Signal Analysis: Homework 2

This is the second out of three homework sheets for the 2019 Signal Analysis problem class. Each homework sheet includes three different examples resulting in a total number of nine homework examples over progress of the course. The presented homework examples are mostly MATLAB examples with a strong real world focus.

The grading of the problem class is based on student presentations (Power Point or Latex presentation) and exercise interviews for five out of the nine examples. For each of the three homework sheets, students have to present one example, which will be selected by the lecturers ("must show"). Students will be informed three days prior to the deadline of the problem sheet, about the number of the example, which has to be presented. The remaining two examples can be selected by the students itself ("will show"). Hereby only one of the remaining two examples per problem sheet can be selected. Note, that the grading is an individual grading.

All presentations and necessary MATLAB-files have to be provided by the deadline to sa@emt.tugraz.at. This holds for all presentations ("must show" and selected "will show"). Please follow the instructions for the upload, which are provided by an additional sheet on the server.

The presentation slides have to consider all necessary derivations, algorithmic considerations, diagrams, conclusions, etc.. A pure presentation of the final results will lead to reduction of your points. Also take care to the representation of necessary diagrams. Slides can be in German of English. Note, that your slides have to include all graphics. A reference to a generating MATLAB-script is not accepted.

Deadline: 16.1.2020

Example 1: Analysis of Damped Exponentials

In the first example we again consider a damped spring mass system, as it was used in the first homework sheet. Figure 1 depicts the system. While the task of the first experiment was to measure the frequency response $H(f)$ of the system, we now want to estimate the system parameters m, c and d. This can be done by using appropriate experiments and sophisticated signal analysis methods to gain knowledge from the system response.

The damped spring mass system follows the differential equation

$$m\ddot{y}(t) + d\dot{y}(t) + cy(t) = F(t), \tag{1}$$

where y is the response in m, m is the mass in kg, c is the spring constant in N/m, and d is the damping constant in Ns/m. For the excitation of the system a force F in N is

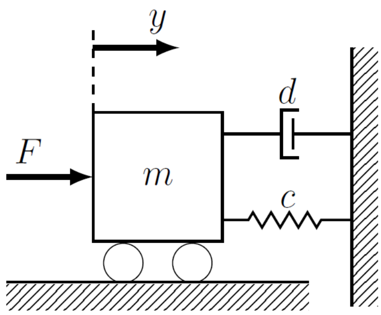

Figure 1: Damped spring mass system.

applied to the mass.

For under-damped systems the homogeneous solution of ODE (1) follows the equation

$$y(t) = Ce^{-\gamma t} \cos(\omega_d t + \varphi), \tag{2}$$

where γ is given by

$$\gamma = \frac{d}{2m}. \tag{3}$$

ω_d is referred to as the damped circular frequency and is given by

$$\omega_d = \sqrt{\omega_0^2 - \gamma^2}, \tag{4}$$

where ω_0 is given by

$$\omega_0 = \sqrt{\frac{c}{m}}. \tag{5}$$

In this example you have to analyze the damped spring mass system, which is provided by the MATLAB function

```
y_out=mass_spring_system(F,Td,id)
```

The function requires the following inputs

- `F`: vector presenting the force excitation in N.
- `Td`: sampling period in s
- `id`: your student ID (integer number). You can find your student ID on the server.

Note, that the system is different from the first homework sheet. The output is already corrupted by noise. The elongation is also limited to $\pm 0.2\,\mathrm{m}$, so be careful about the maximal force you apply to the system.

Please provide answers to the following points:

(A) Consider equation (1). Which parameter can be estimated from a single experiment only, e.g. steady state, harmonic solution, step response, etc.? Provide a mathematically based answer and estimate the parameter by performing the correct experiment.

(B) Use an appropriate excitation signal and determine the homogeneous solution given by equation (2). The parameters C and φ are not important for the following tasks.

