

Review Article

A Guide on Spectral Methods Applied to Discrete Data in One Dimension

Martin Seilmayer and Matthias Ratajczak

Helmholtz-Zentrum Dresden-Rossendorf, Bautzner Landstraße 400, 01328 Dresden, Germany

Correspondence should be addressed to Martin Seilmayer; m.seilmayer@hzdr.de

Received 2 January 2017; Accepted 10 April 2017; Published 24 July 2017

Academic Editor: Zhichun Yang

Copyright © 2017 Martin Seilmayer and Matthias Ratajczak. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper provides an overview about the usage of the Fourier transform and its related methods and focuses on the subtleties to which the users must pay attention. Typical questions, which are often addressed to the data, will be discussed. Such a problem can be the origin of frequency or band limitation of the signal or the source of artifacts, when a Fourier transform is carried out. Another topic is the processing of fragmented data. Here, the Lomb-Scargle method will be explained with an illustrative example to deal with this special type of signal. Furthermore, the time-dependent spectral analysis, with which one can evaluate the point in time when a certain frequency appears in the signal, is of interest. The goal of this paper is to collect the important information about the common methods to give the reader a guide on how to use these for application on one-dimensional data. The introduced methods are supported by the `spectral` package, which has been published for the statistical environment *R* prior to this article.

1. Introduction

Because the field of interest is as wide as the amount of information about spectral methods, it becomes necessary to compose this article out of the most important methods. It is split into three main parts, which point at different levels of abstraction and complexity. Section 2 will give a brief and short overview of the mathematical principles and ideas behind the common methods of spectral analysis. For further reading, the Appendix contains a selection of derivations. Remember, the intention of this work is not the completeness of mathematical proofs. Its focus lies on the application of methods on discrete data and the mistakes to be avoided. Section 3 will focus on simple one-dimensional methods, their properties, advantages, and drawbacks.

The individual sections are supported by examples which are programmed in the statistical language *R* [1] with the help of Seilmayer's `spectral` [2] package which has been published prior to this article. The main goal of the provided *R* code examples and the introduced package is an easy-to-use user interface which computes scaled output data according to the input. This means if the input time vector would be in units of seconds the output of the `spec.fft(x, y)`-function

would be the natural frequency unit Hz. Besides the analysis of data sets it will be shown how spectral methods can be utilized to carry out (or even accelerate) several calculations on discrete data, like derivation, integration, or folding. Here, source code snippets of the `spectral` package highlight the implementation and try to underline the methodology.

2. Mathematical Concepts

This chapter introduces the basic ideas and principles which lead to the methods of spectral analysis for discrete data. Thereby, the focus lays on measurements and time series of physical processes. For now, utilizing the time as variable makes it easier to understand the explained methods. Of course, the time variable t might be replaced by the space variable x , if a distribution of spatial events is of interest. So in this sense space and time are meant to be the *location* where the signal is defined.

The following definitions and explanations are short and sweet. For further reading, it is suggested that you refer to “The Scientist and Engineer’s Guide to Digital Signal Processing” by Smith [3] or to “Time-Frequency Analysis” by Cohen [4]. These two comprehensive works elaborate

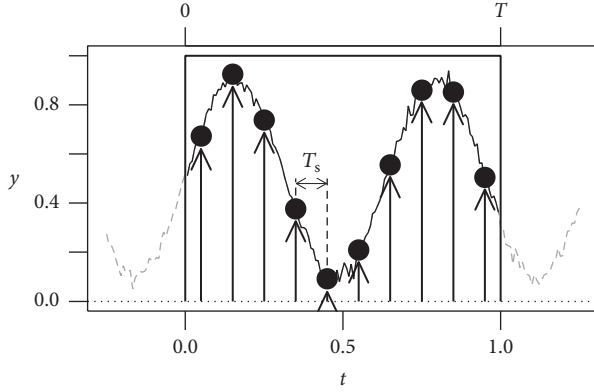


FIGURE 1: The arbitrary signal with overlapped normal distributed noise $p(t) = 0.4 \cos(2\pi \cdot 1.5t - \pi/2) + \mathcal{N}(0, 0.02) + 0.5$ is shown here. It is only within the rectangular window that the measurement takes place and the signal $s(t)$ can exist. Outside this window the physical process might go on but is not recognized anymore. The points and arrows mark the sampling of that signal.

the underlying mathematical framework in detail. Both books are highly recommended if theoretical approaches and mathematical subtleties are of interest. But from now on the practical application to measured data and understanding of the complex connections is of interest.

2.1. Signal Definition. In the context of this article, a signal $s(t)$ corresponds to a physical measure $p(t)$ and is therefore real-valued and causal. This means that with the measurement of the process $p(t)$ the signal $s(t)$ starts to exist at a certain point in time and ends later when the measurement is finished. With that in mind, the signal function $s(t)$ represents a slice of the length T for times $t > 0$. This can be properly defined as follows:

$$s(t) = \begin{cases} p(t) & 0 < t \leq T, \\ 0 & \text{elsewhere.} \end{cases} \quad (1)$$

The simple noisy signal in Figure 1 illustrates that. For this example, the underlying physical process $p(t)$ could be the temperature, which is measured in the time range of $0 < t \leq 1$. Evidently the temperature of an object exists before and after the measurement takes place, so the content of $s(t)$ only maps to a time interval of this process.

With respect to a digital representation of information, the measurement of $p(t)$ takes place by acquiring data in the form of sampling the continuous signal at N different instances in time. The resulting sampled signal series

$$\hat{s}_n = \sum_{n=0}^{N-1} s(n \cdot T_s) \cdot \vec{e}_n \quad (2)$$

is the data vector \hat{s}_n . Here, the index n addresses the n th element in this N -dimensional data vector \hat{s} . In (2), the basis vector \vec{e}_n ensures that \hat{s} becomes a vector in a mathematical sense, which simply represents an ordered list of elements. Remember the absolute value $|\vec{e}_n| = 1$ is always equal to one,

so the sum only helps to iterate along the continuous signal $s(t)$.

2.2. The Fundamental Idea of Frequency Analysis. To explain the idea of frequency analysis right from the beginning, it is necessary to redefine the signal function $s(t)$. From now on, the signal represents a sum of sinusoidal functions in the form:

$$s(t) = \sum_i A_i \cdot \cos\left(\frac{2\pi f_i t}{\omega_i} + \varphi_i\right), \quad (3)$$

which, in the final consequence, leads to the expansion into orthogonal functions.

The aim of frequency analysis is to calculate the amplitude and phase spectrum of a signal. The result of this procedure is a mapping of amplitudes to their corresponding frequencies. If a measured time series of a physical process is the sum of many individual sinusoidal functions, it could be of interest to estimate each of the amplitudes A_i and phases φ_i for a given frequency f_i .

The first approach to do would be to estimate each A and φ by a least square fit against a sinusoidal model for each frequency of interest. This is a legitimate attempt with certain drawbacks and some outstanding advantages, which are discussed in Section 3.8.

The roots of a more general method to determine the spectral components of a signal are the two common addition theorems:

$$\begin{aligned} \cos \alpha \cdot \cos \beta &= \frac{1}{2} (\cos(\alpha - \beta) + \cos(\alpha + \beta)), \\ \cos \alpha \cdot \sin \beta &= \frac{1}{2} (\sin(\beta - \alpha) + \sin(\alpha + \beta)), \end{aligned} \quad (4)$$

which mark the starting point of the following explanations. A collection of addition theorems can be found in almost any book of formulae and tables like Bronštejn and Semendjajew [5].

Given a signal in the continuous domain

$$s(t) = A \cdot \cos(\omega t + \varphi), \quad (5)$$

then multiplying this with $\cos(\omega_0 t)$ as well as $\sin(\omega_0 t)$, and finally integrating over all times yield, for instance, for the cos-term

$$\begin{aligned} &\int_{-\infty}^{\infty} s(t) \cdot \cos(\omega_0 t) dt \\ &= \frac{A}{2} \left(\int_{-\infty}^{\infty} \cos((\omega - \omega_0)t + \varphi) dt \right. \\ &\quad \left. + \underbrace{\int_{-\infty}^{\infty} \cos((\omega + \omega_0)t + \varphi) dt}_{=0} \right) \end{aligned} \quad (6)$$

$$= \begin{cases} \frac{A}{2} \cos(\varphi), & \omega = \omega_0 \\ \rightarrow 0, & \omega \neq \omega_0, \end{cases} \quad (7)$$

$$= R \text{ (real part or inphase component)}, \quad (8)$$

which is going to be named R . Herein, the variable ω_0 denotes the frequency of interest. The sin-term is then very similar

$$\int_{-\infty}^{\infty} s(t) \cdot \sin(\omega_0 t) dt = \begin{cases} \frac{A}{2} \sin(\varphi), & \omega = \omega_0 \\ \rightarrow 0, & \omega \neq \omega_0, \end{cases} \quad (9)$$

$$= I \text{ (imaginary part or quadrature component)}$$

and is called I . In both equations on the right hand side, the right integral vanishes to zero, because each positive area element of a sin-function can be mapped to a negative area element. Only in the case of $\omega_0 = \omega$, a phase-dependent part is left.

With the identity $\sin^2 \alpha + \cos^2 \alpha = 1$, the amplitude can be calculated as

$$A = 2\sqrt{R^2 + I^2}. \quad (10)$$

The phase information is then gained by

$$\varphi = \arctan\left(\frac{I}{R}\right). \quad (11)$$

The introduced procedure is called quadrature demodulation technique (QDT). Finally, by evaluating the amplitude A at many different frequencies ω_0 , a periodogram emerges. Note the R and I part can also be interpreted as the real and imaginary part of a signal. The complex exponential formulation then corresponds to

$$s(t) = \Re e\left(A \cdot e^{i\omega t + \varphi_i}\right) \quad (12)$$

with the identity $e^{i\phi} = \cos \phi + i \sin \phi$.

One of the main disadvantages arises for finite time ranges $0 < t \leq T$. In that case, this method produces a truncation error ε , which depends on the remainder after the modulus division

$$n = (2\pi f \cdot T) \bmod (2\pi) \quad (13)$$

with respect to the total time interval T . In other words, if the period $2\pi/\omega$ of the signal does not fit completely in the integration range, the integrals containing $\omega + \omega_0$ will not vanish completely. However, remember the right term of (6).

Here the integral can be split into an $n \cdot 2\pi$ periodic part and a residue. The periodic integral of $\cos((\omega + \omega_0)t + \varphi)$ vanishes,

$$\begin{aligned} R &= \frac{A}{2} \left(\int_0^T \cos((\omega - \omega_0)t + \varphi) dt \right. \\ &\quad \left. + \int_0^T \cos((\omega + \omega_0)t + \varphi) dt \right) \Big|_{\omega=\omega_0} \\ &= \frac{A}{2} \left(\cos(\varphi) \right. \\ &\quad \left. + \underbrace{\int_0^{(n \cdot 2\pi - \varphi)/(\omega + \omega_0)} \cos((\omega + \omega_0)t + \varphi) dt}_{\rightarrow 0} \right. \\ &\quad \left. + \underbrace{\int_{(n \cdot 2\pi - \varphi)/(\omega + \omega_0)}^T \cos((\omega + \omega_0)t + \varphi) dt}_{=\varepsilon} \right) \\ &= \frac{A}{2} (\cos(\varphi) + \varepsilon), \end{aligned} \quad (14)$$

whereas a residue ε remains in the second integral from the last $n \cdot 2\pi$ period up to the full length T . Obviously, a minimum of ε is reached if $T = n \cdot 2\pi$. All this holds if $\omega = \omega_0$. If this is not the case, also the integral $\int \cos((\omega - \omega_0)t + \varphi)$ splits into two parts from 0 to $n \cdot 2\pi$ and a residue, which leads to a second error ε_2 . Finally, with $\omega \approx \omega_0$ this leakage effect will produce artificial amplitudes in the nearest neighborhood of ω_0 . Anyway, note that, from inserting the result (14) in (10) or (11), mixed products arise, which makes it difficult or even impossible to reject the error ε . All this is far away from being mathematically complete, but it points in the direction towards an error discussion, which is presented in more detail by Jerri [6, Chap. 6].

At this point, the statistical approach “fitting a sinusoidal model to the data set” becomes attractive. The nature of a fitting algorithm enables the minimization of the phase error which leads to an optimal result. A brief discussion of this concept is given in “Studies in Astronomical Time Series Analysis. II” by Scargle [7]. Section 3.8 provides an introduction to the Lomb-Scargle method, which gives equal results compared to the already mentioned least square fit.

2.3. Integral Transforms. In contrast to the previously described QDT, which is more or less a statistical approach, the integral transforms are the mathematical backbone of spectral analysis. An integral transform represents the bijective mapping $s(t) \leftrightarrow S(\omega)$ between the time and frequency domains, which states that all information present in the first will be transformed to the second domain. In the following, capital letters denote always the frequency domain, whereas lowercase letters represent the time domain of the corresponding variable.

2.3.1. Fourier Transform. First of all, the Fourier transform should be introduced. Thereby, the definitions and some of

TABLE 1: Continuous and discrete Fourier transform.

	Continuous	Discrete
Transform	$S(\omega) = \int_{-\infty}^{\infty} s(t) \cdot e^{-i\omega t} dt$	$\hat{S}(m) = \frac{1}{N} \sum_{n=0}^{N-1} \hat{s}(n) e^{-im(2\pi/N)n}$
Back transform	$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \cdot e^{i\omega t} d\omega$	$\hat{s}(n) = \sum_{m=0}^{N-1} \hat{S}(m) e^{im(2\pi/N)n}$

the properties in the continuous and discrete time domain are different. A comparison is given in Table 1. Nevertheless, the signal $s(t)$ forms a Fourier pair with $S(\omega)$. The following notation describes a transform of the signal $s(t)$ into the frequency domain,

$$\mathcal{F}(s(t)) = S(\omega). \quad (15)$$

This uses the Fourier operator \mathcal{F} , which expresses the calculus according to Table 1. The inverse back transform is denoted by \mathcal{F}^{-1} . The Fourier transform gains different features like the frequency shifting

$$\mathcal{F}(e^{ibt} f(t)) = F(\omega - b) \quad b \in \mathbb{R}, \quad (16)$$

which becomes important when amplitude modulation in Section 3.4 is discussed. All necessary properties are listed in Appendix B.

In addition to the characteristics of the continuous Fourier transform, the discrete Fourier transform (DFT) has a 2π -periodic spectrum. As shown in Appendix C, the result of the DFT is a complex valued data vector, in which the first element \hat{S}_1 *always* contains the mean value of the data series. Subsequent bins hold the amplitude values up to $\hat{S}_{N/2}$. From this position on the spectrum is mirrored and repeats up to the sampling frequency. In Section 3, the symmetry of the DFT and its consequences are discussed in more detail. How to take advantage of this to recover the “negative” frequencies will be shown. In case of more dimensional data, it turns out that the result of the DFT contains additional information, where also the “negative” frequency components play a role.

