InnoTecUK™

# Modelling and Control

Professor Gurvinder Singh Virk

Technical Director, InnotecUK

Gurvinder.virk@innotecUK.com

# Example references

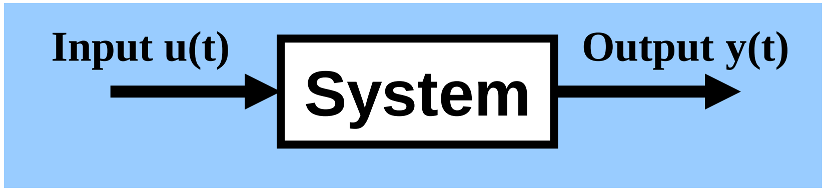
## Following may be useful:

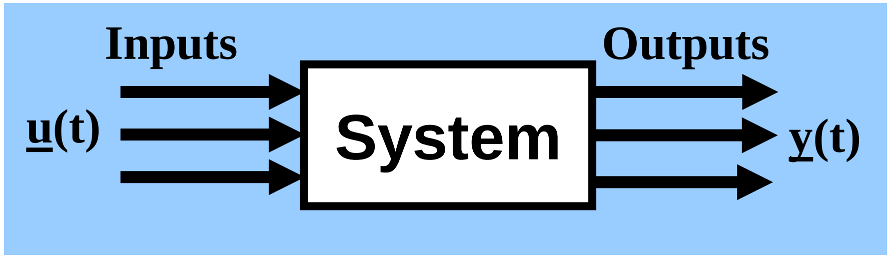
1. Dorf RC & Bishop RH, Modern control systems, 12th ed, Prentice-Hall, 2011
2. Paraskevopoulos PN, Modern control engineering, Marcel Dekker, 2001
3. Franklin GF, Powell JD and Emami-Naeini A, Feedback Control of Dynamic Systems, 6th edition, Prentice-Hall, 2010.
4. Nise NS, Control systems engineering, 4th edition, Wiley, 2004.
5. Dutton K, Thompson S and Barraclough B, The Art of Control Engineering, Addison-Wesley, 1998.
6. Franklin GF, Powell JD and Workman ML, Digital control of dynamic systems, 3rd edition, Addison-Wesley, 1997.
7. Hines W, Matlab Supplement in Fuzzy and Neural approaches in Engineering, Wiley, 1997.
8. Mathworks, Matlab, Simulink, Control Systems and System Identification Toolboxes, available on-line in pdf form, www.mathworks.com

I would like to thank all the colleagues (too many to mention individually) across the world who have helped with the formulation of this lecture on Modelling and Control from material placed on the www

# Control systems basics

- What is Control Engineering?
  - Control is concerned with the area of designing and making systems behave in some defined/ specified manner
- Control systems have inputs which “drive” outputs
  - Simple systems
    - Single input – single output, Linear
  - Complex systems:
    - Multivariable I/Ps – O/Ps
    - Non-linear, stochastic, etc



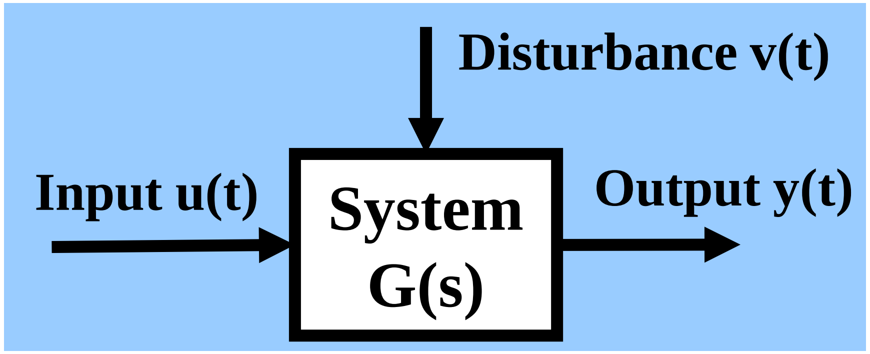


- Input-output related by TF
  - y(t)=Gu(t)

