

## LAB3: Study the Effects of Steady-State Error for a Physical System

### Objective:

- To study the effects of steady-state errors by varying system type, input waveforms and loop gains.

**Description of system:** Modern robotic manipulators that act directly upon their target environments must be controlled so that impact forces as well as steady-state forces do not damage the targets. At the same time, the manipulator must provide sufficient force to perform the task. In order to develop a control system to regulate these forces, the robotic manipulator and target environment must be modeled. Assuming the model shown in Figure (1) (Chiu, 1997)

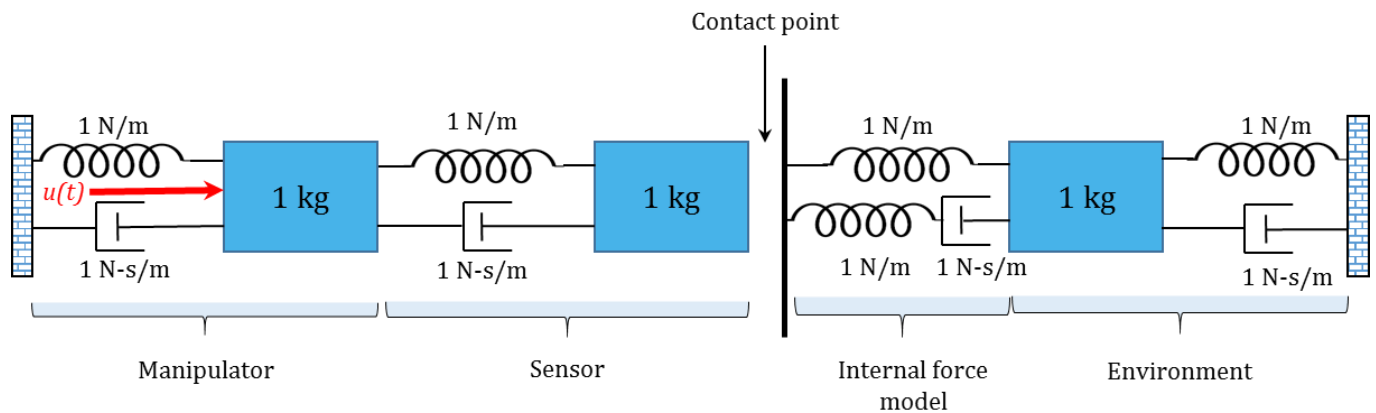


Figure: 1 Robotic manipulator and target environment (© 1997 IEEE)

### Pre lab: (by hand)

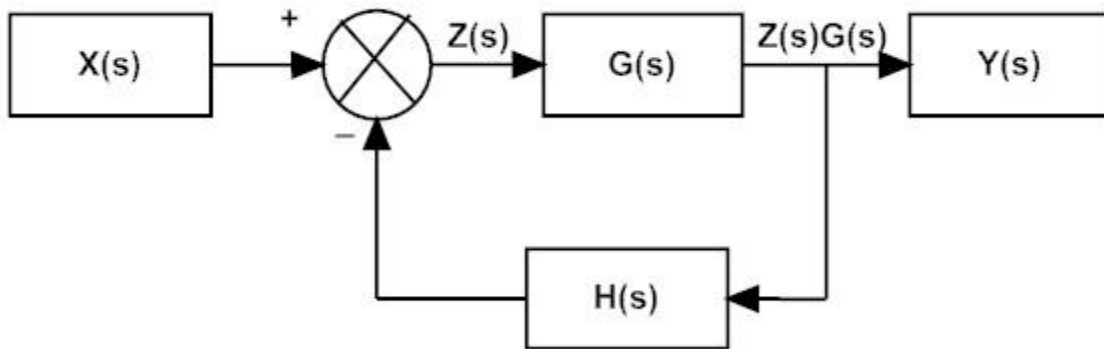
Represent in *transfer function* of the manipulator and its environment under the following conditions

1. The manipulator is not in contact with its target environment.
2. The manipulator is in constant contact with its target environment.

### Background:

Useful Matlab commands to *check your prelab work*

- 'sys ('s'); [B, A] = tf (H(s))' returns the vector of numerator coefficients, B, and the vector of denominators, A, for the equivalent transfer function.



