SPRINGER BRIEFS IN STATISTICS
JSS RESEARCH SERIES IN STATISTICS

Kohei Ohtsu
Hui Peng
Genshiro Kitagawa

Time Series Modeling for Analysis and Control

Advanced Autopilot and Monitoring Systems

Springer

SpringerBriefs in Statistics

JSS Research Series in Statistics

The current research of statistics in Japan has expanded in several directions in line with recent trends in academic activities in the area of statistics and statistical sciences over the globe. The core of these research activities in statistics in Japan has been the Japan Statistical Society (JSS). This society, the oldest and largest academic organization for statistics in Japan, was founded in 1931 by a handful of pioneer statisticians and economists and now has a history of about 80 years. Many distinguished scholars have been members, including the influential statistician Hirotugu Akaike, who was a past president of JSS, and the notable mathematician Kiyosi Itô, who was an earlier member of the Institute of Statistical Mathematics (ISM), which has been a closely related organization since the establishment of ISM. The society has two academic journals: the Journal of the Japan Statistical Society (English Series) and the Journal of the Japan Statistical Society (Japanese Series). The membership of JSS consists of researchers, teachers, and professional statisticians in many different fields including mathematics, statistics, engineering, medical sciences, government statistics, economics, business, psychology, education, and many other natural, biological, and social sciences.

The JSS Series of Statistics aims to publish recent results of current research activities in the areas of statistics and statistical sciences in Japan that otherwise would not be available in English; they are complementary to the two JSS academic journals, both English and Japanese. Because the scope of a research paper in academic journals inevitably has become narrowly focused and condensed in recent years, this series is intended to fill the gap between academic research activities and the form of a single academic paper.

The series will be of great interest to a wide audience of researchers, teachers, professional statisticians, and graduate students in many countries who are interested in statistics and statistical sciences, in statistical theory, and in various areas of statistical applications.

More information about this series at http://www.springer.com/series/13497

Kohei Ohtsu · Hui Peng
Genshiro Kitagawa

Time Series Modeling for Analysis and Control

Advanced Autopilot and Monitoring Systems

Springer

Kohei Ohtsu
Ohtsu Maritime Institute, Co. Ltd.
Tokyo
Japan

Hui Peng
School of Information Science
and Engineering
Central South University
Zuojialong, Changsha, Hunan
China

Genshiro Kitagawa
Research Organization of Information
and Systems
Minato-ku, Tokyo
Japan

ISSN 2191-544X ISSN 2191-5458 (electronic)
SpringerBriefs in Statistics
ISSN 2364-0057 ISSN 2364-0065 (electronic)
JSS Research Series in Statistics
ISBN 978-4-431-55302-1 ISBN 978-4-431-55303-8 (eBook)
DOI 10.1007/978-4-431-55303-8

Library of Congress Control Number: 2015932242

Springer Tokyo Heidelberg New York Dordrecht London

Printed on acid-free paper

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

Preface

The autopilot and main engine governor of a ship are typical examples of feedback systems that have a long history. Autopilot systems to maintain the heading angle of a ship in order to hold a desired course were developed by the Sperry Corporation in the 1910s, and since then helmsmen have become free from the arduous task of steering in course navigation. A governor mechanism to maintain the revolution rate of the engine shaft was invented much earlier than the autopilot system and can be traced back to the centrifugal governor invented by James Watt for regulating a steam engine in the eighteenth century. The classical control theory for designing these analog control systems has contributed to control in numerous mechanical systems.

In the latter half of the twentieth century, however, the circumstances of control engineering have changed rapidly due to dramatic developments in digital computers and microelectronics, and digital computers have overtaken analog systems in several fields. In the first stage of digital control, the analog control law was digitized to realize a digital control system. However, a more essential innovation in control system design was to apply the modern control theory based on the state-space model of the control system.

In the 1970s, modern control theory was also introduced, allowing innovations in ship autopilot systems. The critical problem in designing an autopilot system, however, is to obtain a model of the ship that can properly represent the complicated and inherently stochastic behavior of a ship at sea. Without a reasonable model of the control system, it is not possible to apply modern control theory, which is a bottleneck when applying modern control theory to complicated large systems with strong disturbance noise.

As a practical solution to this problem, Dr. Hirotugu Akaike proposed the use of the autoregressive (AR) model in the analysis and control of complicated systems. The crucial problem in statistical modeling was the identification of the model, including the selection of variables, model type, and model orders, and the estimation of unknown parameters. For this problem, he proposed final prediction errors (FPE) for identifying the stochastic behaviors of a cement rotary kiln system using a multivariate autoregressive (MAR) model, and generalized FPE to the

Akaike information criterion (AIC) for evaluating a more general statistical model. Many successful applications of analysis and control of complicated stochastic systems through statistical modeling based on AIC have appeared in the literature.

The purpose of this book is to present an appropriate time series modeling method for the analysis and control of complicated systems, for which it is difficult to obtain a precise model that can express the behavior of a controlled system based on the theory of the domain. Throughout this book, we will use multivariate autoregressive modeling with exogenous variables based on AIC. However, we will also consider a nonstationary version and a nonlinear version of the model to cope with real problems. A special feature of this book is to consider modeling, analysis, and control of a real ship's behavior at sea, and we herein develop various types of autopilot systems. We present not only the results of simulation studies, but also many results of actual sea tests. Although we treat only applications related to ships, we hope that the readers of this book will gain a deeper general knowledge and useful tools for the analysis and control of complicated systems and will be able to apply these methods to solve problems in their own fields.

This book is the result of long and intensive collaboration of three researchers who have different research fields. Kohei Ohtsu's research interests include the analysis, monitoring, and control of ship motions at sea using time series modeling techniques. He developed a novel autopilot system using an autoregressive model in cooperation with Genshiro Kitagawa in the 1970s. Hui Peng's research interests include nonlinear system modeling, nonlinear optimization, and optimal control. He developed a practical modeling technique for nonlinear time series using a radial bases function ARX model and, together with the two other authors of this book, recently succeeded in developing tracking control of a ship using this model. Genshiro Kitagawa's primary interests are in statistical modeling, nonstationary time series analysis, and optimal control of stochastic systems. He developed a Monte Carlo filter technique for a nonlinear state-space model which is now referred to as a "particle filter".

The authors would like to thank the numerous people who have supported our research in its various stages. In particular, we would like to express our sincere thanks to the late Dr. Hirotugu Akaike, former Director General of the Institute of Statistical Mathematics, Japan, for his guidance and valuable suggestions regarding our research. We are also grateful to Prof. Michio Horigome, Dr. Hiroyuki Oda, Dr. Jun Wu, the crew members of Shioji-Maru, and numerous other people for their collaboration and contributions to our research. Finally, we would like to thank Ms. Michiko Oda for her help in editing this book.

Tokyo, Japan, January 2015 — Kohei Ohtsu
Changsha, China — Hui Peng
Tokyo, Japan — Genshiro Kitagawa

Contents

Chapter 1
Introduction

Abstract In the following, the necessity of statistical modeling for analysis and control of complex, large systems with large disturbances and the aim of this book are first presented. We then present the basic concepts of ship motion and course keeping control problems, which are the primary applications of the time series modeling treated in this book. A brief explanation of the real ships that were used in actual sea tests is then presented. Finally, the organization of this book is described.

Keywords Ship motion · Statistical modeling · Autopilot · Ship propelling · Outline of chapters

1.1 Necessity of Statistical Modeling for Complex, Large Systems

In the identification of ship motion on the ocean, it is important to adopt a statistical model because external disturbances caused by wind, waves, and the motion of the hull itself in response to such oceanic disturbances are intrinsically irregular. Moreover, the dynamic range of the external disturbances is very wide, from mirror-like calm seas to rough seas with violent storms. Thus, changes of ship motions are so large that they would not be imaginable in other vehicles. A method of practical analysis of such irregular phenomena has been established in the frequency domain (Blackman and Tukey 1959; Isobe 1960), and the ship motion under disturbances has also been dealt with as a stochastic process in the field of ship-building engineering.

Statistical methods for analyzing time series obtained from model tests conducted in irregular waves or using records of actual-sea tests in the frequency domain have been established by 1960s (Yamanouchi 1961). However, there are few rigorous statistical methods by which to fit a model in the time domain (Åström and Wittenmark 1984). A breakthrough came with the development of objective model evaluation criteria, such as the final prediction error, FPE, and the Akaike information criterion, AIC, proposed by Akaike (Akaike 1971, 1974; Nakamura and Akaike 1988; Akaike and Nakagawa 1988; Konishi and Kitagawa 2008), which enables identification of a multivariate time series model for real data. The AIC was a useful tool for identifying the actual irregular data observed onboard a ship and for controlling ship motions.

K. Ohtsu et al., *Time Series Modeling for Analysis and Control*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55303-8_1

The present authors have worked to identify actual onboard data and design a marine control system similar to an autopilot or main engine governor. The purpose of this book is to discuss a statistical approach by which to identify a time series model, in particular, a multivariate autoregressive model of observed onboard data, and to control ship motion and main engine behavior using statistical models. Since the models of a ship's autopilot and engine governor discussed herein are typical feedback systems, the authors hope that the readers of this book will be able easily understand the proposed method and apply it in solving problems in their own fields.

1.2 Model of Ship Motion and Main Engine

Before discussing the problems treated herein, we briefly explain a model of the ship motion and main engine. The ship considered herein is a conventional vessel. Thus, special vessels including high-speed launches are not considered herein. As shown in Fig. 1.1, a ship navigating on the sea can be described as moving with six degrees of freedom (Fossen 1994; Lewis 1988).

Roll, pitch, and heave are motions that have restoring forces, whereas sway, surge, and yaw are motions having no restoring force. A ship generally installs with a propeller to control surge motion. Moreover, it is usually not necessary to control sway motion for maneuvering the ship at ocean. However, it is important to maintain and settle her course into a desired one. Thus, a ship is generally equipped with an autopilot system in order to appropriately control yaw. The primary role of autopilot system is to control yaw by rudder and steer the ship to directly follow a desired course. The motion induced by such steering is referred to as a course keeping motion. In this book, the heading deviation from the desired course is referred to as yaw. The secondary role of the autopilot is to alter the course so as to follow another desired course. Course-keeping motions require small-deviation control,

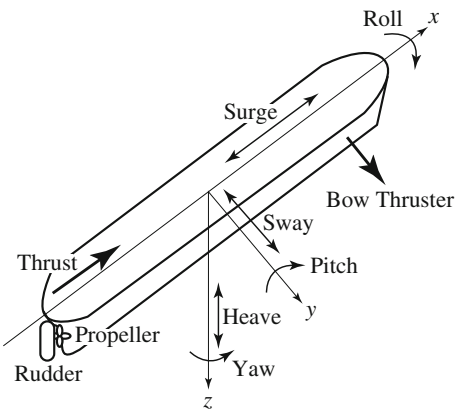

Fig. 1.1 Terms used to describe ship motion

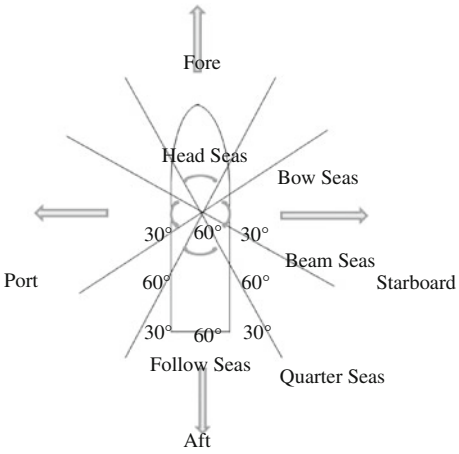

Fig. 1.2 Terms used to describe directions at sea

whereas course-changing motions require large-deviation control. Another role of the autopilot system is to maintain a ship's trajectory along a desired track. Recently, research on a ship's tracking system has been conducted. We also discuss the tracking system in this book.

A ship receives strong disturbances, especially, by wind and waves, from the sea. The scale of the wind force is classified according to the Beaufort scale. However, a general measurement instrument by which to measure the wave height and direction has not yet been standardized. Figure 1.2 shows the terms used in described directions at sea.

On the other hand, the thrusting force of ship is generally generated through the propeller. The rotating force of the propeller is generated by the main engine (Fig. 1.1). The rotation of the propeller cannot maintain a set rotational frequency unless a regulator is properly applied. The engine governor is a device for regulating the amount of fuel supplied to the engine (so that the propeller can maintain the desired rotational frequency). At present, centrifugal governors, which have been widely used as governors in ships, have been gradually replaced by electronic governors because of rapid progress in electronic equipment.

1.3 Experimental Ships and Outline of Topics Discussed in Remaining Chapters

In the following, we use various actual sea test data for modeling and designing autopilot systems. The data were obtained primarily through experiments conducted on "Shioji-Maru II" and "Shioji-Maru III", training ships of Tokyo University of Mercantile Marine. The principal dimensions and main engine specifications of both ships are listed in Table 1.1. Figure 1.3 shows a photograph of T.S. Shioji-Maru III.

Table 1.1 Principal dimensions of Shioji-Maru II and Shioji-Maru III

	T.S. Shioji-Maru II	T.S. Shioji-Maru III
Length	41.70 m	49.93 m
Breadth	8.00 m	10.00 m
Draft	2.575 m	3.01 m
Gross tonnage	331.37 tons	425 tons
Velocity (in voyage)	11.49 knots	14.12 knots
Engine type	Diesel engine	Diesel engine
Horse power	300 PS × 2	1,400 PS
Rated rotation frequency	1,200 rpm	700 rpm

Fig. 1.3 T.S. Shioji-Maru III

We also used a time series that was obtained experimentally on a large container ship. Table 1.2 shows the principal dimensions of the large container ship, referred to herein as "Ship A". Figure 1.4 depicts a typical example of the time history of the ship's yaw, rudder, roll, pitch, propeller shaft revolution, and thrust data. These data are usually obtained through a motion gyro (roll, pitch), an autopilot (yaw, rudder), and a main engine data logger (propeller RPM, thrust, torque, etc.). The sampling rate is generally set to 1–2 s based on sampling theory.

Chapter 2 presents an autoregressive modeling method for both univariate and multivariate stochastic systems. Section 2.1 presents the basic univariate autoregressive (AR) model, its identification method, and the application of the fitted AR model for the analysis of time series. Section 2.2 shows an application of AR modeling to the analysis of a ship's motion.

Section 2.3 presents a multivariate AR model, its identification, and ARX modeling for control systems. Section 2.4 presents a power contribution analysis method, which is a powerful tool for the analysis of control system with feedback loops.

Table 1.2 Principal dimensions of "Ship A"

Length over all	266.65 m
Length between perpendiculars	250.00 m
Breadth	35.40 m
Full load draught	12.72 m
Gross tonnage	44,459 G.T
Engine type	Diesel engine × 1
Max engine power	40,680 kW
Service speed	24.00 knot

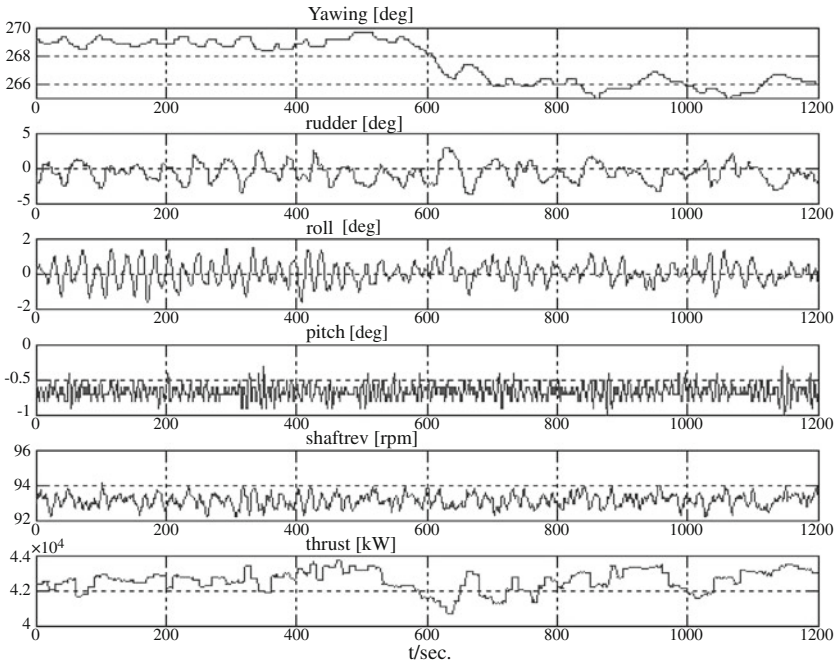

Fig. 1.4 Typical example of time histories of ship motion and propeller and thrust motions on "Ship A"

Section 2.5 shows a state-space representation of the AR model and the Kalman filter, which is a powerful tool for prediction, state estimation, and parameter estimation of the time series model. This state-space model also provides a base for developing an optimal controller of the ship's control system.

As the simplest nonstationary time series model, a locally stationary AR model and its on-line identification method are presented in Sect. 2.6. This model will be used in developing a noise-adaptive autopilot system in Sect. 4.1. A model-based monitoring system will be presented in Sect. 2.7. A radial basis function type ARX

model (RBF-ARX) will be presented in Sect. 2.8. This model provides a practical tool for modeling nonlinear time series and will be used in developing a course tracking autopilot system in Chap. 4.

Chapter 3 presents a method of statistical control and its application to a ship's autopilot system. An optimal control law for the state-space model based on the ARX model and the quadratic loss function will be derived in Sect. 3.1. Two types of criterion function will be used in this chapter. Applications of the stochastic optimal controller for designing an AR-model-based autopilot system and the results of simulations and experiments on an actual ship will be presented in Sect. 3.2.

A rudder-roll control system will be presented in Sect. 3.3. In this section, we show that a proper rudder control enables reduction of both the yaw and roll motions. Section 3.4 is concerned with the control of a marine engine governor.

Chapter 4 presents the extension of the standard statistical controller. A noise-adaptive autopilot system that can adapt to changes in sea disturbances will be presented in Sect. 4.1. An RBF-ARX-model-based nonlinear controller will be presented in Sect. 4.2. A GPS-signal-based computational approach to determining a ship's tracking error and course deviation for implementing tracking control will be presented in Sect. 4.3. Finally, in Sect. 4.4, an RBF-ARX-model-based predictive tracking controller for a marine vehicle will be developed, and the results of simulations and experiments on a real ship will be presented.

References

Akaike, H.: Autoregressive model fitting for control. Ann. Inst. Stat. Math. **23**, 163–180 (1971)

Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **AC–19**, 716–723 (1974)

Akaike, H., Nakagawa, T.: Statistical Analysis and Control of Dynamic System. KTK Scientific Publishers, Tokyo (1988)

Åström, K.J., Wittenmark, B.: Computer Controlled Systems, Theory and Design. Prentice Hall Inc., Englewood Criffs (1984)

Blackman, R.B., Tukey, J.W.: The Measurement of Power Spectra. Dover, New York (1959)

Fossen, T.I.: Guidance and Control of Ocean Vehicles. Wiley, New York (1994)

Isobe, T. (ed.): Covariance Function and Spectrum. Tokyo University Press, Tokyo (1960)

Konishi, S., Kitagawa, G.: Information Criteria and Statistical Modeling. Springer, New York (2008)

Lewis, E.V.: Principles of Naval Architecture, vols. 1–3. Society of Naval Architects and Marine Engineers (1988)

Nakamura, H., Akaike, H.: Statistical identification for optimal control of supercritical thermal power plants. Automatica **17**, 143–155 (1981)

Yamanouchi, Y.: On the analysis of ship oscillation among waves, part 1. J. Soc. Nav. Archit. Jpn. **109**, 169–183 (1961)

Chapter 2
Time Series Analysis Through AR Modeling

Abstract The features of dynamic phenomena can be described using time series models. In this chapter, we present various types of autoregressive models for the analysis of time series, such as univariate and multivariate autoregressive models, an autoregressive model with exogenous variables, a locally stationary autoregressive model, and a radial basis function autoregressive model. Various tools for analyzing dynamic systems such as the impulse response function, the power spectrum, the characteristic roots, and the power contribution are obtained through these models (Akaike and Nakagawa 1989; Kitagawa 2010).

Keywords AR(X) modeling · Ship motion analysis · LSAR model · Ship motion monitoring · RBF-ARX modeling for nonlinear system

2.1 Univariate Time Series Analysis Through AR Modeling

2.1.1 AR Model and Its Identification

A model that expresses a univariate time series y_n as a linear combination of past observations y_{n-i} and white noise v_n is referred to as an autoregressive model (AR model) and has the form

$$y_n = \sum_{i=1}^{m} a_i y_{n-i} + v_n, \tag{2.1}$$

where m and a_i are the autoregressive order and the autoregressive coefficient (AR coefficient), respectively. We assume that v_n is a white noise that follows a normal distribution with mean 0 and variance σ^2 and is independent of the past time series y_{n-i}. In other words, v_n satisfies $\mathrm{E}[v_n] = 0$, $\mathrm{E}[v_n^2] = \sigma^2$, $\mathrm{E}[v_n v_m] = 0$, for $n \neq m$, and $\mathrm{E}[v_n y_m] = 0$, for $n > m$, where E denotes expectation.

K. Ohtsu et al., *Time Series Modeling for Analysis and Control*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55303-8_2

2.1.1.1 Autocovariance Function

Given the time series y_n, the autocovariance function C_k is defined by $C_k = \mathrm{E}[y_n y_{n-k}]$, $k = 0, \pm 1, \ldots$, where k is the lag and, for simplicity, the mean of the time series is assumed to be 0. Taking the expectation after multiplying by y_{n-k} on both sides of (2.1) yields

$$\mathrm{E}[y_n y_{n-k}] = \sum_{i=1}^{m} a_i \mathrm{E}[y_{n-i} y_{n-k}] + \mathrm{E}[v_n y_{n-k}]. \tag{2.2}$$

Therefore, we obtain the following Yule-Walker equation:

$$C_0 = \sum_{i=1}^{m} a_i C_i + \sigma^2, \tag{2.3}$$

$$C_k = \sum_{i=1}^{m} a_i C_{k-i}, \qquad k = 1, 2, \ldots \tag{2.4}$$

A time series is said to be stationary if the mean and the autocovariance function exist and are invariant with time.

Note that, since for univariate time series, the autocovariance function satisfies $C_{-k} = C_k$, Eq. (2.4) also holds, even if C_{k-i} is replaced by C_{k+i}. This means that the backward model satisfies the same equation, and that given the autocovariance function, the forward and backward AR models are identical.

2.1.1.2 Estimation of the AR Model

In order to identify an AR model, it is necessary to determine the order m and estimate the AR coefficients $a_1, \ldots, a_m$ and the variance σ^2 based on the data. Given the time series $y_1, \ldots, y_N$, by computing the sample autocovariance functions

$$\hat{C}_k = \frac{1}{N} \sum_{n=k+1}^{N} y_n \, y_{n-k}, \quad k = 0, 1, \ldots, \tag{2.5}$$

and substituting them into (2.4), we obtain a system of linear equations for the unknown AR coefficients, $a_1, \ldots, a_m$,

$$\begin{bmatrix} \hat{C}_0 & \hat{C}_1 & \cdots & \hat{C}_{m-1} \\ \hat{C}_1 & \hat{C}_0 & \cdots & \hat{C}_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{m-1} & \hat{C}_{m-2} & \cdots & \hat{C}_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \hat{C}_1 \\ \hat{C}_2 \\ \vdots \\ \hat{C}_m \end{bmatrix}. \tag{2.6}$$

By solving this equation, the estimates $\hat{a}_i$ of the AR coefficients are obtained. Then, from (2.3), an estimate of the variance σ^2 is obtained as follows:

$$\hat{\sigma}^2 = \hat{C}_0 - \sum_{i=1}^{m} \hat{a}_i \hat{C}_i. \tag{2.7}$$

The estimates $\hat{a}_1, \ldots, \hat{a}_m$, and $\hat{\sigma}^2$ obtained by this method are referred to as the Yule-Walker estimates.

The log-likelihood of the estimated model is approximately given by

$$\ell = -\frac{N}{2}\left(\log\ 2\pi\hat{\sigma}^2 + 1\right). \tag{2.8}$$

More precise estimates of the AR coefficients and the variance are obtained by the least squares method based on the Householder transformation or the maximum likelihood method (for details, see Kitagawa 2010).

Then, the Akaike information criterion (AIC) of the AR model is obtained approximately as follows:

$$\begin{aligned} \text{AIC} &= -2\ (\text{maximum}\ \ \text{log-likelihood}) + 2\ (\text{number}\ \ \text{of}\ \ \text{parameters}) \\ &\approx N\ \log\ 2\pi\hat{\sigma}^2 + N + 2(m+1). \end{aligned} \tag{2.9}$$

In order to select the AR order m by the minimum AIC method, we calculate the AICs of the AR models with orders of up to M, that is, $\text{AIC}_0, \ldots, \text{AIC}_M$, and select the order that attains the minimum of the AIC values (Akaike 1974; Sakamoto et al. 1986; Konishi and Kitagawa 2008; Kitagawa 2010).

According to Levinson's algorithm, these solutions can be obtained quite efficiently. Hereinafter, the AR coefficients and the innovation variance of the AR model of order m are denoted as a_i^m, $i = 1, \ldots, m$, and σ_m^2, respectively. Then, Levinson's algorithm for obtaining the parameters of the AR models of orders $m = 0, 1, \ldots, M$ is defined as follows (Kitagawa 2010):

1. Set $\hat{\sigma}_0^2 = \hat{C}_0$ and $\text{AIC}_0 = N(\log 2\pi\hat{\sigma}_0^2 + 1) + 2$.
2. Repeat the following steps (a)–(d) for $m = 1, \ldots, M$:
   (a) $\hat{a}_m^m = \left(\hat{C}_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1}\hat{C}_{m-j}\right)\left(\hat{\sigma}_{m-1}^2\right)^{-1}$.
   (b) $\hat{a}_i^m = \hat{a}_i^{m-1} - \hat{a}_m^m \hat{a}_{m-i}^{m-1}$, for $i = 1, \ldots, m-1$.
   (c) $\hat{\sigma}_m^2 = \hat{\sigma}_{m-1}^2\{1 - (\hat{a}_m^m)^2\}$.
   (d) $\text{AIC}_m = N(\log 2\pi\hat{\sigma}_m^2 + 1) + 2(m+1)$.
3. The AIC best order is defined as the minimizer of $\text{AIC}_0, \ldots, \text{AIC}_M$.

2.1.2 Time Series Analysis Using the Univariate AR Model

The AR model estimated from the observed time series can be used to obtain various kinds of information about the time series. In this subsection, we present the impulse response function, the power spectrum, and the characteristic roots.

2.1.2.1 Impulse Response Function

The impulse response function (IRF) of a dynamic system is defined as the output of the IRF when an impulsive input signal is added to the system. Since the impulse function contains all frequencies, the impulse response defines the response of a linear, time-invariant system for all frequencies. All dynamic features of a system can be obtained from the IRF. Using the lag operator B defined by $By_n \equiv y_{n-1}$, the AR model can be expressed as follows:

$$a(B)y_n \equiv \left(1 - \sum_{i=1}^{m} a_i B^i\right) y_n = v_n, \tag{2.10}$$

where $a(B)$ is called the AR operator. Dividing both sides of (2.10) by $a(B)$, the AR model can be expressed as $y_n = a(B)^{-1} v_n$. Therefore, if we define a formal infinite series $g(B)$ as

$$g(B) \equiv a(B)^{-1} = \sum_{i=0}^{\infty} g_i B^i, \tag{2.11}$$

the AR model can be expressed as a linear combination of present and past values of white noise v_n (a moving average model of infinite order):

$$y_n = g(B)v_n = \sum_{i=0}^{\infty} g_i v_{n-i}. \tag{2.12}$$

The coefficients $g_i;\ i = 0, 1, \ldots$, reveal the influence of the noise at time $n = 0$ on the time series at time i, and is referred to as the impulse response function of the AR model. Here, g_i is obtained by the following recursive formula:

$$g_i = \sum_{j=1}^{i} a_j g_{i-j}, \qquad i = 1, 2, \ldots, \tag{2.13}$$

where $g_0 = 1$, $a_j = 0$ for $j > m$. A linear, time-invariant system is completely characterized by its impulse response. That is, for any input, we can calculate the output in terms of the input and the impulse response.

2.1.2.2 Power Spectrum of the AR Process

The Fourier transform of the autocovariance function is referred to as the power spectrum (or spectrum) and expresses the power of the signal at each frequency. If an AR model (2.1) of a time series is given, the power spectrum can be obtained as follows:

$$p(f) = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi ikf} = \frac{\sigma^2}{\left|1 - \sum_{j=1}^{m} a_j e^{-2\pi ijf}\right|^2}, \tag{2.14}$$

where i is the imaginary unit.

There is a close relationship between the AR order and the number of peaks in the spectrum (Kitagawa 2010). The logarithm of the spectrum, $\log p(f)$, is expressible as

$$\log p(f) = \log \sigma^2 - 2 \log \left|1 - \sum_{j=1}^{m} a_j e^{-2\pi ijf}\right|. \tag{2.15}$$

The peaks of the spectrum appear at the local minima of $|1 - \sum_{j=1}^{m} a_j e^{-2\pi ijf}|$. The number of peaks corresponds to the number of roots of the AR operator. Therefore, in order to express k spectral peaks, the AR order must be greater than or equal to $2k$. As will be discussed below, the locations and heights of the peaks are determined by the angles and absolute values, respectively, of the complex roots of the characteristic equation.

2.1.2.3 Characteristic Roots

The characteristics of an AR model are determined by the roots of the following polynomial equation:

$$a(B) = 1 - \sum_{j=1}^{m} a_j B^j = 0. \tag{2.16}$$

Equation (2.16) is referred to as the characteristic equation associated with the AR operator. The roots of this equation are called the characteristic roots. If all roots of the characteristic equation $a(B) = 0$ lie outside the unit circle, or equivalently, the roots of $a(B^{-1}) = 0$ lie inside the unit circle, the influence of noise turbulence at a certain time decays as time progresses. Therefore, the AR model becomes stationary, and the system characterized by the AR model is stable.

As can be seen from Eq. (2.15), the positions of the roots of the characteristic polynomial are closely related to the shape of the spectrum. The peak of the spectrum appears at around $f = \theta/2\pi$, if the complex root of the AR operator is expressed in the following form:

$$z = \alpha + i\beta = re^{i\theta}. \tag{2.17}$$

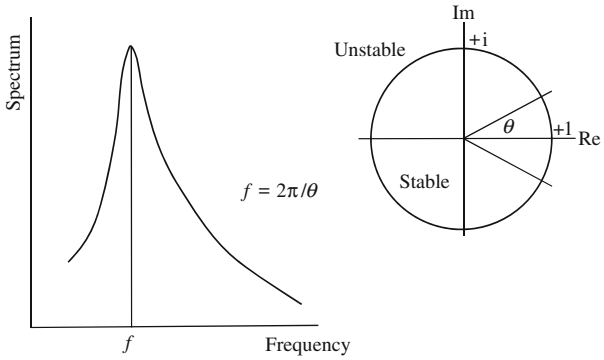

Fig. 2.1 Relationship between the characteristic roots and the spectrum peak frequency

Figure 2.1 illustrates the relationship between θ and f. Furthermore, the more closely the root r approaches 1, the higher the peak of the spectrum becomes.

2.2 Analysis of Ship Motion Through Univariate AR Modeling

In this section, we present an example of time series analysis of ship motion through univariate AR modeling. We discuss roll and pitch, which are typical angular motions of a ship that are induced by restoration forces. For example, a ship is said to have roll stability when the restoration moment is larger than the capsizing moment. Figure 2.2 illustrates the relationship between these moments. In Fig. 2.2, G indicates the center of gravity, B′ indicates the center of buoyancy, and M indicates the metacenter height, which are all technical terms in naval architecture. As the distance GM increases, the lateral stability of the ship increases. Pitch stability can be explained in a similar manner.

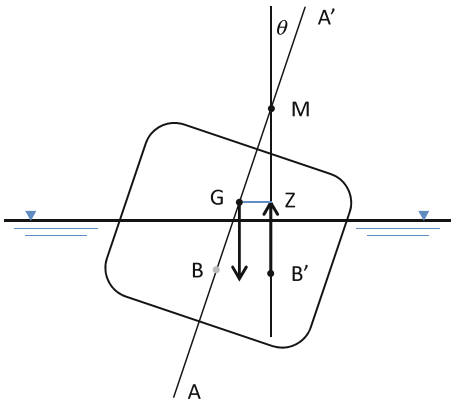

Fig. 2.2 Roll stability. G: center of gravity; B′: center of buoyancy

2.2.1 Features of Roll and Pitch

The salient features of roll and pitch are represented in the spectra as shown in Fig. 2.3. The graph on the left shows the spectra of roll, and the graph on the right shows the spectra of pitch. The data used to plot these graphs were obtained for a large container ship. The dominant peaks of roll motion are concentrated at a specific frequency, and the bandwidth is narrow. In contrast, the spectra for pitch scatter as the bandwidth becomes broad. These observations indicate implicitly that the damping force of the roll is weak, whereas that of the pitch is very strong. In other words, the rolling motion with a natural period is strong, whereas the pitching motion with a natural period is weak. Thus, the pitch responds significantly to external disturbances such as wave forces. Therefore, seafarers know that pitch is an index of wave height.

Figure 2.4 represents the impulse response functions of roll and pitch motions obtained by Eq. (2.13). The roll damping force countering an impulsive disturbance is weak, as shown in Fig. 2.4 (left), whereas the pitch damping force countering an impulsive disturbance is very strong, and the regular response disappears after 15 s.

Figure 2.5 shows the time changes of the roll spectra (left) and the locations of the dominant characteristic roots (right) for successive 50 data sets each of which includes 1,200 observations sampled at 1 s intervals. The power spectra are very stable, and although the magnitudes of the characteristic roots are very close to one, their locations are concentrated at 18-second-cycle movement.

Figure 2.6 shows the time changes of pitch spectra (left) and the locations of their dominant characteristic roots (right) in the local stationary AR model fitting. In this case, the power spectra fluctuate more than the roll spectra. Moreover, the magnitudes of the characteristic roots of pitch are smaller than those of roll. The dominant periods are scattered from 24 to 6 s. As mentioned in Sect. 2.1.2.3, the peak of the spectrum appears at $f = \theta/2\pi$.

Figure 2.7 shows the scatter plot of the dominant and subdominant periods of roll with four typical wave patterns of roll in different positions. The horizontal axis denotes the dominant period of roll, and the vertical axis indicates the subdominant period of roll. As the dominant and subdominant periods approach each other, the wave pattern of roll time history forms a "group wave".

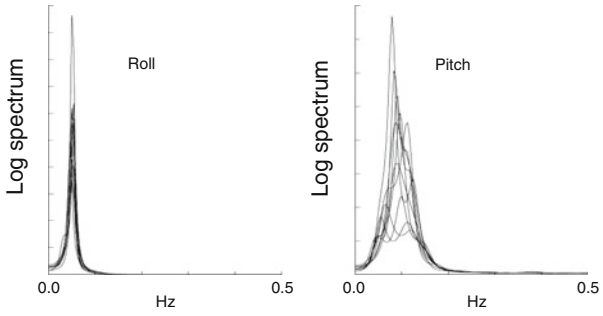

Fig. 2.3 Power spectra of roll (*left*) and pitch (*right*)

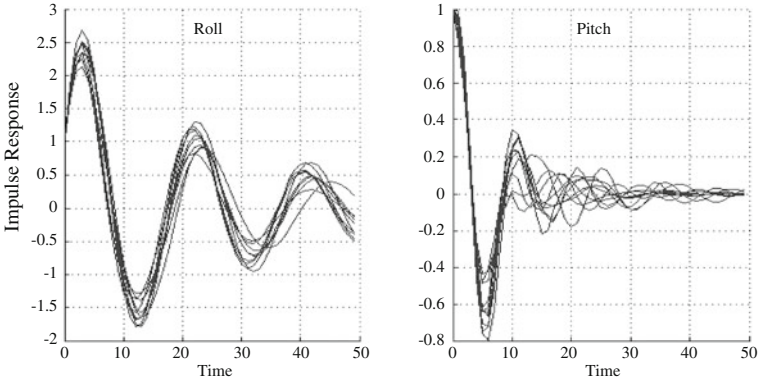

Fig. 2.4 Impulse response of roll (*left*) and pitch (*right*) motions

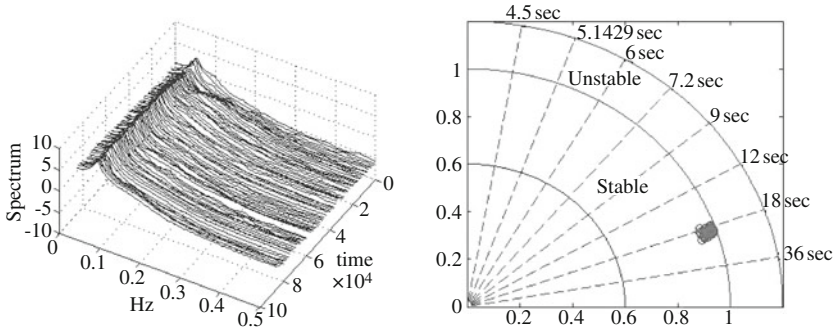

Fig. 2.5 Changes of roll spectra (*left*) and the locations of the dominant characteristic roots (*right*)

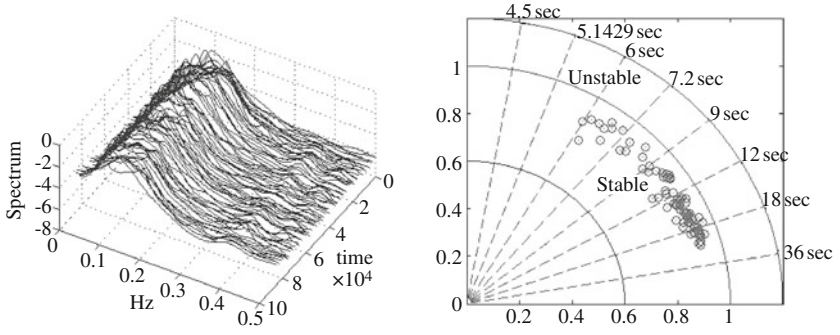

Fig. 2.6 Changes of pitch spectra (*left*) and the locations of the dominant characteristic roots (*right*)

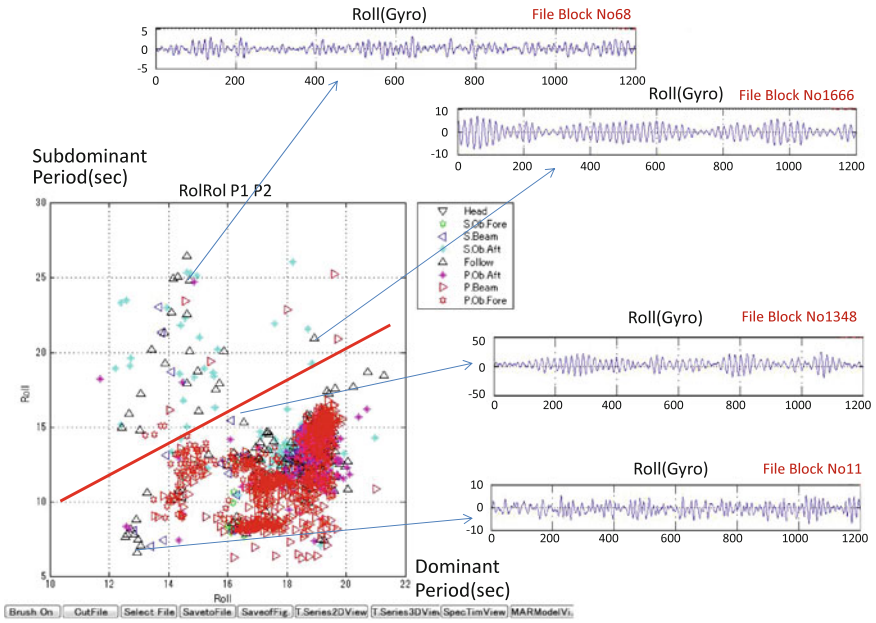

Fig. 2.7 Locations of dominant and sub-dominant periods and patterns of roll

2.2.2 Roll Stability

Roll stability is an important matter not only for ship designers, but also for mariners. When the roll period abruptly shifts longer during sailing, in other words, the magnitude of the characteristic root exceeds a radius of 1 in the complex plane, the risk of capsizing immediately increases.

We can use the scatter diagram of roll and pitch dominant periods to avoid the risk of losing roll stability. Figure 2.8 shows a scatter diagram of the dominant period of roll (horizontal axis) and the dominant period of pitch (vertical axis). This diagram reveals important information on roll stability.

Points plotted on the 45-degree line in the diagram indicate that both motions will be synchronized and that the shape of the roll motion will gradually form as a group wave, which means that the risk of capsizing increases, as shown in Fig. 2.7. We refer to this line as the synchronizing roll line.

On the other hand, for points plotted on the 22.5-degree line, along which the roll period is equal to twice the pitch period, the rolling motion may become a large rolling motion known as parametric rolling. This phenomenon is well known in nonlinear vibration theory. We refer to the 22.5-degree line as the parametric rolling line.

