ME6404 – Fall 2016
Lab 4: Model Reference Control

Objectives:

1. Implement a model reference controller in simulation.
2. Implement model reference control on a real crane with a swinging payload.

Background:

Figure 1 shows the block diagram for a typical model reference controller system.

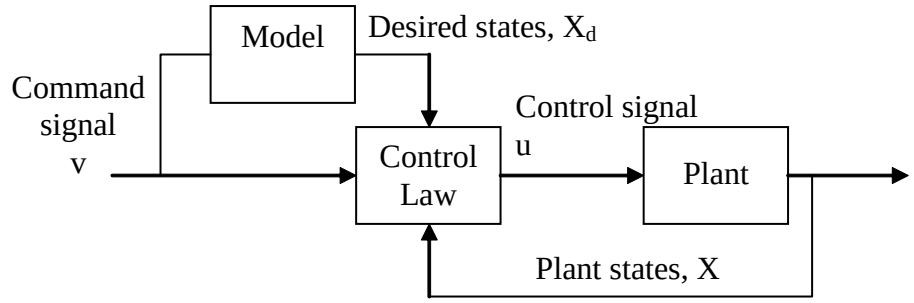


Figure 1: Model Reference Control

Model reference control uses a control law to calculate the control signal, u, such that the plant states, X, are driven to the desired states of the model, X_d. The command signal, v, dictates the model response (and therefore the desired model states, X_d). We can formulate the control law using the second method of Liapunov. We can use state space to represent the system:

$$Model: \dot{x}_d = Ax + Bv$$
$$Plant: \dot{x} = f(x,u,t)$$
$$Error: e = x_d - x$$
$$\dot{e} = \dot{x}_d - \dot{x}$$
$$\dot{e} = Ax_d + Bv - f(x,u,t)$$
$$\dot{e} = Ae + Ax + Bv - f(x,u,t)$$

Then, define a Liapunov function such that the error will be asymptotically stable:

$$V(e) = e^T Pe$$
$$\dot{V}(e) = \dot{e}^T Pe + e^T P\dot{e}$$
$$\dot{V}(e) = e^T (A^T P + PA)e + 2M$$
$$where,\ \ M = e^T P\left[Ax - f(x,u,t) + Bv\right] = scalar$$

The goal is to make $\dot{V}(e)$ negative so that the error will go to zero. To do this there are two parts:

1. Choose *P* so that $A^TP + PA = -Q$, where *Q* is positive-definite-real-symmetric
2. Choose *u* so that *M* is non-positive

You will need to simulate the system (a Matlab template is provided to help you get started). Then, you will implement your controllers on the bridge crane. An important lesson from this lab

is to understand the challenges involved in transforming a controller that works in the ideal, simulated-world, to a controller that works in the non-ideal, real-world. Often, this process involves making intelligent assumptions about the real-world plant.

Model reference control on the payload oscillation

The goal for this part is to force the payload oscillation to behave like a second-order model of your choosing. The block diagram is shown in Figure 2. The overall plant is the trolley and payload.

