Please note

There is no lecture on Monday 5th March since I am away at a conference

I will be away from the laboratory sessions on:

- Thursday 1th March
- Monday 5th March
- Thursday 8th March

The lab sessions will still run and Jake will be available for assistance

I have already uploaded the first part of the assessed coursework

ROCO218: Control Engineering
Dr Ian Howard

Lecture 5

Coursework assessments

Coursework assessment

- There will be practical assessments in the form of laboratory practical work
  - 40% ROCO218
  - 50% ROCO223
- Your complete lab report must be submitted via the DLE by

Thursday 4pm, 17th May 2018

- There will also be a final exam for the course
  - 60% ROCO218
  - 50% ROCO223

- The total mark for this coursework will be the sum of the marks for each individual report with each report contributing to the total coursework mark. The mark for each report will be awarded as indicated in the coursework sheet

Important note on assessed coursework practical

You are required to write a short report on each of the assessed **coursework.** This should describe:

- The task or problems you were asked to solve.
- An explanation of your solution.
- Your results and findings.
- Your insights and conclusions.
- You need to work in pairs for the practical sessions but submit a report **INDIVIDUALLY!**
- Please also look at: StudentxxxROCO218223ReportTemplate.doc

Report requirements:

- The report must only be a single PDF document of *several hundred* words per practical, although this is just a rough suggestion.
- You are strongly encouraged to include photos, diagrams, plots and even linked videos to support your work.
- Embed any code in-line the report (do not use appendices)
- There is a **150MB limit** on the final submission (but your PDF should be much less than this).
- Please check that your submitted files are correct by downloading them again and checking that the content is correct

Report format

Please only submit your overall report as a PDF.

When you write your report, you must include you student number on every page of the report in a header and also include it in the document filename!

Each laboratory report should contain a few pages of written explanation (although it can be more than this) as well as a set of images and diagrams to explain your solutions to the practical exercises.

Embed any code in-line the report (do not use appendices)

Please also include embedded links to short video to illustrate operation of your project if you use videos

Please also look at:
StudentxxxROCO218223ReportTemplate

ROCO218: Control Engineering
Dr Ian Howard

Lecture 5

Introduction to state space models

PID verses state space methods

PID control can be very successful

- But assumes one input signal and one output signal!
- Transfer functions only represent input and out relationships
- Do not handle systems that have several inputs and outputs (MIMO) well
- Especially if there is strong cross coupling within the system!
- Typically, tuning such systems using intuition and experience results in no more than mediocre performance

We need rigorous mathematical methods!

- A state space model is a description of system dynamics using state variables.
- Formulated using matrix notation
- Supports multivariable systems and control (MIMO)
- Works with time varying systems
- Works with non-linear systems (after linearization)
- Under certain conditions, “variables inside the system” (states) can be:

  Controlled without actuators

  Observed without sensors
- State space approach handles optimality and robustness

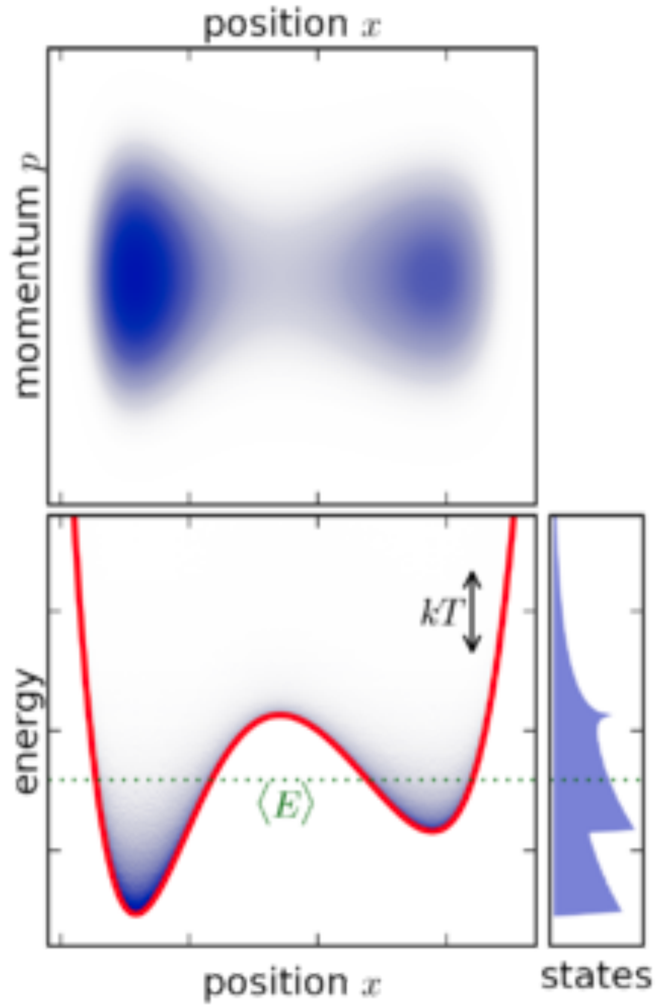
State space variables

The state of a dynamic system often directly describes the distribution of internal energy in the system

- i.e. internal energy can always be computed from the state variable).

For example, it is common to select the following as state variables:

- Position (potential energy)
- Velocity (kinetic energy)
- Capacitor voltage (electric energy)
- Inductor current (magnetic energy)
- The state provides us with important internal information
- Analysis then makes use of the concepts of
  - Stability
  - Controllability
  - Observability



Addresses realization issues of:

- Design
- Linear Quadratic Regulator
- Kalman Filter
- State feedback
- Using Observers

State space system representation

- A state space representation describes a system as a set of 1st order ODEs
- We want to describe the system dynamics with a matrix equation of the form

