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:

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

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, $\mathrm{AIC}_0, \ldots, \mathrm{AIC}_M$, 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 $\mathrm{AIC}_0 = N(\log 2\pi\hat{\sigma}_0^2 + 1) + 2$.
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) $\mathrm{AIC}_m = N(\log 2\pi\hat{\sigma}_m^2 + 1)\ +\ 2(m+1)$.
3. The AIC best order is defined as the minimizer of $\mathrm{AIC}_0, \ldots, \mathrm{AIC}_M$.

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:

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

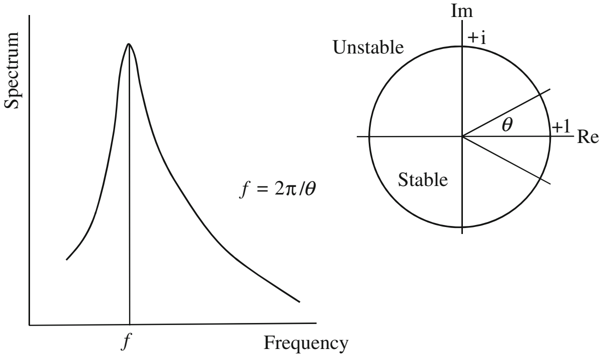


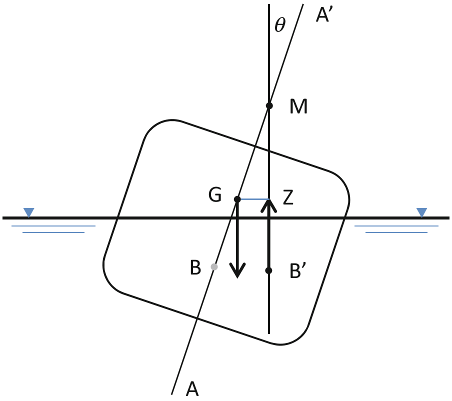
Fig. 2.1 Relationship between the characteristic roots and the spectrum peak frequency

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".

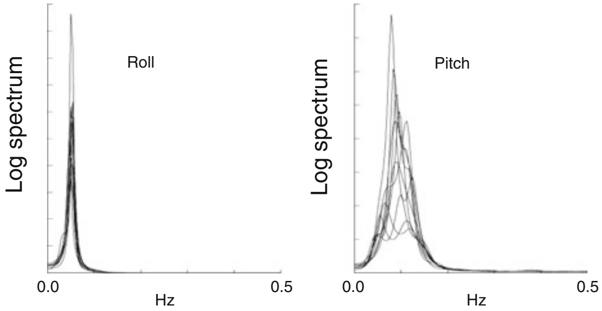


Fig. 2.3 Power spectra of roll (*left*) and pitch (*right*)

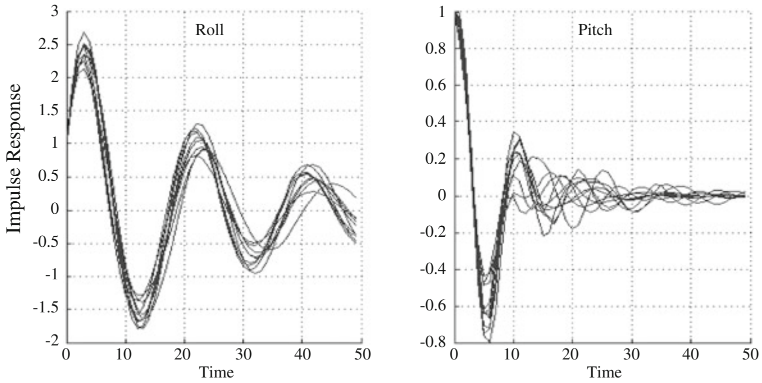


Fig. 2.4 Impulse response of roll (*left*) and pitch (*right*) motions

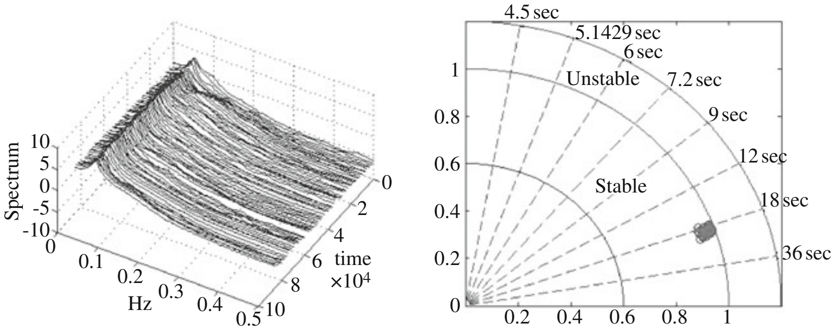


Fig. 2.5 Changes of roll spectra (*left*) and the locations of the dominant characteristic roots (*right*)