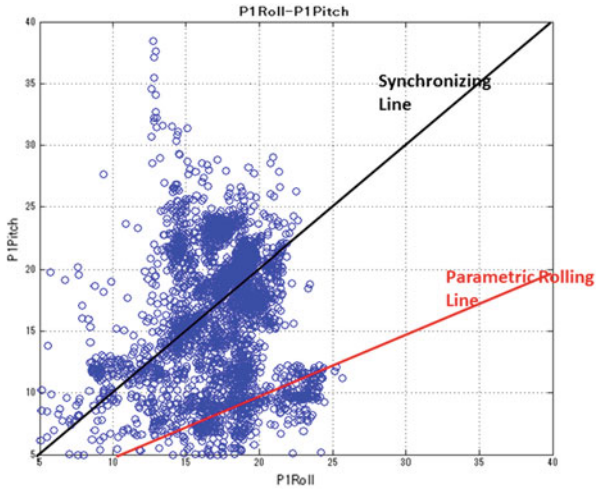

Fig. 2.8 Scatter diagram of the period of roll (*horizontal axis*) and the period of pitch (*vertical axis*)

2.2.3 Increasing Horizon Prediction of Roll and Pitch

Increasing horizon prediction of ship motion is important for safely operating a ship. Applying Kalman filtering to the state-space representation of the AR model described in Sect. 2.5.2, we can reasonably predict ship motion. Figure 2.9 shows the long-term predictions of roll and pitch motions. In these figures, the AR models of roll and pitch are fitted using data from the beginning until 340 s, and the roll and pitch motions from 340 to 360 s are then predicted. The AR model can reasonably predict the future values.

2.3 Multivariate AR Modeling of Controlled Systems

2.3.1 Multivariate AR Model

Assume that $\boldsymbol{y}_n = (y_n(1), \ldots, y_n(\ell))^T$ is a stationary multivariate time series, where ℓ is the dimension of the time series and n is the time. For simplicity, the mean of the time series, $\mathrm{E}[y_n(i)]$, is assumed to be 0 for $i = 1, \ldots, \ell$. A model that expresses the present value of the time series as a linear combination of past values $\boldsymbol{y}_{n-1}, \ldots, \boldsymbol{y}_{n-M}$ and the white noise $\boldsymbol{v}_n$

$$\boldsymbol{y}_n = \sum_{j=1}^{m} A_j \boldsymbol{y}_{n-j} + \boldsymbol{v}_n, \tag{2.18}$$

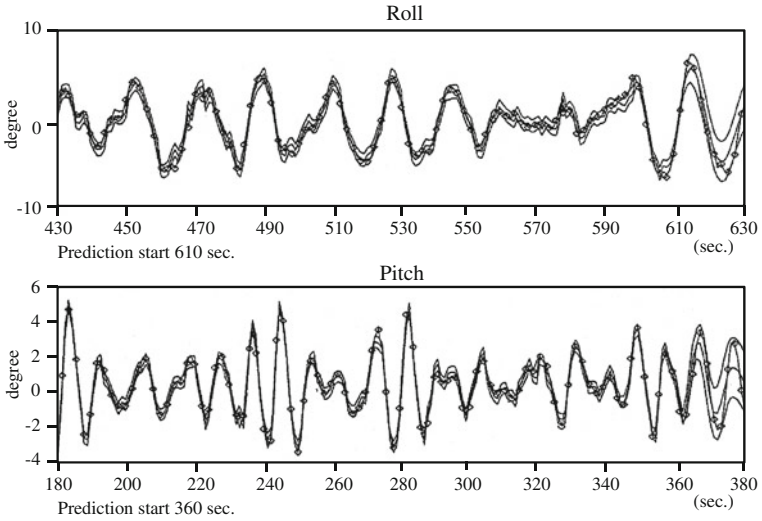

Fig. 2.9 Long-term prediction of roll and pitch. Prediction starts at $n = 610$ for roll and $n = 360$

is referred to as a multivariate autoregressive model (MAR model), where m is the order, A_j is the autoregressive coefficient matrix whose (i, k)th element is given by $a_j(i, k)$, and $\boldsymbol{v}_n$ is an ℓ-dimensional white noise that satisfies $\mathrm{E}[\boldsymbol{v}_n] = [0, \ldots, 0]^T$,

$$\mathrm{E}[\boldsymbol{v}_n \boldsymbol{v}_n^T] = W = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1\ell} \\ \vdots & \ddots & \vdots \\ \sigma_{\ell 1} & \cdots & \sigma_{\ell\ell} \end{bmatrix}, \tag{2.19}$$

$\mathrm{E}[\boldsymbol{v}_n \boldsymbol{v}_j^T] = \mathrm{O}$, for $n \neq j$ and $\mathrm{E}[\boldsymbol{v}_n \boldsymbol{y}_j^T] = \mathrm{O}$, for $n > j$. Here, O denotes the $\ell \times \ell$ matrix with 0 elements, and W is an $\ell \times \ell$ symmetric positive semi-definite matrix satisfying $\sigma_{ij} = \sigma_{ji}$.

The cross-covariance of $\boldsymbol{y}_n(i)$ and $\boldsymbol{y}_n(j)$ at time lag k is defined as $C_k(i, j) = \mathrm{E}\big[y_n(i) y_{n-k}(j)\big]$. Then, the $\ell \times \ell$ matrix $C_k = \mathrm{E}[\boldsymbol{y}_n \boldsymbol{y}_{n-k}^T], k = 0, 1, \cdots$, the (i, j)th component of which is given by $C_k(i, j)$, is referred to as the cross-covariance function. For the multivariate AR model, the cross-covariance function C_k satisfies the Yule-Walker equation

$$C_0 = \sum_{j=1}^{m} A_j C_{-j} + W \tag{2.20}$$

$$C_k = \sum_{j=1}^{m} A_j C_{k-j} \qquad (k = 1, 2, \ldots). \tag{2.21}$$

Note that, unlike in the univariate case, the cross-covariance function is not symmetric with respect to time lag and satisfies $C_{-k} = C_k^T$.

2.3.2 Identification of Multivariate AR Model

In this subsection, identification methods, such as the parameter estimation and the order selection of the multivariate AR model, are presented. The parameters of the multivariate AR model of order m

$$\boldsymbol{y}_n = \sum_{i=1}^{m} A_i^m \boldsymbol{y}_{n-i} + \boldsymbol{v}_n, \qquad \boldsymbol{v}_n \sim N(0, W_m), \tag{2.22}$$

are the variance-covariance matrix W_m of the innovation $\boldsymbol{v}_n$ and the AR coefficient matrices $A_1^m, \ldots, A_m^m$ (Akaike and Nakagawa 1989). The number of unknown parameters is $m\ell^2 + \ell(\ell+1)/2$. In this subsection, since we consider AR models of various orders, the order m is explicitly shown as the superscript of the coefficient.

If a multivariate AR model is given, the cross-covariance function is obtained by Eqs. (2.20) and (2.21). Here, assume that the sample cross-covariance function $\hat{C}_k$, $k = 0, 1, \ldots, m$, is obtained from observed time series by

$$\hat{C}_k(i, j) = \frac{1}{N} \sum_{\ell=k+1}^{N} y_n(i) y_{n-\ell}(j). \tag{2.23}$$

Then, by substituting these into Eq. (2.21), the AR coefficients of the parameters of the multivariate AR model, $\hat{A}_j^m$, can be obtained by solving the Yule-Walker equation:

$$\hat{C}_k = \sum_{j=1}^{m} A_j^m \hat{C}_{k-j}, \qquad (k = 1, \ldots, m). \tag{2.24}$$

Substituting the estimated AR coefficient matrices $\hat{A}_j^m$ into Eq. (2.20) yields the estimate of the variance covariance matrix W_m, as follows:

$$\hat{W}_m = \hat{C}_0 - \sum_{j=1}^{m} \hat{A}_j^m \hat{C}_j. \tag{2.25}$$

In actual modeling, the order m is unknown and must be determined based on the data. The order m can be determined using the minimum AIC procedure (Akaike 1974, 1998; Konishi and Kitagawa 2008). In this method, we compute the AIC_m $m = 0, 1, \ldots, M$, for a properly selected highest-order M (Kitagawa 2010),

$$\begin{aligned}\mathrm{AIC}_m &= -\max(\text{log-likelihood}) + 2(\text{number of parameters}) \\ &= N\ell \log 2\pi + N \log |W_m| + 2m\ell^2 + \ell(\ell+1). \end{aligned} \tag{2.26}$$

Then, the AIC best order m^* is determined as the minimizer of the AIC_m from among the orders $m = 0, 1, \ldots, M$, i.e.,

$$m^* = \arg \min_{m=0,1,\ldots,M} \mathrm{AIC}_m. \tag{2.27}$$

Therefore, in order to perform the minimum AIC procedure, it is necessary to fit all AR models and compute the associated AIC values for the orders, $0, 1, \ldots, M$. The AIC values can be obtained efficiently using the following algorithms. In the case of a univariate time series, the forward AR model coincides with the backward AR model, because the autocovariance function is an even function. The computationally efficient Levinson's algorithm is derived based on this property, which is not satisfied by a multivariate time series. Therefore, in order to derive an efficient algorithm similar to Levinson's algorithm for multivariate time series, in addition to Eq. (2.22), we should consider the backward multivariate AR model

$$\boldsymbol{y}_n = \sum_{i=1}^{m} B_i^m \boldsymbol{y}_{n+i} + \boldsymbol{u}_n, \qquad u_n \sim N(0, U_m), \tag{2.28}$$

and we need to estimate the variance-covariance matrix U_m and the coefficients B_i^m, as well as A_i^m and W_m, simultaneously (Whittle 1963; Kitagawa 2010).

Assume that the sample cross-covariance function $\hat{C}_j$, $j = 0, \ldots, M$ is given. The minimum AIC procedure for fitting the multivariate AR model based on the Levinson-Whittle algorithm is then defined as follows. In this recursive algorithm, the mth AR coefficient A_m^m plays a crucial role and is called the PARCOR (Partial autocorrelation) matrix.

1. Set $W_0 = U_0 = \hat{C}_0$ and compute the AIC of the AR model of order 0 as $\mathrm{AIC}_0 = N(k \log 2\pi + \log |\hat{W}_0| + k) + \ell(\ell+1)$.
2. For $m = 1, \ldots, M$, repeat the following steps (a)–(e).
   (a) $W_m = \hat{C}_m - \sum_{i=1}^{m-1} A_i^{m-1} \hat{C}_{m-i}$.
   (b) Obtain the PARCOR matrices of the forward and backward AR models by $A_m^m = W_m U_{m-1}^{-1}$ and $B_m^m = W_m^T V_{m-1}^{-1}$.
   (c) Compute the AR coefficients of the forward and backward AR models by $A_i^m = A_i^{m-1} - A_m^m B_{m-i}^{m-1}$ and $B_i^m = B_i^{m-1} - B_m^m A_{m-i}^{m-1}$ for $i = 1, \ldots, m-1$.
   (d) Estimate the innovation variance-covariance matrices by $\hat{W}_m = \hat{C}_0 - \sum_{i=1}^{m} A_i^m \hat{C}_i^T$ and $\hat{U}_m = \hat{C}_0 - \sum_{i=1}^{m} B_i^m \hat{C}_i$.
   (e) Compute the AIC value of the AR model of order m by $\mathrm{AIC}_m = N(k \log 2\pi + \log |\hat{W}_m| + k) + \ell(\ell+1) + 2\ell^2 m$.
3. Find the minimum of the AIC_m among $m = 0, \ldots, M$. The AIC best order, m^*, is the minimizer of the AICs, and m^*, $\hat{A}_j^{m^*}$ and $\hat{W}_{m^*}$ are the identified model.

Remark According to the above-mentioned algorithm, we compute $\mathrm{AIC}_0, \ldots,$ AIC_M, and select the m that attains the minimum of the AIC values as the best order of the multivariate AR model. In this method, it is implicitly assumed that the AR coefficients $a_m(i, j)$ have common orders for all i and j. Using the least squares method based on the Householder transformation, it is possible to determine the AIC best order for each pair of (i, j) (Kitagawa 2010). Furthermore, applying this method, it is possible to determine the best order without explicitly obtaining the AR coefficient matrices. The FORTRAN program MULMAR in the program package TIMSAC-78 (Akaike et al. 1979) can be used for this purpose. Furthermore, the maximum likelihood estimates of the multivariate AR model are obtained using the state-space representation of the model and the Kalman filter (Kitagawa 2010).

2.3.3 ARX Model for a Control System

Suppose that the ℓ-dimensional time series $\boldsymbol{y}_n$ consists of p-dimensional output variables $\boldsymbol{s}_n = (s_n(1), \ldots, s_n(p))^T$ and q-dimensional input variables $\boldsymbol{r}_n = (r_n(1), \ldots, r_(q))^T$, so that $\ell = p + q$ and $\boldsymbol{y}_n = (\boldsymbol{s}_n^T, \boldsymbol{r}_n^T)^T$.

The autoregressive exogenous model (ARX model) with inputs $\boldsymbol{r}_n$ and outputs $\boldsymbol{s}_n$ is given by

$$\boldsymbol{s}_n = \sum_{j=1}^{m} a_j \boldsymbol{s}_{n-j} + \sum_{j=1}^{m} b_j \boldsymbol{r}_{n-j} + \boldsymbol{u}_n \,, \tag{2.29}$$

where a_j and b_j are $p \times p$ and $p \times q$ matrices, and $\boldsymbol{u}_n$ is a p-dimensional white noise with covariance matrix $W_{r,m}$.

Note that this ARX model is a part of the AR model for ℓ-dimensional time series

$$\boldsymbol{y}_n = \sum_{j=1}^{m} A_j \boldsymbol{y}_{n-j} + \boldsymbol{v}_n, \tag{2.30}$$

with the relation

$$A_j = \begin{bmatrix} a_j & b_j \\ * & * \end{bmatrix}, \quad v_n = \begin{bmatrix} u_n \\ * \end{bmatrix}, \quad W_m = \begin{bmatrix} W_{r,m} & * \\ * & * \end{bmatrix}. \tag{2.31}$$

The symbol $*$ indicates that this part of the matrix is not used in the ARX model. This means that the parameters of the ARX model are obtained as part of the multivariate AR model for the time series $\boldsymbol{y}_n$. Therefore, the Yule-Walker estimates of a_j and b_j, $j = 1, \ldots, m$ can be obtained from those of A_j, $j = 1, \ldots, m$.

However, the best order for this ARX model is not necessarily the same as that of the multivariate AR model for $\boldsymbol{y}_n$. The AIC for the ARX model is given by

$$\mathrm{AIC}_m = N \log |W_{r,m}| + 2p(p+q)m + p(p+1), \tag{2.32}$$

where N is the data length, and $|W_{r,m}|$ is the determinant of the estimate of the variance covariance matrix of the innovation $\boldsymbol{u}_n$ of the ARX model of order m. Moreover, the sum of the second and third terms on the right-hand side is equal to twice the number of parameters of this model. According to the minimum AIC procedure, the order that attains the minimum of AIC_m is considered to be the best model (Akaike 1974; Konishi and Kitagawa 2008).

Remark Similar to the case of the AR model, a more sophisticated model with a different order for each variable can be obtained by the least squares method based on the Householder transformation.

2.4 Power Contribution Analysis of a Feedback System

2.4.1 Power Contribution of a Feedback System

For multivariate time series, $\boldsymbol{y}_n = (y_n(1), \ldots, y_n(\ell))^T$, assume that a multivariate autoregressive model (MAR model) is given as

$$\boldsymbol{y}_n = \sum_{j=1}^{m} A_j \boldsymbol{y}_{n-j} + \boldsymbol{v}_n, \tag{2.33}$$

where A_j is the autoregressive coefficient matrix whose (i, j)th element is given by $a_m(i, j)$, and $\boldsymbol{v}_n$ is an ℓ-dimensional white noise with mean 0 and cross-covariance matrix W.

The cross-covariance function of the time series $y_n(i)$ and $y_n(j)$ is defined as $C_k(i, j) = \mathrm{E}\big[y_n(i)y_{n-k}(j)\big]$ for $k = 0, 1, \ldots, M$ and the $\ell \times \ell$ matrix whose (i, j) component is $C_k(i, j)$ is denoted by C_k.

The cross-spectrum matrix $P(f)$ is defined as

$$P(f) = \begin{bmatrix} p_{11}(f) & \cdots & p_{1\ell}(f) \\ \vdots & \ddots & \vdots \\ p_{\ell 1}(f) & \cdots & p_{\ell\ell}(f) \end{bmatrix} = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f}. \tag{2.34}$$

For time series that follow the multivariate AR model, the cross-spectrum can be obtained by Whittle (1963), Akaike and Nakagawa (1989)

$$P(f) = A(f)^{-1} W \left(A(f)^{-1}\right)^*, \tag{2.35}$$

where A^* denotes the complex conjugate matrix of A, and $A(f)$ denotes the $\ell \times \ell$ matrix whose (j, k)th component is defined by

$$A_{jk}(f) = \sum_{m=0}^{M} a_m(j, k) e^{-2\pi i m f}, \tag{2.36}$$

with $a_0(j, j) = -1$ and $a_0(j, k) = 0$ for $j \neq k$.

For convenience, $A(f)^{-1}$ will be denoted as $B(f) = (b_{jk}(f))$ in the following. If the components of the white noise $\boldsymbol{v}_n$ are mutually uncorrelated and the variance-covariance matrix becomes the diagonal matrix $W = \mathrm{diag}\{\sigma_1^2, \ldots, \sigma_\ell^2\}$, then the power spectrum of the ith component of the time series can be expressed as follows:

$$p_{ii}(f) = \sum_{j=1}^{\ell} b_{ij}(f)\sigma_j^2 b_{ij}(f)^* \equiv \sum_{j=1}^{\ell} |b_{ij}(f)|^2 \sigma_j^2. \tag{2.37}$$

This indicates that the power of the fluctuation of $\boldsymbol{y}_n(i)$ at frequency f can be decomposed into the effects of ℓ noises and the term $|b_{ij}(f)|^2\sigma_j^2$ is referred to as the absolute power contribution. Therefore, if we define $r_{ij}(f)$ as follows:

$$r_{ij}(f) = \frac{|b_{ij}(f)|^2\sigma_j^2}{p_{ii}(f)}, \tag{2.38}$$

then $r_{ij}(f)$ represents the ratio of the power of fluctuation that can be expressed as the effect of $v_n(j)$ to the power of the fluctuation of $y_n(i)$ at frequency f. Here, $r_{ij}(f)$ is referred to as the relative power contribution, which is a convenient tool for the analysis of a feedback system (Akaike 1968; Akaike and Nakagawa 1989). When drawing figures, it is convenient to use the cumulative power contribution defined by

$$s_{ij}(f) = \sum_{k=1}^{j} r_{ik}(f) = \frac{1}{p_{ii}(f)} \sum_{k=1}^{j} |b_{ik}(f)|^2 \sigma_k^2. \tag{2.39}$$

Remark For many real time series, the assumption of the diagonality of the variance covariance matrix W of the white noise v_n may be too restrictive. A general positive definite matrix W can be expressed as follows:

$$W = \sum_{i=2}^{\ell} \sum_{j=1}^{i-1} s_{ij} J_{ij} J_{ij}^T + \sum_{i=1}^{\ell} s_i J_i J_i^T. \tag{2.40}$$

Using this expression, the generalized power contribution is defined as

$$r_{ijk}(f) = \begin{cases} \dfrac{|\rho_{jk}||\sqrt{\sigma_{jj}}b_{ij} \pm \sqrt{\sigma_{kk}}b_{ik}|^2}{P_{ii}(f)} & (j = 2, \ldots, \ell; k = 1, \ldots, j-1), \\ \dfrac{\tau_j \sigma_{jk}|b_{ij}|^2}{P_{ii}} & (j = 1, \ldots, \ell; k = j). \end{cases} \tag{2.41}$$

For details and some applications to financial time series analysis, see Tanokura and Kitagawa (2004) and Tanokura et al. (2012).

Remark Most real-world systems have feedback loops. If significant feedback loops exist, it is impossible to obtain unbiased estimates of the impulse response function unless the noise input to the system is a white noise sequence. However, under the assumption of diagonality of the noise covariance matrix W, it is possible to obtain unbiased estimates of the impulse response function, even in the presence of feedback loops (Akaike and Nakagawa 1989). Assume that the feedback system can be expressed as

$$y_n(i) = \sum_{j=1}^{\ell} \sum_{m=1}^{\infty} \alpha_m(i, j) y_{n-m}(j) + u_n(j), \tag{2.42}$$

where it is assumed that $\alpha_m(i, i) = 0$ and that the noise input $u_n(j)$ can be expressed by an AR model

$$u_n(j) = \sum_{i=1}^{m} c_i(j) u_{n-i}(j) + \varepsilon_n(j), \tag{2.43}$$

with $\varepsilon_n(j)$ being mutually independent white noise.

In this situation, if a multivariate AR model of order m, Eq. (2.33), is given, we can obtain estimates of the coefficients $\alpha_m(i, j)$ and $c_i(j)$ by

$$\begin{aligned} c_m(i) &= A_m(i, i), \quad m = 1, \ldots, M \text{ and } i = 1, \ldots, \ell, \\ \alpha_m(i, j) &= A_m(i, j) + \sum_{k=1}^{m-1} c_k(i) \alpha_{m-k}(i, j), \quad m = 1, \ldots, M \\ \alpha_m(i, j) &= \sum_{k=1}^{m-1} c_k(i) \alpha_{m-k}(i, j), \quad m = M + 1, \ldots. \end{aligned} \tag{2.44}$$

2.4.2 Analysis of Ship Feedback Motion

Table 2.1 and Fig. 2.10 shows the results of fitting multivariate AR models of orders of up to 20 by the Yule-Walker method to the data obtained from a large container ship. The determinant of the prediction error covariance matrix W_m decreases monotonically with the order. However, the AIC is minimized at $m^* = 7$, and gradually increases for orders $m > m^*$. The identified multivariate AR model will be used in the power contribution analysis mentioned in Sect. 2.2.

Figure 2.11 shows the power contribution obtained by fitting a multivariate AR model to the five-variate time series composed of the yaw rate, the roll, the pitch rate, the propeller revolutions per minute (RPM), and the rudder angle ($N = 500$ and $\triangle t = 2\,\mathrm{s}$). The AR order determined by the AIC criterion was 7 (see Table 2.1 and Fig. 2.10). Since the correlation matrix calculated from W_m is

Table 2.1 AICs of multivariate AR models fitted to ship data

m	$\lvert W_m \rvert$	AIC_m	m	$\lvert W_m \rvert$	AIC_m	m	$\lvert W_m \rvert$	AIC_m
0	54107.23	12574.05	7	0.674	7277.31	14	0.429	7401.03
1	40.310	9023.00	8	0.611	7278.28	15	0.405	7422.76
2	2.397	7661.76	9	0.582	7307.31	16	0.388	7451.84
3	1.498	7476.90	10	0.537	7313.84	17	0.370	7477.29
4	0.965	7306.65	11	0.510	7338.02	18	0.346	7494.14
5	0.830	7281.64	12	0.474	7351.75	19	0.330	7520.22
6	0.749	7280.16	13	0.446	7370.59	20	0.316	7548.03

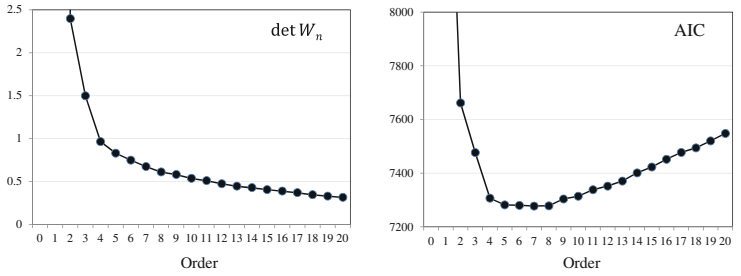

Fig. 2.10 Changes in $|W_m|$ and AIC_m of multivariate AR models for $m = 0, 1, \ldots, 20$, fitted to five-variate ship data under manual control

$$R = \begin{bmatrix} 1 & -0.032 & 0.299 & 0.165 & -0.001 \\ -0.032 & 1 & -0.107 & 0.156 & 0.036 \\ 0.299 & -0.107 & 1 & 0.255 & -0.043 \\ 0.165 & 0.156 & 0.255 & 1 & 0.083 \\ -0.001 & 0.036 & -0.043 & 0.083 & 1 \end{bmatrix}, \tag{2.45}$$

the assumption of the power contribution analysis seems reasonable.

Figure 2.11 shows the power contribution analysis of ship motions when a large container ship was steered in manual mode. From top to bottom, the power contributions to the power of the yaw rate, the roll, the pitch, the propeller RPM, and the rudder angle from each motion are shown. The panels on the left-hand side show the cumulative absolute power contributions defined in Eq. (2.38), whereas the panels on the right-hand side show the cumulative relative power contribution defined in Eq. (2.39). From these plots, we can see that the yaw rate, the pitch, the propeller RPM, and the rudder angle have the same dominant frequency of the power spectra, around a frequency of $f = 0.04$ Hz. On the other hand, the dominant frequency of the roll is located at approximately $f = 0.06$ Hz. The yaw rate and the propeller RPM have the second spectral peaks near this frequency.

Strong contributions to yaw rate are observed in the yaw rate itself, in their dominant frequency range under these sea conditions. On the other hand, at 0.04 Hz, the contribution to pitch is mostly due to the pitch itself and the propeller RPM. Approximately 50 % of the contribution to the power of the propeller RPM at its

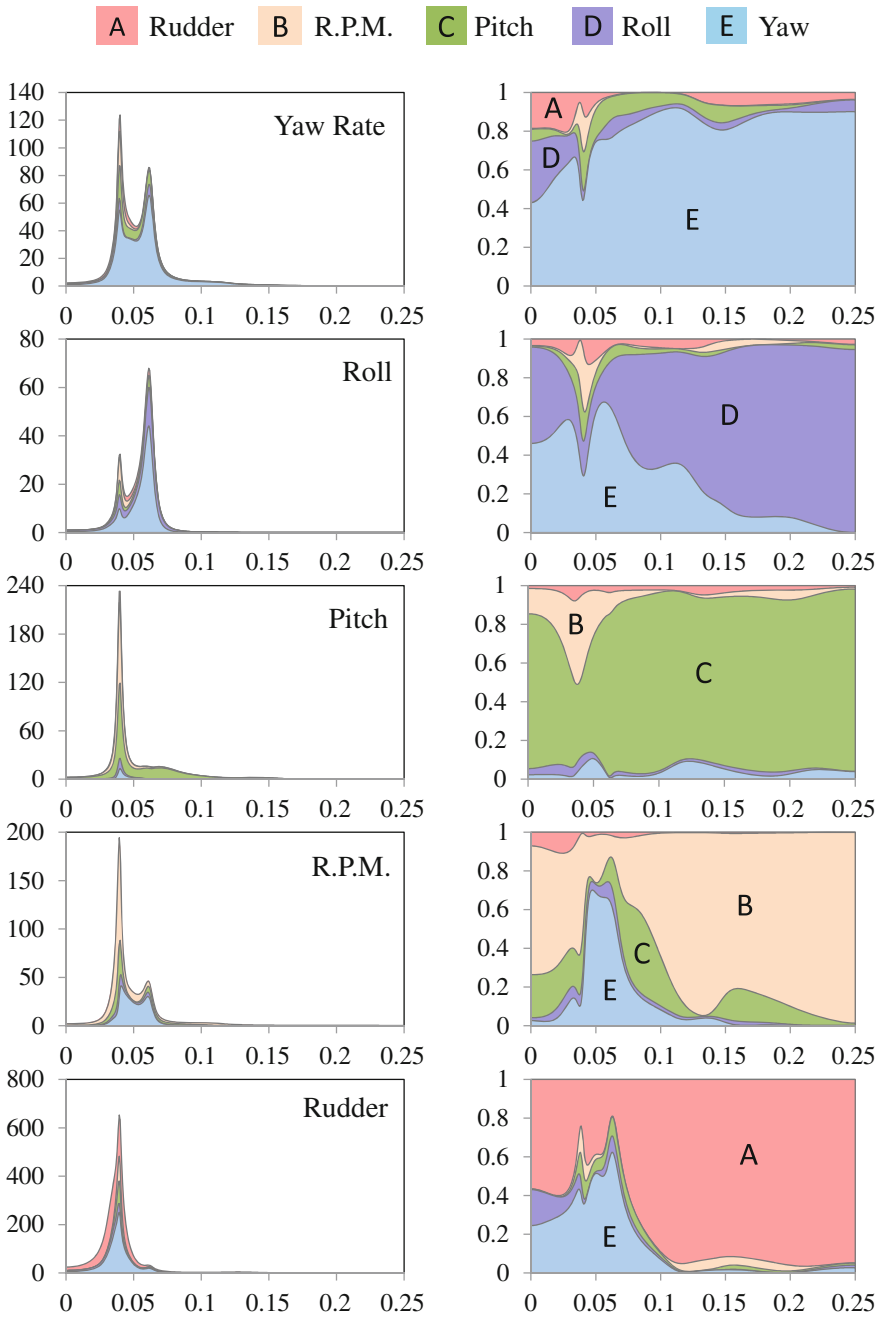

Fig. 2.11 Power contribution analysis of ship motion under manual control

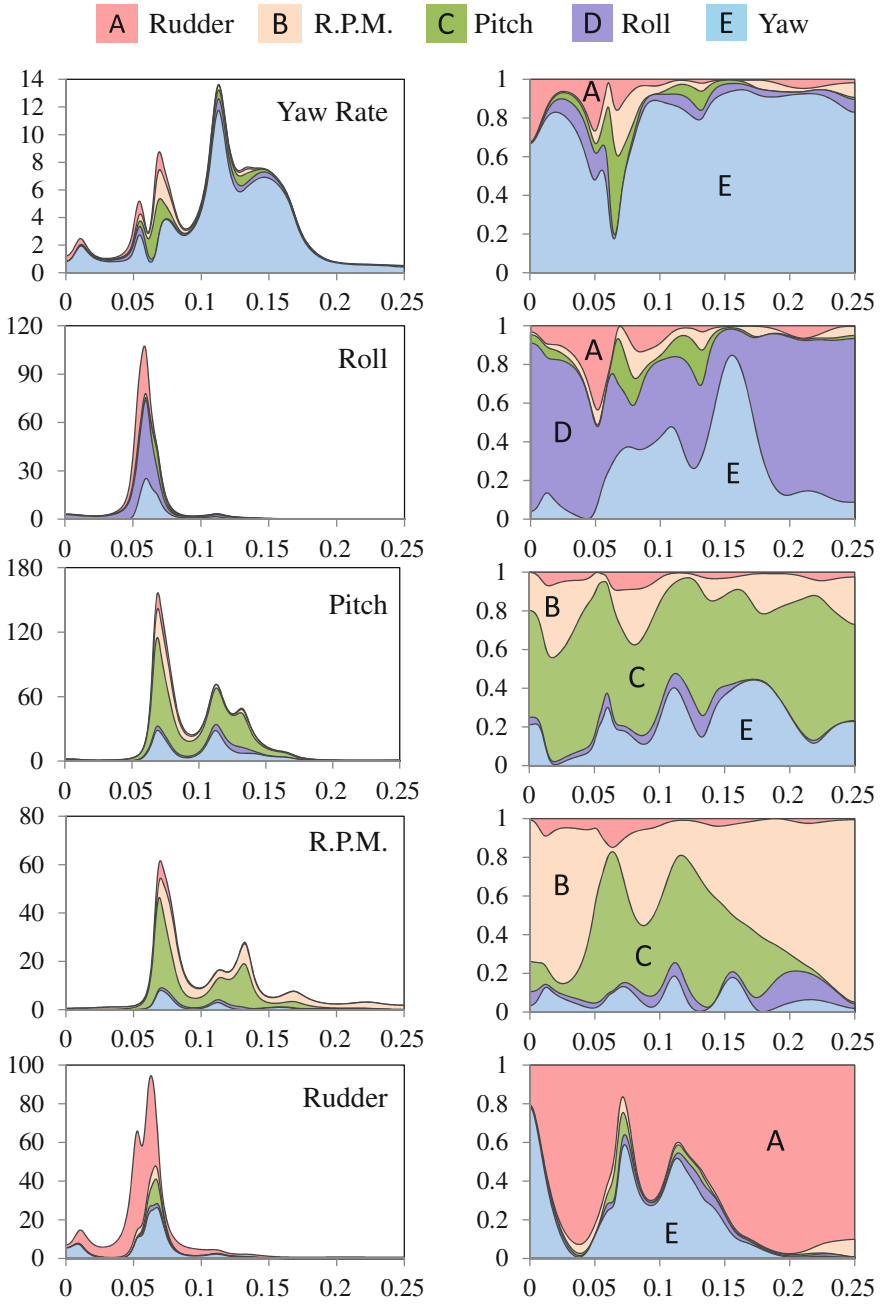

Fig. 2.12 Power contribution analysis of ship motion under autopilot control

dominant frequency is provided by the yaw rate and comes primarily from the yaw rate around the second spectrum peak. At the frequency domain $f < 0.07$ Hz, the contribution of the yaw rate to the power of the rudder angle is very strong, and is as high as 50 %. This suggests that the human operator reasonably feeds back the yaw rate at a frequency lower than 0.1 Hz.

Figure 2.12 shows the power contribution analysis of the same ship under a conventional PID autopilot system. In this case, the yaw rate has the largest spectrum power at frequency $f = 0.12$ Hz and smaller peaks at $f = 0.07$ Hz and 0.05 Hz. The roll has its spectral peak at 0.05 Hz. On the other hand, the pitch, the propeller RPM, and the rudder angle have the highest peaks at approximately 0.07 Hz. The pitch and the propeller RPM have other peaks around $0.1 < f < 0.15$ Hz.

Then, we can see from the figure that approximately 50 % of the power of roll is contributed by the yaw rate and the rudder angle. It might be suggested that the ship sailed under the condition of the well-known rudder-roll-yaw effect. Concerning the propeller RPM, the effect of pitch is significant, e.g., approximately 60–80 % at the dominant frequency and the second largest peak. This suggests that the pitch has an especially strong influence on the change of the propeller RPM, and it is necessary to take this effect into account in designing a new governor.

Approximately 50 % of the effect of pitch on the rudder angle is from the yaw rate or the pitch at the dominant frequency, $f = 0.07$ Hz. On the other hand, at lower frequencies, $0.02\,\text{Hz} < f < 0.05\,\text{Hz}$, the effect of pitch on the rudder angle is primarily from its own peak frequency. This suggests that the conventional PID controller feeds back only in the frequency range of $f > 0.05$ Hz.

2.5 State-Space Model and Kalman Filter

Various time series models can be treated entirely within the state-space model framework. Many problems in time series analysis can be formulated in terms of the state estimation of a state-space model. This section presents algorithms for the Kalman filter and a smoothing algorithm for efficient state estimation.

2.5.1 State-Space Model

Assume that $\boldsymbol{y}_n$ is an ℓ-variate time series. The following model for the time series is called a state-space model.

$$\boldsymbol{x}_n = F\boldsymbol{x}_{n-1} + G\boldsymbol{v}_n, \qquad \text{(system model)} \tag{2.46}$$

$$\boldsymbol{y}_n = H\boldsymbol{x}_n + \boldsymbol{w}_n, \qquad \text{(observation model)}, \tag{2.47}$$

where $\boldsymbol{x}_n$ is a k-dimensional unobservable vector, referred to as the state (Anderson and Moore 1979). $\boldsymbol{v}_n$ and $\boldsymbol{w}_n$ are m-dimensional and ℓ-dimensional Gaussian white noises with mean vector zero and variance-covariance matrices Q and R and are

referred to as system noise and observation noise, respectively. Moreover, F, G, and H are $k \times k$, $k \times m$, and $\ell \times k$ matrices, respectively. Many linear models used in time series analysis are expressible in terms of state-space models.

2.5.1.1 State-Space Representation of an AR Model

Assume that an autoregressive (AR) model for ℓ-dimensional time series $\boldsymbol{y}_n$

$$\boldsymbol{y}_n = \sum_{j=1}^{m} A_j \ \boldsymbol{y}_{n-j} + \boldsymbol{v}_n \ , \tag{2.48}$$

is given, where A_j is an $\ell \times \ell$ matrix and $\boldsymbol{v}_n$ is an ℓ-dimensional white noise with mean 0 and covariance matrix Q.

In order to obtain the state-space representation of an AR model, we define an $m\ell$-dimensional vector $\boldsymbol{x}_n$ by $\boldsymbol{x}_n \equiv \left[\boldsymbol{y}_n^T, \boldsymbol{y}_{n-1}^T, \ldots, \boldsymbol{y}_{n-m}^T\right]^T$. Then, it can be easily confirmed that the MAR model can be expressed in state-space model form (Kitagawa 2010):

$$\begin{cases} \boldsymbol{x}_n = F\boldsymbol{x}_{n-1} + G\boldsymbol{v}_n \\ \boldsymbol{y}_n = H\boldsymbol{x}_n \ , \end{cases} \tag{2.49}$$

where F, G, and H are defined as

$$F = \begin{bmatrix} A_1 & A_2 & \cdots & A_m \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad H = [\ I \ \ 0 \ \ \cdots \ \ 0\ ]. \tag{2.50}$$

This state-space representation is not unique, and another state-space representation of the AR model is obtained by defining a variable $\tilde{\boldsymbol{y}}_{n+k|n-1}$ by

$$\tilde{\boldsymbol{y}}_{n+k|n-1} = \sum_{j=k+1}^{m} A_j \boldsymbol{y}_{n+k-j}, \qquad (k = 1, \ldots, m-1). \tag{2.51}$$

$\tilde{\boldsymbol{y}}_{n+k|n-1}$ is a part of $\boldsymbol{y}_{n+k}$ that can be directly expressed by the time series and the white noise until time $n-1$. Whether the following relation holds can be confirmed

$$\begin{aligned} \boldsymbol{y}_n &= A_1 \boldsymbol{y}_{n-1} + \tilde{\boldsymbol{y}}_{n|n-2} + \boldsymbol{v}_n \\ \tilde{\boldsymbol{y}}_{n+k|n-1} &= A_{k+1} \boldsymbol{y}_{n-1} + \tilde{\boldsymbol{y}}_{n+k|n-2}, \quad k = 1, \ldots, m-1 \\ \tilde{\boldsymbol{y}}_{n+m-1|n-1} &= A_m \boldsymbol{y}_{n-1}. \end{aligned} \tag{2.52}$$

Defining the $m\ell$-dimensional state vector $\boldsymbol{x}_n$ as $\boldsymbol{x}_n \equiv [\boldsymbol{y}_n^T, \tilde{\boldsymbol{y}}_{n+1|n-1}^T, \ldots, \tilde{\boldsymbol{y}}_{n+m-1|n-1}^T]^T$, the AR model (2.48) can be expressed in state-space model form (2.49), where F, G, and H are defined as follows:

$$F = \begin{bmatrix} A_1 & I & \cdots & 0 \\ A_2 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & I \\ A_m & 0 & \cdots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad H = [\, I \;\; 0 \;\; \cdots \;\; 0 \,]. \tag{2.53}$$

Furthermore, another state space representation is obtained by defining the state-space as $\boldsymbol{x}_n \equiv [\boldsymbol{y}_n^T, \boldsymbol{y}_{n+1|n}^T, \ldots, \boldsymbol{y}_{n+m-1|n}]^T$, where $\boldsymbol{y}_{n+k|n}$ is the best predictor of $\boldsymbol{y}_{n+k}$ given the observations up to time n. In this case, the matrices are given by

$$F = \begin{bmatrix} 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \\ A_m & A_{m-1} & \cdots & A_1 \end{bmatrix}, \quad G = \begin{bmatrix} I \\ g_1 \\ \vdots \\ g_{m-1} \end{bmatrix}, \quad H = [\, I \;\; 0 \;\; \cdots \;\; 0 \,]. \tag{2.54}$$

where $g_j, \quad j = 1, \ldots, m-1$, is the impulse response of the AR model defied by $g_0 = I$ and $g_j = \sum_{i=1}^{j} A_i g_{j-i}$.

In general, given the state-space representation given by Eqs. (2.46) and (2.47), for any non-singular $k \times k$ matrix T, by defining a new state x_n' and the matrices F', G', and H' by

$$x_n' = T x_n, \quad F_n' = T F_n T^{-1}, \quad G_n' = T G_n, \quad H_n' = H T^{-1}, \tag{2.55}$$

we obtain an equivalent state-space model.

2.5.1.2 State-Space Representation of the ARX Model

Assume that the following autoregressive exogenous (ARX) model is given as

$$\boldsymbol{y}_n = \sum_{j=1}^{m} A_j \boldsymbol{y}_{n-j} + \sum_{j=1}^{m} B_j \boldsymbol{r}_{n-j} + \boldsymbol{v}_n \,, \tag{2.56}$$

where $\boldsymbol{y}_n$ and $\boldsymbol{r}_n$ are p-dimensional output variables and q-dimensional input variables, A_j is a $p \times p$ matrix, B_j is a $p \times q$ matrix and $\boldsymbol{v}_n$ is p-dimensional white noise.