$$\dot{X} = AX + BU$$

Where

$\dot{X}$ Is the time derivative of the state vector

A Is the system matrix

X Is the state vector

B Is the control input matrix

U Is the control input

State space output representation

- Similarly we want to describe the output with a matrix equation of the form

$$Y = CX + DU$$

Where

Y Is the output

C Is the output or observation matrix

X Is the state vector

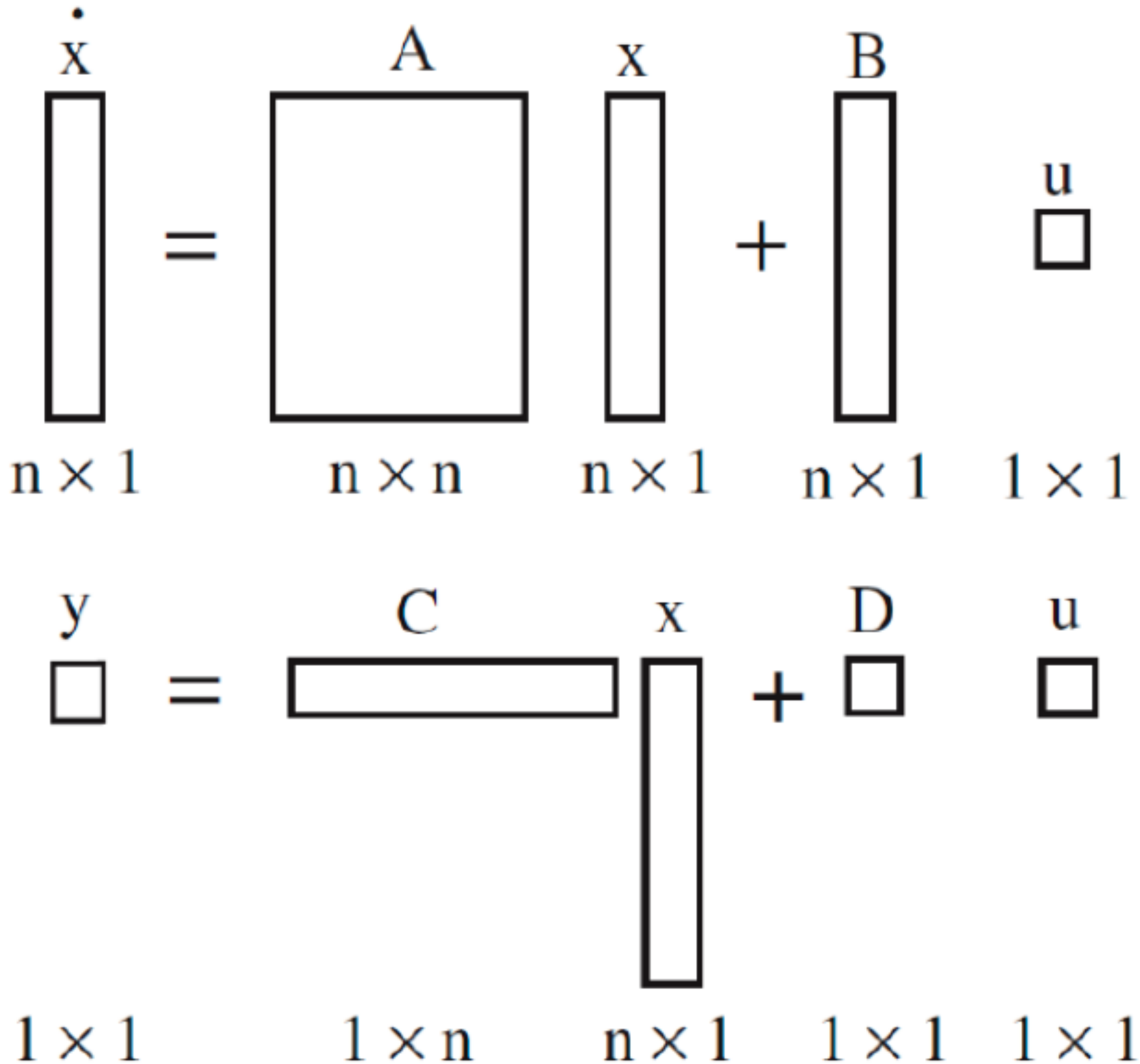
D Is the transmission matrix

U Is the control input

State space SISO representation summary

State Equation

Output Equation



Matrix Products for a SISO and Linear Time-Invariant System

State space representation

We need to write a high order ODE as set of 1st order ODEs

$$\dot{X} = AX + BU$$
$$Y = CX + DU$$

For example we can rewrite a second-order differential equation as a set of two first-order differential equation.
Consider the following general second-order differential equation:

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

Where the dot denotes differentiation w.r.t. time
We can then define y as

$$y = \dot{x}$$

Differentiation of y w.r.t. time, we obtain:

$$\dot{y} = \ddot{x} \qquad = f(x, \dot{x})$$

Therefore, we end up with a system of two first-order differential equations:

$$\Rightarrow \dot{x} = y$$
$$\Rightarrow \dot{y} = f(x, y)$$

State space representation

Similarly, if we consider a general third-order differential equation of the form:

$$\dddot{x} = f(x, \dot{x}, \ddot{x})$$

Then we can define

$$y = \dot{x}$$

$$z = \dot{y} = \ddot{x}$$

Which after time differentiation gives:

$$\dot{z} = \dddot{x} \quad = f(x, \dot{x}, \ddot{x}) \quad = f(x, y, z)$$

Therefore, we end up with a system of three first-order differential equations:

$$\dot{x} = y$$

$$\dot{y} = z$$

$$\dot{z} = f(x, y, z)$$

The same strategy works with an nth-order differential equation that can be transformed into a set of n first-order differential equation.

ROCO218: Control Engineering
Dr Ian Howard

Lecture 5

Mapping 2nd order ODE to state space

2^{nd} order ODE to state space: Easy case

Consider the differential equation

$$\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_2 y = b_1 u$$

Re-writing with highest derivative on the LHS we have

$$\frac{d^2y}{dt^2} = -a_1\frac{dy}{dt} - a_2 y + b_1 u$$

This is a 2^{nd} order equation!
Therefore highest derivative state variable must be less than 2^{nd} order – i.e. 1^{st} order

Choosing the state variables

$$x_1 = y \qquad \Rightarrow \dot{x}_1 = \frac{dy}{dt}$$

$$x_2 = \frac{dy}{dt} \qquad \Rightarrow \dot{x}_1 = x_2$$

First state equation

$$\Rightarrow \dot{x}_2 = \frac{d^2y}{dt^2}$$

2^{nd} order ODE to state space: Easy case

Substituting

Remember

$$\dot{x}_2 = \frac{d^2y}{dt^2} \quad \text{into} \quad \frac{d^2y}{dt^2} = -a_1\frac{dy}{dt} - a_2y + b_1u \qquad x_1 = y \qquad x_2 = \frac{dy}{dt}$$

$$\Rightarrow \dot{x}_2 = -a_1\frac{dy}{dt} - a_2y + b_1u \qquad \Rightarrow \dot{x}_2 = -a_1x_2 - a_2x_1 + b_1u$$ Second state equation

Therefore we have the state equations

$$\boxed{\begin{aligned} &\Rightarrow \dot{x}_1 = x_2 \\ &\Rightarrow \dot{x}_2 = -a_1x_2 - a_2x_1 + b_1u \end{aligned}}$$

Writing state space form

$$\dot{X} = AX + BU \quad \text{and} \quad Y = CX + DU$$

This leads to the dynamics equation

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b_1 \end{bmatrix}u$$

And the output equation

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

2^{nd} order ODE to state space: Harder case

Consider the differential equation

$$\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_2 y = b_0\frac{du}{dt} + b_1 u$$

We now have as differential term on the RHS
We need to choose state variable so this is cancelled out!

