

Lab 1 – DT signals, sampling, frequency

Entry test example questions

1. $x_a(t) = \cos(2\pi f_a t)$ was sampled with sampling period T_s . Find normalized frequency, normalized angular frequency θ or period of the sampled signal. Plot the (sampled) signal spectrum. (f_a , T_s or f_s given, their proportion rational or irrational...)
2. An analog signal with spectrum extending from $-f_a$ to $+f_a$ has been sampled with a sampling period T_s (or frequency f_s). Plot the spectrum after sampling (different values of f_a w.r.t. f_s)

Matlab notes

For help, use `help <subject>`, note that UPPERCASE is used to mark keywords in help only, not in real usage in Matlab....

An exception is in the scripts developed for this lab - their names ARE uppercase.

For sampling a real, analog signal with an A/D converter use Matlab command: `y=LCPS_getdata(Nsamples_in_block, [Kblocks, [Tsampling, [leave_bias]]])`

Tsampling is in seconds, “[]” denote optional arguments.

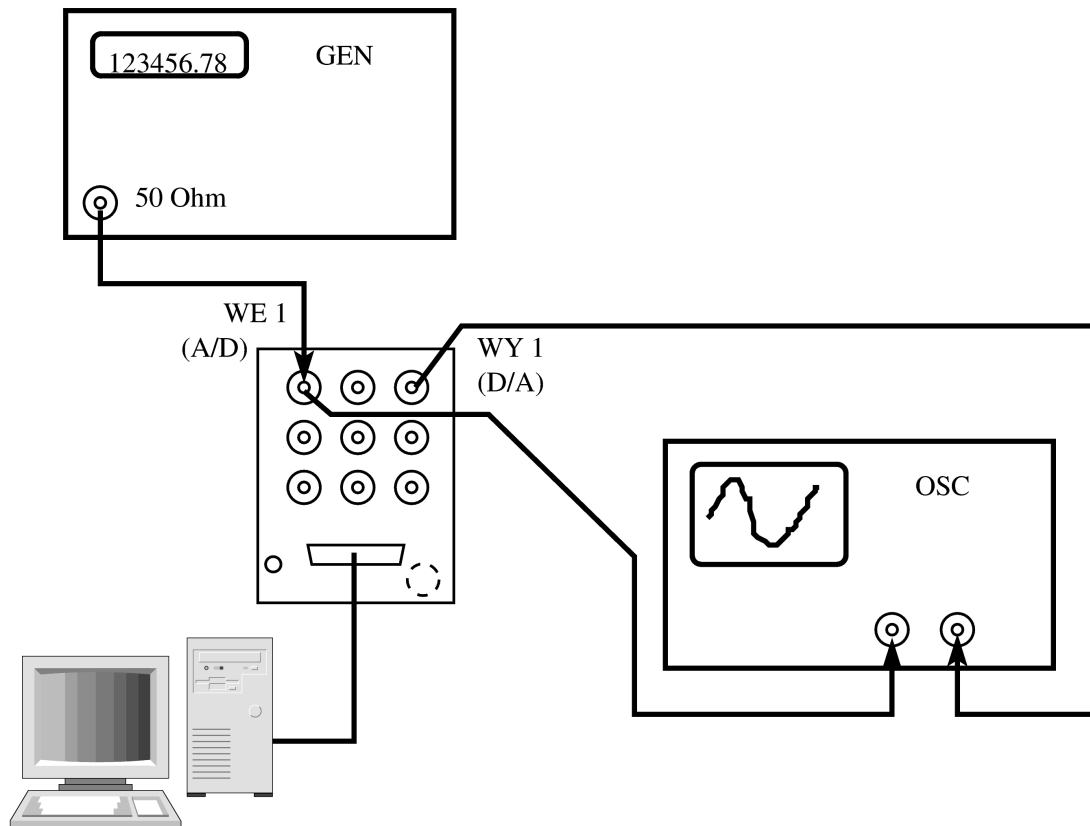
We usually use `Kblocks=1`. Note that actual sampling frequency will be taken from a small predefined set available with the used soundcard – we mostly use 48 kHz. Our card input follows the “professional” standard – you can sample signals within ± 10 Volts.

For plotting DT signals, use markers (`plot(n,x,'o')` or `'-o'`). For their continuous counterparts, use lines.

