### 16.513 Control Systems

**Summary of Results From Last Lecture:**

Consider the system: \( \dot{x} = Ax + Bu; \quad y = Cx + Du \)

Given \( x(0) \) and \( u(t) \) for \( t \geq 0 \). The solution is

\[
\begin{align*}
x(t) &= e^{At}x(0) + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau; \\
y(t) &= Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)} Bu(\tau) d\tau + Du(t)
\end{align*}
\]

The main problem involved is to compute \( e^{At} \).

The system is **internally stable** iff all the eigenvalues of \( A \) have negative real parts: \( \text{Re} \lambda_i(A) < 0 \) for all \( i \)

\( \Rightarrow x(t) \to 0 \) if \( u(t)=0 \)

---

**Solution of Discrete-time Equations**

The DT system: \( x[k+1] = Ax[k] + Bu[k] \)
\( y[k] = Cx[k] + Du[k] \)

Given \( x[0] \) and \( u(k) \), \( k \geq 0 \), the solution is:

\[
\begin{align*}
x[k] &= A^k x[0] + \sum_{m=0}^{k-1} A^{k-m-1} Bu[m] \\
y[k] &= CA^k x[0] + \sum_{m=0}^{k-1} CA^{k-m-1} Bu[m] + Du[k]
\end{align*}
\]

The main problem involved is to compute \( A^k \).

The system is **internally stable** iff all the eigenvalues of \( A \) are within the unit disk: \( |\lambda_i(A)| < 1 \) for all \( i \)

\( \Rightarrow x[k] \to 0 \) if \( u[k]=0 \)
Discretizing a Continuous-Time System

A continuous-time system: \( \dot{x} = Ax + Bu; \quad y = Cx + Du \)

Given a sampling period \( T > 0 \), the discretized system:
\[
\begin{align*}
x[k + 1] &= A_dx[k] + B_du[k] \\
y[k] &= C_dx[k] + D_du[k]
\end{align*}
\]

where \( A_d = e^{AT} \), \( B_d = A^{-1}(A_d - I)B \), \( C_d = C \), \( D_d = D \)

- This exactly describes the input-state, input-output relationship at instants \( T, 2T, \ldots, kT, \ldots \)
- The discretization relies on the computation of \( e^{AT} \).

Let the eigenvalues of \( A \) be \( \lambda_i \), then the eigenvalues of \( e^{AT} \) are \( e^{\lambda iT} \). \( \text{Re}[\lambda_i] < 0 \rightarrow |e^{\lambda iT}| < 1 \). Stability is unchanged after discretization.

Today: we will address some miscellaneous problems about LTI systems

- How to deal with complex eigenvalues
- Realization of a transfer function
- Simulation of systems by using Simulink
  - Course project

To prepare for new topics in this course, we will also study

- Quadratic functions and positive-definiteness

Next Time: We start another topic (Chapter 6)

- Controllability and observability
**Deal with complex eigenvalues**

Typically, we would like to transform a matrix $A \in \mathbb{R}^{n \times n}$ into a diagonal form through equivalent transformation

$$
\tilde{A} = Q^{-1}AQ
$$

Suppose that $A$ has complex eigenvalues.

- Some $\lambda_i$ will be complex
- The transformation matrix $Q^{-1}$ will have complex entries
- What does the new state $z = Q^{-1}x$ stand for?

Physically, it is meaningless. Numerically, it may render analysis or design results invalid, such as a feedback law with complex numbers.

---

Since $A$ is a real matrix. If $A$ has complex eigenvalues.

- The complex eigenvalues appear as conjugate pairs: $\alpha_i+j\beta_i$, $\alpha_i-j\beta_i$.
- The eigenvectors also appear as conjugate pairs $v_i+jw_i$, $v_i-jw_i$.

