Discrete-time systems: discretization, models and their properties

Two main design approaches: a. discretize the analog controller, b. discretize the process and do the design totally in discrete time

Let us consider the design approach b: A discrete system from the controller viewpoint
A continuous function $f(t)$

$$f(t) = 2e^t$$

Set of integers $Z$

$$Z = \{..., -1, 0, 1, \ldots \}$$

Sampling instants $\{ t_k : k \in Z \}$

$$\{0, 1, 2, 3, \ldots \}$$

Sequence $\{ f(t_k) : k \in Z \}$

$$\{2, 2e, 2e^2, 2e^3, \ldots \}$$

Approximately $\{2.00, 5.44, 14.78, 40.17, \ldots \}$

---

**Zero-order-hold ZOH and sampling**

A continuous process is described by a linear state-space-representation.

$$\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + Du(t)
\end{align*}$$

$\dim\{u\} = r$

$\dim\{x\} = n$

$\dim\{y\} = p$

---

**Sampling of continuous-time signals**

Periodic sampling

Sampling interval, $h$

$$t_k = k \cdot h$$

Sampling frequency, $f_s$

$$f_s = \frac{1}{h} \text{ (Hz)}$$

Sampling angular frequency $\omega_s$

$$\omega_s = \frac{2\pi}{h} = \frac{2\pi \cdot f_s}{(\text{rad} / \text{s})}$$

Nyquist frequency, $f_N$

$$f_N = \frac{1}{2h} \text{ (Hz)}$$

$$\omega_N = \frac{2\pi}{(2h) / \pi} = \frac{2\pi}{h} = \frac{2\pi \cdot f_N}{(\text{rad} / \text{s})}$$

---

**The matrix exponential**

Let $A$ be a square matrix, define

$$e^{At} = I + tA + \frac{1}{2}t^2A^2 + \frac{1}{6}t^3A^3 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!}t^nA^n$$

which is always convergent.

From the definition

$$\frac{de^{At}}{dt} = Ae^{At} = e^{At}A$$
The solution to the homogenous set of differential equations
\[ \dot{x}(t) = Ax(t), \quad x(0) = x_0 \]
is then
\[ x(t) = e^{At}x(0) = e^{At}x_0 \]
The term \( e^{At} \) is called the state transition matrix (in this context).

The solution by using the Laplace transformation
\[ \dot{x}(t) = Ax(t), \quad x(0) = x_0 \quad sX(s) - x_0 = AX(s) \]
It follows
\[ (sI - A)X(s) = x_0 \]
\[ (sI - A)^{-1}(sI - A)X(s) = (sI - A)^{-1}x_0 \]
\[ X(s) = (sI - A)^{-1}x_0 \]
The solution is obtained by the inverse transformation.

Solution of the state equation
Solution for the input-output representation
\[ \dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \]
\[ y(t) = Cx(t) \]
is
\[ x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \]
\[ y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau \]
Solution of the state equation

in which $e^{At}$ is the above state transition matrix.

To prove the solution check the initial condition and differentiate the solution to see that the original differential equation holds.

Zero-order hold ZOH and sampling

At the next sampling instant

$$ t = t_{k+1} \implies \begin{cases} x(t_{k+1}) = \Phi(t_{k+1} - t_k) x(t_k) + \Gamma(t_{k+1} - t_k) u(t_k) \\ y(t_k) = C x(t_k) + D u(t_k) \\ \Phi(t_{k+1} - t_k) = e^{A(t_{k+1} - t_k)} \\ \Gamma(t_{k+1} - t_k) = \int_{t_k}^{t_{k+1}} e^{As} B \, ds \end{cases} $$

Zero-order hold ZOH and sampling

By a periodic sampling the equations become

$$ t_k = k \cdot h \implies t_{k+1} - t_k = h \quad \text{(constant)} $$

$$ \begin{cases} x(kh + h) = \Phi(h) x(kh) + \Gamma(h) u(kh) \\ y(kh) = C x(kh) + D u(kh) \\ \Phi(h) = e^{Ah} \\ \Gamma(h) = \int_0^h e^{As} B \, ds \end{cases} $$

 Usually this is written in the form ($h$ is constant)

$$ \begin{cases} x(k + 1) = \Phi x(k) + \Gamma u(k) \\ y(k) = C x(k) + D u(k) \end{cases} $$
How to solve for $\Phi$ and $\Gamma$?

