

MEM01: DC-Motor Servomechanism

Interdisciplinary Automatic Controls Laboratory - ME/ECE/CHE 389

February 5, 2016

Contents

1	Introduction and Goals	1
2	Description	2
3	Modeling	2
4	Lab Objective	5
5	Model Identification	5
5.1	Model Identification: Open-loop Step Response Analysis	5
5.2	Model Identification: Open-loop Frequency Response Analysis	7
6	Velocity Controller Design	8
7	Position Controller Design	8

1 Introduction and Goals

The Precision Modular Servo (PMS) workshop serves as a model of a very popular device, the DC Servo Motor. This is often used in robotic applications. The word ‘servo’ comes from the Latin word servus meaning servant or slave. Thus the servo motor is intended to react to a given command, for example a desired position or velocity. In order for the motor to be called a servo motor it has to be equipped with a velocity and position measurement unit and motor driver. It may also include a simple servo motor controller. The main features distinguishing servo motors from normal motors are:

- minimized rotor inertia,
- minimized armature inductance,
- improved withstand voltage and magnet saturation,
- designed to withstand sudden acceleration and high frequency operation.

The PMS unit allows for the design of different servo motor controllers and to test them in real time using the Matlab and Simulink environment.

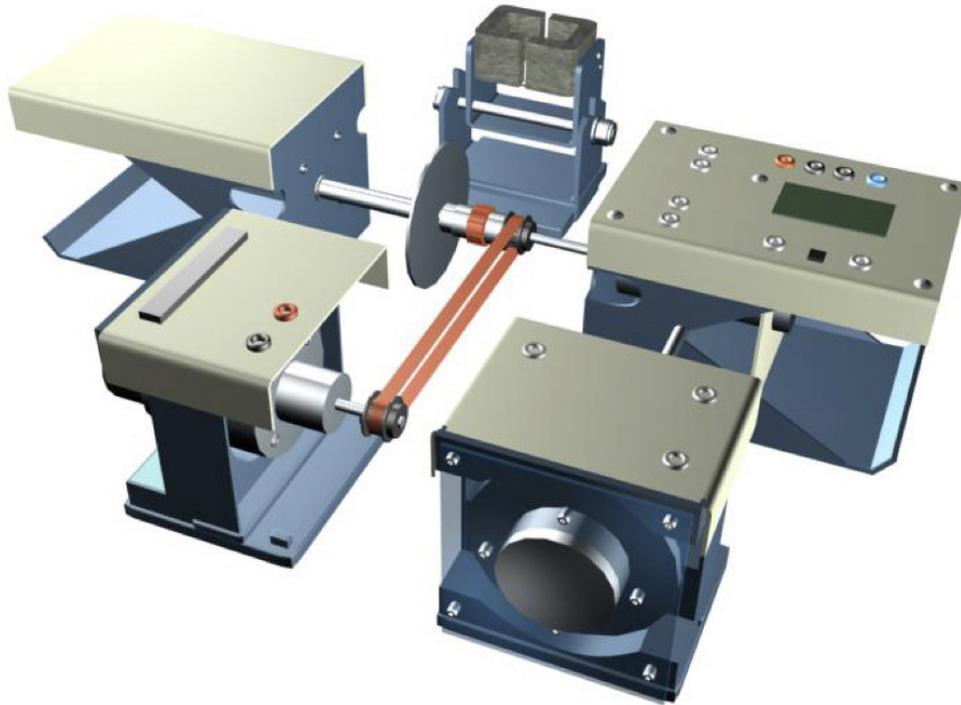


Figure 1: Precision Modular Servo mechanical unit

2 Description

The description of the PMS setup in this section refers mostly to the mechanical part and the control aspect. For details of the mechanical and electrical connection, the interface and an explanation of how the signals are measured and transferred to the PC, refer to the “Installation & Commissioning” manual. As shown in Figure 1, the PMS mechanical unit consists of a DC Motor, Tachometer/Gearbox, Digital Encoder, Input and Output Potentiometers and Magnetic Brake.

