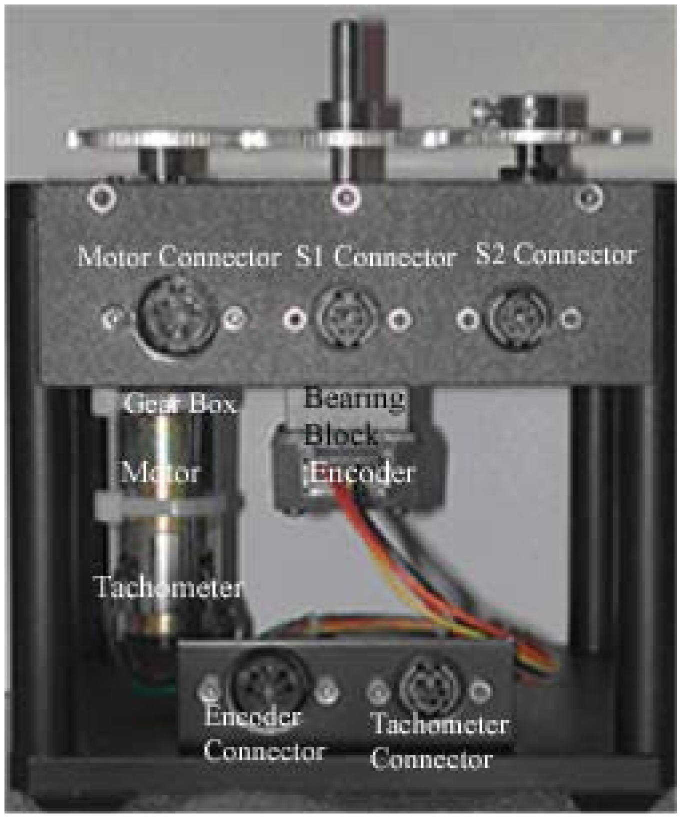
EE-371 CONTROL SYSTEMS LABORATORY

Session 5
Servo Position Control Design Project
(Position and Rate Feedbacks)

Purpose

Position control systems are used extensively in industrial applications such as robotics and drive control. Modern position control systems are achieved using incremental encoder sensors. In this project you design and implement a position control system for low frequency square wave input. The objectives of this project are:

- To obtain the servo plant model
- To design a position control system such that the output angle tracks a commanded position using position and velocity feedback and determine the feedback gains to achieve the given time-domain specifications.
- Build the compensated servo plant in SIMULINK and simulate offline to obtain the response to a square wave input and verify the design.
- Build the WinCon application, implement and test the system on the real-time hardware



Introduction

The position control system is a system that converts a position input command to a position output response.

A schematic layout of the servomotor is shown in Figure 5.1

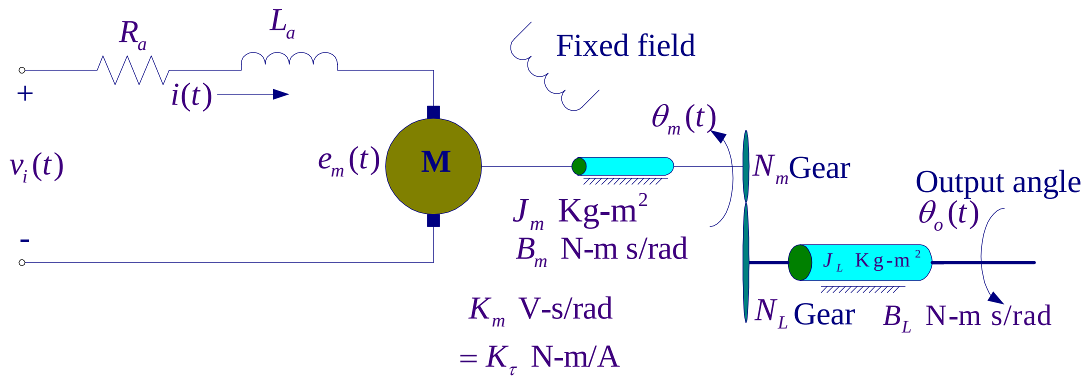


Figure 5.1 Servo plant schematic

The first step in the design of a control system is the mathematical modeling of the physical system as follows

1. Servo plant Modeling

The system parameters are as follows:

Armature resistance, $R_a = 2.6\ \Omega$

Armature inductance, $L_a = 0.18$ mH (neglected)

Motor voltage constant, $K_m = 0.00767$ V-s/rad

Motor torque constant, $K_\tau = 0.00767$ N-m/A

Armature inertia, $J_m = 3.87 \times 10^{-7}$ Kg m^2

Tachometer inertia, $J_{tach} = 0.7 \times 10^{-7}$ Kg m^2

Low gear ratio, $K_g = \dfrac{N_L}{N_m} = 14$

Gear inertia, $J_{72} = 5.4435 \times 10^{-6}$ Kg m^2

Load inertia, $J_L = 3(J_{72}) = 16.333 \times 10^{-6}$ Kg m^2

Armature frictional coefficient, B_m

Load frictional coefficient, B_L

Equivalent friction referred to the secondary gear $B_{eq} = K_g^2 B_m + B_L \simeq 1 \times 10^{-3}$ Nm/(rad/s)

Motor efficiency due to rotational loss $\eta_{mr} \simeq 0.87$

Gearbox efficiency, $\eta_{gb} \simeq 0.85$

From the s-domain motor electrical circuit shown in Figure 5.2, we have

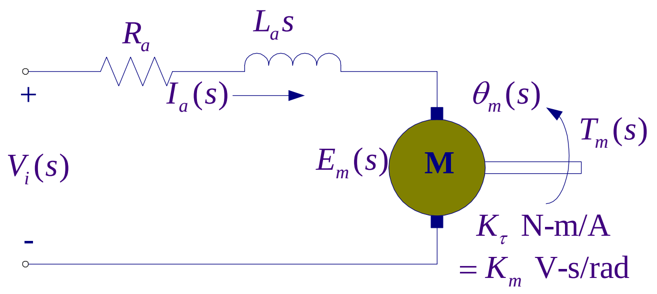


Figure 5.2 The s-domain motor electric circuit

$$I_a = \frac{1}{R_a + L_a s}\left[V_i(s) - E_m(s)\right] \tag{5.1}$$

For a separately excited dc motor with constant field current or a permanent magnet dc motor, the armature produces a torque, which is proportional to the armature current, i.e., $K_\tau i_a(t)$. The gearbox efficiency as well as motor efficiency due to rotational loss would affect the output torque. The gearbox efficiency is not constant, an average value of $\eta_{gb} = 0.85$ is assumed.

