DIRECT DISCRETE CONTROL DESIGN

Topics Covered

- Directly designing a discrete controller.
- Advantages and disadvantages of direct discrete control design.

Prerequisites

- Hardware Interfacing laboratory experiment.
- PD Control laboratory experiment.
- Introduction to Discrete Control laboratory experiment.
- Discrete Stability laboratory experiment.

1 Background

In the Introduction to Discrete Control and Discrete Stability laboratory experiments, we discussed how the results of the continuous control design process can be modified to work in a similar fashion in the discrete time domain. In this section, we will investigate how to design a discrete control in the native discrete domain.

To directly design a digital controller, we are interested in the transfer function between the input $u(nT)$ and the output $y(nT)$ of the system, two purely digital signals. The transformation techniques described in the previous labs used approximation techniques to find discrete equivalents of the continuous plant model, and consequently introduced an approximation error into the design. In direct digital design, everything that happens between the input $u(nT)$ and the output $y(nT)$ depends only on the input at that particular sample time, nT. Therefore, by the definition of Zero-Order Hold (ZOH), it describes the system behavior *exactly* without introducing any additional error.

The discrete closed-loop unity feedback system is shown in Figure 1.1 with the discrete compensator, $G_c(z)$, and the discrete plant, $G_p(z)$.

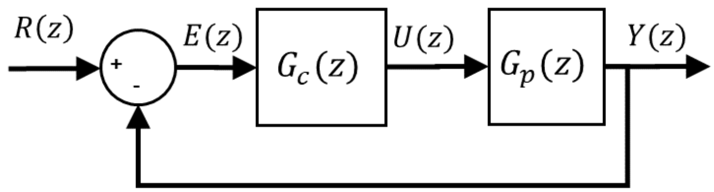


Figure 1.1: Closed-loop unity feedback of discrete system.

The first step in the discrete control process is to obtain a discrete transfer function for our continuous plant using zero-order hold. Formally, for a continuous plant $P(s)$ that is preceded by a Zero-Order Hold (ZOH), the discrete transfer function is given by

$$G_p(z) = \mathcal{Z}\left\{\frac{1-e^{-Ts}}{s}P(s)\right\} = (1-z^{-1})\mathcal{Z}\left\{\frac{P(s)}{s}\right\}, \tag{1.1}$$

where $\mathcal{Z}\{P(s)\}$ is the z-transform of the sampled time series of the s-domain transfer function $P(s)$. The additional integrator in the s-domain expression is based on the step input from the ZOH during each sampling period. In the discrete domain, a single step with a unit duration is equivalent to a step of infinite duration combined with the same step of opposite sign one time step delayed, i.e. $(1-z^{-1})$.

Recall the QUBE-Servo 2 plant voltage-to-position transfer function is

$$P(s) = \frac{\Theta_m(s)}{V_m(s)} = \frac{K}{s(\tau s+1)}, \tag{1.2}$$

where K is the model steady-state gain, τ is is the model time constant, $\Theta_m(s) = \mathcal{L}[\theta_m(t)]$ is the motor/disk position, and $V_m(s) = \mathcal{L}[v_m(t)]$ is the applied motor voltage.

This derivation also contains the biggest disadvantage of direct digital control design: to perform the z-transform in Equation 1.1, the sampling rate has to be known and fixed. Therefore, choosing a different sampling period requires a complete re-design from scratch. For the control design using another technique, such as the Zero-Pole Matching or Bilinear (Tustin) methods, only the continuous-time controller had to be re-evaluated.

Once $P(z)$ is obtained, the actual discrete controller design is similar to the controller design of continuous systems. Generally speaking, the design rules for continuous and discrete control are identical, with one important exception: the stability boundary for continuous systems is the imaginary axis; for discrete systems, it is the unit circle, as described in Discrete Stability laboratory experiment.