Apart from the mechanical units, the electrical units play an important role for motor control. They allow measured signals to be transferred to the PC via an I/O card. The amplifiers are used to transfer control signals from the PC to the DC Motor. The mechanical and electrical units provide a complete control system setup presented in Figure 2.

In order to design any control algorithms one must first understand the physical background behind the process and carry out identification experiments. The next section explains the modelling process of the PMS.

3 Modeling

Every control project starts with plant modeling, so as much information as possible is given about the process itself. The mechanical-electrical model of the servo motor is presented in Figure 3. For the DC motor model, nonlinearities are so small that they can be neglected. According to the electrical-mechanical diagram presented in Figure 3 the linear model equations can be derived.

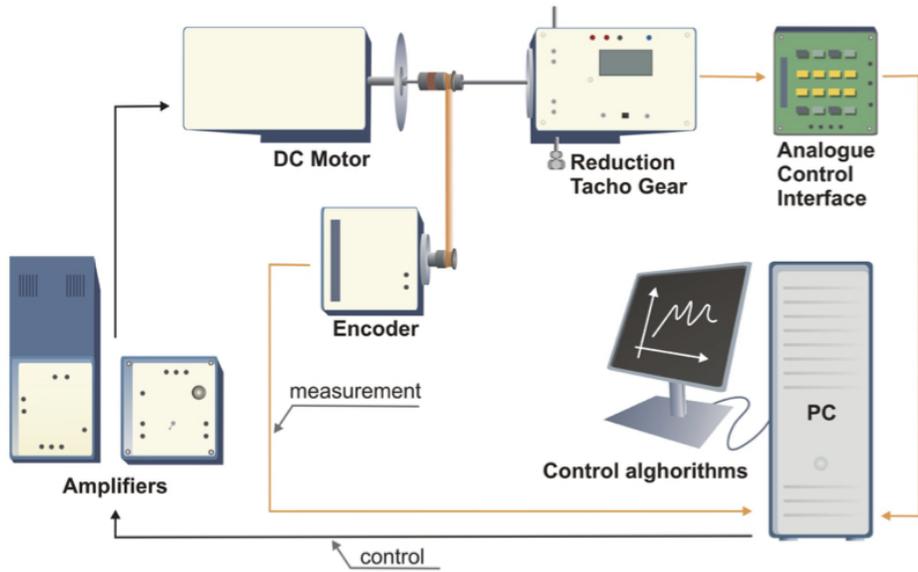


Figure 2: PMS control system

The torque τ generated by the DC motor is proportional to the motor current i :

$$\tau = K_t i, \quad (1)$$

where K_t is the torque constant. The induced electromotive force, v_{emf} , is a voltage proportional to the angular shaft velocity $\dot{\theta}$:

$$v_{emf} = K_b \dot{\theta}, \quad (2)$$

where K_b is a motor property dependant constant related to physical properties of the motor. Using Newtons law the mechanical dynamics can be described by:

$$J\ddot{\theta} = \sum \tau_i = -d\dot{\theta} + K_t i, \quad (3)$$

where d is a linear approximation of the viscous friction. The equations governing the behaviour of the electrical model are:

$$u - v_{emf} = L \frac{di}{dt} + Ri, \quad (4)$$

where u is the control signal. Including equation (2) transforms (4) into:

$$u = L \frac{di}{dt} + Ri + K_b \dot{\theta}. \quad (5)$$

Q1: Use equations (3) and (5) to write the motor model in state space form by defining the state vector $x = [\omega \ i]^T$, where $\omega = \dot{\theta}$ represents the angular velocity, and the output as $y = \omega$.

Q2: Give an expression for the transfer function $G(s) = Y(s)/U(s)$. What is the order of the transfer function? What are the units of the input and the output?