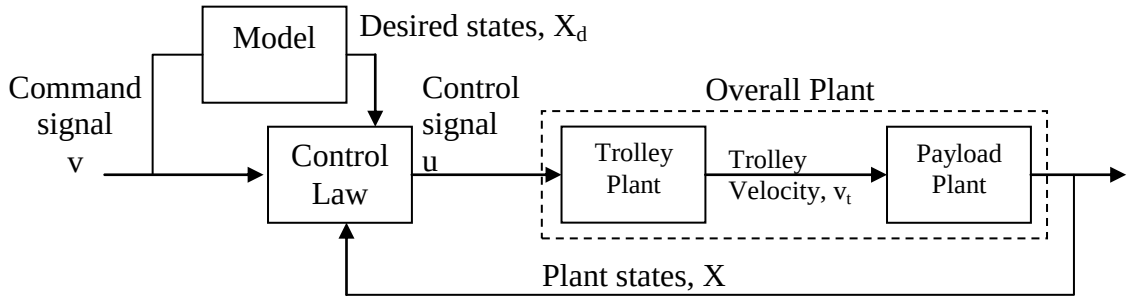


Figure 2: Model Reference Control on Payload Oscillation

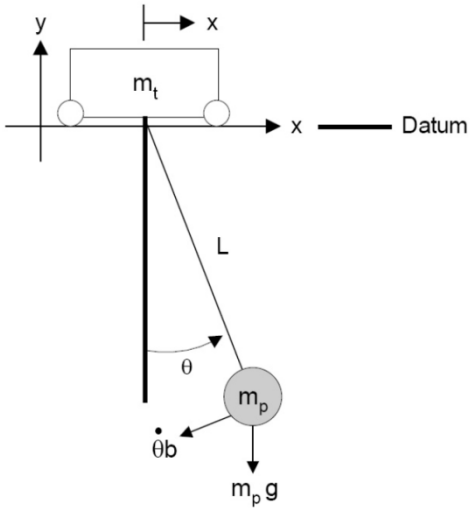


Figure 3: Pendulum Model of the Payload Plant

The pendulum-like payload plant can be obtained by considering Figure 3, which has the transfer function:

$$\frac{\theta(s)}{V_t(s)} = \frac{\left(\frac{-\omega_n^2}{g}\right)s}{s^2 + 2\xi\omega_n s + \omega_n^2} \qquad (3)$$

Where θ is the oscillation angle, v_t is the horizontal velocity of the trolley, g is gravity, ω_n is the natural frequency, and ζ is the damping ratio. The state space form is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\,\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_t(t) \qquad (4)$$

$$\theta(t) = \begin{bmatrix} 0 & \dfrac{-\omega_n^2}{g} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (5)$$

The second state, x_2, is equivalent to $-\theta L$, where L is the suspension length. This is also the horizontal deflection of the payload. The first state, x_1, is the integral of x_2.

1. Design and simulate a model reference controller in the form of Figure 2 to control the payload oscillation. Use the Matlab template or any other simulation software.
    - You can assume that the trolley can follow any velocity.
    - Use the state space model in (4) for the payload plant i.e. f(x,u,t). Assume ω_n = sqrt (g/L), with L = suspension length = 1.0m, and ζ = 0.001.
    - You will need to make the plant follow three models of the form:

      $$\begin{bmatrix} \dot{x}_{d1} \\ \dot{x}_{d2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_m^2 & -2\xi_m\omega_m \end{bmatrix} \begin{bmatrix} x_{d1} \\ x_{d2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v$$

      Model A: $\omega_m = \omega_n$, $\zeta_m = 0.1$

      Model B: $\omega_m = \omega_n$, $\zeta_m = 0.5$
      Model C: $\omega_m = \omega_n$, $\zeta_m = 0.8$
      where x_{d2} has the same physical meaning as x_2 (i.e. the horizontal deflection of the payload; x_{d1} is the integral of x_{d2}, and v is the command signal.
    - Determine what effect the changing model damping has on the system performance and response.
    - Investigate the effect of modeling errors in the natural frequency (the suspension length).
    - Use a pulse input for the command signal, v. Make your own choice for the length of the pulse, t_f.

2. Implement your controller on the bridge crane
    - The second state, $x_2 = -\theta L$, is the negative of the payload deflection. The camera measures this deflection directly. In the PLC code x_1 and x_2 are provided for you.
    - Implement on the trolley direction.
    - Use suspension length of 1.0 m.
    - The command signal, v, comes from the pendent/GUI button. Use an unshaped velocity profile. You choose the length of the command.
3. When you have successfully implemented your controller, investigate the following. Use simulation results to explain what you expect to see in experiment:
    - Investigate your controller performance in following increasingly unrealistic models. What are the largest values of ω_m and ζ_m where the controller output becomes unacceptably large, and the portable bridge crane becomes incapable of tracking the model?
        i. Start with $\omega_m = \omega_n$ and $\zeta_m = 0.5$. Then, increase ζ_m
        ii. Start with $\omega_m = \omega_n$ and $\zeta_m = 0.5$. Then, increase ω_m.

