Chapter 7

Extracting Features from Time Series

Christian Herff and Dean J. Krusienski

7.1 Time-Domain Processing

Raw time-series data, sometimes referred to as a *signal*, is inherently represented in the time-domain. Time-domain processing directly exploits the temporal relations between data points and generally provides an intuitive representation of these relationships. Time-domain techniques often aim to identify and detect the temporal morphology of transient or stereotyped information in the time series. When the information of interest repeats over regular or semi-regular intervals, straightforward transformations can be used to convert the time-domain information to the frequency-domain, which can isolate oscillatory information for comparison within and across oscillatory frequencies present in the time series. This section discusses some fundamental time-(no space) domain techniques and shows how oscillations in the time-domain data lead to frequency-domain representations.

7.1.1 Basic Magnitude Features and Time-Locked Averaging

Peak-picking and integration are two of the most straightforward and basic feature-extraction methods. Peak-picking simply determines the minimum or maximum value of the data points in a specific time interval (usually defined relative to a specific labeled event in the data) and uses that value (and possibly its time of occurrence) as the feature(s) for that time segment. Alternatively, the time series can

C. Herff
Maastricht University, Maastricht, The Netherlands
e-mail: c.herff@maastrichtuniversity.nl

D. J. Krusienski (✉)
Department of Biomedical Engineering, Virginia Commonwealth University, Richmond, VA, USA
e-mail: dkrusien@odu.edu

be averaged or integrated over all or part of the time interval to yield the feature(s) for the segment. Some form of averaging or integration is typically preferable to simple peak-picking, especially when the responses to the stimulus are known to vary in latency and/or when there is noise in the time series that can corrupt a simple peak estimation. These same methods can be applied for tracking transient magnitude peaks in the frequency domain.

When multiple observations of a noisy time series are available, the observations can be time-aligned (typically to an event onset or cyclic phase) and averaged across observations. The resulting average reduces the uncorrelated noise and can reveal the common time-series morphology across observations. For uncorrelated noise, the signal-to-noise ratio of the average increases by a factor of $\sqrt{K}$, where K is the number of observations in the average. When applicable, such averaging increases the reliability of feature estimates. Figure 7.1 shows an example of time-locked averaging relative to an event onset. These events can be, for example, external

Fig. 7.1 Example of time-locked averaging on EEG data. The individual observations are averaged to produce the bottom waveform. Note that, for noisy data, the individual observations may not exhibit obvious amplitude peaks that are characteristic of the underlying signal and clearly revealed in the average

stimuli such as a flashing light or regular measurement intervals. As with many time-series processing approaches, time-locked averaging assumes that the time series remains stationary, meaning that the parameters of the underlying data distribution (e.g., mean and variance) do not change over time. Additional considerations must be taken into account when dealing with non-stationary time series [1].

7.1.2 Template Matching

The similarity of portions of a time series to a predefined template can also be used as a feature. The similarity is generally computed by performing a sliding correlation of the matched filter template with the time series. The output of the filter template will be high for the segments that closely resemble the template and low for segments that differ from the template. Figure 7.2 illustrates an example of matched filtering for the electrocardiogram. Wavelet analysis (see Sect. 7.3 on Time-Frequency Features in this chapter) can be considered a variation of this method; it uses templates with specific analytical properties to produce a frequency decomposition related to the Fourier analysis.

Fig. 7.2 An example of template matching on ECG. The top trace shows the raw ECG time series. The middle trace shows the raw ECG plus uncorrelated random noise. The matched-filter template representing the QRS complex is shown in red. The bottom trace represents the squared output of the matched filter. Note that the peaks clearly and precisely align with each QRS complex in the raw signal, regardless of whether the added noise increases the amplitude beyond the original peaks. By utilizing the characteristic temporal morphology of the desired time-series event, the matched filter can provide a more reliable output than applying a simple amplitude threshold detection on the noisy time series

7.1.3 Weighted Moving Averages: Frequency Filtering

The concept of frequency filtering of a time series is best understood by first exploring how weighted moving averages can be used to manipulate the time series. The basic concept of frequency filtering is shown in Fig. 7.3, where moving average filters (highpass and lowpass) are applied to a time series containing the sum of a high-frequency and low-frequency sinusoidal component. For the *lowpass filter*, the low-frequency oscillation “passes through” the filter and is largely preserved while the high-frequency oscillation is largely suppressed. Likewise, for the *highpass filter*, the high-frequency oscillation passes through and is largely preserved while the low-frequency oscillation is partially suppressed. Note that the degrees of preservation/suppression are determined by the characteristics of the weighted moving average, for which the basic principles are outlined in this section.