Re-writing with highest derivative on the LHS

$$\frac{d^2y}{dt^2} = -a_1\frac{dy}{dt} - a_2 y + \boxed{b_0\frac{du}{dt}} + b_1 u$$

If we choose the state variables as before

$$x_1 = y \qquad \Rightarrow \dot{x}_1 = \frac{dy}{dt} \qquad x_2 = \frac{dy}{dt} \qquad \Rightarrow \dot{x}_2 = \frac{d^2y}{dt^2}$$

Substituting

$$\dot{x}_2 = \frac{d^2y}{dt^2}$$ into $$\frac{d^2y}{dt^2} = -a_1\frac{dy}{dt} - a_2 y + b_0\frac{du}{dt} + b_1 u$$

$$\Rightarrow \dot{x}_2 = -a_1\frac{dy}{dt} - a_2 y + \boxed{b_0\frac{du}{dt}} + b_1 u$$

We will still have as differential term on the RHS We don't want this!
We are not allowed differential terms on the RHS of the matrix equations

2^{nd} order ODE to state space: Harder case

So starting with the re-written equation

$$\frac{d^2y}{dt^2} = -a_1\frac{dy}{dt} - a_2y + b_0\frac{du}{dt} + b_1u$$

Now adding same term to both sides

$$\frac{d^2y}{dt^2} - b_0\frac{du}{dt} = -a_1\frac{dy}{dt} - a_2y + b_0\frac{du}{dt} + b_1u - b_0\frac{du}{dt}$$

We now choose the state variables

$$x_1 = y \qquad \Rightarrow \dot{x}_1 = \frac{dy}{dt}$$

$$x_2 = \frac{dy}{dt} - b_0u \qquad \Rightarrow \dot{x}_2 = \frac{d^2y}{dt^2} - b_o\frac{du}{dt}$$

Substituting

$$\dot{x}_2 = \frac{d^2y}{dt^2} - b_o\frac{du}{dt}$$

Into

$$\frac{d^2y}{dt^2} - b_0\frac{du}{dt} = -a_1\frac{dy}{dt} - a_2y + b_0\frac{du}{dt} + b_1u - b_0\frac{du}{dt}$$

The differential u term is now cancelled out!

$$\Rightarrow \dot{x}_2 = -a_1\frac{dy}{dt} - a_2y + b_0\frac{du}{dt} + b_1u - b_o\frac{du}{dt} \qquad \Rightarrow \dot{x}_2 = -a_1\frac{dy}{dt} - a_2y + b_1u$$

2^{nd} order ODE to state space: Harder case

Since

$$x_2 = \frac{dy}{dt} - b_0 u \qquad \Rightarrow \frac{dy}{dt} = x_2 + b_0 u$$ and $$x_1 = y$$ Output equation

So the equation

$$\dot{x}_2 = -a_1 \frac{dy}{dt} - a_2 y + b_1 u \qquad \Rightarrow \dot{x}_2 = -a_1\left(x_2 + b_0 u\right) - a_2 x_1 + b_1 u$$

$$\Rightarrow \dot{x}_2 = -a_1 x_2 - a_2 x_1 + \left(b_1 - a_1 b_0\right) u$$ Second state equation

Again since $$\frac{dy}{dt} = x_2 + b_0 u$$

$$\Rightarrow \dot{x}_1 = \frac{dy}{dt}$$

$$\dot{x}_1 = x_2 + b_0 u$$ First state equation

2^{nd} order ODE to state space: Harder case

Thus this time we have the state equations

$$\boxed{\begin{aligned} \dot{x}_1 &= x_2 + b_0 u \\ \dot{x}_2 &= -a_1 x_2 - a_2 x_1 + \left(b_1 - a_1 b_0\right) u \\ y &= x_1 \end{aligned}}$$

This leads to the dynamics equation

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 - a_1 b_0 \end{bmatrix} u$$

And the output equation

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

ROCO218: Control Engineering
Dr Ian Howard

Lecture 5

Eigenvalues and Eigenvectors

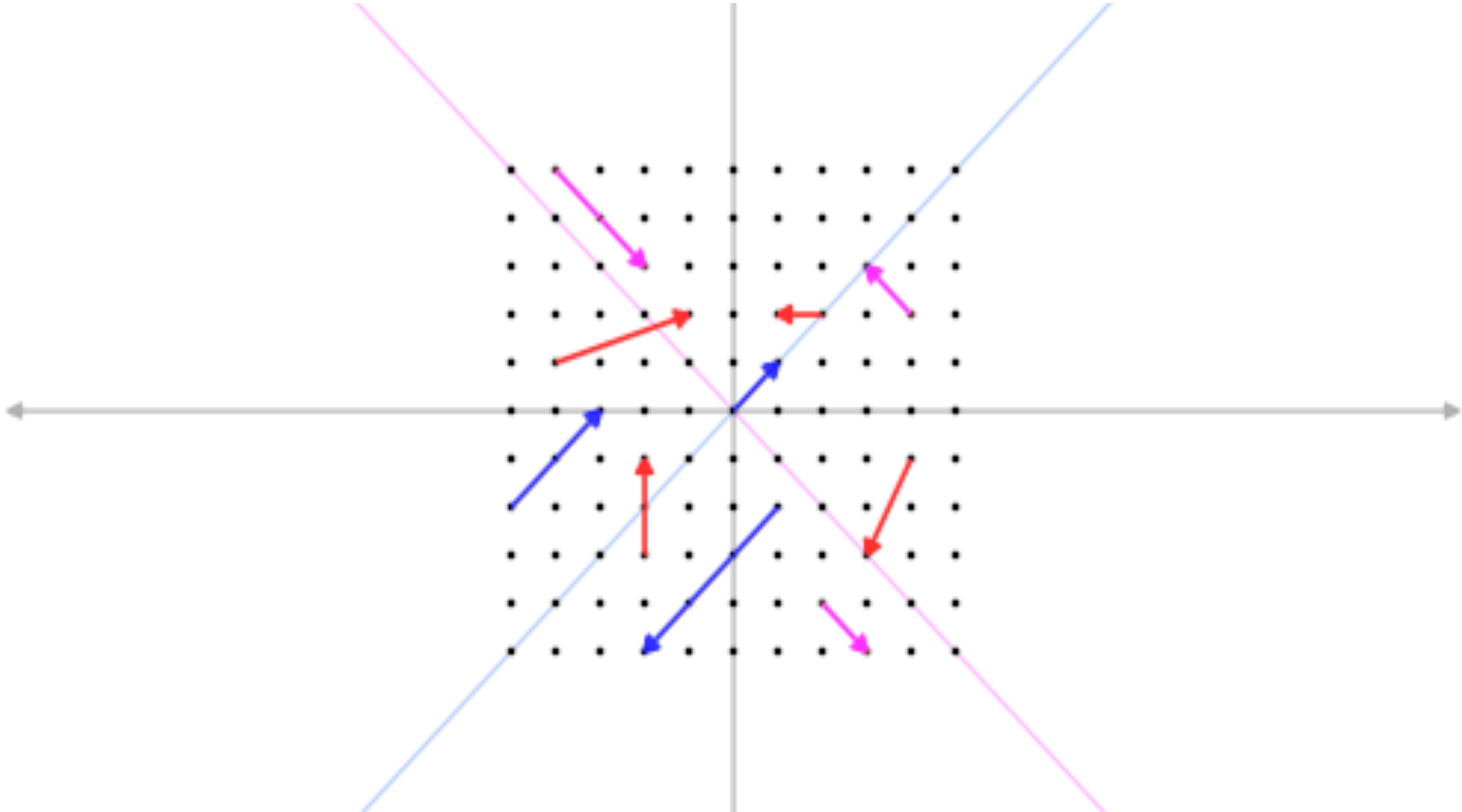
Eigenvalue problems

Consider simultaneous equations of the form:

$$AX = \lambda X$$

- where A is an n x n matrix, x is an n x 1 column vector and λ is a scalar constant
- The solution to the equation is a vector X
- When we transform X by multiplying by A we end up with vector X again but this times scaled by λ
- Therefore the direction of vector X is unaffected by the transformation