- Investigate the robustness of your controller to changes in the plant. Without recalculating the state space matrices, shorten and extend the suspension cable from the nominal 1.0m length. What are the length ranges for which your controller still works?
- Investigate your controller performance to external disturbances. With zero command signal (i.e. v=0, and no button press), and with cable length = 1.0m, disturb the payload (e.g. hit it).

Lab Procedure:

Computer simulation:

- A sample Matlab script file and simple Simulink model is provided to help get you started. There are many tutorials on the internet to help you become familiarized with Matlab and Simulink. The inbuilt help documents are also very useful.
- Be aware of units – all signals are metric in the templates provided, e.g. v and u are in m/s, x_1 m*s, x_2 should be in m.

Implementation on the Portable Bridge Crane:

1. Click on Simotion Scout shortcut link on the desktop.
2. If the "Bridge_Crane_2016master" is not already open, go to Project>Open>... The project you should be using is the "Bridge_Crane_2016master" found on storage path C:\Program Files/Siemens/Step7/s7/Bridge3 **IMPORTANT: make sure to have the correct project open under the correct path or else the lab will not work**
3. Navigate to Crane_of_FG_19042013>D435>Programs>MCC_Test then double click on the program "MCC_shape_Move_Y() to open the program.
4. You will be modifying two areas of the system to change the parameters and the input equation. The equation is located in the program opened in step 3 within Simotion Scout and the parameters are located within a submenu of the Bridge Crane GUI.
5. Double click on the block shown in Figure 4.