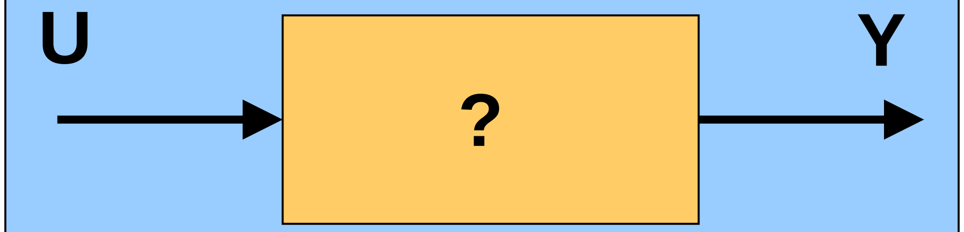
Can control

- System driven by u(t) & v(t)
  u(t), v(t) ⇨ y(s) for s>t
  causal system

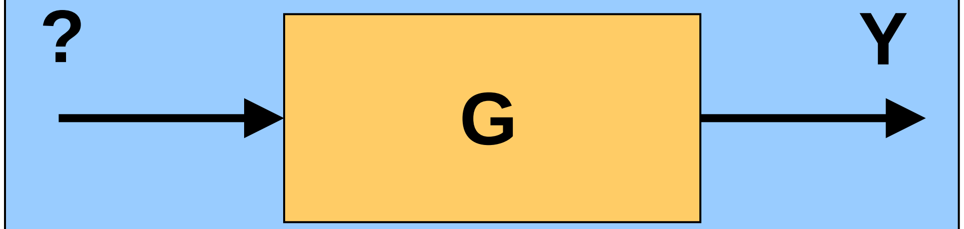
Can’t control?
Can’t measure?



