Lecture Manuscript

Digital Signal Processing

– TSRT78 HT 2023 –

Kailai Li

Division of Automatic Control
Department of Electrical Engineering
Linköping University

kailai.li@liu.se

December 6, 2023

Contents

Chapter 1

Introduction

Please refer to the course page:

- `https://isy.gitlab-pages.liu.se/rt/en/courses/TSRT78/`

and the textbooks:

- Fredrik Gustafsson, Lennart Ljung, and Mille Millnert. “Signal Processing”, Studentlitteratur, 2011.
- Gunnarsson, Fredrik, Fredrik Gustafsson, and Fredrik Tjärnström. “Signal Processing: Exercises”, Studentlitteratur, 2010.

Chapter 2

Frequency Models

2.1 Continuous-Time Signals

Given a continuous-time signal $x(t)$ with $-\infty < t < \infty$, we now aim to model it in frequency domain. For periodic signals, we introduce modeling using Fourier series. If the signal is not periodic, one can use the Fourier transform. Note that Fourier transform can also be applied to periodic signals.

2.1.1 Fourier series

If $x(t)$ is periodic, namely, $x(t + kT_\mathsf{p}) = x(t)$, with $k \in \mathbb{Z}$ being an integer and T_p the period, we can represent the signal using Fourier series (trigonometric series) as follows

$$x(t) = \sum_{n=-\infty}^{\infty} \mathbb{X}_\mathsf{FC}[n] e^{in\omega_\mathsf{p}t}\,, \tag{2.1}$$

with $\omega_\mathsf{p} = 2\pi/T_\mathsf{p}$ being the fundamental (angular) frequency. The complex sinusoid components of a positive integer multiple of the fundamental frequency are called harmonics. Their amplitudes are given by the Fourier coefficients

$$\mathbb{X}_\mathsf{FC}[n] = \frac{1}{T_\mathsf{p}} \int_0^{T_\mathsf{p}} x(t) e^{-in\omega_\mathsf{p}t}\, \mathrm{d}t\,. \tag{2.2}$$

Essentially, Fourier series represents a periodic signal as a superposition of complex sinusoid components of fundamental frequency and harmonics.

Example 2.1.1. Given a real-value signal $x(t)$ of period T_p, its Fourier series representation follows

$$\begin{aligned}
x(t) &= \sum_{n=-\infty}^{\infty} \mathbb{X}_\mathsf{FC}[n] e^{in\omega_\mathsf{p}t} \\
&= \sum_{n=-\infty}^{\infty} \Big(\Re\{\mathbb{X}_\mathsf{FC}[n]\} + i\Im\{\mathbb{X}_\mathsf{FC}[n]\}\Big)\Big(\cos(n\omega_\mathsf{p}t) + i\sin(n\omega_\mathsf{p}t)\Big) \\
&= \sum_{n=-\infty}^{\infty} \Re\{\mathbb{X}_\mathsf{FC}[n]\}\cos(n\omega_\mathsf{p}t) - \Im\{\mathbb{X}_\mathsf{FC}[n]\}\sin(n\omega_\mathsf{p}t) \\
&:= \sum_{n=-\infty}^{\infty} A_n\Big(\cos\phi_n \cos(n\omega_\mathsf{p}t) - \sin\phi_n \sin(n\omega_\mathsf{p}t)\Big) \\
&= \sum_{n=-\infty}^{\infty} A_n \cos(n\omega_\mathsf{p}t + \phi_n)\,,
\end{aligned}$$

$$\text{with} \quad A_n = \sqrt{(\Re\{\mathbb{X}_\mathsf{FC}[n]\})^2 + (\Im\{\mathbb{X}_\mathsf{FC}[n]\})^2}\,, \quad \phi_n = \arctan\left(\frac{\Im\{\mathbb{X}_\mathsf{FC}[n]\}}{\Re\{\mathbb{X}_\mathsf{FC}[n]\}}\right).$$

2.1.2 Fourier Transform (FT)

Given an periodic/aperiodic signal, we define Fourier Transform (FT)

$$\mathbb{X}(i\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}\,\mathrm{d}t\,, \tag{2.3}$$

to model the signal in frequency domain. To transform the frequency model back to time domain, we have inverse Fourier Transform (IFT)

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbb{X}(i\omega)e^{i\omega t}\,\mathrm{d}\omega\,. \tag{2.4}$$

We show the following examples to demonstrate FT/IFT applied to a few common signals.

Example 2.1.2. We compute FT of the Heaviside exponential function

$$x(t) = e^{-\alpha t}u(t)\,, \quad \text{with} \quad \alpha > 0 \quad \text{and} \quad u(t) = \begin{cases} 1, & t > 0 \\ 0, & t \le 0 \end{cases} \tag{2.5}$$

being the Heaviside step function. Applying (2.3) leads to

$$\begin{aligned} \mathbb{X}(i\omega) &= \int_{-\infty}^{\infty} e^{-\alpha t}u(t)e^{-i\omega t}\,\mathrm{d}t \\ &= \int_{0}^{\infty} e^{-(\alpha+i\omega)t}\,\mathrm{d}t \\ &= \frac{-1}{\alpha+i\omega}e^{-(\alpha+i\omega)t}\Big|_0^\infty \\ &= \frac{1}{\alpha+i\omega}\,. \end{aligned} \tag{2.6}$$

Example 2.1.3. We compute IFT of a rectangular pulse in frequency domain (anti-aliasing filter)

$$\mathbb{H}(i\omega) = \begin{cases} 1 & |\omega| < \pi/T\,, \\ 0 & |\omega| \ge \pi/T\,. \end{cases}$$

Applying (2.4) leads to

$$\begin{aligned} h(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbb{H}(i\omega)e^{i\omega t}\,\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} e^{i\omega t}\,\mathrm{d}\omega \\ &= \frac{1}{2\pi}\frac{1}{it}2i\sin(\pi t/T) \quad \text{(Euler's formula)} \\ &= \frac{\sin(\pi t/T)}{\pi t} = \frac{1}{T}\mathrm{sinc}\Big(\frac{t}{T}\Big)\,. \end{aligned}$$

Here, the sinc function is defined as $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$.

Example 2.1.4. We compute the FT of Dirac delta function $x(t) = \delta(t)$[1]. Applying (2.3) to it induces

$$\mathbb{X}(i\omega) = \int_{-\infty}^{\infty} \delta(t)e^{-i\omega t}\,\mathrm{d}t = 1 \tag{2.7}$$

which indicates that the spectrum of $\delta(t)$ has unit amplitude at all frequencies. As a direct side product, we have the following optional definition for Dirac delta function according to IFT in (2.4)

$$\delta(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}\,\mathrm{d}\omega \quad \text{or} \quad \delta(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}\,\mathrm{d}t\,. \tag{2.8}$$

[1] A few useful properties of Dirac delta function are given in (2.10)

It is also worth mentioning that the definition above shows that Dirac delta function is an even function, namely, $\delta(-t) = \delta(t)$.

Example 2.1.5. As mentioned before, FT can handle periodic function as well. As an example, we apply it to a sine function $x(t) = \sin(t)$, leading to

$$\begin{aligned}
\mathbb{X}(i\omega) &= \int_{-\infty}^{\infty} \sin(t) e^{-i\omega t}\,\mathrm{d}t \\
&= \int_{-\infty}^{\infty} \frac{e^{it} - e^{-it}}{2i} e^{-i\omega t}\,\mathrm{d}t \quad \text{(Euler's formula)} \\
&= \frac{1}{2i}\int_{-\infty}^{\infty} e^{i(1-\omega)t} - e^{-i(1+\omega)t}\,\mathrm{d}t \\
&= \frac{\pi}{i}\left(\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i(\omega-1)t}\,\mathrm{d}t - \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i(\omega+1)t}\,\mathrm{d}t\right) \\
&= i\pi\left(\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i(\omega+1)t}\,\mathrm{d}t - \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i(\omega-1)t}\,\mathrm{d}t\right) \\
&= i\pi\big(\delta(\omega+1) - \delta(\omega-1)\big)
\end{aligned}$$

The last step of the derivation above utilizes the optional definition of Dirac delta function as introduced in (2.8).

2.2 Discrete-Time Signals

Discrete-time signals are typically generated by machines or sampled from continuous-time signals measured by sensors. They are the type of signals that appear in engineering practice (instead of continuous-time representation). It is then of interest for us to model them also in frequency domain. For that, the original definition of Fourier transform in Sec. 2.1.2 needs adaptation according to their discrete-time nature. Suppose a continuous-time signal $x(t)$ is sampled over time with interval T. It can be represented by

$$x_{\mathsf{s}}(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t - kT)\,, \tag{2.9}$$

with $\delta(t)$ being the Dirac delta function. A few useful properties of it are shown below

$$\begin{gathered}
\int_{-\infty}^{\infty} \delta(t)\,\mathrm{d}t = 1 \text{ (integration property)} \\
\int_{-\infty}^{\infty} f(t)\delta(t-T)\,\mathrm{d}t = f(T) \text{ (sampling/sifting property)}\,.
\end{gathered} \tag{2.10}$$

Given continuous-time representation (though the sampled data is discrete) in (2.9), we now aim to apply FT to it and derive its discrete-time counterpart in the following section.

2.2.1 Discrete-time Fourier transform (DTFT)

Applying FT in (2.3) to (2.9) leads to

$$\begin{aligned}\mathbb{X}_{\mathsf{s}}(i\omega) &= \int_{-\infty}^{\infty} x_{\mathsf{s}}(t)\, e^{-i\omega t}\,\mathrm{d}t \\ &= \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(t)\,\delta(t-kT)\,e^{-i\omega t}\,\mathrm{d}t \\ &= \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t)\,\delta(t-kT)\,e^{-i\omega t}\,\mathrm{d}t \\ &= \sum_{k=-\infty}^{\infty} x(kT)\,e^{-i\omega kT}\end{aligned} \tag{2.11}$$

By defining $x(kT) =: x[k]$ (square brackets indicate discrete variables) to represent discrete-time series, we then obtain

$$\mathbb{X}_{\mathsf{s}}(i\omega) = \sum_{k=-\infty}^{\infty} x[k]\,e^{-i\omega Tk}\,, \tag{2.12}$$

being the representation of the sampled signal in frequency domain. Note that the frequency model in (2.12) is periodic with a period of $2\pi/T$ with T being the sampling interval. To emphasize this periodicity, we rewrite the expression by using complex sinusoid $e^{i\omega T}$ as the variable, namely,

$$\mathbb{X}_{\mathsf{s}}(e^{i\omega T}) = \sum_{k=-\infty}^{\infty} x[k]\,e^{-i\omega kT}\,. \tag{2.13}$$

In order to further exploit (2.13) as a rectangle approximation ($x[k]$ constant over a segment of T) of the original FT in (2.3), we further multiply it with sampling interval T. This gives the following definition of the discrete-time Fourier transform (DTFT)

$$\mathbb{X}_T(e^{i\omega T}) = T\sum_{k=-\infty}^{\infty} x[k]e^{-i\omega kT}\,. \tag{2.14}$$

Here, the subscript 'T' highlights the sampling interval. Comparing the definition of DTFT in (2.14) with the FT of the sampled continuous-time signal in (2.12) leads to

$$\mathbb{X}_T(e^{i\omega T}) = T\mathbb{X}_{\mathsf{s}}(i\omega) \tag{2.15}$$

Note that $\mathbb{X}_T(e^{i\omega T})$ is a continuous function over frequency domain (w.r.t. ω), though being derived from discrete-time signal $x[k]$, $k \in \mathbb{Z}$. Inversely, we define the inverse discrete-time Fourier transform (IDTFT) to recover signal value at any time stamp k, namely,

$$x[k] = \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} \mathbb{X}_T(e^{i\omega T})\,e^{i\omega kT}\,\mathrm{d}\omega = \frac{1}{2\pi}\int_{-\omega_{\mathsf{s}}/2}^{\omega_{\mathsf{s}}/2} \mathbb{X}_T(e^{i\omega T})\,e^{i\omega kT}\,\mathrm{d}\omega\,, \tag{2.16}$$

with $\omega_{\mathsf{s}} = 2\pi/T$ being the sampling (angular) frequency. We now demonstrate the validation of IDTFT to recover discrete-time signal value $x[k]$ in time domain given its representation in frequency domain.

Proof. We substitute the (2.14) into (2.16) and obtain

$$\begin{aligned}
\text{IDTFT}\{\mathbb{X}_T(e^{i\omega T})\} =& \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} \mathbb{X}_T(e^{i\omega T})\, e^{i\omega kT}\,\mathrm{d}\omega \\
&= \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} \left(T\sum_{m=-\infty}^{\infty} x[m]e^{-i\omega mT}\right) e^{i\omega kT}\,\mathrm{d}\omega \\
&= \frac{T}{2\pi}\sum_{m=-\infty}^{\infty} x[m]\int_{-\pi/T}^{\pi/T} e^{i\omega(k-m)T}\,\mathrm{d}\omega \\
&= \frac{T}{2\pi}\sum_{m=-\infty}^{\infty} x[m]\frac{-i}{(k-m)T}\,2i\sin(\pi(k-m)) \quad \text{(Euler's formula)} \\
&= \sum_{m=-\infty}^{\infty} x[m]\,\mathrm{sinc}(k-m) = x[k]\,,
\end{aligned} \tag{2.17}$$

with $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ that equals to zero when x is a non-zero integer and one when $x = 0$. ■

The multiplication and convolution properties are well-known for Fourier transform and its inverse. We hereby prove these two properties for DTFT and IDTFT, namely, multiplication/convolution in time domain corresponds to convolution/multiplication in frequency domain in the sense of DTFT.

Proof. Given two discrete-time signals $x[k]$ and $y[k]$ of the same sampling interval T, the DTFT of their product follows

$$\begin{aligned}
\text{DTFT}\{x[k]y[k]\} &= T\sum_{k=-\infty}^{\infty} x[k]y[k]e^{-i\omega kT} \\
&= T\sum_{k=-\infty}^{\infty}\left(\frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2} \mathbb{X}_T(e^{i\nu T})e^{i\nu kT}\,\mathrm{d}\nu\right)y[k]e^{-i\omega kT} \\
&= \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2} \mathbb{X}_T(e^{i\nu T})\left(T\sum_{k=-\infty}^{\infty} y[k]e^{-i(\omega-\nu)kT}\right)\mathrm{d}\nu \\
&= \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2} \mathbb{X}_T(e^{i\nu T})\mathbb{Y}_T(e^{i(\omega-\nu)T})\,\mathrm{d}\nu \\
&= \frac{1}{2\pi}\mathbb{X}_T(e^{i\omega T}) \star \mathbb{Y}_T(e^{i\omega T})\,.
\end{aligned} \tag{2.18}$$

Further, the DTFT of convolving the two discrete-time signals follows

$$\begin{aligned}
\text{DTFT}\{x[k]\star y[k]\} &= T\sum_{k=-\infty}^{\infty}\left(\sum_{m=-\infty}^{\infty} x[k-m]y[m]\right)e^{-i\omega kT} \\
&= T\sum_{m=-\infty}^{\infty} y[m]\left(\sum_{k=-\infty}^{\infty} x[k-m]e^{-i\omega kT}\right) \\
&= \sum_{m=-\infty}^{\infty} y[m]\left(T\sum_{k=-\infty}^{\infty} x[k-m]e^{-i\omega(k-m)T}\right)e^{-i\omega mT} \\
&= \frac{1}{T}\left(T\sum_{m=-\infty}^{\infty} y[m]e^{-i\omega mT}\right)\left(T\sum_{k=-\infty}^{\infty} x[k-m]e^{-i\omega(k-m)T}\right) \\
&\overset{n:=k-m}{=} \frac{1}{T}\left(T\sum_{n=-\infty}^{\infty} x[n]e^{-i\omega nT}\right)\left(T\sum_{m=-\infty}^{\infty} y[m]e^{-i\omega mT}\right) \\
&= \frac{1}{T}\mathbb{X}_T(e^{i\omega T})\mathbb{Y}_T(e^{i\omega T})
\end{aligned}$$

■

We further show the following examples to demonstrate DTFT to several important signals.

Example 2.2.1. The DTFT of Dirac delta function $x[k] = \delta[k]$ is

$$\mathbb{X}_T(e^{i\omega T}) = T \sum_{k=-\infty}^{\infty} \delta[k] e^{-i\omega kT} = T$$

Example 2.2.2. Find the IDTFT of a rectangular window function in frequency domain

$$\mathbb{X}_T(e^{i\omega T}) = \begin{cases} 1, & |\omega| < \omega_s/2 \\ 0, & |\omega| \geq \omega_s/2 \end{cases} . \tag{2.19}$$

We apply (2.16) to the low-pass filter above and obtain

$$\begin{aligned} x[k] &= \frac{1}{2\pi} \int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2} \mathbb{X}_T(e^{i\omega T}) e^{i\omega kT} \,\mathrm{d}\omega \\ &= \frac{1}{2\pi} \int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2} e^{i\omega kT} \,\mathrm{d}\omega \\ &= \frac{1}{2\pi} \frac{-i}{kT} \Big(e^{ik\pi} - e^{-ik\pi} \Big) \quad \text{(Euler's formula)} \\ &= \frac{1}{2\pi} \frac{-i}{kT} 2i \sin(k\pi) \\ &= \frac{\sin(k\pi)}{\pi kT} = \frac{1}{T} \mathrm{sinc}(k) \end{aligned}$$

Example 2.2.3. Find the IDTFT of a series of Dirac delta functions

$$\mathbb{X}_T(e^{i\omega T}) = \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_\mathrm{s}) \,, n \in \mathbb{Z}$$

in frequency domain, with ω_s being the sampling frequency in time domain. Applying (2.16) to the equation above leads to

$$x[k] = \frac{1}{2\pi} \int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2} \Big(\sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_\mathrm{s}) \Big) e^{i\omega kT} \,\mathrm{d}\omega = \frac{1}{2\pi} \int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2} \delta(\omega) e^{i\omega kT} \,\mathrm{d}\omega = \frac{1}{2\pi} \tag{2.20}$$

Therefore, we can express Dirac delta function in frequency domain as the DTFT of a constant number, namely,

$$\delta(\omega) = \frac{T}{2\pi} \sum_{k=-\infty}^{\infty} e^{-i\omega kT} \,, \quad \text{with} \quad \omega \in [-\omega_\mathrm{s}, \omega_\mathrm{s}] \,, \tag{2.21}$$

which is the counterpart of the definition in (2.8) for discrete-time case.

Example 2.2.4. Find the DTFT of cosine function series $x[k] = \cos(k)$, with sampling interval $T = 1$.

$$\begin{aligned} \mathbb{X}(e^{i\omega}) &= \sum_{k=-\infty}^{\infty} \cos(k) e^{-i\omega k} \\ &= \frac{1}{2} \sum_{k=-\infty}^{\infty} (e^{ik} + e^{-ik}) e^{-i\omega k} \\ &= \frac{1}{2} \sum_{k=-\infty}^{\infty} \Big(e^{-i(\omega-1)k} + e^{-i(\omega+1)k} \Big) \,. \end{aligned} \tag{2.22}$$

Given the optional definition of Dirac delta function in (2.21), we further obtain

$$\mathbb{X}_1(e^{i\omega}) = \pi\big(\delta(\omega - 1) + \delta(\omega + 1)\big) \tag{2.23}$$

Example 2.2.5. Find the DTFT of sine function series $x[k] = \sin(k)$, with sampling interval $T = 1$.

$$\begin{aligned}\mathbb{X}(e^{i\omega}) &= \sum_{k=-\infty}^{\infty} \sin(k) e^{-i\omega k} \\ &= \frac{1}{2i} \sum_{k=-\infty}^{\infty} (e^{ik} - e^{-ik}) e^{-i\omega k} \\ &= \frac{1}{2i} \sum_{k=-\infty}^{\infty} \left(e^{-i(\omega-1)k} - e^{-i(\omega+1)k}\right).\end{aligned} \tag{2.24}$$

Given the optional definition of Dirac delta function in (2.21), we further obtain

$$\mathbb{X}(e^{i\omega}) = -i\pi\big(\delta(\omega - 1) - \delta(\omega + 1)\big)$$

2.2.2 Sampling Theorem

Given a discrete-time signal in (2.9), we can expand the infinite sum of Dirac delta function values

$$f(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \tag{2.25}$$

via Fourier series in (2.1), as it is periodic of period T. Corresponding Fourier coefficient then follows

$$\mathbb{F}_{\mathsf{FC}}[n] = \frac{1}{T} \int_0^T \sum_{k=-\infty}^{\infty} \delta(t - kT) e^{-in\omega_s t} \, \mathrm{d}t = \frac{1}{T}. \tag{2.26}$$

Here, the fundamental frequency is the sampling frequency $\omega_s = 2\pi/T$. Therefore, we can express the discrete-time series in (2.9) as

$$x_{\mathsf{s}}(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t - kT) = x(t) \sum_{n=-\infty}^{\infty} \mathbb{F}_{\mathsf{FC}}[n] e^{in\omega_s t} = \frac{1}{T} \sum_{n=-\infty}^{\infty} x(t) e^{in\omega_s t}. \tag{2.27}$$

We again apply FT to the expression above and obtain its representation in frequency domain

$$\begin{aligned}\mathbb{X}_{\mathsf{s}}(i\omega) &= \int_{-\infty}^{\infty} x_{\mathsf{s}}(t) \, e^{-i\omega t} \, \mathrm{d}t \\ &= \frac{1}{T} \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(t) e^{in\omega_{\mathsf{s}} t} e^{-i\omega t} \, \mathrm{d}t \\ &= \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) e^{-i(\omega - n\omega_{\mathsf{s}})t} \, \mathrm{d}t \\ &= \frac{1}{T} \sum_{n=-\infty}^{\infty} \mathbb{X}\big(i(\omega - n\omega_{\mathsf{s}})\big).\end{aligned} \tag{2.28}$$

Note that $\mathbb{X}_T(e^{i\omega T}) = T\mathbb{X}_{\mathsf{s}}(i\omega)$ as discussed around (2.15), the derivation above establishes the relationship between DTFT and FT, which is also known as the Poisson summation formula below

$$\mathbb{X}_T(e^{i\omega T}) = \sum_{n=-\infty}^{\infty} \mathbb{X}\big(i(\omega - n\omega_{\mathsf{s}})\big). \tag{2.29}$$

Coming back to (2.28), the formula relates the Fourier transform of a sampled signal to that (FT) of the original signal. And $\mathbb{X}_T(e^{i\omega T})$ is obtained by replicating $\mathbb{X}(i\omega)$, FT of the underlying continuous-time signal, infinite number of times over the frequency domain with a period of sampling (angular) frequency ω_{s}.

What happens when sampling in time domain? Sampling in time domain replicates the spectrum in frequency domain. This replication process is the theoretical foundation of sampling theorem. A follow-up question would be: How to make sure that the sampled values $x[k]$ uniquely represent/reconstruct the underlying continuous-time signal $x(t)$? The answer is YES, when:

1. $x(t)$ is band-limited (e.g., invalid for Dirac delta function) such that frequency model $\mathbb{X}(i\omega) = 0$ for $|\omega| \geq \omega_b$, with ω_b being the so-called bandwidth.

2. sampling interval T is sufficiently small or sampling frequency ω_{s} is sufficiently high.

Sampling Theorem Given a band-limited continuous-time signal $x(t)$ of bandwidth ω_{b} ($\mathbb{X}(i\omega) = 0$ for $|\omega| \geq \omega_{\mathsf{b}}$), then $x(t)$ is uniquely represented by (discrete-time) samples $x[k] = x(kT)$ with $k \in \mathbb{Z}$, if

$$\omega_{\mathsf{s}} > 2\,\omega_{\mathsf{b}} \quad \text{or} \quad \omega_{\mathsf{N}} > \omega_{\mathsf{b}}\,, \tag{2.30}$$

with $\omega_{\mathsf{N}} = \omega_{\mathsf{s}}/2$ denoting the Nyquist frequency (a fundamental characteristic of a sampler). When the signal's highest frequency (bandwidth ω_{b}) exceeds the Nyquist frequency of the sampler, the resulted discrete samples suffer from aliasing, namely, neighboring copies of $\mathbb{X}(i\omega)/T$ overlap with each other as shown in (2.28), and reconstructing the original signal is no longer possible.