Example eigenvector from Wikipedia



- Transformation matrix only preserves the direction of the blue and purple vectors that are parallel to the diagonals
- These are the eigenvectors
- It does not preserve the direction of the red vectors

Finding eigenvalues and eigenvectors

Given the matrix A. e.g. where $A = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$

We will now find its eigenvalues λ and eigenvectors

We must solve the equation $AX = \lambda X$

$$\Rightarrow (A - \lambda I)X = 0$$

There is a trivial solution if the matrix (A-λ) has an inverse since

$$\Rightarrow (A - \lambda I)^{-1}(A - \lambda I)X = 0 \quad \Rightarrow X = 0$$

We want the non-trivial solution!
If there is no inverse then det(A-λ)=0

$$\Rightarrow \det(A - \lambda I) = 0$$ Also written as $$\left|(A - \lambda I)\right| = 0$$

Finding eigenvalues and eigenvectors

Evaluating

$$\det\left(A-\lambda I\right)=0$$

$$\Rightarrow \det\left(\begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix}-\lambda\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)=0$$

$$\Rightarrow \det\begin{vmatrix} 1-\lambda & 0 \\ 1 & 2-\lambda \end{vmatrix}=0$$

Remember

$$\boxed{\det(A)=\left|\begin{pmatrix} a & b \\ c & d \end{pmatrix}\right|=\left(ad-bc\right)}$$

$$\Rightarrow \left(1-\lambda\right)\left(2-\lambda\right)=0$$

This is the characteristic equation of the A matrix

The characteristic equation has solutions that indicates the eigenvalues of the system

$$\lambda=1,\lambda=2$$

Finding eigenvalues and eigenvectors

When $\lambda = 1$

$$\Rightarrow AX = X \qquad \Rightarrow \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Giving

$$x_1 = x_1$$

$$x_1 + 2x_2 = x_2 \quad \Rightarrow x_1 = -x_2 \qquad \Rightarrow X = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

So eigenvectors are proportional to this X vector: e.g.

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 \\ -2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix}, etc$$

Can normalize

$$\Rightarrow X = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Finding eigenvalues and eigenvectors

When $\lambda = 2$

$$\Rightarrow AX = 2X \qquad \Rightarrow \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Giving

$$\begin{aligned} x_1 &= 2x_1 \\ x_1 + 2x_2 &= 2x_2 \end{aligned} \qquad \Rightarrow x_1 = 0 \qquad \Rightarrow X = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Normalized already

So eigenvectors are proportional to this X vector: e.g.

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix}, etc$$

The two solutions

So the the matrix A

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$$

Has the two eigenvalues λ given by

$$\lambda = 1, \lambda = 2$$

And the two associated eigenvectors

$$X_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad X_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

ROCO218: Control Engineering
Dr Ian Howard

Lecture 5

Transfer function of a state space model

Mapping state space models to transfer functions

From state space equations with system dynamics

$$\dot{X} = AX + BU$$

And output relations

$$Y = CX + DU$$

Taking Laplace transformation of system equation assuming zero initial conditions

$$sX(s) = AX(s) + BU(s)$$

Collecting the X(s) terms on the LHS

$$\Rightarrow (sI - A)X(s) = BU(s)$$

NB: need sI here since this is matrix algebra and I is the identity matrix

Multiplying both sides by inverse of (sI-A)

$$\Rightarrow X(s) = (sI - A)^{-1} BU(s)$$

Mapping state space models to transfer functions

Taking Laplace transformation of output equation assuming zero initial conditions

$$Y(s) = CX(s) + DU(s)$$

Substituting in expression for X(s):

$$X(s) = (sI - A)^{-1} BU(s)$$

$$\Rightarrow Y(s) = C(sI - A)^{-1} BU(s) + DU(s)$$

Collecting the U(s) terms on the RHS

$$\Rightarrow Y(s) = \left[C(sI - A)^{-1} B + D \right] U(s)$$

This leads to an expression for the transfer function G(s)

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

Mapping state space models to transfer functions

The expression for the transfer function G(s)

$$G(s) = C(sI - A)^{-1}B + D$$

Expanding the inverse matrix term

$$G(s) = C\left(\frac{adj(sI - A)}{\det(sI - A)}\right)B + D$$

We see that the transfer function has “poles” and goes to infinity when the denominator term is zero, that is when

$$\det(sI - A) = 0$$

Therefore:

The poles of the transfer function correspond to the eigenvalues of the matrix (sI-A)

Example state space model to transfer functions

Given the state space system

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} X + \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} U \qquad Y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} X$$

Transfer function is given by

$$G(s) = C(sI - A)^{-1} B + D$$

Where

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

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

There is no D term and

$$B = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example state space model to transfer functions

And

$$sI - A = s\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} = \begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 1 & 2 & s+3 \end{bmatrix}$$

$$\Rightarrow (sI - A)^{-1} = \begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 1 & 2 & s+3 \end{bmatrix}^{-1}$$

$$\Rightarrow (sI - A)^{-1} = \frac{adj\left(\begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 1 & 2 & s+3 \end{bmatrix}\right)}{\det\left(\begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 1 & 2 & s+3 \end{bmatrix}\right)}$$

This starts to get tricky to evaluate by hand
So lets use Matlab!

Example state space model to transfer functions

```
% symbolic variables
syms a b c d e f g h i
syms s;

% enter A matrix
a = s;
b = -1;
c=0;
d=0;
e=s;
f=-1;
g=1;
h=2;
i=s+3;

% build uo matrix from elements
A = [a b c; d e f; g h i];

% calculate determinant
determinant = det(A);

% calculate adjuct
adjunct = det(A)*inv(A);

disp('A');
disp(A);
disp('determinant');
disp(determinant);
disp('adjunct');
disp(adjunct);
```

Solve using Matlab symbolic processing

```
A
[ s, -1,     0]
[ 0,  s,    -1]
[ 1,  2, s + 3]

determinant
s^3 + 3*s^2 + 2*s + 1

adjunct
[ s^2 + 3*s + 2,      s + 3,   1]
[            -1, s*(s + 3),    s]
[            -s, - 2*s - 1, s^2]
```

$$\Rightarrow (sI - A)^{-1} = \frac{\begin{bmatrix} \left(s^2+3s+2\right) & \left(s+3\right) & 1 \\ -1 & s(s+3) & s \\ -s & -2s & s^2 \end{bmatrix}}{\left(s^3 + 3s^2 + 2s + 1\right)}$$

