Spectral analysis of discrete processes

In the previous chapter, we have examined spectral analysis for continuous-time processes. However, when working with computers, we can only deal with discrete-time processes. So, that's we'll focus on in this chapter. First, we'll look at how we can make a signal discrete. Second, we examine multiple ways to transform such discrete signals. Finally, we find out how can actually estimate the PSD function from a transformed discrete signal.

1 Making a signal discrete: sampling

1.1 The working principle of sampling

Let's suppose that we have a continuous signal $x(t)$. The Fourier transform of this function is $X(\omega)$. But we don't want a continuous signal. Instead, we want a discrete signal. To acquire one, we first define the **pulse train** $y(t)$ as

$$y(t) = \sum_{n=-\infty}^{+\infty} \delta(t - n\Delta t), \tag{1.1}$$

where n is an integer, Δt is the **sampling period** and $\omega_s = \omega_0 = 2\pi/\Delta t$ is the **sampling frequency**. According to a trick from the previous chapter, this pulse train has a Fourier transform of

$$Y(\omega) = \frac{2\pi}{\Delta t} \sum_{n=-\infty}^{+\infty} \delta\left(\omega - n\frac{2\pi}{\Delta t}\right). \tag{1.2}$$

We can use $y(t)$ to create the discrete signal $z(t)$ of $x(t)$. To do this, we multiply $x(t)$ by $y(t)$. Thus,

$$z(t) = x(t)y(t) = \sum_{n=-\infty}^{+\infty} x(t)\,\delta(t - n\Delta t). \tag{1.3}$$

Now we would like to find the Fourier transform $Z(\omega)$ of $z(t)$. Since multiplication in the time domain means convolution in the frequency domain, we get

$$Z(\omega) = \frac{1}{2\pi} X(\omega) * Y(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\xi)Y(\omega - \xi)\,d\xi. \tag{1.4}$$

By inserting the relation for $Y(\omega)$ and by working things out further, we can then find that

$$Z(\omega) = \frac{1}{\Delta t} \sum_{n=-\infty}^{+\infty} X(\omega - n\omega_0). \tag{1.5}$$

So basically, to find $Z(\omega)$, we take $X(\omega)$, scale it by $1/\Delta t$, make copies of it and then shift it by integer multiples of ω_0. This thus makes $Z(\omega)$ periodic. By the way, the above function $Z(\omega)$ is the **continuous-time Fourier transform** (CTFT) of the discrete signal. We will examine the discrete-time Fourier transform later in this chapter.

It is interesting to note the duality with the 'making-a-function-periodic' trick from the last chapter. There, we made a normal function $x(t)$ periodic by copying it and shifting these copies by integer multiple of T_0. The result was a discrete version of the original Fourier transform, scaled by a factor $1/T_0$. Here, we make a function discrete with time step Δt. The result is a copied-and-shifted version of the original Fourier transform, scaled by a factor $1/\Delta t$.

1.2 Losing data due to sampling

Of course, the downside of sampling is that data is lost. In the time domain, we only have data at the points $z(k\Delta t)$ (or alternatively, $z[k]$). The data in between these points is lost. In the frequency domain, something different occurs. Let's examine the function $Z(\omega)$ in the interval $[-\omega_s/2, \omega_s/2]$. In this interval, we don't only have the original function $X(\omega)$. Instead, copies of shifted versions of $X(\omega)$ are also added. This often makes it impossible to extract the original function $X(\omega)$ in this interval. This phenomenon is called **aliasing**.

Luckily, there is a trick to reduce this problem. We simply take the original signal $X(\omega)$ and, before we alias it, we remove all frequencies higher than the so-called **Nyquist frequency** $\omega_n = \omega_s/2$. (Alternatively, we can select the sampling frequency ω_s to be twice as high as the highest frequency that already exists.) This does mean that we lose some data about the higher frequencies. However, the resulting adjusted signal $X(\omega)$ won't be effected by the sampling. In other words, we keep perfect information about the more important lower frequencies.

Based on this information, we can also answer the question when we can reconstruct the exact original signal $x(t)$ from the sampled signal $z(t)$. Let's suppose that $X(\omega)$ is a **bandlimited signal**. This means that there is a frequency ω_M such that if $|\omega| > \omega_M$, then $X(\omega) = 0$. If we now choose the sampling frequency ω_s such that $\omega_s > 2\omega_M$, then we can perfectly reconstruct the original signal $x(t)$ from the sampled signal $z(t)$. If this is not the case, then there will be errors. (This theorem is called the **(Shannon) sampling theorem**.)

