INTRODUCTION TO DISCRETE CONTROL

Topics Covered

- Digitization of continuous controllers.
- Model digital computer using ideal sampler and zero-order hold.
- Implementation issues of continuous controllers on digital computers.

Prerequisites

- Hardware Interfacing laboratory experiment.
- PD Control laboratory experiment.

1 Background

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

$$P(s) = \frac{K}{s(\tau s + 1)}, \tag{1.1}$$

where K is the steady-state gain and τ is time constant. This is a continuous transfer function and, in the PD Control laboratory experiment, a continuous PD controller was designed for this system. Strictly speaking, this implies that the controller is implemented using only analog electronics, such as resistors, capacitors and inductors.

In most cases, it is not feasible to implement a controller using analog electronics. Analog electronics are inherently prone to variations in their nominal values, and thus extensive fine-tuning is necessary for each implemented controller using the same nominal electronics. Furthermore, each pure analog circuit will be very susceptible to environmental changes, in particular changes in temperature and humidity.

Therefore, most control systems are now being implemented on digital computers, i.e. using either a PC/laptop or microprocessor. In this lab, we will be investigating the impact that a direct application of a continuous controller has in a digital environment. Strictly speaking, all controllers implemented with QUARC® so far have been *continuous controllers running on digital equipment*.

1.1 Sampling and Holding of Signals

As illustrated in Figure 1.1, controller such as the PD control shown are implemented on the computer and this interacts with the QUBE-Servo 2 through its Data Acquisition (DAQ) device, as described in the QUBE-Servo 2 User Manual.

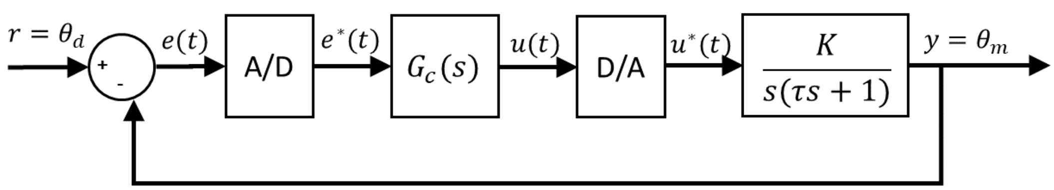


Figure 1.1: Digital control on computer interfaces with QUBE-Servo 2 through DAQ

The digital computer (or microprocessor) can be modeled using an ideal sampler and zero-order hold (ZOH) function, as shown in Figure 1.2. This will be used to investigate what effect running a continuous controller has on a digital system.

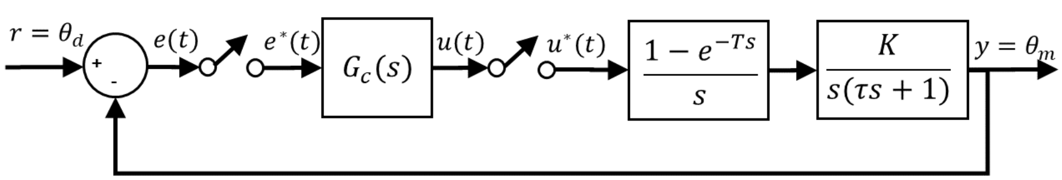


Figure 1.2: Sampling modeled using Zero-Order Hold

The ideal sampler of the error signal is

$$e^*(t) = \sum_{n=0}^{\infty} e(nT)\delta(t - nT) \tag{1.2}$$

where T is the sampling interval time and the δ is the impulse function. The transfer function representation of the zero-order hold function is

$$G_h(s) = \frac{1 - e^{-Ts}}{s}$$

Based on the continuous time controller developed, this experiment will investigate how different sampling times affect the control performance. To measure the controller performance, the percent overshoot and settling time of the response will be analyzed.

