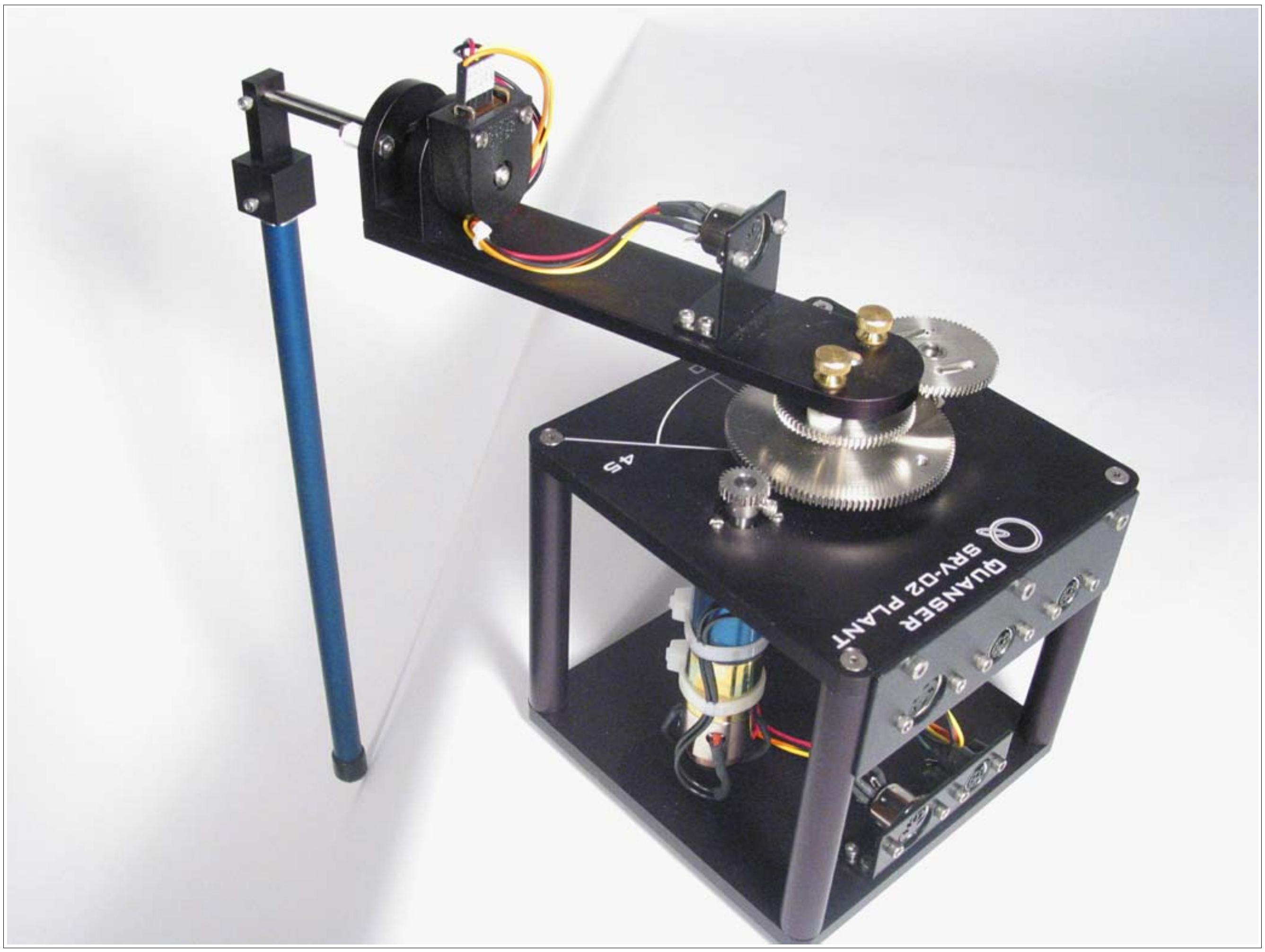
QUANSER
INNOVATE. EDUCATE.

SRV02-Series

Rotary Experiment # 6

Rotary Gantry



Student Handout

SRV02-Series
Rotary Experiment # 6

Rotary Gantry
Student Handout

1. Objectives

The objective in this experiment is to design a state-feedback controller for the rotary gantry module using the pole-placement technique. The controller will allow you to command a load position and should position the gantry tip with no overshoot and no error.