$$T_m(t) = \eta K_\tau i_a(t)$$

Where $\eta = \eta_{mr}\eta_{gb} = (0.87)(0.85) = 0.7395$ and K_τ is the motor torque constant in N-m/A, which is numerically equal to K_m. Thus, in s-domain we have

$$T_m(s) = \eta K_m I_a(s) \tag{5.2}$$

The motor emf is proportional to the angular velocity and the field current. With constant field, the motor emf is given by $e_m(t) = K_m \omega_m(t)$ or in s-domain

$$E_m(s) = K_m \Omega_m(s) \tag{5.3}$$

Figure 5.3 shows the s-domain electric circuit analogy for the mechanical portion.

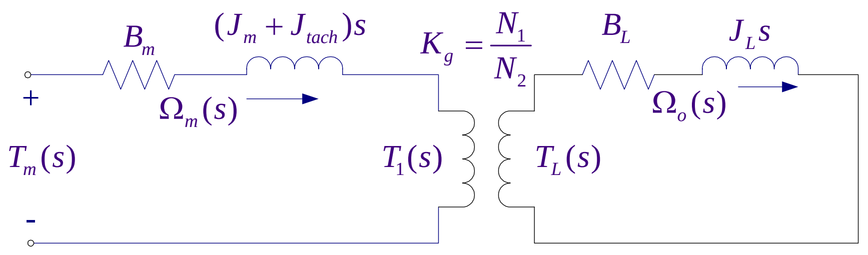
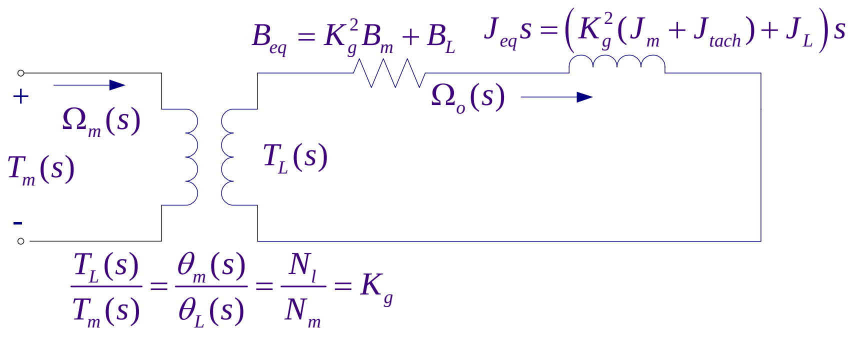


Figure 5.3 The s-domain electric circuit analogy for the mechanical part.

Figure 5.4 shows the circuit analogy with armature frictional coefficient and inertia referred to the secondary gear and combined with the load friction and inertia.



$B_{eq} = K_g^2 B_m + B_L$
$J_{eq}s = \left(K_g^2(J_m + J_{tach}) + J_L\right)s$
$+$
$\Omega_m(s)$
$T_m(s)$
$-$
$\Omega_o(s)$
$T_L(s)$
$\frac{T_L(s)}{T_m(s)} = \frac{\theta_m(s)}{\theta_L(s)} = \frac{N_l}{N_m} = K_g$

Figure 5.4 Motor frictional coefficient and inertia referred to the load side.

From Figure 5.4 we have

$$\Omega_o(s) = \frac{1}{B_{eq} + J_{eq}s} T_L(s) \tag{5.4}$$

$$\frac{\Omega_m(s)}{\Omega_o(s)} = K_g \tag{5.5}$$

The s-domain output angle is

$$\theta_o = \frac{1}{s}\Omega_o(s) \tag{5.6}$$

2. Pre-laboratory Assignment

From equations (5.1)-(5.6) draw a detail block diagram representation for the above servo system with $V_i(s)$ as input and $\theta_o(s)$ as output showing all the variables $I_a(s)$, $T_m(s)$, $T_L(s)$, $\Omega_o(s)$, and $\theta_o(s)$. Label as **Figure 5.5** Servo plant block diagram

Armature circuit time constant $\tau_a = \frac{L_a}{R_a}$ is small compared to the mechanical time constant. Thus, neglect the armature inductance, obtain the closed loop transfer function and show that

$$\frac{\Omega_o(s)}{V_i(s)} = \frac{\dfrac{\eta K_m K_g}{R_a J_{eq}}}{s + \dfrac{B_{eq}}{J_{eq}} + \dfrac{\eta K_m^2 K_g^2}{R_a J_{eq}}} \tag{5.7}$$

$$\frac{\Omega_o(s)}{V_i(s)} = \frac{a_m}{s + b_m} \tag{5.8}$$

Where

$$a_m = \frac{\eta K_m K_g}{R_a J_{eq}}, \qquad b_m = \frac{B_{eq}}{J_{eq}} + \frac{\eta K_m^2 K_g^2}{R_a J_{eq}} \tag{5.9}$$

Or the servo plant transfer function with the output as position $\theta_o(s)$ is

Or

$$\frac{\theta_o(s)}{V_i(s)} = \frac{a_m}{s(s+b_m)} \tag{5.10}$$

Where $J_{eq} = K_g^2(J_m + J_{tach}) + J_L = (14)^2 \times (3.87 + 0.70) \times 10^{-7} + 16.33 \times 10^{-6} = 10.59 \times 10^{-5}$ Kg m^2

Substitute for the parameters, evaluate a_m, and b_m, and find the servo plant transfer function in terms of the numerical values.

$$\frac{\theta_o(s)}{V_i(s)} =$$

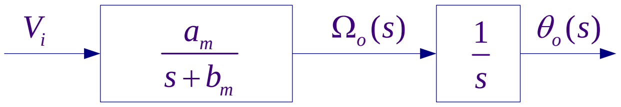


Figure 5.6 Servomotor transfer function model

The s-domain unit step response is

$$\theta_o(s) = \frac{a_m}{s\left(s+b_m\right)}\frac{1}{s}$$

The final value of the response is $\lim_{t\to\infty}\theta_o(t) = \lim_{s\to 0} s\theta_o(s) = \infty$. That is, the response is unbounded.

