1. Continuous-time signals

1.1 Exponential signal

Compute and plot exponential signal

$$x(t) = Ce^{at} \tag{1.1}$$

for values of a and C, and the range of t as in Fig.1.1. Observe this signal for different values of a and C.

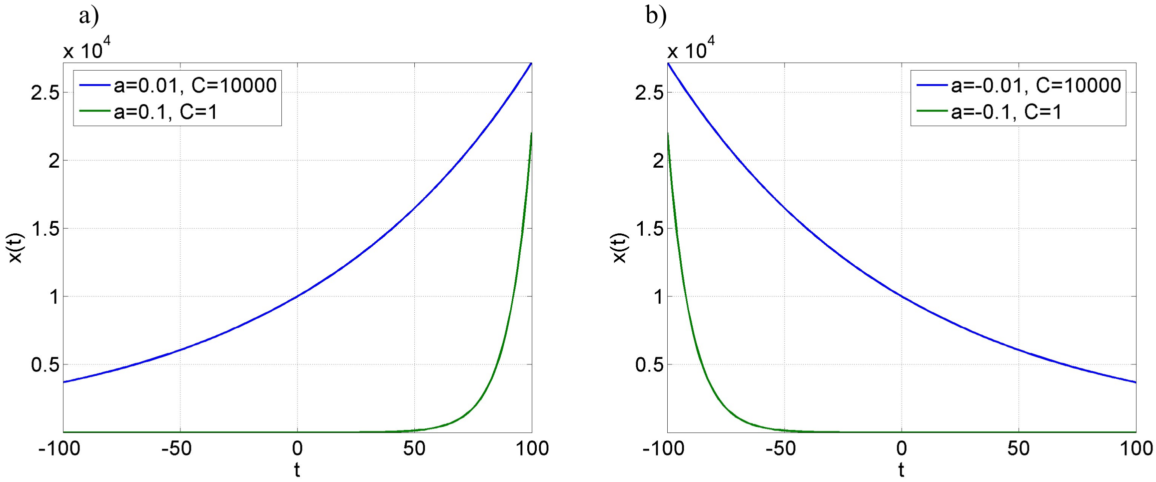

Fig. 1.1 Continuous-time exponential signals (1.1): a) a>0, b) a<0.

1.2 Complex exponential signal

Compute and plot complex exponential signal

$$x(t) = e^{j\Omega_0 t}, \tag{1.2}$$

for Ω_0=1 (rad/s), and the range of t as in Fig.1.2. Observe this signal for different values of Ω_0.

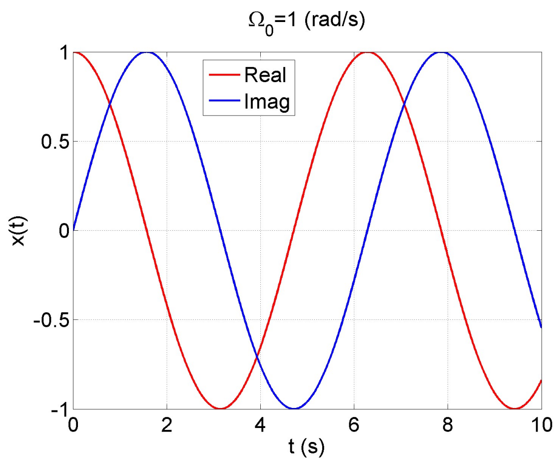

Fig. 1.2 Continuous-time complex exponential signal (1.2).

1.3 Sinusoidal signal

Compute and plot sinusoidal signal

$$x(t) = A\cos(\Omega_0 t + \varphi) = A\cos(2\pi F_0 t + \varphi)\,, \tag{1.3}$$

Verify computationally Euler's relation

$$A\cos(\Omega_0 t + \varphi) = \frac{A}{2} e^{j\varphi} e^{j\Omega_0 t} + \frac{A}{2} e^{-j\varphi} e^{-j\Omega_0 t}\,. \tag{1.4}$$

Plot the signals in equation (1.4).

1.4 Damped sinusoidal signal

Compute and plot damped sinusoidal signal

$$x(t) = Ce^{at}\cos(\Omega_0 t + \varphi), \quad a < 0\,, \tag{1.5}$$

for values of a, C and Ω_0, and the range of t as in Fig.1.3. Observe this signal for different values of a and C.

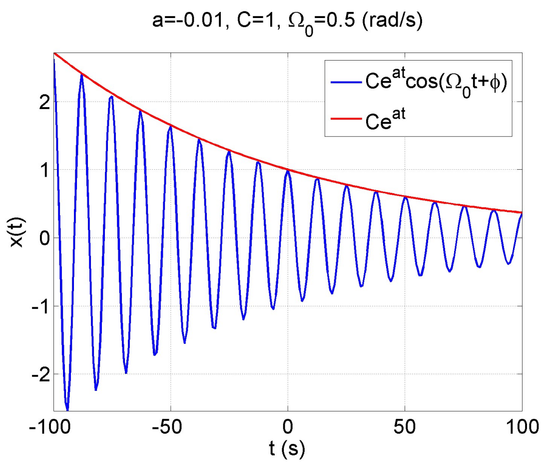

Fig. 1.3 Damped sinusoid (1.5).

1.5 Amplitude and Frequency modulation

Compute and plot sinusoidal signal with sinusoidal amplitude modulation

$$x(t) = (1 + k_{AM}\cos(\Omega_m t))\cos(\Omega_0 t + \varphi)\,, \tag{1.6}$$

for values of Ω_0, Ω_m, and k_{AM} and the range of t as in Fig.1.4. Observe this signal for different values of k_{AM}.

Compute and plot sinusoidal signal with sinusoidal frequency modulation

$$x(t) = \cos(\Omega_0 t + k_{FM}\cos(\Omega_m t))\,, \tag{1.7}$$

for values of Ω_0, Ω_m, and k_{FM} and the range of t as in Fig.1.5. Observe this signal for different values of k_{FM}.

Plot the instantaneous frequency of FM signal (1.7)

$$\Omega_{inst}(t) = \frac{d}{dt}\left(\Omega_0 t + k_{FM}\cos(\Omega_m t)\right) = \Omega_0 - \frac{k_{FM}}{\Omega_m}\sin(\Omega_m t)\,. \tag{1.8}$$

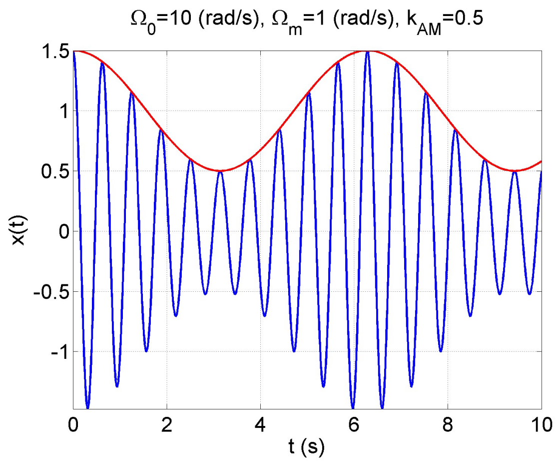

Fig. 1.4 Sinusoidal signal with sinusoidal amplitude modulation (1.6).

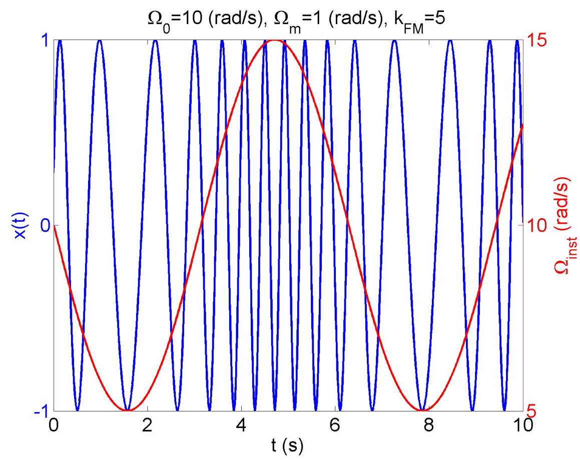

Fig. 1.5 Sinusoidal signal with sinusoidal frequency modulation (1.7), and instantaneous frequency of FM signal (1.8).

2 Continuous-time LTI systems

2.1 Numerical integration

Compute numerically the integral of the signal $x(t) = -t^2 + \frac{1}{2}t + 2$ in the interval $\langle -1,1 \rangle$

$$I = \int_{-1}^{1} x(t)dt = \int_{-1}^{1} (-t^2 + \frac{1}{2}t + 2)dt = \frac{10}{3}. \tag{2.1}$$

Approximate the integral by rectangles (see Fig.2.1)

$$I \approx \sum_{n=-N}^{n=N-1} x(ndt)dt \tag{2.2}$$

and trapezoids (see Fig.2.1)

$$I \approx \sum_{n=-N}^{n=N-1} \frac{x(ndt) + x((n+1)dt)}{2} dt \tag{2.3}$$

where $dt = \frac{1}{N}$ and N is nonnegative integer.