1.2 Overshoot and Settling Time

The absolute peak value of the response is denoted by the y_{max} and it occurs at the peak time t_{max}. For a response to a step signal at time t_0 of magnitude R_0 similar to Figure 1.3, the percentage overshoot is found using

$$PO = \frac{100\,(y_{max} - R_0)}{R_0}. \tag{1.3}$$

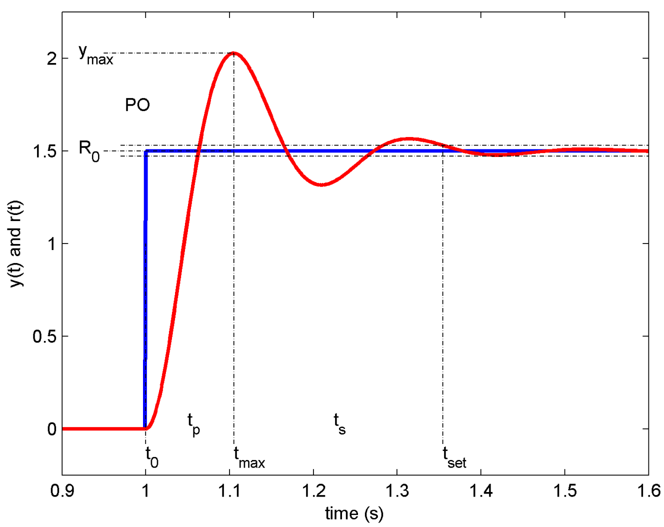


Figure 1.3: Standard second-order step response

The settling time t_s of a system to a step response is the time interval from the initial step time t_0 until the system response has entered and remains within an error margin of 2 % of the step magnitude R_0 called t_{set},

$$t_s = t_{set} - t_0. \tag{1.4}$$

2 In-Lab Exercises

Based on the continuous time controller developed in the PD control laboratory experiment, you will investigates how different sampling times affect the control performance.

2.1 Rate Transitions in MATLAB

In order to be able to investigate how a digitized controller works on a *near continuous* plant, we need to know how to set up our SIMULINK® model such that parts of the model run at different speeds. This can be done using the Zero-Order Hold and Rate Transition blocks, as shown in Figure 2.1

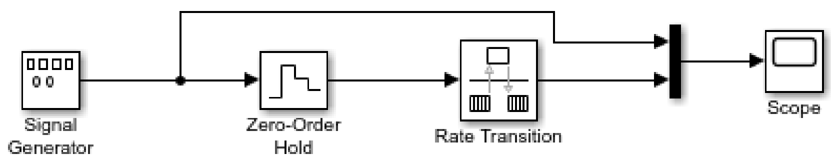


Figure 2.1: Simulating Digital Controller using Zero-Order Hold and Rate Transition blocks

1. Build the Simulink model shown in Figure 2.1. The Zero-Order Hold block can be found in the *Simulink* | *Discrete library* and the Rate Transition block from the *Simulink* | *Signal Attributes* library. To be able to compare our original and discretized signal, we would like to overlay them in the same graph. To do so, we use a signal multiplexer. The Mux block can be found in the *Simulink* | *Signal Routing* library.
2. As shown in Figure 2.2, set the solver to *Fixed-step* and *ode1 (Euler)* and the *Fixed step-size* of the Simulink model to 0.002. The fixed-step size is the sampling interval of the controller, thus the controller is sampling interval of 0.002 s, or 500 Hz.
3. Set the sampling time of the Zero-Order Hold block to 0.1.
4. Set the Rate-Transition block sampling time to -1 and ensure not options are selected.
5. Set the simulation time to 5 seconds, and the simulation mode to *normal*.
6. Run the simulation.
7. Describe the signals that you see in the scope. What happens when you change the parameters in the in Zero-Order Hold block? Attach representative scope responses.
8. Why is the Rate Transition block needed?

2.2 Discretization of a PD Controller

The goal of this lab is to implement a continuous PD controller in a discrete time environment as shown in Figure 2.3. The Simulink model is configured to run with QUARC at a sampling interval of 0.002 s i.e. 500 Hz. The Zero-Order Hold blocks will be used to discretize the PD control and examine the difference between the continuous and discrete versions.