2.1 Position control with position and rate feedback

In order to control the output position to follow an input command, consider the addition of a position feedback and a rate feedback given by

$$V_i(s) = K_P[\theta_i(s) - \theta_o(s)] - K_D\Omega_o(s) \tag{5.11}$$

as shown in Figure 5.7. The purpose of this system is to have the output angle $\theta_o(t)$ follow the input angle $\theta_i(t)$.

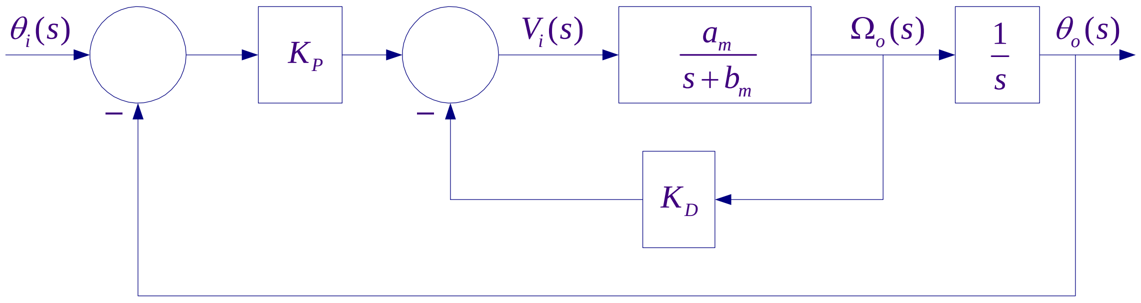


Figure 5.7 Position Control using position and rate feedback.

Applying Mason's gain formula and show that the overall transfer function is

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_P a_m}{s^2 + \left(K_D a_m + b_m\right)s + K_P a_m} \tag{5.12}$$

Substitute for a_m, and b_m.

2.2 Position control design

Let $\theta_i(t)$ be a square wave of amplitude 20° and a frequency of 1 Hz. Design a control system and determine the gains K_P and K_D such that the following time-domain specifications are met:

- Step response damping ratio of $\zeta = 0.707$
- Peak time of $t_p = 0.05$ second.

The second-order response peak time t_p, is given by

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{5.13}$$

and the theoretical peak value of the step response is

$$M_{pt} = \left(1 + e^{-\zeta\pi/\sqrt{1-\zeta^2}}\right)\theta_{i(\text{Amplitude})} \tag{5.14}$$

For $\zeta = 0.707$, we find ω_n.

The servo motor transfer function has the same form as the standard second-order transfer function

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.15}$$

Comparing the plant characteristic equation given in (5.9) with the standard second-order characteristic equation in (5.15), find two equations for K_P and K_D in terms of a_m, b_m, ζ, and ω_n. Substitute for the parameters and obtain the values of K_P, and K_D for the above design specifications.

2.3 Digital Simulation

Construct a SIMULINK diagram similar to the one shown in Figure 5.8 and save it (say Lab5_Sim.mdl). Use a signal generator with amplitude 20 degrees and frequency 1 Hz. Replace am, bm, KP and KD with their values. Alternatively, you can place all equations in a script m-file to compute $am, bm, KP,$ and KD which is sent to the MATLAB Workspace and then simulate the Simulink diagram. From the Simulink/Simulation Parameters select the Solver page and for Solver option Type select Fixed-step and ode4 (Runge-Kutta) and set the fixed step size to 0.001. Actually a Variable-step selection with auto or a smaller Max step size would produce more accurate results, but because in the implementation diagram we are using a uniform sampling rate of 0.001 second, we use the same fixed-step size of 0.001 second for integration.

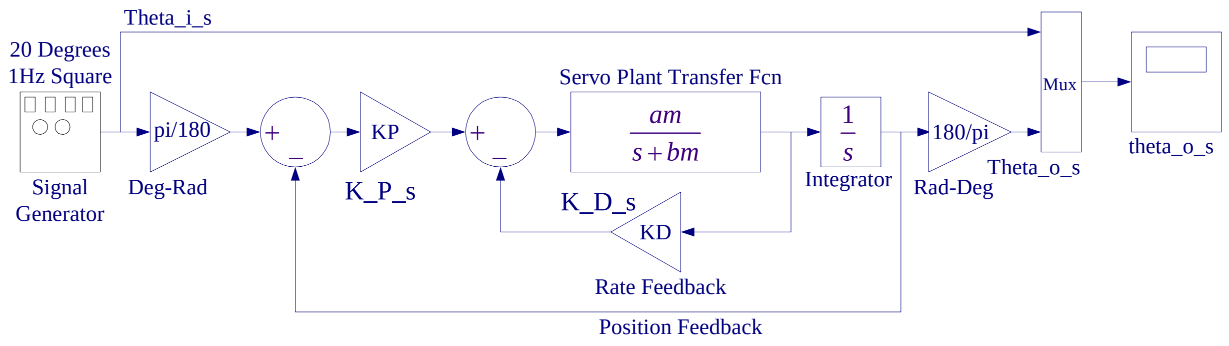


Figure 5.8 Simulink diagram for the servo plant position control.