There is a way to avoid complex numbers. Assume

$$
A(v + jw) = (\alpha + j\beta)(v + jw) = \alpha v - \beta w + j(\beta v + \alpha w);
$$

$$
A(v - jw) = (\alpha - j\beta)(v - jw) = \alpha v - \beta w - j(\beta v + \alpha w);
$$

Add the two: $Av = \alpha v - \beta w$

Subtract: $Aw = \beta v + \alpha w$

$$
A[v \ w] = [\alpha \ -\beta \ \beta \ \alpha]
$$
Suppose we have two real eigenvalues and two pairs of complex eigenvalues, all simple, 

\[ \lambda_1, \lambda_2, \alpha_1 + j\beta_1, \alpha_1 - j\beta_1, \alpha_2 + j\beta_2, \alpha_2 - j\beta_2, \]

with corresponding eigenvectors;

\[ q_1, q_2, v_1 + jw_1, v_1 - jw_1, v_2 + jw_2, v_2 - jw_2, \]

\[ Aq_1 = \lambda_1 q_1, \quad A[v_1 w_1] = [v_1 w_1] \begin{bmatrix} \alpha_1 & -\beta_1 \\ \beta_1 & \alpha_1 \end{bmatrix} \]

In matrix form;

\[ A \begin{bmatrix} q_1 \\ q_2 \\ v_1 \\ w_1 \\ v_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ v_1 \\ w_1 \\ v_2 \\ w_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_1 & -\beta_1 & 0 & 0 \\ 0 & 0 & \beta_1 & \alpha_1 & 0 & 0 \\ 0 & 0 & 0 & \alpha_2 & -\beta_2 & 0 \\ 0 & 0 & 0 & \beta_2 & \alpha_2 & 0 \end{bmatrix} \]

\[ \bar{A} = Q^{-1}AQ \]

Not strictly a diagonal form but real: a block diagonal form

Example:

\[ A = \begin{bmatrix} 0 & 0 & 1 \\ -2 & 0 & 2 \\ -2 & -1 & 3 \end{bmatrix} \]

Find the block diagonal form and the transformation matrix

Use matlab \([V,D]=eig(A)\), you get:

\[ V = \begin{bmatrix} 0.3162 - 0.3162i & 0.3162 + 0.3162i & 0.7071 \\ 0.6325 & 0.6325 & 0.0000 \\ 0.6325 & 0.6325 & 0.7071 \end{bmatrix} \]

\[ D = \begin{bmatrix} 1.0000 + 1.0000i & 0 & 0 \\ 0 & 1.0000 - 1.0000i & 0 \\ 0 & 0 & 1.0000 \end{bmatrix} \]

Where \( inv(V)*A*V=D \)

Let \( q_1=\text{real}(V(:,1)) \);

\( q_2=\text{imag}(V(:,1)); q_3=V(:,3) \)

Form \( Q=[q_1 q_2 q_3] \), then

\[ Q = \begin{bmatrix} 0.3162 & -0.3162 & 0.7071 \\ 0.6325 & 0 & 0.0000 \\ 0.6325 & 0 & 0.7071 \end{bmatrix} \]

\[ >> \text{inv}(Q)*A*Q \]

\[ \text{ans} = \begin{bmatrix} 1.0000 & 1.0000 & 0.0000 \\ -1.0000 & 1.0000 & -0.0000 \\ -0.0000 & 0 & 1.0000 \end{bmatrix} \]
What is $e^{At}$ with $A = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}$?

Use the first definition for a matrix function.

- Let $f(\lambda) = e^{\lambda t}$. Find $g(\lambda) = k_0 + k_1 \lambda$.
- The eigenvalues of $A$: $\lambda_1 = \alpha + j\beta$, $\lambda_2 = \alpha - j\beta$.

\[ g(\lambda_1) = f(\lambda_1) \quad k_0 + k_1 (\alpha + j\beta) = e^{(\alpha+j)\beta t} \]
\[ g(\lambda_2) = f(\lambda_2) \quad k_0 + k_1 (\alpha - j\beta) = e^{(\alpha-j)\beta t} \]

\[ k_1 = \frac{e^{\alpha t} \sin \beta t}{\beta}, \quad k_0 = e^{\alpha t} (\cos \beta t - \frac{\alpha}{\beta} \sin \beta t) \]

\[ e^{At} = g(A) = e^{\alpha t} \begin{bmatrix} \cos \beta t & -\sin \beta t \\ \sin \beta t & \cos \beta t \end{bmatrix} + e^{\alpha t} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} e^{(\alpha+j)\beta t} \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \]

\[ = e^{\alpha t} \begin{bmatrix} \cos \beta t & -\sin \beta t \\ \sin \beta t & \cos \beta t \end{bmatrix} \]

In summary:

With $A = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}$, $e^{At} = e^{\alpha t} \begin{bmatrix} \cos \beta t & -\sin \beta t \\ \sin \beta t & \cos \beta t \end{bmatrix}$,

For the system: $\dot{x} = Ax + Bu$

The zero-input solution is

\[ x(t) = e^{-At} x(0) = e^{\alpha t} \begin{bmatrix} \cos \beta t & -\sin \beta t \\ \sin \beta t & \cos \beta t \end{bmatrix} x(0) \]

The real part $\alpha$ (of the eigenvalue) determines the stability of the system;

The imaginary part $\beta$ determines the frequency of the oscillation.
Today: some miscellaneous problems about LTI systems
- How to deal with complex eigenvalues
- Realization of a transfer function
- Simulation of systems by using Simulink
- Course Project

And more from linear algebra
- Quadratic functions and positive-definiteness
State-space realizations of transfer functions

Given state equations
\[ \dot{x} = Ax + Bu; \quad y = Cx + Du \quad (*) \]
The transfer function is: \[ G(s) = C(sI - A)^{-1}B + D \]
Now, given \( G(s) \), how to find \((A,B,C,D)\)?

Background:
- Sometimes it is hard to obtain a state-space description.
- But you can identify the transfer function using frequency response.
- We have more advanced design methods for state-space models.

Example: \[ G(s) = \frac{4s^2 + 5s + 6}{s^3 + s^2 + 2s + 3} \]
It can be verified that \( G(s) = C(sI - A)^{-1}B + D \) with
\[
A = \begin{bmatrix} -1 & -2 & -3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}, \quad D = 0
\]
First, \[ |sI - A| = \begin{vmatrix} s+1 & 2 & 3 \\ -1 & s & 0 \\ 0 & -1 & s \end{vmatrix} = s^3 + s^2 + 2s + 3 \]
Then, \[ (sI - A)^{-1} = \frac{1}{s^3 + s^2 + 2s + 3} \begin{bmatrix} s^2 & * & * \\ s & * & * \\ 1 & * & * \end{bmatrix} \]
The adjunct matrix
\( A = \begin{bmatrix} -1 & -2 & -3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}, \quad D = 0 \)

\[
(sI - A)^{-1} = \frac{1}{s^3 + s^2 + 2s + 3} \begin{bmatrix} s^2 & * & * \\ s & * & * \\ 1 & * & * \end{bmatrix}
\]

\[
C(sI - A)^{-1}B + D = \frac{1}{s^3 + s^2 + 2s + 3} \begin{bmatrix} 4 & 5 & 6 \\ s^2 & * & * \\ s & * & * \\ 1 & * & * \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
\]

\[
= \frac{4s^2 + 5s + 6}{s^3 + s^2 + 2s + 3} = G(s)
\]

We say that \((A,B,C,D)\) is a realization of \(G(s)\)

If there exists \((A,B,C,D)\) such that \(G(s)=C(sI-A)^{-1}B+D\) then we say that \(G(s)\) is realizable.

\[ \text{Theorem:} \quad \text{A transfer matrix } G(s) \text{ is realizable with LTI state equation if and only if it is a proper rational matrix.} \]

- First observe that \(C(sI-A)^{-1}B+D\) is always proper and rational (necessity proved).
- A proper rational matrix can be decomposed as the sum of a constant matrix and a strictly proper rational matrix: \(G(s)=G_{sp}(s)+D, \quad D=G(\infty)\)
- Let \(d(s)=s^r + a_1s^{r-1} + a_2s^{r-2} + \ldots + a_{r-1}s + a_r\) be the least common denominator of all entries of \(G_{sp}(s)\)
- Then \(G_{sp}(s)\) can be expressed as (assume \(G\) is \(q \times p\))

\[
G_{sp}(s) = \frac{1}{d(s)} \begin{bmatrix} N_1s^{r-1} + N_2s^{r-2} + \ldots + N_{r-1}s + N_r \end{bmatrix}, \quad N_i \in \mathbb{R}^{q \times p}
\]
With
\[ G_{sp}(s) = \frac{1}{d(s)} \left[ N_1 s^{r-1} + N_2 s^{r-2} + \cdots + N_r s + N_r \right], \quad N_i \in \mathbb{R}^{m \times n} \]
\[ d(s) = s^r + a_1 s^{r-1} + \cdots + a_r s + a_r \]