2.3.2. Hilbert Transform. The Hilbert transform (HT) is defined by the real-valued folding operation on the real-valued signal $s(t)$

$$\mathcal{H}(s(t)) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{s(t')}{t - t'} dt'. \quad (17)$$

Additionally, the linear operator \mathcal{H} is introduced to execute the Hilbert transform on a function or data set. From the representation in the Fourier space, the constant phase shifting feature becomes clear,

$$\mathcal{F}(\mathcal{H}(s(t))) = -i \cdot \text{sign}(\omega) \cdot \mathcal{F}(s(t)). \quad (18)$$

Hence, (18) can be used to easily calculate $\mathcal{H}(s(t))$ in terms of a Fourier transform. Remember, here the $\text{sign}(\omega)$ function is defined on $-\infty < \omega < \infty$. This must be taken into account,

when performing $\mathcal{H}(\hat{s}(n))$ on discrete data. In this case, the discrete Hilbert transform is calculated by

$$\mathcal{F}(\mathcal{H}(\hat{s}_n)) = \begin{cases} i \cdot \mathcal{F}(\hat{s}_n) & 0 \leq n < \frac{N}{2} \\ -i \cdot \mathcal{F}(\hat{s}_n) & \frac{N}{2} \leq n < N. \end{cases} \quad (19)$$

This formalism considers the fact that “negative” frequencies are mirrored into the upper parts of the DFT data vector. However, the result of $\mathcal{H}(s(t))$ is real-valued, with each frequency component phase-shifted by $\pm\pi$. In conjunction with a real-valued and causal signal $s(t)$, the HT helps to represent the real physical character of measured data, which should only contain positive frequencies, due to real-world processes. From this, the analytical signal,

$$a(t) = s(t) + i\mathcal{H}(s(t)), \quad (20)$$

follows, which is a complex representation of $s(t)$ containing a one-sided Fourier spectrum.

One derivation of the Hilbert transform in conjunction with the analytical signal is given in Appendix D and its application in terms of data filtering is briefly discussed in Section 3.3 and the following.

In addition to that, the concept of instantaneous frequency can be explained with the help of the HT. This topic is discussed in detail in “The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis” by Huang et al. [8]. For further reading on that “Amplitude, Phase, Frequency—Fundamental Concepts of Oscillation Theory” by Vakman and Vainshtein [9] should be mentioned.

3. Spectral Methods in One Dimension

The application of the introduced Fourier and Hilbert transform will be discussed in detail in this chapter. Each of the following sections includes a theoretical part, which is finally supported by examples. The focus lies on the application and fundamental understanding of the corresponding methods.

First of all, let the signal function

$$s(t) = \sum_{i=0}^m A_i \cdot \cos(\omega_i t + \varphi_i) + \mathcal{N}(0, \sigma) \quad (21)$$

be a set of sinusoidal functions plus an optional noise term. This signal is sampled with an equidistant spacing T_s , so the discrete data vector

$$\hat{s}_n = \sum_{n=0}^{N-1} s(n \cdot T_s) \cdot \vec{e}_n \quad (22)$$

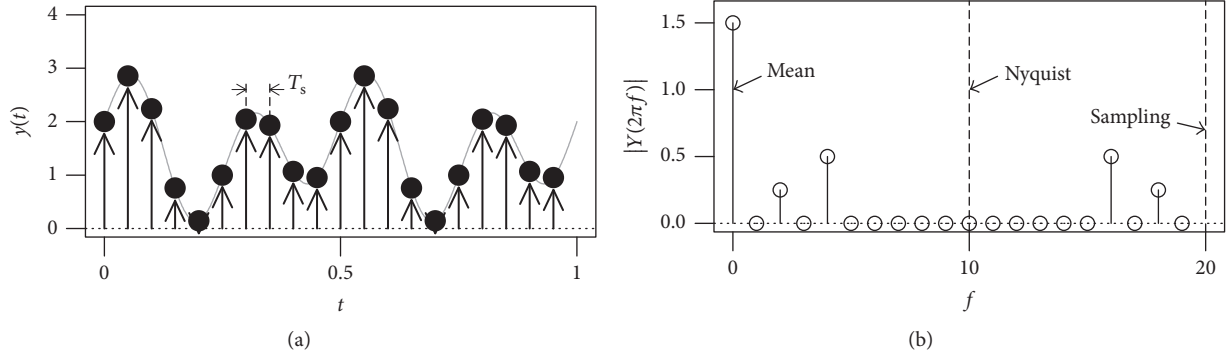


FIGURE 2: Oversampled signal $y(t)$. Band limitation is achieved by definition of the signal. Mention that the first sample and the $i_{th} + 1$ sample at $t = 1$ would be the same, because of the periodicity of the signal.

of length N represents the sampling series. Using the discrete Fourier transform (see Table 1), the spectrum of the data vector \hat{s}

$$\hat{S}_m = \mathcal{F}(\hat{s}_n) \quad (23)$$

can be calculated.

For further reading, a brief introduction can be found in “The Shannon Sampling Theorem—Its Various Extensions and Applications: A Tutorial Review” by Jerri [6]. A detailed comprehensive mathematical work is given in the book “Fourier Analysis” by Duoandikoetxea Zuazo [10].

3.1. The Essence of Band Limitation and the Nyquist Condition.

The question of band limitation is a question of the unique representation of a function $s(t)$ in the frequency space $S(\omega)$. In the simplest case, a band limited signal has a spectrum

$$S(\omega) = \begin{cases} \neq 0 & |\omega| \leq \omega_{\max} \\ 0 & \text{elsewhere} \end{cases} \quad (24)$$

which is only nonzero within the range $|\omega| \leq \omega_{\max}$. Outside this interval, the spectrum is equal to zero. The sampling theorem by Shannon [11] and the proof of Whittaker [12] state that in such a situation even a time-limited continuous function is completely defined by a discrete and finite set of samples, if all the samples outside the time range are exactly zero (see the exemplary function in Figure 1). Closely related to that is the minimal required sampling frequency (twice the “Nyquist frequency”) $1/T_s = f_s \geq 2f_{\max}$, which is necessary to obtain all information about the signal.

The consequence is that there should not be any frequency present in the data set higher than $f_s/2$ to achieve a unique sampling in application. Theoretically, this means a function $s(t)$ is completely determined by its discrete spectrum $S(n \cdot 2\pi f_s)$ only if the signal is bandlimited to $|f| < f_s/2$. Therefore the discrete spectrum of the discrete sampled signal \hat{s}_n provides a unique spectral representation of its continuous counterpart $s(t)$.

3.1.1. Application. The artificial function

$$y(t) = \sin(2\pi \cdot 4t) + 0.5 \cdot \cos(2\pi \cdot 2t) + 1.5 \quad (25)$$

is given in Figure 2(a). It consists of two functions and an offset value. The sampling took place at a rate of $T_s = 0.05$, so that there are theoretically $N = 21$ possible samples. Note $y(t)$ is perfectly periodic, so the first sample would equal the last sample. Because the latter marks the first point of the next sampling period, it must be removed from the data set, which now has $N^* = 20$ elements. Ignoring this fact would lead to an error (leakage effect) with the result that the elements to the left and to the right of the frequency peaks are not zero anymore. The root of this behavior is discussed in Section 2.2 and becomes visible in the subsequent figures in this section. The resulting frequency resolution in this example is

$$\delta f = \frac{1}{T_s N^*} = 1, \quad (26)$$

so the signal’s frequencies fit perfectly into this grid. The absolute amplitude spectrum $|\hat{Y}_n|$ is given in Figure 2(b). Here, one can see that the mean, saved in the first bin, has its original value of $A_0 = 1.5$, whereas the magnitude of the amplitudes at $f_{n=2} = 2$ and $f_{n=4} = 4$ is split into the upper and lower frequency domain. To retrieve the correct signal amplitudes the individual values in both half-planes must be added. Remember the DFT is mirror-symmetric to the Nyquist frequency $f_s/2 = 10$ so that the first amplitude A_1 becomes

$$A_1 = |\hat{Y}_{n=4}| + |\hat{Y}_{n=16}|. \quad (27)$$

Figures 2 and 3(a) display an oversampled signal, where the required Nyquist condition is fulfilled. The result is a unique mapping of the time domain into frequency domain. In other words, the corresponding frequency vector \hat{f}_n is only valid for $0 \leq n < N/2$, because it repeats in the inverse order for $N/2 \leq n < N$.

However, as Figure 3(c) indicates, if the requirement of band limitation ($f_s \geq 2f_{\max}$) is violated, the mapping is not unique anymore. Remember the frequency resolution in all given examples is $\delta f = 1$. In comparison to Figure 3(b), where the condition of band limitation is complied with, Figure 3(d) illustrates that the upper and lower parts of the frequency bands now infiltrate each other. This makes it even more complicated to distinguish between certain frequencies and

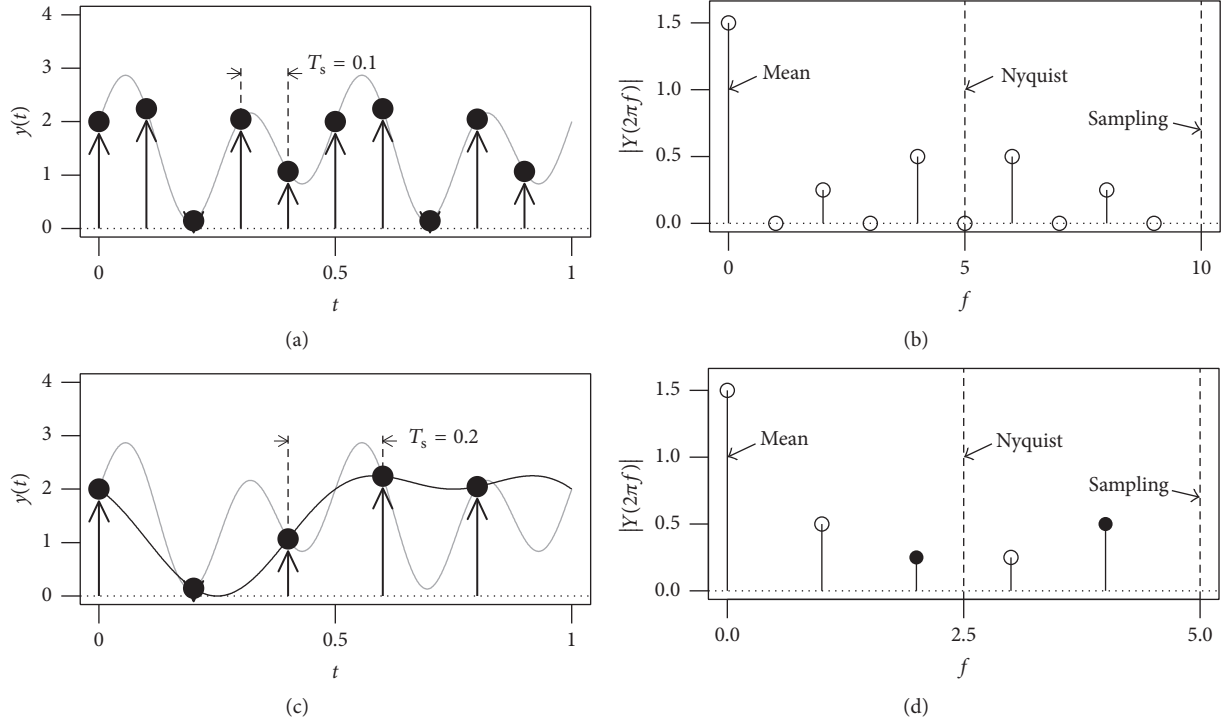


FIGURE 3: (a, b) The signal $y(t)$ is sampled marginally. In (c, d), it is undersampled so the upper and lower parts of the spectrum infiltrate each other. Compare the filled symbols, which correspond to the points in the lower half plane of (b). The black line in (c) indicates the alternative reconstruction.

their physical correspondence. In such a scenario, it becomes completely unclear which of the two functions belong to the sampling points.

3.1.2. Example: Undersampling. The conditions described above can be extended to the requirement

$$f_{\max} - f_{\min} = \Delta f \leq f_s, \quad (28)$$

which means that a unique mapping is still possible, yet only if the bandwidth Δf is not larger than the sampling frequency. In addition to this, the frequency range must be known. This is automatically fulfilled in the most common case, whenever the frequency ranges within $0 \leq f \leq f_s$.

The example given in Figure 4 shows the artificial function

$$y(t) = \cos(2\pi \cdot 25 \cdot t) + 0.5 \cdot \sin(2\pi \cdot 27 \cdot t) + 1.5, \quad (29)$$

which is sampled with a sampling frequency of $f_s = 20$. The amplitude spectrum is given in Figure 4(b). Obviously, $y(t)$ is undersampled. Remember the discrete Fourier transform calculates a periodic spectrum, which projects the high-frequency parts into the lower frequency range between 0 and f_s . Without any additional information, a mapping from spectral domain to time domain will fail. In the present case, we assume that the signal frequency is in the range $25 \leq f \leq 27$, so the whole spectrum can be shifted by an offset of $f_o = f_s$. The second f -axis indicates this. The upper and lower frequency bands can be clearly distinguished, so the

mapping is unique. In contrast to that, the interpenetration of the upper and lower bands prevents such a mapping in the example in Figure 3(d).

3.1.3. Example: The Derivative of a Function. One important property of the Fourier transform is the change of mathematical operations in the spectral domain. According to Appendix B, taking the derivative of a function reduces to a multiplication with $i\omega$

$$\mathcal{F}\left(\frac{d}{dx}y(x)\right) = i\omega \cdot \frac{\mathcal{F}(y(x))}{Y(\omega)} \quad (30)$$

in the spectral domain. In this manner, differential equations can be reduced to simpler linear algebraic representations.

In contrast to the continuous case, discrete sampled functions must be periodic with a finite bandwidth to fulfill the requirements for the DFT. In case the function is nonperiodic within the sampled window, the application of a proper window function is necessary to minimize leakage. The following code explains how to estimate the derivative of the non-band-limited function like

$$y(x) = -x^3 + 3 \cdot x, \quad (31)$$

with $\frac{d}{dx}y(x) = -3 \cdot x^2 + 3$

which is going to be sampled with $N = 40$ discrete points.