Simulate and use ‘plotscope’ function to capture the scope plot and produce a Figure plot. To do this, type **plotscope** at the MATLAB prompt, then click on the Scope Figure (outside the plot area) and hit return you will have a Figure print. You can add label and legend commands or edit the graph, label as Figure 5.9. Check to see if the response meets the design requirements within the simulation numerical integration accuracy.

All the pre-lab calculations, design and simulation must be completed prior to the laboratory session. The plants transfer function and the controller values must be checked and verified by your instructor. Also complete the implementation diagram as outlined in section 3.1 and 3.2.

3. Laboratory Procedure

When you have finished testing your model in SIMULINK, it has to be prepared for implementation on the real-time hardware. This means the plant model has to be replaced by the I/O components that form the interfaces to the real plant.

3.1 Creating the Subsystem Block for the Interface to SRV02

The encoder is used to provide the digital phase information. This signal can also be differentiated to produce a velocity signal. This can be achieved using a low-pass band-limited differentiator. The suitable transfer function used is $\frac{250s}{s+250}$. First the tachometer and encoder interface is created. From the Simulink Library Browser window File menu open a new model. Place an analog input from the Quanser MultiQ3 library on this new window. The tachometer gain is 1.5 V per 1000 RPM. Get a gain block to convert volt to RPM, and another gain block to convert RPM to rad/s, one gain block for the gear ratio. A low pass filter may also be utilized to block the high frequency noise. The tacho input is connected to analog input 2. Double-click on the Tacho input block to open its dialog box and set the Channel Use to 2. Connect the blocks as shown in Figure 5.10. Now get the part Encoder Input; place it on the lower portion of the page. Double-click on the Encoder input block to open its dialog box and set the Channel Use to 0. Since the encoder has 4096 signal periods per revolution, get a gain block and set its value to $-2*pi/4096$. A negative sign is used for the encoder gain, because the negative feedback gain is already implemented in the encoder wiring (i.e. positive voltage => negative counts) and in the Simulink model we have used a negative feedback loop. Acquire the remaining blocks and complete the diagram and label all the blocks properly as shown in Figure 5.10. A subsystem model can be created that can be used in other applications.

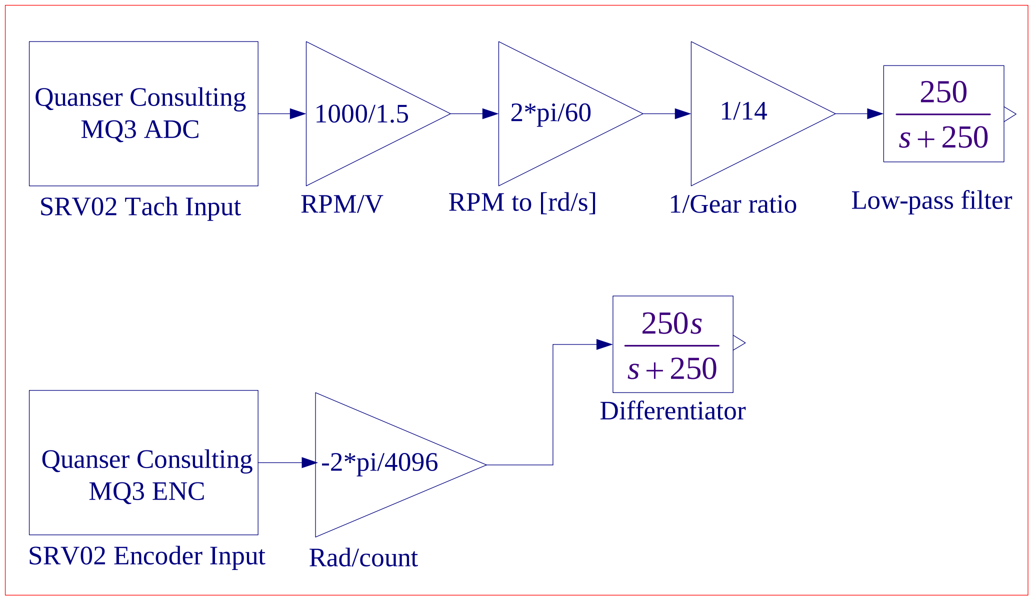


Figure 5.10 Interfaces to the SRV02 Feedback Signals.

To create a subsystem, enclose the blocks within a bounding box as shown in Figure 5.10 (Do not place any outport blocks). Choose **Create Subsystem** from the **Edit menu**. Simulink replaces the selected blocks with a subsystem block as shown in Figure 5.11

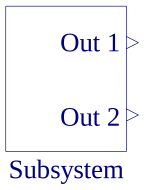


Figure 5.11 Subsystem block for the interface to the SRV02

Double-click to open the subsystem block. Notice that the Simulink automatically adds two Outport blocks Out 1 and Out 2. We need to add another port for position. From the Sink Library get an Outport block and connect it to the output of the Rad/Count gain block. You now have the Out 3 port. Label the Outports appropriately. Rename the title subsystem to Encoder & Tach input, and save it as Encoder_tach.mdl. You know have the following subsystem as shown in Figure 5.12.