1.1 Discrete Lead Compensator Design

The discrete lead-lag compensator can be defined by

$$C(z) = K\frac{z - z_0}{z - p_0}, \tag{1.3}$$

where K is the proportional gain, z_0 is the zero location, and p_0 is the pole location. For a lead compensator, the zero is greater than the pole, i.e. $|z_0| > |p_0|$. The pole must be placed inside the unit circle, $|z| < 1$.

2 In-Lab Exercises

The SIMULINK® model shown in Figure 2.1 implements the discrete lead compensator as shown in Figure 1.1.

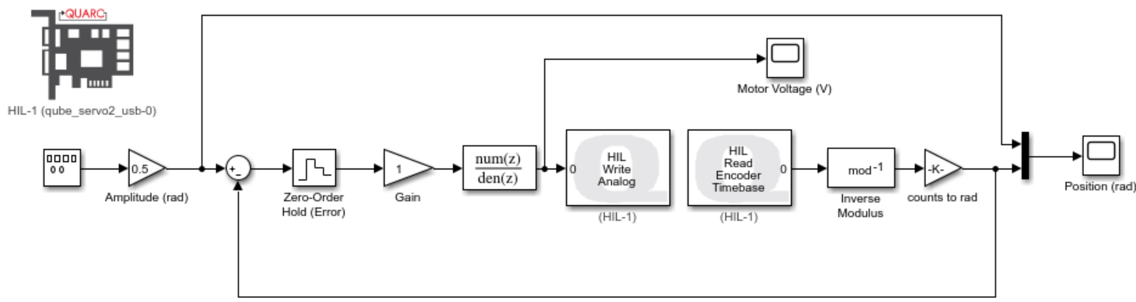


Figure 2.1: Simulink/QUARC model implementing a discrete lead compensator

Design a discrete lead compensator that meets the following requirements for the steady-state error (e_{ss}), settling time (t_s) and percentage overshoot (PO):

$$\begin{aligned} e_{ss} &= 0 \text{ rad} \\ t_s &= 0.30 \text{ s} \\ PO &\leq 5 \text{ \%} \end{aligned} \tag{2.1}$$

2.1 Control Design

1. Similarly as done in Discrete Stability laboratory experiment, the MATLAB® *Control System Designer* application will be used to assess how proportional gain affects the system when it is sampled at 33.3 Hz.
2. Open MATLAB® and save your continuous model transfer function, $G(s)$, as the MATLAB variable `Gs`. The continuous-time transfer function is given by

$$G(s) = \frac{K}{s(s\tau + 1)}, \tag{2.2}$$

   where K is the model steady-state gain, τ s is the model time constant. Use either values $K = 23.2$ rad/Vs and $\tau = 0.15$ s or, for more accurate results, use the model parameters derived for your particular QUBE-Servo 2 system by running one of the modeling labs, e.g. Step Response Modeling laboratory experiment.
3. Discretize the plant transfer function, $G(z)$, with the Zero-Order Hold (ZOH) method and sampling interval of 0.03 s. Call `controlSystemDesigner(Gz)` to open the MATLAB design tool shown in Figure 2.2. The default view shows the root-locus and bode plots of loop transfer function: $L(z) = C(z)G(z)$, along with the step response of the closed-loop system. Show the MATLAB commands you used and attach a screen capture of *Control System Designer*.