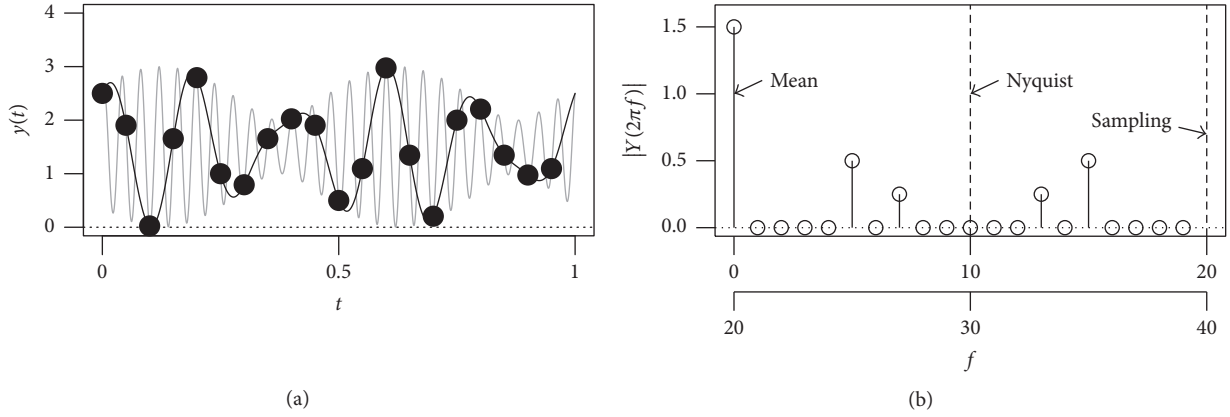


FIGURE 4: A signal with a limited bandwidth is sampled times slower than the signal frequency. The black line in (a) displays again the second possible curve, which would map to the spectrum in (b).

```
require(spectral)
# 40 sample positions
x <- seq(-2.5,2.5,length.out = 40)
# window the function and generate
# the sampling points
y <- win.tukey(x,0.2) * (-x^3+3*x)

# doing the DFT
Y <- spec.fft(y,x,center = T)
# calc deriv.
Y$A <- 1i * 2*pi * Y$fx * Y$A
# the realpart of the back transform
# contains the signal of interesst
dy <- Re(spec.fft(Y,inverse = T)$y)
```

It must be mentioned that the function $y(x)$ has to be windowed in advance to achieve the periodic property. In this example, this is done via the “Tukey-” window, which rises and falls with a quarter cycle of a cosine function (compare dashed curve). Figure 5 shows the output of the listed code above. Pay special attention to the windowed version of $y(x)$ (black curve) and its almost perfect derivative. Here, the effect of the windowing process is to achieve $Y(|\omega| > 0.5 \cdot \omega_s) = 0$, which relaxes the function at the right and left side. In consequence, the resulting derivative equals the analytic solution in the mid-range but also deviates at the boundaries. In comparison to that, the unweighted function in Figure 5(a) (gray line) produces an unusable faulty derivative in Figure 5(b) (crosses). The reason for that is the discussed missing band limitation.

With a look into line 7 of the above code, notice that the additional parameter `center = T` is provided to the `spec.fft()` function. The effect is a zero-symmetric spectrum with the corresponding frequencies ranging within $-f_s/2 \leq f < f_s/2$, which enables the direct computation of $i\omega \cdot Y(\omega)$ utilizing the elements of the list variable `Y`.

Remember a pure DFT would map to frequencies in the range of $0 \leq f < f_s$. It follows that the negative frequencies $-f_s/2 \leq f < 0$ were projected into the upper half of the frequency domain $f_s/2 \leq f < f_s$. In case of `center = F`, the latter frequency range would be generated and used in further calculations, which must be considered to prevent wrong results.

3.2. Center a Discrete Spectrum. As initially explained in Section 2.3, the spectrum of a continuous function $s(t)$ is defined in the range of $-\infty < \omega < \infty$. In contrast to that, its counterpart the discrete Fourier transform produces a 2π -periodic spectrum, which is defined for a positive number of samples $0 \leq n < N$ and $0 \leq m$ frequencies. The negative frequency components are projected into the range of $f_s/2 \leq f < f_s$, so the spectrum becomes mirror-symmetric with respect to the Nyquist frequency $f_s/2$.

Figure 6(a) shows the result of the continuous Fourier transform applied to the artificial example function (25):

$$y(t) = \sin(2\pi \cdot 4t) + 0.5 \cdot \cos(2\pi \cdot 2t) + 1.5. \quad (32)$$

It is clear that the mean value $\overline{y(t)}$ is located at $f = 0$ and the sinusoidal components are located symmetrically around zero. In contrast to that, the discrete Fourier transform produces a data vector \hat{Y}_m , to which only positive discrete frequencies can be assigned. To convert the data vector into a zero-symmetric representation, the function $y(t)$ must be modulated by $f_s/2$:

$$y^*(t) = y(t) \cdot e^{i\pi f_s t}. \quad (33)$$

This method utilizes the shifting or damping property of the Fourier transform as stated in Appendix B (B.4), assuming a damping factor of $a = 1$. By doing so, the complete spectrum (Figure 6(b)), including the mean value, shifts towards the half of the sampling frequency right in the middle, as it is indicated in Figure 6(a).

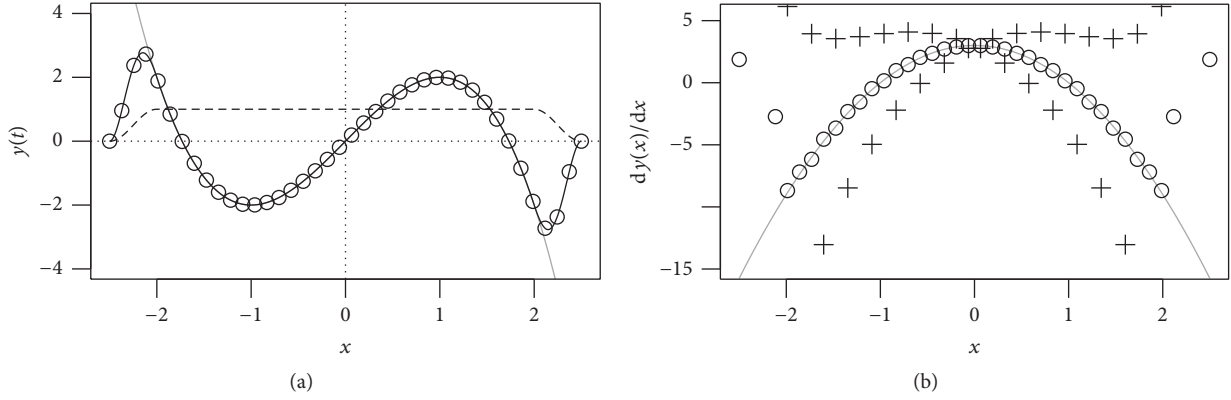


FIGURE 5: Calculating the derivative of $y(x)$. (a) displays the original function (gray), the windowed function (black), and the sample points. The dashed line defines the window function. In (b), the analytic (gray line) and estimated numeric (points) derivative $dy(t)/dx$ is shown. The crosses indicate the result of the calculation without utilizing the window function. Note that some crosses are vertically outside the window for $x < -2$ and $x > 2$.

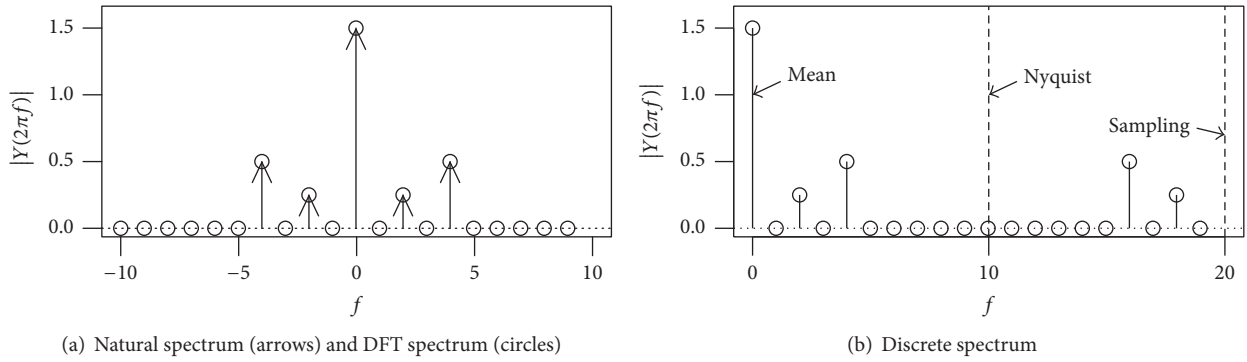


FIGURE 6: Two spectra of function (25). The difference between the natural spectrum in (a) and the normal discrete Fourier transform in (b) is the symmetry. In (a), there is a symmetric spectrum around zero, whereas in (b) the point of symmetry is $f_s/2$. The dots in panel (a) indicate the result of the DFT of the modulated sampling series of (25). Note the exceptional first bin in (b), where the mean value is stored.

3.2.1. *Application.* In an equally spaced sampling series, the discrete time vector $t = n \cdot T_s$ changes (33) to

$$y^*(n \cdot T_s) = \hat{y}_n \cdot e^{i\pi f_s n T_s} = \hat{y}_n \cdot \underbrace{e^{i\pi n}}_{1-1+1-\dots} = \hat{y}_n \cdot (-1)^n \quad (34)$$

and simplifies it. From (34), it follows that the modulation of the data vector can be computed easily by a multiplication of the alternating series $(-1)^n$. Subsequently, all this can be extended into n dimensions, so even 2D spectra can be centered [13, p. 154].

The following R-code which is part of the `spec.fft()` function explains the method. First, assume a vector x for the spatial location where the measurements y were taken.

```
nx <- length(x)
ny <- length(y) # is equal nx
# calculate the sampling
Ts <- min(diff(x))
# modulate the data vector
y <- y*(-1)^(1:ny)
```

```
# determine the corresponding
# frequency vector
fx <- seq(-(nx)/(2*Ts * nx)
          , (nx - 2)/(2*Ts * nx)
          , length.out = nx)
# calculate and normalize the spectrum
A <- fft(y)*1/ny
```

In the calculation of the frequency vector fx , the maximum frequency is $(nx - 2)/(2*Ts * nx)$, in which $(nx - 2)$ considers the position of $f = 0$ and the intrinsic periodicity of the signal y . Remember the examples in Section 3.1, when the last possible sampling point matches the first point of the next signal period.

The back transform of the centered spectrum is the simple reverse of the described procedure. The function `spec.fft()` supports both methods to decompose a data vector into its centered or noncentered spectral representation. The usage of the code is explained below.


```

# defining the sampling
Ts <- 0.05
x <- seq(0,1,by = Ts)
# remove last sample to avoid
# error with periodicity
x <- x[-length(x)]
y <- function(x) sin(2*pi * 4*x) +
  0.5 * cos(2*pi * 2*x) + 1.5

# the normal fft()
Y <- spec.fft(y(x), x, center = F)
# prior modulation
Yc <- spec.fft(y(x), x, center = T)
# spec.fft() returns a list object
# containing the following:
# Y$fx, (Y$fy), Y$A, Y$x, Y$y, (Y$z)

```

3.3. The Analytic Signal. The concept of the analytic signal is based on the idea that a real-valued signal, for example, some measured data, can only contain positive frequencies. At least in the 1D case, the interpretation of negative frequencies is almost impossible. Next to that, a real-world signal does only exist for positive times, usually. This makes the signal causal, which also must be taken into account; see (D.2) in Appendix D. The foundation of the analytic signal is the Hilbert transform. For a short introduction on the Hilbert transformation itself and its properties, refer to Section 2.3.2. Besides that, the attribute “analytic” originates from the Cauchy-Riemann conditions for differentiability, which is fulfilled for analytic functions in this sense [4].

Now, let us introduce the analytic signal

$$a(t) = s(t) + i\mathcal{H}(s(t)). \quad (35)$$

It is formally defined by the sum of a signal $s(t)$ and its Hilbert transform $i\mathcal{H}(s(t))$ multiplied by the imaginary unit. So the real-valued signal $s(t)$ transforms to $a(t)$, which now has an one-sided spectrum. Frequencies above the Nyquist frequency $f_s/2$ are coerced to be zero and frequencies below that are gained by the factor of two, so that energy is conserved. This can be seen in Figure 7, which shows the result of a DFT applied on the analytic representation of the sampled signal from Figure 2. Starting from that several applications can be derived.

First, all the calculations which utilize the signal’s frequencies, for example, the estimation of the derivative introduced in Section 3.1.3, get simplified with respect to discrete data because the distinction of frequencies above and below $f_s/2$ can be omitted. The example of the derivative in Section 3.1.3 instead utilized the discrete *centered* spectrum, which provides a frequency vector with positive and negative parts, which can be multiplied directly to the spectrum.

Second, the imaginary part of $a(t)$ equals the Hilbert transform of its real part. The Hilbert transform’s phase

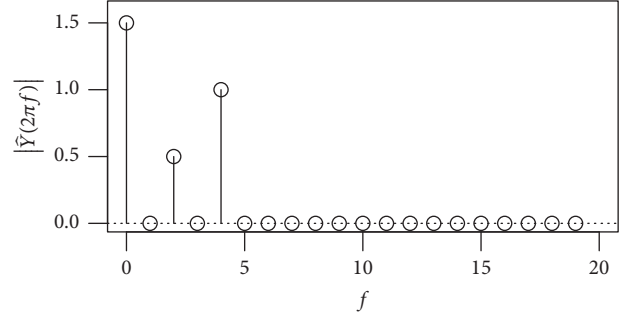


FIGURE 7: Single sided spectrum of the analytic signal of $y(t)$.

shifting properties form the basis to calculate, for instance, the envelope function, which is discussed in the next section. The analytic signal also provides a way to estimate the instantaneous frequency of a signal. The latter ends up in the empirical mode decomposition, which gives a time depending spectral decomposition according to [8], which is out of scope of this guide. In Section 3.7, it will be shown how to utilize the Fourier transform and how to overcome its drawbacks in conjunction with the analytic signal of data of a nonstationary process.

3.3.1. Implementation. According to Appendix D, the function `analyticFunction(y)` provided by the `spectral` package calculates the analytic signal. An example is given in Figure 7, which displays the analytic representation of the sampled artificial example function (25) from Section 3.1. Here the spectrum is single sided and all the amplitudes have their correct value. Note that all components above $f_s/2$ are zero, because the upper half of the spectrum is projected into the lower half by this method. In consequence, the illustration in Figure 7 is in contrast to the given examples (e.g., Figure 6) above, where individual amplitudes only contain the half of the true value.

The program code

```

# normalized DFT-spectrum
X <- fft(x) / length(x)
# synthetic spatial vector
f <- 0:(length(X) - 1)
# shifted by half the length
# so 0 is in the middle
f <- f - mean(f)

# Hilbert transform
X <- X * (1 - sign(f - 0.5))
# correct mean value
X[1] <- 0.5 * X[1]

ht <- fft(X, inverse = T)

```

explains how the analytic signal is calculated. Here the part $(1 - \text{sign}(f - 0.5))$ solves three things. First, this statement shifts the phase with respect to the sign of the virtual frequency vector f , whereby the “negative” frequencies are located in the upper half of the data set. This provides a better solution than (19) because the evenness of the data sets length does not matter anymore. Second, since f is an integer vector, the subtraction $f - 0.5$ circumvents the problem $\text{sign}(0) = 0$, which avoids an error with even length data sets at $f = 0$. And third, the above code calculates the Hilbert transform and the analytic signal in one step in the spectral domain by combining (18) and (33).

The code below produces the single sided spectrum Y given in Figure 7. Note that the amplitudes correspond to the real input values of the function (25) and all spectral parts above $f_s/2$ are zero.

```
x <- seq(0, 1, by = 0.05)
x <- x[ -length(x) ]

y <- function(x) sin(2*pi * 4*x) +
  0.5 * cos(2*pi * 2*x) + 1.5
Y <- spec.fft( analyticFunction(y(x))
  , x, center = F )
```

3.4. Calculating the Envelope. Sometimes it becomes necessary to calculate the envelope function of data. Different approaches are possible, for instance, finding all maxima and then fitting a spline function to these points. However, in the following, the spectral way to do that is introduced.