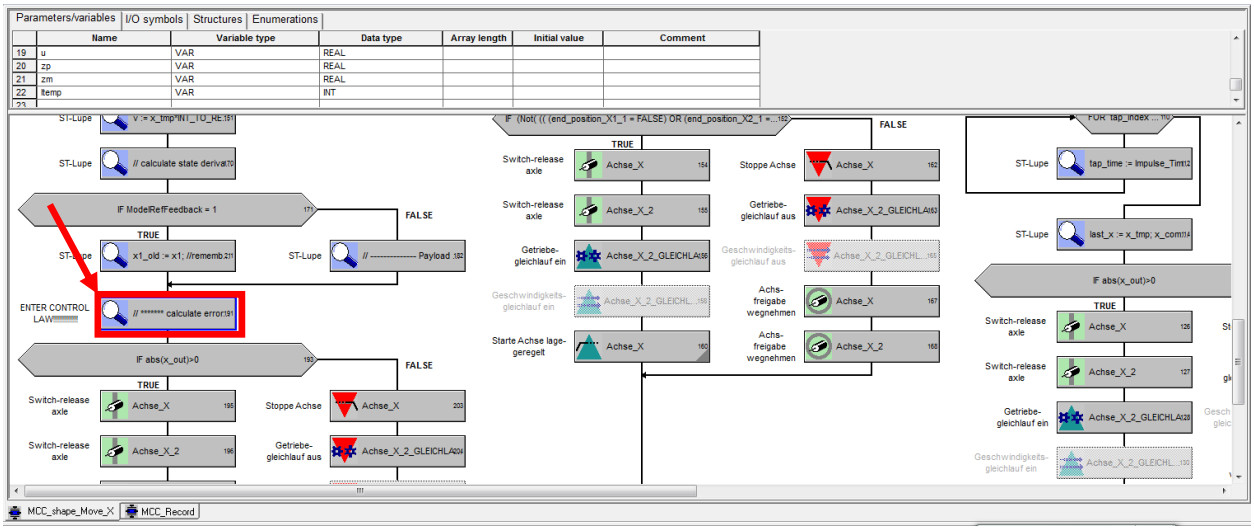


Figure 4: Block program of Y-axis move in Simotion Scout

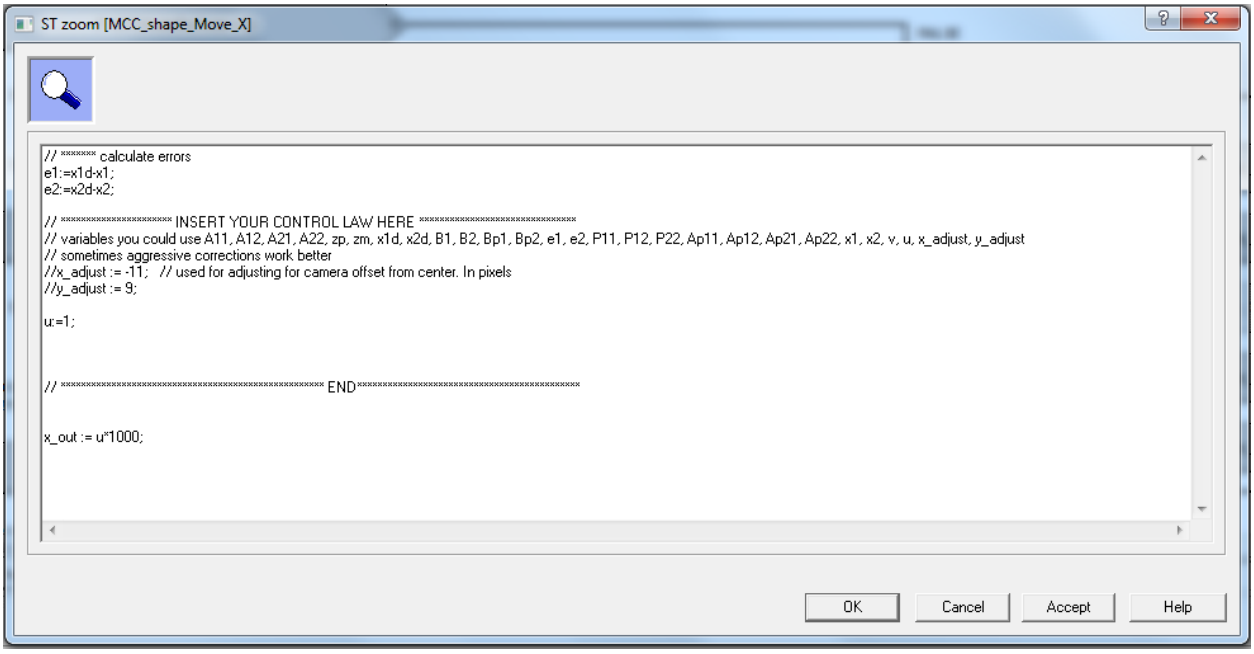


Figure 5: Code in block to enter control law

6. Figure 5 above shows the opened block of code
    a. The executed code is located in this block as well as in program you opened previously. You should read this and look over the program to have a better understanding of the code e.g. the simple approximations used to find the state integrals and derivatives.
    b. You will need to write a few lines of code for your control law in the indicated space. The control signal, u, should be in m/s.
    c. Use '**:=**' as the assignment operator e.g. 'u := 100;'
    d. Use '=' for testing logic statements i.e. IF … THEN… END_IF;
    e. Each line of code must end with '**;**'
    f. Templates for logic statements are in the command library located where the file structure you searched for the program used to be. Drag and drop these functions to use them.
    g. Standard functions (e.g. SQRT() and ABS()) can be found by using Help from the menu bar, and then find 'Standard Functions' in the index.
    h. Compile the code using button (1) in Figure 6. Make sure button (2) is depressed. This puts the PLC online with Simotion Scout.

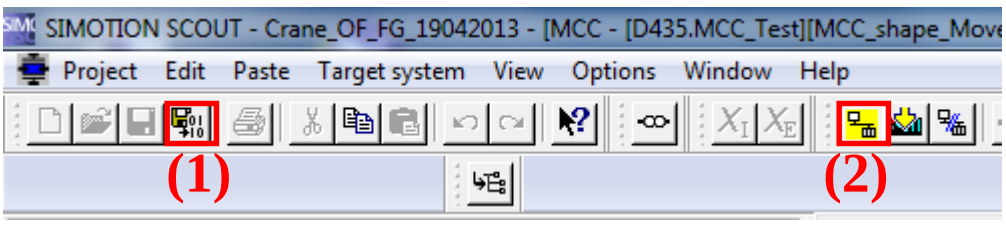


Figure 6: Simotion Scout menu bar

i. Download your new compiled code to the PLC using the “Download” button (3) in Figure 7.