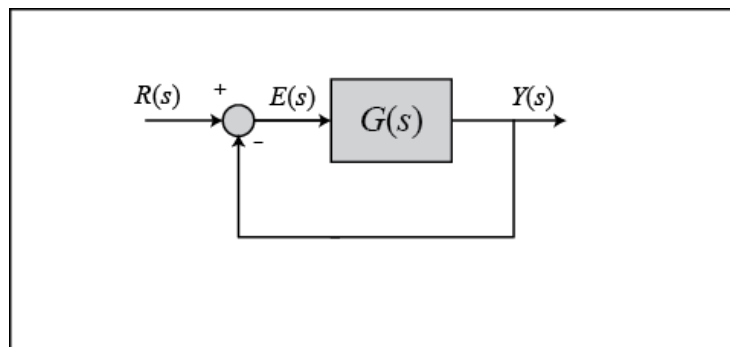
**Figure: 2 General negative feedback system with input  $x(t)$  and output  $y(t)$**

*Steady-State Error* ([http://ctms.engin.umich.edu/CTMS/index.php?aux=Extras\\_Ess](http://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_Ess))

Steady-state error is defined as the difference between the input (command) and the output of a system in the limit as time goes to infinity (i.e. when the response has reached steady state). The steady-state error will depend on the type of input (step, ramp, etc.) as well as the system type (0, I, or II).

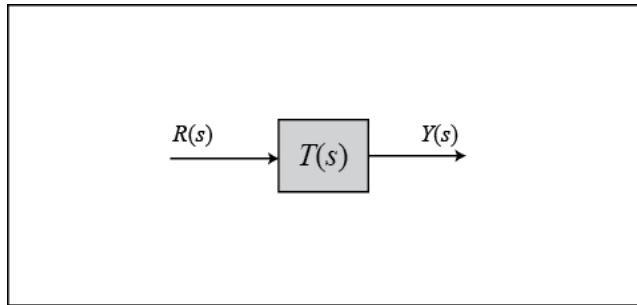
*Calculating steady-state errors*

Steady-state error can be calculated from the open- or closed-loop transfer function for unity feedback systems. For example, let's say that we have the system given below.



**Figure: 3 closed loop system with input  $r(t)$  and output  $y(t)$**

This is equivalent to the following system, where  $T(s)$  is the closed-loop transfer function.



**Figure: 4 closed loop transfer function with input r (t) and output y (t)**

We can calculate the steady-state error for this system from either the open- or closed-loop transfer function using the Final Value Theorem.

$$e(\infty) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{sR(s)}{1 + G(s)} \text{ ---- (1)}$$

$$e(\infty) = \lim_{s \rightarrow 0} sE(s) = \lim_{y \rightarrow 0} sR(s)[1 - T(s)] \text{ ---- (2)}$$

Now, let's plug in the Laplace transforms for some standard inputs and determine equations to calculate steady-state error from the open-loop transfer function in each case.

Step Input  $e(\infty) = \frac{1}{1 + \lim_{s \rightarrow 0} G(s)} = \frac{1}{1 + K_p} \Rightarrow K_p = \lim_{y \rightarrow 0} G(s) \text{ --- (3)}$

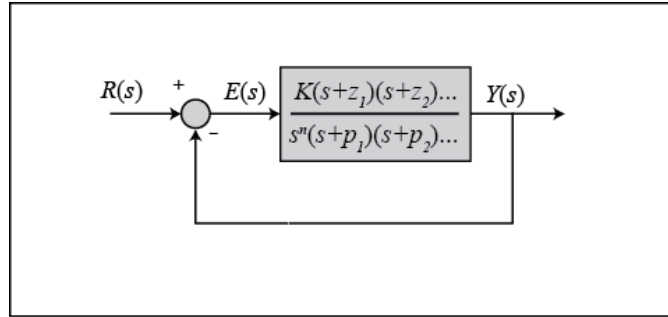
Ramp Input  $e(\infty) = \frac{1}{\lim_{s \rightarrow 0} sG(s)} = \frac{1}{K_v} \Rightarrow K_v = \lim_{s \rightarrow 0} sG(s) \text{ --- (4)}$

Parabolic input  $e(\infty) = \frac{1}{\lim_{s \rightarrow 0} s^2G(s)} = \frac{1}{K_a} \Rightarrow K_a = \lim_{s \rightarrow 0} s^2G(s) \text{ --- (5)}$

### *System type and steady-state error*

The system type is defined as the number of pure integrators in the forward path of a unity-feedback system. That is, the system type is equal to the value of n when the system is represented as in figure (5). It does not matter if the integrators are part of the controller or the plant.

If you refer to the equations (3-5), we have defined certain constants (known as the static error constants). These constants are the position constant ( $K_p$ ), the velocity constant ( $K_v$ ), and the acceleration constant ( $K_a$ ). Knowing the value of these constants, as well as the system type, we can predict if our system is going to have a finite steady-state error.



**Figure: 5 General closed loop transfer function with input  $r(t)$  and output  $y(t)$**

Therefore, a system can be type 0, type 1, etc. The following tables summarize how steady-state error varies with system type.