Exercises

Italics denote optional tasks. Bold suggests what should be in the report

1. NOT using Matlab, **plot** (with a pen or pencil) two periods of sine wave with $f = 4800$ Hz. **Mark** the points where the wave will be sampled at $f_s = 48$ kHz. **Note number of samples per period.**
2. Using Matlab `sin()` function, try to repeat the picture on screen plot. Finally extend the plot to 100 samples length (with the same parameters). **In the report: copy the Matlab commands used to produce the data.**
3. Applying `sign(x+eps)` to your signal `x` obtain a square wave and plot it. (hint: `eps` is added to avoid exact zero in `x` being converted to zero - square wave is either +1 or -1).
4. Connect the equipments as presented on picture 1.



Get similar signals (sine and square wave) from a generator. Set amplitude to about 5 V. **Describe differences between simulated and real-world plots.**

5. Label an x-axis of the plot of real-world sine wave (in Matlab) with time units, then repeat with sample indices (hint: `plot(xvalues, yvalues, 'marker');`). **In the report - copy the Matlab commands used, and copy approx. two periods from the screen plot.**
6. Sample a sinusoidal signal with much bigger amplitude (max possible from generator), and with much smaller amplitude. See the effects of clipping and of noise.
7. Plot a simulated DT sinusoid with normalized frequency $f_n = f/f_s$ equal to 0.1, 0.3, 0.5, 0.9, 1.1, 2.1 ($\theta = 2\pi f_n$, $x(n) = \sin(n\theta)$). Note number of samples in period. For plots @ f_n 0.1 and 1.1 **draw a copy in your report**, and explain **why in some cases different f_n result in identical plots** - try to draw (by pencil) the underlying CT signal. *If you are brave enough, draw the underlying CT signal with Matlab using 9 additional samples between original ones; show it to the teacher and/or copy Matlab expression to the report.*
8. Use the program Spectrum Analyzer located on the desktop. Set parameters:
 - Input - Dev1/ai2, if the cable is connected to AI2;
 - Sampling frequency - 12 kHz;
 - Number of samples - 200;
 - Number of zero - 0;

Calculate the Nyquist Frequency. Set the frequency of signal in the Nyquist range. Slowly increase the frequency of the signal, beyond the Nyquist limit, up to a frequency greater than twice the sampling frequency.

- What happens to the spectrum of the sampled signal, when the signal frequency exceeds the Nyquist frequency?
 - What happens to the spectrum of the sampled signal, when the signal frequency exceeds the sampling frequency?
9. Sample a physical signal: 1.6 kHz sinusoid sampled at $f_s=48$ kHz, 1024 samples; plot it in Matlab using physical time as horizontal axis (create the time axis variable t). Then decimate signal by 16 (i.e. leave every 16th sample) or by 32.
- Hint: use `xdecimated=x(1:16:end)`; `tdecimated=t(1:16:end)`,
 Plot it: `plot(t,x,tdecimated,xdecimated,'-x')`. **Note the resulting sampling frequency and describe undersampling effects.**
10. Keep the plot (or save in some variable the data to replot it again). Repeat the experiment with 0.1 kHz, 3.1 kHz etc. **Compare the results with exercise 9.**
- Hint: to see more plots read help on `figure()` command.
 Hint: to understand results recall the spectrum periodicity.
11. Create a set of test signals (all with length of 32 samples).
- (a) A unit impulse signal $\delta(n)$ by using
- ```
N=32;
n=[-3:(N-4)];
dlt=(n==0);
stem(n,dlt);
```
- (b) A unit step signal  $u(n)$ :
- ```
ustep=(n>=0);
stem(n,ustep);
```
- (c) Shifted unit impulse $\delta(n - K)$; use your table number as K
 You can do it by adding some zeros at the left side of the samples vector and trimming the right side to the previous length (example shown for shift by 4)
- ```
dlt4=[zeros(1,4) dlt]; %add 4 zeros
dlt4=dlt4(1:N); %trim to N samples again
stem(n,dlt,'b');
hold on
stem(n,dlt4,'g');
hold off
```
- (d) A short impulse (e.g with 7 samples length) by subtraction of two unit steps (one shifted in time) or by clever use of `ones()` and `zeros()`.
12. Use the prepared signals ( $\delta(n)$ ,  $u(n)$ ,  $\delta(n - n_0)$ , finite-time impulse) as  $x$  to test a linear system implemented by matlab command `y=filter([1, 2, 1],[1 -.9],x)`; **plot the resulting output signals  $y$** ; try to notice how may the  $y$  depend on  $x$ .