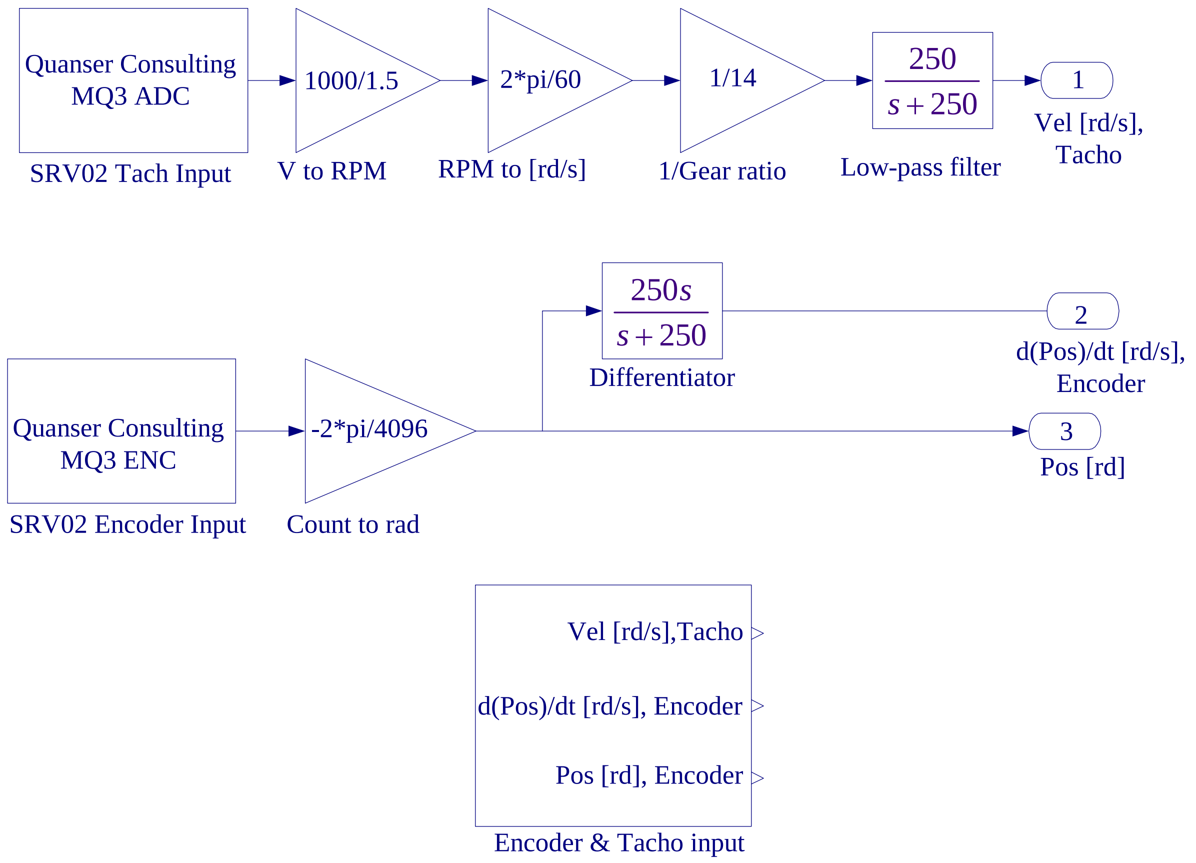


Figure 5.12 Interfaces to the SRV02 Feedback Signals.

3.2 Creating the Implementation model

The implementation model can be added on the previously constructed SIMULINK model. This would enable you to obtain the simulation and actual results simultaneously. Open Lab5_Sim.mdl (your simulation model), save it under a new name (say Lab5_Imp.mdl). Remove the MUX block.

Start constructing the implementation diagram below the simulation diagram. Copy the Encoder_tach.mdl (constructed in part 3.1) to the clipboard and paste it on your implementation model as shown in Figure 5.13. Get the Analog Output block from the Quanser MultiQ3 library and set the Channel Use to 0. Use a Signal Generator with Amplitude 20 degrees, and Frequency 1 Hz, and use a gain block to convert to radians. Add the position and velocity feedback gains to the signal coming from the Motor Encoder complete the feedback loops and connect the resulting signal to the Quanser Analog output. Place as many Scopes as you like to monitor the phase angle, velocity etc. Your completed model should be the same as shown in Figure 5.13. Set the gains KP and KD to the values found in part (2.2), or run the m-file that returns the values of KP and KD. The gains in Simulink Simulation diagram are renamed to KP1 and KP1, so that if the value of the variables KP and KD are changed at the MATLAB prompt for fine tuning, the values in the Simulation diagram are not changed.

