CDS 101/110 Recitation

Daniel Linqi Guo

10/16/2015

Overview

- Stability check
- State-space in MATLAB
- Simulation using ode45
- Rise time
- Discrete-time system

Stability Check

- A linear system

$$\frac{dx}{dt} = Ax$$

is asymptotically stable if and only if all eigenvalues of A have negative real part

- Can use eig(A) to find eigenvalues in MATLAB/Python

Example

- Consider the system given by

$$\frac{dx}{dt} = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} x$$

- To determine its stability, one uses the following commands:
  - A = [1 , 1; -1, -2];
  - evalues = eig(A);

State-space in MATLAB

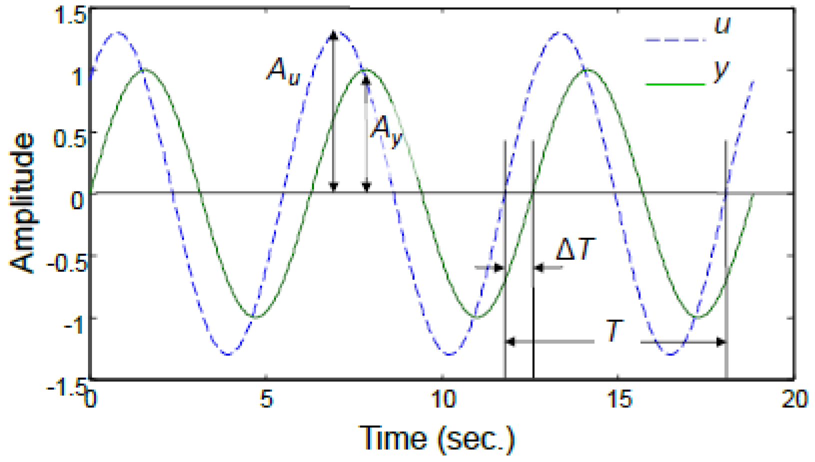
- MATLAB has built-in class for a linear system:

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du$$

- To create an object:
  - sys = ss(A,B,C,D)
- To find its step response, Bode plot and frequency response:
  - step(sys)
  - bode(sys)
  - freqresp(sys)

Bode plot

- A convenient way to graph the frequency response
- For linear system, the response to a sinusoidal input is always sinusoidal with the same frequency



Bode plot

- In Bode plot
  - Gain is plotted on log-log scale
  - Phase is plotted on linear-log scale

- Log scale:
  - A way to "normalize" large and small values to a comparable scale
    - $dB = 20\log_{10}(\cdot)$
  - Sometimes people also use $dB = 10\log_{10}(\cdot)$ depending on the context

