Model Identification And Data Analysis I
Theory

Christian Rossi

Academic Year 2023-2024

Abstract

The course delves into the fundamental concepts of stochastic processes, explores ARMA and ARMAX classes of parametric models for both time series and input-output systems, examines parameter identification techniques for ARMA and ARMAX models, analyzes various identification methods, and addresses model validation and pre-processing.

Contents

CHAPTER 1

Introduction

1.1 Modeling

Definition (*System*)**.** A system denoted by $\mathcal{S}$ refers to a physical entity designed to convert inputs into outputs.

Definition (*Model*)**.** A model, symbolized as $\mathcal{M}$, constitutes a mathematical description of a system.

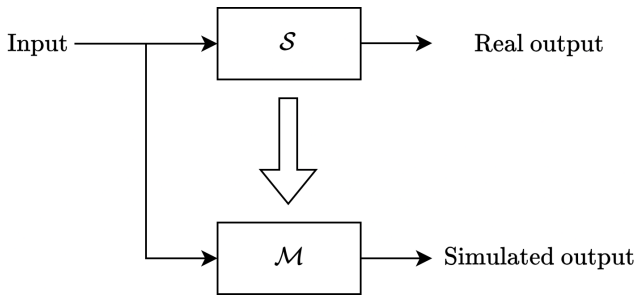


Figure 1.1: Visual representation of system and model

A model can be constructed through various methodologies:

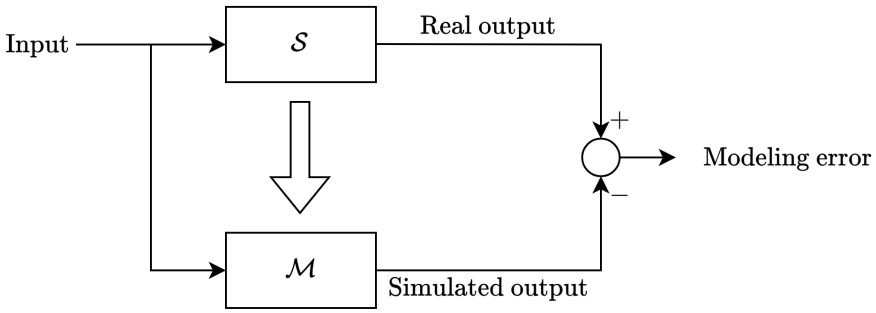
1. *White-box modeling*: this approach relies on established physical laws or existing knowledge. The resultant model is typically generalizable, with clear physical interpretations for each variable. However, precise knowledge of all parameters beforehand is necessary, making it a costly and time-intensive process. Consequently, it's often impractical for complex systems.

2. *Black-box modeling*: this method is based on experimental data. Parameters of the model are estimated using statistical relationships derived from the data. It's feasible even without in-depth knowledge of the underlying processes, and it's comparatively faster and less expensive. However, models generated through this method lack physical interpretability and may not be universally applicable; changes in the system often necessitate repeating the experiment.

3. *Gray-box modeling*: hybrid methodology that commences with equations where only certain parameters are unknown. Its goal is to ascertain these parameters by leveraging both statistical correlations and physical principles. The benefits include the clear interpretation of variable significance and faster model construction compared to white-box approaches. However, a prerequisite for gray-box modeling is the possession of some prior knowledge.

1.2 Modeling error

The modeling error, also known as the residual, is calculated as the disparity between the system output and the model output generated with the same input.



When the outputs exhibit similarity based on certain metrics, it signifies that the model accurately mirrors the dynamics of the system. However, if patterns persist within the error graph, it indicates that not all information has been effectively extracted from the data. Conversely, if the error graph lacks of patterns, it is termed as White Noise, suggesting an inability to extract further meaningful information from the data.

Definition (*Complete model*)**.** A model is deemed complete only when the error demonstrates a completely unpredictable pattern.

1.2.1 Classification

Static and dynamic A system can be categorized as follows:

- *Static system*: in this type of system, knowledge of the input variables alone is adequate to determine the output value.
- *Dynamic system*: this refers to a system with memory, wherein the past behavior of the output impacts its current value.

Discrete and continuous Systems can be further categorized based on their time description, which can be either discrete or continuous. Natural and physical phenomena are inherently continuous and are often mathematically described using ordinary differential equations. On the other hand, discrete systems are mathematically described using difference equations.

However, a computer can only handle a limited amount of data. This necessitates the sampling of signals at discrete intervals with a sampling time T_s. This ensures that only a finite amount of data is stored at discrete time points $t \cdot T_s$, where $t = 1, \ldots, N$:

$$y(t) = y(t \cdot T_s)$$

CHAPTER 2

Stochastic processes

2.1 Introduction

Definition (*Stochastic process*)**.** A stochastic process (SP) is an infinite sequence of random variables, all defined on the same probabilistic space.

A stochastic process can be represented as an infinite sequence:

$$\ldots, v(1,s), v(2,s), v(3,s), \ldots$$

Here, s represents the realization of the random experiment, and it is the same for all elements of the sequence.

In general, each random variable in a sequence is indexed by t, representing the sequence index, and denoted by the realization of the random experiment, s.

When we set $s = \bar{s}$, we obtain a deterministic time signal.

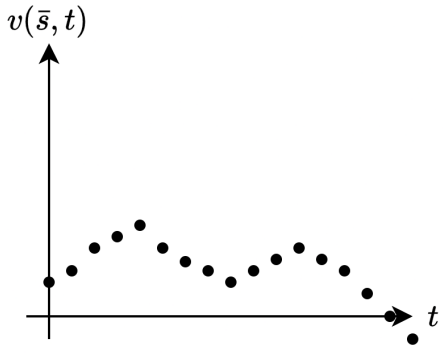


Figure 2.1: Fixed realization

Alternatively, if we constrain the value of time to $t = \bar{t}$, we acquire a simple random variable.

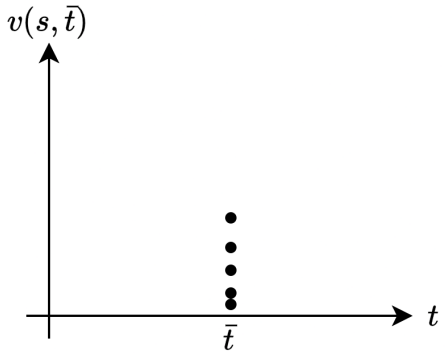


Figure 2.2: Fixed time

Definition (*Signal equivalence*)**.** Two signals are considered equivalent from a stochastic perspective if they are realizations of the same stochastic process.

Wide-sense characterization A stochastic process is fully characterized by the probability distribution of $v(t,s)$. Wide-sense characterization is the description of a stochastic process solely through its mean and covariance functions.

2.1.1 Mean value

The mean value of a stochastic process $v(t,s)$ is defined as:

$$m(t) = \mathbb{E}\left[v(t,s)\right] = \int_{\mathrm{P}} v(t,s) pdf(s)\, ds$$

Graphically, the mean value can be computed by inspecting each time instant and determining the average point for each.

2.1.2 Covariance function

The covariance of a stochastic process $v(t,s)$ is defined as:

$$\gamma(t_1,t_2) = \mathbb{E}\left[\left(v(t_1,s) - m(t_1)\right)\left(v(t_2,s) - m(t_2)\right)\right]$$

This function expresses the degree of correlation between two points at time instants t_1 and t_2.

Variance When the two time instants coincide ($t_1 = t_2 = t$), we have:

$$\gamma(t,t) = \mathbb{E}\left[\left(v(t,s) - m(t)\right)^2\right] = \mathrm{Var}\left[v(t,s)\right] = \gamma(\bar{s})$$

This function is referred to as the variance, and it contains information about the variability of the stochastic process around its mean.

2.2 Stationary stochastic process

Definition (*Stationary stochastic process*)**.** A stationary stochastic process (in a wide sense) is a stochastic process characterized by the following properties:

- The mean must be constant:
$$m(t) = m \qquad \forall t$$
- The covariance is dependent solely on $\tau = t_2 - t_1$:
$$\gamma(t_1, t_2) = \gamma(t_2 - t_1) = \gamma(\tau)$$

The properties of this function are:

1. The variance is non-negative:
$$\gamma(0) = \mathbb{E}\left[(v(t) - m)^2\right] \geq 0$$
2. The covariance is less or less or equal than the variance:
$$|\gamma(\tau)| \leq \gamma(0) \qquad \forall \tau$$
3. The function is even:
$$\gamma(\tau) = \gamma(-\tau) \qquad \forall \tau$$

Property 2.2.1. Given a stationary stochastic process $v(t, s)$, we denote its mean and covariance function as m_v and $\gamma_v(\tau)$, respectively.

Property 2.2.2. Two stationary stochastic processes $v_1(t, s)$ and $v_2(t, s)$ are wide-sense equivalent if $m_{v_1} = m_{v_2}$ and $\gamma_{v_1}(\tau) = \gamma_{v_2}(\tau)$ for all τ.

Definition (*Correlation function*)**.** The correlation function is defined as:
$$\mathbb{E}\left[v(t, s)v(t - \tau, s)\right]$$

Example:
Consider the process $v(t, s) = \alpha(s)$, where $\alpha(s) \sim \mathcal{N}(1, 3)$.

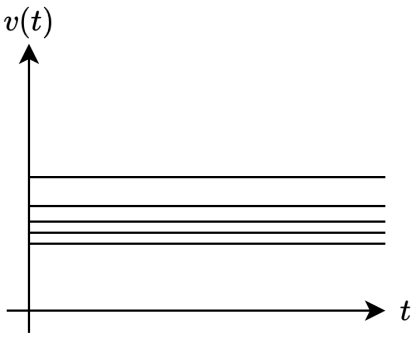


Figure 2.3: Some possible realizations

All the realizations are constants and are more frequent near the value one.
To determine if this is a stationary stochastic process, we need to verify:

1. Constant mean:
$$m(t) = \mathbb{E}\left[v(t, s)\right] = 1$$

2. The covariance must be a function of only τ:

$$\begin{aligned}\gamma(t, t-\tau) &= \mathbb{E}\left[(v(t,s) - m(t))\,(v(t-\tau,s) - m(t))\right] \\ &= \mathbb{E}\left[(\alpha(s) - 1)\,(\alpha(s) - 1)\right] \\ &= 3\end{aligned}$$

Both functions are t-invariant, thus the process is weakly stationary.

Example:
Consider the process $v(t,s) = t\alpha(s) - t$, where $\alpha(s) \sim \mathcal{N}(1,3)$.

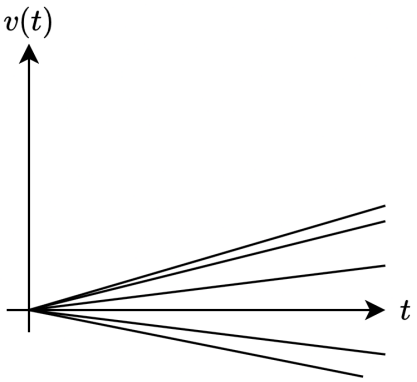


Figure 2.4: Some possible realizations

All the realizations are not constants, but they are more frequent near the value one. To determine if this is a stationary stochastic process, we need to verify:

1. Constant mean:

$$m(t) = \mathbb{E}\left[v(t,s)\right] = \mathbb{E}\left[t\alpha(s) - t\right] = t\mathbb{E}\left[\alpha(s)\right] - t = t - t = 0$$

2. The covariance must be a function of only τ:

$$\begin{aligned}\gamma(t, t-\tau) &= \mathbb{E}\left[(v(t,s) - m(t))\,(v(t-\tau,s) - m(t))\right] \\ &= \mathbb{E}\left[(t\alpha(s) - t - m(t))\,((t-\tau)\alpha(s) - (t-\tau) - m(t))\right] \\ &= \mathbb{E}\left[(t\alpha(s) - t)\,((t-\tau)\alpha(s) - (t-\tau))\right] \\ &= \mathbb{E}\left[t\,(\alpha(s) - 1)\,(t-\tau)\,(\alpha(s) - 1)\right] \\ &= (t^2 - t\tau)\mathbb{E}\left[(\alpha(s) - 1)^2\right] \\ &= 3(t^2 - t\tau)\end{aligned}$$

The covariance function is t-variant, thus the process is not weakly stationary.

2.3 White Noise

Definition (*White Noise*)**.** A stationary stochastic process is termed White Noise WN with mean μ and variance λ^2 if it satisfies the following conditions:

1. The mean is constant:
$$\mathbb{E}\left[e(t)\right] = \mu \qquad \forall t$$
2. The variance is equal to λ^2:
$$\mathrm{Var}\left[e(t)\right] = \gamma_e(0) = \lambda^2$$
3. The covariance is null:
$$\gamma_e(\tau) = \mathbb{E}\left[\left(e(t) - \mu\right)\left(e(t-\tau) - \mu\right)\right] = 0 \qquad \forall t, \tau \neq 0$$

Consequently, the realizations of $e(t)$ exhibit erratic and unpredictable behavior.

The probability distribution of each individual random variable $e(\bar{t}, s)$ is not explicitly specified. In the case of a Gaussian distribution, the White Noise is denoted as WGN.

While a realization of White Noise with a constant value over time is technically feasible, it is highly improbable.

2.4 Moving Average process

Let $e(t) \sim WN(0, \lambda^2)$, and consider the process defined as:

$$y(t) = c_0 e(t) + c_1 e(t-1) + c_2 e(t-2) + \cdots + c_n e(t-n)$$

The process $y(t)$ is termed a Moving Average process of order n, denoted as MA(n).

2.4.1 Mean value

We begin by investigating the mean of the considered process:

$$\begin{aligned} m_y(t) &= \mathbb{E}\left[y(t)\right] \\ &= \mathbb{E}\left[c_0 e(t) + c_1 e(t-1) + c_2 e(t-2) + \cdots + c_n e(t-n)\right] \\ &= c_0 \underbrace{\mathbb{E}\left[e(t)\right]}_{0} + c_1 \underbrace{\mathbb{E}\left[e(t-1)\right]}_{0} + c_2 \underbrace{\mathbb{E}\left[e(t-2)\right]}_{0} + \cdots + c_n \underbrace{\mathbb{E}\left[e(t-n)\right]}_{0} \\ &= 0 \end{aligned}$$

Thus, it is constant.

2.4.2 Covariance function

The covariance must be a function of only τ:

$$\begin{aligned} \gamma_y(t,t) &= \mathbb{E}\left[\left(y(t) - m(t)\right)^2\right] \\ &= \mathbb{E}\left[\left(y(t)\right)^2\right] \\ &= \mathbb{E}\left[\left(c_0 e(t) + c_1 e(t-1) + c_2 e(t-2) + \cdots + c_n e(t-n)\right)^2\right] \end{aligned}$$

Upon computation, we obtain some square terms and some cross terms. However, the cross terms comprise a constant part and the correlation between the White Noise at two time instants, which is zero. Consequently, the only non-null terms are the squares:

$$\begin{aligned}\gamma_y(t,t) &= \mathbb{E}\left[\left(c_0^2 e(t)^2 + c_1^2 e(t-1)^2 + c_2^2 e(t-2)^2 + \cdots + c_n^2 e(t-n)^2\right)\right] \\ &= \mathbb{E}\left[c_0^2 e(t)^2\right] + \mathbb{E}\left[c_1^2 e(t-1)^2\right] + \mathbb{E}\left[c_2^2 e(t-2)^2\right] + \cdots + \mathbb{E}\left[c_n^2 e(t-n)^2\right] \\ &= c_0^2\mathbb{E}\left[e(t)^2\right] + c_1^2\mathbb{E}\left[e(t-1)^2\right] + c_2^2\mathbb{E}\left[e(t-2)^2\right] + \cdots + c_n^2\mathbb{E}\left[e(t-n)^2\right]\end{aligned}$$

Since $\mathbb{E}\left[(e(t)-\mu)^2\right] = \mathbb{E}\left[(e(t))^2\right] = \lambda^2$, we have:

$$\gamma_y(t,t) = \left(c_0^2 + c_1^2 + c_2^2 + \cdots + c_n^2\right)\lambda^2$$

If $y(t)$ is an MA(n) process, it is a stationary stochastic process, and its covariance function is given by:

$$\gamma_y(\tau)\begin{cases}\left(c_0^2 + c_1^2 + \cdots + c_n^2\right)\lambda^2 & \text{if } \tau = 0 \\ \left(c_0c_1 + c_1c_2 + \cdots + c_{n-1}c_n\right)\lambda^2 & \text{if } \tau = \pm 1 \\ \left(c_0c_2 + c_1c_3 + \cdots + c_{n-2}c_n\right)\lambda^2 & \text{if } \tau = \pm 2 \\ \vdots & \\ c_0c_n\lambda^2 & \text{if } \tau = \pm n \\ 0 & \text{if } |\tau| > n\end{cases}$$

By combining different samples of $e(t)$, we can create a process with n non-zero components on the positive τ-axis. Importantly, these components do not depend on time t, ensuring the process remains stationary.