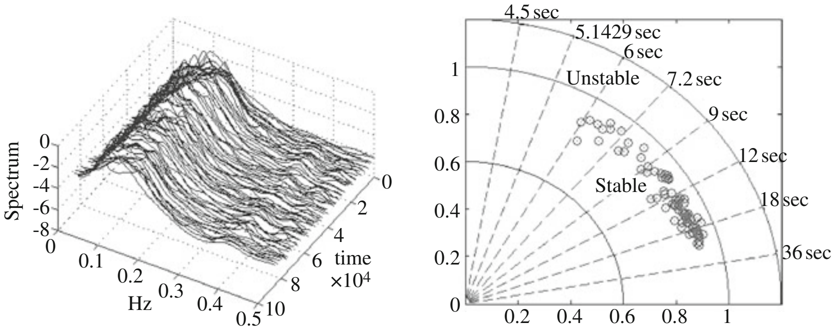


Fig. 2.6 Changes of pitch spectra (*left*) and the locations of the dominant characteristic roots (*right*)

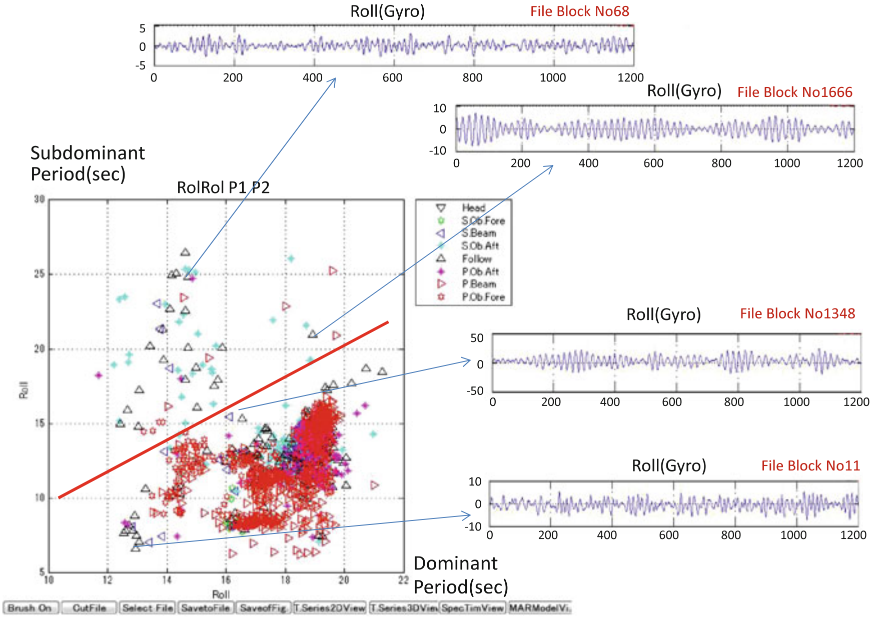


Fig. 2.7 Locations of dominant and sub-dominant periods and patterns of roll

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.

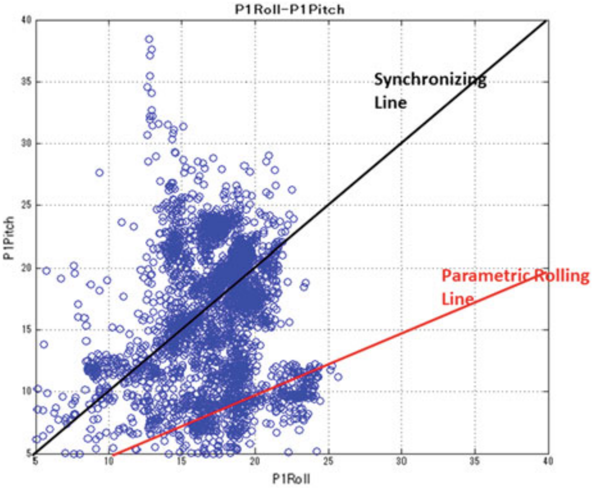


Fig. 2.8 Scatter diagram of the period of roll (*horizontal axis*) and the period of pitch (*vertical axis*)

