State-Space Systems

- What are state-space models?
- Why should we use them?
- How are they related to the transfer functions used in classical control design and how do we develop a state-space model?
- What are the basic properties of a state-space model, and how do we analyze these?
Introduction

- State space model: a representation of the dynamics of an $N^{th}$ order system as a first order differential equation in an $N$-vector, which is called the state.
  - Convert the $N^{th}$ order differential equation that governs the dynamics into $N$ first-order differential equations

- Classic example: second order mass-spring system

$$m\ddot{p} + c\dot{p} + kp = F$$

- Let $x_1 = p$, then $x_2 = \dot{p} = \dot{x}_1$, and

$$\ddot{x}_2 = \ddot{p} = (F - c\dot{p} - kp)/m = (F - cx_2 - kx_1)/m$$

$$\Rightarrow \begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u$$

- Let $u = F$ and introduce the state

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} \Rightarrow \dot{x} = Ax + Bu$$

- If the measured output of the system is the position, then we have that

$$y = p = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = cx$$
The most general continuous-time linear dynamical system has form
\[
\dot{x}(t) = A(t)x(t) + B(t)u(t) \\
y(t) = C(t)x(t) + D(t)u(t)
\]
where:
- \( t \in \mathbb{R} \) denotes time
- \( x(t) \in \mathbb{R}^n \) is the state (vector)
- \( u(t) \in \mathbb{R}^m \) is the input or control
- \( y(t) \in \mathbb{R}^p \) is the output

- \( A(t) \in \mathbb{R}^{n \times n} \) is the dynamics matrix
- \( B(t) \in \mathbb{R}^{n \times m} \) is the input matrix
- \( C(t) \in \mathbb{R}^{p \times n} \) is the output or sensor matrix
- \( D(t) \in \mathbb{R}^{p \times m} \) is the feedthrough matrix

Note that the plant dynamics can be time-varying.
- Also note that this is a MIMO system.

We will typically deal with the time-invariant case
\( \Rightarrow \) **Linear Time-Invariant (LTI)** state dynamics
\[
\dot{x}(t) = Ax(t) + Bu(t) \\
y(t) = Cx(t) + Du(t)
\]
so that now \( A, B, C, D \) are constant and do not depend on \( t \).
Basic Definitions

- **Linearity** – What is a linear dynamical system? A system $G$ is linear with respect to its inputs and output
  \[ u(t) \rightarrow G(s) \rightarrow y(t) \]

  if superposition holds:
  \[ G(\alpha_1 u_1 + \alpha_2 u_2) = \alpha_1 Gu_1 + \alpha_2 Gu_2 \]

  So if $y_1$ is the response of $G$ to $u_1$ ($y_1 = Gu_1$), and $y_2$ is the response of $G$ to $u_2$ ($y_2 = Gu_2$), then the response to $\alpha_1 u_1 + \alpha_2 u_2$ is $\alpha_1 y_1 + \alpha_2 y_2$

- A system is said to be **time-invariant** if the relationship between the input and output is independent of time. So if the response to $u(t)$ is $y(t)$, then the response to $u(t - t_0)$ is $y(t - t_0)$

- $x(t)$ is called the **state of the system** at $t$ because:
  - Future output depends only on current state and future input
  - Future output depends on past input only through current state
  - State summarizes effect of past inputs on future output – like the **memory of the system**

- Example: Rechargeable flashlight – the state is the **current state of charge** of the battery. If you know that state, then you do not need to know how that level of charge was achieved (assuming a perfect battery) to predict the future performance of the flashlight.
Creating Linear State-Space Models

- Most easily created from $N^{\text{th}}$ order differential equations that describe the dynamics
  - This was the case done before.
  - Only issue is which set of states to use – there are many choices.

- Can be developed from transfer function model as well.
  - Much more on this later

- Problem is that we have restricted ourselves here to linear state space models, and almost all systems are nonlinear in real-life.
  - Can develop linear models from nonlinear system dynamics
Linearization

- Often have a nonlinear set of dynamics given by
  \[ \dot{x} = f(x, u) \]
  where \( x \) is once gain the state vector, \( u \) is the vector of inputs, and \( f(\cdot, \cdot) \) is a nonlinear vector function that describes the dynamics

- **Example:** simple spring. With a mass at the end of a linear spring (rate \( k \)) we have the dynamics
  \[ m\ddot{x} = kx \]
  but with a “leaf spring” as is used on car suspensions, we have a nonlinear spring – the more it deflects, the stiffer it gets. Good model now is
  \[ m\ddot{x} = (k_1 x + k_2 x^3) \]
  which is a “cubic spring”.

  - Restoring force depends on the deflection \( x \) in a nonlinear way.