2.4.3 Infinite order Moving Average

We can extend the concept of MA(n) processes by considering an infinite order:

$$y(t) = \sum_{i=0}^{+\infty} c_i e(t-i) \qquad e(t) \sim WN(0,\lambda^2)$$

To define this process, we need to assume that:

$$\sum_{i=0}^{+\infty} c_i^2 < +\infty$$

Mean value We can calculate the mean as follows:

$$m_y(t) = \mathbb{E}\left[y(t)\right] = \mathbb{E}\left[\sum_{i=0}^{+\infty} c_i e(t-i)\right] = \sum_{i=0}^{+\infty} c_i \underbrace{\mathbb{E}\left[e(t-i)\right]}_{0} = 0$$

Variance function We can compute the variance $\gamma_y(t,t)$ as follows:

$$\begin{aligned}
\gamma_y(t,t) &= \mathbb{E}\left[(y(t) - m_y(t))^2\right] \\
&= \mathbb{E}\left[(y(t))^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=0}^{+\infty} c_i e(t-i)\right)\left(\sum_{j=0}^{+\infty} c_j e(t-j)\right)\right] \\
&= \mathbb{E}\left[\sum_{i=0,j=0}^{+\infty} c_i c_j e(t-i)e(t-j)\right] \\
&= \sum_{i=0,j=0}^{+\infty} c_i c_j \mathbb{E}\left[e(t-i)e(t-j)\right] \\
&= \sum_{i=0}^{+\infty} c_i^2 \lambda^2
\end{aligned}$$

Similarly, we can compute the covariance $\gamma_y(t, t-\tau)$ as:

$$\begin{aligned}
\gamma_y(t,t-\tau) &= \mathbb{E}\left[(y(t) - m_y(t))\,(y(t-\tau) - m_y(-\tau))\right] \\
&= \mathbb{E}\left[y(t)y(t-\tau)\right] \\
&= \mathbb{E}\left[\left(\sum_{i=0}^{+\infty} c_i e(t-i)\right)\left(\sum_{j=0}^{+\infty} c_j e(t-j-\tau)\right)\right] \\
&= \mathbb{E}\left[\sum_{i=0,j=0}^{+\infty} c_i c_j e(t-i)e(t-j-\tau)\right] \\
&= \sum_{i=0,j=0}^{+\infty} c_i c_j \mathbb{E}\left[e(t-i)e(t-j-\tau)\right] \\
&= \sum_{i=0}^{+\infty} c_i c_{i+\tau} \lambda^2
\end{aligned}$$

Given that the mean value is constant, and the covariance does not vary with time t, we conclude that the Moving Average process with an infinite order is also a stationary stochastic process.

This model can be applied to represent almost all stationary processes, with a few exceptions. However, such models pose challenges due to their infinite degrees of freedom and the computation of covariances, which necessitates handling infinite series.

2.5 AutoRegressive process

AutoRegressive models enable us to achieve non-zero auto-covariance ($\gamma(\tau) \neq 0$) using a finite set of coefficients. These models are defined as follows:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + e(t) \qquad e(t) \sim WN(\mu, \lambda^2)$$

Here, $a_1, a_2, \ldots, a_m$ represent the AutoRegressive process coefficients, and m denotes the model order. The order of an AutoRegressive model is denoted as AR(m).

To ensure that an AutoRegressive model is stationary, we consider cases where they represent the steady-state solution of the difference equation.

Example:
Let's consider the AR(1) process described by the equation:

$$y(t) = ay(y-1) + e(t) \qquad e(t) \sim WN(\mu, \lambda^2)$$

We aim to determine the steady-state solution of this process. It is possible to rewrite the process as:

$$y(t) = \sum_{i=0}^{+\infty} a^i e(t-1)$$

This form resembles a Moving Average with an infinite order, with coefficients $c_0 = 1, c_1 = a, c_2 = a^2, \ldots, c_i = a^i$.

Thus, we've discovered that an AutoRegressive model in the steady-state is equivalent to a Moving Average process with infinite order.

2.5.1 Operatorial representation

Let's consider the AR(1) process described by the equation:

$$y(t) = ay(t-1) + e(t) \qquad e(t) \sim WN(0, \lambda^2)$$

We can employ operatorial representation to derive the transfer function:

$$y(t)\left(1 - az^{-1}\right) = e(t) \rightarrow y(t) = \frac{z}{z-a}e(t)$$

Stability is ensured if the transfer function is asymptotically stable, and the input process is a stochastic stationary process. By definition, White Noise is a stationary stochastic process. The transfer function is asymptotically stable if $|a| < 1$. Hence, if $|a| < 1$, $y(t)$ constitutes a stochastic stationary process.

From AutoRegressive to Moving Average Now, let's derive the MA(∞) corresponding to the given AutoRegressive model through recursive application of the difference equation or long division. The transfer function is:

$$W(z) = \frac{1}{1 - az^{-1}} = \frac{C(z)}{A(z)}$$

To perform long division, we divide $C(z)$ by $A(z)$. After an infinite number of steps, we obtain:

$$W(z) = 1 + az^{-1} + a^2 z^{-2} + \ldots$$

This result aligns with what we obtained through difference equations, implying that the following condition must hold:

$$\sum_{i=0}^{+\infty} \left|a^2\right|^i < +\infty$$

However, this condition is automatically satisfied if $|a| < 1$ due to the geometric series.

We have once again demonstrated the relationship between AR(1) and MA(∞) processes. However, computing m_y and $\gamma_y(\tau)$ for all τ using the same tools as those for Moving Average processes is impractical.

2.5.2 Mean value

Let's analyze the AR(1) process described by the equation:

$$y(t) = ay(t-1) + e(t) \qquad e(t) \sim WN(0, \lambda^2), |a| < 1$$

We aim to compute the mean value, which is constant because the process is stationary:

$$m_y = \mathbb{E}[y(t)] = \mathbb{E}[ay(t-1) + e(t)] = a\mathbb{E}[y(t-1)] + \underbrace{\mathbb{E}[e(t)]}_{0}$$

Given that it is a stationary stochastic process, we can state that $\mathbb{E}[\tau] = m_y$ for all τ, leading to:

$$m_y = am_y \rightarrow (1-a)m_y = 0 \rightarrow m_y = 0$$

2.5.3 Covariance function

Let's examine the AR(1) process defined by the function:

$$y(t) = ay(t-1) + e(t) \qquad e(t) \sim WN(0, \lambda^2), |a| < 1$$

We'll begin by computing the covariance function at $\tau = 0$, given $m_y = 0$:

$$\begin{aligned}
\gamma_y(0) &= \mathbb{E}[(y(t) - m_y)(y(t) - m_y)] \\
&= \mathbb{E}\left[(y(t))^2\right] \\
&= \mathbb{E}\left[(ay(t-1) + e(t))^2\right] \\
&= \mathbb{E}\left[a^2 y(t-1)^2 + e(t)^2 + 2ay(t-1)e(t)\right] \\
&= a^2 \underbrace{\mathbb{E}\left[y(t-1)^2\right]}_{\gamma_y(0)} + \underbrace{\mathbb{E}\left[e(t)^2\right]}_{\lambda^2} + 2a \underbrace{\mathbb{E}[y(t-1)e(t)]}_{0} \\
&= a^2\gamma_y(0) + \lambda^2
\end{aligned}$$

Solving $\gamma_y(0) = a^2\gamma_y(0) + \lambda^2$ yields:

$$\gamma_y(0) = \frac{\lambda^2}{1-a^2}$$

Next, we'll compute the covariance function at $\tau = 1$:

$$\begin{aligned}
\gamma_y(1) &= \mathbb{E}[(y(t) - m_y)(y(t-1) - m_y)] \\
&= \mathbb{E}[(ay(t-1) + e(t))\, y(t-1)] \\
&= \mathbb{E}\left[ay(t-1)^2 + e(t)y(t-1)\right] \\
&= a\underbrace{\mathbb{E}\left[y(t-1)^2\right]}_{\gamma_y(0)} + \underbrace{\mathbb{E}[e(t)y(t-1)]}_{0} \\
&= a\gamma_y(0) \\
&= a\frac{\lambda^2}{1-a^2}
\end{aligned}$$

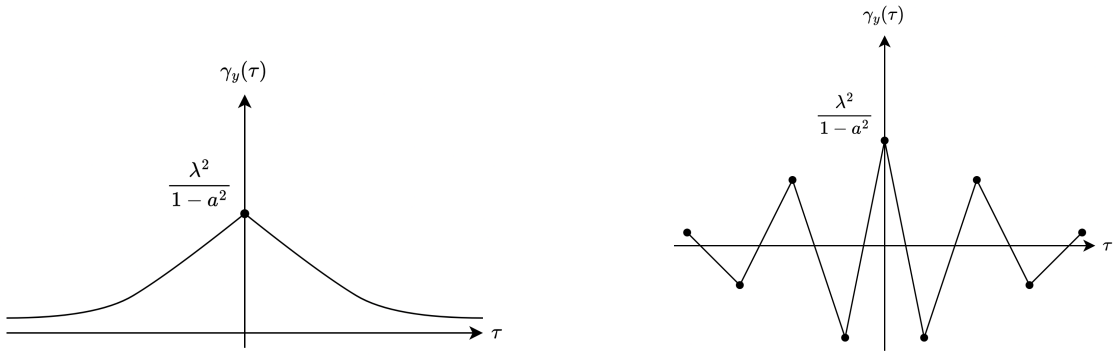
Next, we'll compute the covariance function at $\tau = 2$:

$$\begin{aligned}
\gamma_y(2) &= \mathbb{E}\left[(y(t) - m_y)\,(y(t-2) - m_y)\right] \\
&= \mathbb{E}\left[(ay(t-1) + e(t))\,y(t-2)\right] \\
&= \mathbb{E}\left[ay(t-1)y(t-2) + e(t)y(t-2)\right] \\
&= a\underbrace{\mathbb{E}\left[y(t-1)y(t-2)\right]}_{\gamma_y(1)} + \underbrace{\mathbb{E}\left[e(t)y(t-2)\right]}_{0} \\
&= a\gamma_y(1) \\
&= \frac{a^2\lambda^2}{1-a^2}
\end{aligned}$$

This pattern continues for $\tau > 1$:

$$\gamma_y(\tau) = a^\tau \frac{\lambda^2}{1-a^2}$$

The resulting covariance function for $\tau \geq 0$ is a set of recursive equations, known as the Yule-Walker equations for the AutoRegressive of order one process. These equations describe how each covariance depends on the previous one.



(a) Parameter a positive (b) Parameter a negative

Figure 2.5: Covariance of a stable AR(1) process

2.5.4 White Noise and AutoRegressive covariance

We have derived the covariance of the AutoRegressive process by computing:

$$\mathbb{E}\left[e(t)y(t-\tau)\right] = 0 \qquad \tau > 0$$

To demonstrate the validity of this formula, let's begin by examining an MA(∞) process, which is equivalent to the AutoRegressive process considered:

$$y(t) = e(t) + ae(t-1) + a^2e(t-2) + \dots$$

Now, let's compute the formula for $\tau = 1$:

$$\begin{aligned}
\mathbb{E}\left[e(t)y(t-1)\right] &= \mathbb{E}\left[e(t)\left(e(t-1) + ae(t-2) + a^2e(t-3) + \dots\right)\right] \\
&= \mathbb{E}\left[e(t)e(t-1) + ae(t)e(t-2) + a^2e(t)e(t-3) + \dots\right] \\
&= \mathbb{E}\left[e(t)e(t-1)\right] + a\mathbb{E}\left[e(t)e(t-2)\right] + a^2\mathbb{E}\left[e(t)e(t-3)\right] + \dots
\end{aligned}$$

Since all expected values involve the White Noise at different time instants, and since the White Noise is unpredictable, all these terms become null. Hence, we have:

$$\mathbb{E}\left[e(t)y(t-1)\right] = 0 + a \cdot 0 + a^2 \cdot 0 = 0$$

Similarly, if we consider the same for any $\tau > 1$, we obtain the same result. Therefore, the formula holds true. This demonstrates that the covariance expression is valid for the AutoRegressive process.

2.6 ARMA process

We can define a process that combines elements from both the AutoRegressive and Moving Average processes as follows:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + c_0 e(t) + c_1 e(t-1) + \cdots + c_n e(t-n)$$

Here, $e(t) \sim WN(0, \lambda^2)$.

This process comprises two components: one being the AR(m) process and the other being the MA(n) process. It is denoted as an ARMA(m, n) process. Since it combines elements from two distinct processes, we can make the following observations:

- The MA(n) process is equivalent to an ARMA($0, n$) process.
- The AR(m) process is equivalent to an ARMA($m, 0$) process.

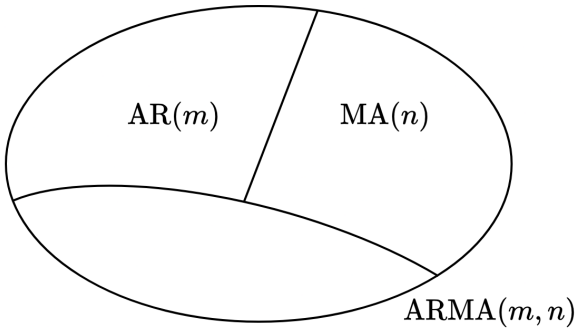


Figure 2.6: Inclusion of ARMA processes

2.6.1 Operatorial representation

Given an ARMA(m, n) process defined as:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + c_0 e(t) + c_1 e(t-1) + \cdots + c_n e(t-n)$$

Here, $e(t) \sim WN(0, \lambda^2)$.

We can rewrite it using operatorial representation as:

$$y(t) = a_1 z^{-1} y(t) + a_2 z^{-2} y(t) + \cdots + a_m z^{-m} y(t) + c_0 e(t) + c_1 z^{-1} e(t) + \cdots + c_n z^{-n} e(t)$$

Rearranging terms, we get:

$$\left(1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_m z^{-m}\right) y(t) = \left(c_0 + c_1 z^{-1} + \cdots + c_n z^{-n}\right) e(t)$$

Finally, we can express $y(t)$ in terms of $e(t)$ using the transfer function notation:

$$y(t) = \frac{c_0 + c_1 z^{-1} + \cdots + c_n z^{-n}}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_m z^{-m}} e(t) = \frac{C(z)}{A(z)} e(t)$$

Here, $C(z)$ and $A(z)$ are polynomials representing the coefficients of the Moving Average and AR parts, respectively. This ratio, denoted as $W(z)$, is a discrete-time transfer function. This operator acts as a digital filter, transforming the input noise sequence $e(t)$ into the output sequence $y(t)$.

2.6.2 Mean value

Given an ARMA(m, n) model defined as:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + c_0 e(t) + c_1 e(t-1) + \cdots + c_n e(t-n)$$

where $e(t) \sim WN(0, \lambda^2)$, we can compute its mean as follows:

$$\begin{aligned}\mathbb{E}[y(t)] &= \mathbb{E}[a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n)] \\ &= a_1 \mathbb{E}[y(t-1)] + \cdots + a_m \mathbb{E}[y(t-m)] + c_0 \underbrace{\mathbb{E}[e(t)]}_{0} + \cdots + c_n \underbrace{\mathbb{E}[e(t-n)]}_{0}\end{aligned}$$

Under the assumption that $a_1, a_2, \ldots, a_m$ are chosen such that $y(t)$ constitutes a stationary stochastic process, thus exhibiting a constant mean value, we can simplify the above to:

$$m_y = a_1 m_y + a_2 m_y + \cdots + a_m m_y \rightarrow m_y = 0$$

2.6.3 Covariance function

Given an ARMA(m, n) model defined as:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + c_0 e(t) + c_1 e(t-1) + \cdots + c_n e(t-n)$$

where $e(t) \sim WN(0, \lambda^2)$, we aim to compute the covariance function at $\tau = 0$, considering $m_y = 0$:

$$\begin{aligned}\gamma_y(0) &= \mathbb{E}[(y(t) - m_y)(y(t) - m_y)] \\ &= \mathbb{E}\left[(y(t))^2\right] \\ &= \mathbb{E}\left[(a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n))^2\right] \\ &= a_1^2 \underbrace{\mathbb{E}\left[y(t-1)^2\right]}_{\gamma_y(0)} + \cdots + c_0^2 \underbrace{\mathbb{E}\left[e(t)^2\right]}_{\lambda^2} + \cdots + c_n^2 \underbrace{\mathbb{E}\left[e(t-n)^2\right]}_{\lambda^2} + \\ &+ 2a_1 a_2 \underbrace{\mathbb{E}[y(t-1)y(t-2)]}_{\gamma_y(1)} + \cdots + 2a_1 c_0 \underbrace{\mathbb{E}[y(t-1)e(t)]}_{0 \text{ if the times are different}} + \cdots + 2c_0 c_1 \underbrace{\mathbb{E}[e(t)e(t-1)]}_{0} \\ &= a_1^2 \gamma_y(0) + \cdots + c_0^2 \lambda^2 + \cdots + c_n^2 \lambda^2 + 2a_1 a_2 \gamma_y(1) + \cdots + 2a_1 c_1 \mathbb{E}[y(t-1)e(t-1)] + \ldots\end{aligned}$$

We require $\gamma_y(1), \gamma_y(2), \ldots, \gamma_y(n)$ to compute $\gamma_y(0)$.

