SERVO MOTOR CONTROL DESIGN ACTIVITY

Worth 20%. See CANVAS for details on the submission format and other requirements.

Background: Servo motor position control

In this activity you will work with the **input to shaft angle position** servo motor transfer function:

$$G_p(s) = \frac{V_p(s)}{V_m(s)} = \frac{K_m}{s(s+\ \alpha)}$$

and investigate (unity) negative feedback for shaft angle position control based on shaft angle measurements. Here V_m is input signal and V_p is output shaft position in units of angle.

Note that this transfer function G_p involves a different output variable (i.e. angle position) than the transfer function G you investigated in the previous assessment activity (whose output variable was angle rate). The transfer functions G_p and G involve the same parameters K_m and α, which arise from the same underlying physical motor, but differ by an additional $\frac{1}{s}$ factor. Their step responses look different.

Assessed Tasks – Control System Design

In the following, you will perform a control system design for the "input to shaft angle position servo motor" with the parameter values you examined in the EGB345 Servo Motor System Identification task.

Preliminary information: <u>your servo motor parameter values</u> for this activity.

In the below tasks use for <u>your servo motor parameter values</u>, K_m and α, the following:

- The K_m and α values provided by the MATLAB *GenerateCSVRandom.m* function provided for Task 1 of the EGB345 Servo Motor System Identification task.

In this activity we investigate (unity) negative feedback closed-loop system, where the "unity" means consider H=1 on the feedback path.

Task 1: Design feedback control for a requested response

Aim: Build a MATLAB representation of your servo motor system, design feedback control and simulate.

Complete the following sub-tasks:

Task 1.1: Build a MATLAB representation of the **input to shaft angle position** servo motor in open loop having <u>your servo motor parameter values</u>. We call this representation <u>*your open-loop system*</u>. Analytically assess the expected unit step response of your open loop system. Plot the unit step response of your open-loop system.

Task 1.2: Analytically create the transfer function expression of a (unity) negative feedback <u>closed-loop system</u> containing a general gain K controller in front of your open-loop system.

Task 1.3: For *K=1*, if underdamped in closed-loop, analytically determine the settling time and % overshoot. If overdamped in closed-loop, use a unit step response plot to graphically estimate settling time (after completing Task 1.5 below).

Task 1.4: Using mathematical analysis, determine a value of *K* that results in a 5% overshoot. Report the location of the closed-loop poles that would result.

Task 1.5: Build an appropriate MATLAB representation of the closed-loop system for *K=1*. Plot the unit step response of the closed-loop system.

Task 1.6: Build an appropriate MATLAB representation of the closed-loop system with the designed value of *K* for 5% overshoot. Plot the unit step response of the closed-loop system

Task 1.7: Comment on whether the three unit step responses seen in the previous tasks (Task 1.1, 1.5 and 1.6) match your analytic results.

Here and below, the word "analytically" means the outcome of mathematical analysis. We would expect several mathematical operations deriving the required mathematical object.

Submit in the provided report template the following:

1. Your servo motor parameter values (K_m and α) as selected from the options on the previous page.
2. The analytical steps and the derived transfer function expression of the (unity) negative feedback closed-loop system for a general gain of *K*. (Include your working.)
3. Calculated settling time and % overshoot, if any.
4. Plot of the unit step response of the open-loop system.
5. Plots of the unit step responses of the closed-loop system with *K=1* and with the designed value of *K* (two plots).
6. Comment on whether the plot of the three unit step responses matches your analytic results (and comment on any differences from the unit step response of your open-loop system).
7. Your MATLAB scripts and functions.

Task 2: Investigation of the gain K

Aim: Evaluate the impact of errors in the value of *K*, such as arises due to imperfect hardware components.

Complete the following sub-tasks:

Task 2.1: Analytically determine the location of the (unity) negative feedback closed-loop poles for at least 5 values of *K* below and at least 5 values above the value you designed in Task 1. Ensure a selection of *K* values so that the closed-loop system shows examples of both over and underdamped closed-loop unit step response behaviour.

We initially suggest increasing and decreasing by a factor of 10% (that is 0.9**K*, 0.8**K*, … and 1.1**K* etc.). However, ensure you select a range of *K* values showing examples for both over and underdamped closed-loop unit step response behaviour in closed-loop.

Task 2.2: Neatly hand draw a graph in the s-plane that reports the closed-loop pole locations used in the previous sub-task (on paper or electronic equivalent but MATLAB-generated plots are not accepted for this specific sub-task). Neatly label a couple of the poles with their corresponding *K* value. Comment on the meaning of what you see in the graph you have drawn. Provide connections to concepts encountered in lectures.

Task 2.3: Using your MATLAB representation of the closed-loop system, plot the unit step response of the following 3 cases: the original *K*, plus the largest and smallest values of *K* you tested in the above sub-tasks. Comment on what you see in the unit step response plots.

<u>Submit in the provided report template the following:</u>

1. Report in a table, which has a new row for each of your ten *K* values and the corresponding closed-loop pole location. Include your calculations.
2. Your drawing of the closed-loop pole location, and your comment about the meaning of what you see you have drawn. Provide connections to concepts encountered in lectures. MATLAB or computer-generated plots are not acceptable for this specific sub-task.
3. The requested plots of unit step responses of the closed-loop system, and the requested comments.
4. Your MATLAB scripts and functions.