Compute the error of above approximations in dependence of N as shown in Fig. 2.2

$$\varepsilon = \frac{10}{3} - I, \tag{2.4}$$

where I is integral approximation by rectangles (2.2) or integral approximation by trapezoids and 10/3 is the true value of (2.1).

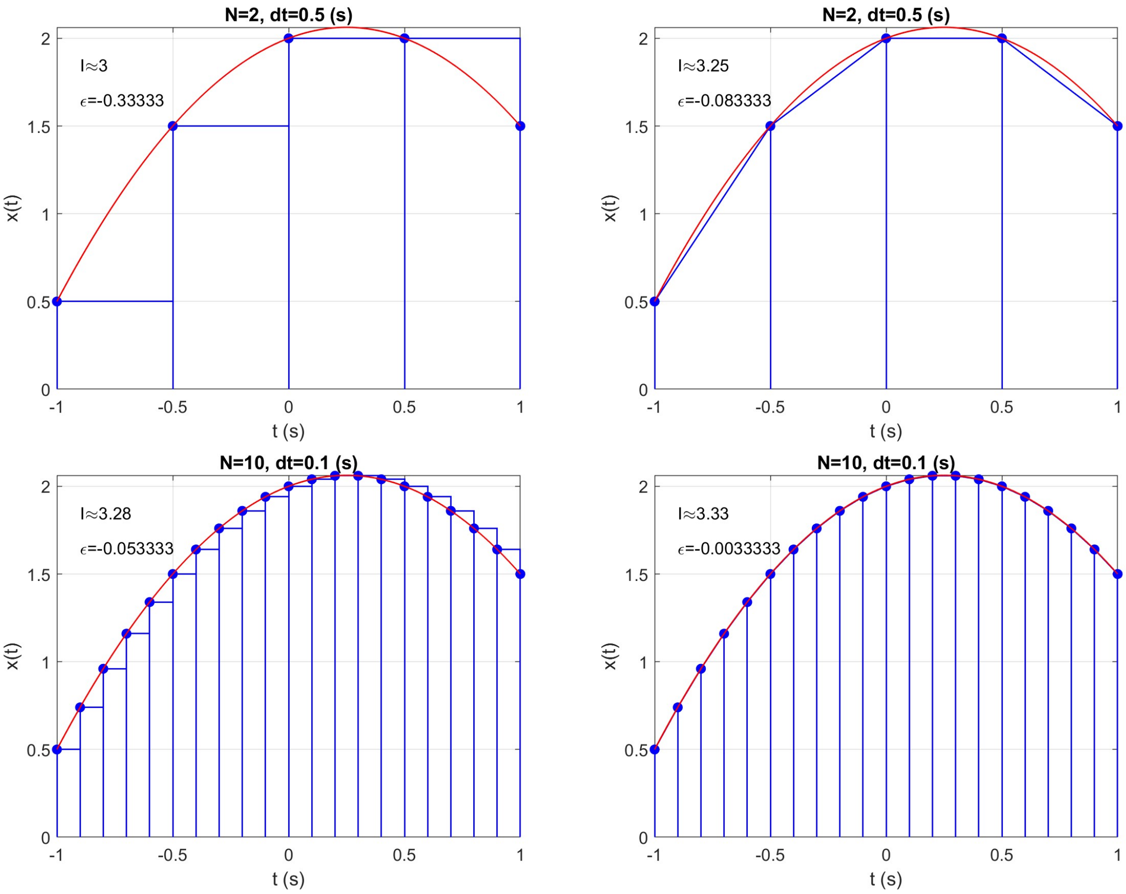

Fig. 2.1 Numerical integration of (2.1).
Left column: rectangle approximation, right column: trapezoids approximation.

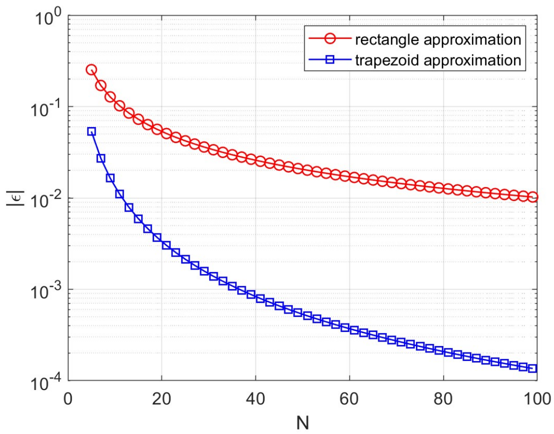

Fig. 2.2 Absolute value of integral approximation error (2.4).

2.2 Convolution integral

Compute numerically the convolution integral

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = x(t) * h(t) \tag{2.5}$$

for signals $x(t) = \begin{cases} 0.5, & |t| < 1 \\ 0, & |t| > 1 \end{cases}$ and $h(t) = \begin{cases} t+1, & |t| < 1 \\ 0, & |t| > 1 \end{cases}$. Plot the results.

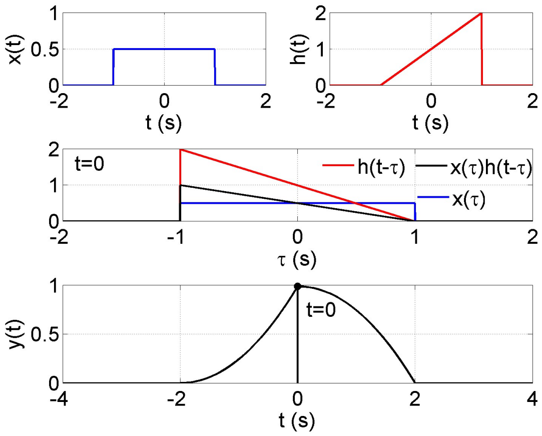

Fig. 2.1 Illustration of computing the convolution integral (2.5).

Compare the results with the true value

$$y(t) = \begin{cases} \frac{1}{4}(t+2)^2, & -2 < t < 0 \\ 1 - \frac{1}{4}t^2, & 0 < t < 2 \\ 0, & |t| > 2 \end{cases} \tag{2.6}$$

2.3 Convolution splines

Compute numerically and plot polynomial (convolution) splines of order P defined as

$$\beta(t)^P = \underbrace{\beta(t)^0 * \beta(t)^0 * \ldots * \beta(t)^0}_{(P+1) \quad \text{times}}, \quad \beta^0(t) = \begin{cases} 1, & |t| < 1 \\ 0, & |t| > 1 \end{cases}. \tag{2.7}$$

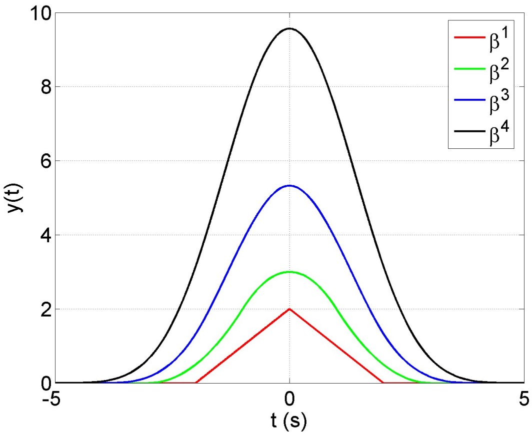

Fig. 2.2 Polynomial (convolution) splines (2.7).

3 Fourier analysis

3.1 The Fourier series

Plot the periodic square wave depicted in Fig. 3.1

$$x(t)=\begin{cases}1, & |t|<T_1\\ 0, & T_1<|t|<\dfrac{T_0}{2}\end{cases}. \tag{3.1}$$

along with the Fourier series coefficients of this signal

$$a_k=\begin{cases}\dfrac{2T_1}{T_0}, & k=0\\ \dfrac{\sin(k\Omega_0 T_1)}{k\pi}, & k\neq 0\end{cases}. \tag{3.2}$$

for T_0=1, T_1=0.25 (symmetric square wave), and T_0=1, T_1=0.1, and the Fourier series coefficients from a_{-15} to a_{15}.

