Real-Time Windows Target, Simulink, and the ECP Model 210

The Short Version

Step 1. First of all, *you must be using Matlab 6.5.1 to use the real-time windows target with the ECP systems.* Once you have started the correct version of Matlab, we need to get some communications started.

Step 2. To install Matlab's Real-Time Windows Target, type

rtwintgt –setup

Step 3. The real-time windows target utilizes Microsoft's Visual C++ compiler. We first need to tell Matlab to use this compiler. To do this, type

mex -setup

You need to select Microsoft's Visual C++ Compiler.

Step 4. Now we need to inform the ECP system that we will be using Simulink and the real-time windows target. To do this, click **Start -> Programs -> ECP**

The ECP window should pop up, something like the window shown below:

First select on **Utility-> Download Controller Personality File.**

Then select **C: -> Program Files -> ECP Systems -> cn**

Finally select **m210_rtwt_3.pmc** and click on **open**. Wait for the ECP system to load the personality file.

Step 5. Close the ECP Executive **(click on the X*)*** ***Do not just minimize it!!!!***

Step 6. Set the Matlab **current directory** to the folder where all of your files are located.

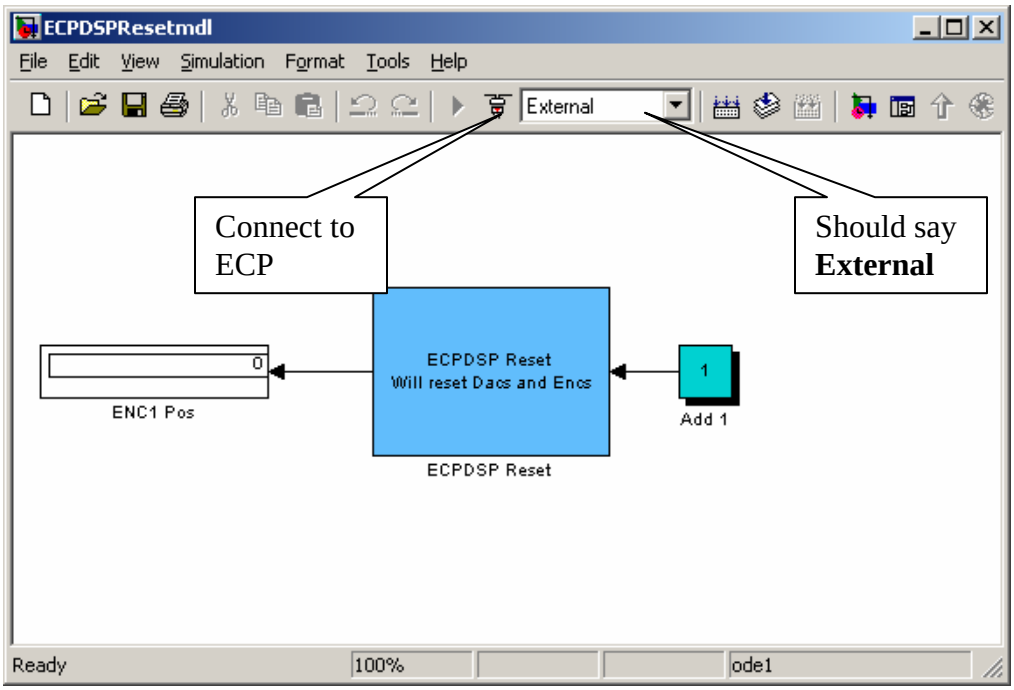
Step 7. Start Simulink by typing **Simulink** in the Matlab window.