Table 1 gives the values of the motor parameters supplied by the manufacturer. The motor used for the experiments is a 24V DC brushed motor with a no-load speed of 4050 rpm. An extra gain term $k_a = 24/2.5 = 9.6$ needs to be added to input of the model to represent the gain provided by the PA150 Preamplifier and SA150 Servo Amplifier. In this way, the model can be run with a 2.5 V control signal

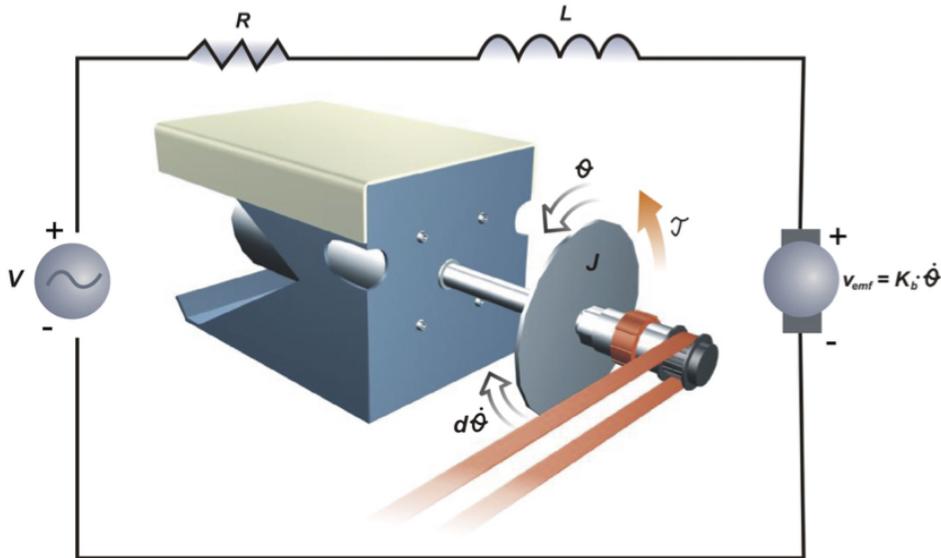


Figure 3: DC motor phenomenological model

Parameter	Description	Value
J	Moment of inertia	$140 \times 10^{-7} \text{ kg} \cdot \text{m}^2$
K_t	Torque constant	0.052 Nm/A
K_b	Electromotive force constant	0.057 Vs/rad
d	Viscous friction coefficient	10^{-6} Nms/rad
R	Resistance	2.4Ω
L	Inductance	$2.5 \times 10^{-3} \text{ H}$

Table 1: DC Motor parameters

amplitude (as used by Simulink to drive the DAC output), which corresponds to the 24 V amplitude signal applied to the motor. In other words, the input of the model is the output of the DAC and not the motor voltage. Moreover, an extra gain term $k_\omega = 60/(2\pi)$ needs to be added to the output of the model to change the output units to rpm.

Q3: How would you modify $G(s)$ to incorporate k_a and k_ω ? Compute the transfer function using the parameter values in Table 1. Where are the poles of the transfer function?

Q4: Obtain a first-order approximation of the obtained transfer function. Explain why this is possible. What is the DC gain and time constant of the model? Plot in the same figure the step response for both actual and approximated transfer functions when the amplitude of the step is 2.5V. Does the result justify the approximation?

Q5: Obtain the experimental step response of the system when the amplitude of the step is 2.5V. Plot in the same figure the obtained experimental step response and that predicted by the approximated first-order transfer function. How well the predictions by the model match the experimental data?

4 Lab Objective

In this laboratory you will obtain an experimentally validated model for the DC motor-based servomechanism and design model-based controllers to regulate both the speed and the position of the servo. A plant transfer function, i.e. the open loop response of the system, will be obtained from experimental data. Identification of model parameters from experimental data is known as system identification.

