MATLAB CHEATSHEET

A basic compilation of functions, helpful and repetative actions, and more!

Random notes and functions

```
qfunc %this will return the value of k i.e. Q(k) which is used for gaussian
      probabilities and more probability items.

bar () %this is a way to display probablility graphs! just like plot, it will
      display blocks on a plot.
```

example code snippets

Filename: Lecture10matlab(1).mlx

```
%this code displays a for/if loop to impliment a rectangular pulse
    fs = 100; ts = 1/fs; % fs = sampling frequency
    t1 = 0; t2 = 5; tau = 1 %start and end time, as well as transition of
amplitude time (tau)
    Av = 1; %amplitude
    time = t1:ts:t2; %time duration which will be plotted
    l = length(time);
    % Generate v(n)
    for n = 1:l %this is one : lowercase L which is the length of time
        if time(n) <= tau %tau is when the amplitude goes from Av to zero again
            v(n) = Av;
        else v(n) = 0;
        end
        end
        plot(time,v)
         axis([0 5 0 1.5])
         xlabel('Time (s)')
         ylabel('v(t)')
%next in this document is the Fourier transform
    % Find the Fourier Transform.
    M = 2^10; %this is the sampling value in frequency domain
    %note that vn is the time domain while Vf is the frequency domain
    Vf = ts*fftshift(fft(v,M)); %note this formatting, this goes for ALL
      TRANSFORMS

    % Generate frequency axis
    df = fs/M;
    freq = -(fs/2 - df):df:fs/2;

    plot(freq,abs(Vf)); %then, note to get the amplitude graph of Vf in frequency
      domain, the abs of Vf must be taken.


    plot (freq, angle(Vf)) %just like abs, angle gives the angle of Vf in the
      frequency domain.
```

Filename: ConvolutionExampleF22.mlx

```
    %stem works just like plot! every time a value is calculated, a line is
placed      to display a sample.
    %rather than doing what plot does which is simply placing the line rather
    %than sample lines.
    stem(time,v)

     % Calculate the convolution
     %note that w and v are two different functions/plots and ts is the
     %sampling time.
     z = ts*conv(v,w); %conv() is the convolution of the two functions.
     stem(time,z(1:l)) %since both v and w in this file are rectangular pulses,
      the ratio from the
     % for/if loop is used to graph z or the convolution equation correctly.
```

Filename: Lecture12matlab.mlx

```
    % Let x (t) = cos (2*pi*100*time) + cos (2*pi*400*time). Find x (n) by
      sampling x(t) with
    % fs = 1000 Hz to generate N = 1000 samples.

    % (a) Plot x(t).

    fs = 1000; ts = 1/fs; N = 1000;
    time = 0: ts : (N-1)*ts; % Generate time
    %time here could be important! this is (samples-1) times the sample
    %time given by the frequency fs. Note that the separation between time
    %markers is also ts or the sampling time.


    f1 = 100.0; f2 = 400.0;
    x100Hz = cos(2*pi*f1*time); %first chunk of x(t)
    x400Hz =  cos(2*pi*f2*time); %second chunk of x(t)
    x = x100Hz + x400Hz; %finally adding the two of them together to graph!
    plot (time, x100Hz);


    plot (time(1:N1), x100Hz(1:N1)) %another way to plot over a specific sample
      size, N1;
    %since time and x100Hz are both dependent on the time interval, both
    %need this reference to work correctly


    wn = 0.4; %cuttoff value that is given in the problem
    [b, a] = butter(4,wn,'low') %this takes the values and low for LPF, and turns
      it into a butterworth graph
    %note that 4 is the order, wn is the cuttoff and low is the type of
    %butterworth
    y1 = filter(b,a,x); %this puts the butter worth values through a filter!
```

Filename:Lecture14Matlab(1).mlx

```
a=[1 1]; b = [1]; %input matrix values
t = 0:0.01:10; % time ratio
y = impulse(b,a,t); % impulse takes matrix and time ratios to graph an
  impulse function
% Calculate frequency response
f=0:0.01:1; %frequency ratio
w=2*pi*f; %converting to radians
H=freqs(b,a,w); % and how creating a frequency graph for the impulse function
  using the same a and b matrix as well as the new radian value.
plot(f,abs(H));
```

Quiz code snippets

Quiz 1 snippet

```
%period = 0.02s
% use fs = 1e4 (10,000 Hz)
fs = 1e4; ts = 1/fs;
time = 0:ts: 0.04;
v = -4 + 8*cos(200*pi*time + pi/6) - 4*sin(500*pi*time + pi/3); %note phasors
  are in pi factors
```

Quiz 2 snippets

```
%okay, so the j in V represents the complex portion of the function.
%note that the function is ALREADY IN THE FUNCTION DOMAIN
f = -4:0.01:4;
V = j*2*pi*f.*sinc(f).*sinc(f); %sinc^2 function works better setting two of
  them multiplied together
```

Quiz 3 snippets

```
% (2f) plot n(x), where n(x) is the number
% of students who scored x points
mx=60; sigmax=15;N=500; %distributions (sigmax), median, and students total
x=0:1:100;%score distribution, zero to 100 points
y=pdf('norm',x,mx,sigmax); %pdf is a pission distribution that takes the
  range of values, median, and distribution
% plot(x,y)
nx=N*y; %displays the distribution times the amount of students
plot(x,nx) %which can then be plotted here


n91_100 = (sum(nx(92:101))); %finds the value of scores between 91 and 101 in
  this cause (101 isnt counted, just includes 100 as a value to count)
n91_100_round=round(n91_100); %rounds value to next integer, no half people
  here
%these two lines allow values to be printed and displayed...could also
%user disp(); to print rather than formatSpec...
formatSpec = '(2g): Number of students scoring 91 to 100 = %d\n';
fprintf(formatSpec,n91_100_round)
```

midterm code snippets

```
%problem 3:
disp('Problem 3')
%note that all values here are from the H(f) function for the RC circuit
B = 1000; C = 1e-6;
L = 1/(4*pi^2*B^2*C);
%B1 = (1/(2*pi))*1*sqrt(L*C) %check on the value of L
R = sqrt(L/(2*C));
c1 = (2*pi*L)/R; c2 - (2*pi)^2*2*L*C;
fprintf('B(Hz) = %f\n', B) %there are ones for C L and R as well


%this format will make it easier to take individual values for frequency!
for k = 1:5001 % df = 1Hz, this is the range of frequency that we will graph
    f(k) = k-1;
    H(k) = 1/(1+j*c1*(k-1)-c2*(k-1)^2); %this is the H(f) equation
end
plot (f, abs(H)); %this is now plotting the graph in the for loop.


%view the format of the for loop which allows this to be possible!
fprintf('|H(0 Hz)| = %f\n', abs(H(1))) %note that the 1 can be any value within
      the range of k in the for loop
%this is how the value of H(f) at a specific f value can be determined.
```