Computerized control - homework 2

Kjartan Halvorsen

Due 2015-09-03

1 Exercises

1.1 Sample the continuous-time transfer function

The harmonic oscillator from Homework 1

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x.\end{aligned}$$

has the transfer function

$$G(s) = C(sI - A)^{-1}B + D = \frac{s}{s^2 + \omega^2}.$$

Sampling the state space system with zero-order-hold gives the discrete-time state space system ($x(kh) = x(k)$)

$$\begin{aligned}x(k+1) &= \begin{bmatrix} \cos\omega h & \sin\omega h \\ -\sin\omega h & \cos\omega h \end{bmatrix} x(k) + \frac{1}{\omega}\begin{bmatrix} \sin\omega h \\ \cos\omega h - 1 \end{bmatrix} u(k), \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(k).\end{aligned}$$

1. **Compute the pulse-transfer function** for the discrete-time system from the state-space representation using the expression
$$H(z) = C(zI - \Phi)^{-1}\Gamma.$$
2. **Compute the pulse-transfer function** by sampling the transfer function $G(s)$.

1.2 Simulation of the continuous- and discrete-time harmonic oscillator

1.2.1 Simulate step responses

Use matlab's control toolbox or the python control module to simulate the systems. Use $\omega = 1$.

First, define the continuous-time system `sys_c` and the sampled system `sys_d` using the `ss` function. The example below uses the python control toolbox. Using the matlab control toolbox is very similar.

```
import numpy as np
import control.matlab as cm
import matplotlib.plot as plt

omega = 1.0
h = omega / 10

A = np.array([[0, omega], [-omega, 0]])
B = np.array([[1],[0]])
```

```
C = np.array([[1, 0]])
D = np.array([[0]])
sys_c = cm.ss(A,B,C,D)

wh = omega*h
F = np.array([[np.cos(wh), np.sin(wh)], [-np.sin(wh), np.cos(wh)]])
G = 1.0/omega* np.array([[np.sin(wh)], [np.cos(wh)-1]])
sys_d = cm.ss(F,G,C,D, h)

Tc = np.linspace(0,4/omega,200)
(yc,tc) = cm.step(sys_c, Tc)
Td = h*np.arange(40)
(yd,td) = cm.step(sys_d, Td)

plt.plot(tc,yc)
plt.plot(td,yd[0], '*')
```

Verify that the step response of the discrete-time system is equal to that of the continuous-time system at the sampling instants. Explain why this is so!

1.2.2 Sampling the system with help of the computer

Use the function `c2d` to sample your continuous-time system `sys_c`. **Verify that you get the same discrete-time system as your `sys_d` above**. *Hint*: Look at the system matrices returned by `ssdata`.

1.2.3 Compute the discrete step response yourself

Write some lines of code that solves the difference equation

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y = Cx(k)$$

given an initial state $x(0) = x_0$ and an input sequence $\{u(k)\}$. Use a step signal ($u(k) = 1$) and verify that your solution is the same as when using the `step` function.