{\boldsymbol{\varepsilon}}$ is the quaternion based attitude error. Let define as $\hat{\boldsymbol{\theta}}_v$ the vector of parameters to be adapted,

$$\boldsymbol{s}_v = \tilde{\boldsymbol{\nu}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{y}}, \tag{3.27}$$

with $\boldsymbol{\Lambda} = \mathrm{blockdiag}\{\lambda_p\boldsymbol{I}_3, \lambda_o\boldsymbol{I}_3\}$, $\boldsymbol{\Lambda} > \boldsymbol{O}$. The control law is given by:

$$\boxed{\boldsymbol{\tau}_v = \boldsymbol{K}_D \boldsymbol{s}_v + \boldsymbol{K}\tilde{\boldsymbol{y}} + \boldsymbol{\Phi}_{v,T}\hat{\boldsymbol{\theta}}_v} \tag{3.28}$$

where $\boldsymbol{K}_D \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{K} = \text{blockdiag}\{k_p \boldsymbol{I}_3, k_o \boldsymbol{I}_3\}$ are positive definite matrices of the gains to be designed, and

$$\boxed{\dot{\hat{\boldsymbol{\theta}}}_v = \boldsymbol{K}_\theta^{-1}\boldsymbol{\Phi}_{v,T}^{\mathrm{T}}\boldsymbol{s}_v\,,} \tag{3.29}$$

where $\boldsymbol{K}_\theta$ is also a positive definite matrix of proper dimensions.

3.6.1 Stability Analysis

Let us consider the following scalar function

$$V = \frac{1}{2}\boldsymbol{s}_v^{\mathrm{T}}\boldsymbol{M}_v\boldsymbol{s}_v + \frac{1}{2}\tilde{\boldsymbol{\theta}}_v^{\mathrm{T}}\boldsymbol{K}_\theta\tilde{\boldsymbol{\theta}}_v + \frac{1}{2}k_p\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_1 + k_o\tilde{\boldsymbol{z}}^{\mathrm{T}}\tilde{\boldsymbol{z}}\,, \tag{3.30}$$

where $\tilde{\boldsymbol{z}} = \left[1\ \boldsymbol{0}^{\mathrm{T}}\right]^{\mathrm{T}} - \boldsymbol{z} = \left[1 - \tilde{\eta}\ -\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\right]^{\mathrm{T}}$. Due to the positive definiteness of $\boldsymbol{M}_v$, $\boldsymbol{K}_\theta$, and $\boldsymbol{K}$, the scalar function $V(\tilde{\boldsymbol{\eta}}_1, \tilde{\boldsymbol{z}}, \boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v)$ is positive definite.

Let us define the following partition for the variable $\boldsymbol{s}_v$ that will be useful later:

$$\boldsymbol{s}_v = \begin{bmatrix}\boldsymbol{s}_p\\ \boldsymbol{s}_o\end{bmatrix}, \tag{3.31}$$

with $\boldsymbol{s}_p \in \mathbb{R}^3$ and $\boldsymbol{s}_o \in \mathbb{R}^3$. In view of (3.31) and the definition of $\boldsymbol{s}_v$, it is

$$\tilde{\boldsymbol{\nu}}_1 = \boldsymbol{s}_p - \lambda_p\boldsymbol{R}_I^B\tilde{\boldsymbol{\eta}}_1 \tag{3.32}$$

$$\tilde{\boldsymbol{\nu}}_2 = \boldsymbol{s}_o - \lambda_o\tilde{\boldsymbol{\varepsilon}}\,. \tag{3.33}$$

Differentiating V with respect to time yields:

$$\begin{aligned}\dot{V} &= \boldsymbol{s}_v^{\mathrm{T}}\boldsymbol{M}_v\dot{\boldsymbol{s}}_v + \tilde{\boldsymbol{\theta}}_v^{\mathrm{T}}\boldsymbol{K}_\theta\dot{\tilde{\boldsymbol{\theta}}}_v + k_p\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\boldsymbol{R}_B^I\tilde{\boldsymbol{\nu}}_1 - 2k_o\tilde{\boldsymbol{z}}^{\mathrm{T}}\boldsymbol{J}_{k,oq}\tilde{\boldsymbol{\nu}}_2\\ &= \boldsymbol{s}_v^{\mathrm{T}}\boldsymbol{M}_v\left(\dot{\boldsymbol{\nu}}_d - \dot{\boldsymbol{\nu}} + \boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{y}}}\right) + \tilde{\boldsymbol{\theta}}_v^{\mathrm{T}}\boldsymbol{K}_\theta\dot{\tilde{\boldsymbol{\theta}}}_v\\ &\quad + k_p\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\boldsymbol{R}_B^I\left(\boldsymbol{s}_p - \lambda_p\boldsymbol{R}_I^B\tilde{\boldsymbol{\eta}}_1\right)\\ &\quad - k_o\left[1 - \tilde{\eta}\ -\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\right]\begin{bmatrix}-\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\\ \tilde{\eta}\boldsymbol{I}_3 + \boldsymbol{S}(\tilde{\boldsymbol{\varepsilon}})\end{bmatrix}(\boldsymbol{s}_o - \lambda_o\tilde{\boldsymbol{\varepsilon}})\,.\end{aligned}$$

Then, defining

$$\boldsymbol{\nu}_a = \boldsymbol{\nu}_d + \boldsymbol{\Lambda}\tilde{\boldsymbol{y}} \tag{3.34}$$

yields (dependencies are dropped out to increase readability):

$$\begin{aligned}\dot{V} &= \boldsymbol{s}_v^{\mathrm{T}}\left[\boldsymbol{M}_v\dot{\boldsymbol{\nu}}_a - \boldsymbol{\tau}_v + \boldsymbol{C}_v\boldsymbol{\nu} + \boldsymbol{D}_{RB}\boldsymbol{\nu} + \boldsymbol{g}_{RB} + \boldsymbol{\tau}_{v,C}\right]\\ &\quad - \tilde{\boldsymbol{\theta}}_v^{\mathrm{T}}\boldsymbol{K}_\theta\dot{\hat{\boldsymbol{\theta}}}_v + k_p\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\boldsymbol{R}_B^I\boldsymbol{s}_p - k_p\lambda_p\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_1\\ &\quad + k_o\tilde{\varepsilon}^{\mathrm{T}}\boldsymbol{s}_o - \lambda_o k_o\tilde{\varepsilon}^{\mathrm{T}}\tilde{\varepsilon},\end{aligned} \tag{3.35}$$

that can be rewritten as:

$$\begin{aligned}\dot{V} &= \boldsymbol{s}_v^{\mathrm{T}}\left[\boldsymbol{\Phi}_{v,T}\boldsymbol{\theta}_v - \boldsymbol{\tau}_v\right] - \boldsymbol{s}_v^{\mathrm{T}}\boldsymbol{D}_{RB}\boldsymbol{s}_v - \tilde{\boldsymbol{\theta}}_v^{\mathrm{T}}\boldsymbol{K}_\theta\dot{\hat{\boldsymbol{\theta}}}_v\\ &\quad + \boldsymbol{s}_v^{\mathrm{T}}\boldsymbol{K}\tilde{\boldsymbol{y}} - k_p\lambda_p\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_1 - k_o\lambda_o\tilde{\varepsilon}^{\mathrm{T}}\tilde{\varepsilon}\,.\end{aligned}$$

By considering the control law (3.28) and the parameters update (3.29), it is:

$$\dot{V} = -\boldsymbol{s}_v^{\mathrm{T}}(\boldsymbol{K}_D + \boldsymbol{D}_{RB})\boldsymbol{s}_v - k_p\lambda_p\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_1 - k_o\lambda_o\tilde{\varepsilon}^{\mathrm{T}}\tilde{\varepsilon}$$

that is negative semi-definite over the state space $\{\tilde{\boldsymbol{\eta}}_1, \tilde{z}, \boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v\}$.

It is now possible to prove the system stability in a Lyapunov-Like sense using the Barbălat's Lemma. Since

- V is lower bounded
- $\dot{V}(\tilde{\boldsymbol{\eta}}_1, \tilde{z}, \boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v) \leq 0$
- $\dot{V}(\tilde{\boldsymbol{\eta}}_1, \tilde{z}, \boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v)$ is uniformly continuous

then

- $\dot{V}(\tilde{\boldsymbol{\eta}}_1, \tilde{z}, \boldsymbol{s}_v, \tilde{\boldsymbol{\theta}}_v) \to 0$ as $t \to \infty$.

Thus $\tilde{\boldsymbol{\eta}}_1, \tilde{\varepsilon}, \boldsymbol{s}_v \to \boldsymbol{0}$ as $t \to \infty$. In view of the definition of $\boldsymbol{s}_v$, this implies that $\tilde{\boldsymbol{\nu}} \to \boldsymbol{0}$ as $t \to \infty$; in addition, due to the properties of the quaternion, it results that $\tilde{\eta} \to 1$ as $t \to \infty$. However, as usual in adaptive control schemes, it is not possible to prove asymptotic stability of the whole state since $\tilde{\boldsymbol{\theta}}_v$ is only guaranteed to be bounded.

Compensation of the persistent effects. The control law has been designed explicitly to take into account the persistent dynamic effects in their proper frame. For this reason it does not suffer from the drawbacks of compensating the restoring forces in the earth-fixed frame, or, dually, the current parameters in the body-fixed frame.

Reduced controller. According to the aim to design a PD-like action plus an adaptive compensation action, the reduced version of the control law is then given by:

$$\boldsymbol{\tau}_v = \boldsymbol{K}_D\boldsymbol{s}_v + \boldsymbol{\Phi}_{v,P}\hat{\boldsymbol{\theta}}_{v,P} \tag{3.36}$$

$$\dot{\hat{\boldsymbol{\theta}}}_{v,P} = \boldsymbol{K}_\theta^{-1}\boldsymbol{\Phi}_{v,P}^{\mathrm{T}}\boldsymbol{s}_v\,, \tag{3.37}$$

where $\boldsymbol{K}_D \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{K}_\theta \in \mathbb{R}^{9\times 9}$ are positive definite matrices of gains to be designed.

3.7 Jacobian-Transpose-Based Controller

Sun and Cheah, in 2003 [31], present an adaptive set-point control inspired by the manipulator control approach proposed in [39].

The vector $\tilde{\boldsymbol{\eta}} \in \mathbb{R}^6$ represents, as usual, the error in the earth-fixed frame. Let further define the function $\boldsymbol{F}$ defined on the single components such as:

- It holds: $F_i(x_i) > 0 \quad \forall x_i \neq 0, \qquad F_i(0) = 0;$
- The function $F_i(x_i)$ is twice continuously differentiable;
- The partial derivative $f_i = \partial F_i/\partial x_i$ is strictly increasing in x_i for $|x_i| < \gamma_i$ for a given γ_i and saturated for $|x_i| \geq \gamma_i$;
- There exists a positive constant c_i such that $F_i(x_i) > c_i^2 f_i$

The function $\boldsymbol{f} \in \mathbb{R}^6$ collecting the elements f_i is basically a saturated function. Guidelines in its selection are given in [39].

The proposed controller is given by:

$$\boxed{\boldsymbol{\tau}_v = \boldsymbol{K}_P \boldsymbol{J}_e^{\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{f}(\tilde{\boldsymbol{\eta}}) - \boldsymbol{K}_D \boldsymbol{\nu} + \boldsymbol{\Phi}_{v,R}(\boldsymbol{R}_B^I)\hat{\boldsymbol{\theta}}_{v,R}} \tag{3.38}$$

where $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are positive definite matrices, selected as scalar in [31] and

$$\boxed{\dot{\hat{\boldsymbol{\theta}}}_{v,R} = \boldsymbol{K}_\theta^{-1} \boldsymbol{\Phi}_{v,R}^{\mathrm{T}}(\boldsymbol{R}_B^I)\left(\boldsymbol{\nu} + \alpha \boldsymbol{J}_e^{\mathrm{T}}(\boldsymbol{R}_B^I)\boldsymbol{f}(\tilde{\boldsymbol{\eta}})\right)} \tag{3.39}$$

where $\alpha > 0$ is a scalar gain.

It can be noticed that the proportional action is similar to the action proposed by Fossen and Balchen in Sect. 3.5. The derivative action is, on the contrary, based on the vehicle-fixed variable $\boldsymbol{\nu}$ that is different from the derivative action developed by Fossen and Balchen based on the term $\boldsymbol{J}_e^{\mathrm{T}}\dot{\tilde{\boldsymbol{\eta}}}$. Finally they both have an adaptive action that is already reduced to the sole restoring terms in [31].

It is worth noticing that the Authors propose scalar gains $\boldsymbol{K}_P$, $\boldsymbol{K}_D$ and α for the controller. There is a main problem related with the fact that the position and orientation variables have different unit measures; using the same gains might force the designer to tune the performance to the lower bandwidth.

Compensation of the persistent effects. The controller proposed by the Authors compensates for the gravity in the vehicle-fixed frame. However, the update law is based on the transpose of the Jacobian, and not on its inverse; the mapping, thus, is not exact and a coupling among the error directions is experienced. This has as a consequence that this control law is not suitable for the restoring force compensation. It is worth noticing, moreover, that, differently from the industrial manipulator case, inversion of the Jacobian is not computational demanding since, being $\boldsymbol{J}_e$ a simple (6×6) matrix (see Eq. (2.19)), its inverse $\boldsymbol{J}_e^{-1}$ can be symbolically computed; from a computational aspect, thus, there is no difference in using $\boldsymbol{J}_e^{\mathrm{T}}$ or $\boldsymbol{J}_e^{-1}$. By visual inspection of $\boldsymbol{J}_e^{\mathrm{T}}$ and $\boldsymbol{J}_e^{-1}$ it can be observed that the difference is in the rotational part $\boldsymbol{J}_{k,o}$, being the transpose of a rotation matrix equal to its inverse. In particular, using the common definition that roll pitch an yaw means the use of elementary rotation around x, y and z in fixed frame [35], the corresponding matrix $\boldsymbol{J}_{k,o}$ is not function of the yaw angle, in case of null pitch angle it is $\boldsymbol{J}_{k,o}^{\mathrm{T}} = \boldsymbol{J}_{k,o}^{-1}$, and it is singular for a pitch angle of $\pm\pi/2$; close to that singularity the numerical difference between $\boldsymbol{J}_{k,o}^{\mathrm{T}}$ and $\boldsymbol{J}_{k,o}^{-1}$ increases.

The ocean current is not taken into account. Using the controller (3.38)–(3.39) under the effect of the current would lead to an error different from zero at steady state.

Reduced controller. The aim of the Authors is to propose already a controller that matches the definition of *reduced* controller that has been given here. The absence of compensation for the current, however, makes the controller not appealing for practical implementation. The proposed reduced controller, modified to take into account a tracking problem, is given by:

$$\boldsymbol{\tau}_v = \boldsymbol{K}_P \boldsymbol{J}_e^{\mathrm{T}}(\boldsymbol{R}_B^I)\tilde{\boldsymbol{\eta}} + \boldsymbol{K}_D \tilde{\boldsymbol{\nu}} + \boldsymbol{\Phi}_{v,P} \hat{\boldsymbol{\theta}}_{v,P} \tag{3.40}$$

$$\dot{\hat{\boldsymbol{\theta}}}_{v,P} = \boldsymbol{K}_\theta^{-1} \boldsymbol{\Phi}_{v,P}^{\mathrm{T}} \left(\tilde{\boldsymbol{\nu}} + \boldsymbol{\Lambda} \boldsymbol{J}_e^{\mathrm{T}}(\boldsymbol{R}_B^I)\tilde{\boldsymbol{\eta}} \right) \tag{3.41}$$

where $\boldsymbol{\Lambda}$, $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are positive definite matrices, selected at least as block-diagonal matrices to keep different dynamics for the position and the orientation.

It is worth noticing that the reduced controller derived is different from the original controller, that would not reach a null steady state error under the effect of the current. Also, the regressor $\boldsymbol{\Phi}_{v,P}$ embeds the gravity regressor as proposed by Sun and Cheah but the drawback in the restoring compensation still exist due to the not proper update law of the parameters.

3.8 Comparison Among Controllers

For easy of readings, Table 3.1 reports the label associated with each controller. The controllers developed by the researchers are quite different one each other, in this Section a qualitative comparison with respect to the compensation performed by the controllers with respect to the persistent dynamic effects is provided.

3.8.1 Compensation of the Restoring Generalized Forces

The controller **A** is totally conceived in the earth-fixed frame and it is model-based. In case of perfect knowledge of the restoring-related dynamic parameters, thus, there is not need to compensate for the restoring generalized forces. This is, however, unpractical in a real situation; in case of partial compensation of the vector $\boldsymbol{g}^{\star}_{RB}$, in fact, this controller experiences the drawback discussed in Sect. 3.2. A similar drawback is shared from the controller **B** that, again, is totally based on earth-fixed variables, the integral actions, thus, does not compensate optimally for the restoring action. The controller **F** compensates the restoring force in the vehicle-fixed frame; the update law, however, is not based on the exact mapping between orientation and moments resulting in a not clean adaptive action. The effect, however, is as significant as the vehicle works with a large pitch angle. The remaining 3 controllers properly adapt the restoring-related parameters in the vehicle-fixed frame.

3.8.2 Compensation of the Ocean Current

The presence of an ocean current is seldom taken into account in the literature. The effect of the current, however, can be really significant [17, 40]. In this discussion a main assumption is made: roughly speaking the current is considered as constant in the earth-fixed frame and its effect on the vehicle is modeled as a force, proportional to the current magnitude, that *pushes* the vehicle. The considerations below, thus, are based on this assumption. It is worth noticing that in the simulation the dynamic effect of the current is correctly taken into account by computing the relative velocity in the vehicle model. The accuracy of this assumption, thus, seems to be verified.

Controller **C** and **F** does not consider at all the current, as a result the developed controller does not reach a null steady state error. Controller **D** compensates the current with a term designed on the purpose; this term adapts on vehicle-fixed parameters, as discussed in Sect. 3.5 this action experiences a drawback. The remaining controllers correctly compensate for the current adapting/integrating on a set of earth-fixed based parameters.

Table 3.2 summarizes the discussed properties of the six controllers with respect to the persistent dynamic terms. It is worth noticing that only the controller **E** correctly compensates for both actions.

3.9 Numerical Comparison Among the Reduced Controllers

A performance comparison among the controllers discussed has been developed by numerical simulations. Two different set of have been run, namely on the 6-DOF mathematical model of ODIN, an AUV built at the Autonomous Systems Laboratory of the University of Hawaii and on a generic ellipsoidal-shaped rigid body. Both models are reported in the appendix.

Table 3.2 Summary of the behavior of the controllers with respect to compensation of the hydrodynamic persistent terms

Control law	Current effect	Restoring forces	Null error with current
A	Fit	Unfit	Yes
B	Fit	Unfit	Yes
C	Unfit	Fit	No
D	Unfit	Fit	Yes
E	Fit	Fit	Yes
F	Unfit	Unfit	No

An aspect to be considered for proper reading of this study is that the control problem at hand is highly nonlinear, coupled and the controller to be compared are intrinsically different one from the other. Any effort to chose the gains so as to ensure *similar* performance to the controllers has been made; nevertheless, this is impossible in a strict sense. For this reason, the presented results have to be interpreted mainly looking at the error behavior rather then focusing on direct numeric comparison. On the other hand, it was chosen to not emphasize vehicle-related effects that would not be present in general applications. In particular, in the first case study ODIN has a small metacentric height, yielding low restoring moments; simulations of the controllers to vehicles of larger metacentric height would make even more evident the drawbacks in the compensation of the restoring forces.

Test trajectory. In order to demonstrate the effects discussed in this chapter, the simplest task to be considered is one involving successive changes of the vehicle orientation in presence of ocean current. The simulation length can be divided in different period of 60 s duration. The vehicle is firstly put in the water at the position

$$\eta_d(t=0) = \begin{bmatrix} 0 \; 0 \; 1 \; 0 \; 0 \; 0 \end{bmatrix}^{\mathrm{T}} \quad [\text{m/deg}]$$

without knowledge of the current but with an estimation of the restoring parameters. The first 60 s are used to adapt the effect of the current. In the successive period the vehicle is required to move in roll and pitch from 0 to 10 and −15 deg, respectively and come back to the original configuration. In the successive two periods the vehicle is required to move of 90 deg in yaw and come back to the initial position. Finally, 60 s of steady state are given. Figure 3.3 plots the desired trajectory.

Ocean current. All the simulations have been run considering the following constant current

$$\nu_c^I = \begin{bmatrix} 0 \; 0.3 \; 0 \; 0 \; 0 \; 0 \end{bmatrix}^{\mathrm{T}} \quad \text{m/s}\,,$$

Vehicle initial position. The vehicle initial position for both case studies is

$$\eta = \begin{bmatrix} 0 \; 0 \; 1 \; 0 \; 0 \; 0 \end{bmatrix}^{\mathrm{T}} \quad [\text{m/deg}]\,,$$

meaning that the vehicle is supposed to start its motion under the water, at 1 m depth, moreover, there is no initial estimation of the ocean current.

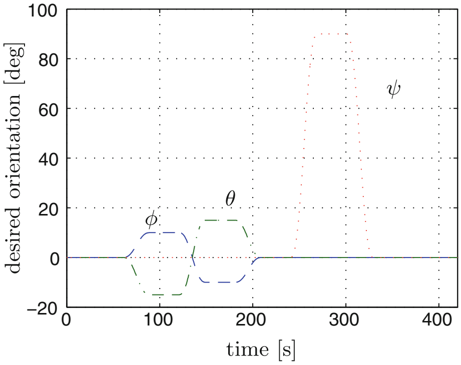


Fig. 3.3 Desired trajectory for both case studies; the desired position is constant

3.9.1 First Case Study: ODIN

Controller initial conditions. Table 3.3 reports the initial condition for the adaptive/integral parameters of the controllers, it can be observed that the same values have been used in the estimation of the restoring forces. This is also true for the controller **A**, where the gravity estimation $\hat{\boldsymbol{g}}^{\star}_{RB}$, even if not adaptive, is obtained resorting to

$$\hat{\boldsymbol{\theta}}_{v,R} = \left[-9\ 0\ 0\ 50\right]^{\mathrm{T}} .$$

From the model in the appendix it can be noticed that the true value of the restoring parameters is

$$\boldsymbol{\theta}_{v,R} = \left[-8.0438\ 0\ 0\ 61.3125\right]^{\mathrm{T}} .$$

Controller gains selection. All the simulated controllers are non-linear. Moreover, significant differences arise among them, this is, in fact, the object of this discussion. For these reasons it is very difficult to perform a fair numerical comparison and the selection of the gains is very delicate. The gains have been tuned so as to give

Table 3.3 First case study

Law	Number of par.	$\hat{\boldsymbol{\theta}}(t=0)$
A	7	$\left[0\ 0\ 0\ 0\ 0\ 0\ 1\right]^{\mathrm{T}}$ (Initial value of the integral)
B	–	Not simulated
C	10	$\left[-9\ 0\ 50\ 0\ 0\ 0\ 0\ 0\ 0\ 0\right]^{\mathrm{T}}$
D	10	$\left[-9\ 0\ 50\ 0\ 0\ 0\ 0\ 0\ 0\ 0\right]^{\mathrm{T}}$
E	9	$\left[0\ 50\ 0\ 0\ 0\ 9\ 0\ 0\ 0\right]^{\mathrm{T}}$
F	9	$\left[0\ 50\ 0\ 0\ 0\ 9\ 0\ 0\ 0\right]^{\mathrm{T}}$

Initial conditions for all the controllers in their reduced version and number of parameter to adapt/integrate to the controller similar control effort in terms of force/moment magnitude and eventual presence of chattering. In few cases, since the controller feeds back the same variable, it has been possible to chose the same control gain, it is the case, e.g., of the controllers **C** and **E**. In other cases, such as, e.g., the controller **F**, the use of the transpose of the Jacobian implicitly change the effective gain acting on the error, the equivalence of the gains, thus, is only apparent.

A separate discussion needs to be done for the controller **B**. Its specific structure makes it really hard to tune the parameters following these simple considerations. As recognized in [17], moreover, the parameter tuning of their controller has not been simple and handy for being used in the experiments with the vehicle ODIN III. For this reason, while the controller has been object of the theoretical discussion, the simulation will not be reported.

The following gains have been used for the controller **A**:

$$\begin{aligned} \boldsymbol{K}_P &= \text{blockdiag}\{8.8\boldsymbol{I}_3, 20\boldsymbol{I}_4\}\,, \\ \boldsymbol{K}_D &= \text{blockdiag}\{110\boldsymbol{I}_3, 40\boldsymbol{I}_4\}\,, \\ \boldsymbol{K}_I &= \text{blockdiag}\{0.2\boldsymbol{I}_3, 1\boldsymbol{I}_4\}\,. \end{aligned}$$

The following gains have been used for the controller **C**:

$$\begin{aligned} \boldsymbol{K}_D &= \text{blockdiag}\{110\boldsymbol{I}_3, 40\boldsymbol{I}_3\}\,, \\ \boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 0.9\boldsymbol{I}_3\}\,, \\ \boldsymbol{K}_\theta^{-1} &= 2\boldsymbol{I}_9\,. \end{aligned}$$

The following gains have been used for the controller **D**:

$$\begin{aligned} \boldsymbol{K}_D &= \text{blockdiag}\{110\boldsymbol{I}_3, 40\boldsymbol{I}_3\}\,, \\ \boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 0.9\boldsymbol{I}_3\}\,, \\ \boldsymbol{K}_\theta^{-1} &= 2\boldsymbol{I}_4\,, \\ \boldsymbol{W}^{-1} &= 2\boldsymbol{I}_6\,. \end{aligned}$$

The following gains have been used for the controller **E**:

$$\begin{aligned} \boldsymbol{K}_D &= \text{blockdiag}\{110\boldsymbol{I}_3, 40\boldsymbol{I}_3\}\,, \\ \boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 0.9\boldsymbol{I}_3\}\,, \\ \boldsymbol{K}_{\theta_P}^{-1} &= 2\boldsymbol{I}_9\,. \end{aligned}$$

The following gains have been used for the controller **F**:

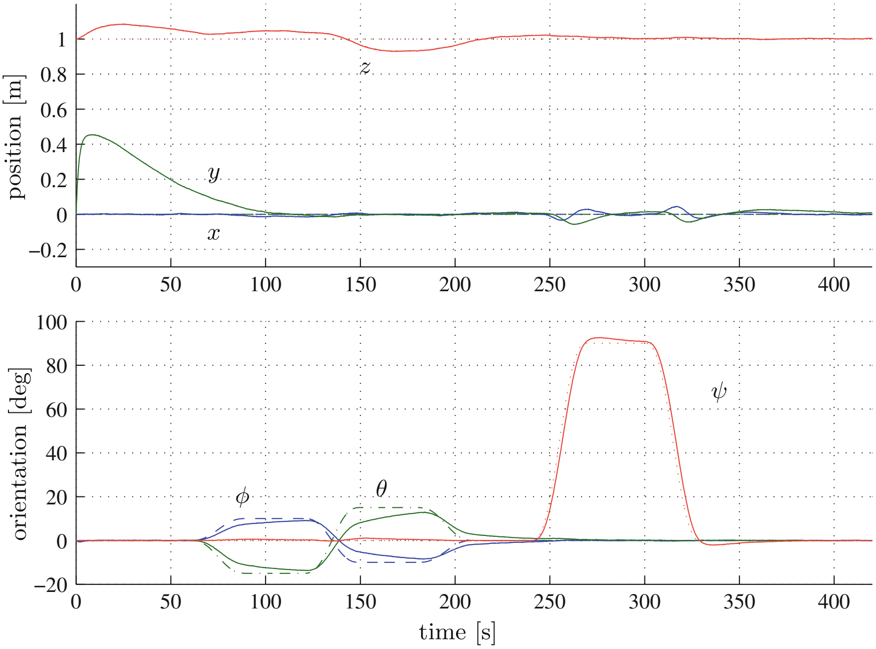


Fig. 3.4 First case study. Simulated position and orientation for the reduced controller **A**

$$
\begin{aligned}
\boldsymbol{K}_D &= \text{blockdiag}\{8.8\boldsymbol{I}_3, 36\boldsymbol{I}_3\}\,,\\
\boldsymbol{K}_V &= \text{blockdiag}\{110\boldsymbol{I}_3, 40\boldsymbol{I}_3\}\,,\\
\boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 0.9\boldsymbol{I}_3\}\,,\\
\boldsymbol{K}_{\theta_P}^{-1} &= 2\boldsymbol{I}_9\,.
\end{aligned}
$$

Results. In Fig. 3.4 the position and orientation for controller **A** are given; it can be noticed that in the first 60 s the control needs to adapt with respect to the current and a movement along y is observed, also, it can be observed that, during the horizontal rotation ($t \in [240, 360]$ s), the controller correctly compensates for the current and a very small coupling is experienced. The drawback in the restoring compensation of this controller is not significant in this numerical case study. Figure 3.5 reports the required control force and moment.

Figures 3.6 and 3.7 report the position, orientation and control effort for controller **C**. It can be observed a strong coupling during the commanded yaw rotation caused by the controller itself. This can be further appreciated in Fig. 3.8 where the projection of the vehicle position on the xy plane during the rotation compared with controller **A** is given.

Figures 3.9 and 3.10 report the position, orientation and control effort for controller **D**. Even in this case, a strong coupling during the commanded yaw rotation caused by the controller itself can be observed.

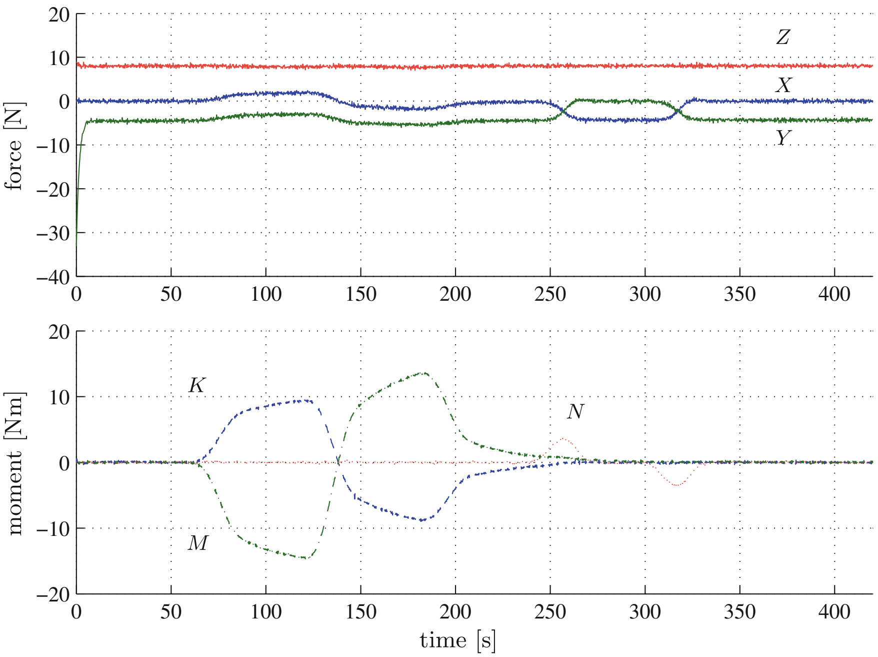


Fig. 3.5 First case study. Simulated force and moment for the reduced controller **A**

Figures 3.11 and 3.12 report the position, orientation and control effort for controller **E**.

Figures 3.13 and 3.14 report the position, orientation and control effort for controller **F**.

A further plot can be used (Fig. 3.15) to understand the problem arising when compensating incorrectly the current, the controller **C** is compared with controller **E** in a normalized polar representation of the sole compensation of the ocean current during the yaw rotation. It can be noticed that, with the controller **E** (and similar) the compensation of the current rotates with the vehicle so that it still compensate with respect to a constant, earth-fixed disturbance.

The given simulation does not exhibit a significant error in the orientation that should be caused by the wrong restoring compensation of the controllers **A** and **F**. The numerical study has been conducted with a model largely used in the literature and tested in experimental cases; moreover, reasonable parameters' estimation has been considered; in particular only the third component of the vector $W\boldsymbol{r}_G - B\boldsymbol{r}_B$ is different from zero; finally small rotations in pitch and yaw have been commanded. Under these considerations both the controllers behave very well; the control effort, moreover, was similar to that of other controllers. It has been considered fair, thus, to report the good numerical result of these controllers despite their theoretical drawback and to avoid specific case studies where those might fail such as, e.g., with large enough ϕ and θ for the controller **F** and a different vector $W\boldsymbol{r}_G - B\boldsymbol{r}_B$. It

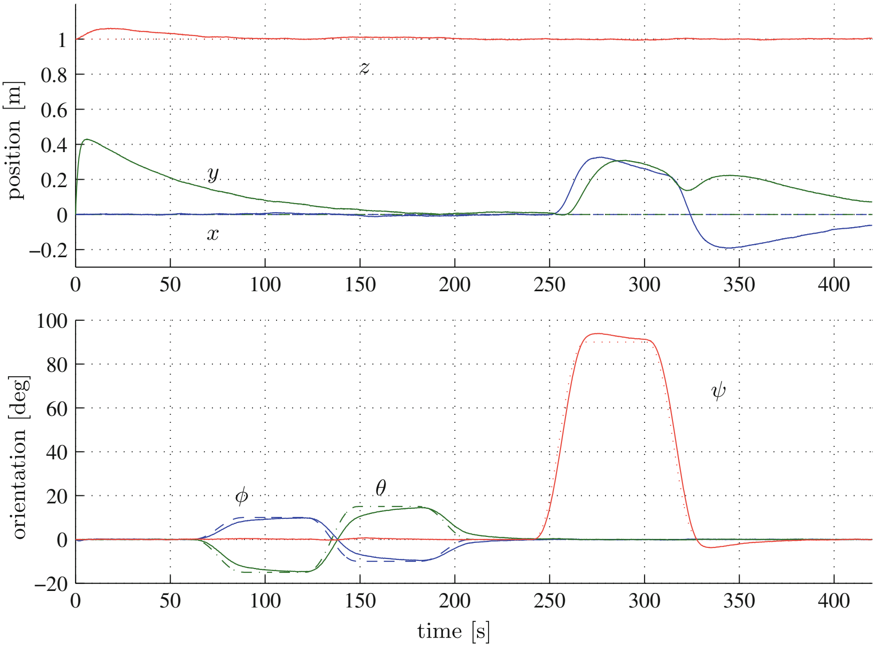


Fig. 3.6 First case study. Simulated position and orientation for the reduced controller **C**

might be noticed that this singularity is a representation singularity and that might arise even with the vehicle in the hovering position due to an unsatisfactory choice of the body-fixed frame or the roll-pitch-yaw convention; however, due to the common marine convention, this possibility is so uncommon that is not a real drawback.

However, to better illustrate this different behavior, a second simulation study was developed in which the vehicle is commanded to change the sole roll angle according to the time law in Fig. 3.16 while keeping constant the vehicle position and the other vehicle angles at zero; for the sake of clarity, it is assumed that no ocean current is acting on the vehicle. The behavior of the reduced version of control law **E** is compared to that of the reduced version of the control law **A** in terms of the measured roll angles (Fig. 3.17, left column). Remarkably, in the simple condition considered, the restoring moment to be compensated for is mainly acting around the $\boldsymbol{x}_b$ axis. In the simulation, it is possible to compute the restoring moment around this axis and compare it with the adaptive compensation of the control law **E** and the integral compensation plus the model-based compensation of the controller **A** respectively; those plots are reported in the right column of Fig. 3.17. It can be observed that during the phase in which the roll angle changes from $+10\,^\circ$ to $-10\,^\circ$ the compensation built by the control law **A** has a lag with respect to the acting restoring moment; this is due to the integral charge/discharge time required to build a time-varying compensation term. The control law **E**, instead, performs proper compensation of the restoring moment since it accounts for the vehicle orientation in the adaptation mechanism.

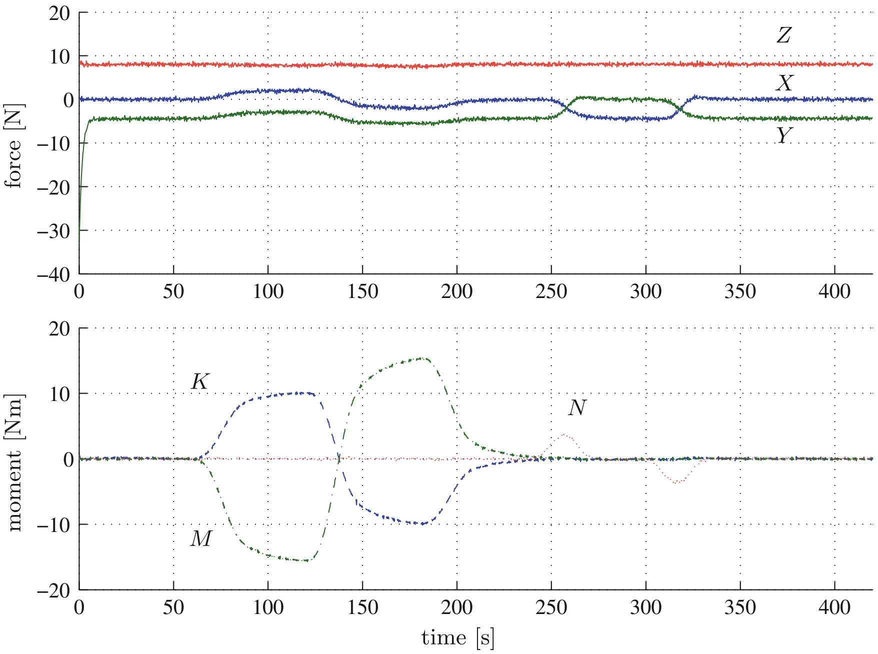
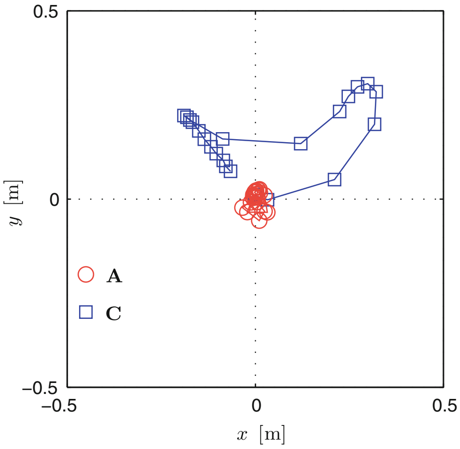


Fig. 3.7 First case study. Simulated force and moment for the reduced controller **C**

Fig. 3.8 First case study. Comparison of the position on the plane xy for the reduced controller **A** and **C**



3.9.2 Second Case Study: Ellipsoidal Shape

ODIN is a sphere, some additional insights may be appreciated by considering an ellipse instead. The anisotropy, in fact, reduces the positive effects of the controllers that correctly compensate for the ocean current.

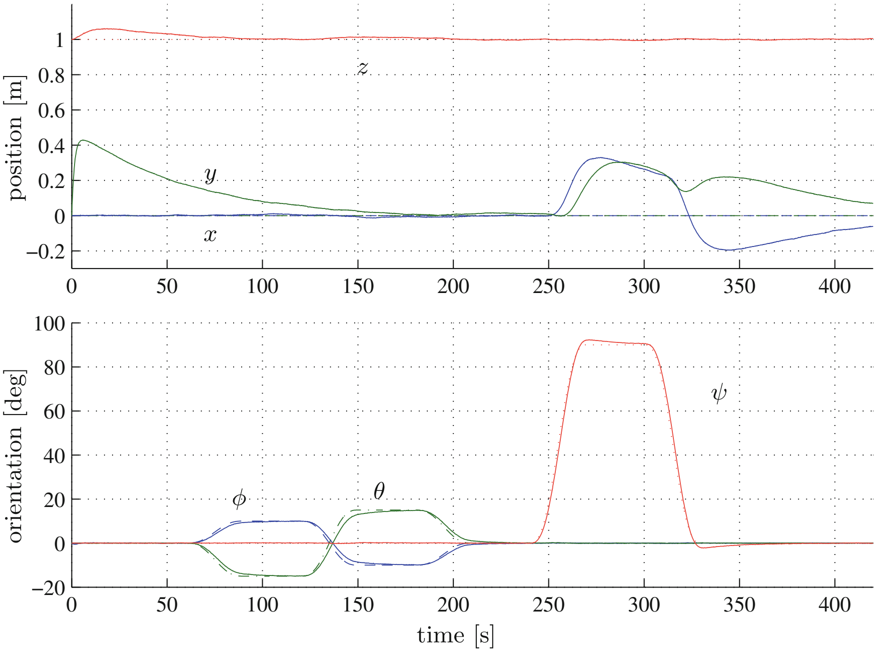


Fig. 3.9 First case study. Simulated position and orientation for the reduced controller **D**

The dynamic parameters of the considered model are listed in Sect. A.4. In view of their digital implementation, all the controllers have been simulated in discrete time with a sampling interval of 0.2 s; the sensor data acquisition has been simulated at the same frequency [41] by assuming the measurement noise to be white and zero-mean. The noise power is 10^{-7} for the position/orientation and 10^{-6} for the linear and angular velocities of proper unit measures.

Controller initial conditions. Table 3.4 reports the initial condition for the adaptive/integral parameters of the controllers for the second case study, in this case too, the same values have been used in the estimation of the restoring forces. This is also true for the controller **A**, where the gravity estimation $\hat{\boldsymbol{g}}^{\star}$, even if not adaptive, is obtained resorting to

$$\hat{\boldsymbol{\theta}}_R = \left[-20\ 0\ 0\ 100\right]^{\mathrm{T}}.$$

From the model in the Appendix it can be noticed that the true value of the restoring parameters is

$$\boldsymbol{\theta}_R = \left[-22.1897\ 0\ 0\ 109.8390\right]^{\mathrm{T}}.$$

Controller gains selection. The same consideration made on the selection of the controllers' gains for the first case study still hold. The following gains have been used for the controller **A**:

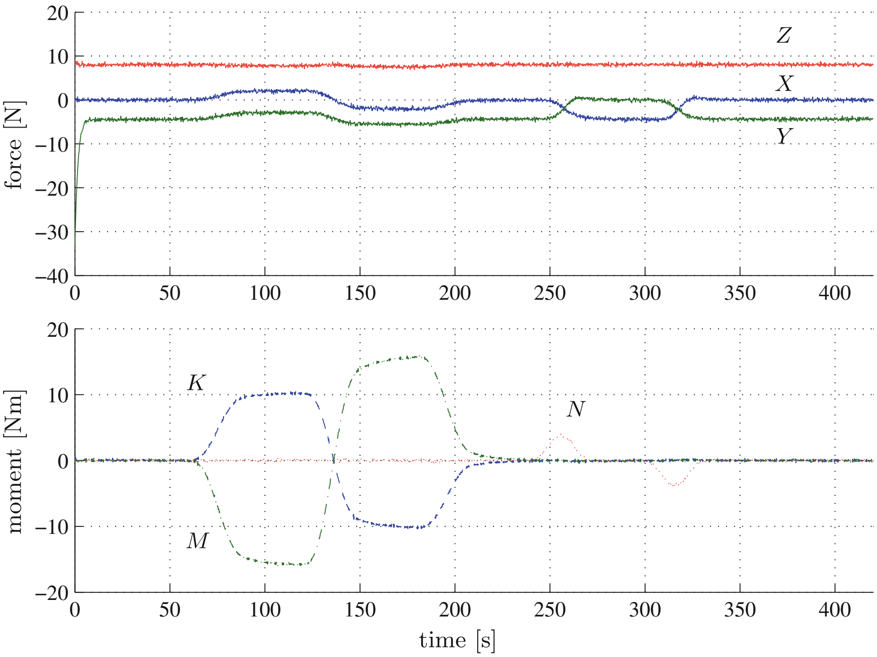


Fig. 3.10 First case study. Simulated force and moment for the reduced controller **D**

$$
\begin{aligned}
\boldsymbol{K}_P &= \text{blockdiag}\{17.6\boldsymbol{I}_3, 33, 99\boldsymbol{I}_3\}\\
\boldsymbol{K}_D &= \text{blockdiag}\{220\boldsymbol{I}_3, 30, 90\boldsymbol{I}_3\}\\
\boldsymbol{K}_I &= \text{blockdiag}\{0.4\boldsymbol{I}_3, 2\boldsymbol{I}_4\}
\end{aligned}
$$

The following gains have been used for the controller **C**:

$$
\begin{aligned}
\boldsymbol{K}_D &= \text{blockdiag}\{220\boldsymbol{I}_3, 30, 90\boldsymbol{I}_2\}\\
\boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 1.1\boldsymbol{I}_3\}\\
\boldsymbol{K}_\theta^{-1} &= 4\boldsymbol{I}_9
\end{aligned}
$$

The following gains have been used for the controller **D**:

$$
\begin{aligned}
\boldsymbol{K}_D &= \text{blockdiag}\{220\boldsymbol{I}_3, 30, 90\boldsymbol{I}_2\}\\
\boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 1.1\boldsymbol{I}_3\}\\
\boldsymbol{K}_\theta^{-1} &= 4\boldsymbol{I}_4\\
\boldsymbol{W}^{-1} &= 4\boldsymbol{I}_6
\end{aligned}
$$

The following gains have been used for the controller **E**:

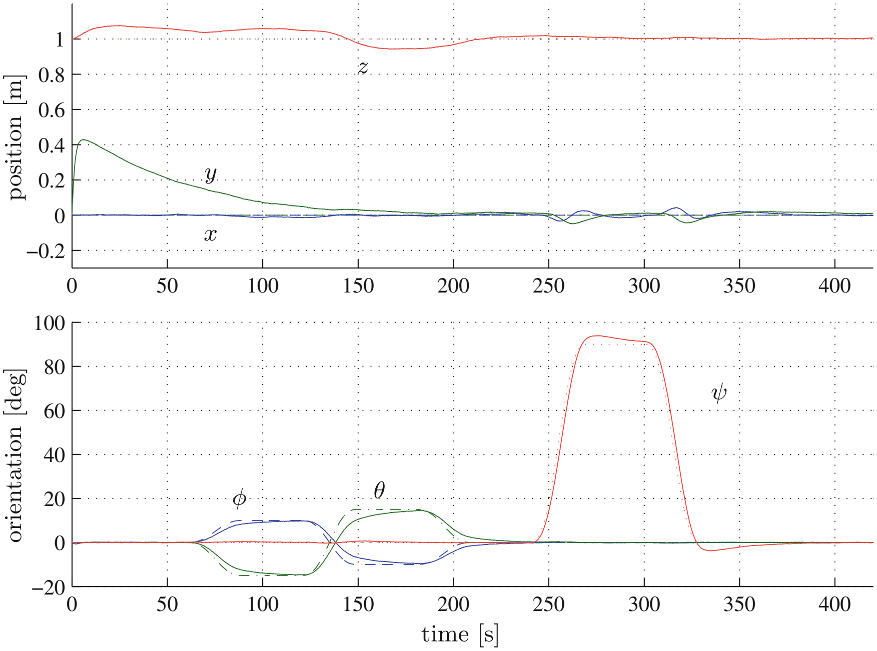


Fig. 3.11 First case study. Simulated position and orientation for the reduced controller **E**

$$
\begin{aligned}
\boldsymbol{K}_D &= \text{blockdiag}\{220\boldsymbol{I}_3, 30, 90\boldsymbol{I}_2\}\\
\boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 1.1\boldsymbol{I}_3\}\\
\boldsymbol{K}_{\theta_P}^{-1} &= 4\boldsymbol{I}_9
\end{aligned}
$$

The following gains have been used for the controller **F**:

$$
\begin{aligned}
\boldsymbol{K}_D &= \text{blockdiag}\{17.6\boldsymbol{I}_3, 33, 99\boldsymbol{I}_2\}\\
\boldsymbol{K}_V &= \text{blockdiag}\{220\boldsymbol{I}_3, 30, 90\boldsymbol{I}_2\}\\
\boldsymbol{\Lambda} &= \text{blockdiag}\{0.08\boldsymbol{I}_3, 1.1\boldsymbol{I}_3\}\\
\boldsymbol{K}_{\theta_P}^{-1} &= 4\boldsymbol{I}_9
\end{aligned}
$$

Results. As for the first case study, for the first reduced controller let us plot all the relevant variables. In Fig. 3.18 the positions are given, in the first 60 s the control needs to adapt with respect to the current and a movement along y is observed, also, during the horizontal rotation ($t \in [240, 360]$ s), the controller correctly still correctly compensate for the current and a very small coupling is experienced. Figure 3.19 reports the orientation, the drawback of this controller is not significant in this numerical case study. Figures 3.20 and 3.21 report the required control

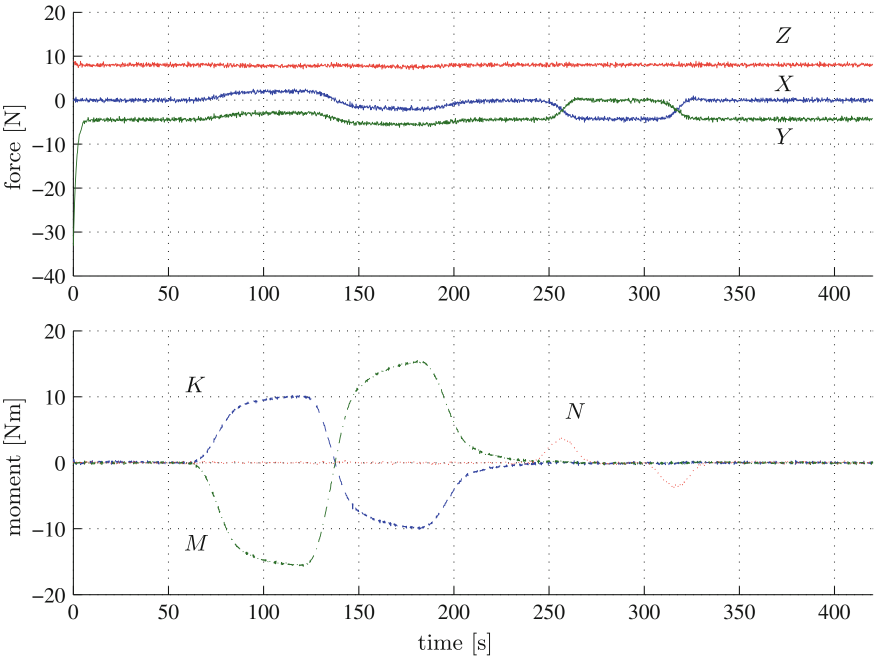


Fig. 3.12 First case study. Simulated force and moment for the reduced controller **E**

force and moment. The control effort is similar for all the controllers, meaning that the different behaviors are not given by a different magnitude of the inputs.

The simulation of the reduced controller **C** is dual with respect to the control **A**. In the former, in fact, a wrong current compensation is experienced. This can be noticed by comparing the xy position of the vehicle during the yaw rotation; Figures 3.22 and 3.23 show the relevant variable for the proper time interval. It can be noticed that the controller **C** is compensating the current in body-fixed frame and, during the rotation, the controller itself is feeding the system with a disturbance. Given the comments made on the current compensation, an additional consideration concerns the symmetry of the vehicle: the *current compensation* of the controllers **A**, **B** and **E** is as much performing as the geometry is close to a sphere. This can be noticed in the first case study where a spherical model is used for the simulations. Similar plots might be drawn for the other controllers following the Table 3.2 to see the behavior exhibited by the different controllers.

A further plot can be used (Fig. 3.24) to understand this aspect, the controller **C** is compared with controller **E** in a normalized polar representation of the sole compensation of the ocean current during the yaw rotation. It can be noticed that, with the controller **E** (and similar) the compensation of the current rotate with the vehicle so that it still compensate with respect to a constant, earth-fixed disturbance.

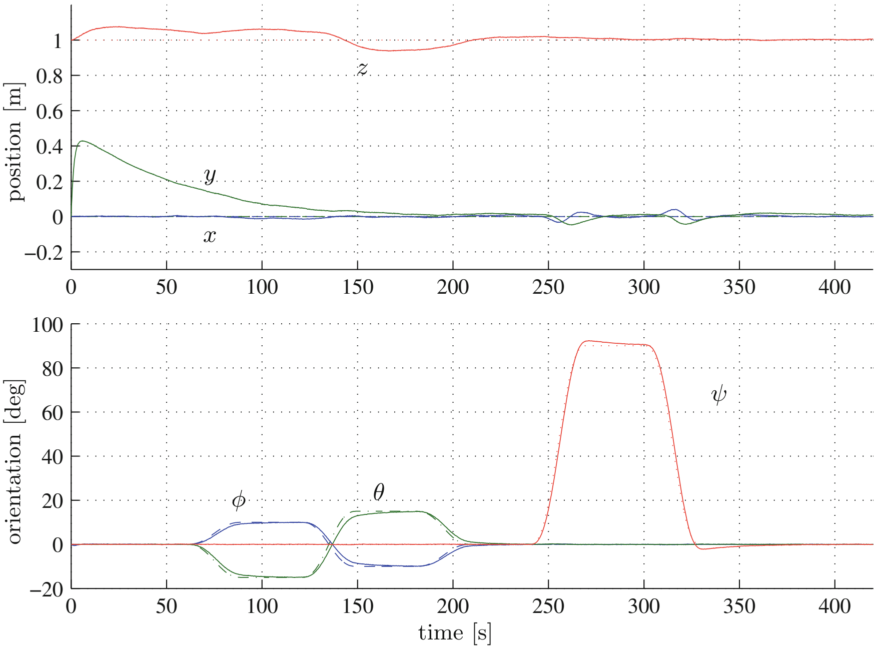


Fig. 3.13 First case study. Simulated position and orientation for the reduced controller **F**

The reduced version of controller **D** shows very similar results of the reduced version of the controller **C**, in particular, the restoring forces are correctly compensated for while the current is not, causing the shown coupling during the yaw rotation.

The given simulation does not exhibit a significant error in the orientation that should be caused by the wrong restoring compensation of the controllers **A** and **F**. The numerical study has been conducted considering a rigid body with a common shape for AUVs, an ellipsoid, moreover, reasonable parameters' estimation has been considered, in particular only the third component of the vector $W\boldsymbol{r}_G - B\boldsymbol{r}_B$ is different from zero; finally small rotations in pitch and yaw have been commanded. Under these considerations both the controllers behave very well with orientation errors practically equal than that shown in Fig. 3.19 (see Fig. 3.25), and close to that of all the simulated controllers; the control effort, moreover, was similar (see Fig. 3.26). It has been considered fair, thus, to report the good numerical result of these controllers despite their theoretical drawback and to avoid specific case studies where those might fails such as, e.g., with large enough ϕ and θ for the controller **F** and a different vector $W\boldsymbol{r}_G - B\boldsymbol{r}_B$. It might be noticed that this singularity is a representation singularity and that might arise even with the vehicle in the hovering position with a not proper choice of the body-fixed frame or the roll-pitch-yaw convention; however, due to the common marine convention, this possibility is so

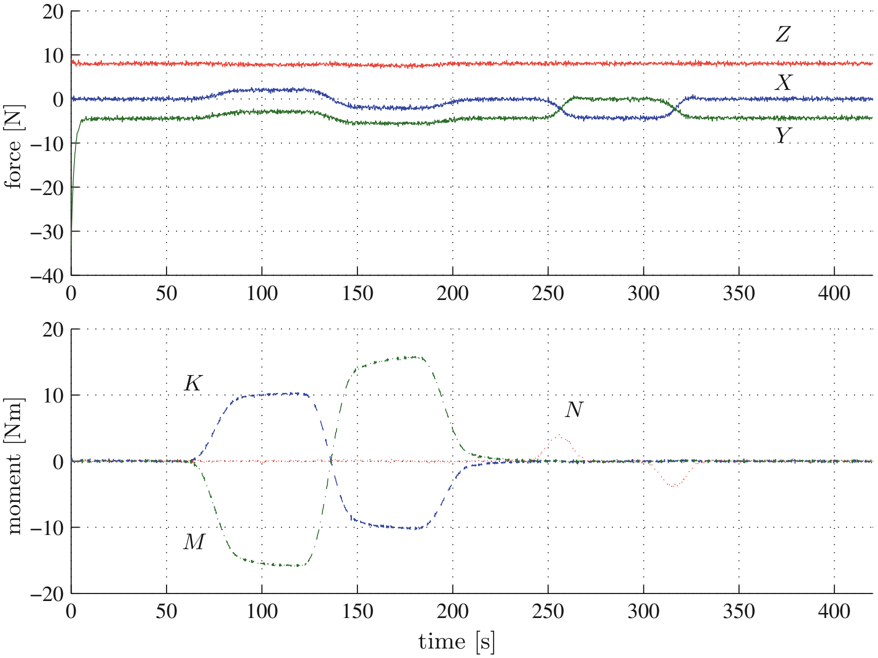


Fig. 3.14 First case study. Simulated force and moment for the reduced controller **F**

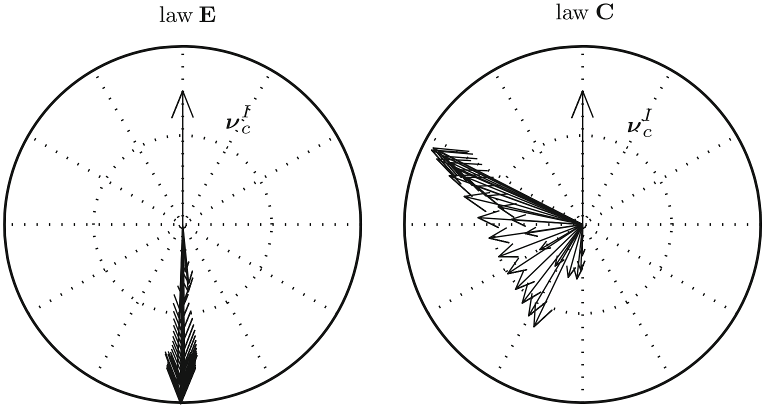


Fig. 3.15 First case study. Earth-fixed-frame polar representation of samples of the current compensation: controller **E** (*left*) and controller **C** (*right*). Notice that the compensation term built by the law **C** rotates together with the vehicle-fixed frame

uncommon that is not a real drawback. In the first case study, and in [38], a pure roll motion is simulated to magnify the drawback of the controller **A**.

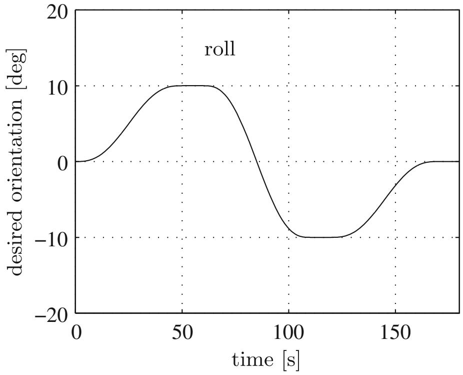


Fig. 3.16 First case study. Comparison of the control law **A** and the control law **E** in the second simulation: Time history of the desired orientation

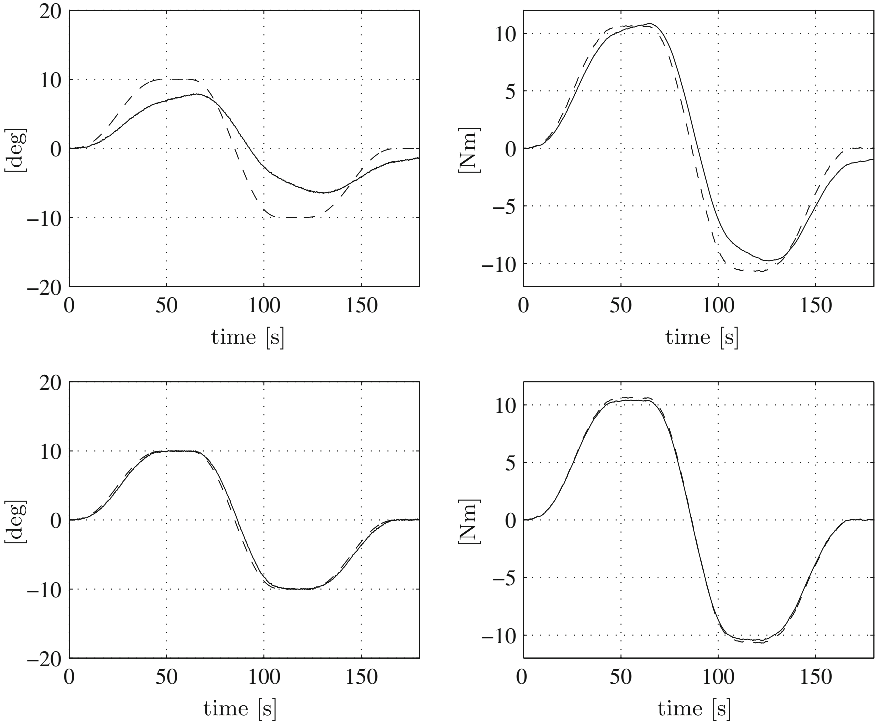


Fig. 3.17 First case study. Comparison of the control law **A** and the control law **E** in the second simulation. *Left column* Time history of measured roll angles (*solid line*) and the desired (*dashed line*); top, control law **A**, bottom, control law **E**. *Right column* Time history of the restoring moment around $\boldsymbol{x}_b$ (*dashed*), the corresponding integral action of the control law **A** (*top, solid line*) and the corresponding adaptive action of the control law **E** (*bottom, solid line*)

Table 3.4 Second case study

Law	Number of par.	$\hat{\boldsymbol{\theta}}(t=0)$
A	7	$[0\ 0\ 0\ 0\ 0\ 0\ 1]^{\mathrm{T}}$ (Initial value of the integral)
B	–	Not simulated
C	10	$[-20\ 0\ 100\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$
D	10	$[-20\ 0\ 100\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$
E	9	$[0\ 100\ 0\ 0\ 0\ 20\ 0\ 0\ 0]^{\mathrm{T}}$
F	9	$[0\ 100\ 0\ 0\ 0\ 20\ 0\ 0\ 0]^{\mathrm{T}}$

Initial conditions for all the controllers in their reduced version and number of parameter to adapt/integrate

Fig. 3.18 Second case study. Simulated position for the reduced controller **A**

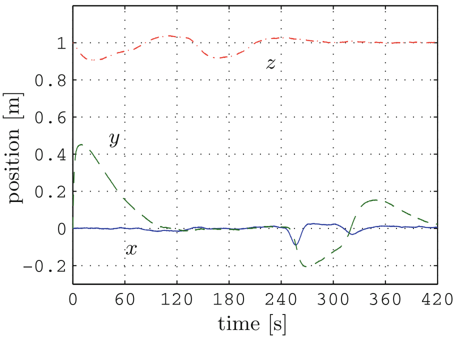
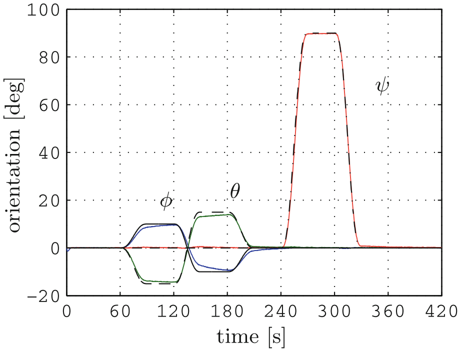


Fig. 3.19 Second case study. Simulated orientation for the reduced controller **A**



Finally, since the reduced version of the controller **F** correctly compensate for the current its behavior is similar to the controller **E**. However it is worth noticing that the original controller does not compensate at all the current.

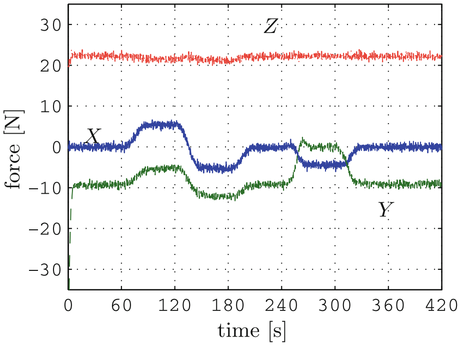


Fig. 3.20 Second case study. Simulated force for the reduced controller **A**

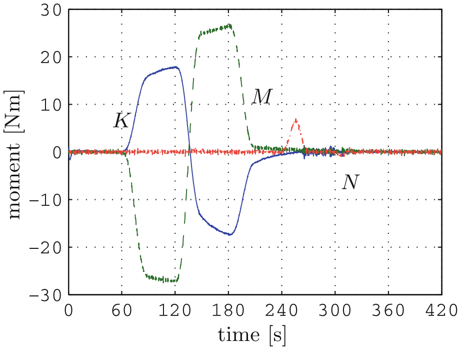


Fig. 3.21 Second case study. Simulated moment for the reduced controller **A**

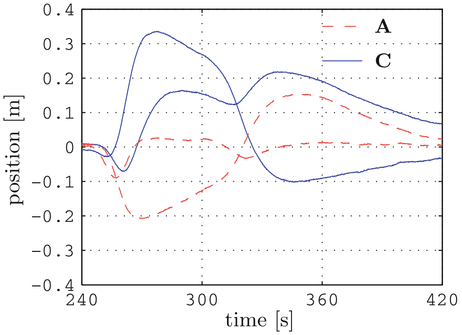


Fig. 3.22 Second case study. Coordinates x and y of the reduced controllers **A** and **C** during the yaw rotation. Controllers **B** and **E** show a behavior similar to **A**; **D** and **F** similar to **C**

3.9.3 Code to Reproduce the Simulations

The code to reproduce the simulations shown in Sects. 3.9.1 and 3.9.2 has been developed in MATLAB© and it is made available to be downloaded at the address:

http://www.eng.docente.unicas.it/gianluca_antonelli/publications/monograph.

For details please refer to the corresponding `readme.txt` file.

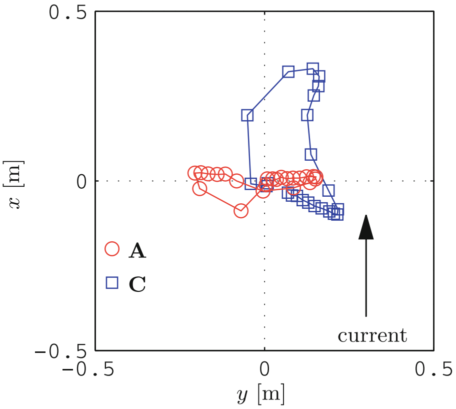


Fig. 3.23 Second case study. Horizontal position of the reduced controllers **A** and **C** during the yaw rotation. Controllers **B** and **E** show a behavior similar to **A**; **D** and **F** similar to **C**

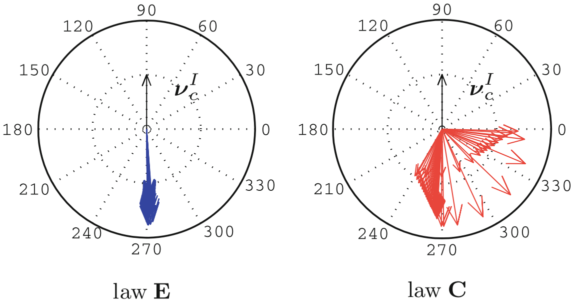


Fig. 3.24 Second case study. Earth-fixed-frame polar representation of samples of the current compensation: controller **E** (*left*) and controller **C** (*right*). Notice that the compensation term built by the law **C** rotates together with the vehicle-fixed frame

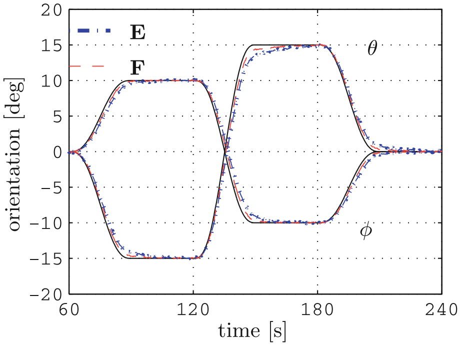


Fig. 3.25 Second case study. Orientation during the roll-pitch movement for the controllers **E** and **F**

3.10 Conclusions and Extension to UVMSs

In this chapter, six controllers have been compared for two different mathematical models with respect to their behavior in presence of modeling uncertainty and presence of ocean current. It is shown that, with a not proper compensation, the integral/adaptive action acts as a *disturbance* during the transients.

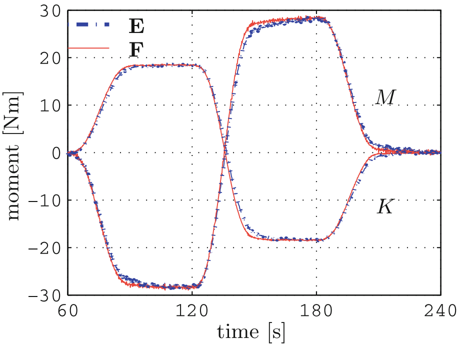


Fig. 3.26 Second case study. Moment during the roll-pitch movement for the controllers **E** and **F**

Numerical simulations better illustrate the theoretical results. However, an aspect to be considered for proper analysis of the simulations is that, despite any effort to chose the gains so as to ensure similar performance to the controllers has been made, this is impossible in a strict sense. Therefore, the presented results have to be read mainly by looking at the error behavior rather then focusing on strict numeric comparison. Notice, however, that the control effort is very similar for all the controllers, meaning that different behaviors are not given by a different magnitude of the inputs.

The extension of these considerations to dynamic control of UVMSs is straightforward using the virtual decomposition [42, 43] approach as detailed in Sect. 7.9.3.