- Symbolically
  Eg. by using the Laplace transformation
  By symbolic programs (Maple, Mathematica, …)
- Numerically
  Eg. by the series expansion of the matrix exponential function
  By numeric software (Matlab)

Example. Discretization by direct calculus
Sampling interval $h = 0.1$

\[
\dot{x}(t) = 2x(t) + u(t) \quad \text{is of the form} \quad \dot{x}(t) = Ax(t) + Bu(t) \\
y(t) = 3x(t) \\
\Phi = e^{Ah} = e^{2.01} = e^{0.2} \\
\Gamma = \int_0^h e^{As}ds \cdot B = \int_0^h e^{2s}ds = \frac{1}{2} \int_0^h e^{2s}ds = \frac{1}{2} \left( e^{2.01} - e^0 \right) = \frac{1}{2} \left( e^{0.2} - 1 \right) \\
\begin{align*}
    x(kh + h) &= \Phi x(kh) + \Gamma u(kh) \\
    y(kh) &= Cx(kh)
\end{align*}
\]

Example. Discretization by the series expansion

The corresponding discrete-time model becomes

\[
\begin{align*}
    x(kh + h) &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(kh) + \begin{bmatrix} \frac{1}{2} h^2 \\ h \end{bmatrix} u(kh) \\
    y(kh) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(kh)
\end{align*}
\]
Example. Discretization by using the Laplace transformation

State-space representation of the DC motor

\[
\begin{align*}
    \dot{x}(t) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \\
    y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(t)
\end{align*}
\]

\[\Phi = e^{A_h} - e^{A_h} L^{-1}\left(\left(sI - A\right)^{-1}\right)\rvert_{s=A_h}\]

\[
\begin{align*}
    \Phi &= L^{-1}\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right] \\
    &= L^{-1}\left[\begin{bmatrix} 1 \\ e^{-h} \end{bmatrix} \right] \\
    &= \begin{bmatrix} 1 & e^{-h} \\ 1 & 1 - e^{-h} \end{bmatrix}
\end{align*}
\]

Pulse transfer function, \(H(z)\)

A pulse transfer function \(H(z)\) can also be calculated directly from \(G(s)\).

\[
\begin{align*}
    \{u(kh)\} &\rightarrow \text{ZOH} \rightarrow \{y(kh)\} \\
    u(t) &\rightarrow \text{continuous process} \rightarrow \text{sampling} \\
    y(t) &\rightarrow \text{discrete process} \rightarrow \text{sampling}
\end{align*}
\]

Example. Discretization by the Laplace ....

\[
\begin{align*}
    \Gamma &= \frac{1}{h}\int_{0}^{h} e^{s} ds B = \left[\begin{array}{cc} 1 & 0 \\ h & 1 \end{array}\right] = \left[\begin{array}{cc} 1 & e^{-h} \\ h & 1 + e^{-h} \end{array}\right] \\
    \text{The discrete-time model is obtained}
\end{align*}
\]

\[
\begin{align*}
    x(kh+h) &= \begin{bmatrix} 1 & 0 \\ 1 - e^{-h} & 1 \end{bmatrix} x(kh) + \begin{bmatrix} 1 - e^{-h} \\ h - 1 + e^{-h} \end{bmatrix} u(kh) \\
    y(kh) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(kh)
\end{align*}
\]

Pulse transfer function, \(H(z)\)

After ZOH \(u(t)\) consists of step functions. The output \(y(t)\) is a set of step responses, which are sampled at constant intervals \(h\).