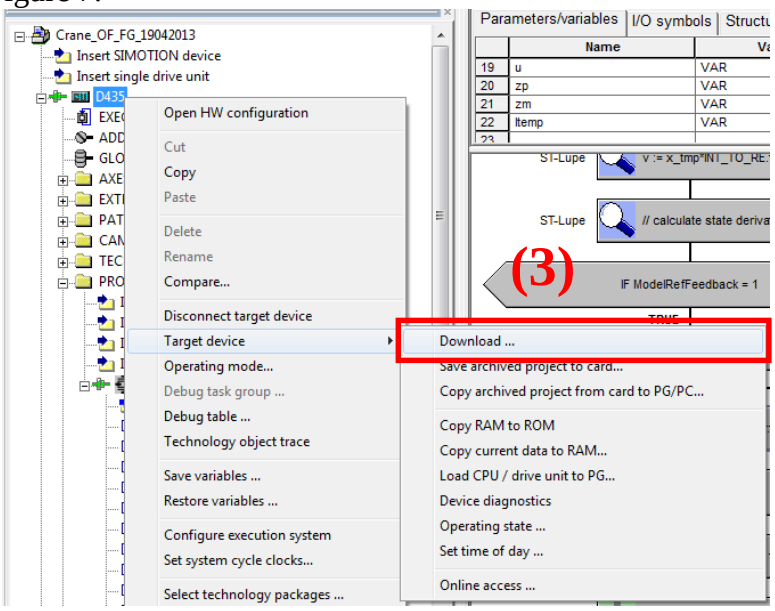


Figure 7: To download to PLC

j. Check the box for Copy Ram to Rom and hit YES. Click yes/OK on all pop up dialogs. See Figure 8.

