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=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. Typically you can sample signals within ±2.5 Volts.

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

Exercises

Italics denote optional tasks.

1. NOT using Matlab, plot (with a pen or pencil) two periods of 3200 Hz sine wave sampled at 32 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). (Then, show your result to the teacher.) In the report copy the Matlab commands used to produde 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. Use A/D converter to get signals (as in 3 and 1) from a generator. Set amplitude to about 1 V. Compare simulated and real-world plots.

5. Label an x-axis of above plot (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 (few volts), 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 the 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. Sample a 1.1 kHz sinusoid @ 32 kHz, then decimate it by 16 (i.e. leave every 16th sample) or by 32. Note the resulting sampling frequency and note undersampling effects.

   Hint: use `xdecimated=x(1:16:length(x));`

9. Keep the plot (or save in some variable the data to replot it again). Repeat the experiment with 0.1 kHz, 2.1 kHz etc. Compare the results with exerise 8.

   Hint: read help on `figure()` command.

10. Create a $\delta(n)$ unit impulse signal by using

```
N=32;
n=[-3:(N-4)];
dlt=(n==0);
stem(n,dlt);
```

    Then make a unit step signal from shifted $\delta(n)$'s

$$u(n) = \sum_{m=-\infty}^{n} \delta(n-m)$$

    Hint: use `cumsum()`, it is easier and faster than `for` loop.

11. Shift your $\delta(n)$ in time 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];
dlt4=dlt4(1:N);
stem(n,dlt,'b');
hold on
stem(n,dlt4,'g');
hold off
```

12. Create a short impulse (e.g with 7 samples length) by subtraction of two unit steps (one shifted in time).

13. Use the above 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`.

File: nwlab1 LaTeXed on October 14, 2011