In order to design an optimal controller for this ARX model based on the optimal control theory, it is convenient to express the model in state-space model form. In order to obtain the state-space representation of an ARX model, we define a new variable $\tilde{\boldsymbol{y}}_{n+k|n-1}$ by

$$\tilde{\boldsymbol{y}}_{n+k|n-1} = \sum_{j=k+1}^{m} A_j \boldsymbol{y}_{n+k-j} + \sum_{j=k+1}^{m} B_j \boldsymbol{r}_{n+k-j}, \qquad (k = 1, \ldots, m-1). \quad (2.57)$$

$\tilde{\boldsymbol{y}}_{n+k|n-1}$ is exactly the part of $\boldsymbol{y}_{n+k}$ that can be directly expressed by the observations of the output, the input, and the white noise until time $n-1$. Whether the following relation holds can be easily confirmed:

$$\begin{aligned} \boldsymbol{y}_n &= A_1 \boldsymbol{y}_{n-1} + B_1 \boldsymbol{r}_{n-1} + \tilde{\boldsymbol{y}}_{n|n-2} + \boldsymbol{v}_n \\ \boldsymbol{y}_{n+k|n-1} &= A_{k+1} \boldsymbol{y}_{n-1} + B_{k+1} \boldsymbol{r}_{n-1} + \tilde{\boldsymbol{y}}_{n+k|n-2}, \quad k = 1, \ldots, m-1 \\ \boldsymbol{y}_{n+m-1|n-1} &= A_m \boldsymbol{y}_{n-1} + B_m \boldsymbol{r}_{n-1}. \end{aligned} \quad (2.58)$$

Defining the pm-dimensional state vector $\boldsymbol{x}_n$ by $\boldsymbol{x}_n \equiv \left[\boldsymbol{y}_n^T, \boldsymbol{y}_{n+1|n-1}^T, \ldots, \boldsymbol{y}_{n+m-1|n-1}^T\right]^T$, the ARX model (2.56) is expressed as (Akaike 1971; Akaike and Nakagawa 1988)

$$\begin{cases} \boldsymbol{x}_n = \Phi \boldsymbol{x}_{n-1} + \Gamma \boldsymbol{r}_{n-1} + G u_n \\ \boldsymbol{y}_n = H \boldsymbol{x}_n \,, \end{cases} \quad (2.59)$$

where Φ, Γ, G, and H are defined by

$$\Phi = \begin{bmatrix} A_1 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{m-1} & 0 & \cdots & I \\ A_m & 0 & \cdots & 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_m \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad H = [\, I \;\; 0 \;\; \cdots \;\; 0 \,]. \quad (2.60)$$

Since we can restore the latest data in a form to be used in the future, the value of $\boldsymbol{y}_n$ in the next step can be predicted by a simple and small calculation when the newest data are obtained.

2.5.2 State Estimation and Kalman Filter

The most important problem in state-space modeling is to estimating the state $\boldsymbol{x}_n$ based on the time series $\boldsymbol{y}_n$. The reason for this is that problems such as prediction, signal extraction, and likelihood computation for the time series can be systematically performed using the estimated state.

In this subsection, we shall consider the problem of estimating the state $\boldsymbol{x}_n$ based on the set of observations $Y_j = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_j\}$. Depending on the relation between j and n, the state estimation problem is classified into three categories: prediction ($j < n$), filter ($j = n$), and smoothing ($j > n$).

For linear-Gaussian state-space model, it is sufficient to obtain the conditional mean $\boldsymbol{x}_{n|j}$ and the covariance matrix $V_{n|j}$, which can be efficiently obtained computationally by means of the recursive computational algorithm shown below. This

algorithm is known as the Kalman filter (Kalman 1960; Anderson and Moore 1979). Nonlinear non-Gaussian extensions of the Kalman filter are given in Kitagawa (1996, 2010) and Doucet et al. (2001).

[One-step-ahead prediction]

$$\begin{aligned} \boldsymbol{x}_{n|n-1} &= F\boldsymbol{x}_{n-1|n-1} \\ V_{n|n-1} &= FV_{n-1|n-1}F^T + GQG^T. \end{aligned} \tag{2.61}$$

[Filter]

$$\begin{aligned} K_n &= V_{n|n-1}H^T(HV_{n*n-1}H^T + R)^{-1} \\ \boldsymbol{x}_{n|n} &= \boldsymbol{x}_{n|n-1} + K_n(\boldsymbol{y}_n - H\boldsymbol{x}_{n|n-1}) \\ V_{n|n} &= (I - K_nH)V_{n|n-1}. \end{aligned} \tag{2.62}$$

The fixed-interval smoothing yields the conditional mean and covariance matrix based on the entirety of the observations.

Fixed-interval smoothing

$$\begin{aligned} A_n &= V_{n|n}F^TV_{n+1|n}^{-1} \\ \boldsymbol{x}_{n|N} &= \boldsymbol{x}_{n|n} + A_n(\boldsymbol{x}_{n+1|N} - \boldsymbol{x}_{n+1|n}) \\ V_{n|N} &= V_{n|n} + A_n(V_{n+1|N} - V_{n+1|n})A_n^T. \end{aligned} \tag{2.63}$$

In order to perform fixed-interval smoothing, we first obtain $\boldsymbol{x}_{n|n-1}$, $\boldsymbol{x}_{n|n}$, $V_{n|n-1}$, $V_{n|n}, n = 1, \ldots, N$, by using the Kalman filter and compute $\boldsymbol{x}_{N-1|N}$, $V_{n-1|N}$ through $\boldsymbol{x}_{1|N}$, $V_{1|N}$ backward in time.

2.5.3 Likelihood Computation and Parameter Estimation for a Time Series Model

The Kalman filter provides a convenient and computationally efficient tool for estimating the parameters of the time series model. Assume that the state-space representation for a time series model specified by a parameter vector $\boldsymbol{\theta}$ is given. When a time series $Y_j = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ of length N is given, the N-dimensional joint density function of Y_N specified by this time series model is denoted by $f_N(Y_N|\theta)$. Then, the likelihood of this model is given as follows:

$$L(\boldsymbol{\theta}) = f_N(Y_N|\boldsymbol{\theta}). \tag{2.64}$$

By repeatedly applying the relation

$$f_n(y_n|\boldsymbol{\theta}) = f_{n-1}(Y_{n-1}|\boldsymbol{\theta})g_n(\boldsymbol{y}_n|Y_{n-1}, \boldsymbol{\theta}),$$

for $n = N,\ N-1, \ldots, 2$, the likelihood of the time series model can be expressed as a product of conditional density functions:

$$L(\boldsymbol{\theta}) = \prod_{n=1}^{N} g_n(\boldsymbol{y}_n|Y_{n-1}, \boldsymbol{\theta}). \tag{2.65}$$

For simplicity of notation, we let $Y_0 = \emptyset$ (empty set), and then $f_1(\boldsymbol{y}_1|\boldsymbol{\theta}) \equiv g_1(\boldsymbol{y}_1|Y_0, \boldsymbol{\theta})$. By taking the logarithm of $L(\boldsymbol{\theta})$, the log-likelihood of the model is obtained as

$$\ell(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log g_n(\boldsymbol{y}_n|Y_{n-1}, \boldsymbol{\theta}). \tag{2.66}$$

Since $g_n(\boldsymbol{y}_n|Y_{n-1}, \boldsymbol{\theta})$ is the conditional distribution of y_n given the observation Y_{n-1} and is, in fact, a normal distribution with mean $y_{n|n-1}$ and variance-covariance matrix $d_{n|n-1}$, $g_n(\boldsymbol{y}_n|Y_{n-1}, \boldsymbol{\theta})$ can be expressed as follows (Kitagawa and Gersch 1996):

$$\begin{aligned} &g_n(\boldsymbol{y}_n|Y_{n-1}, \boldsymbol{\theta}) \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^{\ell} |d_{n|n-1}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_n - \boldsymbol{y}_{n|n-1})^T d_{n|n-1}^{-1}(\boldsymbol{y}_n - \boldsymbol{y}_{n|n-1})\right\}. \end{aligned} \tag{2.67}$$

Therefore, by substituting this density function into Eq. (2.66), the log-likelihood of this state-space model is obtained as

$$\begin{aligned} \ell(\theta) = -\frac{1}{2}\Bigg\{&\ell N \log 2\pi + \sum_{n=1}^{N} \log |d_{n|n-1}| \\ &+ \sum_{n=1}^{N} (\boldsymbol{y}_n - \boldsymbol{y}_{n|n-1})^T d_{n|n-1}^{-1}(\boldsymbol{y}_n - \boldsymbol{y}_{n|n-1})\Bigg\}. \end{aligned} \tag{2.68}$$

Stationary time series models, such as AR models, and many other nonstationary time series models, such as trend and seasonal adjustment models, can be expressed in the form of a linear Gaussian state-space model. Accordingly, for such time series models, a unified algorithm for computing the log-likelihood can be obtained by using the Kalman filter and Eq. (2.68). Furthermore, the maximum likelihood estimates of the parameters of the time series model can be obtained by maximizing this log-likelihood by a numerical optimization method.

The state-space model can be generalized to a nonlinear non-Gaussian version. Although the Kalman filter cannot yield efficient estimates of the state, various types of recursive filtering and smoothing algorithms have been developed (Kitagawa 1987, 1996, 2010; Kitagawa and Gersch 1996; Doucet et al. 2001).

2.6 Piecewise Stationary Modeling

2.6.1 Locally Stationary AR Model

Assume that the ℓ-dimensional time series $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ is nonstationary as a whole, but if the time interval $\{1, \ldots, N\}$ is properly divided into several subintervals, it becomes stationary on each subinterval. Then, it is natural to consider an AR model for each stationary subseries, which are referred to as locally stationary AR models (Ozaki and Tong 1975; Kitagawa and Akaike 1978; Kitagawa and Gersch 1996; Kitagawa 2010) (Fig. 2.13).

More specifically, assume that k and N_j denote the number of subintervals and the number of observations in the jth subinterval ($N_1 + \cdots + N_k = N$), respectively. Note that the starting and end points of the ith subinterval $[n_{i0}, n_{i1}]$ are respectively given by

$$n_{j0} = \sum_{i=1}^{j-1} N_i + 1, \qquad n_{j1} = \sum_{i=1}^{j} N_i.$$

For a multivariate locally stationary AR model, the time series $\boldsymbol{y}_n$ follows an AR model

$$\boldsymbol{y}_n = \sum_{i=1}^{m_j} A_i^{(j)} \boldsymbol{y}_{n-i} + \boldsymbol{v}_{nj}, \tag{2.69}$$

on the jth subinterval, where $\boldsymbol{v}_{nj}$ is an ℓ-dimensional Gaussian white noise with $\mathrm{E}(\boldsymbol{v}_{nj}) = \boldsymbol{0}$, $\mathrm{E}(\boldsymbol{v}_{nj}\boldsymbol{v}_{nj}^T) = \Sigma_j$, and $\mathrm{E}(\boldsymbol{v}_{nj}\boldsymbol{y}_{n-m}^T) = \boldsymbol{0}$ for $m = 1, 2, \ldots$ Here, the unknown parameters of the model include the number of subintervals, k, the length of the jth interval, N_j, the AR order, m_j, the AR coefficient matrices, $A_j = \{A_i^{(j)}, \ldots, A_{m_j}^{(j)}\}$, and the variance covariance matrix of the white noise, Σ_j.

The likelihood of the locally stationary AR model is given by

$$L = p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) = \prod_{j=1}^{k} \prod_{n=n_{j0}}^{n_{j1}} p(\boldsymbol{y}_n | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n-1}), \tag{2.70}$$

where $p(\boldsymbol{y}_n | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n-1})$ denotes the conditional distribution of $\boldsymbol{y}_n$ given the past observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n-1}$. Since the noise input is assumed to be Gaussian, the conditional distribution also becomes Gaussian.

Therefore, for simplicity, ignoring the distributions of the first m_1 data points and replacing N_1 by $N_1 - m_1$ and n_{10} by $m_1 + 1$, the likelihood of the multivariate locally stationary AR model is approximated by

$$\prod_{j=1}^{k} \left\{ \frac{|\Sigma_j|}{(2\pi)^\ell} \right\}^{\frac{N_j}{2}} \exp\left\{ -\frac{1}{2} \sum_{n=n_{j0}}^{n_{j1}} \boldsymbol{v}_{nj}^T \Sigma_j^{-1} \boldsymbol{v}_{nj} \right\} \tag{2.71}$$

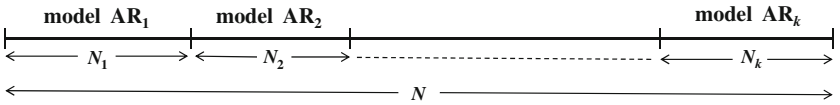

Fig. 2.13 Subdivision of the time interval

Then, the log-likelihood function is given by

$$\begin{aligned}&\ell\,(k, N_j, m_j, A_j, \Sigma_j;\, j = 1, \ldots, k)\\ &\quad = -\frac{1}{2}\sum_{j=1}^{k}\left\{\ell N_j \log 2\pi + N_j \log|\Sigma_j| + \sum_{n=n_{j0}}^{n_{j1}} \boldsymbol{v}_{nj}^T \Sigma_j^{-1} \boldsymbol{v}_{nj}\right\}. \end{aligned} \tag{2.72}$$

Similar to the stationary multivariate AR model shown in Sect. 2.3, the AR coefficients $A_i^{(j)}$ and the innovation variance covariance matrix Σ_j on each interval can be obtained by minimizing

$$N_j \log|\Sigma_j| + \sum_{n=n_{j0}}^{n_{j1}} \boldsymbol{v}_{nj}^T \Sigma_j^{-1} \boldsymbol{v}_{nj}. \tag{2.73}$$

A computationally efficient procedure for the fitting of these models is presented in Takanami and Kitagawa (1991). Substituting these estimates into (2.72), the log-likelihood becomes

$$\begin{aligned}&\ell\ \ (k, N_j, m_j, \hat{A}_j, \hat{\Sigma}_j;\, j = 1, \ldots, k)\\ &\quad = -\frac{1}{2}\sum_{j=1}^{k}\left\{\ell N_j \log(2\pi) + N_j \log|\Sigma_j| + \ell N_j\right\}\\ &\quad = -\frac{\ell(N - m_1)}{2}(\log 2\pi + 1) - \frac{1}{2}\sum_{j=1}^{k} N_j \log|\Sigma_j|. \end{aligned} \tag{2.74}$$

Since the number of estimated parameters is $m_j\ell^2 + \ell(\ell-1)/2$, the AIC value for the locally stationary AR model is given by

$$\mathrm{AIC} = (N - m_1)(\log 2\pi + 1) + \sum_{j=1}^{k} N_j \log|\hat{\Sigma}_j| + \sum_{j=1}^{k}\left\{2m_j\ell^2 + \ell(\ell-1)\right\}. \tag{2.75}$$

This AIC value depends on the number of subintervals k, the length of the jth subinterval N_j, and the order of the AR model for the jth interval m_j. Unknown parameters are obtained by finding the combination of parameters that achieves the minimum AIC value among all possible candidates.

2.6.2 On-Line Identification of the Locally Stationary AR Model

In on-line adaptive control, we can obtain a new dataset successively. A semi-on-line identification procedure for fitting the locally stationary AR (LSAR) model was developed by Ozaki and Tong (1979). According to this procedure, the homogeneity of the data is checked by comparing the AICs of two models, namely, the divided model and the pooled model.

Using this method, we shall determine the basic span L and the highest-order M of the AR model fitted to the subinterval of length L. Here, L is set to a sufficient length so that an AR model of order M can be fitted on an interval of length L. Then, only points $n_i = iL$ are considered as candidates for dividing points. The dividing points of the locally stationary AR model can automatically be decided by the following procedure.

1. **Fit the initial model**: Fit AR models of orders of up to M to the initial time series $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L$, compute $\mathrm{AIC}_0(0), \ldots, \mathrm{AIC}_0(m)$, and find the minimum AIC order $m^* = \arg\min_{j=0,1,\ldots,M} \mathrm{AIC}_0(j)$. Furthermore, set $k = 1$, $n_{10} = m+1$, $n_{11} = L$ and $N_1 = L - m$.
2. **Fit the divided model**: Fit AR models with orders of up to M to the newly obtained time series $\boldsymbol{y}_{n_{k1}+1}, \ldots, \boldsymbol{y}_{n_{k1}+L}$, compute $\mathrm{AIC}_1(0), \ldots, \mathrm{AIC}_1(m)$, and set $\mathrm{AIC}_1 = \min_j \mathrm{AIC}_1(j)$. AIC_1 is the AIC of a new model that was obtained under the assumption that the model changed at time $n_{k1} + 1$. The AIC of the locally stationary AR model that divides the interval $[n_{k0},\ n_{k1} + L]$ into two subintervals, $[n_{ko},\ n_{k1}]$ and $[n_{k1} + 1,\ n_{k1} + L]$, is given by

$$\mathrm{AIC}_D = \mathrm{AIC}_0 + \mathrm{AIC}_1.$$

   This model is referred to as the divided model.
3. **Fit the pooled model**: Assuming $\boldsymbol{y}_{n_{k0}}, \ldots, \boldsymbol{y}_{n_{k1}+L}$ to be stationary, fit AR models of orders of up to M, compute $\mathrm{AIC}_P(0), \ldots, \mathrm{AIC}_P(M)$, and find the minimum AIC order $\mathrm{AIC}_P = \min_j \mathrm{AIC}_P(j)$. Under the assumption that the time series on the entire interval $[n_{k0}, n_{k1} + L]$ is stationary, the model is referred to as the pooled model.
4. **Judge the homogeneity of data**: In order to judge the homogeneity of the two subintervals, compare the values of AIC_D and AIC_P.

   Switch to the new model: If $\mathrm{AIC}_D < \mathrm{AIC}_P$, it is judged that the divided model is better. In this case, $n_{k1} + 1$ becomes the initial point of the current subinterval, and we put $k \equiv k + 1$, $n_{k0} \equiv n_{k-1,1} + 1$, $n_{k1} = n_{k-1,1} + L$, $N_k = L$, and $\mathrm{AIC}_0 = \mathrm{AIC}_D$.

   Pool the new dataset: If $\mathrm{AIC}_D \geq \mathrm{AIC}_P$, the pooled model is adopted. In this case, the new subinterval $[n_{k1}+1, n_{k1}+L]$ is merged with the former subinterval, and $[n_{k0}, n_{k1} + L]$ becomes the new current subinterval. Therefore, we set $n_{k1} \equiv n_{k1} + L$, $N_k = N_k + L$, and $\mathrm{AIC}_0 = \mathrm{AIC}_P$.

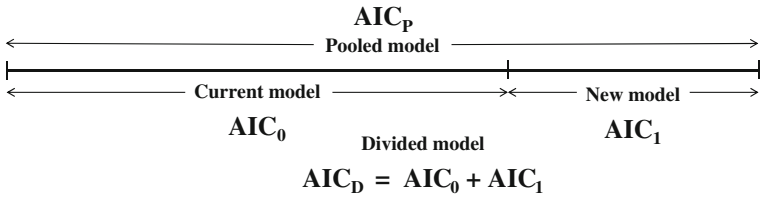

Fig. 2.14 On-line identification of the LSAR model

5. **Repeat the process**: If we have additional time series with L observations, return to step 2. Otherwise, the number of subintervals is k, and the stationary subintervals are judged to be $[1, n_{11}], [n_{20}, n_{21}], \ldots, [n_{k0}, N]$.

In addition, we fit two types of AR models whenever an additional time series of length L remains to be modeled. This estimation process can be carried out efficiently and precisely by the Householder least squares method (Kitagawa and Akaike 1978; Kitagawa 2010) (Fig. 2.14).

2.7 Model-Based Monitoring System

Since actual sea conditions may change gradually or abruptly depending on various factors, seafarers must carefully monitor such changes at all times. When abnormalities occur at sea, installed marine controllers, such as autopilot systems and engine governors, adaptively cope with such difficulties. In such situations, the monitoring models are classified into nonstationary time series. The simplest and most practical approach to modeling nonstationary time series is to subdivide the time interval into several subintervals of appropriate size and then fit a stationary AR model to each subinterval. Using this method, we can obtain a series of models that approximate nonstationary time series. In this section, we develop an onboard model-based system for monitoring ship states and a noise-adaptive autopilot system based on a locally stationary AR model.

2.7.1 Motivation

Data loggers are installed in ships in order to maintain the safety of main engines and reduce fuel consumption. The collected data are summarized every hour and are recorded in the ship's engine logbook. The data are used to estimate the ship's long-term performance. The International Maritime Organization (IMO) has recently recommended an index for energy savings referred to as the Energy Efficiency Design Index (EEDI) for ship builders and the Energy Efficiency Operational Indicator (EEOI) for ship operators.

Based on the recommendations of the IMO, we have designed an entirely new ship-born model-based monitoring system (SBMMS) by applying an AR model to satisfy the requirements of the EEOI (Ohtsu 2009). The proposed monitoring system is designed to instantaneously provide information about the main engine state and the ship motion to the mariners, because the most common reason for state changes of the main engine is hull motion.

2.7.2 Ship-Born Model-Based Monitoring System (SBMMS)

The conditions around a ship at sea are constantly changing. However, the rate of change is slow. As such, we apply the locally stationary fitting procedure described in Sect. 2.6.1 to obtain the current AR model. Table 2.2 shows the state variables used in the monitoring system. Among them, roll, pitch, and yaw are fundamental information of ship motions, which are observed by a motion gyro sensor. The rudder angle can be varied to control the course deviation, and the number of revolutions per minute (RPM) of the propeller is the control variable for thrusting the ship body.

Engine power P is generated from the main engine according to the following relation:

$$P = 2\pi n Q, \tag{2.76}$$

Table 2.2 State variables for the monitoring system

Variable	Recorder	Variable	Recorder
Time	GPS	Heave accel.	Motion gyro
Latitude	GPS	Sway accel.	Motion gyro
Longitude	GPS	Propeller RPM	Shaft
Yaw	Autopilot	Torque	Shaft
Command rudder	Autopilot	Thrust	Shaft
Actual rudder	Autopilot	Power	Engine
Roll	Motion gyro	Fuel oil	Engine
Pitch	Motion gyro		

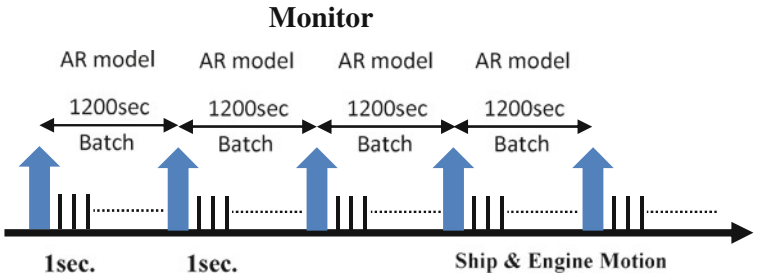

Fig. 2.15 Time schedule for obtaining the model in the monitoring system

where n denotes the propeller RPM/60, and Q is the generated shaft torque. There are two measures of the ship's speed, namely, the speed relative to the water (log speed) and the speed relative to the ground (OG speed), which is usually observed by the Global Positioning System (GPS). As wind resistance information, the wind force W_f and wind direction W_d are observed by ship-born wind meters. The wind resistance, W_R, that a ship experiences is simply calculated as follows:

$$W_R = W_f \cos W_D. \tag{2.77}$$

In contrast, correct wave information has not yet been obtained in the proposed measurement technique. Figure 2.15 shows the time schedule for obtaining the model in the monitoring system designed herein. The fundamental sampling rate is set to 1 s. The length of one data batch is set to 600 s. for a small ship and 1,200 s for a large ship.

Figure 2.16 shows the configuration of the ship-born model-based monitoring system (SBMMS). The system includes a gain scheduling function of the autopilot system, which will be discussed in a later chapter. Moreover, rather than sending all of the raw data that are collected, only the statistical parameters gained from the fitted model are sent from the ship. This means that the need for communication via satellite is extremely rare.

Next, we present examples of the display of the designed SBMMS. The data in all figures displayed here were collected on a large container ship sailing on the Pacific Ocean. Figure 2.17 shows an example of the time history of the data sampled every 1 s.

Figure 2.18 shows a scatter diagram of the periods of roll (horizontal axis) and pitch (vertical axis). The pitch period is equal to the rolling period along the upper red line. We have defined this line as the synchronizing line in Fig. 2.8. On the other hand, twice the pitching period is equal to the rolling period along the lower line.

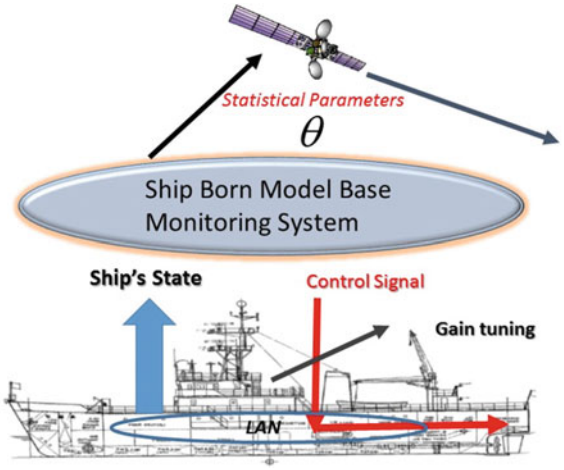

Fig. 2.16 Configuration of the ship-born model-based monitoring system

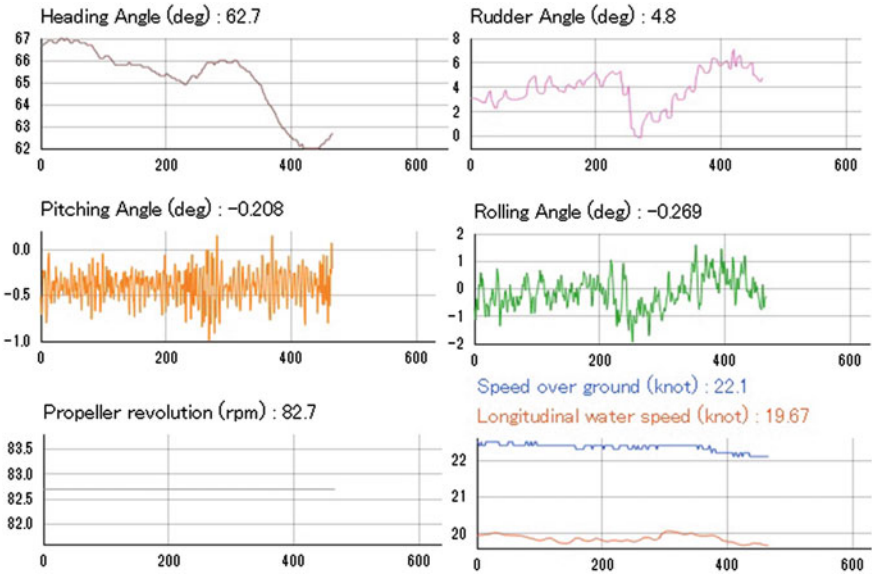

Fig. 2.17 Real-time monitoring of ship and main engine motions (heading angle indicates yaw motion)

When both of the periods are plotted along this line, the risk of large rolling motion increases. We have defined this line as the parametric rolling line in Fig. 2.8.

The lowest two plots in Fig. 2.18 show the changes in the spectra of pitch (left) and roll (right). As described in the previous section, the pitch motion spectrum is strongly affected by wave spectra. According to the pitch spectra shown in bottom left plot, the dominant peaks are located at approximately 18 s in the low-frequency domain. The ship received strong swell wave disturbances from the sea during navigation. However, we also detect other peaks in the higher-frequency domain. We can estimate the peaks generated by wind and waves around the ship.

Figure 2.19 shows the power contribution to the engine power from the engine power, the pitch, the yaw, and the rudder motion in a certain batch. The pie chart shows the contribution from each variable integrated over the frequencies from 0 to 0.5 Hz. The light blue zone shows the contribution from the engine power. Thus, if the light blue zone is larger than the other zones, the ship's main engine is not significantly affected by other motions such as rudder, pitch, and yaw motions. When the contribution of other motions, especially pitch motion, is large, the main engine receives strong effects caused by ship motion through the propeller shaft. Thus, the main engine should not fall into the light blue zone. The lower plots show the time histories of each variable in the latest batch.

Figure 2.20 shows the load diagram of the main engine. The plot shows the outputs of the main engine power (KW) versus the engine RPM. The baseline is the propeller

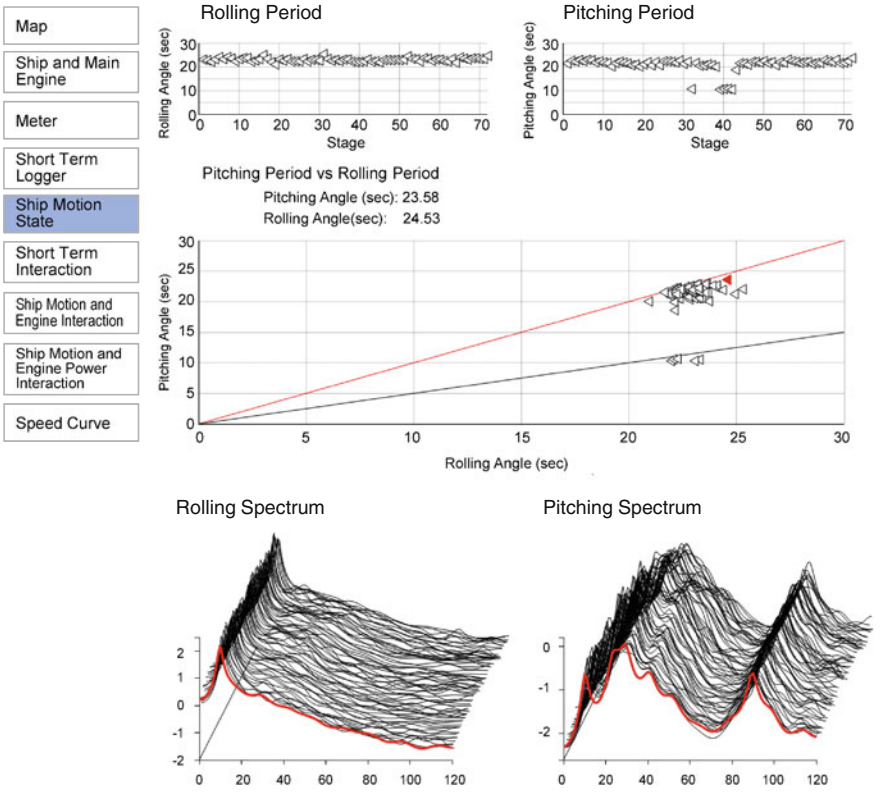

Fig. 2.18 Batch time monitoring of pitch and roll motions

RPM. When the plotted point lies outside of the limit line (red line), the ship's main engine falls into "torque rich" territory.

2.8 RBF-ARX Modeling for a Nonlinear System

Many systems in the real world are inherently nonlinear. In such cases, it is necessary to use a nonlinear model to represent the system behavior and design a controller. This section presents a statistical modeling method for nonlinear systems, based on the Redial Basis Function network-style coefficients AutoRegressive model with eXogenous variable (RBF-ARX) model (Peng et al. 2003, 2004, 2009). In this section, we derive the RBF-ARX model and its estimation method and present some illustrative examples. In Chap. 4, dynamics modeling of a ship using the RBF-ARX model, its application to tracking controller design, and the experiment results for a real ship are presented.

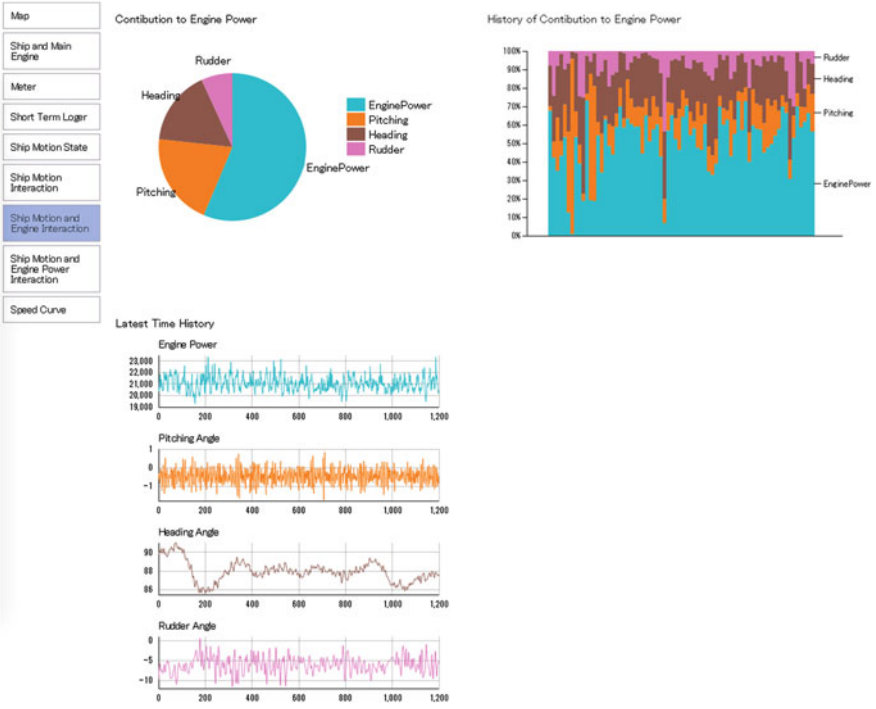

Fig. 2.19 Noise contribution to engine power of engine power, pitch, yaw (heading), and rudder motion

2.8.1 Introduction: Use of the RBF-ARX Model for Nonstationary Nonlinear Systems

There have been many significant advances in research and in the application of linear system modeling and control theory. However, if a system is strongly nonlinear and operates in a wider working range, we should use a nonlinear model to reasonably describe the system. Research on practical nonlinear system modeling and control theory has recently become the focus of attention. Several models for control have been built using statistical methodology for complicated nonlinear system modeling. These models are primarily completely nonlinear models, including various parametric and nonparametric models (such as the bilinear model, the Hammerstein model, the Volterra series model, and neural network models), and local linearization models (such as the state-dependent AR model and piecewise linear model set). Purely nonlinear model-based controller design usually requires solving a nonlinear optimization problem online, which is still not possible, particularly for shorter-sample-period cases.

Local linearization modeling and controller design approaches based on the framework of relatively matured linear system modeling and control theory have many

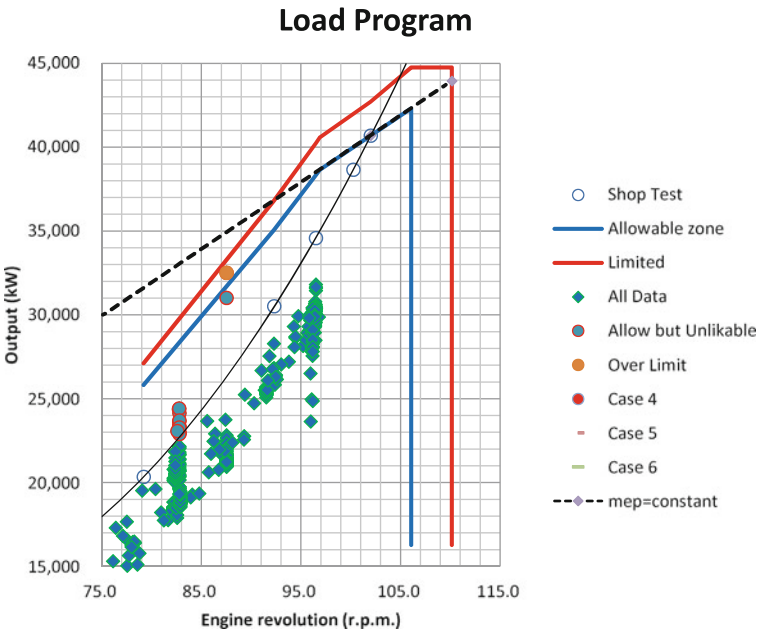

Fig. 2.20 Scattering diagram of the periods of roll and pitch

successful applications. However, some potential problems, such as on-line parameter estimation failure in approaches resorting to parameter estimation online and higher identification cost in the approaches using piecewise linear model set, limit the wider application of these approaches. This section presents a method by which to alleviate these difficulties by using a global modeling and optimization method for nonlinear real-time control, which does not require on-line parameter estimation.

In model-based real-time control strategies for nonlinear systems, radial basis function (RBF) networks offer a framework for the modeling of the controlled system, because of their simple topological structure, their precision in nonlinear dynamics approximation, and their fast learning. However, in many real applications, a large number of RBF centers are needed in order to obtain the required degree of precision, which leads to difficulties in parameter estimation. In modeling complex nonlinear dynamics, a frequently used class of models is the state-dependent autoregressive (AR) model with functional coefficients (Priestley 1980). Using a set of RBF networks to approximate the coefficients of a state-dependent AR model yields an RBF-AR model (Vesin 1993; Ozaki et al. 1999), which has the advantages of both the state-dependent AR models in the description of nonlinear dynamics and of RBF networks in function approximation.

A natural extension of the ideas behind RBF-AR modeling leads us to the RBF-ARX model (RBF-AR model with eXogenous variables) (Peng et al. 2003, 2004, 2009). In general, RBF-ARX models use far fewer RBF centers than a single RBF

network model. The RBF-ARX model is actually a hybrid pseudo-linear model constructed based on the Gaussian RBF networks and linear ARX model structure, which was proposed in order to characterize nonlinear systems having dynamic features that depend on time-varying working-points and which may be locally linearized at each working point. The RBF-ARX model is built as a global model and is estimated off-line so as to avoid the possible failure of on-line parameter estimation during real-time control based on the model.

2.8.2 RBF-ARX Modeling

2.8.2.1 State-Dependent ARX Model for Nonlinear Systems

Consider a discrete-time nonstationary and nonlinear single-input single-output (SISO) system, the dynamic features of which depend on time-varying working points and which may be locally linearized at each working-point. This system can be described by a nonlinear ARX model as follows:

$$y_n = \tilde{f}(y_{n-1}, \ldots, y_{n-n_y}, u_{n-1}, \ldots, u_{n-n_u}, v_{n-1}, \ldots, v_{n-n_v}) + \xi_n, \quad (2.78)$$

where $y_n \in \Re$ is the output, $u_n \in \Re$ is the input, v_n is a measurable disturbance, and ξ_n denotes a noise that is usually regarded as a Gaussian white noise, independent of the observations. Defining the vector

$$\boldsymbol{x}_{n-1} = \left(y_{n-1}, \ldots, y_{n-n_y}, u_{n-1}, \ldots, u_{n-n_u}, v_{n-1}, \ldots, v_{n-n_v}\right)^T, \quad (2.79)$$

model (2.78) may then be rewritten as

$$y_n = f(\boldsymbol{x}_{n-1}) + \xi_n. \quad (2.80)$$

Assuming that the function $f(\cdot)$ in (2.80) is $(j+1)$-times continuously differentiable at an arbitrary equilibrium point $\boldsymbol{x}_0$, then $f(\cdot)$ can be expanded in a Taylor series about $\boldsymbol{x}_0$, yielding

$$\begin{aligned} f(\boldsymbol{x}_{n-1}) = {} & f(\boldsymbol{x}_0) + f'(\boldsymbol{x}_0)^T(\boldsymbol{x}_{n-1} - \boldsymbol{x}_0) \\ & + \frac{1}{2}(\boldsymbol{x}_{n-1} - \boldsymbol{x}_0)^T f''(\boldsymbol{x}_0)(\boldsymbol{x}_{n-1} - \boldsymbol{x}_0) + \cdots + r_j(\boldsymbol{x}_{n-1}), \end{aligned} \quad (2.81)$$

where $f'(\boldsymbol{x}_0)$ and $f''(\boldsymbol{x}_0)$ are the first and second derivatives, respectively, of f with respect to $\boldsymbol{x}_0$, and $r_j(\boldsymbol{x}_{n-1})$ is (the Lagrangian form of) the remainder of the Taylor expansion of $f(\cdot)$ (see, e.g., Bronshtein and Semendyayev 1998). Substituting the above expression into model (2.80) yields

$$y_n = (\varphi_0 + \varphi_1(\boldsymbol{x}_{n-1})) + (\Phi_0 + \Phi_1(\boldsymbol{x}_{n-1}))\,\boldsymbol{x}_{n-1} + \xi_n \tag{2.82}$$

where

$$\begin{aligned}
\varphi_0 &= f(\boldsymbol{x}_0) - f'(\boldsymbol{x}_0)^T \boldsymbol{x}_0 + \frac{1}{2}\boldsymbol{x}_0^T f''(\boldsymbol{x}_0)\boldsymbol{x}_0 + \cdots \\
\varphi_1(\boldsymbol{x}_{n-1}) &= r_j(\boldsymbol{x}_{n-1}) \\
\Phi_0 &= f'(\boldsymbol{x}_0)^T - \frac{1}{2}\boldsymbol{x}_0^T f''(\boldsymbol{x}_0) - \frac{1}{2}\boldsymbol{x}_0^T f''(\boldsymbol{x}_0)^T + \cdots \\
\Phi_1(\boldsymbol{x}_{n-1}) &= \frac{1}{2}\boldsymbol{x}_{n-1}^T f''(\boldsymbol{x}_0) + \cdots,
\end{aligned} \tag{2.83}$$

The above equation can also be rewritten as follows:

$$\begin{aligned}
y_n = \pi_0(\boldsymbol{x}_{n-1}) &+ \sum_{i=1}^{n_y} \pi_i^y(\boldsymbol{x}_{n-1}) y_{n-i} + \sum_{i=1}^{n_u} \pi_i^u(\boldsymbol{x}_{n-1}) u_{n-i} \\
&+ \sum_{i=1}^{n_v} \pi_i^v(\boldsymbol{x}_{n-1}) v_{n-i} + \xi_n
\end{aligned} \tag{2.84}$$

where

$$\begin{aligned}
&\pi_0(\boldsymbol{x}_{n-1}) = \varphi_0 + \varphi_1(\boldsymbol{x}_{n-1}) \\
&\left[\pi_1^y(\boldsymbol{x}_{n-1}), \ldots, \pi_{n_y}^y(\boldsymbol{x}_{n-1}), \pi_1^u(\boldsymbol{x}_{n-1}), \ldots, \pi_{n_u}^u(\boldsymbol{x}_{n-1}), \pi_1^v(\boldsymbol{x}_{n-1}), \ldots, \pi_{n_v}^v(\boldsymbol{x}_{n-1})\right] \\
&\qquad = \Phi_0 + \Phi_1(\boldsymbol{x}_{n-1})
\end{aligned} \tag{2.85}$$

Model (2.84) is in fact a local linearization of the nonlinear ARX model (2.80). Model (2.84) has an autoregressive structure that is similar to that of a linear ARX model, and its state-dependent coefficients enable the behavior of the model to change with the system state. As such, this model may be regarded as a state-dependent ARX model or a functional-coefficient ARX model. Model (2.84) is also an extension of the state-dependent AR model derived in time series modeling (Priestley 1980; Chen and Tsay 1993). Model (2.84) can reasonably treat both nonstationarity and nonlinearity of the time series. Actually, Model (2.84) can deal with a non-stationary process by splitting the parameter space into a large number of small segments and regarding the process as locally stationary within each segment. On the other hand, model (2.84) treats a nonlinear process by splitting the state space into a large number of small segments and regarding the process as locally linear within each segment.

