State-Space Modeling

- The state-space approach takes into account intermediate variables between input and output: state variables
- The state variables may correspond to actual physical quantities or may be just mathematical constructs
- The state vector is composed of $n$ state variables, where $n$ is the system order
- Unlike transfer functions, state-space models can include nonlinearities. Here we focus on linear, time-invariant state-space systems
- Digital controllers are based on state-space representations of the control TF
- It is possible to make conversions between TF and SS.
To go from TF to SS is to obtain a SS realization of the TF.

A given TF admits an infinite number of SS realizations.

A general (possibly nonlinear) \(n\)th-order state-space model with \(m\) inputs and \(p\) outputs has the form:

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
\dot{x}_2 &= f_2(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
&\quad \vdots \\
\dot{x}_n &= f_n(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
y_1 &= h_1(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
&\quad \vdots \\
y_p &= h_p(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m)
\end{align*}
\]

With SS, we replace a single \(n\)th-order ODE by \(n\) 1st order ones.

The column vector \(x = [x_1, x_2, \ldots, x_n]^T\) is called the state vector.

The set where \(x\) belongs is the state space.

For MCE441 we use \(\mathbb{R}^n\) as the state space. Others can be used too.

The column vector \(u = [u_1, u_2, \ldots, u_m]^T\) is the input vector.

The column vector \(y = [y_1, y_2, \ldots, y_p]^T\) is the output vector.
Linear State-Space Systems

- In a linear SS system, the $f_i$ and $h_i$ are linear functions, so that:

\[ \dot{x} = Ax + Bu, \quad y = Cx + Du \]

- $A$ is an $n$-by-$n$ matrix, $B$ is $n$-by-$m$, $C$ is $p$-by-$n$ and $D$ is $p$-by-$m$.

- Conversion to TF (prove it using Laplace transform)

\[ \frac{Y(s)}{U(s)} = G(s) = C(sI - A)^{-1}B + D \]

- It can be shown that the poles of $G(s)$ are the eigenvalues of $A$.

Example

Mass-spring-damper system. Choosing position and velocity as states, we find the following representation:

\[ \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{b} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \]
Example

Following our I/O approach we would have found:

\[ m\ddot{y} + b\dot{y} + ky = u \]

which gives a transfer function

\[ G(s) = \frac{Y(s)}{U(s)} = \frac{1}{ms^2 + bs + k} \]

We can check that \( G(s) = C(sI - A)^{-1}B + D \)

Another Example

Find a ss description of the the following RLC circuit. Find the TF between the input and output voltages.

\[ V_i \rightarrow L \rightarrow R \rightarrow V_o \]

\[ i \rightarrow V_i \rightarrow L \rightarrow R \rightarrow V_o \]
Realization of Transfer Functions

- If there are no zeros, obtain the I/O differential equation and define the \( n \) states to be successive derivatives of the output, starting with the 0th order (the output itself) until the derivative of order \((n - 1)\).

- Write equations of the form \( \dot{x}_i = x_{i+1} \) for \( i = 1, 2, \ldots, n - 1 \).

- Find \( \dot{x}_n \) from the I/O equation itself.

Example: Realize \( G(s) = \frac{2}{s^2 + s + 1} \)

Solution
Realization of Transfer Functions

Often, zeros are present. That is:

\[ G(s) = \frac{b_0 s^n + b_1 s^{n-1} + \ldots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n} \]

Use a *canonical realization* (one choice among infinite)

State definition:

\[
\begin{align*}
  x_1 &= y - \beta_0 u \\
  x_2 &= \dot{y} - \beta_0 \dot{u} - \beta_1 u = \dot{x}_1 - \beta u \\
  \vdots \\
  x_n &= y^{(n-1)} - \beta_0 u^{(n-1)} - \ldots - \beta_{n-1} u = \dot{x}_{n-1} - \beta_{n-1} u
\end{align*}
\]

Coefficient definition:

\[
\begin{align*}
  \beta_0 &= b_0 \\
  \beta_1 &= b_1 - a_1 \beta_0 \\
  \vdots \\
  \beta_n &= b_n - a_1 \beta_{n-1} - \ldots - a_n \beta_0
\end{align*}
\]

System Matrices:

\[
A = \begin{bmatrix}
  0 & 1 & 0 & \ldots & 0 \\
  0 & 0 & 1 & \ldots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & 0 & \ldots & 1 \\
  -a_n & -a_{n-1} & -a_{n-2} & \ldots & -a_1
\end{bmatrix}
\]
Realization of Transfer Functions...

