3.1: The \( z \) transform

- Computer control requires analog-to-digital (A2D) and digital-to-analog (D2A) conversion.
- Everything shown in the dashed box below represents the internals of a computer or embedded system in a digital control scenario.

- The A2D comprises a sampler and quantizer; the D2A performs a zero-order-hold operation.
- The controller is designed in discrete time, where the \( z \) transform is used as a design tool rather than the Laplace transform.

**DEFINITION:** The \( z \) transform is defined for a signal \( x(kT) \) to be

\[
X(z) = x(0) + x(T)z^{-1} + x(2T)z^{-2} + x(3T)z^{-3} + \cdots
\]

\[
= \sum_{k=0}^{\infty} x(kT)z^{-k}
\]

(Or, \( X(z) = \sum_{k=0}^{\infty} x[k]z^{-k} \).)
As with the Laplace-transform variable “s,” the z-transform variable “z” is a complex number.

I assume that you have studied the z transform before, and so include only a short review in these notes.

**EXAMPLE:** Define the digital impulse (pulse) function to be

\[
\delta[k] = \begin{cases} 
1, & k = 0; \\
0, & \text{otherwise}.
\end{cases}
\]

Note: This is very different from an analog impulse (e.g., \(\delta[0]\) is defined), but plays a similar role.

so, let \(x[k] = \delta[k]\)

\[
X(z) = \sum_{k=0}^{\infty} x[k]z^{-k}
\]

\[
= x[0] + x[1]z^{-1} + x[2]z^{-2} + \cdots
\]

\[
= x[0] = 1.
\]

This sum converges regardless of the value of z, so ROC = \(|z| > 0\).

**Selected important properties of the z transform**

Some select properties of the z transform may be useful.

**LINEARITY:** The z transform is linear.

If \(x[k] \leftrightarrow X(z)\) and \(v[k] \leftrightarrow V(z)\), then

\[
a x[k] + b v[k] \leftrightarrow a X(z) + b V(z).
\]
RIGHT SHIFT IN TIME (DELAY): When shifting a signal in time,
\[ x[k - 1] \iff z^{-1}X(z) + x[-1] \]
\[ x[k - 2] \iff z^{-2}X(z) + x[-2] + z^{-1}x[-1] \]
\[ : \]
\[ x[k - q] \iff z^{-q}X(z) + x[-q] + z^{-1}x[-q + 1] + \cdots + z^{-q+1}x[-1]. \]

CONVOLUTION: Discrete-time convolution is analogous to continuous-time convolution:
\[ (x * v)[k] = \sum_{i=0}^{k} x[i]v[k - i] \quad k \geq 0 \]

or,
\[ = \sum_{i=0}^{k} v[i]x[k - i]. \quad k \geq 0. \]

- Then, assuming that \( x[k] \) and \( v[k] \) are zero for negative \( k \),
\[ (x * v)[k] \iff X(z)V(z). \]

INITIAL-VALUE THEOREM: If \( x[k] \iff X(z) \), \( x[0] = \lim_{z \to \infty} X(z). \)

FINAL-VALUE THEOREM: If a finite final value exists (all poles of \( X(z) \) strictly inside the unit circle except perhaps for a single pole on the unit circle at \( z = 1 \)),
\[ \lim_{k \to \infty} x[k] = \lim_{z \to 1} (z - 1)X(z). \]

- The table on the next page lists \( z \) transforms for commonly encountered sampled time-domain signals:
### Example table showing \( z \)-transform of sampled time-domain signals:

<table>
<thead>
<tr>
<th>Time function</th>
<th>Laplace transform</th>
<th>( z )-Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e(t) )</td>
<td>( E(s) )</td>
<td>( E(z) )</td>
</tr>
<tr>
<td>( 1(t) )</td>
<td>( \frac{1}{s} )</td>
<td>( \frac{z}{z-1} )</td>
</tr>
<tr>
<td>( t1(t) )</td>
<td>( \frac{1}{s^2} )</td>
<td>( \frac{Tz}{(z-1)^2} )</td>
</tr>
<tr>
<td>( \frac{t^2}{2}1(t) )</td>
<td>( \frac{1}{s^3} )</td>
<td>( \frac{T^2z(z+1)}{2(z-1)^3} )</td>
</tr>
<tr>
<td>( t^{k-1}1(t) )</td>
<td>( \frac{(k-1)!}{s^k} )</td>
<td>( \lim_{a \to 0} (-1)^k \frac{a^{k-1}}{\theta^{k-1}} \left[ \frac{z-1}{z-e^{-at}} \right] )</td>
</tr>
<tr>
<td>( e^{-at}1(t) )</td>
<td>( \frac{1}{s+a} )</td>
<td>( \frac{z}{z-e^{-at}} )</td>
</tr>
<tr>
<td>( te^{-at}1(t) )</td>
<td>( \frac{1}{(s+a)^2} )</td>
<td>( \frac{Tze^{-at}}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( t^k e^{-at}1(t) )</td>
<td>( \frac{(k-1)!}{(s+a)^k} )</td>
<td>( (-1)^k \frac{a^k}{\theta^k} \left[ \frac{z}{z-e^{-at}} \right] )</td>
</tr>
<tr>
<td>( 1 - e^{-at} )</td>
<td>( \frac{a}{s(s+a)} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( t = \frac{1-e^{-at}}{a} )</td>
<td>( \frac{1}{s^2(s+a)} )</td>
<td>( \frac{z[(aT-1+T^2e^{-at})z+(a-e^{-at}aTe^{-at})]}{a(z-1)(z-e^{-at})} )</td>
</tr>
<tr>
<td>( 1 - (1+at)e^{-at} )</td>
<td>( \frac{e^{-at}-e^{-at}}{(s+a)(s+b)} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( e^{-at} - e^{-bT} )</td>
<td>( \frac{b-a}{(s+a)(s+b)} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( \sin(at) )</td>
<td>( \frac{a}{s^2+a^2} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( \cos(at) )</td>
<td>( \frac{s}{s^2+a^2} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( e^{-at} \sin(bt) )</td>
<td>( \frac{b}{(s+a)^2+b^2} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( e^{-at} \cos(bt) )</td>
<td>( \frac{s-a}{(s+a)^2+b^2} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
<tr>
<td>( 1 - e^{-at} (\cos(bt) + \frac{a}{b} \sin(bt)) )</td>
<td>( \frac{1}{a(s+a)(s+b)} )</td>
<td>( \frac{z}{z-1} - \frac{z}{z-e^{-at}} \frac{aTe^{-at}z}{(z-e^{-at})^2} )</td>
</tr>
</tbody>
</table>

\[
A = 1 - e^{-at} \left( \cos(bt) + \frac{a}{b} \sin(bt) \right)
\]
\[
B = e^{-2at} + e^{-at} \left( \frac{a}{b} \sin(bt) - \cos(bt) \right)
\]

\[
A = \frac{b(1-e^{-at})-a(1-e^{-bt})}{ab(b-a)}
\]
\[
B = \frac{a^2e^{-at} \left( 1-e^{-bt} \right)-be^{-bt} \left( 1-e^{-at} \right)}{ab(b-a)}
\]
3.2: Working with the \( z \) transform

- The following shows some correspondence between the \( z \)-plane and some discrete-time unit-pulse-response signals.

**Unit step:**
- \( h[k] = 1[k] \).
- \( H(z) = \frac{z}{z - 1} \), \( |z| > 1 \).

**Exponential (geometric):**
- \( h[k] = a^k 1[k] \), \( |a| < 1 \).
- \( H(z) = \frac{z}{z - a} \), \( |z| > |a| \).

**General cosinusoid:**
- \( h[k] = a^k \cos(\omega k) 1[k] \), \( |a| < 1 \).
- \( H(z) = \frac{z(z - a \cos \omega)}{z^2 - 2a(\cos \omega)z + a^2} \)
  for \( |z| > |a| \).

- The radius to the two poles is \( a \); the angle to the poles is \( \omega \).
- The zero (not at the origin) has the same real part as the two poles.
  - If \( \omega = 0 \), \( H(z) = \frac{z}{z - a} \ldots \) geometric!
  - If \( \omega = 0 \), \( a = 1 \), \( H(z) = \frac{z}{z - 1} \ldots \) step!
- Pole **radius** \( a \) is the geometric factor, determines settling time.
  1. \( |a| = 0 \), finite-duration response. *e.g.*, \( \delta[k - N] \leftrightarrow z^{-N} \).
  2. \( |a| > 1 \), growing signal that will not decay.
  3. \( |a| = 1 \), signal with constant amplitude; either step or cosine.
  4. \( |a| < 1 \), decaying signal. Small \( a = \) fast decay (see below).
<table>
<thead>
<tr>
<th>$a$</th>
<th>0.9</th>
<th>0.8</th>
<th>0.6</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>≈ duration</td>
<td>$N$</td>
<td>43</td>
<td>21</td>
<td>9</td>
</tr>
</tbody>
</table>

- Pole angle $\omega$ determines number of samples per oscillation.
  - That is, if we require $\cos(\omega k) = \cos(\omega(k + N))$, then
    $$N = \frac{2\pi}{\omega} \mid_{\text{rad}} = \frac{360}{\omega} \mid_{\text{deg}}.$$
- Solid: cst. damping ratio $\zeta$.
- Dashed: constant natural frequency $\omega_n$.