To compute the covariance function at $\tau = 1$, considering $m_y = 0$, we have:

$$\begin{aligned}\gamma_y(1) &= \mathbb{E}\left[(y(t) - m_y)\,(y(t-1) - m_y)\right] \\ &= \mathbb{E}\left[(a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n))\, y(t-1)\right] \\ &= a_1 \mathbb{E}\left[y(t-1)^2\right] + \cdots + c_0 \mathbb{E}\left[e(t) y(t-1)\right] + \ldots \\ &= a_1 \gamma_y(0) + \cdots + c_0 \mathbb{E}\left[e(t) y(t-1)\right] + \ldots\end{aligned}$$

Proceeding with increasing values of τ, we obtain a set of m recursive equations known as Yule-Walker equations for an ARMA(m, n) process:

$$\begin{cases} \gamma_y(0) = a_1^2 \gamma_y(0) + a_2^2 \gamma_y(0) + 2 a_1 a_2 \gamma_y(1) + \ldots \\ \gamma_y(1) = a_1 \gamma_y(0) + c_0 \mathbb{E}\left[e(t) y(t-1)\right] + \ldots \\ \vdots \\ \gamma_y(m-1) = a_1 \gamma_y(m-2) + \ldots \end{cases}$$

At the end we get a set of m recursive equations called Yule-Walker equations for an ARMA(m, n) process. These equations require all covariances $\gamma_y(0), \gamma_y(1), \ldots, \gamma_y(m-1)$ to compute $\gamma_y(m)$.

2.7 ARMAX process

We aim to construct an ARMA process with an additional noise component affecting the output variable. The general structure of the model employed for this purpose is illustrated in the diagram below:

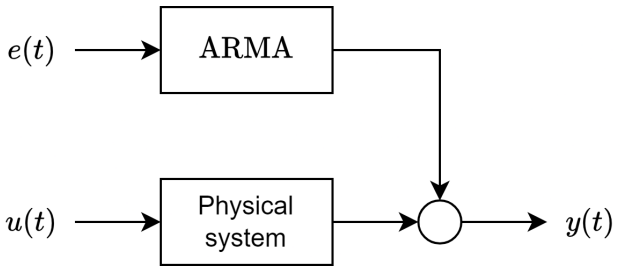


Figure 2.7: ARMAX model

The general formulation of this process is expressed as:

$$y(t) = a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n) + b_0 u(t-k) + \cdots + b_p u(t-k-p)$$

Here, $e(t) \sim WN(\mu, \lambda^2)$ This model comprises three main components: an AutoRegressive process of order m, a Moving Average process of order n, and an exogenous part denoted as $X_{(k,p)}$.

Within the exogenous part, p indicates the order of the process, signifying the number of past values utilized, while k signifies the pure input-output delay. This delay accounts for situations where the influence of $u(t)$ on $y(t)$ is not immediate, typically set to one.

2.7.1 Operatorial representation

Given an ARMAX(m, n, p, k) model:

$$y(t) = a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n) + b_0 u(t-k) + \cdots + b_p u(t-k-p)$$

where $e(t) \sim WN(\mu, \lambda^2)$, the model can be expressed in operatorial representation as:

$$y(t) = a_1 z^{-1} y(t) + \cdots + a_m z^{-m} y(t) + c_0 e(t) + \cdots + c_n z^{-n} e(t) + b_0 u z^{-k}(t) + \cdots + b_p z^{-k-p} u(t)$$

By grouping common terms, we obtain:

$$\underbrace{\left(1 - a_1 z^{-1} - \cdots - a_m z^{-m}\right)}_{A(z)} y(t) = \underbrace{\left(c_0 + \cdots + c_n z^{-n}\right)}_{C(z)} e(t) + \underbrace{\left(b_0 + \cdots + b_p z^{-p}\right)}_{B(z)} z^{-k} u(t)$$

This can be compactly rewritten as:

$$y(t) = \frac{B(z)}{A(z)} z^{-k} u(t) + \frac{C(z)}{A(z)} e(t)$$

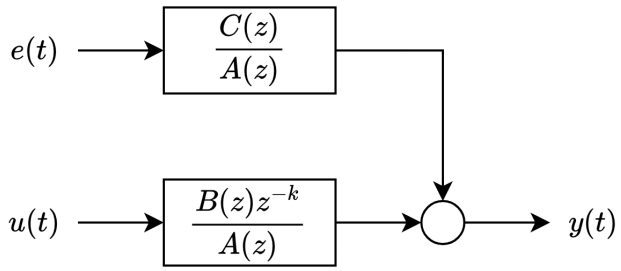


Figure 2.8: ARMAX model in operatorial representation

While it's possible to express the transfer functions with both forward and backward shift operators, it's advisable to maintain consistency by using only one type of shift operator within a formula.

2.8 NARMAX process

Nonlinear ARMAX models are characterized by the expression:

$$y(t) = f\left(y(t-1), \ldots, y(t-m), e(t-1), \ldots, e(t-n), u(t-k), \ldots, u(t-k-p)\right)$$

Here, f represents a nonlinear and parametric function. Commonly utilized functions in this type of models include: polynomials, splines, wavelets, neural networks, and radian basis functions.

2.9 Processes with non-zero mean

Consider an ARMA(m, n) process:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + c_0 e(t) + c_1 e(t-1) + \cdots + c_n e(t-n)$$

Here, $e(t) \sim WN(\mu, \lambda^2)$. In operatorial representation, this process can be expressed as:

$$y(t) = \frac{c_0 + c_1 z^{-1} + \cdots + c_n z^{-n}}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_m z^{-m}} e(t) = \frac{C(z)}{A(z)} e(t)$$

Mean value The mean is computed as:

$$\begin{aligned} m_y = &= \mathbb{E}\left[y(t)\right] \\ &= \mathbb{E}\left[a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n)\right] \\ &= a_1 \mathbb{E}\left[y(t-1)\right] + \cdots + a_m \mathbb{E}\left[y(t-m)\right] + c_0 \underbrace{\mathbb{E}\left[e(t)\right]}_{m_e} + \cdots + c_n \underbrace{\mathbb{E}\left[e(t-n)\right]}_{m_e} \\ &= a_1 \mathbb{E}\left[y(t-1)\right] + \cdots + a_m \mathbb{E}\left[y(t-m)\right] + \left(c_0 + \cdots + c_n\right) m_e \end{aligned}$$

Assuming that $a_1, a_2, \ldots, a_m$ are chosen such that $y(t)$ is a stationary stochastic process with constant mean value, we can rewrite the previous formulation as:

$$m_y = \left(a_1 + a_2 + \cdots + a_m\right) m_y + \left(c_0 + c_1 + \cdots + c_n\right) m_e \rightarrow m_y = \frac{c_0 + c_1 + \cdots + c_n}{1 - a_1 - a_2 - \cdots - a_m} m_e$$

Note that the transfer function is computed at $z = 1$ (the DC gain of $W(z)$), so we have:

$$m_y = W(1)\mathbb{E}\left[e(t)\right]$$

Covariance function The covariance with a generic τ is computed as:

$$\gamma_y(\tau) = \mathbb{E}\left[\left(y(t) - m_y\right)\left(y(t-\tau) - m_y\right)\right]$$

In this case, where $m_y \neq 0$, we have:

$$\mathbb{E}\left[e(t)^2\right] = \mathbb{E}\left[\left(e(t) - \mu\right)^2\right] = \lambda^2$$

and

$$\mathbb{E}\left[e(t)e(t-1)\right] = \mathbb{E}\left[\left(e(t) - \mu\right)\left(e(t-1) - \mu\right)\right] = 0$$

Therefore,

$$\begin{aligned} \lambda^2 &= \mathbb{E}\left[\left(e(t) - \mu\right)^2\right] \\ &= \mathbb{E}\left[e(t)^2\right] + \mathbb{E}\left[\mu^2\right] - \mathbb{E}\left[2\mu e(t)\right] \\ &= \mathbb{E}\left[e(t)^2\right] + \mu^2 - 2\mu\mathbb{E}\left[e(t)\right] \\ &= \mathbb{E}\left[e(t)^2\right] + \mu^2 - 2\mu^2 \end{aligned}$$

Hence:

$$\mathbb{E}\left[e(t)^2\right] = \lambda^2 - \mu^2$$

Similarly, we have:

$$0 = \mathbb{E}\left[\left(e(t) - \mu\right)\left(e(t-1) - \mu\right)\right] = \mathbb{E}\left[e(t)e(t-1)\right] - \mu^2$$

Therefore,

$$\mathbb{E}\left[e(t)e(t-1)\right] = \mu^2$$

2.9.1 Simplification

To simplify the computation, we define two new unbiased processes:

$$\begin{cases} \tilde{y}(t) = y(t) - m_y \\ \tilde{e}(t) = e(t) - m_e \end{cases}$$

In this case, the expected values become:

$$\begin{cases} \mathbb{E}\left[\tilde{y}(t)\right] = \mathbb{E}\left[y(t) - m_y\right] = m_y - m_y = 0 \\ \mathbb{E}\left[\tilde{e}(t)\right] = \mathbb{E}\left[e(t) - m_e\right] = m_e - m_e = 0 \end{cases}$$

Proof. We aim to demonstrate the equivalence between the two processes:

$$\begin{aligned} \tilde{y}(t) &= y(t) - m_y \\ &= a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n) - m_y \\ &= a_1 \left(\tilde{y}(t-1) + m_y\right) + \cdots + a_m \left(\tilde{y}(t-m) + m_y\right) + c_0 \left(\tilde{e}(t) + m_e\right) + \cdots - m_y \\ &= a_1 \tilde{y}(t-1) + \cdots + c_0 \tilde{e}(t) + \cdots - \left(1 - a_1 - \cdots - a_m\right) m_y + \left(c_0 + \cdots + c_n\right) m_e \end{aligned}$$

However, we know that:

$$m_y = \frac{c_0 + c_1 + \cdots + c_n}{1 - a_1 - a_2 - \cdots - a_m} m_e$$

By substitution, we find:

$$\begin{aligned} \tilde{y}(t) &= a_1 \tilde{y}(t-1) + \cdots + c_0 \tilde{e}(t) + \cdots - \left(1 - a_1 - \cdots - a_m\right) m_y + \left(c_0 + \cdots + c_n\right) m_e \\ &= a_1 \tilde{y}(t-1) + \cdots + c_0 \tilde{e}(t) + \cdots - \left(c_0 + \cdots + c_n\right) m_e + \left(c_0 + \cdots + c_n\right) m_e \\ &= a_1 \tilde{y}(t-1) + \cdots + c_0 \tilde{e}(t) + \ldots \end{aligned}$$

Thus, we have shown that the two processes are equivalent. □

Ultimately, we arrive at:

$$\tilde{y}(t) = a_1 \tilde{y}(t-1) + \cdots + a_m \tilde{y}(t-m) + c_0 \tilde{e}(t) + c_n \tilde{e}(t-n) + \ldots \qquad \tilde{e} \sim WN(0, \lambda^2)$$

Subsequently, we can compute the covariance as follows:

$$\gamma_{\tilde{y}}(\tau) = \mathbb{E}\left[\tilde{y}(t)\tilde{y}(t-\tau)\right] = \mathbb{E}\left[\left(y(t) - m_y\right)\left(y(t-\tau) - m_y\right)\right] = \gamma_y(\tau)$$

CHAPTER 3

Frequency analysis

3.1 Spectral density

The spectral density of a stochastic stationary process $y(t)$ is expressed as follows:

$$\Gamma_y(\omega) = \sum_{\tau=-\infty}^{+\infty} \gamma_y(\tau) e^{-j\omega\tau} = \mathcal{F}\{\gamma_y(\tau)\}$$

In simpler terms, it is the Fourier transform of the auto-covariance function $\gamma_y(\tau)$.

This spectral density possesses several key properties:

- It is a real function of a real variable ω:

$$Im(\Gamma_y(\omega)) = 0 \quad \forall\omega \in \mathbb{R}$$

- It is always non-negative:

$$\Gamma_y(\omega) \geq 0 \quad \forall\omega \in \mathbb{R}$$

- It exhibits symmetry (even function):

$$\Gamma_y(\omega) = \Gamma_y(-\omega) \quad \forall\omega \in \mathbb{R}$$

- It follows a 2π periodic pattern:

$$\Gamma_y(\omega) = \Gamma_y(\omega + 2\pi k) \quad \forall\omega \in \mathbb{R}, k \in \mathbb{Z}$$

The Nyquist frequency is denoted as π, implying that π equals the maximum value of ω. Consequently,

$$\omega = \frac{2\pi}{T} \leq \pi \rightarrow T_{\min} \leq 2$$

This signifies that a periodic signal must be represented with a minimum period of two samples.

3.1.1 Inverse transformation

It's feasible to derive the covariance function from the spectral density:

$$\gamma_y(\omega) = \mathcal{F}^{-1}\left(\gamma_y(\omega)\right) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_y(\omega) e^{j\omega\tau}\, d\omega$$

This transformation is valid because both $\gamma_y(\omega)$ and $\Gamma_y(\omega)$ encapsulate identical information about the process $y(t)$.

It's worth noting that the variance determined when $\tau = 0$ is equal to:

$$\gamma_y(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_y(\omega)\, d\omega$$

3.2 White Noise spectral density

Let's explore the representation of White Noise $e(t) \sim WN(\mu, \lambda^2)$, which can be depicted as:

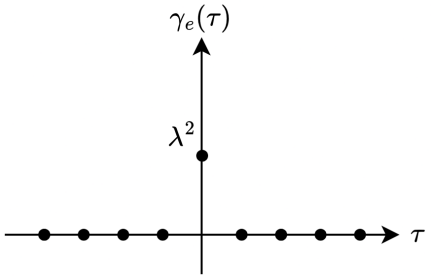


Figure 3.1: White Noise covariance function

The spectral density of White Noise is determined as:

$$\Gamma_e(\omega) = \sum_{\tau=-\infty}^{+\infty} \gamma_e(\tau) e^{-j\omega\tau}$$

Since the covariance function $\gamma_e(\tau)$ solely holds a non-zero value at $\tau = 0$, the spectral density is essentially the sum of the variance and an infinite series of zeros:

$$\Gamma_e(\omega) = \gamma_e(0)e^{-j\omega 0} + 0 + \cdots + 0 = \gamma_e(0) = \lambda^2$$

Here's a visualization of the spectral density function for White Noise:

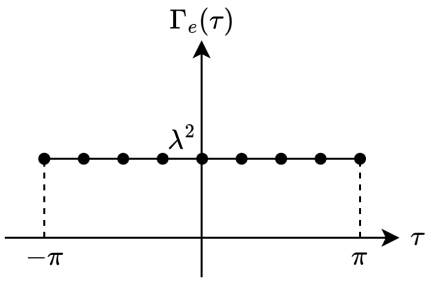


Figure 3.2: White Noise spectral density function

3.3 Stationary stochastic process White Noise

Let's explore the general scenario wherein a stationary stochastic process is generated as outputs of digital filters:

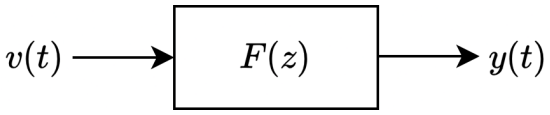


Figure 3.3: Stationary stochastic process

In this setup, $v(t)$ represents a stationary stochastic process, and $F(z)$ is asymptotically stable. Consequently, the output $y(t)$ manifests as a stationary stochastic process.