Upon completion of the exercise, you should have have experience in the following:

- How to mathematically model the rotary gantry system.
- To linearize the model about an equilibrium point.
- To use pole-placement in designing a state-feedback controller.
- To design and simulate a WinCon controller for the system.
- How to introduce Integral Action to eliminate steady-state error.

2. System Requirements

To complete this Lab, the following hardware is required:

[1] Quanser UPM 2405/1503 Power Module or equivalent.
[1] Quanser MultiQ PCI / MQ3 or equivalent.
[1] Quanser SRV02 servo plant.
[1] Quanser ROTPEN – Rotary Pendulum.
[1] PC equipped with the required software as stated in the WinCon user manual.

- The required configuration of this experiment is the **SRV02-E(T)** in the **High-Gear** configuration with a **UPM 2405** power module and a suggested **gain cable** of **1**.

- *It is assumed that the student has successfully completed Experiment #0 of the SRV02 and is familiar in using WinCon to control the plant through Simulink.*

- *It is also assumed that all the sensors and actuators are connected as per dictated in the SRV02 User Manual and the Rotary Pendulum User Manual.*

3. Mathematical Model

Figure 1 below depicts the Rotary Gantry module coupled to the SRV02 plant in the correct configuration. The Module is attached to the SRV02 load gear by two thumbscrews. The Pendulum Arm is attached to the module body by a set screw.

Figure 1 - *SRV02 with a ROTPEN Module*

The following table is a list of the nomenclature used is the following illustrations and derivations.

Symbol	***Description***	***Symbol***	***Description***
L	*Length to Pendulum's Center of mass*	h	*Distance of Pendulum Center of mass from ground*
m	*Mass of Pendulum Arm*	J_{cm}	*Pendulum Inertia about its center of mass*
r	*Rotating Arm Length*	V_x	*Velocity of Pendulum Center of mass in the x-direction*
θ	*Servo load gear angle (radians)*	V_y	*Velocity of Pendulum Center of mass in the y-direction*
α	*Pendulum Arm Deflection (radians)*		