- Plot to right shows discrete-time unit-pulse responses versus pole locations.
Correspondence with continuous-time signals

- Let \( g(t) = e^{-at} \cos(bt)1(t) \).

- Suppose

\[
\begin{align*}
a &= \frac{0.3567}{T} \\
b &= \frac{\pi/4}{T}
\end{align*}
\]

\[ T = \text{sampling period}. \]

- Then,

\[
g[k] = g(kT) = \left(e^{-0.3567k}\right)^k \cos\left(\frac{\pi k}{4}\right)1[k] \\
= 0.7^k \cos\left(\frac{\pi k}{4}\right)1[k].
\]

(This is the cosinusoid example used in the earlier example).

- \( G(s) \) has poles at \( s_{1,2} = -a + jb \) and \( -a - jb \).

- \( G(z) \) has poles at radius \( e^{-aT} \) angle \( \omega = \pm bT \) or at \( e^{-aT \pm jbT} \).

  • So, \( z_{1,2} = e^{s_{1,2}T} \) and \( e^{s_{1,2}T} \).

- In general, poles convert between the \( s \)-plane and \( z \)-plane via \( z = e^{sT} \).

**EXAMPLE:** Some corresponding pole locations:
- $j\omega$-axis maps to unit circle.

- Constant damping ratio $\zeta$ maps to strange spiral.

- When considering system response to a step input for controls purposes, the following diagrams may be helpful:

Higher-order systems:

- Pole moving toward $z = 1$, system slows down.
- Zero moving toward $z = 1$, overshoot.
- Pole and zero moving close to each other cancel.
3.3: Discrete-time state-space form

- Discrete-time systems can also be represented in state-space form.

\[ x[k + 1] = A_d x[k] + B_d u[k] \]
\[ y[k] = C_d x[k] + D_d u[k]. \]

- The subscript “\(d\)” is used here to emphasize that, in general, the “\(A\)”, “\(B\)”, “\(C\)” and “\(D\)” matrices are different for discrete-time and continuous-time systems, even if the underlying plant is the same.

- I will usually drop the “\(d\)” and expect you to interpret the system from its context.

Formulating from transfer functions

- Discrete-time dynamics are represented as difference equations. e.g.,

\[ y[k + 3] + a_1 y[k + 2] + a_2 y[k + 1] + a_3 y[k] = b_1 u[k + 2] + b_2 u[k + 1] + b_3 u[k] \]
\[ y[k] + a_1 y[k - 1] + a_2 y[k - 2] + a_3 y[k - 3] = b_1 u[k - 1] + b_2 u[k - 2] + b_3 u[k - 3]. \]

- This particular example has transfer function

\[ G(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3} = \frac{Y(z)}{U(z)}. \]

- This transfer function may be converted to state-space in a very similar way to continuous-time systems.

- First, consider the poles:

\[ G_p(z) = \frac{1}{z^3 + a_1 z^2 + a_2 z + a_3} = \frac{V(z)}{U(z)} \]
\[ v[k + 3] + a_1 v[k + 2] + a_2 v[k + 1] + a_3 v[k] = u[k]. \]
Choose current and advanced versions of $v[k]$ as state.

$$x[k] = \begin{bmatrix} v[k + 2] & v[k + 1] & v[k] \end{bmatrix}^T.$$ 

Then

$$x[k + 1] = \begin{bmatrix} v[k + 3] \\ v[k + 2] \\ v[k + 1] \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v[k + 2] \\ v[k + 1] \\ v[k] \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u[k].$$

We now add zeros.

$$G(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3} = \frac{Y(z)}{U(z)}.$$ 

Break up transfer function into two parts. $\frac{V(z)}{U(z)}$ contains all of the poles of $\frac{Y(z)}{U(z)}$. Then,

$$Y(z) = [b_1 z^2 + b_2 z + b_3] V(z).$$

Or,

$$y[k] = b_1 v[k + 2] + b_2 v[k + 1] + b_3 v[k].$$ 

Then

$$x[k + 1] = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v[k + 2] \\ v[k + 1] \\ v[k] \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u[k]$$

$$y[k] = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} x[k] + \begin{bmatrix} 0 \end{bmatrix} u[k].$$
Many discrete-time transfer functions are not strictly proper. Solve by polynomial long division, and setting $D$ equal to the quotient.