The most basic form of a weighted moving average is the uniform moving average, where the current data point and the prior $N-1$ data points are summed and divided by N . This is equivalent to multiplying each data point by $1/N$ (i.e., weighting by $1/N$) and summing. The process is repeated for each subsequent data point,

Fig. 7.3 Effect of moving average filters on a time series consisting of a sum of two sinusoids. The high frequency component of the original signal (blue) is attenuated (orange) when using a uniform moving average lowpass filter. In this case a moving average filter of length 40 was used. Note the slight phase shift induced by the filtering process. When using an alternating moving average highpass filter, lower frequency components in the original time series are attenuated (green), primarily leaving the high-frequency component of the original time series. Note that filters attenuate the undesired frequencies (i.e., the stopband) and may not completely remove them, as can be seen in the low frequency oscillations still present in the green time series. Filters can be designed to increase the stopband attenuation by adjusting the filter coefficients and/or filter length

forming a new time series. A uniform moving average with $N = 4$ is illustrated in the upper portion of Fig. 7.4 for a sinusoidal input time series. The left portion of Fig. 7.5 shows how the moving average output differs as the frequency of the sinusoidal input changes. Notice that this weighting perfectly preserves the output time series with no oscillation and progressively attenuates the amplitude of the output time series as the oscillation frequency increases. This is the most basic form of a *lowpass filter*, which preserves the amplitude of low frequency oscillations and attenuates the amplitude of higher frequency oscillations. As N increases, the range of low-end frequencies that are preserved decreases because a longer average covers more cycles of high-frequency oscillations, where the positive and negative half cycles are canceled in the average.

By simply alternating the sign of each weight in the moving average, the opposite effect is observed as shown in the bottom of Fig. 7.4 and the left portion of Fig. 7.5. In this case, the amplitudes of the lower frequencies are attenuated and the higher frequencies are preserved. This is the most basic form of a *highpass filter*, which preserves the amplitude of high frequency oscillations and attenuates the amplitude of lower frequency oscillations. Just as with the lowpass filter, as N increases, the range of high-end frequencies that are preserved decreases because only the oscillation frequencies that are near the oscillation frequency of the weights are preserved.

Fig. 7.4 The resulting output time series generated from applying a uniform moving average of length 4 (top) and alternating moving average of length 4 (bottom) to a 25 Hz sinusoidal input time series. At each time instant, the red filter weights overlapping with the input time series scale each input sample value according to the corresponding filter weight value. The sum of the resulting 4 weighted values produce the output value at the rightmost filter time point. The red filter weights slide from left to right across the input time series to produce each subsequent output value. The top represents a lowpass filter that attenuates the amplitude at this particular input frequency less than it is attenuated by the bottom highpass filter

Fig. 7.5 Simple moving average lowpass ($1/4 \ 1/4 \ 1/4 \ 1/4$) and highpass ($1/4 \ -1/4 \ 1/4 \ -1/4$) filters in the time and frequency domains. The left column compares the input-output relationship for different input frequencies. The right column shows the gain (magnitude) and phase response of the filters. The selected points on the frequency-domain graphs correspond to the frequencies in the left column. Note that the output time series are scaled and shifted with respect to the input time series by the values corresponding to the frequency-domain graphs

Based on these two rudimentary filter types, it can be surmised that the weight values in the moving average (i.e., filter weights or coefficients) and length of the average (i.e., filter length) can be adjusted to preserve and attenuate arbitrary frequency ranges, as well as to produce different output characteristics such as increased attenuation of undesired frequencies. In addition to lowpass and highpass filters, the two other basic filter designs are the bandpass and band-reject filters. A bandpass filter attenuates a range of low and high frequency oscillations, while preserving an intermediate range of frequencies. In contrast, a band-reject filter preserves a range of low and high frequency oscillations, while attenuating an intermediate range of frequencies.