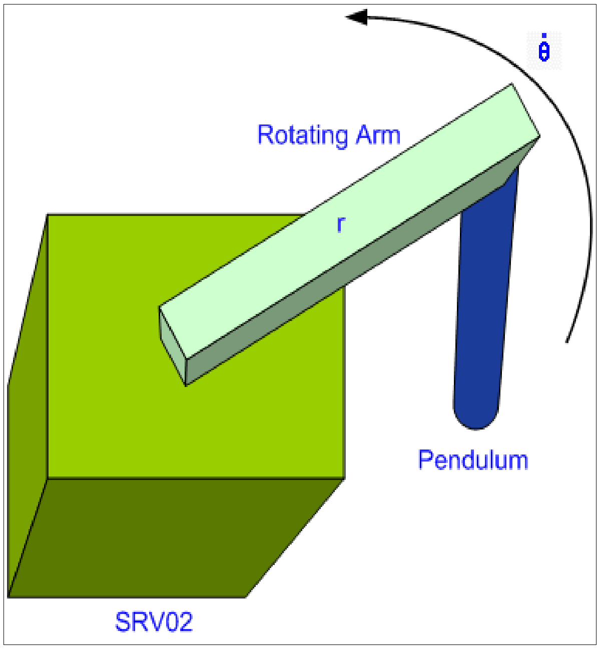


Figure 2 - *Top View of Rotary Gantry*

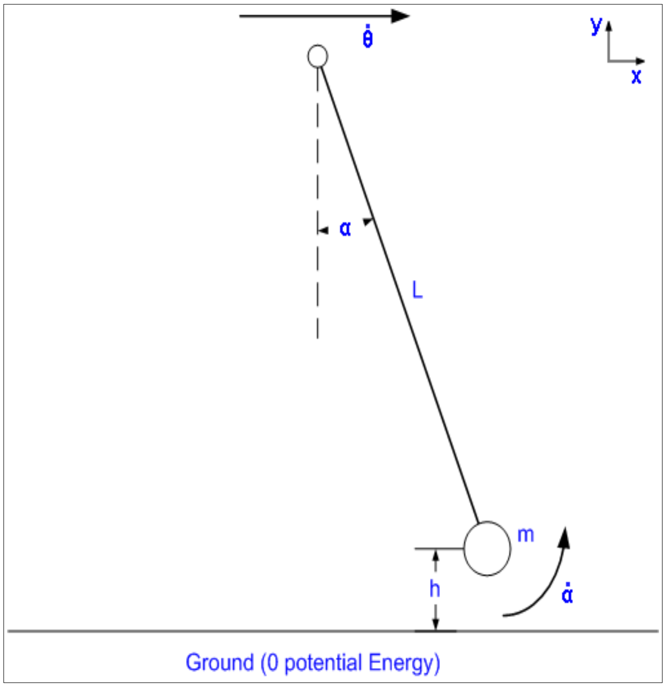


Figure 3 - *Side View with Pendulum in Motion*

Figure 2 above depicts the rotary gantry in motion. Take note of the direction the arm is moving. Figure 3 depicts the pendulum as a lump mass at half the length of the pendulum. The arm is displaced with a given **α**. Notice that the direction of **θ** is now in the x-direction of this illustration. We shall begin the derivation by examining the velocity of the pendulum center of mass.

Referring back to Figure 3, we notice that there are 2 components for the velocity of the Pendulum Lumped Mass:

$$V_{Pendulum\,center\,of\,mass} = L\cos\alpha(\dot{\alpha})\,\hat{x} + L\sin\alpha(\dot{\alpha})\,\hat{y} \qquad \textbf{[3.1]}$$

We also know that the pendulum arm is also moving with the rotating arm at a rate of:

$$V_{arm} = r\dot{\theta} \qquad \textbf{[3.2]}$$

Using equations **[3.1]** & **[3.2]** and solving for the **x** & **y** velocity components:

$$V_x = L\cos\alpha(\dot{\alpha}) + r\dot{\theta}$$
$$V_y = L\sin\alpha(\dot{\alpha}) \qquad \textbf{[3.3]}$$

Equation **[3.3]** leaves us with the complete velocity of the pendulum. We can now proceed to derive the system dynamic equations.

3.1 Deriving The System Dynamic Equations

Now that we have obtained the velocities of the pendulum, the system dynamic equations can be obtained using the Euler-Lagrange formulation. We obtain the Potential and Kinetic energies in our system as:

Potential Energy - The only potential energy in the system is gravity:

$$V = P.E._{Pendulum} = m\,g\,h = m\,g\,L(1-\cos\alpha) \quad \textbf{[3.4]}$$

Kinetic Energy - The Kinetic Energies in the system arise from the moving hub, the velocity of the point mass in the x-direction, the velocity of the point mass in the y-direction and the rotating pendulum about its center of mass:

$$T = K.E._{Hub} + K.E._{Vx} + K.E._{Vy} + K.E._{Pendulum} \quad \textbf{[3.5]}$$

**Note: Since we have modeled the pendulum as a point mass at its center of mass, the total kinetic energy of the pendulum is the kinetic energy of the point mass plus the kinetic energy of the pendulum rotating about its center of mass.*

The moment of inertia of a rod about its center of mass is:

$$J_{cm} = \frac{1}{12} M R^2$$

since we've defined L to be half the pendulum length, then R in this case would be equal to 2L. Therefore the moment of inertia of the pendulum about its center of mass is:

$$J_{cm} = \frac{1}{12} M R^2 = \frac{1}{12} M (2L)^2 = \frac{1}{3} M L^2 \quad \textbf{[3.6]}$$

Finally, our complete kinetic energy **T** can be written as:

$$T = \frac{1}{2} J_{eq} \dot{\theta}^2 + \frac{1}{2} m (L\cos\alpha(\dot{\alpha}) + r\dot{\theta})^2 + \frac{1}{2} m (L\sin\alpha(\dot{\alpha}))^2 + \frac{1}{2} J_{cm} \dot{\alpha}^2 \quad \textbf{[3.7]}$$

After expanding equation **[3.7]** and collecting terms, we can formulate the **Lagrangian**:

$$L = T - V = \frac{1}{2} J_{eq} \dot{\theta}^2 + \frac{2}{3} m L^2 \dot{\alpha}^2 + mLr\cos\alpha(\dot{\alpha})(\dot{\theta}) + \frac{1}{2} m r^2 \dot{\theta}^2 + mgL(\cos\alpha - 1) \quad \textbf{[3.8]}$$

