University of Toronto
Department of Electrical and Computer Engineering
ECE411S – Real-time Computer Control

LAB 1

Sampled-Data Systems Are Different

1 Purpose

The purpose of this experiment is to familiarize you with some differences in the behaviour of sampled-data systems compared to purely continuous time systems or purely discrete time systems. We use Matlab as the analysis tool.

2 Introduction

You have used Matlab in previous control courses to analyze linear time-invariant continuous time systems. The operation of sample and hold brings in discrete time system operations. Aspects related to control design will be discussed in depth in the lectures and course notes. Here we explore some interesting aspects of sampled-data systems not present in their purely continuous time or discrete time counterparts.

3 Experiment

3.1 Aliasing

Aliasing occurs when continuous signals are sampled. Here we illustrate this phenomenon using the frequency response of a continuous time system.

1. Consider a continuous time transfer function

$$G(s) = \frac{1}{s^2 + 0.8s + 1}$$

   Using the Matlab bode.m file, determine and plot the magnitude of the frequency response of $G(s)$ over the frequency range $\begin{bmatrix} 0 & 6 \end{bmatrix}$.

2. Let us first determine the effects of sampling on the frequency response. This means we want to determine the frequency response of the system given by the cascade of $G(s)$ and the sampling operator S. We can proceed as follows:

   (i) Choose the input $u(t)$ to be the Dirac delta function. The output $y(t)$ is then the impulse response $g(t)$. For simplicity, set $T = 1$ and sample the output at $t = kT,\ k \geq 0$. This produces the sequence $g(kT),\ k \geq 0$. Denote its z-transform by $G(z)$. Then

$G(e^{j\omega T})$ corresponds to the frequence response of the system given by $G(s)$ followed by the sampler S. Determine $G(z)$ in terms of $G(s)$. If you and your lab partner cannot get the expression for $G(z)$, ask the TA. In MATLAB, compute $G(z)$ using `c2d(sys,T,'impulse')`.

(ii) Use the function dbode in the form `dbode(sysd,T,w)`, plot the magnitude of the frequency response $G(e^{j\omega T})$ for w ranging over $\begin{bmatrix}0 & 10\end{bmatrix}$.

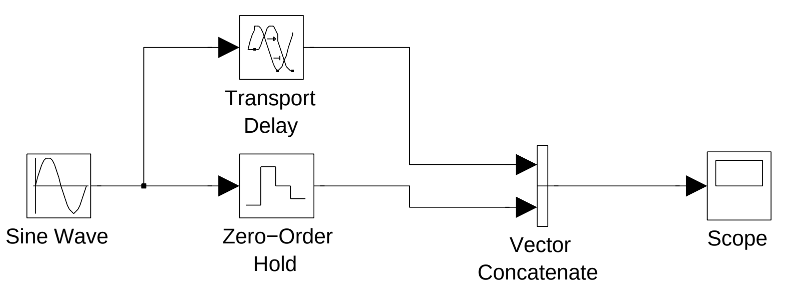
(iii) Compare the frequency response magnitude obtained in (ii) to that obtained in 1 and identify the aliasing that occurs.

3. We now include the effects of the hold operator as well, i.e., we want to determine the discretized transfer function $G_d(z)$, hence the frequency response $G_d(e^{j\omega T})$. We can do this by determining $(A_d, B_d) = c2d(A, B, T)$ and using `dbode(sysd,T,w)`. Plot the magnitude of the frequency response $G_d(e^{j\omega T})$, and display also the frequency responses obtained in 2(i) and 2(ii) in the same plot. Observe the differences among the 3 responses.

3.2 Effects of Sample and Hold on Control Design

As discussed in the lectures as well as in Appendix A, Chapter 3, the hold operator introduces a time delay effect on the input signal. If the sampling time is T, the hold operator can be approximated, at low frequency, by a time delay of $\frac{T}{2}$. In this section, we investigate what this implies in terms of control system analysis and design.

1. Bring up Simulink. Construct a simulink model similar to the one given below:



Choose the sampling time $T = 1$ in the zero-order hold. Choose the transport delay parameter to be $\frac{T}{2} = 0.5$, and choose the sine wave input to be $\sin t$. Run the simulation to show the 2 signals on the scope. Can you recognize which signal corresponds to which system? Is the delay a good approximation? Now vary the frequency of the sinusoid, increasing it and reducing it, and note the change in the closeness of the approximation.

2. Consider now a first order plant with transfer function $G_0(s) = \frac{1}{0.5s+1}$. We know that if we close the loop with a controller $u = k(r - y)$, where r is a reference input and $k > 0$ is the feedback gain, we get the closed loop transfer function to be

$$\frac{Y(s)}{R(s)} = \frac{k}{0.5s + 1 + k}$$

which is stable for all $k > 0$. Let us add a time delay of 0.5 to the open loop transfer function so that the plant model is now

$$G(s) = \frac{e^{-0.5s}}{0.5s + 1}$$

We will investigate stability properties of $G(s)$.

