Lab 4: Sampling and aliasing

Grading

This Lab consists of two exercises (one divided in three subexercises). Once you have submitted your code in Matlab Grader AND once the deadline has past, your code will be checked for correctness. Note here, that upon submission, your code is already subjected to some basic checks that are aimed to verify whether your code will compile; these basics checks don't say anything about the correctness of your submission. You can visit Matlab Grader again after the deadline (give the servers some time to do all the assessments; this might even take a few days) to see how well you did. In case Matlab Grader indicates you failed an exercise, this does not automatically imply that you failed the entire exercise. Each exercise is subjected to n tests, where the number of tests can vary between exercises. In case Matlab Grader indicates you failed the exercise, this means that not all tests were passed (e.g. in an exercise with 7 tests, you could have passed 6 and Matlab Grader will indicate you failed the exercise). Your grade is calculated based on the number of tests you passed and not on the number of exercises you passed.

1 Introduction

In most situations where signal processing is applied, this is done by computers. The main reason for this is that the development of computer-based signal processing methods is very flexible and relatively cheap. However, computers cannot deal with analog (or continuous-time) signals, so in order to use computers in signal processing the signals first need to be converted from the analog domain to the digital (or discrete-time) domain. For this we can use so-called analog-to-digital (A/D) converters or continuous-to-discrete (C-to-D) time converters. Such a C-to-D converter operates at a certain frequency referred to as the sampling frequency f_s. In other words, the signal is sampled by the C-to-D converter at a rate of f_s, or in intervals of $T_s = 1/f_s$ seconds. After the signals are processed by the computer, e.g. FIR filtering which will be discussed in Chapters 5 and 6 of this course, the processed signals are often converted back to the time-continuous domain by a discrete-to-continuous (D-to-C) time converter. The C-to-D and D-to-C converters do not necessarily have to operate at the same sampling frequency, as you can see in the Figure below and as you will see in Exercise 3.

2 Aliasing

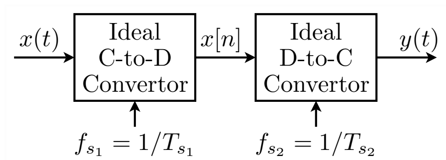

Figure 1: Ideal C/D and D/C conversion

In Figure 1 above, the time-continuous signal at the input is described by $x(t)$. This signal is then converted by the C-to-D converter operating at sampling frequency of f_{s_1} to obtain the time-discrete (or digital) signal samples $x[n]$, which can be processed by digital signal processing methods. Afterwards, the time-discrete $x[n]$ is converted to the time-continuous domain by the D-to-C converter that operates at sampling frequency of f_{s_2} to yield the time-continuous signal $y(t)$.

The topic of this lab is understanding the concepts of sampling and reconstruction and to understand when and how aliasing occurs in this process.

Exercise 1 [7 tests]

Consider the signal $x_1(t) = A_1 \cos(2\pi f_1 t + \phi_1)$, with amplitude $A_1 = 2$, frequency $f_1 = 5[Hz]$, and phase $\phi_1 = \pi/3$. Create 2 plots underneath each other (making use of the subplot function) and plot $x_1(t)$ over two periods in the first plot and use 100 time values per period with 201 time values in total.

Calculate the values of $x_1(t)$ for $t = 0,\ 0.05,\ 0.1,\ 0.15,\ 0.2,\ 0.25,\ 0.3,\ 0.35,\ 0.4$ seconds and store these results in the vector x_{n_1}. In addition, make a time vector t_n that ranges from [0, 0.4] seconds with 9 time values. The vector x_{n1} describes the time-discrete signal $x_1[n]$, sampled at a rate of $1/0.05 = 20$ Hz.

Use the same plot for $x_1(t)$ from the previous exercise and add a "stem" for x_{n1} at the positions specified by t_n. Tip: Use the help function of MATLAB to acquire information about this function. You should use the second variant that gets displayed.

In the second plot, plot the signal $x_2(t) = A_2 \cos(2\pi f_2 t + \phi_2)$ with $A_2 = 2$, $f_2 = 25[Hz]$, and $\phi_2 = \pi/3$. Calculate $x_2(t)$ for 10 periods and use 100 values per period with 1001 time values in total.

Calculate the values of $x_2(t)$ for $t = 0,\ 0.05,\ 0.1,\ 0.15,\ 0.2,\ 0.25,\ 0.3,\ 0.35,\ 0.4$ seconds and store these results in the vector x_{n2}. The vector x_{n2} describes the time-discrete signal $x_2[n]$, also sampled at a rate of 20 Hz.