2.3 Signal Energy

2.3.1 Continuous-Time Signal

Given a signal $x(t)$ of limited energy in the sense of $\mathcal{L}$-2 norm, its energy can be measured via FT in frequency domain according to Parseval's formula as follows

$$\mathscr{E}(x) = \|x(t)\|^2 = \int_{-\infty}^{\infty} |x(t)|^2 \,\mathrm{d}t = \frac{1}{2\pi}\int_{-\infty}^{\infty} |\mathbb{X}(i\omega)|^2 \,\mathrm{d}\omega\,, \tag{2.31}$$

where $\|\cdot\|$ denote the $\mathcal{L}$-2 norm and $|\cdot|$ the complex modulus. Based on the signal energy defined above, the energy spectral density (or energy spectrum) of a $\mathcal{L}$-2 signal ($\mathscr{E}(x) < \infty$) is expressed as

$$\Phi_{xx}(\omega) = |\mathbb{X}(i\omega)|^2\,, \tag{2.32}$$

which is the squared absolute value of its Fourier transform $\mathbb{X}(i\omega)$. It describes how the energy in the signal is distributed over frequency domain, and we have signal energy expressed as its integral

$$\mathscr{E}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi_{xx}(\omega)\,\mathrm{d}\omega \tag{2.33}$$

over frequency domain. We now provide the following proof for (2.31).

Proof. Given the FT $\mathbb{X}(i\omega)$ of $x(t)$ in (2.3), its complex conjugate

$$\mathbb{X}^*(i\omega) = \left(\int_{-\infty}^{\infty} x(t)e^{-i\omega t}\,\mathrm{d}t\right)^* = \int_{-\infty}^{\infty} x^*(t)e^{i\omega t}\,\mathrm{d}t\,.$$

By replacing ω with $-\omega$, we then obtain $\mathbb{X}^*(-i\omega)$ being the FT of $x^*(t)$. Thus, the complex conjugate of $x(t)$ can be represented via IFT in (2.4) as

$$x^*(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbb{X}^*(-i\omega)e^{i\omega t}\,\mathrm{d}\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbb{X}^*(i\omega)e^{-i\omega t}\,\mathrm{d}\omega\,.$$

We then derive energy of $x(t)$ in the form of $\mathcal{L}$-2 norm as

$$\begin{aligned}\mathscr{E}(x) &= \int_{-\infty}^{\infty} |x(t)|^2 \,\mathrm{d}t = \int_{-\infty}^{\infty} x(t)x^*(t)\,\mathrm{d}t \\ &= \int_{-\infty}^{\infty} x(t)\left(\frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbb{X}^*(i\omega)e^{-i\omega t}\,\mathrm{d}\omega\right)\mathrm{d}t \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbb{X}^*(i\omega)\left(\int_{-\infty}^{\infty} x(t)e^{-i\omega t}\,\mathrm{d}t\right)\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbb{X}^*(i\omega)\mathbb{X}(i\omega)\,\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} |\mathbb{X}(i\omega)|^2\,\mathrm{d}\omega\,. \end{aligned} \tag{2.34}$$

■

2.3.2 Discrete-Time Signal

For a sequence of samples $x[k]$ sampled from a $\mathcal{L}$-2 continuous-time signal with sampling interval T, the energy in the discrete-time signal is calculated according to

$$\mathscr{E}(x) = T\big\|x[k]\big\|^2 = T\sum_{k=\infty}^{\infty}\big|x[k]\big|^2 = \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\Big|\mathbb{X}_T(e^{i\omega T})\Big|^2\,\mathrm{d}\omega\,. \tag{2.35}$$

The energy term defined for samples above can be seen as a discrete approximation of the underlying continuous-time signal (therefore scaled by T). Similarly to (2.32), we define the following energy spectral density (or energy spectrum) for discrete-time $\mathcal{L}$-2 signal

$$\Phi_{xx}(\omega) = \big|\mathbb{X}_T(e^{i\omega T})\big|^2\,, \tag{2.36}$$

such that

$$\mathscr{E}(x) = \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\Phi_{xx}(\omega)\,\mathrm{d}\omega\,. \tag{2.37}$$

We now prove the discrete-time version of Parseval's formula in (2.35) as below.

Proof. According to (2.14), the complex conjugate of the DTFT for $x[k]$ follows

$$\mathbb{X}_T^*(e^{i\omega T}) = \left(T\sum_{k=-\infty}^{\infty} x[k]e^{-i\omega kT}\right)^* = T\sum_{k=-\infty}^{\infty} x^*[k]e^{i\omega kT}\,.$$

Thus, we have $\mathbb{X}_T^*(e^{-i\omega T}) = T\sum_{k=-\infty}^{\infty} x^*[k]e^{-i\omega kT}$, and the complex conjugate of $x(t)$ can be represented as

$$x^*[k] = \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\mathbb{X}_T^*(e^{-i\omega T})e^{i\omega kT}\,\mathrm{d}\omega = \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\mathbb{X}_T^*(e^{i\omega T})e^{-i\omega kT}\,\mathrm{d}\omega\,,$$

leading to

$$\begin{aligned}\mathscr{E}(x) &= T\sum_{k=\infty}^{\infty}\left|x[k]\right|^2 = T\sum_{k=\infty}^{\infty}x[k]\,x^*[k]\\ &= T\sum_{k=\infty}^{\infty}x[k]\left(\frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\mathbb{X}_T^*(e^{i\omega T})e^{-i\omega kT}\,\mathrm{d}\omega\right)\\ &= \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\left(T\sum_{k=\infty}^{\infty}x[k]e^{-i\omega kT}\right)\mathbb{X}_T^*(e^{i\omega T})\,\mathrm{d}\omega\\ &= \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\mathbb{X}_T(e^{i\omega T})\mathbb{X}_T^*(e^{i\omega T})\,\mathrm{d}\omega\\ &= \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\left|\mathbb{X}_T(e^{i\omega T})\right|^2\mathrm{d}\omega\,.\end{aligned}\tag{2.38}$$

■

2.4 Computer-Aided Frequency Modeling

Signals processing in real-world applications have limitations compared with ideally-defined mathematical models – discrete-time signals usually have limited time span, and frequency domain for spectral analysis is also discrete. The former is often tackled using truncated DTFT, and the latter with discrete Fourier transform. We introduce the two variants in the remainder of this section.

2.4.1 Truncated discrete-time Fourier transform (Truncated DTFT)

When sampling a signal in practice, we often obtain a discrete-time series $x[k]$ of only finite length, say $k = 0, 1, ..., N-1$. Based thereon, we look for an approximation of the DTFT of the ideal discrete-time signal with $k \in \mathbb{Z}$. One popular approach is to truncate the ideal signal using a window in time domain. Mathematically speaking, an infinite series of samples can be truncated into a series of the first N samples via

$$x_N[k] = x[k]w_N[k]\,, \quad \text{with} \quad w_N[k] = \begin{cases}1\,, & 0 \le k \le N-1\\ 0\,, & \text{otherwise}\end{cases}\tag{2.39}$$

being a rectangular window function. Thus, the DTFT of the truncated discrete-time signal follows

$$\mathbb{X}_T^{(N)}(e^{i\omega T}) = T\sum_{k=-\infty}^{\infty}x_N[k]e^{-i\omega kT} = T\sum_{k=0}^{N-1}x[k]e^{-i\omega kT}\,.\tag{2.40}$$

How does $\mathbb{X}_T^{(N)}(e^{i\omega T})$ relates to $\mathbb{X}_T(e^{i\omega T})$ satisfying the sampling theorem? According to the multiplication property of DTFT in (2.18), the windowed signal has the following DTFT in the form of convolution of the two spectrum components

$$\mathbb{X}_T^{(N)}(e^{i\omega T}) = \frac{1}{2\pi}\int_{-\omega_\mathrm{s}/2}^{\omega_\mathrm{s}/2}\mathbb{X}_T(e^{i\nu T})\mathbb{W}_N(e^{i(\omega-\nu)T})\,\mathrm{d}\nu\tag{2.41}$$

When $\mathbb{W}$ is a Dirac delta function (note that DTFT is still continuous in frequency domain even for DT signals), namely, $\mathbb{W}_N = \delta(\omega-\nu)$, we have

$$\mathbb{X}_T^{(N)}(e^{i\omega T}) = \frac{1}{2\pi}\mathbb{X}_T(e^{i\omega T})\,,$$

meaning truncation in time domain leads to no information loss in frequency domain via DTFT. This is, however, never the case in real world, and some artifact always exists with non-Dirac windowing. For investigation, we first apply DTFT to the rectangular window function in (2.39) and obtain

$$\begin{aligned}\mathbb{W}_N(e^{i\omega T}) &= T\sum_{k=-\infty}^{\infty} w_N[k]e^{-i\omega kT}\\ &= T\sum_{k=0}^{N-1} e^{-i\omega kT}\\ &= T\frac{\sin(-\omega NT/2)}{\sin(-\omega T/2)}e^{-i\omega(N-1)T/2} \quad \text{(see (2.50))}\\ &= Te^{-i\omega(N-1)T/2}\frac{\sin(N\omega T/2)}{\sin(\omega T/2)}\,.\end{aligned} \tag{2.42}$$

Its complex modulus then follows

$$|\mathbb{W}_N(e^{i\omega T})| = T\left|\frac{\sin(N\omega T/2)}{\sin(\omega T/2)}\right|,$$

with zeros at integer multiple of $\frac{2\pi}{NT}$. Essentially, multiplying the discrete signal with a window function refers to convolving its frequency model with the one derived for the window function shown in (2.42). The ability of frequency separation of two nearby sinusoids is then limited to about $\frac{4\pi}{NT}$. The following example further illustrates the artifact of truncating a discrete-time signal in frequency domain.

Example 2.4.1. Derive DTFT to $x_N[k] = \cos(\omega_1 k)$ with sampling interval $T = 1$ and truncation over $k = 0, ..., N-1$. Applying DTFT to the truncated discrete-time signal leads to

$$\begin{aligned}\mathbb{X}^{(N)}(e^{i\omega}) &= \sum_{k=0}^{N-1}\cos(\omega_1 k)e^{-i\omega k}\\ &= \frac{1}{2}\sum_{k=0}^{N-1}(e^{i\omega_1 k} + e^{-i\omega_1 k})e^{-i\omega k} \quad \text{(Euler's formula)}\\ &= \frac{1}{2}\sum_{k=0}^{N-1}\left(e^{-i(\omega-\omega_1)k} + e^{-i(\omega+\omega_1)k}\right).\end{aligned}$$

Again we utilize exponential sum formula above and obtain

$$\mathbb{X}^{(N)}(e^{i\omega}) = \frac{1}{2}\frac{\sin\left(N(\omega-\omega_1)/2\right)}{\sin\left((\omega-\omega_1)/2\right)}e^{-i(\omega-\omega_1)(N-1)/2} + \frac{1}{2}\frac{\sin\left(N(\omega_1+\omega)/2\right)}{\sin\left((\omega_1+\omega)/2\right)}e^{-i(\omega_1+\omega)(N-1)/2}\,.$$

Note that the ideal case ($N \in \mathbb{Z}$) of its frequency model is given by (2.23) in the form of two Dirac delta functions.

Leakage Comparison between Example 2.2.4 and Example 2.4.1 shows that frequency content of the infinite signal, namely, Dirac deltas at two frequencies, has leaked to adjacent frequencies due to the truncation in time domain. Essentially, this is induced by the rectangular window spectrum shifted by convolution as shown in (2.41).

2.4.2 Discrete Fourier transform (DFT)

In order to enable computer-aided spectral analysis, the frequency domain is also to be discretized. Given a discrete-time signal $x[k] = x(kT)$ of sampling interval T truncated over $k = 0, ..., N-1$, we discretize the frequency domain $[0, \omega_s]$[2] into N segments indexed by $\omega_n = n\omega_s/N$ with $n = 0, ..., N-1$. Applying

[2] Sampling in time domain replicates spectrum with period of ω_s

truncated DTFT as in (2.40) then leads to

$$\mathbb{X}_T[n] = T\sum_{k=0}^{N-1} x[k]e^{-ik\omega_n T} = T\sum_{k=0}^{N-1} x[k]e^{-ikn\omega_\mathrm{s}T/N} = T\sum_{k=0}^{N-1} x[k]e^{-i2\pi kn/N}\,. \tag{2.43}$$

By further normalizing the sampling interval T, we obtain the discrete Fourier-transform (DFT) as follows

$$\mathbb{X}[n] = \sum_{k=0}^{N-1} x[k]e^{-i2\pi kn/N}\,. \tag{2.44}$$

Correspondingly, we have inverse discrete Fourier transform (IDFT) given by

$$x[k] = \frac{1}{N}\sum_{n=0}^{N-1} \mathbb{X}[n]e^{i2\pi nk/N} \tag{2.45}$$

Moreover, $\omega_s/N = \frac{2\pi}{TN}$ indicates the frequency grid resolution. Raising the frequency grid resolution can be done by taking more samples in time domain (increase N).

Matrix representation We can also express the DFT/IDFT as a transformation matrix which can be applied to a discrete signals in time/frequency domain through matrix multiplication. This refers to the basic implementation of DFT/IDFT (especially using Matlab) and its faster variants, e.g. fast Fourier transform (FFT). We reformulate the definition in (2.44) and (2.45) as

$$\mathbb{X}[n] = \sum_{k=0}^{N-1} x[k](w_N)^{kn} \quad \text{and}$$
$$x[k] = \frac{1}{N}\sum_{n=0}^{N-1} \mathbb{X}[n](w_N^*)^{nk}\,,$$

respectively, with

$$w_N = e^{-i2\pi/N} \tag{2.46}$$

being a primitive N-th root of unity. We can then construct the so-called DFT matrix $\mathbf{W} \in \mathbb{C}^{N\times N}$ by filling in elements as $(w_N)^{kn}$, where k and n are also the column and row indices, respectively. Consequently, we can express for DFT and IDFT as

$$\underline{\mathbb{X}} = \mathbf{W}\underline{x} \quad \text{and} \quad \underline{x} = \frac{1}{N}\mathbf{W}^{\mathrm{H}}\underline{\mathbb{X}}\,, \tag{2.47}$$

respectively, with discrete signals concatenated into vector forms, i.e.,

$$\underline{x} = \big[\,x[0], \ldots, x[N-1]\,\big]^\top \quad \text{and} \quad \underline{\mathbb{X}} = \big[\,\mathbb{X}[0], \ldots, \mathbb{X}[N-1]\,\big]^\top\,.$$

Moreover, $\mathbf{W}^{\mathrm{H}}$ denotes the conjugate transpose of $\mathbf{W}$. Note that matrix $\mathbf{W}$ is also symmetric, and we can express it explicitly as follows

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & w_N & w_N^2 & \ldots & w_N^{N-1} \\ 1 & w_N^2 & w_N^4 & \ldots & w_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{N-1} & w_N^{2(N-1)} & \ldots & w_N^{(N-1)^2} \end{bmatrix}\,.$$

As shown in (2.47), DTF of a real-valued discrete-time signals of length N requires $2N^2$ multiplications and $2N(N-1)$ additions. Therefore, overall we need $4N^2 - 2N$ computations. As an example of DFT and IDFT, we demonstrate that the transform pair, DFT and IDFT, hold in the following proof.

Proof. Taking the DFT expression in (2.45) into (2.44) leads to

$$\begin{aligned}
\text{IDFT}\{\mathbb{X}[n]\} &= \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{X}[n]e^{i2\pi nk/N} \\
&= \frac{1}{N}\sum_{n=0}^{N-1}\left(\sum_{l=0}^{N-1}x[l]e^{-i2\pi ln/N}\right)e^{i2\pi nk/N} \\
&= \frac{1}{N}\sum_{l=0}^{N-1}x[l]\sum_{n=0}^{N-1}e^{i2\pi(k-l)n/N} \\
&= \frac{1}{N}\sum_{l=0}^{N-1}x[l]\frac{\sin\big(\pi(l-k)\big)}{\sin\big(\pi(l-k)/N\big)}e^{i\pi(l-k)(N-1)/N} \\
&= \frac{1}{N}\sum_{l=0}^{N-1}x[l]N\delta[k-l] \\
&= x[k]\,,
\end{aligned}$$

with $\delta[k-l]$ being the Kronecker delta function defined as

$$\delta[k-l] = \begin{cases} 1, & k=l \\ 0, & \text{otherwise} \end{cases}. \tag{2.48}$$

■

2.4.3 Fast Fourier Transform (FFT)

As discussed above, the standard implementation of DFT has time complexity of $\mathcal{O}(N^2)$, which is not appealing for real-world applications, especially for large data sets. Note that the DFT matrix $\mathbf{W}$ in (2.47) is symmetric, and there exists the periodicity in the entries

$$(w_N^{nk})^2 = e^{-i4\pi/N} = e^{-i2\pi/(N/2)} = w_{N/2}^{nk}\,. \tag{2.49}$$

Therefore, we can compute design a faster algorithm to compute DFT according to divide and conquer, which is one of the basic ideas of fast Fourier transform (FFT). Note that FFT is not another kind of Fourier transform, it is an algorithm to calculate DFT efficiently. Compared with the standard definition of DFT, its time complexity decreased to $\mathcal{O}(N\log(N))$.

2.4.4 Circular Convolution

Note that the multiplication and convolution properties for FT and DTFT does not hold for DFT any more. This is problematic when it comes to designing filters. Often we process signals by handling them in frequency domain with filters (multiplication) and then transform them back to time domain. To solve this issue, we introduce the circular convolution with zero padding.

Suppose $\mathbb{X}[n]$ and $\mathbb{Y}[n]$ are DFTs of truncated discrete-time signals $x[k]$ and $y[k]$ with $k=0,...,N-1$. Given their product $\mathbb{X}[n]\mathbb{Y}[n]$ in frequency domain, we now transform it back to discrete-time domain via

IDFT as follows

$$\begin{aligned}
\text{IDFT}\{\mathbb{X}[n]\mathbb{Y}[n]\} &= \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{X}[n]\mathbb{Y}[n]e^{i2\pi nk/N} \\
&= \frac{1}{N}\sum_{n=0}^{N-1}\left(\sum_{l=0}^{N-1}x[l]e^{-i2\pi ln/N}\right)\left(\sum_{m=0}^{N-1}y[m]e^{-i2\pi mn/N}\right)e^{i2\pi nk/N} \\
&= \frac{1}{N}\sum_{l=0}^{N-1}\sum_{m=0}^{N-1}x[l]y[m]\left(\sum_{n=0}^{N-1}e^{i2\pi(k-l-m)n/N}\right) \\
&= \frac{1}{N}\sum_{l=0}^{N-1}\sum_{m=0}^{N-1}x[l]y[m]\frac{\sin\big(\pi(k-l-m)\big)}{\sin\big(\pi(k-l-m)/N\big)}e^{i\pi(k-l-m)(N-1)/N} \quad \text{(see (2.50))}
\end{aligned}$$

Essentially, function $\sin(\pi\tau)/\sin(\pi\tau/N)$ with $\tau \in \mathbb{Z}$ equals to zero except for the cases when τ is an integer multiple of N. Thus, we further have

$$\begin{aligned}
\text{IDFT}\{\mathbb{X}[n]\mathbb{Y}[n]\} &= \sum_{l=0}^{N-1}x[l]\sum_{m=0}^{N-1}y[m]\delta[(k-l-m)_{\mathrm{mod}\,N}] \quad \text{(Euler's formula)} \\
&= \sum_{l=0}^{N-1}x[l]y[(k-l)_{\mathrm{mod}\,N}] \\
&\coloneqq x\otimes y[k]\,,
\end{aligned}$$

with $\otimes$ denoting the circular convolution. The modulo function is a directional function that describes how much we have to move further after the division in forward direction to catch the numerator(e.g., $-7\%3 = 2$). Therefore, product of two truncated discrete-time signals can be calculated as the circular convolution in frequency domain under DFT. Note that $k-l \in [-N+1, N-1] \subset \mathbb{Z}$, meaning that the modulo operator only take action when $k-l \in [-N+1,-1]$. With zero padding in discrete-time domain, circular convolution coincides with linear convolution in discrete-frequency domain, namely

$$x \star y[k] = \sum_{l=0}^{N-1}x_{\mathsf{p}}[l]y_{\mathsf{p}}[k-l] = \sum_{l=0}^{2N-1}x_{\mathsf{p}}[l]y_{\mathsf{p}}[(k-l)_{\mathrm{mod}\,N}] = x_{\mathsf{p}}\otimes y_{\mathsf{p}}[k]\,,$$

which corresponds to $\mathbb{X}_{\mathsf{p}}\mathbb{Y}_{\mathsf{p}}$ in frequency domain. Therefore, zero-padding keeps the (linear) convolution property under DFT.

Appendix

Exponential sum formulas When analyzing truncated signals in frequency domain, e.g., it is quite often to compute infinite sum of exponential functions. According to the closed-form formula for summing a finite geometric series

$$\sum_{n=0}^{N-1}ar^n = \sum_{n=1}^{N}ar^{n-1} = \begin{cases} a(1-r^N)/(1-r)\,, & r \neq 1 \\ aN\,, & r = 1\end{cases}$$

we have

$$\sum_{n=0}^{N-1}e^{inx} = \frac{\sin(Nx/2)}{\sin(x/2)}e^{ix(N-1)/2}\,. \tag{2.50}$$

Chapter 3

Stochastic Signals and Spectral Models

3.1 Random variable

A random variable, or stochastic variable, is a mathematical formalization of a quantity which depends on random events. Given a random variable X, its uncertainty is characterized by a probability density function (PDF) $p_X(x)$, which integrates to one in the sample space Ω_X (the set of possible values taken by the random variable), namely,

$$\int_{\Omega_X} p_X(x) = 1\,.$$

Statistically, we can now define the expectation and variance of a scalar random variable X as

$$\mathrm{E}\{X\} = \mu = \int_{\Omega_X} x\, p_X(x)\,\mathrm{d}x$$

and

$$\mathrm{var}\{X\} = \sigma^2 = \int_{\Omega_X} \big(x - \mathrm{E}\{X\}\big)^2\, p_X(x)\,\mathrm{d}x = \mathrm{E}\{X^2\} - (\mathrm{E}\{X\})^2\,,$$

respectively. The latter also gives the standard deviation $\sigma = \sqrt{\mathrm{var}\{X\}}$. The definitions above can also be extended to a multi-variate random variable with

$$\mathrm{E}\{X\} = \underline{\mu} = \int_{\Omega_X} \underline{x}\, p_X(\underline{x})\,\mathrm{d}\underline{x}$$

and

$$\mathrm{cov}\{X\} = \mathbf{\Sigma} = \int_{\Omega_X} \big(\underline{x} - \mathrm{E}\{X\}\big)\big(\underline{x} - \mathrm{E}\{X\}\big)^\top p_X(\underline{x})\,\mathrm{d}\underline{x} = \mathrm{E}\{XX^\top\} - \mathrm{E}\{X\}(\mathrm{E}\{X\})^\top\,, \tag{3.1}$$

being the mean and covariance matrix.

Gaussian distribution One of the most important distributions is the Gaussian distribution (or normal distribution) defined in Euclidean spaces. For a scalar Gaussian-distributed random variable, its PDF is given in the following closed form

$$f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right), \quad x \in \mathbb{R}\,.$$

More generally, a d-dimensional multivariate random Gaussian distribution takes the following PDF

$$f(\underline{x}|\underline{\mu},\mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d \det \mathbf{\Sigma}}} \exp\Big(-\frac{1}{2}\big(\underline{x}-\underline{\mu}\big)^\top \mathbf{\Sigma}^{-1}\big(\underline{x}-\underline{\mu}\big)\Big)\,, \quad \underline{x} \in \mathbb{R}^d$$

with $\underline{\mu} \in \mathbb{R}^d$ and $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ denoting the mean and covariance function, respectively.

3.2 Stochastic Process

In comparison with deterministic model, a stochastic process is a sequence of random variable indexed by a certain mathematical set (e.g., a set of time stamps). In this chapter, we mainly focus on real-valued stochastic processes with discrete time indices, and we denote such a stochastic process as $X[k]$ with k indicating the time stamp. An observed time series from it is called a realization, and the same stochastic process can give rise to multiple realizations. Therefore, the random variable at each time stamp has different PDFs.