2.8.2.2 RBF-ARX Model

Although model (2.84) is actually a special version of the nonlinear ARX model (2.80), a problem that remains is how to specify the functional coefficients of model

(2.84). Specifying the state-dependent coefficients of model (2.84) may be considered to be a problem of function approximation from a multi-dimensional input space $\boldsymbol{x}$ to a one-dimensional scalar space $\omega_i = \pi_i(\boldsymbol{x})$. An efficient approach to solving the above problem, without loss of generality, is by the neural networks approximation. Note that RBF networks can be used to approximate the state-dependent coefficients of model (2.84), because they are excellent for approximating nonlinear scalar functions. Moreover, the locality of the basis functions makes RBF networks very suitable for learning local variations. If Gaussian RBF networks are used as approximations to the coefficients of model (2.84), the derived model is referred to as the RBF-ARX model and is given by

$$y_n = \phi_0(\boldsymbol{x}_{n-1}) + \sum_{i=1}^{n_y} \phi_{y,i}(\boldsymbol{x}_{n-1}) y_{n-i} + \sum_{i=1}^{n_u} \phi_{u,i}(\boldsymbol{x}_{n-1}) u_{n-i} + \sum_{i=1}^{n_v} \phi_{v,i}(\boldsymbol{x}_{n-1}) v_{n-i} + \xi_n, \tag{2.86}$$

where $\boldsymbol{z}_k^j = \left[z_{k,1}^j, z_{k,2}^j, \ldots, z_{k,n_x}^j\right]^T$ for $j = y, u, v$ and

$$\phi_0(\boldsymbol{x}_{n-1}) = c_0^0 + \sum_{k=1}^{m} c_k^0 \exp\left\{-\lambda_k^y ||\boldsymbol{x}_{n-1} - \boldsymbol{z}_k^y||_2^2\right\} \tag{2.87}$$

$$\phi_{y,i}(\boldsymbol{x}_{n-1}) = c_{i,0}^j + \sum_{k=1}^{m} c_{i,k}^j \exp\left\{-\lambda_k^j ||\boldsymbol{x}_{n-1} - \boldsymbol{z}_k^j||_2^2\right\}. \tag{2.88}$$

Here n_y, n_u, n_v, m, and $n_x = \dim(\boldsymbol{x}_{n-1})$ are the model orders, $\boldsymbol{z}_k^j (k = 1, \ldots, m;\ j = y, u, v)$ are the centers of RBF networks, $\lambda_k (k = 1, \ldots, m)$ are the scaling parameters, c_k^0 $(k = 0, 1, \ldots, m)$ and $c_{i,k}^j$ $(i = 1, 2, \ldots, n_j;\ j = y, u, v;\ k = 0, 1, \ldots, m)$ are the scalar weighting coefficients, and $||\cdot||_2$ denotes the L_2-norm.

In the general case, the RBF networks in model (2.86) may have different centers for different variables. However, in some applications, all the RBF networks may share the same centers, because model (2.86) possesses the autoregressive structure. Thus, although the RBF centers are the same in this case, the coefficients of the regression polynomials are different. In the RBF-ARX model (2.86), the signal $\boldsymbol{x}_{n-1}$ on which the time varying model coefficients depend may be the output signal, the input signal, or any other measured signal in the system to be considered.

The RBF-ARX model (2.86) with Gaussian RBF network-style coefficients has a basic structure that is similar to that of a linear ARX model. This model actually includes a general RBF network as one of its components and may therefore be regarded as a more general nonlinear model than the RBF neural network.

The RBF-ARX model is a rather general form of the working-point-dependent ARX model by adding the time-varying local mean (offset term) $\phi_0(\boldsymbol{x}_{n-1})$, which is needed in order to describe a non-stationary process in which the equilibrium (working) point varies with time. It is easy to see that the local linearization of model

(2.86) is a linear ARX model at each working point by fixing $\boldsymbol{x}_{n-1}$ at time $n-1$ in (2.86). It is natural and appealing to interpret model (2.86) as a locally linear ARX model in which the evolution of the process at time $n-1$ is governed by a set of AR coefficients $\{\phi_{y,i}, \phi_{u,i}, \phi_{v,i}\}$ and a local mean ϕ_0, all of which depend on the "working point" of the process at time $n-1$. This property is very useful and allows us to use a linear model-based control method to design a controller. On the other hand, this is not possible when we use RBF networks or other nonlinear models, such as the Hammerstein model (see, e.g., Ljung 1999). Model (2.86) may also be conveniently implemented in real-time control, because it avoids the need for on-line parameter estimation.

Because of the satisfactory properties of RBF networks in function approximation, as well as in learning local variation, the use of the working-point-dependent functional coefficients makes the RBF-ARX model capable of effectively representing the dynamic characteristics of the system at each working point. The RBF-ARX model incorporates the advantages of the state-dependent ARX model in nonlinear dynamics description and the RBF network in function approximation. In general, the model does not need many RBF centers compared with a single RBF network model, because the complexity of the model is dispersed into the lags of the autoregressive parts of the model. The SISO RBF-ARX model (2.86) can be extended to the general multi-input multi-output (MIMO) case (Peng et al. 2009) that is presented in Sect. 4.2, in which the RBF-net-type functional coefficients become matrices.

2.8.3 Identification of the RBF-ARX Model

Any kind of RBF and RBF-ARX model parameter estimation procedure must include the selection of appropriate centers and scaling factors for the RBF networks, and the estimation of all of the linear weights of the RBF networks in the model. There are primarily three types of method by which to estimate RBF-type model parameters. The first method estimates all parameters of the model, regardless of parameter features, by using a nonlinear parameter optimization algorithm such as the Levenberg-Marquardt method (LMM) (Marquardt 1963), which is generally based on an exhaustive search in the solution space and therefore requires extensive computation. The LMM is a commonly used method for approaching large problems (McLoone et al. 1998). Gorinevsky (1997) and Gorinevsky et al. (1997) presented some convincing results which were obtained using the Levenberg-Marquardt training algorithm for RBF networks.

The second method is to first select the basis function centers by selecting input vectors either algorithmically or at random and setting them to be the centers (Ozaki et al. 1999; Moody and Darken 1989; Chen et al. 1991). The linear weights may then be estimated by the standard linear least squares method (LSM). Obviously, although this method may provide a rough approximation, it cannot yield optimal parameters.

The third method, presented in this section, is an automatic estimation method that can optimize all of the model parameters simultaneously and accelerate the computa-

tional convergence of the optimization search process compared with the first method. This method, which is implemented off-line, is referred to as the structured nonlinear parameter optimization method (SNPOM) (Peng et al. 2003, 2004) for RBF-type model estimation, which combines the LMM for estimating nonlinear parameters and the LSM for linear weight estimation at each iteration. In the SNPOM, the linear weights are updated several times at each iteration during the process of looking for the search direction to update the nonlinear parameters. Therefore, the SNPOM is a completely structured hybrid algorithm, which can obtain a faster convergence rate and better modeling precision than McLoone's algorithm (McLoone and Irwin 1997) as is shown in Sect. 2.8.4. With the rapid development of computer technology, the speed of convergence and improved modeling accuracy that can be provided by the SNPOM, rather than the computational load, have become the foci of interest.

The RBF-ARX model (2.86) is constructed as a global model, and will be estimated off-line from observation data so as to avoid the potential problem caused by the failure of on-line parameter estimation during real-time control based on the model. The off-line identification procedure for the RBF or RBF-ARX model includes both order selection and estimation of all of the parameters. The order may be selected by comparing the Akaike information criterion (AIC) (Akaike 1974) values for different orders and by looking at the model dynamics. Therefore, we must first have a good model parameter estimation method, and then we can repeat the method for models with different orders, before finally selecting the best model.

2.8.3.1 Model Parameters Estimation

In the general case, the number of linear weights is larger than the number of nonlinear centers and scaling parameters in an RBF-ARX model. Therefore, applying a classical method, such as the Gauss-Newton method (GNM) or the Levenberg-Marquardt method (LMM) (Marquardt 1963), to estimate all parameters simultaneously regardless of their special features may take too much computational time and might not even provide a satisfactory level of modeling precision. In this subsection, an unconstrained structured nonlinear parameter optimization method (SNPOM) (Peng et al. 2003, 2004, 2009) for parameter estimation of RBF-based models is presented, which is a hybrid method that consists of using the LMM for nonlinear parameter estimation and the least squares method (LSM) for linear parameter estimation. Therefore, the SNPOM can greatly accelerate the computational convergence of the parameter optimization search process, especially for the RBF-ARX model with more linear weights and fewer nonlinear parameters. Note that the SNPOM is not a variable rotation method (VRM) (i.e., a method which rotationally fixes one set of variables in order to optimize another set of variables). The main idea behind the SNPOM is to divide the parameter search space into two subspaces (i.e., the linear weight subspace and the nonlinear parameters subspace). The search centers on the optimization in the nonlinear subspace. However, at each iteration in the optimization process, a search in the nonlinear (or linear) subspace is executed based on the estimated values just obtained in the linear (or nonlinear) subspace. The search in the nonlinear subspace uses a method similar to the LMM, and the search in the linear

subspace uses the LSM. The SNPOM for the RBF-ARX model (2.86) is implemented as follows.

Step 1: Parameter classification

For the RBF-ARX model (2.86), the linear and nonlinear parameter sets are respectively defined as follows:

$$\theta_L \equiv \left\{ c_i^0, c_{k,i}^j | i = 0, 1, \ldots, m; k = 1, 2, \ldots, n_j, j = y, u, v \right\} \tag{2.89}$$
$$\theta_N \equiv \left\{ \lambda_1, \ \ldots \ , \lambda_m, \ \boldsymbol{z}_1^T, \ \ldots \ , \boldsymbol{z}_m^T \right\}^T .$$

In general, we can rewrite models (2.86) for estimation purposes as

$$y_n = f(\theta_L, \theta_N, \Omega_{n-1}) + \xi_n, \tag{2.90}$$

or more specifically as

$$y_n = \Phi(\theta_N, \Omega_{n-1})^T \theta_L + \xi_n, \tag{2.91}$$

where $\Omega_{n-1} = \left[\boldsymbol{y}_{n-1}^T, \boldsymbol{u}_{n-1}^T, \boldsymbol{v}_{n-1}^T, \boldsymbol{x}_{n-1}^T \right]^T$, $\boldsymbol{y}_{n-1} = \left[y_{n-1}, y_{n-2}, \ldots, y_{n-n_y} \right]^T$, $\boldsymbol{u}_{n-1} = \left[u_{n-1}, u_{n-2}, \ldots, u_{n-n_u} \right]^T$ and $\boldsymbol{v}_{n-1} = \left[v_{n-1}, v_{n-2}, \ldots, v_{n-n_v} \right]^T$. Note that Eq. (2.91) is the regression form of model (2.90), which is linear with respect to the linear parameter vector.

Step 2: Initialization

For RBF-ARX model (2.86), the orders are n_y, n_u, n_v, m, and $n_x = \dim(\boldsymbol{x}_{n-1})$. In this step, these orders must first be chosen. The best approach for choosing a suitable order is presented in Sect. 2.8.3.2. Then, a subset $\boldsymbol{z}_k^0$ $(k = 1, 2, \ldots, m)$ in the vector space of $\boldsymbol{x}_{n-1}$ is chosen randomly as an initial value for the RBF network centers. The following formula is then used to compute the initial values of the scaling factors in the model:

$$\lambda_k^0 = -\log \varepsilon_k / \max_{n-1} \left\{ ||\boldsymbol{x}_{n-1} - \boldsymbol{z}_k^0||_2^2 \right\}, \quad \varepsilon_k \in [0.1 \sim 0.0001] \tag{2.92}$$

These factors will ensure that the linear weights are bounded when the signal $\boldsymbol{x}_{n-1}$ moves far away from the centers. After selecting an initial nonlinear parameter vector θ_N^0, and keeping it fixed, use the LSM to compute initial linear weights θ_L^0 as follows:

$$\theta_L^0 = R(\theta_N^0)^+ \bar{\boldsymbol{Y}} \tag{2.93}$$

where

$$
\begin{aligned}
R(\theta_N^0)^+ &= \left[R(\theta_N^0)^T R(\theta_N^0)\right]^{-1} R(\theta_N^0)^T \\
R(\theta_N^0) &= \left[\Phi(\theta_N^0, \bar{\Omega}_\tau), \Phi(\theta_N^0, \bar{\Omega}_{\tau+1}), \ldots \Phi(\theta_N^0, \bar{\theta}_{M-1})\right] \\
\bar{Y} &= (\bar{y}_{\tau+1}, \bar{y}_{\tau+2}, \ldots, \bar{y}_M)^T
\end{aligned} \tag{2.94}
$$

and $\{\bar{y}_i, \bar{\Omega}_{i-1} | i = \tau + 1, \ldots, M\}$ is the measured dataset, τ is the largest time lag of the variables in Ω_{n-1}, M is the data length, and $R(\theta_N^0)^+$ is the pseudo-inverse of $R(\theta_N^0)$, calculated using singular value decomposition (SVD) (Golub and Van Loan 1996) for overcoming ill-conditioned problems, which will improve the robustness of the numerical computation.

Step 3: Estimation of θ_N and θ_L

Suppose that we take the following quadratic objective function:

$$
V(\theta_N, \theta_L) \equiv \frac{1}{2} ||F(\theta_N, \theta_L)||_2^2, \tag{2.95}
$$

where

$$
F(\theta_N, \theta_L) \equiv \begin{bmatrix} \hat{y}_{\tau+1|\tau} - \bar{y}_{\tau+1} \\ \hat{y}_{\tau+2|\tau+1} - \bar{y}_{\tau+2} \\ \vdots \\ \hat{y}_{M|M-1} - \bar{y}_M \end{bmatrix}, \tag{2.96}
$$

$$
\hat{y}_{n+1|n} = f(\theta_L, \theta_N, \bar{\Omega}_n), \quad n = \tau, \tau + 1, \ldots, M - 1. \tag{2.97}
$$

Then, the parameters θ_N and θ_L are obtained as the minimizer of the optimization problem

$$
(\hat{\theta}_N, \hat{\theta}_L) = \arg \min_{\theta_N, \theta_L} V(\theta_N, \theta_L). \tag{2.98}
$$

Step 4: Updating the parameters to be estimated

Two (major and minor) iteration processes are used to update all of the parameters to be estimated. The iteration step is denoted by $k (= 1, 2, \ldots, k_{\max})$. For the nonlinear parameter vector θ_N, the updating formula is

$$
\theta_N^{k+1} = \theta_N^k + \beta_k \boldsymbol{d}_k, \tag{2.99}
$$

where $\boldsymbol{d}_k$ is the search direction, and β_k is a scalar step length parameter that gives the distance to the next point, which is determined by a line search procedure after determining the search direction $\boldsymbol{d}_k$ at a minor iteration. In order to increase the robustness of the search process, based on the LMM, the search direction $\boldsymbol{d}_k$ in (2.99) is obtained as the solution of the set of linear equations

$$
\left[J(\theta_N^k)^T J(\theta_N^k) + \gamma_k I\right] \boldsymbol{d}_k = -J(\theta_N^k)^T F(\theta_N^k, \theta_L^k), \tag{2.100}
$$

where

$$J(\theta_N^k) = \left(\partial F(\theta_N^k, \theta_L^k)/\partial \theta_N^k\right)^T, \tag{2.101}$$

and the scalar γ_k controls both the magnitude and the direction of $\boldsymbol{d}_k$. When γ_k tends toward zero, $\boldsymbol{d}_k$ will tend toward the Gauss-Newton direction. As γ_k tends toward infinity, $\boldsymbol{d}_k$ tends toward the steepest descent direction. The magnitude of γ_k is determined at each major iteration using a method similar to that of the "lsqnonlin" function in the Matlab Optimization Toolbox (Coleman et al. 1999).

Following the determination of γ_k, (2.100) is solved in order to obtain a search direction $\boldsymbol{d}_k$. A step length of unity β_k in (2.99) is obtained by a line search procedure similar to the mixed quadratic and cubic polynomial interpolation and extrapolation method given in Coleman et al. (1999).

The optimization calculation centers on the search for θ_N^{k+1} in each iteration using (2.99), followed by the immediate update of the linear weights θ_L^{k+1} using the LSM, as follows:

$$\theta_L^{k+1} = R(\theta_N^{k+1})^+ \bar{Y}, \tag{2.102}$$

where

$$R(\theta_N^{k+1})^+ = \left[R(\theta_N^{k+1})^T R(\theta_N^{k+1})\right]^{-1} R(\theta_N^{k+1})^T \tag{2.103}$$

$$R(\theta_N^{k+1}) = \left[\Phi(\theta_N^{k+1}, \bar{\Omega}_\tau), \Phi(\theta_N^{k+1}, \bar{\Omega}_{\tau+1}), \ldots, \Phi(\theta_N^{k+1}, \bar{\Omega}_{M-1})\right]^T \tag{2.104}$$

$$\bar{Y} = (\bar{y}_{\tau+1}, \bar{y}_{\tau+2}, \ldots, \bar{y}_M)^T, \tag{2.105}$$

and the pseudo-inverse $R(\theta_N^{k+1})^+$ of $R(\theta_N^{k+1})$ is also evaluated using SVD (Golub and Van Loan 1996). The line search procedure for determining the step length β_k in (2.99) ensures that

$$V(\theta_N^{k+1}, \theta_L^{k+1}) < V(\theta_N^k, \theta_L^k) \tag{2.106}$$

holds at each major iteration with respect to the parameters θ_N^{k+1} and θ_L^{k+1} updated by (2.99) and (2.102). Hence, θ_N^{k+1} and θ_L^{k+1} are the parameter choices for decreasing the objective function (2.95) at the $(k+1)$th iteration.

Remark 2.1 In the SNPOM described above, the global optimum of the linear weights θ_L may easily be obtained using (2.102), which adjusts the search direction and the step length to ensure that the objective function decreases in all parameters, rather than only in the nonlinear part θ_N at each iteration. Note that if θ_N^{k+1} is only changed based on the fixed θ_L^{k+1} during the process of looking for the search direction and the step length to update θ_N^{k+1} at the $(k+1)$th iteration, the objective function that is used is then $V(\theta_N^{k+1}, \theta_L^k)$ during the $(k+1)$th iteration, which is only affected by θ_N^{k+1}, and not by θ_L^{k+1}. Thus, the searched θ_N^{k+1} is not better, because it did not consider the effect of θ_L^{k+1}. As a result, the convergence rate in this case would be

slow compared to the SNPOM, where any change in θ_N^{k+1} will also change θ_L^{k+1}. In the SNPOM, since the objective function V is affected by θ_N^{k+1} and θ_L^{k+1}, simultaneously at any time, the searched θ_N^{k+1} and θ_L^{k+1}, based on the V including "full information", should be better. This increases the convergence rate of the SNPOM. In terms of computing efficiency, the SNPOM is much better than general methods of optimizing all parameters regardless of parameter type, especially for the case in which there are more linear parameters than nonlinear parameters in a model.

Remark 2.2 It may be beneficial to use formula (2.92) to re-compute the scaling factor λ_k at any time after updating the RBF center θ_N^k during the search process, in order to avoid divergence of the linear weights. This is a heuristic approach for determining λ_k, but does not greatly affect the optimization effectiveness, because the behavior of RBF-ARX model (2.86) is decided primarily by its RBF centers, model orders, and linear weights.

2.8.3.2 Determination of the Order of the Model

An appropriate order for the identified model (2.90) may be determined by the Akaike information criterion (AIC) (Akaike 1974). The procedure involves repeating the above SNPOM for model (2.90) for different orders and choosing the final model by looking for the smallest AIC value, together with appropriate model dynamics.

For RBF-ARX model (2.86), the AIC is defined as follows:

$$\text{AIC} = M \log \hat{\sigma}_e^2 + 2(s+1), \quad \text{for } M \gg p, \tag{2.107}$$

where $\hat{\sigma}_e^2$ is the residual variance of the model under the chosen orders, p is the largest order of the regression part, and s is the total number of parameters to be estimated.

2.8.4 Illustrative Examples

In order to illustrate the effectiveness of the RBF-ARX modeling, here we use the Mackey-Glass equation (2.108) (a well-known chaotic benchmark time series)

$$\dot{x}(t) = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \tag{2.108}$$

to generate a set of data, and use an RBF-AR model and an RBF network to fit the data. The performance of the RBF-AR(p, m, l) model (2.109)

$$y_n = \phi_0(\boldsymbol{x}_{n-1}) + \sum_{i=1}^{p} \phi_{y,i}(\boldsymbol{x}_{n-1}) y_{n-i} + \xi_n, \tag{2.109}$$

where $\boldsymbol{x}_{n-1} = \left[y_{n-1}, y_{n-2}, \ldots, y_{n-l}\right]^T$, $\boldsymbol{z}_k = \left[z_{k,1}, z_{k,2}, \ldots, z_{k,l}\right]^T$,

$$\phi_0(\boldsymbol{x}_{n-1}) = c_0^0 + \sum_{k=1}^{m} c_k^0 \exp\left\{-\lambda_k ||\boldsymbol{x}_{n-1} - \boldsymbol{z}_k||_2^2\right\} \tag{2.110}$$

$$\phi_{y,i}(\boldsymbol{x}_{n-1}) = c_{i,0}^y + \sum_{k=1}^{m} c_{i,k}^y \exp\left\{-\lambda_k ||\boldsymbol{x}_{n-1} - \boldsymbol{z}_k||_2^2\right\} \tag{2.111}$$

and RBF(p, l) network

$$y_n = \theta_0 + \sum_{k=1}^{p} \theta_k \exp\left\{-\lambda_k ||\boldsymbol{x}_{n-1} - \boldsymbol{z}_k||_2^2\right\} + \xi_n \tag{2.112}$$

are compared. We also compare several optimization methods. In the Mackey-Glass equation (2.108), the selected equation parameters $a = 0.2$, $b = 0.1$, $c = 10$, and $\tau = 20$ will be used.

The original Mackey-Glass series is shown in Fig. 2.21, in which the first 500 data points are used to train the model, and the last 500 data points are used to test the model. Figure 2.22 shows the convergence of the prediction error variance for the RBF-ARX(5,3,2) and RBF(5,5) models during the parameter search iteration using the SNPOM (as presented in this section), Levenberg-Marquardt method (LMM, Marquardt 1963), Gauss-Newton method (GNM) (see, e.g., Coleman et al. 1999), and the trust region method (TRM) (see, e.g., Coleman and Li 1996). It is clear that the SNPOM has the fastest convergence rate, and the predictive error variance of the estimated model using the SNPOM is also smaller than that of other methods.

For the RBF-network/RBF-AR-model and the training data/test data, respectively, Table 2.3 lists the results obtained using the SNPOM and the evolutionary programming algorithm (EPA) (Shi et al. 1999), which is a mutual estimation procedure based on evolutionary programming and the standard linear least squares method, where the AIC values are computed using Eq. (2.107). Table 2.3 shows that in all cases, the estimation performance of the SNPOM is better than that of EPA, especially for

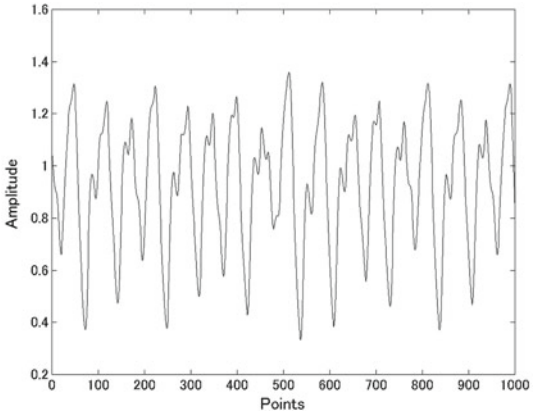

Fig. 2.21 A chaotic series generated from the Mackey-Glass equation

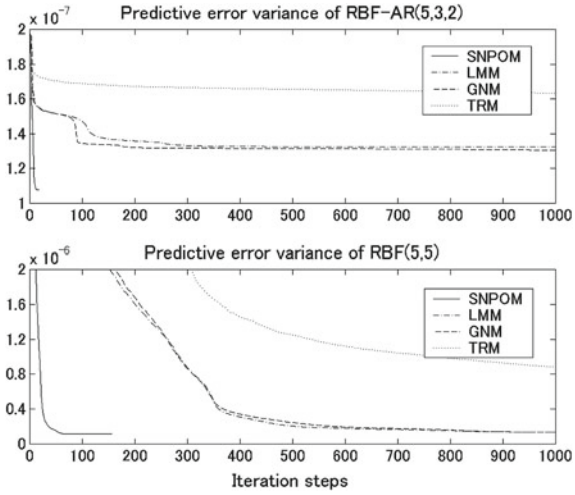

Fig. 2.22 Results for various optimization methods

RBF networks. Table 2.3 also shows that an RBF-AR model with fewer model parameters may attain a better fitting precision than an RBF network. Figure 2.23 shows the characteristic roots of the RBF-AR(5,3,2) model estimated by the SNPOM for the Mackey-Glass series, which shows the complicated dynamics of the time series. In Chap. 4, the RBF-ARX modeling result for a ship's dynamic behavior will be presented for implementing tracking control of a ship.

Table 2.3 Performance of the RBF-AR(p, m, l) model and the RBF(p, l) network estimated using the proposed SNPOM and the evolutionary programming algorithm (EPA) (Shi et al. 1999) for the Mackey-Glass series

Models	Number of centers	Total number of unknown parameters	Training set		Testing set	
			Predictive error variance	AIC	Predictive error variance	AIC
RBF(5,5)[a]	25	31	2.92×10^{-4}	−4007.4	3.50×10^{-4}	−3916.8
RBF(5,5)[b]	25	31	1.16×10^{-7}	−7921.5	1.38×10^{-7}	−7835.7
RBF(20,5)[a]	100	121	4.32×10^{-5}	−4782.8	1.02×10^{-4}	−4353.3
RBF(20,5)[b]	100	121	9.46×10^{-8}	−7848.6	1.16×10^{-7}	−7739.2
RBF-AR(5,3,2)[a]	6	30	1.23×10^{-7}	−7895.5	1.31×10^{-7}	−7864.0
RBF-AR(5,3,2)[b]	6	30	1.08×10^{-7}	−7960.0	1.26×10^{-7}	−7880.3

[a] Estimated by the EPA
[b] Estimated by the SNPOM

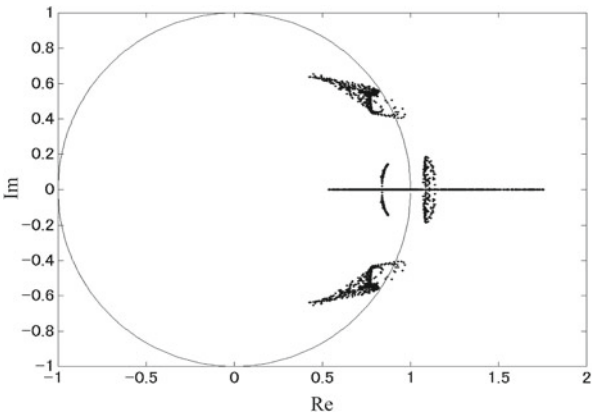

Fig. 2.23 Time-varying eigenvalues of the estimated RBF-AR(5,3,2) model (2.3.22) for the Mackey-Glass series modeling

References

Akaike, H.: On the use of a linear model for the identification of feedback systems. Ann. Inst. Stat. Math. **20**, 425–439 (1968)

Akaike, H.: Autoregressive model fitting for control. Ann. Inst. Stat. Math. **23**, 163–180 (1971)

Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **AC-19**, 716–723 (1974)

Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Parzen, E., Tanabe, K., Kitagawa, G. (eds.) Selected Papers of Hirotugu Akaike, pp. 199–213. Springer, New York (1998)

Akaike, H., Nakagawa, T.: Statistical Analysis and control of Dynamic System. Kluwer, Tokyo (1988)

Akaike, H., Nakagawa, T.: Statistical Analysis and control of Dynamic System. KTK Scientific Publishers, Tokyo (1989)

Akaike, H., Kitagawa, G., Arahata, E., Tada, F.: TIMSAC-78, The Inst. Stat. Math. (1979)

Anderson, B.D.O., Moore, J.B.: Optimal Filtering. Premtics Hall, New Jersey (1979)

Anderson, B.D.O., Moore, J.B.: Optimal Filtering. Courier Dover Publications, Mineola (2012)

Bronshtein, I.N., Semendyayev, K.A.: Handbook of Mathematics. Springer, Berlin (1998)

Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans. Neural Netw. **2**, 302–309 (1991)

Chen, S., Tsay, R.S.: Functional-coefficient autoregressive models. J. Am. Stat. Assoc. **88**, 298–308 (1993)

Coleman, T., Branch, M.A., Grace, A.: Optimization Toolbox User's Guide. The MathWorks Inc., Natick (1999)

Coleman, T.F., Li, Y.: An interior trust region approach for nonlinear minimization subject to bounds. SIAM J. Optim. **6**, 418–445 (1996)

Doucet, A., De Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Springer, New York (2001)

Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)

Gorinevsky, D.: An approach to parametric nonlinear least square optimization and application to task-level learning control. IEEE Trans. Autom. Control **42–7**, 912–927 (1997)

Gorinevsky, D., Torfs, D.E., Goldenberg, A.A.: Learning approximation of feedforward control dependence on the task parameters with application to direct-drive manipulator tracking. IEEE Trans. Robot. Autom. **13–4**, 567–581 (1997)

Kalman, R.E.: A new approach to linear filtering and prediction problems. Trans. Am. Soc. Mech. Eng. J. Basic Eng. **82**, 35–45 (1960)
Kitagawa, G.: Non-Gaussian state-space modeling of nonstationary time series. J. Am. Stat. Assoc. **82–400**, 1032–1041 (1987)
Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. **5**(1), 1–25 (1996)
Kitagawa, G.: Introduction to Time Series Modeling. CRC press (2010)
Kitagawa, G., Akaike, H.: (1978): A procedure for the modeling of nonstationary time series. Ann. Inst. Stat. Math. **30–B**, 215–363 (1978)
Kitagawa, G., Gersch, W.: Smoothness Priors Analysis of Time Series, vol. 116. Springer, New York (1996)
Konishi, S., Kitagawa, G.: Information Criteria and Statistical Modeling. Springer, New York (2008)
Ljung, L.: System Identification: Theory for the User, 2nd edn. Prentice Hall PTR, New Jersey (1999)
Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. SIAM J. Appl. Math. **11**, 431–441 (1963)
McLoone, S., Irwin, G.: Fast parallel off-line training of multilayer perceptrons. IEEE Trans. Neural Netw. **8**, 646–653 (1997)
McLoone, S., Brown, M.D., Irwin, G., Lightbody, G.: A hybrid linear/nonlinear training algorithm for feedforward neural networks. IEEE Trans. Neural Netw. **9**(4), 669–684 (1998)
Moody, J., Darken, C.: Fast learning in networks of locally-tuned processing units. Neural Comput. **1**, 281–294 (1989)
Ohtsu, K.: Statistical analysis and control of ship. In: 8th IFAC Symposium of MCMC, Brazil (2009)
Ohtsu, K.: Model based on monitoring system and the optimal control (in Japanese), Kaibundou (2012)
Ozaki, T., Sosa, P.V., Haggan-Ozaki, V.: Reconstructing the nonlinear dynamics of epilepsy data using nonlinear time series analysis. J Signal Process. **3–3**, 153–162 (1999)
Ozaki, T., Tong, H.: On the fitting of non-stationary autoregressive models in time series analysis. In: Proceedings of 8th Hawaii International Conference on System Science, Western Periodical Company, pp. 224–226 (1975)
Ozaki, T., Tong, H.: On the fitting of non-stationary autoregressive models in time series analysis. In: Proceedings of 8th Hawaii International Conference on System Science, Western Periodical Company, pp. 224–226 (1979)
Peng, H., Ozaki, T., Haggan-Ozaki, V., Toyoda, Y.: A parameter optimization method for the radial basis function type models. IEEE Trans. Neural Netw. **14**, 432–438 (2003)
Peng, H., Ozaki, T., Toyoda, Y., Shioya, H., Nakano, K., Haggan-Ozaki, V., Mori, M.: RBF-ARX model based nonlinear system modeling and predictive control with application to a NOx decomposition process. Control Eng. Pract. **12**, 191–203 (2004)
Peng, H., Wu, J., Inoussa, G., Deng, Q., Nakano, K.: Nonlinear system modeling and predictive control using RBF nets-based quasi-linear ARX model. Control Eng. Pract. **17**, 59–66 (2009)
Priestley, M.B.: State dependent models: a general approach to nonlinear time series analysis. J. Time Ser. Anal. **1**, 57–71 (1980)
Sakamoto, Y., Ishiguro, M., Kitagawa, G.: Akaike Information Criterion Statistics. D. Reidel, Dordrecht (1986)
Shi, Z., Tamura, Y., Ozaki, T.: Nonlinear time series modelling with the radial basis function-based state-dependent autoregressive model. Int. J. Syst. Sci. **30**, 717–727 (1999)
Takanami, T., Kitagawa, G.: Estimation of the arrivaltimes of seismic waves by multivariate time series model. Ann. Inst. Statist. Math. **43**, 407–433 (1991)
Tanokura, Y., Kitagawa, G.: Power contribution analysis for multivariate time series with correlated noise sources. Adv. Appl. Stat. **4**, 65–95 (2004)

Tanokura, Y., Kitagawa, G.: Modeling influential correlated noise sources in multivariate dynamic systems. In: Modelling and Simulation: Fifteenth IASTED International Conference Proceedings (2004)

Tanokura, Y., Tsuda, H., Sato, S., Kitagawa, G.: Constructing a credit default swap index and detecting the impact of the financial crisis. In: Bell, W.R., Holan, S.H., McElroy, T.S. (eds.) Economic Time Series: Modeling and Seasonality, pp. 359–380. CRC Press, Boca Raton (2012)

Vesin, J.: An amplitude-dependent autoregressive signal model based on a radial basis function expansion. In: Proceedings of the International Conference ASSP 3 Minnesota, pp. 129–132 (1993)

Whittle, P.: On the fitting of multivariable autoregressions and the approximate canonical factorization of a spectral density matrix. Biometrika **50**, 129–134 (1963)

Chapter 3
Design of a Model-Based Autopilot System for Course Keeping Motion

Abstract A critical problem in applying the optimal control theory to a large-scale complex system that is subject to large disturbances, such as a ship, an electric power plant, or a large chemical plant, is that it is sometimes difficult to obtain a precise state-space model from the theory of the subject area. For such systems, Akaike, H.: Ann. Inst. Stat. Math. **23**, 163–180 (1971) proposed the ARX model for the identification of the controlled system. In this chapter, we propose a method for obtaining optimal control laws based on the linear stationary state-space model of the controlled system for two performance criteria. Based on this linear quadratic optimal controller, we develop three types of optimal controllers for a ship: an AR model-based autopilot, a roll reduction control system, and an engine governor control system.

Keywords ARX model based optimal control · Ship course keeping · Autopilot · Rudder roll control · Ship main engine governor

3.1 Statistical Optimal Controller Based on the ARX Model

In this section, we consider a standard optimal control problem which will be used in designing the autopilot systems described in Sect. 3.2. A distinct feature of the statistical control approach is that the controlled system model required in the formulation of the optimal control problem is obtained by the state-space representation of the ARX model. The same type of stochastic model was proposed by Åström and Källström (1976). We then derive optimal control laws for two types of performance criteria.

3.1.1 Statistical Optimal Control Problem

Assume that a control system is described by the linear model

$$\boldsymbol{x}_n = \Phi \boldsymbol{x}_{n-1} + \Gamma \boldsymbol{r}_{n-1} + \Psi \boldsymbol{v}_n, \tag{3.1}$$

K. Ohtsu et al., *Time Series Modeling for Analysis and Control*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55303-8_3

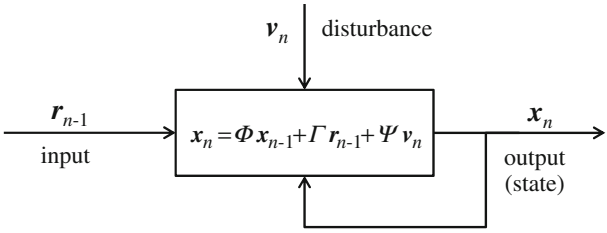

Fig. 3.1 Linear model of the controlled system

where $\boldsymbol{x}_n$ denotes a k-dimensional state vector, $\boldsymbol{r}_n$ is a q-dimensional control input, and $\boldsymbol{v}_n$ is an s-dimensional disturbance to the controlled system (Fig. 3.1).

The standard linear quadratic regulator (LQR) problem for a discrete-time-controlled system (3.1) is to determine the control input sequence $\boldsymbol{r}_0, \ldots, \boldsymbol{r}_{L-1}$ which minimizes the quadratic performance criterion:

$$J_L(\boldsymbol{x}_0) = \sum_{n=1}^{L} \left\{ \boldsymbol{x}_n^T Q \boldsymbol{x}_n + \boldsymbol{r}_{n-1}^T R \boldsymbol{r}_{n-1} \right\}, \tag{3.2}$$

where $\boldsymbol{x}_0$ is a given initial state, L is the time interval used for evaluating the performance of the controlled system, Q is a $k \times k$ positive semi-definite matrix, and R is a $q \times q$ positive definite matrix.

A crucial problem in applying the optimal control theory to a large-scale complex system with large disturbances, such as a ship, an electric power plant, or a large chemical plant, is that it is sometimes difficult to obtain a precise state-space model from the theory of the subject area. For such systems, Akaike (1971) proposed the use of the ARX model for the identification of the system. Many successful applications of this method, which is usually referred to as the statistical controller, can be found in the literature (Otomo et al. 1972; Ohtsu et al. 1979a, b; Nakamura and Akaike 1981; Akaike and Nakagawa 1988; Akaike and Kitagawa 1999).

Assume an autoregressive exogenous (ARX) model:

$$\boldsymbol{y}_n = \sum_{j=1}^{m} A_j \boldsymbol{y}_{n-j} + \sum_{j=1}^{m} B_j \boldsymbol{r}_{n-j} + \boldsymbol{v}_n \,, \tag{3.3}$$

where $\boldsymbol{y}_n$ and $\boldsymbol{r}_n$ are a p-dimensional output variable and a q-dimensional input variable, A_j is a $p \times p$ matrix, B_j is a $p \times q$ matrix, and $\boldsymbol{v}_n$ is a p-dimensional white noise. Then, as shown in the previous chapter, the ARX model (3.3) has a state-space representation (Akaike 1971; Akaike and Nakagawa 1988):

$$\begin{cases} \boldsymbol{x}_n = \Phi \boldsymbol{x}_{n-1} + \Gamma \boldsymbol{r}_{n-1} + \Psi \boldsymbol{v}_n \\ \boldsymbol{y}_n = H \boldsymbol{x}_n \,, \end{cases} \tag{3.4}$$

3.1.2 Optimal Control Law

The optimal controller is an advanced control method, which deals with the problem of finding a control law for a given system such that a certain performance criterion is minimized. The performance criterion is usually defined as a cost functional that is a function of state and control inputs. Optimal control is the sequence of the control inputs that minimize the performance criterion.