From the homogeneous solution we can estimate the parameters ω_d and γ by the following steps:

a) Determine the damped circular frequency ω_d by a DFT analysis from the signal.

b) Design a Hilbert transformer to compute the envelope of the signal.

c) The parameter γ can be found from the slope of the logarithm of the envelope. Select a proper region from the envelope signal and perform a least squares estimation of the slope. You can use the command `polyfit` to perform this least square fit.

(C) Use the relations for γ and ω_d and provide the system parameters m, c, and d.

(D) The given system again offers some nonlinear behavior. To be more specific, the damping constant d depends on the direction of the movement of the mass. For PT2 systems the transfer function can be expressed by

$$G(s) = \frac{Y(s)}{F(s)} = K\frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{6}$$

where ζ is given by

$$\zeta = \frac{d}{2\sqrt{cm}} \tag{7}$$

and K is given by

$$K = \frac{1}{c}. \tag{8}$$

It can be shown, that the peak value of the step response has the value

$$y_{\max} = M_P = K\left(1 + e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}\right) \tag{9}$$

Estimate the parameter ζ from the positive and the negative step response and derive the individual damping constances d_+ and d_-.

Take care about appropriate sampling frequencies to estimate the parameters. You might also find it useful to filter the output signal prior to these steps, to reduce the impact of the measurement noise.

You can find a useful information about the application of the Hilbert transformer in the tutorial bo0437.pdf from Brüel and Kjaer.

Example 2: DFT Analysis

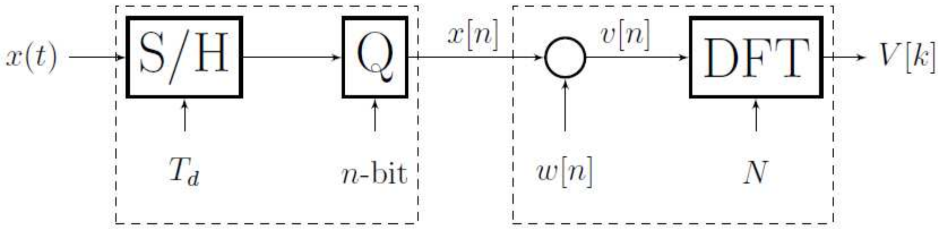

Figure 2: DFT based measurement system with signal acquisition.

In the second example you are dealing with a DFT-based measurement system including the signal acquisition chain provided by the S&H and the DAQ. Figure 2 provides a sketch of the system. You are provided by the MATLAB function

```
xn=osci_input(func_signal,L,fs,id),
```

which provides the signal acquisition system. The function requires the following inputs

- `func_signal`: a MATLAB function, which provides the signal
- `L`: Number of samples
- `fs`: sampling period in Hz
- `id`: your student ID (integer number). You can find your student ID on the server.

See the provided demo file to run the function. Note, that the maximal sampling frequency is limited to 1 MHz.

Please provide answers to the following points:

(A) Determine the input range and the effective number of bits (ENOB) of the signal acquisition system.

(B) Run the signal acquisition function with the strings `sigA1` and `sigA2` for the function argument, e.g. use the line

```
x = osci_input('sigA1',L,fs,id);
```

In this case the signal is provided by personalized signal generators. The signal is an aperiodic signal.

Implement a DFT based analysis function, which provides the Fourier transform. Take care about a correct scaling of the amplitude. Characterize the spectral properties of the signal, e.g. provide the center frequency and the bandwidth. Note: the number of data points is limited in this task.

(C) Run the signal acquisition function with the strings `sigB1` and `sigB2` for the function argument, e.g. use the line

```
x = osci_input('sigB1',L,fs,id);
```

In this case the signal is provided by personalized signal generators. The signal consists of several sinusoidal oscillations.

Analyze the signal and provide its components (frequency and amplitude). Use DFTs of length $N = 2^B$ and find the minimum B, where the analysis provides all frequency components. Provide a discussion about your analysis approach (window, frequency resolution, DFT length, etc.) and result.