References

1. M.L. Corradini, G. Orlando, A discrete adaptive variable-structure controller for MIMO systems, and its application to an underwater ROV. IEEE Trans. Control Syst. Technol. **5**(3), 349–359 (1997)
2. R. Cristi, F. Papoulias, A. Healey, Adaptive sliding mode control of autonomous underwater vehicles in the dive plane. IEEE J. Oceanic Eng. **15**(3), 152–160 (1990)
3. J.P.V.S. Da Cunha, R.R. Costa, L. Hsu, Design of a high performance variable structure position control of ROVs. IEEE J. Oceanic Eng. **20**(1), 42–55 (1995)
4. J. Yuh, K.V. Gonugunta. Learning control of underwater robotic vehicles, in *ICRA '93: Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1 (1993), pp. 106–111
5. J. Yuh, Learning control for underwater robotic vehicles. IEEE Control Syst. **14**(2), 39–46 (1994)
6. T. Fossen, S.I. Sagatun, Adaptive control of nonlinear systems: a case study of underwater robotic systems. J. Robot. Syst. **8**(3), 393–412 (1991)
7. O.E. Fjellstad, T.I. Fossen, Singularity-free tracking of unmanned underwater vehicles in 6 DOF, in *Proceedings of the 33rd IEEE Conference on Decision and Control*, vol. 2 (1994), pp. 1128–1133
8. J. Yuh, Modeling and control of underwater robotic vehicles. IEEE Trans. Syst. Man Cybern. **20**(6), 1475–1483 (1990)
9. C.S.G. Lee, J.S. Wang, J. Yuh, Self-adaptive neuro-fuzzy systems for autonomous underwater vehicle control. Adv. Robot. **15**(5), 589–608 (2001)

10. T.W. Kim, J. Yuh, A novel neuro-fuzzy controller for autonomous underwater vehicles, in *ICRA '01: Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3 (2001), pp. 2350–2355
11. L. Whitcomb, Underwater robotics: Out of the research laboratory and into the field, in *ICRA '00: Proceedings of the IEEE International Conference on Robotics and Automation* (2000), pp. 709–716
12. G. Antonelli, S. Chiaverini, N. Sarkar, M. West, Adaptive control of an autonomous underwater vehicle experimental results on ODIN, in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, (Monterey, Nov 1999), pp. 64–69
13. G. Antonelli, S. Chiaverini, N. Sarkar, M. West, Adaptive control of an autonomous underwater vehicle: experimental results on ODIN. IEEE Trans. Control Syst. Technol. **9**(5), 756–765 (2001)
14. O.E. Fjellstad, T.I. Fossen, Position and attitude tracking of AUV's: A quaternion feedback approach. IEEE J. Oceanic Eng. **19**(4), 512–518 (1994)
15. S.K. Choi, J. Yuh, Experimental study on a learning control system with bound estimation for underwater robots. Auton. Robots **3**(2), 187–194 (1996)
16. J. Nie, J. Yuh., E. Kardash, T. Fossen. On-board sensor-based adaptive control of small UUVS in very shallow water. Int. J. Adapt. Control Signal Process. **14**(4), 441–452 (2000)
17. S. Zhao, J. Yuh, Experimental study on advanced underwater robot control. IEEE Trans. Robot. **21**(4), 695–703 (2005)
18. D.A. Smallwood, L.L. Whitcomb. Toward model based dynamic positioning of underwater robotic vehicles, in *MTS/IEEE Oceans 2001 Conference and Exhibition*, vol. 2 (2001), pp. 1106–1114
19. D.A. Smallwood, L.L. Whitcomb, The effect of model accuracy and thruster saturation on tracking performance of model based controllers for underwater robotic vehicles: Experimental results, in *ICRA '02: Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2 (Washington, 2002), pp. 1081–1087
20. D.A. Smallwood, L.L. Whitcomb, Model-based dynamic positioning of underwater robotic vehicles: theory and experiment. IEEE J. Oceanic Eng. **29**(1), 169–186 (2004)
21. S.C. Martin, L.L. Whitcomb, Preliminary experiments in fully actuated model based control with six degree-of-freedom coupled dynamical plant models for underwater vehicles, in *ICRA '13: Proceedings of the IEEE International Conference on Robotics and Automation* (2013), pp. 4606–4613
22. T. Fossen, J.G. Balchen et al., The NEROV autonomous underwater vehicle, in *Proceedings of the Conference OceansG '91*, (Citeseer, Honolulu, 1991)
23. A. Healey, D. Lienard, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J. Oceanic Eng. **18**(3), 327–339 (1993)
24. J.S. Riedel, Shallow water stationkeeping of an autonomous underwater vehicle: the experimental results of a disturbance compensation controller, in *MTS/IEEE Oceans 2000 Conference and Exhibition*, vol. 2 (2000), pp. 1017–1028
25. J.S. Riedel, A.J. Healey. Shallow water station keeping of AUVs using multi-sensor fusion for wave disturbance prediction and compensation, in *Proceedings of the Oceans '98 Conference*, vol. 2 (1998), pp. 1064–1068
26. D. Marco, A. Healey, Command, control, and navigation experimental results with the NPS ARIES AUV. IEEE J. Oceanic Eng. **26**(4), 466–476 (2001)
27. A.S. Gadre, D.J. Stilwell, A complete solution to underwater navigation in the presence of unknown currents based on range measurements from a single location, in *IROS '05: IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005), pp. 1420–1425
28. G. Marani, S.K. Choi, J. Yuh, Real-time center of buoyancy identification for optimal hovering in autonomous underwater intervention. Intell. Serv. Robot. **3**(3), 175–182 (2010)
29. G. Antonelli, On the use of adaptive/integral actions for 6-degrees-of-freedom control of autonomous underwater vehicles. IEEE J. Oceanic Eng. **32**(2), 300–312 (April, 2007)
30. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, A novel adaptive control law for autonomous underwater vehicles, *Proceedings of the IEEE International Conference on Robotics and Automation* (Seoul, May 2001), pp. 447–451

31. Y.C. Sun, C.C. Cheah, Adaptive setpoint control for autonomous underwater vehicles, in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 2 (2003), pp. 1262–1267
32. J. Yuh, J. Nie, C.S.G. Lee, Experimental study on adaptive control of underwater robots, in *Proceedings of the IEEE International Conference on Robotics and Automation* (1999), pp. 393–398
33. J.J.E. Slotine, W. Li, On the adaptive control of robot manipulators. Int. J. Robot. Res. **6**(3), 49–59 (1987)
34. G. Conte, A. Serrani, Robust lyapunov-based design for autonomous underwater vehicles, in *Proceedings of the 5th IFAC Symposium on Robot Control* (1997), pp. 321–326
35. B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer, London, 2009)
36. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, On the use of integral control actions for autonomous underwater vehicles, in *European Control Conference* (Porto, Sept 2001)
37. G. Antonelli, A new adaptive control law for the Phantom ROV, *7th IFAC Symposium on Robot Control* (Wroclaw, Sept 2003), pp. 569–574
38. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, A novel adaptive control law for underwater vehicles. IEEE Trans. Control Syst. Technol. **11**(2), 221–232 (2003)
39. S. Arimoto, *Control Theory of Nonlinear Mechanical Systems*, (Oxford University Press Inc., New York, 1996)
40. T. Fossen, *Guidance and Control of Ocean Vehicles*, (Wiley, Chichester New York, 1994)
41. L. Hsu, R.R. Costa, F. Lizarrade, J.P.V. Soares da Cunha, Dynamic positioning of remotely operated underwater vehicles. IEEE Robot. Autom. Mag. **7**(3), 21–31 (2000)
42. W.H. Zhu, *Virtual Decomposition Control: Toward Hyper Degrees of Freedom Robots*, vol. 60 (Springer, London, 2010)
43. W.H. Zhu, T. Lamarche, E. Dupuis, D. Jameux, P. Barnard, G. Liu, Precision control of modular robot manipulators: the VDC approach with embedded FPGA. *IEEE Trans. Robot.* **29**(5), 1162–1179 (2013). doi:10.1109/TRO.2013.2265631

Chapter 4
Fault Detection/Tolerance Strategies for AUVs and ROVs

4.1 Introduction

Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs) received increasing attention in the last years due to their significant impact in several underwater operations. Examples are the monitoring and maintenance of off-shore structures or pipelines, or the exploration of the sea bottom; see, e.g., Reference [1] for a complete overview of existing AUVs with description of their possible applications and the main subsystems. The benefit in the use of unmanned vehicles is in terms of safety, due to the possibility to avoid the risk of manned missions, and economic. Generally, AUVs are required to operate over long periods of time in unstructured environments in which an undetected failure usually implies loss of the vehicle. It is clear that, even in case of failure detection, in order to terminate the mission, or simply to recover the vehicle, a fault tolerant strategy, in a wide sense, must be implemented. In fact, simple system failure can cause mission abort [2] while the adoption of a fault tolerant strategy allows to safely terminate the task as in the case of the arctic mission of Theseus [3]. In case of the use of ROVs, a skilled human operator is in charge of command the vehicle; a failure detection strategy is then of help in the human decision making process. Based on the information detected, the operator can decide in the vehicle rescue or to terminate the mission by, e.g., turning off a thruster.

Fault detection is the process of monitoring a system in order to recognize the presence of a failure; fault isolation or diagnosis is the capability to determine which specific subsystem is subject to failure. Often in literature there is a certain overlapping in the use of these terms. Fault tolerance is the capability to complete the mission also in case of failure of one or more subsystems, it is referred also as fault control, fault accommodation or control reconfiguration. In the following the terms fault detection/tolerance will be used.

The characteristics of a fault detection scheme are the capability of isolate the detected failure; the sensitivity, in terms of magnitude of the failure that can be detected and the robustness in the sense of the capability of working properly also

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_4,

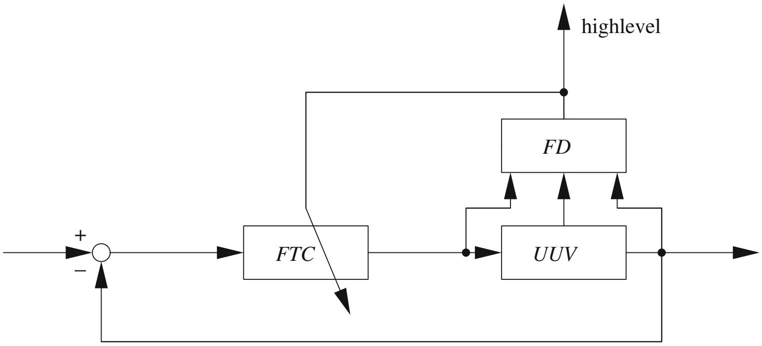


Fig. 4.1 General fault detection/tolerant control scheme for an unmanned underwater vehicle (UUV). The Fault Detection (FD) block is in charge of detecting the failure, send a message to the higher level supervisor and, eventually, modifying the Fault Tolerant Controller (FTC)

in non-nominal conditions. The requirements of a fault tolerant scheme are the reliability, the maintainability and survivability [4]. The common concept is that, to overcome the loss of capability due to a failure, a kind of redundancy is required in the system. A general scheme is presented in Fig. 4.1.

In this chapter, a survey over existing fault detection and fault tolerant schemes for underwater vehicles is presented. For these specific systems, adopting proper strategies, an hardware/software (HW/SW) sensor failure or an HW/SW thruster failure can be successfully handled in different operating conditions as it will be shown in next sections. In some conditions, it is required that the fault detection scheme is also able to diagnostic some external not-nominal working conditions such as a multi-path phenomena affecting the echo-sounder system [5]. It is worth noticing that, for autonomous systems such as AUVs, space systems or aircraft, a fault tolerant strategy is necessary to safely recover the damaged vehicle and, obviously, there is no *panic button* in the sense that the choice of turning off the power or activate some kind of brakes is not available. Strategies as [6] may prevent the occurrence of overall risky missions by presenting a state transition approach, in the form of a Markov chain, which models step sequence from prelaunch to operation to recovery; High risk transitions are then identified.

Most of the fault detection schemes are model-based [5, 7–15] and concern the dynamic relationship between actuators and vehicle behavior or the specific input-output thruster dynamics. A model-free method is presented in [16, 17]. Higher level fault detection schemes are presented in [3, 18–20]. References [21–26] deal with hardware/software aspects of a fault detection implementation for AUVs. Neural Network, Fuzzy and Learning techniques have also been presented [27–32].

Concerning fault tolerant schemes, most of them consider a thruster redundant vehicle that, after a fault occurred in one of the thrusters, still is actuated in 6 Degrees of Freedom (DOFs). Based on this assumption a reallocation of the desired forces on the vehicle over the working thrusters is performed [5, 7–9, 11, 14, 15, 33–37]. Reference [38] too, embeds fault detection, isolation and tolerance by proper control reconfiguration of the redundant thrusters. Of interest is also the study of reconfiguration strategies if the vehicle becomes under-actuated [39].

In recent years, with the increasing number of experimental as well as commercial AUVs, the number of field results is increasing [3, 5, 7–9, 11, 14, 15, 18, 21, 23, 33, 34, 37, 40, 41]. In [42], a discrete fault diagnosis system, Livingstone 2, is designed for hardware and software monitoring of the AUV named Autosub 6000. Being designed for underice missions, this scheme also take into account environmental variables such as the ice concentration and thickness. Effectiveness of the approach has been validated on log data of a series of missions.

In Sect. 4.2 a small list of failures occurred during wet operations is reported; Sects. 4.3 and 4.4 report the description of fault detection and tolerant strategies for underwater vehicles. Since the implementation of such strategies in a real environment is not trivial Sect. 4.5 describes in more detail some successfully experiments. Finally, the conclusions are drawn in Sect. 4.6.

4.2 Experienced Failures

In this section, a small list of possible ROVs/AUVs' failures is reported.

Sensor failure The underwater vehicles are currently equipped with several sensors in order to provide information about their localization and velocity. The problem is not easy, it does not exist a single, reliable sensor that gives the required position/velocity measurement or information about the environment, e.g., about the presence of obstacles. For this reason the use of sensor fusion by, e.g., a Kalman filtering approach, is a common technique to provide to the controller the required variables. This structural redundancy can be used to provide fault detection capabilities to the system. In detail, a failure can occur in one of the following sensors:

- IMU (Inertial Measurement Unit): it provides information about the vehicle's linear acceleration, roll-pitch angles and angular velocity;
- Depth Sensor: by measuring the water pressure gives the vehicle's depth;
- Altitude and frontal sonars: they are used to detect the presence of obstacles and the distance from the sea bottom;
- Ground Speed Sonar: it measures the linear velocity of the vehicle with respect to the ground;
- Currentmeter: it measures the relative velocity between vehicle and water;
- GNSS (Global Navigation Satellite System): it is used to reset the drift error of the IMU and localize exactly the vehicle; it works only at the surface;
- Compass: it gives the vehicle yaw;
- Baseline Acoustic: with the help of one or more transmitters it allows exact localization of the vehicle in a specific range of underwater environment;
- Vision system: it can be used to track structures such as pipelines.

For each of the above sensors the failure can consist in an output zeroing if, e.g., there is an electrical trouble or in a loss of meaning. It can be considered as sensor failure also an external disturbance such as a multi-path reading of the sonar that can be interpreted as a sensor fault and correspondingly detected.

Thruster blocking It occurs when a solid body is between the propeller blades or for the rope entanglements [42]. It can be checked by monitoring the current required by the thruster. It has been observed, e.g., during the Antarctic mission of Romeo [5]: in that occurrence it was caused by a block of ice.

Flooded thruster A thruster flooded with water has been observed during a Romeo's mission [5]. The consequence has been an electrical dispersion causing an increasing blade rotation velocity and thus a thruster force higher then the desired one.

Fin stuck or lost This failure can causes a loss of steering capability as discussed by means of simple numerical simulations in [13]. Other possibilities concern also intermitteng functioning or a non-null offset [42].

Rotor failure A possible consequence of different failures of the thrusters is the zeroing of the blade rotation. The thruster in question, thus, simply stops working. This has been intentionally experienced during experiments with ODIN [14, 15, 33, 34], RAUVER [12] and Roby 2 [11] and during another Romeo's mission [5].

Hardware-software failure A crash in the hardware or software implemented on the vehicle can be experienced. In this case, redundancy techniques can be implemented to handle such situations [21, 42].

Obviously, any other subsystem functional to the achievement of the mission may fail such as, e.g., the power system, the communication module and even the scientific payload carried for the mission [42].

4.3 Fault Detection Schemes

In [9, 11] a model-based fault detection scheme is presented to isolate actuators' failures in the horizontal motion. Each thruster is modeled as in [43]. The algorithm is based on a bank of Extended Kalman Filters (EKFs) the outputs of which are checked in order to detect behaviors not coherent with the dynamic model (Fig. 4.2). In case of two horizontal thrusters and horizontal motion 3 EKFs are designed to simulate the 3 behaviors: nominal behavior, left thruster fault, right thruster fault. The cross-checking of the output allows efficient detection as it has been extensively validated experimentally (details are given in Sect. 4.5). A sketch of this scheme is given in Fig. 4.3 where $\boldsymbol{u}$ is the vector of thruster inputs and the vehicle yaw ψ is measured by means of a compass. In [7], the same approach is investigated with the use of a sliding-mode observer instead of the EKF. The effectiveness of this approach is also discussed by means of experiments.

The work in [5, 8] focuses on the thruster failure detection by monitoring the motor current and the propeller's revolution rate. The non-linear nominal characteristic has been experimentally identified, thus, if the measured couple current-propeller's rate

Fig. 4.2 Romeo (courtesy of M. Caccia, National Research Council-ISSIA, Italy)

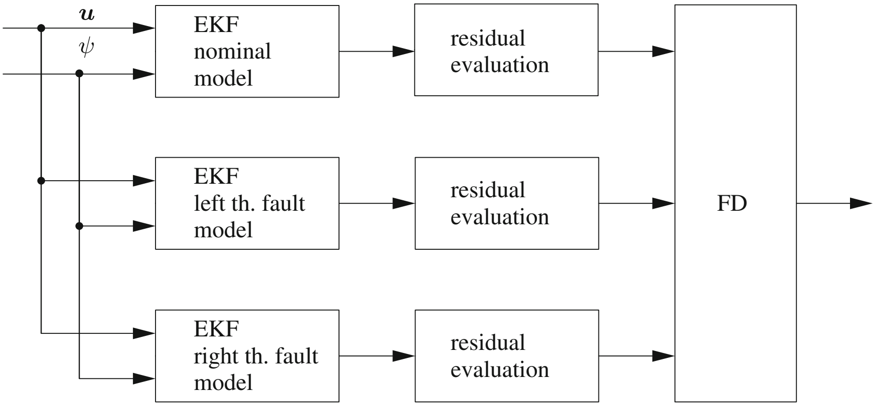


Fig. 4.3 Fault detection strategy for one of the horizontal thruster failure proposed by A. Alessandri, M. Caccia and G. Verrugio

is out of a specific bound, then a fault is experienced. Based on a mapping of the i–o axis the possible cause is also specified with a message to the remote human operator. The two failures corresponding to a thruster flooding or to a rotor failure, in fact, fall in different axis regions and can be isolated. Interesting experiments are given in Sect. 4.5.

The fault in a thruster is also monitored in [14, 15] by the use of a hall-effect sensor mounted on all the thrusters. The input is the desired voltage as computed by the controller and the TCM, the output is the voltage as measured by the hall-effect sensor; the mismatching between the measured and the predicted voltage is considered as a fault. The paper also considers fault tolerance for sensor and actuator faults and experiments, as shown in following sections.

The vehicle Theseus [3, 18] is equipped with a Fault Manager subsystem. This provides some kind of high-level failure detection in the sense that the mission is divided in a number of phases (each phase is a series of manoeuvres between way points); in case of failure of a phase there is a corresponding behavior to be activated. See Sect. 4.5 for detail about a practical intervention of the Fault Manager. A hierarchical control system developed for future implementation on Theseus is described in [20], this is based on the layered control concepts [44].

References [21, 23] present an architecture for AUVs that integrates fault detection capabilities of the subsystems. The hardware and software architecture, named AUVC (Autonomous Underwater Vehicle Controller) implements a fault detection strategy based on five rule-based systems that monitor all the subsystems. The five systems concern the Navigation, the Power/Propulsion, the Direction Control and the Communication; they are coordinated by a Global Diagnoser that avoid contradictory actions. A specific attention has been given at the hardware reliability: as a matter of fact the AUVC is distributed on a redundant network of 18 loosely coupled processors. AUVC has been also used to test the approach proposed in [24], a redundancy management technique based on CLIPS expert system shell to identify faults affecting depth and heading control. In [26] an architecture developed for the vehicle ARICS with fault detection/tolerant capabilities is presented. A software developed for ROVs to help the remote operator that integrates some elementary fault detection algorithms is presented in [25]. The paper in [22] describes the first results on the development of a long endurance AUV that is currently ongoing at the MBARI (Monterey Bay Aquarium Research Institute, California, USA). The fault detection approach is mainly ported from the MIT (Massachusetts Institute of technology, Massachusetts, USA) vehicle Odyssey (I and II) [45] and it is based on the Layered Control Architecture. The software architecture is based on C++, QNX-based modules, which offers multi-tasking capabilities suited for fault-tolerant operations. A single thread suppression and restart can be implemented to recover from software failure. Short-duration operations in open sea and long-duration operations in the lab proved the effectiveness of this approach.

In [13] a model-based observer is used to generate residual between the sensor measured behavior and the predicted one. The model also takes into account the presence of waves in case of operations near the surface. When the residual is larger than a given threshold a Fuzzy Inference System is in charge of isolate the source of this mismatching (see sketch in Fig. 4.4). A planar simulation is provided in case of low speed under wave action and a stuck fin.

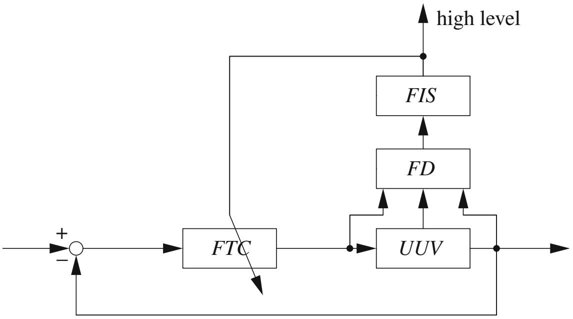


Fig. 4.4 Fuzzy fault detection/tolerant control scheme proposed by A.J. Healey: the FIS (Fuzzy Inference System) block is in charge of isolating faults observed by the FD (Fault Detection) block between fin stroke, servo error, residual and wave activity detectors

A model-free fault detection method is proposed in [16, 17]: this is based on the Hotelling T^2 statistic and it is a data-driven approach. The validation is based on a 6-DOFs simulation affected by stern plane jams and rudder jams.

A model-based, integrated heterogeneous knowledge approach is proposed in [12]. A multi-dimensional correlation analysis allows to increase the confidence in the detected fault and to detect also indirectly sensed subsystems. Some preliminary results with the vehicle RAUVER are also given.

In [10] a model-based fault detection scheme for thrusters and sensors is proposed. It has been designed based on the identified model of the 6-thruster ROV Linotip and it is composed of a bank of single-output Luenberger observers. Its effectiveness is verified by simulations. A robust approach in [46].

A neuro-symbolic hybrid system is used in [27] to perform fault diagnosis on AUVs with learning capability. The method is simulated on the planar motion of the VORTEX mathematical model. Another learning technique is proposed in [28] and verified by means of 6-DOFs simulations. In [31], a neural network mathematical model is used to set-up a self-diagnosis scheme of the AUV. A software for health monitoring of AUVs' missions with learning capabilities is also described in [47].

The work in [48] studies a systematic, quantitative approach in order to maximize the mission and return success probabilities. The failed sensor is de-activated and the information obtained by a backup sensor able to recover the vehicle. No dynamic simulations are provided.

In [49], the wavelet theory is used to detect the fault in the vehicle's navigation angle fault. In [50] developed a software tool to test intelligent controllers for AUVs. This is done by using learning techniques from the artificial intelligence theory.

In [42], a discrete fault diagnosis system, Livingstone 2, is designed for hardware and software monitoring of the AUV named Autosub 6000 belonging to the UK National Oceanography Centre. Being designed for underice missions, this scheme also take into account environmental variables such as the ice concentration and thickness within a Baesyan network architecture.

Reference [41] approaches the fault detection problem within a perspective of a semantic world model framework for hierarchical distributed representation of knowledge. The idea is to raise the abstract level from the raw data by resorting to semantic technologies. The validation is achieved by a injected hardware fault in a REMUS 100 AUV while performing a mission.

4.4 Fault Tolerant Schemes

Most of the fault tolerant controllers developed for thruster-driven underwater vehicles are based on a suitable inversion of the TCM in Eq. (2.51). It is self-evident that, if the matrix is low rectangular it is still possible to turn off the broken thruster and to control the vehicle in all the 6 DOFs. When the vehicle becomes underactuated, or when it is driven by control surfaces, the problem is mathematically more complex. In this case only few solutions to specific set-up have been developed.

The work in [5, 8] reports a fault tolerant approach for ROMEO, a thruster redundant ROV with 8 thrusters. The strategy, experimentally verified, simply consists in deleting the column corresponding to the broken thruster from the TCM. The mapping from the vehicle force/moment to the thrusters' forces thus, does not concern the failed component. A similar approach is used in [14, 15] by exploiting the thruster redundancy of ODIN, an AUV developed at the Autonomous Systems Laboratory (ASL) of the University of Hawaii, HI, USA. The proposed approach is sketched in Fig. 4.5 where the subscript d denotes the desired trajectory, $\boldsymbol{V}_m$ is the motor input voltage and Ω the propeller angular velocity of the thrusters.

In [33–37], a task-space-based, fault tolerant control for vehicles with redundant actuation is proposed. The control law is model based and it handles the thruster redundancy by a pseudo-inverse approach of the TCM that guarantees the minimization of the actuator quadratic norm. The thruster dynamics, with the model described in [51], is also taken into account.

The work [14, 15] presents a fault detection-tolerant scheme for sensor faults. In detail, the depth of the vehicle is measured by using a pressure sensor and a bottom sonar sensor. A third, virtual, sensor is added: this is basically an ARX (AutoRegressive eXogeneous) model of the vehicle depth dynamics. By comparing the measured values with the predicted ones, the residual is calculated and the failed sensor eventually disconnected for the remaining portion of the mission. A sketch of the scheme is shown in Fig. 4.6, in nominal working conditions, the 3 residual R_i are close to the null value. It is worth noticing that this approach requires exact knowledge of the ocean depth.

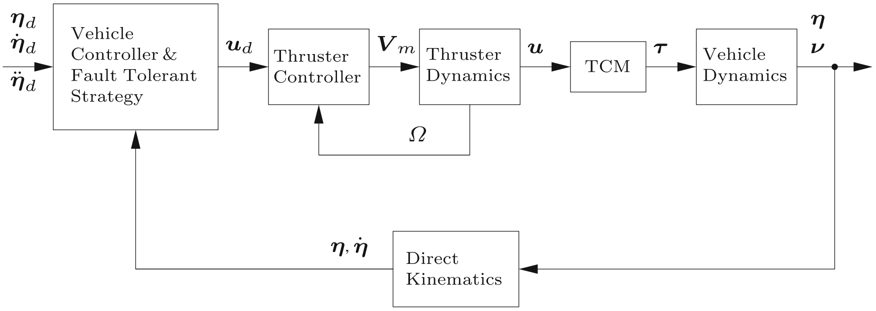


Fig. 4.5 Fault tolerance strategy proposed by K.S. Yang, J. Yuh and S.K. Choi

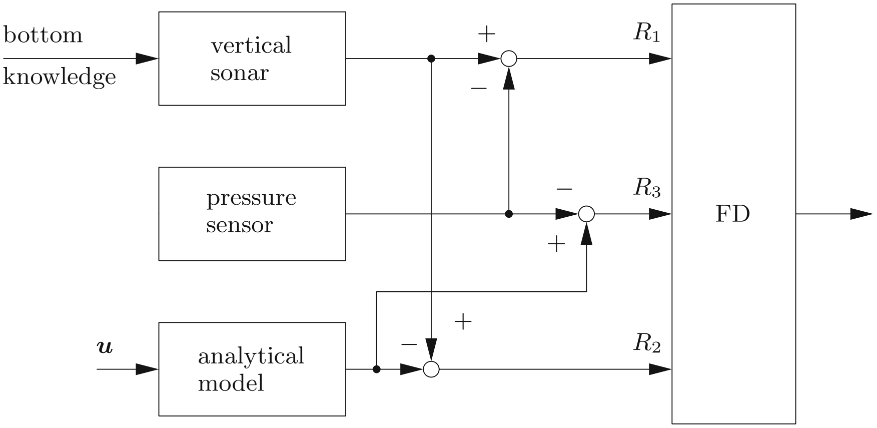


Fig. 4.6 Fault tolerance strategy for sensor fault proposed by K.C. Yang, Y. Yuh and S.K. Choi

In [39], the case of an underactuated AUV controlled by control surfaces is considered. The vehicle tries to move in the unactuated DOFs by using elementary motions in the actuated DOFs. The method has been tested on the vehicle ARCS and showed that, in this form, it is not applicable. This study provides information on structural changes to be adopted in the vehicle in order to develop a vehicle suitable for implementing this method.

One of the first works of reconfiguration control for AUVs is given in [52], where, however, only a superficial description of a possible fault tolerant strategy is provided. Recently, [53] proposed a reconfiguration strategy to accommodate actuator faults: this is based on a mixed H_2/H_∞ problem. Simulation results are provided.

4.5 Experiments

Roby 2 is a ROV developed at the National Research Council-ISSIA, Italy. It has been object of several wet tests aiming at validating fault detection approaches. The horizontal motion is obtained by the use of two fore thrusters that control the surge velocity as well as the vehicle heading; the depth is regulated by means of two vertical thrusters. In [7, 9, 11] experiments of different fault detection schemes have been carried out by causing, on purpose, an actuator failure: one of the thrusters has been simply turned off. The experiments in [9, 11] have been carried out in a pool, in [7] the experiments also concern a comparison between EKFs and sliding-mode observers. The latter is a result of a bilateral project with the Naval Postgraduate School, Monterey, CA.

The Italian National Research Council (CNR-ISSIA) also developed the ROV ROMEO and tested, in an antarctic mission, both fault detection and tolerant schemes [5, 8]. In particular the case of flooded and blocked thrusters occurred.

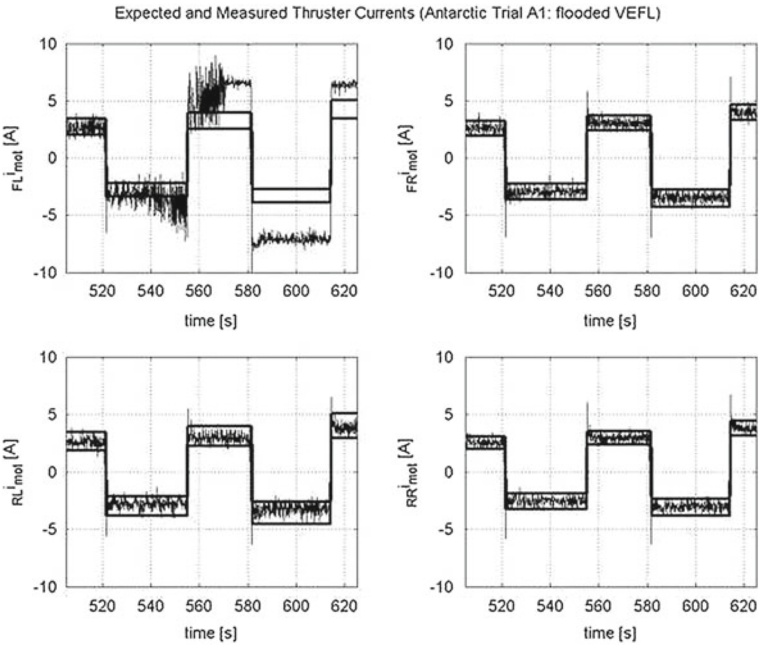


Fig. 4.7 Expected and measured motor currents for the vehicle Romeo in case of flooded thruster (courtesy of M. Caccia, National Research Council-ISSIA)

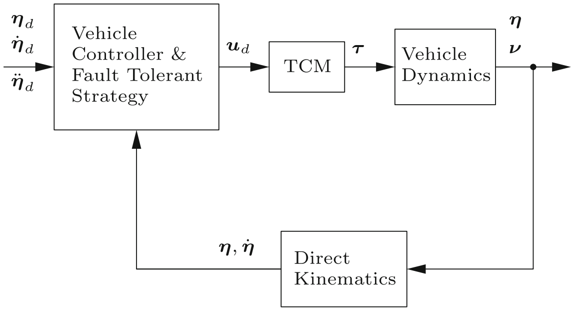


Fig. 4.8 Sketch of the fault tolerance strategy implemented by N. Sarkar, T.K. Podder and G. Antonelli

In both cases the fault has been detected and the information could be reported to the human operator in order to activate the reconfiguration procedure. Figure 4.7 shows the expected and measured motor currents in case of flooded thruster: it can be observed a persistent mismatching between the output of the model and the measured values.

The vehicle Theseus manufactured by ISE Research Ltd with the Canadian Department of National Defence successfully handled a failure during an Arctic mission of cable laying [3, 18]. In details, the vehicle did not terminate a homing step, probably due to poor acoustic conditions and the Fault Manager activated a safe behavior: stop under the ice and wait for further instructions. This allowed to re-establish acoustic telemetry and surface tracking and safely recover the vehicle. Notice that his fault wasn't intentionally caused [3].

The vehicle ODIN, an AUV developed at the Autonomous Systems Laboratory (ASL) of the University of Hawaii, HI, USA, has been used for several experiments. In [14, 15] the fault detection and tolerant schemes are experimentally validated. The thruster fault has been tested by zeroing the output voltage by means of software, the fault detection scheme identified the trouble and correctly reconfigured the force allocation by properly modifying of the TCM. The fault tolerant scheme with respect to depth sensor fault has also been tested by zeroing the sensor reading and verifying that the algorithm, after a programmed time of 1 s, correctly switched on the other sensor. While the theory has been developed for a 6-DOFs vehicle, the experiments results only present the vehicle depth.

The same vehicle has been used to validate the fault tolerant approach developed in [33, 34, 37] in 6-DOFs experiments. Different experiments have been carried out by zeroing the voltage on one or two thrusters simultaneously that, however, did not cause the vehicle becoming under-actuated. The implemented control law is based on an identified reduced ODIN model and does not make use of thruster model neither

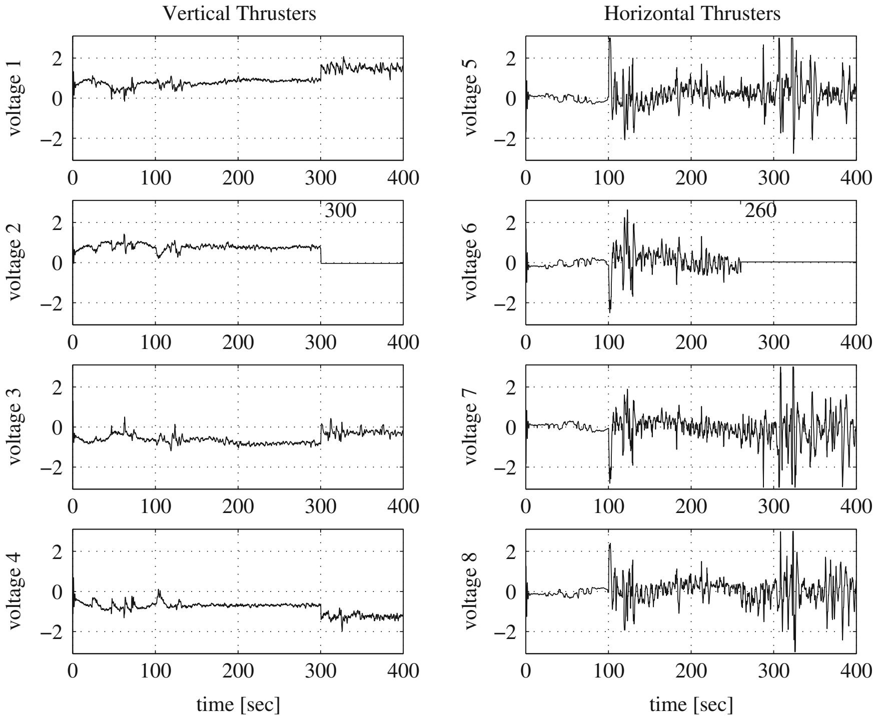


Fig. 4.9 Voltage profile—vertical thrusters (*on the left*) and horizontal thrusters (*on the right*) for the N. Sarkar and T.K. Podder algorithm

it needs the vehicle acceleration as required by the theory; the block diagram, thus, is simply given by Fig. 4.8. Details on the control law are given in the referenced papers, the basic formulation of the controller is given by:

$$\boldsymbol{u}_v = \boldsymbol{E}^{\dagger}\left[\left(\ddot{\boldsymbol{\eta}}_d - \boldsymbol{\beta}\right) + \boldsymbol{K}_v\dot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{K}_p\tilde{\boldsymbol{\eta}}\right] \tag{4.1}$$

where $\boldsymbol{K}_v$ and $\boldsymbol{K}_v$ are control gains, the vector $\tilde{\boldsymbol{\eta}}$ is the position/orientation error, $\boldsymbol{\beta}$ represents the compensation of the nonlinear terms of the equation of motions. The matrix $\boldsymbol{E}$ takes into account the TCM matrix, the inertia matrix and the Jacobian matrix that converts body-fixed to inertial-fixed velocities. Generalization about control of a desired task is given in [34]. The experiments validated the proposed approach; in Fig. 4.9 the voltages are shown: it can be recognized that thrusters 2 and 6 (one horizontal and one vertical) are turned off at $t = 260$ s and $t = 300$ s, respectively, this causes an augmentation of the chattering of the remaining thrusters that, however, still can perform the desired task (Fig. 4.10).

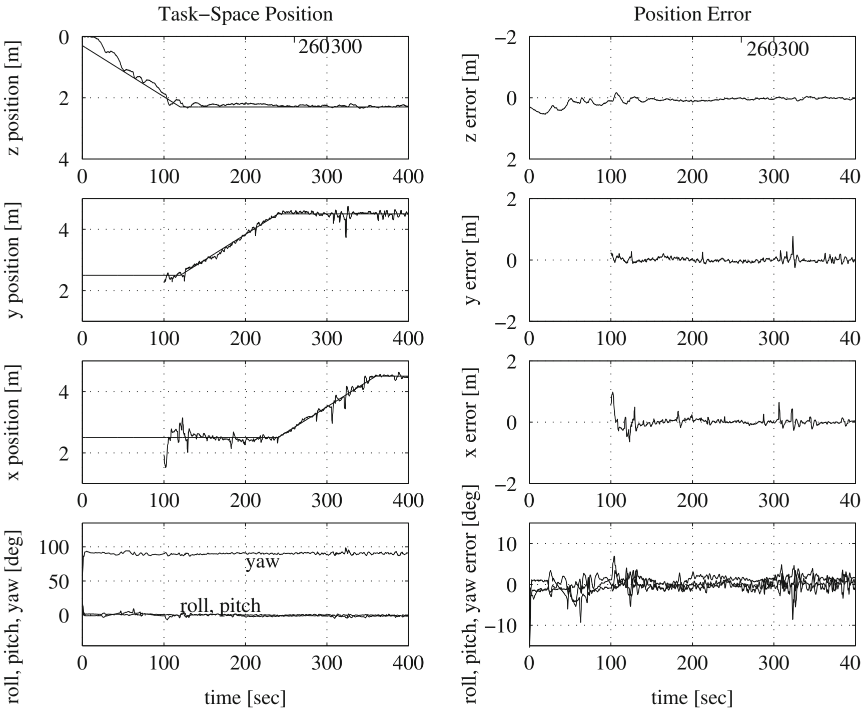


Fig. 4.10 Task-space trajectories (*left column*) and their errors (*right column*) for the N. Sarkar and T.K. Podder algorithm

The AUVC described in [21, 23] has been tested on a six-processor version on the Large Diameter Unmanned Underwater Vehicle of the Naval Undersea Warfare Center.

4.6 Conclusions

An overview over existing fault detection and fault tolerant schemes for underwater vehicles has been presented. The case of failures for autonomous missions in un-structured environment is, obviously, a dramatic occurrence to handle. In this sense, the underwater community would benefit from research studies with a strong practical orientation rather than theoretical-only approaches. Failures in a redundant sensor seem to be a solved problem; however, a particular attention needs to be paid to the tuning of the detection gains, real-data experiments for off-line tuning seems to be a reliable way to select those gains. As far as thrusters are of concern, experiments have shown that current AUVs can be controlled at low velocity with 6 thrusters, also if the original symmetric allocation is lost, without a strong performance deterioration. Some possible research areas concern the case of a thruster-driven AUV that becomes under-actuated and the case of vehicles controlled by means of control surfaces that failed; in both cases some practical approaches might be useful for the underwater community.

References

1. J. Yuh, M. West, Underwater robotics. Adv. Robot. **15**(5), 609–639 (2001)
2. B. Hutchison, Velocity aided inertial navigation, in *Proceedings of Sensor Navigation Issues for UUVs*, CS Draper Lab. (1991)
3. J. Ferguson, A. Pope, B. Butler, R. Verrall, Theseus AUV-two record breaking missions. Sea Technol. **40**(2), 65–70 (1999)
4. H.E. Rauch, Intelligent fault diagnosis and control reconfiguration. IEEE Control Syst. **14**(3), 6–12 (1994)
5. M. Caccia, R. Bono, Ga. Bruzzonea, Gi Bruzzone, E. Spirandelli, G. Veruggio, Experiences on actuator fault detection, diagnosis and accomodation for ROVs, in *International Symposiyum of Unmanned Untethered Sub-mersible Technol*. (Durham, New Hampshire, 2001)
6. M.P. Brito, G. Griffiths, A Markov chain state transition approach to establishing critical phases for AUV reliability. IEEE J. Oceanic Eng. **36**(1), 139–149 (2011)
7. A. Alessandri, A. Gibbons, A.J. Healey, G. Veruggio, Robust model-based fault diagnosis for unmanned underwater vehicles using sliding mode-observers, in *Proceedings International Symposium Unmanned Untethered Submersible Technology* (Durham, New Hampshire, 1999), pp. 352–359
8. R. Bono, Ga. Bruzzone, M. Caccia, ROV actuator fault diagnosis through servo-amplifiers' monitoring: an operational experience, in *Proceedings of OCEANS'99 MTS/IEEE. Riding the Crest into the 21st Century*, vol. 3, pp. 1318–1324. (1999).
9. A. Alessandri, M. Caccia, G. Veruggio, A model-based approach to fault diagnosis in unmanned underwater vehicles, in *OCEANS'98 Conference Proceedings/IEEE*, vol. 2, pp. 825–829. (1998)

10. Y.K. Alekseev, V.V. Kostenko, A.Y. Shumsky, Use of identification and fault diagnostic methods for underwater robotics, in *OCEANS'94'. Oceans Engineering for Today's Technology and Tomorrow's Preservation/IEEE*, pp. 489–494. (1994)
11. A. Alessandri, M. Caccia, G. Veruggio, Fault detection of actuator faults in unmanned underwater vehicles. Control Eng. Pract. **7**(3), 357–368 (1999)
12. K. Hamilton, D. Lane, N. Taylor, K. Brown, Fault diagnosis on autonomous robotic vehicles with RECOVERY: an integrated heterogeneous-knowledge approach, in *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA*, San Francisco, California, 2001, pp. 3232–3237
13. A.J. Healey, Analytical redundancy and fuzzy inference in AUV fault detection and compensation, in *Proceedings Oceanology 1998*, Brighton, 1998, pp. 45–50
14. K.C. Yang, J. Yuh, S.K. Choi, Experimental study of fault-tolerant system design for underwater robots, in *IEEE International Conference on Robotics and Automation, 1998. Proceedings 1998*, vol. 2, pp. 1051–1056. (1998)
15. K.C. Yang, J. Yuh, S.K. Choi, Fault-tolerant system design of an autonomous underwater vehicle ODIN: an experimental study. Int. J. Syst. Sci. **30**(9), 1011–1019 (1999)
16. G. Beale, J. Kim, A robust approach to reconfigurable control, *5th IFAC Conference on Manoeuvring and Control of Marine Craft* (Aalborg, Denmark, 2000), pp. 197–202
17. J. Kim, G. Beale, Fault detection and classification in underwater vehicle using the t^2 statistic, in *9th Mediterranean Conference on Control and Automation*, (Dubrovnik, Croatia, 2001)
18. J. Ferguson, The Theseus autonomous underwater vehicle. Two successful missions, in *Underwater Technology, 1998. Proceedings of the 1998 International Symposium/IEEE*, Tokyo, Japan, 1998, pp. 109–114
19. A.J. Healey, D.B. Marco, Experimental verification of mission planning by autonomous mission execution and data visualization using the NPS AUV II, in *Autonomous Underwater Vehicle Technology, 1992. AUV'92., Proceedings of the 1992 Symposium/IEEE*, Washington D.C., 1992, pp. 65–72
20. X. Zheng, Layered control of a practical AUV. in *Autonomous Underwater Vehicle Technology, 1992. AUV'92., Proceedings of the 1992 Symposium/IEEE*, Washington D.C., 1992, pp. 142–147
21. D. Barnett, S. McClaran, E. Nelson, M. McDermott, G. Williams, Architecture of the Texas A&M autonomous underwater vehicle controller, in *Autonomous Underwater Vehicle Technology, 1996. AUV'96., Proceedings of the 1996 Symposium/IEEE*, Monterey, California, 1996, pp. 231–237
22. W.J. Kirkwood, D. Gashler, H. Thomas, T.C. O'Reilly, R. McEwen, N. Tervalon, F. Shane, D. Au, M. Sibenac, T. Konvalina, et al., Development of a long endurance autonomous underwater vehicle for ocean science exploration, in *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, Honolulu, Hawaii, 2001, pp. 1504–1512
23. E. Nelson, S. McClaran, D. Barnett, Development and validation of the Texas A&M university autonomous underwater vehicle controller, in *Autonomous Underwater Vehicle Technology, 1996. AUV'96., Proceedings of the 1996 Symposium/IEEE*, Monterey, California, 1996, pp. 203–208
24. A. Orrick, M. McDermott, D. Barnett, E. Nelson, G. Williams, Failure detection in an autonomous underwater vehicle, in *Autonomous Underwater Vehicle Technology, 1994. AUV'94., Proceedings of the 1996 Symposium/IEEE*, Cambridge, Massachusetts, 1994, pp. 377–382
25. R.P. Stokey, Software design techniques for the man machine interface to a complex underwater vehicle, in *OCEANS'94'. Oceans Engineering for Today's Technology and Tomorrow's Preservation'. Proceedings of IEEE*, Brest, France, 1994, pp. 119–124
26. A. Yavnai, Architecture for an autonomous reconfigureable intelligent control system (ARICS), in *Autonomous Underwater Vehicle Technology, 1996. AUV'96., Proceedings of the 1996 Symposium/IEEE*, Monterey, California, 1996, pp. 238–245
27. B. Deuker, M. Perrier, B. Amy, Fault-diagnosis of subsea robots using neuro-symbolic hybrid systems, in *OCEANS'98 Conference Proceedings/IEEE*, Nice, France, 1998, pp. 830–834

28. J. Farrell, T. Berger, B.D. Appleby, Using learning techniques to accommodate unanticipated faults. IEEE Control Syst **13**(3), 40–49 (1993)
29. A.J. Healey, A neural network approach to failure diagnostics for underwater vehicles, in *Autonomous Underwater Vehicle Technology, 1992. AUV'92., Proceedings of the 1992 Symposium on IEEE*, Washington D.C., 1992, pp. 131–134
30. A.J. Healey, F. Bahrke, J. Navarrete, Failure diagnostics for underwater vehicles: A neural network approach, *IFAC Conference on Manoeuvring and Control of Marine Craft* (Aalborg, DenmarK, 1992), pp. 293–306
31. M. Takai, T. Fujii, T. Ura, A model based diagnosis system for autonomous underwater vehicles using artifical neural networks, *Proceedings International Symposium Unmanned Untethered Submersible Technology* (Durham, New Hampshire, 1995), pp. 243–252
32. N. Ranganathan, M.I. Patel, R. Sathyamurthy, An intelligent system for failure detection and control in an autonomous underwater vehicle. IEEE Trans. Syst. Man Cybern. A: Syst. Humans **31**(6), 762–767 (2001)
33. T.K. Podder, G. Antonelli, N. Sarkar, Fault tolerant control of an autonomous underwater vehicle under thruster redundancy: simulations and experiments, *Proceedings 2000 IEEE International Conference on Robotics and Automation* (CA, April, San Francisco, 2000), pp. 1251–1256
34. T.K. Podder, G. Antonelli, N. Sarkar, An experimental investigation into the fault-tolerant control of an autonomous underwater vehicle. J. Adv. Robot. **15**(5), 501–520 (2001)
35. T.K. Podder, N. Sarkar, Fault tolerant decomposition of thruster forces of an autonomous underwater vehicle, in *Proceedings 1998 IEEE International Conference on Robotics and Automation*, Leuven, B, May 1998, pp. 84–89
36. T.K. Podder, N. Sarkar, Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy. Robot. Auton. Syst. **34**(1), 39–52 (2001)
37. N. Sarkar, T.K. Podder, G. Antonelli, Fault accommodating thruster force allocation of an AUV considering thruster redundancy and saturation. IEEE Trans. Robot. Autom. **18**(2), 223–233 (April 2002)
38. M.L. Corradini, A. Monteriù, G. Orlando, An actuator failure tolerant control scheme for an underwater remotely operated vehicle. IEEE Trans. Control Syst. Technol. **19**(5), 1036–1046 (2011)
39. D. Perrault, M. Nahon, Fault-tolerant control of an autonomous underwater vehicle, in *OCEANS'98 Conference Proceedings of IEEE*, Nice, France, 1998, pp. 820–824
40. K. Hamiltonand, D.M. Lane, K.E. Brown, J. Evans, N.K. Taylor, An integrated diagnostic architecture for autonomous underwater vehicles: research articles. J. Field Robot. **24**(6), 497–526 (2007)
41. E. Miguelaez, P. Patron, K.E. Brown, Y.R. Petillot, D.M. Lane, Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles. IEEE Trans. Knowl. Data Eng. **23**(5), 759–773 (2011)
42. R. Dearden, J. Ernits, Automated fault diagnosis for an autonomous underwater vehicle. IEEE J. Oceanic Eng. **58**, 11–21 (2013)
43. T. Fossen, *Guidance and Control of Ocean Vehicles*. (McGraw Hill, New York, 1994)
44. R.A. Brooks, A robust layered control system for a mobile robot. IEEE J. Robot. Autom. **2**(1), 14–23 (1986)
45. J.G. Bellingham, C.A. Goudey, T.R. Consi, J.W. Bales, D.K. Atwood, J.J. Leonard, C. Chryssostomidis, A second generation survey AUV, in *Autonomous Underwater Vehicle Technology, 1994. AUV'94., Proceedings of the 1994 Symposium on IEEE*, Cambridge, Massachusetts, 1994, pp. 148–155
46. R.S. Mangoubi, B.D. Appleby, G.C. Verghese, W.E. Vander Velde, A robust failure detection and isolation algorithm, in *Decision and Control, 1995, Proceedings of the 34th IEEE Conference on IEEE*, New Orleans, Louisiana, 1995, pp. 2377–2382
47. W. Hornfeld, E. Frenzel, Intelligent AUV on-board health monitoring software (INDOS), in *OCEANS'98 Conference Proceedings/IEEE*, vol. 2, Nice, France, 1998, pp. 815–819

48. P.S. Babcock IV, J.J. Zinchuk, Fault-tolerant design optimization: application to an autonomous underwater vehicle navigation system, in *Autonomous Underwater Vehicle Technology, 1990. AUV'90., Proceedings of the (1990) Symposium on IEEE*, Washington DC, 1990, pp. 34–43
49. X. Dermin, G. Lei, Wavelet transform and its application to autonomous underwater vehicle control system fault detection, in *Underwater Technology, 2000. UT 00. Proceedings of the 2000 International Symposium on IEEE*, Tokyo, Japan, 2000, pp. 99–104
50. A.C. Schultz, J.J. Grefenstette, K.A. De Jong, Adaptive testing of controllers for autonomous vehicles, in *Autonomous Underwater Vehicle Technology, 1992. AUV'92., Proceedings of the 1992 Symposium on IEEE*, Washington, DC, 1992, pp. 158–164
51. A. Healey, S.M. Rock, S. Cody, D. Miles, J.P. Brown, Toward an improved understanding of thruster dynamics for underwater vehicles. IEEE J. Oceanic Eng. **20**(4), 354–361 (1995)
52. G. Tacconi, A. Tiano, Reconfigurable control of an autonomous underwater vehicle, in *Unmanned Untethered Submersible Technology, 1989. Proceedings of the 6th International Symposium on IEEE*, 1989, pp. 486–493
53. G. Tong, Z. Jimao, A rapid reconfiguration strategy for UUV control, in *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on IEEE*, Tokyo, Japan, 1998, pp. 478–483

Chapter 5
Experiments of Dynamic Control of a 6-DOF AUV

5.1 Introduction

In this chapter some experimental results on dynamic control of a 6-DOF AUV are given; practical aspects of the implementation are also discussed. The experiments have been conducted in the pool of the University of Hawaii using ODIN, an AUV developed at the Autonomous Systems Laboratory (ASL) [1].

Implemented control law. The control law is briefly rewritten:

$$\boldsymbol{u}_v = \boldsymbol{B}_v^{\dagger}[\boldsymbol{K}_D \boldsymbol{s}'_v + \boldsymbol{\Phi}_v(\mathcal{Q}, \boldsymbol{\nu}, \dot{\boldsymbol{\nu}}_a)\hat{\boldsymbol{\theta}}_v] \tag{5.1}$$

$$\dot{\hat{\boldsymbol{\theta}}}_v = \boldsymbol{K}_\theta^{-1} \boldsymbol{\Phi}_v^{\mathrm{T}}(\mathcal{Q}, \boldsymbol{\nu}, \dot{\boldsymbol{\nu}}_a)\boldsymbol{s}_v \tag{5.2}$$

where $\boldsymbol{B}_v^{\dagger}$ is the pseudoinverse of matrix $\boldsymbol{B}_v$ (see the Appendix), $\boldsymbol{K}_\theta > \boldsymbol{O}$ and $\boldsymbol{\Phi}_v$ is the vehicle regressor. The vectors $\boldsymbol{s}'_v \in \mathbb{R}^6$ and $\boldsymbol{s}_v \in \mathbb{R}^6$ are defined as follows

$$\begin{aligned} \mathbf{s}'_v &= \begin{bmatrix} \tilde{\boldsymbol{\nu}}_1 \\ \tilde{\boldsymbol{\nu}}_2 \end{bmatrix} + \left(\boldsymbol{\Lambda} + \boldsymbol{K}_D^{-1}\boldsymbol{K}_P\right) \begin{bmatrix} \boldsymbol{R}_I^B \tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{\varepsilon}} \end{bmatrix} \\ &= \tilde{\boldsymbol{\nu}} + \left(\boldsymbol{\Lambda} + \boldsymbol{K}_D^{-1}\boldsymbol{K}_P\right) \tilde{\boldsymbol{y}}, \end{aligned} \tag{5.3}$$

$$\boldsymbol{s}_v = \tilde{\boldsymbol{\nu}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{y}}\,, \tag{5.4}$$

with $\tilde{\boldsymbol{\eta}}_1 = \begin{bmatrix} x_d - x & y_d - y & z_d - z \end{bmatrix}^{\mathrm{T}}, \tilde{\boldsymbol{\nu}}_1 = \boldsymbol{\nu}_{1,d} - \boldsymbol{\nu}_1$, where the subscript d denotes desired values for the relevant variables. The matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{6\times 6}$ is defined as $\boldsymbol{\Lambda} = \text{blockdiag}\{\lambda_p \boldsymbol{I}_3, \lambda_o \boldsymbol{I}_3\}$, $\boldsymbol{\Lambda} > \boldsymbol{O}$. The matrix $\boldsymbol{K}_P \in \mathbb{R}^{6\times 6}$ is defined as $\boldsymbol{K}_P = \text{blockdiag}\{k_p \boldsymbol{I}_3, k_o \boldsymbol{I}_3\}$, $\boldsymbol{K}_P > \boldsymbol{O}$. Finally, it is

$$\boldsymbol{\nu}_a = \boldsymbol{\nu}_d + \boldsymbol{\Lambda}\tilde{\boldsymbol{y}}$$

and $\boldsymbol{K}_D > \boldsymbol{O}$.

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_5,

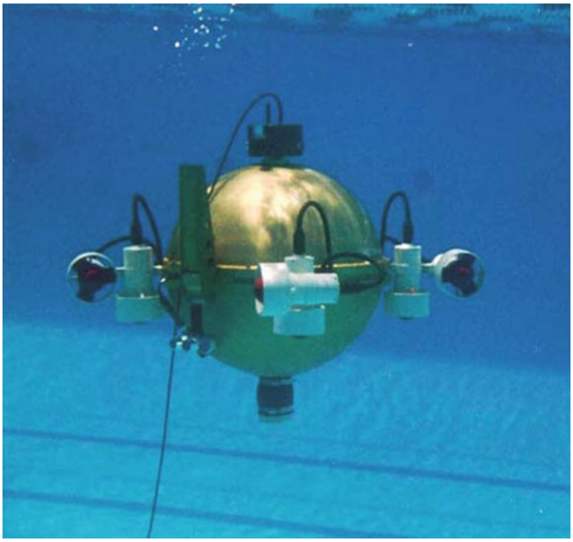


Fig. 5.1 ODIN, the AUV used to experimentally test adaptive and fault tolerant control strategies

5.2 Experimental Set-Up

ODIN is an autonomous underwater vehicle developed at the Autonomous Systems Laboratory of the University of Hawaii. A picture of the vehicle is shown in Fig. 5.1. It has a near-spherical shape with horizontal diameter of 0.63 m and vertical diameter of 0.61 m, made of anodizied Aluminum (AL 6061-T6). Its dry weight is about 125 kg. It is, thus, slightly positive buoyant. The processor is a Motorola 68040/33 MHz working with VxWorks 5.2 operating system. The power supply is furnished by 24 Lead Gel batteries, 20 for the thrusters and 4 for the CPU, which provide about 2 h of autonomous operations. The actuating system is made of 8 marine propellers built at ASL; they are actuated by brushless motors. Each motor weighs about 1 kg and can provide a maximum thrust force of about 27 N. The sensory system is composed of: a pressure sensor for depth measuring, with an accuracy of 3 cm; 8 sonars for position reconstruction and navigation, each with a range $0.1 \div 14.4$ m; an Inertial System for attitude and velocity measures.

5.3 Experiments of Dynamic Control

Despite the closed environment in which the experiments have been conducted, the pool of the University of Hawaii, it is necessary to take into account the presence of a current as an irrotational, constant disturbance [2]. The modeling aspects of including the current in the dynamic model have been discussed in Sect. 2.4.3. Since the measure of the current is not available in ODIN, it has been taken into account as a disturbance $\boldsymbol{\tau}_{v,C}$ acting at the force/moment level on the vehicle-fixed frame. Moreover, since the number of dynamic parameters could be very large ($n_{\theta,v} > 100$, see [2]) it was implemented a reduced version of the regressor matrix in order to adapt only with respect to the restoring force/moments and to the current. In other words, the control law implemented was the reduced controller of the one discussed in Sect. 3.2.

The experiment was conceived in the following way: the vehicle had to follow a desired trajectory with trapezoidal profile. Since the sonars need to be under the surface of the water to work properly the first movement planned was in the z direction (see Fig. 5.2 for the relevant frames). The vehicle planned to move 2 m in the y direction and 2 m in the x direction.

The control law has been designed using quaternions, however the specifications of the desired trajectory and the output results are given in Euler angles because of their immediate comprehension. Notice that the transformation from Euler angles to quaternions is free from representation singularities. The attitude must be kept constant at the value of

$$\boldsymbol{\eta}_{d,2} = \begin{bmatrix} 0 \ 0 \ 90 \end{bmatrix}^{\mathrm{T}} \quad \deg .$$

Since the vehicle is not perfectly balanced, at rest, i.e., with the thrusters off, its position is $\phi \approx 5°$ and $\theta \approx 15°$ with the yaw depending on the current. The desired orientation, thus, is a set-point for the control task.

The following control gains have been used:

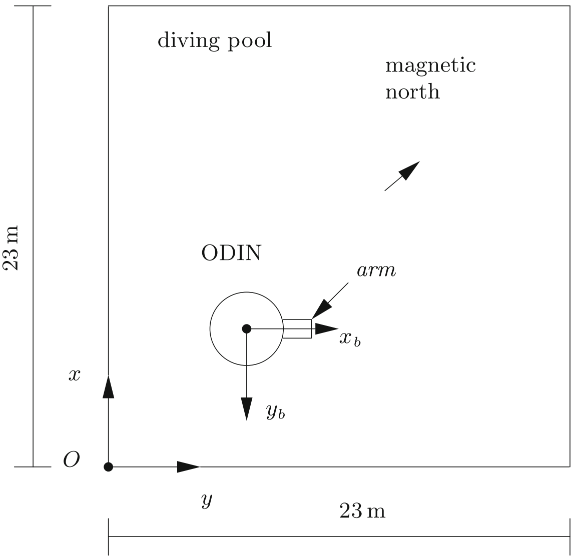


Fig. 5.2 Sketch of the pool used for experiments with relevant frames

$$
\begin{aligned}
\boldsymbol{\Lambda} &= \mathrm{diag}\{0.5, 0.5, 0.15, 0.8, 0.8, 0.8\}\,,\\
\boldsymbol{K}_D &= \mathrm{diag}\{12, 12, 5, 2, 2, 8\}\,,\\
k_p &= 1\,,\\
k_o &= 1\,,\\
\boldsymbol{K}_\theta &= \mathrm{blockdiag}\{0.01, 0.3\boldsymbol{I}_3\}
\end{aligned}
$$

In Fig. 5.3 the time history of the position of the vehicle in the inertial frame and the attitude of the vehicle and the orientation error in terms of the Euler angles is shown. Notice that, due to technical characteristics of the horizontal sonars [3], the x and y position data for the first 100 s are not available. The vehicle is not controlled in those directions and subjected to the pool's current. At $t = 100$ s, it recovers the desired position and starts tracking the trajectory. In the first seconds the vehicle does not track the desired depth. This is based on the assumption that all the dynamic parameters are unknown; at the beginning, thus, the action of the adaptation has to be waited. From this plots it is also possible to appreciate the different noise

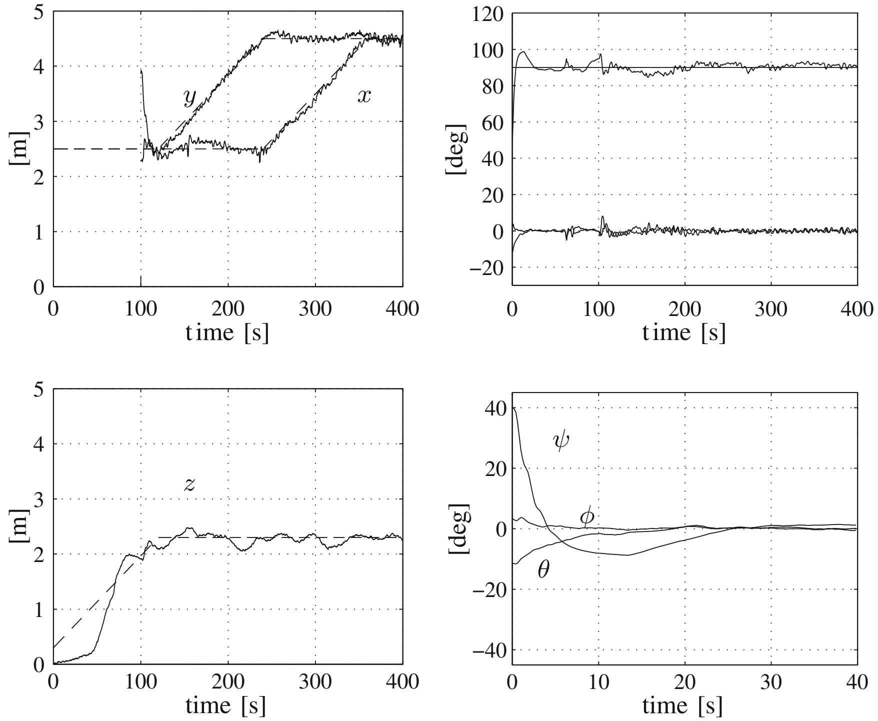


Fig. 5.3 Experiments of adaptive control on ODIN. *Left* time history of the vehicle positions. *Top* x and y positions. *Bottom* z position. *Right Top* time history of the vehicle attitude in terms of Euler angles. *Bottom* vehicle attitude errors in terms of Euler angles (particular)

characteristics of the two position sensors: the sonar for the horizontal plane and the pressure sensor for the depth. It can be noticed that the desired attitude is kept for the overall length of the experiment and that a small coupling can be seen when, at $t = 100$ s the vehicle recovers the desired position in the horizontal plane.

In Fig. 5.4 the control actions are shown. The *peak* in τ_x is due to the big error *seen* by the controller at $t = 100$ s.

In Fig. 5.5 the path in the horizontal plane is shown. It can be noted that the second segment is followed with a smaller error than the first one, due to the adaptation action.

In Fig. 5.6 the 2-norm of the position error (x, y and z components) is shown. It can be noted that the mean error at steady state, in the last 40 s, is about 7 cm. These data need to be be related to the low accuracy and to the big noise that affect the sonar's data.

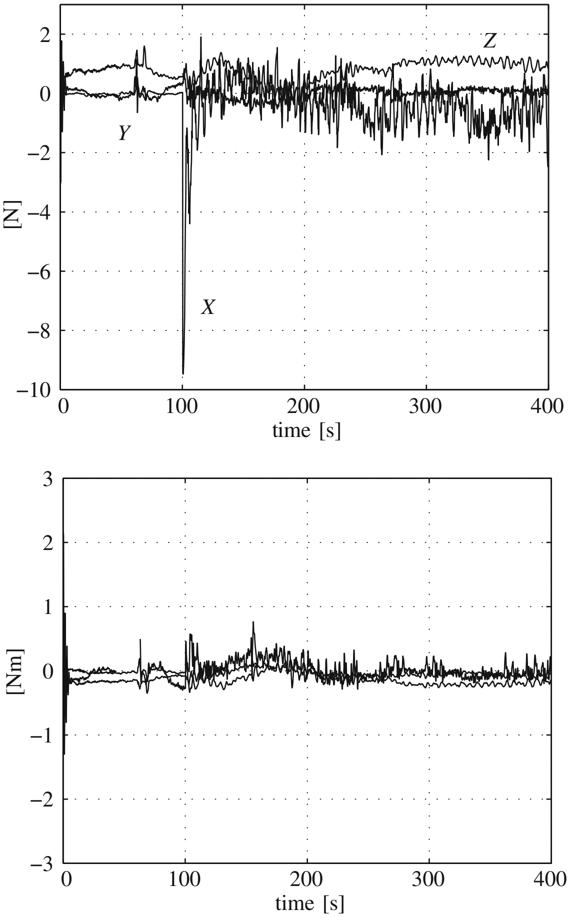


Fig. 5.4 Experiments of adaptive control on ODIN. *Top* control forces. *Bottom* control moments

Fig. 5.5 Experiments of adaptive control on ODIN. Path on the xy plane

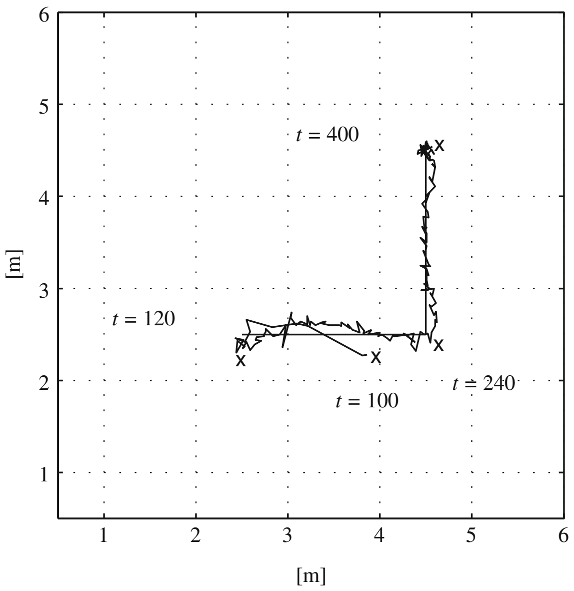
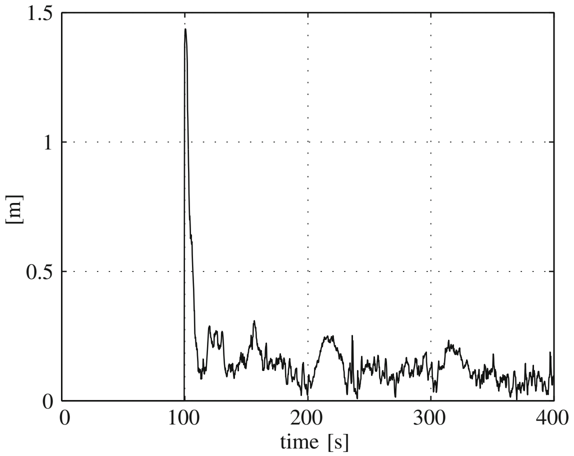


Fig. 5.6 Experiments of adaptive control on ODIN. 2-norm of the 3D position errors in the inertial frame



5.4 Experiments of Fault Tolerance to Thrusters' Fault

The desired trajectory is the same shown in the previous Section. The experiments were run several times simulating different thruster's fault. The fault was simulated via software just imposing zero voltage to the relevant thrusters. It is worth noticing that, while the desired trajectory is always the same, some minor differences arise among the different experiments due to different factors: the presence of strong noise on the sonar that affected the xy movement (see [3] for details), the presence of a pool

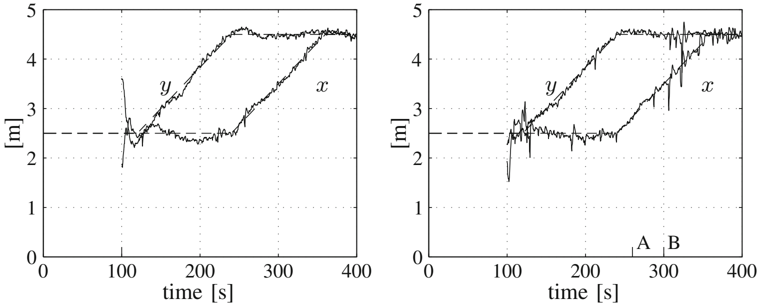


Fig. 5.7 Experiments of fault tolerant control on ODIN. *Left* vehicle movement along xy without thrusters' faults. *Right* vehicle movement along xy with thrusters' faults. *A* first fault at $t = 260$ s at one *horizontal* thruster. *B* second fault at $t = 300$ s at one *vertical* thruster

current. Meanwhile the controller guarantees a good tracking performance tolerant to the occurrence of thruster's fault and it is robust with respect to the described disturbances.

In next plots the movement of the vehicle without fault and with 2 faults are reported. The two faults arise as follow: at $t = 260$ s, one horizontal thruster is off, at $t = 300$ s also the corresponding vertical thruster is off. We chose to test a fault of the same side thrusters because this appears to be the worst situation. Notice that the control law is different from the control law tested in the previous Section. For details, see [4–7].

In Fig. 5.7 the behavior of the vehicle along xy without and with fault is reported. It can be noted that, as for the previous experiments, the vehicle is controlled only after 100 s in order for the sonar to work properly and wait for the transient of the position filter [3]. From the right plot in Fig. 5.7 we can see that the first fault, at $t = 260$ s, it does not affect the tracking error while for the second, at $t = 300$ s, it causes only a small perturbation that is fully recovered by the controller after a transient. Some comments are required for the sonar based error: during the experiment with fault it has been possible to see a small perturbation in the xy plane but, according to the data, this movement has been of 60 cm in 0.2 s, this is by far a wrong data caused by the sonar filter.

In Fig. 5.8 the behavior of the vehicle along z without and with fault is reported. It can be noted that the vehicle tracks the trajectory with the same error with and without thrusters' faults. Since the control law was designed in 6 DOFs, the difference in the behavior between the horizontal plane, where a small perturbation has been observed, and the vertical plane is caused mainly by the different characteristics of the sensors rather than by the controller itself. In other words, we can expect this nice behavior also in the horizontal plane if a more effective sensorial system would be available.

Another comment need to be done about the depth data. In Fig. 5.3 the plot of the depth for a different control law is shown. It could appear that the tracking performance is better in the last experiment shown in Fig. 5.8. However it has to

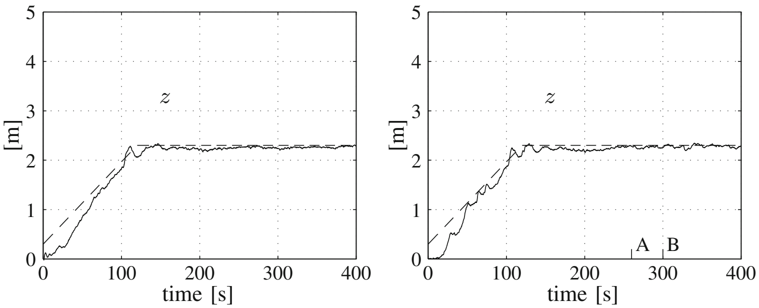


Fig. 5.8 Experiments of fault tolerant control on ODIN. *Left* vehicle movement along z without thrusters' faults. *Right* vehicle movement along z with thrusters' faults. *A* first fault at $t = 260$ s at one *horizontal* thruster. *B* second fault at $t = 300$ s at one *vertical* thruster

be underlined that the experiments are expensive in terms of time and of people involved, the latter are simply successive to the first, the gains, thus, are better tuned.

In Fig. 5.9 the attitude without and with fault is reported. It can be noted that, in case of fault, only after when the two faults arise simultaneously there is a significant transient in the yaw angle that is quick recovered. Comments similar to that done for the linear error in the horizontal plane could be done.

In Fig. 5.10 the voltages of the vertical and horizontal thrusters are reported. We can see that when the thrusters are off, the desired force/moment on the vehicle are redistributed on the working thrusters. From these plots we can also appreciate the different noise on the control caused by the different sensors; due to the null roll and pitch angles, in fact, the vertical thrusters work mainly to track the vertical direction using the depth sensor while the horizontal are mainly use to move the vehicle on the plane.

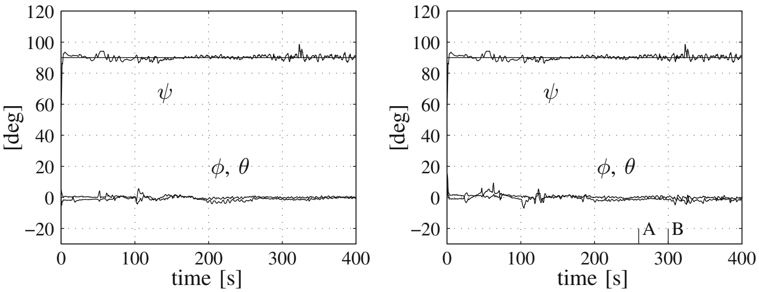


Fig. 5.9 Experiments of fault tolerant control on ODIN. *Left* roll, pitch and yaw without thrusters' faults. *Right* roll, pitch and yaw with thrusters' faults. *A* first fault at $t = 260$ s at one *horizontal* thruster. *B* second fault at $t = 300$ s at one *vertical* thruster

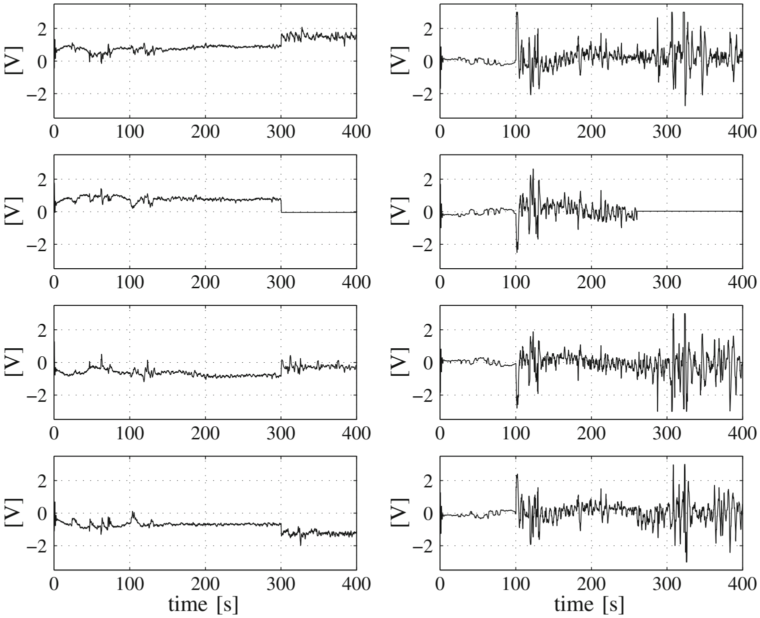
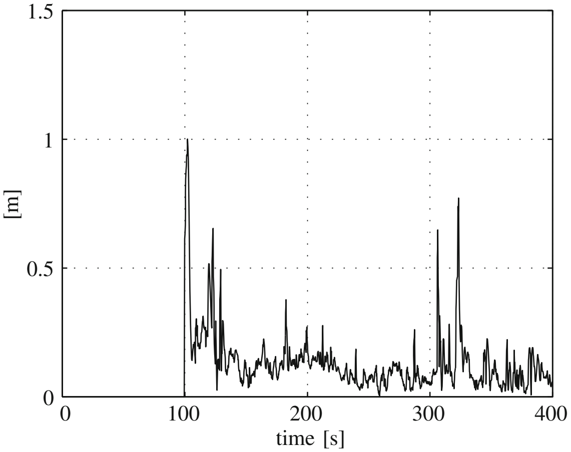


Fig. 5.10 Experiments of fault tolerant control on ODIN. *Left* voltages at *vertical* thrusters. *Right* voltages at *horizontal* thrusters

Fig. 5.11 Experiments of fault tolerant control on ODIN. 2-Norm of the position errors. *A* first fault at $t = 260$ s at one *horizontal* thruster. *B* second fault at $t = 300$ s at one *vertical* thruster



In Fig. 5.11 the 2-norm of the linear error is reported. It can be noted that the first fault does not affect the error while for the second there is a small transient that is recovered.