Resetting the System Using Simulink

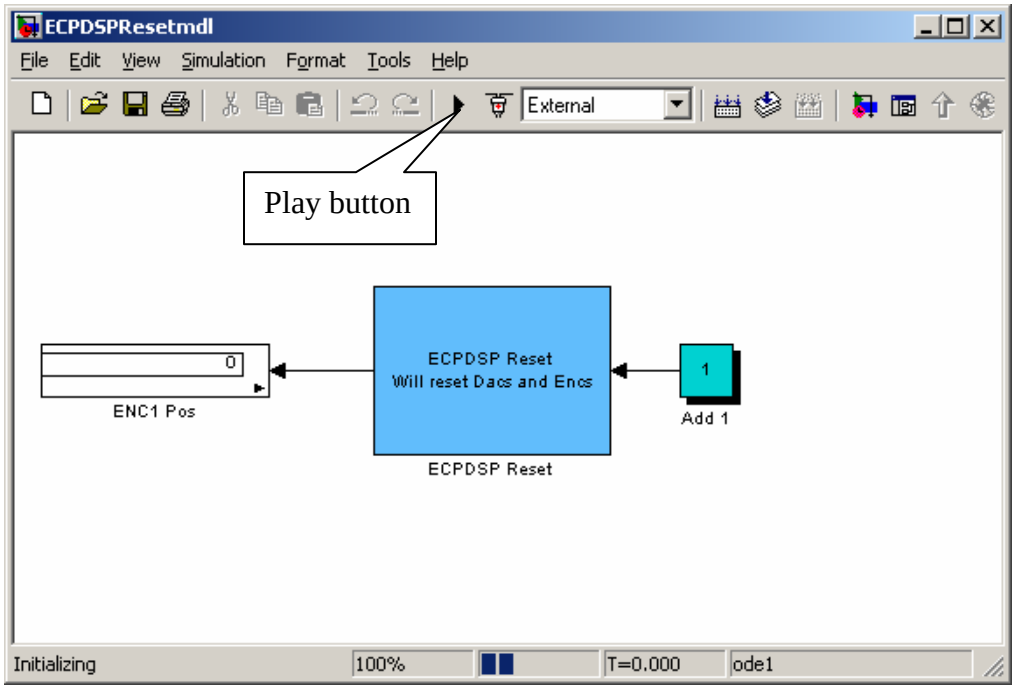
We often have to reset/reinitialize the system. In order to do this from Simulink we use the **ECPDSPResetmdl.mdl** Simulink model file. This model is our friend, you will probably use it a lot.

Step 1. From Simulink select **File->Open ->ECPDSPResetmdl.mdl**

Step 2. Connect to the ECP system



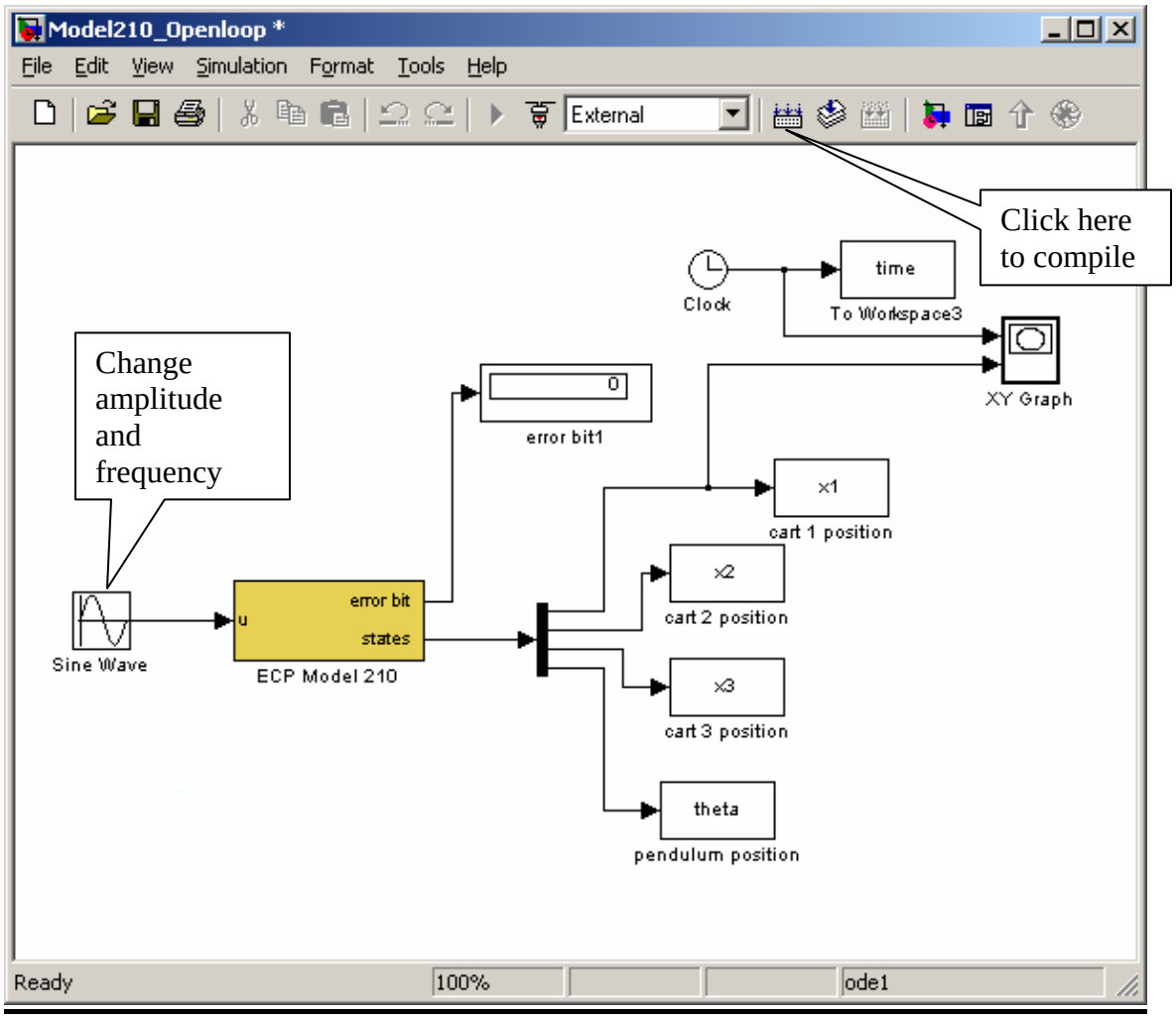
Step 3: Run it.