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, 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 $\boldsymbol{v}_n$

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

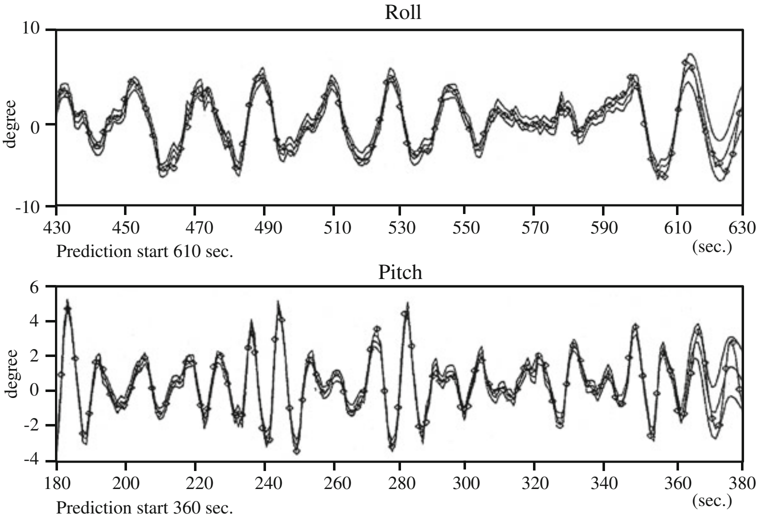


Fig. 2.9 Long-term prediction of roll and pitch. Prediction starts at $n = 610$ for roll and $n = 360$

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),

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

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.,

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

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

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

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 $\text{AIC}_0, \ldots, \text{AIC}_M$, 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

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

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

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

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 $\Delta t = 2$ 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	$\|W_m\|$	AIC$_m$	m	$\|W_m\|$	AIC$_m$	m	$\|W_m\|$	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

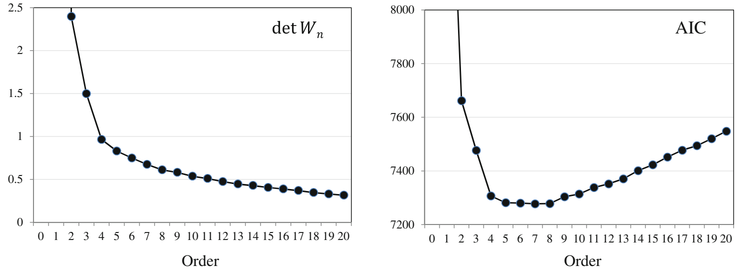


Fig. 2.10 Changes in $|W_m|$ and AIC$_m$ of multivariate AR models for $m = 0, 1, \ldots, 20$, fitted to five-variate ship data under manual control