References

1. H.T. Choi, A. Hanai, S.K. Choi, J. Yuh, Development of an underwater robot. in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems ODIN-III, (IROS 2003)*, vol. 1, (2003), pp. 836–841
2. T. Fossen, *Guidance and Control of Ocean Vehicles* (Wiley, New York, 1994)
3. J. Nie, J. Yuh, E. Kardash, T. Fossen, On-board sensor-based adaptive control of small UUVS in very shallow water. Int. J. Adapt. Control Signal Process. **14**(4), 441–452 (2000)
4. T.K. Podder, G. Antonelli, N. Sarkar, Fault tolerant control of an autonomous underwater vehicle under thruster redundancy: simulations and experiments. in *Proceedings 2000 IEEE International Conference on Robotics and Automation* (San Francisco, CA, 2000), pp. 1251–1256
5. T.K. Podder, G. Antonelli, N. Sarkar, An experimental investigation into the fault-tolerant control of an autonomous underwater vehicle. J. Adv. Robot. **15**(5), 501–520 (2001)
6. T.K. Podder, N. Sarkar, Fault tolerant decomposition of thruster forces of an autonomous underwater vehicle. in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation* (Leuven, Belgium, May 1998), pp. 84–89
7. N. Sarkar, T.K. Podder, G. Antonelli, Fault accommodating thruster force allocation of an AUV considering thruster redundancy and saturation. IEEE Trans. Rob. Autom. **18**(2), 223–233 (2002)

Chapter 6
Kinematic Control of UVMSs

> *"…mais de toutes les sciences la plus absurde, à mon avis et celle qui est la plus capable d'étouffer toute espèce de génie, c'est la géométrie. Cette science ridicule a pour objet des surfaces, des lignes et des points qui n'existent pas dans la nature. On fait passer en esprit cent mille lignes courbes entre un cercle et une ligne droite qui le touche, quoique, dans la réalité, on n'y puisse pas passer un fétu. La géométrie, en vérité, n'est qu'une mauvaise plaisanterie."*
>
> Voltaire, "*Jeannot et Colin* " 1764.

6.1 Introduction

A robotic system is kinematically redundant when it possesses more degrees of freedom than those required to execute a given task. A generic manipulation task is usually given in terms of trajectories for the end effector, specially position and orientation. In this sense, an Underwater Vehicle-Manipulator System is always kinematically redundant due to the DOFs provided by the vehicle itself. However, it is not always efficient to use vehicle thrusters to move the manipulator end effector because of the difficulty of controlling the vehicle in hovering. Moreover, due to the different inertia between vehicle and manipulator, movement of the latter is energetically more efficient. One important results given in [1] concerns, given a generic ground-fixed manipulator with rotational joints, the impossibility to avoid multiple internal singularities. This is true no matter the number of joints and it results in the need for the trajectory planner to always face this problem. Reconfiguration of the whole system is thus required when the manipulator is working at the boundaries of its workspace or close to a kinematic singularity; motion of the sole manipulator, thus, is not always possible or efficient. Also, off-line trajectory planning is not

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_6,

always possible in unstructured environments as in case of underwater autonomous missions.

When a manipulation task has to be performed with an UVMS, the system is usually kept in a confined space (e.g., underwater structure maintenance). The vehicle is then used to ensure station keeping. However, motion of the vehicle can be required for specific purposes, e.g., inspection of a pipeline, reconfiguration of the system, real-time motion coordination while performing end-effector trajectory tracking.

According to the above, a redundancy resolution technique might be useful to achieve system coordination in such a way as to guarantee end-effector tracking accuracy and, at the same time, additional control objectives, e.g., energy savings or increase of system manipulability. To this purpose the task priority redundancy resolution technique [2, 3] is well suited in that it allows the specification of a primary task which is fulfilled with higher priority with respect to a secondary task.

Control of end-effector position/orientation can be obtained also with dynamic control by suitably expressing the mathematical model [4–6]. This approach, successfully implemented for industrial robots, seems not to be suitable for UVMSs for two main reasons: first, in underwater environment the dynamic parameters are usually poorly known; second, the redundancy of the system is not exploited. Some, approaches, moreover, are specifically designed for a 6-DOFs manipulator only.

By limiting our attention to UVMSs, few papers have addressed the problem of inverse kinematics resolution. Reference [7] proposes a local motion planner solved in parallel by a distributed search; this provides an iterative algorithm for an approximate solution. In [8], a task priority approach has been proposed aimed at fulfilling secondary tasks such as reduction of fuel consumption, improvement of system manipulability, and obstacle avoidance. This approach has been further integrated with a fuzzy approach in [9–12] and it will be deeply analyzed in this chapter, an alternative fuzzy approach is proposed in [13]. In [14, 15], a second-order inverse kinematics approach is developed to reduce the total hydrodynamic drag forces of the system. Simulations results are performed on a 6-link vehicle carrying a 3-link manipulator. The same authors also developed a dynamic-based algorithm in [16] that generates the joint trajectories by taking into account the natural frequencies of the two subsystems: vehicle and manipulator; the task-space trajectory is represented by Fourier series and suitably projected on the subsystems. Reference [17] reports an adaptive dynamic controller that uses, as reference trajectory, the output of a first-order inverse kinematics algorithm aimed at satisfying joint limits. In [18], the Authors develop two cost functions devoted at increase the manipulability and respect the joint limits to be used in a task priority approach. In [19], a genetic algorithm-based motion planner is proposed; dividing the workspace in cells, the presence of obstacles and the minimization of the drag forces are taken into consideration. In [20] a distributed kinematic control was developed for coordination of a multi-manipulator system mounted under a free-flying base such as, e.g., an underwater vehicle; the case of a possible under-actuated vehicle is explicitly taken into account. The same group developed an overal strategy, defined as *agility* control [21–24] to also handle control of variables within a range or priorities, a couple of sentences more in Sect. 6.2.4.

The coordinated holding of an object by two underwater manipulators is affoarded in [25] within a task priority strategy and the proper definition of secondary tasks. Numerical simulations illustrate the performances. Further analysisi is proposed in [26].

Reference [27], discusses some interesting practical issues arising in UVMS control; An user interface is developed to cope with the low bitrate of underwater communication, classical Inverse Kinematics algorithms are properly tailored for the SAUVIM set-up and verified in a full-DOFs experimental case study.

Some experimental results of coordinated vehicle-manipulator control are presented in [28], where the manipulator compensate for the vehicle motion.

One of the main concerns in uderwater manipulation is the energy consumption. As discussed widely in Chap. 3, current and restoring forces are the origin of persistent effects that need to be compensated. In addition to the techniques discussed in this chapter, references [29–31] propose a proper index to minimize the restoring influence. Wihtin a similar perspective, in [32], the redundant DOFs of the system are exploited so that the restoring moments assist the UVMS motion by properly optimizing a new performance index within a redundancy resolution approach.

6.2 A Brief Introduction to Kinematic Control

A manipulation task is usually given in terms of position and orientation trajectory of the end effector. The objective of kinematic control is to find suitable vehicle/joint trajectories $\boldsymbol{\eta}(t)$, $\boldsymbol{q}(t)$ that correspond to a desired end-effector trajectory $\boldsymbol{\eta}_{ee,d}(t)$. The output of the inverse kinematics algorithm $\boldsymbol{\eta}_r(t)$, $\boldsymbol{q}_r(t)$ provides the reference values to the control law of the UVMS. This control law will be in charge of computing the driving forces aimed at tracking the reference trajectory for the system while counteracting dynamic effects, external disturbances, and modeling errors.

Equation (2.61)

$$\boldsymbol{\eta}_{ee} = \boldsymbol{k}(\boldsymbol{\eta}, \boldsymbol{q}) \tag{2.61}$$

is invertible only for specific kinematic structures with fixed base. Moreover, the complexity of the relation and the number of solutions, i.e., different joint configurations that correspond to the same end-effector posture, increases with the degrees of freedom. As an example, a simple two-link planar manipulator with fixed base admits two different solutions for a given end-effector position, while up to 16 solutions can be found in the case of 6-DOFs structures. At differential level, however, the relation between joints and end-effector velocities is much more tractable and a theory of kinematic control has been established aimed at solving inverse kinematics of generic kinematic structures.

Equation (2.73)

$$\dot{\boldsymbol{x}}_E = \begin{bmatrix} \dot{\boldsymbol{\eta}}_{ee1} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{R}_B^I, \boldsymbol{q})\boldsymbol{\zeta} \tag{2.73}$$

maps the $(6+n)$-dimensional vehicle/joint velocities into the m-dimensional end-effector task velocities. If the UVMS has more degrees of freedom than those required to execute a given task, i.e., if $(6+n) > m$, the system is redundant with respect to the specific task and the Eqs. (2.61)–(2.73) admit infinite solutions. Kinematic redundancy can be exploited to achieve additional task, beside the given end-effector task. In the following, the typical case $(6+n) \geq m$ will be considered.

The configurations at which $\boldsymbol{J}$ is rank deficient, i.e., $\text{rank}(\boldsymbol{J}) < m$, are termed kinematic singularities. Kinematic singularities are of great interest for several reasons; at a singularity, in fact,

- The mobility of the structure is reduced. If the manipulator is not redundant, this implies that it is not possible to give an arbitrary motion to the end effector;
- Infinite solutions to the inverse kinematics problem might exist;
- *Close* to a kinematic singularity at small task velocities can correspond large joint velocities.

Notice that, in case of UVMS, the Jacobian has always full rank due to the mobility of the vehicle, i.e., a rigid body with 6-DOFs. However, as it will be shown in next sections, movement of the vehicle has to be avoided when unnecessary.

End-effector configuration is not the sole variable of interest. By still keeping the attention on the end-effector one might want to control the sole position, diregarding the orientation, or the opposite. In addition, vehicle-related variables should also be controlled such as, for example, the pitch, to accomodate the restoring forces and avoid an excessive use of thrust.

6.2.1 Possible Tasks

In this section, a possible list of task that need to be controlled is provided. A brief description of why it might be necessary to control this variable is given when needed, together with its dimension m, its symbolic definition and the corresponding Jacobian. For all the tasks, the generic variable

$$\boldsymbol{\sigma}_x = \boldsymbol{\sigma}_x(\boldsymbol{\eta}, \boldsymbol{q}) \in \mathbb{R}^m \tag{6.1}$$

will be used with corresponding Jacobian $\boldsymbol{J}_x(\boldsymbol{\eta}, \boldsymbol{q}) \in \mathbb{R}^{m\times 6+n}$ relating its time derivative to the system velocity $\boldsymbol{\zeta}$

$$\dot{\boldsymbol{\sigma}}_x = \boldsymbol{J}_x(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{\zeta} \tag{6.2}$$

- **End-effector position norm** ($m = 1$). Approaching a target may be achieved by controlling only the norm of the distance between the end-effector and a desired position $\boldsymbol{\eta}_{ee,1d} \in \mathbb{R}^3$:

$$\sigma_x = \sqrt{\left(\boldsymbol{\eta}_{ee,1d} - \boldsymbol{\eta}_{ee,1}\right)^{\mathrm{T}} \left(\boldsymbol{\eta}_{ee,1d} - \boldsymbol{\eta}_{ee,1}\right)} \in \mathbb{R}^1$$

with a corresponding Jacobian

$$\boldsymbol{J}_x = -\frac{\left(\boldsymbol{\eta}_{ee,1d} - \boldsymbol{\eta}_{ee,1}\right)^{\mathrm{T}}}{\left\|\boldsymbol{\eta}_{ee,1d} - \boldsymbol{\eta}_{ee,1}\right\|}\boldsymbol{J}_{pos} \in \mathbb{R}^{1\times 6+n}$$

where $\boldsymbol{J}_{pos} \in \mathbb{R}^{3\times 6+n}$ is the position Jacobian defined in (2.71). Notice that the positional Jacobian in the equation above is simply multiplied by the versor connecting the end-effector to the goal. Numerical issues need to be addressed when the error is *small*, this occurrence may arises not only at steady state but also during the transient for a dynamic desired value. In addition, it is worth noticing the meaning of this task function, the task error is a *sphere* around the desired position, fine position control is thus difficult. This task is implemented in the functions `J01` and `sigma_tilde01`.

- **End-effector obstacle avoidance** ($m = 1$). One effective way to implement end effector obstacle avoidance is to control its distance from the object. The task function, thus, is the same as the previous one but with a desired value different from zero. This task is implemented in the functions `J02` and `sigma_tilde02`. Clearly, if the obstacle is *far* or is not in encumbering the robot motion this task has a negative effect, in the sense that it *attracts* the end-effector to the sphere at a given distance from the obstacle. The need for a supervisor that properly activate the tasks emerges from those considerations.
- **End-effector position** ($m = 3$). A more fine *control* of the end-effector may be achieved by feedback of the whole end-effector position $\boldsymbol{\sigma}_x = \boldsymbol{\eta}_{ee1} \in \mathbb{R}^3$ defined in Sect. 2.9 corresponding to $\boldsymbol{J}_x = \boldsymbol{J}_{pos}$ in which $\boldsymbol{J}_{pos} \in \mathbb{R}^{3\times 6+n}$ is the same Jacobian introduced in Sect. 2.10. This task is implemented in the functions `J03` and `sigma_tilde03`.
- **End-effector orientation** ($m = 3$). Control of the orientation of a rigid body can be addressed, among the other way, by resorting to the quaternion error defined in Eq. (2.12). This has the advantage to allow use of the angular velocity expressed in the inertial frame and correspondingly the orientation Jacobian $\boldsymbol{J}_x = \boldsymbol{J}_{or} \in \mathbb{R}^{3\times 6+n}$ defined in (2.72). This task is implemented in the functions `J04` and `sigma_tilde04`.
- **End-effector configuration** ($m = 6$). The precise control of the end-effector can only be achieved by defining the whole configuration, i.e., position and orientation as task. This is obtained easily by merging the two tasks above and resorting to the Jacobian defined in (2.73) by assuming $\boldsymbol{J}_x = \boldsymbol{J}$. This task is implemented in the functions `J05` and `sigma_tilde05`.
- **End-effector field of view** ($m = 1$). Directional devices or sensors mounted on the end-effector such as, e.g., a laser or a video-camera, do not need that the 3-DOFs of the orientation are controlled but only that the outgoing versor, defined as $\boldsymbol{a} \in \mathbb{R}^3$, is. It is thus possible to define a desired $\boldsymbol{a}_d$ and a corresponding task function

$$\sigma_x = \sqrt{(\boldsymbol{a}_d - \boldsymbol{a})^{\mathrm{T}}(\boldsymbol{a}_d - \boldsymbol{a})} \in \mathbb{R}^1$$

characterized by Jacobian

$$\boldsymbol{J}_x = -\frac{(\boldsymbol{a}_d - \boldsymbol{a})^{\mathrm{T}}}{\|\boldsymbol{a}_d - \boldsymbol{a}\|}\boldsymbol{S}(\boldsymbol{a})\boldsymbol{J}_{or} \in \mathbb{R}^{1\times 6+n} \qquad (6.3)$$

computed by observing that $\boldsymbol{a}_d$ is constant and $\boldsymbol{a}$ may be rewritten as $\boldsymbol{R}_n^I\left[0\ 0\ 1\right]^{\mathrm{T}}$ and its time derivative is given by $\dot{\boldsymbol{a}} = \boldsymbol{S}(\boldsymbol{\omega}_{ee})\boldsymbol{R}_n^I\left[0\ 0\ 1\right]^{\mathrm{T}} = \boldsymbol{S}(\boldsymbol{\omega}_{ee})\boldsymbol{a} = -\boldsymbol{S}(\boldsymbol{a})\boldsymbol{J}_{or}\boldsymbol{\zeta}$. Please notice that, again, there is a denominator zeroing for null errors. This can be handled also by controlling, instead of the norm, the square of the error. This task is implemented in the functions `J06` and `sigma_tilde06`.

- **End-effector relative field of view** ($m = 1$). The task function above allow to assign the end-effector versor $\boldsymbol{a}$ to a desired value $\boldsymbol{a}_d$. If the versor $\boldsymbol{a}_d$, however, is configuration-dependent the Jacobian is not anymore given by (6.3). This is what happens if it is required to *point* no matter what the current position is. The task function, thus remains the same while it is necessary to compute also the Jacobian for $\boldsymbol{a}_d$ now defined as

$$\boldsymbol{a}_d = \frac{\boldsymbol{p}_o - \boldsymbol{\eta}_{ee1}}{\|\boldsymbol{p}_o - \boldsymbol{\eta}_{ee1}\|}$$

where $\boldsymbol{p}_o \in \mathbb{R}^3$ is the position of the point expressed in the inertial frame. On the purpose, let us recall the time derivative of a versor, also called the Poisson formula, that is

$$\frac{\mathrm{d}\hat{\boldsymbol{r}}}{\mathrm{d}t} = \boldsymbol{\omega} \times \hat{\boldsymbol{r}}$$

where $\boldsymbol{\omega}$ is the rotation of the versor that can be computed by observing that it is

$$\dot{\boldsymbol{\eta}}_{ee1} = \boldsymbol{\omega} \times \left(\boldsymbol{p}_o - \boldsymbol{\eta}_{ee1}\right)$$

thus

$$\boldsymbol{\omega} = -\boldsymbol{S}(\boldsymbol{p}_o - \boldsymbol{\eta}_{ee1})^{\dagger}\boldsymbol{J}_{pos}\boldsymbol{\zeta}$$

finally yelding the needed Jacobian as

$$\boldsymbol{J}_x = -\frac{(\boldsymbol{a}_d - \boldsymbol{a})^{\mathrm{T}}}{\|\boldsymbol{a}_d - \boldsymbol{a}\|}\left(-\boldsymbol{S}(\boldsymbol{a}_d)\boldsymbol{S}(\boldsymbol{p}_o - \boldsymbol{\eta}_{ee1})^{\dagger}\boldsymbol{J}_{pos} + \boldsymbol{S}(\boldsymbol{a})\boldsymbol{J}_{or}\right) \in \mathbb{R}^{1\times 6+n}. \qquad (6.4)$$

This task is implemented in the functions `J07` and `sigma_tilde07`.

- **Mechanical joint-limit** ($m = 1$). Any robotic structures exhibits mechanical limits for the joint mobility. Each joint is usually allowed to move in a range, ignoring this limit may cause the robot to incur in an emergency stop. It might be appropriate to define a task that represents the *distance* of the joints from the respective limits. Several metrics may be investigated as the one defined, e.g., in [33]. The simplest way to do it is to control one single joint at once, the task function is thus its position

$$\sigma_x = q_i \in \mathbb{R}^1 \quad \text{with } i \in \{1, \ldots, n\}$$

with a trivial Jacobian $\boldsymbol{J}_x \in \mathbb{R}^{1\times 6+n}$ characterized by null elements except for the one corresponding to the ith joint equal to one. This task is implemented in the functions `J08` and `sigma_tilde08`.

- **Robot manipulability** ($m = 1$). When a Jacobian looses rank its inversion becomes problematic. It is appropriate thus to have a metrics that measure how far the robot is from this so called kinematic singularity. In case of a position-orientation problem of a fixed-based robot a possible index is given by [33]. In case of floating base the Jacobian is always mathematically defined even if, with the manipulator at singular configuration, the motion would be required to the sole vehicle, thus incurring in energetic inefficiencies.
- **Robot nominal configuration** ($m = n$). If the robot exhibits a nominalconfiguration with respect, e.g., to the dynamic manipulability ellipsoid, it might be appropriate to define, probably at a lower priority, the task of controlling its position in the joint space with the simple task

$$\boldsymbol{\sigma}_x = \boldsymbol{q} \in \mathbb{R}^n$$

and trivial Jacobian

$$\boldsymbol{J}_x = \begin{bmatrix} \boldsymbol{O}_{n\times 6} \, \boldsymbol{I}_{n\times n} \end{bmatrix} \in \mathbb{R}^{n\times 6+n}.$$

This task is implemented in the functions `J09` and `sigma_tilde09`.

- **Vehicle orientation** ($m = 3$). Vehicle orientation control falls within the case of rigid body attitude control already briefly reviewed in the Modeling chapter. An effective way to feedback the orientation is by means of quaternions, the task function is thus given by (2.12)

$$\boldsymbol{\sigma}_x = \tilde{\boldsymbol{\varepsilon}} \in \mathbb{R}^3$$

and the Jacobian $\boldsymbol{J}_x \in \mathbb{R}^{3\times 6+n}$

$$\boldsymbol{J}_x = \begin{bmatrix} \boldsymbol{O}_{3\times 3} \, \boldsymbol{R}_B^I \, \boldsymbol{O}_{3\times n} \end{bmatrix}$$

This task is implemented in the functions `J10` and `sigma_tilde10`.

- **Vehicle yaw** ($m = 1$). It is achieved by assigning $\sigma_x = \psi \in \mathbb{R}$ and Jacobian

$$\boldsymbol{J}_x = \begin{bmatrix} \boldsymbol{O}_{1\times 3} \; [0 \; 0 \; 1] \boldsymbol{J}_{k,o}^{-1} \, \boldsymbol{O}_{1\times n} \end{bmatrix}$$

i.e., by simply exciting the third row of the Jacobian $\boldsymbol{J}_{k,o}^{-1}$ defined in (2.2). This task is implemented in the functions `J11` and `sigma_tilde11`.

- **Vehicle roll-pitch** ($m = 2$). In analogous manner, roll and pitch can be controlled by defining $\boldsymbol{\sigma}_x = \begin{bmatrix} \phi \; \theta \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$ and Jacobian

$$\boldsymbol{J}_x = \begin{bmatrix} \boldsymbol{O}_{2\times 3} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{J}_{k,o}^{-1} & \boldsymbol{O}_{2\times n} \end{bmatrix}$$

This task is implemented in the functions `J12` and `sigma_tilde12`.

- **Vehicle-fixed sensor field of view** ($m = 2$). Given a generic vehicle-fixed versor $\boldsymbol{r} \in \mathbb{R}^3$, the task function is defined as

$$\sigma_x = \sqrt{(\boldsymbol{r}_d - \boldsymbol{r})^{\mathrm{T}} (\boldsymbol{r}_d - \boldsymbol{r})} \in \mathbb{R}^1$$

with corresponding Jacobian

$$\boldsymbol{J}_x = \left[\boldsymbol{O}_{1\times 3} \;\; -\tfrac{1}{\sigma}(\boldsymbol{r}_d - \boldsymbol{r})^{\mathrm{T}} \boldsymbol{S}(\boldsymbol{r}) \; \boldsymbol{O}_{1\times n}\right] \in \mathbb{R}^{1\times 6+n} \tag{6.5}$$

- **Vehicle-fixed relative field of view** ($m = 1$). Considerations similar to the ones done for the end-effector motion can be done and are omitted for brevity.
- **Vehicle position norm** ($m = 1$). In some cases it might be necessary to control the position of the vehicle $\boldsymbol{\eta}_{1d} \in \mathbb{R}^3$:

$$\sigma_x = \sqrt{\left(\boldsymbol{\eta}_{1d} - \boldsymbol{\eta}_1\right)^{\mathrm{T}} \left(\boldsymbol{\eta}_{1d} - \boldsymbol{\eta}_1\right)} \in \mathbb{R}^1$$

with a corresponding Jacobian

$$\boldsymbol{J}_x = \left[-\frac{(\eta_1 - \eta_{1d})^{\mathrm{T}}}{\|\eta_1 - \eta_{1d}\|} \boldsymbol{R}_B^I \; \boldsymbol{O}_{1\times 3+n} \right] \in \mathbb{R}^{1\times 6+n}.$$

Here too, the versor connecting the vehicle to the desired position appears and numerical issues need to be addressed when the error is null. This task is implemented in the functions `J13` and `sigma_tilde13`.

- **Vehicle obstacle avoidance** ($m = 1$). The same task above can be used by assigning a task value different from zero as occurred for the end-effector. This task is implemented in the functions `J14` and `sigma_tilde14`.

The most common approaches used to invert the mapping (6.1) are here reported.

Pseudoinverse

The simplest solution is to use the pseudoinverse of the Jacobian matrix [34]

$$\boldsymbol{\zeta}_r = \boldsymbol{J}_x^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q}) \dot{\boldsymbol{\sigma}}_x, \tag{6.6}$$

where

$$\boldsymbol{J}_x^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q}) = \boldsymbol{J}_x^{\mathrm{T}}(\boldsymbol{\eta}, \boldsymbol{q}) \left(\boldsymbol{J}_x(\boldsymbol{\eta}, \boldsymbol{q}) \boldsymbol{J}_x^{\mathrm{T}}(\boldsymbol{\eta}, \boldsymbol{q}) \right)^{-1}.$$

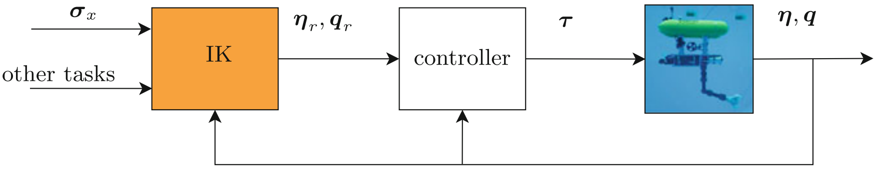


Fig. 6.1 Block scheme illustrating the role of the Inverse Kinematics controller (*in orange*), affoarded in this chapter

This solution corresponds to the minimization of the vehicle/joint velocities in a least-square sense [33], i.e., of the function:

$$E = \frac{1}{2}\zeta^{\mathrm{T}}\zeta \,.$$

Notice that subscript r in ζ_r stands for *reference value*, meaning that those velocities are the desired values for the low-level motion control of the manipulator (see also Fig. 6.1, where a closed-loop inverse kinematics, detailed in next Subsections, is sketched). It is possible to minimize a weighted norm of the vehicle/joint velocities

$$E = \frac{1}{2}\zeta^{\mathrm{T}}\boldsymbol{W}\zeta$$

leading to the *weighted* pseudoinverse:

$$\boldsymbol{J}^{\dagger}_{\mathbf{x},\mathbf{W}} = \boldsymbol{W}^{-1}\boldsymbol{J}_x^{\mathrm{T}}\left(\boldsymbol{J}_x\boldsymbol{W}^{-1}\boldsymbol{J}_x^{\mathrm{T}}\right)^{-1} . \tag{6.7}$$

With this approach, however, the problem of handling kinematic singularities is not addressed and their avoidance cannot be guaranteed.

Augmented Jacobian

Another approach to redundancy resolution is the augmented Jacobian [35]. In this case, a constraint task is added to the end-effector task as to obtain a square Jacobian matrix which can be inverted.

The main drawback of this technique is that new singularities might arise in configurations in which the *original* Jacobian $\boldsymbol{J}_x$ is still full rank. Those singularities, named *algorithmic singularities*, occur when the additional task does cause conflict with the end-effector task.

A similar approach, with the same drawback, is the *extended Jacobian* approach.

Task Priority Redundancy Resolution

By solving (2.73) in terms of a minimization problem of the quadratic cost function $\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{\zeta}$ the general solution [36] is given:

$$\zeta_r = \boldsymbol{J}_a^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{\sigma}}_a + \left(\boldsymbol{I}_N - \boldsymbol{J}_a^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{J}_a(\boldsymbol{\eta}, \boldsymbol{q})\right)\zeta_b = \boldsymbol{J}_a^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{\sigma}}_a + \boldsymbol{N}_a(\boldsymbol{\eta}, \boldsymbol{q})\zeta_b, \tag{6.8}$$

where $N = 6 + n$ and $\boldsymbol{\zeta}_b \in \mathbb{R}^{6+n}$ is an arbitrary vehicle/joint velocity vector.

It can be recognized that the operator $\boldsymbol{N}_a(\boldsymbol{\eta}, \boldsymbol{q})$ projects a generic joint velocity vector in the null space of the Jacobian matrix $\boldsymbol{J}_a$. This corresponds to generating a motion that does not affect the task space $\boldsymbol{\sigma}_a$.

Solution (6.8) can be seen in terms of projection of a secondary task, described by $\boldsymbol{\zeta}_b$, in the null space of the higher priority primary task, i.e., the task described by $\boldsymbol{\sigma}_a$. A first possibility is to choose the vector $\boldsymbol{\zeta}_b$ as the gradient of a scalar objective function $H(\boldsymbol{q})$ in order to achieve a local minimum [36]:

$$\zeta_b = -k_H \nabla H(\boldsymbol{q}), \tag{6.9}$$

where k_H is a scalar gain factor. Another possibility is to chose a primary task $\boldsymbol{\sigma}_{a,d} \in \mathbb{R}^{m_a}$ and a correspondent Jacobian matrix $\boldsymbol{J}_a(\boldsymbol{\eta}, \boldsymbol{q}) \in \mathbb{R}^{m_a \times (6+n)}$

$$\dot{\boldsymbol{\sigma}}_{a,d} = \boldsymbol{J}_a(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{\zeta}.$$

and to design a secondary task $\boldsymbol{\sigma}_{b,d} \in \mathbb{R}^{m_b}$ and a correspondent Jacobian matrix $\boldsymbol{J}_b(\boldsymbol{\eta}, \boldsymbol{q}) \in \mathbb{R}^{m_b \times (6+n)}$:

$$\dot{\boldsymbol{\sigma}}_{b,d} = \boldsymbol{J}_b(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{\zeta}$$

for which the vector of joint velocity is then given by [2, 3]:

$$\zeta_r = \boldsymbol{J}_a^{\dagger}\dot{\boldsymbol{\sigma}}_a + \left(\boldsymbol{J}_b\left(\boldsymbol{I}_N - \boldsymbol{J}_a^{\dagger}\boldsymbol{J}_a\right)\right)^{\dagger}\left(\dot{\boldsymbol{\sigma}}_{b,d} - \boldsymbol{J}_b\boldsymbol{J}_a^{\dagger}\dot{\boldsymbol{x}}_{a,d}\right). \tag{6.10}$$

However, for this solution too, the problem of the algorithmic singularities still remains unsolved. In this case, it is possible to experience an algorithmic singularity when $\boldsymbol{J}_b$ and $\boldsymbol{J}_a$ are full rank but the matrix $\boldsymbol{J}_b\left(\boldsymbol{I}_N - \boldsymbol{J}_a^{\dagger}\boldsymbol{J}_a\right)$ looses rank. Extension of the approach to several tasks for highly redundant systems can be achieved by generalization of (6.10), as described in [37].

Singularity-Robust Task Priority Redundancy Resolution

A robust solution to the occurrence of the algorithmic singularities is based on the following mapping [38]:

$$\zeta_r = \boldsymbol{J}_a^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{\sigma}}_{a,d} + \left(\boldsymbol{I}_N - \boldsymbol{J}_a^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{J}_a(\boldsymbol{\eta}, \boldsymbol{q})\right)\boldsymbol{J}_b^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{\sigma}}_{b,d}. \tag{6.11}$$

This algorithm has a clear geometrical interpretation: the two tasks are separately inverted by the use of the pseudoinverse of the corresponding Jacobian; the vehicle/joint velocities associated with the secondary task are further projected in the null space of the primary task $\boldsymbol{J}_a$. Extension to several tasks requires some care and it can be found in [39]:

$$\boldsymbol{\zeta}_r = \boldsymbol{J}_a^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{\sigma}}_{a,d} + \boldsymbol{N}_a \boldsymbol{J}_b^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{\sigma}}_{b,d} + \boldsymbol{N}_{ab} \boldsymbol{J}_c^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{\sigma}}_{c,d} \tag{6.12}$$

where a third task c, characterized by $\boldsymbol{\sigma}_c$ and $\boldsymbol{J}_c$ has been introduced of dimension m_c and the null-space projector $\boldsymbol{N}_{ab}$ is computed with respect to the matrix

$$\boldsymbol{J}_{ab} = \begin{bmatrix} \boldsymbol{J}_a \\ \boldsymbol{J}_b \end{bmatrix}$$

i.e., the matrix obtained by *impiling* the higher priority tasks.

Damped Least-Squares Inverse Kinematics Algorithms

The problem of inverting ill-conditioned matrices that might occur with all the above algorithms can be avoided by resorting to the damped least-square inverse given by [40]:

$$\boldsymbol{J}_x^{\#}(\boldsymbol{\eta}, \boldsymbol{q}) = \boldsymbol{J}_x^{\mathrm{T}}(\boldsymbol{\eta}, \boldsymbol{q}) \left(\boldsymbol{J}_x(\boldsymbol{\eta}, \boldsymbol{q}) \boldsymbol{J}_x^{\mathrm{T}}(\boldsymbol{\eta}, \boldsymbol{q}) + \lambda^2 \boldsymbol{I}_m \right)^{-1},$$

where $\lambda \in \mathbb{R}$ is a damping factor.

In this case, the introduction of a damping factor allows solving the problem from the numerical point of view but, on the other hand, it introduces a reconstruction error in all the velocity components. Better solutions can be found with variable damping factors or damped least-squares with numerical filtering [40, 41].

Closed-Loop Inverse Kinematic Algorithms

The numerical implementation of the above algorithms would lead to a numerical drift when obtaining vehicle/joint positions by integrating the vehicle/joint velocities. A closed loop version of the above equations can then be adopted. By considering as primary task the end-effector position/orientation, Eq. (6.12), as an example, would become:

$$\begin{aligned} \boldsymbol{\zeta}_r \; = \; & \boldsymbol{J}_a^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q}) \left(\dot{\boldsymbol{\sigma}}_{a,d} + \boldsymbol{k}_a \tilde{\boldsymbol{\sigma}}_a \right) + \boldsymbol{N}_a \boldsymbol{J}_b^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q}) \left(\dot{\boldsymbol{\sigma}}_{b,d} + \boldsymbol{k}_b \tilde{\boldsymbol{\sigma}}_b \right) \\ & + \; \boldsymbol{N}_{ab} \boldsymbol{J}_c^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q}) \left(\dot{\boldsymbol{\sigma}}_{c,d} + \boldsymbol{k}_c \tilde{\boldsymbol{\sigma}}_c \right) \end{aligned} \tag{6.13}$$

where the symbol tilde denotes the errors and $\boldsymbol{k}_a \in \mathbb{R}^{m_a \times m_a}$, $\boldsymbol{k}_b \in \mathbb{R}^{m_b \times m_b}$ and $\boldsymbol{k}_c \in \mathbb{R}^{m_c \times m_c}$ are design matrix gains to be chosen so as to ensure convergence to zero of the corresponding errors.

If the task considered is position control, its reconstruction error is simply given by the difference between the desired and the reconstructed values. In case of the orientation, however, care in the definition of such error is required to ensure convergence to the desired value. In this work, the quaternion attitude representation is used [42]; the vector $\tilde{\boldsymbol{\sigma}}_x$ for the task defined in (2.73) is then given by [43, 44]:

$$\tilde{\boldsymbol{\sigma}}_x = \begin{bmatrix} \boldsymbol{\eta}_{ee1,d} - \boldsymbol{\eta}_{ee1,r} \\ \eta_r \boldsymbol{\varepsilon}_d - \eta_d \boldsymbol{\varepsilon}_r - \boldsymbol{S}(\boldsymbol{\varepsilon}_d)\boldsymbol{\varepsilon}_r \end{bmatrix}, \tag{6.14}$$

where $\mathcal{Q}_d = \{\eta_d, \boldsymbol{\varepsilon}_d\}$ and $\mathcal{Q}_r = \{\eta_r, \boldsymbol{\varepsilon}_r\}$ are the desired and reference attitudes expressed by quaternions, respectively, and $\boldsymbol{S}(\cdot)$ is the matrix operator performing the cross product.

The obtained $\boldsymbol{\zeta}_r$ can then be used to compute the position and orientation of the vehicle $\boldsymbol{\eta}_r$ and the manipulator configuration $\boldsymbol{q}_r$:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\eta}_r(t) \\ \boldsymbol{q}_r(t) \end{bmatrix} &= \int_0^t \begin{bmatrix} \dot{\boldsymbol{\eta}}_r(\sigma) \\ \dot{\boldsymbol{q}}_r(\sigma) \end{bmatrix} d\sigma + \begin{bmatrix} \boldsymbol{\eta}(0) \\ \boldsymbol{q}(0) \end{bmatrix} \\ &= \int_0^t \boldsymbol{J}_k^{-1}(\sigma)\boldsymbol{\zeta}_r(\sigma) d\sigma + \begin{bmatrix} \boldsymbol{\eta}(0) \\ \boldsymbol{q}(0) \end{bmatrix}. \end{aligned} \tag{6.15}$$

As customary in kinematic control approaches, the output of the above inverse kinematics algorithm provides the reference values to the dynamic control law of the vehicle-manipulator system. This dynamic control law will be in charge of computing the driving forces, i.e., the vehicle thrusters and the manipulator torques. The kinematic control algorithm is independent from the dynamic control law as long as the latter is a vehicle/joint space-based control, i.e., it requires as input the reference vehicle-joint position and velocity. In the literature number of such control laws have been proposed that are suitable to be used within the proposed kinematic control approach; a literature survey is presented in Chap. 7.

Remarkably, all those inverse kinematics approaches are suitable for real-time implementation. Of course, depending on the specific algorithm, a different computational load is required [38].

Transpose of the Jacobian

A simple algorithm, conceptually similar to the closed loop approach, is given by the use of the transpose of the Jacobian. In this case, the joint velocities are given by [33]:

$$\boldsymbol{\zeta}_r = \boldsymbol{J}_x^{\mathrm{T}} \boldsymbol{K}_x \tilde{\boldsymbol{\sigma}}_x.$$

6.2.2 How to Select Tasks and Priority

Tasks need to be combined, it is not possible, e.g., to simply control the end-effector position while disregarding the vehicle roll and pitch or the vehicle obstacle avoidance. The possibility to *stack* all the tasks function in one higher-dimensional one would lead to the augmented Jacobian approach, object of drawbacks already discussed.

Let us consider a task-priority approach. While it is *intuitive* that it is meaningless to assign as primary task the end-effector position norm and as secondary the end-effector position, it is less evident if, e.g., an end-effector task may be matched or not with a manipulability task. The answer is not simple and, from an analytical perspective, has been discussed in [39]. Without entering into the details, it is interesting to recall the main result concerning 3 tasks in closed loop:

The gains are properly selected, the Jacobians associated with tasks a and b and the Jacobians associated with tasks c and the augmented task ab satisfy the Independence condition given, i.e.,

$$\begin{cases} \rho(\boldsymbol{J}_a^{\mathrm{T}}) + \rho(\boldsymbol{J}_b^{\mathrm{T}}) = \rho\left(\left[\boldsymbol{J}_a^{\mathrm{T}}\ \boldsymbol{J}_b^{\mathrm{T}}\right]\right) \\ \rho(\boldsymbol{J}_c^{\mathrm{T}}) + \rho(\boldsymbol{J}_{ab}^{\mathrm{T}}) = \rho\left(\left[\boldsymbol{J}_c^{\mathrm{T}}\ \boldsymbol{J}_{ab}^{\mathrm{T}}\right]\right) \end{cases}$$

where $\rho(\cdot)$ is the rank operation, then the origin of the task error vectors is asymptotically stable. Moreover, the assumptions on the gains and the independency conditions are only sufficient.

This condition is, however, mainly a theoretical result since all the Jacobian are configuration dependent and it is not possible to verify this condition in all the configuration space. A possibility is to check the Jacobian ranks during the movement and eventually implement some ad-hoc solutions for incompatible tasks.

The priority of a task is also not trivial. While obstacle avoidance is usually considered as an higher priority task the manipulability prioriryt may depend on the current configuration. It may be considered as a minor optimization when the manipulator is in a *good* configuration that becomes critical for close-to-singularity configurations. There is the need for a *supervisor* that select tasks and priorities.

6.2.3 The Underactuated Case

It may arrive that some of the DOFs are not actuated, it is the case, e.g., of the roll and pitch angles for several ROVs or for quadrotors holding a manipulator [45]. In this case it is still be possible to implement IK algorithms; let us partition the velocity vector $\boldsymbol{\zeta} \in \mathbb{R}^{6+n}$ in actuated $\boldsymbol{\zeta}_{act} \in \mathbb{R}^{n_{act}}$ and not actuated $\boldsymbol{\zeta}_{\overline{act}} \in \mathbb{R}^{n_{\overline{act}}}$ DOFs with $n_{act} + n_{\overline{act}} = 6 + n$. It is possible to rewrite Eq. (6.2) as:

$$\dot{\boldsymbol{\sigma}}_x = \boldsymbol{J}_{x,act}(\boldsymbol{R}_B^I, \boldsymbol{q})\zeta_{act} + \boldsymbol{J}_{x,\overline{act}}(\boldsymbol{R}_B^I, \boldsymbol{q})\zeta_{\overline{act}} \tag{6.16}$$

to be solved in its closed loop version with respect to the actuated velocities

$$\zeta_{act} = \boldsymbol{J}_{x,act}^{\dagger} \left(\dot{\boldsymbol{\sigma}}_x + \boldsymbol{k}_x \tilde{\boldsymbol{\sigma}}_x - \boldsymbol{J}_{x,\overline{act}}\zeta_{\overline{act}}\right). \tag{6.17}$$

6.2.4 Further Readings

Other aspects have not been addressed here for sake of space. The interested reader may find interesting the case of handling transitions when switching among the tasks [46, 47].

Of interest is also the case of controlling task function to a *range* of values instead of zeroing the corresponding error variable. This is the case, e.g., of the joint mechanical limit for which it is not critical to keep the joint at a certain value but rahter to avoid that it is close to its limits. This case has been defined with the term *set-objectives*, in opposition to the *precision-objectives*, and handle together within a framework defined as *agility control* by the group of Genova [21–24]. The various task functions are associated to properly shaped analytical (*penalty*) functions that translate in mathematical language the concepts above. The need to impose a priority among the tasks still hold, in general, this approach consider the set-objective as associated to security and thus impose them an high priority. When the tasks are frozen, i.e., a certain number of that need to be controlled in a fixed priority, the algorithm acts as a normal null-space-based priority approach [48]. However, the use of the penalty functions allow to *automatically* exclude the statisfied tasks from the priority; Intuitively, their associated null space is thus *full* and given unchanged to the lower priority task to try its fullfillment. The tasks are excluded smoothly by weighting their penalty according to the value of the penalty function, during those transients, the algorithms acts similarly to a cooperative behavioral control [49].

In practical implementation it is often good practice to limit the controlling signal. Saturation in the framework of null-space-based kinematic control, has been affoarded in some recent works [50–52].

This chapter deals with the case of kinematic control, i.e., the first-order mapping between joint position and task variables. There is a wide literature dealing also with acceleration or torque level approaches. Among them, the seminal work [53] and its extension to the humanoid case [54].

In [55, 56] an interesting theoretical and experimental comparison among various kind of task space control with redundancy resolution is presented. Resolution at velocity, acceleration and torque level is considered for control problems where the primary task is always the end-effector configuration while the secondary is the optimization of a proper functional.

6.3 The Drag Minimization Algorithm

In 1999 Sarkar and Podder [14, 15] suggest to use the system redundancy in order to minimize the total hydrodynamic drag. Roughly speaking the proposed kinematic control generates a coordinate vehicle/manipulator motion so that the resulting trajectory incrementally reduce the total drag encountered by the system.

The second-order version of (6.8) is given by

$$\dot{\boldsymbol{\zeta}}_r = \boldsymbol{J}^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q}) \left(\dot{\boldsymbol{x}}_{E,d} - \dot{\boldsymbol{J}}(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{\zeta}\right) + \left(\boldsymbol{I}_N - \boldsymbol{J}^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{J}(\boldsymbol{\eta}, \boldsymbol{q})\right) \dot{\boldsymbol{\zeta}}_a \,, \tag{6.18}$$

where, again, the vector $\dot{\boldsymbol{\zeta}}_a$ is arbitrary and can be used to optimize some performance criteria that can be chosen, similarly to Eq. (6.9), as

$$\dot{\boldsymbol{\zeta}}_a = -k_H \nabla H(\boldsymbol{q}) \,. \tag{6.19}$$

The Authors propose the following scalar objective function

$$H(\boldsymbol{q}) = \boldsymbol{D}^{\mathrm{T}}(\boldsymbol{q}, \boldsymbol{\zeta}) \boldsymbol{W} \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta}) \tag{6.20}$$

where $\boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})$ is the damping matrix and $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ is a positive definite weight matrix. By properly selecting the weight matrix it is possible to shape the influence of the drag of the individual components on the total system's drag. A possible choice for $\boldsymbol{W}$ is a diagonal matrix [15].

The method is tested in detailed simulations where the drag coefficients are supposed to be known. As noticed by the Authors, in practical situations, the drag coefficient need to be identified and this can not be an easy task; it must be noted, however, that theoretical drag coefficient are available in the literature for the most common shapes. As a first approximation it is possible to model the vehicle as an ellipsoid and the manipulator arms as cylinders. The proposed method, thus, even being approximated, provides information of wider interest.

It is worth noticing that drag minimization has been the objective of several approaches, even not based on kinematics control, such as, e.g., [19]; in [57], the Authors propose to utilize a genetic algorithm to be trained over a periodic motion in order to estimate the trajectory's parameters that minimize the directional drag force in the task space.

6.4 The Joint Limits Constraints

Sarkar, Yuh and Podder, in 1999 [17] take into account the problem of handling the manipulators' joints limits.

The Authors propose to suitably modifying the weight matrix $\boldsymbol{W}$ in Eq. (6.7). In detail, $\boldsymbol{W}$ is chosen as a diagonal matrix the entries of which are related to a proper function. The approach might also take into account the vehicle position, but, for seek of simplicity, let us consider only the joints positions by defining

$$H(\boldsymbol{q}) = \sum_{i=1}^{n} \frac{1}{c_i} \frac{q_{i,\max} - q_{i,\min}}{(q_{i,\max} - q_i)(q_i - q_{i,\min})}$$

with $c_i > 0$ and the subscript max and min that obviously denotes the two joint limits. This function [17] inherits the concepts developed in [58].

Its partial derivative with respect to the joint positions is given by

$$\frac{\partial H(\boldsymbol{q})}{\partial q_i} = \frac{1}{c_i} \frac{(q_{i,\max} - q_{i,\min})(2q_i - q_{i,\max} - q_{i,\min})}{(q_{i,\max} - q_i)^2 (q_i - q_{i,\min})^2}.$$

The elements of the weight matrix are then defined as

$$W_{i,i} = 1 + \left\| \frac{\partial H(\boldsymbol{q})}{\partial q_i} \right\|,$$

in fact, it can be easily observed that the element goes to 1 when the joint is in the center of its allowed range and goes to infinity when the joint is approaching its limit.

As a further improvement it is possible to relate the weight also to the direction of the joint by defining

$$W_{i,i} = \begin{cases} 1 + \left\| \dfrac{\partial H(\boldsymbol{q})}{\partial q_i} \right\| & \text{if } \Delta \left\| \dfrac{\partial H(\boldsymbol{q})}{\partial q_i} \right\| > 0 \\ 1 & \text{if } \Delta \left\| \dfrac{\partial H(\boldsymbol{q})}{\partial q_i} \right\| \leq 0 \end{cases}$$

In Sect. 6.6, the joint limits are part of a number of tasks handled with a fuzzy approach. In [18], Jun, Lee and Lee propose a first order task priority approach with the optimization of specific cost functions developed for the ROV named KORDI. The joints constraints are taken into account also.

6.5 Singularity-Robust Task Priority

To achieve an effective coordinated motion of the vehicle and manipulator while exploiting the redundant degrees of freedom available, Antonelli and Chiaverini, in [8], resort to the singularity-robust task priority redundancy resolution technique. The velocity vector $\boldsymbol{\zeta}_r$ is then computed as shown in (6.13).

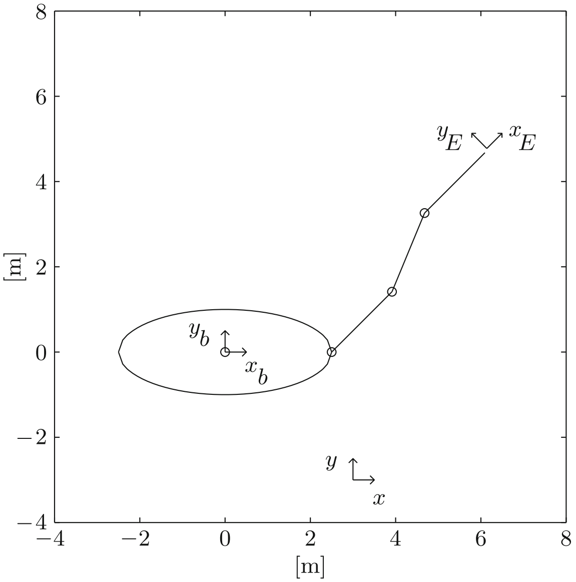
In the case of a UVMS, the primary task vector will usually include the end-effector task vector, while the secondary task vector might include the vehicle position coordinates. This choice is aimed at achieving station keeping of the vehicle as long as the end-effector task can be fulfilled with the sole manipulator arm. It is worth noticing that this approach is conceptually similar to the macro-micro manipulator approach [35]; the main difference is that the latter requires dynamic compensation of the whole system while the former is based on a kinematic control approach. This is advantageous for underwater applications in which uncertainty on dynamic parameters is experienced.

Simulations

Let consider a 9-DOF UVMS constituted by the Naval Postgraduate School AUV Phoenix [59] with a 3-DOF planar manipulator arm. For the sake of clarity, in this first group of simulations, the attention we was restricted to planar tasks described in the plane of the manipulator, that is mounted horizontally. Therefore, let us consider six degrees of freedom in the system which is characterized by the three vehicle coordinates x, y, ψ, and the three end-effector coordinates x_E, y_E, ψ_E all expressed in a earth-fixed frame; the three vehicle coordinates z, θ, ϕ are assumed to be constant. A sketch of the system as seen from the bottom is reported in Fig. 6.2, where the earth-fixed, body-fixed, and end-effector reference frames are also shown.

A station keeping task is considered as first case study. During station keeping, the thrusters must react to the ocean current the strength of which exhibits a quadratic dependence on the relative velocity [60]. However, if the model of the NPS AUV [59] is considered it can be easily recognized that, the drag in the $\boldsymbol{x}_b$ direction is really smaller with respect to the drag in the $\boldsymbol{y}_b$. This suggests to attempt keeping the fore aft direction of the vehicle aligned with the ocean current in order to reduce energy consumption. A similar problem has been affoarded also in [61].

Fig. 6.2 Sketch of the simulated UVMS seen from the bottom



To implement the proposed approach, it is proposed to consider as primary task both the end-effector position + orientation and the vehicle orientation, i.e.,

$$\boldsymbol{x}_p = \left[x_E \, y_E \, \psi_E \, \psi \right]^{\mathrm{T}},$$

and as secondary task the vehicle position, i.e.,

$$\boldsymbol{x}_s = \left[x \, y \right]^{\mathrm{T}}.$$

In a first simulation the end-effector has to maintain its position and orientation while the vehicle will change its orientation to minimize the effect of the ocean current; it is desired to keep the vehicle position constant, if possible. Let the initial configuration of the vehicle be

$$\begin{aligned} x &= 0 \quad \mathrm{m}, \\ y &= 0 \quad \mathrm{m}, \\ \psi &= 0 \quad \mathrm{rad}, \end{aligned}$$

and the manipulator joint angles be

$$\boldsymbol{q} = \left[1.47 \ -1 \, 0.3 \right]^{\mathrm{T}} \quad \mathrm{rad},$$

corresponding to the end-effector location

$$\begin{aligned} x_E &= 5.92 \quad \mathrm{m}, \\ y_E &= 4.29 \quad \mathrm{m}, \\ \psi_E &= 0.77 \quad \mathrm{rad}. \end{aligned}$$

The desired values of the end-effector variables and vehicle position are coincident with their initial value. The desired final value of the vehicle orientation is 0.78 rad as given, e.g., by a current sensor; the time history of the desired value ψ_d is computed according to a quintic polynomial interpolating law with null initial and final velocities and accelerations and a duration of 10 s. The algorithm's gains are

$$\begin{aligned} \boldsymbol{K}_p &= \mathrm{diag}\{10, 10, 10, 1\}, \\ \boldsymbol{K}_s &= \mathrm{diag}\{2, 2\}. \end{aligned}$$

The simulation results are reported in Figs. 6.3 and 6.4. It can be recognized that the task is successfully executed, in that the end-effector location and vehicle position are held while the vehicle body is re-oriented to align with the ocean current. Remarkably, the obtained vehicle reference trajectory is smooth.

A second simulation, starting from the same initial system configuration, considers an end-effector trajectory that cannot be tracked by sole manipulator motion.

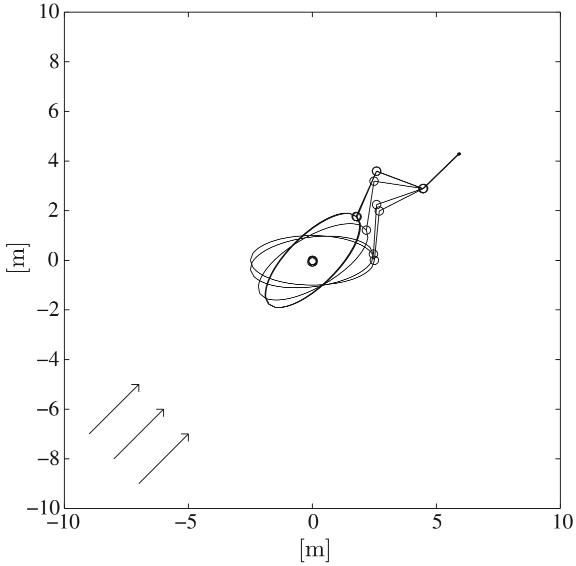


Fig. 6.3 Re-orientation of the vehicle body with fixed end-effector location

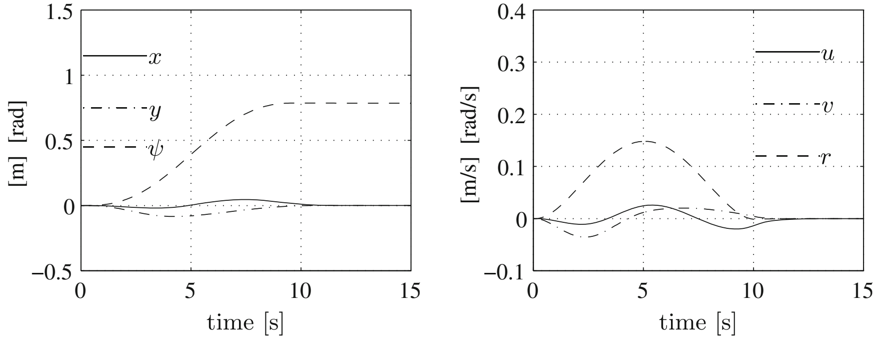


Fig. 6.4 Time history of vehicle position and velocity variables when the vehicle reconfigure itself while keeping a fixed end-effector position/orientation

Therefore, the vehicle must be moved to allow the manipulator end-effector to track its reference trajectory. Also in this simulation, alignment of the vehicle fore aft direction with the ocean current is pursued.

The desired end-effector trajectory is a straight-line motion starting from the same initial location as in the previous simulation and lasting at the final location

$$\begin{aligned} x_E &= 8.00 \quad \text{m}, \\ y_E &= 9.00 \quad \text{m}, \\ \psi_E &= 0.78 \quad \text{rad}. \end{aligned}$$

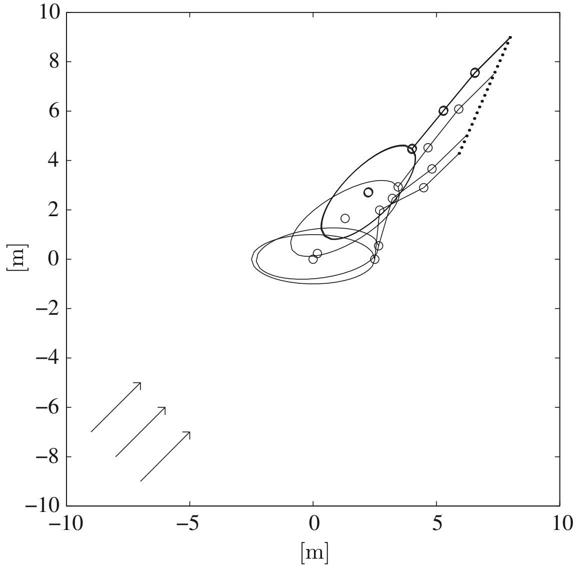
The path is followed according to a quintic polynomial interpolating law with null initial and final velocities and acceleration and a duration of 10 s. The other task variables and gains are the same as in the previous simulation; remarkably, the desired values of the vehicle position variables are coincident with their initial value also in this case.

The simulation results are reported in Figs. 6.5 and 6.6. It can be recognized that the primary task is successfully executed, in that the end-effector location and vehicle orientation achieve their target. On the other hand, the vehicle moves from its initial position despite the secondary task demands for station keeping. Remarkably, the obtained vehicle reference trajectory is smooth.

To show generality of the proposed approach a second case study has been developed. A drawback of the previous case study might be that the manipulator arm is almost completely stretched out when the end-effector trajectory requires large displacements going far from the vehicle body. Nevertheless, this is related to our choice to keep the position of the vehicle constant and to align the fore aft direction with the ocean current. To overcome this drawback, a different choice of the tasks to be fulfilled is necessary. In particular, the task of vehicle re-orientation might be replaced with the task of keeping the manipulator arm in dexterous configurations. To this aim, it would be possible to use a task variable expressing a manipulability measure of the manipulator arm [62]. In this simple case, it is clear that arm singularities occur when $q_2 = 0$; therefore, the use of q_2 as manipulability task variable would reduce the computational burden of the algorithm.

To implement the proposed approach in this second case study, both the end-effector position+orientation and the second manipulator joint variable are thus

Fig. 6.5 Re-orientation of the vehicle body with given end-effector trajectory



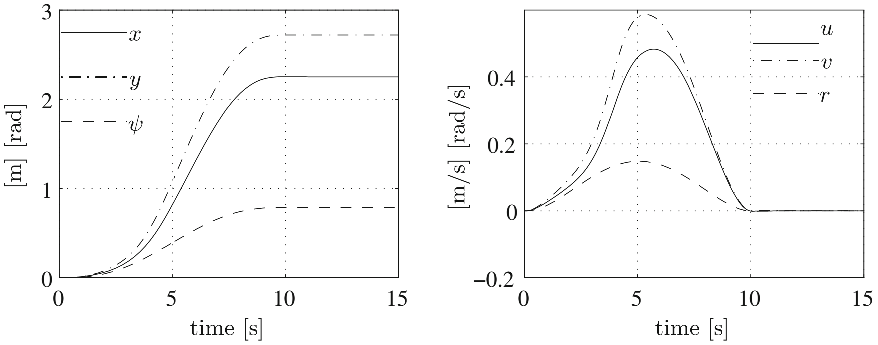


Fig. 6.6 Time history of vehicle position and velocity variables with given end-effector trajectory

considered as primary task, i.e.

$$\boldsymbol{x}_p = \left[x_E \, y_E \, \psi_E \, q_2\right]^{\mathrm{T}},$$

and as secondary task the vehicle position, i.e.

$$\boldsymbol{x}_s = \left[x \, y\right]^{\mathrm{T}}.$$

Starting from the same initial system configuration as before, in the simulation the same end-effector trajectory has been assigned while manipulator joint 2 is driven far from zero; it is desired to keep the vehicle position constant, if possible.

The desired final value of q_2 is -0.78 rad; the time history of the desired value $q_{2,d}$ is computed according to a quintic polynomial interpolating law with null initial and final velocities and acceleration and a duration of 10 s. The algorithm's gains are

$$\boldsymbol{K}_p = \mathrm{diag}\{10, 10, 10, 1\}, \tag{6.21}$$

$$\boldsymbol{K}_s = \mathrm{diag}\{2, 2\}. \tag{6.22}$$

The simulation results are reported in Figs. 6.7 and 6.8.

It can be recognized that the primary task is successfully executed, in that the end-effector location and manipulator joint 2 achieve their target. On the other hand, the vehicle moves despite the secondary task demands for station keeping. Remarkably, the obtained vehicle reference trajectory is smooth.

To underline the energetic difference in the station keeping task considered in the first case study when executed with and without vehicle re-orientation, a third case study has been developed. Two simulations have been performed considering the full-dimensional dynamic model of the NPS AUV under a sliding mode control law [63] and the following constant and irrotational current

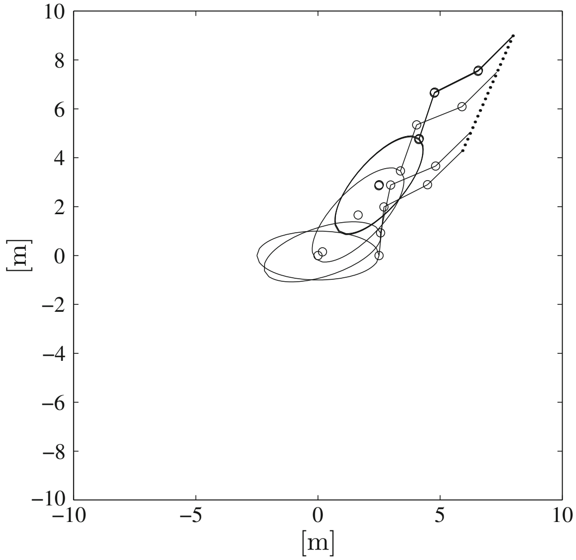


Fig. 6.7 Tracking of a given end-effector trajectory with manipulator dexterity

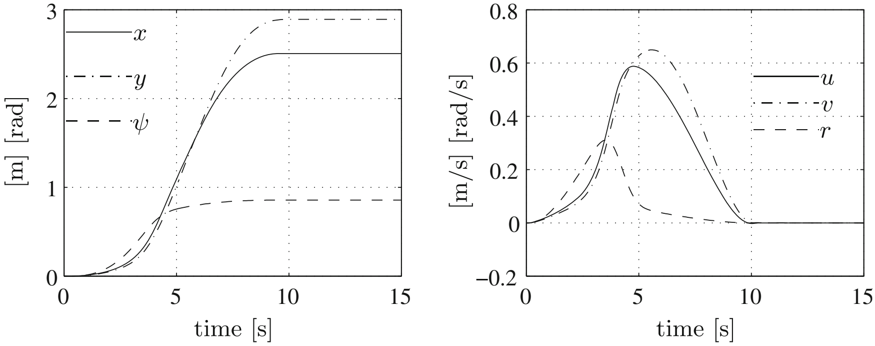


Fig. 6.8 Time history of vehicle position and velocity variables with given end-effector trajectory and manipulator dexterity

$$\boldsymbol{\nu}_c^I = \left[0.1\sqrt{2}\,0.1\sqrt{2}\,0\,0\,0\,0\right]^{\mathrm{T}}.$$

In the first simulation the vehicle stays still and the generalized control forces are required to only compensate the current effect, since the NPS AUV is neutrally buoyant and $\theta = \phi = 0$. In the second simulation the vehicle moves according to the results of the first simulation in the first case study; thus, the generalized control forces are used to both move the vehicle and to compensate the current effect.

Figure 6.9 reports the time histories of the 2-norm of the control forces and moments acting on the vehicle as obtained in the two simulations.

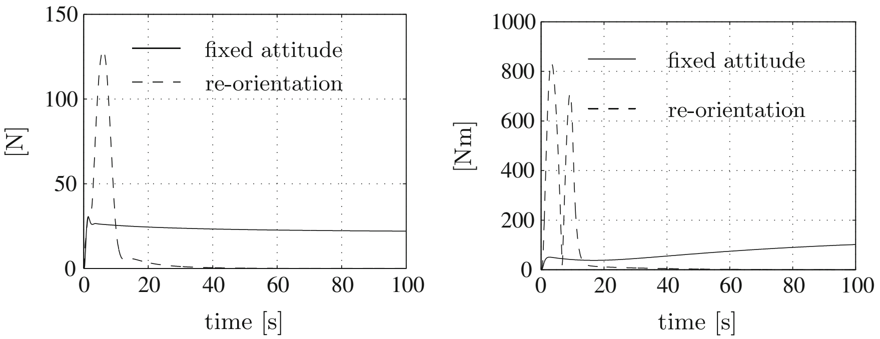


Fig. 6.9 Time history of the 2-norm of the vehicle forces (*left*) and moments (*right*)

Table 6.1 Time integral of the force and moment 2-norms: (a) without re-orientation; (b) with re-orientation

	a	b
$\int \|f\|$	2300	800
$\int \|m\|$	9500	5800

It is easy to recognize that during the reconfiguration the proposed solution is more energy-consuming than the fixed-attitude solution; nevertheless, after the re-orientation has been achieved, the energy consumption required by the proposed technique is negligible. Therefore, the proposed solution becomes the more attractive the longer is the duration of the manipulation task.

For the sake of argument, Table 6.1 reports the time integral of the 2-norms of force and moment obtained in the two simulations over a 100 s task duration.

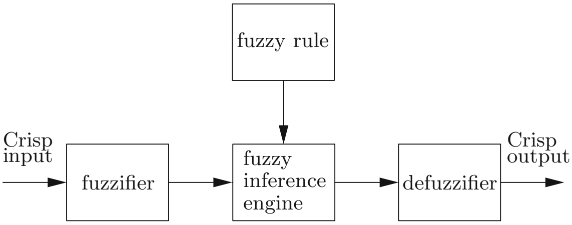
6.6 Fuzzy Inverse Kinematics

Because of the different inertia characteristics of the vehicle and of the manipulator, it would be preferable to perform fast motions of small amplitude by means of the manipulator while leaving to the vehicle the execution of slow gross motions. This might be achieved by adopting the weighted pseudoinverse of Eq. (6.7) with the $(6+n) \times (6+n)$ matrix $\boldsymbol{W}^{-1}$

$$\boldsymbol{W}^{-1}(\beta) = \begin{bmatrix} (1-\beta)\boldsymbol{I}_6 & \boldsymbol{O}_{6\times n} \\ \boldsymbol{O}_{n\times 6} & \beta\boldsymbol{I}_n \end{bmatrix}, \tag{6.23}$$

where β is a weight factor belonging to the interval [0, 1] such that $\beta = 0$ corresponds to sole vehicle motion and $\beta = 1$ to sole manipulator motion.

Fig. 6.10 Mamdani fuzzy inference system



During the task execution, setting a constant value of β would mean to fix the motion distribution between the vehicle and the manipulator. Nevertheless, the use of a fixed weight factor inside the interval [0, 1] has a drawback: it causes motion of the manipulator also if the desired end-effector posture is out of reach; on the other hand, it causes motion of the vehicle also if the manipulator alone could perform the task.

Another problem is the need to handle a large number of variables; UVMSs, in fact, are complex systems and several variables must be monitored during the motion, e.g., the manipulator manipulability, the joint range limits to avoid mechanical breaks, the vehicle roll and pitch angles for correct tuning of the proximity sensors, the yaw angle to exploit the vehicle shape in presence of ocean current, etc. As it can be easily understood, it is quite difficult to handle all these terms without a kinematic control approach. Nevertheless, the existing techniques do not allow to find a flexible and reliable solution.

To overcome this drawback a fuzzy theory approach has therefore been considered at two different levels. First, it is required to manage the distribution of motion between the vehicle and the manipulator; second, it is required to consider multiple secondary tasks that are activated only when the corresponding variable is outside (inside) a desired range. This can be done using different weight factors adjusted on-line according to the Mamdani fuzzy inference system [64] shown in Fig. 6.10.

In detail, the crisp outputs are the scalar β of (6.23) that distributes the desired end-effector motion between the vehicle and the manipulator and a vector of coefficients α_i that are used in the task priority equation as follows

$$\boldsymbol{\zeta} = \boldsymbol{J}_W^{\dagger} \left(\dot{\boldsymbol{x}}_{E,d} + \boldsymbol{k}_E \boldsymbol{e}_E \right) + \left(\boldsymbol{I} - \boldsymbol{J}_W^{\dagger} \boldsymbol{J}_W \right) \left(\sum_i \alpha_i \boldsymbol{J}_{s,i}^{\dagger} \boldsymbol{w}_{s,i} \right), \tag{6.24}$$

where $\boldsymbol{w}_{s,i}$ are suitably defined secondary task variables and $\boldsymbol{J}_{s,i}$ are the corresponding Jacobians. Both β and α_i's are tuned according to the state of the system and to given behavioral rules. The inputs of the fuzzy inference system depend on the variables of interest in the specific mission. As an example, the end-effector error, the ocean current measure, the system's dexterity, the force sensor readings, can be easily taken into account by setting up a suitable set of fuzzy rules.

To avoid the exponential growth of the fuzzy rules to be implemented as the number of tasks is increased, the secondary tasks are suitably organized in a hierarchy.

Table 6.2 D-H parameters [m, rad] of the manipulator mounted on the underwater vehicle

	a	d	θ	α
link 1	0.150	0	q_1	$-\pi/2$
link 2	0.610	0	q_2	0
link 3	0.110	0	q_3	$-\pi/2$
link 4	0	0.610	q_4	$\pi/2$
link 5	0	-0.113	q_5	$-\pi/2$
link 6	0	0.103	q_6	0

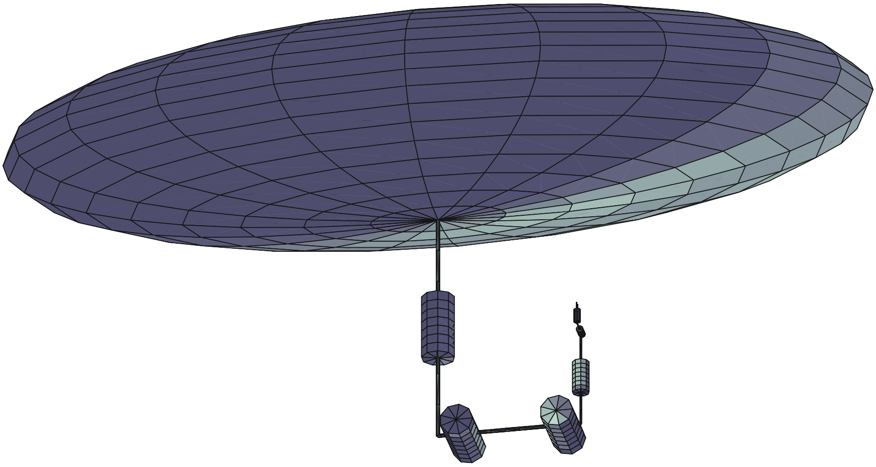


Fig. 6.11 UVMS in the configuration with null joint positions

Also, the rules have to guarantee that only one α_i is high at a time to avoid conflict between the secondary tasks. An example of application of the approach is described in the Simulation Section.

Simulations

The proposed fuzzy technique has been verified in full-DOFs case studies. An UVMS has been considered constituted by a vehicle with the size of the NPS Phoenix [59] and a manipulator mounted on the bottom of the vehicle. The kinematics of the manipulator considered is that of the SMART-3S manufactured by COMAU. Its Denavit-Hartenberg parameters are given in Table 6.2. The overall system, thus, has 12 DOFs. Figure 6.11 shows the configuration in which all the joint positions are zero according to the used convention.

The simulations are aimed at proving the effectiveness of the fuzzy kinematic control approach; for seek of clarity, thus, only the kinematic loop performance is shown (see Fig. 9.2 in Chap. 9). The real vehicle/joint position will be affected by a larger error since the tracking error too has to be taken into account. It is worth noticing that, as long as the law level dynamic controller is suitably designed, this tracking error is bounded. Moreover, it does not affects the kinematic loop performance.

The primary task is to track a position/orientation trajectory of the end effector. The system starts from the initial configuration:

$$\begin{aligned} \boldsymbol{\eta} &= \begin{bmatrix} 0\,0\,0\,0\,0\,0 \end{bmatrix}^{\mathrm{T}} \quad \text{m, deg} \\ \boldsymbol{q} &= \begin{bmatrix} 0\, -\, 30\, -\, 110\,0\, -\, 40\,90 \end{bmatrix}^{\mathrm{T}} \quad \text{deg} \end{aligned}$$

that corresponds to the end-effector position/orientation

$$\begin{aligned} \boldsymbol{\eta}_{ee1} &= \begin{bmatrix} 0.99 \; -0.11\,2.99 \end{bmatrix}^{\mathrm{T}} \quad \text{m} \\ \boldsymbol{\eta}_{ee2} &= \begin{bmatrix} 0\,0 \; -90 \end{bmatrix}^{\mathrm{T}} \quad \text{deg.} \end{aligned}$$

The end effector has to track a segment of −30 cm along z, stop there, and track a segment of 1 m along x. Both segments have to be executed with a quintic polynomial time law in 12 s. During the translation, the end effector orientation has to be kept constant. The initial configuration and the desired path are shown in Fig. 6.12. The duration of the simulation is 50 s. The algorithm is implemented at a sampling frequency of 20 Hz. Notice that the desired path cannot be tracked by the manipulator alone since it goes outside of its workspace. It is then necessary, somehow, to move the vehicle as well. Finally, the task is to be executed in real-time; no off-line knowledge of the task is available.