Discrete pulse transfer function and continuous transfer function correspond to each other, if the outputs are equal at the sampling instants. The step response of the continuous system at the sampling instants \((t = kh)\)

\[
\begin{align*}
    y(kh) &= y(t)\rvert_{t=kh} = L^{-1}\left[\begin{array}{c} Y(s) \\ G(s) \end{array}\right]\rvert_{s=kh} = L^{-1}\left[\begin{array}{c} G(s)Y(s) \\ G(s)U(s) \end{array}\right]\rvert_{s=kh} \\
    &= L^{-1}\left[\begin{array}{c} 1 \\ s \end{array}\right] = \begin{bmatrix} 1 - e^{-h} \\ h - 1 + e^{-h} \end{bmatrix} u(kh)
\end{align*}
\]
Pulse transfer function, $H(z)$

The step response of the discrete system

$$y(kh) = Z^{-1}\{Y(z)\} = Z^{-1}\{H(z)U(z)\} = Z^{-1}\left\{ \frac{H(z)}{z-1} \right\}$$

Make these equal; the final result follows

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

$$H(z) = \frac{1}{z} \cdot Z^{-1}\left\{ \frac{G(s)}{s} \right\}$$

Can a pulse sequence be used to construct the original continuous signal?

Two different continuous signals can be fitted to the pulse train ($h = 1$)

- $y_1(t) = \sin(0.2\pi t)$
- $y_2(t) = \sin(1.8\pi t)$

Actually, an indefinite number of continuous signals can be fitted.

Is it possible to find an ZOH-equivalent continuous model for a given discrete model?

Consider a simple example with one state variable only

$$\begin{align*}
x(k+1) &= \Phi x(k) + \Gamma u(k) \\
y(k) &= Cx(k) \\
\Phi &= e^{\Phi h} \\
\Gamma &= \int e^{\Phi s} \frac{B}{A} e^{\Phi h} - 1 ds
\end{align*}$$

If $\Phi$ is positive, a corresponding continuous model exists. For negative $\Phi$s the logarithm is not real, and the model does not exist.

Is it possible to find an ZOH-equivalent continuous model for a given discrete model?

A more general case (multivariable system of $n$th order): The result is known, but it is difficult to express. Sometimes you can find the following in the literature:

"If matrix $\Phi$ has even one eigenvalue on the negative real axis, the corresponding continuous-time system doesn’t exist."

The above claim is incorrect however. (Take $G(s) = \frac{1}{s^2 + 1}$, do a state-space realization and discretize by using the sampling interval $h = \pi$.)

Moreover: the continuous time model corresponding to a discrete model is not always unique. (Take the harmonic oscillator $G(s) = \frac{s^2}{s^2 + \omega^2}$, do a realization and discretize by using $\omega = \alpha + n2\pi/h$, $n = 0, 1, 2, ...$)
Discretization of systems with delay

With discretization an infinite dimensional (continuous) model becomes finite dimensional (discrete). Analysis and controller design of such systems then becomes easier.

Consider first the case, in which the delay is shorter than the sampling interval ($t < h$)

$$\begin{align*}
    \dot{x}(t) &= Ax(t) + Bu(t - \tau) \\
y(t) &= Cx(t) + Du(t)
\end{align*}$$

The solution can be formed as

$$x(kh + h) = e^{A_h}x(kh) + \int_{kh}^{kh+h} e^{A_{s-h}}B u(s - \tau)ds'$$

Even though $u(t)$ is constant over the integration interval, $u(t-t)$ is not!

Divide into two intervals, where the control signal is constant.

$$t_1 \in [kh , kh + t]$$
$$t_2 \in [kh + t , kh + h]$$

The system model becomes

$$x(kh + h) = \Phi x(kh) + \Gamma_1 u(kh) + \Gamma_2 u(kh - h)$$
Discretization of systems with delay

To form a general state representation the delayed controls can be chosen as additional state components (augmentation of state)

\[
\mathbf{x}_d(kh) = \mathbf{u}(kh) \Rightarrow \mathbf{x}_d(kh+h) = \mathbf{u}(kh) \quad \mathbf{x}_d(kh) = \begin{bmatrix} \mathbf{x}(kh) \\ \mathbf{x}(kh) \end{bmatrix}
\]