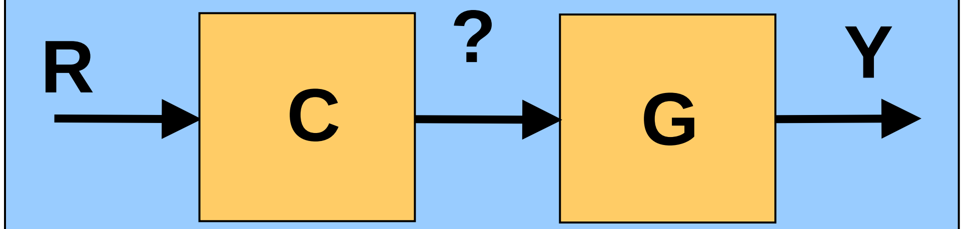
# Problems in control

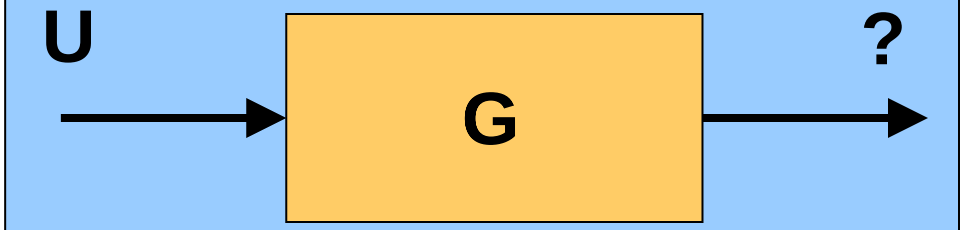


Modelling or synthesis



Controller design
or measurement

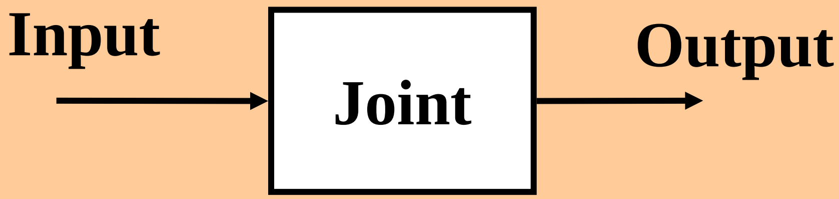




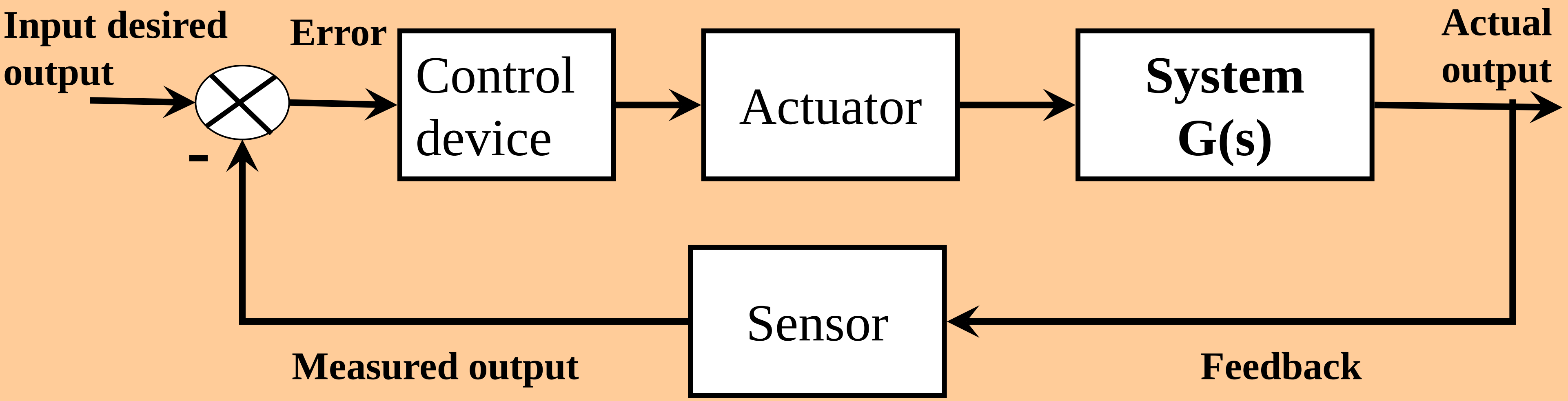
Simulation or analysis

# Control system formats

## Open loop system (eg robot joint)



## Closed loop system



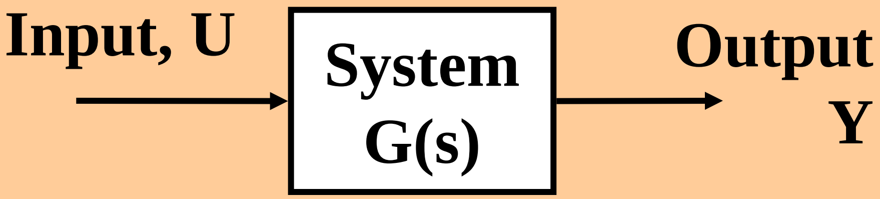
# Open loop – closed loop

Open Loop Control System

Y(s)=G(s)U(s)

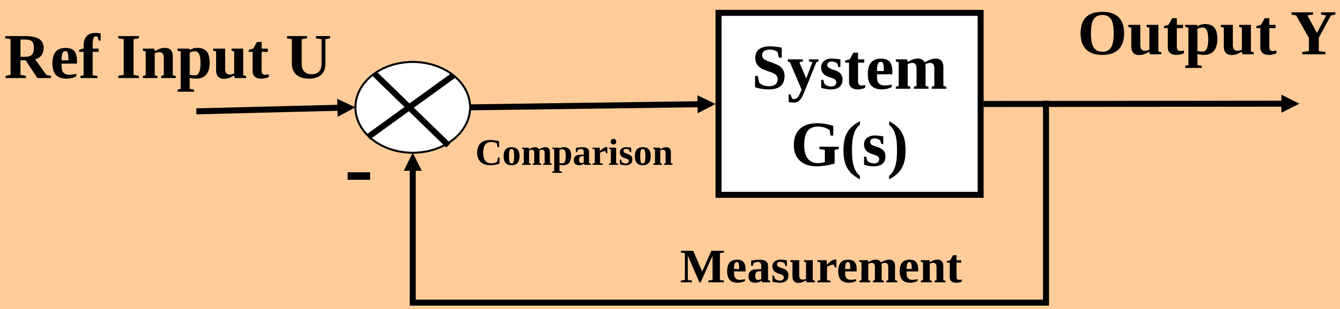
Simple but variable response

Simple gain system example



Closed loop control

Y(s)=G(s)/(1+G(s))



Good behaviour but complex and leads to instability.

