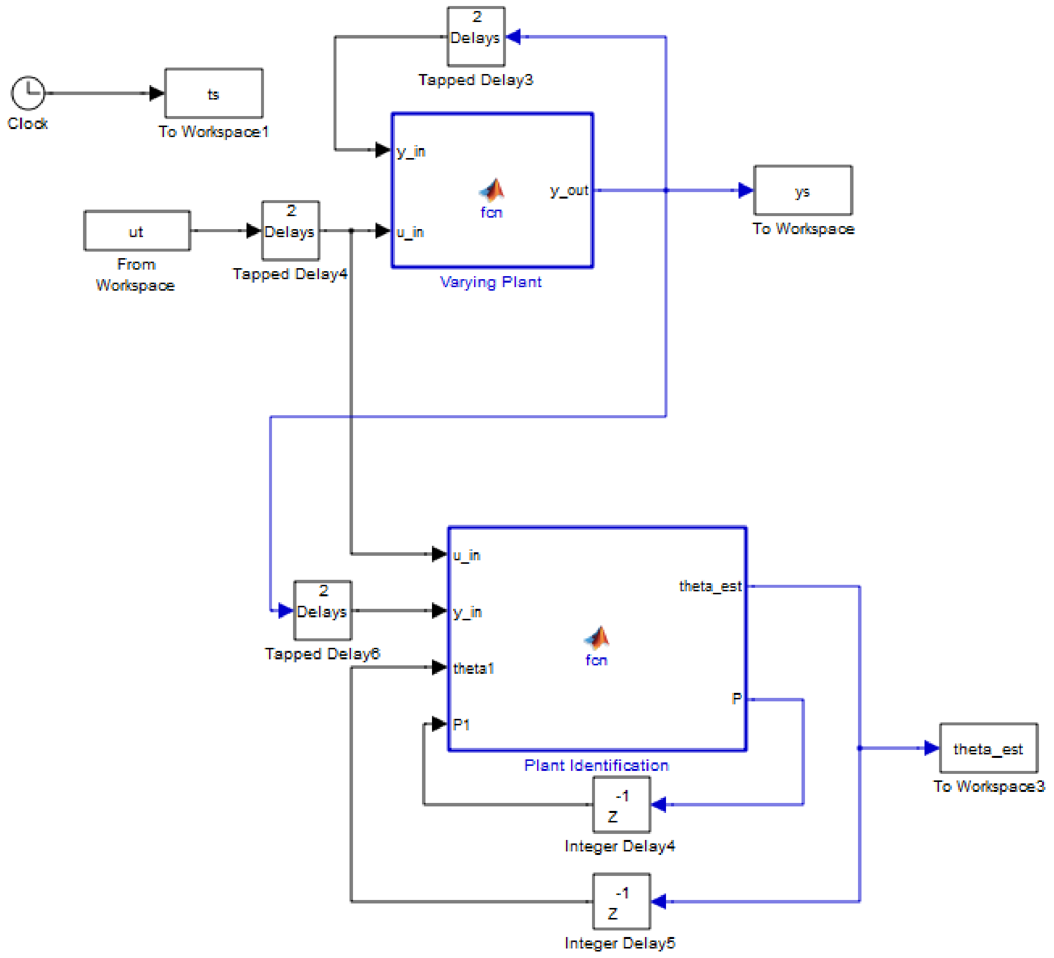
ECE-420: Discrete-Time Control Systems
Project Part C

Due: Friday October 17 at the beginning of class (send me an e-mail with your memo and your code)

In this part of the project you will implement a recursive least-squares algorithm with a forgetting factor in Simulink using the Matlab embedded system toolbox. You should start with the Simulink file **plant_identification.slx** (save the file as **plant_identification2.slx** before you edit it!). You need to implement a recursive least squares algorithm for plant identification. Your final system should look something like that shown below.



You need to run this Simulink model from the Matlab driver file **plant_identification_driver.m** you used in your homework, and show that your results are the same (identical for Matlab and Simulink) for at least three different forgetting factors and the two plants from homework 5 (problem 1 and problem 3).

Just starting off, you should be able to run **plant_identification_driver.m** and show that the Matlab and Simulink results for the plant are the same.
Some things to note:

- all of the variable dimensions in the plant identification routine should be a function of **Nbp**
- you will want the current value of *y* passes through to the identification block
- the code you need inside the plant identification block is very short. Remember that it is updated every sample period, so you are not going to be looping over anything.
- most of the code you need is actually in the Matlab driver file (but you may need to edit it)
- you will need some code like the following in your plant identification block to initialize the algorithms

```
%
%  persistent variables remain in memory
%
persistent initial_flag;% is this the first time through the
algorithm
persistent tol;         % tolerance for resetting P

% initialize the covariance (P) matrix

  if isempty(initial_flag)
    initial_flag = 1;
    tol = 1e-3;
    P1 = eye(5,5);
  end;
```

Mathematical Background: Consider a simple discrete-time transfer function with input $R(z)$ and output $Y(z)$,

$$G_p(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Cross multiplying we get

$$Y(z) + a_1 z^{-1} Y(z) + a_2 z^{-2} Y(z) = b_0 U(z) + b_1 z^{-1} U(z) + b_2 z^{-2} U(z)$$

In the time-domain this becomes

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 u(n) + b_1 u(n-1) + b_2 u(n-2)$$

Defining

$$\phi^T(n) = \begin{bmatrix} y(n-1) & y(n-2) & u(n) & u(n-1) & u(n-2) \end{bmatrix}$$
$$\theta^T = \begin{bmatrix} -a_1 & -a_2 & b_0 & b_1 & b_2 \end{bmatrix}$$

we can write

$$y(n) = \phi^T(n)\,\theta$$

The output of your system identification should be called **theta_est**, to match the Matlab plotting program.

Unfortunately, if you make a mistake in one of the functions, you will get a pretty cryptic message in the Matlab command window. To get more details, click on the block you are having troubles with, click on **Simulink**, and then on **View Report**. At the bottom of this screen there are often helpful hints about what went wrong.

To turn in: write a short memo including your graphs (with captions and figure numbers), and any suggestions you may have for improving this part of the project. e-mail me your memo and both your Matlab and Simulink files.