5 Model Identification

In this section, you will obtain the plant transfer function (open loop system) by two separate system identification methods: i) analysis of the open loop step response and ii) analysis of the open loop frequency response.

5.1 Model Identification: Open-loop Step Response Analysis

Assume the plant $G(s)$ is of the form

$$G(s) = \frac{k}{\tau s + 1}, \quad (6)$$

i.e. a first order linear system with a stable (LHP) pole. The objective of system identification is to determine the steady state gain k and the time constant τ from experimental data. If the system, $G(s)$, is excited by a step input of magnitude A , the response will be

$$U(s) = \frac{A}{s}, \quad Y(s) = G(s)U(s) = \frac{Ak}{s(\tau s + 1)}. \quad (7)$$

Then, the steady-state value y_{ss} of $y(t)$ can be computed by the final value theorem, i.e.,

$$y_{ss} = \lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s) = Ak, \quad (8)$$

and the steady-state gain of the system can be obtained as

$$k = \frac{y_{ss}}{A}. \quad (9)$$

Note that y_{ss} can be obtained directly from the experimentally step response of the system as shown in Figure 4 (left).

By computing the inverse Laplace transform of $Y(s)$ in (7), we can write

$$y(t) = Ak(1 - e^{-t/\tau}) = y_{ss}(1 - e^{-t/\tau}). \quad (10)$$

Therefore,

$$\frac{y(t) - y_{ss}}{y_{ss}} = -e^{-t/\tau} \iff \ln\left(\frac{y_{ss} - y(t)}{y_{ss}}\right) = -\frac{t}{\tau}, \quad (11)$$

which implies that if we plot $\ln((y_{ss} - y(t))/y_{ss})$ versus t as shown in Figure 4 (right), we can obtain $-1/\tau$, and therefore τ , just by computing the slope of the plotted line.

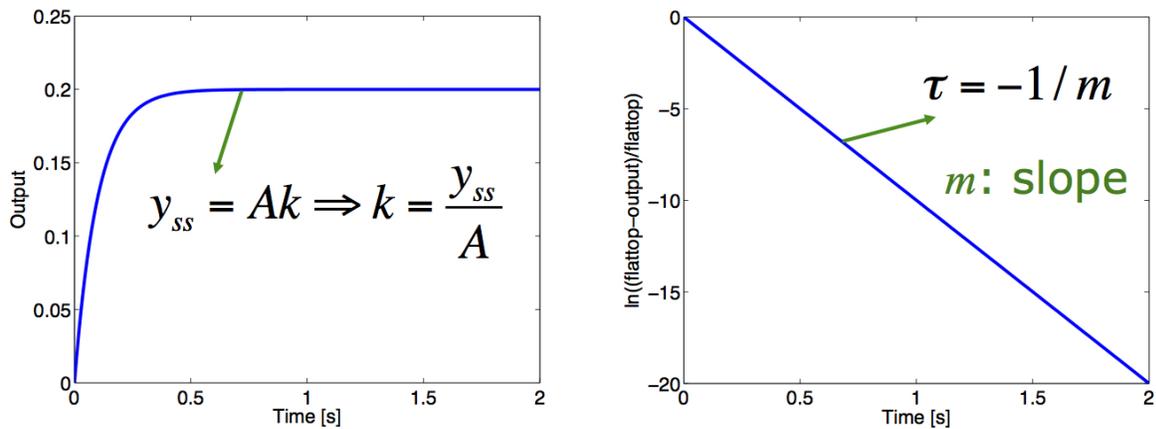


Figure 4: System identification by step response.