In the second plot, also add a "stem" for x_{n2} at the positions specified by t_n. Note: it is not necessary to define a new t_n, you must re-use the t_n defined before. Do you notice any difference between the samples of $x_1[n]$ and $x_2[n]$?

IMPORTANT: The order in which the plots should occur, are for each subplot 'signal' followed by 'stems'.

3 Sampling

From the previous exercise you can see that, in case we would sample the sinusoidal signal with frequency $f_1 = 5[Hz]$ at a sampling frequency of $f_s = 20[Hz]$ and do the same for the sinusoidal signal with frequency $f_2 = 25[Hz]$, we have a problem of ambiguity. The samples we made do not only describe the 5 Hz signal, but also the 25 Hz signal. In fact, they describe an infinite number of other periodic signals, of which the one you plotted with a frequency of 5 Hz is the one with the lowest frequency.

Exercise 2

In this exercise you will write a function that can convert any sinusoidal input signal to its output signal. Aliasing is an important aspect of this exercise that you will deal with. To create this function, the exercise is split up in different parts which will progressively contribute to the final fully functioning function. All sub-exercises are individual exercises, so the requirements per sub-exercise are *only* listed in the corresponding sub-exercise.

Consider the situation from Figure 1, where $f_{s_1} = f_{s_2} = f_s = 1/T_1$. The following equations are a generalization of the calculations required for obtaining the output signal. Just in this example the sampling frequencies are equal.

$$\begin{aligned} x(t) &= A_0 \cos(2\pi \cdot f_0 \cdot t + \phi_0) && (1)\\ x[n] &= x(t)|_{t=n \cdot T_1} = A_0 \cos\left(\left(2\pi \cdot \frac{f_0}{f_s}\right) \cdot n + \phi_0\right) \\ &\equiv A_0 \cos(\theta_0 \cdot n + \phi_0) \quad \text{with } \theta_0 = 2\pi \frac{f_0}{f_s} \quad \in (-\pi, \pi] && (2)\\ y(t) &= x[n]|_{n=t/T_1} = A_1 \cos(2\pi \cdot f_1 \cdot t + \phi_1) && (3) \end{aligned}$$

Exercise 2a [4 tests]

Write a MATLAB function that, given the frequency f_0 of the input signal and the C/D sampling frequency f_s, calculates the normalized frequency θ_0 and manipulates it to fit in the given domain of $\theta_0 \in \langle -\pi, \pi]$. The positive normalized frequency, obtained from the positive frequency of the input signal, should be taken as a starting point. The function should return the *manipulated* normalized frequency θ_0. The use of conditional statements (if, else, while) might be useful. Use the template provided in Matlab Grader!

Exercise 2b [4 tests]

Now assume the *fixed* input signal: $x(t) = cos(2\pi \cdot 175 \cdot t + \frac{\pi}{3})$. Alter the function, written in exercise 2a, to have as input variables: C/D <u>and</u> D/C sampling frequency f_{s1} and f_{s2} respectively. Keep in mind that these sampling frequencies are not always equal! The function has to calculate the (positive) frequency of the output signal (f_1 in Hz) and the corresponding phase (ϕ_1) and output them in this order. The function also has to plot (not the stem-plot!) the output signal $y(t)$ with the calculated variables on the time span $[0, 0.1]$ seconds with a total of 1001 time values.

Exercise 2c [7 tests]

Now the previous exercises will be generalized. You will create a function that accepts (the parameters of) a input signal and the (possibly different) sample rates and that plots the output signal. Create a function that accepts the following input variables: A_0, f_0, ϕ_0, f_{s0} and f_{s1} in this order. The function should use these variables of the input signal and the C/D- and D/C-converters to construct the output signal. The output signal should be plotted on a time domain [0, 0,05] seconds with exactly 1001 time values.

Testing

Because of the complexity of this exercise, the testing criteria of exercise 2c will be given for you so you can verify your output by hand.

Test	A_0	f_0	θ_0	f_{s1}	f_{s2}
1	1	100	0	400	400
2	2	100	$\pi/3$	400	400
3	1	200	$\pi/3$	100	100
4	2	200	$\pi/3$	60	60
5	2	175	$\pi/3$	100	100
6	1	100	$\pi/3$	400	500