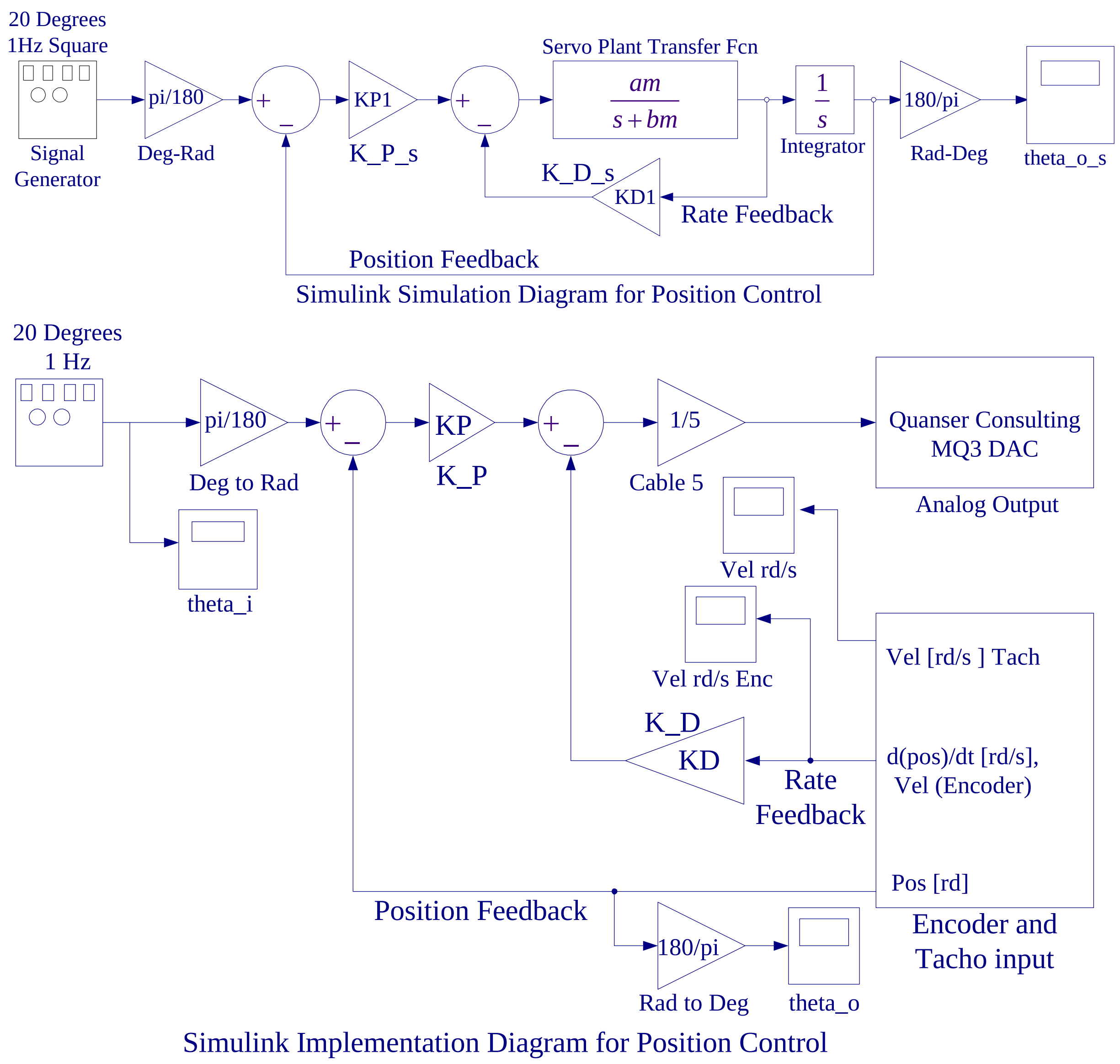


Figure 5.13 Simulation and Implementation diagram for position control.

3.3 Wiring diagram

Using the set of leads, universal power module (UPM), SRV-02 DC-motor, and the connecting board of the MultiQ3 data acquisition board, complete the wiring diagram shown in Figure 5.14 as follows:

From	**To**	**Cable**
Encoder on SRV02	MultiQ/Encoder 0	5 pin Din to 5 pin Din
Motor on SRV02	UPM/To Load	6 pin Din to 4 pin Din, **Gain 5 Cable**
D/A #0 on MultiQ	UPM – From D/A	RCA to 5 Pin Din
A/D # 0, 1, 2, 3, on MultiQ	UPM- TO A/D	5 pin Din to 4xRCA

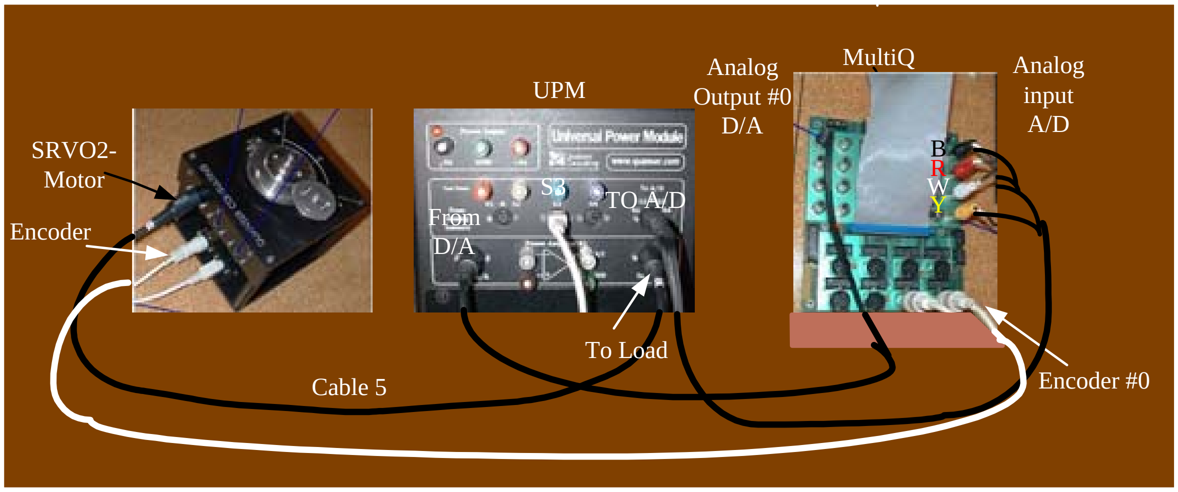


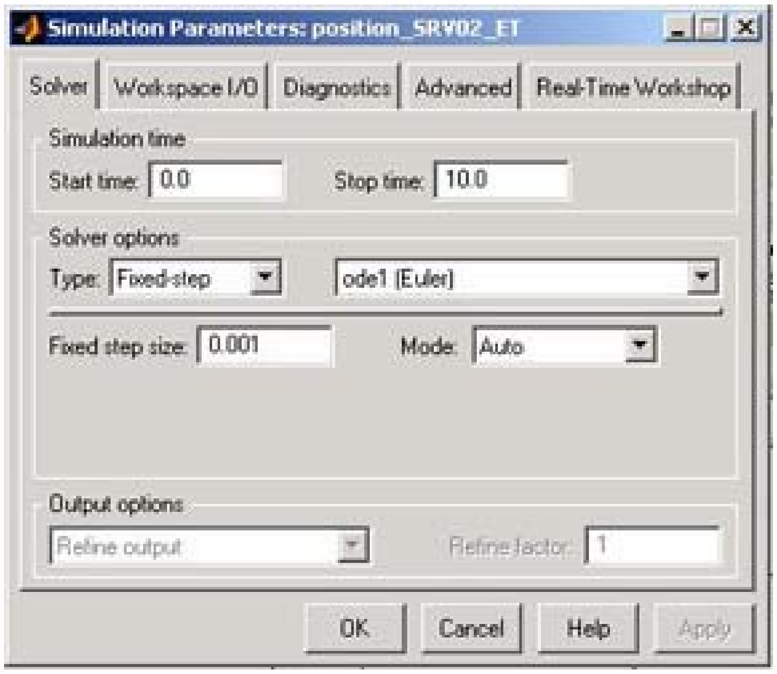
Figure 5.14 Wiring diagram for servo motor position control system.