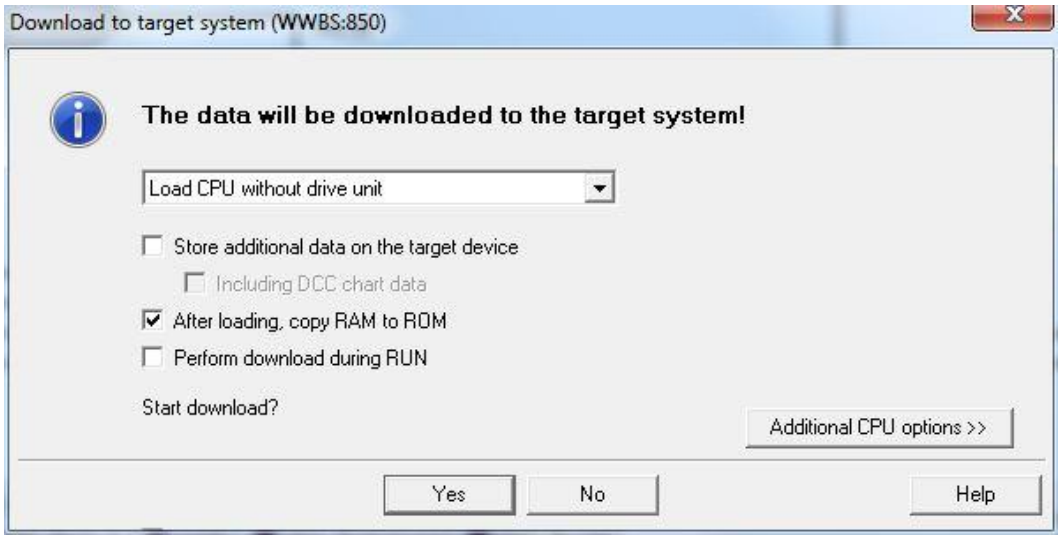
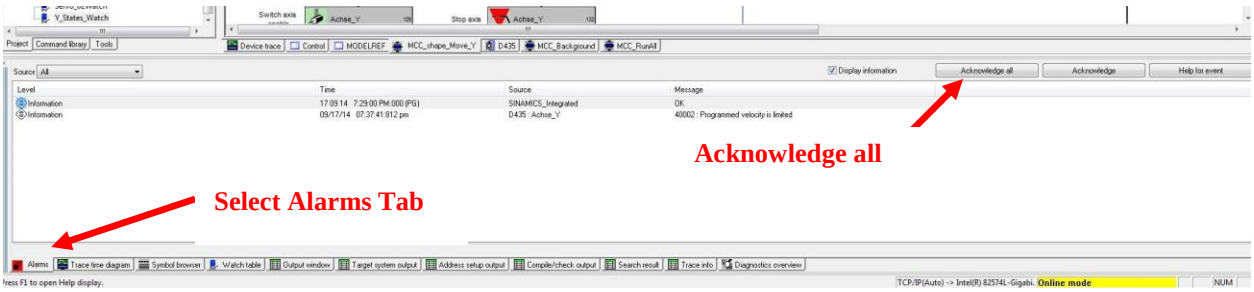
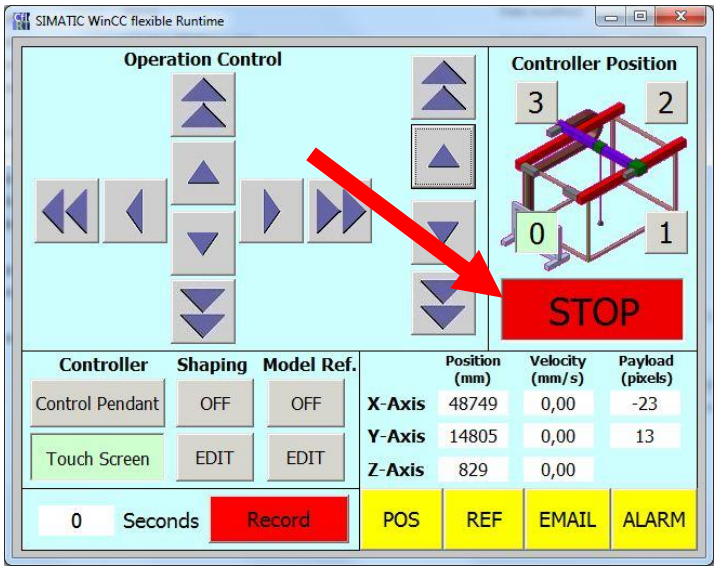


Figure 8: Download pop-up menu

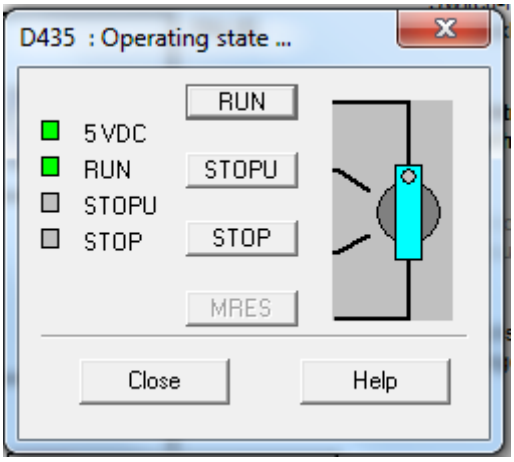
k. If the trolley will not move check the following: (1) Acknowledge any alarms in Scout as shown in Figure 9a, (2) make sure the GUI is in START mode as shown in Figure 9b, (3) check if the program is running as seen in Figure 9c. If not click run. This popup can be accessed by highlighting “D435” and either pressing keyboard command CTRL+i or selecting in the menu bar Target System>>Control Operating State.