![Graphs showing linear and nonlinear responses](image)

**Figure 1:** Response to linear \( k \) and nonlinear \((k_1 = 0, k_2 = k)\) springs (code at the end)
• Typically assume that the system is operating about some nominal state solution $x^0(t)$ (possibly requires a nominal input $u^0(t)$)
  
  Then write the actual state as $x(t) = x^0(t) + \delta x(t)$ and the actual inputs as $u(t) = u^0(t) + \delta u(t)$
  
  The “$\delta$” is included to denote the fact that we expect the variations about the nominal to be “small”

• Can then develop the linearized equations by using the **Taylor series expansion** of $f(\cdot, \cdot)$ about $x^0(t)$ and $u^0(t)$.

• Recall the vector equation $\dot{x} = f(x, u)$, each equation of which

  $$\dot{x}_i = f_i(x, u)$$

  can be expanded as

  $$\frac{d}{dt}(x^0_i + \delta x_i) = f_i(x^0 + \delta x, u^0 + \delta u)$$

  $$\approx f_i(x^0, u^0) + \frac{\partial f_i}{\partial x} \bigg|_0 \delta x + \frac{\partial f_i}{\partial u} \bigg|_0 \delta u$$

  where

  $$\frac{\partial f_i}{\partial x} = \begin{bmatrix} \frac{\partial f_i}{\partial x_1} & \cdots & \frac{\partial f_i}{\partial x_n} \end{bmatrix}$$

  and $\cdot \big|_0$ means that we should evaluate the function at the nominal values of $x^0$ and $u^0$.

• The meaning of “small” deviations now clear – the variations in $\delta x$ and $\delta u$ must be small enough that we can ignore the higher order terms in the Taylor expansion of $f(x, u)$. 
• Since \( \frac{d}{dt}x_i^0 = f_i(x^0, u^0) \), we thus have that

\[
\frac{d}{dt}(\delta x_i) \approx \left. \frac{\partial f_i}{\partial x} \right|_0 \delta x + \left. \frac{\partial f_i}{\partial u} \right|_0 \delta u
\]

• Combining for all \( n \) state equations, gives (note that we also set \( \approx \rightarrow = \)) that

\[
\frac{d}{dt} \delta x = \begin{bmatrix}
\left. \frac{\partial f_1}{\partial x} \right|_0 \\
\left. \frac{\partial f_2}{\partial x} \right|_0 \\
\vdots \\
\left. \frac{\partial f_n}{\partial x} \right|_0
\end{bmatrix} \delta x + \begin{bmatrix}
\left. \frac{\partial f_1}{\partial u} \right|_0 \\
\left. \frac{\partial f_2}{\partial u} \right|_0 \\
\vdots \\
\left. \frac{\partial f_n}{\partial u} \right|_0
\end{bmatrix} \delta u
\]

\[
= A(t) \delta x + B(t) \delta u
\]

where

\[
A(t) \equiv \begin{bmatrix}
\left. \frac{\partial f_1}{\partial x_1} \frac{\partial f_1}{\partial x_2} \cdots \frac{\partial f_1}{\partial x_n} \right|_0 \\
\left. \frac{\partial f_2}{\partial x_1} \frac{\partial f_2}{\partial x_2} \cdots \frac{\partial f_2}{\partial x_n} \right|_0 \\
\vdots \\
\left. \frac{\partial f_n}{\partial x_1} \frac{\partial f_n}{\partial x_2} \cdots \frac{\partial f_n}{\partial x_n} \right|_0
\end{bmatrix}
\]

and

\[
B(t) \equiv \begin{bmatrix}
\left. \frac{\partial f_1}{\partial u_1} \frac{\partial f_1}{\partial u_2} \cdots \frac{\partial f_1}{\partial u_m} \right|_0 \\
\left. \frac{\partial f_2}{\partial u_1} \frac{\partial f_2}{\partial u_2} \cdots \frac{\partial f_2}{\partial u_m} \right|_0 \\
\vdots \\
\left. \frac{\partial f_n}{\partial u_1} \frac{\partial f_n}{\partial u_2} \cdots \frac{\partial f_n}{\partial u_m} \right|_0
\end{bmatrix}
\]
Similarly, if the nonlinear measurement equation is \( y = g(x, u) \), can show that, if \( y(t) = y^0 + \delta y \), then

\[
\delta y = \begin{bmatrix}
\frac{\partial g_1}{\partial x} \bigg|_0 \\
\frac{\partial g_2}{\partial x} \bigg|_0 \\
\vdots \\
\frac{\partial g_p}{\partial x} \bigg|_0
\end{bmatrix} \delta x + \begin{bmatrix}
\frac{\partial g_1}{\partial u} \bigg|_0 \\
\frac{\partial g_2}{\partial u} \bigg|_0 \\
\vdots \\
\frac{\partial g_p}{\partial u} \bigg|_0
\end{bmatrix} \delta u
\]

\[
= C(t) \delta x + D(t) \delta u
\]

Typically think of these nominal conditions \( x^0, u^0 \) as “set points” or “operating points” for the nonlinear system. The equations

\[
\frac{d}{dt} \delta x = A(t) \delta x + B(t) \delta u
\]

\[
\delta y = C(t) \delta x + D(t) \delta u
\]

then give us a linearized model of the system dynamic behavior about these operating/set points.