Why is behaviour good?

Why does instability arise?

InnoTecUK™

# Videos of control examples

- Examples include
  - Herald of Free Enterprise, Estonia sinking (6 March 1987)
  - Tacoma Fall bridge collapsing (7 November 1940)
  - Hot metal rolling processes
  - Ocean Swell Powered Renewable Energy (OSPREY 1), 1995
  - Inverted pendulum control
  - Magnetic suspension
- Important issues for good control are:
  - Need for good modelling
  - Over-engineering needed in practice
  - Feedback is good?
  - How to get good control performance

Movie

# Approaches in control systems

- **Main analysis and design methods include:**
  - **Open-loop versus closed loop**
  - **Time Domain and Frequency Domain methods**
  - **Traditional analysis/design methods: Bode, Nyquist, Routh, Root locus, PID**
  - **Modern methods: State space, adaptive control, multivariable control, model-based control**
  - **Heuristic methods: Fuzzy logic, neural networks**
- **Continuous and Digital Control systems: Laplace versus z-transfer function**

# Laplace- and z-Transforms

- The solutions of linear homogeneous ODEs is complicated and involve handling exponentials
- Continuous systems:
  - The Laplace transform operation reduces exponentials to simple algebraic expressions
  - The Laplace transform can be used to convert a differential equation into an algebraic equation
  - The solution of the algebraic equation can then be written as the sum of terms, each of which is the Laplace transform of an exponential
  - Inverse Laplace transformation needed to convert back to original domain
- Discrete systems:
  - The z-transform reduces difference equations into algebraic equations and also easy solution via following similar process with Laplace transforms for differential equations

# Laplace example

- **Find the Laplace transform of**

$$f(t) = 5u_{step}(t) + 3e^{-2t}$$

- **Solution is**

$$L\{5u_{step}(t)\} = 5L\{u_{step}(t)\} = \frac{5}{s}$$

$$L\{3e^{-2t}\} = 3L\{e^{-2t}\} = \frac{3}{s+2}$$

$$F(s) = \frac{5}{s} + \frac{3}{s+2} = \frac{8s+10}{s(s+2)}$$

# Matrix algebra

- To be used when scalar variables become vector variables
- Column and row vectors

$$a_{row} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \quad and \quad a_{column} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

- Matrices are collection of row/column vectors
- Dimensions of vectors/ matrices must be equal for matrix operations (eg. multiplication) to be valid (AB ≠ BA).
- Square and non-square matrices

# Matrix definitions

- **A diagonal matrix is a square matrix whose element are all zero except those on the main diagonal**
- **Transposing a matrix converts the rows into columns and vice versa. The transpose of a matrix=$(A)^T$=$A^T$**
- **A symmetric matrix has its elements that are mirror image along the diagonal. The transpose of a symmetric matrix is the same, i.e., $A^T$=A**
- **If the determinant of a square matrix is zero, the matrix is called singular. If the determinant is nonzero, the matrix is called non-singular**

# Matrix operations

- **Matrices can be added, multiplied, etc and have various mathematical operations carried out**
- **Determinant of a matrix can be calculated using $c_{ij}$ are the elements of the conjugate matrix**
- **The inverse of a matrix A is denoted $A^{-1}$ and has the property $A*A^{-1}$=I (Identity)**
- **Eigenvalues and eigenvectors, etc**