The frequency response characteristic of a given filter, or the magnification/attenuation factor (i.e., gain) produced with respect to input oscillation frequency, is typically visualized in a *frequency-domain* plot as shown for the 4 fundamental filter types in the right portion of Fig. 7.5. Notice that the time variable is removed and the plots merely track the attenuation for each input oscillation frequency as illustrated in the time domain in Fig. 7.4. The region of the frequency response that preserves the oscillations is referred to as the *passband* and the region that attenuates the oscillations is referred to as the *stopband*. The region between the passband and stopband is referred to as the *transition band*. For practical filters, there is some finite slope to the transition band because a perfect threshold between frequencies (i.e., an ideal filter with infinite slope) requires an infinite length filter. Therefore, by convention, the threshold of the transition band is defined in the frequency

response characteristic as the point where the attenuation drops by 3 decibels (3 dB point) from the passband. This 3 dB point is referred to as the filter's cutoff frequency.

Returning to Fig. 7.4, not only does the amplitude between the input and output time series change depending on the oscillation frequency, but the output time series may be shifted (i.e., delayed) in time. Note that, for moving average filters with symmetric weights about the center of the average, the time shift will be constant for all input frequencies and equal to the length of the moving average divided by 2. This is known as linear phase response. Thus, for real-time applications, the length of the weighted moving average (i.e., filter length) impacts the delay between the input and output time series. Furthermore, because longer averages preserve/attenuate tighter frequency ranges, there is a trade-off between the precision of frequency discrimination and the amount of delay introduced for a given filter length.

The weighted moving average filters discussed to this point are more formally referred to as *finite impulse response (FIR) filters* because they will always produce a finite-length output time series if the input time series is finite in length. A common method to determine FIR filter weights to match a desired frequency response characteristic is known as the equiripple design, which minimizes the maximum error between the approximated and desired frequency response.

7.1.3.1 Weighted Moving Averages with Feedback

By taking an FIR filter structure and including weighted values of the past output values (i.e., feedback), a different filter structure is formed known as an *infinite impulse response (IIR) filter*. The basic idea is that, due to feedback, the output of the filter may continue infinitely in time even if the input time series is finite in length. The advantages of IIR filters over FIR filters is that they offer superior precision of frequency discrimination using fewer data points in the averaging (i.e., lower filter order). This also generally equates to shorter delay times. However, IIR filters tend to distort the output time series because, in contrast to symmetric FIR filters, all input frequencies generally do not experience the same time delay. This is known as nonlinear phase response. Additionally, unlike FIR filters, IIR filters can be unstable if not designed carefully. This occurs when there is a positive feedback loop that progressively increases the amplitude of the output until it approaches infinity, which is highly undesirable and potentially damaging to the system.

To determine the weights of an IIR filter to meet a desired frequency response characteristic, one of four common designs is typically selected (see Fig. 7.6 for the corresponding filter magnitude responses):

Butterworth: Provides a flat passband and stopband with the smallest transition-band slope for a given filter order.

Chebyshev I: Provides a flat passband and rippled stopband with greater transition-band slope for a given filter order compared to Butterworth.

Fig. 7.6 Magnitude response plots of FIR and IIR filters. All filters are 8th-order except a 40th-order FIR filter for comparison. Note the transition-band slope and passband and stopband ripple for the various filter types. Also note that a significantly longer 40th-order FIR filter is needed to approach the transition-band slope of the 8th-order Butterworth filter

Chebyshev II: Provides a flat stopband and rippled passband with greater transition-band slope for a given filter order compared to Butterworth (equivalent to Chebyshev I).

Elliptic: Provides a rippled passband and stopband with greatest transition-band slope for a given filter.

A flat passband or stopband means that the oscillations in the band will be preserved or attenuated by a uniform gain factor. A rippled passband or stopband means that oscillations in the band will be preserved or attenuated by a factor that varies with frequency, which is generally undesirable and can be minimized using design constraints. Thus, if frequency discrimination (i.e., sharp transition bands) is paramount for the filter application and ripple can be tolerated in both bands, then an elliptic filter will provide the best result for a given filter order. If the application requires uniform frequency preservation and attenuation in the respective bands, then a Butterworth is warranted with the compromise of the sharpness of the transition band.

In summary, the main considerations when selecting/designing a filter are:

Filter Order Affects output delay for minimum-latency or real-time applications, longer orders are required for constraints approaching ideal filters such as transition band and stopband attenuation, elliptic IIR generally provides the lowest order for given constraints but other trade-offs must be considered (e.g., nonlinear phase and ripple).