a) Acknowledge Alarms



b) GUI Must be in Start Mode



c) To control operating state

Figure 9: Troubleshooting the Crane

7. Navigate to the Bridge Crane GUI
    a. "ON/OFF" under the "Model Ref." controls whether to do model reference control. You should set this to off when you are not running model reference so that the states can be reset to zero and you can move the crane around normally.

b. In the interface, select edit under “Model Ref.”. This will allow you to modify the parameters for model reference control.
c. “Axis” controls the bridge or trolley axis direction. Make sure to use only the trolley axis. See figure 10.
d. Set “Feedback” to 2 if the payload position is the plant state.
e. There are variables that store the values for the P matrix, model matrices (A and B), and plant matrices (Ap and Bp). The numbers after the letters correspond to the row and column, e.g. “A21” is the second row, first column value for the A matrix. You need to modify these.

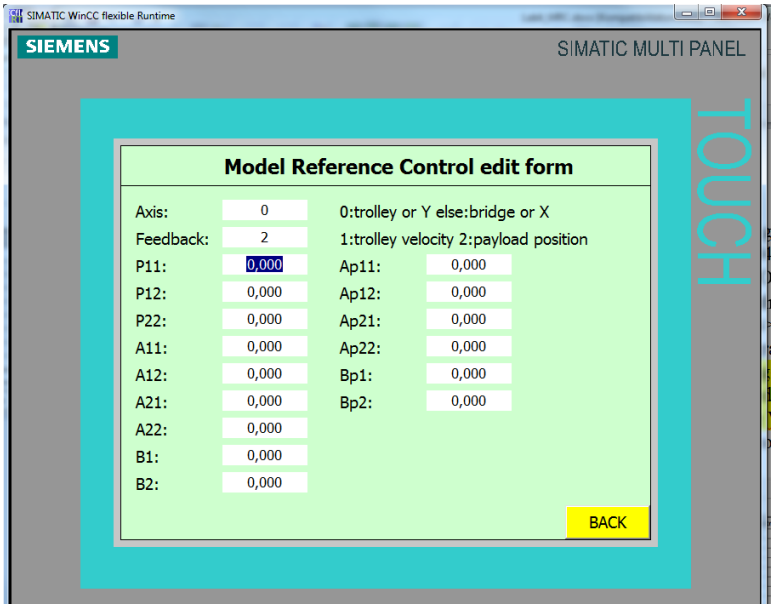


Figure 10: Model reference inputs

f. Be mindful that the commas in the input box indicate a decimal place. Keep in mind that the variables are all “Real” variables with 3 decimal places. If hashtags are displayed in an input box, the value of the variable can be verified by highlighting “Vars” under Programs in the Project Tree. The variables will be displayed in the Symbol Browser as shown in Figure 11.
g. Various variables can be monitored in Simotion Scout by selecting “Global device Variables” and looking at the “symbol browser” located at the bottom of the screen (Figure 11). These variables contain camera measurements, model and plant states, control signal u, and error between the model and plant states. These are there to help you debug. Make sure the system is “online” as shown in Figure 6 to see these values updated in real-time. Alternatively, a watch table called “Y_States_Watch” contains only the most important variables. As shown in Figure 12, the table is accessed under Monitor at the bottom of the Project Tree.