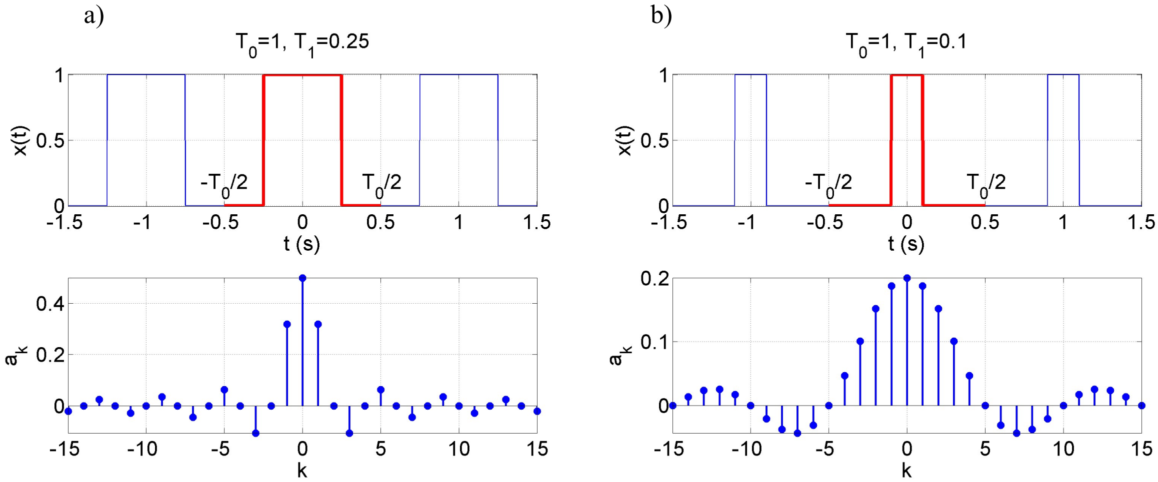

Fig. 3.1 Periodic square wave (3.1) and the Fourier series coefficients (3.2) from a_{-15} to a_{15} for: a) T_0=1, T_1=0.25 (symmetric square wave), and b) T_0=1, T_1=0.1.

Approximate the periodic symmetric square wave (3.1) by considering only a finite number of the Fourier series coefficients, as depicted in Fig. 3.2

$$x_N(t)=\sum_{k=-N}^{N} a_k e^{jk\Omega_0 t}. \tag{3.3}$$

Compute and plot approximation error

$$E_N=\int_{T_0}|x(t)-x_N(t)|^2\,dt. \tag{3.4}$$

as a function of N (3.3), see Fig. 3.3.

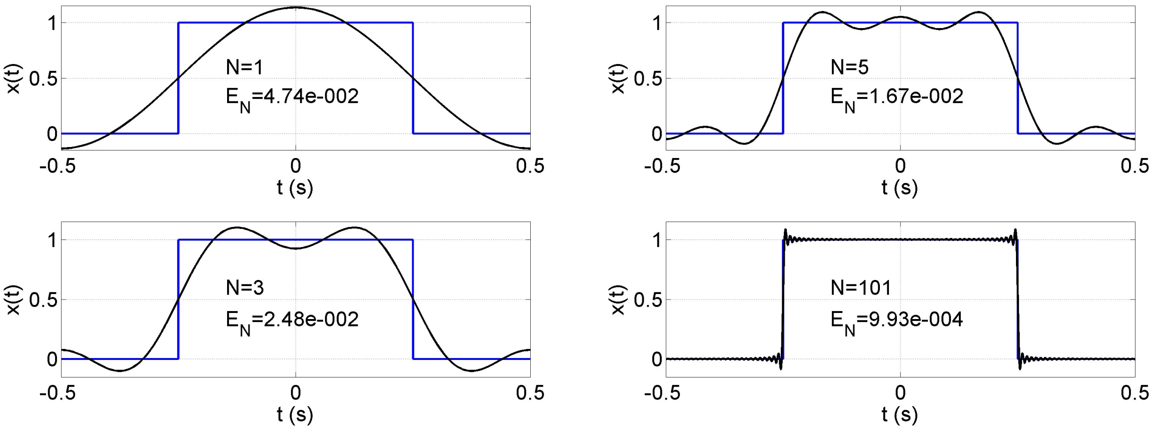

Fig. 3.2 Approximation of the periodic symmetric square wave (3.1) by a finite number of the Fourier series coefficients (3.3).
E_N is the energy in the approximation error over one period (3.4).

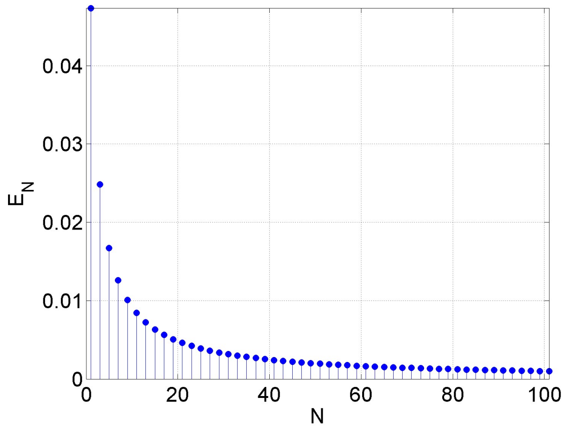

Fig. 3.3 The energy E_N in the approximation error over one period (3.4) for symmetric square wave in dependence of the finite number of the Fourier series coefficients N (3.3)
(actual number of the Fourier coefficients is $2N$+1).

Plot one period on the periodic signals that violate Dirichlet conditions:

1) Condition 1: $x(t) = \dfrac{1}{t}, \quad 0 < t \le 1$ (Fig.3.4 top),

2) Condition 2: $x(t) = \sin\left(\dfrac{2\pi}{t}\right), \quad 0 < t \le 1$ (Fig.3.4 middle),

3) Condition 3: The signal composed of an infinite number of sections each of which is half the height and half the width of the previous section (Fig.3.4 bottom).

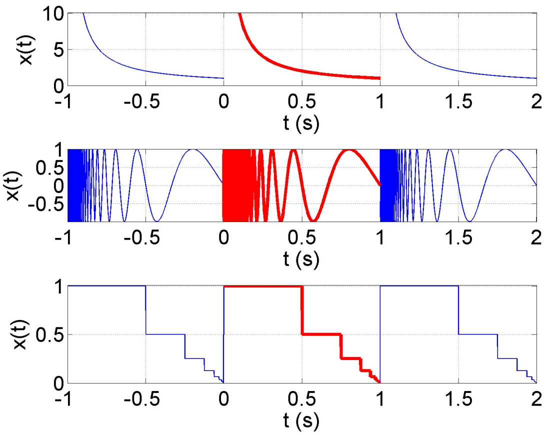

Fig. 3.4 Periodic signals that violate Dirichlet conditions.

3.2 The Fourier integral

Compute and plot the Fourier transform

$$X(\Omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\Omega t} dt \tag{3.5}$$

of the finite length signals:

- finite rectangular impulse

$$x_1(t) = \begin{cases} 1, & |t| < 1 \\ 0, & |t| > 1 \end{cases}, \tag{3.6}$$

- fragment of sinusoidal signal

$$x_2(t) = \begin{cases} A\cos(\Omega_0 t + \varphi), & |t| < 1 \\ 0, & |t| > 1 \end{cases}, \tag{3.7}$$

- fragment of sinusoidal signal with sinusoidal AM

$$x_3(t) = \begin{cases} (1 + k_{AM} \cos(\Omega_m t)) \cos(\Omega_0 t + \varphi), & |t| < 1 \\ 0, & |t| > 1 \end{cases}. \tag{3.8}$$

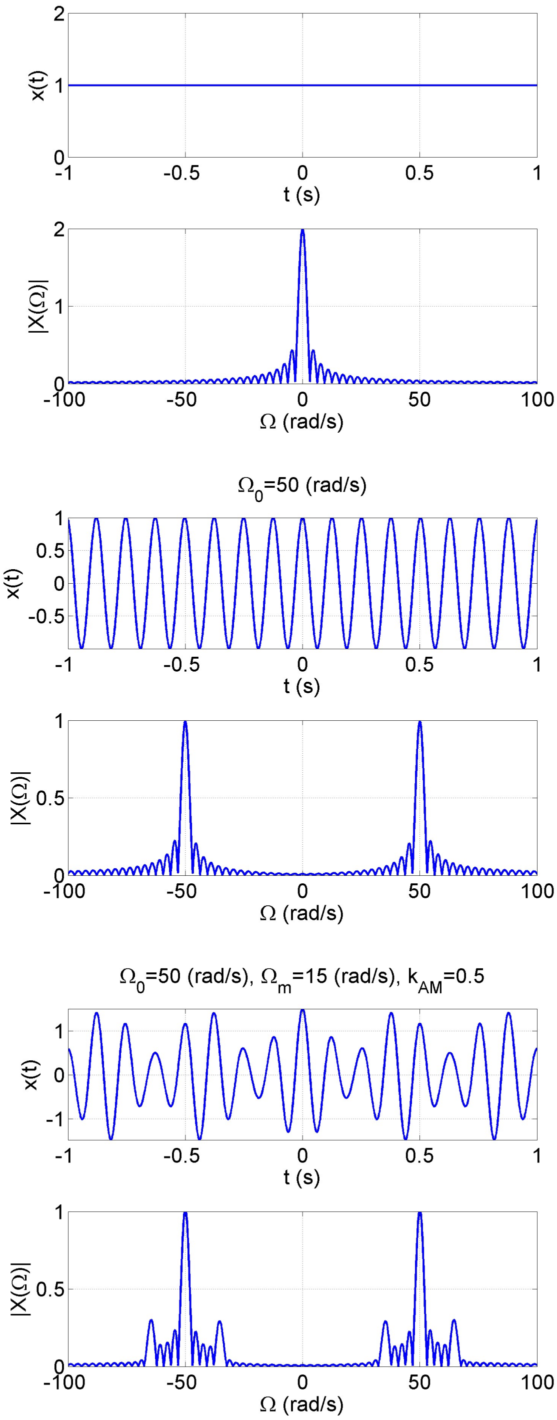

Fig. 3.5 Finite duration signals (3.6-3.8) and theirs Fourier transforms (3.5).

