ELEC-E8101 Digital and Optimal Control
Exercise 7
Solutions
Autumn 2022

1.a System discretization using forward difference approximation

In forward difference approximation,

$$s = \frac{1}{T_s}(z - 1).$$

See the lecture slides to understand why, if you don't remember.

Sampling time $T_s = 1$, thus

$$s = z - 1.$$

The system given was

$$G(s) = \frac{1}{10s+1}.$$

By substitution,

$$G(z) = \frac{1}{10z-10+1} = \frac{1}{10z-9}.$$

A stable continuous-time system may become unstable under this approximation, e.g. try

$$G(s) = \frac{1}{s+3}$$

1.b System discretization using backward difference approximation

In backward difference approximation,

$$s = \frac{1}{T_s}(1 - z^{-1}) = \frac{1}{T_s}\frac{z-1}{z}.$$

See the lecture slides to understand why, if you don't remember.

Sampling time $T_s = 1$, thus

$$s = 1 - z^{-1} = \frac{z-1}{z}.$$

The system given was

$$G(s) = \frac{1}{10s+1}.$$

By substitution,

$$G(z) = \frac{1}{10-10z^{-1}+1} = \frac{1}{11-10z^{-1}} = \frac{z}{11z-10}.$$

An unstable continuous-time system may become stable under this approximation, e.g.

$$G(s) = \frac{1}{2s-1}.$$

1.c System discretization using Tustin approximation

In Tustin approximation,

$$s = \frac{2}{T_s}\frac{z-1}{z+1}.$$

Sampling time $T_s = 1$, thus

$$s = 2\frac{z-1}{z+1}.$$

The system given was

$$G(s) = \frac{1}{10s+1}.$$

By substitution,

$$G(z) = \frac{1}{10\times 2\frac{z-1}{z+1}+1} = \frac{z+1}{21z-19}.$$

Stability of a continuous-time system is preserved by Tustin's approximation. However, frequency warping happens.

1.d Step invariance

Step response of a continuous time system is

$$g_{step}(t) = L^{-1}\left\{\frac{1}{s}G(s)\right\}$$

Sampling this gives the discrete response

$$g_{step}(k) = g_{step}(kT_s)$$

Step response of a discrete time system in z-domain is

$$Z(g_{step}) = \frac{z}{z-1}G(z)$$

Combining these and solving for $G(z)$ gives the step invariance approximation

$$G(z) = \frac{z-1}{z}Z\left\{L^{-1}\left\{\frac{1}{s}G(s)\right\}_{t=kT_s}\right\}$$

Substituting $G(s)$ and T_s and solving the inverse Laplace transform gives

$$G(z) = \frac{z-1}{z} Z\left\{1 - e^{-\frac{1}{10}k}\right\}$$

Solving the Z-transform gives

$$G(z) = \frac{z-1}{z}\left(\frac{z}{z-1} - \frac{z}{z-e^{-\frac{1}{10}}}\right)$$

By algebra

$$G(z) = 1 - \frac{z-1}{z-e^{-\frac{1}{10}}} = \frac{z-e^{-\frac{1}{10}}-z+1}{z-e^{-\frac{1}{10}}} = \frac{1-e^{-\frac{1}{10}}}{z-e^{-\frac{1}{10}}}$$

This method does not preserve stability.

2 Discretization of a PI controller

2.a Tustin approximation and parameter correspondence

The continuous-time PI-controller has the transfer function

$$G(s) = K(1 + \frac{1}{T_I s}).$$

Substituting the Tustin approximation

$$G(z) = K\left(1 + \frac{T_s(z+1)}{2T_I(z-1)}\right)$$

Using partial fractions

$$G(z) = K\left(1 + \frac{T_s}{2T_I}\left(\frac{z-1+2}{2(z-1)}\right)\right) = K\left(1 + \frac{T_s}{2T_I} + \frac{T_s}{T_I}\frac{1}{z-1}\right)$$

$$= K\left(1 + \frac{T_s}{2T_I}\right)\left(1 + \frac{2T_I}{2T_I+T_s}\frac{T_s}{T_I}\frac{1}{z-1}\right)$$

$$= K\left(1 + \frac{T_s}{2T_I}\right)\left(1 + \frac{T_S}{(T_I+T_s/2)(z-1)}\right)$$

Noting that basic discrete-time PI-controller is written

$$U(z) = K_d\left(1 + \frac{T_s}{T_{I,d}(z-1)}\right)$$

we can determine the parameter correspondence

$$K_d = K\left(1 + \frac{T_s}{2T_I}\right)$$

$$T_{I,d} = T_I + \frac{T_s}{2}$$

Thus, the gain is increased by a factor that reduces as the sample time reduces. The time constant of the integration behaves in a similar fashion.

2.b Incremental algorithm

Discrete PI controller in time domain can be written (dropping the sampling time / assuming $T_s = 1$ for notational simplicity)

$$u(k) = Ke(k) + \frac{K}{T_I}\sum_{i=0}^{k} e(i)$$

A trivial (non-incremental) PI-controller implementation would calculate the sum from *k=0* until current *k* every time the control is used. This wastes a lot of computation power and is not practical.

This can be overcome by either storing the sum (cumulative error) at the current step and updating it at each step by adding the error of that step to the current sum.

Another choice is the so-called incremental algorithm which stores the control across steps and determines the change in the control $\Delta u(k)$ instead of its value at each step. That is, at every time step,

$$u(k) = u(k - 1) + \Delta u(k)$$

It turns out, that calculation of $\Delta u(k)$ does not require calculation of the sum.
This is because

$$\Delta u(k) = u(k) - u(k - 1) = Ke(k) + \frac{K}{T_I}\sum_{i=0}^{k} e(i) - Ke(k - 1) - \frac{K}{T_I}\sum_{i=0}^{k-1} e(i)$$

$$= K(e(k) - e(k - 1)) + \left(\frac{K}{T_I}\sum_{i=0}^{k} e(i) - \frac{K}{T_I}\sum_{i=0}^{k-1} e(i)\right) = K(e(k) - e(k - 1)) + \frac{K}{T_I}e(k)$$

Note that only the most recent *e(k)* affects $\Delta u(k)$. Thus, there is no need to calculate the sum, as it is implicitly calculated when summing the $\Delta u(k)$ to the previous $u(k)$.