Theorem 3.3.1. *The spectral density of the output is given by:*

$$\Gamma_y(\omega) = \left|F(e^{j\omega})\right|^2 \Gamma_v(\omega)$$

Example:
Let's examine an MA(1) process:

$$y(t) = e(t) + ce(t-1) \qquad c \in \mathbb{R}, e(t) \sim WN(0,1)$$

The computation of the covariance function is straightforward:

$$\gamma_y(\tau) = \begin{cases} 1 + c^2 & \tau = 0 \\ c & \tau = \pm 1 \\ 0 & |\tau| \geq 1 \end{cases}$$

We can compute the spectral density using the definition:

$$\begin{aligned} \Gamma_y(\omega) &= \sum_{\tau=-\infty}^{+\infty} \gamma_y(\omega) e^{-j\omega\tau} \\ &= \gamma_y(0)e^{-j\omega 0} + \gamma_y(1)e^{-j\omega 1} + \gamma_y(-1)e^{-j\omega(-1)} \\ &= \gamma_y(0) + \gamma_y(1)e^{-j\omega} + \gamma_y(-1)e^{j\omega} \\ &= 1 + c^2 + ce^{-j\omega} + ce^{j\omega} \end{aligned}$$

Utilizing Euler's formula:

$$\begin{cases} e^{-j\omega} = \cos\omega - j\sin\omega \\ e^{j\omega} = \cos\omega + j\sin\omega \end{cases}$$

We find:

$$\begin{aligned} \Gamma_y(\omega) &= 1 + c^2 + ce^{-j\omega} + ce^{j\omega} \\ &= 1 + c^2 + c\left(\cos\omega - j\sin\omega + \cos\omega + j\sin\omega\right) \\ &= 1 + c^2 + 2c\cos\omega \end{aligned}$$

Alternatively, the spectral density can be computed using the theorem:

$$\Gamma_y(\omega) = \left|F(e^{j\omega})\right|^2 \Gamma_v(\omega)$$

In this case, considering $y(t) = e(t) + ce(t-1) = (1 + cz^{-1})\, e(t) = F(z)e(t)$ where $F(z) = (1 + cz^{-1})$, and since the input process $e(t)$ is a White Noise, the spectral density is $\lambda^2 = 1$.

$$\Gamma_y(\omega) = \left|F(e^{j\omega})\right|^2 \Gamma_e(\omega) = \left|1 + ce^{-j\omega}\right|^2 \cdot 1$$

Computing the square of a complex number by multiplying it by its conjugate:

$$\begin{aligned}
\Gamma_y(\omega) &= \left|1 + ce^{-j\omega}\right|^2 \\
&= \left(1 + ce^{-j\omega}\right)\left(1 + ce^{+j\omega}\right) \\
&= 1 + c^2 + ce^{-j\omega} + ce^{+j\omega} \\
&= 1 + c^2 + 2c\cos\omega
\end{aligned}$$

Now, we aim to compute the covariance function using the found spectral density:

$$\begin{aligned}
\gamma_y(0) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_y(\omega) e^{j\omega 0}\, d\omega \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} 1 + c^2 + 2c\cos\omega\, d\omega \\
&= \frac{1}{2\pi}\left[\int_{-\pi}^{\pi} 1\, d\omega + \int_{-\pi}^{\pi} c^2\, d\omega + \int_{-\pi}^{\pi} 2c\cos\omega\, d\omega\right] \\
&= \frac{1}{2\pi}\left[1 + c^2 + 2c\int_{-\pi}^{\pi} \cos\omega\, d\omega\right] \\
&= \frac{1}{2\pi}\left[\left[\left(1 + c^2\right)\omega\right]_{-\pi}^{\pi} + 2c[\sin\omega]_{-\pi}^{\pi}\right] \\
&= \frac{1}{2\pi}\left[2\pi\left(1 + c^2\right)\right] \\
&= 1 + c^2
\end{aligned}$$

CHAPTER 4

Prediction

4.1 Possible representations

An ARMA process can be expressed equivalently in various representations:

1. Time domain representation:
$$y(t) = a_1 y(t-1) + \cdots + a_m y(t-m) + c_0 e(t) + \cdots + c_n e(t-n)$$
2. Operatorial representation:
$$y(t) = \frac{C(z)}{A(z)} e(t)$$
3. Probabilistic representation:
$$m_y, \gamma_y(\tau)$$
4. Frequency domain representation:
$$m_y, \Gamma_y(\omega)$$

4.1.1 Canonical representation

Equivalent form one Let's introduce a parameter $\alpha \in \mathbb{R}$ and define a process as:
$$y(t) = W(z)\xi(t)$$
We can express it as:
$$y(t) = W(z)\xi(t) = W(z)\frac{\alpha}{\alpha}\xi(t) = \tilde{W}(z)\tilde{\xi}(t)$$
Here, $\tilde{W}(z) = \frac{1}{\alpha}W(z)$, and $\tilde{\xi} \sim WN(0, \alpha^2\lambda^2)$.

Equivalent form two Consider the process defined as:
$$y(t) = W(z)\xi(t)$$
We can represent it as:
$$y(t) = W(z)\xi(t) = W(z)\frac{z^n}{z^n}\xi(t) = \tilde{W}(z)\tilde{\xi}(t)$$
Here, $\tilde{W}(z) = z^n W(z)$, and $\tilde{\xi} = \xi(t-n)$.

Equivalent form three Let's introduce a complex number p such that $|p| < 1$ and define a process as:

$$y(t) = W(z)\xi(t)$$

We can rewrite it as:

$$y(t) = W(z)\xi(t) = W(z)\frac{z-p}{z-p}\xi(t) = \tilde{W}(z)\xi(t)$$

Here, $\tilde{W}(z) = W(z)\frac{z-p}{z-p}$.

Equivalent form four Consider the process defined as:

$$y(t) = W(z)\xi(t)$$

Suppose $W(z) = W_1(z)(z-q)$, then:

$$y(t) = W(z)\xi(t) = W_1(z)(z-q)\frac{z-\frac{1}{q}}{z-q}\xi(t) = W_(z)\left(z-\frac{1}{q}\right)\xi(t) = \tilde{W}(z)\tilde{\xi}(t)$$

Here, $\tilde{\xi}(z) = \dfrac{z-q}{z-\frac{1}{q}}\xi(t)$. The fraction $\frac{z-q}{z-\frac{1}{q}}$ is called an all-pass filter.

Theorem 4.1.1 (Spectral factorization)**.** *Let $y(t)$ be a stationary stochastic process with a rational spectral density $\Gamma_y(\omega)$, there exists a unique pair $\xi(t)$ and $W(z)$ such that $y(t) = W(z)\xi(t)$ if and only if:*

1. *$C(z)$ and $A(z)$ are monic, i.e., $c_0 = 1$ and $a_0 = 1$.*
2. *$C(z)$ and $A(z)$ have null relative degree (they have the same degree).*
3. *$C(z)$ and $A(z)$ are co-prime (they do not have common factors).*
4. *The absolute value of poles and zeros of $W(z)$ is less than one.*

4.2 Prediction problem

Consider a zero-mean ARMA process represented as:

$$y(t) = W(z)e(t) \qquad e(t) \sim WN(0, \lambda^2)$$

We make the following assumptions:

1. The pair $(W(z), e(t))$ constitutes a canonical representation of $y(t)$.
2. $W(z)$ has no zeros on the unit circle boundary.

K-step prediction Given observations up to time t:

$$\dots, y(t-100), y(t-99), \dots, y(t-1), y(t)$$

we aim to predict the future value of the process at time $t+k$:

$$\hat{y}(t+k|t)$$

Considering the sample set $y(t), y(t-1), y(t-2)$, potential one-step predictors include:

- Simple average of past values: $\hat{y}(t+1|t) = \dfrac{y(t)+y(t-1)+y(t-2)}{3}$.
- Weighted average of past values: $\hat{y}(t+1|t) = \dfrac{2y(t)+\frac{1}{2}y(t-1)+\frac{1}{2}y(t-2)}{3}$.
- Geometric mean of past values: $\hat{y}(t+1|t) = \sqrt[3]{y(t)y(t-1)y(t-2)}$.
- Complex rules based on prior experimental knowledge.

However, in these examples, the underlying model generating the samples isn't fully considered, neglecting valuable information. For an optimal predictor, we seek to leverage both the information from the model (data generation mechanism) and the information from past observations (specific realization).

4.3 Optimal predictor

The quality of prediction can be quantified through the Mean Square Prediction Error (MSPE), defined as:

$$\text{MSPE} = \mathbb{E}\left[(y(t+k) - \hat{y}(t+k|t))^2\right]$$

Minimization Our goal is to minimize the squares of prediction errors. The prediction depends on past samples through a function:

$$\hat{y}(t+k|t) = f(y(t), y(t-1), \dots)$$

Therefore, we would need to minimize all suitable functions for prediction and select the best one, which is generally impractical due to the vast number of possible functions. However, by constraining ourselves to linear predictors, this task becomes more manageable.

Linear predictors take the form:

$$\hat{y}(t+k|t) = \alpha_0 y(t) + \alpha_1 y(t-1) + \dots = \sum_{i=0}^{+\infty} \alpha_i y(t-i)$$

Here, $\alpha_i \in \mathbb{R}$ such that $\sum_{i=0}^{+\infty} \alpha_i < +\infty$. We can express this general formulation using operatorial representation as:

$$\hat{y}(t+k|t) = \left(\alpha_0 + \alpha_1 z^{-1} \alpha_2 z^{-2} + \dots\right) y(t) = F_\alpha(z) y(t)$$

We assume that $\hat{y}(t+k|t)$ is the steady-state output of a linear digital filter with transfer function $F_\alpha(z)$.

4.3.1 Optimal predictor design

We aim to determine the optimal coefficients $\alpha_0^\circ, \alpha_1^\circ, \alpha_2^\circ, \dots$ of the linear predictor:

$$\hat{y}(t+k|t,\alpha) = \sum_{i=0}^{+\infty} \alpha_i y(t-i)$$

such that the mean squared prediction error is minimized:

$$\min_{\alpha_0,\alpha_1,\alpha_2,\dots} \mathbb{E}\left[(y(t+k) - \hat{y}(t+k|t,\alpha))^2\right]$$

We begin by considering a generic Moving Average process:

$$y(t) = W(z)e(t) = \sum_{i=0}^{+\infty} w_i e(t-i) \qquad e(t) \sim WN(0,\lambda^2)$$

When we perform a time shift, we have:

$$y(t-1) = \sum_{i=0}^{+\infty} w_i e(t-1-i)$$

We can then insert this expression into the predictor, resulting in:

$$\begin{aligned}\hat{y}(t+k|t) &= \alpha_0 \sum_{i=0}^{+\infty} w_i e(t-i) + \alpha_1 \sum_{i=0}^{+\infty} w_i e(t-1-i) + \dots \\ &= \beta_0 e(t) + \beta_1 e(t-1) + \dots \\ &= \sum_{i=0}^{+\infty} \beta_i e(t-1)\end{aligned}$$

where the factor β is found by grouping all elements referring to the same time instant.

We can reformulate the optimization problem with respect to α as a new optimization problem with respect to β. This can be expressed as:

$$\min_{\beta_0,\beta_1,\beta_2,\dots} \mathbb{E}\left[(y(t+k) - \hat{y}(t+k|t,\beta))^2\right]$$

Considering that $y(t+k)$ admits the Moving Average representation, we write:

$$y(t+k) = \sum_{i=0}^{+\infty} w_i e(t+k-i) = \underbrace{\sum_{j=0}^{k-1} w_j e(t+k-j)}_{\text{future } e(t)} + \underbrace{\sum_{i=0}^{+\infty} w_{k+i} e(t-i)}_{\text{past and present } e(t)}$$

We can express the prediction error as follows:

$$\begin{aligned}\mathbb{E}\left[(\varepsilon(t+k|t))^2\right] &= \mathbb{E}\left[\left(\sum_{i=0}^{k-1} w_j e(t+k-i) + \sum_{i=0}^{+\infty} w_{k+i} e(t-i) - \sum_{i=0}^{+\infty} \beta_i e(t-1)\right)^2\right] \\ &= \mathbb{E}\left[\left(\sum_{i=0}^{k-1} w_j e(t+k-i) + \sum_{i=0}^{+\infty} w_{k+i} e(t-i) - \beta_i e(t-1)\right)^2\right]\end{aligned}$$

Expanding the square, we have:

$$\mathbb{E}\left[(\varepsilon(t+k|t))^2\right] = \mathbb{E}\left[\underbrace{\left(\sum_{i=0}^{k-1} w_j e(t+k-i)\right)^2}_{\beta \text{ independent}}\right] + \mathbb{E}\left[\underbrace{\left(\sum_{i=0}^{+\infty} w_{k+i}e(t-i) - \beta_i e(t-1)\right)^2}_{\beta \text{ dependent}}\right] + \\ + 2\mathbb{E}\left[\underbrace{\left(\sum_{i=0}^{k-1} w_j e(t+k-i) \sum_{i=0}^{+\infty} w_{k+i}e(t-i) - \beta_i e(t-1)\right)}_{0}\right]$$

The first term cannot be minimized since there will always be some error in predicting future samples, while the second term depends on β, requiring minimization to solve the optimization problem:

$$\sum_{i=0}^{+\infty} \beta_i e(t-i) = \sum_{i=0}^{+\infty} w_{k+i} e(t-i)$$

Thus, an optimal predictor satisfies:

$$\beta_i^\circ = w_{k+i} \qquad i = 0, 1, 2, \dots$$

This is termed the optimal predictor from the White Noise, and its expression is:

$$\hat{y}^\circ = \sum_{i=0}^{+\infty} w_{k+i} e(t-i)$$

Operatorial representation The operatorial representation of a process is:

$$W(z) = \frac{C(z)}{A(z)} = w_0 + w_1 z^{-1} + \dots$$

By performing long division between $C(z)$ and $A(z)$, we obtain:

$$\frac{C(z)}{A(z)} = E(z) + \frac{z^{-k}F(z)}{A(z)}$$

Here, k is the number of steps in the long division. This equation is termed the Diophantine equation, useful for rewriting the process at a specific time lag k as:

$$\begin{aligned}
\hat{y}(t+k|t) &= \frac{C(z)}{A(z)} e(t+k) \\
&= \left[E(z) + \frac{z^{-k}F(z)}{A(z)}\right] e(t+k) \\
&= E(z)e(t+k) + \frac{z^{-k}F(z)}{A(z)} e(t+k) \\
&= E(z)e(t+k) + \frac{F(z)}{A(z)} e(t) \\
&= \underbrace{e_0 e(t+k) + e_1 e(t+k-1) + \dots + e_{k-1} e(t+1)}_{\text{future } e(t)} + \underbrace{\frac{F(z)}{A(z)} e(t)}_{\text{past and present } e(t)}
\end{aligned}$$

Thus, the optimal predictor in this representation is:

$$\hat{y}(t+k|t) = \frac{F(z)}{A(z)}e(t)$$

characterized by a finite set of coefficients (those of $F(z)$ and $A(z)$).

4.3.2 White Noise reconstruction

To practically utilize the optimal predictor we've derived, we require the White Noise values at each time step. However, obtaining these values directly is often impractical. Therefore, we need to reconstruct plausible White Noise values from the available samples.

The ARMA process under consideration is formulated as:

$$y(t) = W(z)e(t) = \frac{C(z)}{A(z)}e(t)$$

Given our assumption that the zeros lie inside the unit circle and that $W(z)$ is asymptotically stable, we can invert the formula to find:

$$e(t) = \frac{A(z)}{C(z)}y(t)$$

The transfer function $W(z)^{-1}$ is termed the whitening filter.

Optimal predictor Finally, we can express the optimal predictor from data as:

$$\hat{y}(t+k|t) = \frac{F(z)}{C(z)}y(t)$$

4.3.3 Summary

We summarize our findings as follows:

- The optimal predictor, denoted as $\hat{y}(t+k|t,s)$, can be expressed as $\frac{F(z)}{C(z)}y(t,s)$.
- Due to the presence of roots of $C(z)$ inside the unit circle, $\hat{y}(t+k|t)$ represents a stationary stochastic process. Thus, we can represent it equivalently as:
  - $\hat{y}(t+k|t) = \frac{F(z)}{C(z)}y(t)$.
  - $\hat{y}(t+k|t-k) = \frac{F(z)}{C(z)}y(t-k)$.

4.3.4 Optimal prediction error

The optimal prediction error is defined as:

$$\varepsilon(t+k|t) = E(z)e(t+k)$$

Alternatively, it can be expressed as:

$$\varepsilon(t|t-k) = E(z)e(t)$$

This equivalence holds because the prediction error is also a stationary stochastic process, specifically an MA(-1).

Variance function In the optimal scenario, the variance of the prediction error is given by:

$$\mathrm{Var}\left[\varepsilon(t+k|t)\right] = \left(w_0^2 + w_1^2 + \cdots + w_{k-1}^2\right)\lambda^2$$

Since we're dealing with a stationary stochastic process, the variance must converge to a specific value, namely the steady-state variance of the process.