4. Continuous-time filters

4.1 Notch filter

Chose the parameters a, b, and c of the following transmittance $H(s)$

$$H(s) = \frac{(s + a + jb)(s + a - jb)}{(s + c + jb)(s + c - jb)} \tag{4.1}$$

to obtain strong attenuation (elimination) of the signal with frequency 50 Hz. Plot zeros and poles of this transmittance. Observe impulse response and magnitude response of the filter.

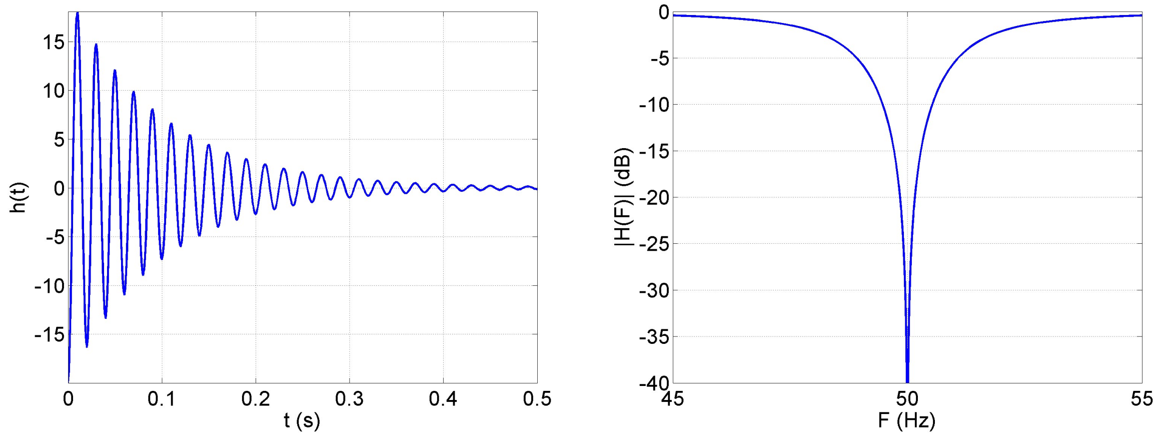

Fig. 4.1 Impulse response and frequency response of exemplary notch filter with the transmittance H(s) (4.1).

4.2 Frequency response

Write Matlab function for computing frequency response of the continuous-time system with transmittance $H(s)$. Use the following definition

```
function H=freqs_lab(B,A,Om);
%Frequency response of transmittance H(s)=B(s)/A(s)
%Om - frequency in rad/s, s=j*Om
```

Compare the results with Matlab function **`freqs`**.

4.3 Butterworth, Chebyshev Type I, Chebyshev Type II, Elliptic, and Bessel filters

Design continuous-time lowpass filter having following parameters:
- passband corner frequency F_{pass}=1 kHz.
- stopband corner frequency F_{stop}=1.5 kHz.
- passband ripple R_p = 1 dB.
- stopband ripple R_s = 30 dB.

For the above requirements give the transmittance $H(s)$ of
- Butterworth filter,
- Chebyshev Type I filter,
- Chebyshev Type II filter,
- Elliptic filter,
- Bessel filter.

What are the orders of those filters?
Plot zeros and poles of those filters.
Plot in one figure magnitude responses of those filters. Verify if the filters fulfill specified requirements.
Plot, in one figure, phase characteristics of those filters.
Plot, in one figure, impulse responses of those filters.

Use Matlab functions **`butter`**, **`cheby1`**, **`cheby2`**, **`ellip`**, **`besself`**.

5. Discrete-time signals

5.1 Sinusoidal signal

Compute sinusoidal signal

$$x[n] = A\cos(\omega_0 n + \phi)\,, \tag{5.1}$$

observe this signal for different values of ω_0 rad. Compare two signals with frequencies ω_0 rad and $\omega_0+2\pi$ rad.

Compute sinusoidal signal

$$x[n] = A\cos(2\pi \frac{F_0}{F_s} n + \phi)\,, \tag{5.2}$$

observe this signal for different values of $F_0<F_s/2$ Hz.

Compute sinusoidal signal by using the following difference equation

$$x[n] = 2\cos(\omega_0)x[n-1] - x[n-2]\,. \tag{5.3}$$

5.2 Damped sinusoidal signal

Compute damped sinusoidal signal by using the following difference equation

$$x[n] = 2\cos(\omega_0)e^{-d}x[n-1] - e^{-2d}x[n-2]\,. \tag{5.4}$$

Compare the results with the signal computed from definition

$$x[n] = A\cos(\omega_0 n + \phi)e^{-dn}\,. \tag{5.5}$$

6. Discrete-time systems

6.1 Convolution sum

Write the program implementing convolution sum

$$y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]=x[n]*h[n]\,. \tag{6.1}$$

Compare the results with Matlab function **`conv`**.

6.2 Linear Constant-Coefficient Difference Equation

Write the program implementing Nth-order linear constant-coefficient difference equation

$$y[n]=\sum_{m=0}^{M}\frac{b[m]}{a[0]}x[n-m]-\sum_{k=1}^{N}\frac{a[k]}{a[0]}y[n-k]\,. \tag{6.2}$$

Compare the results with Matlab function **`filter`**.

6.3 Notch filter

Chose the parameters r_1, r_2, and ω of the following transmittance $H(z)$

$$H(z)=\frac{(z-r_1e^{j\omega})(z-r_1e^{-j\omega})}{(z-r_2e^{j\omega})(z-r_2e^{-j\omega})}, \tag{6.3}$$

to obtain strong attenuation (elimination) of the signal with frequency 50 Hz for the sampling frequency F_s=200 Hz. Observe impulse response and magnitude response of the filter.

6.4 Frequency response

Write Matlab function for computing frequency response of the discrete-time system with transmittance $H(z)$. Use the following definition

```
function H=freqz_lab(B,A,w);
%Frequency response of transmittance H(z)=B(z)/A(z)
%w - frequency in rad, z=exp(j*w)
```

Compare the results with Matlab function **`freqz`**.

6.5 IIR filters

Design discrete-time IIR lowpass filter having following parameters for the sampling frequency F_s=5 kHz:
- passband corner frequency F_{pass}=1 kHz.
- stopband corner frequency F_{stop}=1.5 kHz.
- passband ripple R_p = 1 dB.
- stopband ripple R_s = 30 dB.

For the above requirements give the transmittance $H(z)$ of
- Butterworth filter,
- Chebyshev Type I filter,
- Chebyshev Type II filter,
- Elliptic filter.

Use Matlab functions **`butter, cheby1, cheby2,`** and **`ellip.`**
What are the orders of those filters?
Plot zeros and poles of those filters.
Plot, in one figure, magnitude responses of those filters. Verify if the filters fulfill specified requirements.
Plot, in one figure, phase characteristics of those filters.
Plot impulse responses of those filters (use function **`filter`** for computing impulse response and **`stem`** for plotting impulse response).

6.6 FIR filters

Design by window method lowpass FIR filter that for the sampling frequency F_s=4 kHz has cutoff frequency f_c=1 kHz. The impulse response of the filter is given by

$$h_{LP}[n] = \frac{\sin(\omega_c n)}{\pi n} = \begin{cases} \sin(\omega_c n)/(\pi n), & n \neq 0 \\ \omega_c / \pi, & n = 0 \end{cases}, \tag{6.4}$$

where ω_c is a cutoff frequency in radians.

Plot, in one figure, magnitude responses of the filter designed with rectangular window for different window length M (e.g. M={11, 51, 101}).

Plot, in one figure, magnitude responses of the filters designed with different windows (e.g. rectangular and Hamming) for the same length M.

Design discrete-time FIR lowpass filter having following parameters for the sampling frequency F_s=5 kHz:
- passband corner frequency F_{pass}=1 kHz.
- stopband corner frequency F_{stop}=1.5 kHz.
- passband ripple R_p = 1 dB.
- stopband ripple R_s = 30 dB.

Design the filter by window method with the Kaiser window (Matlab functions **`kaiserord`** and **`fir1`**) and by Parks-McClellan algorithm (Matlab function **`firpm`**).
What are the lengths (orders) of those filters?
Plot, in one figure, magnitude responses of those filters. Verify if the filters fulfill specified requirements.

7. DTFT and DFT

7.1 DTFT

Write Matlab program for computing DTFT

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} . \tag{7.1}$$

Compute and plot magnitude characteristic of the following finite length signals

- constant signal $x_M[n] = 1, \quad -M \le n \le M$,
- complex exponential signal $x_M[n] = e^{j\omega_0 n}, \quad -M \le n \le M$,
- sinusoidal signal $x_M[n] = \cos(\omega_0 n), \quad -M \le n \le M$.