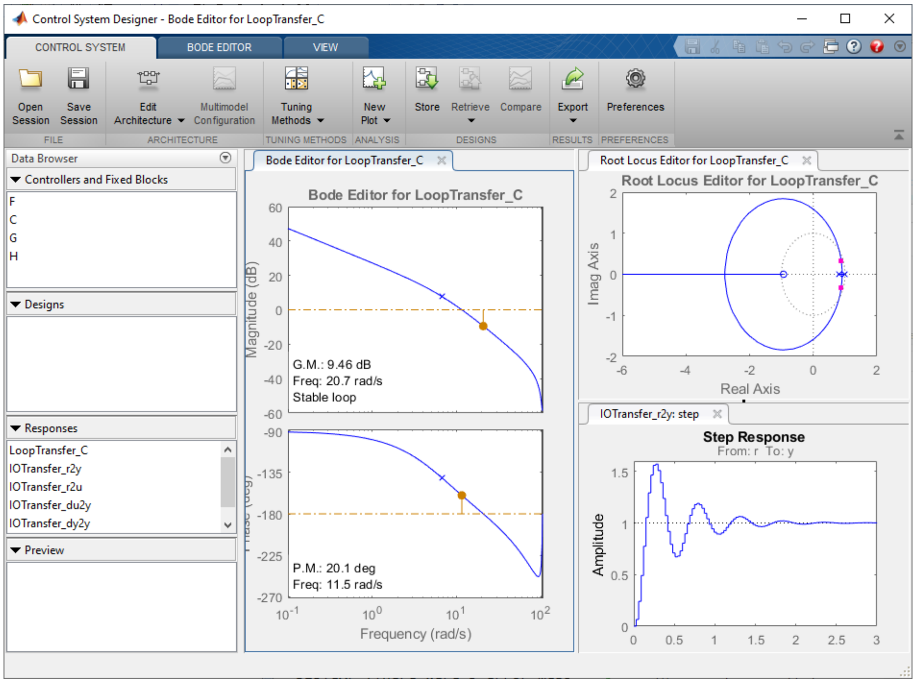


Figure 2.2: MATLAB Control System Designer used to assess stability of discrete system.

4. For now, do not change the closed-loop pole/zero locations. Right click on the root-locus diagram and add a z-plane grid to the view.

5. Use the *Design Requirements* function in *Control System Designer* to limit the feasible closed-loop location area. To add controller design requirements, right-click on the *Root-Locus Editor* plot and select: *Design Requirements* | *New*. Use this to add the percent overshoot and settling time requirements given in Equation 2.1. Describe the resulting feasible region and attach a screen capture of the root-locus.

6. The compensator in the *Control System Designer* is initially set to a proportional controller (with a gain of 1). The solid pink squares and circles are the closed-loop pole and zero locations, respectively, that can be modified by moving them on the *Root-Locus* plot. Are you able to move some the poles into the feasibility region?

7. Add a `Lead` compensator to the *Control System Designer*. To do this, in the *Controllers and Fixed Blocks* window of *Control System Designer*, right-click on the compensator `C` and choose *Open Selection*. In the *Compensator Editor* window, right-click on the *Pole/Zero* pane and select *Add Pole/Zero* | *Lead*. This will add a zero and a pole to your compensator and update your root-locus diagram and step-response plot. In the *Root-Locus Editor* plot, the blue `x` and `o` are the plant pole and zero locations and the red `x` and `o` are the controller pole and zero locations, respectively.

8. As a rule of thumb for a closed-loop system with three closed-loop poles, you should try to vary the controller pole and zero **locations such that the root-locus goes through the intersection of the feasible regions of your requirements**, i.e. ensure that both requirements can be fulfilled at their maximal (or minimal) admissible value.

   - Once the root-locus goes through these points, change the control gain by drag-and-dropping the closed-loop poles until a complex conjugate pair is close to the intersection points.
   - Verify that the third closed-loop poles (on the real axis) is inside the unit circle.
   - For best results, you should try to find a combination of controller pole and zero locations where all poles are approximately equidistant to the origin.

9. Go to the root-locus diagram and **only move the controller pole** (don't change the location of the zero or controller gain by moving the closed-loop poles around) to see the effect of different pole locations.

Describe what happens when you move a closed-loop pole along the root-locus. Can you satisfy the design requirements? Explain and attach a screen capture of *Control System Designer*.