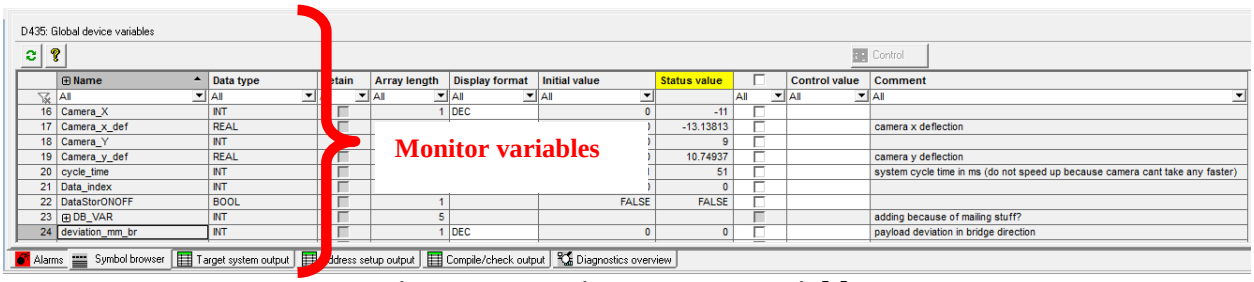


Figure 11: Various system Variables

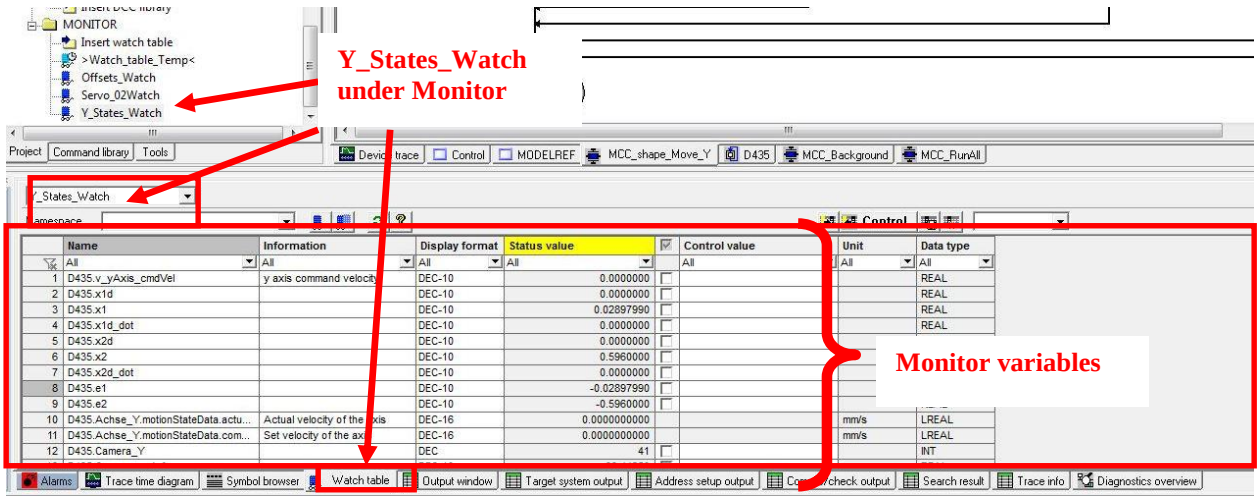


Figure 12: "Y_States_Watch" Watch Table

8. When you have finished modifying the variables, set "Model Ref." to ON in the GUI window and begin moving the crane around.
9. Use the shaper menu in the GUI to set a specific input. Use the trapezoidal command as used in other labs. Make sure to turn the shaping "ON" in order to use it. Both shaping and model reference control can be on at the same time. Make sure the pulse length is short enough so that the crane does not hit the ends. Remember to keep the button held down long enough to ensure that the command input signal, v, contains only one pulse (Recall that when the button is released, there will be a second, negative pulse that takes the crane back to its starting point).
    a. NOTE: If the model reference control does not seem to be responding correctly, first check the GUI to see that the payload position recorded by the camera is being updated. If not, turn the model reference control to OFF and place your hand under the camera lens until the camera begins updating again.
10. Use the recording option get the data as you did in previous labs.

Lab report (one per team)

Present your work and prove that you have met the objectives using **less than 3 pages of text** (not including Figures). Your report should include the following:

1. Summarize how you formed your control law.
2. Compare the simulated results with the experimental results in following models A, B, and C.

3. Summarize your findings in task.
4. Discuss some of the challenges in transferring a simulated, theoretical controller, to a real-world, digitized PLC implementation. Your discussion should mention the following:
    a. Assumptions you needed to make
    b. Sampling time and approximate digital methods for calculating continuous quantities (e.g. integrals and derivatives)
    c. Reliance on sensors for feedback
    d. Actuator limitations

The report is due at the beginning of lab 5 on October 7th