The realization of \( G_{sp}(s) \) is given as:
\[
A = \begin{bmatrix}
-a_1 I_p & -a_2 I_p & \cdots & -a_r I_p & -a_r I_p \\
I_p & 0 & \cdots & 0 & 0 \\
0 & I_p & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & I_p & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
I_p \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]
\[
C = \begin{bmatrix}
N_1 & N_2 & \cdots & N_{r-1} & N_r
\end{bmatrix}
\]

Another form of realization: Problem 4.9

Example: \( G(s) = \begin{bmatrix}
\frac{4s-10}{2s+1} & \frac{3}{(s+2)(s+1)} \\
\frac{1}{(2s+1)(s+2)(s+2)} & \frac{s+1}{s+1}
\end{bmatrix} \)

Step 1: break it into a constant part and a strictly proper part
\[ G(s) = \begin{bmatrix}
2 & 0 \\
0 & 0
\end{bmatrix} + \begin{bmatrix}
\frac{-6}{(s+0.5)(s+2)} & \frac{3}{(s+2)^2} \\
\frac{3}{s+0.5(s+2)} & \frac{s+2}{s+1}
\end{bmatrix} \]
\[ \iff \frac{4s-10}{2s+1} + \frac{4s+2}{2s+1} + \frac{-12}{2s+1} = G_{sp}(s) \]

Step 2: the monic least common denominator
\[ d(s) = (s+0.5)(s+2)(s+2) = s^3 + 4.5 s^2 + 6 s + 2 \]
\[ \hat{a}_1 \hat{a}_2 \hat{a}_3 \]

Step 3:
\[ G_{sp}(s) = \frac{1}{(s+0.5)(s+2)^2} \begin{bmatrix}
-6(s+2)^2 & 3(s+2)(s+0.5) \\
0.5(s+2) & (s+1)(s+0.5)
\end{bmatrix} \]
\[ = \frac{1}{d(s)} \begin{bmatrix}
-6s^2 - 24s - 24 & 3s^2 + 6s + 3 \\
0.5s + 1 & s^2 + 1.5s + 0.5
\end{bmatrix} \]
\[ d(s) = s^3 + 4.5s^2 + 6s + 2 = a_1s^2 + a_2s + a_3 \]

From step 3: \[ G_p(S) = \frac{1}{d(s)} \begin{bmatrix} -6s^2 - 24s - 24 & 3s^2 + 7.5s + 3 \\ 0.5s + 1 & s^2 + 1.5s + 0.5 \end{bmatrix} \]

\[ = \frac{1}{d(s)} \begin{bmatrix} -6 & 3 \\ 0 & 1 \end{bmatrix} s^2 + \begin{bmatrix} -24 & 7.5 \\ 0.5 & 1.5 \end{bmatrix} s + \begin{bmatrix} -24 & 3 \\ 1 & 0.5 \end{bmatrix} \]

Step 4:
\[
A = \begin{bmatrix}
-4.5 & 0 & 0 & -2 \\
0 & -4.5 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
\end{bmatrix},
B = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\end{bmatrix},
\]
\[
C = \begin{bmatrix}
-6 & 3 & -24 & 7.5 & -24 & 3 \\
0 & 1 & 0.5 & 1.5 & 1 & 0.5 \\
\end{bmatrix},
D = \begin{bmatrix}
2 & 0 \\
0 & 0 \\
\end{bmatrix}
\]

Another realization of the same system is given in p.105 Example 4.7, where the dimension is only 4.

**Discussion:**
- The realization \((A,B,C,D)\) for a particular \(G(s)\) is not unique;
- All the equivalence transformations are also valid realizations;
- With different methods, the dimensions of the resulting systems, i.e., the number of state variables, may be different. There exist a minimal-order realization
- We will learn later how to reduce the order of a realization to the minimal number.
Today: some miscellaneous problems about LTI systems
- How to deal with complex eigenvalues
- Realization of a transfer function
  - Simulation of systems by using Simulink
- Course project
And more from linear algebra
- Quadratic functions and positive-definiteness

A Tool for System Simulation: SIMULINK
Can be used for simulation of various systems:
- Linear, CT or DT,
- Nonlinear;
- Switched;
- Hybrid: CT + DT components, signals;

Input signals can be arbitrarily generated:
- Standard: sinusoidal, polynomial, square, impulse
- Customized: from a function, look-up table

Output signals can be stored or demonstrated in different ways.
Example:

\[
\ddot{y} = u - 3\dot{y} - 2\dot{y} - y
\]

\[
\ddot{y} + 3\dot{y} + 2\dot{y} + y = u
\]

Click simulation and use plot(t,y), you will get a time response of \(y\)

- The parameters can be easily changed;
- The initial condition can be easily changed.

