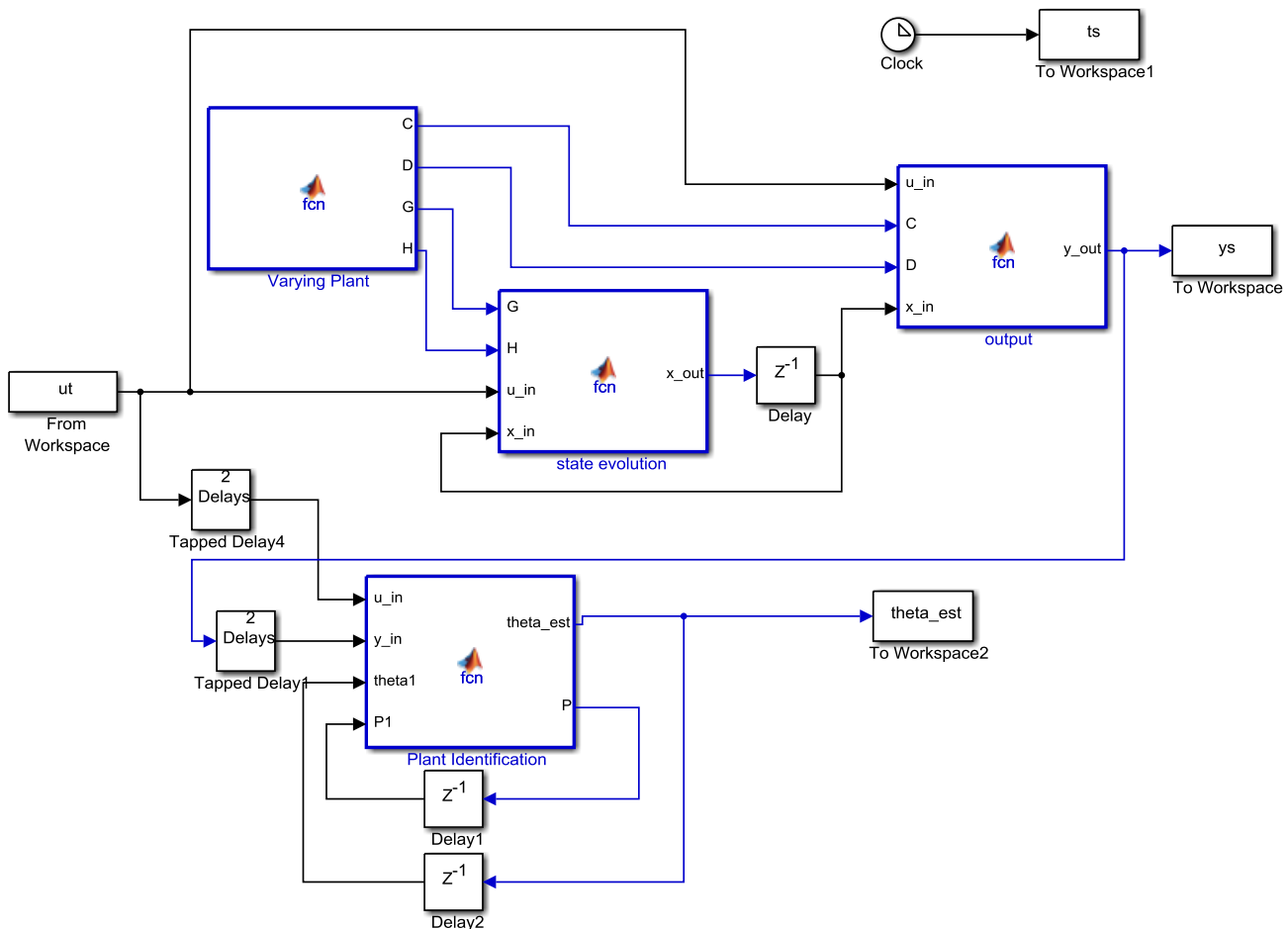


ECE-420: Discrete-Time Control Systems

Project Part C State: System Identification

Due: By 5 PM on Monday November 9 (e-mail me a memo and your modified Simulink model)

In this part of the project you will simulate an open loop plant written in state variable form, and utilize a recursive least squares algorithm to identify the plant, just as you did with a plant written with transfer functions. Note that for simplicity we are assuming the system is written in controllable canonical form. You will need to download the project files from the class website for this. This project was written for Matlab R2012b with a 64 bit operating system. You should start with the Simulink file **openloop_state_model_varying_plant_system_identification.slx**. In this project you need to implement a recursive least squares algorithm for the plant identification. Your final system should look something like the following:



You need to run **plant_identification_driver.m** and show that the Matlab and Simulink results for the plant are the same. You will need some code like the following in your plant identification block to initialize the algorithms

```

persistent initial_flag; % is this the first time through the
algorithm
persistent tol; % number of times this routine has been
calledView Report
%
if isempty(initial_flag)
    initial_flag = 1;
    tol = 1e-3;
    P1 = eye(5,5);
    phi = [0; 0; u_in(1); 0; 0];
    K=P1*phi/(lambda+phi'*P1*phi);
    P=(P1-K*phi')*P1/lambda;
    theta_est = K*y_in(1);
end;

```

Mathematical Background: Consider a simple discrete-time state variable model

$$\begin{aligned}
 x(n+1) &= Gx(n) + Hu(n) \\
 y(n) &= Cx(n) + Du(n)
 \end{aligned}$$

Let's assume we have a second order system and we write it in controllable canonical form. Then it will look like

$$\begin{aligned}
 x(n+1) &= \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} x(n) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(n) \\
 y(n) &= [b_2 - a_2 b_0 \quad b_1 - a_1 b_0] x(n) + b_0 u(n)
 \end{aligned}$$

We can write this in transfer function form as

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

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

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

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. You will need to uncomment code within the Matlab program **plant_identification_driver.m** to run your Simulink model and plot both the Simulink and matlab simulations on the same graph.

Final Note: This is an awful lot like Project C you did before!