1.3 Reconstruction of the original signal

The transformation of the discrete-time signal $z(t)$ back to the continuous-time signal $x(t)$ is called **signal reconstruction**. To do this, we should simply take $Z(\omega)$ and only use the values in the interval $[-\omega_s/2, \omega_s/2]$. In other words, if we take as **reconstruction filter** $R(\omega)$ a block function with width ω_s, then we will simply have as reconstructed signal

$$X_r(\omega) = Z(\omega)R(\omega). \tag{1.6}$$

If we put this equation in the time-domain, then the reconstruction filter will be a sinc function. So, $r(t) = \operatorname{sinc}\left(\frac{\omega_s t}{2}\right) = \operatorname{sinc}\left(\frac{\pi t}{\Delta t}\right)$. To find the reconstructed signal $x_r(t)$ in the time domain, we can then use the convolution integral

$$x_r(t) = z(t) * r(t) = \int_{-\infty}^{+\infty} x(\theta)\delta(\theta - n\Delta t) \operatorname{sinc}\left(\frac{\omega_s}{2}(t-\theta)\right). \tag{1.7}$$

The above integrand only gives a nonzero value if $\theta = n\Delta t$. Thus, we find that

$$x_r(t) = \sum_{-\infty}^{+\infty} x(n\Delta t) \operatorname{sinc}\left(\frac{\omega_s}{2}(t - n\Delta t)\right). \tag{1.8}$$

So, the reconstructed signal is a sum of sinc functions. The nth sinc function has its center at $t = n\Delta t$ (so, at the nth sample) and has magnitude $x(n\Delta t)$ (which is the magnitude of the nth sample). Also, the sinc function is zero at the position of the other samples. So, at least we can always be sure that $x(n\Delta t) = x_r(n\Delta t)$ for all n. That is, the reconstructed signal equals the original signal at the position of the samples. In between these samples. $x_r(t)$ usually only approximates $x(t)$.

1.4 Reconstruction with insufficient data

The previous reconstruction method requires us to have $z(t)$ available at all times t. But in practice, this often isn't the case, as we can't look into the future. To solve this problem, we use a different

reconstruction method, called the **zero-order hold** (ZOH). We now simply say that the reconstructed signal $x_r(t)$ equals the last sample that we have found. So,

$$x_r(t) = z(k\Delta t) \quad \text{for} \quad k\Delta t \leq t < (k+1)\Delta t. \tag{1.9}$$

This method is actually equivalent with using as reconstruction signal $r(t)$ a block with width Δt, shifted in time by $\Delta t/2$. So, $r(t) = b(t - \Delta t/2)$. In the frequency domain, we then have

$$R(\omega) = e^{-j\omega\Delta t/2} B(\omega) = e^{-j\omega\Delta t/2} \Delta t \,\mathrm{sinc}\left(\frac{\omega\Delta t}{2}\right). \tag{1.10}$$

Alternatively, there is also the **first-order hold** (FOH), but we won't discuss that method here.

2 Discrete Fourier transforms

2.1 The discrete-time Fourier transform

It is time to really start to work in discrete time. First, we will denote $x(n\Delta t)$ simply as $x[n]$, and similarly, $z(n\Delta t) = z[n]$. (Note that $x[n]$ and $z[n]$ in fact are the same.) Now we define the **discrete-time Fourier transform** (DTFT) of a discrete signal $x[n]$ as

$$X(\Omega) = \mathcal{F}\{x[n]\} = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\Omega n} \qquad \text{and} \qquad x[n] = \mathcal{F}^{-1}\{X(\Omega)\} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} X(\Omega) e^{j\Omega n}\, d\Omega. \tag{2.1}$$

The DTFT $X(\Omega)$ is very similar to the CTFT $Z(\omega)$ of the discrete signal $z(t)$. The fundamental difference is that we now don't use ω but Ω. The relation between the two is given by $\Omega = \omega\Delta t$. Also, as can be seen from the above relation, the DTFT $X(\Omega)$ has a period of 2π, whereas the CTFT $Z(\omega)$ has a period of $2\pi/\Delta t$. And finally, we don't use times $n\Delta t$ anymore, but we simply use indices n. So, the DTFT can be seen as a 'normalized' version of the CTFT, such that the factor Δt has been taken out.