First, remember the trigonometric identity

$$\sin^2(x) + \cos^2(x) = 1, \quad (36)$$

which becomes the key component of the calculation later. Next to that, assume a function

$$y(t) = A(t) \cdot \cos(\omega_0 t) \quad (37)$$

with $A(t)$ as the envelope function, which is modulated with the carrier $\cos(\omega_0 t)$. It is clear now that calculating the envelope is equivalent to an amplitude demodulation.

Now, the amplitude function is gained by

$$\begin{aligned} A(t) &= \sqrt{y(t)^2 + y^*(t)^2}, \\ &= A(t) \cdot \underbrace{\sqrt{\cos^2(\omega_0 t) + \sin^2(\omega_0 t)}}_{=1}, \end{aligned} \quad (38)$$

where $y^*(t)$ denotes the signal $y(t)$ phase-shifted by $\pi/2$, so that $y^*(t)$ equals the Hilbert transform

$$y^*(t) = A(t) \cdot \sin(\omega_0 t) \quad (39)$$

$$= \mathcal{H}(y(t)). \quad (40)$$

Under the condition of *sufficient* band limitation, the statement above can be expressed as

$$A(t) = \sqrt{y(t)^2 + \mathcal{H}(y(t))^2}, \quad (41)$$

which works, because (40) acts like an ideal phase shifter as discussed in Section 2.3 and Appendix D.

3.4.1. Application. The calculation is straightforward and follows (41). The `spectral` package includes the function `envelope()` to perform the calculation of 1D envelope. The only problem with this is the band limitation, which is necessary to achieve reasonable results. Compared with Figure 8(b), here the required bandwidth to demodulate the envelope becomes clearly visible in the negative and positive half plane.

3.5. Convolution. The Fourier transform not only maps a spatial function into the spectral domain, but also converts several mathematical operations. One example is the derivation of

$$\frac{df(t)}{dt} \longleftrightarrow i\omega F(\omega), \quad (42)$$

which equals a simple multiplication with the complex variable $i\omega$ in the spectral domain.

In the following, the convolution $\int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$ will be introduced by the identities

$$\begin{aligned} \mathcal{F}\left(\underbrace{\int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau}_{=f(t)*g(t)}\right) \\ = \mathcal{F}(f(t)) \cdot \mathcal{F}(g(t)), \end{aligned} \quad (43)$$

$$\mathcal{F}(f(t) \cdot g(t)) = \int_{-\infty}^{\infty} F(\tau)G(\omega-\tau)d\tau, \quad (44)$$

which are valid only if the integrals $\int_{-\infty}^{\infty} |f(t)|^2 dt$ and $\int_{-\infty}^{\infty} |g(t)|^2 dt$ are existent. Equation (43) describes the folding operation (convolution) in the spacial domain, which corresponds to a simple multiplication in spectral domain.

The working principle and the implementation of (43) and (44) via the fast Fourier transform (FFT) do form one of the most powerful tools in the field of numeric computation.

3.5.1. Example: Polynomial Multiplication. Suppose two polynomials of the degree $n = 2$. Then the multiplication of these two would be

$$\begin{aligned} (a_0 + a_1x + a_2x^2) \cdot (b_0 + b_1x + b_2x^2) \\ = a_0b_0 + a_0b_1x + a_0b_2x^2 + a_1b_0x + a_1b_1x^2 \\ + a_1b_2x^3 + a_2b_0x^2 + a_2b_1x^3 + a_2b_2x^4 = \sum_{i=0}^{2n} c_i x^i, \end{aligned} \quad (45)$$

which finally gives an expression of degree of $2n = 4$. Performing the expansion will end in a convolution of the two coefficient vectors, which must be $2n + 1$ elements long to fit the result in the output vector. Doing all that will spend

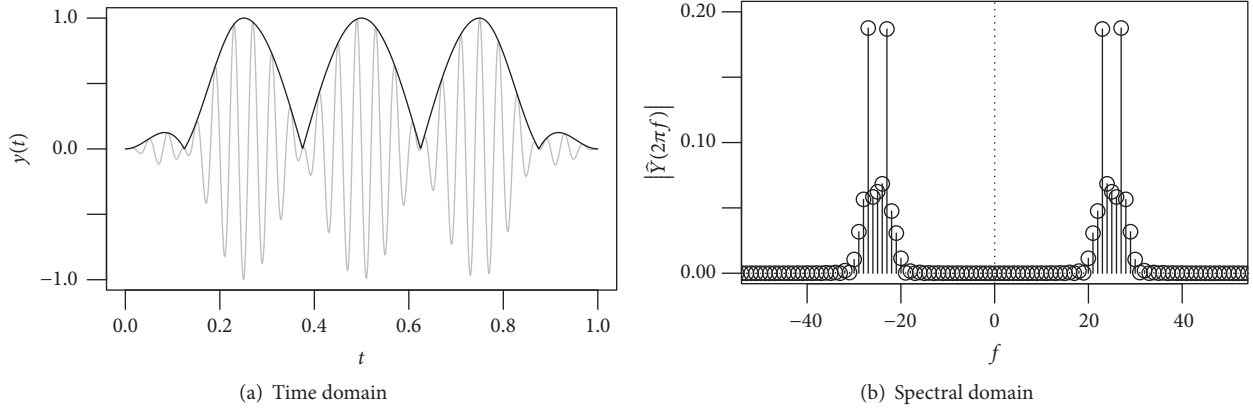


FIGURE 8: The envelope function of a signal. The spectrum (b) clearly indicates the bandwidth, which is needed to reconstruct the envelope. In case of a noisy signal the band pass filter must have at least this width to resolve reasonable results.

TABLE 2: Computation costs for the polynomial multiplication. The calculation amount for the FFT is estimated by $3 \cdot (2n+1) \log_2(2n+1)$.

Degree	Expansion	FFT
8	64	209
16	256	500
32	1024	1175
64	4096	2714

$(n+1)^2 = 9$ floating point multiplications on a computer. The statement

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \\ 0 \end{pmatrix} * \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_0 b_0 \\ a_0 b_1 + a_1 b_0 \\ a_0 b_2 + a_1 b_1 + a_2 b_0 \\ a_1 b_2 + a_2 b_1 \\ a_2 b_2 \end{pmatrix} \quad (46)$$

expresses the convolution and its result, where the symbol $*$ denotes the folding operator. Alternatively, this operation can be done with the help of the Fourier transform. According to (43),

$$\vec{c} = \mathcal{F}^{-1}(\mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b})) \quad (47)$$

requires the transform to be calculated three times.

The great advantage of using the Fourier transform, instead of the straightforward expansion, is the scaling of $n \cdot \log_2 n$ for the implementation of the FFT [14, Chap. 30]. As shown in Table 2, the FFT accelerates the calculation from a value of $n > 32$ for this example.

The standard algorithm of the discrete FFT is limited by a vector size of $N = 2^n$, which can be overcome with the implementation of the DFT method as described by Frigo and Johnson [15], which even allows prime numbered lengths without a loss of performance.

As a simple example, the following code illustrates the method described above.

```
# defining the coefficients
a <- c(1:3, 0, 0)
b <- c(5:7, 0, 0)
# calculating the FFT
A <- fft(a)
B <- fft(b)

# convolve via multiplication
# of A and B
c <- Re(fft(A * B, inverse=T)
      / length(a))
# would give the same result
c2 <- convolve(a, b, conj = F)
print(c)
##### OUTPUT #####
> 5 16 34 32 21
```

The result is given in the last line of the program listing. The R-command `convolve(x, y, conj = F)` uses the same mechanism and produces exactly the same results.

3.6. Spectral Filtering in Conjunction with the Sampling Theorem. Every sampled signal has its starting point $t = 0$, followed by a point in time T where it ends. The sampling procedure itself and the nature of the signal finally define the underlying band limitation. The intention of the first part of this section is to show the relation between the measurement and the consequences which arise if a causal signal is sampled. It turns out that in conjunction with the Fourier transform several convolutions take place in the time and frequency domain by selecting the maximum time T and the maximum frequency.

Regarding that, it is important to understand that the filter process already begins with the sampling of the signal. Figure 9 illustrates that for the function

$$y(t) = 0.8 \cdot \cos(2\pi \cdot 2t), \quad (48)$$

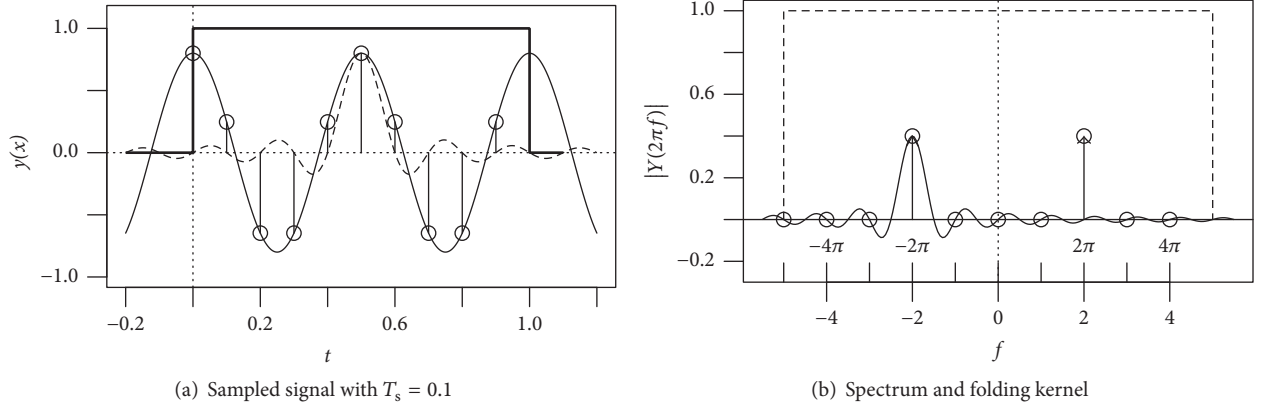


FIGURE 9: In (a), the function $y(t)$ from (48) is sampled in the time range $0 \leq t < 1$. The finite time interval corresponds to a rectangular weight function. The continuous spectrum (arrows in (b)) is then convoluted with the spectral representation of that rectangular window. The black line indicates the convolution step for one left spectral part. Note that only the Dirac impulse generates a value, because all other components remain zero. The dashed line in (b) indicates the limitation to a maximum frequency, which corresponds to a convolution in the time domain (a). Open symbols show the sampling (a) and the result of the DFT (b).

which fits perfectly into the period of the sampling window. In principle, this function is defined on the interval of $-\infty < t < \infty$. Then, the data acquisition process defines a starting point at $t = 0$ where the measurement begins and an end point $t = T$ where the measurement stops. This is illustrated by the bold rectangular window

$$w(t) = \begin{cases} 1, & 0 \leq t < T = 1 \\ 0, & \text{elsewhere} \end{cases} \quad (49)$$

in Figure 9(a). This window function picks out a range $0 \leq t < T$ of $y(t)$ so that the signal

$$s(t) = w(t) \cdot y(t) \quad (50)$$

is now defined by $y(t)$ multiplied by $w(t)$. According to the convolution properties of the Fourier transform (see Section 3.5), this operation corresponds to the convolution

$$\begin{aligned} S(\omega) &= \int_{-\infty}^{\infty} W(\omega - \tau) \cdot Y(\omega) d\tau \\ &= W(\omega) * Y(\omega) \end{aligned} \quad (51)$$

of the ideal spectrum $Y(\omega) = \mathcal{F}(y(t))$ with the spectral representation

$$W(\omega) = \frac{1}{i\omega} (1 - e^{-i\omega T}) \quad (52)$$

of the window function $w(t)$, which is in fact Shannon's sampling function [11]. The consequence for the signal's spectrum $S(\omega)$ is that the ideal Dirac-spectrum $Y(\omega)$ smears out to $\sin(x)/x$ like functions, because each arrow is weighted with $W(\omega)$. Note the zeros of the resulting spectrum are in an equidistant spacing of $\delta f = 1/T$ now. At this moment, the original function $y(t)$ is just windowed and not sampled.

The next step is to limit the spectrum in Figure 9(b) to a maximum frequency; for example, $f_{\max} = 1/2T_s$, by applying an additional rectangular window in the frequency domain

$$V(\omega) = \begin{cases} 1, & |\omega| \leq (2T_s)^{-1} \\ 0, & \text{elsewhere.} \end{cases} \quad (53)$$

The parameter T_s becomes the period of the sampling frequency. The important link to the Nyquist frequency would be clarified later. However, the corresponding representation of $V(\omega)$ in the time domain

$$v(t) = \frac{\sin((2T_s)^{-1} \cdot t)}{t} \quad (54)$$

is displayed by the dashed line in Figure 9(a). In the context of the convolution, which now takes place in time domain, the windowed signal

$$s(t) = (w(t) \cdot y(t)) * v(t) \quad (55)$$

is finally convoluted with $v(t)$. But still, the result (55) is not sampled yet which is done in the next step.

Sampling is a procedure where individual values of the function $s(t)$ are selected and stored. With regard to the nature of the sinc-like functions in Figure 9, the sampling should take place at each possible zero of $v(t)$, as well as $v(t = 0)$ which is the only sampling point with a nonzero value of $v(t)$. This behavior guarantees that the convolution (55) rejects everything but the points of sampling. In the same moment, also the distance in between them is set to $\delta t = T_s$, because the function $v(t)$ provides two zeros in an interval of $(2T_s)^{-1} \cdot t \leq 2\pi$, which correspond to the next nonzero samples of the neighboring points. In other words, by choosing the next sample, $v(t)$ must be shifted by exactly T_s to reject the former sample and to take the current value. Note violating this restriction, because the location of sampling

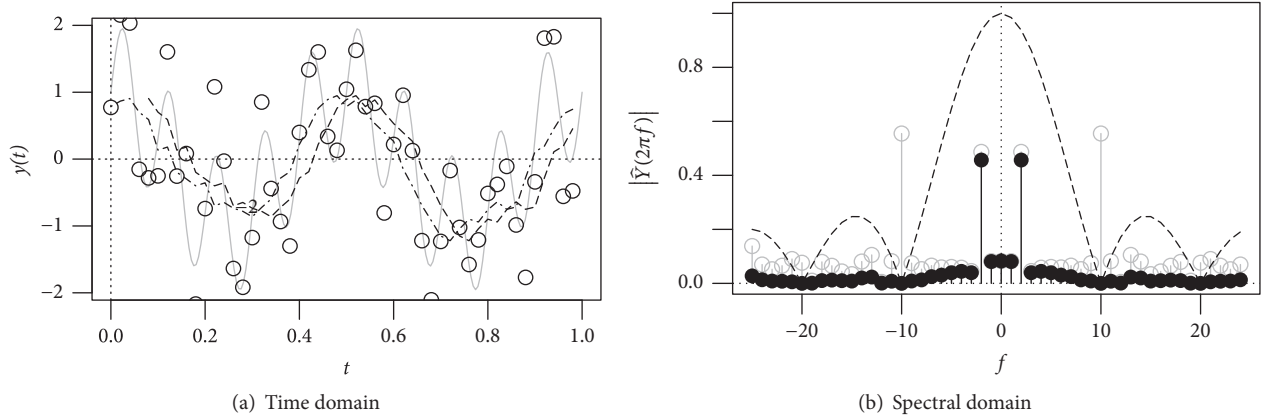


FIGURE 10: Calculating the moving average to reduce the noise and to reject the high frequency. Panel (a) shows the original function (gray line) and the noisy sampling points (open symbols). The result of the moving average in the time domain is given in (a) for the convolution (dashed) and the shifted FFT (dash-dotted). (b) displays the spectrum of the original function (gray), the filtered result (black), and the kernel function (dashed).

points jitters, will lead to an error in the discrete spectral representation \hat{S}_m .