3.2.1 Expectation and covariance functions

Similar to the case of random variables introduced in Sec. 3.1, we can define the so-called expectation function and covariance function (since PDF of X changes over time) to characterize a stochastic process. More specifically, the expectation function (ensemble mean) is given by

$$\underline{m}_X[k] = \mathrm{E}\{\underline{x}[k]\}\,, \tag{3.2}$$

and the (auto)covariance function follows

$$\mathbf{K}_{XX}[k,l] = \mathrm{E}\Big\{\big(\underline{x}[k] - \underline{m}_X[k]\big)\big(\underline{x}[l] - \underline{m}_X[l]\big)^\top\Big\} = \mathrm{E}\Big\{x[k](x[l])^\top\Big\} - \underline{m}_X[k]\underline{m}_X^\top[l]\,. \tag{3.3}$$

Note that two parameters above are both functions w.r.t. given the time stamps as input. If the stochastic process is zero-mean (meaning for all time stamps), we then have $\mathbf{K}_{XX}[k,l] = \mathrm{E}\{x[k]x^\top[l]\}$. Given two stochastic processes $X[k]$ and $Y[k]$ obtained on the same time scale, we can also define the cross covariance function

$$\mathbf{K}_{XY}[k,l] = \mathrm{E}\Big\{(\underline{x}[k] - \underline{m}_X[k])(\underline{y}[l] - \underline{m}_Y[l])^\top\Big\} \tag{3.4}$$

to measure the similarity between the two stochastic processes. By definition, we have $\mathbf{K}_{XY}[k,l] = \mathbf{K}_{YX}[l,k]$ for cross-covariance function.

Autocorrelation function One important measure to characterize a stochastic process is the autocorrelation function. Given a real-valued stochastic process $X[k]$, its autocorrelation between time stamps k and l is given in the following form [1]

$$\mathbf{R}_{XX}[k,l] = \mathrm{E}\big\{\underline{x}[k]\underline{x}^\top[l]\big\} = \mathbf{K}_{XX}[k,l] + \underline{m}_X[k]\,\underline{m}_X^\top[l]\,. \tag{3.5}$$

Subtracting the mean before multiplication then yields the autocovariance function in (3.3), and we have $\mathbf{R}_{XX}[k,l] = \mathbf{K}_{XX}[k,l]$ when $X[k]$ is zero-mean.

3.2.2 Wide-sense stationary stochastic process

A stochastic process which statistically (PDF) does not change over time (time-invariant) is called stationary stochastic process. In generalm it is hard to measure and identify if the underlying stochastic properties change with time. However, we can introduce the so-called weakly/wide-sense stationary stochastic process w.r.t. a few key parameters that are measurable for practice use. Given a stochastic process $X[k]$, if its mean $\underline{m}_X[k]$ and autocovariance $\mathbf{K}[k,l]$ (or autocorrelation $\mathbf{R}[k,l]$) functions are time-independent (constant regardless of time indices), and its second moment is finite. These three conditions can be summarized as follow

- $\underline{m}_X[k] = \underline{m}_X[k+\tau], \forall\, \tau \in \mathbb{Z}\,;$
- $\mathbf{K}_{XX}[k,l] = \mathbf{K}_{XX}[k-l, 0], \forall\, k,l \in \mathbb{Z}\,;$
- $\mathrm{E}\big\{\underline{x}[k]\underline{x}^\top[k]\big\} < \infty, \forall\, k \in \mathbb{Z}\,;$

[1] The general definition of autocorrelation function involve complex conjugate (for complex signals).

In this case, covariance or autocorrelation function depend only on the time-distance between the pair of values but not on their position in time. In other words, autocorrelation function defined in (3.5) can be expressed as a function w.r.t. a time-lag τ

$$\mathbf{R}_{XX}[k,l] = \mathbf{R}_{XX}[k-l,0] =: \mathbf{R}_{XX}[\tau]\,, \quad \text{with} \quad k = l + \tau \tag{3.6}$$

which is the more common form for defining autocorrelation. In other words, writing an autocorrelation function as in (3.6) already assumes that the stochastic process is (at least) wide-sense stationary. In this case, the autocorrelation function is symmetric by definition, namely, $\mathbf{R}_{XX}[-k] = \mathbf{R}_{XX}[k]$, $\forall\, k \in \mathbb{Z}$. In many real-world applications, one common step for analyzing a stochastic process is to confirm whether it is realistic to assume it to be is weakly stationary.

3.2.3 White noise

As an important example of a stationary stochastic process, white noise is a sequence of independent and identically distributed (i.i.d.) random variables. Note that white noise could have any type of underlying distribution - the PDF is not necessarily Gaussian. The whiteness describes how the stochastic variables are correlated or dependent on each other over time. Whereas the probability distribution describes the uncertainty at a given time instant. For a scalar-valued white noise $X[k]$, we have its covariance function

$$\mathbf{R}_{XX}[k] = \sigma^2 \delta[k]\,, \tag{3.7}$$

with $\delta[k]$ being the Kronecker delta function in (2.48) and σ the standard deviation of the white noise. Note that white noise has constant power spectral density.

Wold's decomposition theorem White noise can be used as the basic constructor of all other stationary processes. Any stationary stochastic process $Y[k]$ can be realized via convolution

$$y[k] = \sum_{l=-\infty}^{\infty} x[l]h[k-l]\,, \tag{3.8}$$

where $x[k]$ denotes the white noise and $h[k]$ a certain **deterministic** sequence (filter). In this regard, the white noise is considered to “inject energy” to the signal, and the deterministic sequence here shapes the signal.

3.3 Frequency Modeling and Spectrum

We now aim to model a stochastic process statistically in frequency domain given its realizations, since they are just sequences of numbers as deterministic signals handled in the last chapter. However, Fourier transform of a stochastic process has no physical meaning statistically, because computing expectation is a linear operation. More specifically, we apply DTFT to a stochastic process of zero mean and obtain

$$\mathrm{E}\Big\{\mathbb{X}_T(e^{i\omega T})\Big\} = \mathrm{E}\Bigg\{T\sum_{k=-\infty}^{\infty} x[k]e^{-i\omega kT}\Bigg\} = T\sum_{k=-\infty}^{\infty} \mathrm{E}\big\{x[k]\big\}e^{-i\omega kT} = 0\,,$$

which provides no useful information about the original signal in time domain. In frequency domain, the transform of the first-order moment will only output a sequence of random numbers. However, transforming the second-order moment, namely, the autocorrelation, produces important observable information in frequency domain.

3.3.1 Spectrum

The spectrum, or spectral density, of a zero-mean weakly stationary stochastic process $X[k]$ is uniquely defined by its second-order moment, namely, its autocorrelation function, as follows

$$\Phi_{XX}(\omega) = \mathrm{DTFT}\{\mathbf{R}_{XX}[k]\} = T \sum_{k=-\infty}^{\infty} \mathbf{R}_{XX}[k]\, e^{-i\omega kT} \,. \tag{3.9}$$

Here, the definition can also be expressed by means of covariance function since we can always extract the mean from the data as preprocessing. Note that we have introduced the energy spectral density in (2.36) with notation $\Phi_{xx}(\omega)$, which refers to no statistical meaning as it handles deterministic signals. For a signal $x[k]$ of sampling interval $T = 1$ (for brevity), we have the following relationship between its spectrum and frequency model

$$\Phi_{XX}(\omega) = \mathrm{E}\big\{|\mathbb{X}(e^{i\omega})|^2\big\} \,. \tag{3.10}$$

For white noise of variance σ^2 shown in (3.7), its spectrum follows

$$\Phi_{EE}(\omega) = \sum_{k=-\infty}^{\infty} \sigma^2 \delta[k] e^{-i\omega k} = \sigma^2 \,, \tag{3.11}$$

which is constant in frequency domain.

3.3.2 Spectral estimation

Given one realization of a stationary stochastic process $x[k]$, with $k = 0, \ldots, N-1$, we now aim to (statistically) estimate the spectrum of the underlying stochastic process. One idea is to first estimate the autocorrelation by averaging its value over the sequence, i.e.,

$$\hat{\mathbf{R}}_{XX}^{(N)}[k] = \frac{1}{N} \sum_{m=0}^{N-k-1} x[m+k]x[m] \,, \quad \text{with} \quad k = 0, ..., N-1 \,, \tag{3.12}$$

denoting the time lag. Then we estimate the spectral density as $\hat{\Phi}_N(\omega) = T \sum_{k=0}^{N-1} \hat{\mathbf{R}}_{XX}^{N}[k] e^{-i\omega kT}$.

The second idea is based on the power spectral density of deterministic signals. The energy spectrum (energy spectral density) introduced in (2.36) is suitable for “pulse-like” signals whose energy is concentrated around one time window, for which FT is applicable. For stochastic processes with infinite energy, however, the energy spectrum can no longer characterize the process. However, we can define the energy per unit, which is measurable, to characterize the signal.

Power spectral density Given one realization of a stochastic process $x[k]$ with $k = 0, ..., N-1$, we now introduce the power spectrum, or power spectral density, to describe how signal power (energy per time unit) is distributed over frequency. For that, we adapt the energy spectral density $\Phi_{xx}(\omega)$ in (2.36) to truncated signal of length N and define power spectral density (PSD) as

$$\bar{\Phi}_{xx}^{(N)}(\omega) = \frac{1}{NT} \Phi_{xx}^{(N)}(\omega) = \frac{1}{NT} \big|\mathbb{X}_T^{(N)}(e^{i\omega T})\big|^2 \,. \tag{3.13}$$

Note here the subscripts have changed back to lower case to indicate the deterministic nature of the variable. The PSD given above changes value w.r.t. different realizations of the stochastic process. Statistically, however, its expectation approaches the spectrum in (3.9) when truncation length goes to infinity, namely,

$$\lim_{N\to\infty} \mathrm{E}\big\{\bar{\Phi}_{xx}^{(N)}(\omega)\big\} = \lim_{N\to\infty} \frac{1}{NT} \mathrm{E}\big\{\Phi_{xx}^{(N)}(\omega)\big\} = \Phi_{XX}(\omega) \,. \tag{3.14}$$

Note the difference of the subscripts denoting deterministic and stochastic variables. We provide the following proof for (3.14).

Proof. Given the expression in (3.13), we have

$$\mathrm{E}\Big\{\bar{\Phi}_{xx}^{(N)}(\omega)\Big\} = \frac{1}{NT}\mathrm{E}\Big\{\big|\mathbb{X}_T^{(N)}(e^{i\omega T})\big|^2\Big\} = \frac{1}{NT}\mathrm{E}\Big\{\mathbb{X}_T^{(N)}(e^{i\omega T})\big(\mathbb{X}_T^{(N)}(e^{i\omega T})\big)^*\Big\}$$

Given the expression of truncated DTFT in (2.40), we further obtain

$$\begin{aligned}
\mathrm{E}\big\{\bar{\Phi}_{xx}^{(N)}(\omega)\big\} &= \frac{T}{N}\mathrm{E}\Bigg\{\Big(\sum_{k=0}^{N-1} x[k]e^{-i\omega kT}\Big)\Big(\sum_{l=0}^{N-1} x[l]e^{-i\omega lT}\Big)^*\Bigg\} \\
&= \frac{T}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\mathrm{E}\big\{x[k]x^*[l]\big\}e^{-i\omega(k-l)T} \\
&= \frac{T}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\mathbf{R}_{XX}[k,l]e^{-i\omega(k-l)T} \quad \text{see (3.5)} \\
&= \frac{T}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\mathbf{R}_{XX}[k-l]e^{-i\omega(k-l)T} \quad \text{see (3.6)}
\end{aligned}$$

For an arbitrary function $f(\cdot)$, the following double summation formula holds [1, (1.7.6)]

$$\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} f(k-l) = \sum_{m=-N+1}^{N-1}(N-|m|)f(m)\,.$$

Therefore, the expected power spectrum can be further derived as

$$\begin{aligned}
\mathrm{E}\big\{\bar{\Phi}_{xx}^{(N)}(\omega)\big\} &= \frac{T}{N}\sum_{m=-N+1}^{N-1}(N-|m|)\mathbf{R}_{XX}[m]e^{-i\omega mT} \\
&= T\sum_{m=-N+1}^{N-1}\mathbf{R}_{XX}[m]e^{-i\omega mT} - T\sum_{m=-N+1}^{N-1}\frac{|m|}{N}\mathbf{R}_{XX}[m]e^{-i\omega mT}
\end{aligned}$$

For a wide-sense stationary stochastic process, its second-order moment is limited, thus

$$\lim_{N\to\infty}\sum_{m=-N+1}^{N-1}\frac{|m|}{N}\mathbf{R}_{XX}[m] = 0$$

Consequently, we obtain

$$\lim_{N\to\infty}\frac{1}{NT}\mathbf{E}\big\{\Phi_{xx}^{(N)}(\omega)\big\} = \Phi_{XX}(\omega)\,,$$

meaning the expected power spectrum given realizations of a stationary stochastic process approach its spectrum. ■

3.3.3 Periodogram

However, computing the spectrum according to (3.14) refers to continuous-frequency domain and infinite time extension. In engineering practice, however, we often have only one realization of finite length (not an infinite ensemble of realizations), and analyzing in discrete-frequency domain is necessary. We hereby introduce one common spectral estimator, the periodogram. Given one realization $\{x[k]\}_{k=0}^{N-1}$, we use its time normalized squared magnitude of its DTFT as a measure of the spectrum

$$\hat{\hat{\Phi}}_N(\omega) = \frac{1}{NT}\big|\mathbb{X}_T^{(N)}(e^{i\omega T})\big|^2\,.$$

Here, the double "hat" over Φ hints that it is an estimate from only one realization of finite length as indicated by (3.14). Computationally, we use DFT in (2.44) implemented with FFT for spectral estimation.

Note (2.43), we have

$$\hat{\hat{\Phi}}_N[n] = \frac{1}{NT}\big|T\mathbb{X}[n]\big|^2 = \frac{T}{N}\big|\mathbb{X}[n]\big|^2\,, \quad \text{with} \quad \mathbb{X}[n] = \sum_{k=0}^{N-1} x[k]e^{-i2\pi kn/N}\,.$$

One problem with periodogram is that the variance does not decrease with increasing the number of data points N, though the bias decreases with

$$\lim_{N\to\infty} \mathrm{E}\big\{\hat{\hat{\Phi}}_N[n]\big\} = \Phi_{XX}(\omega)\,.$$

Consequently, the periodogram is noisy no matter how long we record the time sequence. We introduce the following two methods to alleviate this issue.

Welch's method The Welchs's method decreases noise in periodogram by averaging. It can be summarize into the following two step

(1) splitting the data into R segments of length M (thus overall $N = RM$)

(2) computing periodogram for each segment and average them

The estimated spectrum is then obtained as

$$\hat{\hat{\Phi}}_{XX}[n] = \frac{1}{R}\sum_{r=1}^{R} \frac{T}{M}\hat{\hat{\Phi}}_M^{(r)}[n] \tag{3.15}$$

Thus, we can reduce the variance by a factor of R. However, dividing the whole sequence into segments also induced worse grid resolution in frequency domain for each periodogram. Given a realization of an unknown stochastic signal, one has to tune the parameter (R or M) to have enough resolution while reducing the variance for spectral estimation. One can also can this method as periodogram averaging.

Blackman-Tukey method In comparison to Welch's method, Blackman-Tukey method aims to reduce the noise (variance) of spectral estimate by weighting the corresponding autocorrelation estimate $\hat{\mathbf{R}}_{XX}[k]$ with a window function $w[k]$, which is equal to applying a low-pass filter to the periodogram (thus you can also call it periodogram smoothing). The key idea is shown in the formula below

$$\hat{\hat{\Phi}}_N(\omega) = T \sum_{k=-N+1}^{N-1} w[k]\,\hat{\mathbf{R}}_{XX}^{(N)}[k]e^{-i\omega kT}\,, \tag{3.16}$$

with $\hat{\mathbf{R}}_{XX}^{(N)}[k]$ given by (3.12). Multiplication in time domain corresponds to convolution of the periodogram in frequency domain according to (2.18).

Chapter 4

Filtering

We consider signals represented in discrete-time domain. Mathematically speaking, a filter (system) is a mapping $\mathcal{H}$ from an input sequence $u[k]$ to an output sequence $y[k]$, with $k \in \mathbb{Z}$, namely,

$$y[k] = \mathcal{H}(u[k])\,. \tag{4.1}$$

The definition above is general. In the lecture, we focus on linear filters. Given two input signals u_1 and u_2, a system $\mathcal{H}$ is linear if the following holds (time index is omitted for brevity)

$$\mathcal{H}(a_1u_1 + a_2u_2) = a_1\mathcal{H}(u_1) + a_2\mathcal{H}(u_2)\,,$$

with $a_1, a_2 \in \mathbb{R}$.

4.1 Representation of Linear Filter

There are multiple ways of representing linear filters. In the remainder of this section, we introduce representations in the form of pulse response, transfer function, difference equations, shift operator, and state space model.

4.1.1 Pulse response

For linear filters, we can explicitly express the mapping in (4.1) as convolution

$$y[k] = h \star u[k] = \sum_{n=-\infty}^{\infty} h[k-n]u[n] = \sum_{n=-\infty}^{\infty} h[n]u[k-n] \quad \text{(commutativity)}\,, \tag{4.2}$$

with $h[k]$ ($k \in \mathbb{Z}$) being the pulse response of the filter.

4.1.2 Transfer function

For discrete-time signals, one can apply z-transform for representation. Given a discrete-time sequence $x[k]$ ($k \in \mathbb{Z}$), its z-transform is defined as

$$\mathbb{X}(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k}\,, \tag{4.3}$$

with z being a continuous complex variable. By expressing $z = re^{i\omega}$ w.r.t. polar coordinates on the complex plane, we can reformulate z-transform as

$$\mathbb{X}(z)\Big|_{z=re^{i\omega}} = \sum_{k=-\infty}^{\infty} x[k](re^{i\omega})^{-k} = \sum_{k=-\infty}^{\infty} (x[k]r^{-k})e^{-i\omega k}\,, \tag{4.4}$$

which is the DTFT in (2.14) for signal $x[k]r^{-k}$ with sampling interval $T = 1$. One important property for z-transform is the time shifting property. A time shift of τ in $x[k]$ corresponds to a multiplication of $z^{-\tau}$ in z-domain, namely,

$$x[k-\tau] \leftrightarrow z^{-\tau}\,\mathbb{X}(z)\,. \tag{4.5}$$

The infinite series in definition (4.3) may not converge for some z values. The set of z values for which $\mathbb{X}(z)$ converges, namely,

$$|\mathbb{X}(z)| = \left|\sum_{k=-\infty}^{\infty} x[k]z^{-k}\right| \leq \sum_{k=-\infty}^{\infty} \left|x[k]z^{-k}\right| < \infty \quad \text{(triangle inequality)}\,,$$

is called region of convergence (ROC). Note that the ROC should always be specified along with its z-transform for the sake of completeness. If there is no ROC, then the z-transform does not exist.

Given the basic definition of linear filter in (4.2), one can apply z-transform in (4.3) to the pulse response $h[k]$

$$\mathbb{H}(z) = \sum_{k=-\infty}^{\infty} h[k]z^{-k}$$

leading to the filter's transfer function. For rational filters (or finite dimensional filters), their transfer function can be further expressed as ratio of polynomials

$$\mathbb{H}(Z) = \frac{\mathbb{B}(z)}{\mathbb{A}(z)}\,.$$

Setting $\mathbb{A}(z)$ and $\mathbb{B}(z)$ to be zero then yields the poles and zeros, respectively.

4.1.3 Difference equation

A rational linear filter can be expressed as a difference equation between the input $u[k]$ and output $y[k]$

$$y[k] + \sum_{i=1}^{n} a_i y[k-i] = \sum_{j=0}^{m} b_j u[k-j]\,. \tag{4.6}$$

By applying z-transform to (4.6) and exploiting the time-shift property in (4.5), one can obtain

$$\mathbb{Y}(z)\Big(1 + \sum_{i=1}^{n} a_i z^{-i}\Big) = \mathbb{U}(z)\sum_{j=0}^{m} b_j z^{-j}\,,$$

leading to the transfer function representation of the filter

$$\mathbb{H}(z) = \frac{\mathbb{Y}(z)}{\mathbb{U}(z)} = \frac{\sum_{j=0}^{m} b_j z^{-j}}{1 + \sum_{i=1}^{n} a_i z^{-i}} = \frac{b_0 z^n + b_1 z^{n-1} + \ldots \cdots + z^{n-m}}{z^n + a_1 z^{n-1} + \cdots + a_{n-1}z^1 + a_n}\,. \tag{4.7}$$

Therefore, the convolution given in (4.2) can be expressed in z-domain as $\mathbb{Y}(z) = \mathbb{H}(z)\mathbb{U}(z)$.

4.1.4 Transfer operator

Similar to (4.5), one can define a shift operator q in time domain, such that

$$qu[k] = u[k+1]\,. \tag{4.8}$$

Correspondingly, the transfer function in (4.7) can be expressed as the so-called transfer operator

$$\mathbb{H}(q) = \frac{\sum_{j=0}^{m} b_j q^{-j}}{1 + \sum_{i=1}^{n} a_i q^{-i}}\,, \tag{4.9}$$

which can be directly applied to signals in time-domain via

$$y[k] = \mathbb{H}(q)u[k]\,.$$

4.1.5 State-space model

The state-space representation has been very popular for designing filters in real-world applications. Given a system input $\underline{u}[k]$, mapping it to the output $y[k]$ can be done using the following difference equations

$$\begin{aligned} \underline{x}[k+1] &= \mathbf{A}\underline{x}[k] + \mathbf{B}\underline{u}[k]\,, \\ \underline{y}[k] &= \mathbf{C}\underline{x}[k] + \mathbf{D}\underline{u}[k]\,. \end{aligned} \tag{4.10}$$

Here, $\underline{x}[k] \in \mathbb{R}^d$ is the so-called state vector. Applying z-transform to (4.10) then yields its representation in transform function, i.e.,

$$\mathbb{H}(z) = \frac{\mathbb{Y}(z)}{\mathbb{U}(z)} = \mathbf{C}(z\mathbf{I} - A)^{-1}\mathbf{B} + \mathbf{D}\,, \tag{4.11}$$

with $\mathbf{I}$ being an identity matrix.

4.2 LTI System

In the lecture, we mainly focus on investigating properties of linear time invariant (LTI) filters and their design. A system is shift-invariant (or time-invariant) if the relation between the input $u[k]$ and the output $y[k]$ does not vary w.r.t. time. In other words, a time shift on the input, say $u_n[k] = u[k-n]$ with n being the difference, directly equates to a time shift of the output $y_n[k] = y[k-n]$. We show the following example of identifying LTI system and introduce a few properties.

Example 4.2.1. Consider the following two systems in discrete-time domain $k \in \mathbb{Z}$

(1) $y[k] = ku[k]$

(2) $y[k] = (u[k])^2$

Suppose a time shift of n occurs in the input with $u_n[k] = u[k-n]$. For system (1), we have output $ku_n[k] = ku[k-n]$. If we shift the output with the same difference, we obtain $y_n[k] = (k-n)u[k-n] \neq ku[k-n]$. Therefore, system (1) is time-variant, thus not LTI. Given the same shifted input, system (2) outputs $(u[k-n])^2$. And its shifted output $y_n[k] = (u[k-n])^2$. Thus, system (2) is time-invariant, but not LTI, since it is nonlinear.

4.2.1 Causality

A filter is causal if the output is independent of future inputs. In other words, causal filters have pulse response that exists only for $k > 0$, i.e.,

$$h[k] = 0\,, \quad k < 0. \tag{4.12}$$

If the condition above is satisfied for $k \leq 0$, the filter is strictly causal. Reversely, an anti-causal filter has output independent of past input and has pulse response

$$h[k] = 0\,, \quad k > 0. \tag{4.13}$$

If the condition holds when $k \geq 0$, then the system is strictly anti-causal. If a filer is neither causal nor anti-causal, then it is non-causal. In real-time implementations, causal filters must be deployed. Given a transfer function shown in (4.6), if the degree of numerator does not exceed the degree of the denominator, then it is a proper transfer function. If the degree of numerator is smaller than the degree of the denominator, then it is a strictly proper transfer function. A system with a proper transfer function is causal.

4.2.2 Stability

A filter is bounded input-bounded output (BIBO) stable if a bounded input leads to a bounded output. For a LTI filter, if its pulse response $h[k]$ is an l_1 signal, namely,

$$\sum_{k=-\infty}^{\infty} \big|h[k]\big| < \infty\,,$$

then it is stable. For a rational filter represented by a transfer function (4.7) or a transfer operator (4.9), we can use the following rules to tell its stability.