(i) You can produce a transfer function with time delay for Matlab stability analysis in the following way: In the Matlab command window, type

```
s = tf('s'); G = exp(-0.5*s)/(0.5*s+1)
```

Further usage details can be found by typing "help tf". Run nyquist(G) to plot the Nyquist plot for $G(s)$. What can you conclude? If we apply the same control law $u = k(r - y)$, will the closed loop system be stable for all $k > 0$? If not, can you estimate from the Nyquist plot the range of values of k for which closed loop stability is obtained?

(ii) Construct a simulink model with the open loop transfer function to be $G(s)$ and the feedback control law to be $u = k(r - y)$. Take r to be the unit step. Experimentally estimate the largest value of k beyond which the closed loop system becomes unbounded. How does this value compare with your previous estimate from the Nyquist plot? Change the reference input r to a sinusoid. Experimentally estimate the values of the frequency and gain k for the closed loop system to become unbounded. Compare this also to the values from the Nyquist plot.

(iii) Replace the transport delay block with a zero-order hold block with sampling time $T = 1$. Again experimentally investigate stability of the closed loop system.

(iv) Now discretize the originally continuous time plant $G_0(s)$ withe a sampling time of $T = 1$ using the Matlab command `c2d`. Use the same feedback law $u = k(r - y)$ for the discretized plant. Determine the range of value of k for closed loop stability.

3.3 Intersample Ripple

In this course, we focus on the design of the digital feedback control law to achieve desired closed loop properties for the discrete time system, i.e., control the behaviour of the system at the sampling instants. Usually, the values of the state and output between the samples will take care of themselves. However, it is possible, with poor design, to get undesirable response between the samples, even though the behaviour at the sampling instants appears good. This exhibits itself as the so-called "intersample ripple", undesirable oscillations between samples. In this part, you investigate such an example.

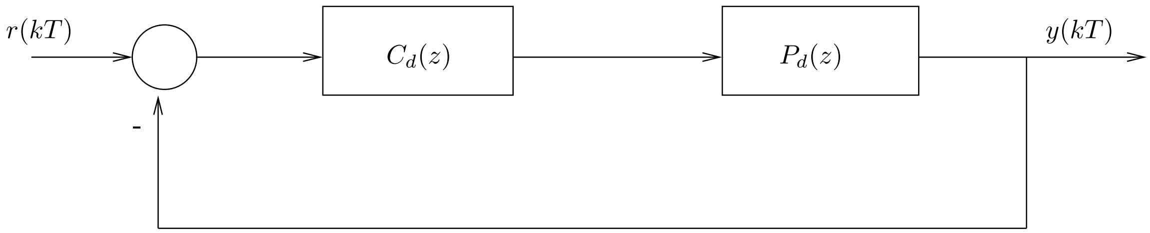
1. Consider a continuous time system having a transfer function

$$P(s) = \frac{0.1}{s(s + 0.1)}$$

Take the sampling time $T = 1$. Suppose we choose a digital control law given by

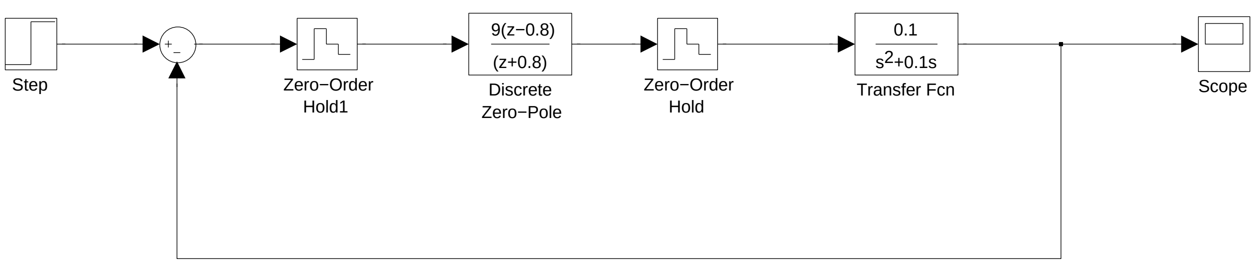
$$C_d(z) = 9\frac{z - 0.8}{z + 0.8}$$

First we discretize the continuous time plant to get $P_d(z)$. The discretized system is given by the following



Determine the output $y(kT)$ when the input $r(kT)$ is the discrete time unit step.

2. Construct the simulink model shown below as the sampled-data implementation of the digital controller for the original continuous time plant.



Run the simulation and examine the output on the scope. Compare it with the behaviour obtained from the discrete time closed loop system.