The new state representation is

\[
\mathbf{x}_d(kh+h) = \begin{bmatrix} \Phi_d & \Gamma_d \\ 0 & 0 \end{bmatrix} \mathbf{x}_d(kh) + \begin{bmatrix} \Gamma_0 \\ 1 \end{bmatrix} \mathbf{u}(kh)
\]

Example. Double integrator with delay

The system matrix \( \Phi \) was discretized earlier

\[
\begin{aligned}
\dot{\mathbf{x}}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t)
\end{aligned}
\]

\[
\mathbf{\Phi}(\tau) = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}
\]

\[
\begin{aligned}
\Gamma_1 &= \int_0^{\frac{\tau}{h}} e^{s} ds = \begin{bmatrix} \frac{h \tau}{1} \\ 1 \end{bmatrix} \\ \Gamma_0 &= \int_0^{\frac{\tau}{h}} s ds = \frac{h \tau^2}{2}
\end{aligned}
\]

Example. Double integrator with delay

By using state augmentation, the general representation follows

\[
\begin{aligned}
\mathbf{x}_d(kh+h) &= \begin{bmatrix} \Phi_d & \Gamma_d \\ 0 & 0 \end{bmatrix} \mathbf{x}_d(kh) + \begin{bmatrix} \Gamma_0 \\ 1 \end{bmatrix} \mathbf{u}(kh) \\ \mathbf{y}(kh) &= \begin{bmatrix} C & 0 \end{bmatrix} \mathbf{x}_d(kh)
\end{aligned}
\]

\[
\begin{aligned}
\mathbf{x}_d(kh+h) &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \mathbf{x}_d(kh) + \begin{bmatrix} \frac{h \tau}{1} \\ 1 \end{bmatrix} \mathbf{u}(kh) \\ \mathbf{x}_d(kh) &= \begin{bmatrix} \frac{h \tau}{1} \\ 1 \end{bmatrix} \mathbf{u}(kh)
\end{aligned}
\]

\[
\begin{aligned}
\mathbf{x}_d(kh+h) &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \mathbf{x}_d(kh) + \begin{bmatrix} \frac{h \tau}{1} \\ 1 \end{bmatrix} \mathbf{u}(kh) \\ \mathbf{y}(kh) &= \begin{bmatrix} C & 0 \end{bmatrix} \mathbf{x}_d(kh)
\end{aligned}
\]
Example. Double integrator with delay

\[
\begin{align*}
\mathbf{x}_t(kh + h) &= 
\begin{bmatrix}
1 & h & \tau(h - \frac{1}{2} \tau) \\
0 & 1 & \tau \\
0 & 0 & 0
\end{bmatrix}
\mathbf{x}_t(kh) + 
\begin{bmatrix}
\frac{1}{2} \tau \\
-\tau \\
1
\end{bmatrix}
\mathbf{u}(kh) \\
y(kh) &= 
\begin{bmatrix}
1 & 0
\end{bmatrix}
\mathbf{x}_t(kh)
\end{align*}
\]

Long delay

If the delay is longer than the sampling interval \((t > h)\), the formulas must be modified a bit:

\[
\tau = (d - 1)h + \tau' \quad 0 < \tau' \leq h \quad d \in \mathbb{Z}_+ = \{1, 2, 3, \ldots\}
\]

A similar result as in the previous case can be derived where the coefficients are as earlier but \(\tau'\) is used instead of \(\tau\)

The delay terms can be taken as augmented state variables

Example. “paper machine” model

\[
h = 1, d = 3, \tau = 2.6, \tau' = 0.6 \quad \dot{x}(t) = -x(t) + u(t - 2.6) \quad \Phi = e^{-1} \approx 0.3679
\]

\[
\Gamma_1 = e^{(h-1)\tau} \int_0^{h-1} e^{s\tau} dsB = e^{0.6} \int_0^{0.6} e^{-s} ds = e^{-0.4} - e^{-1} \approx 0.3024
\]

\[
\Gamma_2 = \int_0^{h-1} e^{s\tau} dsB = \int_0^{0.6} e^{-s} ds = 1 - e^{-0.4} \approx 0.3297
\]