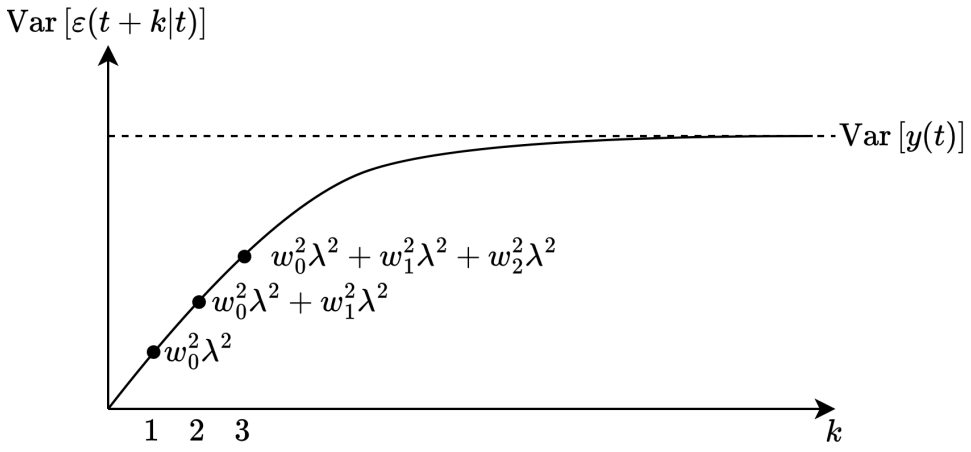


Figure 4.1: Variance of the optimal predictor

Generally, the following inequality holds:

$$\mathrm{Var}\left[e(t)\right] \leq \mathrm{Var}\left[\varepsilon(t+k|t)\right] < \mathrm{Var}\left[y(t)\right]$$

4.4 Predictor implementation

We have determined that the general form of the optimal predictor is:

$$\hat{y}(t+k|t) = \frac{F(z)}{C(z)}y(t)$$

Now, let's focus on a specific form of the optimal predictor:

$$\hat{y}(t+k|t) = \frac{1}{1+cz^{-1}}y(t)$$

For implementation in a program, it needs to be converted into a time-domain representation:

$$\hat{y}(t+k|t) = y(t) - c\hat{y}(t+k-1|t-1)$$

This form represents an AutoRegressive model.

However, there's a problem with the second term on the right side of the equation. When trying to compute $\hat{y}(-1+k|-1)$, we encounter an initialization problem because we only have samples for $t \geq 0$.

Heuristic To solve this problem, a heuristic solution is employed. It sets $\hat{y} = m_y$ when data is not available. This effectively utilizes the trivial predictor.

This heuristic is acceptable due to the asymptotic stability of $\frac{F(z)}{C(z)}$, causing the effect of this initialization to rapidly diminish. Consequently, this approximation becomes negligible as t grows sufficiently large.

4.5 Non-zero mean ARMA process

Consider the ARMA process:

$$y(t) = \frac{C(z)}{A(z)}e(t) \qquad e(t) \sim WN(\mu, \lambda^2)$$

The expected value is:

$$\mathbb{E}\left[y(t)\right] = W(1)\mu = \bar{y}$$

Assuming the process is in canonical representation, we can express it as:

$$\begin{cases} \tilde{y}(t) = y(t) - \bar{y} \\ \tilde{e}(t) = e(t) - \mu \end{cases} \rightarrow \tilde{y}(t) = W(z)\tilde{e}(t)$$

Applying the previously found prediction algorithm:

1. Perform the long division.
2. Take $\frac{F(z)}{A(z)}\tilde{e}(t)$.
3. Use the whitening filter.

We obtain:

$$\hat{\tilde{y}}(t+k|t) = \frac{F(z)}{C(z)}\tilde{y}(t)$$

Trivial solution To obtain the prediction for the original problem, we follow these steps:

- Remove the mean from each data point: $\tilde{y} = y(t) - \bar{y}$.
- Compute $\hat{\tilde{y}}(t+k|t)$.
- Finally, the predicted value is given by: $\hat{y}(t+k|t) = \hat{\tilde{y}}(t+k|t) + \bar{y}$.

Optimized solution Alternatively, we can integrate the computation of the correct predictor by modifying the formula.

Given that $y(t+k|t) = \tilde{y}(t+k|t) + \bar{y}$, we deduce:

$$\hat{y}(t+k|t) = \hat{\tilde{y}}(t+k|t) + \bar{y} = \frac{F(z)}{C(z)}\tilde{y}(t) + \bar{y} = \frac{F(z)}{C(z)}(y(t) - \bar{y}) + \bar{y}$$

Asymptotically, $\dfrac{F(z)}{C(z)}\bar{y}$ tends to $\dfrac{F(1)}{C(1)}\bar{y}$:

$$\hat{y}(t+k|t) = \frac{F(z)}{C(z)}(y(t) - \bar{y}) + \bar{y} = \frac{F(z)}{C(z)}y(t) + \left(1 - \frac{F(1)}{C(1)}\right)\bar{y}$$

This represents the form of the predictor for an ARMA process with non-zero mean.

4.6 ARMAX process

Consider the ARMAX process:

$$y(t) = \frac{C(z)}{A(z)}e(t) + \frac{B(z)}{A(z)}u(t-d) \qquad e(t) \sim WN(0, \lambda^2)$$

Assuming:

- The process is written in canonical representation.
- $u(t-d)$ is measured from $t = -\infty$ until $t = +\infty$.

We define a new process by removing the deterministic part:

$$z(t) = y(t) - \frac{B(z)}{A(z)}u(t-d) = \frac{C(z)}{A(z)}e(t)$$

We know that we can compute the prediction of this stochastic part as:

$$\hat{z}(t+k|t) = \frac{F(z)}{C(z)}z(t)$$

We can rewrite the process also as:

$$y(t) = z(t) + \frac{B(z)}{A(z)}u(t-d)$$

We know that we can compute the prediction of this process as follows:

$$\begin{aligned}
\hat{y}(t+k|t) &= \frac{B(z)}{A(z)}u(t+k-d) + \hat{z}(t+k|t) \\
&= \frac{B(z)}{A(z)}u(t+k-d) + \frac{F(z)}{C(z)}z(t) \\
&= \frac{B(z)}{A(z)}u(t+k-d) + \frac{F(z)}{C(z)}\left(y(t) - \frac{B(z)}{A(z)}u(t-d)\right) \\
&= \frac{B(z)}{C(z)}\left(\frac{C(z)}{A(z)} - \frac{F(z)}{A(z)}z^{-k}\right)u(t+k-d) + \frac{F(z)}{C(z)}y(t) \\
&= \frac{B(z)E(z)}{C(z)}u(t+k-d) + \frac{F(z)}{C(z)}y(t)
\end{aligned}$$

This expression depends on past values of input and outputs and the model information.

CHAPTER 5

Identification

5.1 Introduction

System identification is the discipline focused on techniques for extracting models from data.

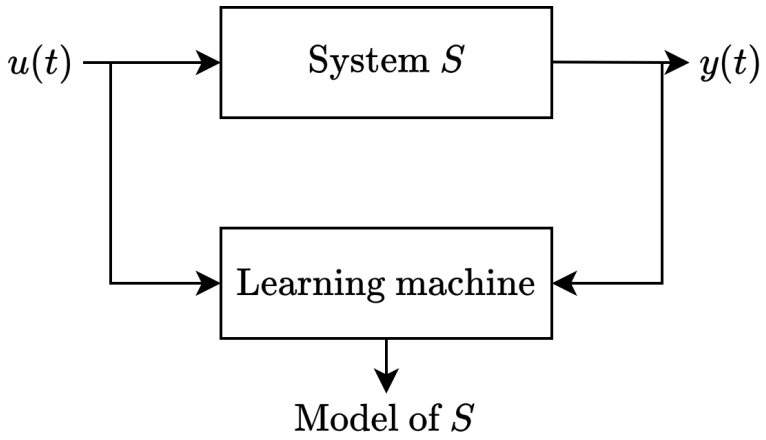


Figure 5.1: Identification process

In essence, the core components of the identification challenge include:

1. The system $\mathcal{S}$.
2. The class of models $\mathcal{M}$.
3. The method of identification $\mathcal{I}$.
4. The experimental setup for identification $\mathcal{E}$.

5.1.1 Parametric system identification

The steps involved in parametric system identification are outlined below:

1. *Experiment design and data collection*: the initial step involves designing experiments and gathering data. Sometimes data are predefined, making it impossible to conduct further experiments. In other cases, where the system is accessible, additional data can be collected. In such scenarios, the choice of input signal $u(t)$ impacts the informativeness, while the data length N affects the confidence level.

2. *Selection of a parametric model class* $\mathcal{M}(\vartheta) = \{M(\vartheta), \vartheta \in \vartheta\}$: this step entails choosing a suitable parametric model class. Considerations include discrete or continuous time, linear or nonlinear, time-invariant or time-varying, and static or dynamic models. To fully define $y(t)$ as ARMA or ARMAX process, ϑ alone isn't sufficient; parameters such as n_a, n_b, n_c, pure delay, and White Noise characteristics expressed through λ^2 must also be specified. Notably, the pure delay and λ^2 aren't critical as they're derived from the n parameters. Admissible values for ϑ must also be determined.

3. *Choice of the identification criterion* $J_N(\vartheta) \geq 0$: the identification criterion is chosen, often based on prediction error minimization. For instance, using the predictive approach, the criterion $J_N(\vartheta)$ can be defined as the expected square prediction error:

$$J_N(\vartheta) = \mathbb{E}\left[(y(t+1) - \hat{y}(t+1|t))^2\right]$$

   A low prediction error indicates a good model. Typically, a one-step-ahead predictor is chosen for optimality checks, ensuring the prediction error equals λ^2. Therefore, $J_N(\hat{\vartheta}_N) \approx \lambda^2$

4. *Minimization of* $J_N(\vartheta)$ with respect to ϑ: this step involves minimizing the criterion function $J_N(\vartheta)$ to estimate the parameters $\hat{\vartheta}_N$. Depending on the model, the criterion function could be quadratic (e.g., for AR and ARX models) or non-quadratic (e.g., for ARMA and ARMAX models).

5. *Model validation*: despite assumptions made during the process, such as the system belonging to a specific model set and fixed values for n_a, n_b, and n_c, it's essential to validate the model's quality. This involves performing quality checks to ensure the identified model accurately represents the system dynamics.

5.2 AR and ARX models

Let's consider the AR and ARX class of models:

$$\mathcal{M}(\vartheta) : y(t) = \frac{B(z)}{A(z)}u(t-d) + \frac{1}{A(z)}e(t) \qquad e(t \sim WN(0, \lambda^2))$$

Here, the polynomials are defined as:

$$\begin{cases} A(z) = 1 - a_1 z^{-1} - a_z z^{-2} - \cdots - a_m z^{-m} \\ B(z) = b_1 + b_2 z^{-1} + b_3 z^{-2} + \cdots + b_p z^{-p+1} \\ C(z) = 1 \end{cases}$$

Where:

$$\vartheta = \begin{bmatrix} a_1 & a_2 & a_n & \dots & a_m & b_1 & b_2 & \dots & b_p \end{bmatrix}^T$$

We can rewrite the model as:

$$\begin{aligned}\mathcal{M}(\vartheta) :& A(z)y(t) = B(z)u(t-d) + e(t)\\ & y(t) - y(t) + A(z)y(t) = B(z)u(t-d) + e(t)\\ & y(t) = (1 - A(z))y(t) + B(z)u(t-d) + e(t)\\ & \underbrace{\left(a_1 z^{-1} + a_z z^{-2} + \dots\right) y(t) + \left(b_1 + b_2 z^{-1} + b_3 z^{-2} + \dots\right) u(t-d)}_{\text{predictable at time } t} + e(t)\end{aligned}$$

The prediction for the predictable component can be expressed as:

$$\hat{\mathcal{M}}(\vartheta) : \hat{y}(t|t-1) = a_1 y(t-1) + a_2 y(t-2) + \dots + b_1 u(t-d) + b_2 u(t-d-1) + \dots$$

From this, we can derive the regression vector:

$$\varphi(t) = \begin{bmatrix} y(t-1) & y(t-2) & \dots & y(t-m) & u(t-d) & u(t-d-1) & \dots \end{bmatrix}^T$$

These values are obtained from past instances of the process. Consequently, we can rewrite the predictor as:

$$\hat{y}(t|t-1) = \vartheta^T \varphi(t) = \varphi(t)^T \vartheta$$

This formulation reveals the linear relationship with ϑ.

Since $\hat{y}$ exhibits linearity with respect to ϑ, $J_N(\vartheta)$ becomes quadratic in ϑ. For optimizing the function $J_N(\vartheta)$, the following conditions are necessary:

- $\hat{\vartheta}_N$ is a stationary point: $\frac{\partial J_N(\vartheta)}{\partial \vartheta}$ in $\vartheta = \hat{\vartheta}_N$ should equal zero.
- $\hat{\vartheta}_N$ is a minimum: $\frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2}$ in $\vartheta = \hat{\vartheta}_N$ must be greater than or equal to zero.

First derivative The general expression for the derivative is:

$$\begin{aligned}\frac{\partial J_N(\vartheta)}{\partial \vartheta} &= \frac{d}{d\vartheta}\left[\frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \varphi(t)^T\vartheta\right)^2\right]\\ &= \frac{1}{N}\sum_{t=1}^{N}\frac{d}{d\vartheta}\left[\left(y(t) - \varphi(t)^T\vartheta\right)^2\right]\\ &= \frac{1}{N}\sum_{t=1}^{N} 2\left(y(t) - \varphi(t)^T\vartheta\right)\frac{d}{d\vartheta}\left(y(t) - \varphi(t)^T\vartheta\right)\\ &= \frac{2}{N}\sum_{t=1}^{N}\left(y(t) - \varphi(t)^T\vartheta\right)\left(-\varphi(t)\right)\\ &= -\frac{2}{N}\sum_{t=1}^{N}\varphi(t)\left(y(t) - \varphi(t)^T\vartheta\right)\end{aligned}$$

Now, we impose:

$$\begin{aligned}&-\frac{2}{N}\sum_{t=1}^{N}\varphi(t)\left(y(t) - \varphi(t)^T\hat{\vartheta}_N\right) = 0 \rightarrow\\ &-\frac{2}{N}\sum_{t=1}^{N}\varphi(t)y(t) + \frac{2}{N}\sum_{t=1}^{N}\varphi(t)\varphi(t)^T\hat{\vartheta}_N = 0 \rightarrow\\ &\hat{\vartheta}_N\sum_{t=1}^{N}\varphi(t)\varphi(t)^T = \sum_{t=1}^{N}\varphi(t)y(t)\end{aligned}$$

These are known as the least-squares normal equations, forming a system of n_ϑ linear equations in n_ϑ unknowns.

If $\sum_{t=1}^{N} \varphi(t)\varphi(t)^T$ is non-singular, we can express $\hat{\vartheta}_N$ as:

$$\hat{\vartheta}_N = \left(\sum_{t=1}^{N} \varphi(t)\varphi(t)^T \right)^{-1} \left(\sum_{t=1}^{N} \varphi(t)y(t) \right)$$

This represents the Ordinary Least Squares formula.

We encounter two scenarios:

- If the Hessian is positive definite, $\sum_{t=1}^{N} \varphi(t)\varphi(t)^T$ is invertible and represents a well-shaped paraboloid.
- If the Hessian is singular, $\sum_{t=1}^{N} \varphi(t)\varphi(t)^T$ is not invertible, and the ordinary least-squares formula may not hold. In this case, there are multiple solutions that are all equivalent due to a degenerate paraboloid. Multiple global minima pose a problem because the data were generated by a single system. Causes of multiple minima include:
  - The data record may not adequately represent the underlying physical phenomenon.
  - The selected model might be overly complex, leading to equivalent models describing the same phenomenon.

Second derivative The general expression for the second derivative is:

$$\frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2} = \frac{d}{d\vartheta}\left[\frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2} \right] = \frac{2}{N} \sum_{t=1}^{N} \varphi(t)\varphi(t)^T$$

This matrix must be positive semidefinite.

Definition (*Positive semi-definite matric*)**.** A square matrix M is positive semi-definite if for any nonzero vector x, we have $x^T M x \geq 0$.

In our context, we require:

$$x^T \frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2} x = x^T \frac{2}{N} \sum_{t=1}^{N} \varphi(t)\varphi(t)^T x = \frac{2}{N} \sum_{t=1}^{N} \underbrace{\left(x^T \varphi(t)\right)}_{q(t)} \underbrace{\left(\varphi(t)^T x\right)}_{q(t)} = \frac{2}{N} \sum_{t=1}^{N} q(t)^2 \geq 0$$

Thus, the Hessian is always positive semidefinite.