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

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

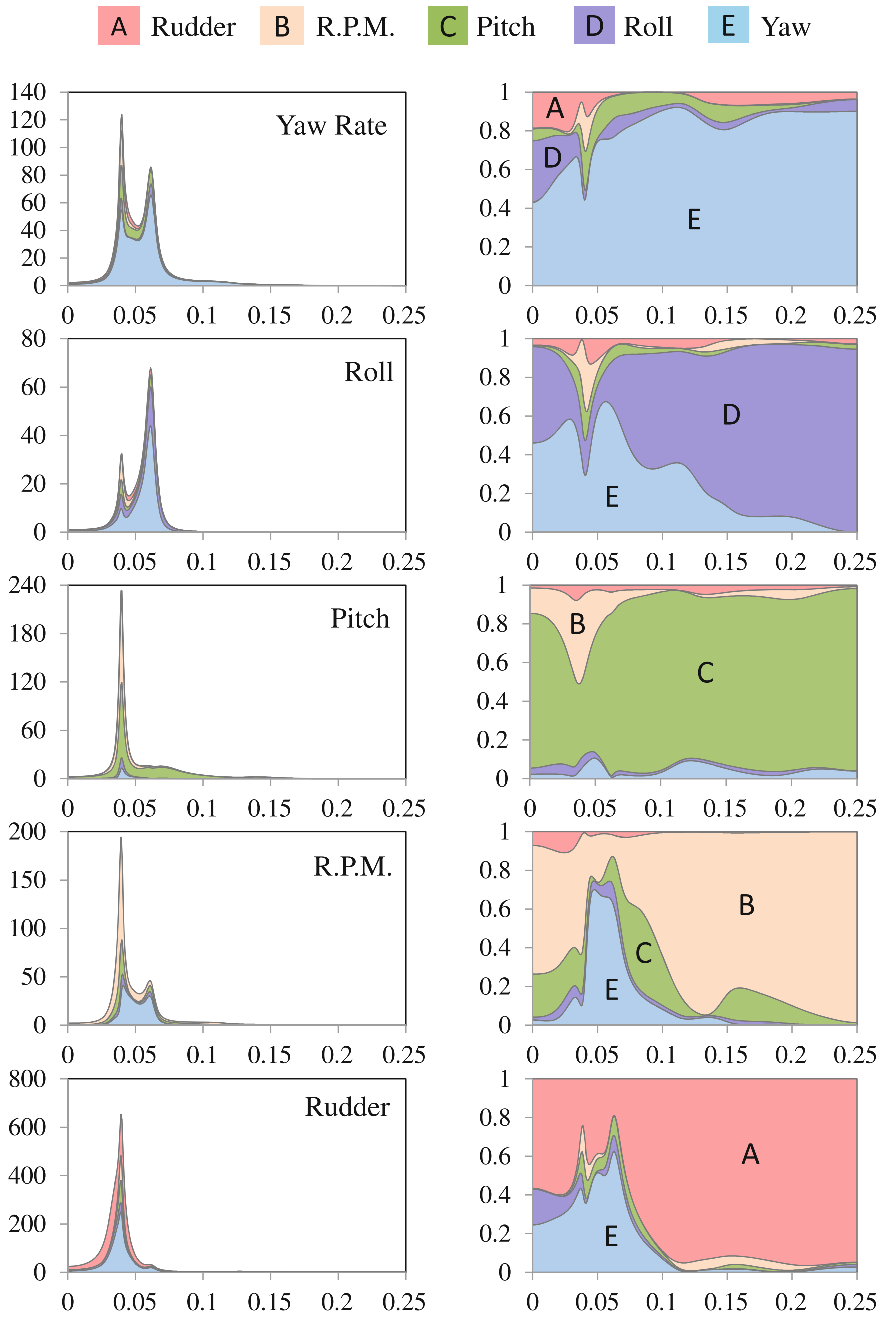


Fig. 2.11 Power contribution analysis of ship motion under manual control

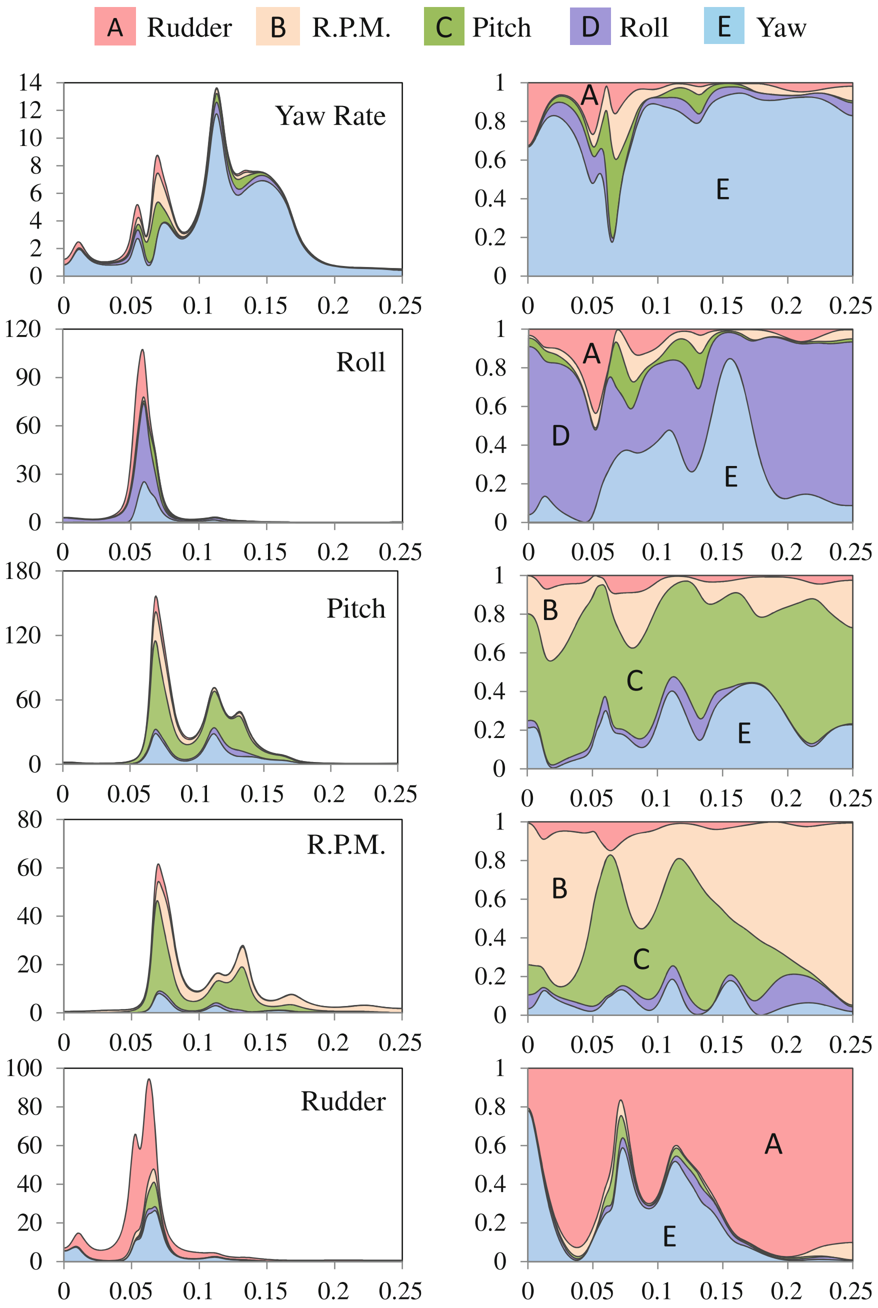


Fig. 2.12 Power contribution analysis of ship motion under autopilot control

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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):

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

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 $\boldsymbol{y}_1, \dots, \boldsymbol{y}_N$ is nonstationary as a whole, but if the time interval $\{1, \dots, N\}$ 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 $m = 1, 2, \dots$ 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, $A_j = \{A_i^{(j)}, \dots, A_{m_j}^{(j)}\}$, and the variance covariance matrix of the white noise, Σ_j.

The likelihood of the locally stationary AR model is given by

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

where $p(\boldsymbol{y}_n | \boldsymbol{y}_1, \dots, \boldsymbol{y}_{n-1})$ denotes the conditional distribution of $\boldsymbol{y}_n$ given the past observations $\boldsymbol{y}_1, \dots, \boldsymbol{y}_{n-1}$. 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