In MATLAB, `[A, B, C, D] = tf2ss(num, den)` converts a transfer function form to state-space form (same syntax as for cts.-time, but `num` and `den` must have same order, maybe zero padded).

As with continuous-time systems, we have a lot of freedom when making state-space models (*i.e.*, in choosing components of $x[k]$).

**Canonical forms**

In discrete-time we have the same canonical forms: Controller, observer, controllability, observability, modal, and Jordan.

They are derived in the same way, as demonstrated above for the controller form.

A block diagram for controller form is:

**Time (dynamic) response**

*Homogeneous part*
First, consider the scalar case

\[ x[k + 1] = ax[k], \quad x[0]. \]

- Take \( z \)-transform. \( X(z) = (z - a)^{-1}z x[0]. \)
- Inverse \( z \)-transform. \( x[k] = a^k x[0]. \)
- Similarly, the full solution (vector case) is

\[ x[k] = A^k x[0]. \]

Aside: Nilpotent systems

- \( A \) is nilpotent if some power of \( n \) exists such that

\[ A^n = 0. \]

- \( A \) does not just decay to zero, it is exactly zero!
- This might be a desirable control design! (Why?) You might imagine that all the eigenvalues of \( A \) must be zero for this to work.

Forced solution

- The full solution is:

\[ x[k] = A^k x[0] + \sum_{j=0}^{k-1} A^{k-1-j} Bu[j]. \]

- This can be proved by induction from the equation

\[ x[k + 1] = Ax[k] + Bu[k], \quad x[0] \]

- Clearly, if \( y[k] = C x[k] + Du[k], \)

\[ y[k] = CA^k x[0] + \sum_{j=0}^{k-1} CA^{k-1-j} Bu[k] + Du[k]. \]
3.4: More on discrete-time state-space models

State-space to transfer function

- Start with the state equations
  \[ x[k + 1] = Ax[k] + Bu[k] \]
  \[ y[k] = Cx[k] + Du[k] \]

- \( z \)-transform
  \[ zX(z) - zx[0] = AX(z) + BU(z) \]
  \[ Y(z) = CX(z) + DU(z) \]

or
  \[ (zI - A)X(z) = BU(z) + zx[0] \]
  \[ X(z) = (zI - A)^{-1}BU(z) + (zI - A)^{-1}zx[0] \]

and
  \[ Y(z) = [C(zI - A)^{-1}B + D]U(z) + C(zI - A)^{-1}zx[0] \]

- So,
  \[ \frac{Y(z)}{U(z)} = C(zI - A)^{-1}B + D \]

- Same form as for continuous-time systems.

- Poles of system are roots of \( \det[zI - A] = 0 \).

Transformation

- State-space representations are not unique. Selection of state \( x \) are quite arbitrary.
Analyze the transformation of

\[ x[k + 1] = Ax[k] + Bu[k] \]
\[ y[k] = Cx[k] + Du[k] \]

Let \( x[k] = Tw[k] \), where \( T \) is an invertible (similarity) transformation matrix.

\[ w[k + 1] = T^{-1}ATw[k] + T^{-1}B u[k] \]
\[ y[k] = CT w[k] + Du[k] \]

so, \( w[k + 1] = \tilde{A}w[k] + \tilde{B}u[k] \)
\[ y[k] = \tilde{C}w[k] + \tilde{D}u[k] \].

Same as for continuous-time.