Open Loop Step Response: Lab Procedure

NOTE: There should be no need to adjust the hardware

1. Open up matlab and point the matlab path to
 - C:\Program Data\Fedback Instruments\33-927 Modular Servo
2. Run the matlab file “startup”
 - the “startup” function adds relevant directories to the matlab path
 - type path in the matlab workspace and notice the additional directories in the current path
3. Obtain the open-loop step response
 - (a) Navigate to the directory ...\Examples\Real-time examples
 - (b) Use the function “ME389_Ident_step.mdl”
 - (c) Excite the system with step functions of 0.5 and 1.0 volts
 - (d) When the model file completes, it saves the measured input / output data to the workspace under the variable “IdentStepData”
 - (e) Use the function “ME389_PlotIdent_step.m” to plot the step data
 - The function can be called from the workspace: ME389_PlotIdent_step(IdentStepData)
 - Make your own copy of the plotting routine and edit as needed
 - You may find the function “ginput.m” for sampling data points from the plot, test it out by calling var = ginput(1) from the workspace
4. Obtain k and τ using the method described above.

Q6: Plot both $y(t)$ versus t and $\ln((y_{ss} - y(t))/y_{ss})$ versus t for the system. What are the values of the DC gain k and the time constant τ ? Write down the identified first-order transfer function.

Q7: Plot in the same figure the obtained experimental step response and that predicted by the identified first-order transfer function when the amplitude of the step is 2.5V. How well the predictions by the identified model match the experimental data?

5.2 Model Identification: Open-loop Frequency Response Analysis

For stable linear systems the open loop transfer function can be determined from the frequency response. The frequency response can be obtained from experimental data by exciting the system with a sinusoidal signal of varying frequency. The measured gain and phase shift between the input and output signal constitute the frequency response. The transfer function of the system can then be inferred from the experimental Bode plot. For review of frequency response and bode plots see the lectures available on the web.

Open Loop Frequency Response: Lab Procedure

NOTE: There should be no need to adjust the hardware

- Excite the system with a sinusoidal function
 - Use the function “ME389_Ident_freq.mdl”
 - Connect the DAQ to the hardware
 - Build the mdl file (compile the code), use the build button at the top left corner (Three arrows pointing down)
 - Connect the DAQ to the hardware with the connect button
 - Power on the motor (green switch on the power supply)
 - When ready, run the mdl file with the green arrow
 - Excite the system with a sinusoidal function of amplitude 0.5 Volts and frequency 2 rad/s
 - When the model file completes, it saves the measured input / output data to the workspace under the variable “IdentFreqData”
- Use the function “ME389_PlotIdent_freq.m” to plot the input / output data
 - The function can be called from the workspace: ME389_PlotIdent_freq(IdentFreqData)
 - First the function plots the sampled data. Only the last 10 s are plotted since we are only interested in steady state data.
 - The plot routine, then queries for a frequency, this is used as an initial guess to a fitting routine, which will fit the measured data to a sinusoid of the form
$$\text{Input} : A_{\text{in}} \sin(\omega t + \phi_{\text{in}}) + \text{offset} \quad \text{and} \quad \text{Output} : A_{\text{out}} \sin(\omega t + \phi_{\text{out}}) + \text{offset} \quad (12)$$
 - The details of the fitting routine are not important.
 - However, it is sensitive to initial guess values
 - If you are not satisfied with the resulting fit, try again with a different frequency guess
 - Once you are satisfied with the fit, the plotting function will print the gain and phase shift, which are determined from the fitted sinusoids
$$\text{gain} = \frac{A_{\text{out}}}{A_{\text{in}}} \quad \text{and} \quad \text{Phase Shift} = \phi_{\text{out}} - \phi_{\text{in}} \quad (13)$$
- Repeat the process for varying frequencies, with intervals of 5 per decade.

Q8: Plot the experimental Bode plot for the system. Write down the identified first-order transfer function.

Q9: Plot in the same figure the obtained experimental step response and that predicted by the identified first-order transfer function when the amplitude of the step is 2.5V. How well the predictions by the identified model match the experimental data?