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

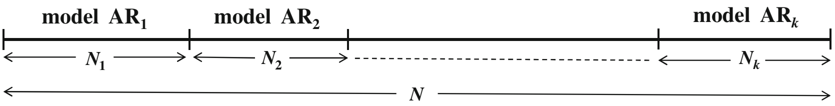


Fig. 2.13 Subdivision of the time interval

Then, the log-likelihood function is given by

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

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

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

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 $y_1, \ldots, y_L$, 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 $y_{n_{k1}+1}, \ldots, y_{n_{k1}+L}$, compute $\mathrm{AIC}_1(0)$, …, $\mathrm{AIC}_1(m)$, 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 $y_{n_{k0}}, \ldots, y_{n_{k1}+L}$ 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 $\mathrm{AIC}_0 = \mathrm{AIC}_P$.

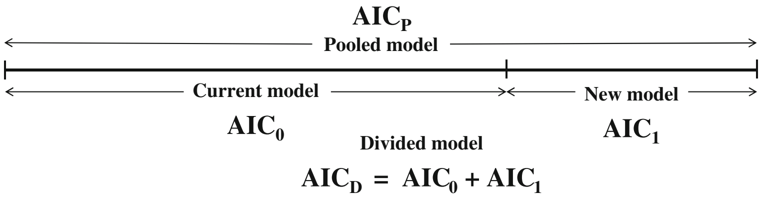


Fig. 2.14 On-line identification of the LSAR model

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		

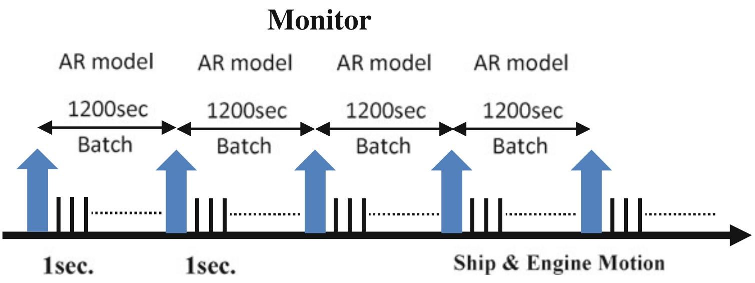


Fig. 2.15 Time schedule for obtaining the model in the monitoring system

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.