7.2 DFT definition

Write Matlab program for computing DFT (7.2) and inverse DFT (7.3)

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi/N)kn} \ , \ k = 0,1,,\ldots,N-1 \tag{7.2}$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j(2\pi/N)kn} \ , \ n = 0,1,,\ldots,N-1 . \tag{7.3}$$

Compare the results with Matlab function **`fft`** for a chosen test signal (e.g. sinusoidal signal). Plot the reconstruction error defined as

$$\varepsilon = x[n] - \mathrm{IDFT}\{\mathrm{DFT}\{x[n]\}\} . \tag{7.4}$$

7.3 DFT analysis

Consider discrete-time test signal

$$x[n] = DC + A\cos(2\pi Ft + \varphi) \ , \ n = 0,1,2,\ldots,N-1 , \tag{7.5}$$

where F=20 Hz, DC=1, A=1, $\varphi=\pi/5$ and sampling frequency F_s=100 Hz.

For the test signal (7.5) compute DFT (Matlab function **`fft`**) from N=32 and N=64 samples. Plot, in one figure, magnitude characteristics for both cases. Scale the frequency axis in hertz, and magnitude axis in magnitude values (i.e. read the value of DC and A from the plot).

For the test signal (7.5) compute DFT from N=64 samples, and from N=64 samples after appending zeros to the length of 1024 samples. Plot, in one figure, magnitude characteristics for both cases. Scale the frequency axis in hertz, and magnitude axis in magnitude values.

For the test signal (7.5) with the rectangular window and with the Hamming window compute DFT from N=64 samples after appending zeros to the length of 1024 samples. Plot, in one figure, magnitude characteristics for both cases. Scale the frequency axis in hertz, and magnitude axis in magnitude values.

Consider discrete-time test signal

$$x[n] = A_1 \cos(2\pi F_1 t + \varphi_1) + A_2 \cos(2\pi F_2 t + \varphi_2)\,,\ n = 0,1,2,\ldots,N-1\,, \qquad (7.6)$$

where F_1=20 Hz and sampling frequency F_s=100 Hz.

For the test signal (7.6) compute DFT from N=256 samples after appending zeros to the length of 1024 samples. Observe magnitude spectra for A_1=A_2=1, and F_2 in the interval (20, 40) Hz.

For the test signal (7.6) compute DFT from N=256 samples after appending zeros to the length of 1024 samples. Observe magnitude spectra for A_1=1, F_2=24 Hz and A_2 in the interval (0, A_1) Hz.

For the test signal (7.6) compute DFT from the number of samples from N=256 to N=2048 after appending zeros to the length of 2048 samples. Observe magnitude spectra for A_1=A_2=1, and F_2=22 Hz.

7.4 Spectrogram

Write Matlab function for computing spectrogram

Use the following function definition

```
function [X,n_spe,w_spe]= specgram_lab(x,w,Nfft,R);
%x - signal
%w - time window with the length L
%Nfft - DFT length >= L
%R - window shift in samples 1<=R<=L
%X - spectrogram time-frequency plane
%n_spe - time index
%w_spe - frequency in radians
```

Observe spectrograms of sinusoidal signal with amplitude modulation (1.6) and frequency modulation (1.7). Compare the results with Matlab function **`spectrogram`**.

8. Signal filtration

8.1 Filtering desired frequency component

Use Matlab function **`square`** to compute the fragment of the periodic square wave with following parameters: sampling frequency F_s=100 Hz, number of samples N=1e3, and period T=1 s. Design the bandpass filter of your choice and filter the 3rd harmonic of this signal. Observe the signals before and after filtration and theirs spectra, see Fig. 8.1.

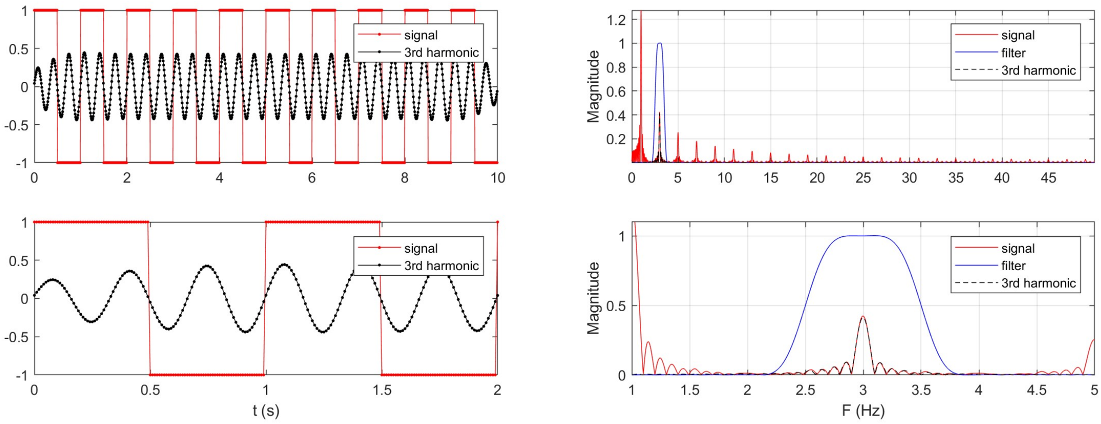

Fig. 8.1 Left: Fragment of periodic square wave and filtered the 3rd harmonic of this signal.
Right: Magnitude spectra of the periodic square wave, applied bandpass filter, and filtered the 3rd harmonic.

8.2 Attenuating undesired frequency component

Load the ECG signal by **`load('ecgsig.mat')`**. The signal has a sampling frequency of 360 Hz. Select the first N=1e3 samples of the signal. Distort the ECG signal by adding 50 Hz sinusoidal signal with amplitude 0.2. Remove sinusoidal distortion by filtration. Observe the signals before and after filtration and theirs spectra, see Fig. 8.2.

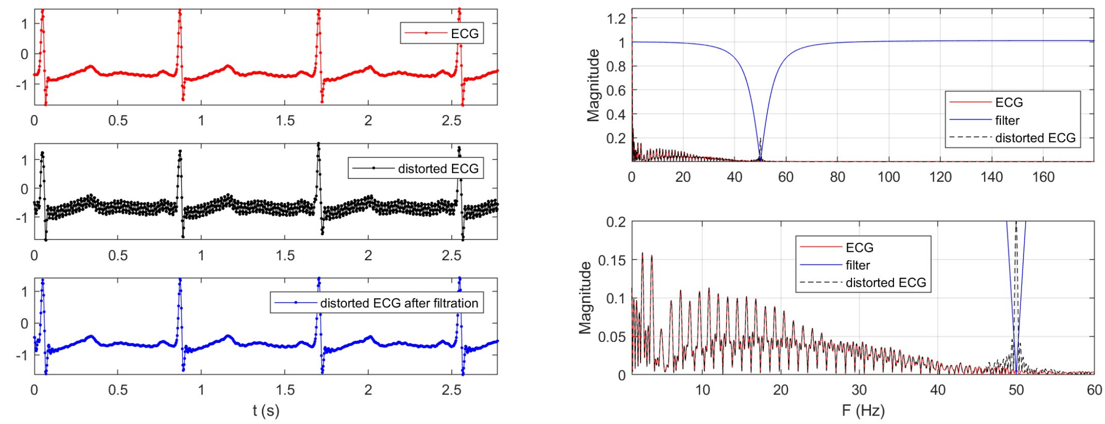

Fig. 8.2 Left: From top to bottom: the ECG signal, the ECG signal distorted by additive 50 Hz sinusoid, and distorted ECG signal after filtration.
Right: Magnitude spectra of the ECG signal, applied filter, and ECG signal distorted by 50 Hz sinusoid.

8.3 Lowpass filtering of noisy signal

Load the ECG signal by `load('ecgsig.mat')`. The signal has a sampling frequency of 360 Hz. Select the first N=1e3 samples of the signal. Distort the ECG signal by additive zero mean Gaussian noise with standard deviation 0.1 (use Matlab function `randn`). Filter the distorted signal by lowpass filter of your choice to reduce the noise in the signal. Observe the signals before and after filtration and theirs spectra, see Fig. 8.3.

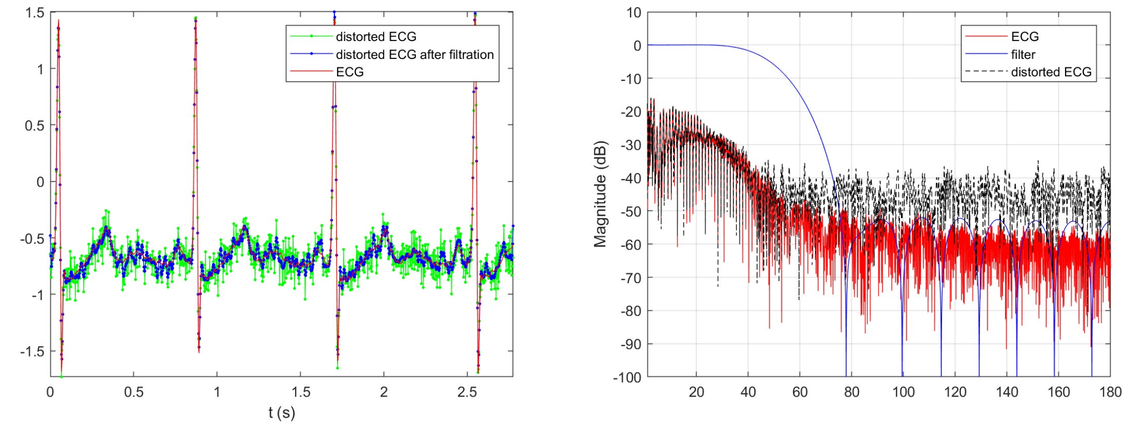

Fig. 8.3 Left: The ECG signal, the ECG signal distorted by AWGN (Additive White Gaussian Noise), and distorted ECG signal after filtration.
Right: Magnitude spectra of the ECG signal, applied lowpass filter, and ECG signal distorted by AWGN.