5.3 ARMA and ARMAX models

Consider the ARMA and ARMAX class of models:

$$\mathcal{M}(\vartheta) : y(t) = \frac{B(z)}{A(z)} u(t-d) + \frac{C(z)}{A(z)} e(t) \qquad e(t \sim WN(0, \lambda^2))$$

Here, we have:

$$\begin{cases} A(z) = 1 - a_1 z^{-1} - a_z z^{-2} - \cdots - a_m z^{-m} \\ B(z) = b_1 + b_2 z^{-1} + b_3 z^{-2} + \cdots + b_p z^{-p+1} \\ C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots - c_n z^{-n} \end{cases}$$

Where:

$$\vartheta = \begin{bmatrix} a_1 & \dots & a_m & b_1 & \dots & b_p & c_1 & \dots & c_n \end{bmatrix}^T$$

We begin by determining the predictor using the long division between $C(z)$ and $A(z)$. Both transfer functions are monic since they are in canonical form. Consequently, the division between the two yields unity after a single step. Thus, we have $E(z) = 1$ and $z^{-k}F(z) = C(z) - A(z)$.

From this outcome, we derive the one-step ahead predictor as:

The one-step ahead prediction error associated with this predictor equals:

$$\hat{y}(t|t-1) = \frac{F(z)}{C(z)}z^{-1}y(t) + \frac{B(z)}{C(z)}u(t-d) = \frac{C(z)-A(z)}{C(z)}y(t) + \frac{B(z)}{C(z)}u(t-d)$$

The one-step ahead prediction error associated with this predictor can be expressed as:

$$\begin{aligned}\varepsilon(t|t-1,\vartheta) &= y(t) - \hat{y}(t|t-1,\vartheta) = y(t) - \frac{C(z)-A(z)}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d) \\ &= y(t)\left(1 - \frac{C(z)A(z)}{C(z)}\right) - \frac{B(z)}{C(z)}u(t-d) \\ &= \frac{A(z)}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d)\end{aligned}$$

The cost function related to this predictor then becomes:

$$J_N(\vartheta) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t|t-1,\vartheta)^2$$

This function is no longer quadratic in ϑ, necessitating the handling of a nonlinear optimization problem. To address this problem, we employ an iterative approach:

1. Initialize the algorithm with an initial estimate ϑ_1 randomly chosen.
2. Define an update rule $\vartheta^{i+1} = f\left(\vartheta^i\right)$.
3. The sequence of estimates should converge to $\hat{\vartheta}_N$.

However, there are challenges with this procedure, notably the selection of an update rule ensuring convergence and finding appropriate initialization values.

5.3.1 Initialization

We have two main approaches for initialization:

- *Iterative algorithms*: these algorithms are guaranteed to converge to a minimum, albeit potentially a local one.
- *Multiple initializations* (empirical approach): here, we start the problem with multiple random guesses and apply the update rule to each. Eventually, we obtain several results, from which we select the solution with the minimum value of $J_N(\vartheta)$. While this method increases the chance of finding the best result, it also escalates computational costs, especially with a higher number of attempts.

5.3.2 Update rule

The most commonly used update rule is Newton's method. Let $V^i(\vartheta)$ denote the second-order Taylor expansion of $J_N(\vartheta)$ in the vicinity of ϑ^i:

$$V^i(\vartheta) = J_N(\vartheta^i) + (\vartheta - \vartheta^i)\frac{\partial J_N(\vartheta)}{\partial \vartheta} + \frac{1}{2}(\vartheta - \vartheta^i)^T \frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2}(\vartheta - \vartheta^i)$$

This is evaluated at $\vartheta = \vartheta^i$. To locate the minimum of this curve, we set the first derivative equal to zero:

$$\frac{\partial V^i(\vartheta)}{\partial \vartheta} = 0$$

This is done at $\vartheta = \vartheta^{i+1}$. Consequently, we have:

$$\frac{\partial V^i(\vartheta)}{\partial \vartheta} = \frac{\partial J_N(\vartheta)}{\partial \vartheta} + \frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2}(\vartheta - \vartheta^i) = 0$$

This is evaluated at $\vartheta = \vartheta^{i+1}$. Ultimately, we obtain:

$$\vartheta^{i+1} = \vartheta^i - \left[\frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2}\right]^{-1} \frac{\partial J_N(\vartheta)}{\partial \vartheta}$$

This is evaluated at $\vartheta = \vartheta^i$.

Gradient Let's compute the gradient of the cost function:

$$\frac{\partial J_N(\vartheta)}{\partial \vartheta} = \frac{d}{d\vartheta}\left[\frac{1}{N}\sum_{t=1}^{N}\varepsilon(t,\vartheta)^2\right] = \left[\frac{1}{N}\sum_{t=1}^{N}\frac{d}{d\vartheta}\varepsilon(t,\vartheta)^2\right] = \frac{2}{N}\sum_{t=1}^{N}\varepsilon(t,\vartheta)\frac{d\varepsilon(t,\vartheta)}{d\vartheta}$$

Now, let's compute the second-order gradient of the cost function:

$$\begin{aligned}
\frac{\partial^2 J_N(\vartheta)}{\partial \vartheta^2} &= \frac{d}{d\vartheta}\left[\frac{2}{N}\sum_{t=1}^{N}\varepsilon(t,\vartheta)\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right] \\
&= \frac{2}{N}\sum_{t=1}^{N}\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\left(\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right)^T + \underbrace{\frac{2}{N}\sum_{t=1}^{N}\varepsilon(t,\vartheta)\left(\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right)}_{\text{negligible}} \\
&= \frac{2}{N}\sum_{t=1}^{N}\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\left(\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right)^T
\end{aligned}$$

The second part is negligible because the value of $\varepsilon(\vartheta)$ is small. By considering only the first part, the estimate of the Hessian will always be non-negative, ensuring convergence.

Final update rule After all considerations, the final version of the update rule is:

$$\vartheta^{i+1} = \vartheta^i - \left[\frac{2}{N}\sum_{t=1}^{N}\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\left(\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right)^T\right]^{-1}\left[\frac{2}{N}\sum_{t=1}^{N}\varepsilon(t,\vartheta)\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right]$$

This is evaluated at ϑ^i.

Estimation error The estimation error $\varepsilon(t, \vartheta)$ is given by:

$$\begin{aligned}\varepsilon(t,\vartheta) &= \frac{A(z)}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d) \\ &= \frac{1 + a_1 z^{-1} + \cdots + a_m z^{-m}}{1 + c_1 z^{-1} + c_2 z^{-2} + \cdots - c_n z^{-n}}y(t) - \frac{b_1 + b_2 z^{-1} + b_3 z^{-2} + \cdots + b_p z^{-p+1}}{1 + c_1 z^{-1} + c_2 z^{-2} + \cdots - c_n z^{-n}}u(t-d)\end{aligned}$$

We need to compute the derivative with respect to all elements of the vector:

$$\vartheta = \begin{bmatrix} a_1 & \dots & a_m & b_1 & \dots & b_p & c_1 & \dots & c_n \end{bmatrix}^T$$

Parameters of $A(z)$ Let's begin by considering the parameter a_1:

$$\begin{aligned}\frac{\partial \varepsilon(t,\vartheta)}{\partial a_1} &= \frac{\partial}{\partial a_1}\left[\frac{1 + a_1 z^{-1} + \cdots + a_m z^{-m}}{1 + c_1 z^{-1} + c_2 z^{-2} + \cdots - c_n z^{-n}}y(t)\right] - \underbrace{\frac{\partial}{\partial a_1}\left[\frac{B(z)}{C(z)}u(t-d)\right]}_{0} \\ &= \frac{z^{-1}y(t)}{C(z)} \\ &= \frac{1}{C(z)}y(t-1)\end{aligned}$$

Let's define a signal $\alpha(t) = \frac{1}{C(z)}y(t)$, obtaining:

$$\frac{\partial \varepsilon(t,\vartheta)}{\partial a_1} = \alpha(t-1)$$

If we perform the same operation for all the a_i in ϑ, we obtain:

$$\begin{cases} \frac{\partial \varepsilon(t,\vartheta)}{\partial a_1} = \alpha(t-1) \\ \frac{\partial \varepsilon(t,\vartheta)}{\partial a_2} = \alpha(t-2) \\ \vdots \\ \frac{\partial \varepsilon(t,\vartheta)}{\partial a_m} = \alpha(t-m) \end{cases}$$

Parameters of $B(z)$ Let's begin by considering the parameter b_1:

$$\begin{aligned}\frac{\partial \varepsilon(t,\vartheta)}{\partial b_1} &= \underbrace{\frac{\partial}{\partial b_1}\left[\frac{A(z)}{C(z)}y(t)\right]}_{0} - \frac{\partial}{\partial a_1}\left[\frac{b_1 + b_2 z^{-1} + b_3 z^{-2} + \cdots + b_p z^{-p+1}}{1 + c_1 z^{-1} + c_2 z^{-2} + \cdots - c_n z^{-n}}u(t-d)\right] \\ &= -\frac{1}{C(z)}u(t-d)\end{aligned}$$

Let's define a signal $\beta(t) = -\frac{1}{C(z)}u(t)$, obtaining:

$$\frac{\partial \varepsilon(t,\vartheta)}{\partial b_1} = \beta(t-d)$$

If we perform the same operation for all the b_i in ϑ, we obtain:

$$\begin{cases} \frac{\partial \varepsilon(t,\vartheta)}{\partial b_1} = \beta(t-d) \\ \frac{\partial \varepsilon(t,\vartheta)}{\partial b_2} = \beta(t-d-1) \\ \vdots \\ \frac{\partial \varepsilon(t,\vartheta)}{\partial a_p} = \beta(t-d-p+1) \end{cases}$$

Parameters of $C(z)$ Considering the equation:

$$\varepsilon(t) = \frac{A(z)}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d)$$

This can be rewritten as:

$$C(z)\varepsilon(t) = A(z)y(t) - B(z)u(t-d)$$

Let's start by considering the parameter c_1:

$$\begin{aligned}
&\frac{d}{dc_1}\left[C(z)\varepsilon(t)\right] = \frac{d}{dc_1}\left[A(z)y(t) - B(z)u(t-d)\right] \rightarrow \\
&\frac{dC(z)}{dc_1}\varepsilon(t) + \frac{d\varepsilon(t)}{dc_1}C(z) = 0 \rightarrow \\
&z^{-1}\varepsilon(t) + C(z)\frac{\partial\varepsilon(t)}{\partial c_1} = 0 \rightarrow \\
&\frac{\partial\varepsilon(t)}{\partial c_1} = -\frac{z^{-1}\varepsilon(t)}{C(z)}
\end{aligned}$$

Let's define a signal $\gamma(t) = -\frac{1}{C(z)}\varepsilon(t)$, obtaining:

$$\frac{\partial\varepsilon(t,\vartheta)}{\partial c_1} = \gamma(t-1)$$

If we perform the same operation for all the c_i in ϑ, we obtain:

$$\begin{cases}
\frac{\partial\varepsilon(t,\vartheta)}{\partial c_1} = \gamma(t-1) \\
\frac{\partial\varepsilon(t,\vartheta)}{\partial c_2} = \gamma(t-2) \\
\vdots \\
\frac{\partial\varepsilon(t,\vartheta)}{\partial c_m} = \gamma(t-m)
\end{cases}$$

The block diagram for this method is as follows:

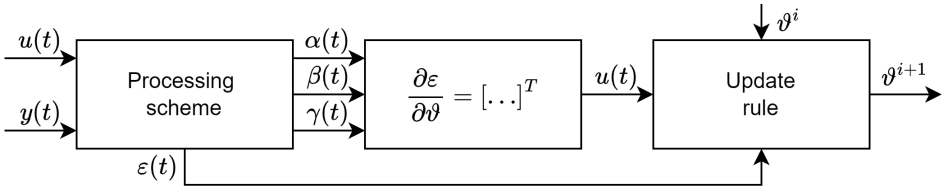


Figure 5.2: ARMAX identification procedure

In summary, the steps for the update rule are as follows:

1. Compute polynomials $A(z,\vartheta^i)$, $B(z,\vartheta^i)$, and $C(z,\vartheta^i)$ at step i.
2. Compute the signals $\varepsilon(t,\vartheta^i)$, $\alpha(t,\vartheta^i)$, $\beta(t,\vartheta^i)$, and $\gamma(t,\vartheta^i)$.
3. Compute the gradient $\frac{\partial\varepsilon(t,\vartheta)}{\partial\vartheta}$.

4. Update the parameter estimate:

$$\vartheta^{i+1} = \vartheta^i - \left[\frac{2}{N}\sum_{t=1}^{N}\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\left(\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right)^T\right]^{-1}\left[\frac{2}{N}\sum_{t=1}^{N}\varepsilon(t,\vartheta)\frac{d\varepsilon(t,\vartheta)}{d\vartheta}\right]$$

After acquiring the inverse part via Quasi-Newton optimization, we strive for a balanced approach that prioritizes both robustness and accuracy during the estimation procedure. In instances where the estimate of the Hessian is numerically poorly conditioned, we employ:

$$\frac{\partial J_N(\vartheta)}{\partial \vartheta^2} = \frac{2}{N}\sum\frac{\partial\varepsilon(t,\vartheta)}{\partial\vartheta}\frac{\partial\varepsilon(t,\vartheta)}{\partial\vartheta}^T + \delta I$$

Here, δ, referred to as the tuning parameter, needs to be kept sufficiently small to prevent significant fluctuations in the final outcome.

CHAPTER 6

Model validation

6.1 Introduction

We aim to establish a metric for assessing the efficacy of a model, denoted as $\mathcal{M}(\hat{\vartheta}_N)$, in representing a specific process, $y(t)$. Given the complexities associated with finite N, we will focus our analysis on the more tractable scenario of large N ($N \to \infty$).

Small number of experiments In the scenario of a few experiments, each experiment yields a singular outcome for every considered time instant:

$$\mathcal{D}\left\{(u(1,\bar{s}),y(1,\bar{s})),(u(2,\bar{s}),y(2,\bar{s})),\dots\right\}$$

Consequently, all predictors are contingent upon $\bar{s}$ independently at each prediction step, with the prediction error also being reliant on $\bar{s}$. When considering multiple experiments individually, divergent outcomes are obtained.

The points $\hat{\vartheta}_N(s)$ derived from each experiment constitute a collection of points. Consequently, this scenario presents challenges as it results in multiple valid minima points.

Large number of experiments In the scenario where $N \to \infty$, multiple curves converge, leading to the emergence of a well-defined point. Consequently, in this case, the cost function $J_N(\vartheta, s)$ converges to a single asymptotic curve, with the corresponding minima gradually approaching each other.

Theorem 6.1.1. *Under the given assumptions, as $N \to \infty$, we observe that:*

$$J_N(\vartheta, s) \to \hat{J}(\vartheta) = \mathbb{E}\left[\varepsilon(t,\vartheta,s)^2\right]$$

Furthermore, defining $\Delta = \left\{\vartheta^* : \bar{J}(\vartheta^*) \leq \bar{J}(\vartheta) \quad \forall\vartheta\right\}$ as the set of global minimum points of $\bar{J}(\varepsilon)$, it follows that:

$$\hat{\vartheta}_N(s) \to \Delta$$

Corollary 6.1.1.1. *If $\Delta = \{\vartheta^*\}$, i.e. $\bar{J}(\vartheta)$ possesses a unique minimum, then:*

$$\hat{\vartheta}_N(s) \to \vartheta^*$$

Our aim, considering the true model of the system $\mathcal{M}(\vartheta^\circ)$, is to converge towards $\hat{\vartheta}_N(s) \to \vartheta^\circ$. This equivalence suggests that $\vartheta^\circ \in \Delta$, if $\Delta = \{\vartheta^*\}$, then $\vartheta^\circ = \vartheta^*$.