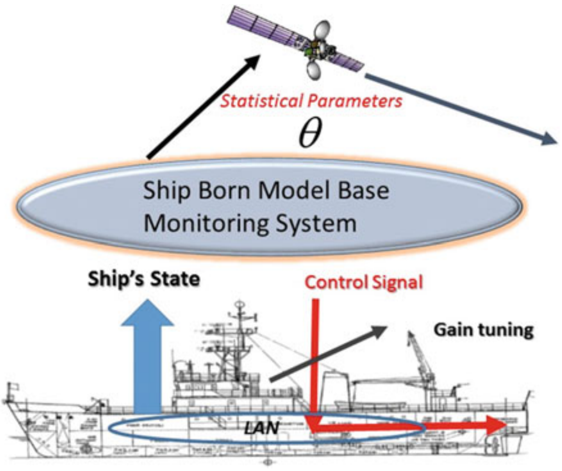


Fig. 2.16 Configuration of the ship-born model-based monitoring system

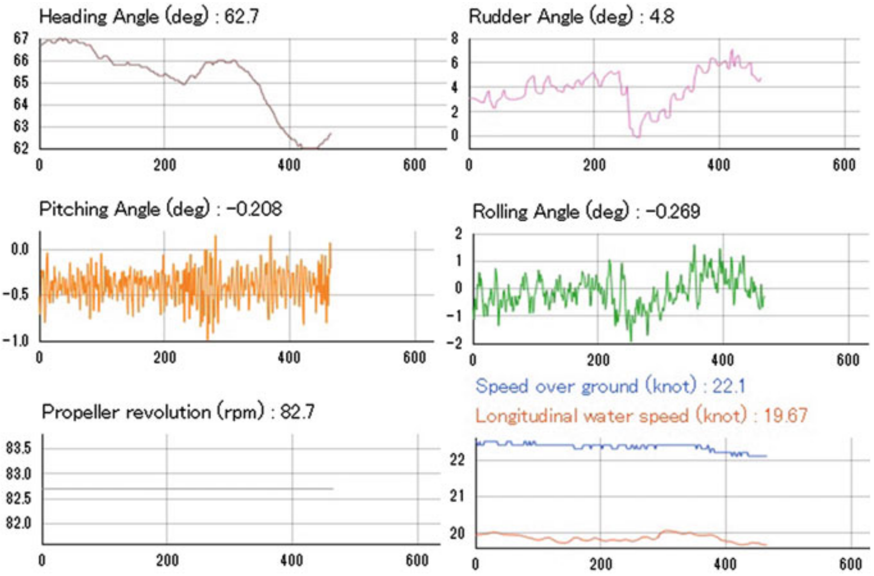


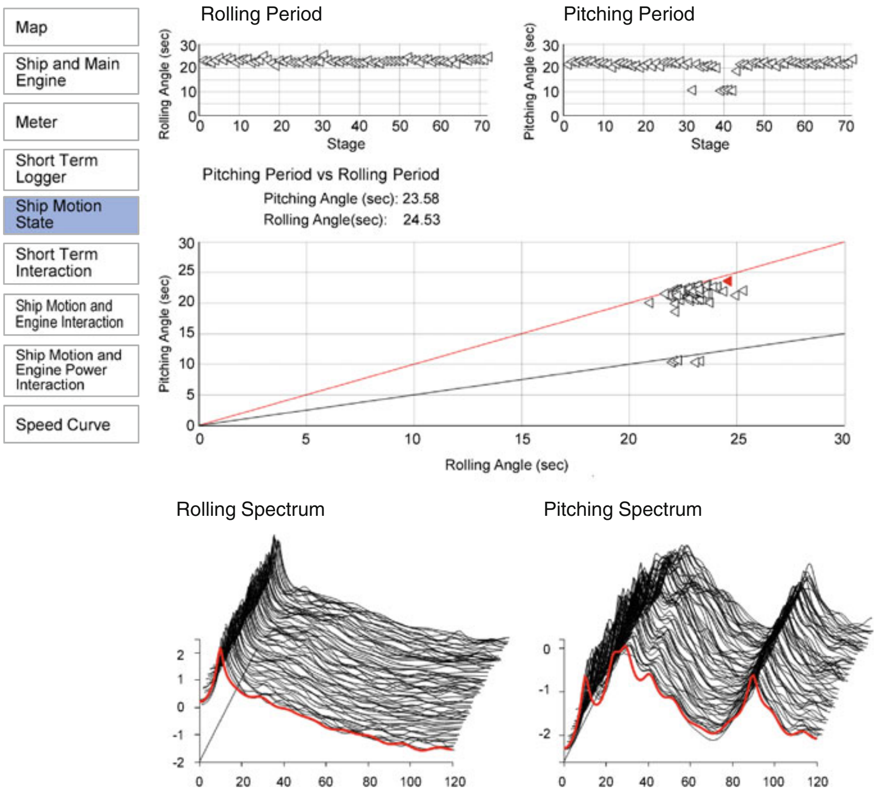
Fig. 2.17 Real-time monitoring of ship and main engine motions (heading angle indicates yaw motion)