**Simulink for linear systems**

Main components with dynamics:
- integrators,
- state-space description (A,B,C,D)
- transfer function
- derivative (rarely used)

The first two components need initial conditions

Math components:
- addition (a+b+c); product (a\times b);
- dot (inner) product \(<x,y>\);
- gain (amplifier) \(kx : x\) a scalar
- matrix gain \(Kx : x\) a vector
Sources: input signals
- constant, step, ramp
- pulse, sine wave, square wave
- from data file
- signal generator

Sinks: for output demonstration or storage
- digital display
- scope
- save to file
- export to workspace
- XY graph

Nonlinear: functions and operations
- saturation, deadzone, switch

Signals and systems:
- Demux: input a vector signal and output all the components
- Mux: input a bunch of scalar signals and output a vector signal

Functions and tables:
- input u → output y: y = f(u); f composed from available functions or operations; e.g., y = \sin(u_1) + u_1 u_2
- matlab function: y = f(u); f written by a matlab file
- look-up table.
**Example:** Find the solution to the LTI systems

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
-1 & -2 & -3
\end{bmatrix}
\begin{bmatrix}
x \\
u
\end{bmatrix}
\begin{bmatrix}
0 \\
1
\end{bmatrix}
\]

\[
y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x
\]

where \( x(0) = 0; \) \( u(t) \) is a square wave.

**Steps:**
1. Open matlab workspace
2. Type simulink and return
   - simulink library browser window is open
3. Click file and choose new then choose model
   - a blank window is open
4. Open one of the commonly used blocks and drag and drop whatever you need to the blank window.
5. Connect the components by arrows.

**First approach: use state-space description:**

- Click each component to setup the parameters properly.
- Sinks labeled “t”, “u”, “y”: choose “array” for save format.
- Sampling time can be a parameter inputted from workspace.
- When ready, click simulation and choose configuration parameters to setup simulation time. Finally, click simulation and choose start.
- When finished, type \texttt{plot(t,y,t,u)} to plot the input and output.
Second approach: use integrators and amplifiers:

You can make any kind of changes to the model: Change the parameters, the sampling time, add some nonlinear component such as a saturation:
Simulation for nonlinear system:

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

\[ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \]

Matlab function:

```matlab
function dx=fun1(v)
x1=v(1);
x2=v(2);
xn=v(n)
u1=v(n+1);
u2=v(n+2);
um=v(n+m);
dx(1)=f1(x1,...,u1,...)
dx(2)=f2(x1,...,u1,...)
dx(n)=f3(x1,...,u1,...)
```

Simulation for a two-link pendulum

\[ \begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= \frac{g}{l_1} \sin x_1 + \frac{m_2 g}{m_1 l_1} \cos x_3 \sin (x_3 - x_1) - 0.2 x_2 \\
\dot{x}_3 &= x_4, \\
\dot{x}_4 &= \frac{g}{l_2} \sin x_3 - 0.2 x_4
\end{align*} \]

Matlab function:

```matlab
function dx=ff(x)
g=9.8; m1=1;m2=1;a1=1;a2=1;
x1=x(1);x2=x(2);
x3=x(3);x4=x(4);
dx2=-(g/a1)*sin(x1)+(m2*g/(m1*a1))*cos(x3)*sin(x3-x1)-0.2*x2;
dx4=-(g/a2)*sin(x3)-0.2*x4;
dx=[dx2;dx4]
```
At this point, it is time to give a summary on what we have achieved and what will be studied.