Linear Phase Constant phase delay (no phase distortion), achieved by symmetric FIR and can be approximated with an IIR, particularly Butterworth.

Filter Precision the sharpness of the transition band for separating two adjacent oscillations, elliptic IIR generally provides the sharpest transition for a given filter order but other trade-offs must be considered (e.g., nonlinear phase and ripple).

Passband/Stopband Smoothness Degree of amplitude distortion in the passband and stopbands. Butterworth IIR provides a smooth passband and stopband. The Chebychev variants can be used to obtain sharper transition bands if it is critical for only one band (passband or stopband) to be smooth.

Stopband Attenuation How well the filter will block the undesired oscillations in the stopband. For a fixed filter order, there will be a trade-off between filter precision and stopband attenuation.

7.2 Frequency-Domain Processing

Thus far, we have shown how weighted moving averages (i.e., FIR and IIR filters) can preserve/attenuate specific oscillatory frequencies present in an input time series. This forms the basis of frequency-domain processing. What has not yet been emphasized is that practical time series are comprised of a mixture of many (possibly infinite) oscillations at different frequencies. Specifically, a time series can be uniquely represented as a sum of sinusoids, with each sinusoid having a specific oscillation frequency, amplitude, and phase shift (delay factor). The method of determining these frequency, amplitude, and phase values for a given time series is known as the Fourier Transform. The Fourier transform converts the time series from the time-domain to the frequency-domain, similar to what is described in the previous section for the frequency response characteristic of a filter. The significance of converting a time series to the frequency-domain is that the specific oscillations present in the time series and their relative amplitudes and phases can be more easily identified, particularly compared to a time-domain visualization of a mixture of many different oscillations. By representing and visualizing in the frequency domain, filter response characteristics can be better designed to preserve/attenuate specific oscillations present in the time series. The filters described previously can operate on a time series that is comprised of a mixture of oscillations in a way that the mixture of oscillations observed at the output is completely defined by the frequency response characteristic of the filter. In other words, if a time series is a simple sum of a low-frequency oscillation and a high-frequency oscillation, an appropriately-designed lowpass filter would preserve only the low-frequency oscillation at the output and sufficiently attenuate the high-frequency oscillation.

7.2.1 Band Power

One of the most straightforward and intuitive methods for tracking amplitude modulations at a particular frequency is to first isolate the frequency of interest by filtering the signal with a bandpass filter. This produces a signal that is largely sinusoidal. Next, to estimate the positive amplitude envelope, the signal is rectified by squaring the signal or by computing its absolute value. Finally, the adjacent peaks are

Fig. 7.7 Straight forward approach to extract amplitude modulations in a specific frequency range. The original signal (blue line) is bandpass filtered (orange line) first. The filtered signal is then rectified (yellow) to estimate the positive amplitude envelope. Adjacent peaks are smoothed (purple) using integration or low-pass filtering. The slight delay in the resulting instantaneous magnitude can be seen in the difference in the lower panel

smoothed together via integration or low-pass filtering. The effects of each of these steps are illustrated in Fig. 7.7. Although the smoothed signal (Fig. 7.7) tracks the magnitude envelope of the frequency of interest, the resulting instantaneous magnitude estimate will be slightly delayed due to the smoothing step. When tracking and comparing multiple frequency bands, it is typically preferable to use an FFT- or AR-based method rather than computing band power at multiple frequencies.

7.2.2 Spectral Analysis

7.2.2.1 Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) is an efficient algorithm to transfer a time series into a representation in the frequency-domain. The FFT represents the frequency spectrum of a digital signal with a frequency resolution of sample-rate/FFT-points,

where the FFT-point is a selectable scalar that must be greater or equal to the length of the time series and is typically chosen as a base-2 value for computational efficiency. Because of its simplicity and effectiveness, the FFT often serves as the baseline method to which other spectral analysis methods are compared.