Different simulations will be shown:

Case Study n. 1. Simple pseudoinversion of (2.73);
Case Study n. 2. Pseudoinversion of (2.73) by the use of a weighted pseudoinverse;
Case Study n. 3. Singularity robust task priority algorithm;
Case Study n. 4. Integration of the former algorithm with the proposed fuzzy technique in presence of several secondary tasks.

For all the simulations a CLIK algorithm is considered, moreover, the orientation error is represented by the use of quaternions. The desired orientation, however, is still assigned in terms of Euler angles, since the transformation from Euler angles to quaternions is free from representation singularities.

Case Study n. 1

The first simulation has been run by the use of a simple pseudoinversion of (2.73) with CLIK gain:

$$\boldsymbol{K}_E = \text{blockdiag}\{1.6\boldsymbol{I}_3, \boldsymbol{I}_3\}\,.$$

This solution is not satisfactory in presence of such a large number of degrees of freedom. The system, in fact, is not taking into account the different nature of the degrees of freedom (vehicle and manipulator) leading to evident drawbacks: large movement of the vehicle in position and orientation, final configuration not suitable

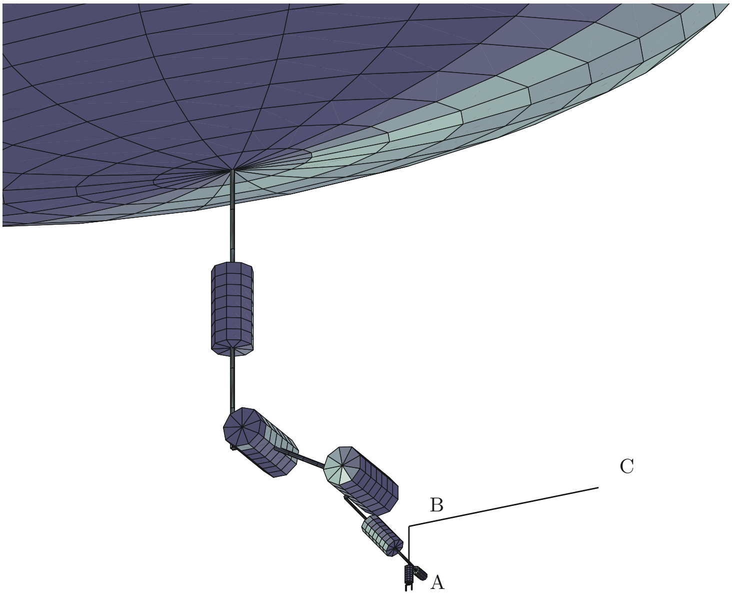


Fig. 6.12 Initial configuration of the UVMS for all the case studies. Desired end-effector position: at the start time (*A*); after the first movement of 12 s (*B*); at the final time (*C*)

for sensor tuning, possible occurrence of kinematic singularities or joint mechanical limits. As an example, Fig. 6.13 reports the sketch of the final configuration where the bottom sonar would not work properly since the pitch is ≈18 deg.

Case Study n. 2

In the second case study a weighted pseudoinverse is added in order to redistribute the motion between vehicle and manipulator including a cost factor that can be considered, e.g., proportional to the ratio of their inertias. The following matrix of gain has been used:

$$\boldsymbol{W}^{-1} = \text{blockdiag}\{0.01\boldsymbol{I}_6, \boldsymbol{I}_6\}\,.$$

Despite the much different costs of the two movements, the vehicle is still required to move in order to contribute to the end effector motion. However it would be preferable to move the vehicle only when absolutely necessary, leading to sole movement of the manipulator in ordinary working conditions. As an example, the vehicle attitude, in terms of Euler angles, for the last simulation is shown in Fig. 6.14. The vehicle still has a pitch of about 20 deg.

Case Study n. 3

The drawback shown by the algorithm as presented in the Case Study n. 2 can be easily avoided by resorting to a singularity-robust task priority redundancy resolution [38]. The same task is now simulated with the introduction of the secondary task:

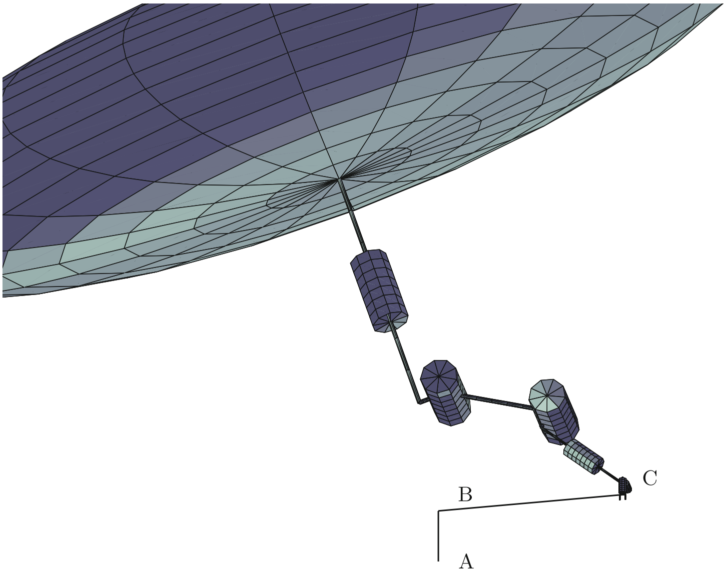
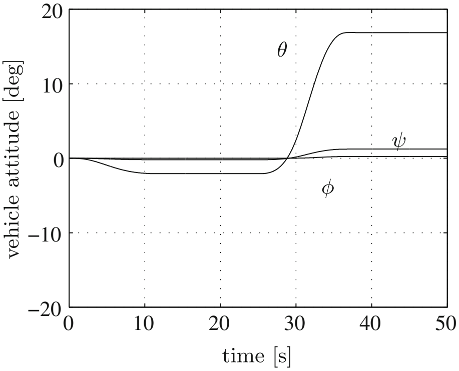


Fig. 6.13 Final configuration of the UVMS for the first case study. The redundancy is not exploited and the possible occurrence of undesired configurations is not avoided

Fig. 6.14 Case study n. 2. Vehicle attitude in terms of Euler angles. Despite the weight factor, the vehicle can reach non-dexterous configurations



$$\boldsymbol{x}_s = \begin{bmatrix} \phi \\ \theta \end{bmatrix}$$

with $\boldsymbol{x}_{s,d} = \begin{bmatrix} 0 \ 0 \end{bmatrix}^{\mathrm{T}}$, meaning that the vehicle has to maintain an horizontal configuration all along the task execution. Its Jacobian is given in Sect. 6.2.1. Notice that, for this simple matrix, it is $\boldsymbol{J}_s^{\dagger} = \boldsymbol{J}_s^{\mathrm{T}}$. The vehicle position and yaw are not constrained.

Figure 6.15 reports the sketch of the final configuration; it can be observed that the pitch is now close to zero, as can be seen also from Fig. 6.16.

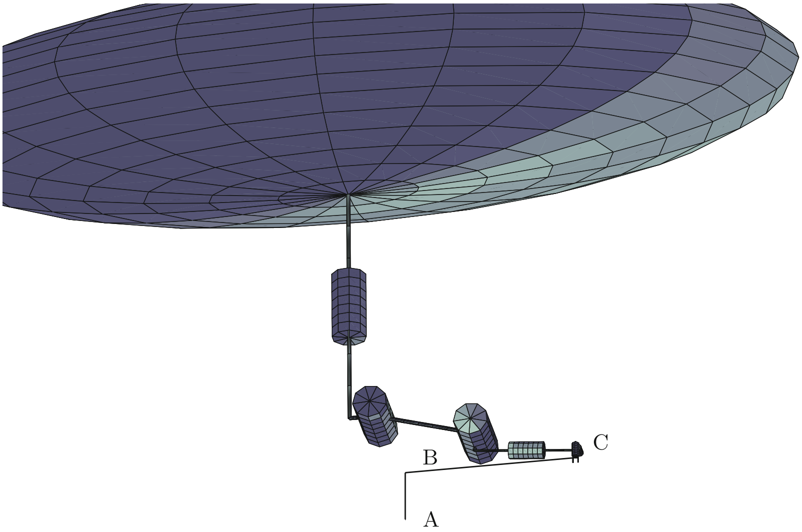
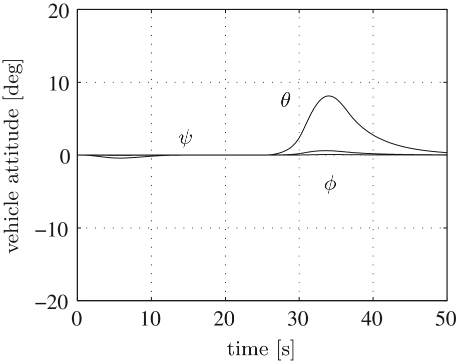


Fig. 6.15 Case study n. 3. Final configuration; the roll and pitch angles are now kept close to zero by exploiting the redundancy with the singularity-robust task priority algorithm

Fig. 6.16 Case study n. 3. Vehicle attitude in terms of Euler angles



Case Study n. 4

In this simulation, the implementation of the proposed kinematic control approach is presented. In an UVMS several variables are of interest in order to achieve a successful mission:

- avoidance of kinematic singularities;
- keeping the joints far from the mechanical limits;
- keeping the vehicle with small roll and pitch;
- avoidance of obstacles;
- alignment of the vehicle fore-aft direction with the ocean current.

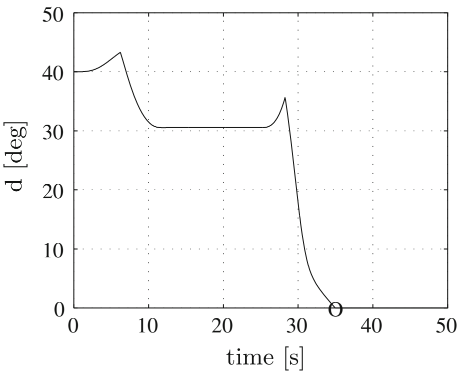


Fig. 6.17 Case study n. 3. Minimum distance of the 6 joints from their mechanical limits. It can be observed that large movements of the manipulator may cause hitting of the joint limits

It is underlined that some of the above items are critical; the alignment with the current, however, can be significant in order to reduce power consumption [8, 61]. As an example, in the previous case study, the use of a weight factor requires now a larger movement of the manipulator. This can lead to the occurrence of kinematic singularities. In fact, if the trajectory is assigned in real-time, it is possible that the manipulator is asked to move to the border of its workspace, where the possibility to experience a kinematic singularity is high. Also, when the manipulator is outstretched, mechanical joint limits can be encountered. In the simulations, the following joint limits have been assumed:

$$\begin{aligned}\boldsymbol{q}_{\min} &= [-100\ \ -210\ \ -210\ \ -150\ \ -80\ \ -170]^{\mathrm{T}}\mathrm{deg}\\ \boldsymbol{q}_{\max} &= [100\quad\ 30\qquad 10\quad\ 150\quad 80\quad 170]^{\mathrm{T}}\mathrm{deg}\,.\end{aligned}$$

In Fig. 6.17, the minimum distance to such limits for the previous case study is shown. It can be observed that the manipulator hits a mechanical limit at $t \approx 35\,\mathrm{s}$.

In order to match all the constraints of such problem, a solution as been proposed based on the use of the singularity-robust task priority merged with fuzzy techniques.

Let consider the following tasks:

- **End-effector position/orientation**. The primary task is given, as for the previous simulations by the end-effector position and orientation. The corresponding Jacobian $\boldsymbol{J}_p$ is $\boldsymbol{J}$ given in (2.73);
- **Manipulability**. Since the fuzzy approach tries to move the manipulator alone, its manipulability has to be checked. A computationally limited measure of the manipulability can be obtained by checking the minimum singular value of the Jacobian [2, 65]. Since the manipulability function is strongly non-linear it is possible to adopt the following approach: when close to a singular configuration, the system tries to reconfigure itself in a dexterous configuration. The task, thus, is a nominal manipulator configuration whose Jacobian is given by:

$$\boldsymbol{J}_{s1} = \left[\boldsymbol{O}_{6\times6}\,\boldsymbol{I}_6\right];$$

- **Mechanical limits**. Due to the mechanical structure, each joint has a limited allowed range. In case of a real-time trajectory, avoidance of such limits is crucial. For this reason the minimum distance from a mechanical limit is considered as another secondary task. Notice that the Jacobian is the same of to the previous task:
$$\boldsymbol{J}_{s2} = \left[\boldsymbol{O}_{6\times 6}\boldsymbol{I}_6\right];$$
- **Vehicle attitude**. As for the previous cases, the vehicle attitude (roll and pitch angles) has to be kept null when possible. The Jacobian is given in Sect. 6.2.1.

Due to the simple structure of the matrices, the pseudoinversion of the secondary tasks is trivial: $\boldsymbol{J}_{s1}^{\dagger} = \boldsymbol{J}_{s1}^{\mathrm{T}}, \boldsymbol{J}_{s2}^{\dagger} = \boldsymbol{J}_{s2}^{\mathrm{T}}, \boldsymbol{J}_{s3}^{\dagger} = \boldsymbol{J}_{s3}^{\mathrm{T}}$.

As shown in Sect. 6.6, the above tasks are activated by fuzzy variables. The fuzzy inference system has 3 inputs, namely: a measure of the robot manipulability, a measure of the distance from the joints limits and a measure of the vehicle attitude. Hence, 3 linguistic variables can be defined that can take the values:

$$\begin{aligned}
\textit{manipulability} &= \{\text{singular, notsingular}\}\\
\textit{joint limits} &= \{\text{close, notclose}\}\\
\textit{vehicle attitude} &= \{\text{small, notsmall}\}
\end{aligned}$$

The output is given by the linguistic variables β and the 3 α_i's. The latter can take the following values:

$$\alpha_i = \{\text{high, low}\}.$$

The linguistic variable β, named *motion* can take the following values:

$$\textit{motion} = \{\text{vehicle, manipulator}\}.$$

As an example, the membership function of the linguistic variable *joint limits* is reported in Fig. 6.18.

The FIS outputs are considered at two different levels. The variable *motion* (β in (6.23)) can be considered at a higher level with respect to the variables α_i's. Roughly speaking, the motion has to be normally assigned to the manipulator, i.e., if all the α_i's are null. Hence, the value of β is related to the value of the α_i's and the first set of rules to be designed concerns the 3 linguistic variables α_1, α_2 and α_3. A complete and consistent set of fuzzy rules for 3 linguistic variables each of them defined in two fuzzy sets requires 2^3 rules for every output leading to 32 total rules. The rules have been arranged in a hierarchical structure that gives higher priority to the kinematic singularity of the manipulator and lower to the vehicle attitude. Hence, the following 8 rules have been used:

1. `if (manipulator is singular) then (`α_1` is high);`
2. `if (manipulator is not singular) then (`α_1` is low);`

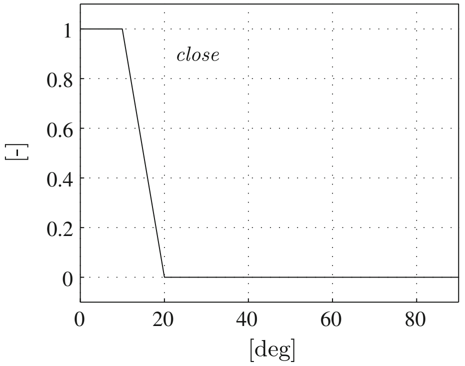


Fig. 6.18 Membership function of the linguistic variable: *joint limits* used in the fuzzy singularity-robust task priority technique

3. if (manipulator is singular) then (α_2 is low);
4. if (joint limits is not close) then (α_2 is low);
5. if (manipulator is not singular) and (joint limits is close)
   then (α_2 is high);
6. if (manipulator is singular) or (joint limits is close)
   then (α_3 is low);
7. if (vehicle attitude is small) then (α_3 is low);
8. if (manipulator is not singular)
   and (joint limits is not close)
   and (vehicle attitude is not small) then (α_3 is high).

In detail, the rules are developed as follows:

- The first two rules concern the primary task. In this case, the manipulator singularity is of concern and the variable α_1 is activated when the manipulator is close to a singularity.
- A second task (α_2), with lower priority with respect to the first, has to be added. Rule n. 3, thus, is aimed at avoiding activation of this task when the primary (α_1) is high.
- Rule n. 4 and n. 5 are aimed at activating α_2. Notice that the activation of α_2 is in and with the condition that does not activate the higher priority task (α_1).
- Repeat for the third task, in order of priority, the same rules as done for the second task by taking into account that two tasks are now of higher priority.

Table 6.3 is aimed at clarifying the rules development with respect to two tasks, 1 and 2 in which the first is of higher priority with respect to the second. The fuzzy sets are very simple, i.e., an input u_i high requires the activation of this task by imposing α_i high. It can be recognized, thus, that α_2 respects its lower priority.

It is worth noticing that the rules presented could be grouped, e.g., rules n. 1 and n. 3. The list presented, however, keeps the logical structure used to develop the

Table 6.3 Examples of the fuzzy set rules for two tasks

	$u_1 = \text{low}$	$u_1 = \text{high}$
$u_2 = \text{low}$	$\alpha_1 = \text{low}$	$\alpha_1 = \text{high}$
	$\alpha_2 = \text{low}$	$\alpha_2 = \text{low}$
$u_2 = \text{high}$	$\alpha_1 = \text{low}$	$\alpha_1 = \text{high}$
	$\alpha_2 = \text{high}$	$\alpha_2 = \text{low}$

u_1 is the input of a generic task of higher priority with respect to u_2 corresponding to a secondary task. α_1 and α_2 are the corresponding output

rules and should be clearer to the reader. Obviously, in the simulation the rules have been compacted. With this logical approach the rules are complete, consistent and continuous [64].

The `and–or` operations have been calculated by resorting to the *min–max* operations respectively, the implication–aggregation operations too have been calculated by resorting to the *min–max* operations respectively, the values of $\alpha_i \in \ [0, 1]$ are obtained by defuzzification using the centroid technique and a normalization. Finally, the value of $\beta \in \ [0, 1]$ is given by $\beta = 1 - \max_i(\alpha_i)$. Notice that the extremities of the range in which β is defined do not involve a singular configuration

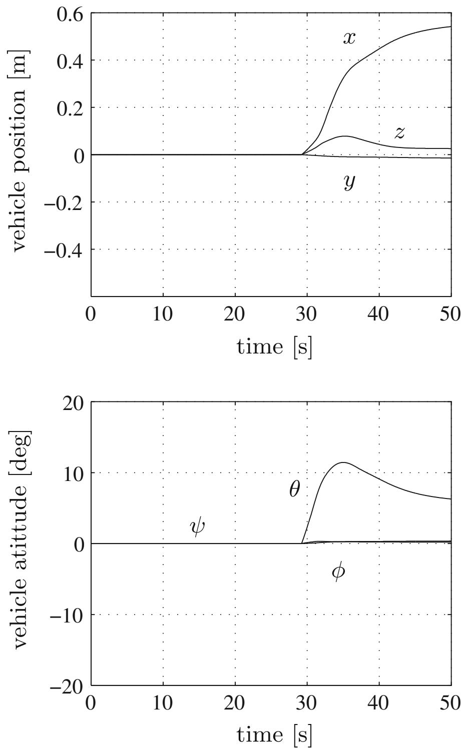


Fig. 6.19 Case study n. 4. Vehicle position (*top*) and attitude in terms of Euler angles (*bottom*). The movement of the vehicle is not required in the execution of the first segment (A-B, first 12 s) when the manipulator is working in dexterous configuration

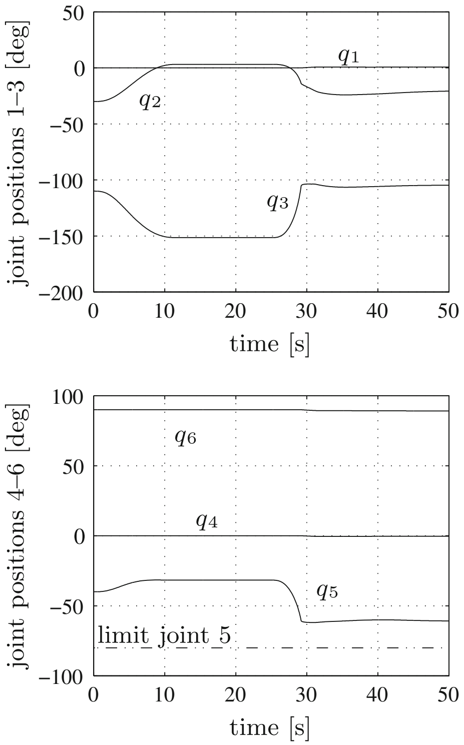


Fig. 6.20 Case study n. 4. Joint positions. The mechanical limit of joint 5 is highlighted; it can be observed that the system reconfigures itself in order to avoid working close to the mechanical limit

since, if $\beta = 1$ the manipulator alone is moving and it is not close to a kinematic singularity. On the other hand, it is preferable to have a certain degree of mobility of the manipulator avoiding $\beta = 0$; this to guarantee that the manipulator reconfigures itself in a dexterous posture.

A simulation has been run with the proposed kinematic control leading to satisfactory results. Figures 6.19, 6.20, 6.21, 6.22, 6.23 and 6.24 show some plots of interest. In detail, Fig. 6.19 shows the vehicle position and attitude, Fig. 6.20 the joint positions, Fig. 6.21 reports the variables considered as secondary tasks and the corresponding FIS outputs. It can be observed that, in the execution of the segment A-B, the vehicle is not requested to move since the manipulator is working in a safe posture; this can be observed from Fig. 6.21 where it can be noticed that the α_i's are null in the first part of the simulation. When $t \approx 30$ s joint 5 is approaching its mechanical limit and the corresponding α_2 is increasing, requesting the vehicle to contribute to the end-effector motion while the manipulator reconfigures itself; thus, by always keeping a null end-effector position/orientation error, the occurrence of an hit is avoided. The same can be observed for the pitch of the vehicle; since it is at the lower hierarchical level in the FIS, its value is recovered to the null value only when the other α_i's are null.

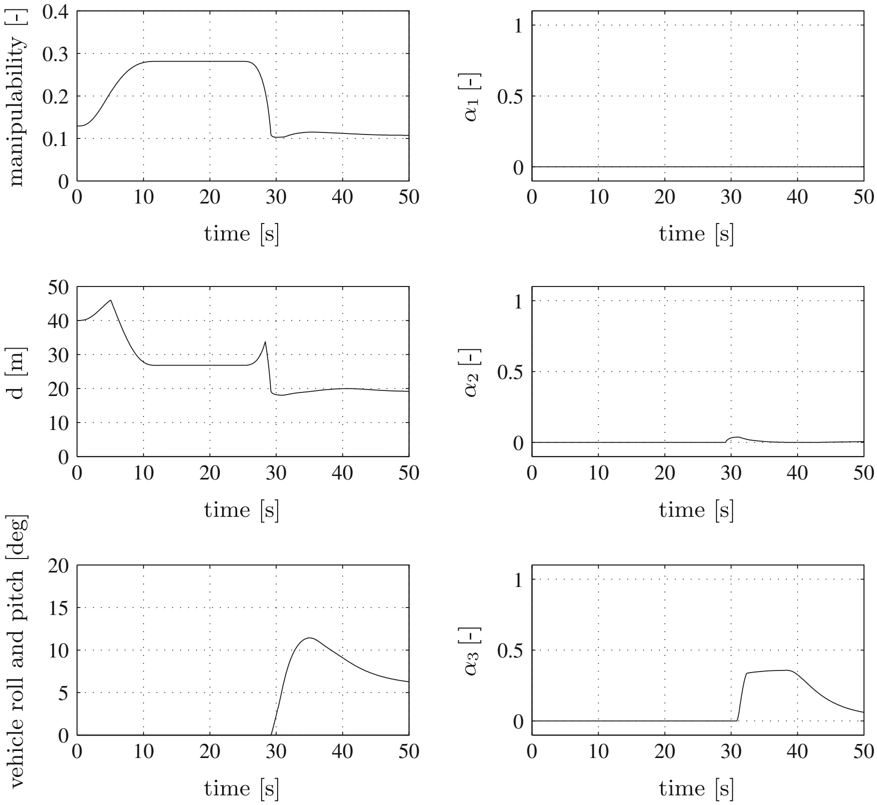


Fig. 6.21 Case study n. 4. Variables of interest for the secondary task (*left*) and output of the fuzzy inference system (*right*). For this specific mission, the manipulability task is not excited, the distance from the mechanical limit and the vehicle roll and pitch tasks are kept in their safe range

Figure 6.22 shows a sketch of the initial and final configuration of the system. Figure 6.23 shows the system velocities. It can be remarked that the proposed algorithm outputs smooth trajectories. As for all the simulations shown, the end-effector position/orientation error is practically null (Fig. 6.24) due to the use of a CLIK algorithm.

In order to show handling of the fuzzy rules under the proposed approach while avoiding exponential growth of their number, we finally add as 4th task, specification of the vehicle yaw. Our aim is to align the vehicle fore-aft direction with the current in order to get energetic benefit from the low drag of such configuration. To limit the number of rules to be implemented we assign to this task the last priority among the secondary tasks. In this case, considering two fuzzy sets also for this last variable ($yaw = \{$aligned, not aligned$\}$), only the following 3 rules have to be added to the previous 8, leading to 11 rules in total instead of 64:

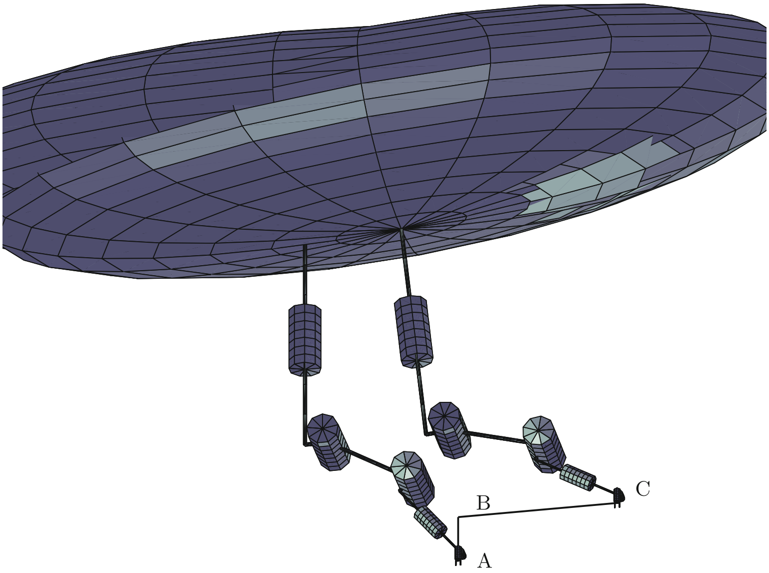


Fig. 6.22 Case study n. 4. Final configuration. The proposed kinematic control allows handling several variables of interest

```
 9. if (manipulator is singular) or (joint limits is
    close) or
    (vehicle attitude is not small) then ($\alpha_4$ is low);
10. if (yaw is aligned) then ($\alpha_4$ is low);
11. if (manipulator is not singular)
    and (joint limits is not close)
    and (vehicle attitude is small)
    and (yaw is not aligned) then ($\alpha_4$ is high).
```

The 3 rules have the following aim: rule n. 9 is aimed at giving the lower priority to this specific task; rule n. 10 is aimed at guaranteeing that the output is always low when the corresponding input is inside the safe range; finally, rule n. 11 activates α_4 only for the given specific combination of inputs.

The integration of fuzzy technique with established inverse kinematic techniques exhibits promising results, the fuzzy theory can give an added value in handling complex situations as missions in remotely, unknown, hazardous underwater environments. In a certain way, a fuzzy approach could be considered to implement an higher level supervisor that is in charge of distributing the motion between vehicle and manipulator while taking into account the big amount of constraints of UVMSs: joint's limits, vehicle's roll and pitch, robot's manipulability, obstacle avoidance, etc.

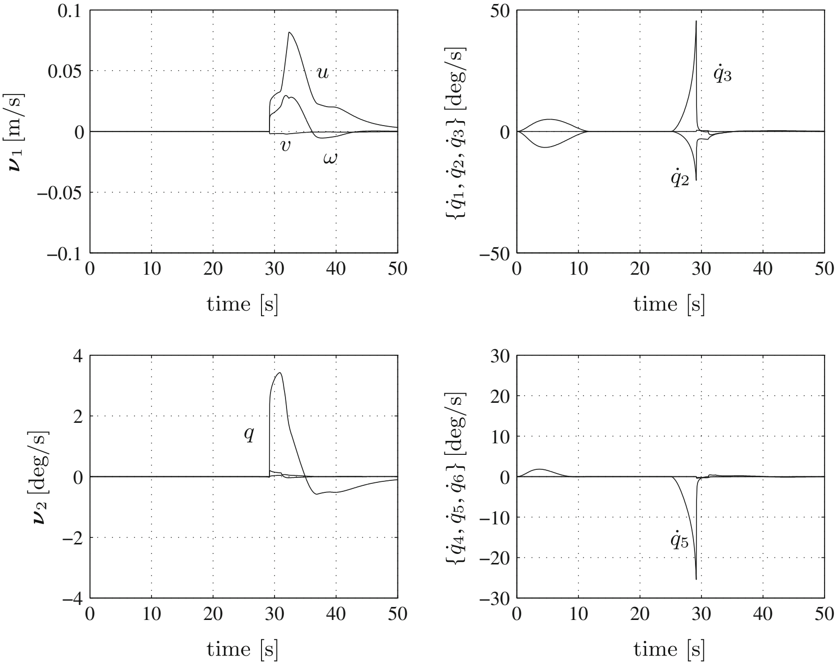
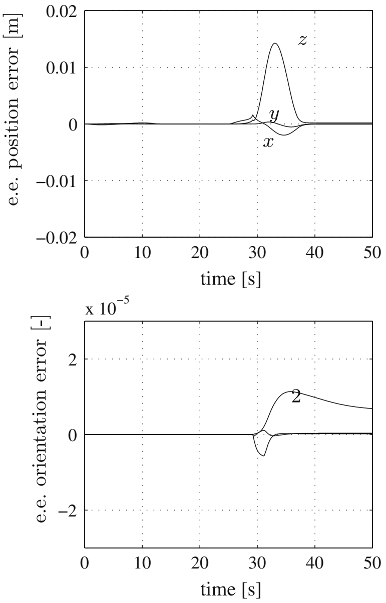


Fig. 6.23 Case study n. 4. System velocities

Fig. 6.24 Case study n. 4. End-effector position/orientation errors



6.7 Conclusions

With a view to implementing autonomous missions of robotic systems, kinematic control plays an important role. The manipulation task is naturally defined in the operational space. A mapping between the task space and the vehicle/joint space is then necessary to achieve the desired task.

If this mapping is implicitly performed via a model based dynamic control the natural redundancy of the system is not exploited, e.g., is not possible to take into account additional constraints. Moreover dynamic compensation of underwater robotic systems is difficult to obtain. On the other hand, off-line planning of the vehicle/joint positions is not advisable, since the mission has to be accomplished in an unstructured, generally unknown, environment. For these reasons real-time kinematic control seems to be the right approach to motion control of UVMSs.

The use of techniques well known in robotics such as the task priority approach seems to offer good results. They allow to reliably exploit the redundancy and do not require compensation of the system's dynamics. As it will be shown in Chap. 8, kinematic control techniques can be successfully integrated with interaction control schemes.

References

1. J.M. Hollerbach, Optimum kinematic design for a seven degree of freedom manipulator, in *Proceedings 2nd International Symposium on Robotics Research*, 1985
2. A. Maciejewski, C. Klein, Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. Int. J. Robot. Res. **4**(3), 109–117 (1985)
3. Y. Nakamura, H. Hanafusa, T. Yoshikawa, Task-priority based redundancy control of robot manipulators. Int. J. Robot. Res. **6**(2), 3–15 (1987)
4. I. Schjølberg, T. Fossen, Modelling and control of underwater vehicle-manipulator systems, in *Proceedings of 3rd Conferemce on Marine Craft maneuvering and control*, Southampton, UK, 1994, pp. 45–57
5. I. Schjølberg, Modeling and Control of Underwater Robotic Systems. PhD thesis, Doktor ingeniør degree, Norwegian University of Science and Technology, Trondheim, Norway, 1996
6. T.J. Tarn, S.P. Yang, Modeling and control for underwater robotic manipulators—an example, in *Proceedings of IEEE International Conference on Robotics and Automation, 1997*, vol. 3, IEEE, Albuquerque, NM, 1997, pp. 2166–2171
7. A.W. Quinn, D. Lane, Computational issues in motion planning for autonomous underwater vehicles with manipulators, in *Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology, 1994. AUV'94*, IEEE, 1994, pp. 255–262
8. G. Antonelli, S. Chiaverini, Task-priority redundancy resolution for underwater vehicle-manipulator systems, in *Proceedings 1998 IEEE International Conference on Robotics and Automation*, Leuven, B, May 1998, pp. 768–773
9. G. Antonelli, S. Chiaverini, Fuzzy inverse kinematics for underwater vehicle-manipulator systems, in *7th International Symposium on Advances in Robot Kinematics, Advances in Robot Kinematics*, ed. by J. Lenarčič, M.M. Stanišić, Piran-Portorož, SLO, June 2000, (NL Kluwer Academic Publishers, Dordrecht, 2000), pp. 249–256

10. G. Antonelli, S. Chiaverini, A fuzzy approach to redundancy resolution for underwater vehicle-manipulator systems, in *Proceedings 5th IFAC Conference on Manoeuvring and Control of Marine Craft*, Aalborg, Denmark, Aug 2000
11. G. Antonelli, S. Chiaverini, A fuzzy approach to redundancy resolution for underwater vehicle-manipulator systems. Control Eng. Pract. **11**(4), 445–452 (2003)
12. G. Antonelli, S. Chiaverini, Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems. IEEE Trans. Fuzzy Syst. **11**(1), 109–120 (2003)
13. S. Soylu, B.J. Buckham, R.P. Podhorodeski, Redundancy resolution for underwater mobile manipulators. Ocean Eng. **37**(2), 325–343 (2010)
14. N. Sarkar, T.K. Podder, Motion coordination of underwater vehicle-manipulator systems subject to drag optimization, in *Proceedings of IEEE International Conference on Robotics and Automation, 1999*, vol. 1, IEEE, 1999, pp. 387–392
15. N. Sarkar, T.K. Podder, Coordinated motion planning and control of autonomous underwater vehicle-manipulator systems subject to drag optimization. IEEE J. Oceanic Eng. **26**(2), 228–239 (2001)
16. T.K. Podder, N. Sarkar, Dynamic trajectory planning for autonomous underwater vehicle-manipulator systems, in *Proceedings of ICRA'00, IEEE International Conference on Robotics and Automation, 2000*, vol. 4, IEEE, 2000, pp. 3461–3466
17. N. Sarkar, J. Yuh, T.K. Podder, Adaptive control of underwater vehicle-manipulator systems subject to joint limits, in *Proceedings of 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999, IROS'99*, vol. 1, IEEE, 1999
18. B.H. Jun, J. Lee, P.M. Lee, Manipulability analysis of underwater robotic arms on ROV and application to task-oriented joint configuration, in *OCEANS'04, MTTS/IEEE TECHNO-OCEAN'04*, vol. 3, IEEE, 2004, pp. 1548–1553
19. S. Ishibashi, E. Shimizu, M. Ito, *The motion planning for underwater manipulators depend on genetic algorithm* (In IFAC World Congress, Barcelona, Spain, 2002)
20. G. Casalino, A. Turetta, A. Sorbara, Computationally distributed control and coordination architectures for underwater reconfigurable free-flying multi-manipulator, in *Workshop on Underwater Robotics, November 2005*, Genova, Italy, 2005
21. G. Casalino, A. Turetta, A computationally distributed self-organizing algorithm for the control of manipulators in the operational space, in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, ICRA 2005*, IEEE, 2005, pp. 4050–4055
22. G. Casalino, A. Turetta, Dynamic programming based, computationally distributed control of modular manipulators in the operational space, in *Mechatronics and Automation, 2005 IEEE International Conference*, 2005, pp. 1460–1467
23. A. Turetta, G. Casalino, A. Sorbara, Distributed control architecture for self-reconfigurable manipulators. Int. J. Robot. Res. **27**(3–4), 481–504 (2008)
24. G. Casalino, E. Zereik, E. Simetti, S. Torelli, A. Sperindé, A. Turetta, Agility for underwater floating manipulation task and subsystem priority based control strategy, in *International Conference on Intelligent Robots and Systems (IROS 2012)*, Sept 2012
25. T. Padir, Kinematic redundancy resolution for two cooperating underwater vehicles with on-board manipulators, in *IEEE International Conference on Systems, Man and Cybernetics, 2005*, vol. 4, 2005, pp. 3137–3142
26. T. Padir, J.D. Nolff, Manipulability and maneuverability ellipsoids for two cooperating underwater vehicles with on-board manipulators, in *IEEE International Conference on Systems, Man and Cybernetics, 2007, ISIC*, 2007, pp. 3656–3661
27. G. Marani, J. Yuh, S.K. Choi, Autonomous manipulation for an intervention AUV. IEE Control Eng. Ser. **69**, 217 (2006)
28. M. Hildebrandt, L. Christensen, J. Kerdels, J. Albiez, F. Kirchner, Realtime motion compensation for ROV-based tele-operated underwater manipulators, in *OCEANS 2009—EUROPE*, 2009, pp. 1–6
29. J. Han, W.K. Chung, Redundancy resolution for underwater vehicle-manipulator systems with minimizing restoring moments, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, IROS 2007*, San Diego, California, IEEE, 2007, pp. 3522–3527

30. J. Han, W.K. Chung, Coordinated motion control of underwater vehicle-manipulator system with minimizing restoring moments, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, IROS 2008*, IEEE, 2008, pp. 3158–3163
31. Z.H. Ismail, M.W. Dunnigan, Redundancy resolution for underwater vehicle-manipulator systems with congruent gravity and buoyancy loading optimization, in *IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009* Guilin, PRC, IEEE, 2009, pp. 1393–1399
32. J. Han, W.K. Chung, Active use of restoring moments for motion control of an underwater vehicle-manipulator system. IEEE J. Oceanic Eng. **32**, 300–312 (2013)
33. B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer, London, 2009)
34. D.E. Whitney, Resolved motion rate control of manipulators and human prostheses. IEEE Trans. Man Mach. Syst. **10**(2), 47–53 (1969)
35. O. Egeland, Task-space tracking with redundant manipulators. IEEE J. Robot. Autom. **3**(5), 471–475 (1987)
36. A. Liégeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans. Syst. Man Cybern. B Cybern. **7**, 868–871 (1977)
37. B. Siciliano, J.-J.E. Slotine, A general framework for managing multiple tasks in highly redundant robotic systems, in *Proceedings 5th International Conference on Advanced Robotics*, Pisa, I, 1991, pp. 1211–1216
38. S. Chiaverini, Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Trans. Robot. Autom. **13**(3), 398–410 (1997)
39. G. Antonelli, Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. IEEE Trans. Robot. **25**(5), 985–994 (October 2009)
40. Y. Nakamura, H. Hanafusa, Inverse kinematic solutions with singularity robustness for robot manipulator control. ASME Trans. J. Dyn. Syst. Meas. Contr. **108**, 163–171 (1986)
41. A.A. Maciejewski, Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. J. Robot. Syst. **5**(6), 527–552 (1988)
42. R.E. Roberson, R. Schwertassek, *Dynamics of Multibody Systems*, vol. 18 (Springer, Berlin, 1988)
43. F. Caccavale, C. Natale, B. Siciliano, L. Villani, Resolved-acceleration control of robot manipulators: a critical review with experiments. Robotica **16**(5), 565–573 (1998)
44. S. Chiaverini, B. Siciliano, The unit quaternion: a useful tool for inverse kinematics of robot manipulators. Syst. Anal. Modell. Simul. **35**(1), 45–60 (1999)
45. G. Arleo, F. Caccavale, G. Muscio, F. Pierri, Control of quadrotor aerial vehicles equipped with a robotic arm, in *21st Mediterranean Conference on Control and Automation*, Crete, GR, Jun 2013
46. N. Mansard, F. Chaumette, Task sequencing for high-level sensor-based control. IEEE Trans. Robot. Autom. **23**(1), 60–72 (2007)
47. J. Lee, N. Mansard, J. Park, Intermediate desired value approach for task transition of robots in kinematic control. Int. J. Robot. Res. **25**, 575-591 (2012)
48. S. Chiaverini, G. Oriolo, I.D. Walker, in *Kinematically Redundant Manipulators* , ed. by B. Siciliano, O. Khatib. Springer Handbook of Robotics, (Springer-Verlag, Heidelberg, 2008), pp. 245–268
49. G. Antonelli, F. Arrichiello, S. Chiaverini, The Null-Space-based Behavioral control for autonomous robotic systems. J. Intel. Serv. Robot. **1**(1), 27–39 (2008) (online March 2007, printed January 2008)
50. F. Arrichiello, S. Chiaverini, G. Indiveri, P. Pedone, The null-space based behavioral control for mobile robots with velocity actuator saturations. Int. J. Robot. Res. **29**(10), 1317–1337 (Sept. 2010)
51. G. Antonelli, G. Indiveri, S. Chiaverini, Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems with velocity saturations, *Proceedings 2009 IEEE/RSJ International Conference on Intelligent RObots and Systems*, St. Louis, MO, USA, Oct 2009, pp. 5892–5897

52. F. Flacco, A. De Luca, O. Khatib, Motion control of redundant robots under joint constraints: Saturation in the null space, in *IEEE International Conference on Robotics and Automation (ICRA), 2012*, IEEE, 2012, pp. 285–292
53. O. Khatib, A unified approach for motion and force control of robot manipulators: the operational space formulation. IEEE J. Robot. Autom. **3**(1), 43–53 (1987)
54. L. Sentis, Synthesis and Control of Whole-Body Behaviors in Humanoid Systems. Ph.D. thesis, Stanford University, 2007
55. J. Nakanishi, R. Cory, M. Mistry, J. Peters, S. Schaal, Comparative experiments on task space control with redundancy resolution, *Proceedings 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, CA, Aug 2005, pp. 3901–3908
56. J. Nakanishi, R. Cory, M. Mistry, J. Peters, S. Schaal, Operational space control: a theoretical and empirical comparison. Int. J. Robot. Res. **27**(6), 737–757 (2008)
57. B.H. Jun, J. Lee, P.M. Lee, A repetitive periodic motion planning of articulated underwater robots subject to drag optimization, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, (IROS 2005)*, IEEE, 2005, pp. 1917–1922
58. A. Ben-Israel, T. Greville, *Generalized Inverses: Theory and Applications*, vol. 15 (Springer, New York, 2003)
59. A. Healey, D. Lienard, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J. Oceanic Eng. **18**(3), 327–339 (1993)
60. T. Fossen, *Guidance and Control of Ocean Vehicles* (Wiley, Chichester, 1994)
61. S. Mohan, K. Jinwhan, K. Yonghyun, A null space control of an underactuated underwater vehicle-manipulator system under ocean currents, in *OCEANS, 2012—Yeosu*, 2012, pp. 1–5
62. T. Yoshikawa, Manipulability of robotic mechanisms. Int. J. Robot. Res. **4**(2), 3–9 (1985)
63. G. Antonelli, S. Chiaverini, Singularity-free regulation of underwater vehicle-manipulator systems, in *Proceedings 1998 American Control Conference*, PA, Philadelphia, Jun 1998, pp. 399–403
64. D. Driankov, H. Hellendoorn, M. Reinfrank, *An Introduction to Fuzzy Control* (Springer, Heidelberg, 1996)
65. S. Chiaverini, Estimate of the two smallest singular values of the Jacobian matrix: application to damped least-squares inverse kinematics. J. Robot. Syst. **10**(8), 991–1008 (1993)

Chapter 7
Dynamic Control of UVMSs

7.1 Introduction

This chapter addresses the dynamic control of UVMSs. Assuming that the control problem is decomposed according to the scheme in Fig. 7.1, the matching of the scheme in Fig. 6.1 with a simple switch of the orange color, the dynamic controller receives as input the desired vehicle and manipulator positions and output the generalized forces to be sent to the motors.

In Chap. 2 the equations of motion of Underwater Vehicle-Manipulator Systems (UVMSs) have been presented. Their expression in matrix form in Eq. (2.77) is formally close to the equations of motion of ground fixed manipulators for which a wide control literature exists. This has suggested a suitable translation/implementation of existing control algorithms. However, some differences, crucial from the control aspect, need to be underlined. UVMSs are complex systems characterized by several strong constraints:

- Uncertainty in the model knowledge, mainly due to the poor knowledge of the hydrodynamic effects;
- Complexity of the mathematical model;
- Kinematic redundancy of the system;
- Difficulty to control the vehicle in hovering, mainly due to the poor thrusters performance;
- Dynamic coupling between vehicle and manipulator;
- Low bandwidth of the sensor's readings.

In [1, 2] a discrete adaptive control strategy for coordinated control of UVMSs is presented. Numerical simulations, on a planar task, show that the use of a centralized controller, better than two separate controllers, one for the vehicle and one for the manipulator, guarantees performance improvement.

Reference [3] shows an adaptive macro-micro control for UVMSs. Inverse kinematics is obtained by inversion of the Jacobian matrix; hence, a manipulator with 6 degrees of freedom is required. A stability analysis in Lyapunov sense is provided.

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_7,

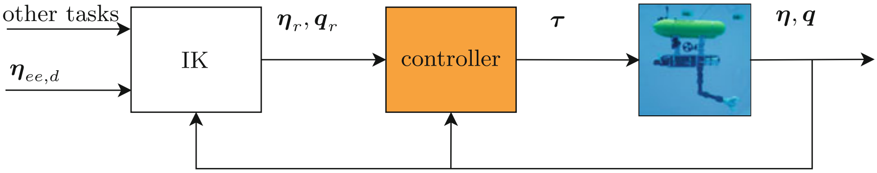


Fig. 7.1 Block scheme of the dynamic controller (in *orange*), affoarded in this chapter

The use of multiple manipulators to be used as stabilizing paddles is investigated by means of simulations in [4]. Those concern a vehicle carrying a 6-DOF manipulator plus a pair of 2-link manipulators counteracting the interaction force between vehicle and manipulator as paddles.

By means of moving loads, in [5, 6] the vehicle roll and pitch are controlled despite the manipulator movments. Interesting pool experiments are provided.

In [7, 8] some dynamic considerations are given to underline the existence of a dynamic coupling between vehicle and manipulator. From this analysis, based on a specific structure of an UVMS, a Sliding Mode approach with a feedforward compensation term is presented. Numerical simulation results show that the knowledge of the dynamics allows improvement of the tracking performance.

In [9], during the manipulation motion, the vehicle is assumed to be inactive and modeled as a passive joint. A robust controller, with a disturbance observer-based action is used and its effectiveness verified in a 1-DOF vehicle carrying a 3-DOF manipulator.

Reference [10] presents a sliding mode controller that benefits from the compensation of a multilayer neural network. Simulation with a 2-DOF, ground-fixed, underwater manipulator is provided. Neuro-fuzzy for underwater manipulation is also presented in [11] where a 3-link test bed of a manipulator with a fixed based, i.e., without vehicle, is proposed.

In [12, 13] the possibility to mount a force/torque sensor at the base of the manipulator is considered in order to compensate for the vehicle/manipulator dynamic interaction. In case of absence of the sensor, [13] also proposes a disturbance observer.

Reference [14] presents an iterative learning control experimentally validated on a 3-DOF manipulator with fixed base. This is first moved in air and then in water in order to *learn* the hydrodynamic dynamic contribution and then use it in a feedforward compensation.

Reference [15], after having reported some interesting dynamic considerations about the interaction between the vehicle and the manipulator, propose a two-time scale control. The vehicle, characterized by low bandwidth actuators that can not compensate for the high manipulator bandwidth, is controlled by a simple P-type action, while the manipulator is controlled by a feedback linearizing controller.

In [16, 17] the model of a planar motion of the AUV Twin-Burger equipped with a 2-link manipulator is developed. A Resolved Acceleration Control is then applied and simulated on the 5-DOFs, some preliminary experiments are also reported.

Reference [18] proposes an adaptive action mainly based on the transpose of the Jacobian. The approach is validated on simulations involving a 2-DOF model of ODIN carrying a 2-DOF planar manipulator.

A problem slightly different is approached in [19], where the 4-DOF manipulator *Sherpa*, mounted under the ROV Victor 6,000 developed at the Ifremer, is considered. This manipulator has been originally deigned to be controlled in open-loop by a remote operator via a master/slave configuration using a joystick: it is, thus, not-provided with proprioceptive sensors. The Authors propose closed-loop system based on an eye-to-hand visual servoing approach to control its displacement.

Concerning systems experimentally tested in full-DOFs surveys, it is worth noticing the UVMS described in [20, 21] and shown in Fig. 1.8: the manipulator is controlled independently from the vehicle with a joint-based dynamic control law.

7.2 Feedforward Decoupling Control

In 1996 McLain, Rock and Lee, in [22, 23], present a control law for UVMSs with some interesting experimental results conducted at the Monterey Bay Aquarium Research Institute (MBARI). A 1-link manipulator is mounted on the vehicle OTTER (see Fig. 7.2) controlled in all the 6-DOFs by mean of 8 thrusters. A coordinated control is then implemented to improve the tracking error of the end effector.

In order to explain the control implemented let rewrite the equations of motion for the sole vehicle in matrix form as

$$\boldsymbol{M}_v\dot{\boldsymbol{\nu}} + \boldsymbol{C}_v(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}_{RB}(\boldsymbol{R}_B^I) = \boldsymbol{\tau}_v - \boldsymbol{\tau}_m(\boldsymbol{R}_B^I, \boldsymbol{q}, \zeta, \dot{\zeta}) \tag{7.1}$$

where $\boldsymbol{\tau}_m(\boldsymbol{R}_B^I, \boldsymbol{q}, \zeta, \dot{\zeta}) \in \mathbb{R}^6$ represents the coupling effect caused by the presence of the manipulator. The control of the vehicle and of the arm is achieved, independently, by classical control technique. The coordination action is obtained by adding to the vehicle thrusters a feedforward compensation term that is an estimate of $\boldsymbol{\tau}_m$:

$$\boldsymbol{\tau}_v = \boldsymbol{\tau}_{control} + \hat{\boldsymbol{\tau}}_m(\boldsymbol{R}_B^I, \boldsymbol{q}, \zeta, \dot{\zeta}) \tag{7.2}$$

where $\boldsymbol{\tau}_{control} \in \mathbb{R}^6$ is the control action output by the sole vehicle controller.

In [23] experimental results conducted with the vehicle OTTER are reported. This is about 2.5 m long, 0.95 m wide, and 0.45 m tall and weights about 145 kg in air. The arm used has a 7.1 cm diameter and is 1 m long. The control benefits from a variety of commercial sensors: an acoustic short-baseline for the horizontal position, a pressure transducer for the depth, a dual-axis inclinometer for the roll and pitch angles, a flux-gate compass for the yaw, and solid-state gyros for the angular velocities. The control loop has been implemented at a frequency of 100 Hz, i.e., the frequency of all the sensors expect the baseline acoustic that worked at 2.5 Hz.

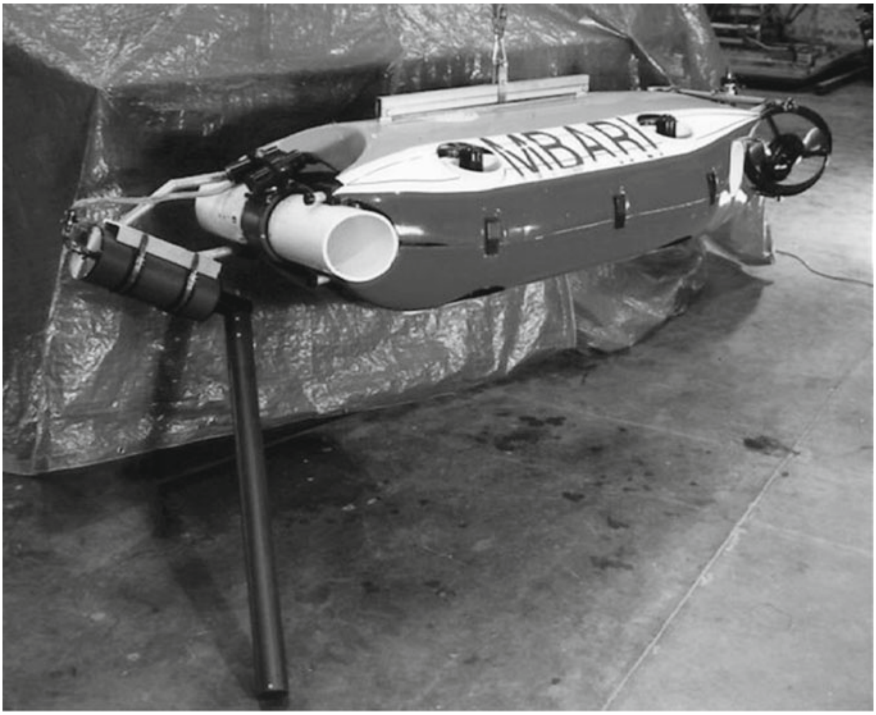


Fig. 7.2 OTTER vehicle developed in the Aerospace Robotics Laboratory at Stanford University (courtesy of T. McLain, Brigham Young University)

The experiment is conducted under some assumptions: the vehicle does not influence the manipulator dynamic due to its small movements; the desired joint acceleration are used instead of the measured/filtered ones; there is no current and the lift forces are small compared to the in-line forces. For this specific experiment, moreover, the control of the vehicle is obtained by resorting to PID-like actions at the 6 DOFs independently. In view of these assumptions it is:

$$\boldsymbol{\tau}_v = \boldsymbol{\tau}_{PID} + \hat{\boldsymbol{\tau}}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_d) \, .$$

Figure 7.3 reports some experimental results of the end-effector error commanded to a periodic motion. In the top-left plot the vehicle is not controlled at all; the influence of the arm on the vehicle can be observed. In the top-right and the bottom-left plots the sole decoupling force and vehicle feedback are considered. Finally, in the bottom-right plot the benefit of considering the proposed control strategy can be fully appreciated.

In Fig. 7.4 the end-effector step response is given and its settling time can be observed. It can be remarked that, without the proposed decoupling strategy, the end effector does not even reaches the given set point.

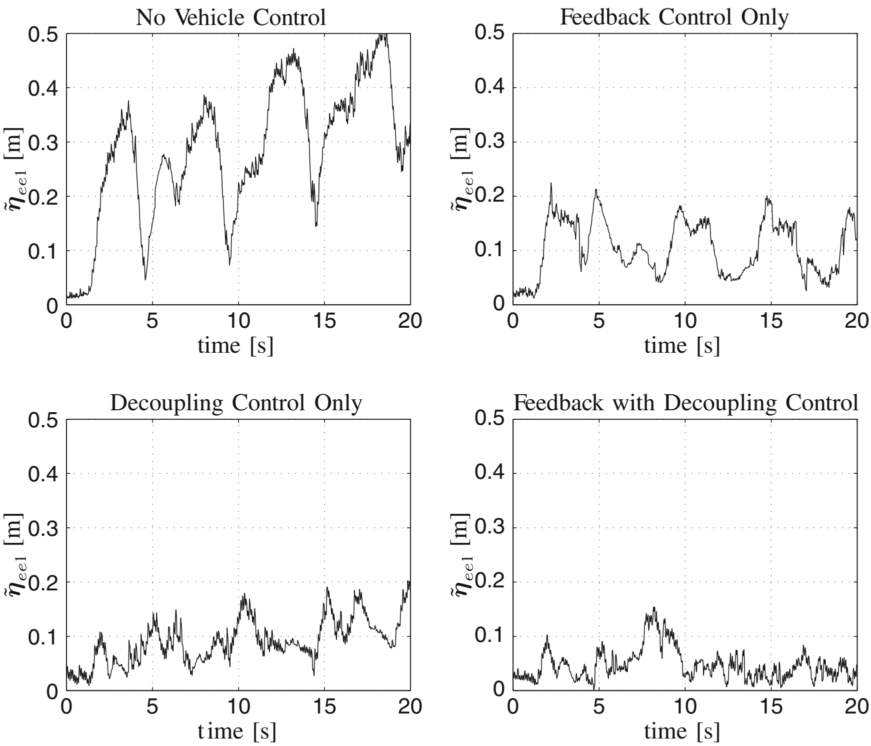


Fig. 7.3 Feedforward Decoupling control. End-effector errors of a periodic motion. *Top-left* without vehicle control; *top-right* with the sole decoupling action; *bottom-left* with the sole vehicle feedback control; *bottom-right* the proposed approach (courtesy of T. McLain, Brigham Young University)

The overall improvement was of a factor 6 with respect to the vehicle without control and a factor 2.5 with respect to a separate control action. The applied thrust only showed an increase of about 5 %.

7.3 Feedback Linearization

Reference [24] presents a model based control law. In detail, the symbolic dynamic model is derived using the Kane's equations [25]; this is further used in order to apply a full dynamic compensation. Simulations of a 6-DOF vehicle carrying two 3-link manipulators are provided. A similar approach has been presented in [26, 27].

From the mathematical point of view, the dynamics of an UVMS can be completely cancelled by resorting to the following control action:

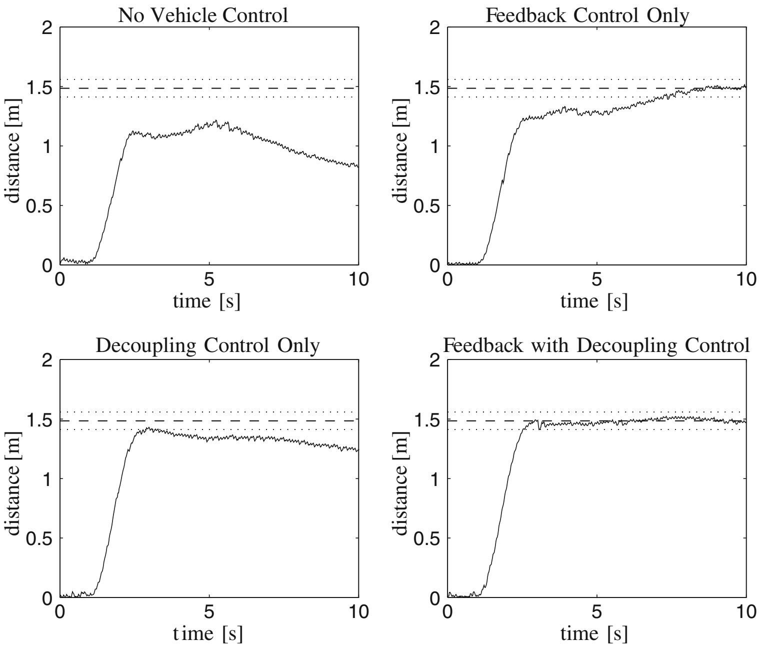


Fig. 7.4 Feedforward Decoupling control. End-effector step response. *Top-left* without vehicle control; *top-right* with the sole decoupling action; *bottom-left* with the sole vehicle feedback control; *bottom-right* the proposed approach (courtesy of T. McLain, Brigham Young University)

$$\boxed{\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}}_a + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q},\boldsymbol{R}_B^I)} \tag{7.3}$$

where the $6+n$ dimensional vector

$$\dot{\boldsymbol{\zeta}}_a = \begin{bmatrix} \dot{\boldsymbol{\nu}}_a \\ \ddot{\boldsymbol{q}}_a \end{bmatrix}$$

is composed by a 6×1 vector defined as

$$\begin{aligned} \dot{\boldsymbol{\nu}}_a &= \boldsymbol{J}_e \ddot{\boldsymbol{\eta}}_e + \dot{\boldsymbol{J}}_e \dot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{\eta}}_e &= \ddot{\boldsymbol{\eta}}_d + \boldsymbol{K}_{pv}\tilde{\boldsymbol{\eta}} + \boldsymbol{K}_{vv}\dot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{K}_{iv}\int_0^t \tilde{\boldsymbol{\eta}} \end{aligned}$$

and a $n \times 1$ vector defined as

$$\ddot{\boldsymbol{q}}_a = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_{pq}\tilde{\boldsymbol{q}} + \boldsymbol{K}_{vq}\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_{iq}\int_0^t \tilde{\boldsymbol{q}}\,.$$

The stability analysis is straightforward; assuming perfect dynamic compensation, in fact, gives two different linear models for the earth-fixed vehicle variables and the joint positions. With a proper choice of the matrix gains, moreover, the designer can shape the response of a second-order dynamic system.

7.4 Nonlinear Control for UVMSs with Composite Dynamics

In [28–31] the singular perturbation theory has been considered due to the composite nature of UVMSs. The different bandwidth characteristics of the vehicle/manipulator dynamics are used as a basis for the control design. This has been developed having in mind a Vortex vehicle together with a PA10, 7-DOF, manipulator for which Tables 7.1 and 7.2 reports some time response of the subsystems.

Let consider a state vector composed by the earth-fixed-frame-based coordinates of the vehicle and the joint position. In the following the dependencies will be dropped out to increase readability. It can be demonstrated that its $(6+n) \times (6+n)$ inertia matrix has the form [26]:

$$\begin{bmatrix} \boldsymbol{M}_v^\star + \boldsymbol{M}_{qq} & \boldsymbol{M}_{vq} \\ \boldsymbol{M}^{\mathrm{T}}{}_{vq} & \boldsymbol{M}_q \end{bmatrix}$$

where $\boldsymbol{M}_v^\star \in \mathbb{R}^{6\times6}$ is the earth-fixed inertia matrix of the sole vehicle (see Eq. (2.56)), $\boldsymbol{M}_q \in \mathbb{R}^{n\times n}$ is the inertia matrix of the sole manipulator, $\boldsymbol{M}_{qq} \in \mathbb{R}^{6\times6}$ is the contribution of the manipulator on the inertia matrix *seen* from the vehicle, $\boldsymbol{M}_{vq} \in \mathbb{R}^{6\times n}$ is the coupling term between vehicle and manipulator; all the matrices include the added mass. Its inverse can be parameterized in:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{M}_{11}^{-1} & -\boldsymbol{M}_{12} \\ -\boldsymbol{M}^{\mathrm{T}}{}_{12} & \boldsymbol{M}_{22}^{-1} \end{bmatrix} \tag{7.4}$$

Table 7.1 Vortex/PA10's sensor time response

Measured variable	Time response (ms)
Surge and sway x, y	400
Depth z	1000
Yaw ψ	1000
Pitch and roll ϕ, θ	100
Joint position $\boldsymbol{q}$	1

Table 7.2 Vortex/PA10's actuators time response

Type	Time response (ms)
Vehicle thruster	160
Manipulator's electrical CD motor	0.5

where the block diagonal matrices $\boldsymbol{M}_{11}$ and $\boldsymbol{M}_{22}$ are of dimension 6×6 and $n \times n$, respectively, and $\boldsymbol{M}_{12} \in \mathbb{R}^{6 \times n}$.

The equations of motion can be written as:

$$\ddot{\boldsymbol{\eta}} = \boldsymbol{M}_{11}^{-1} \left(\boldsymbol{\tau}_v^{\star} - \boldsymbol{n}^{\star}\right) - \boldsymbol{M}_{12} \left(\boldsymbol{\tau}_q - \boldsymbol{n}_q\right) \tag{7.5}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}_{22}^{-1} \left(\boldsymbol{\tau}_q - \boldsymbol{n}_q\right) - \boldsymbol{M}_{12}{}^{\mathrm{T}} \left(\boldsymbol{\tau}_v^{\star} - \boldsymbol{n}^{\star}\right) \tag{7.6}$$

where the nonlinear terms of the equations of motion have been collected in a $(6+n)$ dimensional vector:

$$\boldsymbol{n} = \begin{bmatrix} \boldsymbol{n}^{\star} \\ \boldsymbol{n}_q \end{bmatrix}$$

with $\boldsymbol{n}^{\star} \in \mathbb{R}^6$ and $\boldsymbol{n}_q \in \mathbb{R}^n$.

In [28] the following controller is assumed for the vehicle:

$$\boxed{\boldsymbol{\tau}_v^{\star} = \left(\hat{\boldsymbol{M}}_v^{\star} + \hat{\boldsymbol{M}}_{qq}\right) \left(\ddot{\boldsymbol{\eta}}_d + k_{vv}\dot{\tilde{\boldsymbol{\eta}}} + k_{pv}\tilde{\boldsymbol{\eta}}\right) + \Delta\boldsymbol{\tau}_v^{\star} ,} \tag{7.7}$$

and for the manipulator

$$\boxed{\boldsymbol{\tau}_q = \hat{\boldsymbol{M}}_q \left(\ddot{\boldsymbol{q}}_d + k_{vq}\dot{\tilde{\boldsymbol{q}}} + k_{pq}\tilde{\boldsymbol{q}}\right)} \tag{7.8}$$

where, as usual, the symbol hat: $\hat{\cdot}$ denotes an estimate, positive definite in this case, of the corresponding matrix and the tilde: $\tilde{\cdot}$ represents the error defined as the desired minus the current variable. The scalar gains are chosen so that:

$$\begin{aligned} k_{vv} &= 2\xi\omega_{0v} , \\ k_{pv} &= \omega_{0v}^2 , \\ k_{vq} &= 2\xi\omega_{0q} , \\ k_{pq} &= \omega_{0q}^2 , \end{aligned}$$

i.e., they are defined by the damping ratio and the natural frequency of the linearized model. The bandwidth ratio

$$\varepsilon = \frac{\omega_{0v}}{\omega_{0q}} \ll 1$$

is given by the closed-loop bandwidths the the vehicle and the manipulator and it is small due to the dynamics of the two subsystems. The additional control action $\Delta\boldsymbol{\tau}_v^{\star}$ can be chosen in different ways, in [30] a *partial singular perturbed model-based compensation* and a *robust non-linear control* have been proposed.

A singular perturbation analysis can be carried out when ε is small. It can be demonstrated the the manipulator is not affected by the slow vehicle dynamics and that, after the fast transient, the approximated model has an error $O(\varepsilon)$. On the other side, the vehicle dynamics is strongly affected by the manipulator's motion. For this reason the additional control action $\Delta\boldsymbol{\tau}_v^\star$ is required. In [28], a robust control is developed for the vehicle in order to counteract the coupling effects.

Simulations are carried out considering a dry weight of 150 kg and a length of 1 m for the vehicle and a dry weight of 40 kg and a length of 1 m for the manipulator. Moreover, the natural frequencies have been chosen as $\omega_{0v} = 1.5$ rad/s and $\omega_{0q} = 15$ rad/s leading to a value of $\varepsilon = 0.1$.

7.5 Non-Regressor-Based Adaptive Control

In 1999 reference [32, 33] extend to UVMSs the non-regressor-based adaptive control developed by Yuh [34] and experimentally validated in [35–38] with respect to AUVs. In [39], this controller is integrated with a disturbance observer to improve its tracking performance and simulated on a 1-DOF vehicle carrying a 2-DOF manipulator.

The main idea is to consider the vehicle subsystem separate from the manipulator subsystem and to develop two controllers with different bandwidth, independent one from the other. It is worth noticing that the controller has been developed together with a kinematic control approach (see Chap. 6).

The vehicle generalized force $\boldsymbol{\tau}_v^\star$, thus, can be computed by considering the controller

$$\boxed{\boldsymbol{\tau}_v^\star = \boldsymbol{K}_{1,v}\ddot{\boldsymbol{\eta}}_d + \boldsymbol{K}_{2,v}\dot{\boldsymbol{\eta}} + \boldsymbol{K}_{3,v} + \boldsymbol{K}_{4,v}\dot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{K}_{5,v}\tilde{\boldsymbol{\eta}} = \sum_{i=1}^{5} \boldsymbol{K}_{i,v}\phi_{i,v}\,,}$$

already defined and discussed in Sect. 3.3.

The manipulator is controlled with the following

$$\boxed{\boldsymbol{\tau}_q = \boldsymbol{K}_{1,q}\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_{2,q}\dot{\boldsymbol{q}} + \boldsymbol{K}_{3,q} + \boldsymbol{K}_{4,q}\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_{5,q}\tilde{\boldsymbol{q}} = \sum_{i=1}^{5} \boldsymbol{K}_{i,q}\phi_{i,q}\,,}$$

where $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$, the gains $\boldsymbol{K}_{i,q} \in \mathbb{R}^{n\times n}$ are computed as

$$\boldsymbol{K}_{i,q} = \frac{\hat{\gamma}_{i,q}\boldsymbol{s}_q{\phi_{i,q}}^{\mathrm{T}}}{\|\boldsymbol{s}_q\|\,\|\phi_{i,q}\|} \qquad i = 1,\ldots,5\,,$$

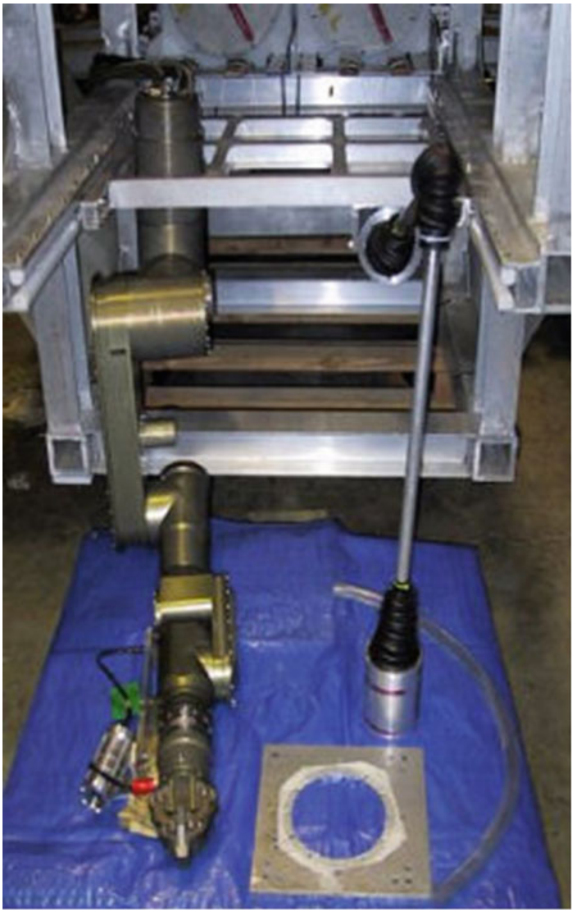


Fig. 7.5 SAUVIM under development and the Autonomous Systems Laboratory, University of Hawaii (courtesy of J. Yuh). The passive joint used for position measurement can be observed

where

$$\boldsymbol{s}_q = \dot{\tilde{\boldsymbol{q}}} + \sigma\tilde{\boldsymbol{q}} \qquad \text{with } \sigma > 0\,,$$

and the factors $\hat{\gamma}_{i,q}$'s are updated by

$$\dot{\hat{\gamma}}_{i,q} = f_{i,q}\,\|\boldsymbol{s}_q\|\,\|\phi_{i,q}\| \qquad \text{with } f_{i,q} > 0 \qquad i = 1, \ldots, 5\,.$$

The stability analysis can be found in [32, 33]. In [33], this controller is used together with the kinematic control detailed in Sect. 6.4 in a simulation study involving a 6-DOF vehicle together with a 3-DOF planar manipulator subject to joint limits. The controller has been developed to be used with the UVMS SAUVIM under development and the Autonomous Systems Laboratory, University of Hawaii (Fig. 7.5). Other works concerning this set-up may be found in [40–42]

7.6 Sliding Mode Control

Robust techniques such as Sliding Mode Control have been successfully applied in control of a wide class of mechanical systems. In this Section the application of a Sliding Mode based approach to motion control of UVMSs is discussed.

The basic idea of this approach is the definition of a sliding surface

$$\boldsymbol{s}(\boldsymbol{x}, t) = \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{x}} = \boldsymbol{0} \tag{7.9}$$

where a second-order mechanical system has been assumed, $\boldsymbol{x}$ is the state vector, t is the time, $\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}$ and $\boldsymbol{\Lambda}$ is a positive definite matrix. When the sliding condition is satisfied, the system is forced to *slide* toward the value $\tilde{\boldsymbol{x}} = \boldsymbol{0}$ with an exponential dynamic (for the scalar case, it is $\dot{\tilde{x}} = -\lambda\tilde{x}$). The control input, thus, has the objective to force the state laying in the sliding surface. With a proper choice of the sliding surface, n-order systems can be controlled considering a first-order problem in $\boldsymbol{s}$.

The only information required to design a stable sliding mode controller is a bound on the dynamic parameters. While this is an interesting property of the controller, one must pay the price of an high control activity. Typically, sliding mode controllers are based on a switching term that causes chattering in the control inputs.

While the first concepts on the sliding surface appeared in the Soviet literature in the end of the fifties, the first robotic applications of sliding mode control are given in [43, 44]. An introduction on Sliding Mode Control theory con be found in [45].