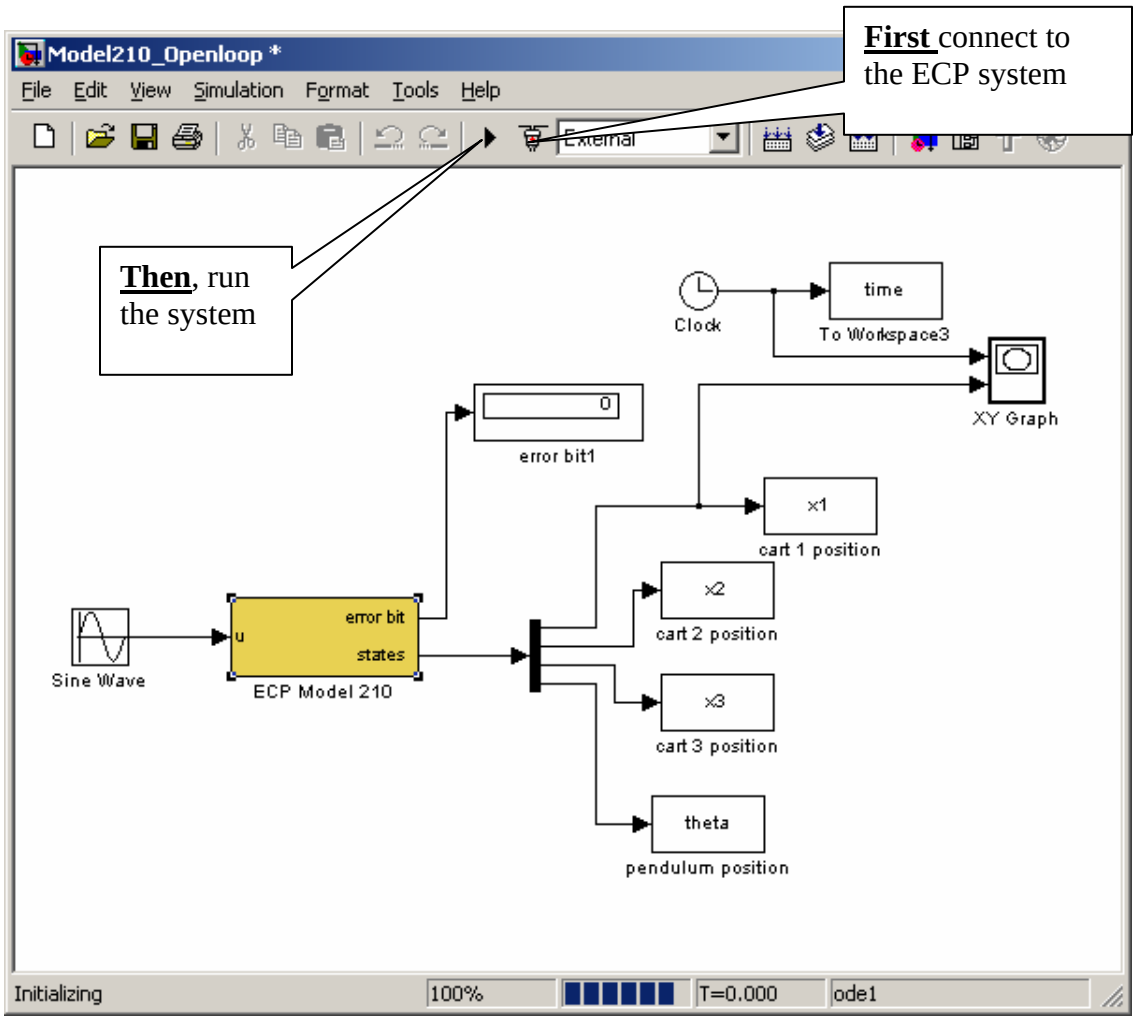
Model210_Openloop.mdl

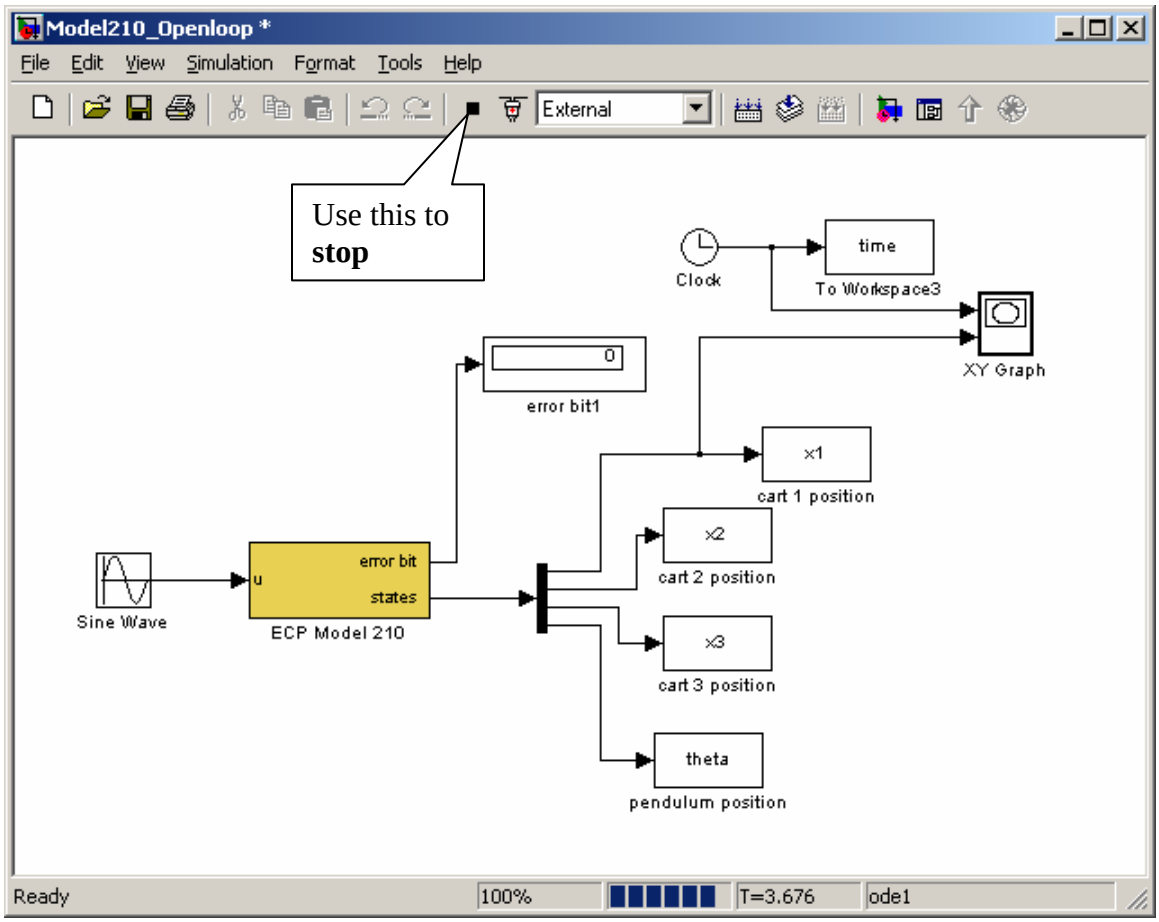
The first thing we usually need to do is to identify the system by constructing a Bode plot and fitting the measured frequency response to the expected frequency response of the transfer function we expect. To do this we will usually use the Simulink file **Model210_openloop.mdl**.