Finally, this conceptual description gives second access to the question why the sampling frequency should be at least twice the maximum signal frequency. A detailed mathematical derivation of the discrete Fourier transform and the properties of sampling is given in the book of Debnath and Shah [16, Chap 3]. However, the discrete sampling given in Figure 9(a) and the finite length of the sampling series cause the convolution (51) to take the only possible peaks in the spectrum at $f = \{-2, 2\}$, because all other 2π -periodic values multiplied with the sampling kernel (sinc-function) evaluate to exactly zero.

If the signal period does not fit perfectly into the window, which is the case in reality, the kernel function $W(\omega)$ never matches the true Dirac, but instead of that it sums up values into the left and right bin of the corresponding discrete Fourier vector \hat{S}_m . To reduce these side effects, other window functions can be used in advance, before the DFT takes place. The intent of the window functions (Hamming-, Blackman-, or Tukey-window) is to convolve the spectrum with a steeper decreasing function amplitude compared to $W(\omega)$ to minimize the effect of the period's misfit. Moreover, $w(t)$ has an infinite spectrum, which is then truncated in the finite and discrete data processing, which leads to additional side effects if the period does not fit the sampling window. However, window functions can help to reduce such interferences, which are generated when local nonperiodic events or infinite spectra are present.

3.6.1. Application. Convolution filter in the time domain, such as the moving average filter, is easy to calculate with the help of the FFT, given the signal

$$s(t) = \cos(4\pi \cdot t) + \sin(20\pi \cdot t) + \mathcal{N}(0, 0.5)|_{0 \leq t < 1}, \quad (56)$$

which is defined in the range of $0 \leq t < 1$. The moving average of the sampled signal \hat{s}_n with the filter kernel length of $N_K = 5$,

$$\overline{\hat{s}}_n = \sum_{i=1}^{N_K} \hat{k}_i \cdot \hat{s}_{n-i}, \quad (57)$$

is then calculated for the n th element by taking the weighted sum of N_K elements before that element. The filter kernel \hat{k} for the example in Figure 10 is represented by the weighting coefficients

$$\hat{k} = \frac{1}{5} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \quad (58)$$

With respect to the spectrum of \hat{k} , it becomes clear that the length of \hat{k} is chosen in a way that the high-frequency part $\sin(20\pi \cdot t)$ is rejected completely. But it is also evident that other frequency components still remain in the result or were damped falsely, like the low frequency part $\cos(4\pi \cdot t)$.

Figure 10(a) illustrates that the application of the moving average filter in the time domain will lead to a phase-shift in the signal. Working in the spectral domain, one will gain the possibility of shifting the signal back and reconstruct the whole definition range as follows:

$$\begin{aligned} \mathcal{F}(\overline{s(t)}) &= e^{i\omega(N_K/2)} \cdot S(\omega) \cdot K(\omega), \\ \mathcal{F}(\overline{\hat{s}(n)}) &= e^{i2\pi(((0 \dots (N-1)) \cdot N_K)/2N)} \cdot \hat{S}_m \cdot \hat{K}_m. \end{aligned} \quad (59)$$

The result of this operation illustrates the dash-dotted line in Figure 10(a). The code which calculates this is presented below.


```

# Convert the kernel k into
# its spectral representation.
# The different lengths are adapted.
K <- fft( c(k,rep(0, length(y)-length(k))) )

exp_ex <- exp( 2i*pi * (1:length(y)-1) / length(y) * length(k)%/%2 )
Y <- fft( y ) / length(y)
y_spec <- Re( fft( exp_ex * Y * K, inverse=T) )

```

Pay attention to the statement `fft(c(k, rep(0, length(y) - length(k))))` in the first line. It first fills up k with zeros until the length of the whole data set, before the FFT is calculated. This is because the multiplication in frequency space can only take place between vectors of equal lengths.

With respect to the spectrum of the kernel \widehat{K} , it becomes evident that the moving average filter is not able to suppress higher frequencies completely.

3.6.2. Example: The Ideal Low Pass Filter. Since the moving average has some drawbacks (because of its infinite frequency response), one should think about an ideal (low pass) filter with a finite frequency response. Such a filter has the advantage that unwanted spectral components can be rejected completely. On the other hand, this results in an infinite long kernel in the time domain, which is nevertheless calculated in the spectral domain.

The basic principle of such a filter is to set each component above a certain frequency $|\omega_{\max}|$ value to zero

$$S(|\omega| > \omega_{\max}) = 0. \quad (60)$$

Figure 11 illustrates the procedure. Compared to the previous moving average example in Figure 10, the output signal's shape changes to a more smooth and accurate form. However, there remains a difference compared to the ideal low frequency part (dashed line in Figure 11(a)), which is the result of the remnant spectral components around the signal frequency $f = 2$.

A code snippet to calculate the ideal low pass is presented below.

```

# assuming x and y to hold the time
# vector and the sampling points

Y <- spec.fft(y, x)
# doing the filtering
Y$A[abs(Y$fx) > 3] <- 0
# reconstruct the filtered signal
yf <- spec.fft(Y, inverse = T)$y

```

The spectral package provides the `filter.fft(y,x, BW, fc=0, n=3)` function to do that in one step. It utilizes

the `analyticFunction(y)` to filter the one-sided spectrum and enables the user to apply a band pass filter with the arbitrary polynomial weight function

$$w \propto \omega^n. \quad (61)$$

The parameter n sets the polynomial degree, whereas 3 is close to the moving average. Passing higher values, say $n = 10$, will successively converge to the ideal bandpass solution. All this is because sometimes it is desirable to have a smooth weight of frequencies and sometimes not.

The used code then changes to the following example.

```
yf <- filter.fft(y, x, fc=0, BP=3, n=10)
```

As seen in Figure 11, the bandwidth BP is symmetric around the center frequency $fc = 0$, so the values must be set in accordance with the desired window which should remain in the spectrum. Utilizing the analytic signal representation avoids mistakes when programming the weight function and keeps the code clean.

3.6.3. Example: Noise Reduction via Autocorrelation. By choosing an appropriate kernel function, it becomes possible to extract certain features of the signal. The previous example illustrated how a low pass filter can filter out a noisy signal by zeroing all upper frequency components in the spectrum. In the time domain, this corresponds to a folding operation of the input data with an infinite “sinc-” like kernel function.

Now the opposite approach is to ask what the most significant periodic components within the signal are. First of all, the continuous autocorrelation function

$$\text{acf}(\tau) = \int_{-\infty}^{\infty} s(t) s^*(t - \tau) dt \quad (62)$$

$$= \mathcal{F}^{-1}(S(\omega) \cdot S^*(\omega)) \quad (63)$$

provides a mechanism to examine a data set or function with respect to its self-similarity. The equations above convolute the signal $s(t)$ with its complex conjugate $s^*(t)$. According to the convolution properties, (62) can be expressed as the inverse Fourier transform of the product of the signals spectra. Note for a real-valued signal $s(t) = s^*(t)$. Suppose the signal consists of a stationary sinusoidal signal. It is quite evident that the acf rejects the noise which overlays $s(t)$. The underlying noise equals a nonstationary process and therefore it cancels out by utilizing the acf. A generalization of this

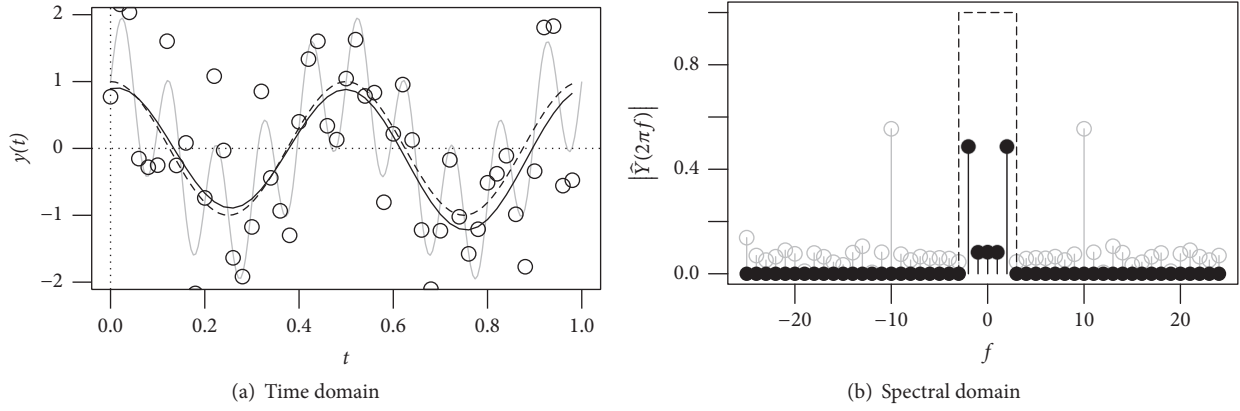


FIGURE 11: Calculating the ideal low pass filter. Panel (a) shows the original function (gray line) and the noisy sampling points (open symbols). The result of the ideal low pass filter in the time domain is given in (a) by the solid line. The dashed line in (a) illustrates the low frequency part of the signal with $f = 2$. Panel (b) displays the spectrum of the original function (gray), the filtered result (black), and the spectral kernel function (dashed).

statement is the Wiener-Khinchin theorem. Its application is discussed in detail by Cohen [4, Chap 1.9], Marks [17, Chap. 4], and Wiener [18].

Assuming the function

$$s(t) = \cos(2\pi \cdot 2 \cdot t) + \sin(2\pi \cdot 10 \cdot t) + \mathcal{N}(0, 0.5) \quad (64)$$

with some normal distributed noise overlaid, the signal \hat{s}_n is then the sampled function $s(n \cdot T_s)$ with $T_s = 0.01$, so it fits perfectly into one sampling period of $T = 1$. This example is very artificial and real-world signals often cause aliasing effects, so attention must be paid.

The discrete spectrum of acf_n can now be calculated by

$$\text{ACF}_m = \hat{S}_m \cdot \hat{S}_m^* \quad (65)$$

```
x <- seq(0, 1, by = Ts)
# for perfect periodicity
x <- x[ -length(x) ]
y <- cos(2*pi * 2*x) + sin(2*pi * 10*x)
  + rnorm(length(x), sd = 1)
# calculating the autocorrelation
Y <- fft( analyticFunction(y) ) / length(y)
ACF <- Y * Conj(Y)

# calculating the weight vector
w <- ACF
w[abs(w) < sd( abs(w) )] <- 0
w[w != 0] <- 1

# backtransform of the filtered signal
yf <- Re(fft( Y * w, inverse = T ) )
```

Note the resulting spectrum is real-valued, so the phase information will be lost. A proper noise reduction can be achieved by defining a weighting vector

$$W_m = \begin{cases} 0, & |\text{ACF}| < \text{sd}(|\text{ACF}_m|) \\ 1, & \text{else,} \end{cases} \quad (66)$$

which sets every spectral component smaller than the standard deviation of the ACF_m to zero. The resulting filtered signal

$$\hat{s}_f(n) = \mathcal{F}^{-1}(W_m \cdot \hat{S}_m) \quad (67)$$

is given in Figure 12. For the given example, it is important to calculate everything with the analytic function representation to preserve the energy content of the data vectors.

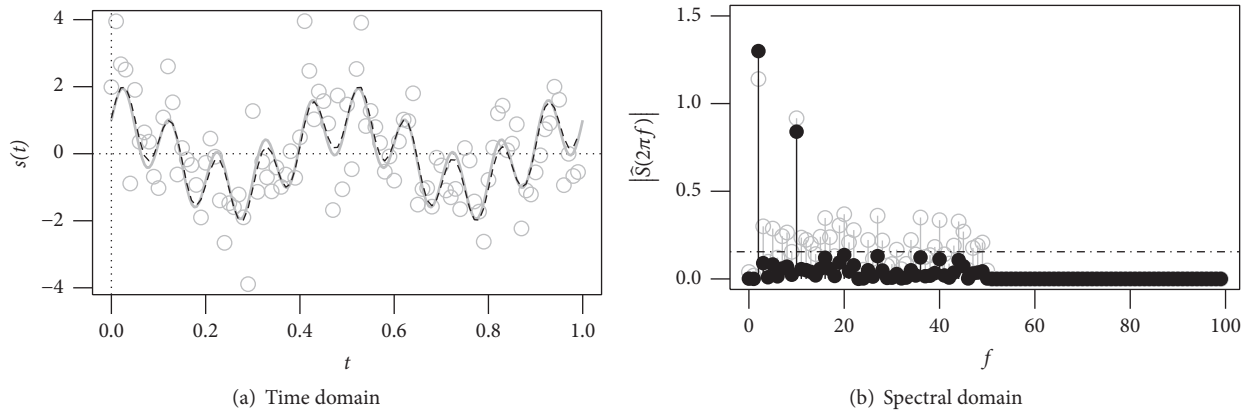


FIGURE 12: Noise reduction with autocorrelation. In the left panel (a) the sampled function (gray line) is overlaid with noise (circles). The bold line is the result of the weighted filtering procedure. In the frequency domain (b) the single sided spectra of the analytic functions are displayed. The input function (gray symbols) contains much noise which is reduced due to the autocorrelation (dotted black). The horizontal line indicates the threshold below which all spectral components are ignored.

3.7. Nonstationary Processes: Spatially Dependent Spectral Analysis. Many signals in reality do not fulfill the requirement of stationarity. It is very often the case that the measured signal is overlaid by a trend or a “temporal-local” event that takes place only once. All these in-stationary and nonperiodic processes will spread out into all frequencies in the corresponding Fourier decomposition of the data. If a proper band limitation cannot be achieved, the mapping between physical frequencies and the spectral representation might fail too, as stated in Section 3.1.

The equation

$$y(t) = \underbrace{\frac{1}{\sqrt{2\pi \cdot 0.05}} \cdot e^{-(t-0.2)^2/(2 \cdot 0.05^2)}}_{A_1(t)} \cdot \sin(2\pi \cdot 20 \cdot t) + \underbrace{\frac{1}{\sqrt{2\pi \cdot 0.1}} \cdot e^{-(t-0.7)^2/(2 \cdot 0.1^2)}}_{A_2(t)} \cdot \sin(2\pi \cdot 40 \cdot t) \quad (68)$$

describes a very artificial example of a nonstationary signal, which is illustrated in Figure 13(a). Here two different events occur. The first one has a low frequency at $f_1 = 20$ Hz, whereas the second one oscillates faster with $f_2 = 40$ Hz. Both parts are modulated with a Gaussian envelope to switch each of them on and off. Finally, Figure 13(b) shows the resulting time variant decomposition of the signal $y(t)$. The `waterfall()` function from the `spectral` package can be used to calculate such kinds of waterfall diagrams.