A simple model is obtained

\[
x(kh + h) = 0.3679x(kh) + 0.3297u(kh - 2h) + 0.3024u(kh - 3h)
\]
Example. “paper machine” model

\[
\begin{bmatrix}
0.3679 & 0.3024 & 0.3297 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x(kh) \\
x(kh + h) \\
x(kh + 2h) \\
x(kh + 3h)
\end{bmatrix}
+
\begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix}

y(kh) = [1 \ 0 \ 0 \ 0] x(kh)

\[
x(kh) =
\begin{bmatrix}
x(kh) \\
x(kh - h) \\
x(kh - 2h) \\
x(kh - 3h)
\end{bmatrix}
\]

Discrete time systems

\[
\begin{aligned}
x(k + 1) &= \Phi x(k) + \Gamma u(k), \text{ initial value } x(k_0) \\
y(k) &= C x(k) + D u(k)
\end{aligned}
\]

Solution by direct recursive calculus starting from a given initial state \(x(k_0)\) up to any time \(k\).

\[
\begin{aligned}
x(k + 1) &= \Phi x(k) + \Gamma u(k) \\
x(k + 2) &= \Phi^2 x(k) + \Phi \Gamma u(k) + \Gamma u(k + 1) \\
&\vdots \\
x(k) &= \Phi^k x(k_0) + \Phi^{k-1} \Gamma u(k_0) + \cdots + \Gamma u(k - 1)
\end{aligned}
\]

Input-Output-models, I/O-models, pulse response

Input and output signals as sequences

\[
\begin{align*}
U &= [u(0) \ u(1) \ \cdots \ u(N-1)]^T \\
Y &= [y(0) \ y(1) \ \cdots \ y(N-1)]^T
\end{align*}
\]

A linear model giving the relationship between \(Y\) and \(U\) is

\[
Y = H U + Y_s
\]

For a causal system \(H\) is a lower triangular matrix (element \(h(k, m)\))

\[
y(k) = \sum_{m=0}^{k} h(k, m) u(m) + y_s(k)
\]

\(h(k, m)\) is the pulse response or weighting function.

Input-Output-models, I/O-models, pulse response

For a time-invariant system the pulse response depends only on \(k-m\):

\[
h(k, m) = h(k-m)
\]

For a state representation the pulse response is calculated as

\[
\begin{aligned}
x(k) &= \Phi^k x(k_0) + \sum_{j=0}^{k-1} \Phi^{k-j-1} \Gamma u(j) \\
y(k) &= C x(k) + D u(k)
\end{aligned}
\]

\[
\begin{bmatrix}
0 & k < 0 \\
D & k = 0 \\
C \Phi^{k+1} & k \geq 1
\end{bmatrix}
\]

(assume \(k_0 = 0\))
Linear time-invariant systems: The pulse transfer function, weighting function, pulse response and convolution sum

\[
Y(z) = H(z)U(z)
\]

Input-output: Z-domain

\[
y(k) = \sum_{i=0}^{p} h(k-i)u(i) + \sum_{j=-\infty}^{k} s(j)u(k-j)
\]

Input-output: time domain, convolution sum

The z-transform of the weighting function \(h(k)\) is the pulse transfer function \(H(z)\).

The (im)pulse response coincides with the weighting function from the time that the pulse enters.