Control law. The vehicle attitude control problem has been addressed among the others in the paper [46] which extends the work in [47] and [48] to obtain a singularity-free tracking control of an underwater vehicle based on the use of the unit quaternion. Inspired by the work in [46], a control law is presented for the regulation problem of an UVMS. To overcome the occurrence of kinematic singularities, the control law is expressed in body-fixed and joint-space coordinates so as to avoid inversion of the system Jacobian. Further, to avoid representation singularities of the orientation, attitude control of the vehicle is achieved through a quaternion based error. The resulting control law is very simple and requires limited computational effort.

Let us recall the dynamic equations in matrix form (2.77):

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) = \boldsymbol{B}\boldsymbol{u}, \tag{2.77}$$

the control law is

$$\boxed{\boldsymbol{u} = \boldsymbol{B}^{\dagger}[\boldsymbol{K}_D\boldsymbol{s} + \hat{\boldsymbol{g}}(\boldsymbol{q}, \boldsymbol{R}_B^I) + \boldsymbol{K}_S \operatorname{sign}(\boldsymbol{s})]\,,} \tag{7.10}$$

where $\boldsymbol{B}^{\dagger}$ is the pseudoinverse of matrix $\boldsymbol{B}$, $\boldsymbol{K}_D$ is a positive definite matrix of gains, $\hat{\boldsymbol{g}}(\boldsymbol{q}, \boldsymbol{R}_B^I)$ is the estimate of gravitational and buoyant forces, $\boldsymbol{K}_S$ is a positive definite matrix, and $\operatorname{sign}(\boldsymbol{x})$ is the vector function whose i-th component is

$$\operatorname{sign}(\boldsymbol{x})_i = \begin{cases} 1 & \text{if } x_i \geq 0 \\ -1 & \text{if } x_i < 0 \end{cases}$$

In (7.10), $\boldsymbol{s}$ is the $\big((6+n) \times 1\big)$ sliding manifold defined as follows

$$\boldsymbol{s} = \boldsymbol{\Lambda} \begin{bmatrix} \boldsymbol{R}_I^B \tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\boldsymbol{q}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \dot{\boldsymbol{q}} \end{bmatrix} = \boldsymbol{y} - \boldsymbol{\zeta}\,, \tag{7.11}$$

with $\boldsymbol{\Lambda} > \boldsymbol{O}$, $\tilde{\boldsymbol{\eta}}_1 = \big[x_d - x \; y_d - y \; z_d - z\big]^{\mathrm{T}}$, $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$ where the subscript d denotes desired values for the relevant variables.

7.6.1 Stability Analysis

In this Section it will be demonstrated that the discussed control law is asymptotically stable in a Lyapunov sense. Let us consider the function

$$V = \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{s}\,, \tag{7.12}$$

that is positive definite being $\boldsymbol{M}(\boldsymbol{q}) > \boldsymbol{O}$.

Differentiating V with respect to time yields

$$\dot{V} = \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\dot{\boldsymbol{M}}\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{s}}$$

that, taking into account the model (2.77), (7.11) and the skew-symmetry of $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$, can be rewritten as

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}[\boldsymbol{M}\dot{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{u} + \boldsymbol{C}\boldsymbol{y} + \boldsymbol{D}\boldsymbol{y} + \boldsymbol{g}]\,. \tag{7.13}$$

Plugging (7.10) into (7.13) gives

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}}(\boldsymbol{D} + \boldsymbol{K}_D)\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}[\boldsymbol{M}\dot{\boldsymbol{y}} + (\boldsymbol{C} + \boldsymbol{D})\boldsymbol{y} + \tilde{\boldsymbol{g}} - \boldsymbol{K}_S \operatorname{sign}(\boldsymbol{s})]$$

that, in view of positive definiteness of $\boldsymbol{K}_D$ and $\boldsymbol{D}$, can be upper bounded as follows

$$\begin{aligned} \dot{V} \leq & - \lambda_{\min}(\boldsymbol{K}_D + \boldsymbol{D})\, \|\boldsymbol{s}\|^2 - \lambda_{\min}(\boldsymbol{K}_S)\, \|\boldsymbol{s}\| \\ & + \|\boldsymbol{M}\dot{\boldsymbol{y}} + (\boldsymbol{C} + \boldsymbol{D})\boldsymbol{y} + \tilde{\boldsymbol{g}}\|\, \|\boldsymbol{s}\| \;, \end{aligned}$$

where $\lambda_{\min}$ denotes the smallest eigenvalue of the corresponding matrix.

By choosing $\boldsymbol{K}_S$ such that

$$\lambda_{\min}(\boldsymbol{K}_S) \geq \|\boldsymbol{M}\dot{\boldsymbol{y}} + (\boldsymbol{C} + \boldsymbol{D})\boldsymbol{y} + \tilde{\boldsymbol{g}}\| \ , \tag{7.14}$$

the time derivative of V is negative definite and thus $\boldsymbol{s}$ tends to zero asymptotically.

If an estimate of the dynamic parameters in (2.77) is available, it might be convenient to consider the control law

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}[\boldsymbol{K}_D\boldsymbol{s} + \hat{\boldsymbol{g}} + \hat{\boldsymbol{M}}\dot{\boldsymbol{y}} + (\hat{\boldsymbol{C}} + \hat{\boldsymbol{D}})\boldsymbol{y} + \boldsymbol{K}_S \text{sign}(\boldsymbol{s})] \tag{7.15}$$

in lieu of (7.10). Starting from the function in (7.12) and plugging (7.15) in (7.13) gives

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}}(\boldsymbol{D} + \boldsymbol{K}_D)\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}[\widetilde{\boldsymbol{M}}\dot{\boldsymbol{y}} + (\widetilde{\boldsymbol{C}} + \widetilde{\boldsymbol{D}})\boldsymbol{y} + \tilde{\boldsymbol{g}} - \boldsymbol{K}_S \text{sign}(\boldsymbol{s})]$$

that, in view of positive definiteness of $\boldsymbol{K}_D$ and $\boldsymbol{D}$, leads to negative definiteness of $\dot{V}$ if

$$\lambda_{\min}(\boldsymbol{K}_S) \geq \left\|\widetilde{\boldsymbol{M}}\dot{\boldsymbol{y}} + (\widetilde{\boldsymbol{C}} + \widetilde{\boldsymbol{D}})\boldsymbol{y} + \tilde{\boldsymbol{g}}\right\| \ . \tag{7.16}$$

It is worth noting that condition (7.16) is weaker than condition (7.14) in that the matrix $\boldsymbol{K}_S$ must overcome the sole model parameters mismatching.

Stability of the sliding manifold. It was demonstrated that the discussed control law guarantees convergence of $\boldsymbol{s}$ to the sliding manifold $\boldsymbol{s} = \boldsymbol{0}$. In the following it will be demonstrated that, once the sliding manifold has been reached, the error vectors $\tilde{\boldsymbol{\eta}}_1, \tilde{\boldsymbol{\varepsilon}}, \tilde{\boldsymbol{q}}$ converge asymptotically to the origin, i.e., that regulation of the system variables to their desired values is achieved.

By taking $\boldsymbol{\Lambda} = \text{blockdiag}\{\boldsymbol{\Lambda}_p, \boldsymbol{\Lambda}_o, \boldsymbol{\Lambda}_q\}$ where $\boldsymbol{\Lambda}_p \in \mathbb{R}^{3\times3}$, $\boldsymbol{\Lambda}_o \in \mathbb{R}^{3\times3}$, $\boldsymbol{\Lambda}_q \in \mathbb{R}^{n\times n}$, from (7.11) it is possible to notice that the stability analysis can be *decoupled* in 3 parts as follows.

Vehicle position error dynamics. The vehicle position error dynamics on the sliding manifold is described by the equation

$$-\boldsymbol{\nu}_1 + \boldsymbol{\Lambda}_p \boldsymbol{R}_I^B \tilde{\boldsymbol{\eta}}_1 = \boldsymbol{0} \, .$$

Notice that the rotation matrix $\boldsymbol{R}_I^B$ is a function of the vehicle orientation.

By considering

$$V = \frac{1}{2}\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_1$$

as Lyapunov function candidate and observing that $\dot{\boldsymbol{\eta}}_1 = \boldsymbol{R}_B^I \boldsymbol{\nu}_1$, it is easily obtained

$$\dot{V} = -\tilde{\boldsymbol{\eta}}_1^{\mathrm{T}} \boldsymbol{R}_B^I \boldsymbol{\Lambda}_p \boldsymbol{R}_I^B \tilde{\boldsymbol{\eta}}_1 \ ,$$

that is negative definite for $\boldsymbol{\Lambda}_p > \boldsymbol{O}$. Hence, $\tilde{\boldsymbol{\eta}}_1$ converges asymptotically to the origin.

Vehicle orientation error dynamics. The vehicle orientation error dynamics on the sliding manifold is described by the equation

$$-\boldsymbol{\nu}_2 + \boldsymbol{\Lambda}_o \tilde{\varepsilon} = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\nu}_2 = \boldsymbol{\Lambda}_o \tilde{\varepsilon} \,. \tag{7.17}$$

Further, by taking into account the quaternion propagation and (7.17), it can be recognized that

$$\dot{\tilde{\eta}} = \frac{1}{2}\tilde{\varepsilon}^{\mathrm{T}}\boldsymbol{\nu}_2 = \frac{1}{2}\tilde{\varepsilon}^{\mathrm{T}}\boldsymbol{\Lambda}_o\tilde{\varepsilon} \,. \tag{7.18}$$

Let consider the Lyapunov function candidate

$$V = \tilde{\varepsilon}^{\mathrm{T}}\tilde{\varepsilon} \,. \tag{7.19}$$

The time derivative of V is:

$$\dot{V} = 2\tilde{\varepsilon}^{\mathrm{T}}\dot{\tilde{\varepsilon}} = -\tilde{\varepsilon}^{\mathrm{T}}\tilde{\eta}\boldsymbol{\nu}_2 - \tilde{\varepsilon}^{\mathrm{T}}\boldsymbol{S}(\tilde{\varepsilon})\boldsymbol{\nu}_2 \,. \tag{7.20}$$

Plugging (7.17) into (7.20) and taking $\boldsymbol{\Lambda}_o = \lambda_o \boldsymbol{I}_3$ with $\lambda_o > 0$, gives

$$\dot{V} = -\tilde{\eta}\lambda_o\tilde{\varepsilon}^{\mathrm{T}}\tilde{\varepsilon} - \lambda_o\tilde{\varepsilon}^{\mathrm{T}}\boldsymbol{S}(\tilde{\varepsilon})\tilde{\varepsilon} = -\tilde{\eta}\lambda_o\tilde{\varepsilon}^{\mathrm{T}}\tilde{\varepsilon} \,.$$

which is negative semidefinite with $\tilde{\eta} \geq 0$. It must be noted that, in view of (7.18), $\tilde{\eta}$ is a not-decreasing function of time and thus it stays positive when starting from a positive initial value.

The set R of all points $\tilde{\varepsilon}$ where $\dot{V} = 0$ is given by

$$R = \{\tilde{\varepsilon} = \mathbf{0}, \quad \tilde{\varepsilon} : \tilde{\eta} = 0\} \,;$$

from (2.7), however, it can be recognized that

$$\tilde{\eta} = 0 \quad \Rightarrow \quad \|\tilde{\varepsilon}\| = 1$$

and thus $\dot{\tilde{\eta}} > 0$ in view of (7.18). Therefore, the largest invariant set in R is

$$M = \{\tilde{\varepsilon} = \mathbf{0}\}$$

and the invariant set theorem ensures asymptotic convergence to the origin.

Manipulator joint error dynamics. The manipulator joint error dynamics on the sliding manifold is described by the equation

$$-\dot{\boldsymbol{q}} + \boldsymbol{\Lambda}_q\tilde{\boldsymbol{q}} = \mathbf{0}$$

whose convergence to $\tilde{\boldsymbol{q}} = \mathbf{0}$ is evident taking $\boldsymbol{\Lambda}_q > \boldsymbol{O}$.

7.6.2 Simulations

Dynamic simulations have been performed in order to show the effectiveness of the discussed control law. The UVMS simulator was developed in the MATLAB©/ SIMULINK© environment and it is described in Chap. 9.

For this simulations, the vehicle data are taken from [49]; they refer to the experimental Autonomous Underwater Vehicle NPS Phoenix. A two-link manipulator with rotational joints has been considered which is mounted under the vehicle body with the joint axes parallel to the fore-aft direction; since the vehicle inertia along that axis is minimum, this choice increases dynamic coupling between the vehicle and the manipulator. The length of each link is 1 m, the center of gravity is coincident with the center of buoyancy and it is supposed to be in the geometrical center of the link; each link is not neutrally buoyant. Dry and viscous joint frictions are also taken into account.

As for the control law, implementation of (7.10) was considered; however, it is well known that the sign function would lead to chattering in the system. Practical implementation of (7.10), therefore, requires replacement of the sign function, e.g., with the sat function

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}[\boldsymbol{K}_D\boldsymbol{s} + \hat{\boldsymbol{g}}(\boldsymbol{q}, \boldsymbol{R}_I^B) + \boldsymbol{K}_S \mathrm{sat}(\boldsymbol{s}, \varepsilon)]\,, \tag{7.21}$$

where the sat$(\boldsymbol{x}, \varepsilon)$ is the vector function whose i-th component is

$$\mathrm{sat}(\boldsymbol{x}, \varepsilon)_i = \begin{cases} 1 & \text{if } x_i > \varepsilon \\ -1 & \text{if } x_i < -\varepsilon \\ \frac{x_i}{\varepsilon} & \text{otherwise} \end{cases}$$

Convergence to the equilibrium of the UVMS under this different control law can be easily demonstrated starting from (7.12) following the guidelines in [45]. In detail, it is obtained that $\dot{V} < 0$ in the region characterized by $\|\boldsymbol{s}\| \geq \varepsilon$, while the sign of $\dot{V}$ is undetermined in the boundary layer characterized by $\|\boldsymbol{s}\| < \varepsilon$. This approach is well established in sliding mode control and does not represent a practical drawback since ε can be taken sufficiently small.

In the simulation $\boldsymbol{B}$ is supposed to be the identity matrix, meaning that direct control of forces and moments acting on the vehicle and joint torques is available. The control law parameters are

$$\begin{aligned} \boldsymbol{\Lambda}_o = \boldsymbol{\Lambda}_p &= \mathrm{diag}\{0.5, 0.5, 0.5\}\,, \\ \boldsymbol{\Lambda}_q &= \mathrm{diag}\{3, 2\}\,, \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{K}_D &= \text{blockdiag}\{10^4 \boldsymbol{I}_6, 3000, 500\}\,, \\ \boldsymbol{K}_S &= 1000 \boldsymbol{I}_8\,, \\ \varepsilon &= 0.1\,. \end{aligned}$$

A station keeping task for the vehicle in the initial location was considered

$$\boldsymbol{\eta}_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \quad \text{m, rad}$$

with the manipulator in the initial configuration $\boldsymbol{q}_i = \begin{bmatrix} -\frac{\pi}{4} & \frac{\pi}{2} \end{bmatrix}^{\mathrm{T}}$ rad. The vehicle must be then kept still, i.e., $\boldsymbol{\eta}_d = \boldsymbol{\eta}_i$, while moving the manipulator arm to the desired final configuration $\boldsymbol{q}_f = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$ rad according to a fifth-order polynomial.

It should be noted that the vehicle orientation set point is assigned in terms of Euler angles; these must be converted into the corresponding rotation matrix so as to extract the quaternion expressing the orientation error from the rotation matrix computed as in Sect. 2.2.3. Remarkably, this procedure is free of singularities.

The obtained simulation results are reported in Figs. 7.6, 7.7 and 7.8 in terms of the time histories of the vehicle position, the vehicle control forces, the vehicle attitude expressed by Euler angles, the vehicle moments, the manipulator joint errors, and the manipulator joint torques, respectively.

Figure 7.6 shows that, as expected, the vehicle position is affected by the manipulator motion; however, the displacements are small and the target position is recovered after a transient. It can be recognized that at steady state the force along z is non null; this happens because the manipulator is not neutrally buoyant.

Figure 7.7 shows that the dynamic coupling is mostly experienced along the roll direction because of the chosen UVMS structure. This effect was intentional in order to test the control robustness. It can be recognized that vehicle control moments are zero at steady state; this happens because the center of gravity and the center of

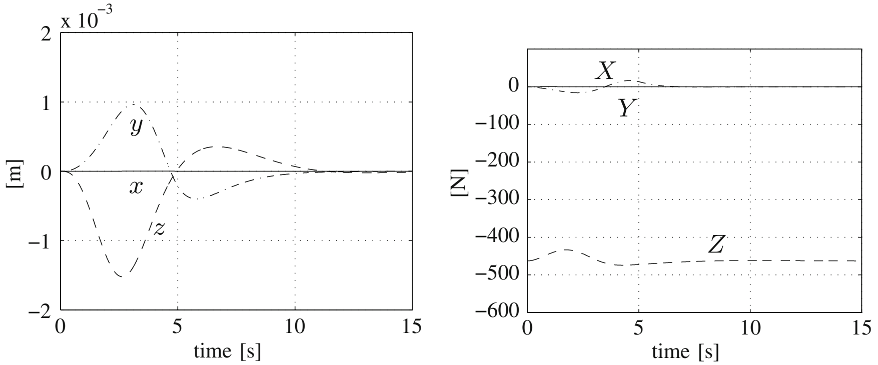


Fig. 7.6 Sliding mode control. *Left* vehicle positions. *Right* vehicle control forces

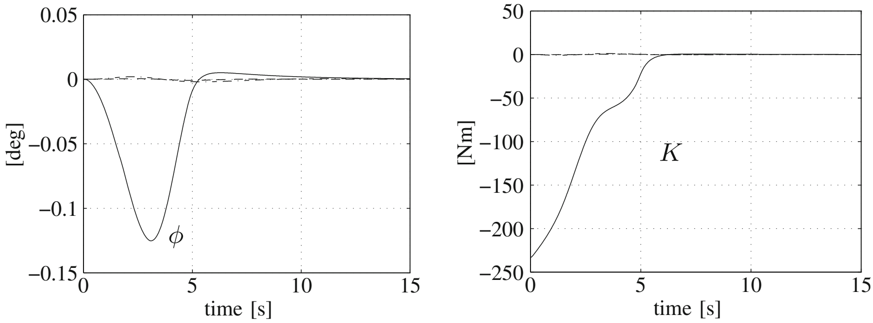


Fig. 7.7 Sliding mode control. *Left* vehicle attitude in terms of Euler angles. *Right* vehicle control moments

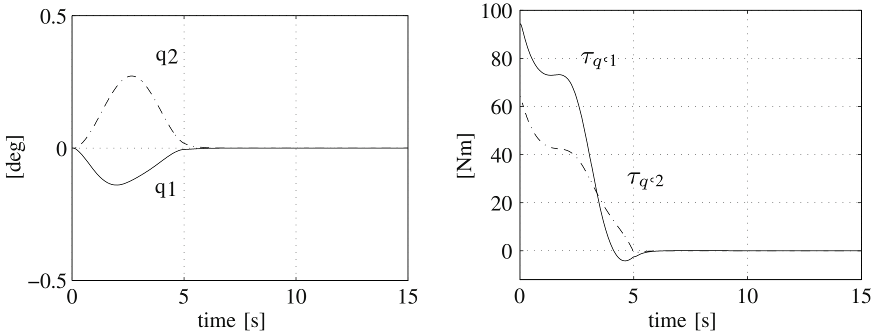


Fig. 7.8 Sliding mode control. *Left* manipulator joint errors. *Right* manipulator joint torques. The steady state vehicle moment and manipulator torques are null due to the restoring force characteristic of this specific UVMS

buoyancy of vehicle body and manipulator links are all aligned with the z-axis of the earth-fixed frame at the final system configuration.

Figure 7.8 shows the time histories of manipulator joint errors and torques. It is worth noting that the initial value of the joint torques is non null because of gravity and buoyancy compensation, while they are null at steady state in view of the particular final system configuration. It can be recognized that control generalized forces are smooth while the task is successfully executed.

7.7 Adaptive Control

Adaptive Control is a wide topic in control theory. The basic idea of adaptive control is to modify on-line some control gains to *adapt* the controller to the plant's parameters that are supposed to be unknown or slowly varying. However, the term *adaptive control* can assume slightly different meanings. In this work, a controller is considered adaptive if it includes explicitly on-line system parameters estimation.

Often, the mathematical model of the system to be controlled is known but the dynamic parameters are not known or may depend from a load. In this case, the adaptive control is mainly based on a PD action plus a dynamic compensation the parameters of which are updated on-line. This dynamic compensation can be intended to cancel the system dynamics, thus achieving decoupling and linearization of the system, or preserve the passivity properties of the closed loop system [50].

While the first concepts of adaptive control appeared, without success, in the aircraft control in the early fifties, the first robotic applications appeared later [50, 51].

Control law. Based on the control law developed in the previous Section, a control law is presented for the tracking problem of UVMSs. As in the previous control law, to overcome the occurrence of kinematic singularities, the control law is expressed in body-fixed and joint-space coordinates so as to avoid inversion of the system Jacobian. Further, to avoid representation singularities of the orientation, attitude control of the vehicle is achieved through a quaternion based error. To achieve good tracking performance, the control law includes model-based compensation of the system dynamics. An adaptive estimate of the model parameters is provided, since they are uncertain and slowly varying.

Given the dynamic equations in matrix form (2.77)–(2.79):

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q},\boldsymbol{R}_B^I) = \boldsymbol{\Phi}(\boldsymbol{q},\boldsymbol{R}_I^B,\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\boldsymbol{\theta} = \boldsymbol{B}\boldsymbol{u}\,,$$

the control law is

$$\boxed{\boldsymbol{u} = \boldsymbol{B}^{\dagger}[\boldsymbol{K}_D\boldsymbol{s}' + \boldsymbol{\Phi}(\boldsymbol{q},\boldsymbol{R}_I^B,\boldsymbol{\zeta},\boldsymbol{\zeta}_r,\dot{\boldsymbol{\zeta}}_r)\hat{\boldsymbol{\theta}}]\,,} \tag{7.22}$$

with the update law given by

$$\boxed{\dot{\hat{\boldsymbol{\theta}}} = \boldsymbol{K}_{\theta}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{q},\boldsymbol{R}_I^B,\boldsymbol{\zeta},\boldsymbol{\zeta}_r,\dot{\boldsymbol{\zeta}}_r)\boldsymbol{s}\,,} \tag{7.23}$$

where $\boldsymbol{B}^{\dagger}$ is the pseudoinverse of matrix $\boldsymbol{B}$, $\boldsymbol{K}_{\theta} > \boldsymbol{O}$ and $\boldsymbol{\Phi}$ is the system regressor defined in (2.79). The vectors $\boldsymbol{s}' \in R^{(6+n)\times 1}$ and $\boldsymbol{s} \in R^{(6+n)\times 1}$ are defined as follows

$$\boldsymbol{s}' = \begin{bmatrix} \tilde{\boldsymbol{\nu}}_1 \\ \tilde{\boldsymbol{\nu}}_2 \\ \dot{\tilde{\boldsymbol{q}}} \end{bmatrix} + \left(\boldsymbol{\Lambda} + \boldsymbol{K}_D^{-1}\boldsymbol{K}_P\right) \begin{bmatrix} {}^{B}\boldsymbol{R}_I\tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\boldsymbol{q}} \end{bmatrix} = \tilde{\boldsymbol{\zeta}} + \left(\boldsymbol{\Lambda} + \boldsymbol{K}_D^{-1}\boldsymbol{K}_P\right)\tilde{\boldsymbol{y}}, \tag{7.24}$$

$$\boldsymbol{s} = \tilde{\boldsymbol{\zeta}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{y}}\,, \tag{7.25}$$

with $\tilde{\boldsymbol{\eta}}_1 = \left[x_d - x \; y_d - y \; z_d - z\right]^{\mathrm{T}}$, $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$, $\tilde{\boldsymbol{\nu}}_1 = \boldsymbol{\nu}_{1,d} - \boldsymbol{\nu}_1$, $\dot{\tilde{\boldsymbol{q}}} = \dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}$, where the subscript d denotes desired values for the relevant variables.

$\boldsymbol{\Lambda}$ is defined as $\boldsymbol{\Lambda} = \text{blockdiag}\{\lambda_p \boldsymbol{I}_3, \lambda_o \boldsymbol{I}_3, \boldsymbol{\Lambda}_q\}$ with $\boldsymbol{\Lambda}_q \in R^{n\times n}$, $\boldsymbol{\Lambda} > \boldsymbol{O}$.$\boldsymbol{K}_P$ is defined as $\boldsymbol{K}_P = \text{blockdiag}\{k_p \boldsymbol{I}_3, k_o \boldsymbol{I}_3, \boldsymbol{K}_q\}$, with $\boldsymbol{K}_q \in R^{n\times n}$, $\boldsymbol{K}_P > \boldsymbol{O}$.$\boldsymbol{K}_q$ and $\boldsymbol{\Lambda}_q$ must be defined so as $\boldsymbol{K}_q \boldsymbol{\Lambda}_q > \boldsymbol{O}$. Finally, it is $\boldsymbol{\zeta}_r = \boldsymbol{\zeta}_d + \boldsymbol{\Lambda}\tilde{\boldsymbol{y}}$ and $\boldsymbol{K}_D > \boldsymbol{O}$.

7.7.1 Stability Analysis

In this Section it will be shown that the control law (7.22)–(7.23) is stable in a Lyapunov-Like sense. Let define the following partition for the variable $\boldsymbol{s}$ that will be useful later:

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_p \\ \boldsymbol{s}_o \\ \boldsymbol{s}_q \end{bmatrix} \tag{7.26}$$

with $\boldsymbol{s}_p \in R^3$, $\boldsymbol{s}_o \in R^3$, $\boldsymbol{s}_q \in R^n$ respectively.

Let us consider the scalar function

$$\begin{aligned} V = & \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{s} + \frac{1}{2}\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{K}_\theta\tilde{\boldsymbol{\theta}} \\ & + \frac{1}{2}\begin{bmatrix} \tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{z}} \\ \tilde{\boldsymbol{q}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} k_p\boldsymbol{I}_3 & \boldsymbol{O}_{3\times 4} & \boldsymbol{O}_{3\times n} \\ \boldsymbol{O}_{4\times 3} & 2k_o\boldsymbol{I}_4 & \boldsymbol{O}_{4\times n} \\ \boldsymbol{O}_{n\times 3} & \boldsymbol{O}_{n\times 4} & \boldsymbol{K}_q \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{z}} \\ \tilde{\boldsymbol{q}} \end{bmatrix} \end{aligned} \tag{7.27}$$

where $\tilde{\boldsymbol{z}} = \left[1 \; \boldsymbol{0}^{\mathrm{T}}\right]^{\mathrm{T}} - \boldsymbol{z} = \left[1 - \tilde{\eta} \; -\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\right]^{\mathrm{T}}$.$V \geq 0$ in view of positive definiteness of $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{K}_\theta$, k_p, k_o and $\boldsymbol{K}_q$.

Differentiating V with respect to time yields

$$\begin{aligned} \dot{V} = & \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\dot{\boldsymbol{M}}\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{s}} + \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{K}_\theta\dot{\tilde{\boldsymbol{\theta}}} \\ & + k_p\tilde{\boldsymbol{\eta}}_1{}^{\mathrm{T}}\boldsymbol{R}_B^I\tilde{\boldsymbol{\nu}}_1 - 2k_o\tilde{\boldsymbol{z}}^{\mathrm{T}}\boldsymbol{J}_{k,oq}(\boldsymbol{z})\tilde{\boldsymbol{\nu}}_2 + \tilde{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{K}_q\dot{\tilde{\boldsymbol{q}}}\,. \end{aligned} \tag{7.28}$$

Observing that, in view of (7.25) and (7.26) it is

$$\tilde{\boldsymbol{\nu}}_1 = \boldsymbol{s}_p - \lambda_p\boldsymbol{R}_I^B\tilde{\boldsymbol{\eta}}_1, \tag{7.29}$$

$$\tilde{\boldsymbol{\nu}}_2 = \boldsymbol{s}_o - \lambda_o\tilde{\boldsymbol{\varepsilon}}, \tag{7.30}$$

$$\dot{\tilde{\boldsymbol{q}}} = \boldsymbol{s}_q - \boldsymbol{\Lambda}_q\tilde{\boldsymbol{q}}, \tag{7.31}$$

and taking into account (2.77), (2.16), (7.29)–(7.31), and the skew-symmetry of $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$, (7.28) can be rewritten as

$$
\begin{aligned}
\dot{V} = {} & -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{s} - \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{K}_{\theta}\dot{\hat{\boldsymbol{\theta}}} + \boldsymbol{s}^{\mathrm{T}}[\boldsymbol{M}\dot{\boldsymbol{\zeta}}_r - \boldsymbol{B}\boldsymbol{u} + \boldsymbol{C}(\boldsymbol{\zeta})\boldsymbol{\zeta}_r + \boldsymbol{D}(\boldsymbol{\zeta})\boldsymbol{\zeta}_r + \boldsymbol{g}] \\
& + k_p\tilde{\boldsymbol{\eta}}_1{}^{\mathrm{T}}\boldsymbol{R}_B^I\boldsymbol{s}_p - k_p\lambda_p\tilde{\boldsymbol{\eta}}_1{}^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_1 + k_o\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\boldsymbol{s}_o - \lambda_o k_o\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\tilde{\boldsymbol{\varepsilon}} \\
& + \tilde{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{K}_q\boldsymbol{s}_q - \tilde{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{K}_q\boldsymbol{\Lambda}_q\tilde{\boldsymbol{q}} \qquad (7.32)
\end{aligned}
$$

where $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$ was assumed, i.e., the dynamic parameters are constant or slowly varying.

Exploiting (2.79), (7.32) can be rewritten in compact form:

$$
\begin{aligned}
\dot{V} = {} & -\begin{bmatrix}\tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\boldsymbol{q}}\end{bmatrix}^{\mathrm{T}} \begin{bmatrix} k_p\lambda_p\boldsymbol{I}_3 & \boldsymbol{O}_{3\times3} & \boldsymbol{O}_{3\times n} \\ \boldsymbol{O}_{3\times3} & k_o\lambda_o\boldsymbol{I}_3 & \boldsymbol{O}_{3\times n} \\ \boldsymbol{O}_{n\times3} & \boldsymbol{O}_{n\times3} & \boldsymbol{K}_q\boldsymbol{\Lambda}_q\end{bmatrix} \begin{bmatrix}\tilde{\boldsymbol{\eta}}_1 \\ \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\boldsymbol{q}}\end{bmatrix} \\
& + \boldsymbol{s}^{\mathrm{T}}\left[\boldsymbol{\Phi}(\boldsymbol{q}, \boldsymbol{R}_I^B, \boldsymbol{\zeta}, \boldsymbol{\zeta}_r, \dot{\boldsymbol{\zeta}}_r)\boldsymbol{\theta} - \boldsymbol{B}\boldsymbol{u} + \boldsymbol{K}_P\tilde{\boldsymbol{y}}\right] - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{s} - \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{K}_{\theta}\dot{\hat{\boldsymbol{\theta}}} \,. \qquad (7.33)
\end{aligned}
$$

Plugging the control law (7.22)–(7.23) into (7.33), one finally obtains:

$$
\dot{V} = -\tilde{\boldsymbol{y}}^{\mathrm{T}}\boldsymbol{K}'\tilde{\boldsymbol{y}} - \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{K}_D + \boldsymbol{D}\right)\boldsymbol{s}
$$

that is negative semi-definite over the state space $\{\tilde{\boldsymbol{y}}, \boldsymbol{s}, \tilde{\boldsymbol{\theta}}\}$.

It is now possible to prove the system stability in a Lyapunov-Like sense using the Barbălat's Lemma. Since V is lower bounded, $\dot{V}(\tilde{\boldsymbol{y}}, \boldsymbol{s}, \tilde{\boldsymbol{\theta}}) \leq 0$ and $\dot{V}(\tilde{\boldsymbol{y}}, \boldsymbol{s}, \tilde{\boldsymbol{\theta}})$ is uniformly continuous then $\dot{V}(\tilde{\boldsymbol{y}}, \boldsymbol{s}, \tilde{\boldsymbol{\theta}}) \rightarrow 0$ as $t \rightarrow \infty$. Thus $\tilde{\boldsymbol{y}}, \boldsymbol{s} \rightarrow \boldsymbol{0}$ as $t \rightarrow \infty$. However it is not possible to prove asymptotic stability of the state, since $\tilde{\boldsymbol{\theta}}$ is only guaranteed to be bounded.

7.7.2 Simulations

By considering the same UVMS as the previous Section, numerical simulations have been performed to test the discussed control law.

The full simulated model includes a large number of dynamic parameters, thus the symbolic regressor $\boldsymbol{\Phi} \in R^{(6+n)\times n_{\theta}}$ has a complex expression. While the simulation is performed via the Newton-Euler based algorithm to overcome this complexity, the control law requires the expression of the symbolic regressor. Practical implementation of (7.22)–(7.23), then, might benefit from some simplifications. Considering that the vehicle is usually kept still during the manipulator motion it is first proposed to decouple the vehicle and manipulator dynamics in the regressor computation. The only coupling effect considered is the vehicle orientation in the manipulator restoring effects. Further, it is possible to simplify the hydrodynamic effects taking a linear and a quadratic velocity dependent term for each degree of freedom.

In this reduced form the vehicle regressor is the same as if it was without manipulator. On the other hand the manipulator regressor is the ground fixed regressor, except for the computation of the restoring forces in which the vehicle orientation cannot be omitted; this means that a reduced set of dynamic parameters has been obtained. In this simulation the vehicle parameter vector is $\hat{\boldsymbol{\theta}}_v \in R^{n_{\theta,v}}$ with $n_{\theta,v} = 23$, the manipulator parameter vector is $\hat{\boldsymbol{\theta}}_m \in R^{n_{\theta,m}}$ with $n_{\theta,m} = 13$; it can be recognized that $n_{\theta,v} + n_{\theta,m} \ll n_\theta$.

The regressor implemented in the control law (7.22)–(7.23) has then the form:

$$\boldsymbol{\Phi}(\boldsymbol{q}, \boldsymbol{R}_I^B, \boldsymbol{\zeta}, \boldsymbol{\zeta}_r, \dot{\boldsymbol{\zeta}}_r) = \begin{bmatrix} \boldsymbol{\Phi}_v(\boldsymbol{R}_I^B, \boldsymbol{\nu}, \boldsymbol{\zeta}_{r,v}, \dot{\boldsymbol{\zeta}}_{r,v}) & \boldsymbol{O}_{6\times 13} \\ \boldsymbol{O}_{2\times 23} & \boldsymbol{\Phi}_m(\boldsymbol{q}, \boldsymbol{R}_I^B, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}_{r,m}, \dot{\boldsymbol{\zeta}}_{r,m}) \end{bmatrix}$$

where $\boldsymbol{\Phi}_v \in R^{6\times 23}$ is the vehicle regressor and $\boldsymbol{\Phi}_m \in R^{2\times 13}$ is the manipulator regressor. The corresponding parameter vector is $\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_v^{\mathrm{T}} & \hat{\boldsymbol{\theta}}_m^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. The vectors $\boldsymbol{\zeta}_{r,v}$ and $\boldsymbol{\zeta}_{r,m}$ are the vehicle and manipulator components of $\boldsymbol{\zeta}$.

In the simulation the initial value of $\hat{\boldsymbol{\theta}}$ is affected by an error greater then the 50 % of the true value. The control law parameters are

$$\begin{aligned} \boldsymbol{\Lambda} &= \mathrm{blockdiag}\{0.5\boldsymbol{I}_3, 0.5\boldsymbol{I}_3, 3, 2\}\,, \\ \boldsymbol{K}_D &= 1000\boldsymbol{I}_8\,, \\ \boldsymbol{K}_P &= 100\boldsymbol{I}_8\,. \end{aligned}$$

$\dot{\tilde{\boldsymbol{y}}}$ is computed by a filtered numerical time derivative:

$$\dot{\tilde{\boldsymbol{y}}}_k = \alpha\dot{\tilde{\boldsymbol{y}}}_{k-1} + (1-\alpha)\frac{\tilde{\boldsymbol{y}}_k - \tilde{\boldsymbol{y}}_{k-2}}{2\Delta T}\,,$$

with ΔT being the simulation sampling time.

A station keeping task for the system in the initial configuration was considered

$$\begin{aligned} \boldsymbol{\eta}_i &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} && \mathrm{m, rad}\,, \\ \boldsymbol{q}_i &= \begin{bmatrix} \frac{\pi}{4} & \frac{\pi}{6} \end{bmatrix}^{\mathrm{T}} && \mathrm{rad}\,. \end{aligned}$$

The vehicle must be then kept still, i.e., $\boldsymbol{\eta}_d = \boldsymbol{\eta}_i$, while moving the manipulator arm to the desired final configuration $\boldsymbol{q}_f = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$ rad according to a fifth-order polynomial. The trajectory is executed two twice without resetting the parameter update.

It should be noted that, if the vehicle orientation trajectory was assigned in terms of Euler angles, these should be converted into the corresponding rotation matrix so as to extract the quaternion expressing the orientation error from the rotation matrix computed as described in Sect. 2.2.3. Remarkably, this procedure is free of singularities.

The obtained simulation results are reported in Figs. 7.9, 7.10 and 7.11 in terms of the time histories of the vehicle position, the vehicle control forces, the vehicle

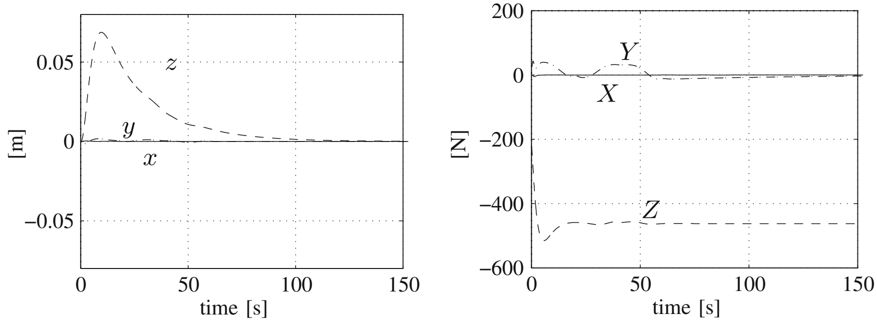


Fig. 7.9 Adaptive control. *Left* vehicle positions. *Right* vehicle control forces

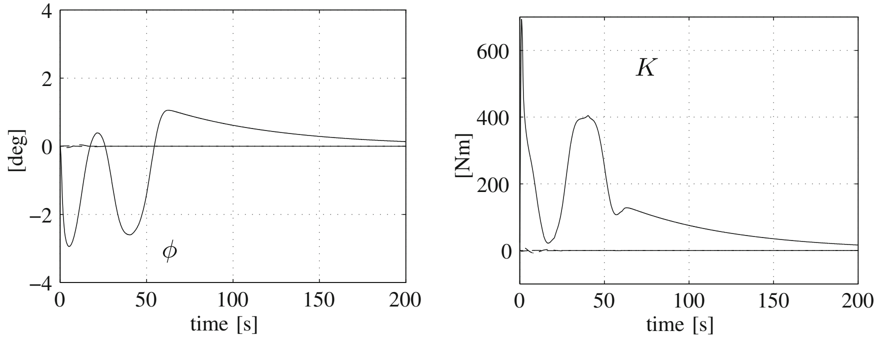


Fig. 7.10 Adaptive control. *Left* vehicle attitude in terms of Euler angles. *Right* vehicle control moments

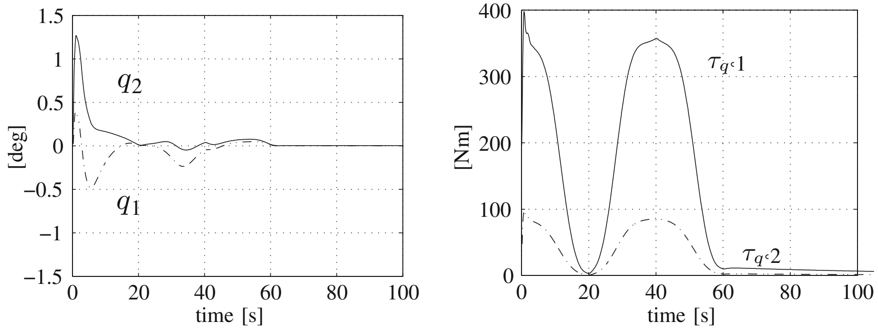


Fig. 7.11 Adaptive control. *Left* joint position errors. *Right* joint control torques

attitude expressed by Euler angles, the vehicle moments, the manipulator joint errors, and the manipulator joint torques, respectively.

Figure 7.9 shows that, as expected, the vehicle position is affected by the manipulator motion. The main displacement is observed along z; this is due to the intentional large initial error in the restoring force compensation. However, the displacements are small and the target position is recovered after a transient. It can be recognized that at steady state the force along z is non null; this happens because the manipulator is not neutrally buoyant. The mismatching in the initial restoring force compensation is recovered by the update of the parameter estimation. The manipulator weight is not included in the vehicle regressor, nevertheless it is compensated as a gravitational vehicle parameter and a null steady state error is obtained.

Figure 7.10 shows that the dynamic coupling is mostly experienced along the roll direction because of the chosen UVMS structure. This effect was intentional in order to test the control robustness. It can be recognized that vehicle control moments are zero at steady state; this happens because the center of gravity and the center of buoyancy of vehicle body and manipulator links are all aligned with the z-axis of the earth-fixed frame at the final system configuration.

Figure 7.11 shows the time histories of manipulator joint errors and torques. It is worth noting that the initial value of the joint torques is non null because of gravity and buoyancy compensation, while they are null at steady state in view of the particular final system configuration. The large initial joint error is due to mismatching in the restoring torques compensation; the integral action provided by the parameters update gives a null steady state error.

Figure 7.12 finally shows a performance comparison between (7.22) and (7.23) with and without adaptation. Remarkably at the very beginning of the trajectory both control laws perform the same error; afterward the adaptive controller provides a significant error reduction.

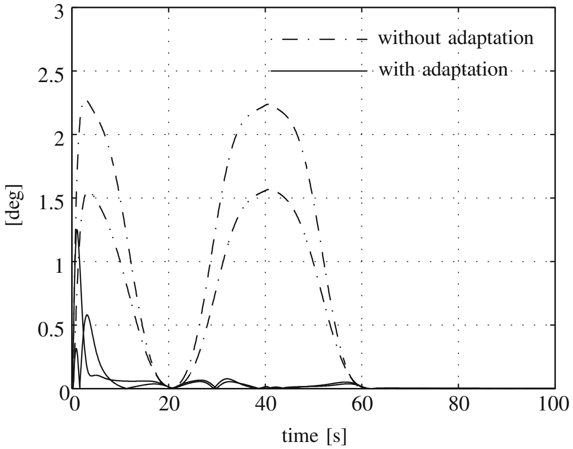


Fig. 7.12 Adaptive control. Comparison between adaptive and PD + dynamic compensation in terms of joint position error absolute values

7.8 Output Feedback Control

Underwater vehicles are typically equipped with acoustic sensors or video systems for position measurements, while the vehicle attitude can be obtained from gyroscopic sensors and/or compasses. Velocity measurements are usually obtained from sensors based on the Doppler effect.

In the case of underwater vehicles operating close to off-shore structures, position and orientation measurements are fairly accurate, while velocity measurements are poor, especially during slow maneuvers. Hence, it is worth devising algorithms for position and attitude control that do not require direct velocity feedback.

A nonlinear observer for vehicle velocity and acceleration has been proposed in [52, 53], although a combined controller-observer design procedure is not developed. On the other hand, a passivity-based control law is proposed in [54], where the velocities are reconstructed via a lead filter; however, this control scheme achieves only regulation of position and orientation variables for an underwater vehicle-manipulator system.

In this section, the problem of output feedback tracking control of UVMSs is addressed. The output of the controlled system is represented by the position and the attitude of the vehicle, together with the position of the manipulator's joints. Remarkably, the unit quaternion is used to express the orientation of a vehicle-fixed frame so as to avoid representation singularities when expressing the vehicle attitude.

The new control law here discussed is inspired by the work in [55] in that a model-based control law is designed together with a nonlinear observer for velocity estimation; the two structures are tuned to each other in order to achieve exponential convergence to zero of both motion tracking and estimation errors. It must be remarked that differently from the work in [55], where a simple time-derivative relates position and velocity variables at the joints, in the control problem considered in this chapter, a nonlinear mapping exists between orientation variables (unit quaternion) and angular velocity of the vehicle; this makes the extension of the previous approach to our case not straightforward.

As a matter of fact, the use of numerical differentiation of noisy position/orientation measurements may lead to chattering of the control inputs, and thus to high energy consumption and reduced lifetime of the actuators. Moreover, low-pass filtering of the numerically reconstructed velocities may significantly degrade the system's dynamic behavior and, eventually, affect the closed-loop stability. In other words, such a filter has to be designed together with the controller so as to preserve closed-loop stability and good tracking performance.

This is the basic idea which inspired the approach described in the following: namely, a nonlinear filter (observer) on the position and attitude measures is designed together with a model-based controller so as to achieve exponential stability and ensure tracking of the desired position and attitude trajectories.

A Lyapunov stability analysis is developed to establish sufficient conditions on the control and observer parameters ensuring exponential convergence of tracking and estimation errors.

In view of the limited computational power available in real-time digital control hardware, simplified control laws are suggested aimed at suitably trading-off tracking performance against reduced computational load. Also, the problem of evaluating some dynamic compensation terms, to be properly estimated, is addressed.

A simulation case study is carried out to demonstrate practical application of the discussed control scheme to the experimental vehicle NPS AUV Phoenix [49]. The obtained performance is compared to that achieved with a control scheme in which velocity is reconstructed via numerical differentiation of position measurements.

Controller-observer scheme. The desired position for the vehicle is assigned in terms of the vector $\boldsymbol{\eta}_{1,d}(t)$, while the commanded attitude trajectory can be assigned in terms of the rotation matrix $\boldsymbol{R}^I_{B,d}(t)$ expressing the orientation of the desired vehicle frame Σ_d with respect to Σ_i. Equivalently, the desired orientation can be expressed in terms of the unit quaternion $\mathcal{Q}_d(t)$ corresponding to $\boldsymbol{R}^I_{B,d}(t)$. Finally, the desired joint motion is assigned in terms of the vector of joint variables $\boldsymbol{q}_d(t)$.

The desired velocity vectors are denoted by $\dot{\boldsymbol{\eta}}_{1,d}(t)$, $\boldsymbol{\nu}^I_{2,d}(t)$, and $\dot{\boldsymbol{q}}_d(t)$, while the desired accelerations are assigned in terms of the vectors $\ddot{\boldsymbol{\eta}}_{1,d}(t)$, $\dot{\boldsymbol{\nu}}^I_{2,d}(t)$, and $\ddot{\boldsymbol{q}}_d(t)$.

Notice that all the desired quantities are naturally assigned with respect to the earth-fixed frame Σ_i; the corresponding position and velocity in the vehicle-fixed frame Σ_b are computed as

$$\boldsymbol{\eta}^B_{1,d} = \boldsymbol{R}^B_I \boldsymbol{\eta}_{1,d}, \quad \boldsymbol{\zeta}_d = \begin{bmatrix} \boldsymbol{R}^B_I \dot{\boldsymbol{\eta}}_{1,d} \\ \boldsymbol{R}^B_I \boldsymbol{\nu}^I_{2,d} \\ \dot{\boldsymbol{q}}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{\nu}_{1,d} \\ \boldsymbol{\nu}_{2,d} \\ \dot{\boldsymbol{q}}_d \end{bmatrix} .$$

It is worth pointing out that the computation of the desired acceleration $\dot{\boldsymbol{\zeta}}_d$ requires knowledge of the actual angular velocity $\boldsymbol{\nu}_2$; in fact, in view of $\dot{\boldsymbol{R}}^B_I = -\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{R}^B_I$, it is

$$\dot{\boldsymbol{\zeta}}_d = \begin{bmatrix} \boldsymbol{R}^B_I \ddot{\boldsymbol{\eta}}_{1,d} - \boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{\nu}_{1,d} \\ \boldsymbol{R}^B_I \dot{\boldsymbol{\nu}}^I_{2,d} - \boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{\nu}_{2,d} \\ \ddot{\boldsymbol{q}}_d \end{bmatrix} .$$

Hence, it is convenient to use in the control law the modified acceleration vector defined as

$$\boldsymbol{a}_d = \begin{bmatrix} \boldsymbol{R}^B_I \ddot{\boldsymbol{\eta}}_{1,d} - \boldsymbol{S}(\boldsymbol{\nu}_{2,d})\boldsymbol{\nu}_{1,d} \\ \boldsymbol{R}^B_I \dot{\boldsymbol{\nu}}^I_{2,d} - \boldsymbol{S}(\boldsymbol{\nu}_{2,d})\boldsymbol{\nu}_{2,d} \\ \ddot{\boldsymbol{q}}_d \end{bmatrix} ,$$

which can be evaluated without using the actual velocity; the two vectors are related by the equality

$$\dot{\boldsymbol{\zeta}}_d = \boldsymbol{a}_d + \boldsymbol{S}_{PO}(\widetilde{\boldsymbol{\nu}}_{2,d})\boldsymbol{\zeta}_d ,$$

where $\boldsymbol{S}_{PO}(\cdot) = \text{blockdiag}\{\boldsymbol{S}(\cdot), \boldsymbol{S}(\cdot), \boldsymbol{O}_{n\times n}\}$, and $\widetilde{\boldsymbol{\nu}}_{2,d} = \boldsymbol{\nu}_{2,d} - \boldsymbol{\nu}_2$.

Hereafter it is assumed that $\left\|\boldsymbol{\zeta}_d(t)\right\| \leq \zeta_{dM}$ for all $t \geq 0$.

A tracking control law is naturally based on the *tracking error*

$$\boldsymbol{e}_d = \begin{bmatrix} \widetilde{\boldsymbol{\eta}}^B_{1,d} \\ \widetilde{\boldsymbol{\varepsilon}}_d \\ \widetilde{\boldsymbol{q}}_d \end{bmatrix}, \tag{7.34}$$

where $\widetilde{\boldsymbol{\eta}}^B_{1,d} = \boldsymbol{\eta}^B_{1,d} - \boldsymbol{\eta}^B_1$, $\widetilde{\boldsymbol{q}}_d = \boldsymbol{q}_d - \boldsymbol{q}$ and $\widetilde{\boldsymbol{\varepsilon}}_d$ is the vector part of the unit quaternion $\widetilde{\mathcal{Q}}_d = \mathcal{Q}^{-1} * \mathcal{Q}_d$.

It must be noticed that a derivative control action based on (7.34) would require velocity measurements in the control loop. In the absence of velocity measurements, a suitable estimate ζ_e of the velocity vector has to be considered. Let also $\boldsymbol{\eta}^B_{1,e}$ and $\mathcal{Q}_e$ denote the estimated position and attitude of the vehicle, respectively; the estimated joint variables are denoted by $\boldsymbol{q}_e$. Hence, the following error vector has to be considered

$$\boldsymbol{e}_{de} = \begin{bmatrix} \widetilde{\boldsymbol{\eta}}^B_{1,de} \\ \widetilde{\boldsymbol{\varepsilon}}^e_{de} \\ \widetilde{\boldsymbol{q}}_{de} \end{bmatrix}, \tag{7.35}$$

where $\widetilde{\boldsymbol{\eta}}^B_{1,de} = \boldsymbol{\eta}^B_{1,d} - \boldsymbol{\eta}^B_{1,e}$, $\widetilde{\boldsymbol{q}}_{de} = \boldsymbol{q}_d - \boldsymbol{q}_e$, and $\widetilde{\boldsymbol{\varepsilon}}^e_{de}$ is the vector part of the unit quaternion $\widetilde{\mathcal{Q}}_{de} = \mathcal{Q}_e^{-1} * \mathcal{Q}_d$.

In order to avoid direct velocity feedback, the corresponding velocity error can be defined as

$$\widetilde{\zeta}_{de} = \begin{bmatrix} \boldsymbol{R}^B_I \dot{\widetilde{\boldsymbol{\eta}}}_{1,de} - \boldsymbol{S}(\boldsymbol{\nu}_{2,d}) \widetilde{\boldsymbol{\eta}}^B_{1,de} \\ \dot{\widetilde{\boldsymbol{\varepsilon}}}^e_{de} \\ \dot{\widetilde{\boldsymbol{q}}}_{de} \end{bmatrix},$$

which is related to the time derivative of $\boldsymbol{e}_{de}$ as follows

$$\dot{\boldsymbol{e}}_{de} = \widetilde{\zeta}_{de} + \boldsymbol{S}_P(\widetilde{\boldsymbol{\nu}}_{2,d}) \boldsymbol{e}_{de},$$

where $\boldsymbol{S}_P(\cdot) = \text{blockdiag}\{\boldsymbol{S}(\cdot), \boldsymbol{O}_3, \boldsymbol{O}_{n \times n}\}$.

In order to design an observer providing velocity estimates, the *estimation error* has to be considered

$$\boldsymbol{e}_e = \begin{bmatrix} \widetilde{\boldsymbol{\eta}}^B_{1,e} \\ \widetilde{\boldsymbol{\varepsilon}}_e \\ \widetilde{\boldsymbol{q}}_e \end{bmatrix},$$

where $\widetilde{\boldsymbol{\eta}}^B_{1,e} = \boldsymbol{\eta}^B_{1,e} - \boldsymbol{\eta}^B_1$, $\widetilde{\boldsymbol{q}}_e = \boldsymbol{q}_e - \boldsymbol{q}$, and $\widetilde{\boldsymbol{\varepsilon}}_e$ is the vector part of the unit quaternion $\widetilde{\mathcal{Q}}_e = \mathcal{Q}^{-1} * \mathcal{Q}_e$.

Finally, consider the vectors

$$\zeta_r = \zeta_d + \boldsymbol{\Lambda}_d \boldsymbol{e}_{de} \tag{7.36}$$

$$\zeta_o = \zeta_e + \boldsymbol{\Lambda}_e \boldsymbol{e}_e. \tag{7.37}$$

where $\boldsymbol{\Lambda}_d = \text{blockdiag}\{\boldsymbol{\Lambda}_{dP}, \lambda_{dO}\boldsymbol{I}_3, \boldsymbol{\Lambda}_{dQ}\}$, $\boldsymbol{\Lambda}_e = \text{blockdiag}\{\boldsymbol{\Lambda}_{eP}, \lambda_{eO}\boldsymbol{I}_3, \boldsymbol{\Lambda}_{eQ}\}$ are diagonal and positive definite matrices. It is worth remarking that $\boldsymbol{\zeta}_r$ and $\boldsymbol{\zeta}_o$ can be evaluated without using the actual velocity $\boldsymbol{\zeta}$.

Let us recall the dynamic equations in matrix form (2.77):

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) = \boldsymbol{B}\boldsymbol{u}, \tag{2.77}$$

the control law is

$$\boxed{\begin{aligned} \boldsymbol{u} = \boldsymbol{B}^{\dagger}\big[&\boldsymbol{M}(\boldsymbol{q})\boldsymbol{a}_r + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta}_o)\boldsymbol{\zeta}_r\boldsymbol{K}_v(\boldsymbol{\zeta}_r - \boldsymbol{\zeta}_o) \\ &+ \boldsymbol{K}_p\boldsymbol{e}_d + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) + \tfrac{1}{2}\boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta}_r)(\boldsymbol{\zeta}_r + \boldsymbol{\zeta}_o)\big], \end{aligned}} \tag{7.38}$$

where $\boldsymbol{K}_p = \text{blockdiag}\{k_{pP}\boldsymbol{I}_3, k_{pO}\boldsymbol{I}_3, \boldsymbol{K}_{pQ}\}$ is a diagonal positive definite matrix and $\boldsymbol{K}_v$ is a symmetric positive definite matrix. The reference acceleration vector $\boldsymbol{a}_r$ is defined as

$$\boldsymbol{a}_r = \boldsymbol{a}_d + \boldsymbol{\Lambda}_d\widetilde{\boldsymbol{\zeta}}_{de}\,, \tag{7.39}$$

and thus the control law (7.38) does not require feedback of the vehicle and/or manipulator velocities.

The estimated velocity vector $\boldsymbol{\zeta}_e$ is obtained via the observer defined by the equations:

$$\boxed{\left\{\begin{aligned} \dot{z} &= \boldsymbol{M}(\boldsymbol{q})\boldsymbol{a}_r - \big(\boldsymbol{L}_p + \boldsymbol{L}_v\boldsymbol{A}(\widetilde{\mathcal{Q}}_e)\boldsymbol{\Lambda}_e\big)\boldsymbol{e}_e + \boldsymbol{K}_p\boldsymbol{e}_d \\ &\quad + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta}_o)\boldsymbol{\zeta}_r + \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q}, \boldsymbol{\zeta}_r)\boldsymbol{\zeta}_o \\ \boldsymbol{\zeta}_e &= \boldsymbol{M}^{-1}(\boldsymbol{q})\,(z - \boldsymbol{L}_v\boldsymbol{e}_e) - \boldsymbol{\Lambda}_e\boldsymbol{e}_e, \end{aligned}\right.} \tag{7.40}$$

where the matrix $\boldsymbol{L}_p = \text{blockdiag}\{l_{pP}\boldsymbol{I}_3, l_{pO}\boldsymbol{I}_3, \boldsymbol{L}_{pQ}\}$ is diagonal positive definite. The matrix $\boldsymbol{L}_v = \text{blockdiag}\{\boldsymbol{L}_{vP}, l_{vO}\boldsymbol{I}_3, \boldsymbol{L}_{vQ}\}$ is symmetric and positive definite, and

$$\boldsymbol{A}(\widetilde{\mathcal{Q}}_e) = \text{blockdiag}\left\{\boldsymbol{I}_3,\ \boldsymbol{E}(\widetilde{\mathcal{Q}}_e)/2,\ \boldsymbol{I}_n\right\}.$$

The estimated quantities $\boldsymbol{\eta}_{1,e}$ and $\boldsymbol{q}_e$ are computed by integrating the corresponding estimated velocities $\dot{\boldsymbol{\eta}}_{1,e} = \boldsymbol{R}_B^I\boldsymbol{\nu}_{1,e}$ and $\dot{\boldsymbol{q}}_e$, respectively, whereas the estimated orientation $\mathcal{Q}_e$ is computed from the estimated angular velocity $\boldsymbol{\nu}_{2,e}^I = \boldsymbol{R}_B^I\boldsymbol{\nu}_{2,e}$ via the quaternion propagation rule.

Implementation issues. Implementation of the above controller-observer scheme (7.38), (7.40), requires computation of the dynamic compensation terms. While this can be done quite effectively for the terms related to rigid body dynamics, the terms related to hydrodynamic effects are usually affected by some degree of approximation and/or uncertainty. Besides the use of adaptive control schemes aimed at on-line estimation of relevant model parameters, e.g. [1, 56, 57], it is important to have an estimate of the main hydrodynamic coefficients.

An estimate of the added mass coefficients can be obtained via strip theory [58]. A rough approximation of the hydrodynamic damping is obtained by considering only the linear skin friction and the drag generalized forces.

Another important point concerns the computational complexity associated with dynamic compensation against the limited computing power typically available on board. This might suggest the adoption of a control law computationally lighter than the one derived above. A reasonable compromise between tracking performance and computational burden is achieved if the compensation of Coriolis, centripetal and damping terms is omitted resulting in the controller

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}\big(\boldsymbol{M}(\boldsymbol{q})\boldsymbol{a}_r + \boldsymbol{K}_v(\zeta_r - \zeta_o) + \boldsymbol{K}_p\boldsymbol{e}_d + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I)\big)\,, \tag{7.41}$$

with the simplified observer

$$\begin{cases} \dot{z} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{a}_r - \big(\boldsymbol{L}_p + \boldsymbol{L}_v\boldsymbol{A}(\widetilde{\mathcal{Q}}_e)\boldsymbol{\Lambda}_e\big)\boldsymbol{e}_e + \boldsymbol{K}_p\boldsymbol{e}_d \\ \zeta = \boldsymbol{M}^{-1}(\boldsymbol{q})\,(z - \boldsymbol{L}_v\boldsymbol{e}_e) - \boldsymbol{\Lambda}_e\boldsymbol{e}_e, \end{cases} \tag{7.42}$$

The computational load can be further reduced if a suitable constant diagonal inertia matrix $\widehat{\boldsymbol{M}}$ is used in lieu of the matrix $\boldsymbol{M}(\boldsymbol{q})$, i.e.,

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}\big(\widehat{\boldsymbol{M}}\boldsymbol{a}_r + \boldsymbol{K}_v(\zeta_r - \zeta_o) + \boldsymbol{K}_p\boldsymbol{e}_d + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I)\big)\,, \tag{7.43}$$

with the observer

$$\begin{cases} \dot{z} = \widehat{\boldsymbol{M}}\boldsymbol{a}_r - \big(\boldsymbol{L}_p + \boldsymbol{L}_v\boldsymbol{A}(\widetilde{\mathcal{Q}}_e)\boldsymbol{\Lambda}_e\big)\boldsymbol{e}_e + \boldsymbol{K}_p\boldsymbol{e}_d \\ \widehat{\zeta} = \widehat{\boldsymbol{M}}^{-1}\,(z - \boldsymbol{L}_v\boldsymbol{e}_e) - \boldsymbol{\Lambda}_e\boldsymbol{e}_e\,, \end{cases} \tag{7.44}$$

Table 7.3 shows the computational load of each control law, in terms of required floating point operations, in the case of a 6-DOF vehicle equipped with a 3-DOF manipulator. Where required, inversion of the inertia matrix has been obtained via the Cholesky factorization since $\boldsymbol{M}(\boldsymbol{q})$ is symmetric and positive definite; of course, the inverse of the constant matrix $\widehat{\boldsymbol{M}}$ is computed once off-line. As shown by the results in Table 7.3, the computational load is reduced by about 80 % when the control law (7.43), (7.44) is considered.

Table 7.3 Computational burden for different output feedback controllers

	Mult/div	Add/sub
Control law (7.38), (7.40)	1831	1220
Control law (7.41), (7.42)	1216	849
Control law (7.43), (7.44)	354	147

7.8.1 Stability Analysis

In order to derive the closed-loop dynamic equations, it is useful to define the variables

$$\boldsymbol{\sigma}_d = \boldsymbol{\zeta}_r - \boldsymbol{\zeta} = \widetilde{\boldsymbol{\zeta}}_d + \boldsymbol{\Lambda}_d \boldsymbol{e}_{de} \tag{7.45}$$

$$\boldsymbol{\sigma}_e = \boldsymbol{\zeta}_o - \boldsymbol{\zeta} = \widetilde{\boldsymbol{\zeta}}_e + \boldsymbol{\Lambda}_e \boldsymbol{e}_e \,, \tag{7.46}$$

where

$$\widetilde{\boldsymbol{\zeta}}_d = \boldsymbol{\zeta}_d - \boldsymbol{\zeta} \tag{7.47}$$

$$\widetilde{\boldsymbol{\zeta}}_e = \boldsymbol{\zeta}_e - \boldsymbol{\zeta} \,. \tag{7.48}$$

Combining (2.77) with the control law (7.38), (7.39), and using the equality

$$\boldsymbol{a}_r = \dot{\boldsymbol{\zeta}}_r + \boldsymbol{S}_{PO}(\widetilde{\boldsymbol{\nu}}_{2,d})\boldsymbol{\zeta}_d + \boldsymbol{\Lambda}_d \boldsymbol{S}_P(\widetilde{\boldsymbol{\nu}}_{2,d})\boldsymbol{e}_{de} \,,$$

the tracking error dynamics can be derived

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\sigma}}_d + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\sigma}_d + \boldsymbol{K}_v\boldsymbol{\sigma}_d - \boldsymbol{K}_p\boldsymbol{e}_d& \\
= \boldsymbol{K}_v\boldsymbol{\sigma}_e - \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\sigma}_e)\boldsymbol{\zeta}_r - \boldsymbol{M}(\boldsymbol{q})\boldsymbol{S}_{PO}(\widetilde{\boldsymbol{\nu}}_{2,d})\boldsymbol{\zeta}_d - \boldsymbol{M}(\boldsymbol{q})\boldsymbol{\Lambda}_d\boldsymbol{S}_P(\widetilde{\boldsymbol{\nu}}_{2,d})\boldsymbol{e}_{de}& \\
+ \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} - \frac{1}{2}\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)(\boldsymbol{\zeta}_r + \boldsymbol{\zeta}_o) \,.&
\end{aligned} \tag{7.49}$$

The observer equation (7.40), together with (7.49), yields the estimation error dynamics

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\sigma}}_e + \left(\boldsymbol{L}_v\boldsymbol{A}(\widetilde{\mathcal{Q}}_e) - \boldsymbol{K}_v\right)\boldsymbol{\sigma}_e - \boldsymbol{L}_p\boldsymbol{e}_e& \\
= -\boldsymbol{K}_v\boldsymbol{\sigma}_d - \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\sigma}_e& \\
+ \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q},\boldsymbol{\sigma}_d)\boldsymbol{\zeta}_o + \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta}& \\
- \frac{1}{2}\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)(\boldsymbol{\zeta}_r + \boldsymbol{\zeta}_o) \,.&
\end{aligned} \tag{7.50}$$

A state vector for the closed-loop system (7.49), (7.50) is then

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{\sigma}_d \\ \boldsymbol{e}_d \\ \boldsymbol{\sigma}_e \\ \boldsymbol{e}_e \end{bmatrix} .$$

Notice that perfect tracking of the desired motion together with exact estimate of the system velocities results in $\underline{\boldsymbol{x}} = \boldsymbol{0}$. Therefore, the control objective is fulfilled if the closed loop system (7.49), (7.50) is asymptotically stable at the origin of its state space. This is ensured by the following theorem:

Theorem *There exists a choice of the controller gains $\boldsymbol{K}_p$, $\boldsymbol{K}_v$, $\boldsymbol{\Lambda}_d$ and of the observer parameters $\boldsymbol{L}_p$, $\boldsymbol{L}_v$, $\boldsymbol{\Lambda}_e$ such that the origin of the state space of system* (7.49), (7.50) *is locally exponentially stable.*

Consider the positive definite Lyapunov function candidate

$$\begin{aligned}V = &\ \frac{1}{2}\boldsymbol{\sigma}_d{}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{\sigma}_d + \frac{1}{2}\boldsymbol{\sigma}_e{}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{\sigma}_e \\ &+ \frac{1}{2}k_{pP}\widetilde{\boldsymbol{\eta}}_{1,d}^{B\mathrm{T}}\widetilde{\boldsymbol{\eta}}_{1,d}^{B} + k_{pO}\left((1-\widetilde{\eta}_d)^2 + \widetilde{\varepsilon}_d{}^{\mathrm{T}}\widetilde{\varepsilon}_d\right) + \frac{1}{2}\widetilde{\boldsymbol{q}}_d{}^{\mathrm{T}}\boldsymbol{K}_{pQ}\widetilde{\boldsymbol{q}}_d \\ &+ \frac{1}{2}l_{pP}\widetilde{\boldsymbol{\eta}}_{1,e}^{B\mathrm{T}}\widetilde{\boldsymbol{\eta}}_{1,e}^{B} + l_{pO}\left((1-\widetilde{\eta}_e)^2 + \widetilde{\varepsilon}_e{}^{\mathrm{T}}\widetilde{\varepsilon}_e\right) + \frac{1}{2}\widetilde{\boldsymbol{q}}_e{}^{\mathrm{T}}\boldsymbol{L}_{pQ}\widetilde{\boldsymbol{q}}_e \,. \end{aligned} \tag{7.51}$$

The time derivative of V along the trajectories of the closed-loop system (7.49), (7.50) is given by

$$\begin{aligned}\dot{V} = &-\boldsymbol{\sigma}_d{}^{\mathrm{T}}\boldsymbol{K}_v\boldsymbol{\sigma}_d - \boldsymbol{e}_{de}{}^{\mathrm{T}}\boldsymbol{\Lambda}_d\boldsymbol{K}_p\boldsymbol{e}_d - \boldsymbol{e}_e{}^{\mathrm{T}}\boldsymbol{\Lambda}_e\boldsymbol{L}_p\boldsymbol{e}_e \\ &-\boldsymbol{\sigma}_e{}^{\mathrm{T}}\left(\boldsymbol{L}_v\boldsymbol{A}(\widetilde{\mathcal{Q}}_e) - \boldsymbol{K}_v\right)\boldsymbol{\sigma}_e - \boldsymbol{\sigma}_d{}^{\mathrm{T}}\boldsymbol{C}(\boldsymbol{q},\boldsymbol{\sigma}_e)\boldsymbol{\zeta}_r \\ &-\boldsymbol{\sigma}_e{}^{\mathrm{T}}\boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\sigma}_e + \boldsymbol{\sigma}_e{}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q},\boldsymbol{\sigma}_d)\boldsymbol{\zeta}_o \\ &+(\boldsymbol{\sigma}_d+\boldsymbol{\sigma}_e)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} - \frac{1}{2}(\boldsymbol{\sigma}_d+\boldsymbol{\sigma}_e)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)(\boldsymbol{\zeta}_r+\boldsymbol{\zeta}_o) \\ &-\boldsymbol{\sigma}_d{}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{S}_{PO}(\widetilde{\boldsymbol{\nu}}_{2,d})\boldsymbol{\zeta}_d - \boldsymbol{\sigma}_d{}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{\Lambda}_d\boldsymbol{S}_P(\widetilde{\boldsymbol{\nu}}_{2,d})\boldsymbol{e}_{de} \,. \end{aligned} \tag{7.52}$$

In the following it is assumed that $\widetilde{\eta}_d > 0$, $\widetilde{\eta}_e > 0$; in view of the angle/axis interpretation of the unit quaternion, the above assumption corresponds to considering orientation errors characterized by angular displacements in the range $]-\pi, \pi[$.

From the equality $\widetilde{\mathcal{Q}}_{de} = \widetilde{\mathcal{Q}}_e^{-1} * \widetilde{\mathcal{Q}}_d$, the following equality follows

$$\widetilde{\varepsilon}_{de}^{eT}\widetilde{\varepsilon}_d = \widetilde{\eta}_e\widetilde{\varepsilon}_d{}^{\mathrm{T}}\widetilde{\varepsilon}_d - \widetilde{\eta}_d\widetilde{\varepsilon}_d{}^{\mathrm{T}}\widetilde{\varepsilon}_e \,,$$

where $\widetilde{\eta}_d$ and $\widetilde{\eta}_e$ are the scalar parts of the quaternions $\widetilde{\mathcal{Q}}_d$ and $\widetilde{\mathcal{Q}}_e$, respectively. The above equation, in view of $\widetilde{\boldsymbol{\eta}}_{1,de}^{B} = \widetilde{\boldsymbol{\eta}}_{1,d}^{B} - \widetilde{\boldsymbol{\eta}}_{1,e}^{B}$ and $\widetilde{\boldsymbol{q}}_{de} = \widetilde{\boldsymbol{q}}_d - \widetilde{\boldsymbol{q}}_e$, implies that

$$\begin{aligned}\boldsymbol{e}_{de}{}^{\mathrm{T}}\boldsymbol{\Lambda}_d\boldsymbol{K}_p\boldsymbol{e}_d \ = &\ k_{pP}\widetilde{\boldsymbol{\eta}}_{1,d}^{B\mathrm{T}}\boldsymbol{\Lambda}_{dP}\widetilde{\boldsymbol{\eta}}_{1,d}^{B} + \lambda_{dO}k_{pO}\widetilde{\eta}_e\left\|\widetilde{\varepsilon}_d\right\|^2 + \widetilde{\boldsymbol{q}}_d{}^{\mathrm{T}}\boldsymbol{\Lambda}_{dQ}\boldsymbol{K}_{pQ}\widetilde{\boldsymbol{q}}_d \\ &-k_{pP}\widetilde{\boldsymbol{\eta}}_{1,d}^{B\mathrm{T}}\boldsymbol{\Lambda}_{dP}\widetilde{\boldsymbol{\eta}}_{1,e}^{B} - \lambda_{dO}k_{pO}\widetilde{\eta}_d\widetilde{\varepsilon}^{\mathrm{T}}{}_d\widetilde{\varepsilon}_e + \widetilde{\boldsymbol{q}}_d{}^{\mathrm{T}}\boldsymbol{\Lambda}_{dQ}\boldsymbol{K}_{pQ}\widetilde{\boldsymbol{q}}_e \,, \end{aligned}$$

and thus

$$\boldsymbol{e}_{de}{}^{\mathrm{T}}\boldsymbol{\Lambda}_d\boldsymbol{K}_p\boldsymbol{e}_d \geq \lambda_{\min}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)\widetilde{\eta}_e\left\|\boldsymbol{e}_d\right\|^2 - \lambda_{\max}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)\left\|\boldsymbol{e}_d\right\|\left\|\boldsymbol{e}_e\right\| \,, \tag{7.53}$$

where $\lambda_{\min}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)$ ($\lambda_{\max}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)$) is the minimum (maximum) eigenvalue of the matrix $\boldsymbol{\Lambda}_d\boldsymbol{K}_p$. Moreover, in view of the block diagonal structure of the matrix $\boldsymbol{L}_v$ and of the skew-symmetry of the matrix $\boldsymbol{S}(\cdot)$, the following inequality holds

$$\boldsymbol{\sigma}_e{}^{\mathrm{T}}\boldsymbol{L}_v\boldsymbol{A}(\widetilde{\mathcal{Q}}_e)\boldsymbol{\sigma}_e \geq \frac{1}{2}\lambda_{\min}(\boldsymbol{\Lambda}_v)\widetilde{\eta}_e\,\|\boldsymbol{\sigma}_e\|^2\ , \tag{7.54}$$

where $\lambda_{\min}(\boldsymbol{\Lambda}_v)$ is the minimum eigenvalue of the matrix $\boldsymbol{\Lambda}_v$.