After changing the input signal, and changing any simulation parameters (select **Simulation -> Simulation Parameters),** you are ready to compile**.**



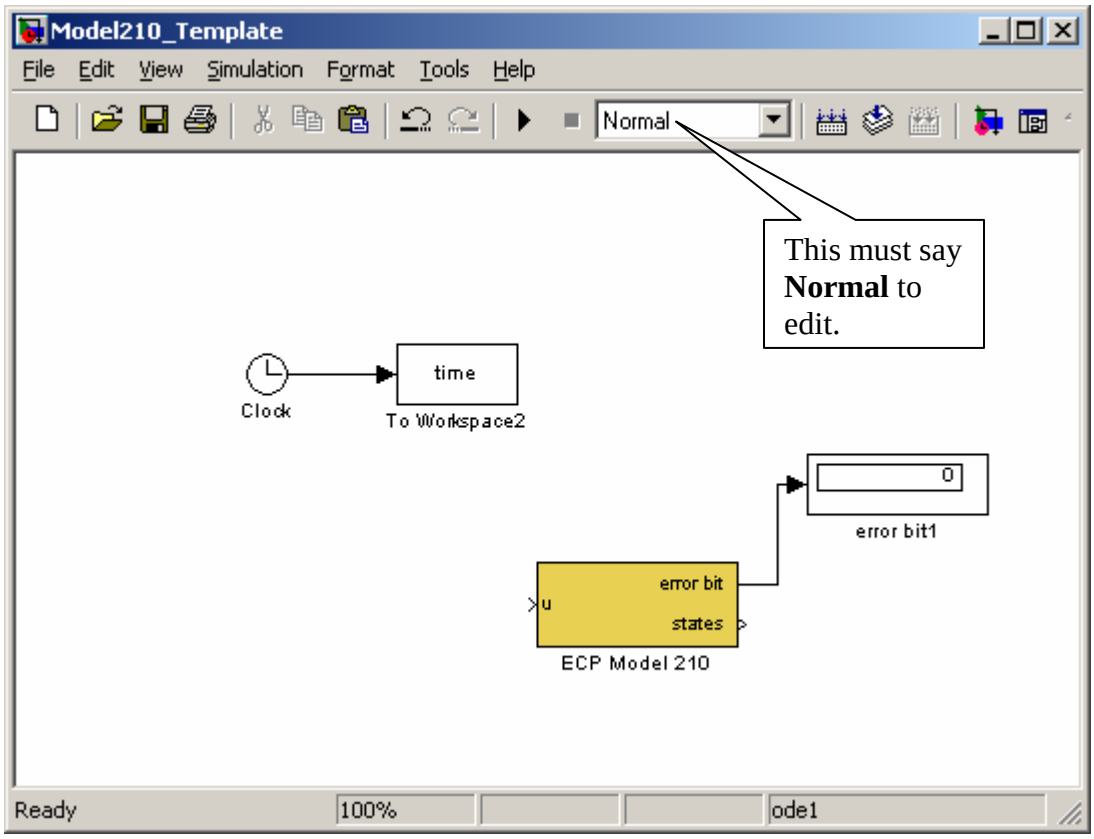
This particular implementation has the input signal in cm and the position of the three carts measured in cm. The output of the pendulum attachment is shown in radians. For this implementation, the position of the first mass is plotted as a function of time as the ECP system runs (the default is to plot for the first 20 seconds).