Main Problems of the Course

- Analysis: Solutions to LTI systems, stability etc.
- Controllability and observability;
- Feedback design and construction of observers
- Optimal control
- Lyapunov stability

Course project will involve feedback design of an inverted pendulum system.
- Design a feedback law through the linearized system
- Apply the feedback law to the nonlinear system
- Use simulink to check if desirable performance requirements are satisfied.
Course Project

A cart with an inverted pendulum (page 22, Chen’s book)

The control problems are
1: Stabilization: Design a feedback law $u = Fx$ such that $x(t) \to 0$ for $x(0)$ close to the origin.
2: For $x(0) = (0,0,\pi,0)$, apply an impulse force ($u(t) = u_{\text{max}}$ for $t \in [0,0.1]$) to bring $\theta$ to a certain range and then switch to the linear controller so that $x(t) \to 0$.

Assume that there is no friction or damping. The nonlinear model is as follows.

\[
\begin{bmatrix}
M + m & ml \cos \theta \\
\cos \theta & l
\end{bmatrix}
\begin{bmatrix}
\dot{y} \\
\dot{\theta}
\end{bmatrix}
= \begin{bmatrix}
u + ml\theta^2 \sin \theta \\
g \sin \theta
\end{bmatrix}
\]

$m = 1kg$ : mass of the pendulum
$l = 0.2m$ : length of the pendulum
$M = 5kg$ : mass of the cart, $g = 9.835$

Linearize the system at $x=0$;

\[
\begin{bmatrix}
M + m & ml \\
1 & l
\end{bmatrix}
\begin{bmatrix}
\dot{y} \\
\dot{\theta}
\end{bmatrix}
= \begin{bmatrix}
u \\
g \theta
\end{bmatrix}
\]

The state space description for the linearized system

\[
\dot{x} = Ax + Bu
\]

Problems:
1. Find matrices $A$, $B$ for the state space equation.
2. Design a feedback law $u = F_1x$ so that $A + BF_1$ has eigenvalues at $-2\pm j2; -4$ and $-8$. Build a simulink model for the closed-loop linear system.
   Plot the response under initial condition $x(0) = [1.5,0,1,-3]$.
3. Build a simulink model for the original nonlinear system, verify that stabilization is achieved by $u = F_1x$ when $x(0)$ is close to the origin.
   Find the maximal $\theta_0$ so that the nonlinear system can be stabilized from $x(0) = (0,0,\theta_0,0)$.
4. For $x(0) = (0,0,\pi/5,0)$, compare the response $y(t)$ and $\theta(t)$ for the linearized system and the nonlinear system under the same feedback $u = F_1x$. 
5. Assume that the initial condition is \( x(0) = (0, 0, \pi, 0) \).
   For the nonlinear system, construct a switching law to bring the pendulum upward and stabilized at \( x = 0 \). (cart still at \( y = 0 \), pendulum inverted, \( \theta = 0 \)).
   An initial impulse control is applied with \( u(t) = u_{\text{max}} \) for \( t \in (0, t_0] \) and \( u(t) = 0 \) for \( t \geq t_0 \). After the angle is within a small range, i.e., \( |\theta| \leq \theta_d \), switch to a linear controller \( u = F_2 x \).
   Find \( u_{\text{max}}, t_0, \theta_d \) and \( F_2 \) so that the following requirements are satisfied:
   1) \( |y(t)| \leq 1 \) for all \( t > 0 \) or keep the maximal \( y \) as small as possible.
   2) \( |y(t)| \leq 0.02 \) for \( t > 2.5 \).
   3) \( |u| \leq 150 \) for all \( t > 0 \).

Note:
In all the simulation, please choose a fixed sampling period: 0.001 second

Some guidelines:

The simulink model

Use a matlab function to realize the nonlinear/linear model

Use a matlab function to realize the switching control law

You may use a not so good control law to check if you simulink model is built correct.

\[
u = \begin{bmatrix} 0.7071 & 3.1831 & 125.5455 & 16.5057 \end{bmatrix}^T \\
= 0.7071y + 3.1831y + 125.5455 \dot{\theta} + 16.5057 \ddot{\theta}
\]
We are going to learn how to design a good control law.

Before that, we need to study

- **Controllability and observability**;

We need some background on linear algebra:
- positive-definiteness of a square matrix.

They are also essential to Lyapunov stability and optimal control.
Quadratic functions and positive-definiteness (§3.9)

Given a symmetric matrix $P=P^\top$ ($p_{ij}=p_{ji}$).