9. Signal resampling

Compute N=1e3 samples of sinusoidal signal with sinusoidal amplitude modulation

$$x[n]=\left(1+k_{AM}\cos(2\pi\frac{F_m}{F_s}n)\right)\cos(2\pi\frac{F_0}{F_s}n),\quad n=0,1,\ldots,N-1, \tag{9.1}$$

for sampling frequency F_s=2e3 Hz, carrier frequency F_0=100 Hz, modulating frequency F_m=10 Hz, and modulating coefficient k_{AM}=0.75. Resample this signal to obtain sampling frequency F_{s1}=750 Hz, and F_{s2}=3500 Hz. Observe the signals and theirs spectra for all sampling frequencies, see Fig. 9.1. Use Matlab function `resample`.

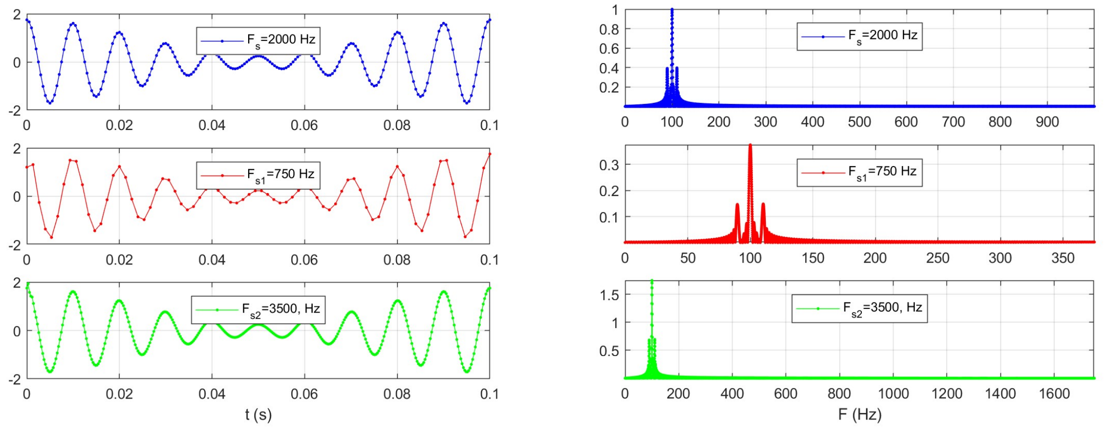

Fig. 9.1 Left: Beginning fragment of the signals, from top to bottom: original signal sampled at 2e3 Hz, signal resampled to 750 Hz, and signal resampled to 3500 Hz.
Right: Magnitude spectra of the signals from the left plot.

Compute N=1e3 samples of multicomponent sinusoidal signal

$$x[n]=\cos(2\pi\frac{F_0}{F_s}n)+\cos(2\pi\frac{2.5F_0}{F_s}n)+\cos(2\pi\frac{4.5F_0}{F_s}n),\quad n=0,1,\ldots,N-1, \tag{9.2}$$

for sampling frequency F_s=2e3 Hz, and frequency F_0=100 Hz. Resample this signal to obtain sampling frequency F_{s1}=750 Hz, and F_{s2}=3500 Hz. Observe the signals and theirs spectra for all sampling frequencies, see Fig. 9.2.

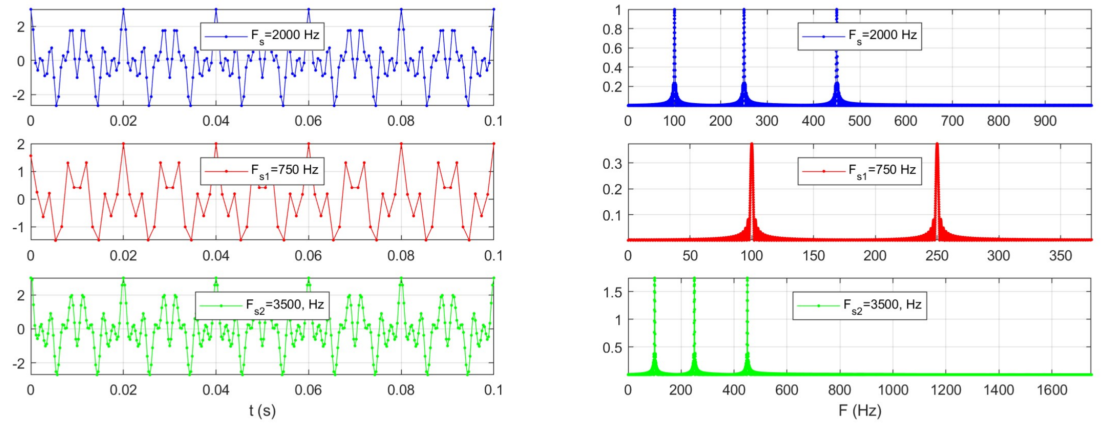

Fig. 9.2 Left: Beginning fragment of the signals, from top to bottom: original signal sampled at 2e3 Hz, signal resampled to 750 Hz, and signal resampled to 3500 Hz.
Right: Magnitude spectra of the signals from the left plot.

10. Periodogram, modified periodogram, and periodogram averaging

Write Matlab function named **periodogram_lab** for computing modified periodogram according to

$$I[k] = \frac{1}{LU} \,|\, \mathrm{DFT}\{w[n]x[n]\} \,|^2, \; U = \frac{1}{L}\sum_{n=0}^{L-1}(w[n])^2 \; . \qquad (10.1)$$

where $w[n]$ is applied time window, and L is the length of the window. When the window $w[n]$ is rectangular window the power spectrum estimator $I[k]$ is called *periodogram*. If the window is not rectangular, $I[k]$ is called *modified periodogram*.

Use the following function definition

```
function [Ik,om] = periodogram_lab(x,w,Nfft)
%Ik - periodogram
%om - frequency in rad
%x - signal
%w - window
%Nfft - DFT length
```

Compute the periodogram of Relative Sunspot Numbers, that can be loaded in Matlab by **load sunspot.dat**, for several time windows, e.g. the rectangular window, the Hann window, and the Hamming window. Estimate periodicity of the sun activity. Compare the results with Matlab function **periodogram**.

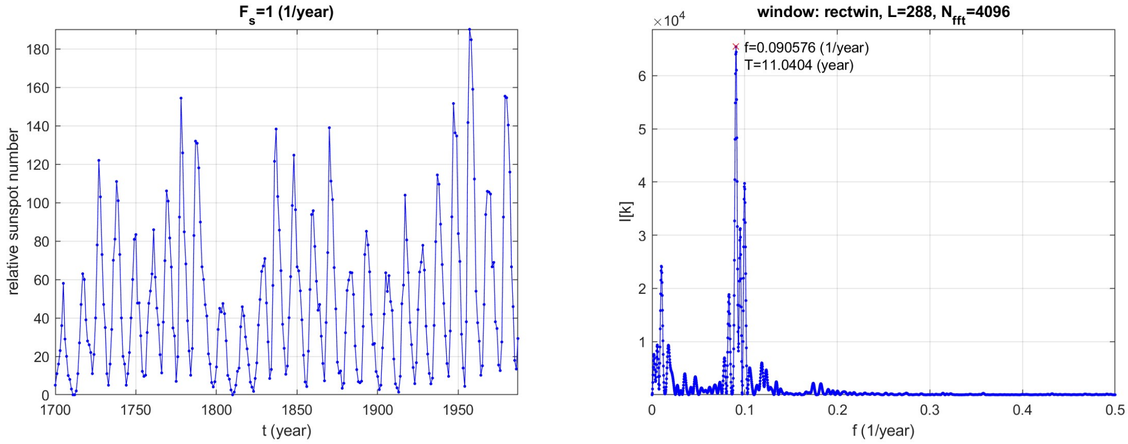

Fig. 10.1 Left: Relative sunspot number data sampled yearly between 1700 and 1987.
Right: Periodogram of relative sunspot numbers after DC value subtraction.