Here, we present a method for designing an optimal control law that is obtained as the solution to the linear quadratic optimal control problem:

$$\min_{r_0,\ldots,r_{L-1}} J_L(\boldsymbol{x}_0), \tag{3.5}$$

with respect to the state-space model (3.4). As a performance criterion, we use the following two criteria.

[Type 1] A quadratic performance criterion evaluating the expected quadratic loss of a state variable $\boldsymbol{x}_n$ and a control variable $\boldsymbol{r}_{n-1}$,

$$J_L(\boldsymbol{x}_0) = \mathrm{E}\left[\sum_{n=1}^{L}\left\{\boldsymbol{x}_n^T Q\boldsymbol{x}_n + \boldsymbol{r}_{n-1}^T R\boldsymbol{r}_{n-1}\right\}\right]. \tag{3.6}$$

[Type 2] A performance criterion that also penalizes the difference of the control input:

$$J_L(\boldsymbol{x}_0) = \mathrm{E}\left[\sum_{n=1}^{L}\left\{\boldsymbol{x}_n^T Q\boldsymbol{x}_n + \boldsymbol{r}_{n-1}^T R\boldsymbol{r}_{n-1} + \Delta\boldsymbol{r}_{n-1}^T T\Delta\boldsymbol{r}_{n-1}\right\}\right], \tag{3.7}$$

where $\Delta\boldsymbol{r}_{n-1} = \boldsymbol{r}_{n-1} - \boldsymbol{r}_{n-2}$, Q is a $pm \times pm$ positive semi-definite matrix, R is a $q \times q$ positive definite matrix, and T is an $q \times q$ positive semi-definite matrix. In designing an autopilot system, it is possible to reduce the mechanical loss of the actuator by using the type-2 performance criterion.

3.1.2.1 Optimal Control Law for the Type-1 Criterion

We consider the optimal control input when we adopted the type-1 performance criterion. In order to solve the optimal control problem by dynamic programming (Bellman 2003), we define $f_n(\boldsymbol{x})$, $(n = 1, \ldots, N)$ by

$$f_{n+1}(\boldsymbol{x}) = \min_{\substack{\boldsymbol{r}_n,\ldots,\boldsymbol{r}_{L-1}\\ \boldsymbol{x}_n=\boldsymbol{x}}} E\left[\sum_{i=n+1}^{L}\left\{\boldsymbol{x}_i^T Q\boldsymbol{x}_i + \boldsymbol{r}_{i-1}^T R\boldsymbol{r}_{i-1}\right\}\right]. \tag{3.8}$$

Assume that this $f_{n+1}(\boldsymbol{x})$ can be expressed in simple quadratic form:

$$f_{n+1}(\boldsymbol{x}) = \min_{\substack{\boldsymbol{r}_n \\ \boldsymbol{x}_n = \boldsymbol{x}}} E\Big[\boldsymbol{x}_{n+1}^T\, S_n\, \boldsymbol{x}_{n+1} + \boldsymbol{r}_n^T R \boldsymbol{r}_n\Big], \tag{3.9}$$

for some properly defined matrix S_n. Then, from (3.4), it holds that

$$\begin{aligned}
&E\Big[\boldsymbol{x}_{n+1}^T S_n \boldsymbol{x}_{n+1} + \boldsymbol{r}_n^T R \boldsymbol{r}_n\Big] \\
&= E\Big[(\Phi \boldsymbol{x}_n + \Gamma \boldsymbol{r}_n + \Psi \boldsymbol{v}_{n+1})^T S_n (\Phi \boldsymbol{x}_n + \Gamma \boldsymbol{r}_n + \Psi \boldsymbol{v}_{n+1}) + \boldsymbol{r}_n^T R \boldsymbol{r}_n\Big] \\
&= \boldsymbol{r}_n^T (\Gamma^T S_n \Gamma + R) \boldsymbol{r}_n + \boldsymbol{x}_n^T \Phi^T S_n \Gamma \boldsymbol{r}_n + \boldsymbol{r}_n^T \Gamma^T S_n \Phi \boldsymbol{x}_n \\
&\quad + \boldsymbol{x}_n^T \Phi^T S_n \Phi \boldsymbol{x}_n + E\Big[\boldsymbol{v}_{n+1}^T \Psi^T S_n \Psi \boldsymbol{v}_{n+1}\Big] \\
&= \Big\{\boldsymbol{r}_n + (\Gamma^T S_n \Gamma + R)^{-1} \Gamma^T S_n \Phi \boldsymbol{x}_n\Big\})^T (\Gamma^T S_n \Gamma + R) \\
&\quad \Big\{\boldsymbol{r}_n + (\Gamma^T S_n \Gamma + R)^{-1} \Gamma^T S_n \Phi \boldsymbol{x}_n\Big\} + \boldsymbol{x}_n^T \Phi^T S_n \Phi \boldsymbol{x}_n \\
&\quad - \boldsymbol{x}_n^T \Phi^T S_n^T \Gamma (\Gamma^T S_n \Gamma + R)^{-1} \Gamma^T S_n \Phi \boldsymbol{x}_n + E\Big[\boldsymbol{v}_{n+1}^T \Psi^T S_n \Psi \boldsymbol{v}_{n+1}\Big].
\end{aligned} \tag{3.10}$$

Since the first term on the right-hand side of the above equation is non-negative and the other terms are independent of the control input $\boldsymbol{r}_n$, the control input that minimizes this quantity is obtained by

$$\boldsymbol{r}_n = -(\Gamma^T S_n \Gamma + R)^{-1} \Gamma^T S_n \Phi \boldsymbol{x}_n. \tag{3.11}$$

Furthermore, for this optimal input, the first term vanishes and $f_n(\boldsymbol{x})$ is reduced to

$$\begin{aligned}
f_{n+1}(\boldsymbol{x_n}) = \boldsymbol{x}_n^T \Phi^T S_n \Phi \boldsymbol{x}_n - \boldsymbol{x}_n^T \Phi^T S_n^T \Gamma (\Gamma^T S_n \Gamma + R)^{-1} \Gamma^T S_n \Phi \boldsymbol{x}_n \\
+ \text{constant}.
\end{aligned} \tag{3.12}$$

Then, by the principle of optimality of dynamic programming (Bellman 2003), $f_n(\boldsymbol{x})$ satisfies the following relation:

$$f_n(\boldsymbol{x}) = \min_{\substack{y_{n-1} \\ \boldsymbol{x}_{n-1} = \boldsymbol{x}}} E\left[\boldsymbol{x}_n^T Q \boldsymbol{x}_n + \boldsymbol{r}_{n-1}^T R \boldsymbol{r}_{n-1} + f_{n+1}(\boldsymbol{x}_n)\right]. \tag{3.13}$$

Here, since $f_{n+1}(\boldsymbol{x})$ is given by (3.12), $f_n(\boldsymbol{x})$ can be expressed as

$$f_n(\boldsymbol{x}) = \min_{\substack{y_{n-1} \\ \boldsymbol{x}_{n-1} = \boldsymbol{x}}} E\Big[\boldsymbol{x}_n^T\, S_{n-1}\, \boldsymbol{x}_n + \boldsymbol{r}_{n-1}^T R \boldsymbol{r}_{n-1}\Big], \tag{3.14}$$

where S_n is given by

$$S_{n-1} = Q + \Phi^T \left\{ S_n - S_n^T \Gamma (\Gamma^T S_n \Gamma + R)^{-1} \Gamma^T S_n \right\} \Phi, \tag{3.15}$$

with $S_L = Q$.

The optimal control input $\boldsymbol{r}_n^*(n = 1, \ldots L-1)$ is given by multiplying the feedback gain G_n by the state $\boldsymbol{x}_n$

$$\boldsymbol{r}_n^* = G_n \boldsymbol{x}_n, \tag{3.16}$$

where the control gain G_n is given by

$$G_n = -(\Gamma^T S_{n+1} \Gamma + R)^{-1} \Gamma^T S_n \Phi. \tag{3.17}$$

Remark The optimal control input given in Eq. (3.16) is seemingly the standard feedback control and is obtained by the linear transformation of the current state vector $\boldsymbol{x}_n$. However, since the state vector contains a type of predictor of the future variable $\tilde{y}_{n+k|n-1}$ and the performance criterion evaluates the current and future states, the state vector can be considered to be a predictive controller.

3.1.2.2 Optimal Control Law for the Type-2 Criterion

Similarly, the optimal control input for the type-2 performance criterion is determined by considering a generalized version of $f_n(\boldsymbol{x})$, $(n = 1, \ldots, N)$,

$$f_n(\boldsymbol{x}) = \min_{\substack{\boldsymbol{r}_{n-1}, \ldots, \boldsymbol{r}_{L-1} \\ \boldsymbol{x}_{n-1} = \boldsymbol{x}}} E\left[\sum_{i=n}^{L} \left\{ [\boldsymbol{x}_i^T, \boldsymbol{r}_{i-1}^T] \begin{bmatrix} S & P \\ P^T & R \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{r}_{i-1} \end{bmatrix} + \Delta \boldsymbol{r}_{i-1}^T T \Delta \boldsymbol{r}_{i-1} \right\} \right]. \tag{3.18}$$

For details of the derivation of the optimal control law for the criterion, refer to Ohtsu et al. (1976a, b) and Ohtsu (2012).

The algorithm for obtaining the optimal control gains for the type-2 criterion is summarized as follows:

1. Set the initial values, $S_0 = Q$, $R_0 = R$, and $P_0 = 0$.
2. Compute S_i, P_i, and R_i, recursively for $i = 1, 2, \ldots, L$ by

$$\begin{aligned} S_i &= S_{i-1} + \Phi^T \left\{ S_{i-1} - (S_{i-1}^T \Gamma + P_{i-1})(\Gamma^T S_{i-1} \Gamma + P_{i-1}^T \Gamma + \Gamma^T P_{i-1} \right. \\ &\qquad \left. + R_{i-1} + T)^{-1} (\Gamma^T S_{i-1} + P_{i-1}^T) \right\} \Phi \\ P_i &= P + \Phi^T (S_{i-1} \Gamma + P_{i-1})(\Gamma^T S_{i-1} \Gamma + \Gamma^T P_{i-1} + P_{i-1}^T \Gamma + R_{i-1} + T)^{-1} T \\ R_i &= T + R - T^T (\Gamma^T S_{i-1} \Gamma + \Gamma^T P_{i-1} + P_{i-1} \Gamma^T + R_{i-1} + T)^{-1} T \,. \end{aligned} \tag{3.19}$$

The optimal control input $\boldsymbol{r}_i$ is then given by the feedback control law

$$\boldsymbol{r}_i = G_i \boldsymbol{x}_i + F_i \boldsymbol{r}_{i-1} , \tag{3.20}$$

where the optimal control gains G_i and F_i are obtained by

$$\begin{aligned} G_i &= -(\Gamma^T S_{i-1}\Gamma + P_{i-1}^T\Gamma + \Gamma^T P_{i-1} + R_{i-1} + T)^{-1}(\Gamma^T S_{i-1} + P_{i-1}^T)\Phi \\ F_i &= (\Gamma^T S_{i-1}\Gamma + P_{i-1}^T\Gamma + \Gamma^T P_{i-1} + R_{i-1} + T)^{-1} T . \end{aligned} \tag{3.21}$$

Here, if the evaluation period L is taken to be long enough, then G_i and F_i converge to some G and F, respectively. Therefore, the optimum control law for a stationary state is determined as

$$\boldsymbol{r}_i = G\boldsymbol{x}_i + F\boldsymbol{r}_{i-1} . \tag{3.22}$$

In actual autopilot system design, it will assuredly be reasonable and practical to use this control law with constant gains.

Remark This algorithm, which is based on dynamic programming, is not stable and may diverge due to the computational error at each time step if the evaluation period L is taken to be too large. Therefore, it is recommended that L be taken as not so large, e.g., $L = 30$. However, this problem can be mitigated by using the square root algorithm (Kitagawa 1983).

3.2 AR Model-Based Autopilot System

3.2.1 Autopilot System for Ships

The autopilot of a ship is a typical example of a classical control system that was designed based on the concept of feedback control. Figure 3.2 shows a schematic diagram of an autopilot system, where $\varphi_{0,n}$ denotes the desired yaw deviation at time n in the course commanded by a navigator. On the other hand, φ_n denotes the actual

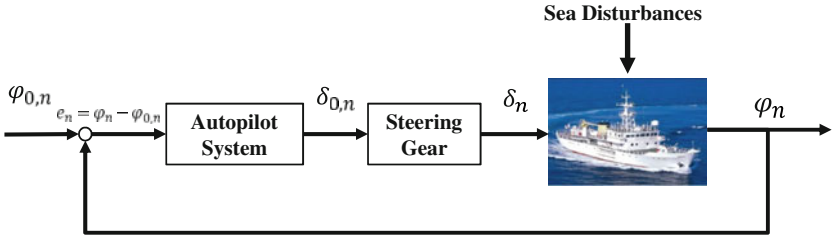

Fig. 3.2 Configuration of the autopilot system of a ship

yaw angle at time n observed under the influence of strong external disturbances, such as wind, current, or waves. Thus, the difference, $e_n = \varphi_n - \varphi_{0,n}$, is the yaw deviation from the desired course, and this angle is hereinafter referred to as the yaw deviation, or simply the yaw. The yaw, e_n, is an input to the controller (autopilot system), and the controller calculates an appropriate rudder angle $\delta_{0,n}$, which is then used as an input to the actuator (steering gear) to compensate for the yaw deviation. The actuator generates the actual rudder angle δ_n and applies a rudder force to the hull of the ship.

By formulating the autopilot system as an optimal control problem, it is possible to obtain the optimal feedback gain for the rudder to reduce the yaw deviation with a smaller rudder angle with respect to the type-1 or type-2 performance criterion given in Sect. 3.1.2.

3.2.2 Design of the ARX Model-Based Autopilot System

Here, we present a five-step procedure for designing a statistical optimal controller based on the time series of the controlled system.

(1) Acquisition of Time Series under Random Steering

The design of an AR autopilot system starts with the acquisition of time series data from an actual ship sailing at sea. During this experiment to obtain time series data, the following two points must be taken into account. First, if a control input is determined by a linear function of the output variable x_n, it is impossible to identify the input-output dynamics using the obtained data. Therefore, it is important that the control input contains some "noise" during the actual sea experiment that cannot be expressed as a linear feedback of the output. Second, it is desirable that the input signal, namely the rudder motion, contains frequency components that are as broad as possible. In order to satisfy these requirements, we recommend the method of steering according to the commanded rudder angle rather than randomly in manual mode. Figure 3.3 shows an example of using random steering to identify the AR model for actual ship data observed on Shioji-Maru III.

(2) Selection of Variables

Figure 3.4 shows a record of actual ship motions observed on a middle-sized container ship on the Pacific Ocean. The sea state was rough, and the observed wind scale was 9 (strong gale) to 10 (storm). The sampling rate Δt was 1 s. The time series of pitch, roll, yaw, sway acceleration ($\boldsymbol{r}_{acc}$), and rudder motions are shown from top to bottom. In designing an autopilot system, the selection of variables actually used for modeling is a crucial problem. It is clear that the rudder angle is the input variable, and yaw deviation is the output variable. However, it is possible to consider other variables as input or output variables.

For this selection stage, important information might be obtained from the power contribution analysis of the autopilot system, as discussed in Sect. 2.4.2, in which the importance of the roll-yaw-rudder effect was emphasized. Furthermore, if we

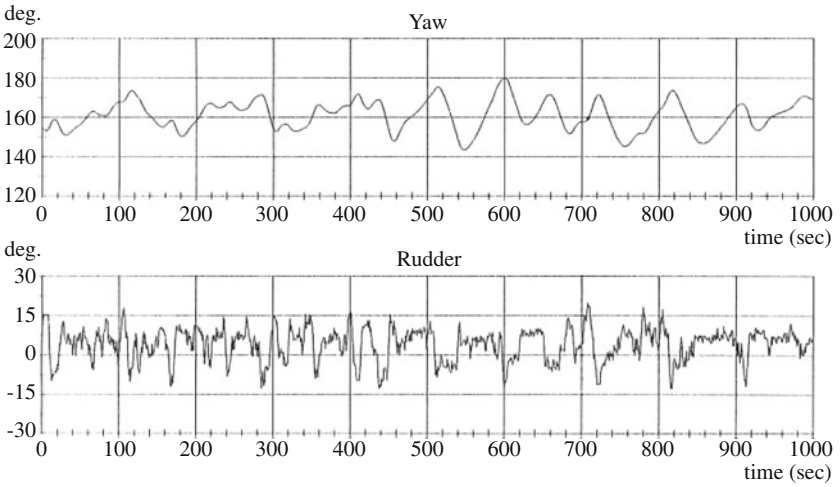

Fig. 3.3 "Random steering" data for Shioji-Maru III

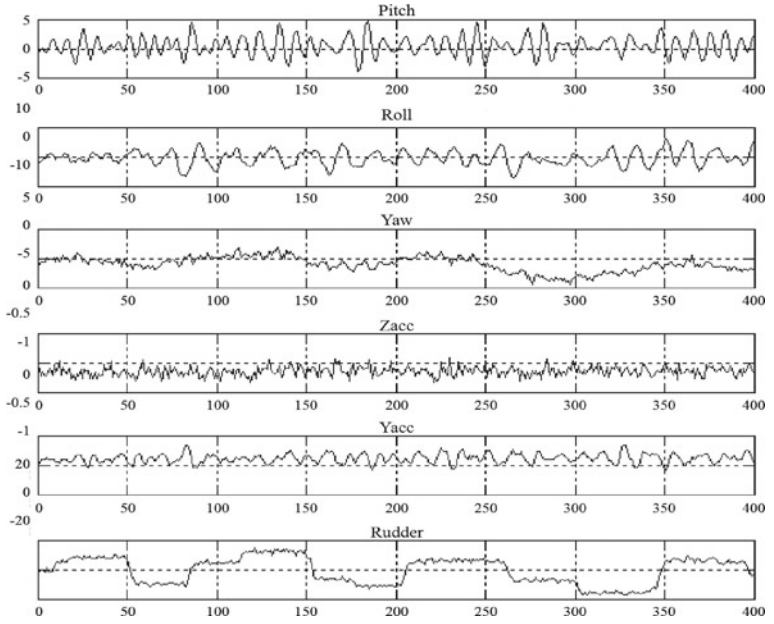

Fig. 3.4 Example of the observed motion of a ship steered by a conventional autopilot system

attempt to avoid rapid changes in yaw and yaw magnitude, it is necessary to consider the yaw rate signal, as well as the yaw. If we need to reduce the lateral drifting due to the rudder motion, it might be important to select the lateral sway acceleration ($\boldsymbol{r}_{acc.}$). However, we should not select the lateral sway acceleration when it includes harmful noise.

(3) White Noise Simulation Study of an Optimal Controller

In order to select suitable variables and weighting matrices Q, R, and T in the performance criterion for the selected variables of Eqs. (3.6) and (3.7), white noise simulation is useful. The following state-space model is used in the white noise simulation:

$$\begin{cases} \boldsymbol{x}_n = \Phi \boldsymbol{x}_{n-1} + \Gamma \boldsymbol{r}^*_{n-1} + W_n \\ \boldsymbol{r}^*_n = G\boldsymbol{x}_n + F\boldsymbol{r}^*_{n-1}. \end{cases} \tag{3.23}$$

In this model, the variance covariance matrix of the residual of the fitted model is used as the white noise W_n. Moreover, G and F denote the optimal feedback gains determined by Eq. (3.21). See Kitagawa and Ohtsu (1976) and Ohtsu et al. (1979b) for a semi-automatic procedure for obtaining appropriate weighting values in the Q, R, and T matrices in Eq. (3.7). Table 3.1 shows the results of the simulation study.

Figure 3.5 shows an example of the white noise simulation for $Q = \text{diag}(2,7,35,1)$, $T = 3$, and $R = 5$. Figure 3.6 shows the spectra of rudder, roll, and yaw motions. Repeating these simulations numerous times, we obtained the following results:

1. The AR model effectively fitted to the actual data represents the actual steering dynamics of the ship,
2. Compared with manual control, optimal control of the rudder is effective for significantly reducing both the yaw and roll, but is not so effective for reducing the pitch or $\boldsymbol{r}_{acc}$ (lateral acceleration),
3. The weight T that penalizes the change of the rudder motion is effective for making the rudder motion smoother.

Based on these results, the effect of rudder steering on roll motion is especially important, because the roll may increase or decrease depending on the steering law, i.e., on the selection of the Q and T weights.

(4) AR Autopilot

Next, we discuss the results of AR autopilot experiments performed onboard Shioji-Maru II and Shioji-Maru III. The principal dimensions of the two ships and photographs are presented in Chap. 1. The course deviation, i.e., the yaw, is selected as the output, and rudder is selected as the input.

Table 3.1 Results of the simulation study

No.	Q	R	Pitch	Roll	Yaw	Y_{acc}	Rudder	Diff.
(1)	(2,7,35,1)	0.850	1.726	4.575	1.709	0.0063	18.72	4.546
(2)	(3.6,7,40,1.67)	1.015	1.444	4.915	1.674	0.0062	19.12	5.052
(3)	(0,6.33,57.8,0)	0.860	2.242	4.473	1.672	0.0068	18.66	4.931
(4)	Manual		2.771	6.557	7.781	0.0081	17.62	0.832

Nos. (1) through (3): optimal controllers with various weighting matrices; No. (4): manual control

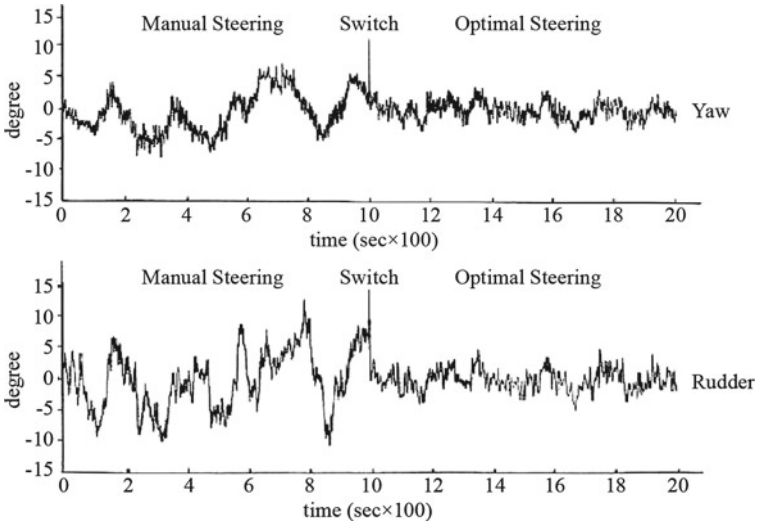

Fig. 3.5 Example of the white noise simulation for $Q = \text{diag}(2,7,35,1)$, $T = 3$, and $R = 5$

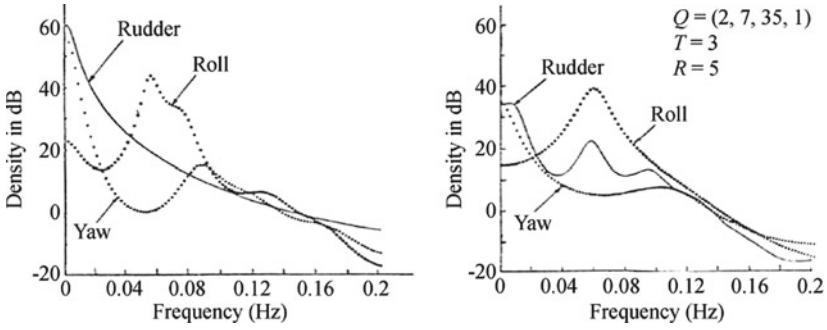

Fig. 3.6 Spectra of rudder, yaw, and roll motions in the white noise simulations (*left* conventional autopilot, *right* optimal control)

Figure 3.7 shows the analog record of the actual data collected on Shioji-Maru II. The plots on the left show the results for the AR autopilot, and the plots on the right show the results for conventional PID autopilot steering, which was carried out soon after the AR autopilot steering.

As shown in the figure, the AR autopilot can reduce the yaw motion and make the rudder motion smoother. Furthermore, the roll motion is also reduced by the AR autopilot steering.

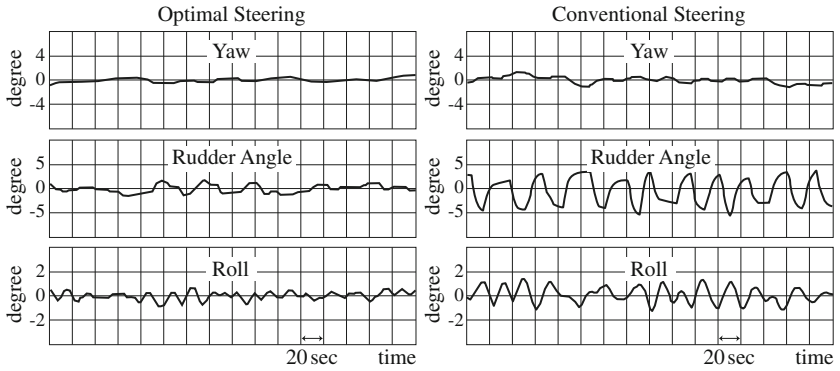

Fig. 3.7 Analog records of data collected onboard Shioji-Maru II (*left* AR autopilot steering, *right* conventional PID type autopilot)

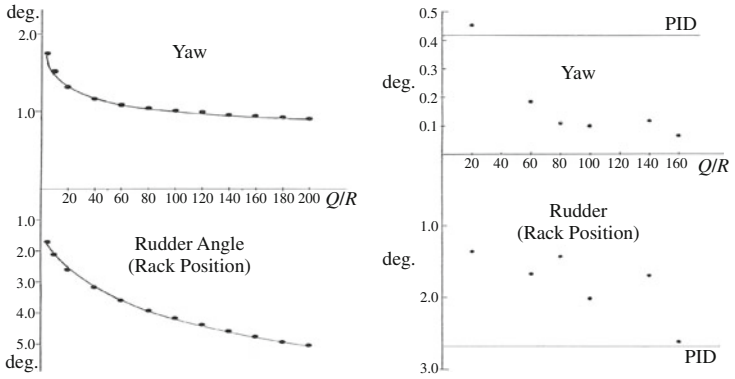

Fig. 3.8 Relationship between the gain ratio Q/R and the standard error of the heading deviation and rudder angle. *Left* simulation study, *right* actual sea test on Shioji-Maru III

(5) Performance of the AR Autopilot

The performances of the AR autopilot were considered from various points of view based on actual experiments onboard Shioji-Maru III. The plot on the left in Fig. 3.8 shows the standard deviations of yaw and rudder angle in the white noise simulations as the ratio Q/R is changed from 5 to 200. When the ratio is increased, the standard deviation of the yaw decreases, but this reduction becomes marginal for Q/R larger than 100. The plot on the right in Fig. 3.8 shows the corresponding results for the actual experiments onboard Shioji-Maru III. Similar tendencies appear in the simulation results. In this figure, 'PID' indicates the conventional ship autopilot (Ogata 1990). Even in the case of $Q/R = 160$, for which the standard deviation of yaw is less than 1/4 of the PID steering, the amount of rudder motion is approximately the same as that for the PID autopilot.

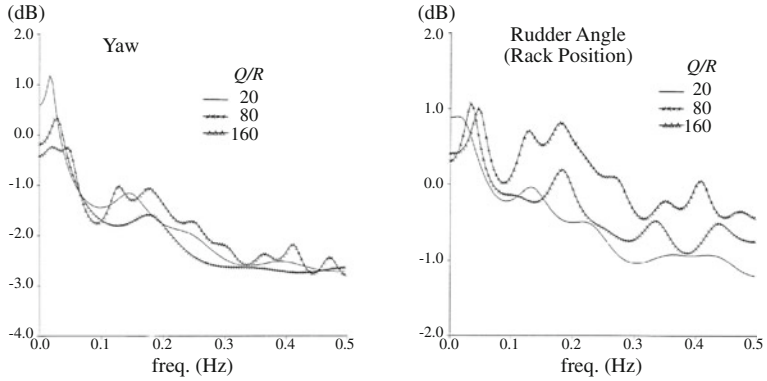

Fig. 3.9 Power spectra of AR autopilot steering with $Q/R = 20, 80$, and 180 (*left* yaw, i.e., heading deviation, *right* rudder angle)

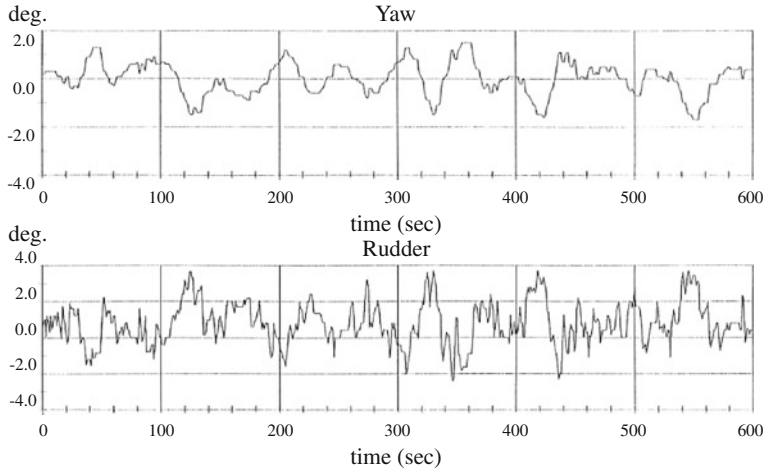

Fig. 3.10 Time history of the yaw and rudder motions for Q/R = 160

Figure 3.9 shows the spectra of the yaw and rudder motions. According to these spectra, when Q/R increases, the peak frequency of the rudder shifts to a higher frequency domain. In the power spectrum of the rudder angle shown in the right-hand plot, a strong spectral peak appears around $f = 0.15$–0.2 Hz as Q/R increases. The spectral peak of the yaw decreases, but its peak frequency shifts higher. This means that the yaw is reduced but fluctuates more frequently. Figure 3.10 shows the time histories of the AR autopilot with $Q/R = 160$.

During the series of experiments, the sea disturbances come from various directions. The three plots in Fig. 3.11 show the changes of the standard deviations of roll rate, yaw, and rudder angle in the AR autopilot steering for various wave encounter directions. The yaw and rudder angles become larger when the wave

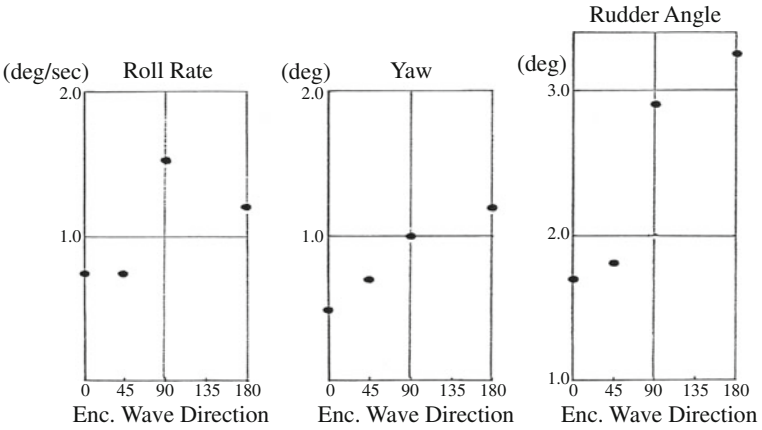

Fig. 3.11 Relationship between the encounter angle of the wave and the standard deviations of roll rate, yaw, and rudder angle (*horizontal axis* encounter angle, *vertical axis* standard deviation)

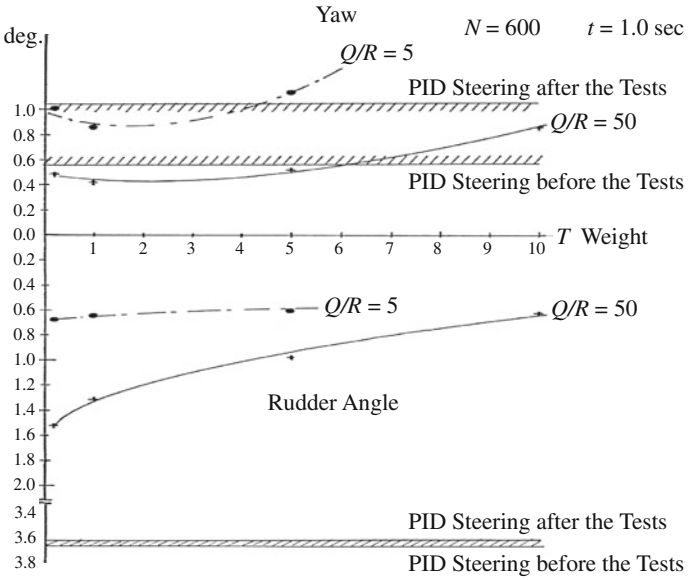

Fig. 3.12 Effect of weight T

direction changes to follow sea (180° from the ship's head), whereas the roll rate becomes large in the beam sea (±90° from the ship's head).

Figure 3.12 shows the effect of weight T of the type-2 performance criterion on the standard deviations of the yaw and rudder motions. In these actual experiments, Q/R was fixed to a constant value and only T was changed. This figure indicates that in the case of the AR autopilot steering with $Q/R = 50$, the standard deviations of the yaw are approximately the same as those the conventional autopilot steering,

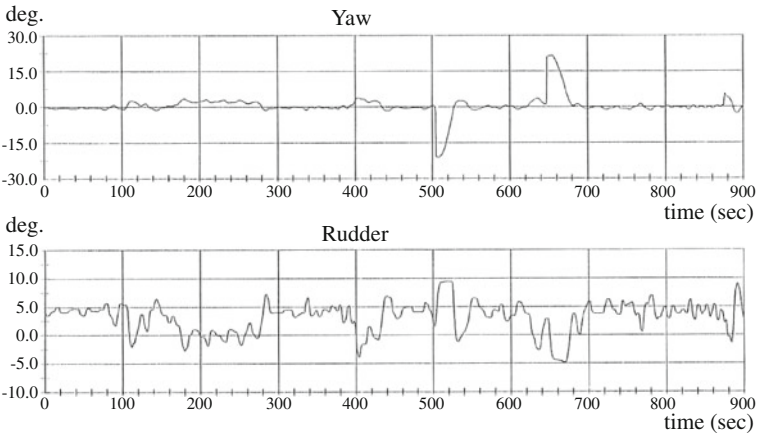

Fig. 3.13 Course-changing ability of the AR autopilot. Note that the ordered course is always taken as 0°)

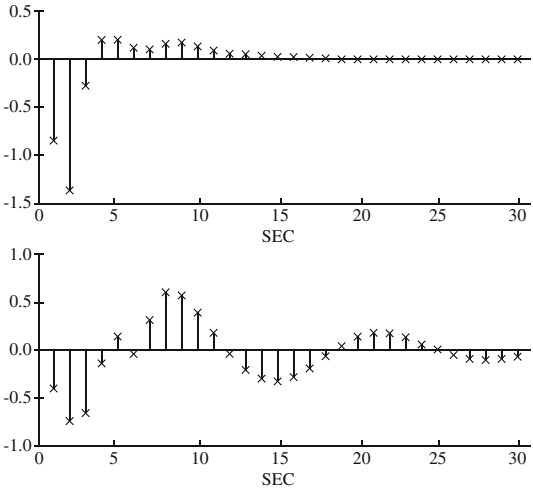

Fig. 3.14 Impulse response functions of the rudder to the impulsive change of yaw. *Top* AR autopilot, *bottom* conventional PID autopilot

whereas the standard deviation of the rudder motion became 1/2–1/6 of that of the conventional PID controller.

Figure 3.13 shows the course changing ability of the AR autopilot system. At approximately 500 s, the ship's course changes by 20 ° from the original course and then returns to the original course at approximately 640 s. This figure shows that the overshoot during this maneuvering is extremely small and that this autopilot system, which was designed for course-keeping control, also provides reasonable course-changing control.

Figure 3.14 shows the difference in properties between the AR autopilot and the conventional PID autopilot. Namely, the top and bottom plots show the impulse

response function of the rudder motion to the impulsive change of yaw for the AR autopilot and the conventional autopilot, respectively. The AR autopilot has a quick and sharp response compared with the PID autopilot, and the overshoot is significantly smaller for the AR autopilot.

3.3 Rudder-Roll Control System

As discussed in Sect. 2.4, the rudder motion has a strong influence on both yaw and roll. Figure 3.15 shows the absolute power contributions to the roll motion by the roll, rudder, and yaw motions for a large container ship. This figure also indicates that the rudder motion has a strong influence on roll motion at its significant peak frequency. Figure 3.16 shows a physical explanation of the rudder-roll effect. In this figure, G_{rudder} is the center of gravity of the rudder control surface and G_{ship} is the center of gravity of the hull body. As can be understood from the locations of these two points, as lever h between these two points becomes longer, the roll moment generated by the rudder moment becomes stronger.

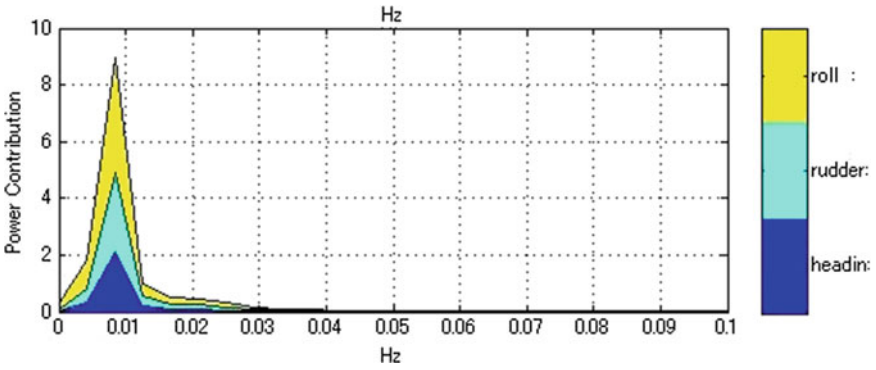

Fig. 3.15 Power contributions to roll from roll, rudder, and yaw (heading) motions

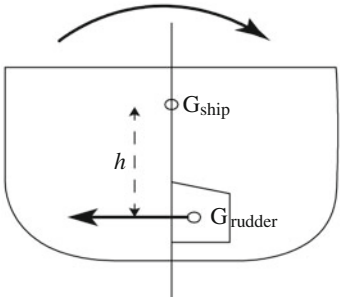

Fig. 3.16 Rudder effect on roll motion

In this section, considering this effect, we design a rudder-roll controller to reduce both the roll motion and the yaw motion (Oda et al. 1991). In this rudder-roll control system (RRCS), we consider the rudder command as the input variable and both yaw and roll as the output variables. Therefore, in the performance criterion, both the variances of yaw and roll are penalized with properly determined weighting coefficients.

However, inherent difficulties might arise in the above feedback control system. Moreover, it is conceivable that too frequent rudder motions are used to suppress the roll motion. According to the white noise simulations described in the previous section, the following three countermeasures may suppress such frequent rudder motions:

1. Select the roll rate signal rather than the roll angle as the state variable.
2. Adopt the type-2 performance criterion with weight T that penalizes the changes of the rudder angle described in Sect. 3.1.2.
3. Speed up of the movement of rudder steering actuator.

The following are examples of the third type of countermeasure. Figure 3.17 and Table 3.2 show the data obtained for the conventional PID autopilot and rudder-roll control (RRCS) systems when the conventional steering actuator and the improved

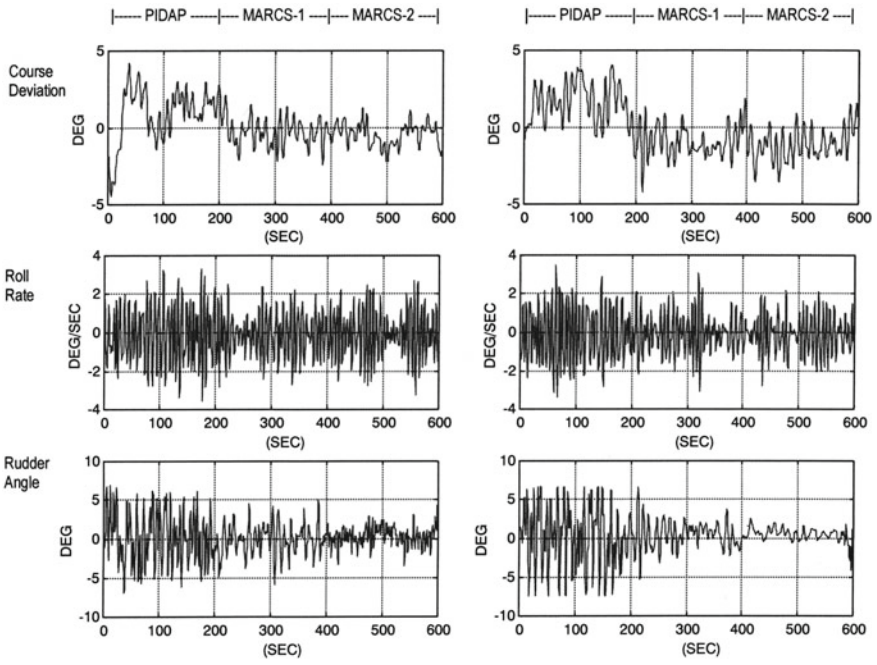

Fig. 3.17 Course deviation, roll rate, and rudder angle obtained using the conventional PID autopilot and the rudder-roll control system (MARCS). *Left* Conventional steering actuator, *right* EHS steering actuator (Oda et al. 2008)

Table 3.2 Results of the full-scale experiments on the rudder-roll controller

Test no. (platform)	Control mode	Yaw (variance)	Roll rate (variance)	Yaw reduction (%)	Roll rate reduction (%)
Ju32 (conventional)	PID	3.178	2.525	–	–
	MARCS-1	0.886	1.291	72.13	48.87
	MARCS-2	0.548	1.535	82.77	39.20
Ju35 (DDVC)	PID	1.907	2.219	–	–
	MARCS-1	1.101	1.063	42.29	52.07
	MARCS-2	1.167	0.945	38.84	57.41

(Modified from Table 4 of Oda et al. 2008)

direct drive volume type electronic hydraulic system (EHC) were used. In the experiment, the ship was steered by the conventional PID autopilot for the first 200 s, after which steering was performed by RRCS (MARCS), followed by RRCS with different parameters at 400 s.