Moreover, the following two terms in (7.52) can be rewritten as:

$$\begin{aligned}
&(\boldsymbol{\sigma}_d+\boldsymbol{\sigma}_e)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta}-\frac{1}{2}(\boldsymbol{\sigma}_d+\boldsymbol{\sigma}_e)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)(\boldsymbol{\zeta}_r+\boldsymbol{\zeta}_o)\\
&\quad=-\frac{1}{2}(\boldsymbol{\sigma}_d+\boldsymbol{\sigma}_e)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})(\boldsymbol{\sigma}_d+\boldsymbol{\sigma}_e)\\
&\qquad-\frac{1}{2}(\boldsymbol{\sigma}_d+\boldsymbol{\sigma}_e)^{\mathrm{T}}(\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)-\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}))(\boldsymbol{\zeta}_r+\boldsymbol{\zeta}_o)\,.
\end{aligned}\tag{7.55}$$

In view of the properties of the model (2.77) and Eqs. (7.45), (7.46), (7.53), (7.54), by taking into account that $\boldsymbol{\zeta}=\boldsymbol{\zeta}_d-\widetilde{\boldsymbol{\zeta}}_d$ with $\|\boldsymbol{\zeta}_d\|\leq\zeta_{dM}$ and $\|\boldsymbol{e}_{de}\|\leq\|\boldsymbol{e}_d\|+\|\boldsymbol{e}_e\|$, the function $\dot{V}$ can be upper bounded as follows

$$\begin{aligned}
\dot{V}=&-\lambda_{\min}(\boldsymbol{K}_v)\,\|\boldsymbol{\sigma}_d\|^2-\lambda_{\min}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)\widetilde{\eta}_e\,\|\boldsymbol{e}_d\|^2-\frac{\lambda_{\min}(\boldsymbol{\Lambda}_v)}{2}\widetilde{\eta}_e\,\|\boldsymbol{\sigma}_e\|^2\\
&+\lambda_{\max}(\boldsymbol{K}_v)\,\|\boldsymbol{\sigma}_d\|^2-\lambda_{\min}(\boldsymbol{L}_p)\,\|\boldsymbol{e}_e\|^2+\lambda_{\max}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)\,\|\boldsymbol{e}_d\|\,\|\boldsymbol{e}_e\|\\
&+C_M\,\|\boldsymbol{\sigma}_d\|\,\|\boldsymbol{\sigma}_e\|\left(2\left\|\widetilde{\boldsymbol{\zeta}}_d\right\|+2\zeta_{dM}+\|\boldsymbol{\sigma}_d\|+\|\boldsymbol{\sigma}_e\|\right)\\
&+C_M\,\|\boldsymbol{\sigma}_e\|^2\left(\left\|\widetilde{\boldsymbol{\zeta}}_d\right\|+\zeta_{dM}\right)\\
&+\frac{D_M}{2}(\|\boldsymbol{\sigma}_d\|^2+\|\boldsymbol{\sigma}_d\|\,\|\boldsymbol{\sigma}_e\|)(2\left\|\widetilde{\boldsymbol{\zeta}}_d\right\|+2\zeta_{dM}+\|\boldsymbol{\sigma}_d\|+\|\boldsymbol{\sigma}_e\|)\\
&+\lambda_{\max}(\boldsymbol{M})\,\|\boldsymbol{\sigma}_d\|\left\|\widetilde{\boldsymbol{\zeta}}_d\right\|(\zeta_{dM}+\lambda_{\max}(\boldsymbol{\Lambda}_d)(\|\boldsymbol{e}_d\|+\|\boldsymbol{e}_e\|))
\end{aligned}\tag{7.56}$$

where $\lambda_{\min}(\boldsymbol{K}_v)$ ($\lambda_{\max}(\boldsymbol{K}_v)$) denotes the minimum (maximum) eigenvalue of the matrix $\boldsymbol{K}_v$, $\lambda_{\min}(\boldsymbol{L}_p)$ denotes the minimum eigenvalue of $\boldsymbol{L}_p$ and $\lambda_{\max}(\boldsymbol{\Lambda}_d)$ denotes the maximum eigenvalue of the matrix $\boldsymbol{\Lambda}_d$.

Consider the state space domain defined as follows

$$B_\rho=\left\{\underline{\boldsymbol{x}}:\ \left\|\underline{\boldsymbol{x}}\right\|<\rho,\ \rho<1\right\}\ ,$$

with $\widetilde{\eta}_d>0, \widetilde{\eta}_e>0$. It can be recognized that in the domain B_ρ the following inequalities hold

$$0<\sqrt{1-\rho^2}<\widetilde{\eta}_e<1, \tag{7.57}$$

$$\left\|\widetilde{\boldsymbol{\zeta}}_d\right\|=\|\boldsymbol{\sigma}_d-\boldsymbol{\Lambda}_d\boldsymbol{e}_{de}\|\leq(1+2\lambda_{\max}(\boldsymbol{\Lambda}_d))\rho. \tag{7.58}$$

By completing the squares in (7.56) and using (7.57), (7.58), it can be shown that there exists a scalar $\kappa > 0$ such that

$$\dot{V} \leq -\kappa \left\|\underline{\boldsymbol{x}}\right\|^2 \tag{7.59}$$

in the domain B_ρ, provided that the controller and observer parameters satisfy the inequalities

$$\begin{aligned}
\lambda_{\min}(\boldsymbol{K}_v) &> \alpha_1\left(C_M + \frac{3D_M}{2}\right) + \alpha_2\lambda_{\max}(\boldsymbol{M})(1+\lambda_{\max}(\boldsymbol{\Lambda}_d))\\
\lambda_{\min}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p) &> \frac{\alpha_2\lambda_{\max}(\boldsymbol{M})\lambda_{\max}(\boldsymbol{\Lambda}_d)}{\sqrt{1-\rho^2}},\\
\lambda_{\min}(\boldsymbol{L}_p) &> \max\left\{\alpha_2\lambda_{\max}(\boldsymbol{M})\lambda_{\max}(\boldsymbol{\Lambda}_d),\ \frac{\lambda_{\max}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)^2}{\lambda_{\min}(\boldsymbol{\Lambda}_d\boldsymbol{K}_p)\sqrt{1-\rho^2}}\right\},\\
\lambda_{\min}(\boldsymbol{\Lambda}_v) &> \frac{2}{\sqrt{1-\rho^2}}\left(\lambda_{\max}(\boldsymbol{K}_v) + (2\alpha_1+\rho)C_M + \frac{\alpha_1 D_M}{2}\right)
\end{aligned}$$

where $\alpha_1 = 2(1+\lambda_{\max}(\boldsymbol{\Lambda}_d))\rho + \zeta_{dM}$ and $\alpha_2 = \zeta_{dM} + 2\lambda_{\max}(\boldsymbol{\Lambda}_d)\rho$.

Therefore, given a domain B_ρ characterized by any $\rho < 1$, there always exists a set of observer and controller gains such that $\dot{V} \leq 0$ in B_ρ. Moreover, for $\widetilde{\eta}_d \geq 0$, $\widetilde{\eta}_e \geq 0$ the following inequality holds

$$0 \leq (1-\widetilde{\eta}_d)^2 \leq (1-\widetilde{\eta}_d)(1+\widetilde{\eta}_d) = \|\widetilde{\varepsilon}_d\|^2\ ,$$

and a similar inequality can be written in terms of $\widetilde{\eta}_e$ and $\widetilde{\varepsilon}_e$. Hence, function V can be bounded as

$$c_m \left\|\underline{\boldsymbol{x}}\right\|^2 \leq V(\underline{\boldsymbol{x}}) \leq c_M \left\|\underline{\boldsymbol{x}}\right\|^2\ , \tag{7.60}$$

with

$$\begin{aligned}
c_m &= \frac{1}{2}\min\{\lambda_{\min}(\boldsymbol{M}), \lambda_{\min}(\boldsymbol{K}_p), \lambda_{\min}(\boldsymbol{L}_p)\}\\
c_M &= \frac{1}{2}\max\{\lambda_{\max}(\boldsymbol{M}), 4\lambda_{\max}(\boldsymbol{K}_p), 4\lambda_{\max}(\boldsymbol{L}_p)\}\,,
\end{aligned}$$

where $\lambda_{\min}(\boldsymbol{K}_p)$ ($\lambda_{\max}(\boldsymbol{K}_p)$) is the minimum (maximum) eigenvalue of the matrix $\boldsymbol{K}_p$, $\lambda_{\max}(\boldsymbol{L}_p)$ is the maximum eigenvalue of the matrix $\boldsymbol{L}_p$.

Since $V(t)$ is a decreasing function along the system trajectories, the inequality (7.60) guarantees that, for a given $0 < \rho < 1$, all the trajectories $\underline{\boldsymbol{x}}(t)$ starting in the domain

$$\Omega_\rho = \left\{\underline{\boldsymbol{x}} :\ \left\|\underline{\boldsymbol{x}}\right\| < \rho\sqrt{\frac{c_m}{c_M}}\right\}\ ,$$

remain in the domain B_ρ for all $t > 0$ provided that $\widetilde{\eta}_d(t) > 0, \widetilde{\eta}_e(t) > 0$ for all $t > 0$. The latter condition is fulfilled when $\widetilde{\eta}_d(0)$ and $\widetilde{\eta}_e(0)$ are positive; in fact, $\|\widetilde{\boldsymbol{\varepsilon}}_d\| < \rho < 1$ and $\|\widetilde{\boldsymbol{\varepsilon}}_e\| < \rho < 1$ for all $t > 0$ implies that $\widetilde{\eta}_d(t)$ and $\widetilde{\eta}_e(t)$ cannot change their sign.

Moreover, from (7.59) and (7.60), the convergence in the domain B_ρ is exponential [59], which implies exponential convergence of $\boldsymbol{e}_d$, $\boldsymbol{e}_e$, $\widetilde{\boldsymbol{\zeta}}_d$ and $\widetilde{\boldsymbol{\zeta}}_e$.

The condition $\rho < 1$ is due to the unit norm constraint on the quaternion components, and gives a rather conservative estimate of the domain of attraction. However, it must be pointed out that this limitation arises when spheres are used to estimate the domain of attraction; better estimates can be obtained by using domain of different shapes, e.g. ellipsoids.

7.8.2 Simulations

As for the previous sections, numerical simulations have been performed resorting to SIMURV 4.0 described in Chap. 9. To test this control law, however, a different manipulator has been considered. A three-link manipulator with elbow kinematic structure has been simulated that is mounted under the vehicle body. Since the vehicle is neutrally buoyant, but the arm is not neutrally buoyant, the whole system results to be not neutrally buoyant. The 3 links are cylindrical, thus hydrodynamic effects can be computed by simplified relations as in [26]. Matrix $\boldsymbol{B}$ is supposed to be constant and full-rank (for simplicity it has been set to identity), meaning that direct control of forces and moments acting on the vehicle and joint torques is available.

A task involving motion of both the vehicle and the manipulator has been considered. At the initial time, the initial vehicle configuration is

$$\boldsymbol{\eta}_i = \begin{bmatrix} 0 \; 0 \; 0.1 \; 15 \; 0 \; -15 \end{bmatrix}^{\mathrm{T}} \quad \text{m, deg}$$

and the initial manipulator configuration is

$$\boldsymbol{q}_i = \begin{bmatrix} 20 \; -30 \; 40 \end{bmatrix}^{\mathrm{T}} \quad \text{deg.}$$

The vehicle must move to the final location

$$\boldsymbol{\eta}_f = \begin{bmatrix} 0 \; 0 \; 0 \; 0 \; 0 \; 0 \end{bmatrix}^{\mathrm{T}} \quad \text{m, deg}$$

in 20 s according to a fifth-order polynomial time law. The manipulator must move to

$$\boldsymbol{q}_f = \begin{bmatrix} 0 \; 0 \; 0 \end{bmatrix}^{\mathrm{T}} \quad \text{deg.}$$

in 3 s according to a fifth-order polynomial time law too. Notice that the assigned trajectories correspond to a fast desired motion for the manipulator while the vehicle is kept almost in hovering. Figure 7.13 shows the desired trajectories. It must be

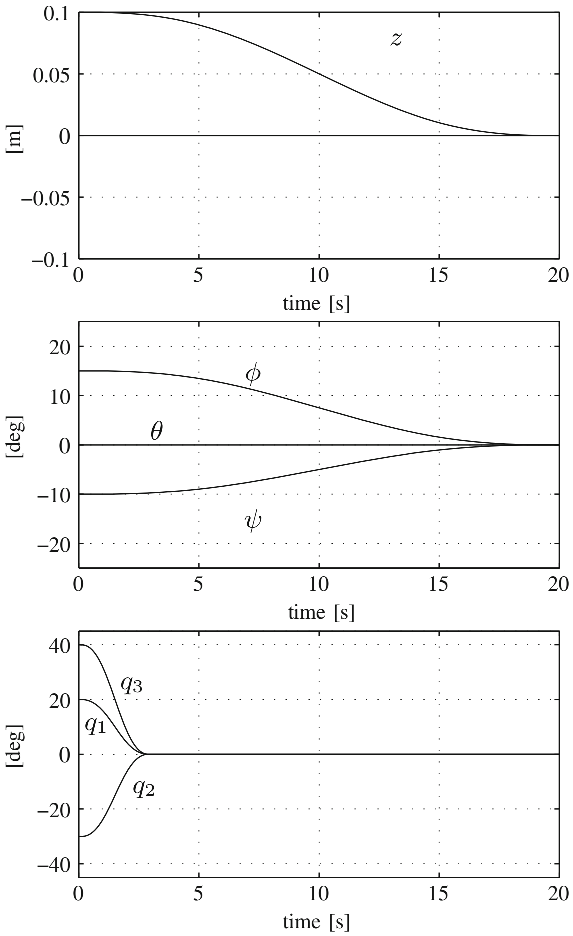


Fig. 7.13 Output feedback control. Desired trajectories used in the case studies. *Top* vehicle position; *Middle* vehicle orientation (RPY angles); *Bottom* joint positions

noticed that the vehicle orientation set point is assigned in terms of Euler angles, as usual in navigation planning; these are converted into the corresponding rotation matrix so as to extract the quaternion expressing the orientation error. Remarkably, this procedure is free of singularities [60].

First case study. The performance of the control law (7.41), (7.42) has been compared to that obtained with a control scheme of similar structure in which the velocity feedback is implemented through numerical differentiation of position measurements. The following control law has then been considered

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}\left(\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}}_r + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta}_r + \boldsymbol{K}_v(\boldsymbol{\zeta}_r - \boldsymbol{\zeta}) + \boldsymbol{K}_p\boldsymbol{e}_d + \boldsymbol{g}(\boldsymbol{q},\boldsymbol{R}_B^I)\right), \tag{7.61}$$

where $\zeta_r = \zeta_d + \Lambda_d e_d$. The vectors $\dot{\zeta}$ and $\dot{\zeta}_r$ are computed via first-order difference. The above control law is analogous to the operational space control law proposed in [61] and extended in [47] in the framework of quaternion-based attitude control. To obtain a control law of computational complexity similar to that of (7.41), the algorithm (7.61) has been modified into the simpler form

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}\left(\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}}_r + \boldsymbol{K}_v(\boldsymbol{\zeta}_r - \boldsymbol{\zeta}) + \boldsymbol{K}_p\boldsymbol{e}_d + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I)\right) . \tag{7.62}$$

The parameters in the control laws are set to

$$\begin{aligned}
\boldsymbol{\Lambda}_d &= \text{blockdiag}\{0.005\boldsymbol{I}_3, 0.01\boldsymbol{I}_3, 0.01\boldsymbol{I}_3\}\,,\\
\boldsymbol{\Lambda}_e &= \text{blockdiag}\{5\boldsymbol{I}_3, 10\boldsymbol{I}_3, 10, 10, 5\}\,,\\
\boldsymbol{L}_v &= \text{blockdiag}\{\boldsymbol{I}_3, 20\boldsymbol{I}_3, 160, 160, 900\}\,,\\
\boldsymbol{L}_p &= \text{blockdiag}\{5000\boldsymbol{I}_3, 10^5\boldsymbol{I}_3, 10^3, 10^3, 2\cdot 10^3\}\,,\\
\boldsymbol{K}_p &= \text{blockdiag}\{400\boldsymbol{I}_3, 500\boldsymbol{I}_3, 1500\boldsymbol{I}_3\}\,,\\
\boldsymbol{K}_v &= \text{blockdiag}\{4000\boldsymbol{I}_3, 4000\boldsymbol{I}_3, 400\boldsymbol{I}_3\}\,.
\end{aligned}$$

A digital implementation of the control laws has been considered. The sensor update rate is 100 Hz for the joint positions and 20 Hz for the vehicle position and orientation, while the control inputs to the actuators are updated at 100 Hz, i.e., the control law is computed every 10 ms. The relatively high update rate for the vehicle position and orientation measurements has been chosen so as to achieve a satisfactory tracking accuracy.

Quantization effects have been introduced in the simulation by assuming a 16-bit A/D converter on the sensors outputs. Also, Gaussian zero-mean noise has been added to the signals coming from the sensors.

Figure 7.14 shows the time history of the norm of the tracking and estimation errors obtained with the control laws (7.41), (7.42) and (7.62), respectively.

Figures 7.15, 7.16 and 7.17 show the corresponding control forces, moments and torques. It can be recognized that good tracking is achieved in both cases, although the performance in terms of tracking errors is slightly better for the control law (7.62), where numerical derivatives are used. On the other hand, the presence of measurement noise and quantization effects results in chattering of the control commands to the actuators; this is much lower when the control law (7.41), (7.42) is adopted, as compared to control law (7.62). An indicator of the energy consumption due to the chattering at steady state is the variance of the control commands reported in Table 7.4 for each component; this data clearly show the advantage of using the controller-observer scheme. Of course, the improvement becomes clear when the noise and quantization effects are larger than a certain threshold. The derivation of such a threshold would require a stochastic analysis of a nonlinear system, which

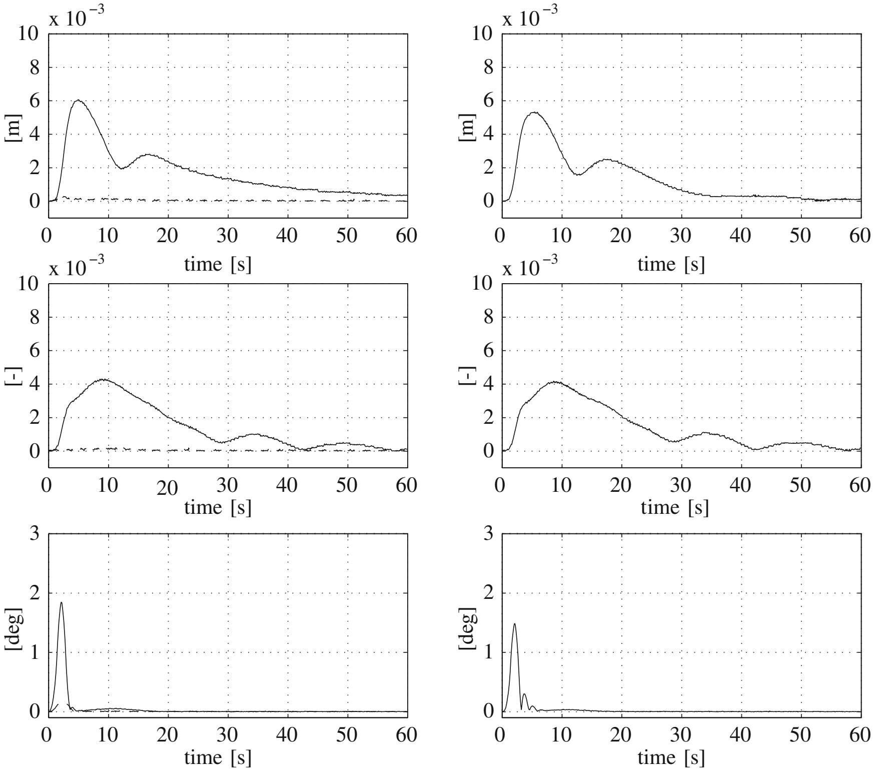


Fig. 7.14 Output feedback control; first case study. Comparison between the control laws (7.41), (7.42) and (7.62): norm of the tracking (*solid*) and estimation (*dashed*) errors. *Left* control law (7.41), (7.42). *Right* control law (7.62). *Top* vehicle position error; *Middle* vehicle orientation error (vector part of the quaternion); *Bottom* joint position errors

is beyond the scope of the present work. Moreover, it can be easily recognized that such a bound is strongly dependent on the characteristics of the actuators.

Second case study. In this case study the same task as above is executed by adopting the control law (7.43), (7.44) and its counterpart using numerical differentiation of the measured position/orientation, i.e.,

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}\left(\widehat{\boldsymbol{M}}\dot{\boldsymbol{\zeta}}_r + \boldsymbol{K}_v(\boldsymbol{\zeta}_r - \boldsymbol{\zeta}) + \boldsymbol{K}_p\boldsymbol{e}_d + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I)\right) , \tag{7.63}$$

where the same parameters as in the previous case study have been used. Also, the same measurement update rates, quantization resolution and sensory noise have been considered in the simulation.

The results are shown in Fig. 7.18 in terms of tracking and estimation errors. It can be recognized that the errors are comparable to those obtained with the control

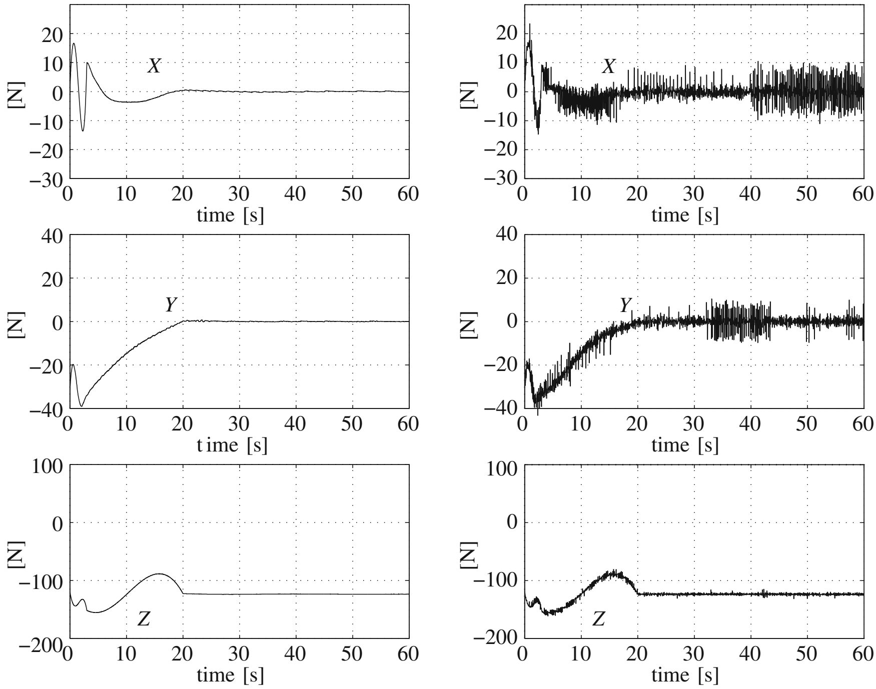


Fig. 7.15 Output feedback control; first case study. Comparison between the control laws (7.41),(7.42) and (7.62): vehicle control forces. *Left* control law (7.41), (7.42). *Right* control law (7.62)

scheme (7.41), (7.42) in spite of the extremely simplified control structure; also, the control inputs remain free of chattering phenomena and are not reported for brevity.

The tracking performance obtained with the simplified control law confirms that the controller-observer approach is intrinsically robust with respect to uncertain knowledge of the system's dynamics, thanks to the exponential stability property. Hence, perfect compensation of inertia, Coriolis and centripetal terms, as well as of hydrodynamic damping terms, is not required.

Third case study. The control laws (7.43), (7.44) and (7.63) have been tested in more severe operating conditions. Namely, the update rate for the vehicle position/orientation measurements has been lowered to 5 Hz and the A/D word length has been set to 12 bit for all the sensor output signals. Gaussian zero-mean noise is still added to the measures. The parameters in the control laws are the same as in the previous case studies.

Figure 7.19 shows a small degradation of the tracking performance for both the control schemes. In fact, the tracking errors remain of the same order of magnitude as in the previous case studies, because the computing rate of the control law is unchanged (100 Hz). Namely, the update rate of the measurements relative to the subsystem with faster dynamics (i.e., the manipulator) remains the same (100 Hz), while

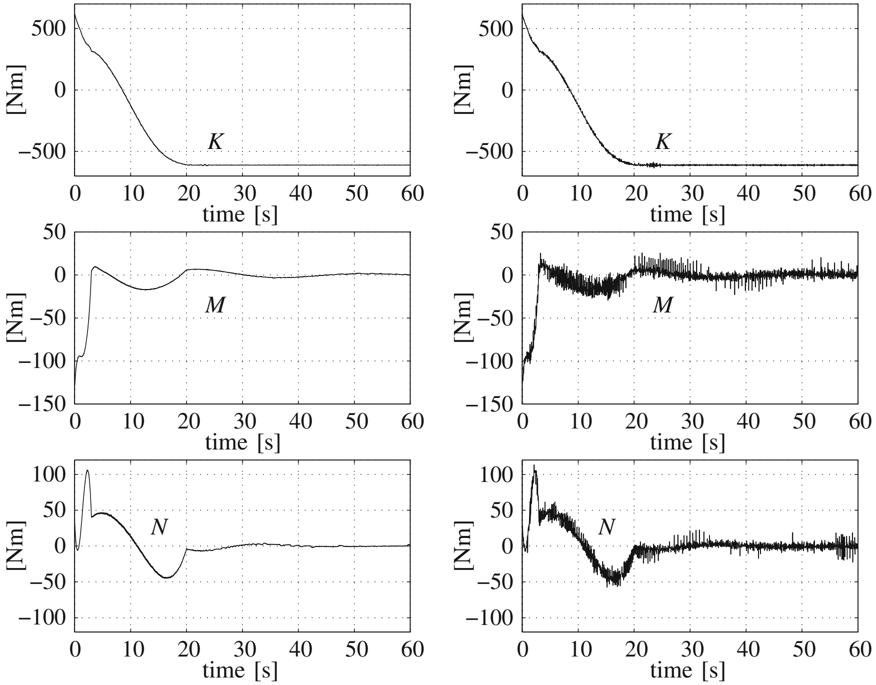


Fig. 7.16 Output feedback control; first case study. Comparison between the control laws (7.41), (7.42) and (7.62): vehicle control moments. *Left* control law (7.41), (7.42); *Right* control law (7.62)

the update rate of the measurements relative to the vehicle (5 Hz) is still adequate to its slower dynamics.

Figures 7.20, 7.21 and 7.22 show the corresponding control forces, moments and torques. It can be recognized that unacceptable chattering on the control inputs is experienced when numerical derivatives are used, which is almost completely cancelled when the controller-observer scheme is adopted.

7.9 Virtual Decomposition Based Control

Divide et Impera.

Anonymous from the middle age.

Usually, adaptive control approaches for UVMSs look at the system as a whole, giving rise to high-dimensional problems: differently from the case of earth-fixed manipulators, in the case of UVMSs it is not possible to achieve a reduction of the number of dynamic parameters to be adapted, since the base of the manipulator—i.e., the vehicle—has full mobility. As a matter of fact, the computational load of such control algorithms grows as much as the fourth-order power of the number of

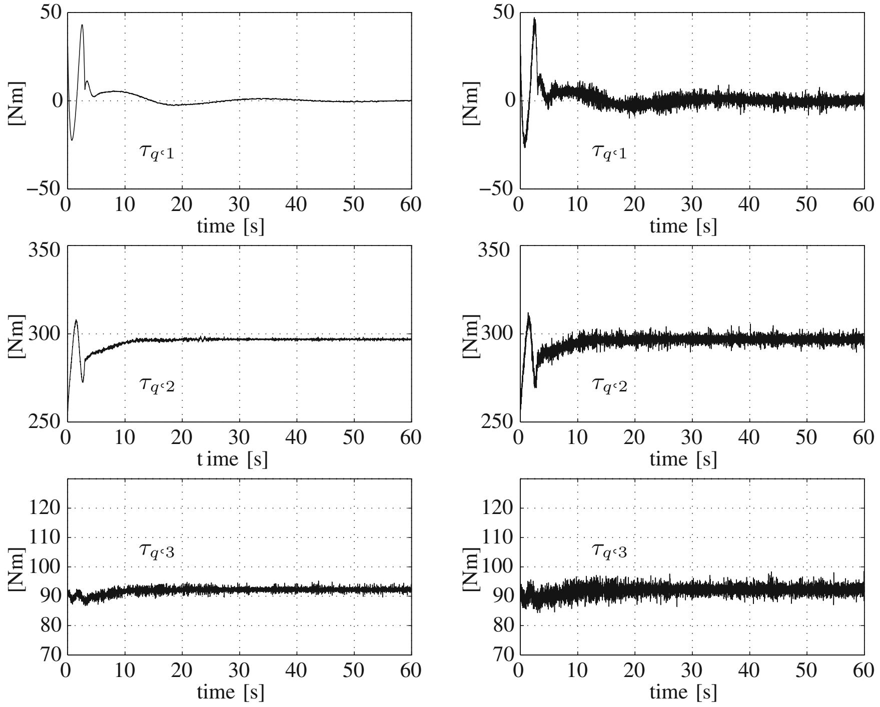


Fig. 7.17 Output feedback control; first case study. Comparison between the control laws (7.41), (7.42) and (7.62): joint control torques. *Left* control law (7.41), (7.42); *Right* control law (7.62)

the system's degrees of freedom. For this reason, practical application of adaptive control to UVMS has been limited, even in simulation, to vehicles carrying arms with very few joints (i.e., two or three) and usually performing planar tasks.

In this section an adaptive control scheme for the tracking problem of UVMS is discussed, which is based on the approach in [62–64] and presented in [65, 66]. Differently from previously proposed schemes, the serial-chain structure of the UVMS is exploited to decompose the overall motion control problem in a set of elementary control problems regarding the motion of each rigid body in the system, namely the manipulator's links and the vehicle. For each body, a control action is designed to assign the desired motion, to adaptively compensate for the body dynamics, and to counteract force/moment exchanged with its neighborhoods along the chain.

The resulting control scheme has a modular structure which greatly simplifies its application to systems with a large number of links; furthermore, it reduces the required computational burden by replacing one high-dimensional problem with many low-dimensional ones; finally it allows efficient implementation on distributed computing architecture; it can be embedded in a kinematic control scheme, which allows handling of kinematic redundancy, i.e., to achieve joint limits avoidance and dexterity optimization; finally, it reduces the size of the control software code and

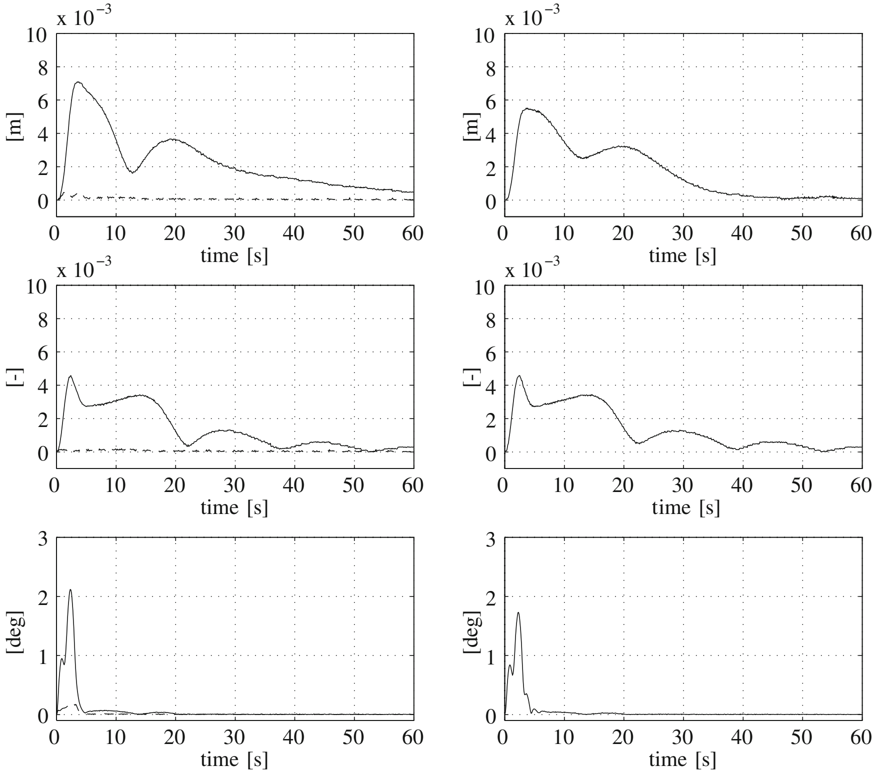


Fig. 7.18 Output feedback control; second case study. Comparison between the control laws (7.43), (7.44) and (7.63): norm of the tracking (*solid*) and estimation (*dashed*) errors. *Top* vehicle position error; *Middle* vehicle orientation error (vector part of the quaternion); *Bottom* joint position errors

Table 7.4 Variance of control commands

	Control law (7.41), (7.42)	Control law (7.62)
X (N^2)	0.0149	14.6326
Y (N^2)	0.0077	11.4734
Z (N^2)	0.0187	3.0139
K (N^2m^2)	0.2142	10.6119
M (N^2m^2)	2.8156	23.2798
N (N^2m^2)	1.6192	26.0350
$\tau_{q,1}$ (N^2m^2)	0.3432	4.0455
$\tau_{q,2}$ (N^2m^2)	0.0725	3.2652
$\tau_{q,3}$ (N^2m^2)	0.3393	1.8259

improves its flexibility, i.e., its structure is not modified by changing the system's mechanical structure.

Remarkably, the control law is expressed in terms of body-fixed coordinates so as to overcome the occurrence of kinematic singularities. Moreover, a non-minimal

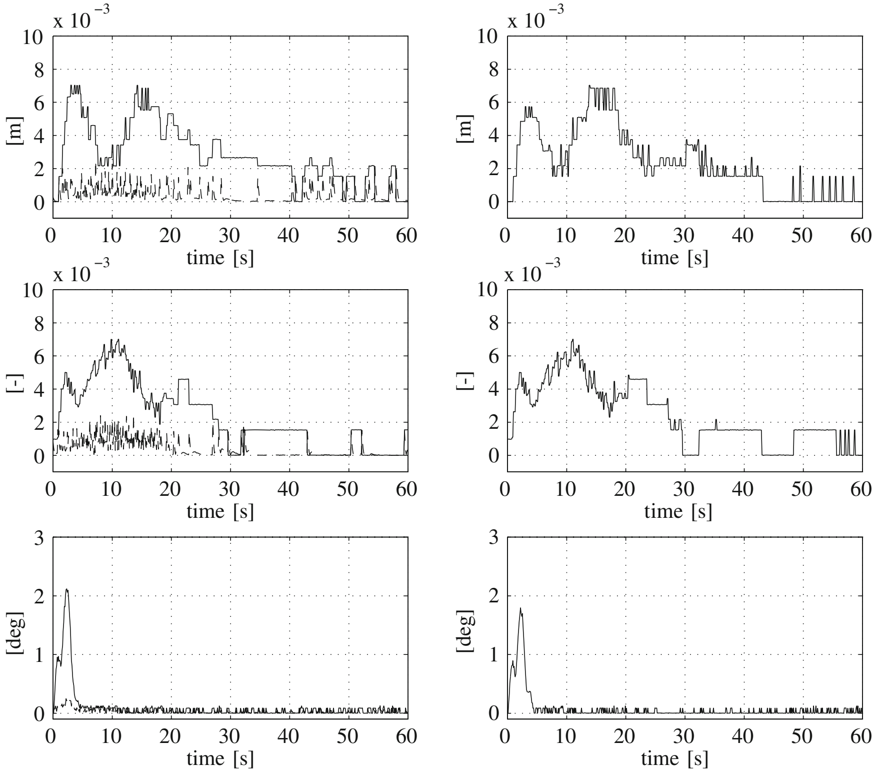


Fig. 7.19 Output feedback control; third case study. Comparison between the control laws (7.43), (7.44) and (7.63): norm of the tracking (*solid*) and estimation (*dashed*) errors. *Left* control law (7.43), (7.44). *Right* control law (7.63). *Top* vehicle position error; *Middle* vehicle orientation error (vector part of the quaternion); *Bottom* joint position errors

representation of the orientation—i.e., the unit quaternion [67]—is used in the control law; this allows overcoming the occurrence of representation singularities.

The discussed control scheme is tested in a numerical case study. A manipulation task is assigned in terms of a desired position and orientation trajectory for the end effector of a 6-DOF manipulator mounted on a 6-DOF vehicle. Then, the system's behavior under the discussed control law is verified in simulation.

Control law. The dynamics of an UVMS is rewritten in a way to remark the interaction between the different rigid bodies, i.e., between the links and between links and the vehicle. Consider an UVMS composed of a vehicle and of a n-DOF manipulator mounted on it.

The vehicle and the manipulator's links are assumed to be rigid bodies numbered from 0 (the vehicle) to n (the last link, i.e., the end effector). Hence, the whole system can be regarded as an open kinematic chain with floating base.

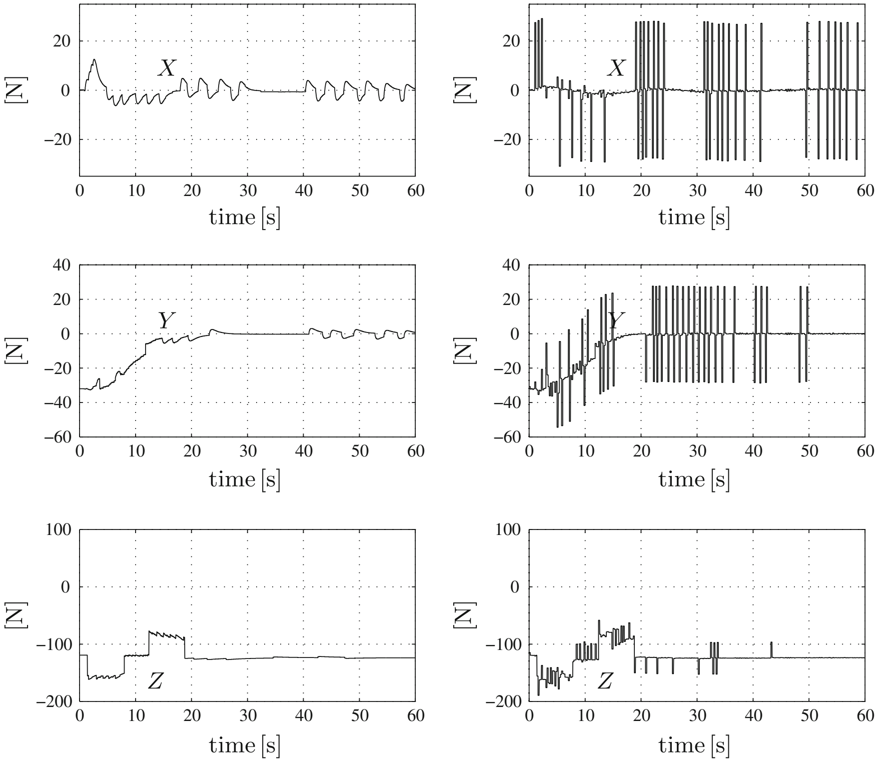


Fig. 7.20 Output feedback control; third case study. Comparison between the control laws (7.43), (7.44) and (7.63): vehicle control forces. *Left* control law (7.43), (7.44). *Right* control law (7.63)

A reference frame $\mathcal{T}_i$ is attached to each body according to the Denavit-Hartenberg formalism, while Σ_i is the earth-fixed inertial reference frame. Hereafter, a superscript will denote the frame to which a vector is referred to, the superscript will be dropped for quantities referred to the inertial frame.

Notice that some differences may arise in the symbology of the vehicle's variables due to the different approach followed in this section. Coherently with the virtual decomposition approach, the vehicle is considered as link number 0.

The (6 × 1) vector of the total generalized force (i.e., force and moment) acting on the ith body is given by

$$\begin{aligned} \boldsymbol{h}^i_{t,i} &= \boldsymbol{h}^i_i - \boldsymbol{U}^i_{i+1}\boldsymbol{h}^{i+1}_{i+1}, \qquad i = 0, \ldots, n-1 \\ \boldsymbol{h}^n_{t,n} &= \boldsymbol{h}^n_n\,, \end{aligned} \tag{7.64}$$

where $\boldsymbol{h}^i_i$ is the generalized force exerted by body $i-1$ on body i, $\boldsymbol{h}^{i+1}_{i+1}$ is the generalized force exerted by body $i+1$ on body i. The matrix $\boldsymbol{U}^i_{i+1} \in \mathbb{R}^{6\times 6}$ is defined as

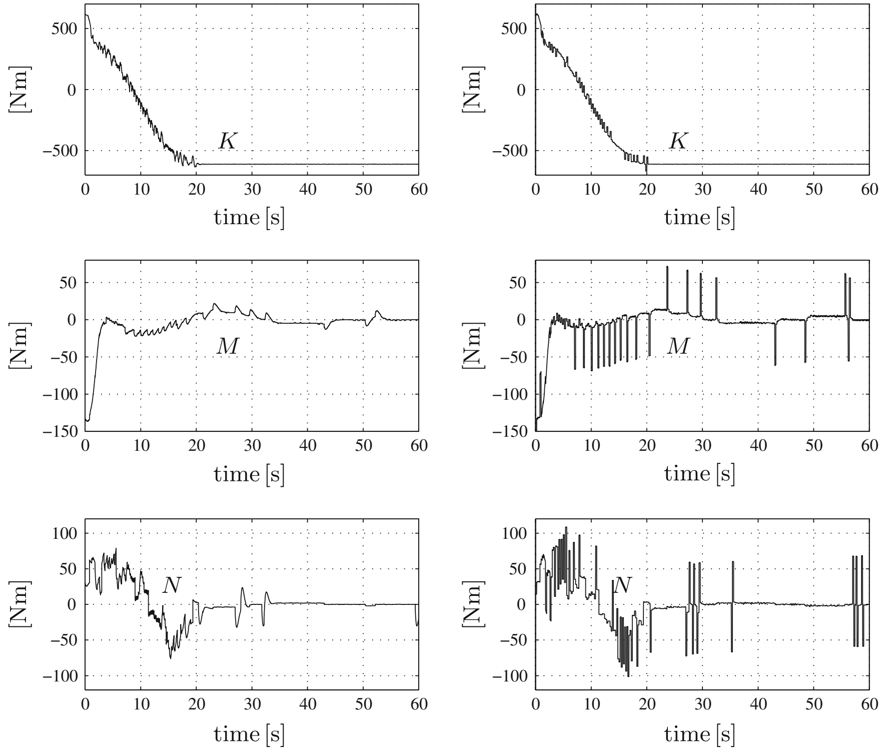


Fig. 7.21 Output feedback control; third case study. Comparison between the control laws (7.43), (7.44) and (7.63): vehicle control moments. *Left* control law (7.43), (7.44). *Right* control law (7.63)

$$\boldsymbol{U}_{i+1}^{i} = \begin{bmatrix} \boldsymbol{R}_{i+1}^{i} & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{S}(\boldsymbol{r}_{i,i+1}^{i})\boldsymbol{R}_{i+1}^{i} & \boldsymbol{R}_{i+1}^{i} \end{bmatrix},$$

where $\boldsymbol{R}_{i+1}^{i} \in \mathbb{R}^{3\times 3}$ is the rotation matrix from frame $\mathcal{T}_{i+1}$ to frame $\mathcal{T}_i$, $\boldsymbol{S}(\cdot)$ is the matrix operator performing the cross product between two (3×1) vectors, and $\boldsymbol{r}_{i,i+1}^{i}$ is the vector pointing from the origin of $\mathcal{T}_i$ to the origin of $\mathcal{T}_{i+1}$.

The equations of motion of each rigid body can be written in body-fixed reference frame in the form [57, 58]:

$$\boldsymbol{M}_i\dot{\boldsymbol{\nu}}_i^i + \boldsymbol{C}_i(\boldsymbol{\nu}_i^i)\boldsymbol{\nu}_i^i + \boldsymbol{D}_i(\boldsymbol{\nu}_i^i)\boldsymbol{\nu}_i^i + \boldsymbol{g}_i(\boldsymbol{R}_i) = \boldsymbol{h}_{t,i}^{i}, \tag{7.65}$$

where $\boldsymbol{\nu}_i^i \in \mathbb{R}^6$ is the vector of generalized velocity (i.e., linear and angular velocities defined in Sect. 2.10), $\boldsymbol{R}_i$ is the rotation matrix expressing the orientation of $\mathcal{T}_i$ with respect to the inertial reference frame, $\boldsymbol{M}_i \in \mathbb{R}^{6\times 6}$, $\boldsymbol{C}_i(\boldsymbol{\nu}_i^i)\boldsymbol{\nu}_i^i \in \mathbb{R}^6$, $\boldsymbol{D}_i(\boldsymbol{\nu}_i^i)\boldsymbol{\nu}_i^i \in \mathbb{R}^6$ and $\boldsymbol{g}_i(\boldsymbol{R}_i) \in \mathbb{R}^6$ are the quantities introduced in (2.54) referred to the generic rigid

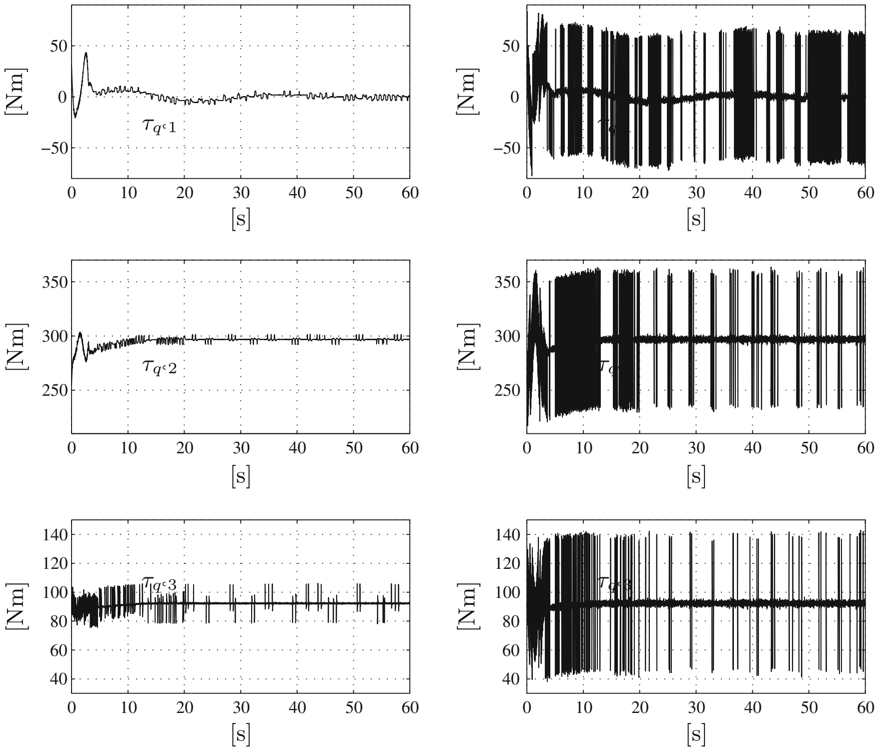


Fig. 7.22 Output feedback control; third case study. Comparison between the control laws (7.43), (7.44) and (7.63): joint control torques. *Left* control law (7.43), (7.44). *Right* control law (7.63)

body. In Chap. 2, the details on the dynamics of a rigid body moving in a fluid are given.

According to the property of linearity in the parameters (7.65) can be rewritten as:

$$\boldsymbol{Y}(\boldsymbol{R}_i, \boldsymbol{\nu}_i^i, \dot{\boldsymbol{\nu}}_i^i)\boldsymbol{\theta}_i = \boldsymbol{h}_{t,i}^i$$

where $\boldsymbol{\theta}_i$ is the vector of dynamic parameters of the ith rigid body. Notice that, for the vehicle, i.e., for the body numbered as 0, the latter is exactly (2.57); only for this section, however, the notation of the vehicle forces and regressor will be slightly different from the rest of the book.

The input torque $\tau_{q,i}$ at the ith joint of the manipulator can be obtained by projecting $\boldsymbol{h}_i$ on the corresponding joint axis via

$$\tau_{q,i} = \boldsymbol{z}_{i-1}^{i\mathrm{T}} \boldsymbol{h}_i^i, \tag{7.66}$$

where $\boldsymbol{z}_{i-1}^i = \boldsymbol{R}_i{}^{\mathrm{T}} \boldsymbol{z}_{i-1}$ is the z-axis of the frame $\mathcal{T}_{i-1}$ expressed in the frame $\mathcal{T}_i$.

The input force and moment acting on vehicle are instead given by the vector $\boldsymbol{h}_{t,0}^{0}$. Notice that this vector was introduced in Sect. 2.6 with the symbol $\boldsymbol{\tau}_v$, in this Section, however, it was preferred to modify the notation consistently with the serial chain formulation adopted here.

Let $\boldsymbol{p}_{d,o}(t)$, $\mathcal{Q}_{d,0}(t)$, $\boldsymbol{q}_d(t)$, $\boldsymbol{\nu}_{d,0}^{0}(t)$, $\dot{\boldsymbol{q}}_d(t)$, $\dot{\boldsymbol{\nu}}_{d,0}^{0}(t)$, $\ddot{\boldsymbol{q}}_d(t)$ represent the desired trajectory. Let define

$$\boldsymbol{\nu}_{r,0}^{0} = \boldsymbol{\nu}_{d,0}^{0} + \begin{bmatrix} \lambda_{p,0}\boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \lambda_{o,0}\boldsymbol{I}_3 \end{bmatrix} \boldsymbol{e}_0, \tag{7.67}$$

$$\dot{q}_{r,i} = \dot{q}_{d,i} + \lambda_i \tilde{q}_i \qquad i = 1, \ldots, n \tag{7.68}$$

$$\boldsymbol{\nu}_{r,i+1}^{i+1} = \boldsymbol{U}_{i+1}^{i\mathrm{T}} \boldsymbol{\nu}_{r,i}^{i} + \dot{q}_{r,i+1} \boldsymbol{z}_i^{i+1} \qquad i = 0, \ldots, n-1\,, \tag{7.69}$$

where the (6 × 1) vector

$$\boldsymbol{e}_0 = \begin{bmatrix} \boldsymbol{R}_0{}^{\mathrm{T}} \tilde{\boldsymbol{p}}_0 \\ \tilde{\boldsymbol{\varepsilon}}_0^{0} \end{bmatrix}$$

denotes position and orientation errors for the vehicle, $\lambda_{p,0}$, $\lambda_{o,0}$, λ_i are positive design gains.

It is useful considering the following variables:

$$\begin{aligned} \boldsymbol{s}_i^{i} &= \boldsymbol{\nu}_{r,i}^{i} - \boldsymbol{\nu}_i^{i} && i = 0, \ldots, n \\ s_{q,i} &= \dot{q}_{r,i} - \dot{q}_i && i = 1, \ldots, n \\ \boldsymbol{s}_q &= [s_{q,1} \ \ldots \ s_{q,n}]^{\mathrm{T}} \end{aligned}$$

The discussed control law is based on the computation of the *required* generalized force for each rigid body in the system. Then, the input torques for the manipulator and the input generalized force for the vehicle are computed from the required forces according to (7.64) and (7.66).

In the following it is assumed that only a nominal estimate $\hat{\boldsymbol{\theta}}_i$ of the vector of dynamic parameters is available for the ith rigid body. Hence, a suitable update law for the estimates has to be adopted so as to ensure asymptotic tracking of the desired trajectory.

For the generic rigid body (including the vehicle) the required force has the following structure

$$\boldsymbol{h}_{r,i}^{i} = \boldsymbol{h}_{c,i}^{i} - \boldsymbol{U}_{i+1}^{i} \boldsymbol{h}_{c,i+1}^{i+1}$$

that, including the designed required force, implies

$$\boldsymbol{h}_{c,i}^{i} = \boldsymbol{Y}\left(\boldsymbol{R}_i, \boldsymbol{\nu}_i^{i}, \boldsymbol{\nu}_{r,i}^{i}, \dot{\boldsymbol{\nu}}_{r,i}^{i}\right) \hat{\boldsymbol{\theta}}_i + \boldsymbol{K}_{v,i} \boldsymbol{s}_i^{i} + \boldsymbol{U}_{i+1}^{i} \boldsymbol{h}_{c,i+1}^{i+1} \tag{7.70}$$

with $\boldsymbol{K}_{v,i} > \boldsymbol{O}$. The parameters estimate $\hat{\boldsymbol{\theta}}_i$ is dynamically updated via

$$\dot{\hat{\boldsymbol{\theta}}}_i = \boldsymbol{K}_{\theta,i}^{-1} \boldsymbol{Y}^{\mathrm{T}} \left(\boldsymbol{R}_i, \boldsymbol{\nu}_i^{i}, \boldsymbol{\nu}_{r,i}^{i}, \dot{\boldsymbol{\nu}}_{r,i}^{i}\right) \boldsymbol{s}_i^{i} \tag{7.71}$$

with $\boldsymbol{K}_{\theta,i} > \boldsymbol{O}$.

The control torque at the ith manipulator's joint is given by

$$\tau_{q,i} = \boldsymbol{z}_{i-1}^{i}{}^{\mathrm{T}} \boldsymbol{h}_{c,i}^{i} \,. \tag{7.72}$$

Finally, the generalized force for the vehicle needed to achieve the corresponding required force is computed as

$$\boldsymbol{h}_{c,0}^{0} = \boldsymbol{h}_{r,0}^{0} + \boldsymbol{U}_{1}^{0} \boldsymbol{h}_{c,1}^{1} \,.$$

7.9.1 Stability Analysis

In this section, the stability analysis for the discussed control law is provided.

Let consider the following scalar function

$$V_i(\boldsymbol{s}_i^i, \tilde{\boldsymbol{\theta}}_i) = \frac{1}{2} \boldsymbol{s}_i^{i\mathrm{T}} \boldsymbol{M}_i \boldsymbol{s}_i^i + \frac{1}{2} \tilde{\boldsymbol{\theta}}_i{}^{\mathrm{T}} \boldsymbol{K}_{\theta,i} \tilde{\boldsymbol{\theta}}_i \,. \tag{7.73}$$

The scalar $V_i(\boldsymbol{s}_i^i, \tilde{\boldsymbol{\theta}}_i) > 0$ in view of positive definiteness of $\boldsymbol{M}_i$ and $\boldsymbol{K}_{\theta,i}$.

By differentiating V_i with respect to time yields

$$\dot{V}_i = \boldsymbol{s}_i^{i\mathrm{T}} \boldsymbol{M}_i (\dot{\boldsymbol{\nu}}_{r,i}^i - \dot{\boldsymbol{\nu}}_i^i) - \tilde{\boldsymbol{\theta}}_i{}^{\mathrm{T}} \boldsymbol{K}_{\theta,i} \dot{\hat{\boldsymbol{\theta}}}_i \,,$$

where the parameters was considered constant or slowly varying, i.e.,

$$\dot{\tilde{\boldsymbol{\theta}}}_i = -\dot{\hat{\boldsymbol{\theta}}}_i \,.$$

Taking into account the equations of motions (7.65), and considering the vector $\boldsymbol{n}_i = \boldsymbol{C}_i(\boldsymbol{\nu}_i^i)\boldsymbol{\nu}_{r,i}^i + \boldsymbol{D}_i(\boldsymbol{\nu}_i^i)\boldsymbol{\nu}_{r,i}^i + \boldsymbol{g}_i^i(\boldsymbol{R}_i)$ it is:

$$\dot{V}_i = -\boldsymbol{s}_i^{i\mathrm{T}} \boldsymbol{D}_i(\boldsymbol{\nu}_i^i) \boldsymbol{s}_i^i + \boldsymbol{s}_i^{i\mathrm{T}} \left(\boldsymbol{M}_i \dot{\boldsymbol{\nu}}_{r,i}^i + \boldsymbol{n}_i(\boldsymbol{\nu}_i^i, \boldsymbol{\nu}_{r,i}^i, \boldsymbol{R}_i) - \boldsymbol{h}_{t,i}^i \right) - \tilde{\boldsymbol{\theta}}_i{}^{\mathrm{T}} \boldsymbol{K}_{\theta} \dot{\hat{\boldsymbol{\theta}}}_i \,.$$

By adding and subtracting the term $\boldsymbol{s}_i^{i\mathrm{T}} \boldsymbol{h}_{c,i}^i$, where $\boldsymbol{h}_{c,i}^i$ is the control law as introduced in (7.70), and by exploiting the linearity in the parameters, the previous equation can be rewritten as

$$\begin{aligned} \dot{V}_i = \; & -\boldsymbol{s}_i^{i\mathrm{T}} \boldsymbol{D}_i \boldsymbol{s}_i^i + \boldsymbol{s}_i^{i\mathrm{T}} \left(\boldsymbol{Y}_i \boldsymbol{\theta}_i - \boldsymbol{h}_{t,i}^i - \boldsymbol{Y}_i \hat{\boldsymbol{\theta}}_i - \boldsymbol{K}_{v,i} \boldsymbol{s}_i^i - \boldsymbol{U}_{i+1}^i \boldsymbol{h}_{c,i+1}^{i+1} \right) \\ & - \tilde{\boldsymbol{\theta}}_i{}^{\mathrm{T}} \boldsymbol{K}_{\theta,i} \dot{\hat{\boldsymbol{\theta}}}_i + \boldsymbol{s}_i^{i\mathrm{T}} \boldsymbol{h}_{c,i}^i , \end{aligned}$$

where $\boldsymbol{Y}_i = \boldsymbol{Y}(\boldsymbol{R}_i, \boldsymbol{\nu}_i^i, \boldsymbol{\nu}_{r,i}^i, \dot{\boldsymbol{\nu}}_{r,i}^i)$ and $\boldsymbol{D}_i = \boldsymbol{D}_i(\boldsymbol{\nu}_i^i)$.

By rearranging the terms one obtains:

$$\dot{V}_i = -\boldsymbol{s}_i^{i\mathrm{T}}(\boldsymbol{K}_{v,i} + \boldsymbol{D}_i)\boldsymbol{s}_i^i + \boldsymbol{s}_i^{i\mathrm{T}}\left(\boldsymbol{Y}_i\tilde{\boldsymbol{\theta}}_i - \boldsymbol{h}_{t,i}^i + \boldsymbol{h}_{r,i}^i\right) - \tilde{\boldsymbol{\theta}}_i{}^{\mathrm{T}}\boldsymbol{K}_{\theta,i}\dot{\hat{\boldsymbol{\theta}}}_i\,,$$

that, taking the update law for the dynamic parameters (7.71), finally gives

$$\dot{V}_i = -\boldsymbol{s}_i^{i\mathrm{T}}(\boldsymbol{K}_{v,i} + \boldsymbol{D}_i)\boldsymbol{s}_i^i + \boldsymbol{s}_i^{i\mathrm{T}}\tilde{\boldsymbol{h}}_{t,i}^i\,, \tag{7.74}$$

where $\tilde{\boldsymbol{h}}_{t,i}^i = \boldsymbol{h}_{r,i}^i - \boldsymbol{h}_{t,i}^i$. Equation (7.74) does not have any significant property with respect to its sign.

The Lyapunov candidate function for the UVMS is given by

$$V(\boldsymbol{s}_0^0, \ldots, \boldsymbol{s}_n^n, \tilde{\boldsymbol{\theta}}_0, \ldots, \tilde{\boldsymbol{\theta}}_n) = \sum_{i=0}^{n} V_i(\boldsymbol{s}_i^i, \tilde{\boldsymbol{\theta}}_i)\,,$$

that is positive definite in view of positive definitiveness of $V_i,\ i = 0, \ldots, n$.

Its time derivative is simply given by the time derivatives of all the scalar functions:

$$\dot{V} = \sum_{i=0}^{n}\left[-\boldsymbol{s}_i^{i\mathrm{T}}(\boldsymbol{K}_{v,i} + \boldsymbol{D}_i)\boldsymbol{s}_i^i + \boldsymbol{s}_i^{i\mathrm{T}}\tilde{\boldsymbol{h}}_{t,i}^i\right],$$

where the last term is null. In fact, let us consider the two equations:

$$\begin{aligned}\boldsymbol{h}_{t,i}^i &= \boldsymbol{h}_i^i - \boldsymbol{U}_{i+1}^i\boldsymbol{h}_{i+1}^{i+1}\,,\\ \boldsymbol{\nu}_{i+1}^{i+1} &= \boldsymbol{U}_{i+1}^{i\mathrm{T}}\boldsymbol{\nu}_i^i + \dot{q}_{i+1}\boldsymbol{z}_i^{i+1}\end{aligned}$$

it is possible to observe that the same relationships hold for $\tilde{\boldsymbol{h}}_{t,i}^i$ and $\boldsymbol{s}_i^i$:

$$\begin{aligned}\tilde{\boldsymbol{h}}_{t,i}^i &= \tilde{\boldsymbol{h}}_i^i - \boldsymbol{U}_{i+1}^i\tilde{\boldsymbol{h}}_{i+1}^{i+1}\,,\\ \boldsymbol{s}_{i+1}^{i+1} &= \boldsymbol{U}_{i+1}^{i\mathrm{T}}\boldsymbol{s}_i^i + s_{q,i+1}\boldsymbol{z}_i^{i+1}\,.\end{aligned}$$

where $\tilde{\boldsymbol{h}}_i^i = \boldsymbol{h}_{c,i}^i - \boldsymbol{h}_i^i$. Recalling that $\boldsymbol{\tau} = \boldsymbol{J}_w{}^{\mathrm{T}}\boldsymbol{h}_e$, the last term of the time derivative of the Lyapunov function candidate is then given by:

$$\sum_{i=0}^{n}\boldsymbol{s}_i^{i\mathrm{T}}\tilde{\boldsymbol{h}}_{t,i}^i = \boldsymbol{s}_0^{0\mathrm{T}}\tilde{\boldsymbol{h}}_0^0 + \sum_{i=1}^{n} s_{q,i}\boldsymbol{z}_{i-1}^i{}^{\mathrm{T}}\tilde{\boldsymbol{h}}_i^i$$

$$
\begin{aligned}
&= \boldsymbol{s}_0^{0\mathrm{T}} \tilde{\boldsymbol{h}}_0^0 + \sum_{i=1}^{n} s_{q,i} \tilde{\tau}_{q,i} \\
&= \boldsymbol{s}_0^{0\mathrm{T}} \tilde{\boldsymbol{h}}_0^0 + \boldsymbol{s}_q{}^{\mathrm{T}} \tilde{\boldsymbol{\tau}}_q \\
&= \begin{bmatrix} \boldsymbol{s}_0^0 \\ \boldsymbol{s}_q \end{bmatrix}^{\mathrm{T}} \boldsymbol{J}_w{}^{\mathrm{T}} \tilde{\boldsymbol{h}}_e \\
&= 0
\end{aligned}
\tag{7.75}
$$

since, in absence of contact at the end effector, $\tilde{\boldsymbol{h}}_e = \boldsymbol{0}$. To understand the first equality let rewrite the first term for two consecutive links:

$$
\begin{aligned}
\boldsymbol{s}_i^{i\mathrm{T}} \tilde{\boldsymbol{h}}_{t,i}^{i} + \boldsymbol{s}_{i+1}^{i+1\mathrm{T}} \tilde{\boldsymbol{h}}_{t,i+1}^{i+1} &= (\boldsymbol{U}_i^{i-1\mathrm{T}} \boldsymbol{s}_{i-1}^{i-1} + s_{q,i} \boldsymbol{z}_{i-1}^{i})^{\mathrm{T}} \tilde{\boldsymbol{h}}_i^i - \boldsymbol{s}_i^{i\mathrm{T}} \boldsymbol{U}_{i+1}^{i} \tilde{\boldsymbol{h}}_{i+1}^{i+1} \\
&\quad + (\boldsymbol{U}_{i+1}^{i}{}^{\mathrm{T}} \boldsymbol{s}_i^i + s_{q,i+1} \boldsymbol{z}_i^{i+1})^{\mathrm{T}} \tilde{\boldsymbol{h}}_{i+1}^{i+1} - \boldsymbol{s}_{i+1}^{i+1\mathrm{T}} \boldsymbol{U}_{i+2}^{i+1} \tilde{\boldsymbol{h}}_{i+2}^{i+2} \\
&= (\boldsymbol{U}_i^{i-1\mathrm{T}} \boldsymbol{s}_{i-1}^{i-1} + s_{q,i} \boldsymbol{z}_{i-1}^{i})^{\mathrm{T}} \tilde{\boldsymbol{h}}_i^i \\
&\quad + (s_{q,i+1} \boldsymbol{z}_i^{i+1})^{\mathrm{T}} \tilde{\boldsymbol{h}}_{i+1}^{i+1} - \boldsymbol{s}_{i+1}^{i+1\mathrm{T}} \boldsymbol{U}_{i+2}^{i+1} \tilde{\boldsymbol{h}}_{i+2}^{i+2},
\end{aligned}
$$

that, considering that the first and the last term are null for the first and the last rigid body respectively, gives the relation required in (7.75).

It is now possible to prove the system stability in a Lyapunov-Like sense using the Barbălat's Lemma. Since

- $V(\boldsymbol{s}_0^0, \ldots, \boldsymbol{s}_n^n, \tilde{\boldsymbol{\theta}}_0, \ldots, \tilde{\boldsymbol{\theta}}_n)$ is lower bounded;
- $\dot{V} \leq 0$;
- $\dot{V}$ is uniformly continuous.

then

- $\dot{V} \rightarrow 0$ as $t \rightarrow \infty$.

Thus $\boldsymbol{s}_0^0, \ldots, \boldsymbol{s}_n^n \rightarrow \boldsymbol{0}$ as $t \rightarrow \infty$. Due to the recursive definition of the vectors $\boldsymbol{s}_i^i$, the position errors converge to the null value as well. In fact, the vector $\boldsymbol{s}_0^0 \rightarrow \boldsymbol{0}$ implies $\boldsymbol{\nu}_0^0 \rightarrow \boldsymbol{0}$ and $\boldsymbol{e}_0 \rightarrow \boldsymbol{0}$ ($\lambda_{p,0} > 0$ and $\lambda_{o,0} > 0$). Moreover, since the rotation matrix has full rank, the vehicle position error $\tilde{\boldsymbol{\eta}}_1$ too decreases to the null value. Convergence to zero of $\boldsymbol{s}_i^i$ directly implies convergence to zero of $s_{q,1}, \ldots, s_{q,n}$; consequently, $\dot{\tilde{\boldsymbol{q}}} \rightarrow \boldsymbol{0}$ and $\tilde{\boldsymbol{q}} \rightarrow \boldsymbol{0}$. As usual in adaptive control technique, it is not possible to prove asymptotic stability of the whole state, since $\tilde{\boldsymbol{\theta}}_i$ is only guaranteed to be bounded.

Remarks

- Achieving null error of $n + 1$ rigid bodies with 6-DOF with only $6 + n$ inputs is physically coherent since the control law is computed by taking into account the kinematic constraints of the system.

Table 7.5 Masses, vehicle length and Denavit-Hartenberg parameters (m, rad) of the manipulator mounted on the underwater vehicle

	Mass (kg)	Length (m)	a	d	θ	α
Vehicle	5454	5.3	–	–	–	–
Link 1	80	–	0.150	0	q_1	$-\pi/2$
Link 2	80	–	0.610	0	q_2	0
Link 3	30	–	0.110	0	q_3	$-\pi/2$
Link 4	50	–	0	0.610	q_4	$\pi/2$
Link 5	20	–	0	−0.113	q_5	$-\pi/2$
Link 6	25	–	0	0.103	q_6	0

- In [62] the control law performs an implicit kinematic inversion. This approach requires to work with a 6-DOF robot for the stability analysis of a position/orientation control of the end effector. In case of a redundant robot, the authors suggest the implementation of an augmented Jacobian approach in order to have a square Jacobian to work with. However, the possible occurrence of algorithmic singularities is not avoided (see also Sect. 6.2). On the other hand, if the kinematic control is kept separate from the dynamic loop, as in the discussed approach, it is possible to use inverse kinematic techniques robust to the occurrence of algorithmic singularities.

7.9.2 Simulations

Dynamic simulations have been performed to show the effectiveness of the discussed control law based on the simulation tool SIMURV, described in Chap. 9. The vehicle data are taken from [49] and are referred to the experimental Autonomous Underwater Vehicle NPS Phoenix. For simulation purposes, a six-degree-of-freedom manipulator with large inertia has been considered which is mounted under the vehicle's body. The manipulator structure and the dynamic parameters are those of the Smart-3S manufactured by COMAU. Its weight is about 5 % of the vehicle weight, its length is about 2 m stretched while the vehicle is 5 m long. The overall structure, thus, has 12 degrees of freedom. The relevant physical parameters of the system are reported in Table 7.5.

Notice one of the features of the controller: by changing the manipulator does not imply redesigning the control but it simply modifies the Denavit-Hartenberg table and the initial estimate of the dynamic parameters. As a matter of fact, the code would be exactly the same, just reading the parameters' values from different data files, with obvious advantages for software debugging and maintenance.

Notice, that, for all the simulations the following conditions have been considered:

- full degrees-of-freedom dynamic simulations;
- detailed mathematical model used in the simulation;

- inaccurate initial parameters estimates used by the controller (about 15 % for each parameter);
- reduced order regressor used in the control law;
- digital implementation of the control law (at a sampling rate of 200 Hz);
- quantization of the sensor outputs and measurement noise.

The only significant simplification is the neglect of the thruster dynamics that causes limit cycle in the vehicle behavior [68]. It is worth remarking that the adaptive control law uses a reduced number of dynamic parameters for each body: body mass and mass of displaced fluid (1 parameter), first moment of gravity and buoyancy (3 parameters). Those are the parameters that, in absence of current, can affect a steady state error. Thus, each vector $\hat{\boldsymbol{\theta}}_i$ is composed of 4 elements. In Chap. 3, a detailed analysis of adaptive/integral actions in presence of the ocean current is discussed.

The desired end-effector trajectory is a straight line the projection of which on the inertial axis is a segment of 0.2 m. The line has to be executed four times, with duration of the single trial of 4 s with a fifth-order polynomial time law. Then, 14 s of rest are imposed. The attitude of the end effector must be kept constant during the motion.

The initial configuration is (see Fig. 7.23):

$$\begin{aligned}\boldsymbol{\eta} &= \begin{bmatrix}0 \ 0 \ 0 \ 0 \ 0 \ 0\end{bmatrix}^{\mathrm{T}} \ \mathrm{m, deg}\\ \boldsymbol{q} &= \begin{bmatrix}0 \ 180 \ 0 \ 0 \ 90 \ 180\end{bmatrix}^{\mathrm{T}} \ \mathrm{deg}\end{aligned}$$

The redundancy of the system is used to keep still the vehicle. Notice, that this choice is aimed only at simplifying the analysis of the discussed controller. More complex secondary tasks can easily be given with the inverse kinematics algorithm [69–71].

The control law parameters are:

$$\begin{array}{lll}\lambda_{p,0} & = 0.4 & \\ \lambda_{o,0} & = 0.6 & \\ \lambda_i & = 0.9 & i = 1, \ldots, n\\ \boldsymbol{K}_{v,0} & = \mathrm{blockdiag}\,\{15000\boldsymbol{I}_3,\, 16000\boldsymbol{I}_3\} & \\ \boldsymbol{K}_{v,i} & = \mathrm{blockdiag}\,\{900\boldsymbol{I}_3,\, 1100\boldsymbol{I}_3\} & i = 1, \ldots, n\\ \boldsymbol{K}_{\theta,0} & = 14\boldsymbol{I}_4 & \\ \boldsymbol{K}_{\theta,i} & = 21\boldsymbol{I}_4 & i = 1, \ldots, n\end{array}$$

Figures 7.24 and 7.25 report the time history of the end-effector position and orientation errors. It can be noted that, since the trajectory is repeated 4 times in the first 16 s, after a large initial error, the adaptive action can significantly reduce the tracking error. At steady state the errors tend to zero.