- a causal filter is stable ⇔ all poles are strictly inside the unit circle
- an anti-causal filter is stable ⇔ all poles are strictly outside the unit circle

Example 4.2.2. Tell the causality and stability of the following system

$$2y[k] - y[k-1] = 2u[k]\,.$$

The output $y[k]$ depends on present and past value of input $u[k]$. Thus, it is causal. Converting it to z-domain shows the following transfer function

$$\frac{\mathbb{Y}(z)}{\mathbb{U}(z)} = \frac{2}{2 - z^{-1}}\,,$$

leading to one pole at $z = 1/2$ inside the unit circle. Therefore, the system is stable.

4.3 Filtering

4.3.1 Signal and noise problem

Consider the following signal and noise model in discrete-time domain

$$u[k] = s[k] + n[k]\,, \quad k \in \mathbb{Z}$$

with $u[k]$ denoting the observation (measurements), $n[k]$ disturbances (noise), and $s[k]$ signal of interest. We assume that the noise and signal are stochastically independent. Without loss of generality, we set sampling interval $T = 1$ for brevity. Our goal is to obtain as much information as possible about $s[k]$ from observations $u[k]$. The approach to find a linear filter $\mathbb{H}(q)$ (represented in transfer operator) to perform filtering on $u[k]$ to obtain an estimate $\hat{s}[k] = \mathbb{H}(q)u[k]$ of the true signal $s[k]$. For evaluating the estimation quality, we introduce the following error term

$$\lambda[k] := \hat{s}[k] - s[k] = \big(\mathbb{H}(q) - 1\big)s[k] + \mathbb{H}(q)n[k]\,, \tag{4.14}$$

denoting the difference between the estimate and the underlying true signal. We further exploit the mean squared error for minimization, namely,

$$\mathrm{E}\big\{|\lambda[k]|^2\big\} = \mathbf{R}_{\Lambda\Lambda}[0]\,,$$

which is the autocorrelation (3.6) at time stamp $k = 0$. Based on the definition of spectrum in (3.9) (i.e., spectrum is the DTFT of autocorrelation), we can quantify it in frequency domain via the IDTFT of spectrum $\Phi_{\Lambda\Lambda}(\omega)$. It then follows

$$\mathrm{E}\big\{|\lambda[k]|^2\big\} = \text{IDTFT}\big\{\Phi_{\Lambda\Lambda}(\omega)\big\}|_{k=0} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{\Lambda\Lambda}(\omega)e^{i\omega k}\,\mathrm{d}\omega\Big|_{k=0} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{\Lambda\Lambda}(\omega)\,\mathrm{d}\omega\,.$$

We now derive the spectrum of the error term. According to the z-transform in (4.3) and its connection to DTFT in (4.4), the error term in (4.14) can be expressed in frequency domain as

$$\Lambda(e^{i\omega}) = \big(\mathbb{H}(e^{i\omega}) - 1\big)\mathbb{S}(e^{i\omega}) + \mathbb{H}(e^{i\omega})\mathbb{N}(e^{i\omega})\,.$$

Here, $\mathbb{H}(e^{i\omega})$ denotes the frequency function of the filter $\mathbb{H}(q)$. According to (3.10), the error spectrum can be derived as

$$\begin{aligned}
\Phi_{\Lambda\Lambda}(\omega) &= \mathrm{E}\big\{|\Lambda(e^{i\omega})|^2\big\} \\
&= \mathrm{E}\big\{|\mathbb{H}(e^{i\omega}) - 1|^2|\mathbb{S}(e^{i\omega})|^2 + |\mathbb{H}(e^{i\omega})|^2|\mathbb{N}(e^{i\omega})|^2\big\} \\
&= |\mathbb{H}(e^{i\omega}) - 1|^2\mathrm{E}\{|\mathbb{S}(e^{i\omega})|^2\} + |\mathbb{H}(e^{i\omega})|^2\mathrm{E}\{|\mathbb{N}(e^{i\omega})|^2\} \\
&= |\mathbb{H}(e^{i\omega}) - 1|^2\Phi_{SS}(\omega) + |\mathbb{H}(e^{i\omega})|^2\Phi_{NN}(\omega)\,.
\end{aligned} \tag{4.15}$$

Since the signal and noise are not correlated, their cross covariance function given by (3.4) equals to zero. Thus, the expected product of their frequency models equals to zero. To minimize the error spectrum in (4.15), we can design our filter $\mathbb{H}(q)$ with the following property in frequency domain

- $|\mathbb{H}(e^{i\omega}) - 1|$ is small where the signal spectrum $\Phi_{SS}(\omega)$ is large
- $|\mathbb{H}(e^{i\omega})|$ is small where the noise spectrum $\Phi_{NN}(\omega)$ is large

Therefore, knowledge about the signal and noise is essential for a good filtering performance. For that, we have the following two strategies.

- exploiting parametric models of signal $s[k]$ and noise $n[k]$, based on which the filter $\mathbb{H}[q]$ can be designed such that the mean squared error can be minimized → model-based optimal filter
- relying on a rough idea about the signal spectrum and designing a filter $\mathbb{H}[q]$ of a certain frequency model subjectively → frequency-selective filter

The first strategy yields a few very popular filters, such as the Wiener filter and Kalman filter, which will be introduced in the following chapters. The second strategy is a rough design (not optimal) according to our understanding about the signal and noise. Some basic filter types in this regard include low-pass filter, high-pass filter, band-pass filter, band-stop filter, etc.

4.3.2 Frequency property of linear filters

Given a linear filter $\mathbb{H}(q)$ of pulse response $h[k]$ and an input of complex exponential $u[k] = e^{i\omega k}$ (sampling interval $T = 1$). The output according to (4.2) in time domain follows convolution

$$y[k] = \sum_{n=-\infty}^{\infty} h[n]u[k-n] = \sum_{n=-\infty}^{\infty} h[n]e^{i\omega(k-n)} = e^{i\omega k}\sum_{n=-\infty}^{\infty} h[n]e^{-i\omega n} = e^{i\omega k}\,\mathbb{H}(e^{i\omega})\,.$$

Therefore, we obtain the filtering above expressed as the transfer operator

$$y[k] = \mathbb{H}(q)u[t] = \mathbb{H}(q)e^{i\omega k} = e^{i\omega k}\mathbb{H}(e^{i\omega})\,.$$

Rewriting the input via Euler's formula then yields

$$\mathbb{H}(q)\big(\cos(\omega k) + i\sin(\omega k)\big) = e^{i\omega k}|\mathbb{H}(e^{i\omega})|e^{i\phi} = |\mathbb{H}(e^{i\omega})|e^{i(\omega k+\phi)}\,, \tag{4.16}$$

where the frequency function of the filter is expressed in polar coordinates on the complex plane. $\phi(\omega) = \arg\big(\mathbb{H}(e^{i\omega})\big)$ quantifies the argument of frequency function. Decomposing the formula above w.r.t. the real and imaginary parts leads to

$$\begin{aligned}
\mathbb{H}(q)\cos(\omega k) &= |\mathbb{H}(e^{i\omega})|\cos(\omega k + \phi)\,, \\
\mathbb{H}(q)\sin(\omega k) &= |\mathbb{H}(e^{i\omega})|\sin(\omega k + \phi)\,.
\end{aligned}$$

Therefore, applying a linear filter $\mathbb{H}(q)$ to (a superposition of) sinusoidal inputs results in the same (superposition of) sinusoids amplified/attenuated by $|\mathbb{H}(e^{i\omega})|$ and delayed by $-\arg\big(\mathbb{H}(e^{i\omega})\big)/\omega$.

Filter phase As previously introduced, applying a linear filter $\mathbb{H}(q)$ leads to a phase shift of $\phi(\omega) = \arg\left(\mathbb{H}(e^{i\omega})\right)$ in normalized frequency domain. Ideally, a filter should not cause a time delay, namely $\phi(\omega) = 0, \forall\omega$. This gives the definition of the so-called zero-phase filter. Therefore, we have frequency function $\mathbb{H}(e^{i\omega}) = |\mathbb{H}(e^{i\omega})|$, which is a positive real number. Correspondingly, its pulse response is symmetric, namely,

$$h[k] = h[-k]\,.$$

Thus, a zero-phase filter is a non-causal filter, which cannot be implemented in real time.

In comparison with zero-phase filters, a linear phase filter has constant delay over all frequencies. The argument of its frequency function can be denoted as

$$\phi = -\omega\tau$$

with $-\tau$ denoting the phase delay. Correspondingly, the output follows (4.16) with

$$y[k] = \mathbb{H}(q)e^{i\omega k} = |\mathbb{H}(e^{i\omega})|e^{i\omega(k-\tau)}\,.$$

For implementing a low pass filter with linear phase delay, the linear phase needs to exist at least in the pass band.

4.3.3 Implementation

Given a desired frequency function $\mathbb{H}(e^{i\omega})$, we now aim to implement it in time domain. For online filtering, namely, the output $y[k]$ must exist as soon as the input $u[k]$ comes, a causal filter is required. Filters of other causalities can be implemented for offline filtering.

Causal filters According to pulse response representation in (4.2), a causal filter $h[k]$ can be implemented as

$$y[k] = \sum_{n=0}^{\infty} h[n]u[k-n]\,,$$

given its definition that $h[k] = 0, \forall k < 0$. Correspondingly, its transfer function follows

$$\mathbb{H}(z) = \sum_{k=0}^{\infty} h[k]z^{-k}\,, \tag{4.17}$$

and it can be implemented in terms of difference equation as in (4.6).

Anti-causal filters An anti-causal filter can be implemented with the following pulse response representation

$$y[k] = \sum_{n=-\infty}^{0} h[n]u[k-n]\,,$$

or transfer function

$$\mathbb{H}(z) = \sum_{k=-\infty}^{0} h[k]z^{-k}\,,$$

Given a stable causal filter with the transfer function in (4.17), we reverse its pulse response in time domain into $\tilde{h}[k]$ and obtain a non-causal filter represented by

$$\tilde{\mathbb{H}}(z) = \sum_{k=-\infty}^{0} \tilde{h}[k]z^{-k} = \sum_{k=-\infty}^{0} h[-k]z^{-k} = \sum_{k=0}^{\infty} h[k]z^{k} = \sum_{k=0}^{\infty} h[k](1/z)^{-k} = \mathbb{H}(z^{-1})\,.$$

Therefore, we can implement a stable non-causal filter with the following steps

- time-reverse of input $u[k]$ to get $u_\mathsf{r}[k]$

- filter u_{r} with $\mathbb{H}(z^{-1})$ to get $y_{\mathrm{r}}[k]$
- time-reverse of $y_{\mathrm{r}}[k]$ to get $y[k]$

Non-causal filters Zero-phase filters are non-causal. Given a typical filter $\mathbb{H}(q)$, a naive way to implement an offline zero-phase filter $\mathbb{H}'(q)$ can be expressed according to transfer function

$$H'(z) = \mathbb{H}(z)\mathbb{H}(z^{-1})\,.$$

Thus, its frequency function $\mathbb{H}'(e^{i\omega}) = \mathbb{H}(e^{i\omega})\mathbb{H}(e^{-i\omega}) = |\mathbb{H}(e^{i\omega})|^2$, which is a positive real number. This indicates that the filter is zero-phase.

4.4 Spectral Factorization

Given a filtered signal $y[n]$, its spectrum defined in (3.9) can also be formulated into transfer function via z-transform (sampling interval $T = 1$)

$$\tilde{\Phi}(z) = \sum_{k=-\infty}^{\infty} \mathbf{R}_{YY}[k]\, z^{-k}\,.$$

As discussed around (3.6), autocorrelation function is symmetric over time. Therefore, we have

$$\tilde{\Phi}(z) = \tilde{\Phi}(z^{-1})\,.$$

Therefore, the spectrum function is symmetric w.r.t. the unit circle on the complex plane, namely, a point $z = |z|e^{i\alpha}$ and its mirrored point $z^{-1} = 1/|z|\, e^{-i\alpha}$ have the same spectrum value. Thus, if n_i is a zero (or p_i a pole) of $\tilde{\Phi}(z)$, then n_i^{-1} must be also a zero (or p_i^{-1} a pole). Assuming there is no poles or zeros on the unit circle, the spectrum can then be decomposed as

$$\tilde{\Phi}(z) = C\,\frac{\prod_{i=1}^{m}(z - n_i)}{\prod_{i=1}^{n}(z - p_i)} \cdot \frac{\prod_{i=1}^{m}(z^{-1} - n_i)}{\prod_{i=1}^{n}(z^{-1} - p_i)} =: C\,\mathbb{H}(z) \cdot \mathbb{H}(z^{-1})\,, \tag{4.18}$$

with $\{n_i\}_{i=1}^{m}$ and $\{p_i\}_{i=1}^{m}$ denoted as zeros and poles inside the unit circle by convention. According to the conditions in (4.2.2), $\mathbb{H}(z)$ is then causal and stable, and $\mathbb{H}(z^{-1})$ is anti-causal and stable. The spectral factorization theorem plays a fundamental role in spectral analysis and system realization theory. According to (4.18), any spectrum $\Phi(\omega)$ that is non-zero and finite in $[-\pi, \pi]$ can be formulated as

$$\Phi(\omega) = \lambda|\mathbb{H}(e^{i\omega})|^2\,, \tag{4.19}$$

where systems $\mathbb{H}(z)$ and $1/\mathbb{H}(z)$ are both causal and stable. This also corresponds to the Wold's decomposition theorem introduced in (3.8).

Chapter 5

Signal Models

So far, we have covered a few nonparametric signal approaches to characterize stochastic signals, including realization, autocorrelation/autocovariance functions and spectra. These approaches have complexity in the same order of the a signal vector. We investigate in this chapter a few parametric models using a finite set of parameters, leading to much more compact representations for modeling stochastic signals. These parametric models will also be exploited for model-based optimal filtering introduced later. We first come back to the signal and noise problem

$$y[k] = s[k] + n[k]\,. \tag{5.1}$$

Given a series of measurements $\{y[l]\}_{l=0}^{k}$, estimation can be categorized into the following three cases w.r.t. the scenario when we perform it.

(1) filtering: estimate $s[k]$, denoted as $\hat{s}[k|k]$ or $\hat{s}_{k|k}$

(2) prediction: estimate $s[k+n]$, denoted as $\hat{s}[k+n|k]$ or $\hat{s}_{k+n|k}$

(3) smoothing: estimate $s[k-n]$, denoted as $\hat{s}[k-n|k]$ or $\hat{s}_{k-n|k}$

Here, n is a positive integer that is given. Note that a mathematical signal model is useful for (1), but necessary for (2) and (3).

5.1 Signal Models as Filtered White Noise

Given a series of measurements $\{y_k\}_{k=0}^{N-1}$ realized from (5.1), we now want to model the system mathematically in a parametric manner (refers to this chapter) and determine the parameters of the model (refers to next chapter). According to the Wold's decomposition theorem in (3.8), any stationary process $Y[k]$ can be realized by a white noise $u[k]$ and a filter, namely,

$$y[k] = \mathbb{T}(q)u[k]\,, \tag{5.2}$$

Here, $\mathbb{T}(q)$ denotes the filter represented by a transfer operator. Suppose that the white noise $u[k]$ has a variance of λ, we then obtain the following spectral relation similar to (4.15)

$$\Phi_{YY}(\omega) = \left|\mathbb{T}(e^{i\omega})\right|^2\Phi_{UU}(\omega) = \lambda\left|\mathbb{T}(e^{i\omega})\right|^2,$$

which is also shown in (4.19) by spectral factorization. Intuitively, the white noise injects the energy and the filter shapes the spectrum of the signal in frequency domain. However, many different stochastic processes have constant spectrum of λ in frequency domain. In practice, we can select a specific $u[k]$ of constant spectrum to shape the desired waveform in time domain. In general, we have the following design rules: (1) Select $\mathbb{T}(q)$ to get desired frequency content; (2) Select input $u[k]$ to get an output $y[k]$ approximating the desired signal shape in time domains. In the textbook, several signal sources holding of constant spectrum are introduced for shaping different waveforms.

5.2 Parametric Models

We now focus on parametric modeling of the filter $\mathbb{T}(q)$ in (5.2). In the remainder of this section, we introduce three parametric models, the autoregressive (AR) model, the autoregressive moving average (ARMA) model, and the state-space model.

5.2.1 Autoregressive (AR) model

An autoregressive model has its current output dependent only on the previous output and current input. Based on (4.6), it can be represented by the following difference equation

$$y[k] = -\sum_{i=1}^{n} a_i y[k-i] + e[k]\,, \tag{5.3}$$

with $e[k]$ being a white noise. It can be also based on the transfer operator $\mathbb{T}(q)e[k]$ with the filter represented by

$$\mathbb{T}(q) = \frac{1}{1+\sum_{i=1}^{n} a_i q^{-i}} = \frac{1}{1+a_1 q^{-1}+\cdots+a_n q^{-n}}\,, \tag{5.4}$$

which takes the inverse of a monic polynomials. Here, n denotes the order of the model, which corresponds to the number of unknown coefficients. Thus, we denote the model as $\mathrm{AR}(n)$.

5.2.2 Autoregressive moving average (ARMA) model

Based on the AR model, we can also integrate the past inputs via moving average. The difference equation in (5.3) is then converted into

$$y[k] = -\sum_{i=1}^{n} a_i y[k-i] + \sum_{j=0}^{m} c_j e[k-j]\,, \tag{5.5}$$

with the second polynomial term averaging the input white noise over a finite window m of the past besides the autoregressive term in (5.3). Therefore, we can denote an ARMA model as $\mathrm{ARMA(n{,}m)}$, and the AR model is a special case of ARMA model with $m=0$. For ARMA model, we have the shift operator

$$\mathbb{T}(q) = \frac{\sum_{j=0}^{m} c_j q^{-j}}{1+\sum_{i=1}^{n} a_i q^{-i}} = \frac{c_0 + c_1 q^{-1}+\cdots+c_m q^{-m}}{1+a_1 q^{-1}+\cdots+a_n q^{-n}}\,, \tag{5.6}$$

such that $y[k] = \mathbb{T}(q)e[k]$. Note that the denominator is also a monic polynomial. Given the transfer operator in (5.6), we obtain the spectrum of ARMA model as follows

$$\Phi_{YY}(\omega) = \lambda\big|\mathbb{T}(e^{i\omega})\big|^2 = \lambda\frac{\big|c_0 + c_1 e^{-i\omega} + \cdots c_m e^{-i\omega m}\big|^2}{\big|1 + a_1 e^{-i\omega} + \cdots + a_n e^{-i\omega n}\big|^2}\,. \tag{5.7}$$

If we express all the complex exponential functions according to Euler's formula, the result will always be a rational function of $\cos(\omega)$, which is generally valid for all ARMA models (therefore also valid for AR models). We show the following example for demonstration.

Example 5.2.1. Given a system in the form $y[k] = \mathbb{T}(q)e[k]$, with filter $\mathbb{T}(q)$ defined as

$$\mathbb{T}(q) = \frac{1+0.5q^{-1}}{1+0.8q^{-1}} = \frac{q+0.5}{q+0.8}\,.$$

We compute its spectrum according to (5.7) and obtain

$$\Phi_{YY}(\omega) = \lambda\big|\mathbb{T}(e^{i\omega})\big|^2 = \frac{(e^{i\omega}+0.5)(e^{-i\omega}+0.5)}{(e^{i\omega}+0.8)(e^{-i\omega}+0.8)} = \lambda\,\frac{1.25+\cos(\omega)}{1.64+1.6\cos(\omega)}\,,$$

which is a rational function in $\cos(\omega)$.

A natural follow-up question now is how to convert a spectrum to an ARMR model (if applicable). Given a non-zero and finite spectrum $\Phi(\omega)$, one can always perform spectral factorization as shown in (4.18). Note that spectrum of an ARMA model is a rational functions of $\cos(\omega)$, we can then replace each $\cos(\omega)$ with $\frac{1}{2}(q + q^{-1})$ according to Euler's formula. Then, the spectrum can be factorized as

$$\Phi(\omega) = \lambda \big|\mathbb{T}(e^{i\omega})\big|^2 = \lambda \mathbb{T}(e^{i\omega})\mathbb{T}(e^{-i\omega})\,,$$

where the filter $\mathbb{T}(q)$ and $\mathbb{T}^{-1}(q)$ is causal and stable with all poles and zeros inside the unit circle.

Covariance/Autocorrelation There is no simple general procedure to compute the autocorrelation of ARMA models by definition. The following simplified example of an AR model gives an idea about the typical way of handling a specific case.

Example 5.2.2. We compute the spectrum for the following $AR(2)$ model

$$y[k] + a_1 y[k-1] + a_2 y[k-2] = e[k]\,, \tag{5.8}$$

with $e[k]$ being a white Gaussian noise of variance λ (zero-mean). We multiply the equation above with $y[k]$, $y[k-1]$ and $y[k-2]$ and take expectation, and obtain

$$\begin{aligned}
\mathrm{E}\big\{y[k](y[k] + a_1 y[k-1] + a_2 y[k-2])\big\} &= \mathrm{E}\big\{y[k]e[k]\big\}\,,\\
\mathrm{E}\big\{y[k-1](y[k] + a_1 y[k-1] + a_2 y[k-2])\big\} &= \mathrm{E}\big\{y[k-1]e[k]\big\}\,,\\
\mathrm{E}\big\{y[k-2](y[k] + a_1 y[k-1] + a_2 y[k-2])\big\} &= \mathrm{E}\big\{y[k-2]e[k]\big\}\,.
\end{aligned}$$

By definition of autocorrelation w.r.t time lag, the equations above leads to

$$\begin{aligned}
&\mathbf{R}_{YY}[0] + a_1\mathbf{R}_{YY}[1] + a_2\mathbf{R}_{YY}[2] = \mathrm{E}\big\{e[k]e[k]\big\} = \lambda\,,\\
&\mathbf{R}_{YY}[1] + a_1\mathbf{R}_{YY}[0] + a_2\mathbf{R}_{YY}[1] = 0\,,\\
&\mathbf{R}_{YY}[2] + a_1\mathbf{R}_{YY}[1] + a_2\mathbf{R}_{YY}[0] = 0\,.
\end{aligned} \tag{5.9}$$

Note that $y[k-1]$ and $y[k-2]$ are not correlated with input $e[k]$ due to causality during the derivations above. Further, we can obtain the value of $\mathbb{R}_{YY}[n]$ for $n = 0, 1, 2$ from the three equations in (5.9). We can again multiply $y[k-n]$ with $n \geq 3$ to both sides of (5.8) and take the expectation. This yields

$$\mathbf{R}_{YY}[n] + a_1\mathbf{R}_{YY}[n-1] + a_2\mathbf{R}_{YY}[n-2] = 0\,, \tag{5.10}$$

based on which we can compute the autocorrelation $\mathbf{R}_{YY}[n]$ ($n \geq 3$) recursively. Afterward, the function value of $\mathbf{R}_{YY}[n]$ can be computed over all integers due to symmetry of autocorrelation.

5.2.3 State-space model

State-space models are very powerful parametric tool to for modeling time-invariant as well as time-variant systems. Also they often reflect the underlying physical systems in a straightforward manner. Given the original signal model in (5.2), it can be converted to a state-space model of the following "innovation form"

$$\begin{aligned}
\underline{x}[k+1] &= \mathbf{A}\underline{x}[k] + \mathbf{B}\underline{e}[k]\,,\\
\underline{y}[k] &= \mathbf{C}\underline{x}[k] + \underline{e}[k]\,.
\end{aligned}$$

Here, the innovation term refers to the white noise $\underline{e}[k]$ of variance λ corresponding to the Wold's decomposition theorem. Based on the shift operator q, we have the following transfer operator to represent the filter in (5.2), namely,

$$\mathbb{T}(q) = \mathbf{C}(q\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{I}\,.$$

When we start up with a state-space model, it may not be initially a stationary process. However, if $\mathbf{A}$ is stable, i.e., the absolute value of all eigenvalues of $\mathbf{A}$ are all below one, then all poles are inside of the unit circle. Consequently, the autocorrelation of the state, namely, $\pi_k = \mathrm{E}\{\underline{x}[k]\underline{x}^\top[k]\}$, will converge to a constant matrix $\bar{\pi}$ as $k \to \infty$, with $\bar{\pi}$ being the solution to the Lyapunov equation

$$\bar{\pi} = \mathbf{A}\bar{\pi}\mathbf{A}^\top + \lambda\mathbf{B}\mathbf{B}^\top \,. \tag{5.11}$$

Autocorrelation for state-space model State-space models are suitable for computing the covariance/autocorrelation functions. The cross-correlation function between the input noise $\underline{e}[k]$ and the noise follows

$$\mathbf{R}_{YE}[n] = \mathrm{E}\{\underline{y}[k]\underline{e}[k-n]\} = \begin{cases} \lambda\mathbf{C}\mathbf{A}^{n-1}\mathbf{B}\,, & n > 0 \\ \lambda\,, & n = 0 \\ 0\,, & n < 0 \end{cases}$$

The derivation is omitted here as it resembles the procedure in Example 5.2.2, and one can refer to (3.7) for autocorrelation of white noise. Similarly, we also have the autocorrelation of the output $y[k]$ derived as the following form.

$$\mathbf{R}_{YY}[n] = \begin{cases} \mathbf{C}\mathbf{A}\bar{\pi}\mathbf{C}^\top + \lambda\mathbf{C}\mathbf{A}^{n-1}\mathbf{B}\,, & |n| > 0 \\ \mathbf{C}\bar{\pi}\mathbf{C}^\top + \lambda\,, & n = 0 \end{cases}$$

Here, $\bar{\pi}$ is the autocorrelation of the state given by the Lyapunov equation in (5.11).