Our 2 generalized co-ordinates are **θ** and **α**. We therefore have 2 equations:

$$\frac{\delta}{\delta t}\left(\frac{\delta L}{\delta \dot{\theta}}\right)-\frac{\delta L}{\delta \theta}=T_{output}-B_{eq}\dot{\theta} \quad \textbf{[3.9]}$$

$$\frac{\delta}{\delta t}\left(\frac{\delta L}{\delta \dot{\alpha}}\right)-\frac{\delta L}{\delta \alpha}=0 \quad \textbf{[3.10]}$$

Solving Equations **[3.9]** & **[3.10]** and linearizing about α = 0, we are left with :

$$(J_{eq}+mr^2)\ddot{\theta}+mLr\ddot{\alpha}=T_{output}-B_{eq}\dot{\theta} \quad \textbf{[3.11]}$$

$$\frac{4}{3}mL^2\ddot{\alpha}+mLr\ddot{\theta}+mgL\alpha=0 \quad \textbf{[3.12]}$$

Referring back to *Experiment # 1 – Position Control*, we know that the output Torque on the load from the motor is:

$$T_{output}=\frac{\eta_m\eta_g K_t K_g(V_m-K_g K_m\dot{\theta})}{R_m} \quad \textbf{[3.13]}$$

Finally, by combining equations **[3.11], [3.12] & [3.13]**, we are left with the following state-space representation of the complete system:

$$\begin{bmatrix}\dot{\theta}\\ \dot{\alpha}\\ \ddot{\theta}\\ \ddot{\alpha}\end{bmatrix}=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{bd}{E} & \frac{-cG}{E} & 0\\ 0 & \frac{-ad}{E} & \frac{bG}{E} & 0\end{bmatrix}\begin{bmatrix}\theta\\ \alpha\\ \dot{\theta}\\ \dot{\alpha}\end{bmatrix}+\begin{bmatrix}0\\ 0\\ c\frac{\eta_m\eta_g K_t K_g}{R_m E}\\ -b\frac{\eta_m\eta_g K_t K_g}{R_m E}\end{bmatrix}V_m$$

Where:

$$a=J_{eq}+mr^2 \qquad E=ac-b^2$$

$$b=mLr \qquad G=\frac{\eta_m\eta_g K_t K_m K_g^2+B_{eq}R_m}{R_m}$$

$$c=\frac{4}{3}mL^2$$

$$d=mgL$$

In the typical configuration of the SRV02 & the ROTPEN (Pendulum/Gantry) system, the above state space representation of the system is:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 39.32 & -14.52 & 0 \\ 0 & -81.78 & 13.98 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 25.54 \\ -24.59 \end{bmatrix} V_m$$

3.2 Pre Lab Assignment

The purpose of the lab is to design a state-feedback controller that will place the tip of the arm (**θ + α**) at a given command. The controller must also meet the following criteria:

- The Controller must place the tip of the Pendulum at the desired location with zero overshoot.
- There should be zero steady-state error in the position of the Pendulum tip.

As a pre-requisite to the lab, you are required to calculate the state-feedback gain vector ***k*** such that the feedback law ***u = -kx*** will produced the following closed-loop poles:

p = (-30 , -8 , -4 , -3.5)

These particular poles were chosen to practically meet the controller specifications stated above. By choosing all poles to be real, we can ensure that there will be no overshoot when placing the tip of the pendulum.

**Hint: This problem requires you to setup the eigenvalue equation:*

*s**x** = A**x** , where A is your closed loop matrix (use the feedback law), and s is a closed-loop pole. Since there are 4 poles, you will need to solve 4 equations that will yield the feedback gain vector **k**.*

4. In Lab Procedure

The Rotary Gantry is an ideal experiment intended to model an actual gantry crane. It is an interesting control experiment as the emphasis isn't on a fast response, rather the emphasis is on the motion of the pendulum tip where in reality a heavy payload would be. An example would be a loading dock crane that must place a crate in a cargo container. This experiment is also ideal in studying the practical applications of introducing some integral action to eliminate steady-state error.

The purpose of the lab is to design a state-feedback controller that will place the tip of the arm (**θ + α**) at a given command. The controller must also meet the following criteria:

- The Controller must place the tip of the Pendulum at the desired location with zero overshoot.
- There should be zero steady-state error in the position of the Pendulum tip.