2.2 The discrete Fourier transform

A different type of transform is the **discrete Fourier transform** (DFT). (Don't confuse the DTFT with the DFT!) Let's suppose that we have N samples $x[n]$ ($0 \leq n < N$), taken with a sampling time Δt over a measurement time T. (So, $T = N\Delta t$.) The DFT $X[k]$ of $x[n]$ is now given by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n} \qquad \text{and, as inverse,} \qquad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{jn\frac{2\pi}{N}k}. \tag{2.2}$$

Just like the DTFT resembles the CTFT, so does the DFT resemble the CTFS. To see how, compare the above equation with the relations for the CTFS, given by

$$c_k = \frac{1}{T}\int_{t_0}^{t_0+T} x(t) e^{-jk\frac{2\pi}{T}t}\, dt \qquad \text{and} \qquad x(t) = \sum_{k=-\infty}^{+\infty} c_k e^{jk\frac{2\pi}{T}t}. \tag{2.3}$$

So, the period T from the CTFS resembles the number of measurements N in the DFT. In fact, the function $X[k]$ is periodic with period N. So, $X[k] = X[k+N]$ for all k. Also, it can be noted that $X[-k] = X[n-k] = X^*[k]$, where $X^*[k]$ is the complex conjugate of $X[k]$. (Remember that we also had $c_{-k} = c_k^*$ with the CTFS.)

Let's denote $f_s = 1/\Delta t$ as the sampling frequency (in Hertz). Also, we say that $f_s/N = 1/N\Delta t = 1/T$ is the **frequency resolution** (FR) of the DFT in Hertz. (The FR in radians per second is $2\pi f_s/N$.) The

DFT can now be used to measure how 'strong' the frequency $f = mf_s/N$ (with $-N/2 < m < N/2$) is present in the signal $x[n]$. To see how this works, let's examine a signal with frequency f_0, like

$$x(t) = \cos(2\pi f_0 t), \qquad \text{or equivalently,} \qquad x[k] = \cos(2\pi f_0 k \Delta t). \tag{2.4}$$

Let's first assume that we can write $f_0 = mf_s/N$ for some integer m. (That is, f_0 is a multiple of the frequency resolution. We do ought to have $m < N/2$ though, since higher frequencies can't be measured with only N measurements.) In this case, $|X[\pm m]|$ will have a relatively high value, while $X[k] = 0$ if $k \neq \pm m$. But now assume that we can't write $f_0 = mf_s/N$ for some integer m. In this case, **spectral leakage** occurs. This means that $X[k]$ will have a value for about all k. This makes it hard to determine the actual frequency of the signal.

2.3 Applying a window

The DFT has leakage, while the DTFT does not. So, to prevent leakage, we can in a way draw inspiration from the DTFT. We start with a signal $x[n]$. (For example, the one of equation (2.4).) When we use the DFT to acquire $X[k]$, we only consider the signal $x[n]$ for $0 \leq n < N$. We want to do a similar thing with the DTFT. So, we define the **rectangular time window** $w[n]$ as

$$w[n] = \begin{cases} 1 & \text{if } 0 \leq n < N, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

We then define the signal $y[n] = w[n]x[n]$. We put this signal $y[n]$ into the DTFT and find $Y(\Omega)$. It can now be shown that $X[k]$ actually equals $Y(\Omega)$ on the corresponding points. (That is, if $\Omega = 2\pi k/N$.)

Although the above is interesting, it doesn't solve the leakage problem. However, things are different if we use a different time window. Several good time windows are available. We could, for example, try the **Hanning window**

$$h[k] = \begin{cases} \frac{1}{2}\left(1 - \cos\left(\frac{2\pi k}{N}\right)\right) = \sin^2\left(\frac{\pi k}{N}\right) & \text{if } 0 \leq k < n, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

If we apply this window to the signal $x[k]$ (and thus get $y[k] = h[k]x[k]$) and put this new signal into the DFT, then there generally is less leakage. So, this significantly reduces our problem.

2.4 The fast Fourier transform

Let's suppose that we have a signal $x[n]$ and we want to find the DFT $X[k]$. We could simply use equation (2.2). However, this would require N^2 computations. The **fast Fourier transform** (FFT) is a very efficient algorithm that finds the DFT with only $N \log_2 N$ computations. Especially for big N, this saves a lot of computation time.

To apply the FFT, we first split the sequence $x[n]$ up into even terms $y[r]$ and odd terms $z[r]$. So,

$$y[r] = x[2r] \qquad \text{and} \qquad z[r] = x[2r+1] \qquad \text{for } 0 \leq r < N/2. \tag{2.7}$$

For each of the sequences $y[r]$ and $z[r]$, we now find the DFTs $Y[k]$ and $Z[k]$ for $0 \leq k < N/2$. We then put these two together to find $X[k]$, according to

$$X[k] = Y[k] + e^{-j\frac{2\pi k}{N}} Z[k]. \tag{2.8}$$

There is, however, a problem. During computations, we only have the values for $Y[k]$ and $Z[k]$ for $0 \leq k < N/2$. So the above equation only works for $0 \leq k < N/2$. But we need to find $X[k]$ for $0 \leq K < N$. To solve this problem, we make use of the fact that $Y[k]$ and $Z[k]$ are, in reality, periodic

functions with period $N/2$. So, $Y[k] = Y[k + N/2]$ and $Z[k] = Z[k + N/2]$. By using this fact, we can find that, for $N/2 \leq K < N$, we have

$$X[k + N/2] = Y[k + N/2] + e^{-j\frac{2\pi(k+N/2)}{N}} Z[k + N/2] = Y[k] - e^{-j\frac{2\pi k}{N}} Z[k]. \tag{2.9}$$

(Note that we have used the fact that $e^{-j\pi} = -1$.) In this way, we can derive $X[k]$ from $Y[k]$ and $Z[k]$ in a very efficient way. The question remains, how do we find $Y[k]$ and $Z[k]$? Well, this is simply done recursively: we again use the FFT.