5.3 Signal Prediction

In many cases, we also aim to predict $y[k+n]$ $(n > 0)$ given our past measurements $y[0], ..., y[k]$. According to (5.2) for linear filters, this can be formulated as

$$y[k+n] = \mathbb{T}(q)e[k+n]\,.$$

For a linear filter, we can convert the transfer operator above into its convolution representation via the pulse response of the filter (causal) $t[i]$ and obtain

$$y[k+n] = \sum_{i=0}^{\infty} t[i]e[k+n-i] = \sum_{i=0}^{n-1} t[i]e[k+n-i] + \sum_{i=n}^{\infty} t[i]e[k+n-i]\,.$$

The first sum has unknown future noise at time stamp k, which we can assume to be zero given it is realized by white noise. Now we can predict the estimate

$$\hat{y}[k+n|k] = \sum_{i=n}^{\infty} t[i]e[k+n-i] =: \tilde{\mathbb{T}}_n(q)e[k]\,. \tag{5.12}$$

Here, the noise term $e[k]$ can be obtained via the whitening filter $\mathbb{T}^{-1}(q)$, i.e., $e[k] = \mathbb{T}^{-1}(q)y[k]$. The spectral factorization theorem guarantees that this exists and it is stable. Therefore, we obtain the predicted estimate

$$\hat{y}[k+n|k] = \tilde{\mathbb{T}}_n(q)\mathbb{T}^{-1}(q)y[k]\,. \tag{5.13}$$

The following example demonstrate prediction for one step ($n = 1$).

Example 5.3.1. For deriving the one-step predictor, we first compute the filter applied to non-zero input in (5.12). According to the definition of z-transform and its relation to the transfer operator, we have

$$\tilde{\mathbb{T}}_1(q) = \sum_{i=1}^{\infty} t[i]q^{1-i} = q\sum_{i=1}^{\infty} t[i]q^{-i} = q\Big(\sum_{i=0}^{\infty} t[i]q^{-i} - t_0\Big) = q\big(\mathbb{T}(q) - t_0\big)\,. \tag{5.14}$$

Further, we apply the whitening filter to recover the input term. Following (5.13), we then obtain the predicted estimate

$$\begin{aligned}\hat{y}[k+1|k] &= q\big(\mathbb{T}(q) - t_0\big)\mathbb{T}^{-1}(q)y[k] \\ &= q\big(1 - t_0\mathbb{T}^{-1}(q)\big)y[k]\,. \end{aligned} \tag{5.15}$$

Based on the property of shift operator q, we also have $\hat{y}[k+1|k] = \big(1 - t_0\mathbb{T}^{-1}(q)\big)y[k+1]$.

Chapter 6

Model Estimation

In the last chapter, we have introduced a few typical parametric filter models and their properties for signal modeling. We now focus on estimating the parameters of a signal model given observed data.

6.1 Model Estimation

Given a model structure $y[k] = \mathbb{T}(q)u[k]$ and a finite length of observations $\{y[k]\}_{k=0}^{N-1}$, we aim to determine the model parameters, namely, the set of coefficients of $\mathbb{T}(q)$, which are summarized as θ, and the variance λ of the input noise. According to (5.15), the one-step prediction of a linear filter yields estimate

$$\hat{y}_{k+1|k}(\theta) = qy_k - t_0\mathbb{T}^{-1}(q)y_k = y_{k+1} - t_0e_k\,, \tag{6.1}$$

based on which the variance of the predicted error follows

$$\mathrm{E}\big\{(\hat{y}_{k+1|k}(\theta) - y_{k+1})^2\big\} = \mathrm{E}\{\big|t_0e_k\big|^2\} = t_0^2\lambda\,. \tag{6.2}$$

provided that true model parameters are available. Here, we rewrite time stamps as subscripts for conciseness. Given measurements $\{y_i\}_{i=0}^{N-1}$ of finite length, the basic idea for parameter estimation is then to find a set of optimal parameters θ^* such that the mean squared error of prediction is minimized, namely,

$$\theta^* = \arg\min_{\theta}\left\{\frac{1}{N}\sum_{k=0}^{N-1}\Big(y_{k+1} - \hat{y}_{k+1|k}(\theta)\Big)^2\right\}. \tag{6.3}$$

The quadratic formulation above is therefore called a least squares problem, which is a fundamental principal for parameter estimation.

6.2 Parameter Estimation of AR Models

Consider an AR model $AR(n)$ as defined in (5.3)

$$y_k + a_1y_{k-1} + a_2y_{k-2} + \cdots + a_ny_{k-n} = e_k\,.$$

Its one-step prediction gives estimate as follows

$$\hat{y}_{k|k-1}(\theta) = -a_1y_{k-1} - a_2y_{k-2} - \cdots - a_ny_{k-n}\,. \tag{6.4}$$

We concatenate the model parameters and outputs into vectors

$$\underline{\theta} = [\,a_1,\,a_2,\,\ldots,a_n\,]^\top \quad \text{and} \quad \underline{\varphi}_k = [\,-y_{k-1},\,-y_{k-2},\,\ldots,y_{k-n}\,]^\top\text{(regression vector)}\,,$$

respectively. The signal model then takes the following form based on scalar product

$$y_k = \underline{\varphi}_k^\top \underline{\theta} + e_k \,, \tag{6.5}$$

and the predicted estimate is given by $\hat{y}_{k|k-1}(\theta) = \underline{\varphi}_k^\top \underline{\theta}$. Given the measurements $\{y_k\}_{k=1}^N$, we can then formulate the least-squares problem for estimating parameter vector θ, i.e.,

$$\hat{\underline{\theta}} = \arg\min_{\underline{\theta}} \mathcal{V}(\underline{\theta}) \,, \quad \text{with} \quad \mathcal{V}_N(\underline{\theta}) = \frac{1}{N}\sum_{k=1}^N (y_k - \underline{\varphi}_k^\top \underline{\theta})^2 \tag{6.6}$$

being the loss function. For solving the optimization above, we compute the first derivative of the objective w.r.t. θ

$$\frac{\mathrm{d}\mathcal{V}_N(\underline{\theta})}{\mathrm{d}\underline{\theta}} = -\frac{2}{N}\sum_{k=1}^N \underline{\varphi}_k (y_k - \underline{\varphi}_k^\top \underline{\theta})$$

and set it to be zero. This yields

$$\frac{1}{N}\left(\sum_{k=1}^N \underline{\varphi}_k \underline{\varphi}_k^\top\right)\underline{\theta} = \frac{1}{N}\sum_{k=1}^N \underline{\varphi}_k y_k \,. \tag{6.7}$$

The equation above can be formulated as the normal equations of the least-squares problem

$$\mathbf{R}_N \underline{\theta} = \underline{f}_N \,, \quad \text{with} \quad \mathbf{R}_N = \frac{1}{N}\sum_{k=1}^N \underline{\varphi}_k \underline{\varphi}_k^\top \quad \text{and} \quad \underline{f}_N = \frac{1}{N}\sum_{k=1}^N \underline{\varphi}_k y_k \,. \tag{6.8}$$

Consequently, we obtain the optimal parameters in least-squares sense of the following closed form

$$\hat{\theta} = \mathbf{R}_N^{-1} \underline{f}_N \,. \tag{6.9}$$

We also obtain the variance estimate of the input noise via

$$\hat{\lambda} = \mathcal{V}_N(\hat{\theta}) \,. \tag{6.10}$$

Besides the ordinary least-squares method above, there are also other approaches to estimate the parameters. We show one alternative solution in the following example.

Example 6.2.1. Consider the following $\mathrm{AR}(2)$ model

$$y_k - 1.6\, y_{k-1} + 0.64\, y_{k-2} = e_k \,, \tag{6.11}$$

with $\mathrm{var}\{e_k\} = 1$. According to (6.5), we have regression vector $\underline{\varphi}_k = [\, y_{k-1}, y_{k-2}\,]^\top$ and parameter vector $\underline{\theta} = [\, a_1, a_2\,]^\top$ of true values $\underline{\theta}^\circ = [\, 1.6, -0.64\,]^\top$. Suppose we are given measurements $\{y_k\}_{k=1}^N$ construct the normal equations in (6.8), we have

$$\mathbf{R}_N = \frac{1}{N}\sum_{k=1}^N \underline{\varphi}_k \underline{\varphi}_k^\top = \frac{1}{N}\sum_{k=1}^N \begin{bmatrix} y_{k-1}^2 & y_{k-1}y_{k-2} \\ y_{k-2}y_{k-1} & y_{k-2}^2 \end{bmatrix} \quad \text{and} \quad \underline{f}_N = \frac{1}{N}\sum_{k=1}^N \underline{\varphi}_k y_k = \frac{1}{N}\sum_{k=1}^N \begin{bmatrix} y_{k-1}y_k \\ y_{k-2}y_k \end{bmatrix} \,.$$

Thus, we obtain normal equations of the following form

$$\frac{1}{N}\begin{bmatrix} \sum_{i=1}^N y_{k-1}^2 & \sum_{i=1}^N y_{k-1}y_{k-2} \\ \sum_{i=1}^N y_{k-2}y_{k-1} & \sum_{i=1}^N y_{k-2}^2 \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \frac{1}{N}\begin{bmatrix} \sum_{i=1}^N y_{k-1}y_k \\ \sum_{i=1}^N y_{k-2}y_k \end{bmatrix} \,. \tag{6.12}$$

For a stationary stochastic process, the equation above can be reformulated by exploiting (estimated) autocorrelation, namely,

$$\begin{bmatrix} \hat{R}_{YY}[0] & \hat{R}_{YY}[1] \\ \hat{R}_{YY}[1] & \hat{R}_{YY}[0] \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \hat{R}_{YY}[1] \\ \hat{R}_{YY}[2] \end{bmatrix} \,. \tag{6.13}$$

When $N \to \infty$, the autocorrelation terms will converge, and one can express the normal equations in (6.8) asymptotically as

$$\bar{\mathbf{R}}\theta = \underline{\bar{f}}, \tag{6.14}$$

with $\bar{\mathbf{R}}$ and $\underline{\bar{f}}$ given by the asymptotic values of autocorrelation. Note that matrix $\bar{\mathbf{R}}$ is a Toeplitz matrix, in which the diagonal elements are constant. (6.14) is also called Yule-Walker equation. It unveils the correspondence between the AR parameters and the autocorrelation/covariance function of the process. Inversely, one can exploit this correspondence to determine the parameters as a function w.r.t. the autocorrelation.

6.2.1 Statistical properties

Suppose the measurements $\{y_k\}_{k=1}^{N}$ are realized by the model parameters $\underline{\theta}^{\circ}$ and noise variance λ° of the white noise input, i.e.,

$$y_k = \underline{\varphi}_k^{\top}\underline{\theta}^{\circ} + e_k \,.$$

The estimate given by least squares in (6.9) can then be expressed as

$$\begin{aligned}
\hat{\underline{\theta}} = \mathbf{R}_N^{-1}\underline{f}_N &= \mathbf{R}_N^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\underline{\varphi}_k y_k\right) \\
&= \mathbf{R}_N^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\underline{\varphi}_k\left(\underline{\varphi}_k^{\top}\underline{\theta}^{\circ} + e_k\right)\right) \\
&= \mathbf{R}_N^{-1}\left(\mathbf{R}_N\underline{\theta}^{\circ} + \frac{1}{N}\sum_{k=1}^{N}\underline{\varphi}_k e_k\right) \\
&= \underline{\theta}^{\circ} + \mathbf{R}_N^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\underline{\varphi}_k e_k\right).
\end{aligned} \tag{6.15}$$

Therefore, the least-square estimate $\hat{\underline{\theta}}$ is unbiased, namely, $\mathrm{E}\{\hat{\underline{\theta}}\} = \underline{\theta}^{\circ}$. According to the law of large numbers, when $N \to \infty$, we have the parameter estimate approaching the true value, i.e., $\hat{\underline{\theta}} \to \underline{\theta}^{\circ}$. It is important to measure the uncertainty of the estimate in parameter estimation. For that, we compute covariance of the estimated parameter $\hat{\underline{\theta}}$ as follows

$$\begin{aligned}
\mathrm{cov}\{\hat{\underline{\theta}}\} &= \mathrm{E}\left\{\left(\hat{\underline{\theta}} - \mathrm{E}\{\hat{\underline{\theta}}\}\right)\left(\hat{\underline{\theta}} - \mathrm{E}\{\hat{\underline{\theta}}\}\right)^{\top}\right\} \\
&= \mathrm{E}\left\{\left(\hat{\underline{\theta}} - \underline{\theta}^{\circ}\right)\left(\hat{\underline{\theta}} - \underline{\theta}^{\circ}\right)^{\top}\right\}.
\end{aligned}$$

Based on the expression (6.15) and note that $\mathbf{R}_N$ is symmetric, we further obtain

$$\begin{aligned}
\mathrm{cov}\{\hat{\underline{\theta}}\} &= \mathrm{E}\left\{\mathbf{R}_N^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\underline{\varphi}_k e_k\right)\left(\frac{1}{N}\sum_{n=1}^{N}\underline{\varphi}_n e_n\right)^{\top}\mathbf{R}_N^{-1}\right\} \\
&= \frac{1}{N^2}\mathbf{R}_N^{-1}\left(\sum_{k=1}^{N}\sum_{n=1}^{N}\underline{\varphi}_k\underline{\varphi}_n^{\top}\mathrm{E}\{e_k e_n\}\right)\mathbf{R}_N^{-1}\,.
\end{aligned}$$

The equation above refers to the autocorrelation of white noise, which is given in (3.7). Thus, the equation above can be further derived as

$$\begin{aligned}
\mathrm{cov}\{\hat{\underline{\theta}}\} &= \frac{1}{N^2}\mathbf{R}_N^{-1}\left(\sum_{k=1}^{N}\sum_{n=1}^{N}\underline{\varphi}_k\underline{\varphi}_n^{\top}\lambda\delta[k-n]\right)\mathbf{R}_N^{-1} \\
&= \frac{\lambda}{N}\mathbf{R}_N^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\underline{\varphi}_k\underline{\varphi}_k^{\top}\right)\mathbf{R}_N^{-1}\,.
\end{aligned}$$

Note the definition of $\mathbf{R}_N$ in (6.8), we now obtain the covariance estimate of the in the following form

$$\mathrm{cov}\{\hat{\underline{\theta}}\} = \frac{\hat{\lambda}}{N}\mathbf{R}_N^{-1} \,. \tag{6.16}$$

The uncertainty measure above can then be exploited for setting up confidence intervals for the estimated parameters.

6.3 Parameter Estimation of ARMA model

Given an $\mathrm{ARMA}(n, m)$ structure according to (5.5) (coefficients for inputs are divided with c_0)

$$y_k + \sum_{i=1}^{n} a_i y_{k-i} = 1 + \sum_{j=1}^{m} c_j e_{k-j} \,, \tag{6.17}$$

we concatenate its coefficients into vector

$$\underline{\theta} = [\, a_1, \ldots, a_n,\, c_1, \ldots, c_m \,]^\top \,.$$

With the transfer operator in (5.6) formulated as rational function

$$\mathbb{T}(q; \underline{\theta}) = \frac{1 + c_1 q^{-1} + \ldots c_m q^{-m}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}} =: \frac{\mathbb{C}(q; \underline{\theta})}{\mathbb{A}(q; \underline{\theta})} \,,$$

we can further obtain estimate of one-step prediction according to (5.15) as

$$\begin{aligned} \hat{y}_{k|k-1}(\underline{\theta}) &= q\big(1 - \mathbb{T}^{-1}(q)\big) y_{k-1} \\ &= \left(1 - \frac{\mathbb{A}(q; \underline{\theta})}{\mathbb{C}(q; \underline{\theta})}\right) y_k \end{aligned} \tag{6.18}$$

Note the derivation above refers to the property of the shift operator in (4.8). The residual term then follows

$$\epsilon_k(\underline{\theta}) = y_k - \hat{y}_{k|k-1}(\underline{\theta}) = \frac{\mathbb{A}(q; \underline{\theta})}{\mathbb{C}(q; \underline{\theta})} y_k \,, \tag{6.19}$$

and we have the following least squares problem of estimating the coefficients in ARMA models

$$\hat{\underline{\theta}} = \arg\min_{\underline{\theta}} \mathcal{V}_N(\underline{\theta}) \,, \quad \text{with} \quad \mathcal{V}_N(\underline{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \left(\frac{\mathbb{A}(q; \underline{\theta})}{\mathbb{C}(q; \underline{\theta})} y_k \right)^2 \tag{6.20}$$

being the loss function. Unlike estimating parameters of AR models, the optimization problem in (6.20) refers to nonlinear least squares w.r.t. $\underline{\theta}$ and cannot be solved in closed form. Thus, numerical solvers, e.g., Newton's methods, can be employed. In general, we start up with an initial guess of $\hat{\underline{\theta}}_0$. Afterward, we update the estimate with iterations according to

$$\hat{\underline{\theta}}_{i+1} = \hat{\underline{\theta}}_i - \alpha \big(\nabla^2 \mathcal{V}_N(\underline{\theta})\big)^{-1} \nabla \mathcal{V}_N(\underline{\theta})$$

until convergence. Here, α denotes step-length in each iteration. $\nabla \mathcal{V}_N(\theta)$ and $\nabla^2 \mathcal{V}_N(\theta)$ are the gradient and the Hessian of the loss function w.r.t. $\underline{\theta}$, respectively[1]. Afterward, we obtain the variance estimate of the input $e[k]$, and the covariance estimate of $\underline{\theta}$ is approximated by

$$\mathrm{cov}(\hat{\underline{\theta}}) \approx \frac{\hat{\lambda}}{N} \Big(\frac{1}{N} \sum_{k=1}^{N} \nabla \epsilon_k (\nabla \epsilon_k)^\top \Big)^{-1} \,,$$

[1] In many cases, the Hessian can be approximated using the gradient, resulting in the Gauss-Newton method.

with residual term $\epsilon_k = \epsilon_k(\underline{\theta})$ given by (6.19). The derivation of asymptotic properties (including the covariance estimate given above) resembles the one given in Sec. 6.2.1 with approximations via linearization of the residual.

6.4 Practical Aspects

6.4.1 Preprocessing

It is very common that data needs to be first preprocessed before applying least squares. The following issues are recommended to check.

- detrending: remove non-zero mean or linear trend (e.g., via fitting to a linear model via least squares) in the data to avoid biased estimates.
- decimation: downsampling to 10 times of the bandwidth, if data is oversampled (redundant information); resampling needed if sampling frequency too small.
- pre-filtering: removing disturbances and frequency components that should not be modeled.

6.4.2 Model validation

In practice, we split the data set into two parts, one for parameter estimation, the other for validation. As a rule of thumb, given N observations, we typically use $2N/3$ for estimation and the rest for validation. The following criteria are exploited to tell if the obtain estimates deliver a model close to the underlying true process.

Prediction We compare predicted values with observation. Multiple steps of prediction is more revealing in general.

Spectral analysis We compare the spectrum given by the estimated model, namely

$$\hat{\Phi}_{YY}(\omega) = \hat{\lambda}\big|\mathbb{T}(e^{i\omega}, \hat{\underline{\theta}})\big|^2$$

with the periodogram (non-parametric) in Sec. 3.3.3 directly computed given the data.

Residual whiteness tests We can plot the residual estimate $\epsilon_k = y_k - \hat{y}_{k|k-1}$ to check if it resembles a realization of white noise. Further, we can plot the autocorrelation $\hat{\mathbf{R}}_{\epsilon\epsilon}[k]$ of the residual and the cross correlation $\hat{\mathbf{R}}_{\epsilon y}[k]$ between the residual and the observation. In principle, we should have $\hat{\mathbf{R}}_{\epsilon\epsilon}[n] = \lambda\delta[n]$ (a peak at $n = 0$), and $\hat{\mathbf{R}}_{\epsilon y}[n] = \mathrm{E}\big\{\epsilon[k]y[k-n]\big\} = 0$ when $n > 0$ due to causality.

The methods above are good at detecting under-modeling (model is not descriptive enough). However, over-modeling are harder to detect. For a specific parameter estimate $\hat{\theta}_i$, we can check its significance to see if its absolute value is smaller than its uncertainty, namely, $|\hat{\underline{\theta}}_i|^2 < (\mathrm{cov}\{\hat{\underline{\theta}}\})_{ii}$. If so, then the model order can be reduced correspondingly.

6.4.3 Model order selection

We denote the parameter estimates of order n, e.g., for an $\mathrm{AR}(n)$, using N samples as $\hat{\underline{\theta}}_N^n$. Suppose there exists a nested model structure, namely, a model of lower order is a special case of a higher-order model. Then the loss function value $\mathcal{V}^{\mathrm{e}}(\hat{\underline{\theta}}_N^n)$ for the estimation data declines as the order n grows, namely, $\mathcal{V}^{\mathrm{e}}(\hat{\underline{\theta}}_N^{n+1}) < \mathcal{V}^{\mathrm{e}}(\hat{\underline{\theta}}_N^n)$, $\forall n$. In this case, it is not possible to find the proper order of the model using the data for estimation.

One solution is to predict the loss function value $\mathcal{V}^{\mathsf{v}}(\hat{\underline{\theta}}_N^n)$ on the validation data. It can be shown that the minimum

$$\hat{n} = \arg\min_n \mathcal{V}^{\mathsf{v}}(\hat{\underline{\theta}}_N^n)$$

is a consistent estimator of the true model order n° (when $N \to \infty$). Thus, finding an optimal order for the model is possible by minimizing the loss on validation data. Another solution is to add a penalty factor $\mathcal{P}_N(n)$ to the loss computation of estimation data via

$$\mathcal{U}_N(n) = \mathcal{V}^{\mathrm{e}}(\underline{\hat{\theta}}_N^n)\big(1 + \mathcal{P}_N(n)\big)\,,$$

which delivers consistent estimate of the model order n. Some well-known penalty term include

- Akaike's information criterion A (AIC) with $\mathcal{P}_N(n) = 2n/N$,
- Akaike's information criterion B (BIC) with $\mathcal{P}_N(n) = n\log(N)/N$ (also known as the Bayesian information criterion).

Note that AIC tends to over-estimate the model (large order n) when the sample size N is small.

Chapter 7

Wiener Filtering

The frequency selective filter introduced in Chap. 4 only relies on a rough idea about the noise and signal spectra. They are in general applicable when the signal and noise are concentrated in different frequency bands. However, when signal and noise have overlapping spectra, issues may occur as information can be lost by filtering. In this regard, optimal filtering is more desired which aims to deliver estimates by exploiting the precise signal and noise spectra.