Shift-operators

Corresponding to the differential operator \(p\) used in continuous systems a (forward) shift operator \(q\) is defined for discrete systems

\[
q \cdot f(k) = f(k+1)
\]

\[
q^{-1} \cdot f(k) = f(k-1)
\]

By using the shift operator input-output relationships (difference equations) can easily be described

\[
y(k + n_q) + a_1y(k + n_q - 1) + \cdots + a_d y(k) = b_0u(k) + b_1u(k + n_q) + \cdots + b_d u(k)
\]

\[
(q^{n_q} + a_1q^{n_q-1} + \cdots + a_d) y(k) = (b_0q^{n_q} + b_1q^{n_q-1} + \cdots + b_d) u(k)
\]

Shift-operators

Difference equations can be described by polynomials

\[
A(z) = z^n + a_1z^{n-1} + \cdots + a_n
\]

\[
B(z) = b_0z^n + b_1z^{n-1} + \cdots + b_n
\]

A backward shift operator \(q^{-1}\) can also be used

\[
y(k) + a_1y(k-1) + \cdots + a_d y(k-n_d) = b_0u(k-d) + \cdots + b_{n_d} u(k-d-n_d)
\]

\[
(1 + a_1q^{-1} + \cdots + a_d q^{-n_d}) y(k) = (b_0 + b_1q^{-1} + \cdots + b_{n_d} q^{-n_d}) u(k)
\]
Pulse-transfer operator, $H(q)$

The pulse transfer operator is an I/O-representation obtained by eliminating internal variables. E.g. from state representation

\[
\begin{align*}
\dot{x}(k+1) &= \Phi x(k) + \Gamma u(k) \\
y(k) &= Cx(k) + Du(k)
\end{align*}
\]

\[
\begin{align*}
(qI - \Phi)x(k) &= \Gamma u(k) \\
y(k) &= Cx(k) + Du(k)
\end{align*}
\]

\[
y(k) = (C(qI - \Phi)^{-1}\Gamma + D)u(k) = H(q)u(k)
\]

Pulse-transfer operator, $H(q)$

From polynomial representations

\[
A(q)y(k) = B(q)u(k) \quad H(q) = \frac{B(q)}{A(q)}
\]

or

\[
A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) \quad H'(q^{-1}) = \frac{q^{-d}B'(q^{-1})}{A'(q^{-1})}
\]

Representation by reciprocal polynomials

\[
H'(q^{-1}) = H(q)
\]

Pulse-transfer operator, $H(q)$

Consider a simple example of a pulse transfer function.

\[
\begin{align*}
x(k + 1) &= 0.5x(k) + 0.5u(k) \\
y(k) &= 2x(k)
\end{align*}
\]

We obtain

\[
H(q) = C(qI - \Phi)^{-1}\Gamma + D = \frac{2 \cdot 0.5}{q - 0.5} = \frac{1}{q - 0.5}
\]

Starting from IO-difference equation the calculation is equally easy

Pulse-transfer operator, $H(q)$

\[
\begin{align*}
x(k + 1) &= \frac{1}{2}y(k) \\
y(k + 1) &= 0.5y(k) + 0.5u(k)
\end{align*}
\]

Form $A$- and $B$-polynomials

\[
(0.5q - 0.25)y(k) = 0.5u(k) \quad (q - 0.5)y(k) = u(k)
\]

\[
A(q)y(k) = B(q)u(k)
\]

\[
A(q) = q - 0.5 \quad B(q) = 1
\]

The pulse transfer operator is

\[
H(q) = \frac{B(q)}{A(q)} = \frac{1}{q - 0.5}
\]
Pulse-transfer operator, $H(q)$

Let us calculate the same with the $q^{-1}$-operator

\[ 0.5y(k+1) = 0.25y(k) + 0.5u(k) \quad 0.5y(k) = 0.25y(k-1) + 0.5u(k-1) \]

\[ (0.5-0.25q^{-1})y(k) = 0.5q^{-1}u(k) \quad (1-0.5q^{-1})y(k) = 1-u(k-1) \]

\[ A'(q^{-1})y(k) = q^{-d}B'(q^{-1})u(k) \]

The result becomes

\[ H'(q^{-1}) = \frac{q^{-d}B'(q^{-1})}{A'(q^{-1})} = \frac{q^{-1} \cdot 1}{1-0.5q^{-1}} = \frac{1}{q - 0.5} \]