1. Design the QUARC® controller from Figure 2.3. The Manual Switch block can be found in the *Simulink* | *Signal Routing* library.

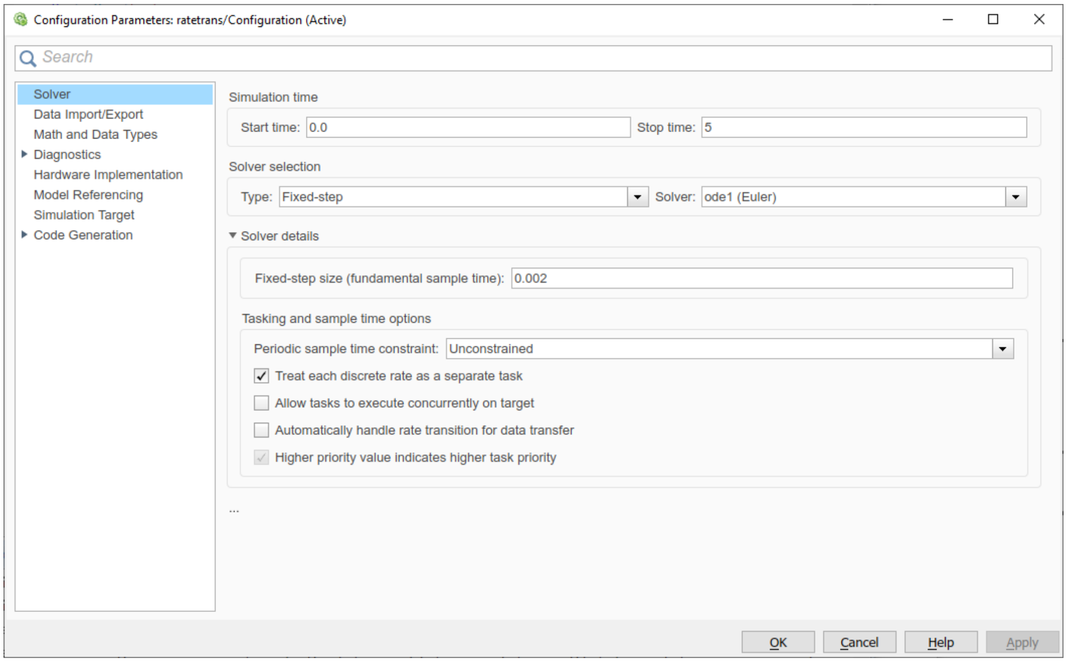


Figure 2.2: Simulink model solver options

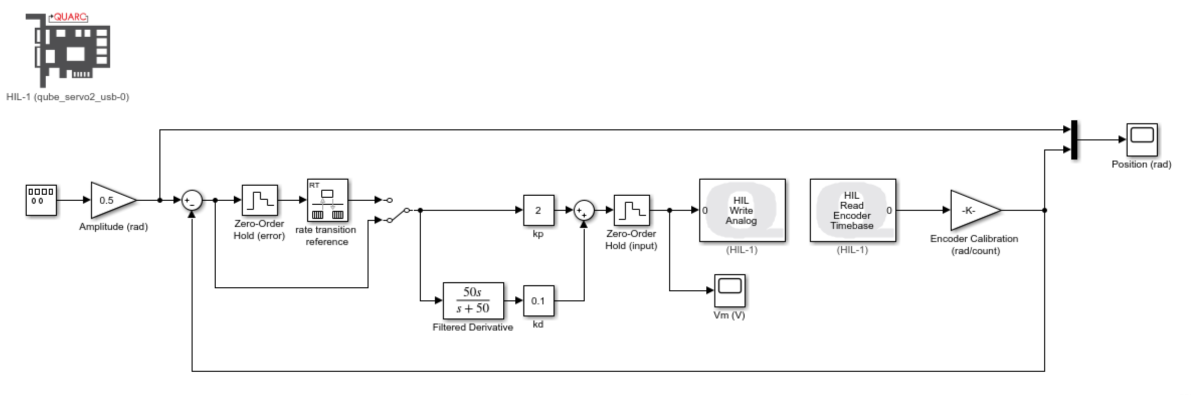


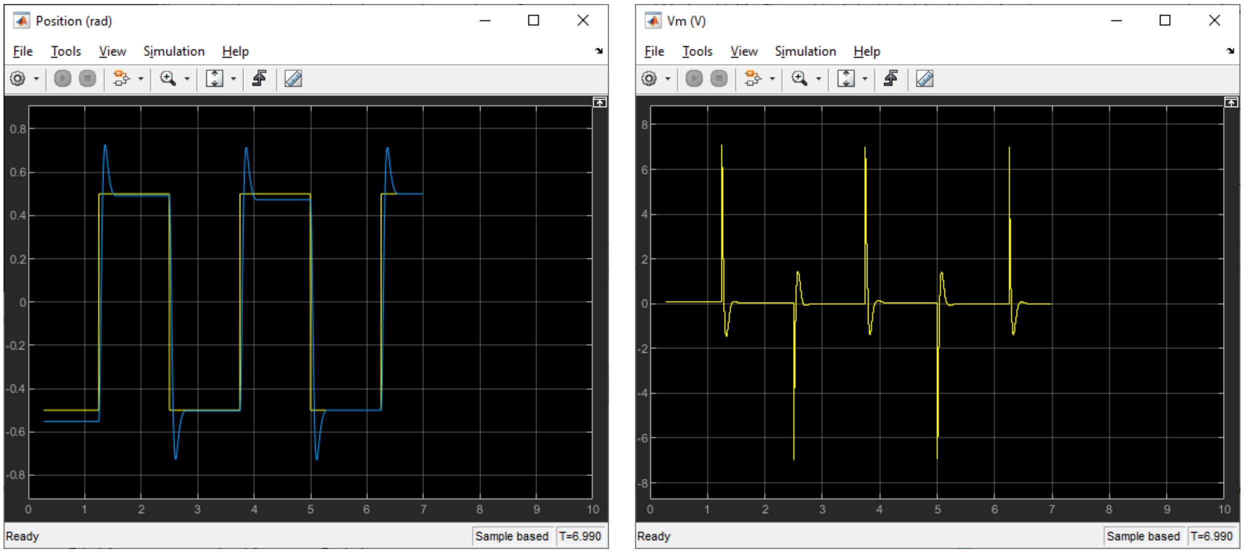
Figure 2.3: Digitization of QUBE-Servo 2 Controller

2. Set the proportional gain, kp, to 2.0 V/rad and the derivative gain, kd, to 0.1 V.s/rad.

3. The sampling frequency for the Zero-Order Hold blocks can be controlled by the variable `Ts`. For a sampling frequency of 500 Hz, enter `Ts = 1/500` in the MATLAB® workspace. Make sure that in the Rate-Transition block none of the options selected and the output port sampling time to -1.

   Important: The MATLAB variable `Ts` must be defined to run the model.

4. Build and run the QUARC® controller. The response should look similarly as shown in Figure 2.4.



(a) Position Response

(b) Motor Voltage

Figure 2.4: QUBE-Servo 2 PD control with $T_s = 0.002$ s

5. As the controller is running, change the position of Manual Switch and compare the response between the continuous and discrete PD controls. Do you notice any difference between the two responses? Explain why or why not. Measure the percent overshoot and settling time of the response of the digitized controller.
6. Stop the QUARC controller.
7. Change the sampling interval of the discrete control to 0.03 s by entering `Ts = 0.03` in the MATLAB Command Window.
8. What is the sampling frequency of the digital control?
9. Build and run the QUARC controller.
10. Investigate what happens if you are using a *sampled error signal instead of the continuous error signal* by changing the position of the Manual Switch. Compute the percentage overshoot and settling time when using the sampled error. Compare the performance to the case with continuous error signal. Comment on the similarities and differences.
11. Stop the QUARC controller.
12. Summarize your observations on how the sampling rate affects the performance of controllers designed for continuous systems once they are directly digitized.

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