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



Pitching Angle (sec): 23.58
Rolling Angle(sec): 24.53

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.

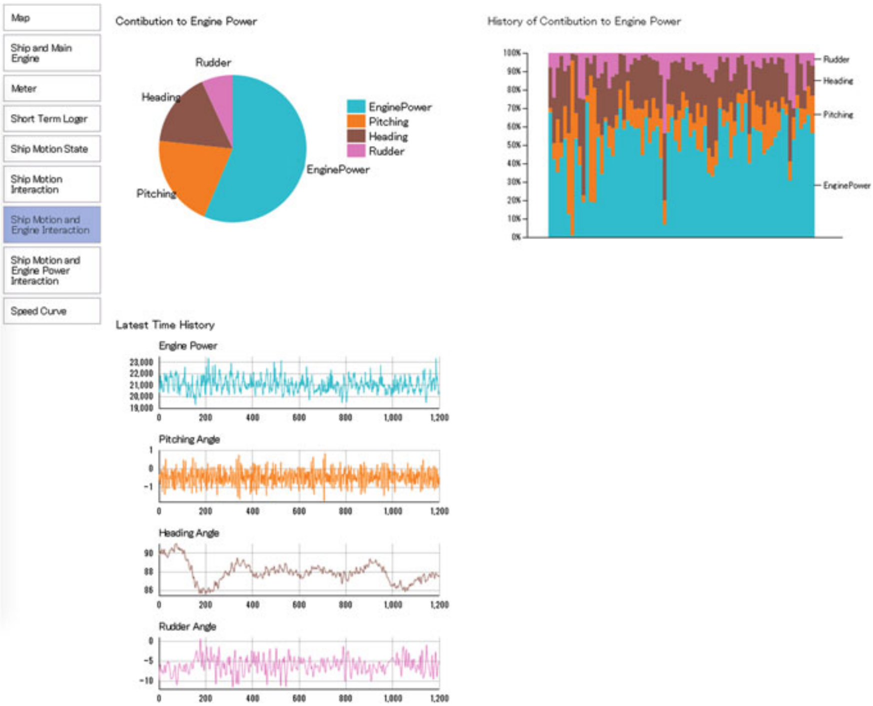


Fig. 2.19 Noise contribution to engine power of engine power, pitch, yaw (heading), and rudder motion