3 Calculating spectral estimates

3.1 An initial estimate

The DFT is quite popular, because it can be used to approximate the PSD function. Let's suppose that we have a stochastic process $\bar{x}(t)$. We have several samples $x[n]$ of a realization of this process. We have also found the DFT $X[k]$ of $x[n]$. The **Periodogram** $I_{N_{\bar{x}\bar{x}}}[k]$ of this signal $x[n]$ is now defined as

$$I_{N_{\bar{x}\bar{x}}}[k] = \frac{1}{N} X[-k]X[k] = \frac{1}{N} X^*[k]X[k] = \frac{1}{N}|X[k]|^2. \tag{3.1}$$

This periodogram is an unbiased estimate of the discrete-time PSD function $S_{\bar{x}\bar{x}}[k]$. That is,

$$\lim_{N\to\infty} \mathrm{E}\left\{I_{N_{\bar{x}\bar{x}}}[k]\right\} = S_{\bar{x}\bar{x}}(\omega). \tag{3.2}$$

However, it is not a consistent estimate: the variance doesn't go to zero as $N \to \infty$. Instead, we have

$$\lim_{N\to\infty} \mathrm{var}\left\{I_{N_{\bar{x}\bar{x}}}[k]\right\} = \sigma^2_{I_N} = \sigma^4_{\bar{x}} \neq 0. \tag{3.3}$$

3.2 Improving the estimate

We would like to reduce the variance of our estimate $I_{N_{\bar{x}\bar{x}}}[k]$. One way to do this is to take multiple signals $x[n]$ and derive an estimate $I_{N_{\bar{x}\bar{x}}}[k]$ for each one of them. However, the problem is that we usually only have one signal $x[n]$. According to **Bartlett's procedure**, we then simply divide this signal into K signals, each having M samples, with $N = KM$. The ith signal would thus be $x^{(i)}[n] = x[n + (i-1)M]$, with $0 \leq i < K$ and $0 \leq n < M$. The estimate $\hat{S}_{\bar{x}\bar{x}}[k]$ of the discrete-time PSD function now becomes

$$\hat{S}_{\bar{x}\bar{x}}[k] = \frac{1}{K}\sum_{i=1}^{K} I_M^{(i)}[k], \qquad \text{with} \qquad I_M^{(i)}[k] = \frac{1}{M}\left|\sum_{n=0}^{M-1} x^{(i)}[n] e^{-j\frac{2\pi k}{M}n}\right|^2. \tag{3.4}$$

The variance of the estimate $\hat{S}_{\bar{x}\bar{x}}[k]$ is now $1/K$ times the variance of each periodogram. So, that's a significant improvement. However, there of course is a downside. The frequency resolution equals $2\pi f_s/M$. So, decreasing the variance of the estimate means that you decrease the frequency resolution as well. When applying Bartlett's procedure, you thus have to make a tradeoff between these two parameters.

Let's suppose that we've finally found a satisfactory estimate $\hat{S}_{\bar{x}\bar{x}}[k]$ of the discrete-time PSD function $S_{\bar{x}\bar{x}}[k]$. Now we want to find an estimate of the actual continuous-time PSD function $\hat{S}_{\bar{x}\bar{x}}(\omega)$. What is the relation between the two? Well, at the end of paragraph 1.1, we've seen that transforming a discrete version of a signal with time step Δt is the same as transforming the continuous signal, copying-and-shifting it and scaling it by $1/\Delta t$. To reverse this, we thus need to scale the function $\hat{S}_{\bar{x}\bar{x}}[k]$ back by a factor Δt. So,

$$\hat{S}_{\bar{x}\bar{x}}(\omega) = \Delta t \hat{S}_{\bar{x}\bar{x}}[k]. \tag{3.5}$$

Here, we have $\omega\Delta t = \frac{2\pi k}{N}$. Do note though, that the estimate $\hat{S}_{\bar{x}\bar{x}}(\omega)$ is only valid for $-2\pi f_s/2 < \omega < 2\pi f_s/2$. Also, the resulting frequency resultion is given by $2\pi f_s/M$.