Converting plant dynamics to discrete time

Combine the dynamics of the zero-order hold and the plant.

\[ u[k] \xrightarrow{\text{ZOH}} u(t) \xrightarrow{\text{A, B, C, D}} y(t) \]

The continuous-time dynamics of the plant are:

\[ \dot{x}(t) = Ax(t) + Bu(t) \]
\[ y(t) = Cx(t) + Du(t). \]

Evaluate \( x(t) \) at discrete times. Recall

\[ x(t) = \int_0^t e^{A(t-\tau)} Bu(\tau) \, d\tau \]
\[
x[k + 1] = x((k + 1)T) = \int_0^{(k+1)T} e^{A((k+1)T-\tau)} B u(\tau) \, d\tau
\]

- With malice aforethought, break up the integral into two pieces. The first piece will become \(A_d\) times \(x(kT)\). The second part will become \(B_d\) times \(u(kT)\).

\[
= \int_0^{kT} e^{A((k+1)T-\tau)} B u(\tau) \, d\tau + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)} B u(\tau) \, d\tau
\]

\[
= \int_0^{kT} e^{AT} e^{A(kT-\tau)} B u(\tau) \, d\tau + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)} B u(\tau) \, d\tau
\]

\[
= e^{AT} x(kT) + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)} B u(\tau) \, d\tau.
\]

In the remaining integral, note that \(u(\tau)\) is constant from \(kT\) to \((k + 1)T\), and equal to \(u(kT)\); let \(\sigma = (k + 1)T - \tau; \tau = (k + 1)T - \sigma\); \(d\tau = -d\sigma\).

\[
x((k + 1)T) = e^{AT} x(kT) + \left[ \int_0^T e^{A\sigma} B \, d\sigma \right] u(kT)
\]

or, \(x[k + 1] = e^{AT} x[k] + \left[ \int_0^T e^{A\sigma} B \, d\sigma \right] u[k]\).

- So, we have a discrete-time state-space representation from the continuous-time representation.

\[
x[k + 1] = A_d x[k] + B_d u[k]
\]

where \(A_d = e^{AT}\), \(B_d = \int_0^T e^{A\sigma} B \, d\sigma\).

- Similarly,

\[
y[k] = C x[k] + D u[k].
\]

That is, \(C_d = C; D_d = D\).
Calculating $A_d$, $B_d$, $C_d$ and $D_d$

- $C_d$ and $D_d$ require no calculation since $C_d = C$ and $D_d = D$.
- $A_d$ is calculated via the *matrix* exponential $A_d = e^{A T}$. This is different from taking the exponential of each element in $A T$.
- If MATLAB is handy, you can type in $A_d = \text{expm}(A \times T)$
- If MATLAB is not handy, then you need to work a little harder. Recall from the previous set of notes that $e^{A t} = \mathcal{L}^{-1}[(s I - A)^{-1}]$. So,

$$e^{A T} = \mathcal{L}^{-1}[(s I - A)^{-1}]|_{t = T},$$

which is probably the “easiest” way to work it out by hand. Or,

$$e^{A T} = I + A T + \frac{A^2 T^2}{2!} + \frac{A^3 T^3}{3!} + \cdots$$

which is a convergent series so may be approximated with only a few terms.

- Now we focus on computing $B_d$. Recall that

$$B_d = \int_0^T e^{A \sigma} B \, d\sigma$$

$$= \int_0^T \left( I + A \sigma + A^2 \sigma^2 + \cdots \right) B \, d\sigma$$

$$= \left( I T + A \frac{T^2}{2!} + A^2 \frac{T^3}{3!} + \cdots \right) B$$

$$= A^{-1}(e^{A T} - I) B$$

$$= A^{-1}(A_d - I) B,$$

if $A^{-1}$ exists.

- So, calculating $B_d$ is easy once we have already calculated $A_d$.
- Also, in MATLAB, $[A_d, B_d] = \text{c2d}(A, B, T)$
3.5: Linear time-varying and nonlinear discrete-time systems

Linear time-varying discrete-time systems

- A linear time-varying system can be written as
  \[
  y[k] = C[k]x[k] + D[k]u[k].
  \]
  
- Analysis is somewhat easier than for the continuous-time counterpart.

- Consider first the homogeneous case
  \[
  x[k + 1] = A[k]x[k], \quad x[k_0] = x_0, \quad k \geq 0.
  \]
  The solution to this is
  \[
  x[k] = \Phi[k, k_0]x_0
  \]
  where
  \[
  \Phi[k, k_0] = \begin{cases}
  I, & k = k_0 \\
  A[k - 1]A[k - 2] \cdots A[k_0], & k > k_0.
  \end{cases}
  \]

- We see that the state-transition matrix can be computed readily, and does not involve any difficult integrals.

