Planning Doc

In your report, clearly describe for each of these 3 systems:

i) your methodology for testing each of the properties listed above, and

ii) your reasoning for why you think that system has that particular property, e.g., what signal(s) you put into the system to determine whether it is causal or non-causal, and how you can tell that from the output.

For input signals, you can create your own set of signals, as described above for the unit impulse for example. It is recommended that you test out the system properties with a few different signals, but in your report, you can include just those input signals cases that are sufficient to support your decisions on each of the system properties.

Sep 27th Meeting Minute

Linearity: Daniel

Causality & Memory: Kevin

Time Invariance: David

Oct 2nd Meeting Minute

Meeting on Tuesday 8pm

Finish the bonus 2 parts

Draft 1 for report done

Oct 4th Meeting Minute

Finish the remaining of the report (individual parts)

Finish the code for bonus 2

**Stuff daniel suspects:**

**System0:**

Y[n] = 0.25\*x[n] + 0.5\*x[n-1] + 0.25\*x[n-2]

Linear

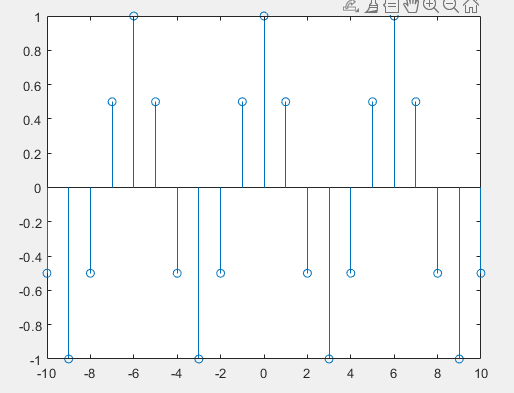
Time invariant

Causal

With memory

**System1:**

Y[n]: g[n] multiplied by x[n], g[n] = {1,0.5,-0.5,-1,-0.5,0.5} repeating for n = {0,1,2,3,4,5}



Output for input of all ones between -10 and 10 (g[n] between -10 and 10)

Linear

Time varying

Causal

Memoryless

**System2:**

Y[n] = x[n] if x[n] >= 0, y[n] = -x[n] - 2 if n < 0

Non-linear

Time invariant

Casual

Memoryless

**System3:**

Y[n] = (x[n-1] + x[n] + x[n+1]) / 3

Linear

Time invariant

Non-causal

With memory

**Test Functions to test linearity with:**

Impulse (scaled with pos/neg numbers)

Unit step (scaled with pos/neg numbers)

Unit ramp (scaled with pos/neg numbers)

Sinusoidal (scaled with pos/neg numbers)

Random numbers (scaled with pos/neg numbers)

Check additivity by mixing these inputs

Time varying

x(n) → y(x(n))

x(n+a) → y(?)

Shift y(x(n)) *a* steps → y(x(n+a))

See if y(?) = y(x(n+a))

Memory

x1(n) -> y(x1(n))

x1’(n) -> y(x1’(n))

(change one number in the previous input) and check number of differences

Causality

x1(n) -> y(x1(n))

Compare value of x1 and y when x1 is equal to 0, if non zero, it’s non causal