7.1 Geometric Interpretation of Linear Estimation

In Sec. 6.2 of Chap. 6, we have shown the derivation of linear estimators with the example of AR models. We now recall a few tools in linear algebra and interpret linear regression from a geometric perspective. Given $\forall \underline{x}, \underline{y} \in \mathbb{R}^d$, their inner product is defined as

$$\langle \underline{x}, \underline{y} \rangle = \underline{x}^\top \underline{y} \,.$$

The norm of an arbitrary vector $\underline{x} \in \mathbb{R}^d$ can be computed based thereon as

$$\|\underline{x}\| = \sqrt{\langle \underline{x}, \underline{x} \rangle} \,.$$

Further, two orthogonal vectors have an inner product of zero i.e., $\langle \underline{x}, \underline{y} \rangle = 0$. We can also project any vector $\underline{x} \in \mathbb{R}^d$ to another vector $\underline{y} \in \mathbb{R}^d$, and the projection is given by

$$\underline{x}_\mathsf{p} = \frac{\langle \underline{x}, \underline{y} \rangle}{\langle \underline{y}, \underline{y} \rangle} \, \underline{y} \,.$$

Consequently, we have $\underline{x} - \underline{x}_\mathsf{p}$ is orthogonal to y, namely $\langle x - \underline{x}_\mathsf{p}, \underline{y} \rangle = 0$ (projection theorem). The projection operators can be generalized to infinite dimension of Euclidean space $\mathbb{R}^\infty$ and further to stochastic variables. In this sense, the *fundamental idea underlying linear estimations is to project the estimated quantities onto a plane, which is spanned by the available observations*. More specifically, a general description of a linear model similar to (6.5) can be formulated as

$$y_k = \underline{\psi}_k^\top \underline{\theta} + e_k \,, \tag{7.1}$$

with $\underline{\theta} \in \mathbb{R}^d$ being the parameter vector and $\{y_k\}_{k=1}^N$ the measurements. We can stack measurements and error terms into vector

$$\underline{y} = [\, y_1, y_2, \dots, y_N]^\top \in \mathbb{R}^N \quad \text{and} \quad \underline{e} = [\, e_1, e_2, \dots, e_N\,]^\top \in \mathbb{R}^N \,,$$

respectively, and concatenate the N regression vectors $\underline{\psi}_k \in \mathbb{R}^d$ column-wise as

$$\Psi = \left[\, \underline{\psi}_1, \underline{\psi}_2, ..., \underline{\psi}_N \right]^\top \in \mathbb{R}^{N \times d} \,.$$

leading to the so-called regressor/design matrix given by observed input. Consequently, the point-wise regression in (7.1) can be reformulated into the following compact form

$$\underline{y} = \Psi\underline{\theta} + \underline{e}\,.$$

Our target for linear regression is to minimize the error term

$$\underline{\epsilon} = \underline{y} - \underline{\hat{y}}(\underline{\hat{\theta}}) = \underline{y} - \Psi\underline{\hat{\theta}}\,, \tag{7.2}$$

which can be achieved when $\underline{\epsilon}$ is orthogonal to the hyperplane spanned by the columns of Ψ, i.e., $\langle\Psi, \underline{\epsilon}\rangle = 0$, according to the projection theorem. We now perform the projection to the error term in (7.2) and obtain

$$\langle\Psi, \underline{\epsilon}\rangle = \langle\Psi, y\rangle - \langle\Psi, \Psi\underline{\hat{\theta}}\rangle = \Psi^\top\underline{y} - \Psi^\top\Psi\,\hat{\theta}\,.$$

Setting the expression above to be zero then induces the normal equations in (6.8), and the parameter estimate is given by

$$\underline{\hat{\theta}} = \left(\Psi^\top\Psi\right)^{-1}\Psi^\top\underline{y}\,.$$

Essentially, the ordinary least squares method aims to minimize the projection residual $\underline{\epsilon}$ when projecting $\underline{y}$ to the hyperplane spanned by the columns of regressor matrix Ψ.

7.2 Wiener Filter

The formulation of Wiener Filtering is based on the signal and noise problem in Sec. 4.3.1 with its expression

$$y[k] = s[k] + n[k]\,. \tag{7.3}$$

Suppose that the signal $S[k]$ and the noise $N[k]$ are independent stationary stochastic processes with zero-mean, namely, $\mathrm{E}\{S[k]\} = \mathrm{E}\{N[k]\} = 0$ and $\mathbf{R}_{SN}[k] = 0$. We also assume that their second-order moments $\mathbf{R}_{SS}[k]$ and $\mathbf{R}_{NN}[k]$ are given. The Wiener filter, say $\mathbb{H}(q)$, is a linear filter

$$\hat{s}[k] = \mathbb{H}(q)y[k] = \sum_{n\in\Omega} h[n]y[k-n]\,, \tag{7.4}$$

which minimizes the mean squared error (MSE) of the estimate $\hat{s}[k]$ w.r.t. the true signal $s[k]$, namely, $\mathrm{E}\{(s[k]-\hat{s}[k])^2\}$. Here, $h[k]$ denotes the pulse response of the filter $\mathbb{H}(q)$. According to the setup of the domain Ω, several variants of the Wiener filter can be derived as follows

- $\Omega = \{0, \pm1, \pm2, \dots\} \Rightarrow$ non-causal Wiener filter
- $\Omega = \{0, 1, 2, \dots\} \Rightarrow$ causal Wiener filter for filtering
- $\Omega = \{m, m+1, m+2, \dots\} \Rightarrow$ causal Wiener filter for prediction ($m > 0$)
- $\Omega = \{-m, -m+1, -m+2, \dots, 0, 1, 2, \dots\} \Rightarrow$ causal Wiener filter for smoothing ($m > 0$)

Note that the causality of the filter variants above is determined w.r.t. the estimator $\hat{s}[k-m]$ (at time stamp of estimation). According to the geometric interpretation of ordinary least squares in Sec. 7.1, the filter $\mathbb{H}(q)$ in (7.4) is determined based on

$$\langle s[k] - \hat{s}[k], y[k-\tau]\rangle = 0\,, \forall\tau\in\Omega\,,$$

which minimizes the expected projection residual when projecting the signal to the plane spanned by all available observations $\{y[k-\tau]\}_{\tau\in\Omega}$. This further leads to

$$\begin{aligned}&\mathrm{E}\Big\{\Big(s[k] - \sum_{n\in\Omega} h[n]\,y[k-n]\Big)y[k-\tau]\Big\}\\ =&\mathrm{E}\Big\{s[k]y[k-\tau]\Big\} - \sum_{n\in\Omega} h[n]\,\mathrm{E}\Big\{y[k-n]y[k-\tau]\Big\} = 0\end{aligned}$$

Consequently, we obtain the so-called Wiener-Hopf equation

$$\mathbf{R}_{SY}[\tau] = \sum_{n\in\Omega} h[n]\,\mathbf{R}_{YY}[\tau - n]\,, \quad \forall\tau \in \Omega\,. \tag{7.5}$$

with $\mathbf{R}_{SY}$ denoting correlations between signal and observation, and $\mathbf{R}_{YY}$ the autocorrelation of observations. Designing the filter then boils down to solving the equation (7.5) in different circumstances corresponding to the Wiener filter variants formerly introduced. We show their derivations in the remainder of this section.

7.2.1 Non-Causal Wiener filter

For a non-causal Wiener filter, the convolution sum has infinite domain $\Omega = \mathbb{Z}$, and (7.5) can be specified as

$$\mathbf{R}_{SY}[\tau] = \sum_{n=-\infty}^{\infty} h[n]\,\mathbf{R}_{YY}[\tau - n]\,, \forall\tau \in \mathbb{Z}\,, \tag{7.6}$$

which refers to infinite numbers of equations. However, the right-hand side of (7.6) essentially denotes the convolution of the filter and the autocorrelation of observations. Therefore, the Wiener-Hopf equation can be reformulated into

$$\mathbf{R}_{YY} \star h[\tau] = \mathbf{R}_{SY}[\tau]\,,$$

which we can convert to frequency domain given the definition of spectrum in (3.9). Applying DTFT to both sides of the equation above then leads to

$$\Phi_{YY}(\omega)\mathbb{H}(e^{i\omega}) = \Phi_{SY}(\omega)\,.$$

Here, $\Phi_{SY}(\omega)$ denotes the cross-spectrum between the two processes $S[k]$ and $Y[k]$. Based on (4.4), the equation above can be established in z-domain. Then, we can obtain the transform function as

$$\mathbb{H}(z) = \frac{\Phi_{SY}(z)}{\Phi_{YY}(z)}\,. \tag{7.7}$$

Given the setup of Wiener filtering in (7.3) where the signal and noise processes are independent ($\mathbf{R}_{SN} = 0$), the non-causal Wiener filter can be expressed as

$$\mathbb{H}(z) = \frac{\Phi_{SS}(z)}{\Phi_{SS}(z) + \Phi_{NN}(z)}\,. \tag{7.8}$$

which is a rational, symmetric and real positive function. Thus, the filter can be further implemented base on the spectral factorization in (4.18). As shown in (7.8), the non-causal Wiener filtering introduces the noise spectrum into the denominator for damping the observed signal in certain frequency band where the noise is considerable. We demonstrate this procedure with the example below.

Example 7.2.1. Suppose the following signal and noise problem

$$y[k] = s[k] + n[k]\,, \quad \text{with} \quad s[k] := w[k] \quad \text{and} \quad n[k] := \mathbb{F}_{\mathsf{n}}(q)v[k]\,.$$

Here, $W[k]$ and $V[k]$ are both white noise of variances $\operatorname{var}\{w[k]\} = \frac{1}{2}$ and $\operatorname{var}\{v[k]\} = 1$, respectively. The transfer operator follows $\mathbb{F}_n(q) = 1 + q^{-1}$. We now aim to find the non-causal Wiener filter for $\hat{s}[k]$. The spectrum of $s[k]$ (white noise) expressed in z-domain follows

$$\Phi_{SS}(z) = \frac{1}{2}\,.$$

And we have noise spectrum given by

$$\Phi_{NN}(z) = |\mathbb{F}_{\mathsf{n}}(z)|^2\Phi_{VV}(z) = (1 + z^{-1})(1 + z) = 2 + z + z^{-1}\,.$$

According to (7.8), the transfer function of the non-causal Wiener filter follows

$$\mathbb{H}(z) = \frac{\Phi_{SS}(z)}{\Phi_{SS}(z) + \Phi_{NN}(z)} = \frac{0.5}{2.5 + z + z^{-1}} \,.$$

According to spectral factorization, we can further decompose the function above as

$$\mathbb{H}(z) = \frac{0.5}{2(z+0.5)(z^{-1}+0.5)} = -\frac{1}{6}\frac{z^{-1}}{1+0.5z^{-1}} + \frac{1}{3}\frac{1}{1+0.5z} \tag{7.9}$$

The partial fraction expansion above can also be done using Matlab via function `residue`. Based on the geometric series

$$\sum_{k=0}^{\infty} z^k = \frac{1}{1-z}\,, \quad \text{with} \quad |z| < 1\,,$$

the transfer function can be further expressed as

$$\begin{aligned}\mathbb{H}(z) &= -\frac{1}{6}z^{-1}\sum_{k=0}^{\infty}(-0.5z^{-1})^k + \frac{1}{3}\sum_{k=0}^{\infty}(-0.5z)^k \\ &= -\frac{1}{6}z^{-1}\sum_{k=0}^{\infty}(-0.5)^k z^{-k} + \frac{1}{3}\sum_{k=-\infty}^{0}(-2)^k(z)^{-k}\,,\end{aligned}$$

with region of convergence $|z| \in (0.5, 2)$ for z-transform. Consequently, we can apply the non-causal Wiener filter $\mathbb{H}(q)$ to the observed signal to obtain the signal estimate

$$\begin{aligned}\hat{s}[k] = \mathbb{H}(q)y[k] &= \sum_{n=-\infty}^{\infty} h[n]y[k-n] \\ &= -\frac{1}{6}\sum_{n=0}^{\infty}(-0.5)^n\, y[k-n-1] + \frac{1}{3}\sum_{n=-\infty}^{0}(-2)^n y[k-n] \\ &= -\frac{1}{6}\sum_{n=0}^{\infty}(-0.5)^n\, y[k-n-1] + \frac{1}{3}\sum_{n=0}^{\infty}(-0.5)^n y[k+n]\,.\end{aligned}$$

Note that the first term above is causal stable, and the second term anti-causal stable. For implementation, one can apply `filtfilt` in Matlab to get the non-causal filter done once the spectrum is factorized in (7.9).

7.2.2 FIR Wiener filter

Given the basic filtering equation in (7.4), we design the Wiener filter to be a finite impulse response (FIR) over $\Omega = \{0, \ldots, N-1\}$. The resulted FIR Wiener filter has the following transfer function

$$\mathbb{H}(z) = \sum_{k=0}^{N-1} h[k]z^{-k}\,.$$

The Wiener-Hopf equation in (7.5) can then be specified as

$$\mathbf{R}_{SY}[\tau] = \sum_{n=0}^{N-1} h[n]\mathbf{R}_{YY}[\tau - n]\,, \quad \text{with} \quad \tau = 0, ..., N-1\,,$$

leading to the following finite and quadratic system of equations

$$\begin{bmatrix} \mathbf{R}_{SY}[0] \\ \mathbf{R}_{SY}[1] \\ \vdots \\ \mathbf{R}_{SY}[N-1] \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{YY}[0] & \mathbf{R}_{YY}[-1] & \dots & \mathbf{R}_{YY}[-N+1] \\ \mathbf{R}_{YY}[1] & \mathbf{R}_{YY}[0] & \dots & \mathbf{R}_{YY}[-N+2] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{YY}[N-1] & \mathbf{R}_{YY}[N-2] & \dots & \mathbf{R}_{YY}[0] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[N-1] \end{bmatrix} . \qquad (7.10)$$

The setup of Wiener filtering facilitates computing the correlation terms above. Given that the signal and noise are uncorrelated and their spectra $\Phi_{SS}(\omega)$ and $\Phi_{NN}(\omega)$ are available, one can obtain the cross-correlation between the signal and observation with $\mathbf{R}_{SY}[\tau] = \text{IDTFT}\{\Phi_{SS}(\omega)\}$, and the autocorrelation of the observed signal $R_{YY}[\tau] = \text{IDTFT}\{\Phi_{SS}(\omega)\} + \text{IDTFT}\{\Phi_{NN}(\omega)\}$. Solving a linear system in (7.10) denoted as $\underline{b} = \mathbf{A}\underline{h}$ can be done via $\underline{h} = \mathbf{A}^{-1}\underline{b}$ (suppose matrix $\mathbf{A}$ full rank).

7.2.3 Causal Wiener filter

We now focus on causal Wiener filtering for filtering, prediction and smoothing.

Filtering Given the general description of Wiener filter in (7.4), its causal variant for filtering is expressed as

$$\hat{s}[k] = \sum_{n=0}^{\infty} h[n]y[k-n] , \qquad (7.11)$$

and its Wiener-Hopf equation follows

$$\mathbf{R}_{SY}[\tau] = \sum_{n=0}^{\infty} h[n]\mathbf{R}_{YY}[\tau-n] , \quad \text{with} \quad \tau \geq 0 . \qquad (7.12)$$

According to the spectral factorization theorem in (4.18) and Wold's decomposition theorem, we have

$$\Phi_{YY}(z) = |\mathbb{T}(z)|^2\Phi_{EE}(z) = \mathbb{T}(z)\mathbb{T}(z^{-1})\Phi_{EE}(z) , \qquad (7.13)$$

where $\mathbb{T}(z)$ have all zeros and poles inside the unit circle, and $e[k]$ is given by a white noise with its variance $\text{var}\{e[k]\} = \sigma_e^2$. Therefore, we can exploit $1/\mathbb{T}(q)$ as a whitening filter (stable and causal), such that

$$e[k] = \frac{1}{\mathbb{T}(q)}y[k] .$$

Filtering the whitened output via the Wiener-Hopf equation in (7.12) then boils down to

$$\begin{aligned} \mathbf{R}_{SE}[\tau] &= \sum_{n=0}^{\infty} h_e^{\text{c}}[n]\mathbf{R}_{EE}[\tau-n] \\ &= \sigma_e^2 \sum_{n=0}^{\infty} h_e^{\text{c}}[n]\delta[\tau-n] \\ &= \sigma_e^2 h_e^{\text{c}}[\tau] , \quad \forall \tau \geq 0 \end{aligned}$$

Here, $h_e^{\text{c}}[k]$ denotes the causal Wiener filter applied to the white noise $e[k]$. For a non-causal Wiener filter filtering the same white noise, its Wiener-Kopf equation introduced in (7.6) follows

$$\mathbf{R}_{SE}[\tau] = \sum_{n=-\infty}^{\infty} h_e^{\text{nc}}[n]\mathbf{R}_{EE}[\tau-n] = \sigma_e^2 h_e^{\text{nc}}[\tau] , \quad \forall \tau \in \mathbb{Z} ,$$

with $h_e^{\text{nc}}[k]$ denoting the pulse response of the non-causal Wiener filter. Therefore, given whitened output $e[k] = 1/\mathbb{T}(q)y[k]$, we can derive a causal Wiener filter $h_e^{\text{c}}[k]$ by truncating its non-causal variant $h_e^{\text{nc}}[k]$ to positive indices according to

$$h_{\mathrm{e}}^{\mathsf{c}}[k] = \left[\, h_{\mathrm{e}}^{\mathsf{nc}}[k] \,\right]_+ := \begin{cases} h_{\mathrm{e}}^{\mathsf{nc}}[k]\,, & k \geq 0 \\ 0\,, & k < 0 \end{cases} . \tag{7.14}$$

Further, we can represent the truncated non-causal filter in z-domain with the transfer function

$$[\mathbb{H}_{\mathrm{e}}(z)]_+ := \sum_{k=0}^{\infty} h_{\mathrm{e}}^{\mathsf{nc}}[k] z^{-k}\,. \tag{7.15}$$

The operator $[\cdot]_+$ above can also be directly implemented for frequency models via partial fraction expansion and picking the causal part (also stable, with poles inside the unit circle). The cascading of whitening and truncation above then induces the causal Wiener filter

$$\mathbb{H}(z) = \frac{1}{\mathbb{T}(z)} [\mathbb{H}_{\mathrm{e}}(z)]_+\,. \tag{7.16}$$

Here, $\mathbb{H}_{\mathrm{e}}(z)$ denotes the transfer function of the non-causal Wiener filter in (7.14), which is given by

$$\mathbb{H}_{\mathrm{e}}(z) = \frac{\Phi_{SY}(z)}{\sigma_{\mathrm{e}}^2\, \mathbb{T}(z^{-1})}\,. \tag{7.17}$$

We now provide the following proof to the expression (7.17).

Proof. Notice that cascading the whitening $1/\mathbb{T}(z)$ and the follow-up non-causal filtering $\mathbb{H}_{\mathrm{e}}(z)$ essentially yields the non-causal Wiener filtering of signal $y[k]$ given by (7.7). Mathematically, this relation is formulated as

$$\frac{1}{\mathbb{T}(z)} \mathbb{H}_{\mathrm{e}}(z) = \frac{\Phi_{SY}(z)}{\Phi_{YY}(z)}\,.$$

Given the factorization to the observed signal spectrum $\Phi_{YY}(z)$ in (7.13), the equation above can be further derived as follows

$$\begin{aligned} \mathbb{H}_{\mathrm{e}}(z) &= \frac{\Phi_{SY}(z)}{\Phi_{YY}(z)} \mathbb{T}(z) \\ &= \frac{\Phi_{SY}(z)}{\mathbb{T}(z)\mathbb{T}(z^{-1})\Phi_{EE}(z)} \mathbb{T}(z) \\ &= \frac{\Phi_{SY}(z)}{\sigma_{\mathrm{e}}^2\, \mathbb{T}(z^{-1})}\,. \end{aligned}$$

Thus, the expression in (7.17) is valid. ■

Prediction and smoothing We can exploit the causal Wiener filter for prediction and smoothing. For that, we construct a “delayed” signal of m steps via the shift operator q, namely,

$$\bar{s}[k] = s[k+m] = q^m s[k]\,.$$

Given observed signal $y[k]$, we now apply the causal Wiener filtering given by (7.16) to the delayed signal and obtain its estimate via

$$\hat{\bar{s}}[k] = \bar{\mathbb{H}}(q) y[k] = \frac{1}{\mathbb{T}(q)} \left[\frac{\Phi_{\bar{S}Y}(q)}{\sigma_{\mathrm{e}}^2\, \mathbb{T}(q^{-1})} \right]_+ y[k]\,,$$

with the cross-spectrum between delayed signal and output derived as follows

$$\Phi_{\bar{S}Y}(q) = q^m \Phi_{SY}(q)\,.$$

To obtain the predicted/smoothed estimate, we apply the reverse delay q^{-m} to the delayed signal estimate. In summary, the m-step predictor ($m > 0$) or smoother ($m < 0$) given by causal Wiener filter follow

$$\hat{s}[k+m|k] = \frac{1}{q^m \mathbb{T}(q)} \left[\frac{q^m \Phi_{SY}(q)}{\sigma_{\mathrm{e}}^2 \, \mathbb{T}(q^{-1})} \right]_+ y[k] \, .$$

7.2.4 Residual variance

We now aim to quantify the uncertainty of the estimate given by Wiener filtering. Given estimates $\hat{s}[k]$ given by (7.4), we compute the mean squared error

$$\begin{aligned} \operatorname{var}\big\{ s[k] - \hat{s}[k] \big\} &= \mathrm{E}\big\{ (s[k] - \hat{s}[k])^2 \big\} \\ &= \mathrm{E}\big\{ (s[k] - \hat{s}[k]) s[k] \big\} - \mathrm{E}\big\{ (s[k] - \hat{s}[k]) \hat{s}[k] \big\} \, . \end{aligned}$$

According to the projection theorem, the second term on the right-hand side of the formula above goes to zero. Therefore, we obtain

$$\begin{aligned} \operatorname{var}\big\{ s[k] - \hat{s}[k] \big\} &= \mathrm{E}\big\{ (s[k] - \hat{s}[k]) s[k] \big\} \\ &= \mathrm{E}\big\{ s[k] s[k] \big\} - \mathrm{E}\Bigg\{ \bigg(\sum_{n \in \Omega} h[n] y[k-n] \bigg) s[k] \Bigg\} \\ &= \mathbf{R}_{SS}[0] - \sum_{n \in \Omega} h[n] \mathrm{E}\big\{ y[k-n] s[k] \big\} \\ &= \mathbf{R}_{SS}[0] - \sum_{n \in \Omega} h[n] \mathbf{R}_{SY}[n] \, . \end{aligned}$$

The first term denotes the energy in signal $s[k]$, and the second term indicates the amount of information about signal $s[k]$ in the observation $y[k]$.

Chapter 8

Kalman Filtering

The Wiener filters are theoretically sound and fulfills major signal estimation tasks including filtering, prediction, and smoothing. However, their applications are restricted to time-invariant and scalar signal models. They assume measurements from $-\infty$ and cannot handle transient during start-up. Moreover, there has been a lack of modeling and computational tools to deploy them in practice. Being set up based on state-space models, the Kalman filter can be implemented more closely to the physical system (instead of spectra). Also, they require less computational complexity and can handle multi-variate and time-variant signal models. Since its successful deployment in Project Apollo, it has gained extensive applications in science and engineering.

8.1 Problem Formulation

We exploit the state-space model for system representation in the setup of Kalman filter. A process model is introduced to describe the state dynamics in the form of the following difference equation

$$\underline{x}_{t+1} = \mathbf{A}\underline{x}_t + \underline{w}_t \,, \tag{8.1}$$

with $\underline{x}_t$ and $\underline{x}_{t+1}$ being the state vectors at time steps t and $t+1$, respectively, and matrix $\mathbf{A}$ denoting the state transition. $\underline{w}_t$ denotes the process noise which is a zero-mean white noise of covariance $\mathrm{cov}\{\underline{w}_t\} = \mathrm{E}\{\underline{w}_t\underline{w}_t^\top\} = \mathbf{Q}$. Further, the system state is observed according to the following measurement model

$$\underline{y}_t = \mathbf{C}\underline{x}_t + \underline{v}_t \,, \tag{8.2}$$

with $\underline{y}_t$ being the measurement and matrix $\mathbf{C}$ denoting the observation. $\underline{v}_t$ denotes the measurement noise that is a white noise of covariance $\mathrm{cov}\{\underline{v}_t\} = \mathrm{E}\{\underline{v}_t\underline{v}_t^\top\} = \mathbf{R}$. We further assume the initial value of the system state to be a random variable of expectation $\mathrm{E}\{\underline{x}_{t=0}\} = \underline{x}_0$ and covariance $\mathrm{cov}\{\underline{x}_{t=0}\} = \mathrm{E}\{(\underline{x}_{t=0}-\underline{x}_0)(\underline{x}_{t=0}-\underline{x}_0)^\top\} = \mathbf{P}_0$. In practice, we typically (however, not necessarily) assume all stochastic variables in the setup above, namely, $\underline{w}_t$, $\underline{v}_t$, $\underline{x}_{t=0}$ to be Gaussian-distributed. In this case, a Kalman filter delivers estimates with Gaussian-distributed conditional distribution given available measurements (shown later in Sec. 8.3.1).