A quadratic function can be defined as

$$V(x) = x'Px$$

Example:

$$V_1(x) = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = ax_1^2 + 2bx_1x_2 + cx_2^2$$

$$V_2(x) = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} a & d & e \\ d & b & f \\ e & f & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = ax_1^2 + bx_2^2 + cx_3^2 + 2dx_1x_2 + 2ex_1x_3 + 2fx_2x_3$$

For higher order vector spaces, $V(x) = x'Px = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}x_ix_j$

Definition:

A symmetric matrix $P$ is said to be positive definite, denoted by $P > 0$, if $x'Px > 0$ for all $x \neq 0$. It is said to be positive semidefinite, denoted by $P \geq 0$, if $x'Px \geq 0$ for all $x$.

• Under what condition is $V(x)=x'Px$ positive definite?
• This depends on the eigenvalues of $P$.

Compare the eigenvalues of

$$P_1 = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \quad \text{and} \quad P_2 = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

$\{a+jb, a-jb\}$ for $P_1$, $a + c \pm \sqrt{(a-c)^2 + 4b^2}$ for $P_2$. \[42\]
**Theorem:** A real symmetric matrix has real eigenvalues.

**Proof:** Suppose that $\lambda$ is an eigenvalue, possibly complex, $v$ is the eigenvector such that $Pv = \lambda v$. The complex conjugate transpose of $v$ is $v^*$, the complex conjugate transpose of $P$ is $P'$. We have

$$(v^*Pv)^* = v^*P^*v = v^*P'v = v^*Pv$$

$v^*Pv$ must be a real number. Also recall that $v^*v$ is a real number. From $Pv = \lambda v$, we have

$$v^*Pv = \lambda v^*v \quad \Rightarrow \quad \lambda \text{ must be a real number}$$

**Theorem:** A real symmetric matrix $P$ is always diagonalizable.

More than diagonalizable, the transformation matrix has nice properties.

**Definition:** A square matrix $A$ is orthogonal if all columns of $A$ are orthonormal:

Let $A = [a_1, a_2, \ldots, a_n]$. $A$ is orthogonal if

$$a_j \cdot a_k = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases}$$

$A'A = ?$  $AA' = ?$

$A'A = I$,  $A^{-1} = A'$,  $AA' = I$.

- If $A$ is orthogonal, then $\|Ax\|_2 = \|x\|_2$ for all $x \in \mathbb{R}^n$.

**Proof:** $\|Ax\|_2 = \langle Ax, Ax \rangle^{1/2} = \langle x, A'Ax \rangle^{1/2} = \langle x, x \rangle^{1/2} = \|x\|_2$

- An orthogonal matrix is also called a unitary matrix.
- $\|A\|_2 = 1$, $x \rightarrow Ax$ transforms a unit ball to a unit ball.
**Theorem:** For every symmetric matrix $P$, there exists an orthogonal matrix $U$ and a real diagonal matrix $D$, such that

$$U'PUD = \iff P = UDU'$$

Equivalently, $PU = UDU' = UDU$,

$$P[u_1 \ u_2 \ldots u_n] = [u_1 \ u_2 \ldots u_n] \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d_{nn} \end{bmatrix},$$

$$Pu_j = d_j u_j$$

- The diagonal elements of $D$ are the eigenvalues of $P$;
- The columns of $U$ are the eigenvectors of $P$.

---

**Example:** copied from matlab workspace

$$P = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

$$P = UDU' = UDU'$$

$$D = \begin{bmatrix} -0.6235 & 0 & 0 \\ 0 & 0.0000 & 0 \\ 0 & 0 & 9.6235 \end{bmatrix}$$

$$U'U = I$$

$$>> [U,D]=eig(P)$$

$$U = \begin{bmatrix} 0.8277 & 0.4082 & 0.3851 \\ 0.1424 & -0.8165 & 0.5595 \\ -0.5428 & 0.4082 & 0.7339 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.6235 & 0 & 0 \\ 0 & 0.0000 & 0 \\ 0 & 0 & 9.6235 \end{bmatrix}$$

$$>> U'PU$$

$$1.0e-014 *$$

$$\begin{bmatrix} 0.0111 & -0.0416 & -0.0167 \\ -0.0609 & 0.0719 & 0.0155 \\ 0 & -0.0888 & 0.5329 \end{bmatrix}$$

$$>> U'U$$

$$\begin{bmatrix} 1.0000 & -0.0000 & -0.0000 \\ -0.0000 & 1.0000 & 0.0000 \\ -0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

$$U'U = I$$
Theorem: A symmetric matrix $P$ is positive definite ($P > 0$) if and only if its eigenvalues are all positive.