System Matrices...

\[
B = [\beta_1, \beta_2, \ldots, \beta_{n-1}, \beta_n]^T
\]
\[
C = [1, 0, \ldots, 0]
\]
\[
D = \beta_0 = b_0
\]

System Description:

\[
\dot{x} = Ax + Bu
\]
\[
y = Cx + Du
\]

Example (Dorf 3.1)

Find a state-space realization of the transfer function

\[
G(s) = \frac{2s^2 + 8s + 6}{s^3 + 8s^2 + 16s + 6}
\]
State-Space to TF Conversions in Matlab

Old-fashioned method:
To go from tf to ss: \[ [A, B, C, D] = \text{tf2ss}(\text{num}, \text{den}) \]
To go from ss to tf: \[ [\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D) \]

New method:
Create a tf object: \[ \text{tf} \_ \text{sys} = \text{tf}(\text{num}, \text{den}) \]
Convert to ss: \[ \text{sys} \_ \text{ss} = \text{ss}(\text{tf} \_ \text{sys}) \]
Extract system matrices: \[ [A, B, C, D] = \text{ssdata}(\text{sys} \_ \text{ss}) \]
Go back to tf: \[ \text{tf} \_ \text{sys} = \text{tf}(\text{sys} \_ \text{ss}) \]

Note: The ss realization in the old method uses the control canonical form. The new method uses an efficient numerical procedure. The second method is preferred since it is required for conversions to and from discrete time.

Linearization - Equilibrium Points

An equilibrium point or fixed point of a control system under constant control \( u_0 \) is defined by the property

\[ \dot{x} = f(x, u_0) = 0 \]

In nonlinear systems, multiple equilibrium points can exist. For example, the system

\[ \begin{align*}
\dot{x}_1 &= x_1^2 - u \\
\dot{x}_2 &= x_1 x_2^2 - x_2 u
\end{align*} \]

has 4 equilibrium points for \( u_0 = 1 \). Find them.
Linearization - Equilibrium Points

We may linearize a nonlinear system about an equilibrium point by using the Jacobian of $f$. In doing so, we obtain a linear state-space system for state and control deviations: $\dot{\Delta x} = A\Delta x + B\Delta u$. As an example, we linearize the above system about $u_0 = 1$ and $x_0 = [1, 0]^T$

\[
\begin{align*}
\dot{\Delta x}_1 &= \left. \frac{\partial f_1}{\partial x_1} \right|_{u_0,x_0} \Delta x_1 + \left. \frac{\partial f_1}{\partial x_2} \right|_{u_0,x_0} \Delta x_2 + \left. \frac{\partial f_1}{\partial u} \right|_{u_0,x_0} \Delta u \\
\dot{\Delta x}_2 &= \left. \frac{\partial f_2}{\partial x_1} \right|_{u_0,x_0} \Delta x_1 + \left. \frac{\partial f_2}{\partial x_2} \right|_{u_0,x_0} \Delta x_2 + \left. \frac{\partial f_2}{\partial u} \right|_{u_0,x_0} \Delta u
\end{align*}
\]

The matrix $A$ is obtained by taking all partial derivatives of $f$ with respect to all variables ($A = \{a_{ij}\} = \frac{\partial f_i}{\partial x_j}$), evaluated at $(u_0, x_0)$. The matrix $B$ is obtained by taking partial derivatives w.r.t. all control components: $B = \{b_i\} = \frac{\partial f_i}{\partial u_i}$.

Note that the linearized system is valid only in close proximity to $(u_0, x_0)$. 