- Some properties of the state-transition matrix include:
  \[
  \Phi[k + 1, k_0] = A[k]\Phi[k, k_0] \\
  \Phi[k_0, k_0] = I \\
  \Phi[k, s]\Phi[s, \tau] = \Phi[k, \tau].
  \]

- The solution to the nonhomogeneous case can be shown to be
  \[
  x[k] = \Phi[k, k_0]x_0 + \sum_{\tau=k_0}^{k-1} \Phi[k, \tau + 1]B[\tau]u[\tau]
  \]
Nonlinear discrete-time systems

- The approach to working with nonlinear discrete-time systems is similar to that used for nonlinear continuous-time systems.
  - We can linearize around an equilibrium point or around a solution trajectory.
- We define a nonlinear state-space form as
  \[
  x[k + 1] = f(x[k], u[k]) \\
  y[k] = g(x[k], u[k]).
  \]

Linearizing around an equilibrium point

- We can linearize around an equilibrium point if there exists an equilibrium constant solution
  \[
  u[k] = u^{eq}, \quad x[k] = x^{eq}, \quad y[k] = y^{eq}.
  \]
- Then, let the actual input signal and initial state be written as
  \[
  u[k] = u^{eq} + \delta u[k], \quad x[k_0] = x^{eq} + \delta x^{eq}[k_0]
  \]
  for small \( \delta u[k] \).
- Then, we can write actual state and output as
  \[
  x[k] = x^{eq} + \delta x[k], \quad y[k] = y^{eq} + \delta y[k].
  \]
- Following the same process used in the prior chapter, the linearized perturbation system is

\[
y[k] = C[k] \Phi[k, k_0] x_0 + \sum_{\tau=k_0}^{k-1} C[k] \Phi[k, \tau + 1] B[\tau] u[\tau] + D[k] u[k].
\]
\[
\delta x[k + 1] = A \delta x[k] + B \delta u[k]
\]
\[
\delta y[k] = C \delta x[k] + D \delta u[k],
\]
where the linearized state-space matrices are
\[
A = \left( \frac{df(x^{eq}, u^{eq})}{dx} \right) \quad B = \left( \frac{df(x^{eq}, u^{eq})}{du} \right)
\]
\[
C = \left( \frac{dg(x^{eq}, u^{eq})}{dx} \right) \quad D = \left( \frac{dg(x^{eq}, u^{eq})}{du} \right)
\]
and the overall state and output can be computed as
\[
x[k] = x^{eq} + \delta x[k] \quad \text{and} \quad y[k] = y^{eq} + \delta y[k].
\]

**Linearizing around a solution trajectory**

- Alternately, suppose it is known that
  \[u^{sol}[k], \quad x^{sol}[k], \quad y^{sol}[k]\]
  form a time-varying solution to the nonlinear dynamics of the system.
- This means that
  \[x^{sol}[k + 1] = f(x^{sol}[k], u^{sol}[k])\]
  \[y^{sol}[k] = g(x^{sol}[k], u^{sol}[k]).\]
- Then, let general input, state, and output be
  \[u[k] = u^{sol}[k] + \delta u[k]\]
  \[x[k] = x^{sol}[k] + \delta x[k]\]
  \[y[k] = y^{sol}[k] + \delta y[k].\]
- Proceeding as before, we find the perturbation system
\[ \delta x[k + 1] \approx \left( \frac{d f(x^\text{sol}, u^\text{sol})}{dx[k]} \right) \delta x[k] + \left( \frac{d f(x^\text{sol}, u^\text{sol})}{du[k]} \right) \delta u[k] \]

\[ \delta y[k] \approx \left( \frac{d g(x^\text{sol}, u^\text{sol})}{dx[k]} \right) \delta x[k] + \left( \frac{d g(x^\text{sol}, u^\text{sol})}{du[k]} \right) \delta u[k]. \]

- The overall state and output can be computed as

\[ x[k] = x^\text{sol} + \delta x[k] \quad \text{and} \quad y[k] = y^\text{sol} + \delta y[k]. \]

- Notice that even if the nonlinear system is time-invariant, the linearized system will be time-varying, in general.

Where to from here?

- We have now seen how to model both continuous-time and discrete-time systems in state-space form.

- There are many commonalities between the two, which we will take advantage of in the remainder of the course.

- The main difference is in interpreting the result. For example,
  - Eigenvalues of the continuous-time \( A \) matrix correspond to \( s \)-plane locations, and should be in the left-half \( s \)-plane for stability;
  - Eigenvalues of the discrete-time \( A \) matrix correspond to \( z \)-plane locations, and should be in the unit circle for stability.

- In fact, stability is a very important concept, which must be evaluated and ensured before we talk more about control.

- This will be our next topic.