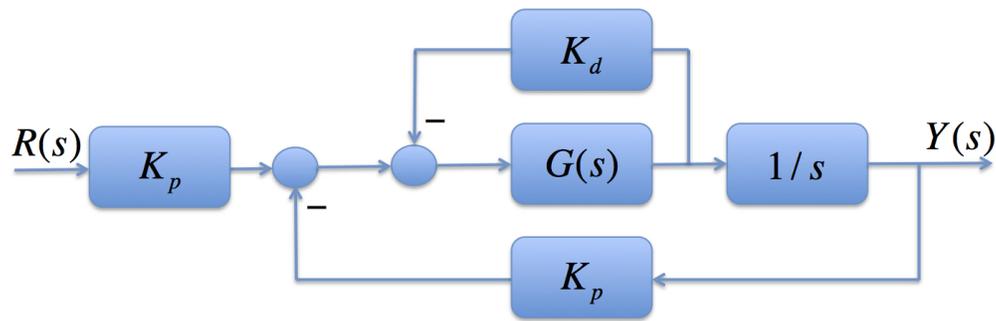


Figure 5: Block diagram of servomechanism-based regulator.

6 Velocity Controller Design

Q10: Using the identified model of the plant obtained either in Section 5.1 or Section 5.2, compute the step response of the open loop system for $0 \leq t \leq 2$ using the Matlab command “step.” Use the Matlab command “stepinfo” to summarize the characteristics of the step response.

Q11: Consider a P controller ($K(s)=K_p$). Plot the position of the closed-loop poles as a function of the controller gain. Give an expression for the steady state error in the closed loop step response. Will this controller guarantee zero steady-state error? How could you reduce the steady-state error? Could you destabilize the closed-loop system by pursuing a smaller steady-state error? Could you drive the closed-loop system response into saturation by pursuing a smaller steady-state error. Compute the step response of the closed-loop system for a “small” gain of $K_p = 0.01$ and for a “big” gain of $K_p = 100$. Plot both the input and the output of the plant.

Q12: Consider a PI controller ($K(s)=K_p+K_i/s$). Give an expression for the steady state error in the closed-loop step response. Will this controller guarantee zero steady-state error? Choose K_p and K_i such that the overshoot is less than 3% and the settling time is less than 0.5 s. Compute the step response of the closed-loop system for $0 \leq t \leq 4$ using the Matlab command “step.” Use the Matlab command “stepinfo” to summarize the characteristics of the step response. Does the step response undershoot (goes below zero)? Explain why in the positive case. What can you do to eliminate the undershoot?

Q13: Test developed controller experimentally. Compare simulated (use ‘step’ command) and experimental step responses. Compare simulated (use ‘slim’ command) and experimental sinusoidal responses.

7 Position Controller Design

Q14: Using the identified model $G(s)$ for the angular velocity in response to the voltage command, obtain the model $H(s)$ for the angular position in response to the same voltage command.

Q15: Using the identified model of the plant obtained either in Section 5.1 or Section 5.2, compute the step response of the open loop system for $0 \leq t \leq 2$ using the Matlab command “step.” Use the Matlab command “stepinfo” to summarize the characteristics of the step response.

Q16: Consider a P controller ($K_1(s) = K_p$) with the structure shown in Figure 5 ($K_d \equiv 0$). Plot the position of the closed-loop poles as a function of the controller gain. Give an expression for the steady state error in the closed loop step response. Will this controller guarantee zero steady-state error? Obtain K_p to guarantee an overshoot of 10%.

Q17: By root locus, show that you can improve performance by adding tachometer feedback ($K_2(s) = K_d$) as proposed in Figure 5. Use the value of K_p obtained in the previous question. Improved performance is evidenced by reduced overshoot without extending the risetime, or reduced risetime without incurring increased overshoot. What value of K_d will allow you to demonstrate this improved performance?

Q18: Test developed controller experimentally. Compare simulated (use 'step' command) and experimental step responses. Compare simulated (use 'slim' command) and experimental sinusoidal responses.