Note that if \( x^0, u^0 \) are constants, then the partial fractions in the expressions for \( A–D \) are all constant \( \rightarrow \text{LTI linearized model} \).

One particularly important set of operating points are the equilibrium points of the system. Defined as the states & control input combinations for which

\[
\dot{x} = f(x^0, u^0) \equiv 0
\]

provides \( n \) algebraic equations to find the equilibrium points.
Example

- Consider the nonlinear spring with (set \( m = 1 \))

\[
\dot{y} = k_1 y + k_2 y^3
\]

gives us the nonlinear model \((x_1 = y \text{ and } x_2 = \dot{y})\)

\[
\frac{d}{dt} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ k_1 y + k_2 y^3 \end{bmatrix} \Rightarrow \dot{x} = f(x)
\]

- Find the equilibrium points and then make a state space model

- For the equilibrium points, we must solve

\[
f(x) = \begin{bmatrix} \dot{y} \\ k_1 y + k_2 y^3 \end{bmatrix} = 0
\]

which gives

\[
\dot{y}^0 = 0 \quad \text{and} \quad k_1 y^0 + k_2 (y^0)^3 = 0
\]

- Second condition corresponds to \( y^0 = 0 \) or \( y^0 = \pm \sqrt{-k_1/k_2} \),

which is only real if \( k_1 \) and \( k_2 \) are opposite signs.

- For the state space model,

\[
A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ k_1 + 3k_2(y)^2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ k_1 + 3k_2(y^0)^2 & 0 \end{bmatrix}
\]

and the linearized model is \( \dot{x} = A\delta x \)
For the equilibrium point $y = 0, \dot{y} = 0$

$$A_0 = \begin{bmatrix} 0 & 1 \\ k_1 & 0 \end{bmatrix}$$

which are the standard dynamics of a system with just a linear spring of stiffness $k_1$

- Stable motion about $y = 0$ if $k_1 < 0$

Assume that $k_1 = 1$, $k_2 = -1/2$, then we should get an equilibrium point at $\dot{y} = 0, y = \pm \sqrt{2}$, and since $k_1 + k_2(y^0)^2 = 0$

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2k_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & 0 \end{bmatrix}$$

are the dynamics of a stable oscillator about the equilibrium point

- Will explore this in detail later

Figure 2: Nonlinear response ($k_1 = 1, k_2 = -.5$). The figure on the right shows the oscillation about the equilibrium point.
Linearized Nonlinear Dynamics

- Usually in practice we drop the “δ” as they are rather cumbersome, and (abusing notation) we write the state equations as:

\[
\begin{align*}
\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\
y(t) &= C(t)x(t) + D(t)u(t)
\end{align*}
\]

which is of the same form as the previous linear models
Example: Aircraft Dynamics

- Assumptions:
  1. Earth is an inertial reference frame
  2. A/C is a rigid body
  3. Body frame $B$ fixed to the aircraft $(\vec{i}, \vec{j}, \vec{k})$

- The basic dynamics are:

  \[
  \vec{F} = m \dot{\vec{v}}_c \quad \text{and} \quad \vec{T} = \dot{\vec{H}}
  \]

  \[
  \Rightarrow \frac{1}{m} \vec{F} = \dot{\vec{v}}_c^B + B^I \vec{\omega} \times \vec{v}_c \quad \text{Transport Thm.}
  \]

  \[
  \Rightarrow \vec{T} = \dot{\vec{H}} + B^I \vec{\omega} \times \vec{H}
  \]

- Instantaneous mapping of $\vec{v}_c$ and $B^I \vec{\omega}$ into the body frame is given by

  \[
  B^I \vec{\omega} = P\vec{i} + Q\vec{j} + R\vec{k} \quad \vec{v}_c = U\vec{i} + V\vec{j} + W\vec{k}
  \]

  \[
  \Rightarrow B^I \omega_B = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad \Rightarrow (v_c)_B = \begin{bmatrix} U \\ V \\ W \end{bmatrix}
  \]
• The overall equations of motion are then:

\[
\frac{1}{m} \vec{F} = \dot{\vec{v}}_c^B + B I \vec{\omega} \times \vec{v}_c
\]

\[
\Rightarrow \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} + \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix}
\]

\[
= \begin{bmatrix} \dot{U} + QW - RV \\ \dot{V} + RU - PW \\ \dot{W} + PV - QU \end{bmatrix}
\]

• These are clearly nonlinear – need to linearize about the equilibrium states.