The table and figure indicate that on both platforms, the mean squared course deviation by RRCS was reduced by 39–83 % and the variance of the roll rate was suppressed by 39–57 % as compared to control by the PID autopilot. Furthermore, Fig. 3.17 indicates that this reduction of yaw and roll rate was achieved with a significantly smaller (say 50–10 %) rudder angle.

Figure 3.18 shows the results of an experiment to determine the effect of roll reduction in the RRCS. In this figure, "Rudder neutral" indicates the results after impulsively induced rolling by a fin stabilizer, after which the rudder is set to the neutral position. On the other hand, "RRCS" indicates the results for the case in which RRCS control is performed after the impulsively induced roll motion. The roll motion is drastically decreased by RRCS.

Figure 3.19 shows the relative power contributions of roll rate motion from the roll rate, rudder angle, and yaw. In this figure, the left-hand plot shows the results for

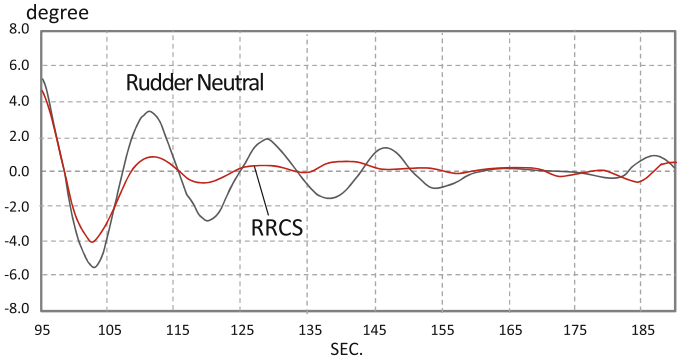

Fig. 3.18 Roll motions obtained by the RRCS steering and neutral rudder after impulsive forced roll (experimental data)

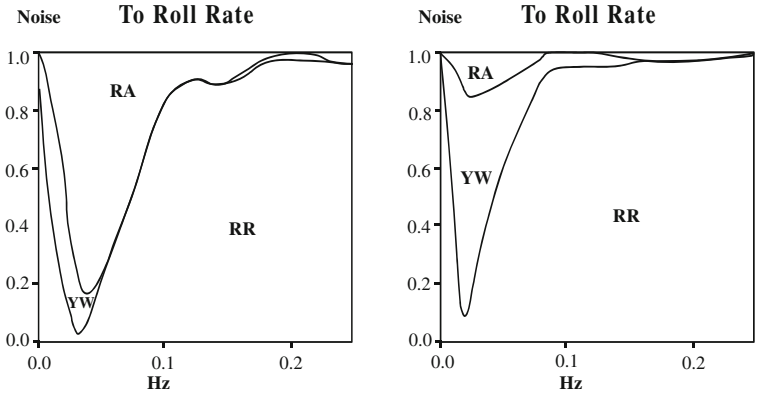

Fig. 3.19 Relative power contributions to the roll rate from roll rate (RR), rudder angle (RA), and yaw (YW). *Left* Conventional autopilot mode, *right* RRCS mode

conventional autopilot steering and the right-hand plot shows the results for the case of RRCS steering, both of which were carried out on Shioji-Maru III. Comparison of the results reveals a significant reduction of rudder effect on roll motion in the RRCS mode.

A major shipbuilding company in Japan has successfully used RRCS for practical applications on various types of ships.

3.4 Application to the Marine Main Engine Governor System

3.4.1 Marine Main Engine Governor

In addition to autopilot systems, the marine engine governor is another typical feedback system in the field of marine engineering. Figure 3.20 shows a schematic diagram of the marine main engine governor. The main engine transmits the revolution energy to the propeller through the propeller shaft. The function of the governor is to suppress the fluctuation of propeller revolution by adjusting the quantity of fuel supplied to piston cylinder in the main engine. Conventional governors are classified as either mechanical governors, which use the centrifugal force on the propeller shaft due to the propeller revolution, or electric governors. However, neither mechanical nor electric governors directly take into account ship motion, which is the primary cause of fluctuations in propeller revolution speed.

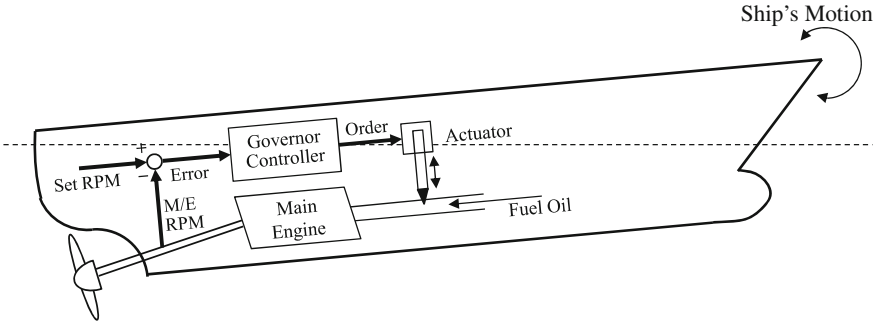

Fig. 3.20 Configuration of the main engine governor

3.4.2 Dynamic Characteristics of the Main Engine Governor System

Larger marine engines have been developed, and the reliability of such engines has increased. However, analysis of the dynamic characteristics of the main engine in actual navigation has been limited. In this section, we analyze a main engine governor system using the autoregressive model

$$\boldsymbol{x}_n = \sum_{m=1}^{M} A_m \boldsymbol{x}_{n-m} + \boldsymbol{u}_n \,, \tag{3.24}$$

fitted to the data discussed in the previous section, using the MAICE method (Ishizuka et al. 1991, 1992). Here, $\boldsymbol{x}_n$ is a two-dimensional vector time series and is composed of the propeller revolution speed and the governor command (hereinafter, the propeller rotation speed shall be expressed in revolutions per minute, RPM). Using this model, the frequency response function of the governor-propeller RPM and the impulse response function are calculated. They can be easily calculated using the autoregressive model (Akaike and Nakagawa 1988).

Figures 3.21 and 3.22 show the impulse response function and the frequency response function of the propeller RPM to the desired signal of a governor (rack-bar

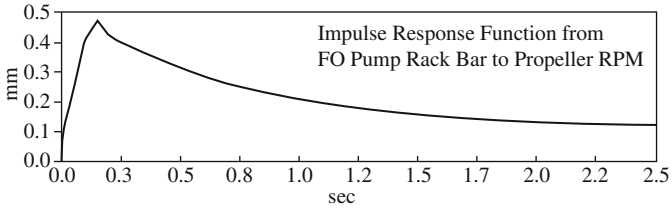

Fig. 3.21 Impulse response function of the governor-propeller revolution system

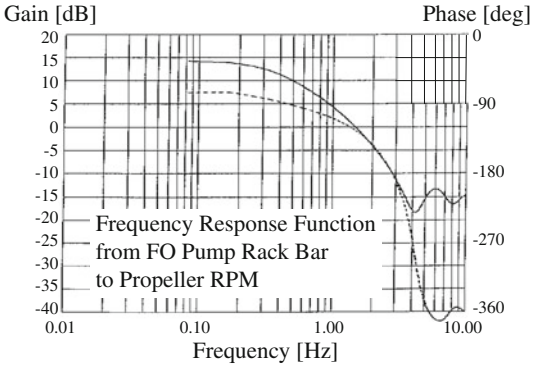

Fig. 3.22 Frequency response function of the governor-propeller revolution system (*solid line* amplitude, *dashed line* phase)

position). Based on these figures, the transfer function of this system is expected to be as follows:

$$D(s) = e^{-Ls} \frac{K}{s(T_1 s + 1)(T_2 s + 1)}. \tag{3.25}$$

Such knowledge will become available as computer-aided control system design is conducted using the PID control rule.

Figure 3.23 shows the noise contribution of the roll, pitch, and yaw to the change of the propeller RPM. The figure shows that the change in propeller RPM is subject to strong effects from pitch motion.

3.4.3 Design of the ARX Model-Based Governor

Considering the abovementioned theory, we designed the following two types of governor, applying the optimal controller based on the ARX model. The first governor is a SISO-type feedback controller (type-1 AR governor) that considers the actuator (the rack bar position for supplying fuel to the piston cylinder) as the input and the change of the propeller revolution as the output. The sampling rate is set to 0.1 s. First, random signal through the actuator is inlet to the main engine. Then, using the response of the propeller revolution to the input, the SISO AR model is constructed and the optimal gains are computed.

Figure 3.24 shows the results of an experiment using a middle-sized marine engine. The upper plots show the propeller RPM and fuel oil pump rack position of a conventional ship governor. The target RPM was 700 RPM. The lower plots show the

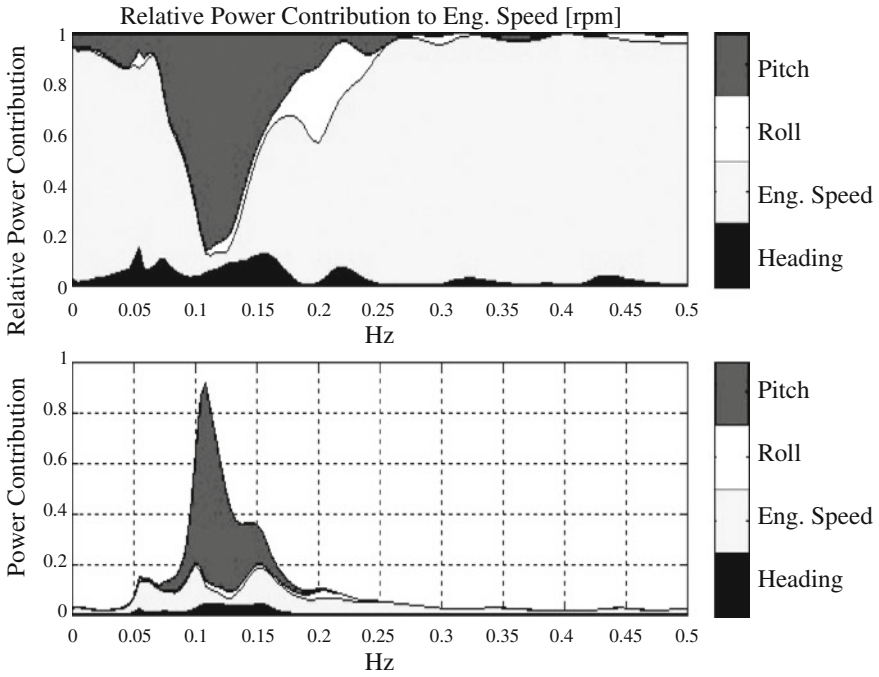

Fig. 3.23 Relative power contribution to propeller RPM from pitch, roll, propeller RPM (engine speed), and yaw (heading angle)

corresponding results for the optimal AR governor. The fluctuation of the propeller RPM became smaller with smaller governor changes for the optimal AR governor, as compared to the conventional governor.

Table 3.3 shows the variances of the engine RPM and the rack bar (Governor) motions for three values of Q/R in the performance criterion (3.6). The bottom row shows the results for the conventional governor. Figure 3.25 plots these results. In the case of $Q/R = 0.001$, the gain is relatively weak, and the change in RPM becomes greater than that for the conventional governor. In other cases, however, a significant reduction in RPM fluctuation is achieved by very small rack bar motions.

3.4.4 Design of the AR Governor Considering Pitch Motion

In the second type of marine engine governor, the ship's pitch rate signal is added to increase the accuracy of prediction of the change in the propeller revolution. The

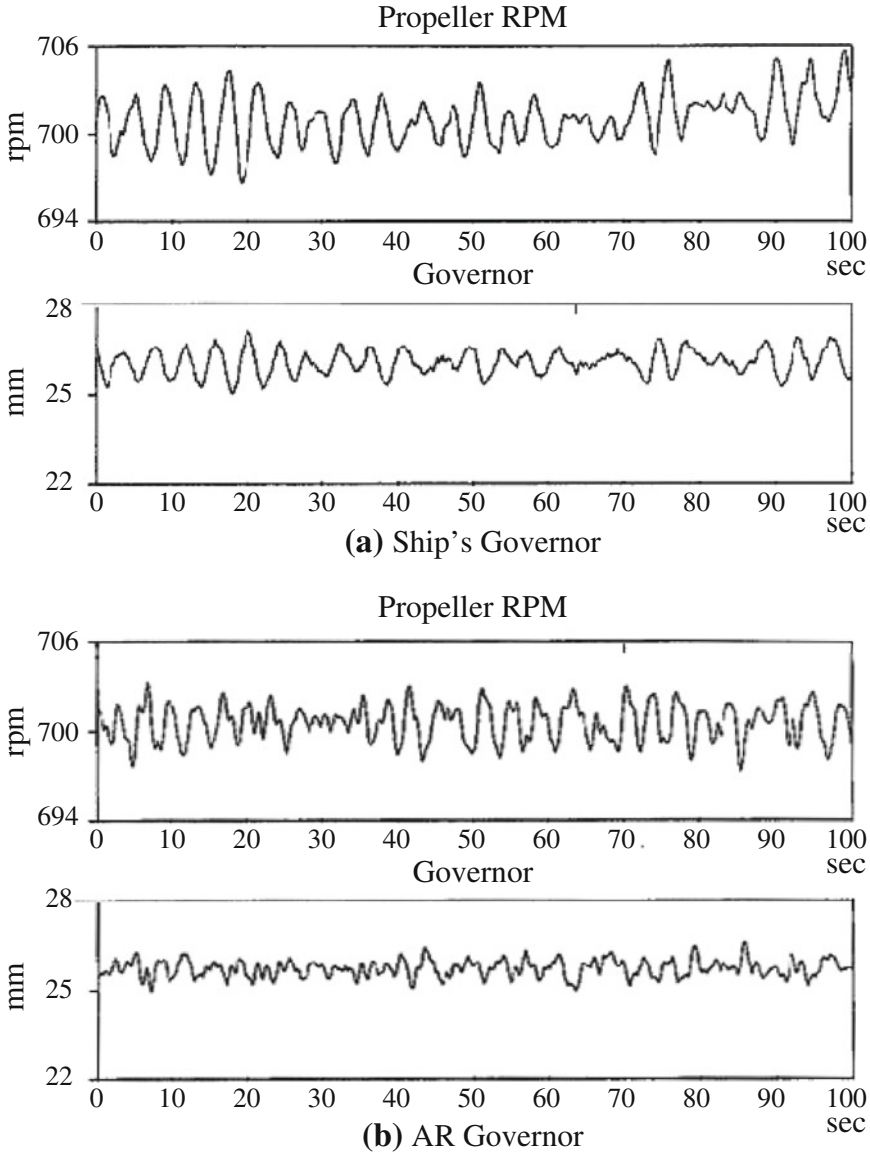

Fig. 3.24 Experimental results for engine RPM and engine governor. **a** Ship's governor. **b** AR governor

Table 3.3 Variances of the RPM and rack bar position of the optimal governor for three values of Q/R and those of the conventional governor obtained through a sea experiment

Gain's name	Q/R	Variance of RPM	Variance of rack bar
A0001	0.001	7.195	0.0196
A0005	0.005	2.251	0.0254
A0010	0.010	1.273	0.0756
Ship's governor	–	4.012	0.1710

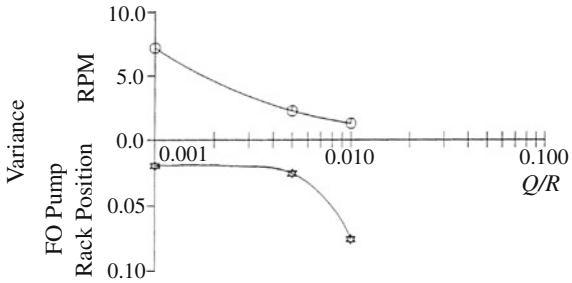

Fig. 3.25 Change of variances of engine RPM with respect to the change in Q/R. *Horizontal axis* Q/R, *vertical axis* variances of RPM and governor (FO pump rack position)

reason for this is that the pitch motion changes the depth of the propeller, which significantly affects the load on the engine. In this case, the weight to pitch motion in the performance criterion vanishes, because the pitch motion is not supposed to be reduced by the governor. The results of an experiment using the two types of governors are shown in Fig. 3.26. In this figure, the results for the type-2 governor, which took into account the pitch motion, are shown on the right, and the results for the type-1 governor, which did not use information on pitch motion, are shown on the left.

Comparing the results for two types of governors reveals that the propeller RPM is reduced significantly for the type-2 governor, and so the inclusion of pitch motion yields better engine governor control performance. The reason for this is likely because the inclusion of pitch motion allows more accurate prediction of the fluctuation of the RPM.

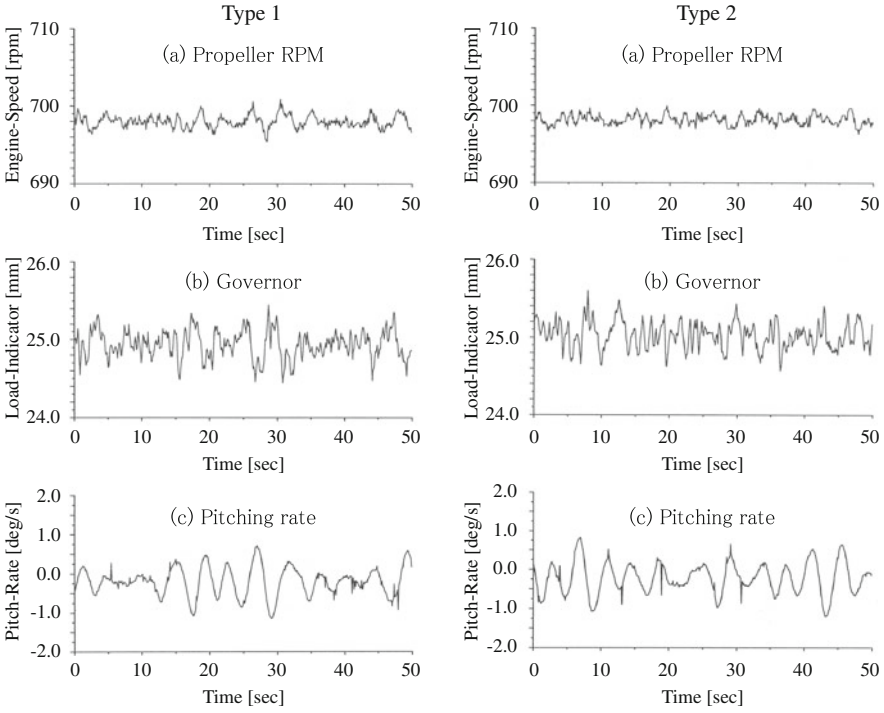

Fig. 3.26 Results for the type-1 and type-2 governors taking into account the pitch rate and the engine RPM

References

Akaike, H.: Autoregressive model fitting for control. Ann. Inst. Stat. Math. **23**, 163–180 (1971)

Akaike, H., Kitagawa, G. (eds.): The Practice of Time Series Analysis. Springer, New York (1999)

Akaike, H., Nakagawa, T.: Statistical Analysis and Control of Dynamic System. KTK Scientific Publishers, Tokyo (1988)

Åström, K.J., Källström, C.G.: Identification of ship steering dynamics. Automatica **12**(9), 9–22 (1976)

Bellman, R.: Dynamic Programming. Dover, New York (2003)

Ishizuka, M., Ohtsu, K., Hotta, T., Horigome, M.: Statistical identification and optimal control of marine engine system, part 1, part 2. J. Soc. Nav. Archit. Jpn. **170** 211–220, **171** 425–433 (1991)

Ishizuka, M., Ohtsu, K., Hotta, T., Horigome, M.: Statistical identification and optimal control of marine engine system, part 1, part 2. J. Soc. Nav. Archit. Jpn. **170** 211–220, **171** 425–433 (1992)

Kitagawa, G.: A square root algorithm for optimal controller design. (in Japanese with English abstract and FORTRAN source code). Proc. Inst. Stat. Math. **31**(1), 71–80 (1983)

Kitagawa, G., Ohtsu, K.: The statistical control of ship's course keeping motion (in Japanese with English abstract). Proc. Inst. Stat. Math. **23**(2), 105–128 (1976)

Nakamura, H., Akaike, H.: Statistical identification for optimal control of supercritical thermal power plants. Automatica **17**(1), 143–155 (1981)

Oda, H., Ohtsu, K., Sasaki, M., Seki, Y., Hotta, T.: Roll stabilization by rudder control through multivariate auto-regressive model. J. Kansai Soc. Nav. Archit. **216**, 165–173 (1991)

Oda, H., Ohtsu, K., Sato, H., Kanehiro, K.: Designing advanced rudder roll stabilization system. In: Proceedings of the 7th JFPS International Symposium on Fluid Power, Toyama, pp. 169–174. 15–18 Sept 2008

Ogata, K.: Modern Control Engineering, 2nd edn. Prentice Hall, Englewood Cliffs (1990)

Ohtsu, K.: Model Based on Monitoring System and the Optimal Control (in Japanese). Kaibundou, Tokyo (2012)

Ohtsu, K., Horigome, M., Kitagawa, G.: Statistical identification of ship's course keeping motion and optimal control. J. Soc. Nav. Archit. Jpn. **139** 31–44 (1976a)

Ohtsu, K., Horigome, M., Kitagawa, G.: Statistical identification of ship's course keeping motion and optimal control (2). J. Soc. Nav. Archit. Jpn. **143** 216–224 (1976b)

Ohtsu, K., Horigome, M., Kitagawa, G.: A new ship's auto pilot through a stochastic model. Automatica **15–3**, 255–268 (1979a)

Ohtsu, K., Horigome, M., Kitagawa, G.: A robust autopilot system against the various sea conditions. In: Proceedings of ISSOA Symposium, Tokyo (1979b)

Otomo, T., Nakagawa, T., Akaike, H.: Statistical approach to computer control of cement rotary kilns. Automatica **8**(1), 35–48 (1972)

Chapter 4
Advanced Autopilot Systems

Abstract In the previous chapter, we presented three types of autopilot system based on the stationary linear AR model. However, actual sea conditions may change gradually or abruptly due to various factors, and in such a situation, we must consider a nonstationary time series. Furthermore, it may become necessary to consider the nonlinear response of the ship, which will be of particular importance to tracking control. In this chapter, we propose extensions of our statistical optimal controller based on the locally stationary AR model and the RBF-ARX model and develop a noise-adaptive autopilot and a path-tracking autopilot.

Keywords Noise adaptive autopilot · RBF-ARX model based predictive control · Ship tracking control · GPS signal based ship tracking error measuring · Ship trajectory tracking dynamics modeling

4.1 Noise-Adaptive Autopilot System

The simplest and most practical approach to modeling nonstationary time series is to subdivide the time interval into several subintervals of appropriate size and fit a stationary AR model to each subinterval. Using this method, we can obtain a series of models that approximate nonstationary time series. In this section, we develop a noise-adaptive autopilot system based on the locally stationary AR model.

4.1.1 Construction of a Noise-Adaptive Control System

The noise-adaptive controller is an extension of the optimal controller, which has the ability to adapt to the changes in the characteristics of an external disturbance. We assume the ARX model for the system dynamics

$$\boldsymbol{y}_n = \sum_{i=1}^{m} A_i \boldsymbol{y}_{n-1} + \sum_{i=1}^{\ell} B_i \boldsymbol{r}_{n-i} + \boldsymbol{u}_n, \tag{4.1}$$

K. Ohtsu et al., *Time Series Modeling for Analysis and Control*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55303-8_4

where $\boldsymbol{y}_n$ is the p-dimensional output, $\boldsymbol{r}_n$ is the q-dimensional control input, and $\boldsymbol{u}_n$ is the p-dimensional system noise. It is assumed that the ship's dynamics is time invariant and known but the external disturbance $\boldsymbol{u}_n$ is not necessarily a white noise and its characteristics change gradually with time. Therefore, we consider a locally stationary AR model for the system noise:

$$\boldsymbol{u}_n = \sum_{j=1}^{k} C_j \boldsymbol{u}_{n-j} + \boldsymbol{\varepsilon}_n, \tag{4.2}$$

where $\boldsymbol{\varepsilon}_n$ is a p-dimensional Gaussian white noise with mean 0 and variance covariance matrix W. Substituting (4.1) into (4.2) yields the overall behavior model:

$$\boldsymbol{y}_n = \sum_{i=1}^{m+k} A_i' \boldsymbol{y}_{n-i} + \sum_{i=1}^{\ell+k} B_i' \boldsymbol{r}_{n-i} + \boldsymbol{\varepsilon}_n, \tag{4.3}$$

where A_n' and B_n' are given by

$$A_i' = A_i + C_i - \sum_{j=1}^{i} C_j A_{i-j}, \quad B_i' = B_i + C_i - \sum_{j=1}^{i} C_j B_{i-j} \tag{4.4}$$

with $A_i = O$ for $i > m$, $B_i = O$ for $i > \ell$, and $C_i = O$ for $i > k$.

As shown in Fig. 4.1, the noise-adaptive controller is composed of the following five modules:

External Disturbance Estimator: The external disturbance estimator estimates the external disturbance $\boldsymbol{u}_n$ from the observations of the output $\boldsymbol{y}_n$ and the input $\boldsymbol{r}_n$ using Eq. (4.1).

Nonstationary Noise Model Builder: The nonstationary noise model builder fits a locally stationary AR model to $\boldsymbol{u}_n$ and obtains the current noise model (4.2).

Overall Behavior Model Modifier: The overall behavior model modifier computes the current overall behavior model (4.3) using the original system dynamics model (4.1) and the current model of the external disturbance (4.2). The state-space representation of the overall behavior model is then obtained.

Optimal Controller Designer: The optimal controller designer designs the optimal feedback gain (3.21) with respect to a predetermined quadratic performance criterion.

Optimal Controller: The optimal controller controls the system using the optimal feedback gain by (3.20).

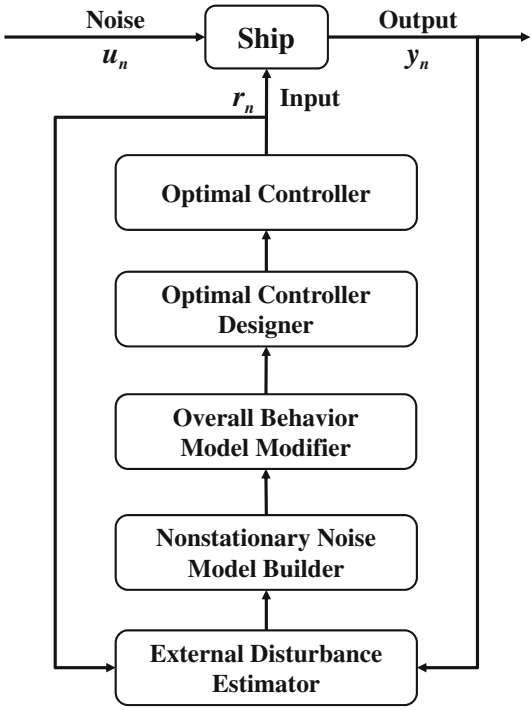

Fig. 4.1 Construction of the noise-adaptive controller

4.1.2 Actual Sea Test of the Noise-Adaptive Autopilot System

In order to realize a noise-adaptive ship's autopilot system based on the method described in the previous section, it is necessary to run two programs simultaneously. The first program, which should be primarily executed, inputs the ship's yaw deviation, calculates the optimal control input using Eq. (3.20), and sends the commands to the ship's steering gear. Usually, this program is executed at sampling rate $\Delta t = 1$ in the autopilot system.

The second program builds the current overall model based on the on-line identification procedure of the locally stationary AR model for the noise sequence and calculates the optimal gain to be used for the next stage at every $N\Delta t$ s ($N \geq 200$ points).

Figure 4.2 shows a record of the data collected on the training ship with similar size with Shioji-Maru III, used to demonstrate the effect of using the on-line identification procedure of the locally stationary AR model. The principal dimensions of the ship are listed in Table 1.1. In this experiment, the length of the basic span for fitting the AR model is fixed to $N_1 = N2 = \cdots = 200$ s.

Just before the start point of the results in Fig. 4.2, the ship navigated into open sea from leeward side of a peninsula and large deviation from the desired course occurred. The model of the disturbance was switched to a new model in order to

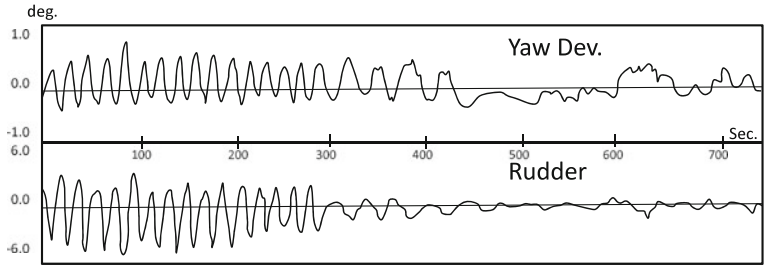

Fig. 4.2 Actual experiment using a noise-adaptive autopilot on a small ship

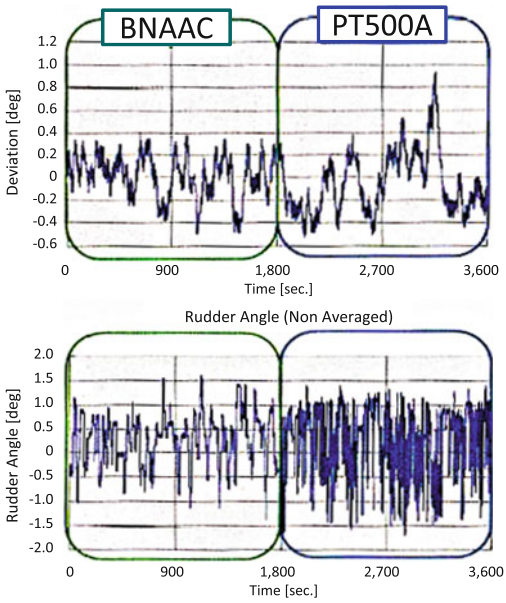

Fig. 4.3 Data obtained using the noise-adaptive autopilot on a large container ship. *Top* yaw deviation, *bottom* rudder angle

adapt to this change in the environment, and the ship's deviations were gradually reduced after approximately 300 s under the updated optimal gains.

Figure 4.3 shows another example of data obtained on a large container ship having a length of 250 m, a breadth of 35.40 m, a gross tonnage of 44,459, and 4,432 TEU. Two upper plots in this figure show the course deviation by the noise-adaptive autopilot (left: BNAAC) and the conventional autopilot (right: PT500A). The lower two plots show the corresponding rudder motions. The noise-adaptive autopilot system exhibited an excellent ability to suppress the yaw deviations by significantly smoother rudder motions, far better than the conventional autopilot system.

The noise-adaptive ship's autopilot gained formal approval from the Det Norske Veritas (DNV) as a trademark, the batch noise-adaptive autopilot controller (BNAAC) has been installed in approximately 500 ships worldwide.

4.2 RBF-ARX Model-Based Predictive Control

The principal role of the classical autopilot of a ship is primarily concerned with maintaining the yaw at a desired course angle. However, nowadays by making use of the position feedback signals obtained from a Global Positioning System (GPS) navigation device, a ship guidance system, namely, a tracking control system, can be designed to make the ship track a desired trajectory. This section introduces a multi-input and multi-output (MIMO) RBF-ARX model-based predictive controller design method for general nonlinear plant control, including ship tracking control issues (Peng et al. 2009). In Sect. 4.3, a GPS signal-based calculation approach to the tracking error and course deviation of a ship for better implementation of the arc trajectory tracking of a ship is presented, and the dynamics modeling of the ship using the RBF-ARX model and tracking controller design along with its application are described in Sect. 4.4.

4.2.1 MIMO RBF-ARX Model and Its State-Space Form

Based on the idea of building the SISO RBF-ARX model described in Sect. 2.8, the multivariable form of the model is introduced in this section (Peng et al. 2009). Consider a MIMO nonlinear and nonstationary system with a measurable disturbance, which is represented by the following nonlinear autoregressive model:

$$\boldsymbol{y}_n = f(\boldsymbol{w}_{n-1}) + \boldsymbol{\zeta}_n, \tag{4.5}$$

where

$$\boldsymbol{w}_{n-1} = \left[\boldsymbol{y}_{n-1}^T \cdots \boldsymbol{y}_{n-k_a}^T, \boldsymbol{u}_{n-1}^T \cdots \boldsymbol{u}_{n-k_b}^T, \boldsymbol{v}_{n-1}^T \cdots \boldsymbol{v}_{n-k_d}^T\right]^T \tag{4.6}$$

$\boldsymbol{y}_n \in \Re^k$ is the output, $\boldsymbol{u}_n \in \Re^m$ is the input, $\boldsymbol{v}_n \in \Re^l$ is the measurable disturbance, and $\boldsymbol{\zeta}_n \in \Re^k$ denotes the white noise sequence. If the function $f(\cdot)$ in (4.5) is assumed to be continuously differentiable at an arbitrary equilibrium point, $f(\cdot)$ may be expanded in a Taylor series, and model (4.5) may then be rewritten as the following state $(\boldsymbol{w}_{n-1})$-dependent autoregressive model, as describe for the SISO case in Sect. 2.8:

$$\begin{aligned}\boldsymbol{y}_n = \pi_0(\boldsymbol{w}_{n-1}) &+ \sum_{i=1}^{k_a} \pi_{yi}(\boldsymbol{w}_{n-1})\boldsymbol{y}_{n-i} + \sum_{i=1}^{k_b} \pi_{ui}(\boldsymbol{w}_{n-1})\boldsymbol{u}_{n-i} \\ &+ \sum_{i=1}^{k_d} \pi_{vi}(\boldsymbol{w}_{n-1})\boldsymbol{v}_{n-i} + \boldsymbol{\zeta}_n,\end{aligned} \tag{4.7}$$

where π_0, π_{yi}, π_{ui}, and π_{vi} are state-dependent coefficients matrices of suitable dimensions. The above problem may be rewritten by allowing $\boldsymbol{w}_n$ to be the process variables causing the working point of the system to change with time. $\boldsymbol{w}_n$ may be

directly or indirectly related to the input or output of the system, in some cases being the input or output itself. For example, in a nonlinear thermal power plant, $\boldsymbol{w}_n$ could be the load demand of the plant (Peng et al. 2009). Similar to the derivation of the SISO RBF-ARX model as described in Sect. 2.8, if Gaussian RBF networks are used to approximate the functional elements of the coefficient matrices of model (4.7), the MIMO RBF-ARX model can then be obtained as follows:

$$\begin{aligned}\boldsymbol{y}_n = \boldsymbol{\phi}_0(\bar{\boldsymbol{w}}_{n-1}) + \sum_{i=1}^{k_a} \boldsymbol{\phi}_{yi}(\bar{\boldsymbol{w}}_{n-1})\boldsymbol{y}_{n-i} + \sum_{i=1}^{k_b} \boldsymbol{\phi}_{ui}(\bar{\boldsymbol{w}}_{n-1})\boldsymbol{u}_{n-i} \\ + \sum_{i=1}^{k_d} \boldsymbol{\phi}_{vi}(\bar{\boldsymbol{w}}_{n-1})\boldsymbol{v}_{n-i} + \boldsymbol{\xi}_n,\end{aligned} \tag{4.8}$$

where

$$\boldsymbol{\phi}_0(\bar{\boldsymbol{w}}_{n-1}) = c_0^0 + \sum_{p=1}^{h} c_p^0 \exp\left\{-||\bar{\boldsymbol{w}}_{n-1} - \boldsymbol{z}_{yp}||^2_{\lambda_{yp}}\right\} \tag{4.9}$$

$$\begin{aligned}\boldsymbol{\phi}_{ji}(\bar{\boldsymbol{w}}_{n-1}) &= c_{i0}^j + \sum_{p=1}^{h} c_{ip}^j \exp\left\{-||\bar{\boldsymbol{w}}_{n-1} - \boldsymbol{z}_{jp}||^2_{\lambda_{jp}}\right\},\ j = y, u, v \\ \bar{\boldsymbol{w}}_{n-1} &= \left[\boldsymbol{w}_{n-1}^T, \boldsymbol{w}_{n-2}^T, \ldots, \boldsymbol{w}_{n-n_w}^T\right]^T \\ \boldsymbol{z}_{jp} &= \left[z_{jp1}^T, z_{jp2}^T, \ldots, z_{jpn_w}^T\right]^T,\ j = y, u, v\end{aligned} \tag{4.10}$$

and k_a, k_b, k_d, h, and n_w are the model orders, $\boldsymbol{z}_{jk}$'s are the centers of RBF networks, $\boldsymbol{c}_{ik}^j$'s and $\boldsymbol{c}_k^0$'s are the weighting coefficient matrices of suitable dimension, $||\boldsymbol{x}||^2_{\hat{\lambda}} = \boldsymbol{x}^T\hat{\boldsymbol{\lambda}}\boldsymbol{x}$, $\hat{\boldsymbol{\lambda}} = \text{diag}\left(\hat{\lambda}_1^2, \ldots, \hat{\lambda}^2_{\dim(\boldsymbol{x})}\right)$, $\left\{\hat{\lambda}_1, \ldots, \hat{\lambda}_{\dim(\boldsymbol{x})}\right\}$ are the scaling factors, and $\left\{\boldsymbol{\xi}_n \in \Re^k\right\}$ denotes the white noise sequence that is assumed to satisfy

$$E\left\{\boldsymbol{\xi}_n | F_{n-1}\right\} = 0, \quad E\left\{\boldsymbol{\xi}_n\ \boldsymbol{\xi}_n^T\right\} = \boldsymbol{\Omega}, \tag{4.11}$$

in which F_t denotes the σ-algebra generated by the data up to and including time n, and $\boldsymbol{\Omega}$ is a positive definite matrix. The off-line estimation approach to the SISO RBF-ARX model, referred to as the structured nonlinear parameter optimization method (SNPOM) introduced in Sect. 2.8, may be applied to off-line estimation of the parameters of MIMO RBF-ARX model (4.8) after a minor alteration (Peng et al. 2009).

In order to design the MIMO RBF-ARX model-based (4.8) predictive controller, we need to obtain the state-space form of the model. To this end, we can rewrite model (4.8) in the following matrix polynomial form:

$$\boldsymbol{y}_n = \sum_{i=1}^{k_n} \boldsymbol{a}_{i,n-1}\boldsymbol{y}_{n-i} + \sum_{i=1}^{k_n} \boldsymbol{b}_{i,n-1}\boldsymbol{u}_{n-i} + \boldsymbol{\phi}_{n-1} + \boldsymbol{\xi}_n \tag{4.12}$$

where $k_n = \max\{k_a, k_b\}$ and

$$\boldsymbol{a}_{i,n-1} = \begin{cases} c_{i0}^{y} + \sum_{p=1}^{h} c_{ip}^{y} \exp\left\{-||\overline{\boldsymbol{w}}_{n-1} - \boldsymbol{z}_{yp}||^2_{\lambda_{yp}}\right\}, & i \le k_a \\ 0, & i > k_a \end{cases}.$$

$$\boldsymbol{b}_{i,n-1} = \begin{cases} c_{i0}^{u} + \sum_{p=1}^{h} c_{ip}^{u} \exp\left\{-||\overline{\boldsymbol{w}}_{n-1} - \boldsymbol{z}_{up}||^2_{\lambda_{up}}\right\}, & i \le k_b \\ 0, & i > k_b \end{cases}.$$

$$\boldsymbol{\phi}_{n-1} = \boldsymbol{\phi}_0(\overline{\boldsymbol{w}}_{n-1}) + \sum_{i=1}^{k_d} \boldsymbol{\phi}_{vi}(\overline{\boldsymbol{w}}_{n-1})\boldsymbol{v}_{n-i}. \tag{4.13}$$

Defining the state vector as

$$\begin{aligned} \boldsymbol{x}_n &= \left[\boldsymbol{x}_{1,n}^T \cdots \boldsymbol{x}_{k_n,n}^T\right]^T \\ \boldsymbol{x}_{1,n} &= \boldsymbol{y}_n \\ \boldsymbol{x}_{j,n} &= \sum_{i=1}^{k_n+1-j} \boldsymbol{a}_{i+j-1,n-1}\boldsymbol{y}_{n-i} + \sum_{i=1}^{k_n+1-j} \boldsymbol{b}_{i+j-1,n-1}\boldsymbol{u}_{n-i}, \quad j = 2, 3, \ldots, k_n, \end{aligned} \tag{4.14}$$

a state-space model corresponding to model (4.8) or (4.12) can be then obtained by

$$\begin{cases} \boldsymbol{x}_{n+1} = \boldsymbol{A}_n\boldsymbol{x}_n + \boldsymbol{B}_n\boldsymbol{u}_n + \boldsymbol{\Phi}_n + \boldsymbol{\Xi}_{n+1} \\ \boldsymbol{y}_n = \boldsymbol{C}\boldsymbol{x}_n \end{cases}, \tag{4.15}$$

where

$$\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{a}_{1,n} & \mathbf{1} & 0 & \cdots & 0 \\ \boldsymbol{a}_{2,n} & 0 & \mathbf{1} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{k_n-1,n} & 0 & 0 & \cdots & \mathbf{1} \\ \boldsymbol{a}_{k_n,n} & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{B}_n = \begin{bmatrix} \boldsymbol{b}_{1,n} \\ \boldsymbol{b}_{2,n} \\ \vdots \\ \boldsymbol{b}_{k_n,n} \end{bmatrix}$$

$$\boldsymbol{\Phi}_n = \begin{bmatrix} \boldsymbol{\phi}_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \boldsymbol{\Xi}_{n+1} = \begin{bmatrix} \boldsymbol{\xi}_{n+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} \mathbf{1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T. \tag{4.16}$$

The model given by (4.15) and (4.16) is a state-space representation of MIMO RBF-ARX model (4.8). Note that the state $\boldsymbol{x}_n$ at time n in (4.15) can be easily calculated by (4.14) according to the present output $\boldsymbol{y}_n$, the past input/output data, and the off-line estimated MIMO RBF-ARX model (4.8).