To establish that $\vartheta^\circ \in \Delta$, we present the following theorem:

Theorem 6.1.2. $\vartheta^\circ \in \Delta$

Proof. Consider the prediction error for a generic ϑ:

$$\varepsilon(t, \vartheta) = y(t) - \hat{y}(t|t-1, \vartheta)$$

By adding and subtracting the predictor for ϑ°, we obtain:

$$\varepsilon(t, \vartheta) = \underbrace{y(t) + \hat{y}(t|t-1, \vartheta^\circ)}_{e}(t) - \hat{y}(t|t-1, \vartheta^\circ) - \hat{y}(t|t-1, \vartheta)$$

The expected value of $\varepsilon(t, \vartheta)$ is then computed as follows:

$$\mathbb{E}\left[\varepsilon(t,\vartheta)^2\right] = \underbrace{\mathbb{E}\left[e(t)^2\right]}_{\lambda^2} + \mathbb{E}\left[\left(\hat{y}(t|t-1,\vartheta^\circ) - \hat{y}(t|t-1,\vartheta)\right)^2\right] + \\ + 2\underbrace{\mathbb{E}\left[e(t)\left(\hat{y}(t|t-1,\vartheta^\circ) - \hat{y}(t|t-1,\vartheta)\right)\right]}_{0}$$

Consequently, we have:

$$\mathbb{E}\left[\varepsilon(t,\vartheta)^2\right] = \lambda^2 + \mathbb{E}\left[\left(\hat{y}(t|t-1,\vartheta^\circ) - \hat{y}(t|t-1,\vartheta)\right)^2\right]$$

Here, λ^2 is termed the unavoidable error term, independent of ϑ. On the other hand, the second term is ϑ-dependent and is greater than or equal to zero. Hence, it follows that:

$$\bar{J}(\vartheta^\circ) \leq \bar{J}(\vartheta) \quad \forall \vartheta$$

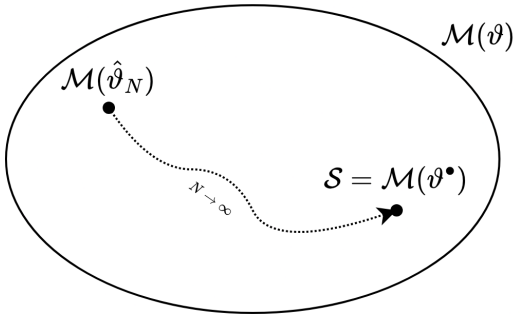
Therefore, if $J \in \mathcal{M}(\vartheta)$, then the parametric identification ensures that $\mathcal{M}(\hat{\vartheta}_N) \to \mathcal{S}$. □

6.1.1 Possible cases

There exist four potential scenarios that may occur:

1. $\mathcal{S} \in \mathcal{M}(\vartheta)$ and Δ is a singleton:

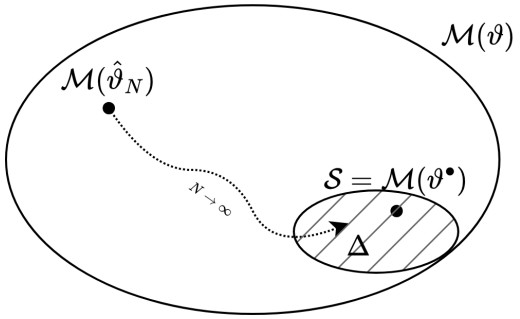
$$\mathcal{S} = \mathcal{M}(\vartheta^\circ) \qquad \Delta = \{\vartheta^\circ\}$$



As N increases, we approach the correct model but only achieve a satisfactory approximation.

2. $\mathcal{S} \in \mathcal{M}(\vartheta)$ and Δ is not a singleton:

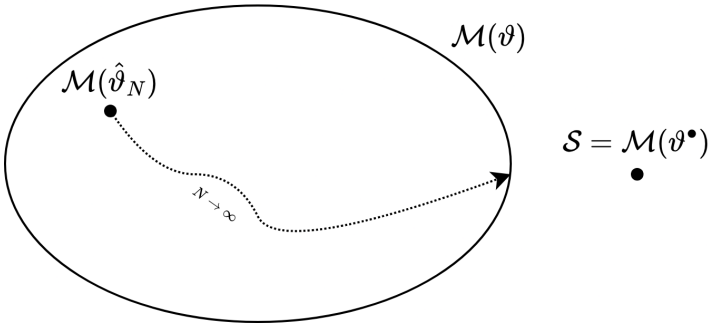
$$\mathcal{S} \in \mathcal{M}(\vartheta^\circ) \qquad \Delta = \{\vartheta_1^\circ, \vartheta_2^\circ, \ldots, \vartheta_n^\circ\}$$



With increasing N, we approach the set of global minima comprising the correct model, but with less precision compared to the first scenario.

3. $\mathcal{S} \notin \mathcal{M}(\vartheta)$ and Δ is a singleton:

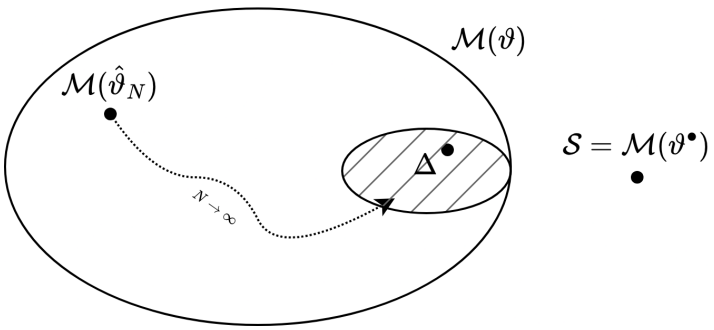
$$\mathcal{S} \notin \mathcal{M}(\vartheta^\circ) \qquad \Delta = \{\vartheta^\circ\}$$



As N increases, we approach the correct model, but we are constrained by the limits of the considered set $\mathcal{M}(\vartheta^\circ)$. Consequently, we obtain a result closest to correctness but not entirely accurate.

4. $\mathcal{S} \notin \mathcal{M}(\vartheta)$ and Δ is not a singleton:

$$\mathcal{S} \notin \mathcal{M}(\vartheta^\circ) \qquad \Delta = \{\vartheta_1^\circ, \vartheta_2^\circ, \ldots, \vartheta_n^\circ\}$$



With increasing N, we approach the set of global minima encompassing the correct model, albeit with less precision than the second scenario, as the set of points is contained within the model, but the sought-after model is not.

6.2 Model order selection

We have defined the models as:

$$\mathcal{M}(\vartheta) = \{M(\vartheta), \vartheta \in \Theta \subseteq \mathbb{R}^{n_\theta}\}$$

This representation constitutes a fixed-order model class, implying that we cannot guarantee $\mathcal{S} \in \mathcal{M}(\vartheta)$. It's important to note that we lack a method to expand $\mathcal{M}(\vartheta)$ if the system lies outside it.

For simplicity, let's assume $n_\theta = n = m = p$, meaning a single parameter describes the model order. The process for selecting the parameter n_θ begins by choosing an initial value and incrementing it if necessary, up to a maximum of $n_{\max}$:

Algorithm 1 Model order selection algorithm

1: $n \leftarrow 1$
2: **repeat**
3: $\quad \mathcal{M}(\vartheta) \leftarrow \{M(\vartheta), \vartheta \in \Theta \subseteq \mathbb{R}^{n_\theta}\}$
4: $\quad \hat{\vartheta}_N^{(n)} \leftarrow \text{argmin}_\vartheta \, J_N(\vartheta)$
5: **until** $n = n_{max}$

As the model order increases, the total cost decreases. When we achieve a total cost of approximately λ^2, we have reached the optimal order for the selected model.

Overfitting occurs with increased order, eventually leading to a cost of zero. This happens when $n_\vartheta = N$.

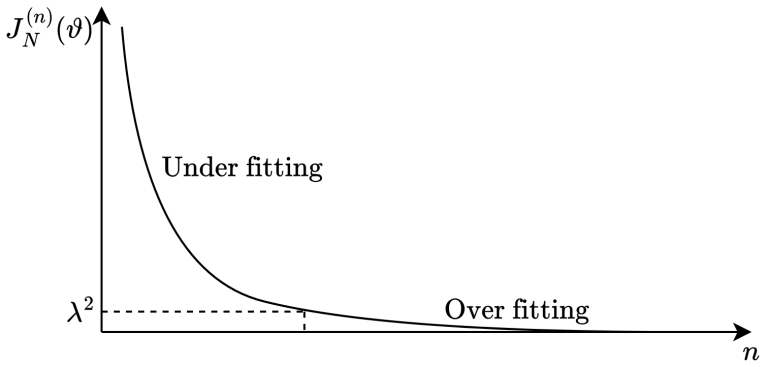


Figure 6.1: Cost function with respect to the order

However, this method isn't ideal because the value of λ^2 is generally unknown. Possible alternatives include:

1. Whiteness test on residuals.
2. Cross-validation.
3. Identification with model order penalties.

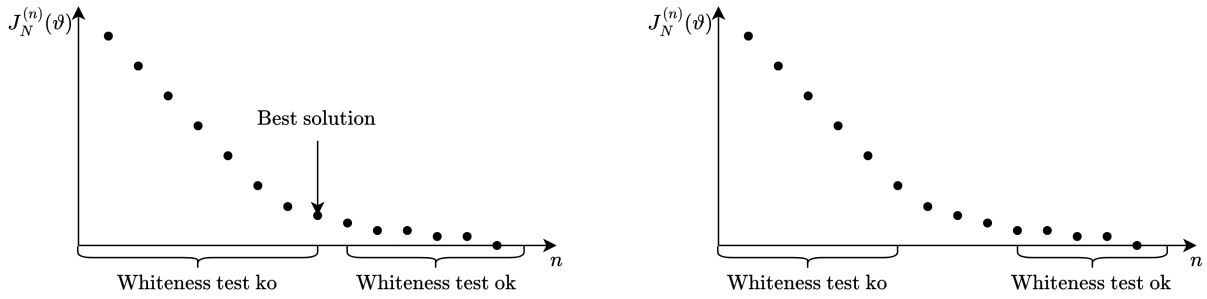
6.2.1 Whiteness test on residuals

We understand that if $\mathcal{S} \in \mathcal{M}(\vartheta)$ for a certain n, then $\hat{\vartheta}_N^{(n)} \approx \vartheta^\circ$ and $\varepsilon(t, \hat{\vartheta}_N^{(n)}) \approx e(t)$, where $\varepsilon(t, \hat{\vartheta}_N^{(n)})$ approximates White Noise.

To determine the appropriate model order n, we assess whether $\varepsilon(t, \hat{\vartheta}_N^{(n)})$ exhibits whiteness. We select the first n for which the whiteness test yields favorable results.

The whiteness test typically involves a covariance check, which should be non-zero at $\tau = 0$ and zero otherwise. Alternatively, a spectral density check can be performed: White Noise exhibits a constant spectrum value.

Problems The limitation of this method arises from the fact that the covariance at time instants other than zero may vary, making it challenging to determine whether a process exhibits White Noise characteristics. Similarly, the spectral check may show slight variations across different instants instead of a perfectly horizontal line.



(a) Ideal situation

(b) Real situation

Figure 6.2: Whiteness test outcomes

In the real scenario, we can narrow down the set of candidate model orders, but identifying the exact order remains challenging.

6.2.2 Cross-validation

Let's consider a scenario where we have a total of N data points, which we divide into two halves:

- Identification dataset: $\left\{y(1), y(2), \ldots, y(\frac{N}{2})\right\}$.
- Validation dataset: $\left\{y\left(\frac{n}{2}+1\right), y\left(\frac{n}{2}+2\right), \ldots, y(N)\right\}$.

Algorithm 2 Model order selection algorithm

1: **for** $n = 1$ **to** $n = n_{max}$ **do**
2: find $\hat{\vartheta}_{N/2} \leftarrow \operatorname{argmin}_\vartheta J_N(\vartheta) = \operatorname{argmin}_\vartheta \frac{1}{N/2} \sum_{t=1}^{N/2} \left(y(t) - \hat{y}(t|t-1, \vartheta)\right)^2$
3: evaluate $J_V\left(\hat{\vartheta}_{N/2}^{(n)}\right) \leftarrow \frac{1}{N/2} \sum_{t=n/2+1}^{N} \left(y(t) - \hat{y}(t|t-1, \hat{\vartheta}_{N/2}^{(n)})\right)^2$
4: **end for**
5: choose n minimizing J_V

Note that the $\frac{N}{2}$ used in the evaluation step is not the same as the $\frac{N}{2}$ used in the preceding step. This distinction is important because, in the evaluation step, we only consider the second portion of the available N data using the value of $\hat{\vartheta}_N$ found with the other half of the data. We then compute the variance of the prediction error in the second dataset.

The difference between the costs computed on the whole dataset and half of it is significant. In the former case, the cost keeps decreasing, while in the latter, it initially decreases and then rises again.

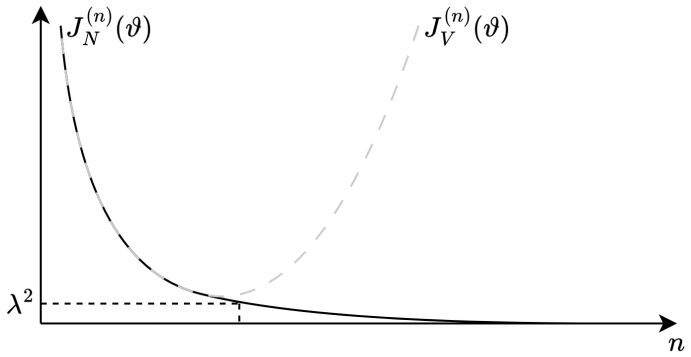


Figure 6.3: Difference between the cost functions J_N and J_V

Problems The primary challenge associated with this method is the requirement for many elements in the initial dataset N to ensure sufficient data in the considered half.

6.2.3 Identification with model order penalties

When the dataset's size is insufficient, identification with model order penalties becomes impractical. Unlike cross-validation, where the same cost is applied to different data, the penalized-cost approach employs a distinct cost on the same dataset.

Instead of minimizing $J_N(\vartheta)$ directly, we aim to minimize one of the following measures:

- *Final prediction error*: we seek to minimize $\mathbb{E}[J_N(\vartheta)]$ with respect to noise realizations. As computing the expected value directly isn't feasible, we minimize the final prediction error:

$$\mathrm{FPE}(n) = \frac{N+n}{N-n} J_N\left(\hat{\vartheta}_N^{(n)}\right)$$

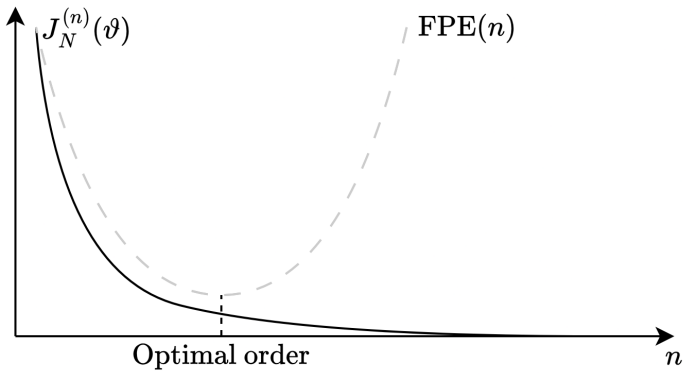


Figure 6.4: Difference between the cost functions J_N and and final prediction error

- *Akaike information criterion*:

$$\mathrm{AIC}(n) = \ln\left(J_N(\hat{\vartheta}_N^{(n)})\right) + 2\frac{n}{N}$$

  The penalty on the current order n increases with n, while the cost decreases as the order increases. Balancing these two components is crucial for optimal results.

- *Minimum description length*:

$$\mathrm{MDL}(n) = \ln\left(J_N(\hat{\vartheta}_N^{(n)})\right) + \ln(N)\frac{n}{N}$$

  Minimum description length operates similarly to Akaike information criterion.

FPE and AIC We begin with:

$$\begin{aligned}
\ln(\mathrm{FPE}) &= \ln\left(\frac{N+n}{N-n} J_N\left(\hat{\vartheta}_N^{(n)}\right)\right) \\
&= \ln\left(\frac{1+\frac{n}{N}}{1-\frac{n}{N}} J_N\left(\hat{\vartheta}_N^{(n)}\right)\right) \\
&= \ln\left(1+\frac{n}{N}\right) - \ln\left(1-\frac{n}{N}\right) + \ln\left(J_N\left(\hat{\vartheta}_N^{(n)}\right)\right)
\end{aligned}$$

Given that $\frac{n}{N}$ is usually small, we approximate:

$$\begin{aligned}
\ln(\mathrm{FPE}) &= \ln\left(1+\frac{n}{N}\right) - \ln\left(1-\frac{n}{N}\right) + \ln\left(J_N\left(\hat{\vartheta}_N^{(n)}\right)\right) \\
&= \frac{n}{N} - \left(-\frac{n}{N}\right) + \ln\left(J_N\left(\hat{\vartheta}_N^{(n)}\right)\right) \\
&= 2\frac{n}{N} + \ln\left(J_N\left(\hat{\vartheta}_N^{(n)}\right)\right) \\
&= \mathrm{AIC}(n)
\end{aligned}$$

Thus, both methods yield the same information criterion. Consequently, minimizing one metric also minimizes the other.

AIC and MDL The primary distinction between the Akaike Information Criterion (AIC) and Minimum Description Length (MDL) lies in the coefficient of the penalty term. AIC maintains a fixed penalty value of two, while MDL incorporates a variable term $\ln(N)$.

Consequently, if $\ln(N) > 2$, indicating $N > 8$, MDL penalizes the model order more severely than AIC.

In cases where $\mathcal{S} \in \mathcal{M}(\vartheta)$ and $\mathcal{M}(\vartheta)$ represents the set of ARX models, MDL is the preferred choice. However, in general scenarios where $\mathcal{S} \notin \mathcal{M}(\vartheta)$, a slight overfitting is acceptable, making AIC the preferred criterion.