The FFT takes an N -sample time series and produces N frequency samples uniformly spaced over a frequency range of sampling rate/2, thus making it a one-to-one transformation that incurs no loss of information. The maximum frequency of sampling rate/2 in this transformation is called Nyquist frequency and refers to the highest frequency that can be reconstructed using the FFT. These frequency domain samples are often referred to as frequency bins. Each bin of the FFT magnitude spectrum tracks the sinusoidal amplitude of the signal at the corresponding frequency. The FFT will produce complex values that can be converted to magnitude and phase. The FFT spectrum of a real signal has symmetry such that only half of the bins are unique, from zero to + sampling rate/2. The bins from zero to sampling rate/2 are a mirror image of the positive bins about the origin (i.e., zero frequency). Therefore, for an N -sample real signal, there are $N/2$ unique frequency bins from zero to sampling rate/2. Knowing this fact allows one to apply and interpret the FFT without a firm grasp of the complex mathematics associated with the notion of “negative frequencies.”

Finer frequency sampling can be achieved by appending M zeros to the N -sample signal, producing $(M + N)/2$ bins from zero to the sampling rate/2. This is known as zero padding. Zero padding does not actually increase the spectral resolution since no additional signal information is being included in the computation, but it does provide an interpolated spectrum with different bin frequencies.

Note that it is also common to refer to the power spectrum or power spectral density (PSD) rather than the magnitude spectrum. Signal power is proportional to the squared signal magnitude. A simple estimate of the power spectrum can be obtained by simply squaring the FFT magnitude. More robust FFT-based estimates of the power spectrum can be obtained by using variants of the periodogram [2]. Figure 7.8 illustrates the power spectral density obtained using the squared FFT on a time series consisting of the sum of two sinusoids. It is observed that the FFT resolves the frequency of each sinusoid.

Fig. 7.8 Signal in the time-(left) and frequency-domain (right). The time-domain signal was sampled at a rate of 5 Hz resulting in a Nyquist-Frequency of 2.5 Hz. In the frequency-domain, it can easily be seen that the time-domain signal was created by a combination of a 0.5 Hz and a 1 Hz sinusoidal signal

7.2.2.2 Windowing

Because N -sample signal blocks may section the signal abruptly to create false discontinuities at the edges of the block, artificial ripples tend to be produced around the peaks of the spectrum. This can be mitigated by multiplying the block of samples by a tapering window function that tapers the samples at the edges of the sample block, thus reducing the ripples in the spectrum. Two of the most common tapering windows are the Hamming and Hanning windows, which both provide a good balance in terms of ripple attenuation and spectral resolution trade-offs produced by windowing. An example of a tapering window and the windowed signal is given in the top pane of Fig. 7.9. Although this acts to smooth the spectral ripples, it also expands the width of the frequency peaks, and thus lowers the overall spectral resolution as shown in the bottom pane of Fig. 7.9. In many cases, this is a tolerable trade-off for obtaining a smoother spectrum.

7.2.2.3 Autoregressive (AR) Modeling

Autoregressive (AR) modeling is an alternative to Fourier-based methods for computing the frequency spectrum of a signal. AR modeling assumes that the signal being modeled was generated by passing white noise through an infinite impulse response (IIR) filter. The specific weights of the IIR filter shape the white noise input to match the characteristics of the signal being modeled. White noise is essentially random noise that has the unique property of being completely uncorrelated when compared to any delayed version of itself. The specific IIR filter structure for AR modeling uses no delayed input terms and p delayed output terms. This structure allows efficient computation of the IIR filter weights. Because white noise has a completely flat power spectrum (i.e., the same power at all frequencies), the IIR filter weights are set so as to shape the spectrum to match the actual spectrum of the time series being analyzed.

Because the IIR filter weights define the signal's spectrum, AR modeling can potentially achieve higher spectral resolution for shorter signal blocks than can the FFT. Short signal blocks are often necessary for real-time applications. Additionally, the IIR filter structure accurately models spectra with sharp, distinct peaks, which are common for biological signals such as ECG or EEG. [2] discusses the theory and various approaches for computing the IIR weights (i.e., AR model) from an observed signal.

The primary issue with AR modeling is that the accuracy of the spectral estimate is highly dependent on the selected model order (p). An insufficient model order tends to blur the spectrum, whereas an overly large order may create artificial peaks in the spectrum, as illustrated in the bottom of Fig. 7.9. The complex nature of many time series should be taken into account for accurate spectral estimation, and this often cannot be reliably accomplished with such small model

Fig. 7.9 Comparison of spectra generated by FFT, FFT with Hanning window, and AR models of 3 different orders. The left panel shows the time-domain signal before and after applying the Hanning window, also showing the shape of the Hanning window envelope, scaled for effect. The right panel shows The FFT of the original signal, the FFT of the Hanning-windowed signal, and spectra for 3 AR model orders using the original signal

orders. It should be noted that the model order is dependent on the spectral content of the signal and the sampling rate. For a given signal, the model order should be increased in proportion to an increased sampling rate. More information about AR modeling can be found in [3, 4].