Model210_Openloop *
File Edit View Simulation Format Tools Help
External
First connect to the ECP system
Then, run the system
Clock
time
To Workspace3
XY Graph
0
error bit1
x1
cart 1 position
x2
cart 2 position
x3
cart 3 position
theta
pendulum position
Sine Wave
u
error bit
states
ECP Model 210
Initializing
100%
T=0.000
ode1

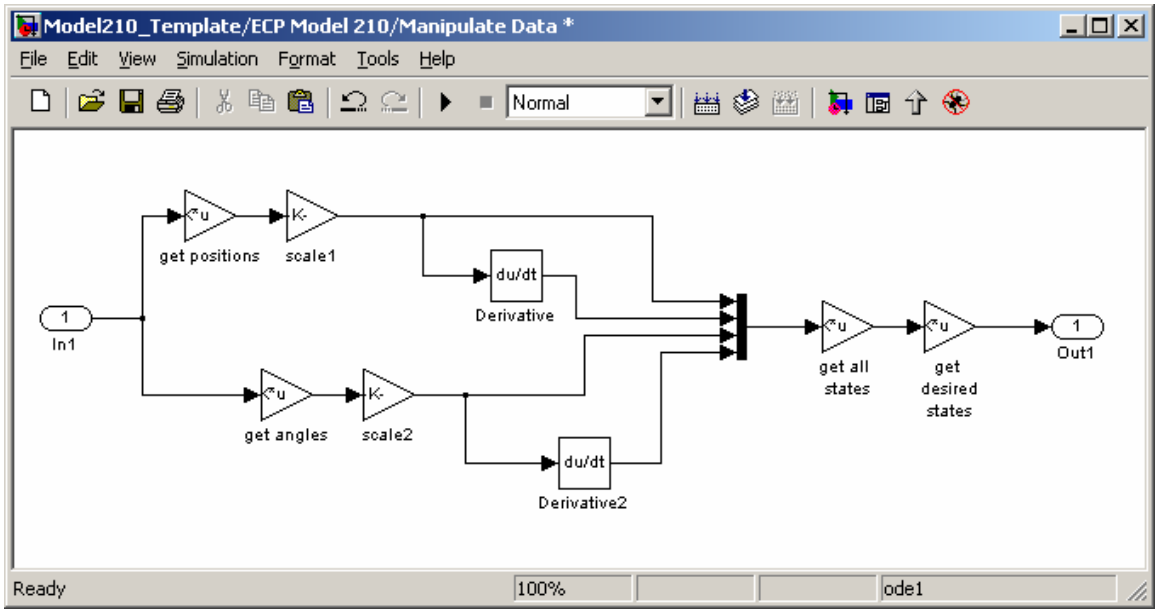

Model210_Openloop *
File Edit View Simulation Format Tools Help
External
Use this to stop
Clock
time
To Workspace3
XY Graph
0
error bit1
x1
cart 1 position
x2
cart 2 position
x3
cart 3 position
theta
pendulum position
Sine Wave
u
error bit
states
ECP Model 210
Ready
100%
T=3.676
ode1

Model210_Template.mdl

For most of our systems, we will utilize the basic skeleton shown below:



Note that this looks like **Model210_Openloop.mdl**, but it has different subsystems in the **ECP Model 210** block. If we click on this block we can see what's inside. It initially looks like the inside of **Model210_Openloop**, but the contents of the **Manipulate Data** block are different.



The matrix **get all states** is a matrix of 1's and 0's that sorts the states so the state vector is what we want:

1. the position of the first cart,
2. the velocity of the first cart,
3. the position of the second cart,
4. the velocity of the second cart,
5. the position of the third cart,
6. the velocity of the third cart,
7. the position of the pendulum, and
8. the velocity of the pendulum

We don't need all of the state for every model, so the matrix **get desired states** is a matrix of 1's and 0's so we only output the states we are going to be using. This is the only matrix you will have to modify (it should be modified in the Matlab driver file**).**