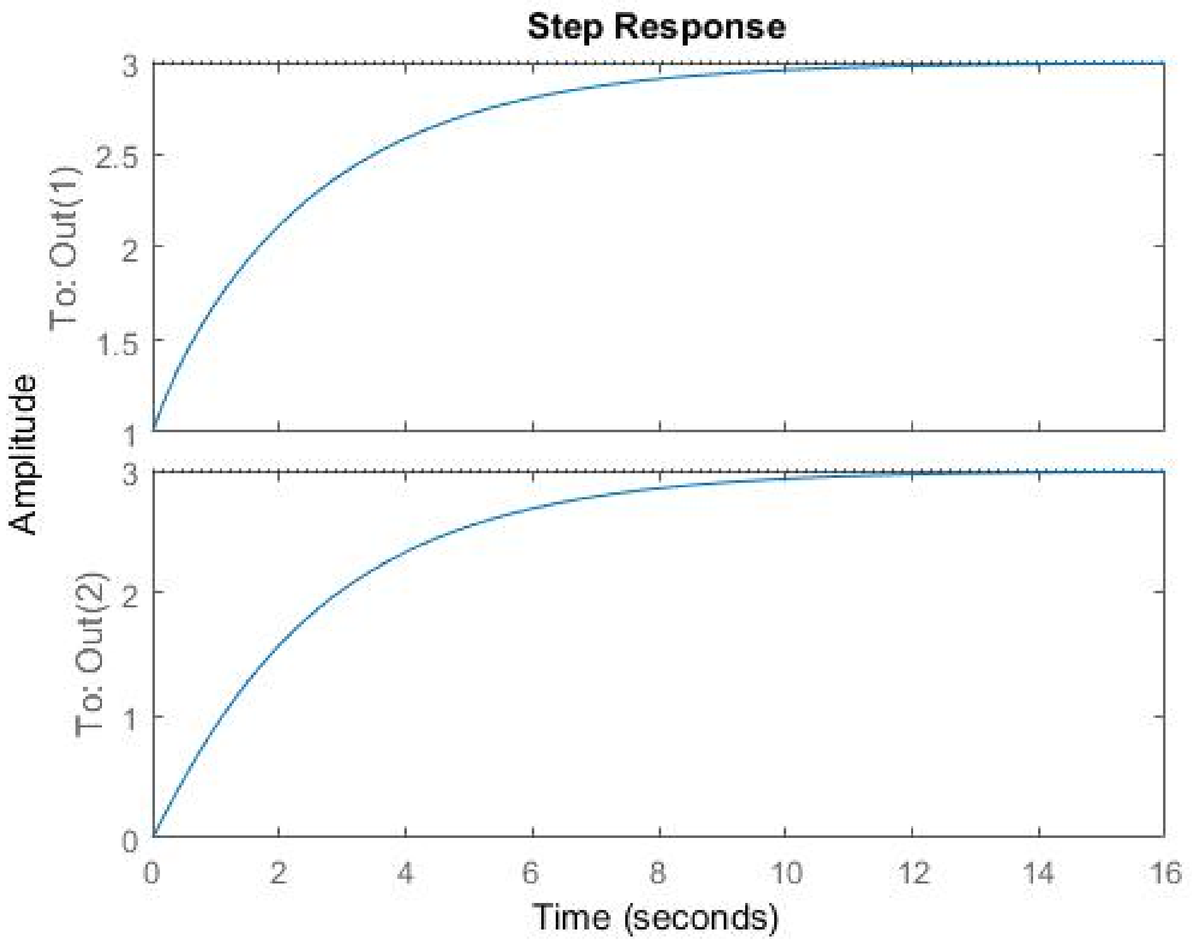
Example

- Consider the system given by

$$\frac{dx}{dt} = \begin{bmatrix} -2 & 1 \\ -1 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

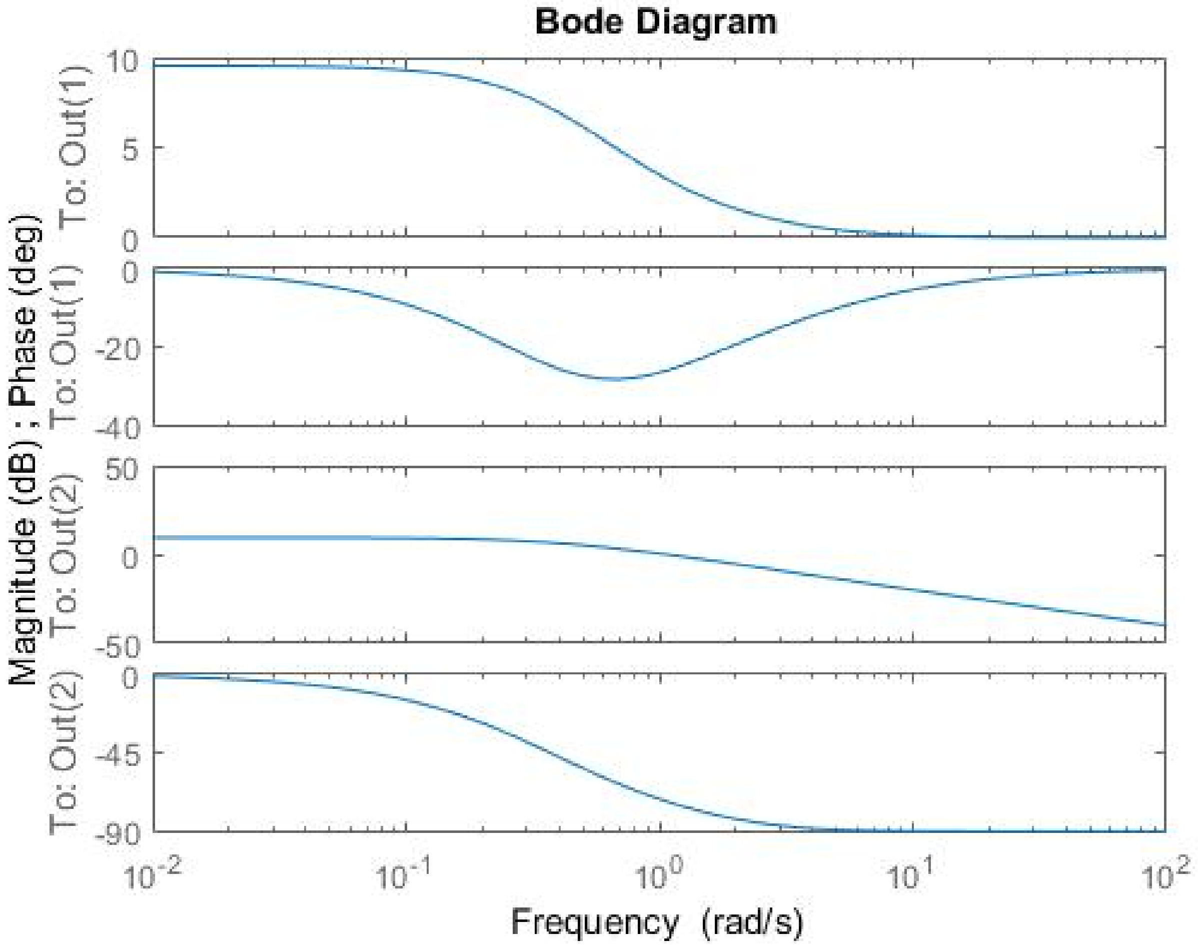
- Illustrative command
  - A = [-2 , 1; 1, -1]; B = [1;1]; C = eye(2); D = [1; 0];
  - sys = ss(A,B,C,D);
  - step(sys);
  - bode(sys);
  - freqresp(sys);