4.1 Part I - Verification & Simulation

The first part of this lab will be to verify the state-feedback gain vector that you have calculated in the pre-lab. Once you have verified and successfully simulated your controller, you will then implement the controller on the actual plant.

The first task upon entering the laboratory is to familiarize yourself with the system. The pendulum deflection signal (**α**) should be connected to encoder channel #1 and the servomotor's position signal (**θ**) should be connected to encoder channel #0. Analog Output channel #0 should be connected to the UPM (Amplifier) and from the amplifier to the input of the servomotor. This system has one input ($\mathbf{V_m}$) and two outputs (**θ** & **α**).

You are now ready to begin the lab. Launch MATLAB from the computer connected to the system. Under the *“SRV02_Exp6_Gantry”* directory, begin by running the file by the name *“Setup_SRV02_Exp6.m”.* This MATLAB script file will setup all the specific system parameters and will set the system state-space matrices A,B,C & D. You are now ready verify your previously calculated gains.

At this point, you should enter in the calculated ***k*** vector from the pre-lab.
Ex: $\mathbf{k = [k_1\ \ k_2\ \ k_3\ \ k_4\,]}$, where k_1, k_2, k_3, k_4 are your calculated gains.

The MATLAB ***eig*** function returns the eigen-vector (set of eigenvalues) of a given matrix. Since we will be implement the feedback control law ***u = -kx***, the closed-loop state-space matrix is ***A – B*k***. At the MATLAB command window, enter : ***eig (A – B*k)***. The returned vector will be the location of the closed-loop poles. If this vector matches the poles that were required: **p = (-30 , -8 , -4 , -3.5)**, then you have calculated the correct gains and you are ready to simulate the controller. If the returned poles are not located at the desired location, the gains were not calculated correctly. You should re-calculate the gains.

Once you have ensured that the closed-loop poles are at the desired location, you can continue on and simulate your controller. Under the same directory, open a Simulink model called *"s_SRV02_Rotary_Gantry.mdl".* This model is a simulation of the Rotary Gantry system with a feedback law ***u = -kx****.* Start the simulation. You should have 3 scopes open that are displaying alpha (**α**), theta (**θ**) and gamma (**θ** + **α**). You should be seeing the simulation response to a square-wave command of ±45°. The 3 responses you are seeing should look similar to Figure 4 below.