CHAPTER 7

Non-parametric identification

7.1 Introduction

If our interest lies in certain features of the stochastic process $y(t)$, such as $\mathbb{E}[y(t)]$ or $\gamma_y(\tau)$, it is unnecessary to identify the complete model $M(\vartheta)$ and estimate these features directly from it.

For a stationary stochastic process $y(t)$, data-driven estimation is feasible for estimating $\mathbb{E}[y(t)]$, $\gamma_y(\tau)$, and $\Gamma_y(\omega)$.

Definition (*Estimator correctness*)**.** An estimator $\hat{Q}_N$ of Q is considered correct if:

$$\mathbb{E}\left[\hat{Q}_N\right] = Q$$

This property is also known as unbiased estimation.

Definition (*Estimator consistency*)**.** An estimator $\hat{Q}_N$ of Q is regarded as consistent if:

$$\mathbb{E}\left[\left(\hat{Q}_N - Q\right)^2\right] \to 0$$

as the number of samples N tends to infinity.

7.2 Mean estimation

For a stationary stochastic process $y(t)$ and a dataset $\mathcal{D}_N = \{y(1), y(2), \ldots, y(n)\}$, our goal is to estimate the mean:

$$m_y = \mathbb{E}\left[y(t)\right]$$

One feasible approach is to define the estimated mean as:

$$\hat{m}_N = \frac{1}{N}\sum_{t=1}^{N} y(t)$$

7.2.1 Correctness

Let's examine the expected value of $\mathbb{E}[\hat{m}_N]$:

$$\mathbb{E}[\hat{m}_N] = \mathbb{E}\left[\frac{1}{N}\sum_{t=1}^{N} y(t)\right] = \frac{1}{N}\sum_{t=1}^{N} \mathbb{E}\underbrace{[y(t)]}_{m_y} = \frac{1}{N} N m_y = m_y$$

Thus, this estimator is correct.

7.2.2 Consistency

Let's consider a process $y(t,s) = v(s)$, where $v \sim \mathcal{N}(0,1)$:

$$\hat{m}_N = \frac{1}{N}\sum_{t=1}^{n} y(t,\bar{s}) = \frac{1}{N} N v(\bar{s}) = v(\bar{s})$$

Computing the expected value of the error:

$$\mathbb{E}\left[(\hat{m}_N - m_y)^2\right] = \mathbb{E}\left[\left(\frac{1}{N}\sum_{t=1}^{n} y(t,\bar{s}) - 0\right)^2\right] = \mathbb{E}\left[\left(\frac{1}{N} N v(\bar{s})\right)^2\right] = \mathbb{E}\left[v(\bar{s})^2\right] = 1$$

Since this value is independent of n, the estimator is not consistent.

7.2.3 Theorem

Theorem 7.2.1. *The estimator $\hat{m}_n$ is correct and consistent for ARMA processes.*

7.3 Variance estimation

Consider a process with zero mean, where the variance is defined as:

$$\gamma(\tau) = \mathbb{E}[y(t)y(t-\tau)]$$

Similar to mean estimation, we have a dataset $\mathcal{D}_N = \{y(1), y(2), \ldots, y(n)\}$, and the covariance is computed as the covariance of all data.

One possible method to estimate the covariance is:

$$\hat{\gamma}_N(|\tau|) = \frac{1}{N - |\tau|}\sum_{t=1}^{N-|\tau|} y(t)y(t+|\tau|)$$

Here, $|\tau| \leq N-1$. The values of the covariance for different τ are:

$$\begin{cases} \hat{\gamma}_N(0) = \frac{1}{N}\left(y(1)^2 + y(2)^2 + \cdots + y(N)^2\right) \\ \hat{\gamma}_N(1) = \frac{1}{N-1}\left(y(1)y(2) + y(2)y(3) + \cdots + y(N-1)y(N)\right) \\ \vdots \\ \hat{\gamma}_N(N-1) = y(1)y(N) \end{cases}$$

As τ increases, we might encounter difficulties in computing accurate covariance due to fewer data points. Therefore, in practice, it's preferable to have $|\tau| \ll N-1$ ($|\tau| \approx 2\%$ of $N-1$) to ensure reliable estimation.

7.3.1 Correctness

Let's examine the expected value of $\gamma(\tau)_N$:

$$\mathbb{E}\left[\gamma(\tau)_N\right] = \mathbb{E}\left[\frac{1}{N-\tau}\sum_{t=1}^{N-\tau} y(t)y(t+\tau)\right] = \frac{1}{N-\tau}\left(N-\tau\right)\underbrace{\mathbb{E}\left[y(t)y(t+\tau)\right]}_{\gamma_y(\tau)} = \gamma_y(\tau)$$

Thus, this estimator is correct.

7.3.2 Consistency

Similar to the mean, since this is not a function of n, the estimator does not depend on the number of samples. Therefore, it is not consistent.

7.3.3 Theorem

Theorem 7.3.1. *The estimator $\gamma(\tau)_N$ is correct and consistent for ARMA processes.*

7.4 Spectral density estimation

Given a stationary stochastic process $y(t)$ and a dataset $\mathcal{D}_N = \{y(1), y(2), \ldots, y(n)\}$, we aim to estimate the spectral density:

$$\Gamma_y(\omega) = \sum_{\tau=-\infty}^{+\infty} \gamma_y(\tau)e^{-j\omega\tau}$$

We can estimate this value as:

$$\hat{\Gamma}_N(\omega) = \sum_{\tau=-N+1}^{N-1} \hat{\gamma}_N(\tau)e^{-j\omega\tau}$$

The sources of approximation lie in the limits of the sum and the function $\hat{\gamma}_N$.

Alternative estimator We can consider an alternative estimator $\hat{\gamma}'_N(\tau)$ instead of $\hat{\gamma}_N(\tau)$:

$$\hat{\gamma}'_N(\tau) = \frac{1}{N}\sum_{\tau=-N+1}^{N-1} y(t)y(t+\tau)$$

With this alternative estimator, we can prove that:

$$\hat{\Gamma}'_N(\omega) = \sum_{\tau=-N+1}^{N-1} \hat{\gamma}'_N(\tau)e^{-j\omega\tau} = \frac{1}{N}\left|\sum_{\tau=-N+1}^{N-1} y(t)e^{-j\omega\tau}\right|^2$$

The last part is known as the Fast Fourier Transform (FFT), which can be computed very efficiently.

7.4.1 Correctness

Now, let's check the correctness of the estimator:

$$\mathbb{E}\left[\hat{\Gamma}_N(\omega)\right] = \mathbb{E}\left[\sum_{\tau=-N+1}^{N-1} \hat{\gamma}_N(\tau)e^{-j\omega\tau}\right] = \sum_{\tau=-N+1}^{N-1} \underbrace{\mathbb{E}\left[\hat{\gamma}_N(\tau)\right]}_{\gamma_y(\tau)} e^{-j\omega\tau} = \sum_{\tau=-N+1}^{N-1} \gamma_y(\tau)e^{-j\omega\tau} \neq \Gamma_y(\tau)$$

This discrepancy arises because the sum's limits are not infinite as in the standard spectrum. However, we can assert that $\mathbb{E}\left[\hat{\Gamma}_N(\omega)\right]$ tends to the real spectrum as N tends to infinity. Therefore, with a large dataset, this estimator is correct, also termed asymptotically correct.

7.4.2 Consistency

This estimator remains inconsistent even for ARMA processes, resulting in a spectrum with significant noise.

To improve the final result, we can employ a technique known as regularization. By dividing the dataset into four equal sections, we obtain estimators for each subset:

$$\hat{\Gamma}^{(1)}_{\frac{N}{4}}(\omega) \qquad \hat{\Gamma}^{(2)}_{\frac{N}{4}}(\omega) \qquad \hat{\Gamma}^{(3)}_{\frac{N}{4}}(\omega) \qquad \hat{\Gamma}^{(4)}_{\frac{N}{4}}(\omega)$$

Then, we compute a new estimator as the average of these four:

$$\tilde{\hat{\Gamma}}_N(\omega) = \frac{1}{4}\sum_{i=1}^{4} \hat{\Gamma}^{(i)}_{\frac{N}{4}}(\omega)$$

This results in a smoother spectral density function.

It's worth noting that:

$$\mathbb{E}\left[\left(\tilde{\hat{\Gamma}}_N(\omega) - \Gamma_y(\omega)\right)^2\right] \approx \frac{1}{4}\mathbb{E}\left[\left(\hat{\Gamma}_N(\omega) - \Gamma_y(\omega)\right)^2\right]$$

Increasing the number of sectors will increase bias and decrease variance.

7.5 Model pre-processing

Inclination For non-stationary processes, a method to achieve stationarity involves:

1. Identifying and quantifying any clear drift present in the time series by computing the angle of the drift.
2. Removing the drift from the data.

This process transforms a non-stationary process into a stationary one.

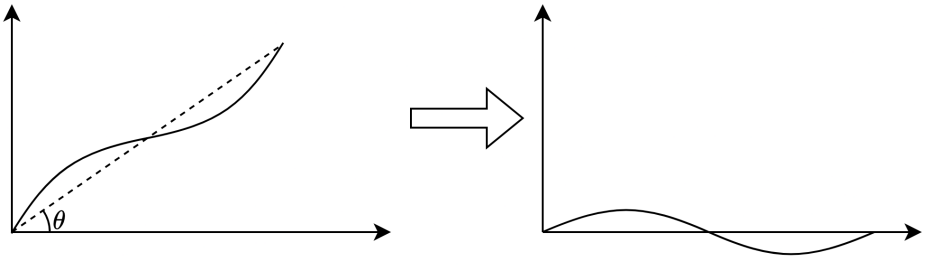


Figure 7.1: Non-stationary process to stationary process

Since the drift may not be readily predictable, decomposing the process into a trend and a stationary stochastic process is necessary. Subsequently, the trend can be eliminated. If the trend follows a linear expression, such as $kt + q$, where k and q are parameters, the expectation of the process can be expressed as:

$$\mathbb{E}\left[y(t)\right] = \mathbb{E}\left[y'(t) - kt - q\right] = 0$$

Following this, the least squares problem can be formulated to minimize:

$$\min_{k,q} \frac{1}{N} \sum_{t=1}^{N} \left(y'(t) - kt - q\right)^2$$

Upon solving this problem, estimates for $\hat{k}$ and $\hat{q}$ are obtained.

Seasonality For processes exhibiting seasonality, in addition to trends, we may encounter periodic variations. To handle seasonality, the process can be decomposed into a seasonal component $s(t)$ and a stationary stochastic process.

Determining the period of seasonality, denoted as T, is crucial. It represents the duration after which the seasonality pattern repeats:

$$s(t) = s(t + Kt) \qquad k \in \mathbb{Z}$$

The period T can be identified by examining the frequency domain and identifying the highest frequency, then computing its inverse.

Once the period T is determined, we can aggregate observations occurring at the same time within each period. This involves averaging all observations at corresponding time instants across multiple periods:

$$\hat{s}(t) = \frac{1}{M} \sum_{h=1}^{M} y(t + hT) \qquad t = 1, 2, \ldots, N$$

Here, M represents the number of periods observed in the dataset. This averaging process yields an estimate for $s(t)$ at each time instant within the season.

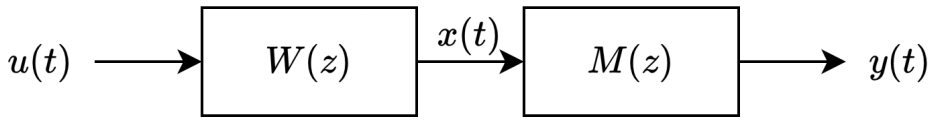
APPENDIX A

Transfer functions

A.1 Operations with transfer functions

Let $W(z)$ and $M(z)$ denote the transfer functions of linear digital filters.

Series connection When connecting these transfer functions in series, the resulting block diagram is as follows:



This configuration yields the following equations:

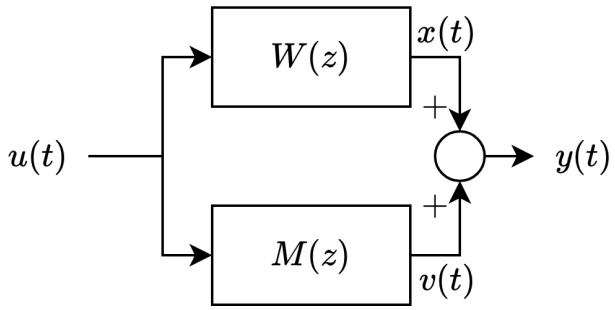
$$\begin{cases} y(t) = M(z)x(t) \\ x(t) = W(z)u(t) \end{cases}$$

By substitution, we obtain:

$$\begin{aligned} y(t) &= M(z)\left[W(z)u(t)\right] \\ G &= M(z)W(z)u(t) \\ &= V(z)u(t) \end{aligned}$$

Hence, the transfer function of n blocks in series is the product of the transfer functions of each block.

Parallel connection When connecting the transfer functions in parallel, the block diagram appears as follows:



The output signal is the sum of the two signals:

$$\begin{aligned} y(t) &= x(t) + v(t) \\ &= W(z)u(t) + M(z)u(t) \\ &= (W(z) + M(z))\, u(t) \\ &= V(z)u(t) \end{aligned}$$

Consequently, the transfer function of n blocks in parallel is the sum of the transfer functions of each block.

A.2 Stability

Consider $W(z)$ as the transfer function of linear digital filters:

- Zeros of $W(z)$ are values of z where $W(z) = 0$. These are the roots of the numerator and are represented graphically with an x.
- Poles of $W(z)$ are values of z where $W(z)^{-1} = 0$. These are the roots of the denominator and are denoted on the graph with a circle.

Theorem A.2.1 (*Asymptotic stability*)**.** *A linear digital filter with transfer function $W(z)$ is asymptotically stable if and only if all its poles are strictly inside the unit circle in the complex plane.*

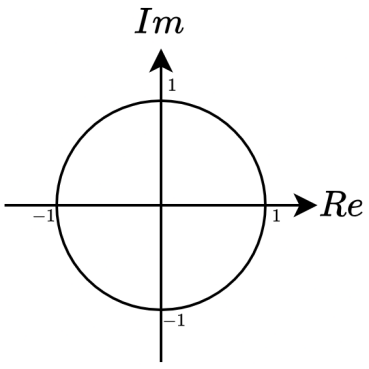


Figure A.1: Unit circle in the complex plane

Definition (*Minimum-phase transfer function*)**.** If the zeros of $W(z)$ are also strictly inside the unit circle, $W(z)$ is termed minimum-phase.

Output stability

Theorem A.2.2. *The steady-state output $y(t)$ is stationary if and only if:*

1. *$v(t)$ is stationary.*
2. *$F(z)$ is asymptotically stable.*

Theorem A.2.3. *There is only one stationary output corresponding to the steady-state solution. However, if $F(z)$ is asymptotically stable, then all potential outputs obtained for different initializations of the digital filter $F(z)$ converge asymptotically to the steady-state solution, i.e., to the stationary output.*

A.2.1 Models's transfer functions

Moving Average An MA(n) process exhibits the following characteristics:

- It possesses n nontrivial zeros.
- It has n poles, all located at the origin.

This is a consequence of all Moving Average processes being generated by asymptotically stable filters.

AutoRegressive An AR(m) process is characterized by:

- m zeros, all positioned at the origin.
- n nontrivial poles.

Consequently, these processes are commonly referred to as all-poles processes.

ARMA An ARMA(m, n) process displays the following features:

- m nontrivial poles.
- n nontrivial zeros.

It's noteworthy that if the input signal is a stationary stochastic process, and the transfer function leading to the output is asymptotically stable, then the resulting process is also a stochastic stationary one.

A.3 Operatorial representation

Definition (*Backward shift operator*)**.** The backward shift operator z^{-1} is defined as:

$$z^{-1} \cdot x(t) = x(t-1)$$

Definition (*Forward shift operator*)**.** The forward shift operator z is defined as:

$$z \cdot x(t) = x(t+1)$$

The properties of the backward and forward shift operators are:

1. Linearity:
$$z^{-1}\left(ax(t)+by(t)\right)=ax(t-1)+by(t-1)$$

2. Recursive application:
$$z^{-1}\left(z^{-1}\left(z^{-1}\left(x(t)\right)\right)\right)=z^{-1}\left(z^{-1}\left(x(t-1)\right)\right)=z^{-1}\left(x(t-2)\right)=x(t-3)=z^{-3}x(t)$$

3. Linear composition:
$$\left(az^{-1}+bz+cz^{-3}+dz^{2}\right)=ax(t-1)+bx(t+1)+cx(t-3)+dx(t+2)$$