Write Matlab function named **`periodogram_averaging`** for periodogram averaging according to

$$\hat{I}[k] = \frac{1}{K}\sum_{r=0}^{K-1} I_r[k]\,. \tag{10.2}$$

where $I_r[k]$ are periodograms computed in the window having length of L samples shifted along analyzed signal by R samples

$$I_r[k] = \frac{1}{LU}\,|\,\mathrm{DFT}\{w[n]x[rR+n]\}\,|^2, \quad n = 0,1,2,\ldots L-1\,. \tag{10.3}$$

When the window $w[n]$ is rectangular window, and successive windows do not overlap, i.e. R=L, the method of averaging periodograms is called *Bartlett's procedure*. For arbitrary window $w[n]$ and 50% overlap of successive windows, i.e. R=L/2 , the method of averaging periodograms is called *Welch's procedure*.

Use the following function definition

```
function [Ik,om] = periodogram_averaging(x,w,Nfft,R)
%Ik - averaged periodogram
%om - frequency in rad
%x - signal
%w - window
%Nfft - DFT length
%R - window overlap in samples
```

Compute N=1e3 samples of sinusoidal signal with amplitude A=1, frequency ω=0.1 rad, and random phase disturbed by zero mean AWGN (Additive White Gaussian Noise) with standard deviation σ=1.5. Observe averaged periodograms of this signal for chosen widows of different kind (e.g. the rectangular window, and the Hammnig window) and different length L and different overlap R, see Fig. 10.2. Compare the results with Matlab function **`pwelch`**.

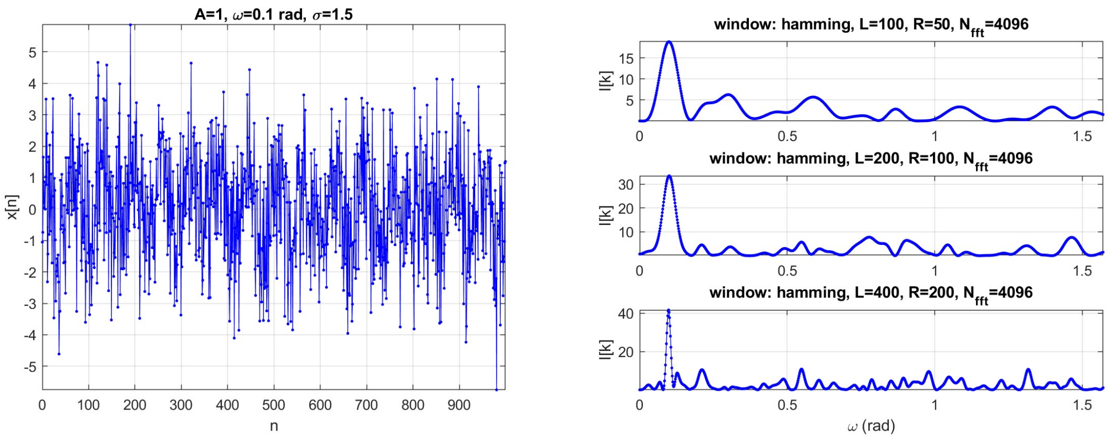

Fig. 10.2 Left: Sinusoidal signal distorted by AWGN (Additive White Gaussian Noise).
Right: Exemplary averaged periodograms of this signal.

11. Frequency and damping estimation

Compute N=1e3 samples of two discrete-time damped sinusoidal signals

$$x[n] = A\cos(\omega_0 n + \phi)e^{-dn},\ n = 0,1,\dots,N-1 \tag{11.1}$$

for amplitude A=1.5, damping d=0.001, phase ϕ=0.3 rad, and two different frequencies $\omega_0 = \frac{2\pi}{N}(10-0.4)$ rad, and $\omega_0 = \frac{2\pi}{N}(10+0.4)$ rad. Use the DFT for estimation of parameters ω_0, d, A, and ϕ of those two signals.

Damping and frequency are estimated as

$$d_E = -\mathrm{Re}\{\ln(\lambda)\}, \tag{11.2}$$

$$\omega_{0E} = \mathrm{Im}\{\ln(\lambda)\}. \tag{11.3}$$

The amplitude and phase are estimated as

$$A_E = 2\,|X[k_{\max}]|\left|\frac{1-\lambda e^{-j\omega_{k_{\max}}}}{1-\lambda^N}\right|, \tag{11.4}$$

$$\varphi_E = \mathrm{angle}\left(X[k_{\max}]\frac{1-\lambda e^{-j\omega_{k_{\max}}}}{1-\lambda^N}\right), \tag{11.5}$$

Where

$$\lambda = \begin{cases} e^{j\omega_{k\max}}\dfrac{1-\dfrac{X[k_{\max}+1]}{X[k_{\max}]}}{1-\dfrac{X[k_{\max}+1]}{X[k_{\max}]}e^{-j2\pi/N}}, & if \quad |X[k_{\max}+1]|>|X[k_{\max}-1]| \\ e^{j\omega_{k_{\max}}}\dfrac{1-\dfrac{X[k_{\max}-1]}{X[k_{\max}]}}{1-\dfrac{X[k_{\max}-1]}{X[k_{\max}]}e^{j2\pi/N}}, & if \quad |X[k_{\max}-1]|>|X[k_{\max}+1]| \end{cases}, \tag{11.6}$$

k_{max} is the index of the DFT bin with the highest magnitude, and $\omega_k = \frac{2\pi}{N}k$.

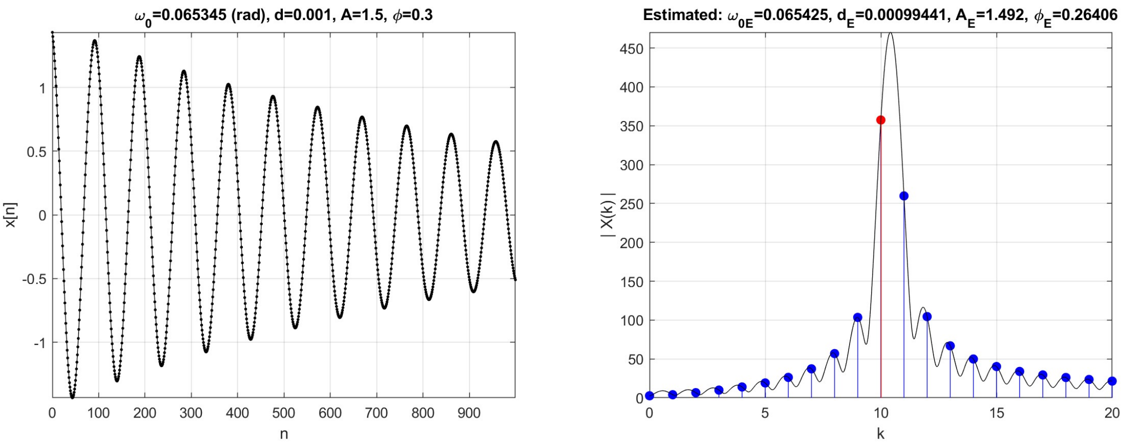

Fig. 11.1 Left: Damped sinusoidal signal.
Right: Magnitude spectrum; red bin denotes $|X[k_{max}]|$; $|X[k_{max}+1]|>|X[k_{max}-1]|$.

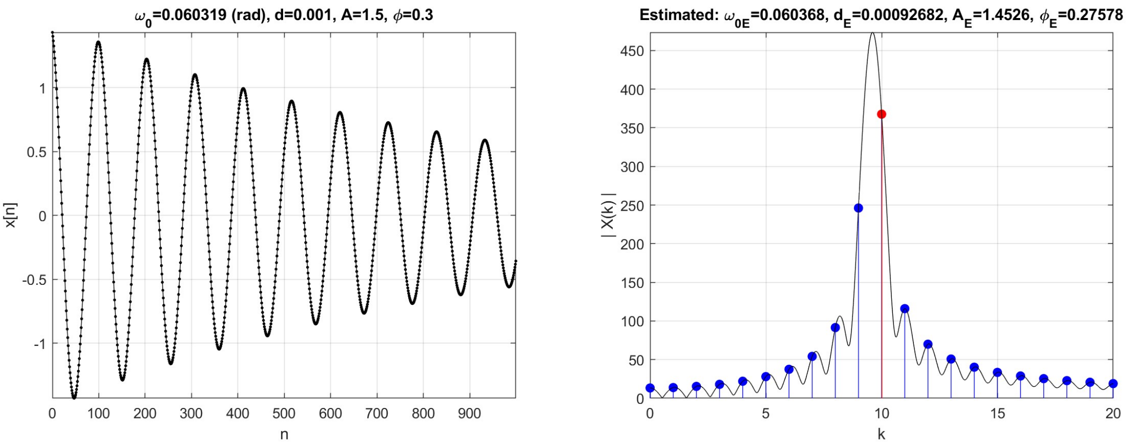

Fig. 11.2 Left: Damped sinusoidal signal.
Right: Magnitude spectrum; red bin denotes $|X[k_{max}]|$; $|X[k_{max}-1]|>|X[k_{max}+1]|$.