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

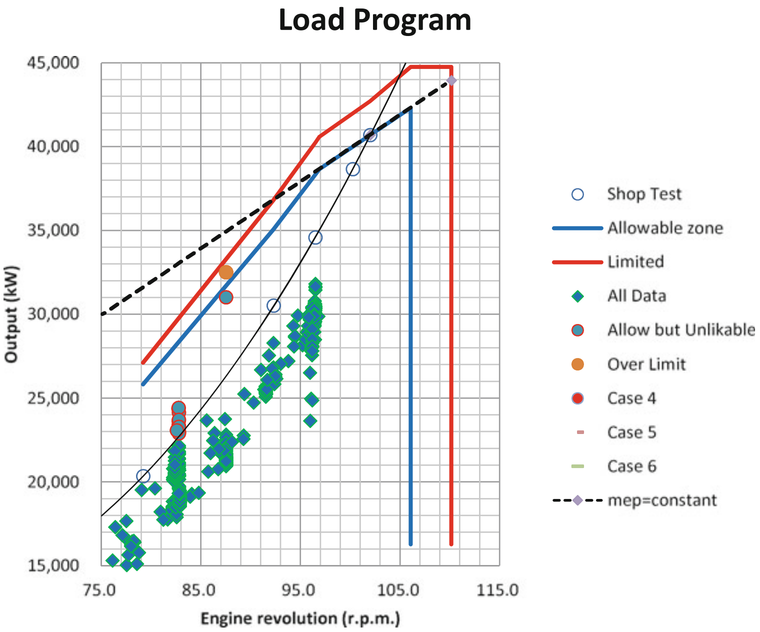


Fig. 2.20 Scattering diagram of the periods of roll and pitch

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

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

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 $\left\{\phi_{y,i}, \phi_{u,i}, \phi_{v,i}\right\}$ 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 computational 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:

$$\begin{aligned}\theta_L &\equiv \left\{c_i^0, c_{k,i}^j | i = 0, 1, \ldots, m; k = 1, 2, \ldots, n_j, j = y, u, v\right\} \\ \theta_N &\equiv \left\{\lambda_1, \ \ldots \ , \lambda_m, \ z_1^T, \ \ldots \ , z_m^T\right\}^T.\end{aligned} \tag{2.89}$$

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 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:

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

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:

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

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

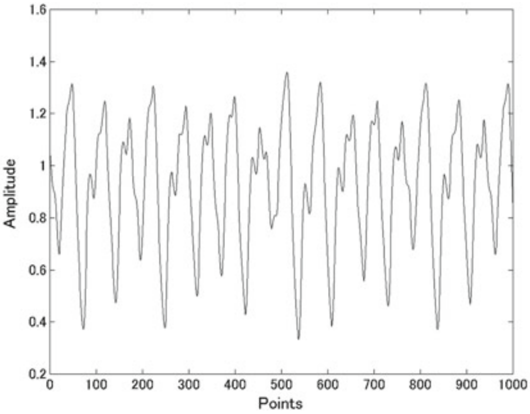


Fig. 2.21 A chaotic series generated from the Mackey-Glass equation

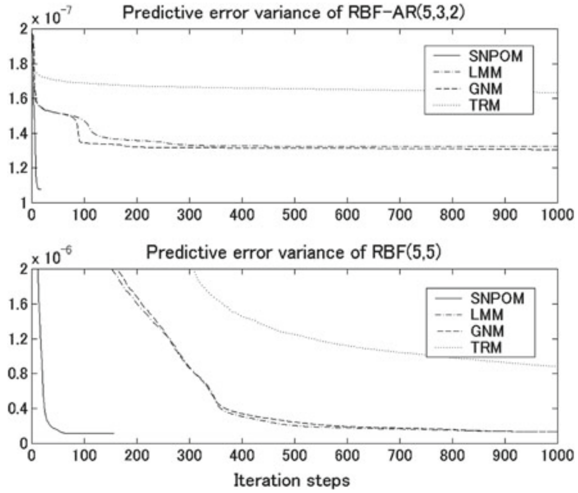


Fig. 2.22 Results for various optimization methods

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

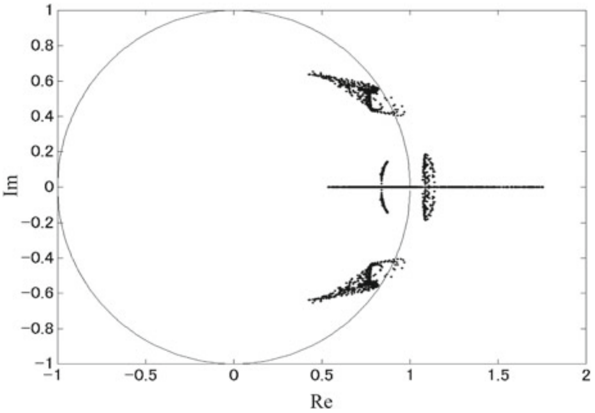


Fig. 2.23 Time-varying eigenvalues of the estimated RBF-AR(5,3,2) model (2.3.22) for the Mackey-Glass series modeling

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)

http://www.springer.com/978-4-431-55302-1

Time Series Modeling for Analysis and Control
Advanced Autopilot and Monitoring Systems
Ohtsu, K.; Peng, H.; Kitagawa, G.
2015, IX, 119 p. 77 illus., 14 illus. in color., Softcover
ISBN: 978-4-431-55302-1