7.3 Time-Frequency Processing: Wavelets

For the conventional spectral-analysis techniques discussed thus far, the temporal and spectral resolution of the resulting estimates are highly dependent on the selected time series length, model order, and other parameters. This is particularly problematic when the time series contains transient oscillations that are localized in time. For instance, for a given time series observation length, the amplitude of a particular high-frequency oscillation (with respect to the observation length) has the potential to fluctuate significantly over each cycle within the observation. In contrast, the amplitude of a lower-frequency oscillation will not do so because a smaller number of cycles occur within the observation. For a given observation length, the FFT and AR methods produce only one frequency bin that represents these fluctuations at the respective frequency. By observing this bin in isolation, it is not possible to determine when a transient oscillation at that particular frequency occurs within the observation. Wavelet analysis solves this problem by producing a time-frequency representation of the signal. However, as predicted by Heisenberg's uncertainty principle, there is always a time/frequency resolution trade-off in time series analysis: it is impossible to precisely determine the instantaneous frequency and time of occurrence of an event. This means that longer observation lengths will produce spectral estimates having higher frequency resolution, while shorter time windows will produce estimates having lower frequency resolution.

Conceptually, rather than representing a time series as a sum of sinusoids as with the FFT, wavelet analysis instead represents the time series as a sum of specific time-limited oscillatory pulses, known as wavelets. These pulses have an identical morphology, referred to as the mother wavelet. The set of wavelets used to represent the time series are simply time-scaled and shifted versions of the mother wavelet, as shown for one common type of mother wavelet (Daubechies 4) in the upper portion of Fig. 7.10. The various time scales of the mother wavelet are roughly analogous to the sinusoidal frequencies represented by the FFT. Thus, each member of the wavelet set effectively represents a specific oscillatory frequency over a specific time interval, resulting in a time-frequency decomposition of the time-series. A comparison of the reconstructions achieved by wavelet and FFT representations is shown in the lower portion of Fig. 7.10.

There are a wide variety of mother wavelets, and each has specific time-frequency characteristics and mathematical properties. In addition, application-specific mother wavelets can be developed if general mother wavelet characteristics are known or desired. [5, 6] provide the theoretical details of wavelets.

Fig. 7.10 **Upper panel:** Example mother wavelet (top row) and sinusoidal functions (bottom row) at different scales/frequencies used to decompose time series for the wavelet transform and the FFT, respectively. Note that the wavelets are time-limited while the sinusoids extend to $t = \infty$. **Lower panel:** Reconstructions (black) of the gold trace using (a) wavelet transform with 32 co-efficients (b) wavelet transform with 180 coefficients (c) FFT with 32 coefficients (d) FFT with 180 coefficients

7.4 Conclusion

This chapter provided a broad overview of common techniques to extract meaningful features from time-series data. The reader should be familiarized with the basic concepts of time-domain analysis and the transition to frequency domain using filtering and Fourier and wavelet analyses. For a deeper understanding of the topic, dedicated textbooks are recommended (e.g. [7, 8]).

References

1. Priestley MB. Non-linear and non-stationary time series analysis. London: Academic Press; 1988.
2. Hayes MH. Statistical digital signal processing and modeling. New York: John Wiley & Sons; 1996.
3. Hamilton JD. Time series analysis, vol. 2. Princeton: Princeton university press; 1994.
4. Madsen H. Time series analysis. Hoboken: CRC Press; 2007.
5. Mallat S. A wavelet tour of signal processing. San Diego: Academic press; 1999.
6. Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. IEEE Trans Inf Theory. 1990;36(5):961–1005.
7. Lyons RG. Understanding digital signal processing, 3/E. Upper Saddle River: Prentice Hall; 2011.
8. Proakis JG, Manolakis DG. Digital signal processing: principles algorithms and applications. Upper Saddle River: Pearson Prentice Hall; 2007.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.