Task 3: Robustness study - an extra unmodelled fast pole

Aim: Investigate the impact of an extra unmodelled fast pole.

You are told the true system has an additional fast pole at β and is described by:

$$G_{extra}(s) = \frac{V_p(s)}{V_m(s)} = \frac{\beta K_m}{s(s+\ \alpha)(s+\beta)}$$

You are told the unmodelled pole at -β is faster than the pole at -α but you don't know how much faster ("unmodelled" means you didn't know the pole was present and/or didn't know its specific value). Note the extra β factor in the numerator ensures the systems with different extra poles have the same final value for their unit step response (in open-loop).

You do not need to consider the gain error issue considered in Task 2.

Consider the three systems:

A) No extra pole (previously used in Task 1 and elsewhere):

$$G_p(s) = \frac{V_p(s)}{V_m(s)} = \frac{K_m}{s(s+\ \alpha)}$$

B) Fast extra pole where $\beta = 10\alpha$

$$G_{fo}(s) = \frac{V_p(s)}{V_m(s)} = \frac{10\alpha K_m}{s(s+\ \alpha)(s+10\alpha)}$$

C) Slow extra pole where $\beta = 2\alpha$

$$G_{so}(s) = \frac{V_p(s)}{V_m(s)} = \frac{2\alpha K_m}{s(s+\ \alpha)(s+2\alpha)}$$

In Task 1 you previously designed a gain *K* (for (unity) negative feedback closed loop system) to achieve a 5% overshoot assuming no extra pole system. Label this previously designed gain *K_NE*.

Complete the following sub-tasks:

Task 3.1: The analytic steps to design a new controller gain *K* (for (unity) negative feedback closed loop system) to achieve 5% overshoot assuming that $G_{fo}(s)$ is the true model. Label this designed gain *K_FO*. Check your design for 5% overshoot by plotting the unit step response. Provide this unit step response.

Task 3.2: The analytic steps to design a new controller gain *K* (for (unity) negative feedback closed loop system) to achieve 5% overshoot assuming that $G_{so}(s)$ is the true model. Label this designed gain *K_SO*. Check your design for 5% overshoot by plotting the unit step response. Provide this unit step response.

Task 3.3: Now consider the three gain designs (*K_NE*, *K_F0* and *K_S0*) and the three systems ($G_{ne}(s), G_{fo}(s)$ and $G_{so}(s)$) and investigate each of gains when they are used in a situation *other than they were initially designed* (this type of situation is called design mismatch). There are three gain designs each with 2 possible mismatched systems (the other two systems) which means there are 6 mismatch cases to consider. For example, *K_NE* can be considered in (unity) negative feedback closed loop with both $G_{fo}(s)$ and $G_{so}(s)$. You are asked to provide plots of the unit step responses for each of the 6 cases where control gains are applied to a system for which not designed. There are 6 plots.

Task 3.4: Compare and contrast the performance of the controllers, and report what design considerations are important. For example, would your *K_NE* design work on $G_{fo}(s)$? Can you explain in terms of any concepts encountered in lectures, tutorials or practicals? If so, provide this explanation. If you are told that the implemented closed-loop system should have %5 overshoot or less, which of the designed gains (*K_NE*, *K_F0* and *K_S0*) would you recommend?

<u>Submit in the provided report template the following:</u>

1. Your analysis for the two new designs above (less than 2 pages).
2. The requested unit step response plots. There are 8 plots.
3. Your requested comments and explanations.
4. Your MATLAB scripts and functions.

Task 4: Improved transient response using a dynamic compensator

Aim: Design dynamic compensation for an improved transient response.

You now introduce dynamic compensation designed so that the closed-loop system has *half the settling time* achieved in Task 1, has the same overshoot, and keeps the same steady-state properties. (Here, “dynamic compensation” is the same as a “dynamic controller” and means a controller with poles and/or zeros). You do not need to consider the gain issue considering in Task 2. Further, you do not need to consider the extra pole examined in Task 3. That is, your work on this task may be based on the Task 1 open loop system $G_p(s)$.

In this task, you are to use a root locus control design approach for a (unity) negative feedback closed loop system. You may use other compensators, but one useful compensator is a lead compensator which has the transfer function with one zero z_c and one pole p_c (with $z_c < p_c$)

$$\frac{s+z_c}{s+p_c}$$

Complete the following sub-tasks:

Task 4.1: Analytically determine the location of the desired closed-loop poles for the designed closed-loop system to achieve the requested response.

Task 4.2: Using an s-domain graph, determine the location of your compensator’s pole and zero location that allows you to achieve the requested unit step response.

Task 4.3: Determine the value of *K* that achieves the requested unit step response (with the above step 2 poles and zeros) of the designed closed loop system.

Task 4.4: Using your MATLAB representation of the closed-loop system, make modifications to include the designed compensator (designed in the above steps 2 & 3), and provide a plot of its unit step response. Comment on the meaning of what you see. What are the closed-loop pole (and zero) locations? What is the settling time and %OS of your design?

<u>Submit in the provided report template the following:</u>

1. The analytic steps and your determined desired closed-loop pole location. (Include your working.)
2. The analytic steps and your desired compensator (pole, zero and gain value). (Include your working.)
3. Plot of the unit step response of the closed-loop system.
4. Comment on whether the Task 4.4 plot, closed-loop pole and time response information matches what you excepted given your answers to Task 4.1, 4.2 and 4.3.
5. Your MATLAB scripts and functions.