Note that the state space model introduced for Kalman filtering in (8.1) and (8.2) is in general non-stationary. And matrices denoting transition and observation, i.e., $\mathbf{A}$ and $\mathbf{C}$, respectively, can be time-varying. Moreover, such a setup can be easily extended when a known control input $\underline{u}_t$ exists. For that, we can incorporate the control vector $\underline{u}_t$ into (8.1) via $\underline{x}_{t+1} = \mathbf{A}\underline{x}_t + \mathbf{B}_t\underline{u}_t + \underline{w}_t$ with matrix $\mathbf{B}$ denoting control-input model that is also time-varying in general.

Given a set of measurements $\mathbb{Y}_s = \{\underline{y}_i\}_{i=1}^{s}$ from time step 1 to s from a system set up above, applying Kalman filtering at time t delivers the best linear estimate of the state

$$\hat{\underline{x}}_{t|s} = \mathrm{E}\{\underline{x}_t|\mathbb{Y}_s\} \tag{8.3}$$

in the sense of minimal estimation error variance

$$\mathrm{E}\{\|\hat{\underline{x}}_{t|s} - \underline{x}_t\|^2\}\,. \tag{8.4}$$

W.r.t. the estimation time step t, we have three variants of the Kalman filter with (1) $t = s$ for filtering, (2) $t > s$ for prediction, and (3) $t < s$ for smoothing.

8.2 Derivation

We first consider a linear state observer that combines information about system dynamics and external measurements to estimate the true system state. Given the setup shown in Sec. 8.1, this can be in general formulated into the following linear form with continuous-time variables

$$\dot{\hat{\underline{x}}} = \mathbf{A}\hat{\underline{x}} + \mathbf{K}(\underline{y} - \mathbf{C}\hat{\underline{x}})\,. \tag{8.5}$$

Here, $\hat{\underline{x}}$ denotes the estimate incorporating information from all previous measurements. The innovation term $\underline{y} - \mathbf{C}\hat{\underline{x}}$ contains information in the present current $\underline{y}$ with $\mathbf{K}$ controlling how much it contributes to the current estimate.

The Kalman filter is a recursive estimator following the concept above. Suppose a previous posterior estimate at time step $t-1$ is available with its mean $\hat{\underline{x}}_{t-1|t-1}$ and covariance $\mathbf{P}_{t-1|t-1}$ [1]. Given the measurement $\underline{y}_t$ at time step t, we aim to obtain the current posterior estimate characterized by its mean $\hat{\underline{x}}_{t|t}$ and covariance $\mathbf{P}_{t|t}$. For that, the Kalman filtering consists of two steps, the time update and measurement update, which are associated with the state-space model formulated in (8.1) and (8.2), respectively. In the remainder of this section, we provide a derivation for the two steps of Kalman filtering.

Time update Given the posterior estimate $\underline{x}_t|\mathbb{Y}_t$ of mean $\hat{\underline{x}}_{t|t}$ and covariance $\mathbf{P}_{t|t}$ at time step t , we aim to predict the estimate at time step $t+1$ with its mean $\hat{\underline{x}}_{t+1|t}$ and its covariance $\mathbf{P}_{t+1|t}$ based on the linear process model in (8.1). According to (8.3), the mean of the predicted estimate follows

$$\begin{aligned}\hat{\underline{x}}_{t+1|t} &= \mathrm{E}\big\{\underline{x}_{t+1}|\mathbb{Y}_t\big\} = \mathrm{E}\big\{\mathbf{A}\underline{x}_t + \underline{w}_t|\mathbb{Y}_t\big\} \\ &= \mathbf{A}\mathrm{E}\big\{\underline{x}_t|\mathbb{Y}_t\big\} + \mathrm{E}\{\underline{w}_t|\mathbb{Y}_t\} \\ &= \mathbf{A}\hat{\underline{x}}_{t|t}\,,\end{aligned} \tag{8.6}$$

given $\mathrm{E}\{\underline{w}_t|\mathbb{Y}_t\} = \mathrm{E}\{\underline{w}_t\} = \underline{0}$ due to zero-mean white process noise. To characterize the uncertainty of the predicted estimate, we compute its covariance matrix

$$\mathbf{P}_{t+1|t} = \mathrm{E}\big\{\tilde{\underline{x}}_{t+1|t}\tilde{\underline{x}}_{t+1|t}^\top\big\} \tag{8.7}$$

with $\tilde{\underline{x}}_{t+1|t}$ denoting the prediction error given by

$$\begin{aligned}\tilde{\underline{x}}_{t+1|t} &= \underline{x}_{t+1|t} - \hat{\underline{x}}_{t+1|t} \\ &= \mathbf{A}\underline{x}_t + \underline{w}_t - \mathbf{A}\hat{\underline{x}}_{t|t} \\ &= \mathbf{A}(\underline{x}_t - \hat{\underline{x}}_{t|t}) + \underline{w}_t \\ &= \mathbf{A}\tilde{\underline{x}}_{t|t} + \underline{w}_t\,.\end{aligned} \tag{8.8}$$

Here, $\tilde{\underline{x}}_{t|t}$ is the estimation error of the previous posterior estimate $\underline{x}_t|\mathbb{Y}_t$. It is trivial to verify that $\tilde{\underline{x}}_{t|t}$ and the process noise $\underline{w}_t$ are independent of each other. Thus, the covariance of predicted estimate in (8.7)

[1] Or initialized estimate characterized by the mean $\underline{x}_0$ and the covariance $\mathbf{P}_0$.

can be further derived as

$$\begin{aligned}\mathbf{P}_{t+1|t} &= \mathrm{E}\big\{(\mathbf{A}\tilde{\underline{x}}_{t|t} + \underline{w}_t)(\mathbf{A}\tilde{\underline{x}}_{t|t} + \underline{w}_t)^\top\big\} \\ &= \mathrm{E}\big\{\mathbf{A}\tilde{\underline{x}}_{t|t}\tilde{\underline{x}}_{t|t}^\top\mathbf{A}^\top + \underline{w}_t\underline{w}_t^\top\big\} \\ &= \mathbf{A}\mathrm{E}\big\{\tilde{\underline{x}}_{t|t}\tilde{\underline{x}}_{t|t}^\top\big\}\mathbf{A}^\top + \mathrm{E}\{\underline{w}_t\underline{w}_t^\top\} \\ &= \mathbf{A}\mathbf{P}_{t|t}\mathbf{A}^\top + \mathbf{Q}\,.\end{aligned} \tag{8.9}$$

Measurement update Given the predicted estimate $\underline{x}_t|\mathbb{Y}_{t-1}$ of mean $\hat{\underline{x}}_{t|t-1}$ and covariance $\mathbf{P}_{t|t-1}$ from time update (or initialization), we now update the prior by incorporating the current measurement $\underline{y}_t$ to obtain the posterior estimate $\underline{x}_t|\mathbb{Y}_t$. For derivation, We start off from the concept of linear observer in (8.5). Given the discrete-time measurement model in (8.2), we have the innovation term w.r.t. the prior given by

$$\tilde{\underline{y}}_t = \underline{y}_t - \mathbf{C}\hat{\underline{x}}_{t|t-1}\,. \tag{8.10}$$

And the posterior mean is obtained via a linear combination of the prior mean and the innovation, i.e.,

$$\hat{\underline{x}}_{t|t} = \hat{\underline{x}}_{t|t-1} + \mathbf{K}_t\tilde{\underline{y}}_t = \hat{\underline{x}}_{t|t-1} + \mathbf{K}_t(\underline{y}_t - \mathbf{C}\hat{\underline{x}}_{t|t-1})\,. \tag{8.11}$$

It is then important to determine the optimal gain matrix $\mathbf{K}_t$ in the sense of minimal variance of estimation error as shown in (8.4). Based on the measurement model in (8.2), we first compute the resulted estimation error for the posterior mean $\hat{\underline{x}}_{t|t}$ in (8.11) as follows

$$\begin{aligned}\tilde{\underline{x}}_{t|t} &= \underline{x}_t - \hat{\underline{x}}_{t|t} = \underline{x}_t - \hat{\underline{x}}_{t|t-1} - \mathbf{K}_t(\underline{y}_t - \mathbf{C}\hat{\underline{x}}_{t|t-1}) \\ &= \underline{x}_t - \hat{\underline{x}}_{t|t-1} - \mathbf{K}_t(\mathbf{C}\underline{x}_t + \underline{v}_t - \mathbf{C}\hat{\underline{x}}_{t|t-1}) \\ &= (\mathbf{I} - \mathbf{K}_t\mathbf{C})(\underline{x}_t - \hat{\underline{x}}_{t|t-1}) - \mathbf{K}_t\underline{v}_t \\ &= (\mathbf{I} - \mathbf{K}_t\mathbf{C})\tilde{\underline{x}}_{t|t-1} - \mathbf{K}_t\underline{v}_t\,.\end{aligned}$$

Here, $\underline{x}_t$ denotes the true value of the state at time step t. $\tilde{\underline{x}}_{t|t-1}$ is the prediction error given by (8.8). In order to minimize the variance of estimation error, we first derive the covariance matrix of the posterior estimate, namely,

$$\begin{aligned}\mathbf{P}_{t|t} &= \mathrm{E}\big\{\tilde{\underline{x}}_{t|t}\tilde{\underline{x}}_{t|t}^\top\big\} \\ &= \mathrm{E}\big\{\big((\mathbf{I} - \mathbf{K}_t\mathbf{C})\tilde{\underline{x}}_{t|t-1} - \mathbf{K}_t\underline{v}_t\big)\big(\tilde{\underline{x}}_{t|t-1}^\top(\mathbf{I} - \mathbf{K}_t\mathbf{C})^\top - \mathbf{K}_t\underline{v}_t\big)\big\}\,.\end{aligned}$$

Notice that the predicted estimate $\hat{\underline{x}}_{t|t-1}$ is independent from the measurement noise $\underline{v}_t$, we further obtain

$$\begin{aligned}\mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathrm{E}\{\tilde{\underline{x}}_{t|t-1}\tilde{\underline{x}}_{t|t-1}^\top\}(\mathbf{I} - \mathbf{K}_t\mathbf{C})^\top + \mathbf{K}_t\mathrm{E}\{\underline{v}_t\underline{v}_t^\top\}\mathbf{K}_t^\top \\ &= (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t|t-1}(\mathbf{I} - \mathbf{K}_t\mathbf{C})^\top + \mathbf{K}_t\mathbf{R}\mathbf{K}_t^\top \\ &= \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{C}^\top\mathbf{K}_t^\top - \mathbf{K}_t\mathbf{C}\mathbf{P}_{t|t-1} + \mathbf{K}_t\mathbf{C}\mathbf{P}_{t|t-1}\mathbf{C}^\top\mathbf{K}_t^\top + \mathbf{K}_t\mathbf{R}\mathbf{K}_t^\top \\ &= \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{C}^\top\mathbf{K}_t^\top - \mathbf{K}_t\mathbf{C}\mathbf{P}_{t|t-1} + \mathbf{K}_t(\mathbf{C}\mathbf{P}_{t|t-1}\mathbf{C}^\top + \mathbf{R})\mathbf{K}_t^\top\,.\end{aligned} \tag{8.12}$$

The variance of the estimation error is equal to the trace of the posterior covariance above, i.e.,

$$\mathrm{E}\big\{\|\hat{\underline{x}}_{t|t} - \underline{x}_t\|^2\big\} = \mathrm{E}\big\{\tilde{\underline{x}}_{t|t}^\top\tilde{\underline{x}}_{t|t}\big\} = \mathrm{tr}\big\{\mathbf{P}_{t|t}\big\}\,.$$

Therefore, we can minimize it by setting its first derivative to be zero via[2]

$$\frac{\mathrm{d}\,\mathrm{tr}\{\mathbf{P}_{t|t}\}}{\mathrm{d}\mathbf{K}_t} = -2\mathbf{P}_{t|t-1}\mathbf{C}^\top + 2\mathbf{K}_t(\mathbf{C}\mathbf{P}_{t|t-1}\mathbf{C}^\top + \mathbf{R}) = \mathbf{0}\,.$$

[2] Derivatives of matrix functions can be obtained using basic matrix calculus. Here is a link for reference.

Therefore, the optimal Kalman gain follows

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{t|t-1}\mathbf{C}^\top + \mathbf{R})^{-1} . \tag{8.13}$$

Consequently, the mean of the posterior estimate is given by

$$\hat{\underline{x}}_{t|t} = \hat{\underline{x}}_{t|t-1} + \mathbf{K}_t(\underline{y}_t - \mathbf{C}\hat{\underline{x}}_{t|t-1}) . \tag{8.14}$$

We further multiply the term $(\mathbf{C}\mathbf{P}_{t|t-1}\mathbf{C}^\top + \mathbf{R})\mathbf{K}_t^\top$ on both sides of (8.13), leading to

$$\mathbf{K}_t(\mathbf{C}\mathbf{P}_{t|t-1}\mathbf{C}^\top + \mathbf{R})\mathbf{K}_t^\top = \mathbf{P}_{t|t-1}\mathbf{C}^\top\mathbf{K}_t^\top$$

And the covariance matrix in (8.12) can be further derived as

$$\begin{aligned}\mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{C}^\top\mathbf{K}_t^\top - \mathbf{K}_t\mathbf{C}\mathbf{P}_{t|t-1} + \mathbf{P}_{t|t-1}\mathbf{C}^\top\mathbf{K}_t^\top \\ &= \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{C}\mathbf{P}_{t|t-1} \\ &= (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t|t-1} .\end{aligned} \tag{8.15}$$

8.3 Properties

We now introduce several important properties of Kalman filters based on their derivation in the previous section.

8.3.1 Optimality

In general, the Kalman filter is the linear estimator with the smallest mean squared error as derived above. If the initial state $\underline{x}_0$, process noise $\underline{w}_t$ and the measurement noise $\underline{v}_t$ are Gaussian-distributed, then the conditional distribution of the states given available measurements (e.g., $\{\underline{y}_i\}_{i=1}^t$) are also Gaussians according to the Kalman filter, namely,

$$\begin{aligned}\underline{x}_{t+1}\big|\{\underline{y}_i\}_{i=1}^t &\sim \mathcal{N}\big(\hat{\underline{x}}_{t|t+1}, \mathbf{P}_{t+1|t}\big) , \\ \underline{x}_t\big|\{\underline{y}_i\}_{i=1}^t &\sim \mathcal{N}\big(\hat{\underline{x}}_{t|t}, \mathbf{P}_{t|t}\big) .\end{aligned}$$

8.3.2 Observability

We consider the following noise-free state-space model of a linear time-variant (LTV) system

$$\begin{aligned}\underline{x}_{t+1} &= \mathbf{A}_t\underline{x}_t , \\ \underline{y}_t &= \mathbf{C}_t\underline{x}_t .\end{aligned} \tag{8.16}$$

for a d-dimensional state $\underline{x}_t \in \mathbb{R}^d$. Given an initial state $\underline{x}_0$, the measurements over N time steps can be computed according to

$$\underline{y}_t = \mathbf{C}_t\prod_{i=0}^{t-1}\mathbf{A}_i\underline{x}_0 , \quad \text{for} \quad t = 0, \dots, N . \tag{8.17}$$

By stacking the set of measurements $\{\underline{y}_t\}_{t=0}^N$ into a vector, we can express the $N+1$ equations above in the following compact form

$$\begin{bmatrix}\underline{y}_0 \\ \underline{y}_1 \\ \underline{y}_2 \\ \vdots \\ \underline{y}_N\end{bmatrix} = \begin{bmatrix}\mathbf{C}_0 \\ \mathbf{C}_1\mathbf{A}_0 \\ \mathbf{C}_2\mathbf{A}_1\mathbf{A}_0 \\ \vdots \\ \mathbf{C}_N\prod_{i=0}^{N-1}\mathbf{A}_i\end{bmatrix}\underline{x}_0 =: \mathbf{O}\underline{x}_0 . \tag{8.18}$$

with $\mathbf{O}$ being the so-called observability matrix. A system is observable if and only if that the initial state x_0 can be determined according to a noise-free system dynamics, e.g., (8.16), after a finite number of steps. As shown in (8.18), this requirement is equal to that the observability matrix $\mathbf{O}$ is of full column rank of d, which intuitively guarantees that the system of equations in (8.17) delivers a unique solution of $\underline{x}_0$ given a finite length of measurements.

For an observable system, the current state can be estimated using only the information from measurements given by the sensors for all possible evolutions of the state. And the Kalman filter is guaranteed to converge to a steady-state values when $t \to \infty$. Thus, before trying to estimate $\underline{x}_t$ using the measurements, it is important to investigate observability. If a system is not observable, extra sensor(s) would be desired to make it observable.

8.4 Stationary Kalman Filter

By substituting the predicted prior covariance $\mathbf{P}_{t|t-1}$ in (8.9) with its expression in (8.15), we can establish the relation between two consecutive posterior covariance estimates as follows

$$\begin{aligned}\mathbf{P}_{t+1|t} &= \mathbf{A}(\mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{C}\mathbf{P}_{t|t-1})\mathbf{A}^\top + \mathbf{Q} \\ &= \mathbf{A}\big(\mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{t|t-1}\mathbf{C}^\top + \mathbf{R})^{-1}\mathbf{C}\mathbf{P}_{t|t-1}\big)\mathbf{A}^\top + \mathbf{Q}\,.\end{aligned} \tag{8.19}$$

If the system is time-invariant and observable and we apply Kalman filtering to it with $t \to \infty$, $\mathbf{P}_{t|t-1}$ and $\mathbf{P}_{t+1|t}$ will converge to the stationary covariance $\bar{\mathbf{P}}$. According to (8.19), the stationary covariance matrix $\bar{\mathbf{P}}$ is the solution to the stationary Riccati equation

$$\bar{\mathbf{P}} = \mathbf{A}\big(\bar{\mathbf{P}} - \bar{\mathbf{P}}\mathbf{C}^\top(\mathbf{C}\bar{\mathbf{P}}\mathbf{C}^\top + \mathbf{R})^{-1}\mathbf{C}\bar{\mathbf{P}}\big)\mathbf{A}^\top + \mathbf{Q}\,.$$

The equation above is in general nonlinear w.r.t. $\bar{\mathbf{P}}$ and can be solved numerically (e.g., by simply iterating the equation until convergence). There may also exist closed-form solution in certain simplified cases. Consequently, we obtain the stationary/asymptotic Kalman gain according to (8.13) with

$$\bar{\mathbf{K}} = \bar{\mathbf{P}}\mathbf{C}^\top(\mathbf{C}\bar{\mathbf{P}}\mathbf{C}^\top + \mathbf{R})^{-1}\,.$$

The mean of the posterior estimate given by a stationary Kalman filter can be then expressed as

$$\begin{aligned}\hat{\underline{x}}_{t|t} &= \hat{\underline{x}}_{t|t-1} + \bar{\mathbf{K}}(\underline{y}_t - \mathbf{C}\hat{\underline{x}}_{t|t-1}) \\ &= \mathbf{A}\hat{\underline{x}}_{t-1|t-1} + \bar{\mathbf{K}}(\underline{y}_t - \mathbf{C}\mathbf{A}\hat{\underline{x}}_{t-1|t-1}) \\ &= (\mathbf{A} - \bar{\mathbf{K}}\mathbf{C}\mathbf{A})\hat{\underline{x}}_{t-1|t-1} + \bar{\mathbf{K}}\underline{y}_t\,.\end{aligned} \tag{8.20}$$

Frequency property (8.20) corresponds to the linear filter

$$\mathbb{H}_{\mathsf{KF}}(z) = \frac{\hat{\mathbb{X}}(z)}{\mathbb{Y}(z)} = \frac{\bar{\mathbf{K}}}{\mathbf{I} - (\mathbf{A} - \bar{\mathbf{K}}\mathbf{C}\mathbf{A})z^{-1}}\,. \tag{8.21}$$

Therefore, the stationary Kalman filter is a frequency selective filter.

8.5 Kalman Filter in Practice

We now demonstrate the workflow of Kalman filtering with common examples in navigation and tracking. In this regard, the state of interest typically refers to motion information. For instance, for tracking an object moving in a one-dimensional space, we can model the system dynamics with a state incorporating the position z and velocity $\dot{z}$, i.e.,

$$\underline{x}_t = [\,z_t, \dot{z}_t\,]^\top\,.$$

Suppose that the object is driven by an random force F_t that is constant from time steps t to $t+1$ over interval T in the positive direction of the coordinate. We further assume that the force is zero-mean

Gaussian-distributed, namely, $F_t \sim \mathcal{N}(0, \sigma_F^2)$. According to the Newton's law of motion, the dynamics of the object can be modeled with the following difference equation

$$\underline{x}_{t+1} = \begin{bmatrix} z_{t+1} \\ \dot{z}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_t \\ \dot{z}_t \end{bmatrix} + \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} \frac{F_t}{m} := \mathbf{A}\underline{x}_t + \underline{w}_t \,, \tag{8.22}$$

with m denoting the mass of the object. Matrix $\mathbf{A}$ above denotes the system transition as given in (8.1). We further reformulate the process noise as

$$\underline{w}_t := \mathbf{B}F_t \,, \quad \text{with} \quad \mathbf{B} = \begin{bmatrix} 0.5T^2/m \\ T/m \end{bmatrix} , \tag{8.23}$$

which is an affine transformation of the one-dimensional Gaussian distribution $\mathcal{N}(0, \sigma_F^2)$ It is straightforward to verify that the process noise $\underline{w}_t$ is also zero-mean Gaussian-distributed with covariance given by

$$\mathbf{Q} = \mathbf{B}\mathbf{B}^\top \sigma_F^2 = \begin{bmatrix} 0.25T^4/m^2 & 0.5T^3/m^2 \\ 0.5T^3/m^2 & T^2/m^2 \end{bmatrix} \sigma_F^2 \,. \tag{8.24}$$

We can then perform time update according to (8.6) and (8.9) to obtain the mean and covariance of the predicted estimate, respectively.

As for the measurement model, its form is in general sensor-dependent. For instance, we can assume that the position of the object is measured by a GPS-like sensor with standard deviation of 5 meters. This leads to a measurement model

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \underline{x}_t + v(t) \,, \quad \text{with} \quad v_t \sim \mathcal{N}(0, 25) \,.$$

For some sensors such as radars or LiDARs, the measurement consists of bearing and range. In this case, the raw measurement can be often converted into the standard form given by (8.2). Another strategy is to directly exploit a nonlinear observation function $h(\underline{x}_t)$ in the measurement model with

$$\underline{y}_t = h(\underline{x}_t) + \underline{v}_t \,. \tag{8.25}$$

For performing measurement update step, we can linearize the model above at the prior to deliver the posterior estimate as given in (8.14) and (8.15).

For bootstrapping Kalman filtering, we can initialize the state $\underline{x}_0$ and its uncertainty $\mathbf{P}_0$ according to the prior knowledge. The process and measurement noise covariances, $\mathbf{Q}$ and $\mathbf{R}$, are typically determined according to the actual physical systems and sensors. For evaluating the estimation performance, we often perform several Monte-Carlo runs of the sequence and compute the mean of the estimation error.

8.6 Extended Kalman Filter (EKF)

As shown in (8.25), it is very common that the system or measurement models are nonlinear equations. In this case, we can formulate them in the following general form

$$\begin{aligned} \underline{x}_{t+1} &= f(\underline{x}_t) + \underline{w}_t \,, \\ \underline{y}_t &= h(\underline{x}_t) + \underline{v}_t \,, \end{aligned} \tag{8.26}$$

with nonlinear functions $f(\cdot)$ and $h(\cdot)$ denoting the transition and observation, respectively. The process and measurement noises, $\underline{w}_t$ and $\underline{v}_t$, are zero-mean white noise of covariances $\mathbf{Q}$ and $\mathbf{R}$, respectively. The standard Kalman filter is not directly applicable. The extended Kalman filter linearizes $f(\cdot)$ and $h(\cdot)$ around the estimates corresponding to the time update and measurement update steps, namely,

$$\begin{aligned} f(\underline{x}_t) &\approx f(\hat{\underline{x}}_{t|t}) + \mathbf{F}_t(\underline{x}_t - \hat{\underline{x}}_{t|t}) \,, \\ h(\underline{x}_t) &\approx h(\hat{\underline{x}}_{t|t-1}) + \mathbf{H}_t(\underline{x}_t - \hat{\underline{x}}_{t|t-1}) \,, \end{aligned}$$

with

$$\mathbf{F}_t = \frac{\partial f}{\partial \underline{x}}\bigg|_{\underline{\hat{x}}_{t|t}} \quad \text{and} \quad \mathbf{H}_t = \frac{\partial h}{\partial \underline{x}}\bigg|_{\underline{\hat{x}}_{t|t-1}}$$

being the Jacobian of the transition and observation functions at estimates $\underline{\hat{x}}_{t|t}$ and $\underline{\hat{x}}_{t|t-1}$, respectively. The nonlinear state-space model in (8.26) can then be approximated as

$$\begin{aligned} \underline{x}_{t+1} &= f(\underline{\hat{x}}_{t|t}) - \mathbf{F}_t\underline{\hat{x}}_{t|t} + \mathbf{F}_t\underline{x}_t + \underline{w}_t\,, \\ \underline{y}_t &= h(\underline{\hat{x}}_{t|t-1}) - \mathbf{H}_t\underline{\hat{x}}_{t|t-1} + \mathbf{H}_t\underline{x}_t + \underline{v}_t\,, \end{aligned}$$

enabling the direct application of Kalman filtering derived in Sec. 8.2. We summarize the equations below.