Example



Step Response
Amplitude
To: Out(1)
To: Out(2)
3
2.5
2
1.5
1
3
2
1
0
0
2
4
6
8
10
12
14
16
Time (seconds)

Example



Bode Diagram
Magnitude (dB) ; Phase (deg)
To: Out(1)
To: Out(1)
To: Out(2)
To: Out(2)
10
5
0
0
-20
-40
50
0
-50
0
-45
-90
10^-2
10^-1
10^0
10^1
10^2
Frequency (rad/s)

Simulation using ode45

- ode45 can simulate a general differential equation:

$$\dot{x} = f(t, x)$$

- Three inputs:
  - The function rule $f(t, x)$
  - Time span
  - Initial condition

Example

- Say we want to solve

$$\frac{dx}{dt} = x - 2x^3 + t, \qquad x(0) = 1$$

Example

- First, we define the function in a separate file

$$f(t,x) = x - 2x^3 + t$$

Editor - C:\Users\daniel\Desktop\Sample\update.m*

update.m*

```matlab
function out = update(t,x)
%Sample dynamical system f(t,x) = x-2x^3+t

out = x - 2 * x^3 + t;

end
```

Example

- Then, we use the following command:

Command Window

```
>> [t,y] = ode45(@update,[0 20],1)
```

  - @update is the function handle (pointer)
  - [0 20] is the time span
  - 1 is the initial condition

Rise time

- The time required for a signal to change from a specified low value to a specified high value
- In lecture, we use 5% to 95%
- In homework, we use 10% to 90%.......
- Both are fine, as long as you state clearly the definition you are using!

Example

- Suppose the step response is given by

$$V = V_0(1 - e^{-t/\tau})$$

- Final value = 1
- The 5% time is given by

$$0.05 = \; 1 - e^{-t_1/\tau}$$

- The 95% time is given by

$$0.95 = \; 1 - e^{-t_2/\tau}$$

- The rise time is then

$$t_r = t_2 - t_1 = \tau \cdot \ln(19) \approx 2.94\tau$$

Discrete-Time System

- Time domain chosen to be the integers
- Continuous-time versus discrete-time

	Continuous-Time	Discrete-Time
Specification	$\dot{x} = \lambda x$	$x[n+1] = \lambda x[n]$
Operator	$\frac{d}{dt}: x \mapsto \dot{x}$	$D: x[n] \mapsto x[n+1]$
Eigen-function	$\exp(\lambda t)$	λ^n

- Results in continuous case usually can find counterparts in discrete case

Discrete-Time System

- For problem 4b, note the matrix A has a full basis of eigenvectors
- You may want to write the initial state as a linear combination of the eigenvectors
- The conclusion is still true without this condition, but the proof is more involved

Questions?