MAE 3724, Systems Analysis and Introduction to Controls

Useful MATLAB Notes

Save script as:

AssignmentTitle_LastName_FirstInitial.m
(copy and paste code to Word document,
then save and submit code as PDF with same name)

Comments at beginning of new script:

```
% Assignment Title
% Systems Analysis
% MAE 3723
% Last Name, First Name
% Date (xx/xx/xxxx)
```

Making a time vector:

```
t = initialTime : stepSize : finalTime ;
t = linspace(intialTime, finalTime, num_of_points);
```

Making a vector of function values - examples (suitable for plotting):

```
t = linspace(0,5,100)  %create a time vector
y1 = 1.2*sin(3*t)   %and a y1 vector (for each value in t)
y2 = 3.7*cos(3*t+2)  %and a y2 vector
```

Plotting:

(1-line, 1-graph) `plot(t1,Y1,'color')`

(2-lines, 1-graph) `plot(t1,Y1,'color1', t2,Y2,'color2',...)`

- or -

```
plot(t1,Y1,'color1')
hold on
plot(t2,Y2,'color2')
```

Line color and style options:

`b`	blue,	`r`	red,	`g`	green,
`b-`	(blue) solid line,	`b--`	(blue) dashed line,	`b:`	(blue) dotted line

Graph Title	`title('Meaningful Title Goes Here')`
X-Axis Label	`xlabel('variable (units)')`
Y-Axis Label	`ylabel('variable (units)')`
Legend	`legend('variable1name','variable2name',...)`

Multiplying two arrays:

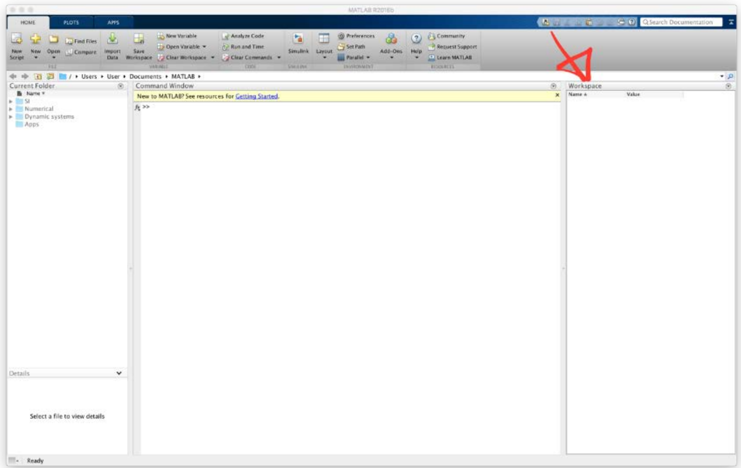
add " `.` " in front of any math operator between two arrays to evaluate element-by-element

(`.*` , `./` , `.^` , etc)

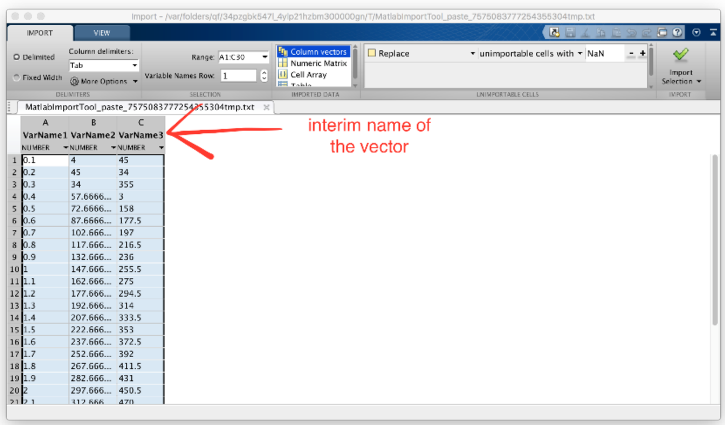
Importing data from Excel:

1) Select the data you want to import from Excel by highlighting it.
   Then right click on the highlighted data and select ‘Copy.’

2) Open MATLAB. In the ‘Workspace’ section, right click and select ‘Paste.’