Proof: There exist diagonal matrix $D$ and orthogonal matrix $U$ such that $P = UDU'$.  
Consider the quadratic form $z'Dz$. Have 
$$z'Dz = d_1z_1^2 + d_2z_2^2 + \ldots + d_nz_n^2 > 0 \text{ for all } z \neq 0$$  
Let $z = U'x$. $z = 0$ iff $x = 0$. Hence  
$$x'Px = x'UDU'x = z'Dz > 0 \text{ for all } x \neq 0$$

Theorem: A symmetric matrix $P$ is positive definite iff there exists a nonsingular matrix $N$ such that $P = NN'$.  
Proof: ....

In summary:  
Given a symmetric matrix $P$.  
- All the eigenvalues and eigenvectors are real.  
- Exists a matrix $U$, $UU' = U'U = I$, and a diagonal $D$, such that $P = UDU'$.  
- $P$ is positive definite iff 
  - all eigenvalues are positive;  
  - exists nonsingular $N$ such that $P = NN'$;  
- $P$ positive semi-definite iff 
  - all eigenvalues are non-negative;  
  - exists $N$ such that $P = NN'$;  
- $P$ negative definite iff 
  - all eigenvalues are negative;  
  - exists nonsingular $N$ such that $P = -NN'$
An additional result on symmetric matrix:

**Theorem:** Let $M(t) \in \mathbb{R}^{n \times n}$ be a symmetric matrix function. Suppose that every element of $M(t)$ is an analytic function, i.e., $m_{ij}(t)$ can be differentiated infinitely many times. Then the eigenvalues of $M(t)$ can be arranged as $\lambda_1(t), \lambda_2(t), \ldots, \lambda_n(t)$ where each $\lambda_i(t)$ is analytic.

If $M(t)$ is nondecreasing, i.e., $M(t_1) \leq M(t_2)$, for any $t_1 < t_2$, then $\lambda_1(t), \lambda_2(t), \ldots, \lambda_n(t)$ are nondecreasing.

---

**Today:** some miscellaneous problems about LTI systems
- How to deal with complex eigenvalues
- Realization of a transfer function
- Simulation of systems by using Simulink
- Course project
- Quadratic functions and positive-definiteness

**Next Time:** Chapter 6.
- Controllability and Observability
Problem set #7:

1. Use the first definition of a matrix function to compute $e^{At}$ for

$$A = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix}$$

2. Use matlab to find the real block diagonal form and transformation matrix for the following:

$$A = \begin{bmatrix} -2 & -4 & -3 & -3 \\ -3 & 2 & -4 & -1 \\ 3 & 5 & 3 & 2 \\ 3 & -1 & 6 & 3 \end{bmatrix}$$

i.e., find a real $Q$ and a real block diagonal $D$ such that $D=Q^{-1}AQ$.

3. Find a state space realization for

$$G(s) = \frac{s^3 + 2s^2 + 3s + 4}{s^4 + 3s^3 + 4s^2 + 4s + 2}.$$

Use integrators and amplifiers to construct a Simulink model for it. Let the input be a step signal: $u(t)=0$ for $t<0$ and $u(t)=2$ for $t > 0$. Choose the sampling time to be $T=0.1$. Simulate the output under 0 initial condition and plot the output response for $t=0$ to $t=15$. (print the model and the output response). You can try different input signals.
4. For the matrix
\[
A(s) = \begin{bmatrix}
1 & 2 & 3 \\
2 & 2s & 4 \\
3 & 4 & s
\end{bmatrix}
\]
As \(s\) is increased, all the eigenvalues of \(A(s)\) increases. Use matlab to find the least integer \(s\) such that \(A(s)\) is positive definite.

5. Construct the simulink model on page 32 (two link pendulum) and run simulation from \(t=0\) to \(t=20\), with initial condition \(x(0)=(0.5,0,1,0)\).