Before proceeding to the next part request the instructor to check your electrical connections and the implementation diagram.

3.4 Compiling the model

In order to run the implementation model in real-time, you must first build the code for it. Turn on the UPM. Start WinCon, Click on the MATLAB icon in WinCon server. This launches MATLAB. In the Command menu set the Current Directory to the path where your model Lab5_Imp.mdl is. Before building the model, you must set the simulation parameters. Pull down the Simulation dialog box and select Parameters. Set the Start time to 0, the Stop time to 5, for Solver Option use Fixed-step and ode4 (Runge-Kutta) method set the Fixed Step size, i.e., the sampling rate to 0.001 as shown in Figure 5.15.

In the Simulation drop down menu set the model to **External**. Make sure all the controller gains are set. Start the WinCon **Server** on your laptop and then use **Client Connect**, in the dialog box type the proper Client workstation IP address. Generate the real-time code corresponding to your diagram by selecting the "**Build**" option of the **WinCon** menu from the Simulink window. The MATLAB window displays the progress of the code generation task. Wait until the compilation is complete. The following message then appears: "*### Successful completion of Real-Time Workshop builds procedure for model: Lab5_Imp*".



Simulation Parameters: position_SRV02_ET

Solver | Workspace I/O | Diagnostics | Advanced | Real-Time Workshop

Simulation time

Start time: 0.0 Stop time: 10.0

Solver options

Type: Fixed-step ode1 (Euler)

Fixed step size: 0.001 Mode: Auto

Output options

Refine output Refine factor: 1

OK Cancel Help Apply

Figure 5.15 Simulation Parameters

3.5 Running the code
Following the code generation, WinCon Server and WinCon Client are automatically started. The generated code is automatically downloaded to the Client and the system is ready to run. To start the controller to run in real-time, click on the **Start** icon from the WinCon Server window shown in Figure 5.16. It will turn red and display STOP. Clicking on the **Stop** icon will stop the real-time code and return to the green button.

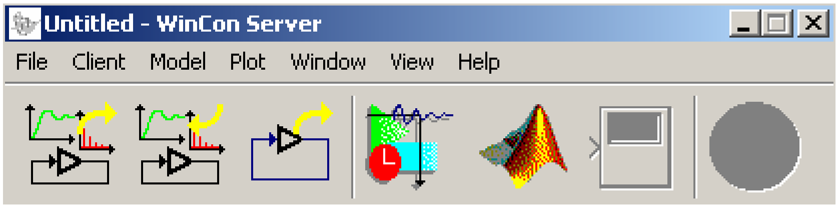
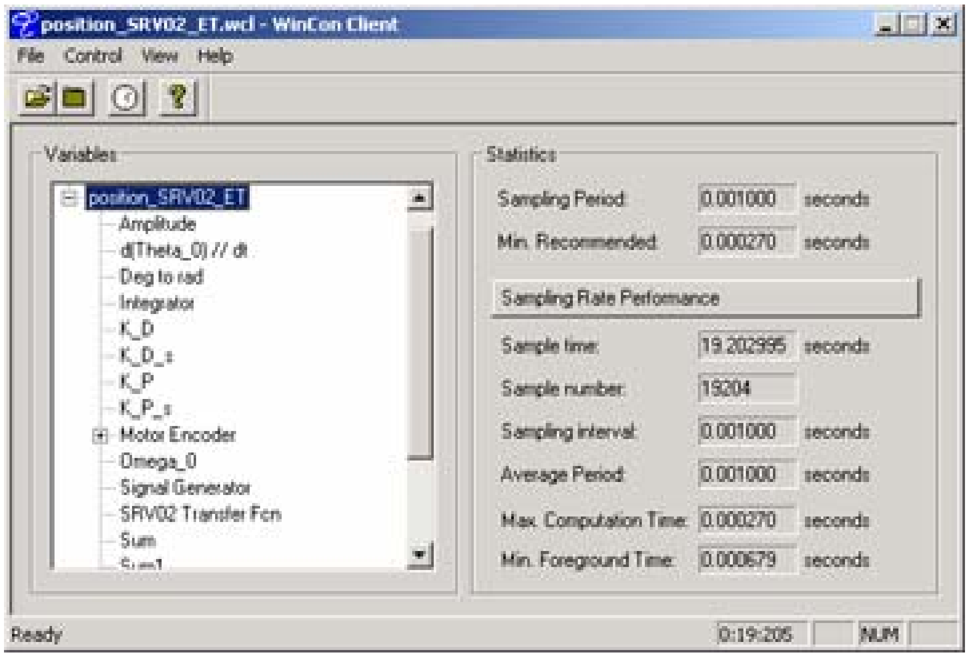


Figure 5.16 WinCon Server

If you hear a whining or buzzing in the motor you are feeding high frequency noise to the motor or motor is subjected to excessive voltage, immediately stop the motor. Ask the instructor to check the implementation diagram and the compensator gains before proceeding again.

If you have access to the host PC you can maximize the WinCon Client icon to show a window similar to the one pictured in Figure 5.17. The WinCon Client is the real-time component of the software and it runs at the period (i.e. sampling rate) specified under **WinCon/Options…/Solver/Fixed step size** from the Simulink window menu, as illustrated in Figure 5.15. In this case, a sampling period of 1 ms was chosen.



position_SRV02_ET
- Amplitude
- d(Theta_0) // dt
- Deg to rad
- Integrator
- K_D
- K_D_s
- K_P
- K_P_s
- Motor Encoder
- Omega_0
- Signal Generator
- SRV02 Transfer Fcn
- Sum
- Sum1

Sampling Period	0.001000	seconds
Min. Recommended	0.000270	seconds
Sample time	19.202995	seconds
Sample number	19204	
Sampling interval	0.001000	seconds
Average Period	0.001000	seconds
Max. Computation Time	0.000270	seconds
Min. Foreground Time	0.000679	seconds

Ready 0:19:205 NUM

Figure 5.17 WinCon Client

You can also change the controller gains on the fly (i.e. while the controller is running in real-time). To do so, double click on the Simulink gain block, change to the desired new value, and select Apply, or OK. Note the changes in the real-time plots.