Note this type of analysis produces a two-dimensional time-frequency map like a shifting FFT, which selects only a window of the data before doing the spectral decomposition for one time step. The amplitude functions $A_1(t)$ and $A_2(t)$

play the role of arbitrary window functions which mask the time series. The interpretation is that $A_i(t)$ contains the information in time where the finite process is located. All this relies on the assumption that the signal of interest can be modeled as

$$y(t) = \sum_i A_i(t) \cdot \cos(\omega_i + \varphi_i). \quad (69)$$

In contrast to the shifting FFT, the introduced approach is to do an amplitude demodulation by calculating the signal's envelope. The result is the amplitude function $A(t)$ for a given center frequency f_c , which corresponds to the frequency of the physical process. Finally, a sufficient bandwidth around f_c guarantees that $A(t)$ will be reconstructed correctly.

The first advantage of this method is that for each single frequency of interest the whole data set is taken into account. Note the shifting FFT would select only a certain range of the data, so information about low frequencies can be lost if the selected window becomes too small. However, the overall frequency resolution decreases to the length of the window size of the shifting FFT. In the end, two signals that are very close in the frequency space might not be distinguishable anymore. The second point is the necessary band pass filter at f_c , which is changed according to the actual frequency. This draws attention to the uncertainty principle, which states that with increasing frequency the locating of an event becomes sharper but the exact frequency gets more incorrect.

3.7.1. Implementation. As stated above, the key component is a band pass filter BP, which must be applied twice to calculate the two envelopes in Figure 13. The following code shows how this can be done in R with the appropriate functions provided by the `spectral` package.

```
A1 <- Re( filter.fft(y,x,fc=20, BW=10, n=10) )
A1 <- Re( envelope( A1 ) )

A2 <- Re( filter.fft(y, x, fc=40, BW=10, n=10) )
A2 <- Re( envelope( A2 ) )
```

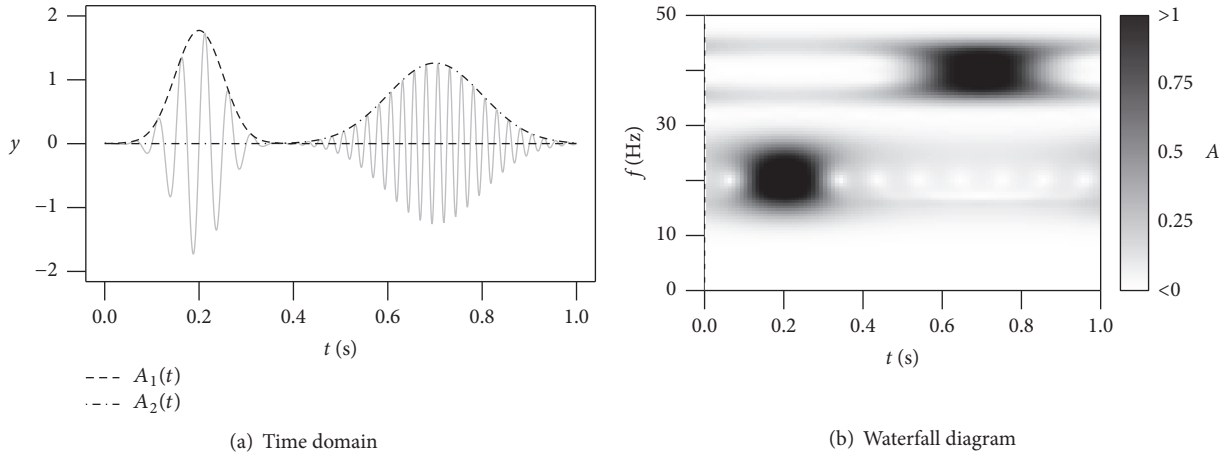


FIGURE 13: Basic simple example of a nonstationary signal (a) with its time depending spectral decomposition in (b). The different line types of the envelope correspond to the first and second event.

The next step is to do this calculation for many frequencies. A reasonable range is to start from $f = 0$ up to $f_s/2$. To accelerate the code the `waterfall()` function uses a fast

version of the `envelope()` function, which combines the filtering and the Hilbert transform as follows.

```
# Defining the frequency vector
Y.f <- seq(0, (n-1)*df, length.out=n)
# calculating the sign for the HT
sY.f <- (1 - sign(Y.f - mean(Y.f)))
# correct first bin
sY.f[1] <- 1

fast_envelope <- function(y, x, fc, BW, nf)
{
  Y <- BP(Y.f, fc, BW, nf) * fft(y)/n * sY.f
  hk <- base::Mod(fft(Y + 1i*Y, inverse=T)/sqrt(2))
  return(hk)
}
```

Here the working band pass filter is implemented as a weighting vector multiplied by the amplitudes. The second line in the `fast_envelope()` function performs the back transform. Compared to the calculation of A_1 and A_2 , this method saves two back transforms and one real part extraction.

Finally, the bandwidth calculation is done by an empirical approach, like

$$BW(f_c) = \begin{cases} 4\delta f & f_c < 16\delta f \\ \frac{f_c}{4} & 16\delta f \leq f_c \leq wd \cdot \delta f \\ wd \cdot \delta f & \text{else.} \end{cases} \quad (70)$$

Here $\delta f = 1/\Delta x$ denotes the frequency step and wd is the normalized width of the resulting band pass. The task of

$BW(f_c)$ is to widen the frequency band for higher frequencies, whereas low frequencies take a very small bandwidth. This somehow pays attention to the uncertainty principle, which states that “a narrow waveform yields a wide spectrum and a wide waveform yields a narrow spectrum and both the time waveform and frequency spectrum cannot be made arbitrarily small simultaneously” [19].

3.7.2. Application. The two following examples illustrate how the `waterfall()` function can be used for intricate signals. First of all the function

$$y(t) = |2t - 1| \cdot \sin(2\pi \cdot 10 \cdot t) + \begin{cases} 0 & t \leq 0.5 \\ \sin(2\pi \cdot 20 \cdot t^2) & \text{else} \end{cases} \quad (71)$$

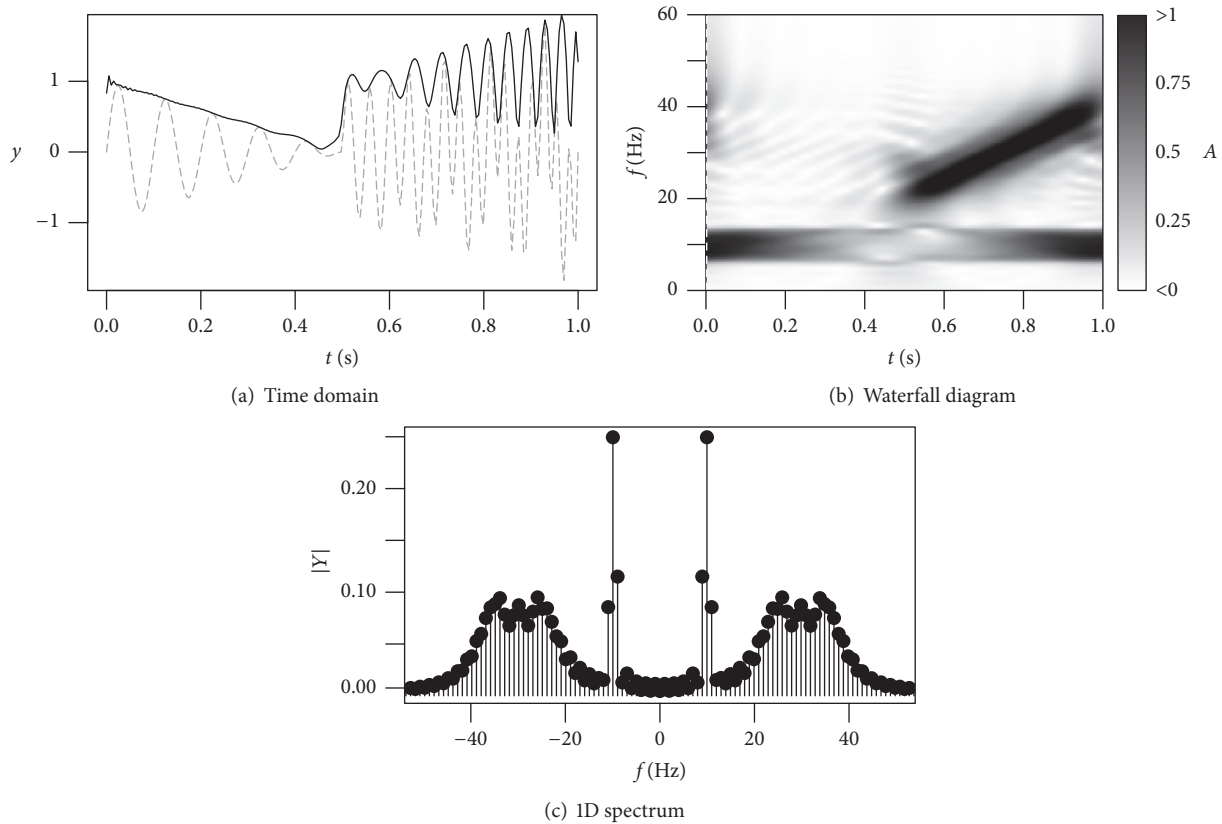


FIGURE 14: Complex example for the temporal-dependent spectral decomposition. (a) shows the time series and (b) its decomposition. Panel (c) demonstrates that the simple 1D spectrum would produce misleading results. Here one can only identify the 10 Hz carrier, but the frequency drift is hidden behind the wide band spectrum at around 30 Hz.

is going to be analyzed. Figure 14 illustrates the results. The waterfall diagram in Figure 14(b) displays all the features of (71). Note that even the $|x|$ function and the $\sin(x^2)$ term can be distinguished, whereas the normal time-invariant Fourier spectrum in Figure 14(c) hides this property completely. Here the 10 Hz carrier is the only “correctly” visible signal component.

A second example is given in Figure 15(a). Here the given data represents the outflow temperature of a cooling system with about 300 kW power. It consists of a heat exchanger outside the building and a secondary water loop inside from which to draw the heat. A sprinkler system improves the performance of the outside heat exchanger, so that outflow temperatures below the ambient temperature become possible. Figure 15(b) illustrates the ambient temperature near the external air heat exchanger. The impact of the sun’s radiation becomes visible in a signal amplitude with a very long period of about $T_{\text{sun}} \approx 80$ min. In terms of period lengths, the high eigenfrequencies of the hydraulic system become visible at the bottom, approximately within a range of 10 min to 20 min, when the control loop reaches its limit.

Finally, the following code shows how to invert the frequencies to periods. Note that a resampling of the matrix must be performed, because the unequally spaced vector $1/f$ now maps to the corresponding spectrum. This task is done

by the `plot.fft()` function, which also plots objects with the attribute `mode = "waterfall"`.

```
# calculating the waterfall diagram
wf <- waterfall(temp,time,nf = 3)
# change to periods = 1/f
wf$fx <- 1/wf$fx
# avoid "infinite" values
c <- !is.infinite(wf$fx)
wf$fx[1] <- 2 * max( wf$fx[c],na.rm = T )

# plot graph with hour-scale on the x-axis
wf$x <- wf$x/60

plot(wf,xlim=c(11,17),ylim=c(3,80)
     ,zlim=c(0,0.5) )
```

3.8. Fragmented and Irregularly Sampled Data. Sometimes the nature of measurement prevents an equally spaced sampling. Observations of astrophysical bodies like the sun or the moon are examples of objects only visible at a certain time of day. Concerning a definite property of these

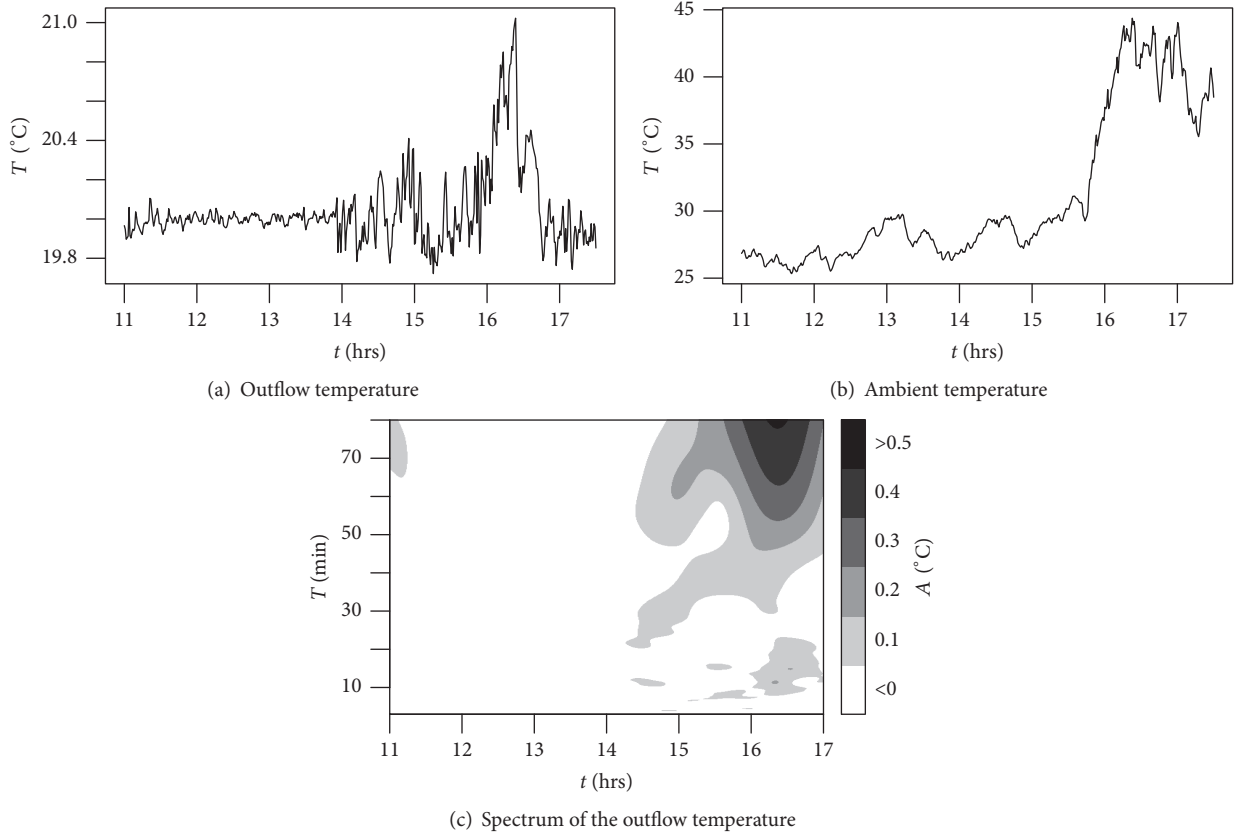


FIGURE 15: Outflow temperature of a 300 kW cooling system. Panel (a) shows the performance of the outflow temperature. At about 4 p.m., the system reaches its control limits, because the ambient temperature (b) exceeds significantly the set-point (20°C) of the outflow temperature. The high eigenfrequencies of the hydraulic system become visible from about 2:30 p.m. on.

bodies, the corresponding time series would be fragmented, which makes a spectral analysis with a standard Fourier transform almost impossible. A nice application example can be found in “The Lick Planet Search: Detectability and Mass Thresholds” by Cumming et al. [20]. Another issue would be the measurement with randomly sampled data, for instance, the occurrence of an event like the appearance of a malfunctioning product in a production lane or even a measurement with a high jitter.