Example state space model to transfer functions

Since transfer function is given by

$$G(s) = C(sI - A)^{-1}B + D$$

$$\Rightarrow G(s) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \frac{\begin{bmatrix} \left(s^2+3s+2\right) & \left(s+3\right) & 1 \\ -1 & s(s+3) & s \\ -s & -2s & s^2 \end{bmatrix}}{\left(s^3 + 3s^2 + 2s + 1\right)} \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

Solve using Matlab symbolic processing

```
C = [1 0 0];
B = [10; 0; 0;];
G =  C * inv(sImA) * B;
disp('G');
disp(G)
```

```
(10*(s^2 + 3*s + 2))/(s^3 + 3*s^2 + 2*s + 1)
```

$$\Rightarrow G(s) = \frac{10(s^2 + 3s + 2)}{\left(s^3 + 3s^2 + 2s + 1\right)}$$

Eigenvalues and stability

- For the system

$$\dot{X} = AX + BU$$

- The stability of the equilibrium is examined by calculating the eigenvalues λ of the system matrix A.

- In particular this is achieved by solving the expression

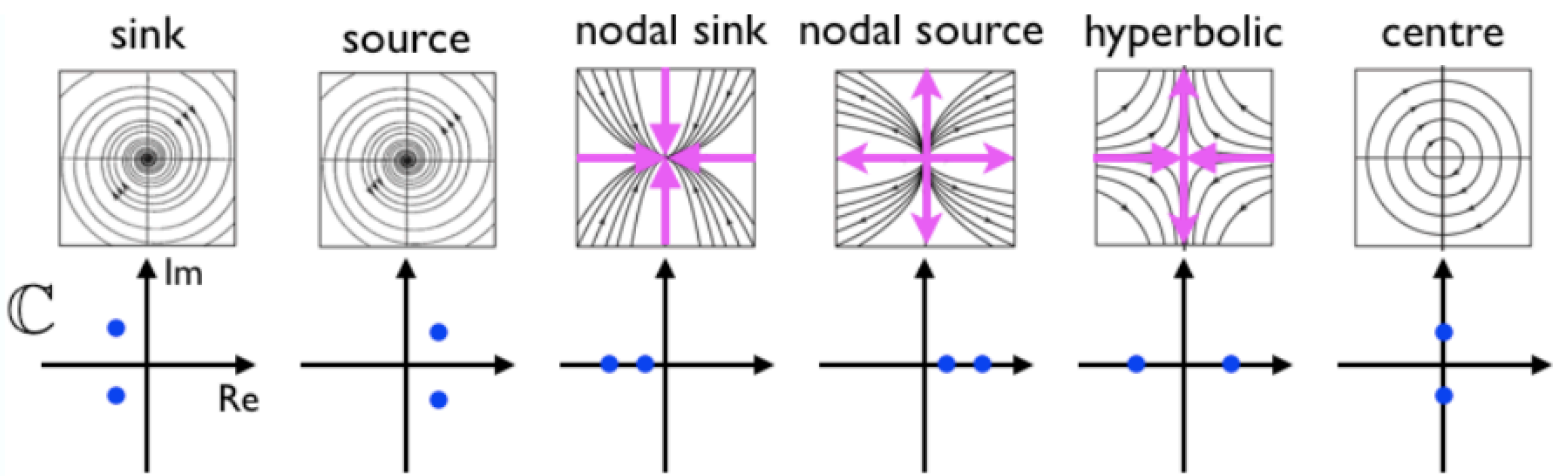
$$\det(A - \lambda I) = 0$$

- The eigenvalues of the system relate to its poles and will determine if the equilibriums are stable or not

- In a 2-dimensional systems, the eigenvalues will have the general form: = a ± ib (i.e. complex numbers).

Eigenvalues and stability

- If eigenvalues has a negative real part, the system will have a component that is stable
- If eigenvalues has a positive real part, the system will have a component that is unstable
- If the real part is zero, then the system is neutral - neither stable nor unstable
- If the eigenvalues have a complex part there are oscillations