Compute N=1e2 samples of discrete-time signal containing two damped sinusoidal components

$$x[n] = A_1 \cos(\omega_1 n + \phi_1) e^{-d_1 n} + A_2 \cos(\omega_2 n + \phi_2) e^{-d_2 n},\ n = 0,1,\ldots,N-1 \tag{11.7}$$

where A_1=1.5, ω_1=0.1 rad, d_1=1e-3, ϕ_1=0, and A_2=2, ω_2=0.25 rad, d_2=3e-3, ϕ_2=1.5. This signal ad its magnitude spectrum is shown in Fig. 10.3. Use the DFT for estimation of frequency, damping, amplitude, and phase of each component of this signal.

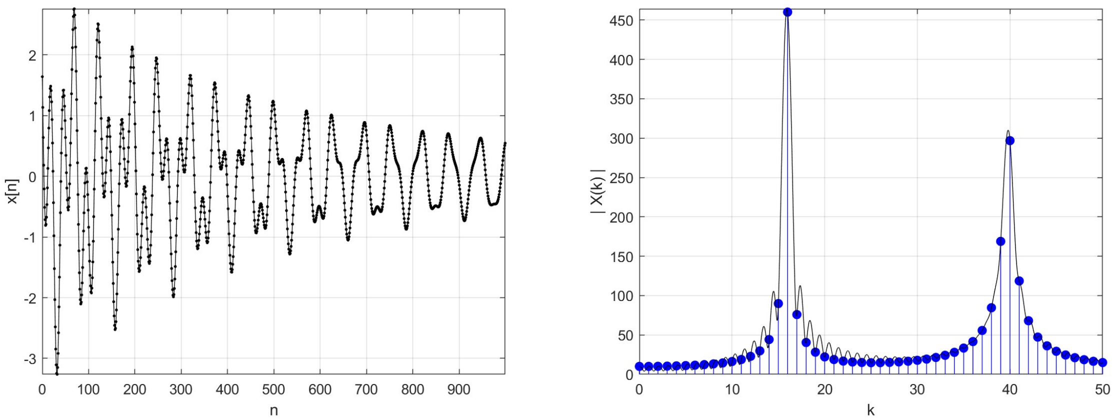

Fig. 11.2 Left: Signal containing two damped sinusoidal components.
Right: Magnitude spectrum of this signal.

12. Amplitude and phase demodulation

12.1 Amplitude demodulation

Compute sinusoidal signal with sinusoidal amplitude modulation

$$x[n] = (1 + k_{AM}\, m[n])\cos(\omega_0 n)\,,\; 0 < k_{AM} \le 1\,, \tag{12.1}$$

where $m[n]=\cos(\omega_m n)$, $\omega_m << \omega_0$, and use the discrete time analytic signal for demodulation of instantaneous amplitude (envelope) of this test signal. Compute the discrete time analytic signal by Matlab function `hilbert`. Instantaneous amplitude is the magnitude of discrete time analytic signal.

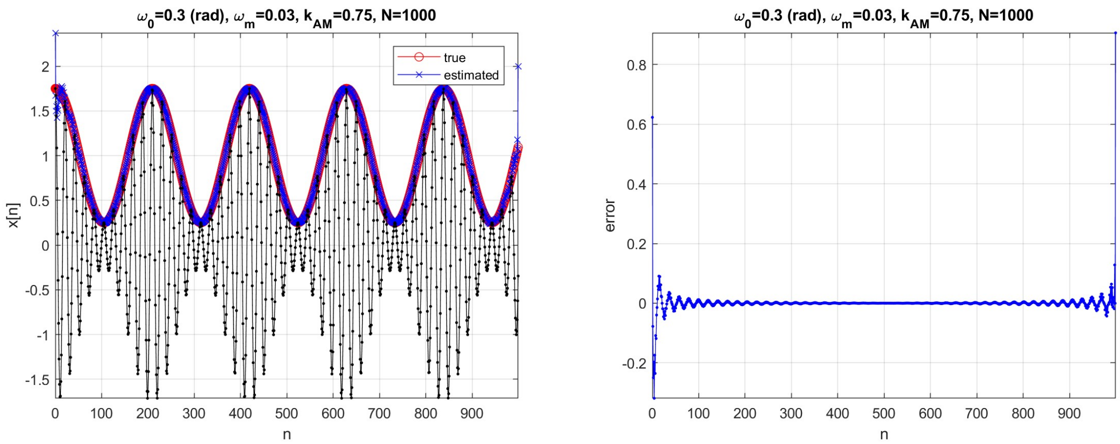

Fig. 12.1 Left: Sinusoidal signal with sinusoidal amplitude modulation.
Right: The error of instantaneous amplitude (envelope) estimation by analytic signal.

12.2 Phase demodulation

Compute sinusoidal signal with sinusoidal frequency modulation

$$x[n] = \cos(\omega_0 n + k_{FM} \cos(\omega_m n) / \omega_m)\,. \tag{12.2}$$

The instantaneous frequency of this signal (i.e. time derivative of the cosine function argument) is $m[n] = \omega_0 - k_{FM} \sin(\omega_m n)$. Use the discrete time analytic signal (Matlab function **`hilbert`**) for demodulation of instantaneous frequency. Instantaneous frequency is the derivative of the phase of discrete time analytic signal.

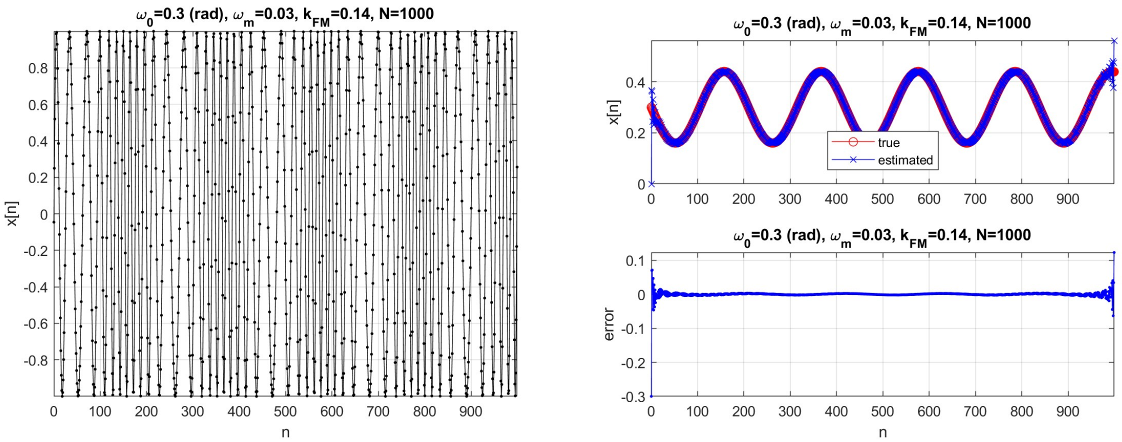

Fig. 12.2 Left: Sinusoidal signal with sinusoidal frequency modulation.
Right: True and estimated instantaneous frequency and the error of instantaneous frequency estimation by analytic signal.

References

[Britt93] C. Britton Rorabaugh, *Digital Filter Designer's Handbook: Featuring C Routines*, McGraw-Hill Education - Europe 1993.

[Kay93] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, 1993.

[Lyons04] R. G. Lyons, *Understanding Digital Signal Processing*, Second Edition, Prentice-Hall, 2004.

[Lyons99] Lyons R.G.: *Wprowadzenie do cyfrowego przetwarzania sygnałów*, WKŁ, Warszawa 1999.

[MatFD] *Matlab Filter Design Toolbox*, available from www.mathworks.com.

[MatIP] *Matlab Image Processing Toolbox*, available from www.mathworks.com.

[MatSP] *Matlab Signal Processing Toolbox*, available from www.mathworks.com.

[MatWT] *Matlab Wavelet Toolbox*, available from www.mathworks.com.

[Moon99] Moon T. K., Stirling W. C., *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall 1999.

[Oppen83] A. V. Oppenheim, A. S. Willsky, I. T. Young, *Signals and Systems*, Prentice-Hall, 1983.

[Oppen99] A.V. Oppenheim, R.W. Schafer, J.R. Buck, *Discrete-Time Signal Processing*, 2nd Ed., Prentice-Hall, 1999.

[Pap62] A. Papoulis, *The Fourier integral and its applications*, McGraw-Hill Book Company, Inc. 1962.

[Phadke09] A. G. Phadke, J. S. Thorp, *Computer Relaying for Power Systems*, Second Edition, John Wiley & Sons Ltd, 2009.

[Poll99] D.S.G. Pollock, *A Handbook of Time-Series Analysis, Signal Processing and Dynamics*, ACADEMIC PRESS, 1999.

[Spe68] M.R. Speigel, *Mathematical Handbook of Formulas and Tables*. Schaum's Outline Series, McGraw-Hill, 1968.

[Stein00] Jonathan Y. Stein, *Digital Signal Processing: A Computer Science Perspective*, John Wiley & Sons, Inc. 2000.

[Ziel05] T.P. Zieliński, *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, WKŁ, Warszawa 2005, 2007, 2009.