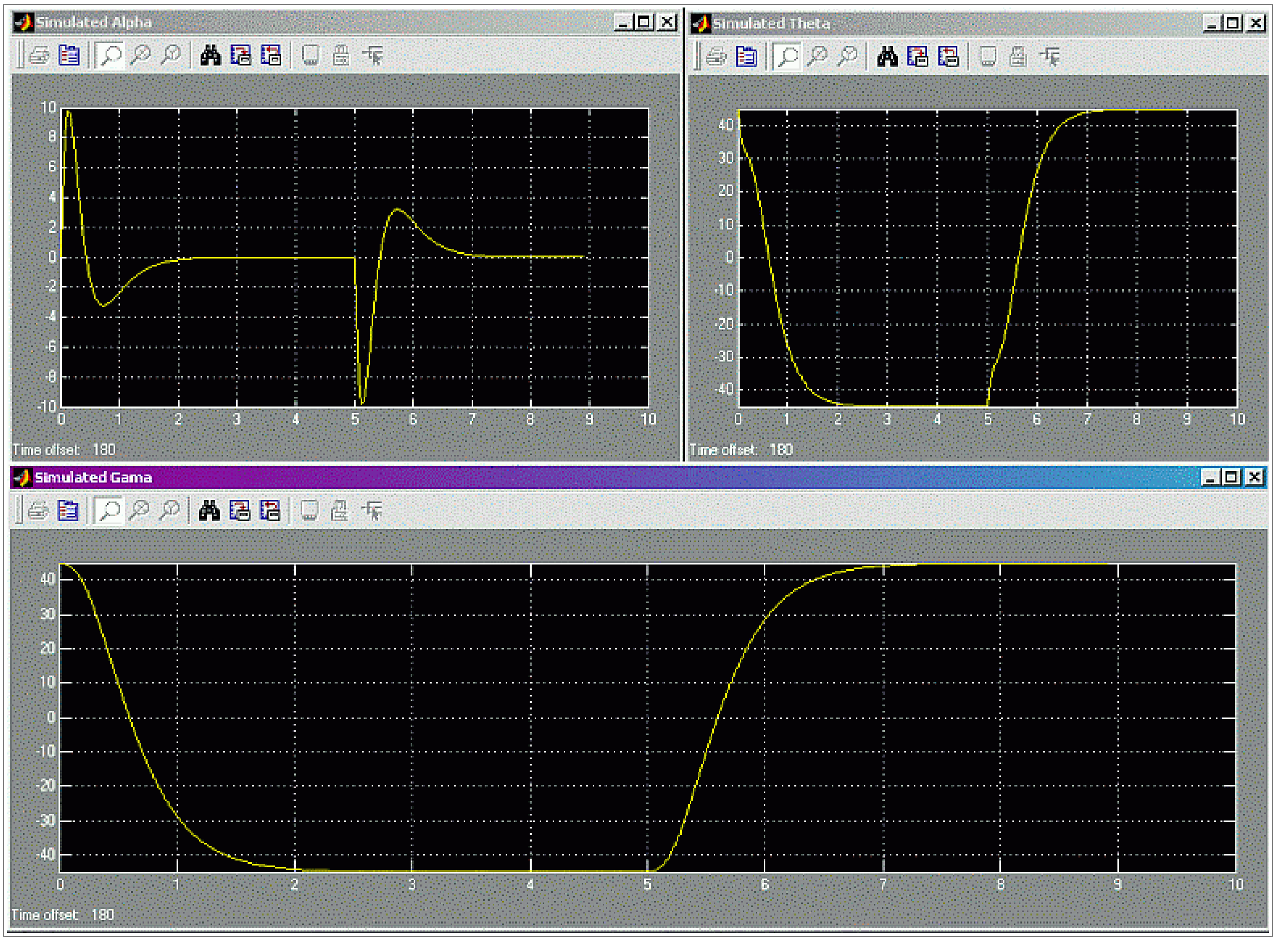


Figure 4 - *Simulation of Closed-Loop System*

As you can seen from the above simulation, the controller places the tip with no overshoot and zero steady state error. In simulation, the controller has met all the objectives. If your simulation matches Figure 4 above, you can continue and implement the controller.

4.2 Part II - Implementing the Controller

After verifying the calculated controller gains, it is time to implement the controllers on the actual system. In the same working directory, open a Simulink model called *"q_SRV02_Rotary_Gantry.mdl".* This model has the I/O connection blocks linking to the physical plant as well as a simulated block to compare real and simulated results. You may now proceed to ***"Build"*** the controller through the WinCon menu. After the code has compiled, start the controller through WinCon and open up two scopes; one for **alpha** and another for **gamma**. The following graph is of **gamma** and its response to a command of 40°.