4.2.2 MIMO RBF-ARX Model-Based Nonlinear MPC

Model predictive control (MPC) refers to a class of computer control algorithms that use an explicit process model to predict the future response of a plant. At each control interval, an MPC algorithm attempts to optimize future plant behavior by computing a sequence of future control variables and taking systematic account of equipment and safety constraints. The first input in the optimal sequence is sent to the plant, and the entire calculation is repeated at subsequent control intervals. Model predictive control has had a significant and widespread impact on industry. Originally developed to meet the specialized control needs of power plants and petroleum refineries, MPC technology is currently used in a wide variety of application areas, including chemicals, food processing, automotive, and aerospace applications (Qin and Badgwell 2003).

Linear model-based MPC has numerous theoretical and practical applications. At present, the nonlinear model-based MPC is the focus of a great deal of research and application. There are three basic types of method for coping with nonlinear system modeling and predictive control problems. The first type of method (e.g., Bloemen et al. 2001; Kothare 1996; Prasad et al. 1998) involves using a piecewise linearization technique to describe the nonlinear behavior of a system, so that the model is linearized in each sampling interval. This results in the solution of quadratic programming problems or linear matrix inequalities (LMIs) for each interval, as in case of linear-model-based predictive control (MPC). However, the identification of numerous linear models that are only effective in a small region is not easy.

The second type of method (e.g., Sentoni et al. 1996; Mahfouf and Linkens 1998; Mizuno et al. 2007) is based on the direct use of nonlinear models, which involve the on-line solution of a higher-order nonlinear optimization problem with constraints, which is usually computationally expensive and may not even guarantee a feasible solution in real-time control.

The third type of method involves using a local linearization approach for representing a nonlinear plant using an on-line estimated affine model (e.g., Lakhdari et al. 1995) or an off-line estimated globally nonlinear and locally linear RBF-ARX model (Peng et al. 2003, 2004, 2009) and then solving a quadratic programming problem on-line in order to obtain optimal control. In the former case, however, fast and accurate on-line estimation of a complicated model providing a good fit to a nonlinear process may be difficult in actual application.

RBF-ARX model-based nonlinear MPC algorithms have been investigated both in simulation and in real industrial applications (Peng et al. 2004, 2006, 2007, 2009, 2010, 2011; Qin et al. 2014; Zeng et al. 2014; Wu et al. 2012), where the satisfactory nonlinear modeling accuracy and significant effectiveness and feasibility of the algorithms have been verified. Furthermore, some stability conclusions on the RBF-ARX model-based nonlinear MPC were also given in Peng et al. (2007, 2011).

This section introduces the framework of the RBF-ARX model-based nonlinear MPC algorithm. A unified form of the MIMO RBF-ARX model-based MPC strategy using only quadratic programming routines to solve an optimization problem on-line

for a class of nonlinear systems is presented in this section. In the first case, the global MIMO RBF-ARX model is used for predicting the future output of the system when the knowledge of the future working point state prediction is available. In the second case, in which a local linearization ARX model that is easily obtained from the global MIMO RBF-ARX model must be used to obtain an affine output prediction in control when the working point state prediction is not available.

In order to design the MIMO RBF-ARX model-based (4.8) or (4.15) MPC strategy, the following vectors of signals are first defined:

$$\begin{aligned}
\hat{\boldsymbol{x}}_n &= \left[\hat{\boldsymbol{x}}_{n+1|n}^T, \hat{\boldsymbol{x}}_{n+2|n}^T, \ldots, \hat{\boldsymbol{x}}_{n+N|n}^T\right]^T, \quad \boldsymbol{x}_{p,n} = \left[\boldsymbol{x}_{n+1|n}^T, \boldsymbol{x}_{n+2|n}^T, \ldots, \boldsymbol{x}_{n+N|n}^T\right]^T, \\
\hat{\boldsymbol{y}}_n &= \left[\hat{\boldsymbol{y}}_{n+1|n}^T, \hat{\boldsymbol{y}}_{n+2|n}^T, \ldots, \hat{\boldsymbol{y}}_{n+N|n}^T\right]^T, \quad \boldsymbol{y}_{p,n} = \left[\boldsymbol{y}_{n+1|n}^T, \boldsymbol{y}_{n+2|n}^T, \ldots, \boldsymbol{y}_{n+N|n}^T\right]^T \\
\hat{\boldsymbol{u}}_n &= \left[\boldsymbol{u}_n^T, \boldsymbol{u}_{n+1}^T, \ldots, \boldsymbol{u}_{n+N_u-1}^T\right]^T, \qquad \bar{\boldsymbol{\Phi}}_n = \left[\boldsymbol{\Phi}_n^T, \boldsymbol{\Phi}_{n+1}^T, \ldots, \boldsymbol{\Phi}_{n+N-1}^T\right]^T
\end{aligned} \tag{4.17}$$

where N is the prediction horizon, whereas N_n ($N_n \leq N$) is the control horizon after which control signals are assumed to have no variation, i.e., $\boldsymbol{u}_{n+j} = \boldsymbol{u}_{n+N_u-1}$ ($j \geq N_u$). If $\{\boldsymbol{u}_{n+j-1} | j = 1, 2, \ldots, N_u\}$ is assumed to be F_t-measurable, then, based on model (4.15) at time n, the j ($j = 1, 2, \ldots, N$) step ahead optimal predictive state and output may be obtained as follows:

$$\begin{cases} \hat{\boldsymbol{x}}_n = E\left\{\boldsymbol{x}_{p,n}\right\} = \bar{\boldsymbol{A}}_n \boldsymbol{x}_n + \bar{\boldsymbol{B}}_n \hat{\boldsymbol{u}}_n + \bar{\boldsymbol{\Gamma}}_n \bar{\boldsymbol{\Phi}}_n \\ \hat{\boldsymbol{y}}_n = E\left\{\boldsymbol{y}_{p,n}\right\} = \bar{\boldsymbol{C}}_n \hat{\boldsymbol{x}}_n, \end{cases} \tag{4.18}$$

where the following system matrices may be defined:

$$\bar{\boldsymbol{A}}_n = \left[\left(\prod_{j=0}^{0} \boldsymbol{A}_{n+j}\right)^T, \left(\prod_{j=0}^{1} \boldsymbol{A}_{n+j}\right)^T \ldots, \left(\prod_{j=0}^{N-1} \boldsymbol{A}_{n+j}\right)^T\right]^T$$

$$\prod_{j=i}^{q} \boldsymbol{A}_{n+j} = \begin{cases} \boldsymbol{A}_{n+q} \boldsymbol{A}_{n+q-1} \ldots \boldsymbol{A}_{n+i}, & i \leq q \\ \boldsymbol{1}, & i > q \end{cases}$$

$$\bar{\boldsymbol{B}}_n = \begin{bmatrix}
\boldsymbol{B}_n & 0 & 0 & \cdots & 0 \\
\left(\prod_{j=1}^{1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_n & \boldsymbol{B}_{n+1} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
\left(\prod_{j=1}^{N_u-1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_n & \left(\prod_{j=2}^{N_u-1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+1} & \cdots & \left(\prod_{j=N_u-1}^{N_u-1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+N_u-2} & \boldsymbol{B}_{n+N_u-1} \\
\left(\prod_{j=1}^{N_u} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_n & \left(\prod_{j=2}^{N_u} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+1} & \cdots & \left(\prod_{j=N_u-1}^{N_u} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+N_u-2} & \sum_{i=N_u-1}^{N_u} \left(\prod_{j=i+1}^{N_u} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+i} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\left(\prod_{j=1}^{N-1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_n & \left(\prod_{j=2}^{N-1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+1} & \cdots & \left(\prod_{j=N_u-1}^{N-1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+N_u-2} & \sum_{i=N_u-1}^{N-1} \left(\prod_{j=i+1}^{N-1} \boldsymbol{A}_{n+j}\right) \boldsymbol{B}_{n+i}
\end{bmatrix}$$

$$\bar{\boldsymbol{\Gamma}}_n = \begin{bmatrix} \mathbf{1} & 0 & 0 & \cdots & 0 \\ \prod_{j=1}^{1} \boldsymbol{A}_{n+j} & \mathbf{1} & 0 & \cdots & 0 \\ \prod_{j=1}^{2} \boldsymbol{A}_{n+j} & \prod_{j=2}^{2} \boldsymbol{A}_{n+j} & \mathbf{1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \prod_{j=1}^{N-1} \boldsymbol{A}_{n+j} & \prod_{j=2}^{N-1} \boldsymbol{A}_{n+j} & \cdots & \prod_{j=N-1}^{N-1} \boldsymbol{A}_{n+j} & \mathbf{1} \end{bmatrix}, \quad \bar{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C} & 0 & \cdots & 0 \\ 0 & \boldsymbol{C} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{C} \end{bmatrix}. \tag{4.19}$$

The system matrices $\bar{\boldsymbol{A}}_n$, $\bar{\boldsymbol{B}}_n$, $\bar{\boldsymbol{C}}$, and $\bar{\boldsymbol{\Gamma}}_n$ in (4.18) are calculated using Eq. (4.19), for which the knowledge of the working point state prediction $\hat{\boldsymbol{w}}_{n+j|n}$ ($j = 1, 2, \ldots, N-1$) is required. It is possible to obtain the knowledge for some controlled processes, such as thermal power plants, under the load-pattern operating condition and regarding the load as $\boldsymbol{w}_n$ in (4.8) to describe the working point state. If the working point state prediction is not available, we may have to replace $\hat{\boldsymbol{w}}_{n+j|n}$ with $\boldsymbol{w}_n$ in order to compute $\bar{\boldsymbol{A}}_n$, $\bar{\boldsymbol{B}}_n$, $\bar{\boldsymbol{C}}$, and $\bar{\boldsymbol{\Gamma}}_n$ in (4.18) and (4.19). From (4.18), the output prediction may also be represented as follows:

$$\hat{\boldsymbol{y}}_n = \boldsymbol{G}_n \hat{\boldsymbol{u}}_n + \boldsymbol{y}_{0,n} \tag{4.20}$$

where

$$\begin{aligned} \boldsymbol{G}_n &= \bar{\boldsymbol{C}} \bar{\boldsymbol{B}}_n \\ \boldsymbol{y}_{0,n} &= \bar{\boldsymbol{C}} \bar{\boldsymbol{A}}_n \boldsymbol{x}_n + \bar{\boldsymbol{C}} \bar{\boldsymbol{\Gamma}}_n \bar{\boldsymbol{\Phi}}_n \end{aligned} \tag{4.21}$$

Define the control move sequence $\Delta \hat{\boldsymbol{u}}_n$ and the desired output sequence $\hat{\boldsymbol{y}}_{r,n}$ as

$$\begin{aligned} \Delta \hat{\boldsymbol{u}}_n &= [\Delta \boldsymbol{u}_n^T, \Delta \boldsymbol{u}_{n+1}^T, \ldots, \Delta \boldsymbol{u}_{n+N_u-1}^T]^T \\ \hat{\boldsymbol{y}}_{r,n} &= [\boldsymbol{y}_{r,n+1}^T, \boldsymbol{y}_{r,n+2}^T, \ldots, \boldsymbol{y}_{r,n+N}^T]^T, \end{aligned} \tag{4.22}$$

where $\Delta \boldsymbol{u}_n = \boldsymbol{u}_n - \boldsymbol{u}_{n-1}$. Now consider the following optimization problem:

$$\begin{aligned} &\min_{\hat{\boldsymbol{u}}_n} J = ||\hat{\boldsymbol{y}}_n - \hat{\boldsymbol{y}}_{r,n}||^2_{\mathbf{1}_{n\times n}} + ||\hat{\boldsymbol{u}}_n||^2_{\boldsymbol{R}_1} + ||\Delta \hat{\boldsymbol{u}}_n||^2_{\boldsymbol{R}_2} \\ &\text{s.t. } \boldsymbol{y}_{\min} \le \hat{\boldsymbol{y}}_n \le \boldsymbol{y}_{\max},\ \boldsymbol{u}_{\min} \le \hat{\boldsymbol{u}}_n \le \boldsymbol{u}_{\max},\ \Delta \boldsymbol{u}_{\min} \le \Delta \hat{\boldsymbol{u}}_n \le \Delta \boldsymbol{u}_{\max}, \end{aligned} \tag{4.23}$$

where $||\boldsymbol{x}||^2_{\boldsymbol{R}} \equiv \boldsymbol{x}^T \boldsymbol{R} \boldsymbol{x}$, $\boldsymbol{R} = \text{diag}(r_1, r_2, \ldots, r_{\dim(\boldsymbol{x})})$ is a positive definite diagonal weighting matrix, and both control levels and control moves are penalized.

Substituting (4.20) into (4.23), after removing the constant terms, a quadratic form of optimization problem that is equivalent to (4.23) may be obtained as follows:

$$\begin{aligned}\min_{\hat{\boldsymbol{u}}_n} \tilde{J} = & \frac{1}{2}\hat{\boldsymbol{u}}_n^T\left[\boldsymbol{G}_n^T\boldsymbol{G}_n + \boldsymbol{R}_1 + \boldsymbol{E}^{-T}\boldsymbol{R}_2\boldsymbol{E}^{-1}\right]\hat{\boldsymbol{u}}_n \\ & + \left[\boldsymbol{y}_{0,n}^T\boldsymbol{G}_n - \hat{\boldsymbol{y}}_{r,n}^T\boldsymbol{G}_n - \boldsymbol{u}_{0,n-1}^T\boldsymbol{E}^{-T}\boldsymbol{R}_2\boldsymbol{E}^{-1}\right]\hat{\boldsymbol{u}}_n \\ & \text{s.t.} \begin{bmatrix}\boldsymbol{G}_n \\ -\boldsymbol{G}_n\end{bmatrix}\hat{\boldsymbol{u}}_n \le \begin{bmatrix}\boldsymbol{y}_{max} - \boldsymbol{y}_{0,n} \\ -\boldsymbol{y}_{min} + \boldsymbol{y}_{0,n}\end{bmatrix} \\ & \boldsymbol{u}_{min} \le \hat{\boldsymbol{u}}_n \le \boldsymbol{u}_{max}, \\ & \boldsymbol{u}_{0,n-1} + \boldsymbol{E}\Delta\boldsymbol{u}_{min} \le \hat{\boldsymbol{u}}_n \le \boldsymbol{u}_{0,n-1} + \boldsymbol{E}\Delta\boldsymbol{u}_{max},\end{aligned} \tag{4.24}$$

where $\hat{\boldsymbol{u}}_n = \boldsymbol{u}_{0,n-1} + \boldsymbol{E}\Delta\hat{\boldsymbol{u}}_n$, $\boldsymbol{u}_{0,n-1} = [\boldsymbol{u}_{n-1}^T, \boldsymbol{u}_{n-1}^T, \ldots, \boldsymbol{u}_{n-1}^T]^T$, and

$$\boldsymbol{E} = \begin{bmatrix}\boldsymbol{1} & & & 0 \\ \boldsymbol{1} & \boldsymbol{1} & & \\ \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} & \\ \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1}\end{bmatrix}. \tag{4.25}$$

The on-line optimization problem (4.24) may be solved by the quadratic programming (QP) routines. In the solution for optimal control $\hat{\boldsymbol{u}}_n$, only the first component $\boldsymbol{u}_n$ is used as a control input. Note that the MIMO RBF-ARX model-based predictive controller proposed for underlying nonlinear systems does not rely on on-line parameter estimation, because its internal model, i.e., MIMO RBF-ARX model (4.8), is a globally nonlinear and locally linear model that can be estimated off-line.

In the MPC optimization problem (4.24), the desired output sequence $\hat{\boldsymbol{y}}_{r,n}$ in (4.22) is used to make the control process smooth and to avoid an excessively large variation of the control (input) variable $\boldsymbol{u}_n$. For the ship's tracking control, the controlled (output) variables are usually the ship cross track error and the yaw deviation. This means that the target values of the output variables are zeros, so the desired output sequence $\hat{\boldsymbol{y}}_{r,n}$ in (4.22) for the ship's tracking control may be designed as an exponential decay curve, as follows:

$$\boldsymbol{y}_{r,n+j} = \alpha\boldsymbol{y}_{r,n+j-1}, \quad \boldsymbol{y}_{r,n} = \boldsymbol{y}_n, \quad j = 1, 2, \ldots, N, \quad 0 \le \alpha < 1. \tag{4.26}$$

If the sample period of a control system is not very short, and the number of input variables in the MPC is not very large, i.e., the on-line optimization problem (4.23) based on the nonlinear global MIMO RBF-ARX model without using the local linearization technique used in (4.24) can be solved on-line within the sample period, we can obtain a globally optimized input variable by solving optimization problem (4.23) based on the MIMO RBF-ARX model (4.8). The MPC algorithm introduced in this section will be applied to the design of the ship's tracking control described in Sect. 4.4.

4.3 GPS Signal-Based Computation of a Ship's Tracking Error and Course Deviation

For a ship's tracking control problem, Fig. 4.4 shows the relationships between the ship's desired trajectory, the cross track error Dy, and the yaw deviation Ψ, in which U_0 is the velocity [m/s] of the ship (Peng et al. 2010). The purpose of the ship's tracking control is to regulate the rudder in order to minimize the ship cross track error Dy and the yaw deviation Ψ. Based on the GPS signals of a ship, we can use an Earth-fixed coordinate system to compute the ship's position, yaw deviation, and tracking error (Wu et al. 2012). Without loss of generality, the setting trajectory for the ship's tracking control can be designed in three parts, namely, two straight lines and one circular arc BC, the radius of which is $\overline{\mathrm{BR}}$ (or $\overline{\mathrm{RC}}$), as shown in Fig. 4.5.

The coordinate system XOY shown in Fig. 4.5 is an earth-fixed north-up coordinate system centered at the executing point O. The ship first sails from point O to point B along a straight line and then sails to point C along a circular arc trajectory. After that, the ship sails along another straight-line trajectory. $\angle$BRC is the turning angle.

Note that in the actual experiments, we filtered the GPS signals using a navigation Kalman filter as a pre-filter in order to increase the accuracy of the ship's position calculation. We transform the coordinate values (x, y) into (x', y'), which are the coordinate values in the $\mathrm{X'OY'}$ coordinate frame depicted in Fig. 4.6, using the following equation:

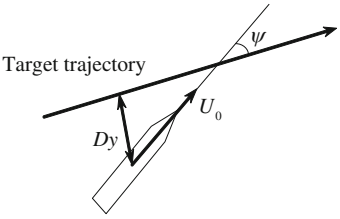

Fig. 4.4 Desired trajectory, tracking error, and yaw deviation

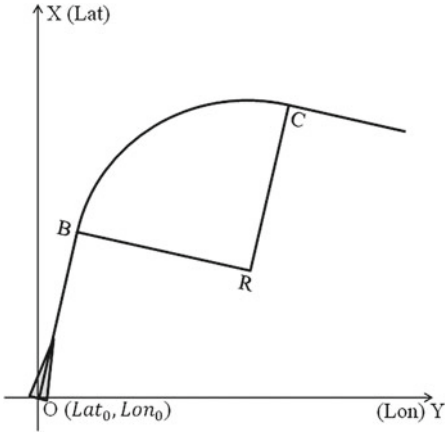

Fig. 4.5 Earth-fixed coordinate XOY and ship's setting trajectory

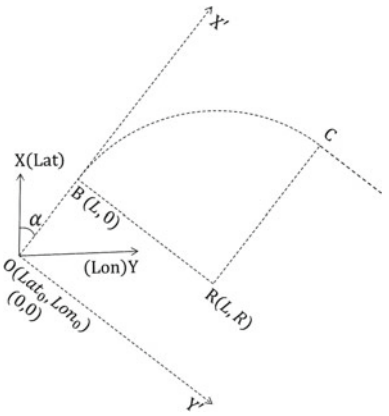

Fig. 4.6 Relative coordinate system $X'OY'$, $OB = L$, $BR = R$

$$x' = x\cos\frac{\alpha\pi}{180} + y\sin\frac{\alpha\pi}{180}$$
$$y' = y\cos\frac{\alpha\pi}{180} - x\sin\frac{\alpha\pi}{180}, \tag{4.27}$$

where α is the turning angle of the two coordinate systems, XOY and $X'OY'$. After that, in the relative coordinate system $X'OY'$, the calculation of the ship's tracking error and the yaw deviation will be divided into three cases as follows.

A. Case 1 (Straight-line tracking from point O to point B)

In Fig. 4.7, the ship sails from point O straight to point B along the X' axis. Point $A(x', y')$ is the present ship position. Also, α is the desired course angle, and θ is the measured present yaw angle. In this straight-line tracking segment, the ship's tracking error and yaw deviation can be computed as follows:

$$\Psi = \theta - \alpha, \quad \Psi \in (-180^\circ, 180^\circ]$$
$$Dy = y'. \tag{4.28}$$

B. Case 2 (Curve tracking)

After arriving at point B, the ship will sail to point C along a circular arc of radius $\overline{BR}$ (see Fig. 4.8). The coordinate values of points O, B, and R in the $X'OY'$ coordinate system are shown in Fig. 4.8 assuming that $\overline{OB} = L$ and $\overline{BR} = R$. In this curve tracking segment, the desired course is constantly changing with the variation of the ship's positions. For example, if point A is the ship's present position and θ is its present yaw angle, then the present desired course can be chosen as $\alpha + \gamma$, which

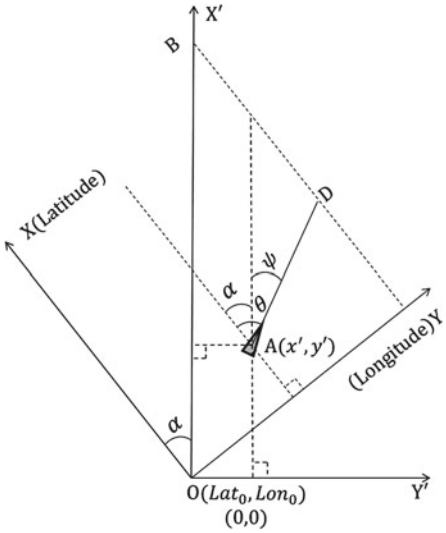

Fig. 4.7 Calculation of the tracking error and yaw deviation in the *straight-line* tracking from point O to point B (Note that this figure was created by the head-up system.)

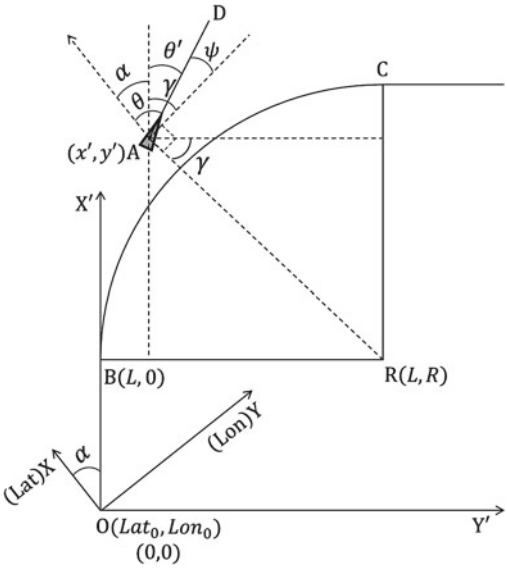

Fig. 4.8 Calculation of the tracking error and yaw deviation in the *curve* tracking segment from point B to point C (Note that this figure was created by the head-up system.)

is the tangent direction of the intersection point of $\overline{\mathrm{AR}}$ and BC in Fig. 4.8. Therefore, in this curve tracking segment, the ship's tracking error and yaw deviation can be computed as follows:

$$\begin{aligned} Dy &= R - \sqrt{(x' - L)^2 + (R - y')^2} \\ \Psi &= \theta' - \gamma, \quad \Psi \in (-180^\circ, 180^\circ], \end{aligned} \tag{4.29}$$

where θ is the observed yaw angle, $\theta' = \theta - \alpha$, $\theta' \in (-180°, 180°]$, and

$$\gamma = \arctan\left(\frac{x' - L}{R - y'}\right) \cdot \frac{180}{\pi}, \quad \gamma \in (-180°, 180°]. \tag{4.30}$$

C. Case 3 (Straight-line tracking from point C)

Starting from point C, the ship track follows another straight-line trajectory, as shown in Fig. 4.9, and the ship's desired course is $\alpha + \gamma$ ($\gamma = 90°$), which is the tangent direction of the intersection point of BC and $\overline{RC}$. Actually, this is the direction of the straight-line starting from C, as shown in Fig. 4.9.

In Fig. 4.9, the point $A(x', y')$ is the present position of the ship, and θ is the present observed yaw. Therefore, the yaw deviation is

$$\Psi = \theta' - \gamma, \quad \Psi \in (-180°, 180°] \tag{4.31}$$

where $\theta' = \theta - \alpha$. The coordinate values of point C in the $X'OY'$ coordinate system are

$$\begin{aligned} x_c' &= R \sin\frac{\beta\pi}{180} + L \\ y_c' &= R\left(1 - \cos\frac{\beta\pi}{180}\right) \end{aligned} \tag{4.32}$$

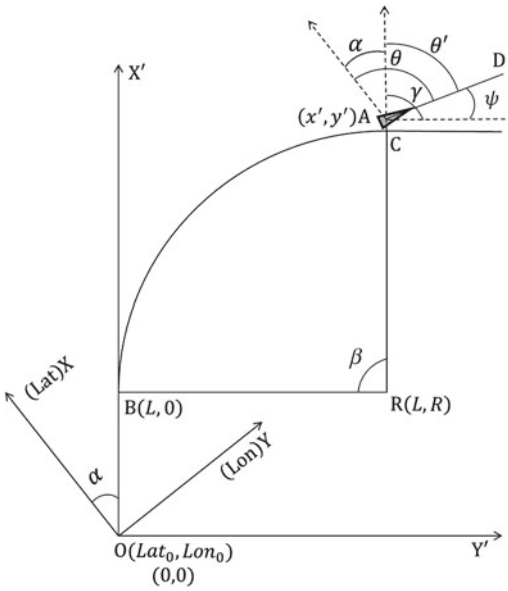

Fig. 4.9 Calculation of the tracking error and yaw deviation in the second *straight-line* tracking segment (Note that this figure was created by the head-up system.)

For obtaining the cross track error Dy in this case, it is necessary to rotate the $\mathrm{X}'\mathrm{OY}'$ coordinate system by β degrees, as in Eq. (4.27). Therefore, the ordinates of points C and A in the new relative coordinate system are

$$\begin{aligned} y_c'' &= y_c' \cos\frac{\gamma\pi}{180} - x_c' \sin\frac{\gamma\pi}{180} \\ y'' &= y' \cos\frac{\gamma\pi}{180} - x' \sin\frac{\gamma\pi}{180}. \end{aligned} \tag{4.33}$$

Then, in the second straight-line tracking segment, the cross track error should be

$$Dy = y'' - y_C''. \tag{4.34}$$

Finally, a multi-step-ahead forecast strategy can be designed for reducing the overshoot around points B and C in Fig. 4.10. As shown in Fig. 4.10, point A is the present position of the ship, $\overline{\mathrm{AT}}$ is the desired direction shown in Fig. 4.8, and $\overline{\mathrm{AT}'}$ is the k-step-ahead desired yaw direction. We start the curve tracking at point B′ and use the k-step-ahead desired yaw direction as the new desired course that can reduce the cross track error in the curve tracking segment. Therefore, we end the curve tracking at point C′, and from point C′, set the desired course to be the same as the course of the straight line starting from point C, as shown in Fig. 4.9. Thus, the additional course angle is $\angle\mathrm{TAT}'$, which is given by

$$\angle\mathrm{TAT}' = \angle\mathrm{C}'\mathrm{RC} = k \cdot \mathrm{BB}' \frac{360}{2\pi R}, \tag{4.35}$$

where $\mathrm{BB}' = V \cdot \Delta t$, V is the ship's speed [m/s], Δt is the sampling period [s], and k is the step number, which can be adjusted online. In the simulation studies and the real-time control experiments presented in Sect. 4.4.3, k is chosen to be 7. If a smaller value of k is used, the controller may not be able to effectively reduce the

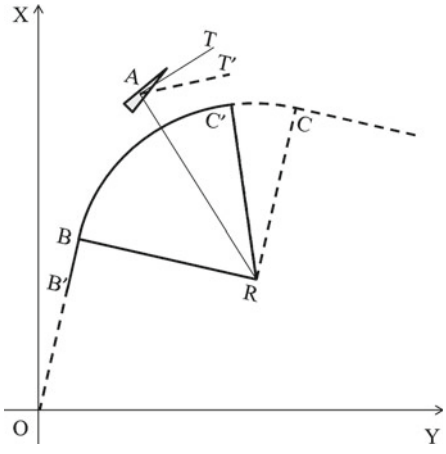

Fig. 4.10 Multi-step-ahead forecast in the *curve* tracking segment

cross track error when the ship is on the left-hand side of the curve tracking segment. If a larger value for k is set, a larger oscillation could occur when the ship is on the right-hand side of the curve.

4.4 Tracking Control Approach to Marine Vehicles

As GPS navigation technologies have become to be widely adopted in ships, the ship's tracking control method has become an area of increasing interest. One of the main reasons why ship owners want to use tracking control technologies is to save fuel. In order to save fuel, the captain needs to determine the optimum route using weather routing techniques considering the ship's resistance due to wind, waves, and current. Namely, in real navigation, a ship must track along the selected route as precisely as possible by means of efficient rudder control. This is an important role of the ship's tracking system. The simplest way of realizing tracking control is to change the ship's course in order to maintain its trajectory to follow the desired trajectory. The principles in many commercial-based tracking systems follow this type of control (Fossen 1994). However, this type of control does not directly consider rudder motion. Thus, unnecessarily large rudder angles might be used in many cases.

Holzhüter (1997) presented a linear-quadratic Gaussian (LQG) approach to high-precision tracking control of a ship, which can avoid unnecessary rudder motions by imposing an appropriate penalty to the rudder motion in the performance criterion. However, the ship's model in their paper contains numerous hydro-dynamic parameters, which were assumed to be known or must be estimated by troublesome maneuvering tank tests. Kvam et al. (2000) and Fukuda et al. (2001) also proposed this type of tracking control based on the Bryson and Ho's time varying control theory. However, these controllers did not consider the roll motion induced by the rudder motion and did not provide a general method of tracking a circular route.

There have been numerous studies on ship tracking control based on ship physical models. Pettersen and Nijmeijer (2001) and Lefeber et al. (2003) studied a complete state-tracking problem for an under-actuated ship that has only a surge control force and a yaw control moment. Using the same physical model, Jiang (2002) designed two systematic tracking controllers with the aid of Lyapunov's direct method. Do and Pan (2005) proposed a method by which to design a controller that forces the position and the orientation of an underactuated ship to globally track a reference trajectory. The reference trajectory is not required to be generated by the ship model. Moreiraa et al. (2007) designed a PID yaw controller and a speed controller to realize the path following control. The guidance system was designed using a way-point guidance scheme based on a line-of-sight projection algorithm. Do et al. (2002) proposed a universal controller that simultaneously solves the stabilization and tracking problems of underactuated surface ships under certain assumptions. Miyoshi et al. (2007) designed a linear optimal controller for tracking control. Du and Guo (2004) established an uncertain nonlinear mathematical model and designed a nonlinear adaptive controller for course-tracking control.

However, these methods were based on simplified physical models and it is not easy to obtain accurate parameters of a physical model for an actual ship, particularly for a large commercial ocean vehicle.

In order to overcome this difficulty, system identification techniques using actual ship sailing data for obtaining ship dynamics models are attractive approaches. Ohtsu et al. (1979), Ohtsu (1999), and Ohtsu and Kitagawa (1984) proposed a multi-variable autoregressive model to describe a ship's motion at sea. Using the fitted model, the ship's behavior was analyzed, and an optimal controller was designed for the ship's autopilot system. Park et al. (2000) proposed a practical batch-adaptive identification for a locally stationary process using an on-line minimum AIC procedure and based on the procedure and designed a roll-reducible autopilot system and a noise-adaptive system. Statistical modeling methods for describing ship motion and course-keeping control have been verified to be very effective and feasible in real applications. This section introduces a statistical modeling method to identify ship's dynamics for the purpose of tracking control along the desired track based on the RBF-ARX modeling method presented in Sects. 2.8 and 4.2.

Following the framework of the RBF-ARX modeling method, in this section, we first construct an RBF-ARX model for characterizing the dynamic behavior of the yaw deviation and the rudder angle of a training ship (Peng et al. 2010). Since yaw motion is strongly affected by the rolling motion, which is referred to as the yaw-heel effect, the rolling angle is used as the RBF-ARX model index to make the model parameters vary with the ship sailing state (Lewis 1967). The ship's dynamic model for tracking control is identified off-line using the real data observed from the ship, and the model parameters may be estimated using the structured nonlinear parameter optimization method (SNPOM) presented in Sect. 2.8. After that, in order to represent the ship's tracking behavior, a state-space model expressing the relationship between the yaw deviations and the tracking position errors, which is used to predict the future movement of the ship for the purpose of tracking control, is introduced (Peng et al. 2010). The ship's state-space model-based predictive controller is then designed to maneuver the ship sailing forward at a constant velocity along a predefined reference tracking path (Peng et al. 2010). Finally, in this section, we present the results of simulation studies and real-time control of "T.S. Shioji-Maru III", which demonstrate the effectiveness of the presented modeling and control methods (Peng et al. 2010; Wu et al. 2012).

4.4.1 RBF-ARX Model-Based Ship Motion Modeling for Tracking Control

As mentioned in Sect. 4.4 and shown in Fig. 4.4, the purpose of the ship's tracking control is to make the ship cross track error Dy and the yaw angle deviation Ψ as small as possible by controlling the rudder angle. For this purpose, we first use the RBF-ARX modeling method, which is introduced in Sect. 2.8, to describe the dynamic behavior between the yaw deviations and the rudder angles considering

the yaw-heel effect, which is usually nonlinear and its accurate physical model is difficult to obtain. The following SISO RBF-ARX model is constructed in order to represent the nonlinear dynamic behavior:

$$\Psi_n = a_0(\boldsymbol{s}_{n-1}) + \sum_{i=1}^{M} a_i(\boldsymbol{s}_{n-1})\Psi_{n-i} + \sum_{i=1}^{L} b_i(\boldsymbol{s}_{n-1})\delta_{n-1} + \xi_n, \tag{4.36}$$

where Ψ_n is the yaw deviation [deg], δ_n is the rudder angle [deg], s_n is the roll [deg], ξ_n denotes the noise, which is usually regarded as Gaussian white noise independent of the observations, a_i and b_i are the Gaussian RBF-net style nonlinear time-varying parameters of the ARX structure-type model,

$$\begin{aligned} a_i(\boldsymbol{s}_{n-1}) &= c^a_{i,0} + \sum_{k=1}^{m} c^a_{i,k} \exp\left(-\lambda^a_k ||\boldsymbol{s}_{n-1} - \boldsymbol{z}^a_k||_2^2\right) \\ b_i(\boldsymbol{s}_{n-1}) &= c^b_{i,0} + \sum_{k=1}^{m} c^b_{i,k} \exp\left(-\lambda^b_k ||\boldsymbol{s}_{n-1} - \boldsymbol{z}^b_k||_2^2\right) \\ \boldsymbol{s}_{n-1} &= \left(s_{n-1}, s_{n-2}, \ldots, s_{n-n_w}\right)^T \\ \boldsymbol{z}^j_k &= \left(z^j_{k,1}, z^j_{k,2}, \ldots, z^j_{k,n_w}\right)^T, \quad j = a, b, \end{aligned} \tag{4.37}$$

and $\boldsymbol{z}^j_k$ is the center of the RBF network, λ^j_k is the scaling factor, and M, L, m, and n_w are the model orders. The RBF-ARX model (4.36) is constructed as a global model and is estimated off-line from observation data so as to avoid the potential problem caused by the failure of on-line parameter estimation during real-time control. It is easy to see that the local linearization of model (4.36) is a linear ARX model at each working point by fixing $\boldsymbol{s}_{n-1}$ at time $n-1$ in (4.36). The RBF-ARX model (4.36) may be estimated off-line by the structured nonlinear parameter optimization method (SNPOM) presented in Sect. 2.8.

From Fig. 4.4, the following formula (4.38), representing the relation between the yaw deviation and the increment of cross track error within a sample period, can be derived as

$$Dy_n - Dy_{n-1} = U_0 \Delta t \sin(\Psi_{n-1}), \tag{4.38}$$

where Dy_n is the cross track error [m] at sample instant n, U_0 is the sailing velocity [m/s], and Δt is the sample period [s] of the tracking control system to be designed. From the RBF-ARX model (4.36) and formula (4.38), by defining the state variable $\boldsymbol{x}_n$ and the output variable $\boldsymbol{y}_n$ as

$$\begin{aligned} \boldsymbol{x}_n &= \left[\Psi_n,\ \Psi_{n-1},\ \ldots,\Psi_{n-M+1},\ \delta_{n-1},\ \delta_{n-2},\ \ldots,\delta_{n-L+1},\ Dy_n\right]^T \\ \boldsymbol{y}_n &= [\Psi_n,\ Dy_n]^T, \end{aligned} \tag{4.39}$$

we can obtain the state-space model that characterizes the tracking motion as follows:

$$\begin{cases} \boldsymbol{x}_n = \boldsymbol{A}_{n-1}\boldsymbol{x}_{n-1} + \boldsymbol{B}_{n-1}\delta_{n-1} + \boldsymbol{\Phi}_{n-1} + \Xi_n \\ \boldsymbol{y}_n = \boldsymbol{C}\boldsymbol{x}_n \end{cases} \tag{4.40}$$

where the matrices and vectors in Eq. (4.40) are given as

$$\boldsymbol{A}_{n-1} = \begin{bmatrix} a_1(\boldsymbol{s}_{n-1}) & \cdots & a_{M-1}(\boldsymbol{s}_{n-1}) & a_M(\boldsymbol{s}_{n-1}) & b_2(\boldsymbol{s}_{n-1}) & \cdots & b_{L-1}(\boldsymbol{s}_{n-1}) & b_L(\boldsymbol{s}_{n-1}) & 0 \\ 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ dy_{n-1}(\boldsymbol{x}_{n-1}) & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \tag{4.41}$$

$$dy_{n-1}(\boldsymbol{x}_{n-1}) = \begin{cases} \dfrac{U_0 \Delta t \sin(\Psi_{n-1}\pi/180)}{\Psi_{n-1}}, & \Psi_{n-1} \neq 0 \\ U_0 \Delta t \pi/180. & \Psi_{n-1} = 0 \end{cases} \tag{4.42}$$

$$\boldsymbol{B}_{n-1} = \begin{bmatrix} b_1(\boldsymbol{s}_{n-1}) \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \boldsymbol{\Phi}_{n-1} = \begin{bmatrix} a_0(\boldsymbol{s}_{n-1}) \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, \quad \Xi_n = \begin{bmatrix} \xi_n \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{4.43}$$

After identifying the RBF-ARX model (4.36), the state-space representation of tracking error dynamic model (4.40) can be constructed. Note that model (4.40) can be used to represent the dynamic behavior of tracking error within a large variation range of yaw deviation, because of the introduction of the accurate nonlinear formula (4.42) instead of its linear approximation.