The time instances t_n when a sample is taken now depend on the sample number n . This means the time interval

$$\Delta t = t_{n-1} - t_n \neq \text{const.} \quad (72)$$

is not constant anymore, so the sampling time becomes an additional data vector. This enables one advantage in case of randomly sampled data. Given a stationary process, it becomes possible to estimate signal amplitudes for frequencies which are in the order of $f_{\max} \approx \mathcal{O}(\min(\Delta t)^{-1})$, with the minimal distance between two points. In fact, this is related to an average Nyquist frequency \bar{f}_c , where

$$f_{\max} > \frac{1}{2} \frac{N}{\max(t) - \min(t)} = \bar{f}_c, \quad (73)$$

which can be defined by the time range of the time series and the total number of samples.

Let us introduce the Lomb-Scargle periodogram as a consistent estimator for spectral properties of discrete data. The estimation in that sense is not a mathematical transform like the Fourier transform. The frequencies of interest can be selected freely, so the whole method analyzes a data set against amplitude and phase. The associated mathematical details of this statistical approach can be found in the papers by Mathias et al. [21], Hocke and Kämpfer [22], Scargle [7], and Lomb [23], who originally developed the method.

In principle, the concept of the procedure is about a least square optimization of the parameters A_i and B_i for the model function

$$y(t) = \sum_i A_i \cdot \cos(\omega_i \cdot (t - \tau_i)) + B_i \cdot \sin(\omega_i \cdot (t - \tau_i)), \quad (74)$$

which is fitted to the data. Remember the introductory Section 2.2 shows a slightly different approach in conjunction with the trigonometric identities (4), which also become the key component here. While the traditional calculation of a Newton algorithm takes several iteration steps until the result converges, the optimal parameters can alternatively be estimated by one single matrix inversion; see [21]. This can be done only if the orthogonal property of the trigonometric model function is taken into account. The mathematical

proof can be found in “A Generalized Inverse for Matrices” by Penrose and Todd [24]. As a necessary condition to obtain an optimal result for A_i and B_i , the relation

$$\sum_{n=1}^N \cos(\omega_i \cdot (t_n - \tau_i)) \cdot \sin(\omega_i \cdot (t_n - \tau_i)) = 0 \quad (75)$$

must be fulfilled. Given that it can be shown that the error of the resulting fit against the normal quadrature approach from Section 2.2 is minimized, from (75), the calculation of τ_i can be derived:

$$\tan(2\omega_i \cdot \tau_i) = \frac{\sum_n \sin(2\omega_i \cdot t_n)}{\sum_n \cos(2\omega_i \cdot t_n)}. \quad (76)$$

Next to that, the parameters R , I , C , and S will be defined as follows:

$$\begin{aligned} R(\omega_i) &\equiv \sum_n y_n \cos(\omega_i \cdot (t_n - \tau_i)), \\ I(\omega_i) &\equiv \sum_n y_n \sin(\omega_i \cdot (t_n - \tau_i)), \\ C(\omega_i) &\equiv \sum_n \cos^2(\omega_i \cdot (t_n - \tau_i)), \\ S(\omega_i) &\equiv \sum_n \sin^2(\omega_i \cdot (t_n - \tau_i)). \end{aligned} \quad (77)$$

Finally, the power spectral density P , the absolute amplitude A , and the phase φ

$$\begin{aligned} P(\omega_i) &= \frac{1}{2\sigma^2} \left(\frac{R(\omega_i)^2}{C(\omega_i)} + \frac{I(\omega_i)^2}{S(\omega_i)} \right), \\ A(\omega_i) &= \sqrt{\frac{2}{N} \left(\frac{R(\omega_i)^2}{C(\omega_i)} + \frac{I(\omega_i)^2}{S(\omega_i)} \right)}, \\ \varphi(\omega_i) &= -\left(\tan^{-1} \left(\frac{I}{R} \right) + \omega_i \tau_i \right) \end{aligned} \quad (78)$$

can be calculated from these coefficients. Hereby σ describes the standard deviation of the discrete data vector y_n .

In comparison to (14) from Section 2.2, the equations above work quite similarly. But now the modified function argument—extended by τ_i , the correction terms $C(\omega_i)$ and $S(\omega_i)$ —leads to the optimal least square fit solution.

In the limit of an infinite number of equally spaced samples, the Lomb-Scargle estimator converges to the Fourier transform, so it becomes a consistent estimator [21]. The last statement is very important, because with an increasing number of samples the error reduces until the result converges to the true amplitude and phase.

To value the significance of the estimated amplitude the “false alarm probability” (FAP)

$$p(P < P_0) = 1 - (1 - e^{-P})^M \quad (79)$$

is defined. Here the probability p that there is no other larger amplitude P than P_0 is expressed in terms of the exponential function above. For small values of p , the approximation

$$p(P < P_0) \approx M \cdot e^{-P} \Big|_{p \ll 1} \quad (80)$$

can be used. The free parameter M counts the *independent* frequencies in the data. These are difficult to measure a priori, but it turns out that with $M = N/2$ sufficient results can be achieved. A brief discussion on this issue can be read in the work of Townsend [25, Cap. 6.2], Zechmeister and Kürster [26], and Cumming et al. [20].

3.8.1. Implementation. The calculation of the phase is tricky. Invoking the `arctan2()` function is mandatory. Nevertheless, to prevent errors, the phase φ is calculated by

$$\varphi = -\left(\arctan 2 \left(\frac{I}{N}, \frac{R}{N} \right) + \omega_i \tau_i \right), \quad (81)$$

which takes the normalized components I and R .

Next to that, if the suggestions made by Townsend [25] are taken into account, then the above algorithm can be shortened. The problem is that the straightforward implementation runs *two* times over the whole data set, while calculating τ prior to the rest of the parameters. A keen refactoring of the equations above will minimize the computational cost. For a certain frequency ω_i , the amplitudes

$$A(\omega_i) = \sqrt{\frac{2}{N} \left(\frac{(c_\tau \cdot XC + s_\tau \cdot XS)^2}{c_\tau^2 \cdot CC + 2c_\tau \cdot s_\tau \cdot CS + s_\tau^2 \cdot SS} + \frac{(c_\tau \cdot XS - s_\tau \cdot XC)^2}{c_\tau^2 \cdot SS + 2c_\tau \cdot s_\tau \cdot CS + s_\tau^2 \cdot CC} \right)} \quad (82)$$

can be calculated out of the parameters

$$\begin{aligned} XC &= \sum_j y_j \cdot \cos(\omega_i \cdot t_j), \\ XS &= \sum_j y_j \cdot \sin(\omega_i \cdot t_j), \\ CC &= \sum_j \cos^2(\omega_i \cdot t_j), \end{aligned}$$

$$SS = \sum_j \sin^2(\omega_i \cdot t_j),$$

$$CS = \sum_j \cos(\omega_i \cdot t_j) \cdot \sin(\omega_i \cdot t_j),$$

$$\tau_L = \tan^{-1} \left(\frac{2 \cdot CS}{2 \cdot CC - 1} \right),$$

$$c_\tau = \cos(\omega \cdot \tau_L),$$

$$s_{\tau} = \sin(\omega_i \cdot \tau_L) \quad (83)$$

in one single loop. A vectorized *R*-code example is given below. Here the data and the corresponding frequencies form the matrix `omega.x`, which is processed successively.

```
# put everything in a matrix.
# One frequency per column.
# x corresponds to the time.
# y_ corresponds to the mean-free
# data vector.
omega.x <- x %*% t(omega)
co <- cos(omega.x); si <- sin(omega.x)
# use trigonometric identities
co2 <- co^2; si2 <- 1 - co2
si <- sqrt(1-co2)
CC <- colSums(co2)
SS <- colSums(si2)
YCS <- colSums(y_ * co)
YSS <- colSums(y_ * si)
CS <- colSums(si*co)
tauL <- atan(2 * CS / (2*CC - 1) / (2*omega) )
ct <- cos(omega * tauL)
st <- sin(omega * tauL)
ct2 <- ct^2; st2 <- 1-ct2

ctstCS <- 2*ct*st*CS
R <- (ct*YCS + st*YSS)
I <- (ct*YSS - st*YCS)

# a trick to reduce numeric error
# in atan2()
l <- sqrt(R^2+I^2)
A <- sqrt( 2/nt* ( R^2/(ct2 * CC + ctstCS + st2 * SS ) + I^2/(ct2 * SS -- ctstCS + st2 * CC) ) )
phi <- - omega*tauL
- atan2((I/l), (R/l))
```

The *R*-code illustrates how to use the specialties of the language. Instead of programming a `for` statement, the faster vector and matrix operations of *R* can be invoked.

3.8.2. Application. To show the power of the method, let us first assume a simple example. The function

$$y(x) = \sin(2\pi \cdot 7 \cdot x) \quad (84)$$

is going to be sampled with $N = 101$ equally spaced samples. Remember the last point of the data vector equals the first point of the period of the signal. As discussed in the application section of Section 3.1, the violation of the periodicity would lead to an error in the result of the Fourier transform. In addition to that, approximately 30% of the data were deleted.

Subsequently, Figure 16 shows the fragmented signal and the corresponding periodogram, which is calculated with the `spec.lomb(x, y, f)` function.

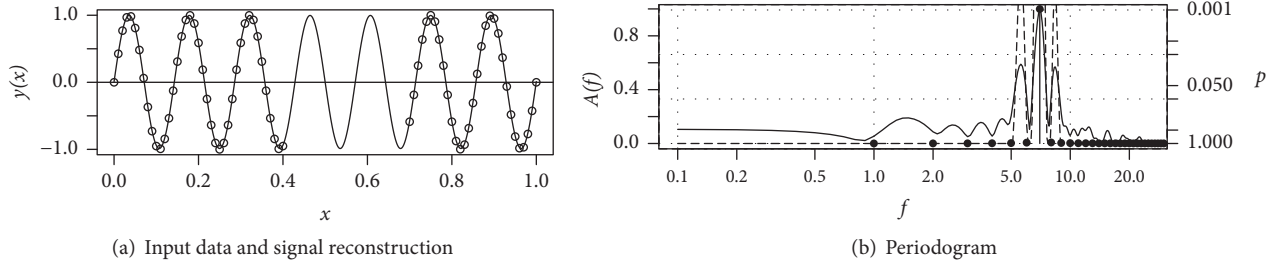


FIGURE 16: Simple Lomb-Scargle periodogram. The open symbols in panel (a) display the input signal. The solid line shows the reconstruction out of the spectrum. Panel (b) focuses on the periodogram. Here, the solid line corresponds to the spectral amplitude and the black points indicate the results of a DFT, as if there was no gap in the data. The dashed line represents the false alarm probability p . Note the value of 1 is located at the bottom, whereas smaller values are displayed above.

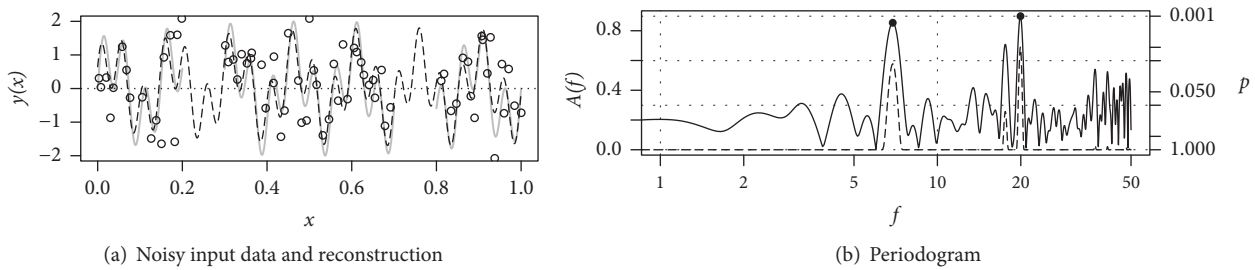


FIGURE 17: Randomly sampled data with noise and gaps. The bold gray line in (a) is the discontinuous signal. The circles represent the sampling points, which are overlaid with some strong noise and jitter. The dashed line represents the reconstruction. In (b) the corresponding Lomb-Scargle periodogram is given. The black line is the amplitude spectrum and the dashed line indicates p . The black points show the amplitude maxima, which the `filter.lomb()` function chose for reconstruction.

```
x <- seq(0,1,by=0.01)
x.new <- seq(0,1,by=1e-3)
yorg <- function(x)
  return(sin(2*pi*7*x))
# create the gap
cond <- !(x > 0.4 & x < 0.7)
x <- x[cond]; y <- yorg(x)

l <- spec.lomb(x=x, y=y, f=seq(0,25,by=0.1) )
lf <- filter.lomb(l,newx=x.new,phase="lin",threshold=3)
```

The code above shows how Figure 16 can be created. Note that the frequency vector f contains 250 different frequencies to analyze the data. Compared to the single sided spectrum of the *unfragmented* data's analytic signal representation (black dots in Figure 16(b)), the Lomb-Scargle periodogram produces large side band amplitudes to the left and right of the main amplitude. This is quite typical if the data is nonuniformly sampled or even patchy. The dashed line represents the false alarm probability, which tends to zero if the corresponding amplitude is significant.

The spectral package also provides a `filter.lomb()` function, with which the most significant amplitudes can be extracted for reconstruction. Provided the continuous

sampling vector `x.new`, the result is a new data set in which the remaining gaps are filled.

A more complex example is given in Figure 17. Here the function

$$y(x) = \sin\left(2\pi \cdot \left(x + \mathcal{N}\left(0, \frac{\Delta x}{4}\right)\right)\right) + \sin\left(2\pi \cdot 20 \cdot \left(x + \mathcal{N}\left(0, \frac{\Delta x}{4}\right)\right)\right) + \mathcal{N}(0, 1) \quad (85)$$

consists of two sin-terms, the location of sampling jitters, and the amplitude itself. In the simple test case here, normal

distributed noise is assumed and the resulting data vectors x_n and y_n are $N = 101$ elements in length. However, for reconstruction, the most significant amplitude values are selected again, so that the dashed line in Figure 17(a) will fill the gaps correspondingly.

For further reading about nonuniformly sampled data, the adaptive approach introduced by Stoica et al. [27] is recommended. This method has a better signal to noise ratio but will have much more computational cost.

4. Conclusion

Today, spectral analysis is one of the key methods in data processing. It helps to filter signals, to accelerate calculations, or to identify processes. While the complex mathematics is already documented in the given references and the common text books, the present work focuses on the *application* of the basic spectral methods in one dimension. Hereby working principles of the introduced methods and their possible misunderstandings are of interest. Small examples support the explanations about the work of operation and show the “dos and don’ts.” The idea of the present work is to show how the complex underlying theory of spectral methods can be applied to solve real-world tasks. A dedicated tool supporting this intent is the `spectral` package in the statistic environment *R*, which was developed contemporaneously to this paper. Further work will cover the applications of spectral methods in two and more dimensions.

