

# EE153 Homework 1

## Chapter 2

Prof. Alyson Fletcher

Complete all problems. The final two problems use MATLAB. When submitting the MATLAB component, provide the MATLAB script, all plots and answers to any questions in those parts. One simple way to create the submission is to place all the code in a single MATLAB script file and use MATLAB's publish capability:

[http://www.mathworks.com/help/matlab/matlab\\_prog/publishing-matlab-code.html](http://www.mathworks.com/help/matlab/matlab_prog/publishing-matlab-code.html)

When you publish your script, MATLAB will run the script and create an html file with all the MATLAB code, comments in the code, figures and output. To make it readable, separate each answer with the double comment (`%%`) Once you have the html file you can print it or convert it to pdf to submit.

1. For each of the following systems, determine whether the systems is: (1) time-invariant, (2) causal, (3) linear and (4) stable. EXPLAIN your answers to receive any credit. In (b),  $u[n]$  is the discrete-time unit step function.
  - (a)  $T(x[n]) = x[-n]$
  - (b)  $T(x[n]) = u[n + 1] + 3x[n]$
  - (c)  $T(x[n]) = \sum_{k=n-n_0}^{n+n_0} x[k]$  for some  $n_0 > 0$ .
2. Find the impulse response,  $h[n]$ , for the following linear time-invariant (LTI) systems. (Hint: For an LTI system,  $h[n] = T(\delta[n])$ .)
  - (a)  $T(x[n]) = \sum_{k=-\infty}^n x[k]$
  - (b)  $T(x[n]) = x[n + 1] - x[n]$
  - (c)  $T(x[n])$  has a step response,  $s[n] = T(u[n]) = 3^n u[n]$ . Hint: Use  $\delta[n] = u[n] - u[n - 1]$ .
3. Are the systems in Problem 2 (i) stable and (ii) causal? EXPLAIN.
4. Are the following signals periodic? For each signal that is period, find the period.
  - (a)  $\cos((8/3)\pi n)$
  - (b)  $\sin(2\pi n) + \cos((2\pi n))$
  - (c)  $\sin(\pi n/3)/(\pi n)$  for  $n \neq 0$ ; 1 for  $n = 0$
  - (d)  $e^{3n}$

5. Convolution: Find the convolution  $y[n] = x[n] * h[n]$
- $x[n] = 3nu[n]$ ,  $h[n] = u[n]$ .
  - $x[n]$  and  $h[n]$  are shown in Fig. 1.
  - $x[n] = (1/4)^n u[n]$ ,  $h[n] = u[n]$ .

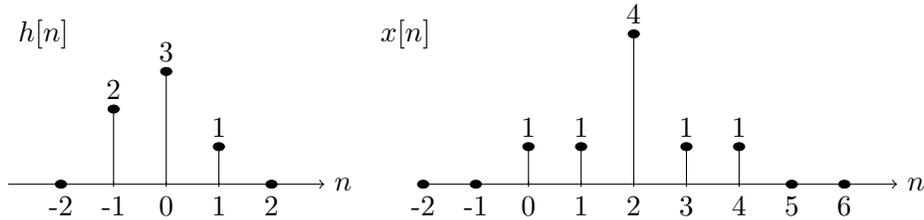


Figure 1:  $h[n]$  and  $x[n]$  for Problem 5(b).

6. *Representing and visualizing signals:*

- In MATLAB, discrete-time signals must be represented as vectors with finite length. Create two MATLAB vectors with components:

$$\begin{aligned}
 t[n] &= (n-1)\Delta, \\
 u[n] &= \sin(2\pi t[n]/T) + 0.5 \cos(6\pi t[n]/T), \quad n = 1, \dots, N,
 \end{aligned}$$

where  $\Delta = 0.01$ ,  $T = 1$  and  $N = 300$ . The vectors represent the sampling times and samples of the signals at those times. To be compatible with the remaining parts, make sure the vectors are **column vectors** – you may need to use the transpose operator in MATLAB (`'`). Note that MATLAB indexes vectors from 1 to  $N$ , not 0 to  $N - 1$ , like many languages such as C/C++ and python.

It is important that you create the vectors without using any for loops. MATLAB is significantly more efficient when performing the commands on vectors as opposed to individual components. To read more, see:

[http://www.mathworks.com/help/matlab/matlab\\_prog/vectorization.html](http://www.mathworks.com/help/matlab/matlab_prog/vectorization.html).

Try to avoid for loops as much as possible in MATLAB.

- MATLAB has a rich set of tools for plotting and visualizing signals that are very easy to use. To plot the signal you created in the last part, use the commands:

```

% Create a line plot
plot(t,u,'-', 'LineWidth',2);
grid on;
set(gca, 'FontSize',16);
xlabel('Time t');
ylabel('u(t)');

```

In addition to the plot command, I have included commands for making the lines thicker (so they print well), adding grid lines and labeling the plot. It is important you learn to make a good plot, since a good plot is the best way to convey data.

Run this code and include the plot in your submission. If you are using MATLAB's publish capability, then the plot will automatically be included in the report. However, if you want a separate image, MATLAB has an excellent `print` command that lets you create a variety of different image types from the plot. Here you could use:

```
print -dpng sinusoids;
```

which creates a PNG file `sinusoids.png`. You can also create JPEG and other formats.

- (c) For discrete-time signals, an alternate plot is a stem plot, which has the advantage of showing exactly where the samples are located.

```
% Create a stem plot of the first 100 samples
I = (1:100);
stem(t(I),u(I));
grid on;
set(gca,'FontSize',16);
xlabel('Time t');
ylabel('u(t)');
```

Again, run the code and include your output in the submission.

### 7. Simulating a simple system

- (a) Now suppose that the signal  $u[n]$  from the previous problem is the input to a system that generates the output

$$y[n] = \tanh(u[n]).$$

Write a single MATLAB line of code that generates the vector of samples of  $y[n]$ . The `tanh(.)` function is often a model of “saturation” since it clips signals to  $\pm 1$ .

- (b) To plot the input and output, we can use MATLAB to show  $u[n]$  and  $y[n]$  on the same graph with code like:

```
% Plot the input and output vs. time
plot(t, [u y], '-','Linewidth', 2);
grid on;
set(gca,'FontSize', 16);
legend('Input', 'Output');
```

Note the use of the `legend` command, which lets you create a legend on the graph. Run this code and include the plot in your submission.

- (c) Finally, you may wish to create a plot of the output vs. the input. For this you can use the scatter plot, which places a dot at each point  $(u[n], y[n])$ :

```
% Plot the output vs. input scatter plot
plot(u, y, '.', 'Linewidth', 2);
grid on;
set(gca,'FontSize', 16);
xlabel('Input u[n]');
ylabel('Output y[n]');
```