ROCO218: Control Engineering
Dr Ian Howard

Lecture 5

State feedback control

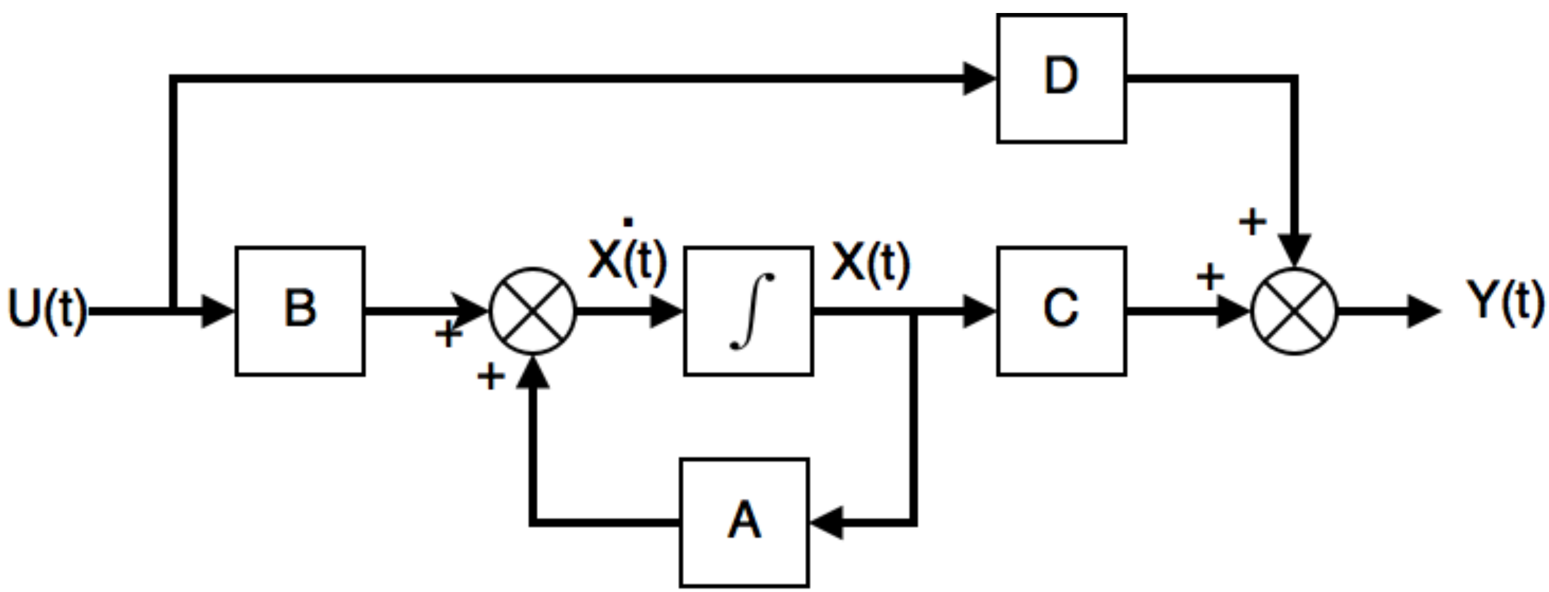
State space feedback control

- State space model is described by the following equations

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

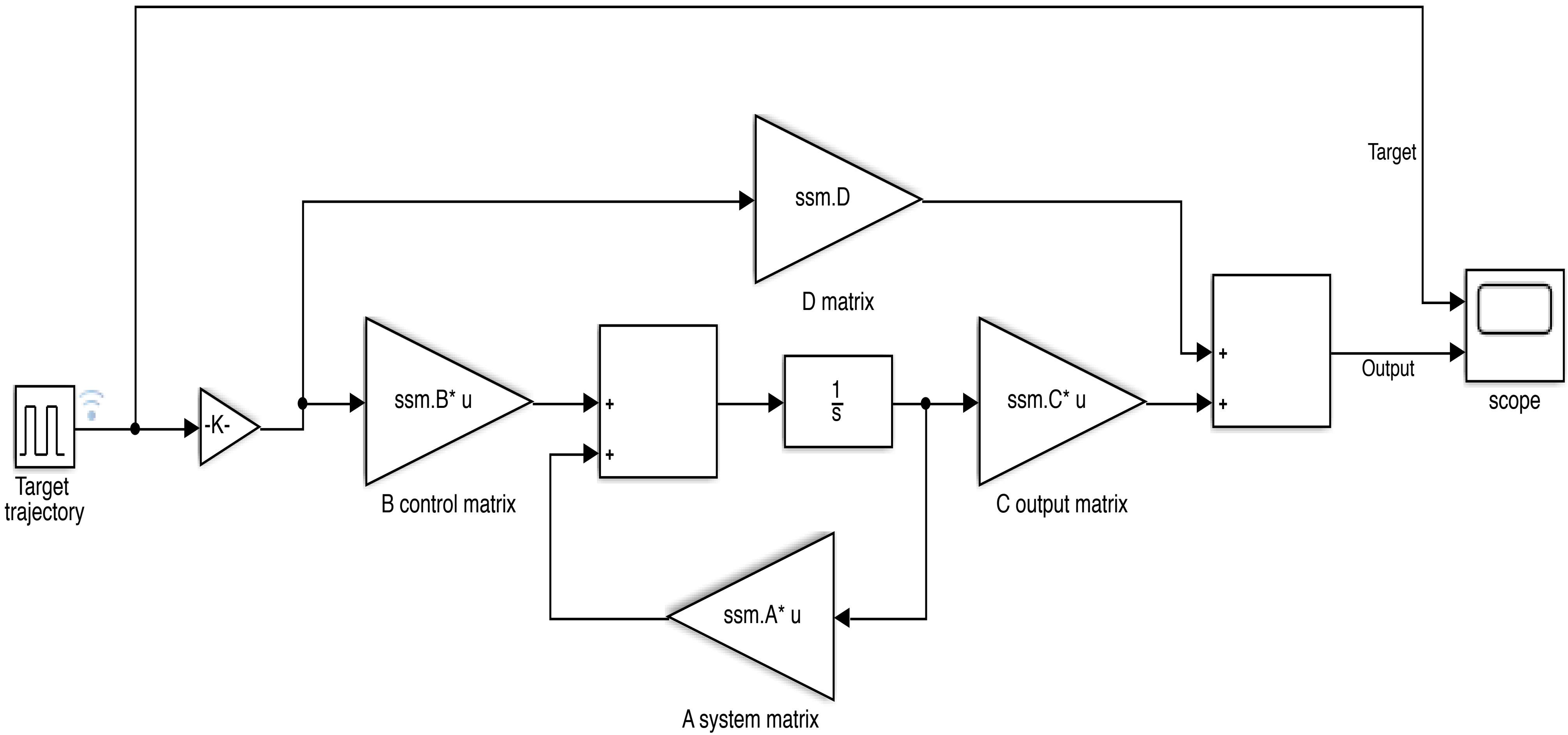
- Where in general we have a control input on the LHS as shown below



Simulink simulation of a state space model

$$\dot{X} = AX + BU$$

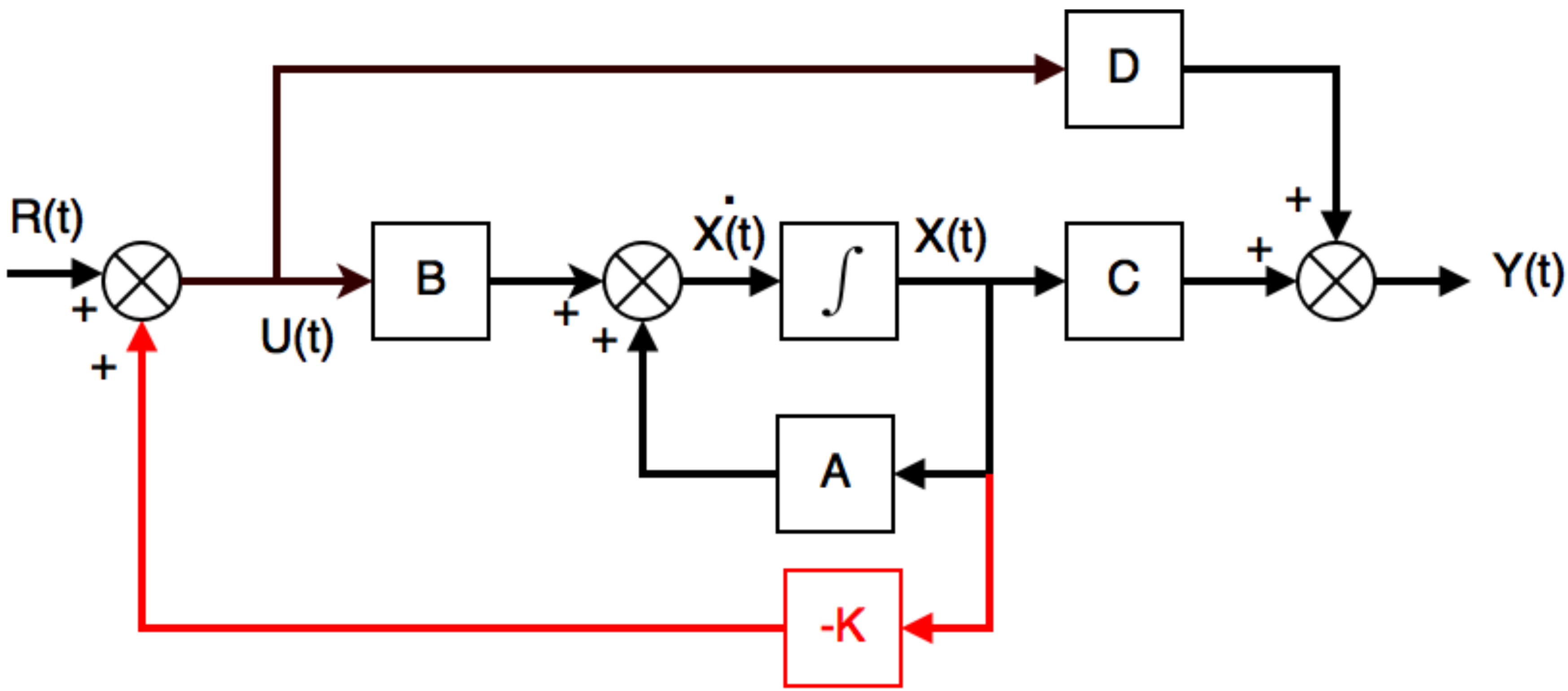
$$Y = CX + DU$$



State space feedback control

- To implement feedback control of this system we must use some form of feedback in the system
- One option is to use state feedback
- This is represented by the expression $U = -KX$