(D) Run the signal acquisition function with the strings `sigC1`, `sigC2` `sigC3`, e.g. use the line

```
x = osci_input('sigC1',L,fs,id);
```

The frequency components of the signals are close to the maximal Nyquist frequency ($f_{\mathrm{N,max}} = 500\,\mathrm{kHz}$), so the visualization of the signal lacks in temporal resolution. Provide a proper visualization of the signal by using the DFT based measurement system.

Example 3: PPM Demodulation

Pulse Position Modulation (PPM) is a modulation principle, where the information is coded by means of the temporal position of a pulses within fixed time slots. The principle of PPM was even used by the ancient Greeks some 350 B.C.

Similar to many other modulation principles, one of the key difficulties for the implementation of PPM is the synchronization between the transmitter and the receiver. This problem can be solved by means of differential PPM. Hereby, the information is coded by means of the relative time difference between pulses, rather than by the absolute temporal position. Figure 3 depicts the principle of the modulation scheme.

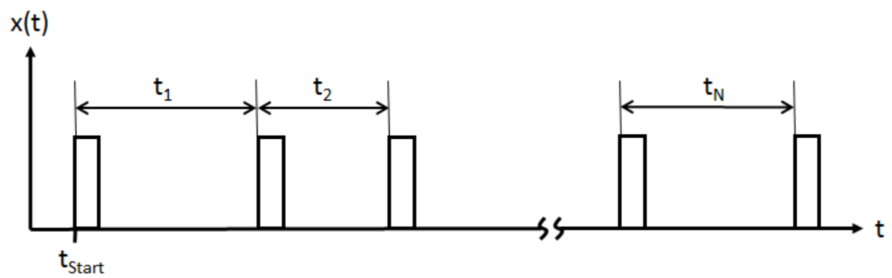

Figure 3: Principle of differential PPM.

In this example you are provided with the MATLAB-Function,

```
[SIG,t] = func_PPMModulator('Message String',f_S,id);
```

which provides a personalize PPM transmitter. The function requires the following inputs

- `Message String`: a string, which will be sent.
- `f_S`: the sampling frequency to measure the output signal of the PPM transmitter. The maximum sampling frequency is limited to 1 MHz.
- `id`: your student ID (integer number). You can find your student ID on the server.

The transmitter features the alphabet

`abcdefghijklmnopqrstuvwxyz 123456790.`

The decoding is done by the function

```
MSGDecode = func_decoder( v_T )
```

where the vector `v_T` holds the time differences (as multiples of the time base). See the MATLAB file `Demo_UE3`, where you can find the implementation of the PPM generator.

Please provide answers to the following points:

(A) Analyze the output of the PPM-modem. What is the RMS value of the noise? Use an appropriate technique and determine the signal pattern to design an optimal detector.

(B) Implement a detector and demonstrate the performance of your detector by means of a plot, where you show the detection of the pattern for the signal in noise case

$$x[n] = s[n] + v[n]. \tag{10}$$

E.g. generate some message string and corrupt the signal with white Gaussian noise, such that the standard deviation the overall noise (noise of the PPM modulator and your added noise) is $\sigma = 0.5$. Then provide a plot, where you show the different signals (signal, signal in noise, signal after filtering).

(C) Measure the jitter of the PPM modem and provide a histogram for the distribution of it. To do so, it is of advantage to implement a state machine, which measures the time difference between detection events. To avoid a multiple detection of the same pulse you have set some hold off time, where you disable the detector after a detection. Below find a possible structure for such a state machine.

```
Define T_holdoff
Init: Flag = 1

Loop
    Get next sample
    If Detection and Flag = 1
        Flag = 0
        Get time from last detection
        Start new time measurement
    End
    If Time > T_holdoff
        Flag = 1
    End
```

(D) Demonstrate the decoding functionality of a message for the signal in noise case, where you corrupt the output signal with white Gaussian noise with $\sigma = 0.5$

(E) Christmas Special: Run the PPM modem with the command

```
[SIG,t] = func_PPMModulator([],f_S,id);
```

In this case the message is a movie quote (personalized). Which character made the quote and what was the movie? Who was the actor playing the character?

Hint: you can use Google or youtube to find the answer, once you have decoded the quote.