Type 0 system	Step Input	Ramp Input	Parabolic Input
Steady-State Error Formula	$1/(1+K_p)$	$1/K_v$	$1/K_a$
Static Error Constant	$K_p = \text{constant}$	$K_v = 0$	$K_a = 0$
Error	$1/(1+K_p)$	infinity	infinity
Type 1 system	Step Input	Ramp Input	Parabolic Input
Steady-State Error Formula	$1/(1+K_p)$	$1/K_v$	$1/K_a$
Static Error Constant	$K_p = \text{infinity}$	$K_v = \text{constant}$	$K_a = 0$
Error	0	$1/K_v$	infinity
Type 2 system	Step Input	Ramp Input	Parabolic Input
Steady-State Error Formula	$1/(1+K_p)$	$1/K_v$	$1/K_a$
Static Error Constant	$K_p = \text{infinity}$	$K_v = \text{infinity}$	$K_a = \text{constant}$
Error	0	0	$1/K_a$

**Table: 1 Relationships between input, system type, static error constants and steady-state errors**

### **Lab: (Using Simulink)**

1. Set up the negative feedback system of  $G(s) = \frac{K(s+2)}{(s+3)(s+4)(s+9)(s+12)}$  and  $H(s) = 1$ .  
Plot on one graph the error signal of the system for an input of  $3u(t)$  and  $K= 10, 100, 1000, 5000$ . Repeat for inputs  $3tu(t)$  and  $3t^2u(t)$ .
2. Set up the negative feedback system of  $G(s) = \frac{K(s+6)(s+2)}{s(s+3)(s+4)(s+9)(s+12)}$  and  $H(s) = 1$ .  
Plot on one graph the error signal of the system for an input of  $3u(t)$  and  $K= 10, 100, 1000, 5000$ . Repeat for inputs  $3tu(t)$  and  $3t^2u(t)$ .
3. Set up the negative feedback system of  $G(s) = \frac{K(s+6)(s+2)(s+8)}{s^2(s+3)(s+4)(s+9)(s+12)}$  and  $H(s) = 1$ .  
Plot on one graph the error signal of the system for an input of  $3u(t)$  and  $K= 10, 100, 1000, 5000$ . Repeat for inputs  $3tu(t)$  and  $3t^2u(t)$ .

### **Lab Procedure:**

- 1) Check your prelab work using matlab commands given in background section.
- 2) Get the Simulink block diagram for the case “*manipulator is not in constant contact with target environment*” and simulate the system with  $u(t) = 3$  for 10 secs. Discuss your output.
- 3) Get the Simulink block diagram for the case “*manipulator is in constant contact with target environment*” and simulate the system with  $u(t) = 3$  for 10 secs. Discuss your output.
- 4) Calculate the steady-state errors by hand for each case described in Lab section.
- 5) Build the simulink diagram and outputs for each case described in Lab section. Note your observations and discuss.
- 6) For Each case in Lab section compare your simulink result with calculation by hand. Explain the reasons for any discrepancies.
- 7) Open the simulink ‘Library Browser’ by clicking on Start → Simulink. Open a new model file (.mdl). Build a model of the above closed loop system by using the respective blocks.
- 8) Double-click on the blocks to open its property editor. This will give you options of changing the parameters of the blocks.
- 9) Arrange the blocks in the proper order, and connect them.
- 10) Also save your workspace to matlab.
- 11) Go to Simulation → Configuration Parameters.

NOTE: Do not forget to save your work

**Post Lab:** Write a report in abstract, objective, theory, procedure, results, conclusion and appendices format. All steps should be clearly mentioned. Include all plots and results.

## **References:**

Chiu, D.K., and Lee, S. *Design and Experimentation of a Jump Impact Controller*. IEEE control systems, June 1997, pp 99-106

Franklin, G.F., Powell, J.D., and Emami-Naeini, A. *Feedback Control of Dynamic Systems*. Pearson, 2015, 7/E

Nise, N.S., *Control Systems Engineering*. John Wiley & sons, Inc, 1995. 6/E

### *Useful Links*

<http://vigir.ee.missouri.edu/~gdesouza/ece4310/index.htm> (intro slides)

[http://vigir.ee.missouri.edu/~gdesouza/ece4310/Lab\\_Assignments/ECE\\_Feedback\\_Lab3.pdf](http://vigir.ee.missouri.edu/~gdesouza/ece4310/Lab_Assignments/ECE_Feedback_Lab3.pdf) (Lab 3 document)

[http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics\\_Matlab](http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics_Matlab) (Matlab basics)

[http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics\\_Simulink](http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics_Simulink) (Simulink basics)

<https://www.youtube.com/watch?v=PXxveGoNRUw> (Video: Final value theorem and steady state errors)