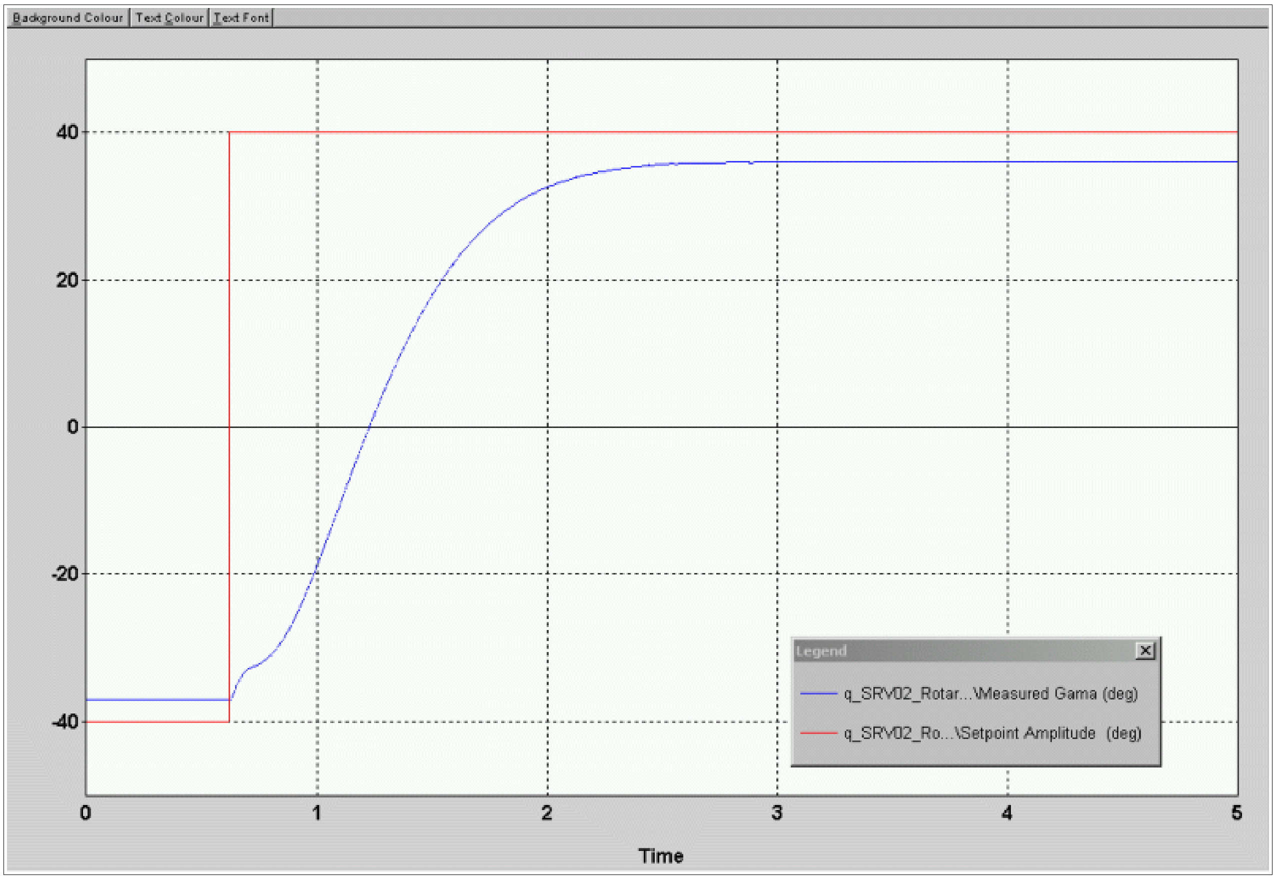


Figure 5 - *Gamma Response*

Notice the error is the response. This response is undesirable and can lead to substantial damage in a real situation. The best way of eliminating steady-state error is to introduce integral action into the controller.

****WARNING: If at any point the system is not behaving as expected, make sure to immediately press STOP on the WinCon server.***

****If at any time you hear a high frequency 'hum' from the system, this is an indication that the gains are too high and you need to re-calculate your controller.***
Under the same directory, open a Simulink file called *"q_SRV02_Rotary_Gantry_I.mdl".*

This model is the exact same as before except for the addition of an integral term in the feedback loop. You should notice that the integral action is being performed on the error in **gamma** as this is the variable we are interested in.

Double click on the K_i block and that will bring up a slider gain that you will be *tuning* to achieve a desired response. It should be noted that the *tuning* of system gains has always been an integral part of a controller's design. The initial gains were calculated based on a simplified linear model and a design is only complete when the gains are *tuned* to the specific plant.

"Build" this model and start it through WinCon. You should bring up a scope of the measured **gamma** and plot it against the simulated response. Start to vary the slider gain K_i and take note on its effect on the response. As you increase the gain, what happens to the response? What happens when you decrease the gain?

When you have *tuned* the integral gain, the response of the system should follow your original simulation and the controller should meet all the required specifications.