Appendix

A. The Origin of $f_s/2$

To derive the band limitation, we have to assume a function $y(t)$ for which the Fourier transform $Y(\omega)$ exists. There might be a natural maximum frequency $\omega_{\max} \leq \omega_B$ in the signal, so the integration boundaries of the back transform

$$y(t) = \frac{1}{2\pi} \int_{-\omega_B}^{\omega_B} Y(\omega) \cdot e^{i\omega t} d\omega \quad (\text{A.1})$$

can be limited to $|\omega_B|$. In the next step, this function is sampled at discrete time instances $t = n \cdot T_s$, with a sampling frequency $f_s = 1/T_s$. It follows

$$\begin{aligned} y(n \cdot T_s) &= \frac{1}{2\pi} \int_{-\omega_B}^{\omega_B} Y(\omega) \cdot e^{i\omega(n/f_s)} d\omega, \\ &= \frac{1}{2\pi} \int_{-2\pi f_B}^{2\pi f_B} Y(\omega) \cdot e^{i2\pi(f/f_s)n} d\omega. \end{aligned} \quad (\text{A.2})$$

By substituting $d\omega \leftrightarrow 2\pi df$, the modified (A.2) can be written as

$$y(n \cdot T_s) = \int_{-f_B}^{f_B} Y(2\pi f) \cdot e^{i2\pi(f/f_s)n} df. \quad (\text{A.3})$$

The pointer $e^{i\varphi}$ is periodic with respect to 2π , so the span of $\Delta f/f_s$ must not exceed unity to reject duplication. It follows

$$\frac{\Delta f}{f_s} \leq 1 \quad (\text{A.4})$$

so that the maximum integration boundaries

$$f_B \leq 0.5 \cdot f_s \quad (\text{A.5})$$

can be concluded in the first instance.

Due to the requirement that the *frequency span* must be less than the sampling frequency, the undersampling of functions becomes possible. This changes the meaning of a typical Nyquist limit to a limitation of bandwidth which is necessary to obtain uniqueness.

B. Properties of the Fourier Operator

In the following, the properties of the Fourier operator [5] are listed.

Linearity:

$$\mathcal{F}(a \cdot f(t) + b \cdot g(t)) = a \cdot F(\omega) + b \cdot G(\omega). \quad (\text{B.1})$$

Scaling:

$$\mathcal{F}\left(f\left(\frac{t}{a}\right)\right) = |a| \cdot F(a \cdot \omega) \quad a \in \mathbb{R}, a \neq 0. \quad (\text{B.2})$$

Translation:

$$\mathcal{F}(f(a \cdot t + b)) = \frac{1}{a} e^{i\omega \cdot b/a} F\left(\frac{\omega}{a}\right) \quad (\text{B.3})$$

with $a, b \in \mathbb{R}, a \neq 0$.

Damping:

$$\mathcal{F}(e^{ib \cdot t} f(a \cdot t)) = \frac{1}{a} F\left(\frac{\omega - b}{a}\right) \quad a \in \mathbb{R}^+. \quad (\text{B.4})$$

n th derivation: the function $f(t)$ must be absolutely integrable and $\lim_{t \rightarrow \pm\infty} f(t) = 0$

$$\mathcal{F}(f^{(n)}(t)) = (i\omega)^n \mathcal{F}(f(t)). \quad (\text{B.5})$$

Integration: only if $\int_{-\infty}^{\infty} f(t) dt = 0$, then

$$\mathcal{F}\left(\int_{-\infty}^t f(t) dt\right) = \frac{1}{i\omega} F(\omega). \quad (\text{B.6})$$

Folding: only if the integrals $\int_{-\infty}^{\infty} |f(t)|^2 dt$ and $\int_{-\infty}^{\infty} |g(t)|^2 dt$ are existent, the two-sided folding of $f(t)$ and $g(t)$,

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau, \quad (\text{B.7})$$

can be calculated with the help of the Fourier transform. In the frequency domain, the folding operation reduces to a multiplication. This is expressed as follows:

$$\mathcal{F}(f(t) * g(t)) = \mathcal{F}(f(t)) \cdot \mathcal{F}(g(t)). \quad (\text{B.8})$$

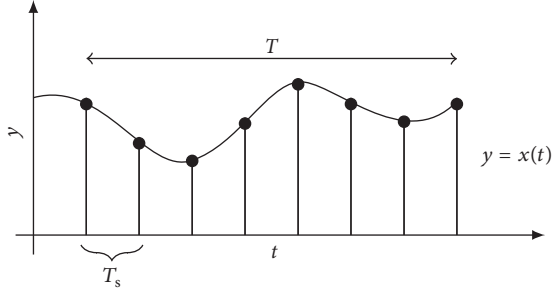


FIGURE 18: Common band limited and real-valued function. The reference sampling points are equally spaced with the distance T_s .

C. Derivation of the Discrete Fourier Transform

The derivation of the discrete Fourier transform (DFT) is described below in detail. The already introduced and sketched out properties of the DFT will become clear with the discrete formulation of the Fourier transform. We will see that the spectral information becomes periodic and in the 1D case redundant.

First, let y be a real-valued function of time, with

$$y = x(t) : \{t \in \mathbb{R} \rightarrow x(t) \in \mathbb{R}\}, \quad (\text{C.1})$$

which is going to be sampled later on. Figure 18 illustrates the continuous function and its sampling points. To perform the sampling, it becomes necessary to define the Dirac function as a distribution as follows:

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0, \end{cases} \quad (\text{C.2})$$

$$\text{with } 1 = \int_{-\infty}^{\infty} \delta(t) dt. \quad (\text{C.3})$$

The Dirac pulse is described as $\delta(t \neq 0) = 0$ for all values of t except $t = 0$. At this point, the Dirac function returns an infinite value, $\delta(t = 0) \rightarrow \infty$. However, the behavior (C.3) turns the Dirac function into a distribution. The finite value of the integral enables subsequent calculations to sample a point from the continuous function $y = x(t)$. This is called “sifting” property and can be written as

$$\int_{-\infty}^{\infty} \delta(t) \phi(t) dt = \phi(0). \quad (\text{C.4})$$

In conjunction with the Fourier transform (C.4), this leads to a necessary simplification for the derivation of the DFT.

Next to that, the continuous function $x(t)$ is going to be sampled with the help of the sampling function $\sum_{n=-\infty}^{\infty} \delta(t - nT_s)$. The sampling series

$$A(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s) \delta(t - nT_s) \quad (\text{C.5})$$

evaluates $x(t)$ at n reference points in time by multiplying the sampling function to $x(t)$. Because of the properties of the Dirac pulse, the sum (C.5) converges to the value $A(t) \rightarrow \infty$.

In the next step, the sampling series $A(t)$ is going to be inserted in the Fourier integral

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt. \quad (\text{C.6})$$

This transforms the function $x(t)$ from time or spatial domain into frequency domain, expressed by the complex valued spectrum $X(\omega)$. The capital letter $X(\omega)$ supports the result, which is a bijective mapping of $x(t) \leftrightarrow X(\omega)$. At this point, both function $x(t)$ and its spectrum $X(\omega)$ are still continuous. For real signals, $x(t)$ is causal, real-valued, and finite in time, with

$$0 < t \leq T. \quad (\text{C.7})$$

This means the signal is defined in a certain range of positive t and stops existing after a maximum time T . With this behavior, the boundaries of integration in (C.6) change. The result is the discrete spectrum of the Fourier series

$$X(m) = \frac{1}{T} \int_0^T x(t) e^{-im\omega_0 t} dt \quad (\text{C.8})$$

with $m \in \mathbb{N}$. Inserting the sampling series $A(t)$ in (C.8), now discrete points enter the discrete spectrum:

$$X(m) = \frac{1}{T} \int_0^T \underbrace{\sum_{n=-\infty}^{\infty} x(nT_s) \delta(t - nT_s)}_{A(t)} e^{-im\omega_0 t} dt. \quad (\text{C.9})$$

The distribution character (C.3) and the sifting property of the Dirac function help to rearrange the equation into the DFT form [5, 28],

$$X(m) = \frac{1}{T} \sum_{n=-\infty}^{\infty} x(nT_s) e^{-im\omega_0 nT_s}. \quad (\text{C.10})$$

For a finite data set of N equidistant data points, $\omega_0 = 2\pi/T$, and a total length of $T = N \cdot T_s$, this can be written as

$$\begin{aligned} X(m) &= \frac{1}{NT_s} \sum_{n=0}^{N-1} x(nT_s) e^{-im \cdot 2\pi(nT_s)/(N \cdot T_s)} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-im \cdot 2\pi(n/N)}. \end{aligned} \quad (\text{C.11})$$

Here T_s denotes the inverse sampling frequency $T_s = f_s^{-1}$ and $m \in \mathbb{N}$ measures the normalized discrete frequencies on which the spectrum is being evaluated.

This discrete form of the Fourier transform provides several properties which define the spectrum $X(m)$.

- (1) The spectrum is periodic with respect to 2π , as indicated in Figure 19. It follows that at least $m = N$ frequencies are necessary to describe a unique mapping of $x(n) \leftrightarrow X(m)$.
- (2) $X(\omega)$ can be interpreted as a line spectrum because it is only evaluated at $\omega = (2\pi/(N \cdot T_s))m$ discrete frequencies.

This is now being successively reduced:

$$\begin{aligned}
 \lim_{\varepsilon \rightarrow 0} \int_0^{\infty} e^{i\omega t - \varepsilon \omega} d\omega &= \lim_{\varepsilon \rightarrow 0} \int_0^{\infty} e^{-\omega \cdot (-it + \varepsilon)} d\omega, \\
 &= \lim_{\varepsilon \rightarrow 0} \frac{-1}{-it + \varepsilon} \cdot e^{-\omega \cdot (-it + \varepsilon)} \Big|_0^{\infty}, \\
 &= \lim_{\varepsilon \rightarrow 0} \frac{-1}{-it + \varepsilon} \cdot (0 - 1), \\
 &= \lim_{\varepsilon \rightarrow 0} \frac{1}{-it + \varepsilon} = \lim_{\varepsilon \rightarrow 0} \frac{i}{t + i\varepsilon}.
 \end{aligned} \tag{D.9}$$

The last term is now expanded by the conjugate denominator $t - i\varepsilon$. It follows

$$\begin{aligned}
 \lim_{\varepsilon \rightarrow 0} \frac{i}{t + i\varepsilon} &= \lim_{\varepsilon \rightarrow 0} \frac{\varepsilon}{t^2 + \varepsilon^2} + \frac{it}{t^2 + \varepsilon^2}, \\
 &\quad \rightarrow \pi \delta(t) \\
 &= \pi \cdot \delta(t) + \frac{i}{t}.
 \end{aligned} \tag{D.10}$$

The last term expresses the Poisson kernel, which is the solution of the Laplace equation in the upper half plane [29]. The expression

$$\lim_{\varepsilon \rightarrow 0} \frac{1}{\pi} \frac{\varepsilon}{t^2 + \varepsilon^2} = \delta(t) \tag{D.11}$$

also equals the Lorentz distribution and converges to the Dirac function in the limit of $\varepsilon \rightarrow 0$.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] R. Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016, <https://www.r-project.org/>.
- [2] M. Seilmayer, "Spectral: Common Methods of Spectral Data Analysis," 2016, R package version 1.0., <http://CRAN.R-project.org/package=spectral>.
- [3] S. W. Smith, *The Scientist & Engineer's Guide to Digital Signal Processing*, California Technical Pub, San Diego, Calif, USA, 1st edition, 1997.
- [4] L. Cohen, *Time-Frequency Analysis, ser. Prentice Hall signal processing series*, Prentice Hall PTR, Englewood Cliffs, N.J., 1995.
- [5] I. N. Bronštejn and K. A. Semendjajew, *Taschenbuch der Mathematik*, Harri Deutsch, Thun, Switzerland, 5th edition, 2001.
- [6] A. J. Jerri, "The shannon sampling theorem—its various extensions and applications: a tutorial review," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.
- [7] J. D. Scargle, "Studies in astronomical time series analysis. II—statistical aspects of spectral analysis of unevenly spaced data," *The Astrophysical Journal*, vol. 263, pp. 835–853, 1982.
- [8] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society A. Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.
- [9] D. E. Vakman and L. A. Vainshtein, "Amplitude, phase, frequency—fundamental concepts of oscillation theory," *Soviet Physics Uspekhi*, vol. 20, no. 12, pp. 1002–1016, 1977.
- [10] J. Duoandikoetxea Zuazo, *Graduate Studies in Mathematics*, vol. 29 of *Fourier Analysis*, American Mathematical Society, Providence, RI, USA, 2001.
- [11] C. E. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.
- [12] J. M. Whittaker, *Interpolatory Function Theory*, Cambridge University Press, Cambridge, UK, 1935.
- [13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.
- [14] T. H. Cormen, C. Stein, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, Mass, USA, 2nd edition, 2001.
- [15] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.
- [16] L. Debnath and F. A. Shah, *Wavelet Transforms And Their Applications*, Birkhäuser Boston, Boston, Mass, USA, Second edition, 2015.
- [17] R. J. Marks, *Handbook of Fourier Analysis & Its Applications*, vol. 800, Oxford University Press, Oxford, Conn, USA, 2009.
- [18] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, vol. 2, MIT Press, Cambridge, Mass, USA, 1949.
- [19] M. I. Skolnik, *Introduction to Radar Systems*, McGraw-Hill, New York, NY, USA, 2nd edition, 1980.
- [20] A. Cumming, G. W. Marcy, and R. P. Butler, "The lick planet search: detectability and mass thresholds," *The Astrophysical Journal*, vol. 526, no. 2, pp. 890–915, 1999.
- [21] A. Mathias, F. Grond, R. Guardans et al., "Algorithms for spectral analysis of irregularly sampled time series," *Journal of Statistical Software*, vol. 11, no. 2, pp. 1–30, 2004.
- [22] K. Hocke and N. Kämpfer, "Gap filling and noise reduction of unevenly sampled data by means of the Lomb-Scargle periodogram," *Atmospheric Chemistry and Physics*, vol. 9, no. 12, pp. 4197–4206, 2009.
- [23] N. R. Lomb, "Least-squares frequency analysis of unequally spaced data," *Astrophysics and Space Science*, vol. 39, no. 2, pp. 447–462, 1976.
- [24] R. Penrose and J. A. Todd, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, p. 406, 1955.
- [25] R. H. D. Townsend, "Fast calculation of the lomb-scargle periodogram using graphics processing units," *The Astrophysical Journal, Supplement Series*, vol. 191, no. 2, pp. 247–253, 2010.
- [26] M. Zechmeister and M. Kürster, "The generalised lomb-scargle periodogram—a new formalism for the floating-mean and keplerian periodograms," *Astronomy & Astrophysics*, vol. 496, no. 2, pp. 577–584, 2009.
- [27] P. Stoica, J. Li, and H. He, "Spectral analysis of nonuniformly sampled data: a new approach versus the periodogram," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 843–858, 2009.

- [28] R. Hoffmann, *Signalanalyse Und -Erkennung: Eine Einführung Für Informationstechniker*, Springer Berlin Heidelberg, Heidelberg, Germany, 1998.
- [29] E. M. Stein and G. L. Weiss, *Introduction to Fourier analysis on Euclidean Spaces*, vol. 32 of *Princeton mathematical series*, Princeton University Press, Princeton, NJ, USA, 1975.