In Fig. 7.26 the time history of the vehicle position is reported. It can be noted that the main tracking error is observed along z. This is due to the fact that the vehicle

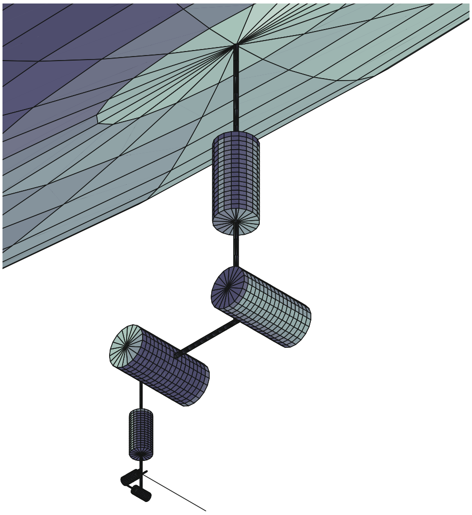


Fig. 7.23 Virtual decomposition control. Sketch of the initial configuration of the system and trace of the desired path

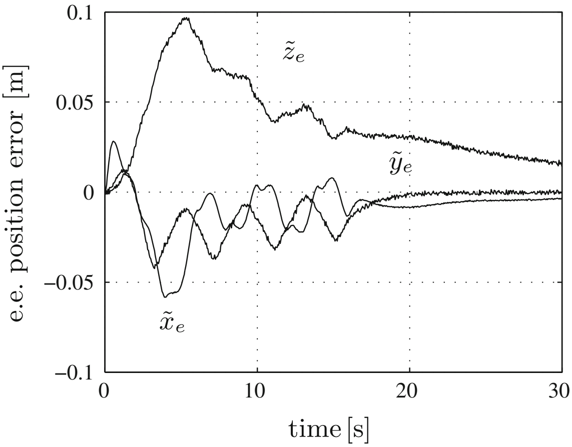


Fig. 7.24 Virtual decomposition control. Time history of the end-effector position error. The periodic desired trajectory makes it clear the advantage of the adaptive action

is not neutrally buoyant and at the beginning the estimation of the restoring forces has to wait for the adaptation of the control law. This can be observed also from Fig. 7.27, where the time history of the vehicle linear forces is reported. Since, the manipulator is not neutrally buoyant, a large force is experienced along the z-axis at rest.

In Fig. 7.28 the time history of the vehicle attitude in terms of Euler angles is shown. It can be noted that the main tracking error is observed along the vehicle pitch angle θ. The manipulator, in fact, interacts with the vehicle mainly in this direction. Also, Fig. 7.29 shows the time history of the vehicle moments; notice the large initial overestimate due to the intentional wrong model compensation. Also,

Fig. 7.25 Virtual decomposition control. Time history of the end-effector attitude error in terms of Euler angles

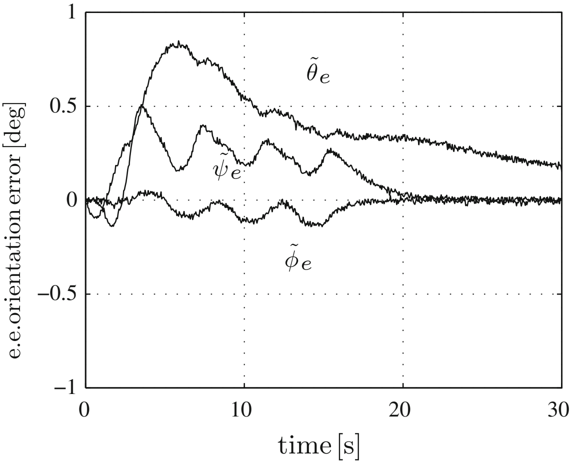
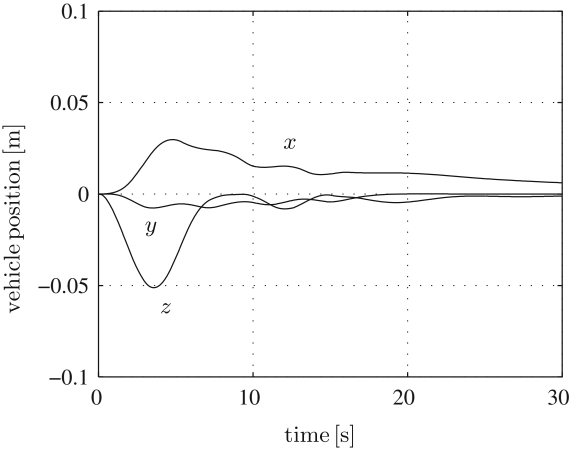


Fig. 7.26 Virtual decomposition control. Time history of the vehicle position



being the manipulator not neutrally buoyant, a large moment is experienced along the *y*-axis at rest.

Figure 7.30 shows the time history of the joint errors. Those are computed with respect to the desired joint positions as output from the inverse kinematics algorithm. The errors are quite large since the aim of the simulation was to show the benefit of the adaptive action in a virtual decomposition approach. The end-effector errors are the composition of all the tracking errors along the structure; the manipulator, characterized by smaller inertia and higher precision with respect to the vehicle, will be in charge of compensating for the effects of the vehicle tracking errors on the end-effector. For this specific simulation, thus, the end-effector tracking error is comparable with the vehicle position/orientation tracking error.

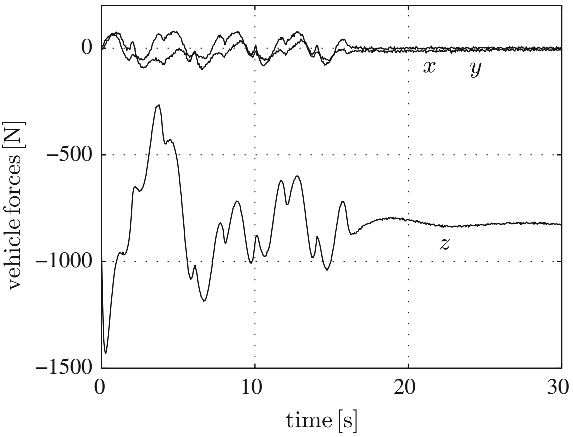


Fig. 7.27 Virtual decomposition control. Time history of the vehicle linear forces

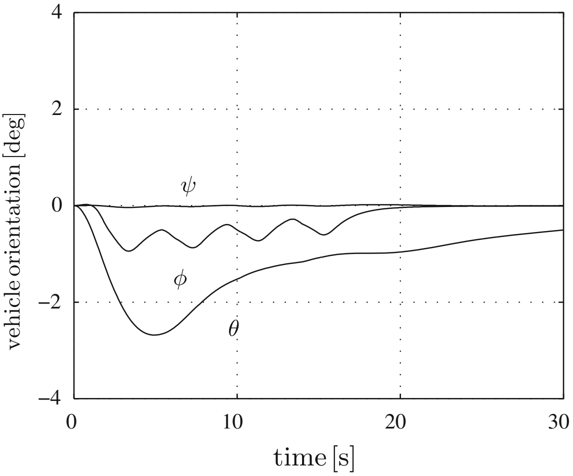


Fig. 7.28 Virtual decomposition control. Time history of the vehicle attitude in terms of Euler angles

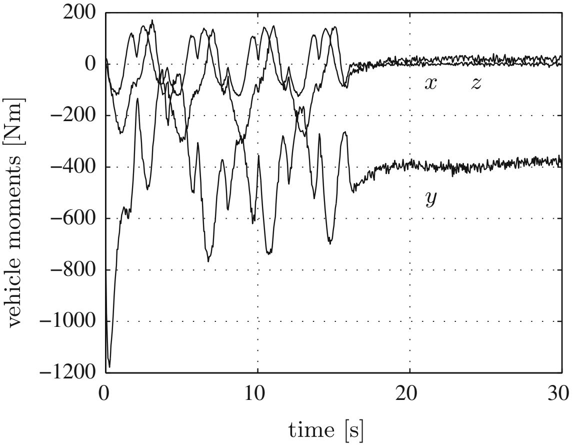


Fig. 7.29 Virtual decomposition control. Time history of the vehicle moments

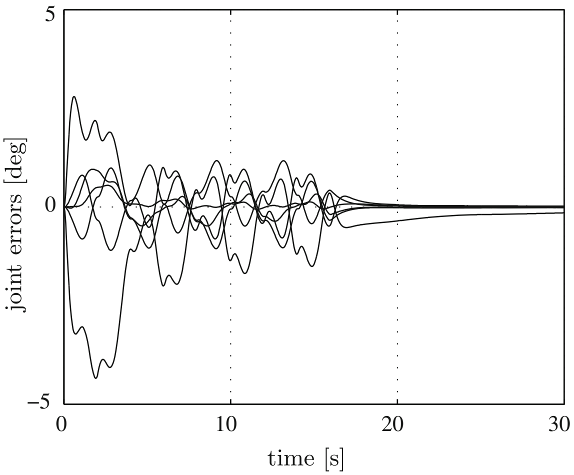


Fig. 7.30 Virtual decomposition control. Time history of the joint errors

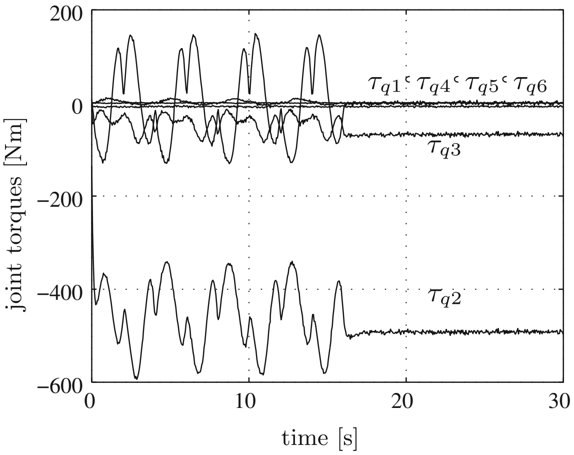


Fig. 7.31 Virtual decomposition control. Time history of the joint torques

In Fig. 7.31 the time history of the joint torques is reported. Notice, that only the torque of the second joint is mainly affected by the manipulator restoring forces. At the very beginning the restoring compensation (≈200 Nm) presents a large difference with respect to the final value (≈500 Nm) corresponding to the same configuration due to the error in the parameter estimation.

7.9.3 Virtual Decomposition with the Proper Adapting Action

In the previous section, an adaptive control law in which the serial-chain structure of the UVMS is exploited is discussed. The overall motion control problem is decomposed in a set of elementary control problems regarding the motion of each rigid body in the system, namely the manipulator's links and the vehicle. For each body, a control action is designed to assign the desired motion, to adaptively compensate for the body dynamics, and to counteract force/moment exchanged with its neighborhoods along the chain.

On the other hand, in Chap. 3 it is shown, that, for a single rigid body, a suitable regressor-based adaptive control law (the mixed earth/vehicle-fixed-frame-based, model-based controller presented in Sect. 3.6) gets improvement in the tracking error by considering the proper adaptation on the sole persistent terms, i.e., the current and the restoring forces. It is worth noticing that the rigid bodies of the manipulator are subject to a fast dynamics and thus the effects numerically shown in Chap. 3 for a single rigid body (the vehicle) are magnified for a fast movement of one of the arm.

It is advisable, thus, to merge these two approaches as done by Antonelli et al. in [72] in order to get benefit from both of them. The resulting control scheme has a modular structure which greatly simplifies its application to systems with a large number of links; furthermore, it reduces the required computational burden by replacing one high-dimensional problem with many low-dimensional ones, finally allows efficient implementation on distributed computing architectures.

Numerical simulations have been performed to show the effectiveness of the discussed control law with the same model used for the virtual decomposition approach; the simulation, thus, uses $\approx$400 dynamic parameters. The overall number of parameters of the controller is $9 * (n + 1) = 63$. Moreover, the software to implement the controller is modular, the same function is used to compensate for all the rigid bodies, either the vehicle or links of the manipulator, with different parameters as inputs. This makes easier the debugging procedure.

The desired end-effector path is a straight line with length of 35 cm, to be executed 4 times according to a fifth-order polynomial time law; the duration of each cycle is 4 s. The desired end-effector orientation is constant along the commanded path. The vehicle is commanded to keep its initial position and orientation during the task execution. Therefore, the inverse kinematics is needed to compute the sole joint vectors $\boldsymbol{q}_d(t)$, $\dot{\boldsymbol{q}}_d(t)$ and $\ddot{\boldsymbol{q}}_d(t)$ in real time, and thus only the inverse of the (6×6) manipulator Jacobian $\boldsymbol{J}_m$ is required in the inverse kinematics algorithm.

A constant ocean current affect the motion with the following components:

$$\boldsymbol{\nu}_c^I = \begin{bmatrix} 0.1 \ 0.3 \ 0 \ 0 \ 0 \ 0 \end{bmatrix}^{\mathrm{T}} \quad \text{m/s} \, .$$

As reported in [73–75], some of the parameters of the controllers are related to the knowledge of the mass and the first moment of gravity/buoyancy. Their initial estimate are set so as to give an estimation error larger then 20 % of the true values. The other parameters are related to the presence of the current. Those, will be initialized to the null value.

The control law parameters have been set to:

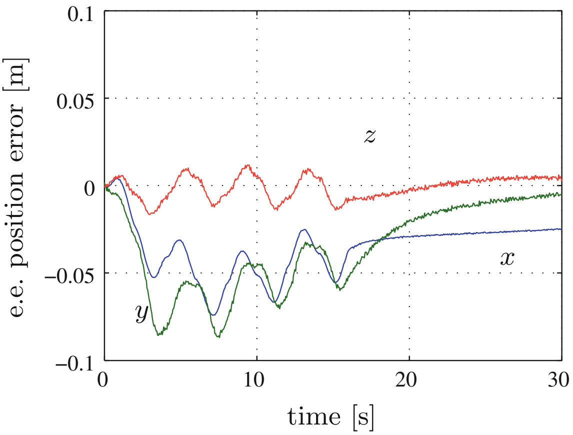


Fig. 7.32 Virtual decomposition control + proper adaptive action. End-effector position error

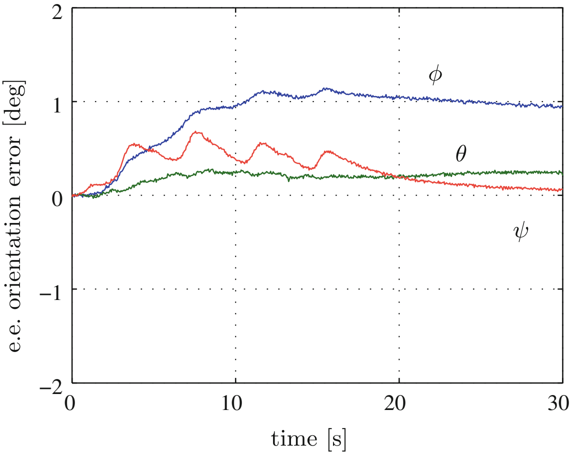


Fig. 7.33 Virtual decomposition control + proper adaptive action. End-effector orientation error

$$\begin{aligned}
\boldsymbol{\Lambda}_0 &= \mathrm{blockdiag}\{0.4\boldsymbol{I}_3, 0.6\boldsymbol{I}_3\}\,,\\
\boldsymbol{\Lambda}_{i=1,6} &= \mathrm{blockdiag}\{0.9\boldsymbol{I}_3, 0.9\boldsymbol{I}_3\}\,,\\
\boldsymbol{K}_{v,0} &= \mathrm{blockdiag}\{9100\boldsymbol{I}_3, 9800\boldsymbol{I}_3\}\,,\\
\boldsymbol{K}_{v,i=1,6} &= \mathrm{blockdiag}\{600\boldsymbol{I}_3, 800\boldsymbol{I}_3\}\,,\\
\boldsymbol{K}_{\theta,0} &= 100\boldsymbol{I}_9\,,\\
\boldsymbol{K}_{\theta,i=1,6} &= 40\boldsymbol{I}_9\,,
\end{aligned}$$

where $\boldsymbol{I}_9$ is the (9×9) identity matrix.

The results are reported in Figs. 7.32 and 7.33 in terms of end-effector position/orientation tracking errors; it can be recognized that, in spite of the demanding task commanded to the system, the errors are kept small in the transients and reach zero values at steady state.

Fig. 7.34 Virtual decomposition control + proper adaptive action. Vehicle control forces

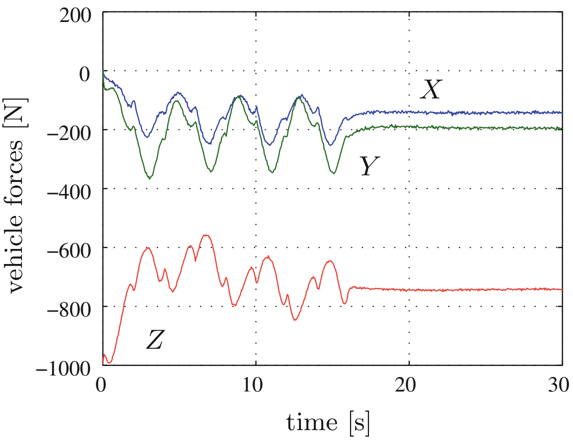


Fig. 7.35 Virtual decomposition control + proper adaptive action. Vehicle control moments

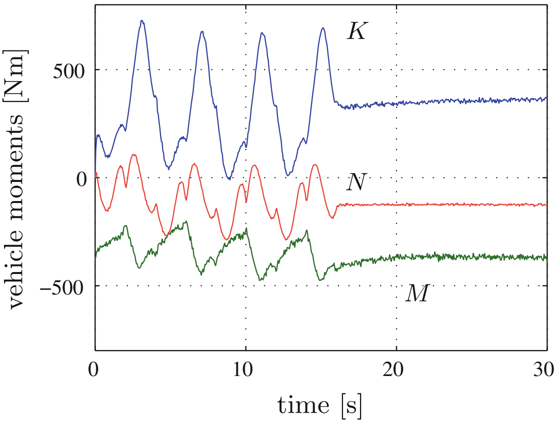


Fig. 7.36 Virtual decomposition control + proper adaptive action. Joint control torques

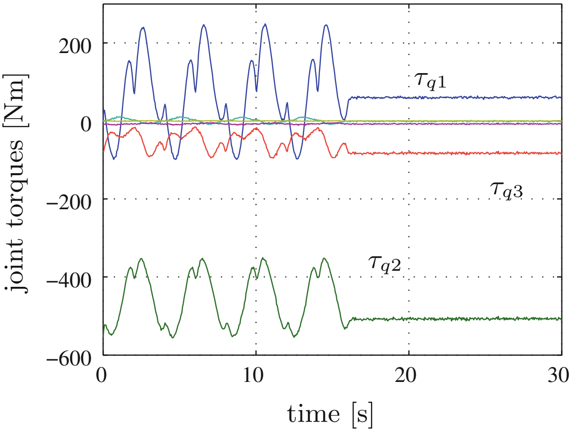


Fig. 7.37 Virtual decomposition control + proper adaptive action. Vehicle position

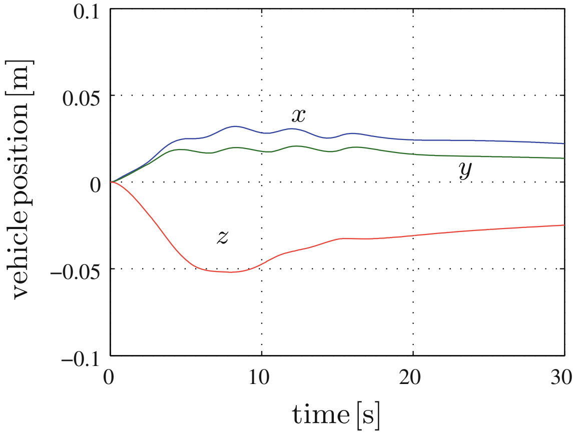


Fig. 7.38 Virtual decomposition control + proper adaptive action. Vehicle orientation

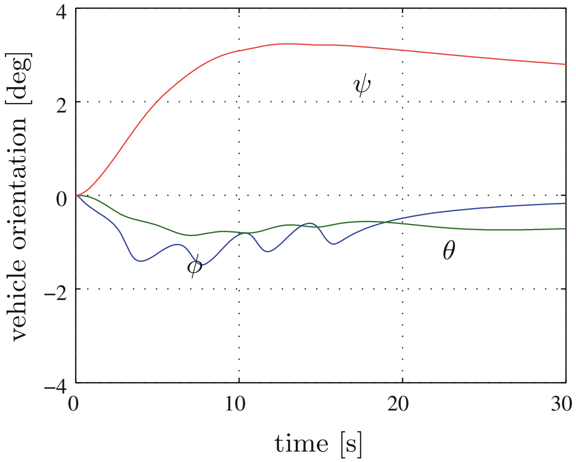
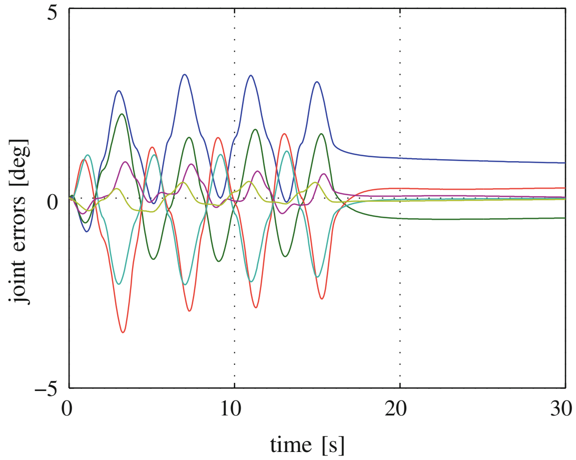


Fig. 7.39 Virtual decomposition control + proper adaptive action. Joint position errors



Moreover, the results in Figs. 7.34, 7.35 and 7.36 show that the control force and moments acting on the vehicle, as well as the control torques applied at the manipulator's joints, are kept limited along all the trajectory and are characterized by a smooth profile. It is worth noticing that, at the beginning of the task, the controller is not aware of the presence of the current; a compensation can be observed mainly along the x and y vehicle linear forces and moments.

In Figs. 7.37, 7.38 and 7.39, the vehicle position and orientation and the joint tracking errors are reported. The tracking errors along the yaw direction can be motivated by the effect of the current, its value is decreasing to the null value according to the control gains.

7.10 Conclusions

In this chapter an overview of possible control strategies for UVMSs has been presented. In view of these first, preliminary, results, it can be observed that the simple translation of control strategies developed for industrial robotics is not possible. The reason can be found both in the different technical characteristics of actuating/sensing system and in the different nature of the forces that act on a submerged body. As shown in Chap. 3, neglecting the physical nature of these forces can cause the controller to feed the system with a disturbance rather than a proper control action.

At the same time, an UVMS is a complex system, neglecting the computational aspect can lead the designer to develop a controller for which the tuning and wet-test phase might be unpractical. Starting from the simulation phase it might be appropriate to test simplified version of the controllers.

Among the control strategies analyzed the Virtual Decomposition approach, merged with the proper adaptation action, has several appealing characteristic: it is adaptive in the dynamic parameters; it is based on a Newton-Euler formulation that keep limited the computational burden; it is compatible with kinematic control strategies; it avoids representation singularities; it is modular, thus simplifying the software debugging and maintenance.

References

1. H. Mahesh, J. Yuh, R. Lakshmi, Control of underwater robots in working mode, in *Proceedings of 1991 IEEE International Conference on Robotics and Automation, 1991*, IEEE, 1991, pp. 2630–2635
2. H. Mahesh, J. Yuh, R. Lakshmi, A coordinated control of an underwater vehicle and robotic manipulator. J. Robot. Syst. **8**(3), 339–370 (1991)
3. T. Fossen, Adaptive macro-micro control of nonlinear underwater robotic systems, in *Fifth International Conference on Advanced Robotics, 'Robots in Unstructured Environments', 1991, 91 ICAR*, IEEE, 1991, pp. 1569–1572

4. N. Kato, D. Lane, Co-ordinated control of multiple manipulators in underwater robots, in *Proceedings of 1996 IEEE International Conference on Robotics and Automation, 1996*, vol. 3, IEEE, Minneapolis, Minnesota, 1996, pp. 2505–2510
5. N. Sakagami, M. Shibata, S. Kawamura, T. Inoue, H. Onishi, S. Murakami, An attitude control system for underwater vehicle-manipulator systems, in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1761–1767
6. N. Sakagami, T. Ueda, M. Shibata, S. Kawamura, Pitch and roll control using independent movable floats for small underwater robots, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4756–4761
7. M.W. Dunnigan, G.T. Russell, Reduction of the dynamic coupling between a manipulator and ROV using variable structure control, in *International Conference on Control, 1994, Control'94*, IET, 1994, pp. 1578–1583
8. M.W. Dannigan, G.T. Russell, Evaluation and reduction of the dynamic coupling between a manipulator and an underwater vehicle. IEEE J. Oceanic Eng. **23**(3), 260–273 (1998)
9. G.B. Chung, K.S. Eom, B.-J. Yi, I.H. Suh, S.-R. Oh, W.K. Chung, J. Kim, Disturbance observer-based robust control for underwater robotic systems with passive joints. Adv. Robot. **15**(5), 575–588 (2001)
10. M. Lee, H.-S. Choi, A robust neural controller for underwater robot manipulators. IEEE Trans. Neural Netw. **11**(6), 1465–1470 (2000)
11. S.R. Pandian, N. Sakagami, A neuro-fuzzy controller for underwater robot manipulators, in *11th International Conference on Control Automation Robotics Vision (ICARCV), 2010*, 2010, pp. 2135–2140
12. P. Fraisse, L. Lapierre, P. Dauchez, F. Pierrot, Position/force control of an underwater vehicle equipped with a robotic manipulator, in *6th IFAC Symposium on Robot Control*, Austria, Wien, 2000, pp. 475–479
13. J.H. Ryu, D.-S. Kwon, P.-M. Lee, Control of underwater manipulators mounted on an ROV using base force information, in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001*, vol. 4, IEEE, Seoul, Korea, 2001, pp. 3238–3243
14. S. Kawamura, N. Sakagami, Analysis on dynamics of underwater robot manipulators based on iterative learning control and time-scale transformation, in *Proceedings of IEEE International Conference on Robotics and Automation, 2002, ICRA'02*, vol. 2, IEEE, Washington, DC, 2002, pp. 1088–1094
15. J. Kim, W.K. Chung, J. Yuh, Dynamic analysis and two-time scale control for underwater vehicle-manipulator systems, in *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, (IROS 2003)*, vol. 1, IEEE, 2003, pp. 577–582
16. M. Ishitsuka, S. Sagara, K. Ishii, Dynamics analysis and resolved acceleration control of an autonomous underwater vehicle equipped with a manipulator, in *2004 International Symposium on Underwater Technology, 2004, UT'04*, IEEE, Taipei, Taiwan, 2004, pp. 277–281
17. M. Ishitsuka, K. Ishii, Development of an underwater manipulator mounted for an AUV, in *Proceedings of MTS/IEEE OCEANS, 2005*, 2005, pp. 1811–1816
18. Y.C. Sun, C.C. Cheah, Adaptive setpoint control of underwater vehicle-manipulator systems, in *2004 IEEE Conference on Robotics, Automation and Mechatronics*, vol. 1, IEEE, Singapore, 2004, pp. 434–439
19. E. Marchand, F. Chaumette, F. Spindler, M. Perrier, Controlling the manipulator of an underwater ROV using a coarse calibrated pan/tilt camera, in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001*, vol. 3, IEEE, Seoul, Korea, 2001, pp. 2773–2778
20. M. Prats, D. Ribas, N. Palomeras, J.C. García, V. Nannen, S. Wirth, J.J. Fernández, J.P. Beltrán, R. Campos, P. Ridao et al., Reconfigurable AUV for intervention missions: a case study on underwater object recovery. Intell. Serv. Robot. **5**(1), 19–31 (2012)
21. J. Fernández, M. Prats, P. Sanz, J. C. García Sánchez, R. Marin, M. Robinson, D. Ribas, P. Ridao, Manipulation in the seabed: a new underwater robot arm for shallow water intervention. IEEE Robot. Autom. Mag. (2013)

22. T.W. McLain, S.M. Rock, M.J. Lee, Coordinated control of an underwater robotic system, in *Video Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1996, pp. 4606–4613
23. T.W. McLain, S.M. Rock, M.J. Lee, Experiments in the coordinated control of an underwater arm/vehicle system. Auton. Robot. **3**(2), 213–232 (1996)
24. T.J. Tarn, S.P. Yang, Modeling and control for underwater robotic manipulators—an example, in *Proceedings of 1997 IEEE International Conference on Robotics and Automation, 1997*, vol. 3, IEEE, Albuquerque, NM, 1997, pp. 2166–2171
25. T.J. Tarn, G.A. Shoults, S.P. Yang, A dynamic model of an underwater vehicle with a robotic manipulator using Kane's method. Auton. Robot. **3**(2), 269–283 (1996)
26. I. Schjølberg, T. Fossen, Modelling and control of underwater vehicle-manipulator systems, in *Proceedings of 3rd Conference on Marine Craft maneuvering and control*, Southampton, UK, 1994, pp. 45–57
27. I. Schjølberg, Modeling and Control of Underwater Robotic Systems. Ph.D. thesis, Doktor ingeniør degree, Norwegian University of Science and Technology, Trondheim, Norway, 1996
28. C. Canudas de Wit, E. Olguin Diaz, M. Perrier, Robust nonlinear control of an underwater vehicle/manipulator system with composite dynamics, in *Proceedings of 1998 IEEE International Conference on Robotics and Automation, 1998*, IEEE, Leuven, Belgium, 1998, pp. 452–457
29. C. Canudas de Wit, O. Olguin Diaz, M. Perrier, Control of underwater vehicle/manipulator with composite dynamics, in *Proceedings of the 1998 American Control Conference, 1998*, vol. 1, IEEE, 1998, pp. 389–393
30. C. Canudas de Wit, O. Olguin Diaz, M. Perrier, Nonlinear control of an underwater vehicle/manipulator with composite dynamics. IEEE Trans. Control Syst. Technol. **8**(6), 948–960 (2000)
31. E. Olguin Diaz, Modélisation et Commande d'un Système Véhicule/Manipulateur Sous-Marin (in French). Ph.D. thesis, Docteur de l'Institut National Polytechnique de Grenoble, Grenoble, France, 1999
32. P.-M. Lee, J. Yuh, Application of non-regressor based adaptive control to an underwater mobile platform-mounted manipulator, in *Proceedings of the 1999 IEEE International Conference on Control Applications, 1999*, vol. 2, IEEE, Kohala Coast, Hawaii, 1999, pp. 1135–1140
33. N. Sarkar, J. Yuh, T.K. Podder, Adaptive control of underwater vehicle-manipulator systems subject to joint limits, in *Proceedings. 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999, IROS'99*, vol. 1, IEEE, 1999, pp. 142–147
34. J. Yuh, An adaptive and learning control system for underwater robots, in *13th World Congress International Federation of Automatic Control*, California, San Francisco, 1996, pp. 145–150
35. S.K. Choi, J. Yuh, Experimental study on a learning control system with bound estimation for underwater robots. Auton. Robot. **3**(2), 187–194 (1996)
36. J. Nie, J. Yuh, E. Kardash, T. Fossen, On-board sensor-based adaptive control of small UUVS in very shallow water. Int. J. Adapt. Control Signal Process. **14**(4), 441–452 (2000)
37. J. Yuh, J. Nie, C.S.G. Lee, Experimental study on adaptive control of underwater robots, in *Proceedings of 1999 IEEE International Conference on Robotics and Automation, 1999*, IEEE, 1999, pp. 393–398
38. S. Zhao, J. Yuh, Experimental study on advanced underwater robot control. IEEE Trans. Robot. **21**(4), 695–703 (2005)
39. J. Yuh, S. Zhao, P.-M. Lee, Application of adaptive disturbance observer control to an underwater manipulator, in *Proceedings of 2001 IEEE International Conference on Robotics and Automation, 2001, ICRA*, vol. 4, IEEE, Seoul, Korea, 2001, pp. 3244–3249
40. G. Marani, J. Yuh, S.K. Choi, Autonomous manipulation for an intervention AUV. IEE Control Eng. Ser. **69**, 217 (2006)
41. G. Marani, S.K. Choi, J. Yuh, Underwater autonomous manipulation for intervention missions AUVs. Ocean Eng. **36**(1), 15–23 (2009)
42. G. Marani, S.K. Choi, J. Yuh, Real-time center of buoyancy identification for optimal hovering in autonomous underwater intervention. Intell. Serv. Robot. **3**(3), 175–182 (2010)

43. V. Utkin, Variable structure systems with sliding modes. IEEE Trans. Autom. Control **22**(2), 212–222 (1977)
44. K.K. Young, Controller design for a manipulator using theory of variable structure systems. IEEE Trans. Syst. Man Cybern. **8**(2), 101–109 (1978)
45. J.J. Slotine, W. Li, *Applied Nonlinear Control*, vol. 199 (Prentice hall, New Jersey, 1991)
46. O.E. Fjellstad, T.I. Fossen, Singularity-free tracking of unmanned underwater vehicles in 6 DOF, in *Proceedings of the 33rd IEEE Conference on Decision and Control, 1994*, vol. 2, IEEE, 1994, pp. 1128–1133
47. O. Egeland, J.-M. Godhavn, Passivity-based adaptive attitude control of a rigid spacecraft. IEEE Trans. Autom. Control **39**(4), 842–846 (1994)
48. J.J. Slotine, M. Di Benedetto, Hamiltonian adaptive control of spacecraft. IEEE Trans. Autom. Control **35**(7), 848–852 (1990)
49. A. Healey, D. Lienard, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J. Oceanic Eng. **18**(3), 327–339 (1993)
50. R. Ortega, M. Spong, Adaptive motion control of rigid robots: a tutorial. Automatica **25**(6), 877–888 (1989)
51. J.J.E. Slotine, W. Li, On the adaptive control of robot manipulators. Int. J. Robot. Res. **6**(3), 49–59 (1987)
52. T. Fossen, J.G. Balchen et al., The NEROV autonomous underwater vehicle, in *Proceeding Conference Oceans 91*, Citeseer, Honolulu, HI, 1991
53. T.I. Fossen, O. Fjellstad, Robust adaptive control of underwater vehicles: a comparative study, in *IFAC Workshop on Control Applications in Marine Systems*, IEEE, Trondheim, Norway, 1995, pp. 66–74
54. F. Lizarralde, J.T. Wen, L. Hsu, Quaternion-based coordinated control of a subsea mobile manipulator with only position measurements, in *Proceedings of the 34th IEEE Conference on Decision and Control, 1995*, vol. 4, IEEE, New Orleans, Louisiana, 1995, pp. 3996–4001
55. H. Berghuis, H. Nijmeijer, A passivity approach to controller-observer design for robots. IEEE Trans. Robot. Autom. **9**(6), 740–754 (1993)
56. G. Antonelli, S. Chiaverini, Adaptive tracking control of underwater vehicle-manipulator systems, *IEEE Conference on Control Applications*, Trieste, Italy, Sept 1998, pp. 1089–1093
57. J. Yuh, Modeling and control of underwater robotic vehicles. IEEE Trans. Syst. Man Cybern. **20**(6), 1475–1483 (1990)
58. T. Fossen, *Guidance and Control of Ocean Vehicles* (Wiley, Chichester, 1994)
59. H. Khalil, *Nonlinear Systems* (Prentice Hall, Upper Saddle River, 1996)
60. S.W. Shepperd, Quaternion from rotation matrix. J. Guid. Control **1**, 223 (1978)
61. J.J. Slotine, W. Li, Adaptive strategies in constrained manipulation, in *Proceedings of 1987 IEEE International Conference on Robotics and Automation*, IEEE, Raleigh, NC, 1987, pp. 595–601
62. W.H. Zhu, Y.-G. Xi, Z.-J. Zhang, Z. Bien, J. De Schutter, Virtual decomposition based control for generalized high dimensional robotic systems with complicated structure. IEEE Trans. Robot. Autom. **13**(3), 411–436 (1997)
63. W.H. Zhu, *Virtual Decomposition Control: Toward Hyper Degrees of Freedom Robots*, vol. 60 (Springer, New York, 2010)
64. W.H. Zhu, T. Lamarche, E. Dupuis, D. Jameux, P. Barnard, G. Liu, Precision control of modular robot manipulators: the VDC approach with embedded FPGA. IEEE Trans. Robot. (2013)
65. G. Antonelli, F. Caccavale, S. Chiaverini, A virtual decomposition based approach to adaptive control of underwater vehicle-manipulator systems, in *9th Mediterranean Conference on Control and Automation*, Dubrovnik, HR, June 2001
66. G. Antonelli, F. Caccavale, S. Chiaverini, Adaptive tracking control of underwater vehicle-manipulator systems based on the virtual decomposition approach. IEEE Trans. Robot. Autom. **20**(3), 594–602 (June 2004)
67. R.E. Roberson, R. Schwertassek, *Dynamics of Multibody Systems*, vol. 18 (Springer, Berlin, 1988)

68. L.L. Whitcomb, D. Yoerger, Development, comparison, and preliminary experimental validation of nonlinear dynamic thruster models. IEEE J. Oceanic Eng. **24**(4), 481–494 (1999)
69. G. Antonelli, S. Chiaverini, Task-priority redundancy resolution for underwater vehicle-manipulator systems, in *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, Leuven, May 1998, pp. 768–773
70. G. Antonelli, S. Chiaverini, in *Fuzzy Inverse Kinematics for Underwater Vehicle-Manipulator Systems*, ed. by J. Lenarčič, M.M. Stanišić. Advances in Robot Kinematics. 7th International Symposium on Advances in Robot Kinematics, Piran-Portorož, SLO, June 2000 (NL Kluwer Academic Publishers, Dordrecht, 2000), pp. 249–256
71. G. Antonelli, S. Chiaverini, A fuzzy approach to redundancy resolution for underwater vehicle-manipulator systems, in *Proceedings 5th IFAC Conference on Manoeuvring and Control of Marine Craft*, Aalborg, Denmark, Aug 2000
72. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, A modular control law for underwater vehicle-manipulator systems adapting on a minimun set of parameters, in *15th Ifac World Congress*, Barcelona, Spain, July 2002
73. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, A novel adaptive control law for autonomous underwater vehicles, in *Proceedings of 2001 IEEE International Conference on Robotics and Automation*, Seoul, KR, May, 2001, pp. 447–451
74. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, On the use of integral control actions for autonomous underwater vehicles, in *2001 European Control Conference*, Porto, Sept 2001
75. G. Antonelli, F. Caccavale, S. Chiaverini, G. Fusco, A novel adaptive control law for underwater vehicles. IEEE Trans. Control Syst. Technol. **11**(2), 221–232 (2003)

Chapter 8
Interaction Control of UVMSs

8.1 Introduction to Interaction Control of Robots

In view of the development of an underwater vehicle able to perform a completely autonomous mission the capability of the vehicle to interact with the environment by the use of a manipulator is of greatest interest. To this aim, control of the force exchanged with the environment must be properly investigated.

Underwater Vehicle-Manipulator Systems are complex systems characterized by several strong constraints that must be taken into account when designing a force control scheme:

- Uncertainty in the model knowledge;
- Complexity of the mathematical model;
- Structural redundancy of the system;
- Difficulty to control the vehicle in hovering;
- Dynamic coupling between vehicle and manipulator;
- Low sensors' bandwidth.

Limiting our attention to elastically compliant, frictionless environments several control schemes have been proposed in the literature. An overview of interaction control schemes can be found, e.g., in [1–5].

Stiffness control is obtained by adopting a suitable position control scheme when in contact with the environment [6]. In this case, it is not possible to give a reference force value; instead, a desired stiffness attitude of the tip of the manipulator is assigned. Force and position at steady state depend on the relative compliance between the environment and the manipulator.

Impedance control allows to achieve the behavior of a given mechanical impedance at the end effector rather than a simple compliance attitude [3, 7]. In this case, while it is not possible to give a reference force value, the force measure at the end effector is required to achieve a decoupled behavior.

To allow the implementation of a control scheme that fulfills contact force regulation one should rather consider *direct force control* [8]. This can be effectively

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_8,

obtained by closing an *external force feedback* loop around a position/velocity feedback loop [9], since the output of the force controller becomes the reference input to the standard motion controller of the manipulator.

Nevertheless, many manipulation tasks require simultaneous control of both the end-effector position and the contact force. This in turn demands exact knowledge of the environment geometry: the force reference, in fact, must be consistent with the contact constraints [10].

A first strategy is the *hybrid force/position control* [11]: the force and position controllers are structurally decoupled according to the analysis of the geometric constraints to be satisfied during the task execution. These control schemes require a detailed knowledge of the environment geometry, and therefore are unsuitable for use in poorly structured environments and for handling the occurrence of unplanned impacts.

To overcome this problem, the *parallel force/position control* can be adopted [12]. In this case, position and force loops are closed in all task-space directions, while structural properties of the controller ensure that a properly assigned force reference value is reached at steady state. Since the two loops are not decoupled, a drawback of parallel control is the mutual disturbance of position and force variables during the transient.

Most force control schemes proposed in the literature do not take into account the possible presence of kinematic redundancy in the robotic system, which is always the case of UVMSs. Reference [13] presents a unified approach for motion and force control, extending the formulation to kinematically redundant manipulators. Reference [14] proposes two control schemes, the *extended hybrid control* and the *extended impedance control*, and some experimental results for a 3-link manipulator are provided. In [15] the Operational Space Formulation is experimentally applied to a coordinated task of two mobile manipulators, an experimental comparison for industrial robots is presented in [16]. Reference [17] proposes an *extended hybrid impedance control*. Finally, [18] develops a spatial impedance control for redundant manipulators, and reports experiments with a 7-DOF industrial manipulator. All the above schemes are based on dynamic compensation. However, while dynamic model parameters of land-based manipulators are usually well known, this is not the case of UVMSs for which the application of these schemes is not straightforward.

In [19] a very specific problem is approach, a floating vehicle, with a manipulator mounted on board, uses its thrusters and the resorting forces in order to apply the required force at the end effector.

A test bench for laboratory reproduction of underwater interaction is proposed in [20].

8.2 Dexterous Cooperating Underwater 7-DOF Manipulators

AMADEUS (Advanced MAnipulator for DEep Underwater Sampling) was a project funded by the European Community within the MAST III framework program on Marine Technology development. One of the project's objectives includes the realization of a set-up composed by two 7-DOF Ansaldo manipulators to be used in cooperative mode [21]. The controller has been developed by Casalino et al. and it is based on a hierarchy of interacting functional subsystems.

At the lowest level there is the joint motion control, grouped in a functional scheme defined VLLC (Very Low Level Control), one for each manipulator, that receive as input the desired joint velocities and outputs the real joint positions. In case of high bandwidth loop this block behaves as an integrator.

On the top there is the LLC (Low Level Control) that receives as input the desired homogeneous transformation for the end-effector and output the desired joint velocities. Details on the handling of the kinematic singularities can be found in [21, 22] and reference therein.

At an higher level works the MLC (Medium Level Control) that operates completely in the operational space and outputs the desired homogeneous transformation matrix for the LLC. This block has been designed to interact with an operator via an human-computer interface and thus implements specific attention to the use of a expressly developed mouse. Moreover, the case of manipulators cooperation is properly taken into account.

The above approach has been developed using the RealTime Workshop tool of the MATLAB© software package. The controllers are based on C-modules running on distributed CPUs with VxWorks operative system properly synchronized at 5 ms.

Figure 8.1 reports a photo taken during an experiment run in a pool, the two manipulators have a fixed base and cooperate in the transportation of a rigid object. At the best of our knowledge, this experiment still is the unique of this kind ever reported in the literature.

In [23] the coordinated control of several UVMSs holding a rigid object is also taken into account by properly extending the UVMS dynamic control proposed in [24].

8.3 Impedance Control

Cui et al. in [25], propose an application for a classical impedance controller [3, 7].

It is required that the *desired* impedance at the end effector is described by the following

$$\boldsymbol{f}_{e,d} = \boldsymbol{M}_d \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d \tilde{\boldsymbol{x}} \tag{8.1}$$

where $\boldsymbol{M}_d \in \mathbb{R}^{3\times3}$, $\boldsymbol{D}_d \in \mathbb{R}^{3\times3}$ and $\boldsymbol{K}_d \in \mathbb{R}^{3\times3}$ are positive definite matrices. In other words it is required to assign a desired impedance at the end effector expressed as a desired behavior of the linear e.e. force in the earth-fixed frame.

Let recall Eq. (2.81)

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) = \boldsymbol{\tau} + \boldsymbol{J}_{pos}^{\mathrm{T}}(\boldsymbol{q}, \boldsymbol{R}_B^I)\boldsymbol{f}_e \tag{2.81}$$

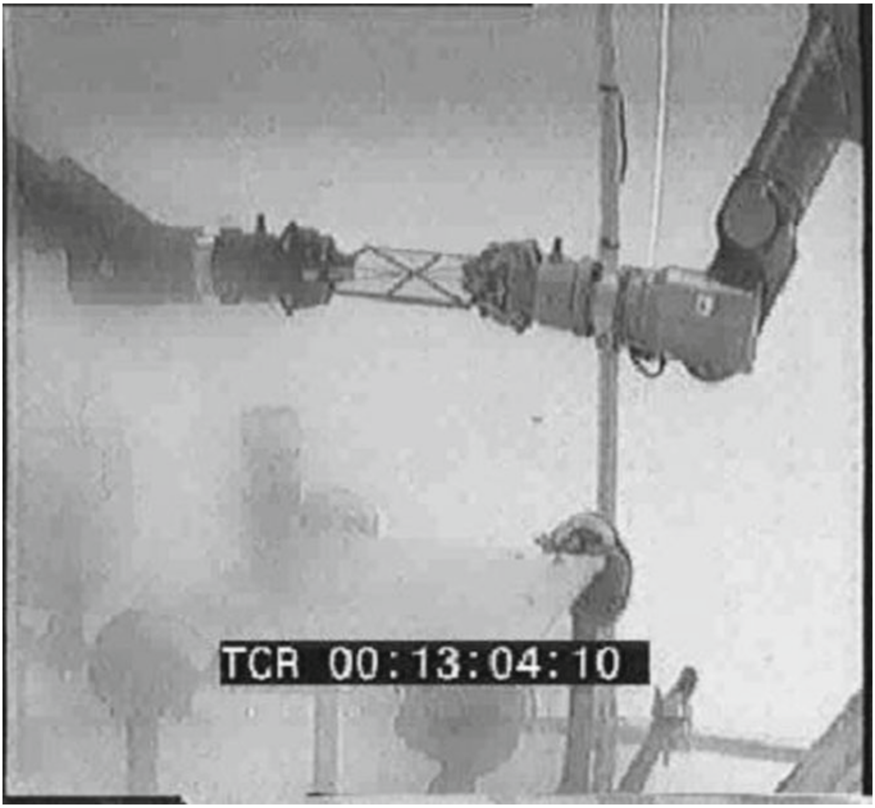


Fig. 8.1 Snapshot of the two 7-link Ansaldo manipulators during a wet test in a pool (courtesy of Casalino, Genoa Robotics And Automation Laboratory, Università di Genova and Veruggio, National Research Council-ISSIA, Italy)

that, by defining

$$\boldsymbol{\tau}_n = \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I)$$

can be written as

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{\tau}_n = \boldsymbol{\tau} + \boldsymbol{J}_{pos}^{\mathrm{T}}(\boldsymbol{q}, \boldsymbol{R}_B^I)\boldsymbol{f}_e \,. \tag{8.2}$$

The second-order differential relationship between the linear end-effector velocity in the earth-fixed frame and the system velocity is given by

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}_{pos}(\boldsymbol{q}, \boldsymbol{R}_B^I)\dot{\boldsymbol{\zeta}} + \dot{\boldsymbol{J}}_{pos}(\boldsymbol{q}, \boldsymbol{R}_B^I)\boldsymbol{\zeta}$$

that, neglecting dependencies, can be inverted in the simplest way as

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{J}_{pos}^{\dagger}\left(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}_{pos}\boldsymbol{\zeta}\right)$$

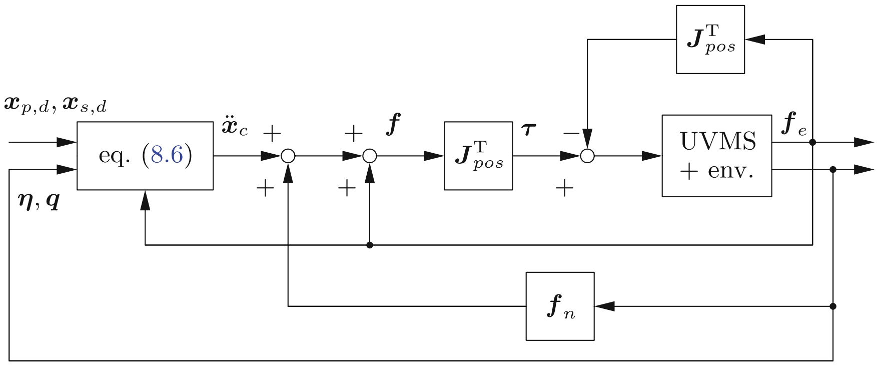

$\boldsymbol{x}_{p,d}, \boldsymbol{x}_{s,d}$
$\boldsymbol{\eta}, \boldsymbol{q}$
eq. (8.6)
$\ddot{\boldsymbol{x}}_c$
+
+
+
+
$\boldsymbol{f}$
$\boldsymbol{J}_{pos}^{\mathrm{T}}$
$\boldsymbol{\tau}$
−
+
$\boldsymbol{J}_{pos}^{\mathrm{T}}$
UVMS
+ env.
$\boldsymbol{f}_e$
$\boldsymbol{f}_n$

Fig. 8.2 Impedance control scheme

that, plugged into (8.2), and defining

$$\boldsymbol{\tau} = \boldsymbol{J}_{pos}^{\mathrm{T}} \boldsymbol{f} \tag{8.3}$$

where $\boldsymbol{f} \in \mathbb{R}^3$ is to be defined, leads to

$$\overline{\boldsymbol{M}} \ddot{\boldsymbol{x}} + \boldsymbol{f}_n = \boldsymbol{f} - \boldsymbol{f}_e \tag{8.4}$$

where $\overline{\boldsymbol{M}} = \boldsymbol{J}_{pos}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{J}_{pos}^{\dagger}$, and $\boldsymbol{f}_n = \boldsymbol{J}_{pos}^{\mathrm{T}} \boldsymbol{\tau}_n - \overline{\boldsymbol{M}} \dot{\boldsymbol{J}}_{pos} \boldsymbol{J}_{pos}^{\dagger} \dot{\boldsymbol{x}}$. The vector $\boldsymbol{f} \in \mathbb{R}^3$ can be selected as

$$\boldsymbol{f} = \hat{\overline{\boldsymbol{M}}} \boldsymbol{x}_c + \boldsymbol{f}_n + \boldsymbol{f}_e \,. \tag{8.5}$$

where

$$\boldsymbol{x}_c = \ddot{\boldsymbol{x}}_d + \boldsymbol{M}_d^{-1} \left(\boldsymbol{D}_d \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d \tilde{\boldsymbol{x}} - \boldsymbol{f}_e \right) \tag{8.6}$$

Details can be found in [25] as well as [3]; Fig. 8.2 represents a block scheme of the impedance approach. In [26] a unified version with the hybrid approach is also proposed.

8.4 External Force Control

In this Section a force control scheme is presented to handle the strong limitations that are experienced in case of underwater systems. Based on the scheme proposed in [27], a force control scheme is analyzed that does not require exact dynamic compensation; however, the knowledge of part of the dynamic model can always be exploited when available. Extension of the original scheme to redundant systems is achieved via a task-priority inverse kinematics redundancy resolution algorithm [28] and suitable secondary tasks are defined to exploit all the degrees of freedom of the system.

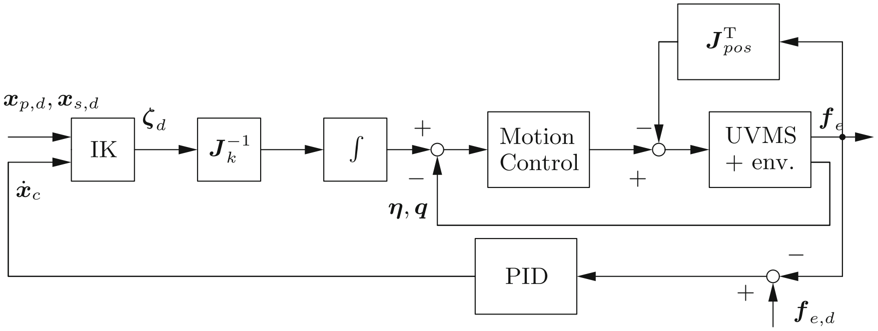


Fig. 8.3 External force control scheme

Force control scheme. A sketch of the implemented scheme is provided in Fig. 8.3. In our case the Inverse Kinematics (i.e., the block IK in the sketch) is solved at the differential level allowing to efficiently handle the system redundancy. The matrix $\boldsymbol{J}_k(\boldsymbol{R}_I^B)$ is the nonlinear, configuration dependent, matrix introduced in Eq. (2.63); its inverse, when resorting to the quaternion attitude representation, is always defined. The matrix $\boldsymbol{J}_{pos}(\boldsymbol{R}_I^B, \boldsymbol{q})$ has been defined in Eq. (2.71). Moreover, a stability analysis for the kinematically redundant case is provided. The Motion Control block can be any suitable motion control law for UVMS; thus, if a partial knowledge of the system is available, a model based control can be applied.

In Sect. 2.12, the mathematical model of an UVMS in contact with the environment is given.

8.4.1 Inverse Kinematics

Let m be the number of degrees of freedom of the mission task. The system DOFs are $6+n$, where 6 are the DOFs of the vehicle and n is the number of manipulator's joints. When $6+n > m$ the system is redundant with respect to the given task and Inverse Kinematics (IK) algorithms can be applied to exploit such redundancy. The IK algorithm implemented is based on a task-priority approach [28] (see also Sect. 6.5), that allows to manage the natural redundancy of the system while avoiding the occurrence of algorithmic singularities. In this phase we can define different secondary tasks to be fulfilled along with the primary task as long as they do not conflict. For example, we can ask the system not to change the vehicle orientation, or not to use the vehicle at all as long as the manipulator is working in a dexterous posture.

Let us define $\boldsymbol{x}_p$ as the primary task vector and $\boldsymbol{x}_s$ as the secondary task vector. The frame in which they are defined depends on the variables we are interested in: if the primary task is the position of the end-effector it should be normally defined in the inertial frame.

The task priority inverse kinematics algorithm is based on the following update law [28]

$$\begin{aligned}\boldsymbol{\zeta}_d = \boldsymbol{J}_p^{\#}\left[\dot{\boldsymbol{x}}_{p,d} + \boldsymbol{\Lambda}_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}_p)\right] \\ + (\boldsymbol{I} - \boldsymbol{J}_p^{\#}\boldsymbol{J}_p)\boldsymbol{J}_s^{\#}\left[\dot{\boldsymbol{x}}_{s,d} + \boldsymbol{\Lambda}_s(\boldsymbol{x}_{s,d} - \boldsymbol{x}_s)\right],\end{aligned} \tag{8.7}$$

where $\boldsymbol{J}_p$ and $\boldsymbol{J}_s$ are the configuration-dependent primary and secondary task Jacobians respectively, the symbol $^{\#}$ denotes any kind of matrix inversion (e.g. Moore-Penrose), and $\boldsymbol{\Lambda}_p$ and $\boldsymbol{\Lambda}_s$ are positive definite matrices. It must be noted that $\boldsymbol{x}_p$, $\boldsymbol{x}_s$ and the corresponding Jacobians $\boldsymbol{J}_p$, $\boldsymbol{J}_s$ are functions of $\boldsymbol{\eta}_d$ and $\boldsymbol{q}_d$. The vector $\boldsymbol{\zeta}_d$ is defined in the body-fixed frame and a suitable integration must be applied to obtain the desired positions: $\boldsymbol{\eta}_d$, $\boldsymbol{q}_d$ (see Eq. 2.63).

Let us assume that the primary task is the end-effector position that implies that $\boldsymbol{J}_p = \boldsymbol{J}_{pos}$. To introduce a force control action Eq. (8.7) is modified as follow: the vector $\dot{\boldsymbol{x}}_c$ is given as a reference value to the IK algorithm and the reference system velocities are computed as:

$$\begin{aligned}\boldsymbol{\zeta}_d = \boldsymbol{J}_p^{\#}\left[\dot{\boldsymbol{x}}_{p,d} + \dot{\boldsymbol{x}}_c + \boldsymbol{\Lambda}_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}_p)\right] \\ + (\boldsymbol{I} - \boldsymbol{J}_p^{\#}\boldsymbol{J}_p)\boldsymbol{J}_s^{\#}\left[\dot{\boldsymbol{x}}_{s,d} + \boldsymbol{\Lambda}_s(\boldsymbol{x}_{s,d} - \boldsymbol{x}_s)\right],\end{aligned} \tag{8.8}$$

where

$$\dot{\boldsymbol{x}}_c = k_{f,p}\tilde{\boldsymbol{f}}_e - k_{f,v}\dot{\boldsymbol{f}}_e + k_{f,i}\int_0^t \tilde{\boldsymbol{f}}_e(\sigma)d\sigma \tag{8.9}$$

being $\tilde{\boldsymbol{f}}_e = \boldsymbol{f}_{e,d} - \boldsymbol{f}_e$ the force error. The direction in which force control is expected are included in the primary task vector, e.g., if the task requires to exert a force along z, the primary task includes the z component of $\boldsymbol{x}$.

8.4.2 Stability Analysis

Pre-multiplying Eq. (8.8) by $\boldsymbol{J}_p \in \mathbb{R}^{3\times(6+n)}$ we obtain:

$$\boldsymbol{J}_p\boldsymbol{\zeta}_d = \dot{\boldsymbol{x}}_{p,d} + \dot{\boldsymbol{x}}_c + \boldsymbol{\Lambda}_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}_p). \tag{8.10}$$

Let us consider a regulation problem, i.e., the reference force $\boldsymbol{f}_{e,d}$ and the desired primary task $\boldsymbol{x}_{p,d}$ are constant. In addition, let us assume that $\boldsymbol{f}_{e,d} \in \mathcal{R}(\boldsymbol{K})$, where $\boldsymbol{K}$ is the stiffness matrix defined in (2.83), and that, as common in external force control approach, the motion controller guarantees perfect tracking, yielding $\boldsymbol{J}_p\boldsymbol{\zeta}_d = \dot{\boldsymbol{x}}$. We finally choose $\boldsymbol{\Lambda}_p = \lambda_p\boldsymbol{I}_m$. Thus, Eq. (8.10) can be rewritten as

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_c + \lambda_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}). \tag{8.11}$$

In view of the above assumptions the vector $\dot{\boldsymbol{x}}_c$ belongs to $\mathcal{R}(\boldsymbol{K})$; it is then simple to recognize that the motion component of the dynamics along the normal direction to the surface is decoupled from the motion components lying onto the contact plane. Therefore, it is convenient to multiply Eq. (8.11) by the two orthogonal projectors $\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}$ and $\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}$ (see Fig. 2.8), that gives the two decoupled dynamics

$$\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_c + \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\lambda_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}) \tag{8.12}$$

$$(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\dot{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\lambda_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}) \tag{8.13}$$

Equation (8.13) clearly shows convergence of the components of $\boldsymbol{x}$ tangent to the contact plane to the corresponding desired values when $\lambda_p > 0$.

To analyze convergence of Eq. (8.12), by differentiating Eq. (8.12) and by taking into account (2.82) and (8.9) we obtain the scalar equation

$$(1 + k_{f,v}k)\ddot{w} + (\lambda_p + k_{f,p}k)\dot{w} + k_{f,i}kw = k_{f,i}\boldsymbol{n}^{\mathrm{T}}(\boldsymbol{f}_d + k\boldsymbol{x}_\infty)\,, \tag{8.14}$$

where the variable $w = \boldsymbol{n}^{\mathrm{T}}\boldsymbol{x}$ is used for notation compactness and k, as defined in the modeling Chapter, is the environment stiffness.

Equation (8.14) shows that, with a proper choice of the control parameters, the component of $\boldsymbol{x}$ normal to the contact plane converges to the value

$$w_\infty = \boldsymbol{n}^{\mathrm{T}}(\frac{1}{k}\boldsymbol{f}_{e,d} + \boldsymbol{x}_\infty).$$

In summary, the overall system converges to the equilibrium

$$\boldsymbol{x}_\infty = \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}(\frac{1}{k}\boldsymbol{f}_{e,d} + \boldsymbol{x}_\infty) + (\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\boldsymbol{x}_{p,d},$$

which can be easily recognized to ensure $\boldsymbol{f}_{e,\infty} = \boldsymbol{f}_{e,d}$.

8.4.3 Robustness

In the following, the robustness of the controller to react against unexpected impacts or errors in planning desired force/position directions is discussed.

The major drawback of hybrid control is that it is not robust to the occurrence of an impact in a direction where motion control has been planned. In such a case, in fact, the end effector is not compliant along that direction and strong interaction between manipulator and environment is experienced. This problem has been solved by resorting to the parallel approach [12] where the force control action overcomes the position control action at the contact.

The proposed scheme shows the same feature as the parallel control. In detail, if a contact occurs along a motion direction, we can see from Eq. (8.8) that the integral action of the force controller guarantees a null force error at steady state while a non-null position error $\boldsymbol{x}_d - \boldsymbol{x}$ is obtained. This implies that, in the direction where a null desired force is commanded, the manipulator reacts to unexpected impacts with a safe behavior. Moreover, in the directions in which a desired force is commanded, the desired position is overcome by the controller.

If $\boldsymbol{f}_{e,d} \notin R(\boldsymbol{K})$, i.e., the direction of $\boldsymbol{f}_{e,d}$ is not parallel to $\boldsymbol{n}$, the controller is trying to interact with the environment in directions in which the environment cannot generate reaction forces. A drift in that direction is then experienced.

8.4.4 Loss of Contact

Due to the floating base and the possible occurrence of external disturbances (such as, e.g., ocean current), it can happen that the end effector loses contact with the environment during the task fulfillment. In such a case the control action might become unsuitable. In fact, from (8.8) it can be noted that the desired force is interpreted as a motion reference velocity scaled by the force control gain. One way to handle this problem is the following: if the force sensor does not read any force value in the desired contact direction, the integrator in the force controller is reset and (8.8) is modified as follows:

$$\begin{aligned}\zeta_d = \boldsymbol{J}_p^{\#} &\left\{\boldsymbol{H}\left[\dot{\boldsymbol{x}}_{p,d} + \dot{\boldsymbol{x}}_c + \boldsymbol{\Lambda}_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}_p)\right] + (\boldsymbol{I} - \boldsymbol{H})\,\dot{\boldsymbol{x}}_l\right\}\\ &+ (\boldsymbol{I} - \boldsymbol{J}_p\boldsymbol{J}_p^{\#})\boldsymbol{J}_s^{\#}\left[\dot{\boldsymbol{x}}_{s,d} + \boldsymbol{\Lambda}_s(\boldsymbol{x}_{s,d} - \boldsymbol{x}_s)\right],\end{aligned} \tag{8.15}$$

where $\boldsymbol{H} \in \mathbb{R}^{m\times m}$ is a diagonal selection matrix with ones for the motion directions and zeros for the force directions, and $\dot{\boldsymbol{x}}_l$ is a desired velocity at which the end effector can safely impact the environment. Basically, the IK is handling the motion directions in the same way as when the contact occurs but, for the force direction, a reference velocity is given in a way to obtain again the contact. Notice that $\dot{\boldsymbol{x}}_c$ is not dropped out in case of loss of contact because it guarantees from unexpected contacts in the direction where motion control is expected (i.e., the directions in which the desired force is zero).

Matrix $\boldsymbol{H}$ can be interpreted as the selection matrix of a hybrid control scheme. Notice that this matrix is used only when there is *no contact* at the end effector and the properties of robustness discussed above still hold.

8.4.5 Implementation Issues

The implementation of the proposed force control scheme might benefit from some practical considerations:

- The vehicle and the manipulator are characterized by a different control bandwidth, due to the different inertia and actuator performance. Moreover limit cycles in underwater vehicles are usually experienced due to the thruster's characteristics. This implies that the use of the desired configuration in the kinematic errors computation in (8.8) would lead to coupling between the force and motion directions, since the Jacobian matrix is computed with respect to a position different from the actual position. In (8.8), thus, the real positions will be used to compute the errors.
- Force control tasks require accurate positioning of the end effector. On the other hand, the vehicle, i.e., the base of the manipulator, is characterized by large position errors. In (8.8), thus, it could be appropriate to decompose the desired end-effector velocity in a way to involve the manipulator alone in the fulfillment of the primary task. Let us define $\boldsymbol{J}_{p,man}(\boldsymbol{R}_B^I, \boldsymbol{q})$ as the Jacobian of the manipulator, it is possible to rewrite (8.8) as

$$\begin{aligned}\zeta_d = &\begin{bmatrix} \mathbf{0}_{6\times 1} \\ \boldsymbol{J}^{\#}_{p,man}\left[\dot{\boldsymbol{x}}_{p,d} + \dot{\boldsymbol{x}}_c + \boldsymbol{\Lambda}_p(\boldsymbol{x}_{p,d} - \boldsymbol{x}_p)\right] \end{bmatrix} \\ &+ (\boldsymbol{I} - \boldsymbol{J}^{\#}_p \boldsymbol{J}_p)\boldsymbol{J}^{\#}_s\left[\dot{\boldsymbol{x}}_{s,d} + \boldsymbol{\Lambda}_s(\boldsymbol{x}_{s,d} - \boldsymbol{x}_s)\right]. \end{aligned} \tag{8.16}$$

  Notice that the same properties of (8.8) applies also for (8.16). A physical interpretation of (8.16) is the following: it is *asked* the manipulator to fulfill the primary task taking into account the movement of its base. At the same time the secondary task is fulfilled, with less strict requirements, with the whole system (e.g., the vehicle must move when the manipulator is working on the boundaries of its workspace).
- UVMSs are usually highly redundant. If a 6-DOF manipulator is used this means that 12 DOFs are available. It is then possible to define more tasks to be iteratively projected on the null space of the higher priority tasks. An example of 3 tasks could be: (1) motion/force control of the end effector, (2) increase manipulability measure of the manipulator, (3) limit roll and pitch orientation of the vehicle. See also Chap. 6.
- To decrease power consumption it is possible to implement IK algorithms with bounded reference values for the secondary tasks. Using some smooth functions, or fuzzy techniques, it is possible to activate the secondary tasks only when the relevant variables are out of a desired range. For the roll and pitch vehicle's angles, for example, it might be convenient to implement an algorithm that keeps them in a range, e.g., ±10°. See also Chap. 6.
- Force/moment sensor readings are usually corrupted by noise. The use of a derivative action in the control law, thus, can be difficult to implement. With the assumption of a frictionless and elastically compliant plane it can be observed a linear relation between $\dot{\boldsymbol{f}}_e$ and $\dot{\boldsymbol{x}}$. The force derivative action can then be substituted by a term proportional to $\dot{\boldsymbol{x}}$. The latter will be computed by differential kinematics from $\boldsymbol{\zeta}$ that is usually available from direct sensor readings or from the position readings via a numerical filter.

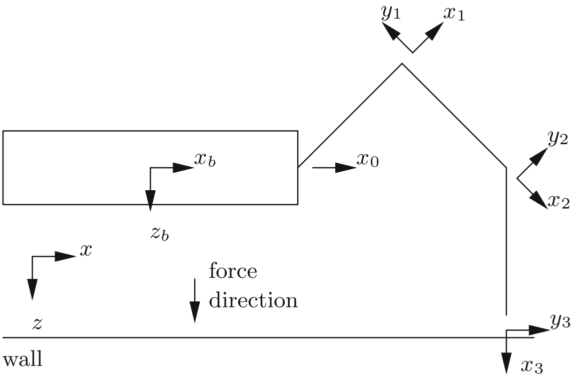


Fig. 8.4 Sketch of the simulated system for both controllers as seen from the xz vehicle-fixed plane

8.4.6 Simulations

To prove the effectiveness of the proposed force control scheme several simulations have been run under MATLAB©/SIMULINK© environment. The UVMS simulated has 9 DOFs, 6 DOFs of the vehicle plus a 3-link manipulator mounted on it [29]. The controller has been implemented with a sampling frequency of 200 Hz. The vehicle is a box of dimensions (2 × 1 × 0.5) m, and the vehicle fixed frame is located in the geometrical center of the body. The manipulator has a planar structure with 3 rotational joints. A sketch of the system seen from the vehicle xz plane is shown in Fig. 8.4. In the shown configuration it is $\boldsymbol{\eta} = \begin{bmatrix}0 \ 0 \ 0 \ 0 \ 0 \ 0\end{bmatrix}^{\mathrm{T}}$ m, deg, $\boldsymbol{q} = \begin{bmatrix}45 \ -90 \ -45\end{bmatrix}^{\mathrm{T}}$ deg. The stiffness of the environment is $k = 10^4$ N/m. In the Appendix some details on the dynamic model are given.

The vehicle is supposed to start in a non dexterous configuration, so as to test the redundancy resolution capabilities of the proposed scheme. The initial configuration is:

$$\boldsymbol{\eta} = \begin{bmatrix}0 \ 0 \ -2 \ 12 \ 0 \ 0\end{bmatrix}^{\mathrm{T}} \text{ m, deg,}$$
$$\boldsymbol{q} = \begin{bmatrix}-90 \ 15 \ 0\end{bmatrix}^{\mathrm{T}} \text{ deg}$$

that corresponds to the end-effector position

$$\boldsymbol{x} = \begin{bmatrix}x_E \ y_E \ z_E\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}1.51 \ -0.60 \ 0.86\end{bmatrix}^{\mathrm{T}} \text{ m.}$$

Since the vehicle is far from the plane ($z = 1.005$ m), the manipulator is outstretched, i.e., close to a kinematic singularity.

The simulated task is to perform a force/position task at the end effector as primary task; specifically, the UVMS is required to move −20 cm along x and apply a desired force of 200 N along z. The secondary tasks are to guarantee manipulator manipulability and to keep roll and pitch vehicle's angles in a safe range of ±10°.

In detail, the 3 task variables are:

$$\begin{aligned}\boldsymbol{x}_p &= \boldsymbol{x}\,,\\ \boldsymbol{x}_s &= \left[\phi\ \theta\right]^{\mathrm{T}},\\ x_t &= q_2\,,\end{aligned}$$

where x_t expresses the third task to be projected in the null space of the higher priority tasks; in fact since the manipulator has a 3-link planar structure a measure of its manipulability is simply given by q_2, where $q_2 = 0$ corresponds to a kinematic singularity. In the initial position the 3 tasks are activated simultaneously and are performed by exploiting kinematic redundancy. Moreover an unexpected impact along x is considered (for $x_E = 1.32\,\mathrm{m}$).

A weighted pseudoinverse has been used to compute $\boldsymbol{J}_p^{\#}$ characterized by the weight matrix $\boldsymbol{W} = \mathrm{blockdiag}\{10\boldsymbol{I}_6, \boldsymbol{I}_3\}$. To simulate an imperfect hovering of the vehicle a control law with *lower* gain for the vehicle was implemented; the performance of the simulated vehicle, thus, has an error that is of the same magnitude as that of a real vehicle in hovering. The motion controller implemented is the virtual decomposition adaptive based control presented in Sect. 7.9. Equation (8.16) has then been used to compute the desired velocities. The secondary task regarding the vehicle orientation has to be fulfilled only when the relevant variable is outside of a desired bound.

The control gains, in S.I. units, are:

$$\begin{aligned}k_{fp} &= 3 \cdot 10^{-3}\,,\\ k_{fi} &= 8 \cdot 10^{-3}\,,\\ k_{fv} &= 10^{-4}\,,\\ \boldsymbol{\Lambda}_p &= 0.6\boldsymbol{I}_3\,,\\ \boldsymbol{\Lambda}_s &= \boldsymbol{I}_3\,.\end{aligned}$$

The simulations have been run by adopting separate motion control schemes for the vehicle and the manipulator, since this is the case of many UVMSs. Better results would be obtained by resorting to a centralized motion control scheme in which dynamic coupling between vehicle and manipulator is compensated for. The initial value of the parameters has been chosen such that the gravity compensation at the beginning is different from the real one, adding an error bounded to about 10 % for each parameter.

In Fig. 8.5 the time history of the end-effector variables for the proposed force control scheme without exploiting the redundancy and without unexpected impact is shown. During the first 3 s the end effector is not in contact with the plane and the algorithm to handle loss of contact has been used. It can be noted that the primary task is successfully achieved.

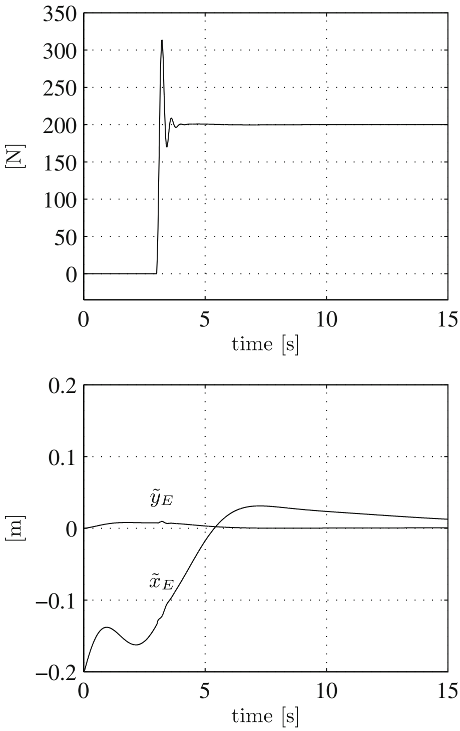


Fig. 8.5 External force control. Force along z (*top*) and end-effector error components along the motion directions (*bottom*) without exploiting the redundancy and without unexpected impact

In Fig. 8.6 the same task has been achieved by exploiting the redundancy. The different behavior of the force can be explained by considering that the system impacts the plane in a different configuration with respect to the previous case because of the internal motion imposed by the redundancy resolution. This also causes a different end-effect velocity at the impact.

Figure 8.7 shows the time history of the secondary tasks in the two previous simulations, without exploiting redundancy (solid) and with the proposed control scheme (dashed). It can be recognized that without exploiting the redundancy the system performs the task in a non dexterous configuration, i.e., with a big roll angle and with the manipulator close to a kinematic singularity. A suitable use of the system's redundancy allows to reconfigure the system and to achieve the secondary task.