4.4.2 Predictive Controller Design for Path Tracking of a Ship

Model (4.40) is a SIMO state-space model, which is similar to model (4.15). Therefore, using the model (4.15)-based MPC approach presented in Sect. 4.2, we can design a predictive controller based on model (4.40) to implement the ship's tracking control. To do this, the following vectors of signals must first be defined

$$\begin{aligned} \hat{\boldsymbol{x}}(n) &= \left[\hat{\boldsymbol{x}}_{n+1|n}^T, \ldots, \hat{\boldsymbol{x}}_{n+L|n}^T\right]^T & \boldsymbol{x}_p(n) &= \left[\boldsymbol{x}_{n+1|n}^T, \ldots, \boldsymbol{x}_{n+L|n}^T\right]^T \\ \hat{\boldsymbol{y}}(n) &= \left[\hat{\boldsymbol{y}}_{n+1|n}^T, \ldots, \hat{\boldsymbol{y}}_{n+L|n}^T\right]^T & \boldsymbol{y}_p(n) &= \left[\boldsymbol{y}_{n+1|n}^T, \ldots, \boldsymbol{y}_{n+L|n}^T\right]^T \\ \boldsymbol{u}(n) &= \left[\delta_n, \ldots, \delta_{n+L_u-1}\right]^T & \bar{\boldsymbol{\Phi}}_n &= \left[\boldsymbol{\Phi}_n^T, \ldots, \boldsymbol{\Phi}_{n+L-1}^T\right]^T \end{aligned} \tag{4.44}$$

where L is the prediction horizon, whereas L_u ($L_u \leq L$) is the control horizon after which control signals are assumed to have no variation, i.e., $\delta_{n+j} = \delta_{n+L_u-1} (j \geq L_u)$. Assuming that $\{\delta_{n+j-1} | j = 1, 2, \ldots, L_u\}$ is F_t-measurable, based on model (4.40) at instant n, the j-step ($j = 1, \ldots, L$) ahead optimal predictive state and output may be obtained as follows:

$$\begin{aligned} \hat{\boldsymbol{x}}(n) &= \mathrm{E}\left\{\boldsymbol{x}_p(n)\right\} = \bar{\boldsymbol{A}}_n \boldsymbol{x}_n + \bar{\boldsymbol{B}}_n \boldsymbol{u}(n) + \bar{\boldsymbol{\Gamma}}_n \bar{\boldsymbol{\Phi}}_n \\ \hat{\boldsymbol{y}}(n) &= \mathrm{E}\left\{\boldsymbol{y}_p(n)\right\} = \bar{\boldsymbol{C}} \hat{\boldsymbol{x}}(n) = \boldsymbol{G}_n \boldsymbol{u}(n) + \boldsymbol{y}_0(n), \end{aligned} \tag{4.45}$$

where

$$\begin{aligned} \boldsymbol{G}_n &= \bar{\boldsymbol{C}} \bar{\boldsymbol{B}}_n \\ \boldsymbol{y}_0(n) &= \bar{\boldsymbol{C}} \bar{\boldsymbol{A}}_n \boldsymbol{x}(n) + \bar{\boldsymbol{C}} \bar{\boldsymbol{\Gamma}}_n \bar{\boldsymbol{\Phi}}_n \end{aligned} \tag{4.46}$$

and $\mathrm{E}\{\boldsymbol{x}\}$ denotes the mathematical expectation of $\boldsymbol{x}$, and matrices $\bar{\boldsymbol{A}}_n$, $\bar{\boldsymbol{B}}_n$, $\bar{\boldsymbol{C}}$, and $\bar{\boldsymbol{\Gamma}}_n$ in Eq. (4.45) can be calculated by Eq. (4.19), for which the prediction of the working point state $s_{n+j|n}$ ($j = 1, 2, \ldots, L-1$) is required. It is possible to obtain the prediction by using a model to predict the future values of the rolling angle signal. If the working point state prediction is not available, $s_{n+j|n}$ ($j = 1, 2, \ldots, L-1$) can be replaced by s_n for computing $\bar{\boldsymbol{A}}_n$, $\bar{\boldsymbol{B}}_n$, $\bar{\boldsymbol{C}}$, and $\bar{\boldsymbol{\Gamma}}_n$ in Eq. (4.45).

Define the control move by the difference of control inputs sequence $\Delta U(n)$ and the reference output sequence $Y_r(n)$ as follows:

$$\begin{aligned} \Delta \hat{\boldsymbol{u}}_n &= [\Delta\delta_n, \Delta\delta_{n+1}, \ldots, \Delta\delta_{n+L_u-1}]^T \\ \boldsymbol{y}_r(n) &= [\boldsymbol{y}_{r,n+1}^T, \boldsymbol{y}_{r,n+2}^T, \ldots, \boldsymbol{y}_{r,n+L}^T]^T, \end{aligned} \tag{4.47}$$

where $\Delta\delta_n = \delta_n - \delta_{n-1}$. Now consider the following optimization problem, which is similar to optimization problem (4.23):

$$\begin{aligned} &\min_{\boldsymbol{u}(n)} \; J = ||\hat{\boldsymbol{y}}(n) - \boldsymbol{y}_r(n)||_{\boldsymbol{Q}}^2 + ||\boldsymbol{u}(n)||_{\boldsymbol{R}_1}^2 + ||\Delta\boldsymbol{u}(n)||_{\boldsymbol{R}_2}^2 \\ &\text{s.t.} \; \boldsymbol{y}_{\min} \leq \hat{\boldsymbol{y}}(n) \leq \boldsymbol{y}_{\max}, \; \boldsymbol{u}_{\min} \leq \Delta\boldsymbol{u}(n) \leq \boldsymbol{u}_{\max}, \; \Delta\boldsymbol{u}_{\min} \leq \Delta\boldsymbol{u}_n \leq \Delta\boldsymbol{u}_{\max}, \end{aligned} \tag{4.48}$$

where $||\boldsymbol{x}||^2_{\boldsymbol{\Omega}} \equiv \boldsymbol{x}^T \boldsymbol{\Omega} \boldsymbol{x}$, $\boldsymbol{R}_1 = \text{diag}(r_{1,1}, \ldots, r_{1,L_u})$, $\boldsymbol{R}_2 = \text{diag}(r_{2,1}, \ldots, r_{2,L_u})$, and $\boldsymbol{Q} = \text{diag}\left\{\bar{\boldsymbol{Q}}_1^T, \bar{\boldsymbol{Q}}_2^T, \ldots, \bar{\boldsymbol{Q}}_L^T\right\}$, $\bar{\boldsymbol{Q}}_i = [Q_1 \;\; Q_2]^T$ are the positive definite diagonal weighting matrices, and both control levels and control moves are penalized. In the simulation studies and the real-time experiments described in the present paper, we set $Q_1 = 1$ and determine the other variables using the reciprocal of the permissible variance of error (Akaike and Nakagawa 1988; Kwakernaak and Sivan 1972).

Substituting (4.45) into (4.48), after removing the constant terms, a quadratic form of the optimization problem that is equivalent to (4.48) may be obtained as follows:

$$\begin{aligned}
\min_{\boldsymbol{u}(n)} \; \tilde{J} = & \frac{1}{2}\boldsymbol{u}(n)^T[\boldsymbol{G}_n^T \boldsymbol{Q} \boldsymbol{G}_n + \boldsymbol{R}_1 + \boldsymbol{F}^{-T}\boldsymbol{R}_2\boldsymbol{F}^{-1}]\boldsymbol{u}(n) \\
& + (\boldsymbol{y}_0(n)^T \boldsymbol{Q}\boldsymbol{G}_n - \boldsymbol{y}_r(n)^T \boldsymbol{Q}\boldsymbol{G}_n - \boldsymbol{u}_0(n-1)^T \boldsymbol{F}^{-T}\boldsymbol{R}_2\boldsymbol{F}^{-1})\boldsymbol{u}(n) \quad (4.49) \\
& \text{s.t.} \; \begin{bmatrix} \boldsymbol{G}_n \\ -\boldsymbol{G}_n \end{bmatrix} \boldsymbol{u}(n) \le \begin{bmatrix} \boldsymbol{y}_{max} - \boldsymbol{y}_0(n) \\ -\boldsymbol{y}_{min} + \boldsymbol{y}_0(n) \end{bmatrix} \\
& \boldsymbol{u}_{min} \le \boldsymbol{u}(n) \le \boldsymbol{u}_{max}, \\
& \boldsymbol{u}_0(n-1) + \boldsymbol{F}\Delta\boldsymbol{u}_{min} \le \boldsymbol{u}(n) \le \boldsymbol{u}_0(n-1) + \boldsymbol{F}\Delta\boldsymbol{u}_{max},
\end{aligned}$$

where $\boldsymbol{u}(n) = \boldsymbol{u}_0(n-1) + \boldsymbol{F}\Delta\boldsymbol{u}(n)$, $\boldsymbol{u}_0(n-1) = [\delta_{n-1}, \delta_{n-1}, \ldots, \delta_{n-1}]^T$ and

$$\boldsymbol{F} = \begin{bmatrix} 1 & & & 0 \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \quad (4.50)$$

The on-line optimization problem (4.49) may be solved by the quadratic programming (QP) routines. In the solution of the optimal control series $\boldsymbol{u}(n)$, only the first component δ_n is used as the control input, namely, the rudder command. Note that this RBF-ARX model-based predictive controller does not require on-line parameter estimation, because its internal model, i.e., RBF-ARX model (4.36), is a global model estimated off-line.

4.4.3 Simulation Study and Real-Time Experiment

In this section, we first introduce the simulation study and then present the experimental results obtained onboard the training ship, i.e., Shioji-Maru III. A photograph and the principal particulars of Shioji-Maru III are given in Fig. 1.3 and Table 1.1, respectively.

4.4.3.1 Simulation Study

The simulation study was performed according to the following procedures. First, a set of observation data from the ship is used to estimate the RBF-ARX model (4.36) by applying the SNPOM introduced in Sect. 2.8, and the state-space model (4.40), which is also used for controlling the system in the simulation study, is then constructed. After that, the predictive control signal obtained by solving the optimization problem (4.49) is used to perform the tracking control simulation. The observation data with a sampling period of $\Delta t = 1$ s are shown in Fig. 4.11. The yaw signals and the ruder signals were sampled from the ship's autopilot, and the rolling signals were sampled from a high-accuracy fiber optical gyro. Figure 4.11 shows that in order to obtain data containing the global dynamic character information of the sailing ship, an experienced seaman steered the rudder as randomly as possible. The modeling results are shown in Figs. 4.12 and 4.13, where the orders of the identified model (4.36) are $M = 11, L = 7, m = 1$, and $n_w = 3$. Figure 4.13 shows the poles of the estimated model (4.36) changing with the variation of the roll shown in Fig. 4.11, from which we can see that the model dynamic behavior changes with the ship state. Therefore, the model could be used to represent the locally linearized dynamics of the ship at different working points.

The simulation results for the tracking control for the ship are shown in Figs. 4.14, 4.15 and 4.16, in which the controlled system is the estimated model (4.36), and $L = 27$ and $L_u = 15$. The noise series, shown in Fig. 4.17, the standard deviation of which is the same as that of the modeling residual in Fig. 4.12, is added to the controlled system, and the roll is also set to vary with time, as in Fig. 4.12, in order to simulate the real sailing situation. Figure 4.14 shows the tracking control performance for the case in which the ship sails forward at a constant velocity along a straight line

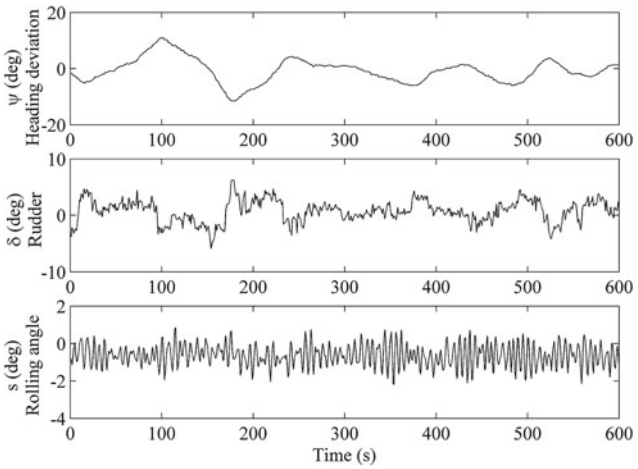

Fig. 4.11 Real-time observation data from Shioji-Maru III

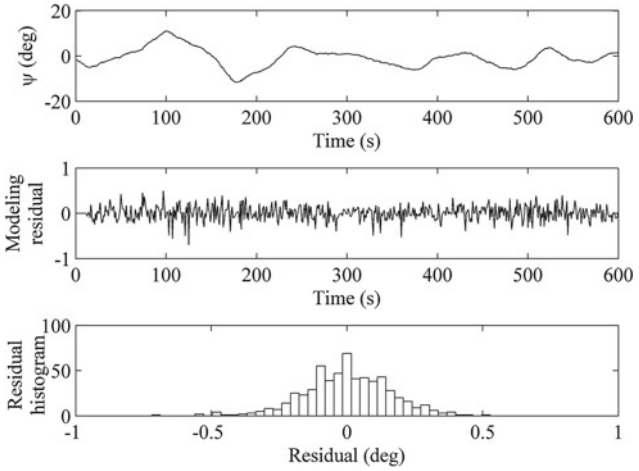

Fig. 4.12 Real value (*dotted line*) and one-step-ahead model predictive output (*solid line*) of the yaw deviation, and the modeling residual and its histogram from the estimated RBF-ARX model (4.36). The standard deviation of the modeling residual is 0.1621

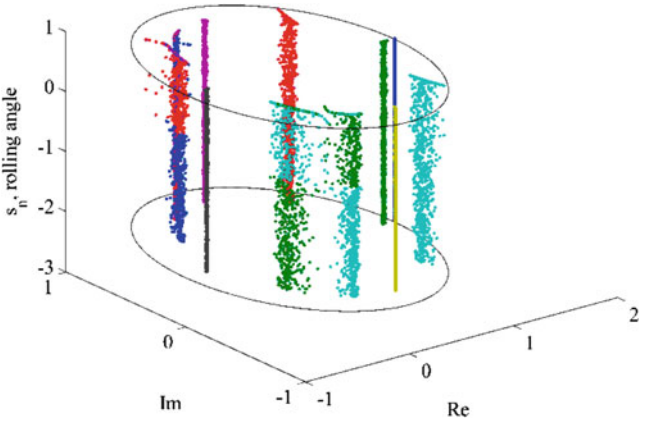

Fig. 4.13 Poles of the estimated RBF-ARX model (4.36) changing with the variation of the rolling angle (*Different colors* indicate *different poles*.)

and with a large initial position deviation. Figures 4.15 and 4.16 show the tracking control performance along the desired trajectory. The dynamic process and the steady process in two tracking control simulations show good results with a quick response, a small overshoot, and a slight tracking error.

Figures 4.15 and 4.16 show that, especially at turning points B and C shown in Fig. 4.10, no large overshoot occurred when using the multi-step-ahead forecast approach presented in Sect. 4.3, and in the curve tracking segment, because of using the seven-step-ahead desired yaw as the desired course, the cross track error Dy is effectively reduced. In other words, from Figs. 4.14 and 4.16, we can see that the

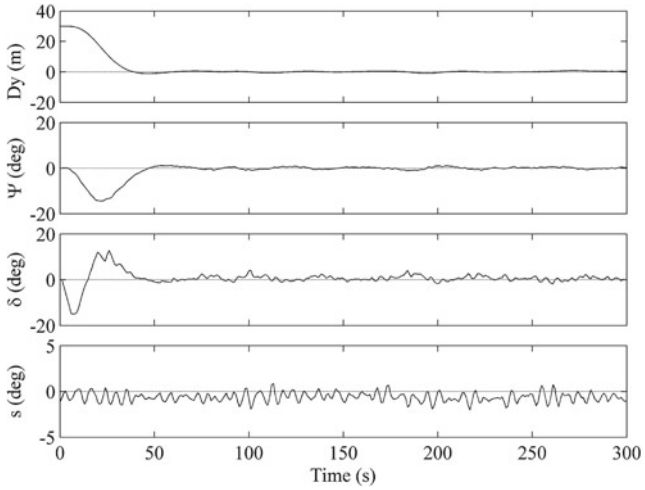

Fig. 4.14 Cross track error, yaw deviation, and rudder angle in the tracking control simulation sailing along a *straight line* at a constant velocity of 10.7 knots. The initial position error is 30 m; controller parameters: $Q_2 = 200$, $r_1 = 120$, $r_2 = 160$, $-15 \leq \delta \leq 15$, and $-3 \leq \Delta\delta \leq 3$

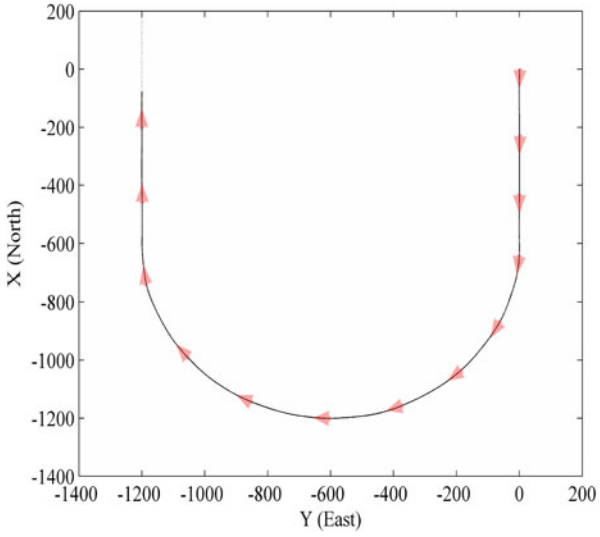

Fig. 4.15 Ship position in tracking control simulation, initially sailing forward at a constant velocity of 11.7 knots along a *straight line* in the negative x direction for 600 m, along a half circle of radius 600 m, and finally along a *vertical line* in the positive x direction. Controller parameters: $Q_2 = 400$, $r_1 = 120$, $r_2 = 160$, $-15 \leq \delta \leq 15$, and $-3 \leq \Delta\delta \leq 3$ (*dotted line* reference trajectory; *solid line* ship position)

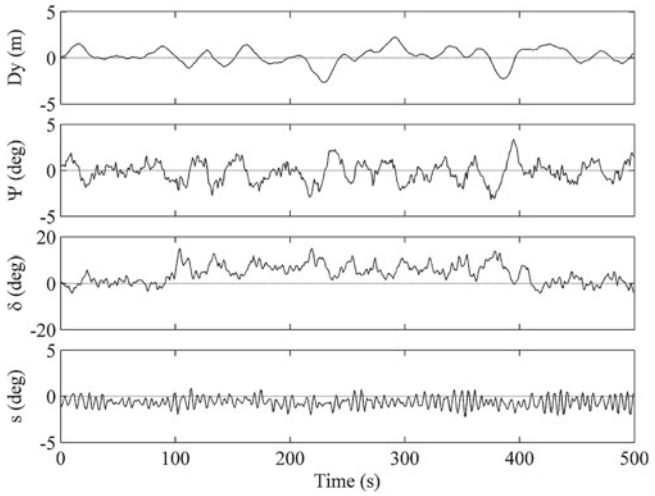

Fig. 4.16 Cross track error, yaw deviation, and rudder angle in tracking control simulation sailing forward at a constant velocity of 11.7 knots along the reference trajectory shown in Fig. 4.15

proposed approach can achieve both the straight-line tracking goal and the curve tracking goal very well.

The RBF-ARX model-based MPC and the reference trajectory design used in the simulation studies and the following real-time control are identical. In the above-described simulation studies, the ship's rolling angle is a sequence of data points sampled from the Shioji-Maru III and is used as the RBF-ARX model's index to make the model parameters vary with the sailing states of the ship. However, the model parameters are independent of the ship's rudder movement in the simulation. In the real-time control, the roll motion is not only affected by the sea condition, but is also directly and continuously influenced by the rudder motion. Large roll motion is always dangerous and should be avoided. Therefore, the controller parameters must be readjusted in order to suppress the rudder motion in the real-time control environment, as described in the next section.

4.4.3.2 Real-Time Experiment

A real-time tracking control experiment on the Shioji-Maru III using the RBF-ARX modeling and MPC design techniques was performed in Tokyo Bay. During the experiment, the sea condition was calm and there was a light breeze (wind force 2). Note that the roll signal in the following figures is used as the ship's state-dependent variables. Figures 4.17, 4.18, 4.19 and 4.20 show the real-time tracking control results of straight-line tracking for Shioji-Maru III under different controller parameters and different initial conditions using the RBF-ARX modeling and MPC design method presented in Sects. 2.8, 4.2–4.4.

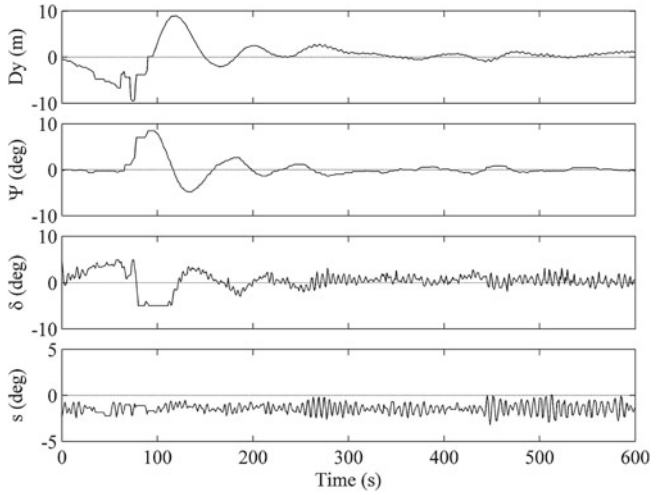

Fig. 4.17 Cross track error, yaw deviation, and rudder angle in real-time tracking control sailing forward at a constant velocity of 9.8 knots along a *straight line*. Controller parameters: $Q_2 = 550$, $r_1 = 600$, $r_2 = 500$, $-5 \leq \delta \leq 5$, and $-2.5 \leq \Delta\delta \leq 2.5$

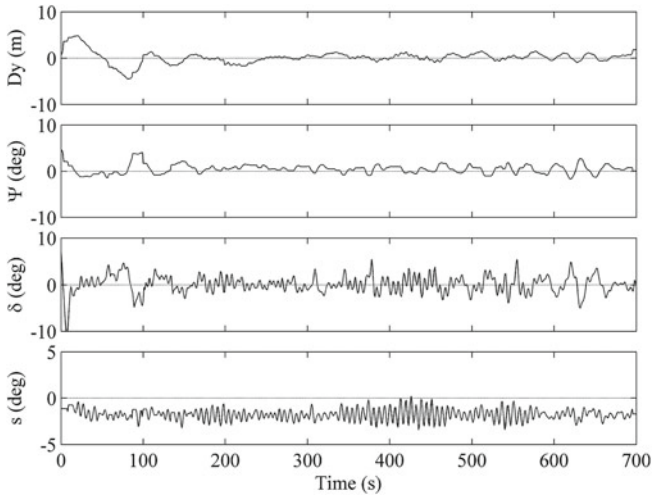

Fig. 4.18 Cross track error, yaw angle deviation, and rudder angle in real-time tracking control sailing forward at a constant velocity of 10.6 knots along a *straight line*. Controller parameters: $Q_2 = 600$, $r_1 = 550$, $r_2 = 1,000$, $-10 \leq \delta \leq 10$, and $-3 \leq \Delta\delta \leq 3$

Figure 4.17 shows the real-time straight-line tracking control results obtained using the present method. Before the method was applied to the ship, the ship was steered by a course keeping controller until the 74th second, as shown in Fig. 4.17, and the desired course was set to 210°. After shifting the control mode to the present

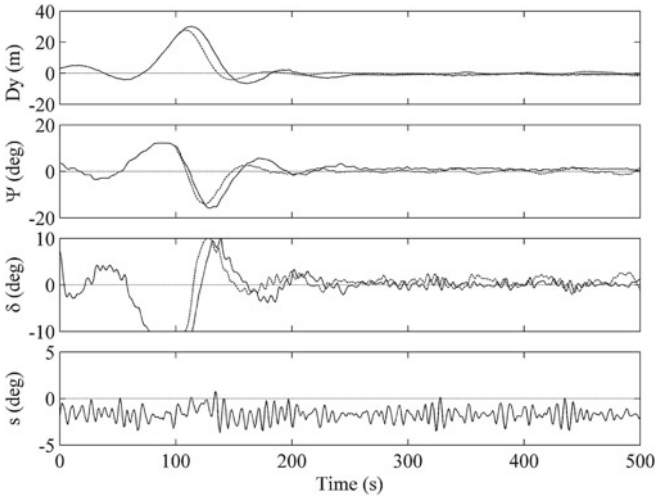

Fig. 4.19 Cross track error, yaw deviation, and rudder angle in tracking control. The *solid-line* shows the real-time tracking control results. The *dotted-lines* show the simulation results, which are for sailing along a *straight line* at a constant velocity of 10.7 knots. The initial position error is 30 m, and the controller parameters are set as $Q_2 = 600$, $r_1 = 550$, and $r_2 = 5{,}000$

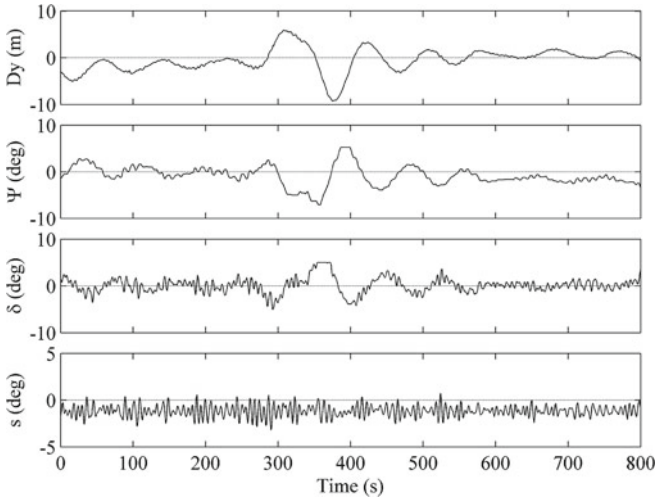

Fig. 4.20 Cross track error, yaw deviation, and rudder angle in real-time tracking control sailing forward at a constant velocity of 10.0 knots along a *straight line*. Controller parameters: $Q_2 = 550$, $r_1 = 600$, $r_2 = 500$, $-5 \leq \delta \leq 5$, and $-2.5 \leq \Delta\delta \leq 2.5$

tracking control strategy starting from the 74th second, we can see that the controller enables a quick response to reduce the gradually increased cross track error Dy, and the ship entered the steady state in approximately 200 s.

Figure 4.18 shows the real-time straight-line tracking control results for another desired course, which was 10°, and using another set of controller parameters. The large initial rudder change in Fig. 4.18 is caused by a large course change. Figures 4.17 and 4.18 show that the control performance is very good in the straight-line tracking control case under different control parameters and initial conditions. The cross track error in the tracking process after the course change can be controlled within ±2 m, as shown in Figs. 4.17 and 4.18. The ship breadth is 10 m (see Table 1.1), which means that the cross track error for the present approach was controlled to within ±0.2B (B is the ship breadth). To our knowledge, the obtained result is better than the results of previous studies, such as Holzhüter (1997) and Fukuda et al. (2001).

Figures 4.18, 4.19 and 4.20 show the control performance in the dynamic process more clearly. In Fig. 4.18, in order to investigate the large initial tracking error canceling ability, the ship is controlled by a helmsman from the 50th second to the 100th second in order to make an initial cross track error. From the 100th second, the control mode is again shifted from manual control to automatic control mode with the RBF-ARX model-based predictive controller. The controller can quickly eliminate the large offset of the cross track error Dy, as shown in Fig. 4.18.

In order to compare the simulation and the real-time control results for large initial tracking error canceling ability, the simulation results are shown in Fig. 4.19 for the same controller parameters in the real-time control in Fig. 4.18. The dynamic behaviors for the two records shown in Fig. 4.19 are very similar. In other words, Fig. 4.19 also demonstrates the effectiveness of the RBF-ARX modeling and MPC method presented in this book.

In Fig. 4.20, a large oscillation of Dy, which was caused by the change in the ocean current direction, occurred from the 300th s to the 400th s. As shown in Fig. 4.20, the presented control strategy also provides good control performance under the influence of a large disturbance.

The results of the curve tracking experiments are shown in Figs. 4.21, 4.22, 4.23 and 4.24. We conducted turning experiments tracking along a circular arc. Since the length of the experimental ship Shioji-Maru III is 46.0 m, the turning radiuses were chosen as 500 and 600 m, which are approximately ten times the ship length. The desired course design approach for tracking control is shown in Sect. 4.3. The setting trajectory and the real-time tracking control results obtained using the RBF-ARX modeling and MPC method are given in Figs. 4.21, 4.22, 4.23 and 4.24, which also show very good curve tracking control performance. Note that in the middle turning process of the 180° circle tracking in Figs. 4.23 and 4.24, since the ship sailed in the opposite direction after turning 180°, the sea condition could gradually influence the ship motion in the opposite manner. This may explain the larger roll motion after approximately 240 s in Fig. 4.24. Even in this special case, the present control strategy can also achieve very good curve tracking control performance, because the parameters of the RBF-ARX model can vary with the ship state on-line.

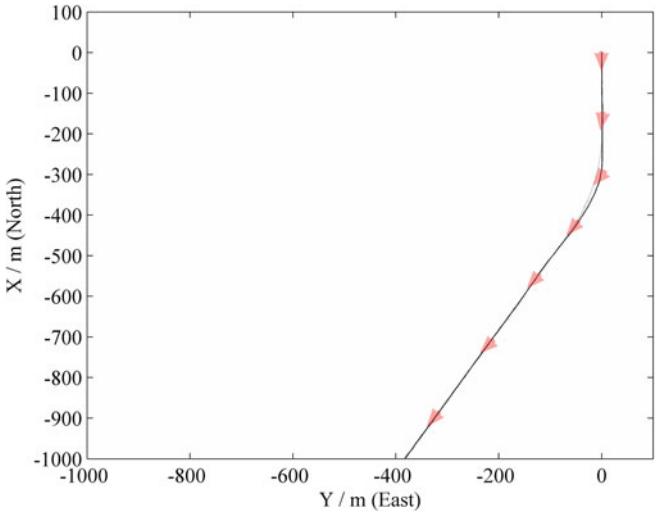

Fig. 4.21 Ship's trajectory of the 30° turning test, whose radius is 500 m; $Q_2 = 600$; $r_1 = 550$; $r_2 = 1{,}000$; *dotted line* setting trajectory, *solid line* real-time tracking control trajectory

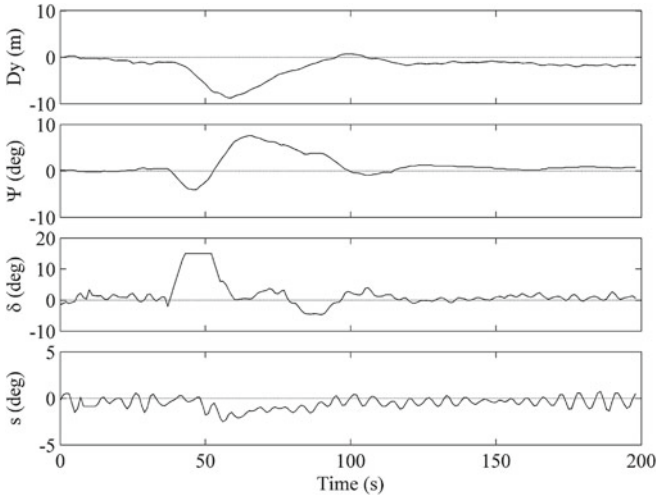

Fig. 4.22 Real-time control results of the 30° turning test: $-15 \leq \delta \leq 15$ and $-3 \leq \Delta\delta \leq 3$

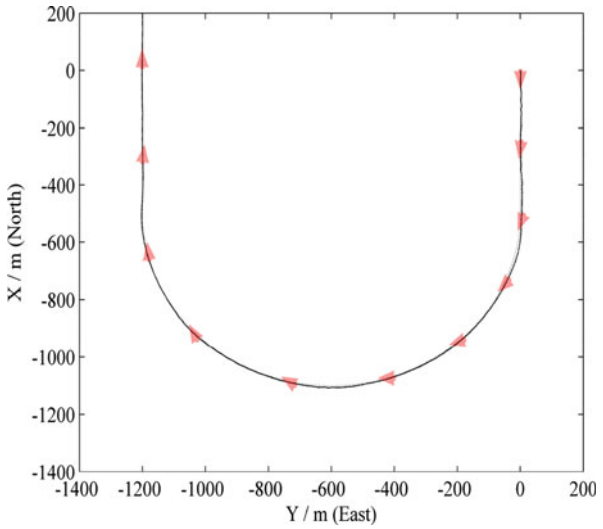

Fig. 4.23 Ship's trajectory of the 180° turning test, the radius of which is 600 m. Controller parameters: $Q_2 = 600$, $r_1 = 550$, and $r_2 = 1{,}000$; *dotted line* setting trajectory, *solid line* real-time tracking control trajectory

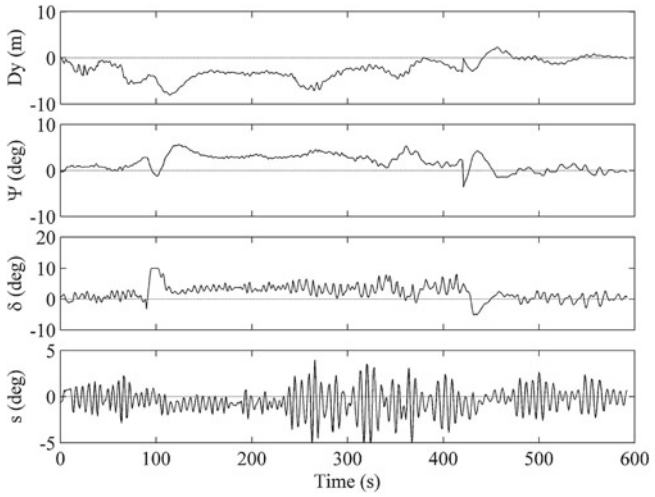

Fig. 4.24 Real-time control results of the 180° turning test: $-10 \leq \delta \leq 10$ and $-3 \leq \Delta\delta \leq 3$

References

Akaike, H., Nakagawa, T.: Statistical Analysis and Control of Dynamic Systems. Kluwer, Dordrecht (1988)

Bloemen, H.H.J., Boom, T.J.V.D., Verbruggen, H.B.: Model-based predictive control for Hammerstein-Wiener systems. Int. J. Control **74**, 482–485 (2001)

Bowditch, N.: The American Practical vol. 9. Navigator, National Imagery and Mapping Agency, (2002)

Do, D., Jiang, Z.P., Pan, J.: Universal controllers for stabilization and tracking of underactuated ships. Syst Control Lett **47**, 299–317 (2002)

Do, K.D., Pan, J.: Global tracking control of underactuated ships with nonzero off-diagonal terms in their system matrices. Automatica **41**, 87–95 (2005)

Du, J., Guo, C.: Nonlinear adaptive ship course tracking control based on backstepping and nussbaum gain. In: Proceeding of the 2004 American Control Conference, pp. 3845–3850 (2004)

Fossen, T.I.: Guidance and Control of Ocean Vehicles. Wiley, New York (1994)

Holzhüter, T.: LQG approach for the high-precision track control of ships. IEE Proc. Control Theory Appl, **144**, 121–127 (1997)

Fukuda, H., Ohtsu, K., Tasaki, T., Okazaki, T.: Study on tracking control system using the time varying gain theory. Soc. Nav. Archit. Jpn. **190** (2001)

Iseki, T., Ohtsu, K.: Bayesian estimation of directional wave spectra based on ship motions. Control Eng. Pract. **8**, 215–219 (2000)

Jiang, Z.P.: Global tracking control of underactuated ships by Lyapunov's direct method. Automatica **38**, 301–309 (2002)

Kothare, M.V.: Robust constrained model predictive control using linear matrix inequalities. Automatica **32**, 1361–1379 (1996)

Kvam, K., Ohtsu, K., Fossen, T.I.: Optimal ship maneuvering using Bryson and Ho's time varying LQ controller. In: Proceedings of the IFAC Conference on Maneuvering and Control of Marine Craft (MCMC'00), Aalborg (2000)

Kwakernaak, H., Sivan, R.: Linear Optimal Control Systems, vol. 1. Wiley, New York (1972)

Lakhdari, Z., Mokhtari, M., Lécluse, Y., Provost, J.: Adaptive predictive control of a class of nonlinear systems—a case study. In: IFAC Proceedings of Adaptive Systems in Control and Signal Processing, 209–214, Budapest, (1995)

Lefeber, E., Pettersen, K.Y., Nijmeijer, H.: Tracking control of an underactuated ship. IEEE Trans. Control Syst. Technol. **11**, 52–61 (2003)

Lewis, E.V.: Principle of Naval Architecture. The Society of Naval Architects and Marine Engineering, Chap. 9 (1967)

Mahfouf, M., Linkens, D.A.: Non-linear generalized predictive control (NLGPC) applied to muscle relaxant anaesthesia. Int. J. Control **71**, 239–257 (1998)

Miyoshi, S., Hara, Y., Ohtsu, K.: Study on optimum tracking control with linearized model for vessel (in Japanese). Jpn. Inst. Navig. **117**, 183–189 (2007)

Mizuno, N., Kuroda, M., Okazaki, T., Ohtsu, K.: Minimum time ship maneuvering method using neural network and nonlinear model predictive compensator. Control Eng. Pract. **15**(6), 757–765 (2007)

Moreiraa, L., Fossenb, T.I., Soares, C.G.: Path following control system for a tanker ship model. Ocean Eng. **34**, 2074–2085 (2007)

Oda, H., Ohtsu, K., Hotta, T.: Statistical analysis and design of a rudder roll stabilization system. Control Eng. Pract. **4**, 351–358 (1996)

Oda, H., Ohtsu, K., Sasaki, M., Seki, Y., Hotta, T.: Roll stabilization by rudder control through multivariate auto-regressive model. J. Kansai Soc. Nav. Archit. **216**, 165–173 (1991)

Ohtsu, K.: Recent development of analysis and control of ship's motions. In: Proceedings of the 1999 IEEE International Conference Control Applications, pp. 1096–1103, Hawaii (1999)

Ohtsu, K., Horigome, M., Kitagawa, G.: A new ship's auto pilot through a stochastic model. Automatica **15–3**, 255–268 (1979)

Ohtsu, K., Kitagawa, G.: Statistical analysis of the AR type ship's autopilot system. J. Dyn. Syst. Meas. Control **106**, 193–202 (1984)
Park, J.S., Ohtsu, K., Kitagawa, G.: Batch-adaptive ship's autopilot. Int. J. Adapt. Control Signal Process. **14**, 427–439 (2000)
Peng, H., Gui, W., Shioya, H., Zou, R.: A predictive control strategy for nonlinear NOx decomposition process in thermal power plants. IEEE Trans. Syst. Man Cybern. Part A **36**(5), 904–921 (2006)
Peng, H., Kitagawa, G., Wu, J., Ohtsu, K.: Multivariable RBF-ARX model-based robust MPC approach and application to thermal power plant. Appl. Math. Model. **35–7**, 3541–3551 (2011)
Peng, H., Nakano, K., Shioya, H.: Nonlinear predictive control using neural nets-based local linearization ARX model—stability and industrial application. IEEE Trans. Control Syst. Technol. **15**, 130–143 (2007)
Peng, H., Ohtsu, K., Kitagawa, G., Oda, H.: A statistical modeling and tracking control approach to marine vehicle. In: 2010 IEEE International Conference on Control Applications (CCA), pp. 640–645 (2010)
Peng, H., Ozaki, T., Haggan-Ozaki, V., Toyoda, Y.: A parameter optimization method for the radial basis function type models. IEEE Trans. Neural Netw. **14**, 432–438 (2003)
Peng, H., Ozaki, T., Toyoda, Y., Shioya, H., Nakano, K., Haggan-Ozaki, V., Mori, M.: RBF-ARX model based nonlinear system modeling and predictive control with application to a NOx decomposition process. Control Eng. Pract. **12**, 191–203 (2004)
Peng, H., Wu, J., Inoussa, G., Deng, Q., Nakano, K.: Nonlinear system modeling and predictive control using RBF nets-based quasi-linear ARX model. Control Eng. Pract. **17**, 59–66 (2009)
Pettersen, K.Y., Nijmeijer, H.: Underactuated ship tracking control: theory and experiments. Int. J. Control **74**, 1435–1446 (2001)
Prasad, G., Swdenbank, E., Hogg, B.W.: A local model networks based multivariable long-range predictive control strategy for thermal power plants. Automatica **34**, 1185–1204 (1998)
Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Eng. Pract. **11**(7), 733–764 (2003)
Qin, Y., Peng, H., Ruan, W., Wu, J., Gao, J.: A modeling and control approach to magnetic levitation system based on state-dependent ARX model. J. Process. Control **24–1**, 93–112 (2014)
Sentoni, G., Agamennoni, O., Desages, A., Romagnoli, J.: Approximate models for nonlinear process control. AIChE J. **42**, 2240–2250 (1996)
Wu, J., Peng, H., Ohtsu, K., Kitagawa, G., Itoh, T.: Ship's tracking control based on nonlinear time series model. Appl. Ocean Res. **36**, 1–11 (2012)
Zeng, X., Peng, H., Wu, J., Wei, J.: Quad-rotor modeling and attitude control using state-dependent ARX type model. Asian J. Control **16–6**, 1–13 (2014)

Index

K. Ohtsu et al., *Time Series Modeling for Analysis and Control*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55303-8