- The feedback matrix K represents feedback gain of the system state and R(t) represent a reference input
- The state feedback path is depicted in red



State space feedback control

- Substituting in $U = -KX$

$$\Rightarrow \dot{X} = AX + BU \quad = AX - BKX \quad = \left(A - BK\right)X$$

$$\Rightarrow Y = CX + DU \quad = CX - DKX = \left(C - DK\right)X$$

- Looking at the the expression

$$\dot{X} = \left(A - BK\right)X$$

- we see than the close- loop stability now depends on eigenvalues λ given by the characteristic equation

$$\left|\lambda I - \left(A - BK\right)\right| = 0$$

- Thus we can influence the location of the eigenvalues of the system by changing the gain matrix K

State space feedback control summary

- In an open loop system

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

- Stability determined by location of poles which are the eigenvalue of matrix A

- In a closed loop system

$$\dot{X} = \left(A - BK\right)X$$

$$Y = \left(C - DK\right)X$$

- Stability determined by location of poles which are the eigenvalue of matrix (A-BK)

- In a state feedback system, choosing the value of K can enable designer to change the location of the poles of the system

- However in general it is not necessarily trivial to determine K

ROCO218: Control Engineering
Dr Ian Howard

Lecture 6

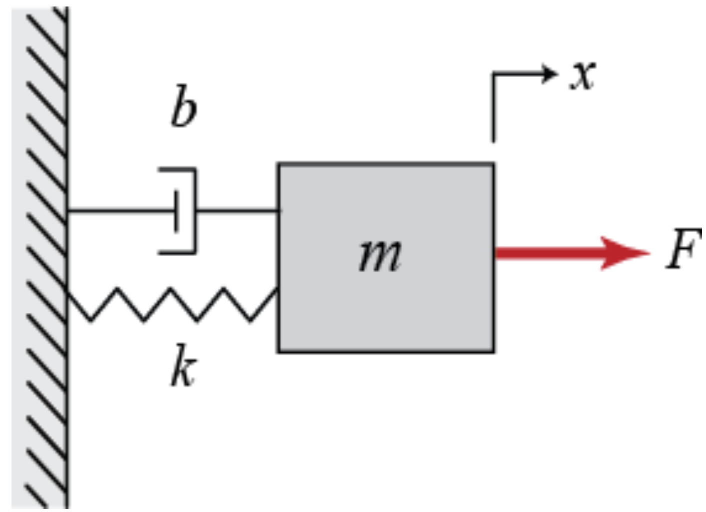
Example: Mass spring damper SFC

Modelling mass spring damper

Consider damped mass on a spring

Where:

m = 1Kg
k = 2 N/m
b = 3 N/ms^{-1}



$$u(t) = m\ddot{x}(t) + b\dot{x}(t) + kx(t)$$

Input force

Opposition by Inertial mass

Viscous resistance

Spring resistance

Mass spring damper state space model

From differential equation

$$u(t) = m\ddot{x}(t) + b\dot{x}(t) + kx(t)$$

Identify its states
The highest state must be less than highest differential eqn term

$$x(t) = x_1$$

$$\dot{x}(t) = x_2 \qquad = \dot{x}_1$$

Also identify the control input

$$u(t) = u_1$$

Mass spring damper state space model

Rewriting differential equation

$$u(t) = m\ddot{x}(t) + b\dot{x}(t) + kx(t)$$

$$\Rightarrow u_1 = m\dot{x}_2 + bx_2 + kx_1$$

$$\Rightarrow m\dot{x}_2 = u_1 - bx_2 - kx_1$$

So given

$$x(t) = x_1$$

$$\dot{x}(t) = x_2 \quad = \dot{x}_1$$

$$u(t) = u_1$$

This gives the state equations

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{u_1}{m} - \frac{kx_1}{m} - \frac{bx_2}{m}$$

Therefore

$$\dot{x}_1 = 0x_1 + 1.x_2 + 0.u_1$$

$$\dot{x}_2 = -\frac{k}{m}x_1 - \frac{b}{m}.x_2 + \frac{1}{m}.u_1$$

Mass spring damper state space model

So from

$$\dot{x}_1 = 0x_1 + 1.x_2 + 0.u_1$$

$$\dot{x}_2 = -\frac{k}{m}x_1 - \frac{b}{m}.x_2 + \frac{1}{m}.u_1$$

In state space matrix in standard form this gives

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}.u_1$$

if output $y = x_1$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}.u_1$$

Mass spring damper uncontrolled behavior

- If m=1, k =2 and b=3, this leads to the numeric matrices

$$\dot{X} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U \qquad \text{So system matrix} \qquad A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

- The eigenvalues λ of this open loop system are given by

$$\det\left(A - \lambda I\right) = 0 \quad \Rightarrow \left\| \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\| = 0$$

$$\Rightarrow \left\| \begin{bmatrix} -\lambda & 1 \\ -2 & \left(-3-\lambda\right) \end{bmatrix} \right\| = 0$$

Remember

$$\boxed{\det(A) = \left| \begin{pmatrix} a & b \\ c & d \end{pmatrix} \right| = \left(ad - bc\right)}$$

This gives the characteristic equation for the system

$$\Rightarrow -\lambda\left(-3-\lambda\right) - \left(-2\right) = 0$$

$$\Rightarrow \lambda\left(3+\lambda\right) + 2 = 0 \quad \Rightarrow \lambda^2 + 3\lambda + 2$$

$$\Rightarrow (\lambda+1)(\lambda+2) = 0 \qquad \Rightarrow \lambda = -1, -2$$

Here Eigenvalues are –ve and real therefore the system is stable

Mass spring damper state feedback control

- From the mass spring damper system

$$\dot{X} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U \quad \text{so} \quad A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Applying state feedback control we have

$$U = -KX \quad \text{where} \quad K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

- The eigenvalues λ of this closed loop system are given by

$$\det\left(A - BK - \lambda I\right) = 0$$

The term

$$BK = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix}$$

Mass spring damper state feedback control

- Therefore $\det\left(A-BK-\lambda I\right)=0$

$$\Rightarrow 0=\left|\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}-\begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix}-\lambda\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right| = \left|\begin{bmatrix} -\lambda & 1 \\ \left(-2-k_1\right) & \left(-3-k_2-\lambda\right) \end{bmatrix}\right|$$