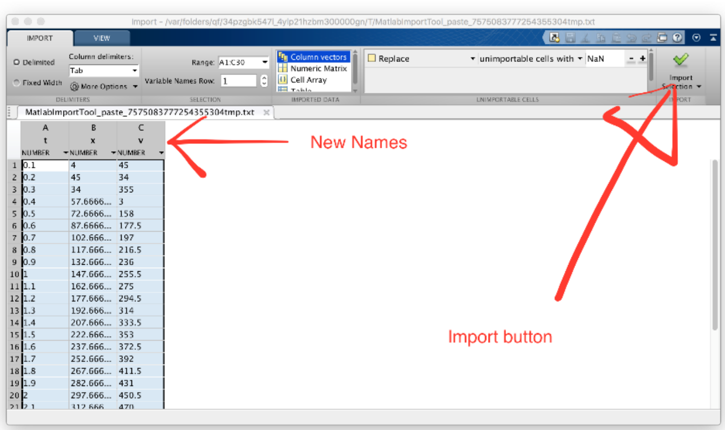


   An import window should pop up with all of the data columns that you copied.
   On the very top of each column, the interim name of the vector exists.



3) Change the names of the vectors, by double clicking on the name. Change the vector name to something that is relevant to your process (i.e x for position, v for velocity, or t for time).

4) Then, click the ‘Import Selection’ button (green check mark in the top right hand corner) to import the data to MATLAB. Now, the vectors should be available in the workspace.



Creating transfer functions and their responses:

`sysName = tf( [num1,num2,...],[denom1,denom2,...] )`

where `sysName` is the variable name assigned to the transfer function
where `[num1,num2,...]` is a list of the numerator's coefficients
where `[denom1,denom2,...]` is a list of the denominator's coefficients

`step(a*sysName, t)`

where `a` is the magnitude (size) of the step input
where `sysName` is the transfer function obtained using the `tf()` command
where `t` is the time vector the response is simulated over

`var1 = lsim(sysName, u, t)`

where `var1` is the variable name assigned to the vector of output values
where `sysName` is the transfer function obtained using the `tf()` command
where `u` is a vector of values developed from an input or forcing function
where `t` is the time vector the response is simulated over

Examples:

Transfer function: $\frac{X(s)}{U(s)} = \frac{8}{s^2+10s+30}$

Forcing function: $f = 10 \cdot \mathrm{u_s}(t)$

```
X = tf( [8] , [1,10,30] )
t = 0 : 0.01 : 2 ;
step(10*X,t)
.
.
.
```

Transfer function: $\frac{X(s)}{U(s)} = \frac{8}{s^2+10s+30}$

Forcing function: $f = \sin(2\pi t)$

```
X = tf( [8] , [1,10,30] )
t = 0 : 0.01 : 2 ;
u = sin(2*pi*t);
x = lsim(X,u,t)
plot(t,x)
.
.
.
```

<u>Using ode45 to obtain system responses</u>

State Variables

1. Make a list of all the time-dependent output variables.
2. List the highest order derivative present for each of the output variables.
3. List all of the lower order derivatives below the highest found in Step 2.
   (a.k.a. "List your *state variables*")
4. Determine each state variable's initial condition.

State Derivative Equations

1. List the derivatives of each of the state variables.
2. Write equations for each of the derivatives, using only state variables and constants.

Forcing Functions and Parameters

1. Determine the model parameters.
2. Write the forcing function equation.

   a. Free response: $f = 0$
   b. Step input: $f = (magnitude\ of\ step)$
   c. Sinusoid: $f = A \cdot \sin(\omega t + \phi)$

Download prepared code from D2L and adjust code

1. Adjust the `deriv` function at the bottom.

   a. Adjust the parameters
   b. Adjust the forcing function
   c. Adjust the state variables
   d. Adjust the derivative equations

2. Adjust the `ode45` command at the top.

   a. Time range
   b. Initial conditions (make sure to match the order assigned in the `deriv` function)

3. Adjust any variables to proper units (i.e. radians to degrees).

4. Adjust the plotting code to obtain graphs of desired data.

```
st_var1 = y(:,1)
```

"all rows in array, column 1 of array"

Global Variables

1. Declare whichever variable is changing as `global` in the:

    - Initial code (up top)
    - and `deriv` function (below)

2. Delete assignment of value to `global` variable in `deriv` function.

3. Write command that assigns value to `global` variable just before `ode45` command.

4. Add additional lines to calculate response at each value of the `global` variable.

Read-in Excel Data

1. Double check that excel file is in same folder as your MATLAB script.

2. Assign data in spreadsheet to a variable using the `xlsread()` function.

```
xlData = xlsread(‘FileName.xlsx’);
```

3. Assign respective portions of that data to specific variable’s vectors.

```
time = xlData(:,1);
x1 = xlData(:,2);
x2 = xlData(:,3);
```

4. Adjust any variables to proper units (i.e. radians to degrees).

5. Plot the excel data and include legend.