3.6 Plotting Data

You can now plot in real-time any variables (e.g. angles, velocities) of your diagram by clicking on the "**Plot\New\Scope**" button in the WinCon Server window and selecting the variable you wish to visualize. Select "Theta_o" and click OK. This opens one real-time plot. To plot more variables in that same window, click on "**File/Variables…**" from the Scope window menu. The names of all blocks in the Simulink model diagram appear in a Multiple Select Variable Tree. You can then select the variable(s) you want to plot. In this case, select, for example, "Theta_i" and "Theta_o_s. In the Scope pull-down menu, using Freeze you can freeze the plot, and Update/Buffer can be used to change the final time to display fewer cycles. From the File menu you can Save and Print the graph. Choose Save As M-File, and save the plot as M-file (say Figure5_18.m). Now at the MATLAB prompt type the file name to obtain the MATLAB Figure plot. You can type "grid" to place a grid on the graph or edit the Figure as you wish.

Zoom in to obtain a better comparison between the simulated response and the actual response.

If you are sufficiently happy with your results and the actual response is close to the simulated response, you can move on and begin the report for this project. Remember there is no such a thing as a perfect model, and your calculated parameters were based on the plant model. A control design usually involve some form of fine-tuning, and will more than likely be an iterative process. If the actual response deviates from the desired response by large values you can fine tune K_P and K_D (only in the implementation model) around the calculated value to get a response to meet the design specifications more closely.

Use File Save, this saves the compiled controller including all plots as a **.wpc** (WinCon project) file. In case you want to run the experiment again, from WinCon Server use File/Open to reload this **.wcp** file, and run the project in real time independent of MATLAB/Simulink.

To prevent excessive wear to the motor and gearbox run the experiment for a short time.

4. Project Report

Discuss the assumption and approximations made in the modeling the servomotor. Describe the effect of adding the velocity feedback and describe your design. In the report show the closed-loop system block diagram and your analysis. Comment on your results; how does experimental response compare to simulated response? Calculate the peak value of the compensated closed-loop system from (5.14). Zoom in and estimate the peak value and the peak time for the experimental response and compare with the specified values. Discuss the reason for any deviation in the actual transient response and the simulated response. What is the system type, and what is the theoretical steady-state error? Estimate the actual steady-state error if any, and discuss the reason for the steady-state error. A brief discussion of frictional forces and the amplifier saturation is included in the Appendix, which you may find helpful for preparing the discussion of results in your project report. After the completion of this lab you should be confident in tuning this type of controller to achieve a desired response. Do you feel this controller can meet any arbitrary system requirements? Explain.

Appendix

The essential requirement in a position control system is for a motor to rotate an output shaft to the same angle as an input shaft. In designing a servomechanism particular attention should be given to both static and transient behavior. The sampling rate in the data acquisition board is high enough that permits continuous system analysis and design techniques.

The static characteristics are the steady-state error, repeatability, resolution and stiffness. Repeatability is the range in which the servo will come to rest whenever a given input signal is repeated. Resolution is the smallest discrete motion, which the servo can make. The steady-state error is primarily limited to the accuracy of the feedback sensors.

The open-loop plan transfer function as given by (5.10) is a type 1 system and the theoretical steady-state error to a step input is zero. However, the actual response will have a steady state error. This is due to the static friction. Because of the static friction the gears will not turn until a certain amount of torque has exceeded. This is known as the deadband, and is defined, as the minimum voltage required for getting a system to respond. The SRV02 servomotor has a motor deadband about 0.1volts. All of the static characteristics are improved with increased proportional gain.

The frictional torque between the gearing mesh depends on many factors, and an accurate mathematical description of frictional torque is difficult. The angular rotation is influenced by three type of friction. These are viscous friction, static friction and Coulomb friction. Viscous friction is function of velocity. In the mathematical model derived we assumed viscous friction, with equivalent frictional coefficient of B_{eq}. Static friction represents a retarding torque that tends to prevent motion from beginning. Coulomb friction has amplitude that reverses with the reversal of the direction of velocity. The frictional torque is largely due to Coulomb and static rather than viscous friction and is nonlinear.

The transient characteristics are relative stability, maximum overshoot and the speed of the response. A satisfactory transient response can be attained by specifying: the step response damping ratio, and a specification for the speed of the response such as rise time, time constant or peak time.

Another factor, which influences the transient response, is the amplifier saturation in the D/A converter. The voltage applied to the motor is proportional to the error signal. For a large step input the initial error signal $e(t)$ is also very large, resulting in a very large instantaneous voltage given by $K_P e(t)$. If the voltage is beyond the saturation limit of the amplifier, then the amplifier will not put out the expected voltage and the transient response will not be as expected and will result in larger overshoot. The step change in radians for the amplifier to operate in its linear range is given by $2\theta_i < \frac{V_{sat}}{K_P}$, where θ_i is the square wave amplitude. The saturation for this amplifier starts at about 5 volts. If $K_P = 27$, $2\theta_i \le \frac{5}{27}$, i.e., $2\theta_i = 0.185$ rad or 10.6°, or an amplitude of 5.3°. When the step change is larger a gain cable can be used. Quanser supplies gain =1, 5 cables for the amplifier. If a gain = 5 cable is used the step change can be as high as 53° without saturating the amplifier.

The servomechanism can never be more accurate than its instrumentation. The gain feedbacks K_P and K_D are deigned based on the derived model. This should give a reasonably accurate result as a first pass. In the implementation diagram, these gains can be tuned to obtain the desired response. It must also be noted that the numerical integration set at fixed step size 0.001 and ode4 (Rung e-Kutta) will produce error in the simulation response.