Poles and zeros

Some basics from matrix calculus. For the pulse transfer operator we can write

\[ H(q) = C(qI - \Phi)^{-1} + D = \frac{C \cdot \text{adj}(ql - \Phi) \Gamma + D \cdot \text{det}(ql - \Phi)}{\text{det}(ql - \Phi)} \]

where 'adj' means the adjugate matrix. For the matrix $\Phi$ the eigenvalues $\lambda_i$ and eigenvectors $e_i$ are defined as follows

\[ \Phi e_i = \lambda_i e_i \Rightarrow (\lambda_i I - \Phi) e_i = 0 \]

There exist non-zero eigenvectors when $\text{det}(\lambda_i I - \Phi) = 0$
Poles and zeros

The poles are the roots of the characteristic polynomial. They belong to the set of eigenvalues of the system matrix $\Phi$.

Consider the polynomial

$$f(\lambda) = \alpha_n \lambda^n + \alpha_{n-1} \lambda^{n-1} + \cdots + \alpha_0$$

The corresponding function for the square matrix $A$ is defined as

$$f(A) = \alpha_n A^n + \alpha_{n-1} A^{n-1} + \cdots + \alpha_0 I$$

Result: Let the eigenvalues of $A$ be $\lambda_i$ and the corresponding eigenvectors $e_i$. It holds

$$f(A)e_i = f(\lambda_i)e_i$$

meaning that $f(\lambda)$ is the eigenvalue of $f(A)$ and the corresponding eigenvector is $e_i$.

The proof is straightforward by writing the above formula and using repeatedly the definitions of eigenvalue and eigenvector $\Rightarrow$ Exercise.

Poles and zeros

The poles of a discrete system are the zeros of the denominator of $H(z)$. The zeros are the zeros of the numerator. The poles are also eigenvalues of the system matrix $\Phi$.

From the location of poles in the complex plane stability, oscillations and speed of the system can be deduced.

The poles of a continuous $n$th order system are mapped to the discrete system poles according to:

$$x(t) = Ax(t) + Bu(t) \quad \text{poles: } \lambda_i(A), \quad i = 1, \ldots, n$$

$$y(t) = Cx(t)$$

Discrete system

$$\begin{cases}
x(kh+h) = \Phi x(kh) + \Gamma u(kh) \\
y(kh) = Cx(kh)
\end{cases}$$

poles: $\lambda_i(\Phi), \quad i = 1, \ldots, n$

$$\Phi = e^{A_h} \Rightarrow \lambda_i(\Phi) = e^{\lambda_i(A)h}$$

A simple relationship for the mapping of zeros does not hold. Even the number of zeros does not necessarily remain invariant. The mapping of zeros is a complicated issue.
Unstable inverse, non-minimum phase systems

- A continuous time system is non-minimum phase, if has zeros on the right half plane (RHP) or if it contains a delay.
- A discrete system has an unstable inverse, if it has zeros outside the unit circle.
- Zeros are not mapped in a similar way as poles, so a minimum phase continuous system may have a discrete counterpart with an unstable inverse and a non-minimum phase continuous system may have a discrete counterpart with a stable inverse.

Selection of the sampling rate

- The proper choice of the sampling interval is very important. Too low sampling frequency may lose so much information that the control performance deteriorates and the system dynamics is lost.
- Too high sampling rate increases the burden of the processor; also, it may lead to discrete representation with bad numerical properties.
- For oscillating systems the sampling interval is often tied to the frequency of the dominating oscillation. For damped systems the sampling interval is usually chosen to be in relation to the time constant.
Selection of the sampling rate

$N_r$ means the amount of samples during the rise time.

$$N_r = \frac{T_r}{h}$$

For a self-oscillating system (2nd order, damping ratio $\zeta$ and natural frequency $w_0$) the rise time is:

$$T_r = \frac{1}{\sqrt{1 - \zeta^2} e^{\frac{\zeta}{2}}}, \quad \zeta = \cos \phi$$

A usual sampling rate is $N_r = \frac{T_r}{h} = 4 \ldots 10$

Selection of the sampling rate

Sampling examples for a sinusoidal and exponential signal.

a. $N_r = 1$

b. $N_r = 2$

c. $N_r = 4$

d. $N_r = 8$