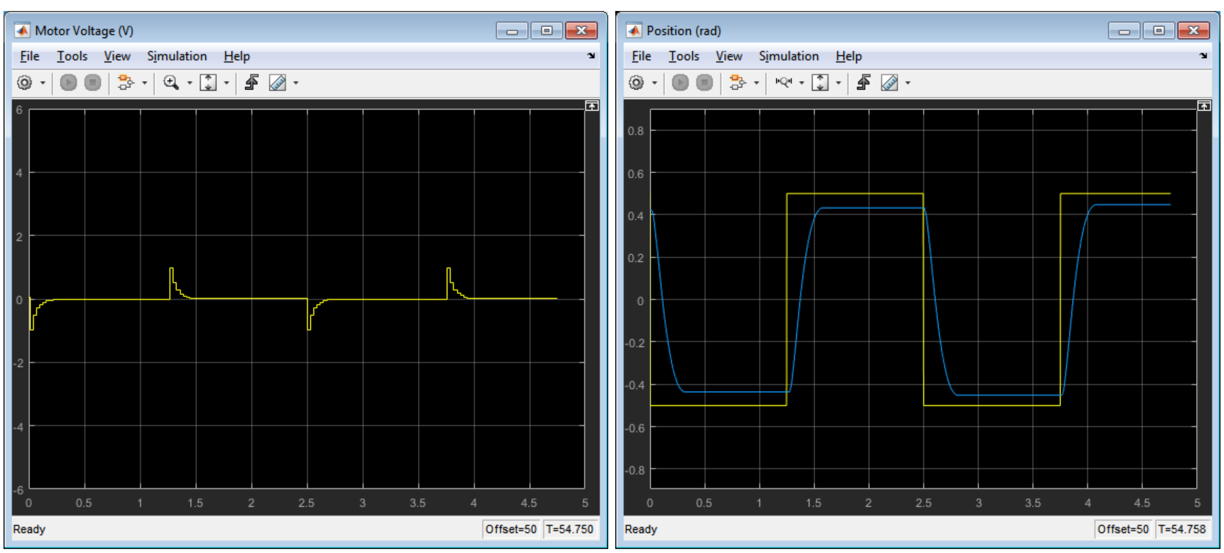
10. Move the pole and/or zero of the lead compensator and adjust the gain until the closed-loop poles are moved such that requirements are satisfied. Give the resulting lead compensator and attach a screen capture of *Control System Designer*. To ensure your design has a large enough gain, make sure the step response has an overshoot between 5% and 15% and a settling time that is less than 0.35 s. The overshoot and settling time typically decrease when the controller is ran on the actual system due to the inherent friction in the motor. Make sure you use the compensator shown in the *Preview* window in *Control System Designer*.

2.2 Lead Control Implementation

1. Connect and power up the QUBE-Servo 2 system and ensure the inertial disc load is mounted.
2. Design (or open, if supplied) the Simulink model shown in Figure 2.1.
3. Set the Signal Generator block such that the servo command (i.e. reference angle) is a square wave with an amplitude of 0.5 rad and at a frequency of 0.4 Hz.
4. Set the sampling interval of the Zero-Order Hold to 0.03 s by entering `Ts = 0.03` in the MATLAB prompt.
5. Based the finalized lead compensator design using *Control System Designer*, enter the gain and lead compensator z-transform in the Simulink model Gain and Discrete Transfer Fcn blocks.
6. Build and run the QUARC® controller.
7. The response shown in Figure 2.3 was obtained using the following *sample* lead compensator:

$$G_c(z) = \frac{z - 0.65}{z - 0.25} \tag{2.3}$$

Note: This is only an example response. The response using *your designed lead compensator will be different.*



(a) Voltage (b) Position

Figure 2.3: Example response of discrete lead compensator at 33.3 Hz.

8. Does your controller meet the desired requirements listed in Equation 2.1? If not, try tuning the gain to minimize the steady-state error and/or improve the settling time. Record the finalized compensator used and attach the voltage and position response of your finalized design. Find the resulting percent overshoot, settling time, and steady-state error as well.

9. Were there any differences between the simulated response in `sisotool` and the measured one? If so, list one reason for this discrepancy.
10. Stop the QUARC controller.
11. Power off the QUBE-Servo 2 if no more experiments will be conducted.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com