Time update

$$\begin{aligned} \underline{\hat{x}}_{t+1|t} &= f(\underline{\hat{x}}_{t|t}) \\ \mathbf{P}_{t+1|t} &= \mathbf{F}_t\mathbf{P}_{t|t}\mathbf{F}_t^\top + \mathbf{Q} \end{aligned}$$

Measurement update

$$\begin{aligned} \mathbf{K}_t &= \mathbf{P}_{t|t-1}\mathbf{H}_t^\top\big(\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^\top + \mathbf{R}\big)^{-1} \\ \underline{\hat{x}}_{t|t} &= \underline{\hat{x}}_{t|t-1} + \mathbf{K}_t\big(\underline{y}_t - h\big(\underline{\hat{x}}_{t|t-1}\big)\big) \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1} \end{aligned}$$

Chapter 9

Adaptive Filtering

In this chapter, we revisit model estimation problems in Chap. 6 and generalize it to modeling with time-varying parameters in the context of adaptive filtering. Adaptive filters are linear filters with variable parameters (coeffecients) which aim to converge to an optimal state. Compared with ordinary linear filters, they are of vital importance in many real-world applications such as system identification, communication systems, change detection, etc.

9.1 Problem Formulation

Many adaptive filtering problems can be formulated as estimating time-varying parameters of linear regression models. By concatenating a set of inputs into the regression vector $\underline{\varphi}_t$ at time stamp t, the observed output y_t is given by

$$y_t = \underline{\varphi}_t^\top \underline{\theta}_t + e_t \,, \tag{9.1}$$

with $\underline{\theta}_t$ denoting the parameter vector that changes over time and e_t the disturbance. Estimating the parameters over time t can be done in a recursive manner according to

$$\hat{\underline{\theta}}_t = \hat{\underline{\theta}}_{t-1} + \mathbf{K}_t \epsilon_t \,, \quad \text{with} \quad \epsilon_t = y_t - \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1} \tag{9.2}$$

denoting the prediction error (residual) and $\mathbf{K}_t$ the gain given measurement y_t in each recursion. For a concrete adaptive filtering algorithm, the gain term needs to be specified.

9.2 Adaptive Approaches

Given the conceptual formulation of adaptive filtering in (9.2), we introduce three approaches in the remainder of this section.

9.2.1 Recursive lest squares (RLS)

We recall the ordinary least squares introduce in Sec. 7.1 for linear regression. Given a set of measurements $\{y_k\}_{k=1}^N$, the time-invariant parameter $\hat{\underline{\theta}}$ is given by minimizing the sum of squares of residuals, i.e.,

$$\hat{\underline{\theta}} = \arg\min_{\underline{\theta}} \sum_{k=1}^{N} \left(y_k - \underline{\varphi}_k^\top \underline{\theta}\right)^2 , \tag{9.3}$$

leading to the closed form $\hat{\underline{\theta}}_N = \mathbf{R}_N^{-1} \underline{f}_N$, with $\mathbf{R}_N = \frac{1}{N}\sum_{k=1}^N \underline{\varphi}_k \underline{\varphi}_k^\top$ being the regressor/design matrix and $\underline{f}_N = \frac{1}{N}\sum_{k=1}^N \underline{\varphi}_k y_k$. In order to estimate a set of time-varying parameters $\underline{\theta}_t$, the ordinary least squares above can be adapted with the following time-varying loss function

$$\mathcal{V}_t(\underline{\theta}_t) = \sum_{k=1}^{t} \lambda^{t-k} (y_k - \underline{\varphi}_k^\top \underline{\theta}_t)^2 \tag{9.4}$$

given a set of observed output $\mathbb{Y}_t = \{y_k\}_{k=1}^t$. A forgetting factor $\lambda \in (0, 1]$ is introduced in (9.4) to put more weights on residual terms of more recent observations (since the recent ones contribute comparably more information to the updating the parameters). Intuitively, the term λ^{t-k} is small if k is small, meaning observations from past are (partially) "forgotten". Solving the optimization problem can be done similarly to the ordinary least squares by setting the first order of the loss function $\mathcal{V}_t$ to be zero. This leads to the following time-varying normal equations

$$\mathbf{R}_t \hat{\underline{\theta}}_t = \underline{f}_t \,, \quad \text{with} \quad \mathbf{R}_t = \sum_{k=1}^{t} \lambda^{t-k} \underline{\varphi}_k \underline{\varphi}_k^\top \quad \text{and} \quad \underline{f}_t = \sum_{k=1}^{t} \lambda^{t-k} \underline{\varphi}_k y_k \,. \tag{9.5}$$

Accordingly, the estimate can be obtained in closed form $\hat{\underline{\theta}}_t = \mathbf{R}_t^{-1} \underline{f}_t$.

Recursive computation The batch-wise procedure given in (9.5) induces growing computational cost as measurements $\mathbb{Y}_t$ accumulates over time, which can make online estimation intractable at certain point. Therefore, we introduce the recursive least squares. Given a newly coming measurement y_t, we can reformulate the regressor matrix $\mathbf{R}_t$ and $\underline{f}_t$ according to

$$\begin{aligned} \mathbf{R}_t &= \lambda \sum_{k=1}^{t-1} \lambda^{t-1-k} \underline{\varphi}_k \underline{\varphi}_k^\top + \underline{\varphi}_t \underline{\varphi}_t^\top = \lambda \mathbf{R}_{t-1} + \underline{\varphi}_t \underline{\varphi}_t^\top \,, \\ \underline{f}_t &= \lambda \sum_{k=1}^{t-1} \lambda^{t-1-k} \underline{\varphi}_k y_k + \underline{\varphi}_t y_t = \lambda \underline{f}_{t-1} + \underline{\varphi}_t y_t \,. \end{aligned} \tag{9.6}$$

Correspondingly, the current estimate can be updated based on the previous normal equations $\underline{f}_{t-1} = \mathbf{R}_{t-1} \hat{\underline{\theta}}_{t-1}$ recursively, namely,

$$\begin{aligned} \hat{\underline{\theta}}_t &= \mathbf{R}_t^{-1} \underline{f}_t = \mathbf{R}_t^{-1} (\lambda \underline{f}_{t-1} + \underline{\varphi}_t y_t) \\ &= \mathbf{R}_t^{-1} \big(\lambda \mathbf{R}_{t-1} \hat{\underline{\theta}}_{t-1} + \underline{\varphi}_t y_t \big) \\ &= \mathbf{R}_t^{-1} \Big((\mathbf{R}_t - \underline{\varphi}_t \underline{\varphi}_t^\top) \hat{\underline{\theta}}_{t-1} + \underline{\varphi}_t y_t \Big) \\ &= \hat{\underline{\theta}}_{t-1} + \mathbf{R}_t^{-1} \underline{\varphi}_t \big(y_t - \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1} \big) \,. \end{aligned}$$

However, the regressor matrix $\mathbf{R}_t$ in the expression above is inverted in each recursion which requires growing computational cost. To resolve this issue, we exploit the matrix inversion lemma

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{C}^{-1} + \mathbf{D} \mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{D} \mathbf{A}^{-1} \,, \tag{9.7}$$

based on which we aim to compute the regressor matrix also recursively. For that, we reformulate the recursion in (9.6) w.r.t. (9.7) into

$$\mathbf{R}_t = \mathbf{A} + \mathbf{BCD} \,, \quad \text{with} \quad \mathbf{A} = \lambda \mathbf{R}_{t-1} \,, \mathbf{B} = \underline{\varphi}_t \,, \mathbf{C} = \mathbf{I} \,, \mathbf{D} = \underline{\varphi}_t^\top \,.$$

We denote the inverted regressor matrix as $\mathbf{P}_t = \mathbf{R}_t^{-1}$, its computation then follows

$$\begin{aligned} \mathbf{P}_t &= \frac{1}{\lambda} \mathbf{P}_{t-1} - \frac{1}{\lambda} \mathbf{P}_{t-1} \underline{\varphi}_t (\mathbf{I} + \frac{1}{\lambda} \underline{\varphi}_t^\top \mathbf{P}_{t-1} \underline{\varphi}_t)^{-1} \underline{\varphi}_t^\top \frac{1}{\lambda} \mathbf{P}_{t-1} \\ &= \frac{1}{\lambda} \left(\mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1} \underline{\varphi}_t \underline{\varphi}_t^\top \mathbf{P}_{t-1}}{\lambda + \underline{\varphi}_t^\top \mathbf{P}_{t-1} \underline{\varphi}_t} \right) . \end{aligned} \tag{9.8}$$

Note that the term $\underline{\varphi}_t^\top \mathbf{P}_{t-1} \underline{\varphi}_t$ is placed in the denominator since it is a scalar value. Based on the recursive expression in (9.8), the estimate can be computed recursively with

$$\hat{\underline{\theta}}_t = \hat{\underline{\theta}}_{t-1} + \mathbf{P}_t \underline{\varphi}_t (y_t - \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1}) \,.$$

In comparison with the conceptual formulation of adaptive filtering given in (9.3), we have the gain in RLS given by $\mathbf{K}_t = \mathbf{P}_t \underline{\varphi}_t$, with $\mathbf{P}_t = \mathbf{R}_t^{-1}$ computed recursively according to (9.8).

9.2.2 Least mean squares (LMS)

As introduced in model estimation in Chap. 6 and Wiener filtering in Chap. 7, estimating a time-invariant system can be done via minimizing the expected value of squared prediction error in a batch-wise manner. For adaptive filtering, we can apply this principle on a stepwise basis given a newly-coming measurement y_t. More specifically, the time-varying parameter θ_t can be computed according to

$$\hat{\underline{\theta}}_t = \arg\min_{\underline{\theta}} \mathcal{V}_t(\underline{\theta}) \,, \quad \text{with} \quad \mathcal{V}_t(\underline{\theta}) = \frac{1}{2}\mathrm{E}\big\{(y_t - \underline{\varphi}_t^\top \underline{\theta})^2\big\} \tag{9.9}$$

being the loss function. In general, the optimization problem above can be solved numerically via gradient descent, where the estimate is updated recursively via

$$\hat{\underline{\theta}}_t = \hat{\underline{\theta}}_{t-1} - \mu \frac{\mathrm{d}\mathcal{V}_t(\underline{\theta})}{\mathrm{d}\underline{\theta}}\bigg|_{\underline{\theta}=\hat{\underline{\theta}}_{t-1}} \,. \tag{9.10}$$

Here, the term μ is the step length (also called learning rate) which regulates how much the estimate is changed in the gradient direction in each recursion. The gradient can be computed in the closed-form given the linear loss function in (9.9), namely,

$$\frac{\mathrm{d}\mathcal{V}_t(\underline{\theta})}{\mathrm{d}\underline{\theta}} = -\mathrm{E}\big\{\underline{\varphi}_t(y_t - \underline{\varphi}_t^\top \underline{\theta})\big\} \,.$$

However, computing the expected value of the residual is not feasible using a single measurement. Therefore, we neglect the expectation and replace it with an approximate given by the current observations $(\underline{\varphi}_t, y_t)$ to denote the gradient direction (therefore called “stochastic” gradient descent)

$$\frac{\mathrm{d}\mathcal{V}_t(\underline{\theta})}{\mathrm{d}\underline{\theta}} \approx -\underline{\varphi}_t\big(y_t - \underline{\varphi}_t^\top \underline{\theta}\big) \,. \tag{9.11}$$

Thus, the recursion in (9.10) can be implemented according to

$$\hat{\underline{\theta}}_t = \hat{\underline{\theta}}_{t-1} + \mu \underline{\varphi}_t(y_t - \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1}) \,. \tag{9.12}$$

Corresponding to the conceptual setup of adaptive filtering in (9.2), we have the gain of LMS given by $\mathbf{K}_t = \mu \underline{\varphi}_t$. The original LMS derived above is in general sensitive to the scaling of the observed input $\underline{\varphi}_t$, making it difficult to determine a proper learning rate μ to guarantee the stability of the optimization. In this regard, one can refer to the normalized least mean squares (NLMS) with recursion

$$\hat{\underline{\theta}}_t = \hat{\underline{\theta}}_{t-1} + \mu_t \underline{\varphi}_t(y_t - \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1}) \,,$$

where the learning rate is given by

$$\mu_t = \frac{\mu}{\underline{\varphi}_t^\top \underline{\varphi}_t + \alpha} \,.$$

Here, α is a small value close to machine precision to avoid zero in the denominator, and a good practice of setting the value μ is $\mu = 0.02$. In general, the normalized learning rate can stabilize the algorithm in the case of energy increase.

9.2.3 Adaptation using the Kalman filter

We can also consider the time-varying parameter as a hidden state undergoing random walk, and the linear regression model in (9.1) can be seen as a measurement model. Such a setup leads to a state-space model to which the Kalman filtering in Chap. 8 is applicable for parameter estimation. We summarize the state-space formulation of adaptive filtering as follows

$$\begin{aligned} \underline{\theta}_{t+1} &= \underline{\theta}_t + \underline{w}_t \\ y_t &= \underline{\varphi}_t^\top \underline{\theta}_t + e_t \,, \end{aligned} \tag{9.13}$$

with $\underline{w}_t$ and e_t being zero-mean process and measurement noises of covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$, respectively[1]. Essentially, the LMS and RLS are special case of the solution given by the Kalman filter. The Kalman gain in (8.13) can then be specified w.r.t. the setup in (9.13) into

$$\mathbf{K}_t = \frac{\mathbf{P}_{t-1}\underline{\varphi}_t}{\underline{\varphi}_t^\top \mathbf{P}_{t-1}\underline{\varphi}_t + \mathbf{R}_t} \,.$$

Based thereon, we can obtain the time-varying estimate via consecutive time updates (with one measurement update in between) as

$$\hat{\underline{\theta}}_t = \hat{\underline{\theta}}_{t-1} + \mathbf{K}_t(y_t - \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1}) \,, \tag{9.14}$$

which corresponds to the conceptual design of adaptive filtering in (9.2). Note that both measurement and time updates have been incorporated into the recursion in (9.14) (The predicted mean equals to the posterior mean at time stamp t given the random-walk process model). According to (8.15) and (8.9), we obtain the covariance of the time-varying estimate given by

$$\begin{aligned} \mathbf{P}_t &= (\mathbf{P}_{t-1} - \mathbf{K}_t \underline{\varphi}_t^\top \mathbf{P}_{t-1}) + \mathbf{Q}_t \\ &= \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1}\underline{\varphi}_t \underline{\varphi}_t^\top \mathbf{P}_{t-1}}{\underline{\varphi}_t^\top \mathbf{P}_{t-1}\underline{\varphi}_t + \mathbf{R}_t} + \mathbf{Q}_t \,. \end{aligned} \tag{9.15}$$

9.3 Properties of Adaptive Algorithms

The adaptive filtering algorithms in Sec. 9.2.1 (RLS) to Sec. 9.2.2 (LMS) can be in general described by the adaptation of Kalman filters in Sec. 9.2.3. In what follows in this section, we investigate several important properties of adaptive filtering.

Estimation error We assume that the true time-varying parameter $\underline{\theta}_t$ undergoes a random walk and are observed as described by the state-space model in (9.13). The estimate given by the linear regression in (9.2) then induces the following estimation error

$$\begin{aligned} \tilde{\underline{\theta}}_t &= \underline{\theta}_t - \hat{\underline{\theta}}_t \\ &= \underline{\theta}_{t-1} + \underline{w}_{t-1} - \hat{\underline{\theta}}_{t-1} - \mathbf{K}_t y_t + \mathbf{K}_t \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1} \,. \end{aligned} \tag{9.16}$$

Given the state-space model in (9.13), the measurement y_t can be expressed as

$$y_t = \underline{\varphi}_t^\top \underline{\theta}_t + e_t = \underline{\varphi}_t^\top (\underline{\theta}_{t-1} + \underline{w}_{t-1}) + e_t \,.$$

Thus, (9.16) can be further derived as

$$\begin{aligned} \tilde{\underline{\theta}}_t &= \underline{\theta}_{t-1} + \underline{w}_{t-1} - \hat{\underline{\theta}}_{t-1} - \mathbf{K}_t \underline{\varphi}_t^\top (\underline{\theta}_{t-1} + \underline{w}_{t-1}) - \mathbf{K}_t e_t + \mathbf{K}_t \underline{\varphi}_t^\top \hat{\underline{\theta}}_{t-1} \\ &= \underline{\theta}_{t-1} - \hat{\underline{\theta}}_{t-1} - \mathbf{K}_t \underline{\varphi}_t^\top (\underline{\theta}_{t-1} - \hat{\underline{\theta}}_{t-1}) - \mathbf{K}_t \underline{\varphi}_t^\top \underline{w}_{t-1} - \mathbf{K}_t e_t + \underline{w}_{t-1} \,. \end{aligned}$$

This further leads to the following recursion

$$\tilde{\underline{\theta}}_t = (\mathbf{I} - \mathbf{K}_t \underline{\varphi}_t^\top)\tilde{\underline{\theta}}_{t-1} + (\mathbf{I} - \mathbf{K}_t \underline{\varphi}_t^\top)\underline{w}_{t-1} - \mathbf{K}_t e_t \,, \tag{9.17}$$

with $\tilde{\underline{\theta}}_{t-1} = \underline{\theta}_{t-1} - \hat{\underline{\theta}}_{t-1}$ being the estimation error from previous step. The expression in (9.17) consists of three terms.

[1] Note that here matrix $\mathbf{R}_t$ denote the process noise covariance, not the regressor matrix in RLS.

- The term $(\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top)\underline{\tilde{\theta}}_{t-1}$ denotes the transient error, which regulates how the error term $\underline{\tilde{\theta}}_t$ develops over time (error transition). For instance, if we have a bad initial estimate of the parameter, the transient error tells how fast the error dies out.
- The term $(\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top)\underline{w}_{t-1}$ denotes the tracking/adaptationr error. According to the state-space model in (9.13), the process noise $\underline{w}_t$ controls how much the parameter changes at each time stamp. Thus, the tracking error quantifies how much the change in parameter affects the error of the estimate. In other words, it shows how well can we track the parameter variation.
- The term $\mathbf{K}_t e_t$ is the variance error describing how measurement noise affects the estimation error.

According to (9.17), it is straightforward to tell that the tracking error is small if the absolute eigenvalues $|\mathrm{eig}(\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top)|$ is small, making the adaptive algorithm good at tracking changes of parameter. And in order to suppress measurement noise, $|\mathrm{eig}(\mathbf{K}_t)|$ should also be small, namely, a small gain every time a new measurement is incorporated.

Transition The transition phase determines how long does it take for the estimate to converge to the true value. A short transition phase is possible if the absolute eigenvalues of the error transition matrix, i.e., $|\mathrm{eig}(\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top)|$, is small for small t (meaning shortly after initialization). In this case, the error will quickly die out once the system is started off.

Stability Based on the difference equation representation in (9.2), we can express an adaptive filter using the transfer function

$$\mathbb{H}(z) = \frac{\hat{\Theta}_t(z)}{\mathbb{Y}_t(z)} = \frac{\mathbf{K}_t}{\mathbf{I}-(\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top)z^{-1}}\,, \tag{9.18}$$

where the gain is specified by an adaptive algorithm as introduced in Sec. 9.2. The filter given in (9.18) is causal, and the necessary condition for stability is that all poles are strictly inside the unit circle. For vector-valued variables, this means that the absolute eigenvalues of the error transition matrix $\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top$ should be constrained by 1, namely, $|\mathrm{eig}(\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top)| < 1$. Given a noise-free process model, the error is asymptotically zero, i.e., $\lim_{t\to\infty}\underline{\tilde{\theta}}_t = 0$ (θ_t is constant). The following example demonstrates a simplified scenario of setting up a stable LMS-based adaptive filter.

Example 9.3.1. We repeat the gain of the least mean squares

$$K_t = \mu\varphi_t\,, \tag{9.19}$$

with μ being the learning rate and φ_t the regression vector (here assumed to be a scalar value). The error transition matrix then follows

$$1 - K_t\varphi_t = 1 - \mu\varphi_t^2$$

According to the condition for stability, we have $|1-\mu\varphi_t^2| < 1$, therefore, $0 < \mu < 2/\varphi_t^2$. In practice, we often set the condition to be more conservative. For instance, we can reduce the upper limit by a factor of 100, i.e., $\mu \approx 0.02/\varphi_t^2$.

9.4 Design Considerations

Based on the discussion for (9.17), we now introduce the several common considerations for designing adaptive filters using the approaches introduced in Sec. 9.2. The major difference among the three methods are different specifications of the gain, which we summarize in Tab. 9.1.

Transient error In order to quickly reach convergence after initialization, the absolute eigenvalues of the error transition matrix $\mathbf{I}-\mathbf{K}_t\underline{\varphi}_t^\top$ needs to be small. Thus, a “large” $\mathbf{P}_0$ (large initial uncertainty, thus large initial gains) is desirable for the KF. For RLS, a “small” initial regressor matrix $\mathbf{R}_0$ is needed for a high initial gain. For the LMS, there is no simple rule to follow. In general, RLS and KF have significantly shorter transient compared to LMS.

adaptive method	recursive least squares	least mean squares	Kalman filter
gain $\mathbf{K}_t$	$\mathbf{R}_t^{-1}\underline{\varphi}_t$	$\mu\underline{\varphi}_t$	$\left(\underline{\varphi}_t^\top \mathbf{P}_{t-1}\underline{\varphi}_t + \mathbf{R}_t\right)^{-1}\mathbf{P}_{t-1}\underline{\varphi}_t$

Table 9.1: List of gains in different adaptive approaches. Here, $\mathbf{R}_t$ is the regressor matrix of RLS in (9.5). μ denotes the learning rate in the LMS as shown in (9.12), and $\mathbf{P}_{t-1}$ is the covariance estimate in the KF according to (9.15).

Tracking/adaptation error To realize a fast adaptation speed in the case of parameter variation, the gain $\mathbf{K}_t$ needs to be high. For RLS, this means that a small value of the forgetting factor λ (e.g., $\lambda = 0.99$) is needed. For LMS, a large learning rate μ is necessary. For the KF, a "large" $\mathbf{Q}_t$ (high process noise) should be set up.

Variance error To suppress measurement noise (or reduce noise sensitivity), the gain $\mathbf{K}_t$ needs to be "small". Thus, a large forgetting factor λ is needed for RLS (e.g., $\lambda \approx 0.999$). For LMS, a small learning rate $\mu > 0$ is desired. For KF, the process noise should be low, meaning a "small" $\mathbf{Q}_t$ is needed.

Therefore, there exists a fundamental trade-off for tuning an adaptive filter between adaptation speed and noise sensitivity. For LMS, a big learning rate leads to fast adaptation, however, making the algorithm more sensitive to noise. For RLS, a rule of thumb for tuning the forgetting factor λ is that mainly the recent $\frac{1}{1-\lambda}$ measurements affect the current estimate (effective window size w_e). Thus, $\lambda = 0.99$ corresponds to the recent 100 measurements. For KF, we often fix the measurement noise covariance $\mathbf{R}_t$ and set process noise covariance in the form of diagonal matrix with $\mathbf{Q}_t = q\mathbf{I}$, where q is a scalar, or alternatively $\mathbf{Q}_t = \mathrm{diag}([q_1, q_2, ..., q_n])$ with q_i being the variance of $[\underline{w}_k]_i$. The latter setting then enables different tracking rate of various components in $\underline{\theta}_t$. In this case, RLS is often simpler to tune compared with KF, since the former requires tuning only for the scalar value λ and the latter for the matrix $\mathbf{Q}_t$.

Bibliography

[1] P. Stoica, R. L. Moses, *et al.*, *Spectral Analysis of Signals*, vol. 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.