This gives the characteristic equation for the system

$$\Rightarrow\left(-\lambda\right)\left(-3-k_2-\lambda\right)-\left(-2-k_1\right)=0 \quad \Rightarrow\lambda\left(3+k_2+\lambda\right)+\left(2+k_1\right)=0$$

$$\Rightarrow\lambda^2+\left(3+k_2\right)\lambda+\left(2+k_1\right)=0$$

- If say we want eigenvalues at λ =-1,-5

This gives the required characteristic equation

$$\Rightarrow\left(\lambda+1\right)\left(\lambda+5\right)=0 \qquad \Rightarrow\lambda^2+6\lambda+5=0$$

- We see our system can achieve this if

$$\left(3+k_2\right)=6 \quad \Rightarrow k_2=3$$

$$\left(2+k_1\right)=5 \quad \Rightarrow k_1=3$$

So choosing gain appropriately we can out the eigenvalues where we want them in this case

Mass spring damper Matlab step response simulation

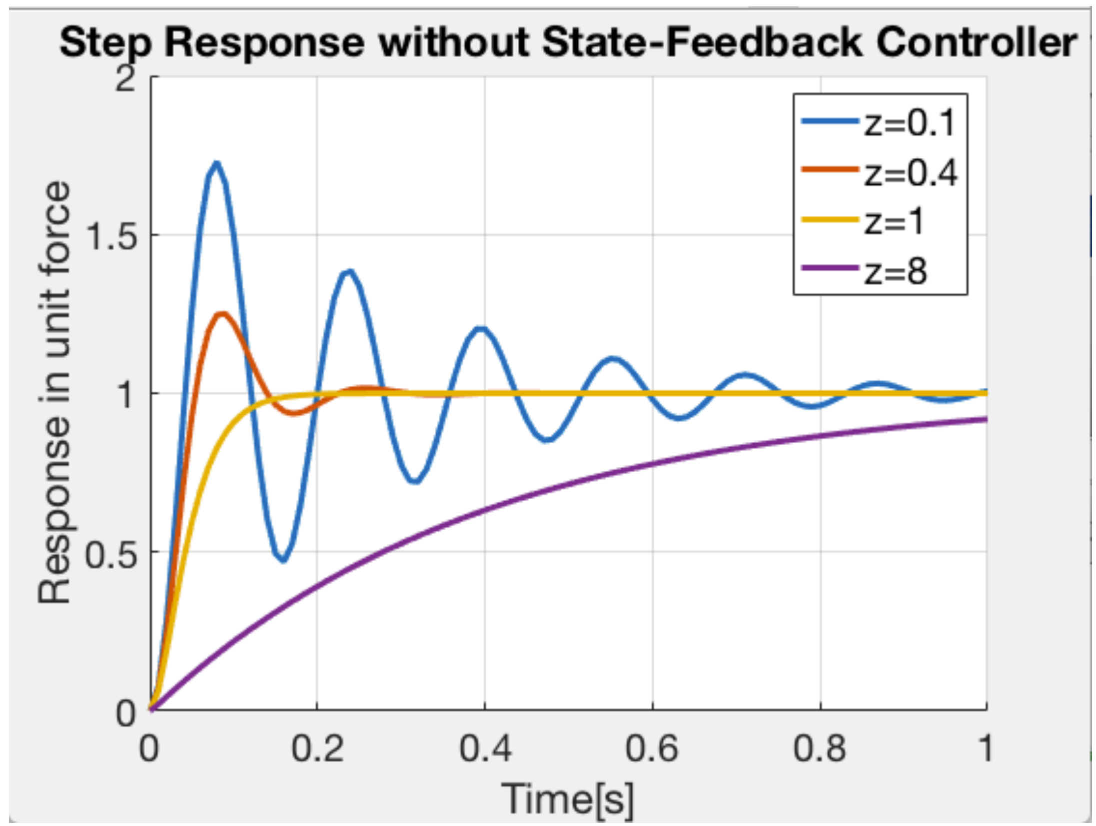
Here we simulate the mass spring damper response to step force input for a range of damping ratio values.

```
p.zeta(1) = 0.1;
p.zeta(2) = 0.4;
p.zeta(3) = 1;
p.zeta(4) = 8;
```

We then sets appropriate values of m, k and mu to achieve these zeta values

```
% get state space matrices
ssm.A = [0 1; -p.k/p.m -p.mu/p.m ];
ssm.B = [0; 1/p.m;];
ssm.C = [1 0;];
ssm.D = [];
```

```
% get response of uncontrolled system
sys = ss(ssm.A,ssm.B,ssm.C,ssm.D);
t = 0:0.01:1;
[Y,T] = step(sys,t);
response(:,tidx) = Y;
time = T;
```

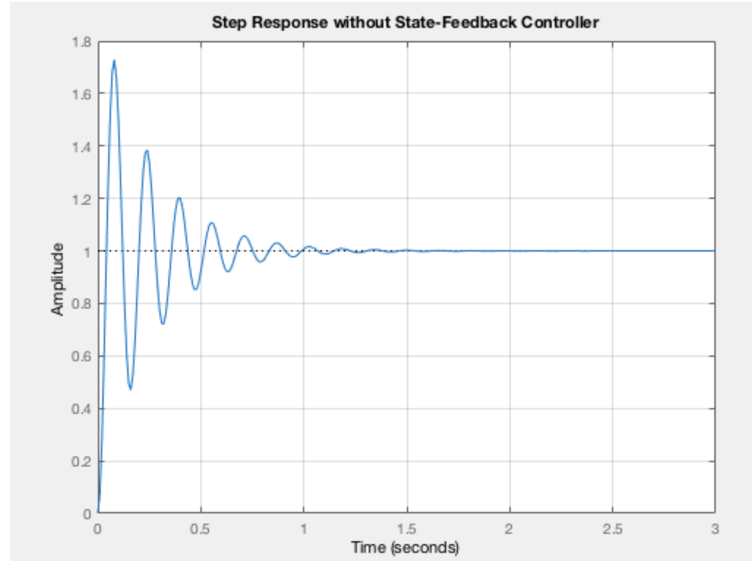


Mass spring damper uncontrolled/SFC response

Now we choose params to give an underdamped uncontrolled response

```
ssm.A =  [0 1; -1600 -8];
ssm.B =  [0; 1600];
ssm.C =  [1 0;];
ssm.D = 0;

% get response of uncontrolled system
sys = ss(ssm.A,ssm.B,ssm.C,ssm.D);
figure
t = 0:0.01:1;
step(sys,t)
grid
title('Open Loop Step Response')
```




```
% for state space model with x, xDot
% compute SFC gains to set eigenvalues
PX=8 * [-1 -1.1 ];
ssm.KPLACE = place(ssm.A, ssm.B, PX);
disp(ssm.KPLACE);

% get response of SFC system
figure
sys_cl = ss(ssm.A-ssm.B*ssm.K,ssm.B,ssm.C,ssm.D);
step(sys_cl,t)
grid
title('Step Response with State-Feedback Controller')
```