Finally, in Fig. 8.8 the time history of the primary task variables in case of an unplanned impact is shown. It can be noted that the undesired force along x, due to the unexpected impact, is recovered yielding a non-zero position error along that direction.

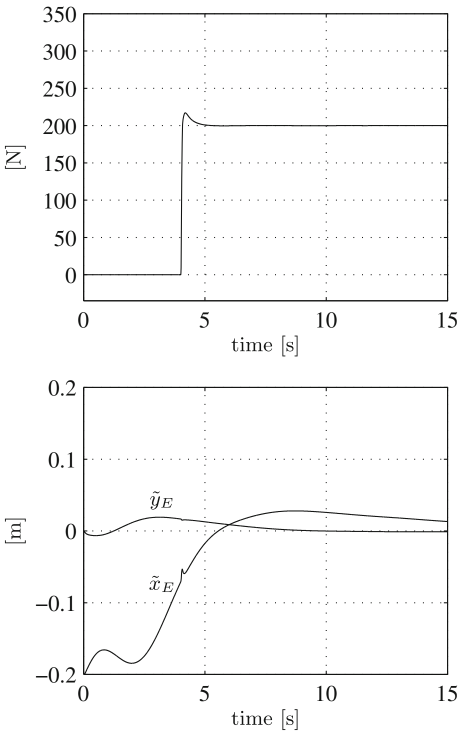


Fig. 8.6 External force control. Force along z (*top*) and end-effector error components along the motion directions (*bottom*) exploiting the redundancy and without unexpected impact

8.5 Explicit Force Control

In this Section, based on [30] two force control schemes for UVMS that overcome many of the above-mentioned difficulties associated with the underwater manipulation are presented. These force control schemes exploit the system redundancy by using a task-priority based inverse kinematics algorithm [28]. This approach allows us to satisfy various secondary criteria while controlling the contact force. Both these force control schemes require separate motion control schemes which can be chosen to suit the objective without affecting the performance of the force control. The possible occurrence of loss of contact due to vehicle movement is also analyzed. The proposed control schemes have extensively been tested in numerical simulation runs; the results obtained in a case study are reported to illustrate their performance.

Two different versions of the scheme are implemented based on different projections of the force error from the task space to the vehicle/joint space.

The first scheme is obtained by using the transpose of the Jacobian to project the force error from the task space directly to the control input space, i.e., force/moments for the vehicle and torques for the manipulator, leading to an evident physical interpretation.

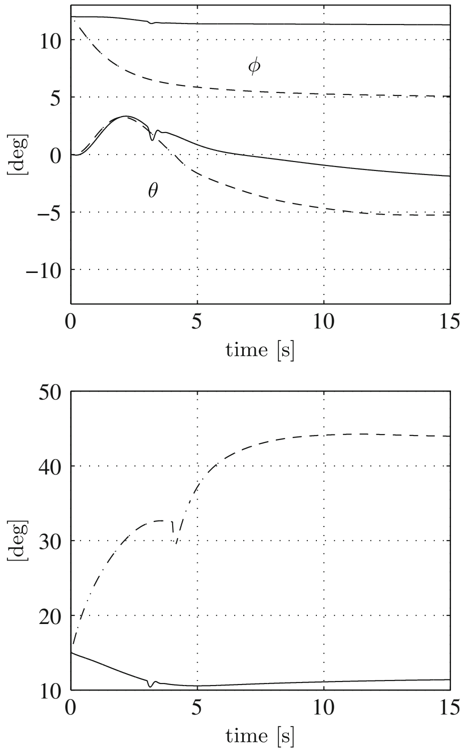


Fig. 8.7 External force control. Roll and pitch vehicle's angles (*top*) and q_2 as manipulability measure (*bottom*). *Solid* without exploiting redundancy; *Dashed* exploiting redundancy with the proposed scheme

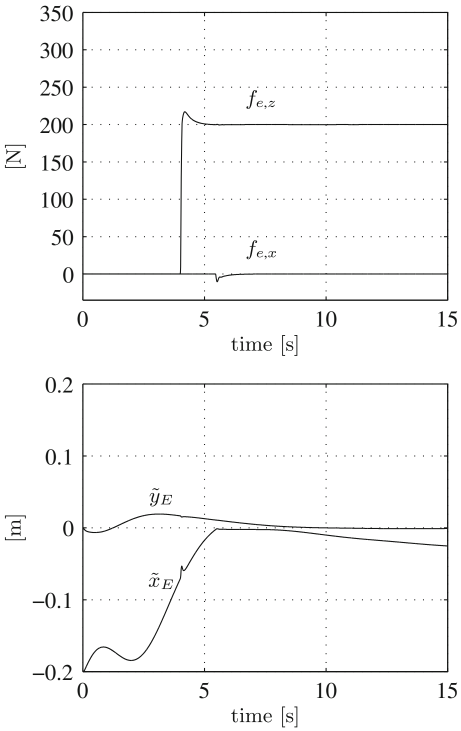
In the second scheme, instead, the force error is projected from the task space to the body-fixed velocities. This is done to avoid the need to directly access the control input; in many cases, in fact, a velocity controller is implemented on the manipulator and control torques are not accessible [30].

For both the control schemes, the kinematic control applied is the same as the algorithm exploited in the external force control scheme, already shown in Sect. 8.4.1.

Explicit force control, scheme 1. A sketch of the implemented scheme is provided in Fig. 8.9. In our case, the inverse kinematics is solved via a numerical algorithm based on velocity mapping to allow the handling of system redundancy efficiently (see previous Section and Sect. 6.5). It is possible to decompose the input torque as the sum of the torque output from the motion control and the torque output from the force control: $\boldsymbol{\tau} = \boldsymbol{\tau}^M + \boldsymbol{\tau}^F$. The force control action $\boldsymbol{\tau}^F$ is computed as:

$$\begin{aligned} \boldsymbol{\tau}^F &= \boldsymbol{J}_{pos}^{\mathrm{T}} \boldsymbol{u}_F \\ &= \boldsymbol{J}_{pos}^{\mathrm{T}} \left(-\boldsymbol{f}_e + k_{f,p} \tilde{\boldsymbol{f}}_e - k_{f,v} \dot{\boldsymbol{f}}_e + k_{f,i} \int_0^t \tilde{\boldsymbol{f}}_e(\sigma) d\sigma \right), \end{aligned} \tag{8.17}$$

Fig. 8.8 External force control. Force along *z* and *x* (*top*) and end-effector error components along the motion directions (*bottom*) exploiting the redundancy and with unexpected impact along *x*



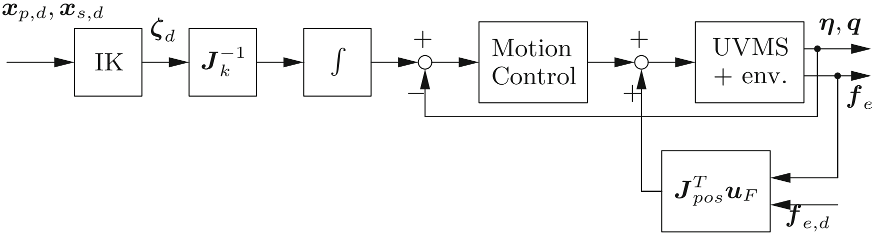

$\boldsymbol{x}_{p,d}, \boldsymbol{x}_{s,d}$
IK
ζ_d
$\boldsymbol{J}_k^{-1}$
$\int$
+
−
Motion Control
+
+
UVMS + env.
$\boldsymbol{\eta}, \boldsymbol{q}$
$\boldsymbol{f}_e$
$\boldsymbol{J}_{pos}^T \boldsymbol{u}_F$
$\boldsymbol{f}_{e,d}$

Fig. 8.9 Explicit force control scheme 1

where $k_{f,p}$, $k_{f,v}$, $k_{f,i}$ are scalar positive gains, and $\tilde{\boldsymbol{f}}_e = \boldsymbol{f}_{e,d} - \boldsymbol{f}_e$ is the force error. Equation (8.17), thus, is a force control action in the task space that is further projected, via the transpose of the Jacobian, on the vehicle/joint space.

Explicit force control, scheme 2. A sketch of the implemented scheme is provided in Fig. 8.10. The force control action is composed of two loops; the action $-\boldsymbol{J}_{pos}^{\mathrm{T}} \boldsymbol{f}_e$ is aimed at compensating the end-effector contact force (included in the block labeled "UVMS + env."). The input of the motion control is a suitable integration of the velocity vector:

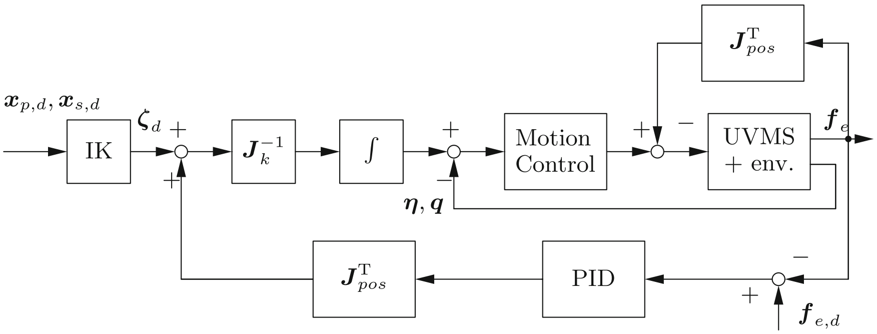


Fig. 8.10 Explicit force control scheme 2

$$\zeta_r = \zeta_d + \zeta_F = \zeta_d + \boldsymbol{J}_{pos}^{\mathrm{T}}(k_{f,p}^{*}\tilde{\boldsymbol{f}}_e - k_{f,v}^{*}\dot{\boldsymbol{f}}_e + k_{f,i}^{*}\int_0^t \tilde{\boldsymbol{f}}_e(\sigma)d\sigma),$$

where $k_{f,p}^{*}, k_{f,v}^{*}, k_{f,i}^{*}$ are positive gains, and ζ_d is the output of the inverse kinematics algorithm described in previous section.

8.5.1 Robustness

In the following, the robustness of the schemes to react against unexpected impacts or errors in planning desired force/position directions is discussed.

The use of the integral action in the force controller gives a higher priority to the force error with respect to the position error. In the motion control directions the desired force is null. An unexpected impact in a direction where the desired force is zero, thus, is handled by the controller yielding a safe behavior, which results in a non-null position error at steady state and a null force error, i.e., zero contact force.

If $\boldsymbol{f}_{e,d} \notin \mathcal{R}(\boldsymbol{K})$, i.e. the direction of $\boldsymbol{f}_{e,d}$ is not parallel to $\boldsymbol{n}$, the controller is commanded to interact with the environment in directions along which no reaction force exists. In this case, a drift motion in that direction is experienced.

These two force control schemes too, possess the safe behavior as the external force control scheme.

The possibility that the end effector loses contact with the environment is also taken into account. The same algorithm as presented in Sect. 8.4.4 were implemented.

8.5.2 Simulations

To test the effectiveness of the proposed force control schemes several numerical simulations were run under the MATLAB©/SIMULINK© environment. The controllers were implemented in discrete time with a sampling frequency of 200 Hz. The environmental stiffness is $k = 10^4$ N/m.

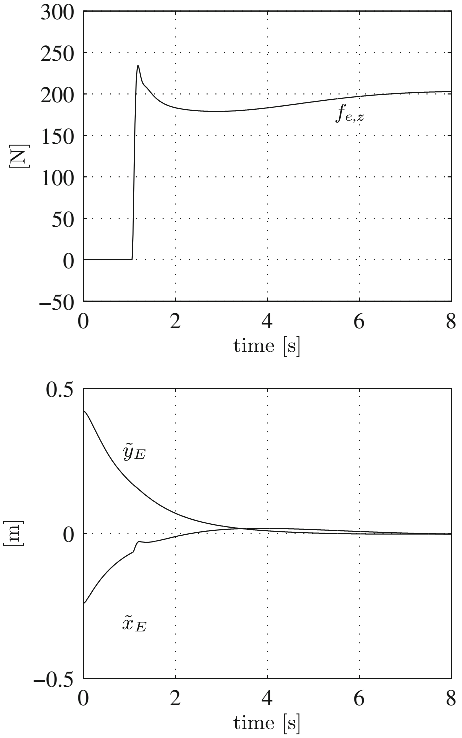


Fig. 8.11 Explicit force control scheme 1. *Top* time history of the contact force. *Bottom* time history of the end-effector errors along the motion directions

The simulated UVMS has 9 DOFs, 6 DOFs of the vehicle plus a 3-link manipulator mounted on it [29]. The vehicle is a box of dimensions $(2 \times 1 \times 0.5)$ m; the vehicle-fixed frame is located in the geometrical center of the body. The manipulator is a 3-link planar manipulator with rotational joints. Figure 8.4 shows a sketch of the system, seen from the vehicle's xz plane, in the configuration

$$\begin{aligned}\boldsymbol{\eta} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \text{ m, deg,} \\ \boldsymbol{q} &= \begin{bmatrix} 45 & -90 & -45 \end{bmatrix}^{\mathrm{T}} \text{ deg}\end{aligned}$$

corresponding to the end-effector position $\boldsymbol{x} = \begin{bmatrix} 2.41 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$ m.

A case study is considered aimed at the following objectives: as primary task, to perform force/motion control of the end effector (exert a force of 200 N along z, moving the end effector from 2.41 to 2.21 m along x, while keeping y at 0 m); as secondary task, to guarantee vehicle's roll and pitch angles being kept in the range ±10°; as tertiary task, to guarantee the manipulator's manipulability being kept in a safe range. In detail, the 3 task variables are:

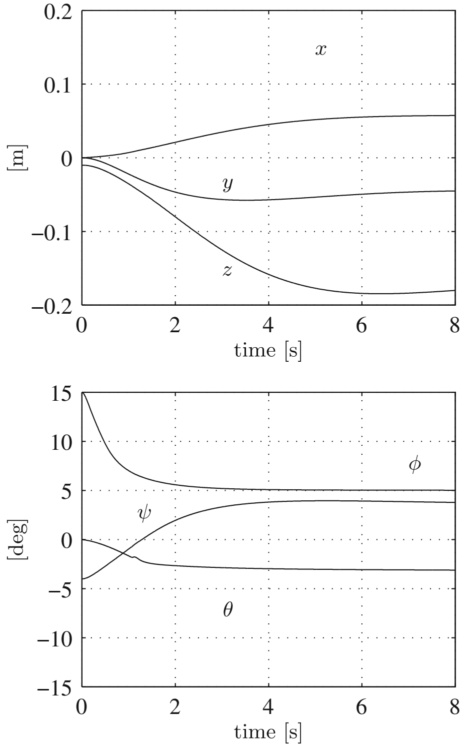


Fig. 8.12 Explicit force control scheme 1. *Top* time history of the vehicle's position. *Bottom* time history of the vehicle's orientation

$$
\begin{aligned}
\boldsymbol{x}_p &= \boldsymbol{x},\\
\boldsymbol{x}_s &= \left[\phi\ \theta\right]^{\mathrm{T}},\\
x_t &= q_2,
\end{aligned}
$$

where x_t expresses the third task; in fact, since the manipulator has a 3-link planar structure, a measure of its manipulability is simply given by q_2, where $q_2 = 0$ corresponds to a kinematic singularity.

Let us suppose that the system starts the mission in the non-dexterous configuration

$$
\begin{aligned}
\boldsymbol{\eta} &= \left[0\ 0\ -0.001\ 15\ 0\ -4\right]^{\mathrm{T}} && \text{m, deg},\\
\boldsymbol{q} &= \left[46\ -89\ -44\right]^{\mathrm{T}} \text{deg}
\end{aligned}
$$

corresponding to the end-effector position $\boldsymbol{x} = \left[2.45\ -0.42\ 0.91\right]^{\mathrm{T}}$ m.

A weighted pseudoinverse was used to compute $\boldsymbol{J}^{\#}$ characterized by the weight matrix $\boldsymbol{W} = \text{blockdiag}\{10\boldsymbol{I}_6, \boldsymbol{I}_3\}$. To simulate an imperfect hovering of the vehicle the control law was implemented with *lower* gains for the vehicle; the performance of the simulated vehicle, thus, has an error that is of the same magnitude of a real vehicle in hovering.

To accomplish the above task, firstly the force control scheme 1 is used together with the sliding mode motion control law described in Sect. 7.6; notice that non-perfect gravity and buoyancy compensation was assumed. Figure 8.11 reports the contact force and the end-effector error components obtained. Since the desired final position is given as a set-point, the initial end-effector position errors are large. In Fig. 8.12, the position and orientation components of the vehicle are shown; it can be recognized that, despite the large starting value, the roll angle is kept in the desired range.

To take advantage of dynamic compensation actions, the force control scheme 2 is used to accomplish the same task as above together with the singularity-free adaptive control presented in Sect. 7.9; notice that only the restoring force terms have been considered to be compensated and a constant unknown error, bounded to ±10 %, of these parameters has been assumed. There are 4 parameters for each rigid body of the UVMS, giving 16 parameters in total. Moreover, during the task execution an unexpected impact occurs along x.

In Fig. 8.13, the contact force and the end-effector error components are shown. It can be recognized that the unexpected impact, occurring at about 3 s, is safely

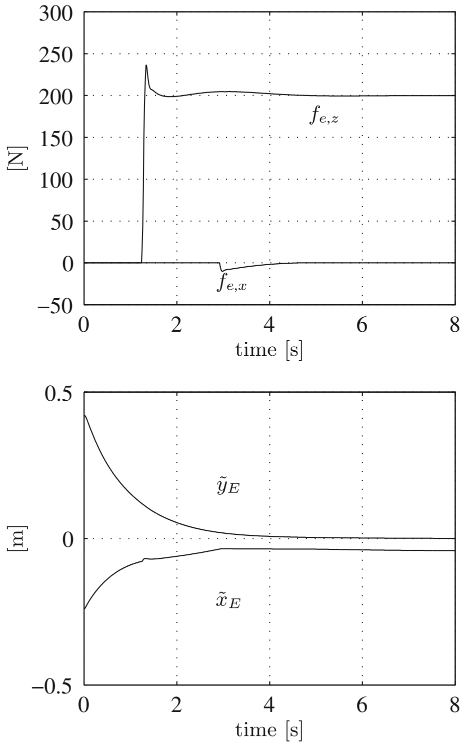


Fig. 8.13 Explicit force control scheme 2. *Top* time history of the contact force in case of unexpected impact along x. *Bottom* time history of the end-effector errors along the motion directions

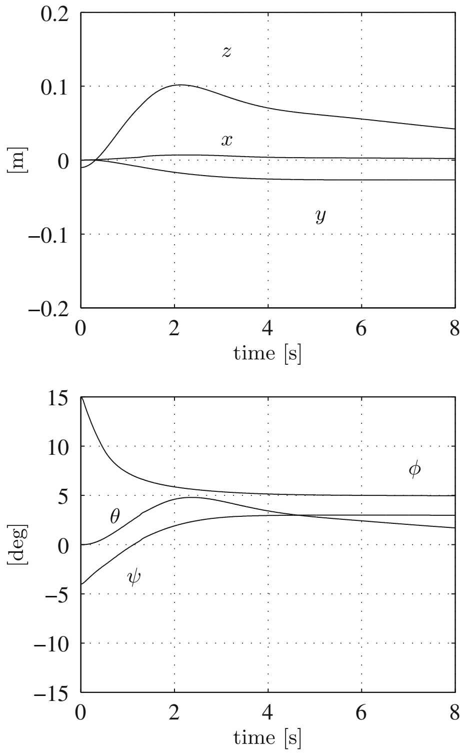


Fig. 8.14 Explicit force control scheme 2. *Top* time history of the vehicle's position. *Bottom* time history of the vehicle's orientation

handled: a transient force component along x is experienced; nevertheless, at the steady state the desired force is achieved with null error.

It can be noted that the expected coupling between the force and the motion directions affects the x direction also before the unplanned impact. This coupling, due to the structure and configuration of the manipulator, is not observed along y.

In Fig. 8.14, the vehicle's position and orientation components are shown. It can be recognized that the vehicle moves by about 10 cm; nevertheless, the manipulator still performs the primary task accurately. Moreover, the large initial roll angle is recovered by exploiting the system redundancy since this task does not conflict with the higher priority task.

In the force control scheme labeled 1 the force error directly modifies the force/moments/torques acting on the UVMS, leading to an evident physical interpretation. In the force control scheme 2, instead, the force error builds a correction term acting on the body-fixed velocity references which fed the available motion control system of the UVMS.

The two proposed control schemes were tested in numerical simulation case studies and their performance is analyzed. Overall, the force control scheme 2 seems to be preferable with respect to the force control scheme 1 for two reasons. Firstly, it allows the adoption of adaptive motion control laws, thus making it possible dynamic compensation actions. Secondly, it naturally embeds the standard motion control of the UVMS, since it acts at the reference motion variables level.

8.6 Conclusions

Interaction control in the underwater environment is a very challenge task. Currently, only theoretical results exist that extended the industrial-based approaches to the UVMSs. Obviously, while these approaches worked in simulation their real effectiveness can be proven only with an exhaustive experimental analysis. Simulations, nevertheless, gave useful information about the need to equip the system with high bandwidth and low noise sensors in order to interact with stiff environment.

References

1. C. Canudas De Wit, G. Bastin, B. Siciliano, *Theory of Robot Control* (Springer, New York Inc., 1996)
2. D. Whitney, Historical perspective and state of the art in robot force control. Int. J. Robot. Res. **6**(1), 3–14 (1987)
3. B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer, London, 2009)
4. B. Siciliano, L. Villani, *Robot Force Control*, vol. 540 (Kluwer Academic Pub, Dordrecht, 1999)
5. L. Villani, J. De Schutter, in *Force Control*, ed. by B. Siciliano, O. Khatib. Springer Handbook of Robotics (Springer-Verlag, Heidelberg, 2008)
6. J.K. Salisbury, Active stiffness control of a manipulator in cartesian coordinates, in *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, vol. 19, pp. 95–100 (Albuquerque, New Mexico, 1980)
7. N. Hogan, Impedance control: an approach to manipulation: Part I-III. J. Dyn. Syst. Meas. Contr. **107**(2), 1–24 (1985)
8. D. Whitney, Force feedback control of manipulator fine motions. ASME J. Dyn. Syst. Meas. Contr. **98**, 91–97 (1977)
9. J. De Schutter, H. Van Brussel, Compliant robot motion II. A control approach based on external control loops. Int. J. Rob. Res. **7**(4), 18–33 (1988)
10. M. Mason, Compliance and force control for computer controlled manipulators. IEEE Trans. Syst. Man Cybern. **11**(6), 418–432 (1981)
11. M.H. Raibert, J.J. Craig, Hybrid position/force control of manipulators. Trans. ASME J. Dyn. Syst. Meas. Contr. **102**, 126–133 (1981)
12. S. Chiaverini, Estimate of the two smallest singular values of the Jacobian matrix: application to damped least-squares inverse kinematics. J. Rob. Syst. **10**(8), 991–1008 (1993)
13. O. Khatib, A unified approach for motion and force control of robot manipulators: The operational space formulation. IEEE J. Rob. Autom. **3**(1), 43–53 (1987)
14. Z.-X. Peng, N. Adachi, Compliant motion control of kinematically redundant manipulators. IEEE Trans. Rob. Autom. **9**(6), 831–836 (1993)
15. O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation, in *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems' 96, IROS 96*, vol. 2, pp. 546–553, Osaka, Japon 1996. IEEE.

16. J. Nakanishi, R. Cory, M. Mistry, J. Peters, S. Schaal, Operational space control: a theoretical and empirical comparison. Int. J. Rob. Res. **27**(6), 737–757 (2008)
17. Y. Oh, W.K. Chung, Y. Youm, I.H. Suh, Motion/force decomposition of redundant manipulator and its application to hybrid impedance control, in *Proceedings of the IEEE International Conference on Robotics and Automation, 1998*, vol. 2 (Leuven, Belgium, 1998), pp. 1441–1446
18. C. Natale, B. Siciliano, L. Villani, Spatial impedance control of redundant manipulators, in *Proceedings of the IEEE International Conference on Robotics and Automation, 1999*, vol. 3 (Detroit, Michigan, 1999), pp. 1788–1793
19. H. Kajita, K. Kosuge, Force control of robot floating on the water utilizing vehicle restoring force, in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1997. IROS'97*, vol. 1 (Grenoble, France, 1997), pp. 162–167
20. S. Lemieux, J. Beaudry, M. Blain, Force control test bench for underwater vehicle-manipulator system applications, in *IECON 2006–32nd Annual Conference on IEEE Industrial Electronics*, 2006, pp. 4036–4042
21. G. Casalino, D. Angeletti, T. Bozzo, G. Marani, Dexterous underwater object manipulation via multi-robot cooperating systems, in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001*, vol. 4 (Seoul, Korea, 2001), pp. 3220–3225
22. G. Casalino, A. Turetta, Coordination and control of multiarm, non-holonomic mobile manipulators, in *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003)*, vol. 3 (Las Vegas, Nevada, 2003), pp. 2203–2210
23. Y.C. Sun, C.C. Cheah, Coordinated control of multiple cooperative underwater vehicle-manipulator systems holding a common load, in *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, vol. 3 (Kobe, Japan, 2004), pp. 1542–1547
24. Y.C. Sun, C.C. Cheah, Adaptive setpoint control of underwater vehicle-manipulator systems, in *2004 IEEE Conference on Robotics, Automation and Mechatronics*, vol. 1 (Singapore, 2004), pp. 434–439
25. Y. Cui, T.K. Podder, N. Sarkar, Impedance control of underwater vehicle-manipulator systems (UVMS), in *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. IROS'99*, vol. 1 (Kyongju, Korea, 1999), pp. 148–153
26. Y. Cui, N. Sarkar, A unified force control approach to autonomous underwater manipulation. Robotica **19**(03), 255–266 (2001)
27. E. Dégoulange, P. Dauchez, External force control of an industrial PUMA 560 robot. J. Rob. Syst. **11**(6), 523–540 (1994)
28. S. Chiaverini, Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Trans. Rob. Autom. **13**(3), 398–410 (1997)
29. T.K. Podder, Dynamic and control of kinematically redundant underwater vehicle-manipulator systems, in *Technical Report ASL 98–01, Autonomous Systems Laboratory Technical Report*, University of Hawaii, Honolulu, Hawaii, 1998
30. G. Ferretti, G. Magnani, P. Rocco, Toward the implementation of hybrid position/force control in industrial robots. IEEE Trans. Rob. Autom. **13**(6), 838–845 (1997)

Chapter 9
Simurv 4.0

Song' 'e fierr ca' fann 'o mast.
Neapolitan *saying.*

9.1 Introduction

SIMURV 4.0 is not a *one-click* simulator, it is rather a libray of functions of general use to test UVMS kinematic and dynamic control algorithms. The code developed is made available to be downloaded at the address:

`http://www.eng.docente.unicas.it/gianluca_antonelli/simurv`

This piece of code has been written with some requirements in mind. First, I needed an "agile" simulator for all the kinematic and dynamic control algorithms described along this book. The focus, thus, is on the word *control* of a mechanical system mathematically described by non linear deterministic differential equations; the bodies are supposed to move in the water at low velocities, all the hydrodinamic terms, thus, may be approximated by their compact expressions without resorting to computational-demanding algorithms based on the discretization of the Navier-Stokes equations.

The user should be able to easily change the controller, tune the gains as well as to modify some modeling parameters. In addition, I did not want to *freeze* the simulator with one single mathematical model, for this reason the symbolic model has not been computed and the inverse dynamics has been implemented with the algorithm described in Sect. 9.4. As a consequence, the user can easily define its own UVMS or modify an existing one.

This simulator ignores totally the perception part and does not consider the possibility to model the environment with, e.g., some additional important features such as impact detection, use of vision, laser or any other exteroceptive sensor. Communication too, is missing from the simulator Fig. 9.1.

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_9,

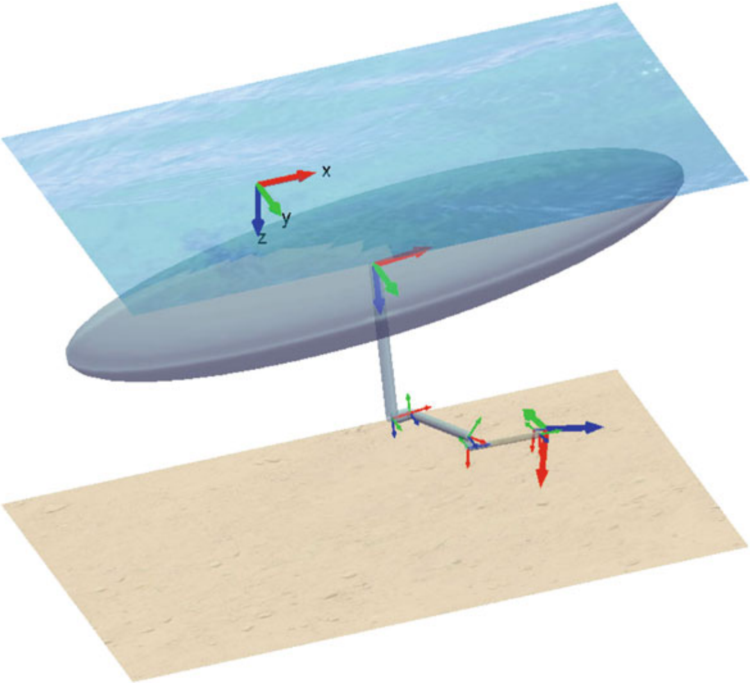


Fig. 9.1 Snapshot of a simulated UVMS equipped with a 6-DOF manipulator

I decided to keep most of the code at *low* level while resorting to a popular programming language for numerical computation such as Matlab© [1]. The result is that the *core* code, i.e., the simulation of the controller and the inverse dynamics, is made by resorting to the basic commands and by using only simple structures. More *high* level or version/machine depended features, such as the graphical representation of the output, are kept separate from the remaining code. This choice leaves the door open for future developments such as, e.g., translation in open source languages such as Octave [2, 3], Scilab [4] or development of a ROS [5] node. At the printing date there is not a schedule for such developments, any help or suggestion is welcome.

There are several other simulators for underwater vehicles only at different levels of complexity, such as, e.g., [6–8]. Concerning UVMS the simulator UWSim [9] (http://www.irs.uji.es/uwsim) developed within the TRIDENT [10] project is available.

9.2 License

SIMURV 4.0 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

SIMURV 4.0 is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with SIMURV 4.0. If not, see `http://www.gnu.org/licenses`.

9.3 Variables' Definition

The basic variables needed to run and understand a simulation are:

Variable	Dimension	Notes
`Ts`	`1 x 1`	Sampling time of the discretization $T_s \in \mathbb{R}$
`t_f`	`1 x 1`	Final simulation time $t_f \in \mathbb{R}$
`npti`	`1 x 1`	Number of simulation samples $n_{pti} \in \mathbb{N}$
`t`	`npti x 1`	Time vector, `t(i)` is $t_i = T * (i - 1)$
`n`	`1 x 1`	Number of joints
`DH`	`n x 4`	Denavit-Hartenberger table
`PARAM`	**`struct`**	Structure with all the dynamic parameters
`eta`	`6 x npti`	Vehicle pos./or. during the simulation `eta(:,i)` is $\boldsymbol{\eta}(t_i) \in \mathbb{R}^6$
`q`	`n x npti`	Joint positions during the simulation `q(:,i)` is $\boldsymbol{q}(t_i) \in \mathbb{R}^n$
`zita`	`6+n x npti`	System velocities during the simulation `zita(:,i)` is $\boldsymbol{\zeta}(t_i) \in \mathbb{R}^{6+n}$
`dzita`	`6+n x npti`	System acceleration during the simulation `dzita(:,i)` is $\dot{\boldsymbol{\zeta}}(t_i) \in \mathbb{R}^{6+n}$
`tau`	`6+n x npti`	Generalized forces during the simulation `tau(:,i)` is $\boldsymbol{\tau}(t_i) \in \mathbb{R}^{6+n}$
`eta_ee`	`6 x npti`	e.-e. pos./or. during the simulation `eta_ee(:,i)` is $\boldsymbol{\eta}_{ee}(t_i) \in \mathbb{R}^6$
`sigma_x`	`mx x npti`	Generic user-defined objective function `sigma_x(:,i)` is $\boldsymbol{\sigma}_x(t_i) \in \mathbb{R}^{m_x}$
`J_x`	`mx x 6+n`	Corresponding Jacobian $\boldsymbol{J}_x \in \mathbb{R}^{m_x \times 6+n}$

9.4 Direct Dynamics Algorithm Description

Computation of the torques from configuration, velocities and acceleration is known as *inverse dynamics*. Numerical simulations of robot control algorithms require the computation of $\dot{\boldsymbol{\zeta}}$ by knowing the system configuration $\boldsymbol{q}$, $\boldsymbol{R}_B^I$, the system velocities $\boldsymbol{\zeta}$ and the control inputs $\boldsymbol{\tau}$, i.e., what it is known as *direct dynamics*:

$$\begin{aligned}\dot{\boldsymbol{\zeta}} &= \boldsymbol{M}(\boldsymbol{q})^{-1}\left(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} - \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} - \boldsymbol{g}(\boldsymbol{q},\boldsymbol{R}_B^I)\right)\\ &= \boldsymbol{M}(\boldsymbol{q})^{-1}\left(\boldsymbol{\tau} - \boldsymbol{n}(\boldsymbol{q},\boldsymbol{R}_B^I,\boldsymbol{\zeta})\right)\end{aligned}$$

One possibility is to compute simbolically $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{R}_B^I, \boldsymbol{\zeta})$, another possibility is given by the algorithm described in the following that allows the computation of the direct dynamics by resorting to a script that computes the inverse dynamics only (2.77):

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{R}_B^I) = \boldsymbol{\tau}. \tag{2.77}$$

The latter is achieved by resorting to the method 1 developed by Walker and Orin based on the Newton-Euler algorithm [11, 12]. This apparently baroque way to solve the problem exhibits, as great advantage, the possibility to simulate robotic structures without computing symbolically for the inertia matrix. The price to pay is an increased computational load.

The algorithm can be understood easily by observing that, by defining as $\boldsymbol{e}_i \in \mathbb{R}^{6+n}$ (variable `ei`) the versor of all null elements but the i th egual to 1, and putting a null gravity, the following call

```
InverseDynamics(eta2,DH,zeros(6+n,1),ei,PARAM);
```

outputs the i th column of the mass matrix. In addition, the term $\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{R}_B^I, \boldsymbol{\zeta})$ is obtained with the call

```
InverseDynamics(eta2,DH,zita,zeros(6+n,1),PARAM);
```

The algorithm runs according to the following pseudocode

```
tau_n = InverseDynamics(eta2,DH,zita,zeros(6+n,1),PARAM)
$g = 0$ {impose null gravity}
for $i = 1$ to $6 + n$ do
  M(:,i) = InverseDynamics(eta2,DH,zeros(6+n,1),ei,PARAM) {ith column of the mass matrix}
end for
$g = 9.81$ {restore gravity}
dzita = inv(M)*(tau-tau_n) {compute the accelerations}
```

The following scripts is available in SIMURV 4.0 for the Direct Dynamics

```
function dzita = DirectDynamics(eta2,DH,zita,tau,
    PARAM)
%
% Computes the direct dynamics
%
% function dzita = DirectDynamics(eta2,DH,zita,tau,
    PARAM)
%
% input:
%       eta2    dim 3x1     vehicle orientation
%       DH      dim nx4     Denavit-Hartenberg table
    (include joint pos)
%       zita    dim 6+nx1   system velocities
%       tau     dim 6+nx1   generalized forces
%        PARAM   struct      parameters for the dynamic
    simulation
%
% output:
%        dzita   dim 6+nx1   system accelerations
```

9.5 Simulation Parameters Description and Customization

Before running a simulation it is usefull to understand the flags variables in the beginning of the file `core_simulator.m`:

```
KinOnly    = 1;
Graphics   = 1;
```

The important flag is `KinOnly` that, when activated, imposes only a kinematic simulation, in which the controller is supposed to output the velocities $\boldsymbol{\zeta}(t)$ that, by (proper) direct integration, return the system configuration as shown in Fig. 9.2. When `KinOnly=0` the controller outputs $\boldsymbol{\tau}(t)$ and the direct dynamics is also computed, in this case, the block scheme representing the simulation is shown in Fig. 7.1.

The discretization of the nonlinear, continuous-time equations describing the system movement is made by resorting to a simple Euler rule of integration, the corresponding code, that the user should not modify is given in the function `Integration.m`.

In order to implement its own controller the user needs to modify the file `core_simulator.m` and eventually create its own functions. If needed, the model can be modified in the `/data/` folder by adding links or modifying kinematic and dynamic parameters of existing one.

At the end of each simulation the three files are generarted in the `/output/` folder. One file copies the data, the second file copies the core simulator file and the last save all the workspace variables. In this way, by preserving the integrity of all the remaining functions, each simulation can be run again by simply restoring the original file names.

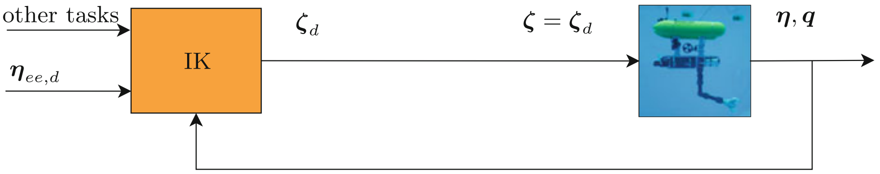


Fig. 9.2 Block scheme for the simulation with flag `KinOnly=1`, the output of the Inverse Kinematics controller is directly integrated

9.6 Run a Demo

9.6.1 *Kinematic Control*

As a first example, a case of kinematic control is simulated concerning the model given in Sect. A.2. It is required to control three task functions in this priority:

(a) tracking of desired end-effector position and orientation
(b) keeping null vehicle roll and pitch
(c) assign a desired −45 deg regulation for the second joint of the manipulator

being (a) the higher priority task. The implemented algorithm is given in Eq. (6.13):

$$\boldsymbol{\zeta}_r = \boldsymbol{J}_a^{\dagger}\left(\dot{\boldsymbol{\sigma}}_{a,d} + \boldsymbol{K}_a\tilde{\boldsymbol{\sigma}}_a\right) + \boldsymbol{N}_a\boldsymbol{J}_b^{\dagger}\left(\dot{\boldsymbol{\sigma}}_{b,d} + \boldsymbol{K}_b\tilde{\boldsymbol{\sigma}}_b\right) + \boldsymbol{N}_{ab}\boldsymbol{J}_c^{\dagger}\left(\dot{\boldsymbol{\sigma}}_{c,d} + \boldsymbol{K}_c\tilde{\boldsymbol{\sigma}}_c\right) \tag{6.13}$$

Under the Matlab© shell, after having selected the installation path (`/simurv4.0/`), run `simurv [enter]`, then, following the information requests by the program, select the model `data_phoenix_smart3s.m` and the controller `core_simulator_demo1.m`. In the lucky case no problem arises the simulation starts, at his end this is how the screen looks like, it contains some information about the model, the output files and some warnings:

```
 ----------------
 ----------------
 -- SIMURV 4.0 --
 ----------------
 ----------------
 Phoenix + Smart3S

 Description of the model
 number of link: 6
 robot dry weight : 285.0 [kg]
 robot buoyancy   : 201.1 [kg]
 robot wet weight : 83.9 [kg]
 link masses      :  80.0  80.0  30.0  50.0  20.0
    25.0  [kg]
 link buoyancies : 106.8  31.4  14.1  25.1  14.1
    9.4  [kg]

model copied in          /output/20130720
   T123319data_phoenix_smart3s.m
output will be copied in /output/20130720T123319out.
   mat
core_simulator in        /output/20130720
   T123319core_simulator_demo1.m
 WARNING in trapezoidal2.m: cruise vel of input 1
    decreased from 0.30 to 0.18

 WARNING in trapezoidal2.m: cruise vel of input 2
    decreased from 0.10 to 0.02
```

```
WARNING in trapezoidal2.m: cruise vel of input 3
    decreased from 0.60 to 0.28

WARNING in trapezoidal2.m: cruise vel of input 1
    decreased from 0.70 to 0.39

WARNING in mypinv.m: task x singular, damped inverse
     used
WARNING in mypinv.m: task x singular, damped inverse
     used

to run another simulation with same model and
    controller type: simurv[enter]
to modify the model type: clear model_name, simurv[
    enter]
to modify the controller type: clear
    core_simulator_name, simurv[enter]
```

Velocities saturation has been implemented by resorting to a basic algorithm implemented in `VectorSat.m`. Each DOF is characterized by a limit (vehicle linear and angular and joint velocities), in case one or more DOF exceeds the given limit all the velocities are scaled by the same amount in order to avoid distortions within the mapping to the task spaces. Refer to citations in Sect. 6.2.4 for more advanced techniques.

Figure 9.3 reports two snapshots of the initial (left) and final (right) configurations. An animation generated with `GenerateMovie.m` may be downloaded from the SIMURV 4.0 link given in the introduction of this Chapter.

9.6.2 Dynamic Control

The second demo concerns a *dummy* controller, i.e., a controller poorly performing due to its intentional simplicity, thus ignoring all the considerations made along the book.

In particular, the vehicle controller is the controller proposed in Sect. 3.6 but *totally ignoring* the presence of the manipulator. This implies that the vehicles experiences a 6-DOF configuration-dependent disturbance that needs to be compensated for each

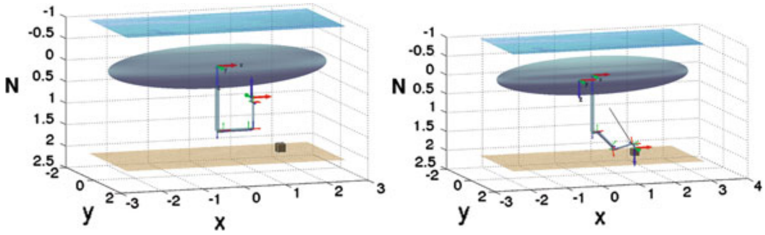


Fig. 9.3 Snapshots of the first demo, initial (*left*) and final (*right*) configurations

desired final configuration with obvious decrease of performance. For the joints a trivial PID is implemented, in this case too, the absence of including any interaction or even the restoring forces in the controller implicitly limit the achieved performances.

The model to be selected still is `data_phoenix_smart3s.m` while the controller is `core_simulator_demo2.m`. It is worth noticing that this simulation is slower than the previous one since it needs to consider the Direct Dynamics, i.e., the algorithm described in Sect. 9.4.

This demo is provided to have a *minimal* working code, the results can not be considered as a valid benchmark.

9.7 Function List

The main functions used in the simulator are listed below, the command `help function_name` gives additional details and the syntax

`DirectDynamics`	Computes the direct dynamics
`DirectKinematics`	Computes the homogeneous transformation matrix from intertial frame to end-effector
`DrawUVMS`	3D rendering of the UVMS in a given configuration
`DrawXYZ`	Functions to draw the various parts of the UVMS
`GenerateDesQuat`	Generate time-varying desired orientation
`GenerateMovie`	Of help in generation of animations
`Homogeneous_dh`	Homogeneous transformation matrix between consecutive frames according to DH convention
`InverseDynamics`	Computes the inverse dynamics
`Jacobian`	Computes the Jacobian in Eq. (2.73)
`J_e`	Computes the Jacobian in Eq. (2.19)
`J_ko_rpy`	Computes the Jacobian in Eq. (2.3) expressed in roll-pitch-yaw
`J_man`	Computes the geometric Jacobian of the sole manipulator expressed with respect to the zero frame
`PlotXYZ`	Functions to plot the various variables of interest
`Quat2Rot`	Rotation matrix expressed in quaternion
`Rot2Quat`	Extract quaternion from rotation matrix
`Rot_dh`	Rotation matrix between consecutive frames according to DH convention
`Rpy2Quat`	Convert roll-pitch-yaw in quaternion
`Rpy2Rot`	Rotation matrix expressed in roll-pitch-yaw
`S`	Skew-symmetric matrix
`VectorSat.m`	*Proportional* saturation of vector

References

1. MATLAB, version 7.10.0 (R2010a). (The MathWorks Inc., Natick, 2010)
2. Octave community. GNU/Octave (2012)
3. J.W. Eaton, D. Bateman, S. Hauberg, *GNU Octave Manual Version 3* (Network Theory Limited, London, 2008)
4. S. Enterprises, *Scilab: free and open source software for numerical computation* (Scilab Enterprises, Orsay, 2012)
5. M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng. ROS: an open-source robot operating system. In *Open-source software workshop of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, 2009
6. T. Perez, T. Fossen, *Marine Systems Simulator (2010)* (Norwegian University of Science and Technology, Trondheim, 2013)
7. P. Senarathne, W. Wijesoma, K. Lee, B. Kalyan, M. Moratuwage, N. Patrikalakis, F. Hover. MarineSIM: robot simulation for marine environments. In *IEEE OCEANS 2010*. (IEEE, 2010)
8. C.W. Chen, J.S. Kouh, J.F. Tsai, Modeling and simulation of an AUV simulator with guidance. System **38**(2), 211–225 (2012)
9. M. Prats, J. Pérez, J.J. Fernández, P. Sanz. An open source tool for simulation and supervision of underwater intervention missions. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference* on, pp. 2577–2582. (IEEE, 2012)
10. P.J. Sanz, P. Ridao, G. Oliver, C. Insaurralde, G. Casalino, C. Silvestre, C. Melchiorri, A. Turetta. TRIDENT: Recent improvements about intervention missions. In *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV2012)* (2012)
11. M.W. Walker, D.E. Orin, Efficient dynamic computer simulation of robotic mechanisms. J. Dyn. Syst. Measur. Control **104**(3), 205–211 (1982)
12. R. Featherstone, D. Orin. Robot dynamics: equations and algorithms. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference* on, vol. 1, pp. 826–834. (IEEE, 2000)

Chapter 10
Concluding Remarks

> *"Cheshire-cat—Alice began—would you tell me, please, which way I ought to go from here?". "That depends a good deal on where you want to get to", said the cheshire-cat.*
> Lewis Carroll, *"Alice's adventures in wonderland"*.

This monograph addressed control of underwater vehicle manipulator systems, a challenging problem that is addressed by the field actors with an increasing interest. Autonomous robotics, in all its domains, is continuously growing reaching new and exciting results, in the underwater environment as well.

From a mathematical perspective, the equations of motion describing the movement of UVMSs, although nonlinear and coupled, are not *so* different from that of any industrial manipulator. They share most of the mathematical properties useful in the analysis and design of feedback control loop. The difference is mainly due to the presence of the water, i.e., a fluid denser than the water. At the operating velocities, however, a *simple* buoyancy and damping terms are required to describe this phenomenon.

This may lead to the apparent conclusion that a simple *translation* of the industrial control schemes is sufficient to handle the underwater case. This may be true in simplified numerical simulations but, as widely discussed in this monograph, a so *naive* approach would probably fail.

The underwater environment is hostile, the communication is difficult and external disturbances such as the ocean current can strongly affect the dynamics of the system involved in the mission. Moreover, the sensing devices are not fully reliable in the underwater environment. This is particularly true for the positioning sensors. The vehicle actuating systems, usually composed of thrusters, is highly nonlinear and subject to limit cycles. Some very simple operations for a ground robotic system, such as *to hold its position* can become difficult for UVMSs.

In this framework, the control of the UVMS must be *robust*, in a wide sense, reliable and of simple implementation. With this in mind the techniques for motion and force control of UVMSs needs to be developed.

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4_10,

The kinematic control approach is important for accomplishing an autonomous robotic mission. Differently from industrial robotics, where an off-line trajectory planning is a possible approach, in case of unknown, unstructured environment, the trajectory has to be generated in real-time. Moreover, UVMSs are usually redundant with respect to the given task; a kinematic control approach, thus, can exploit such a redundancy.

The motion control of UVMSs must deal with the strong constraints discussed along the book. In this work, different approaches were discussed; as an example, it has been stressed the results achieved in Chap. 3: certain control laws devoted at the study of 6-DOFs dynamic control of the vehicle can deteriorate the transient due to a wrong adaptive or integral action. In the Author's opinion, this result is particularly interesting.

In the recent years some interesting results have been achieved in the laboratories worldwide. Few set-up are available and sea tests are now possible in autonomous or semi-autonomous mode. Also, the community is starting discussing the problem of cooperation between two UVMSs.

Nec plus ultra.

Appendix A
Mathematical Models

This Appendix reports the parameters of the different mathematical models used along the book. For all the models the gravity vector, expressed in the inertial frame, and the water density are given by:

$$\begin{aligned} \boldsymbol{g}^I &= \begin{bmatrix} 0 \; 0 \; 9.81 \end{bmatrix}^{\mathrm{T}} \ \mathrm{m/s^2} \\ \rho &= 1000\,\mathrm{kg/m^3} \,. \end{aligned}$$

A.1 Phoenix

The data of the vehicle Phoenix, developed at the Naval Postgraduate School (Monterey, CA, USA), are given in [1]. These have been used as data of the vehicle carrying a manipulator with 2, 3 and 6 DOFs in the simulations of the dynamic control laws.

L	$= 5.3$	m	vehicle length
m	$= 5454.54$	kg	vehicle weight
$\boldsymbol{r}_G^B$	$= \begin{bmatrix} 0 \; 0 \; 0.061 \end{bmatrix}^{\mathrm{T}}$	m	
$\boldsymbol{r}_B^B$	$= \begin{bmatrix} 0 \; 0 \; 0 \end{bmatrix}^{\mathrm{T}}$	m	
W	$= 53400$	N	
B	$= 53400$	N	
I_x	$= 2038$	$\mathrm{Nms^2}$	
I_y	$= 13587$	$\mathrm{Nms^2}$	
I_z	$= 13587$	$\mathrm{Nms^2}$	
I_{xy}	$= -13.58$	$\mathrm{Nms^2}$	
I_{yz}	$= -13.58$	$\mathrm{Nms^2}$	
I_{xz}	$= -13.58$	$\mathrm{Nms^2}$	

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4,
© Springer International Publishing Switzerland 2014

where:

$$\boldsymbol{I}_{O_b} = \begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{xy} & I_y & I_{yz} \\ I_{xz} & I_{yz} & I_z \end{bmatrix}.$$

The inertia matrix, including the added mass terms, is given by:

$$\boldsymbol{M}_v = \left[\begin{array}{ccc} 6.019 \times 10^3 & 5.122 \times 10^{-8} & -1.180 \times 10^{-2} \\ 5.122 \times 10^{-8} & 9.551 \times 10^3 & 3.717 \times 10^{-6} \\ -1.180 \times 10^{-2} & 3.717 \times 10^{-6} & 2.332 \times 10^4 \\ -3.200 \times 10^{-5} & -3.802 \times 10^2 & -1.514 \times 10^{-4} \\ 3.325 \times 10^2 & -3.067 \times 10^{-5} & 2.683 \times 10^3 \\ -6.731 \times 10^{-5} & -4.736 \times 10^2 & -4.750 \times 10^{-4} \end{array}\right. \cdots$$

$$\cdots \left.\begin{array}{ccc} -3.200 \times 10^{-5} & 3.325 \times 10^2 & -6.731 \times 10^{-5} \\ -3.802 \times 10^2 & -3.067 \times 10^{-5} & -4.736 \times 10^2 \\ -1.514 \times 10^{-4} & 2.683 \times 10^3 & -4.750 \times 10^{-4} \\ 4.129 \times 10^3 & 1.358 \times 10^1 & 8.467 \times 10^1 \\ 1.358 \times 10^1 & 4.913 \times 10^4 & 1.357 \times 10^1 \\ 8.467 \times 10^1 & 1.357 \times 10^1 & 2.069 \times 10^4 \end{array}\right].$$

The hydrodynamic derivatives are given by:

$X_{p\lvert p\rvert} = 2.8 \times 10^3$	$X_{q\lvert q\rvert} = -5.9 \times 10^3$	$X_{r\lvert r\rvert} = 1.6 \times 10^3$
$X_{pr} = 2.9 \times 10^2$	$X_{\dot{u}} = -5.6 \times 10^2$	$X_{\omega q} = -1.5 \times 10^4$
$X_{vp} = -2.2 \times 10^2$	$X_{vr} = 1.5 \times 10^3$	$X_{v\lvert v\rvert} = 7.4 \times 10^2$
$X_{\omega\lvert\omega\rvert} = 2.4 \times 10^3$		
$Y_{\dot{p}} = 47$	$Y_{\dot{r}} = 4.7 \times 10^2$	$Y_{pq} = 1.6 \times 10^3$
$Y_{qr} = -2.6 \times 10^3$	$Y_{\dot{v}} = -4.1 \times 10^3$	$Y_{up} = 2.2 \times 10^2$
$Y_r = 2.2 \times 10^3$	$Y_{vq} = 1.8 \times 10^3$	$Y_{\omega p} = 1.7 \times 10^4$
$Y_{\omega r} = -1.4 \times 10^3$	$Y_v = -1.4 \times 10^3$	$Y_{v\omega} = 9.5 \times 10^2$
$Z_{\dot{q}} = -2.7 \times 10^3$	$Z_{p\lvert p\rvert} = 51$	$Z_{pr} = 2.6 \times 10^3$
$Z_{r\lvert r\rvert} = -2.9 \times 10^3$	$Z_{\dot{\omega}} = -1.8 \times 10^4$	$Z_{uq} = -1 \times 10^4$
$Z_{vp} = -3.6 \times 10^3$	$Z_{vr} = 3.3 \times 10^3$	$Z_{u\omega} = -4.2 \times 10^3$
$Z_{v\lvert v\rvert} = -9.5 \times 10^2$		
$K_{\dot{p}} = -2 \times 10^3$	$K_{\dot{r}} = -71$	$K_{pq} = -1.4 \times 10^2$
$K_{qr} = 3.5 \times 10^4$	$K_{\dot{v}} = 47$	$K_{up} = -4.3 \times 10^3$
$K_{ur} = -3.3 \times 10^2$	$K_{vq} = -2 \times 10^3$	$K_{\omega p} = -51$
$K_{\omega r} = 5.5 \times 10^3$	$K_{uv} = 2.3 \times 10^2$	$K_{v\omega} = -1.4 \times 10^4$
$M_{\dot{q}} = -3.5 \times 10^4$	$M_{p\lvert p\rvert} = 1.1 \times 10^2$	$M_{pr} = 1 \times 10^4$
$M_{r\lvert r\rvert} = 6 \times 10^3$	$M_{\dot{\omega}} = -2.7 \times 10^3$	$M_{uq} = -2.7 \times 10^4$
$M_{vp} = 4.7 \times 10^2$	$M_{vr} = 6.7 \times 10^3$	$M_{u\omega} = 7.4 \times 10^3$
$M_{v\lvert v\rvert} = -1.9 \times 10^3$		
$N_{\dot{p}} = -71$	$N_{\dot{r}} = -7.1 \times 10^3$	$N_{pq} = -4.4 \times 10^4$
$N_{qr} = 5.6 \times 10^3$	$N_{\dot{v}} = 4.7 \times 10^2$	$N_{up} = -3.3 \times 10^2$
$N_{ur} = -6.3 \times 10^3$	$N_{vq} = -3.9 \times 10^3$	$N_{\omega p} = -6.7 \times 10^3$
$N_{\omega r} = 2.9 \times 10^3$	$N_{uv} = -5.5 \times 10^2$	$N_{v\omega} = -2 \times 10^3$

Table A.1 Mass [kg], Denavit-Hartenberg parameters [m, rad], radius [m], length [m] and viscous friction [Nms] of the manipulator mounted on the underwater vehicle

	Mass	a	d	θ	α	Radius	Length	Viscous frict.
Link 1	80	0.15	0	q_1	$-\pi/2$	0.2	0.85	30
Link 2	80	0.61	0	q_2	0	0.1	1	20
Link 3	30	0.11	0	q_3	$-\pi/2$	0.15	0.2	5
Link 4	50	0	0.610	q_4	$\pi/2$	0.1	0.8	10
Link 5	20	0	−0.113	q_5	$-\pi/2$	0.15	0.2	5
Link 6	25	0	0.103	q_6	0	0.1	0.3	6

On the sway, heave, pitch and yaw degree of motion the damping due to the vortex shedding is also considered. By defining as $b(x)$ the vehicle breadth and $h(x)$ the vehicle height this term is modeled as a discretized version of the following:

$$
\begin{aligned}
Y &= -\frac{1}{2}\rho \int_{tail}^{nose} \left[C_{dy} h(x)(v+xr)^2 + C_{dz} b(x)(\omega - xq)^2 \right] \frac{(v+xr)}{U_{cf}(x)} dx \\
Z &= \frac{1}{2}\rho \int_{tail}^{nose} \left[C_{dy} h(x)(v+xr)^2 + C_{dz} b(x)(\omega - xq)^2 \right] \frac{(\omega - xq)}{U_{cf}(x)} dx \\
M &= -\frac{1}{2}\rho \int_{tail}^{nose} \left[C_{dy} h(x)(v+xr)^2 + C_{dz} b(x)(\omega - xq)^2 \right] \frac{(\omega + xq)}{U_{cf}(x)} x dx \\
N &= -\frac{1}{2}\rho \int_{tail}^{nose} \left[C_{dy} h(x)(v+xr)^2 + C_{dz} b(x)(\omega - xq)^2 \right] \frac{(v+xr)}{U_{cf}(x)} x dx
\end{aligned}
$$

where the cross-flow velocity is computed as:

$$
U_{cf}(x) = \sqrt{(v+xr)^2 + (\omega - xq)^2}.
$$

In the simulations $h(x)$ and $b(x)$ have been considered constant, in detail $h(x) = 0.5\,\text{m}$ and $b(x) = 1\,\text{m}$. Finally, $C_{dy} = C_{dz} = 0.6$.

A.2 Phoenix + 6DOF SMART 3S

The vehicle, whose model is widely known and used in literature [1, 2], is supposed to carry a SMART-3S manipulator manufactured by COMAU whose main parameters are reported in Table A.1.

The manipulator is supposed to be mounted under the vehicle in the middle of its length, the vector positions of the origin of the frames $i-1$ to frame/center-of-mass/center-of-buoyancy of link i are not reported for brevity. The dry friction has not been considered to avoid chattering behavior and increase output readability. All the links are modeled as cylinders. Their volumes, thus, are computed as $\delta_i = \pi * L_i\, r_i^2$, where L_i and r_i are the link lengths and radius, respectively. The modeling of the

Table A.2 Link inertia [Nms^2] of the manipulator mounted on the underwater vehicle

	$I_{x,i}$	$I_{y,i}$	$I_{z,i}$	$I_{xy,i}$	$I_{xz,i}$	$I_{yz,i}$
Link 1	100	30	100	0	0	0
Link 2	20	80	80	0	0	0
Link 3	2	0.5	2	0	0	0
Link 4	50	9	50	0	0	0
Link 5	5	4	5	0	0	0
Link 6	5	5	3	0	0	0

hydrodynamic effects, using the strip theory, benefit from the simplified geometric assumption on the link shapes [2, 3].

For each of the cylinder the mass is computed as:

$$\boldsymbol{M}_i = \left[\begin{matrix} m_i + \rho\delta_i & 0 & 0 \\ 0 & m_i + \rho\delta_i & 0 \\ 0 & 0 & m_i + 0.1 m_i \\ * & * & * \\ * & * & * \\ * & * & * \end{matrix}\right. \cdots$$

$$\left.\begin{matrix} * & * & * \\ * & * & * \\ * & * & * \\ I_{x,i} + \pi\rho L_i^3 r_i^2/12 & 0 & 0 \\ 0 & I_{y,i} + \pi\rho L_i^3 r_i^2/12 & 0 \\ 0 & 0 & I_{z,i} \end{matrix}\right],$$

where the link inertia are given in Table A.2. In this case, the cylinder length is considered along the z_i axis.

The linear skin and quadratic drag coefficients are given by $D_s = 0.4$ and $C_d = 0.6$, respectively. The lift coefficient C_l is considered null.

A.3 ODIN

The mathematical model of ODIN, an AUV developed at the ASL, University of Hawaii, has been used to test several, experimentally validated, control laws [4–11]:

$$
\begin{array}{lll l}
r & = 0.3 & \mathrm{m} & \text{vehicle radius} \\
m & = 125 & \mathrm{kg} & \text{vehicle weight} \\
\boldsymbol{r}_G^B & = \left[0\ 0\ 0.05\right]^{\mathrm{T}} & \mathrm{m} & \\
\boldsymbol{r}_B^B & = \left[0\ 0\ 0\right]^{\mathrm{T}} & \mathrm{m} & \\
W & = 1226 & \mathrm{N} & \\
I_x & = 8 & \mathrm{Nms}^2 & \\
I_y & = 8 & \mathrm{Nms}^2 & \\
I_z & = 8 & \mathrm{Nms}^2 & \\
I_{xy} & = 0 & \mathrm{Nms}^2 & \\
I_{yz} & = 0 & \mathrm{Nms}^2 & \\
I_{xz} & = 0 & \mathrm{Nms}^2 &
\end{array}
$$

The buoyancy B is computed considering the vehicle as spherical. The hydrodynamic derivatives are given by: It can be observed that a diagonal structures of the matrices $\boldsymbol{M}_A$ and $\boldsymbol{D}$ is obtained.

$$
\begin{array}{llll}
X_{\dot{u}} & = -62.5 & X_{u|u|} & = -48 \\
Y_{\dot{v}} & = -62.5 & Y_{v|v|} & = -48 \\
Z_{\dot{\omega}} & = -62.5 & Z_{\omega|\omega|} & = -48 \\
K_p & = -30 & K_{p|p|} & = -80 \\
M_q & = -30 & M_{q|q|} & = -80 \\
N_r & = -30 & N_{r|r|} & = -80
\end{array}
$$

According to simple geometrical considerations, the following TCM is observed:

$$
\boldsymbol{B}_v = \begin{bmatrix}
s & -s & -s & s & 0 & 0 & 0 & 0 \\
s & -s & -s & -s & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & l_1 s & l_1 s & -l_1 s & -l_1 s \\
0 & 0 & 0 & 0 & l_1 s & -l_1 s & -l_1 s & l_1 s \\
l_2 & -l_2 & l_2 & -l_2 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

with $s = \sin(\pi/4)$, $l_1 = 0.381\,\mathrm{m}$ and $l_2 = 0.508\,\mathrm{m}$.

A.4 Ellispoidal Shape

The vehicle used in the simulation is an ellipsoid characterized by the parametric representation:

$$
\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1
$$

where a, b and c are the semi-axes of dimensions: $a = 0.6\,\mathrm{m}$, $b = 0.3\,\mathrm{m}$, $c = 0.3\,\mathrm{m}$. Being $b = c$ and $a > c$, it can be noticed that a prolate spheroid has been selected whose eccentricity $e_c = 1 - (b/a)^2 = 0.75$. The dry weight is about $m \approx 225\,\text{kg}$. The center of gravity and buoyancy are given by

$$\begin{aligned} \boldsymbol{r}_G &= \begin{bmatrix} 0 \; 0 \; 0 \end{bmatrix} \text{ m} \\ \boldsymbol{r}_B &= \begin{bmatrix} 0 \; 0 \; 0.05 \end{bmatrix} \text{ m}. \end{aligned}$$

It is further possible to compute

$$\begin{aligned} W &= mg \approx 2196\,\text{N} \\ B &= \frac{4}{3}\pi \, abc \, \rho g \approx 2219\,\text{N}, \end{aligned}$$

leading to $W - B = -22.1897\,\text{N}$ and $z_G W = 109.8390\,\text{Nm}$. The inertia tensor is given by

$$\boldsymbol{I}_0 = \frac{m}{5} \begin{bmatrix} (b^2+c^2) & 0 & 0 \\ 0 & (a^2+c^2) & 0 \\ 0 & 0 & (a^2+b^2) \end{bmatrix} \approx \begin{bmatrix} 8 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}.$$

The matrix $\boldsymbol{M}_{RB}$ is given by

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I} & -m\boldsymbol{S}(\boldsymbol{r}_G) \\ m\boldsymbol{S}(\boldsymbol{r}_G) & \boldsymbol{I}_0 \end{bmatrix},$$

where the matrix $\boldsymbol{S}(\cdot)$ has been defined in Eq. (2.6). By defining

$$\begin{aligned} \alpha_0 &= \frac{2(1-e_c^2)(\frac{1}{2}\log\left(\frac{1+e_c}{1-e_c}\right) - e_c)}{e_c^3} = 0.4624 \\ \beta_0 &= \frac{1}{e_c^2} - \frac{(1-e_c^2)\log\left(\frac{1+e_c}{1-e_c}\right)}{2e_c^3} = 0.7688, \end{aligned}$$

it is possible to compute the added mass coefficients as

$$\begin{aligned} X_{\dot{u}} &= -m\frac{\alpha_0}{2-\alpha_0} \approx -67 \\ Y_{\dot{v}} &= -m\frac{\beta_0}{2-\beta_0} \approx -140 \\ Z_{\dot{w}} &= Y_{\dot{v}} \approx -140 \\ K_{\dot{p}} &= 0 \\ M_{\dot{q}} &= -\frac{m}{5}\frac{(b^2-a^2)^2(\alpha_0-\beta_0)}{2(b^2-a^2)-(b^2+a^2)(\alpha_0-\beta_0)} \approx -2.5 \\ N_{\dot{r}} &= M_{\dot{q}} \approx -2.5 \end{aligned}$$

and, being the added mass defined as

$$\boldsymbol{M}_A = \operatorname{diag}\left(\begin{bmatrix} -X_{\dot{u}} & -Y_{\dot{v}} & -Z_{\dot{w}} & -K_{\dot{p}} & -M_{\dot{q}} & -N_{\dot{r}} \end{bmatrix}\right)$$

the simulated mass matrix is

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A.$$

The rigid body Coriolis and Centripetal matrix is given by

$$\boldsymbol{C}_{RB} = \begin{bmatrix} \boldsymbol{O} & -m\boldsymbol{S}(\boldsymbol{\nu}_1) - m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{r}_G) \\ -m\boldsymbol{S}(\boldsymbol{\nu}_1) - m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{r}_G) & m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{r}_G) - \boldsymbol{S}(\boldsymbol{I}_0\boldsymbol{\nu}_2) \end{bmatrix}$$

and the added mass

$$\boldsymbol{C}_A = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix}.$$

leading to $\boldsymbol{C} = \boldsymbol{C}_{RB} + \boldsymbol{C}_A$. Finally, the damping matrix is defined as

$$\begin{aligned} \boldsymbol{D} = \operatorname{diag}\Big(\Big[& -X_{u|u|}\,|u| \;\; -Y_{v|v|}\,|v| \;\; -Z_{w|w|}\,|w| \;\; \ldots \\ & -K_p - K_{p|p|}\,|p| \;\; -M_q - M_{q|q|}\,|q| \;\; -N_r - N_{r|r|}\,|r| \Big]\Big) \end{aligned}$$

where

$$\begin{aligned} X_{u|u|} &= -50 \\ Y_{v|v|} &= X_{u|u|}\frac{a}{b} = -100 \\ Z_{w|w|} &= X_{u|u|}\frac{a}{b} = -100 \\ K_{p|p|} &= -10 \\ M_{q|q|} &= -80 \\ N_{r|r|} &= -80 \\ K_p &= -5 \\ M_q &= -20 \\ N_r &= -20\,. \end{aligned}$$

A.5 9-DOF UVMS

Details of the mathematical model of the vehicle carrying a 3-link manipulator used in the simulations of the interaction control chapter can be found in [12], its main characteristics are reported in the following. The vehicle considered is a box of $2 \times 1 \times 0.5\,\mathrm{m}$ characterized by:

$$\begin{array}{llll}
L & = 2 & \mathrm{m} & \text{vehicle length} \\
m & = 1050 & \mathrm{kg} & \text{vehicle weight} \\
\boldsymbol{r}_G^B & = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} & \mathrm{m} & \\
\boldsymbol{r}_B^B & = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} & \mathrm{m} & \\
W & = 10300 & \mathrm{N} & \\
B & = 7900 & \mathrm{N} & \\
I_x & = 66 & \mathrm{Nms}^2 & \\
I_y & = 223 & \mathrm{Nms}^2 & \\
I_z & = 223 & \mathrm{Nms}^2 & \\
I_{xy} & = 0 & \mathrm{Nms}^2 & \\
I_{yz} & = 0 & \mathrm{Nms}^2 & \\
I_{xz} & = 0 & \mathrm{Nms}^2 &
\end{array}$$

The hydrodynamic derivatives are given by:

$$\begin{array}{ll}
X_{\dot{u}} & = -307 \\
Y_{\dot{v}} & = -1025 \\
Z_{\dot{\omega}} & = -1537 \\
K_{\dot{p}} & = -12 \\
M_{\dot{q}} & = -36 \\
N_{\dot{r}} & = -36
\end{array}$$

The drag coefficient is given by $C_d = 1.8$. All the hydrodynamic effects have been computed by assuming a regular shape for the vehicle and the link (a box and a cylinder, respectively).

In Tables A.3 and A.4 the mass, length, radius and inertia of the 3-link manipulator carried by the vehicle are reported.

Table A.3 Mass [kg], length [m] and radius [m] of the 9-DOF UVMS

	Mass	Length	Radius
Link 1	33	1	0.1
Link 2	19	1	0.08
Link 3	12	1	0.07

Table A.4 Link inertia [Nms^2] of the 3-link manipulator mounted on the underwater vehicle

	$I_{x,i}$	$I_{y,i}$	$I_{z,i}$	$I_{xy,i}$	$I_{xz,i}$	$I_{yz,i}$
Link 1	11	11	1.65	0	0	0
Link 2	6.3	6.3	0.75	0	0	0
Link 3	4	4	0.4	0	0	0

References

1. A. Healey, D. Lienard, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J. Oceanic Eng. **18**(3), 327–339 (1993)
2. T. Fossen, *Guidance and Control of Ocean Vehicles* (Wiley, Chichester, 1994)
3. I. Schjølberg, Modeling and Control of Underwater Robotic Systems. in *Ph.D. thesis*, Doktor ingeniør degree, Norwegian University of Science and Technology, Trondheim, 1996
4. G. Antonelli, S. Chiaverini, N. Sarkar, M. West, Adaptive control of an autonomous underwater vehicle: experimental results on ODIN, in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 64–69, Monterey, 1999
5. G. Antonelli, S. Chiaverini, N. Sarkar, M. West, Adaptive control of an autonomous underwater vehicle: experimental results on ODIN. IEEE Trans. Control Syst. Technol. **9**(5), 756–765 (2001)
6. S.K. Choi, J. Yuh, Experimental study on a learning control system with bound estimation for underwater robots. Auton. Robot **3**(2), 187–194 (1996)
7. J. Nie, J. Yuh, E. Kardash, T. Fossen, On-board sensor-based adaptive control of small UUVS in very shallow water. Int. J. Adapt. Control Sig. Process. **14**(4), 441–452 (2000)
8. T.K. Podder, G. Antonelli, N. Sarkar, Fault tolerant control of an autonomous underwater vehicle under thruster redundancy: simulations and experiments, *Proceedings 2000 IEEE International Conference on Robotics and Automation* (San Francisco, 2000), pp. 1251–1256
9. T.K. Podder, G. Antonelli, N. Sarkar, An experimental investigation into the fault-tolerant control of an autonomous underwater vehicle. J. Adv. Robot. **15**(5), 501–520 (2001)
10. N. Sarkar, T.K. Podder, G. Antonelli, Fault accommodating thruster force allocation of an AUV considering thruster redundancy and saturation. IEEE Trans. Robot. Autom. **18**(2), 223–233 (April 2002)
11. J. Yuh, J. Nie, C.S.G. Lee, Experimental study on adaptive control of underwater robots, in *Proceedings of 1999 IEEE International Conference on Robotics and Automation*, pp. 393–398, 1999
12. T.K. Podder, Dynamic and control of kinematically redundant underwater vehicle-manipulator systems, *Technical Report ASL 98–01*, Autonomous Systems Laboratory Technical Report, University of Hawaii, Honolulu, 1998

About the Author

Gianluca Antonelli was born in Rome, Italy, in 1970. He received the “Laurea” degree in Electronic Engineering and the “Research Doctorate” degree in Electronic Engineering and Computer Science at the University of Naples in 1995 and 2000, respectively. He currently is an Associate Professor at the University of Cassino and Southern Lazio. He is Chair of the IEEE RAS Italian Chapter since 2011. He is Senior Member IEEE since June 2006. He is Associate Editor of the IEEE Transactions on Control Systems Technology since 2013. From 2005 to 2009 has been Associate Editor of the IEEE Transactions on Robotics. From 2007 to 2009 has been an Editor of the Springer Journal of Intelligent Service Robotics and then, from 2009 to 2011, Regional Editor for Europe/Africa. From 2008 to 2011 he has been Chair of the IEEE Robotics and Automation Society Technical Committee in Marine Robotics. He served as Scientific Responsible, Researcher or Member of the Scientific Advisory Board for several projects under the European Community Framework Programme. For the same funding scheme he served both as independent expert and reviewer several times since 2006. His research interests include simulation and control of underwater robotic systems, force/motion control of robot manipulators, multi-robot systems, identification. He has published more than 30 international journal papers and more than 90 conference papers, he Co-authored the Chapter “Underwater Robotics” for the Springer Handbook of Robotics, (Springer-Verlag, 2008).

G. Antonelli, *Underwater Robots*, Springer Tracts
in Advanced Robotics 96, DOI: 10.1007/978-3-319-02877-4,