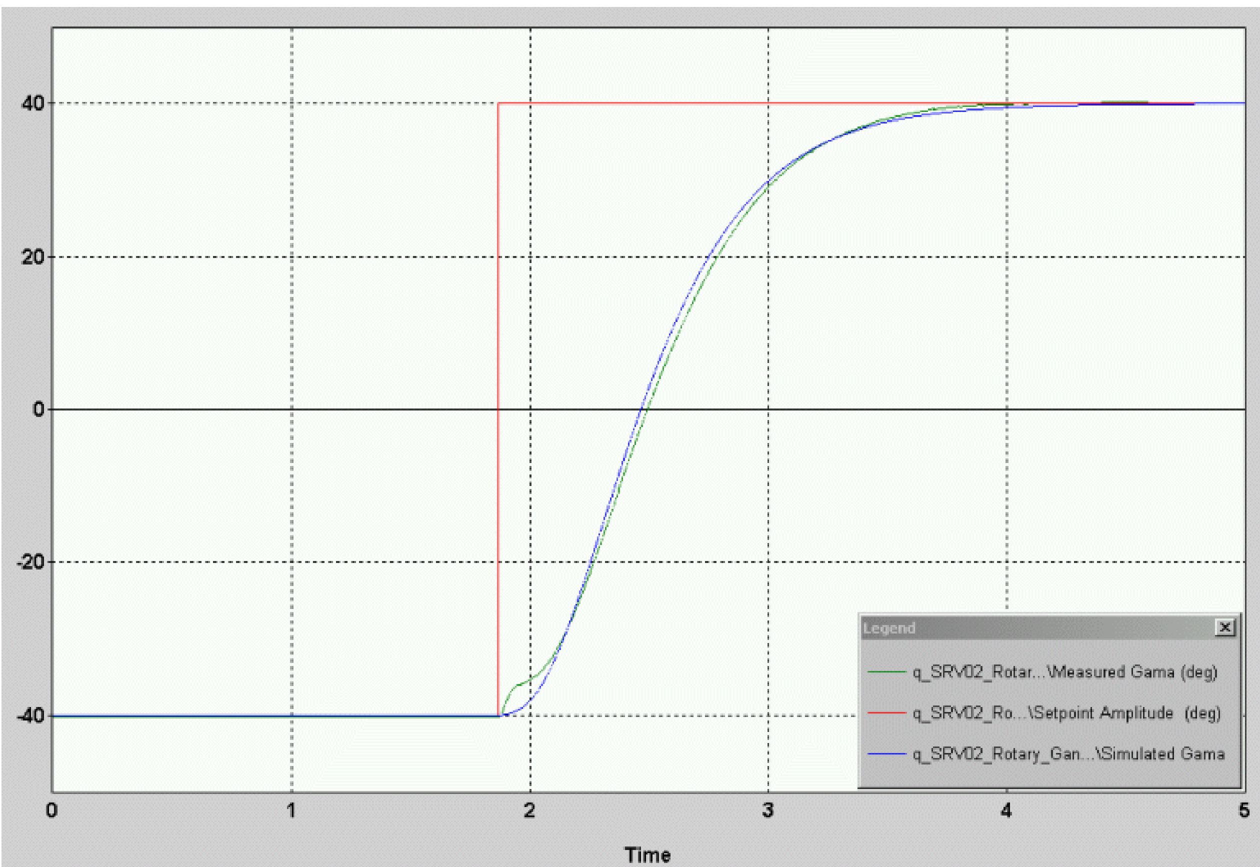


Figure 6 - *Command, Measured and Simulated Response*

With a properly chosen K_i, the controller will place the pendulum tip with no overshoot and zero steady-state error.

5. Post Lab Question and Report

Upon completion of the lab, you should begin by documenting your work into a lab report. Included in this report should be the following:

i. In the Pre-Lab assignment, you were asked to calculate the state-feedback gain vector that will place the closed-loop poles at the desired location. Make sure to include your complete solution and final ***k*** vector.

ii. In part I of the lab, you were asked to simulate the system with the calculated gain. Be sure to Include a simulation printout similar to Figure 4.

iii. In part II of the lab, you implemented the controller on the physical plant. Make sure to include the measured gamma plot to a setpoint command similar to Figure 5.

iv. As you were *tuning* the integral gain, you were asked to make some qualitative observations of the response as you varied the K_i variable. Include these observations.

v. After *fine-tuning* the controller, include the final value you chose for K_i. Make sure to include the final plot demonstrating the controller has met the requirements. This plot should look similar to Figure 6.

vi. Make sure to include your final controller gains and any re-iterative calculations made if any.

5.1 Post Lab Questions

1) After completing this laboratory, look back at the desired closed-loop poles of the system. What are the possible reasons of placing the poles at these locations? Given these poles, would you have anticipated the system to respond as it did? Explain.

2) Having performed this lab by using the pole-placement technique, what other control approaches would you consider for this system?

3) When the system was simulated with the calculated controller, there was no steady-state error. When the same controller was implemented on the actual plant, a state error appeared. What reasons could you determine for the discrepancy of the simulation and the actual response?

4) To eliminate the steady-state error, an integral gain was introduced into the feedback loop. Was this a sufficient approach in eliminating the error? Explain.