• To find suitable equilibrium conditions, must solve

\[
\frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} +QW - RV \\ +RU - PW \\ +PV - QU \end{bmatrix} = 0
\]

• Assume steady state flight conditions with \( \dot{P} = \dot{Q} = \dot{R} = 0 \)
• Define the **trim** angular rates, velocities, and Forces

\[
B_l \omega_B^o = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}, \quad (v_c)_B^o = \begin{bmatrix} U_o \\ 0 \\ 0 \end{bmatrix}, \quad F_B^o = \begin{bmatrix} F_x^o \\ F_y^o \\ F_z^o \end{bmatrix}
\]

that are associated with the flight condition (they define the type of equilibrium motion that we analyze about).

**Note:**

- \( W_0 = 0 \) since we are using the stability axes, and

- \( V_0 = 0 \) because we are assuming symmetric flight

• Can now linearize the equations about this flight mode. To proceed, define

**Velocities**

\[
U_0, \quad U = U_0 + u \quad \Rightarrow \quad \dot{U} = \dot{u}
\]

\[
W_0 = 0, \quad W = w \quad \Rightarrow \quad \dot{W} = \dot{w}
\]

\[
V_0 = 0, \quad V = v \quad \Rightarrow \quad \dot{V} = \dot{v}
\]

**Angular Rates**

\[
P_0 = 0, \quad P = p \quad \Rightarrow \quad \dot{P} = \dot{p}
\]

\[
Q_0 = 0, \quad Q = q \quad \Rightarrow \quad \dot{Q} = \dot{q}
\]

\[
R_0 = 0, \quad R = r \quad \Rightarrow \quad \dot{R} = \dot{r}
\]

**Angles**

\[
\Theta_0, \quad \Theta = \Theta_0 + \theta \quad \Rightarrow \quad \dot{\Theta} = \dot{\theta}
\]

\[
\Phi_0 = 0, \quad \Phi = \phi \quad \Rightarrow \quad \dot{\Phi} = \dot{\phi}
\]

\[
\Psi_0 = 0, \quad \Psi = \psi \quad \Rightarrow \quad \dot{\Psi} = \dot{\psi}
\]
Linearization for symmetric flight

\[ U = U_0 + u, \quad V_0 = W_0 = 0, \quad P_0 = Q_0 = R_0 = 0. \]

Note that the forces and moments are also perturbed.

\[
\frac{1}{m} \left[ F_x^0 + \Delta F_x \right] = \dot{U} + QW - RV \approx \dot{u} + qw - rv \\
\approx \dot{u}
\]

\[
\frac{1}{m} \left[ F_y^0 + \Delta F_y \right] = \dot{V} + RU - PW \approx \dot{v} + r(U_0 + u) - pw \\
\approx \dot{v} + rU_0
\]

\[
\frac{1}{m} \left[ F_z^0 + \Delta F_z \right] = \dot{W} + PV - QU \approx \dot{w} + pv - q(U_0 + u) \\
\approx \dot{w} - qU_0
\]

\[
\Rightarrow \frac{1}{m} \begin{bmatrix} \Delta F_x \\ \Delta F_y \\ \Delta F_z \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} + rU_0 \\ \dot{w} - qU_0 \end{bmatrix}
\]

Which gives the linearized dynamics for the aircraft motion about the steady-state flight condition.

- Need to analyze the perturbations to the forces and moments to fully understand the linearized dynamics – take 16.61
- Can do same thing for the rotational dynamics.
% save this entire code as plant.m
% function [xdot] = plant(t,x)
global n
xdot(1) = x(2);
xdot(2) = -3*(x(1))^n;
xdot = xdot';
return

% the use this part of the code in Matlab®
% to call_plant.m
global n
n=3; %nonlinear
x0 = [-1 2]; % initial condition
[T,x]=ode23('plant', [0 12], x0); %simulate NL equations for 12 sec
n=1; % linear
[T1,x1]=ode23('plant', [0 12], x0);

subplot(211)
plot(T,x(:,1),T1,x1(:,1),'--');
legend('Nonlinear','Linear')
ylabel('X')
xlabel('Time')
subplot(212)
plot(T,x(:,2),T1,x1(:,2),'--');
legend('Nonlinear','Linear')
ylabel('V')
xlabel('Time')