$$|A| = \sum_{i=1}^{n} a_{ij} c_{ij}$$

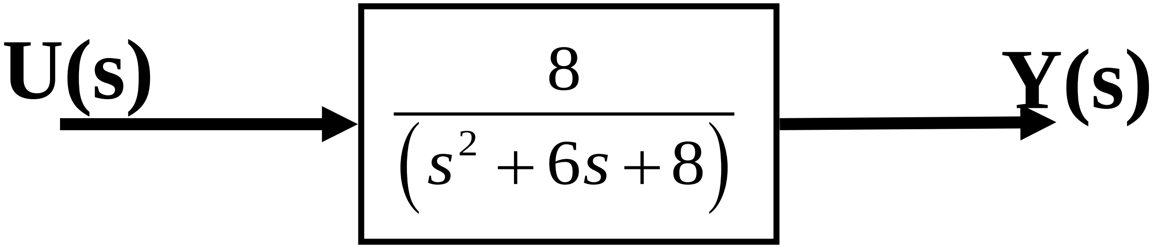
$$|\lambda I - A| = 0 \ \textit{give eigenvalues} \quad \textit{and} \quad Am_i = \lambda_i m_i \ \textit{give eigenvectors}$$

# Digital control systems

- Digital controllers! What is digital control?
  - Digital control is concerned with the area of control systems which are implemented using digital techniques/ computers
- What is needed?
  - Hardware – to transform from the real (“analogue”) world to the computer (“digital”) world and vice versa
  - Hardware – to do the thinking in the digital world. This is the processing hardware.
  - Software – The instructions to be carried out. These are the algorithms run on the computing hardware.
- In digital control systems we need to consider sampled and quantised signals as well as having to use z transforms to make the mathematics easier, etc.

# Numerical example

- Determine the output response of the second order system shown below when a unit step is applied



U(s)

$$\frac{8}{\left(s^2+6s+8\right)}$$

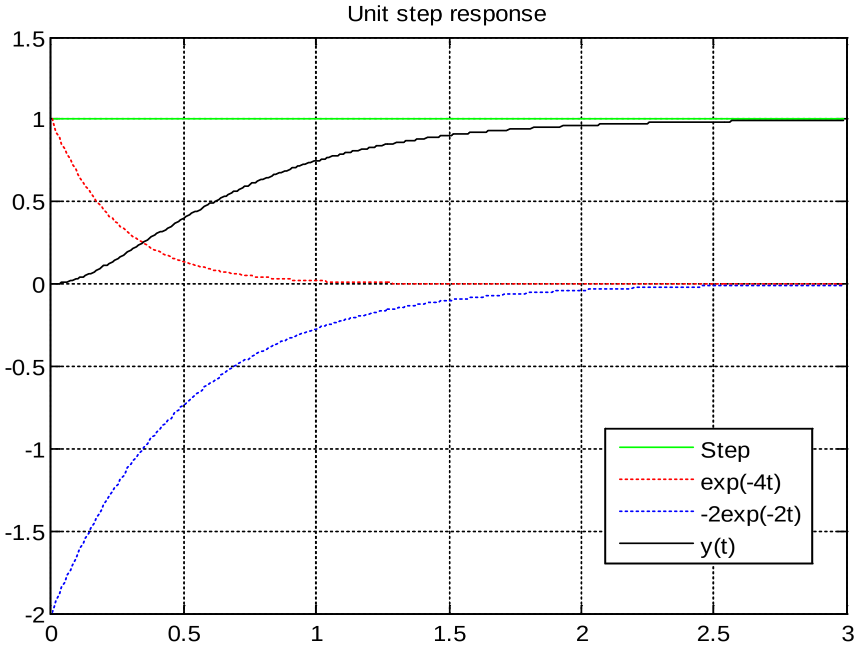
Y(s)

$$Y(s)=\frac{8}{s^2+6s+8}U(s)=\frac{8}{(s+4)(s+2)}\frac{1}{s}$$

$$=\frac{1}{s}+\frac{1}{s+4}-\frac{2}{s+2}$$

**Two 1st order subsystems ⇨ via “Cover up” rule**

$$\Rightarrow y(t)=1+e^{-4t}-2e^{-2t}$$



# Standard second order systems

- Most systems can be studied as a set of first order sub-systems or a standard second order system (dominant behaviour)
- Transfer function of a standard second order system is

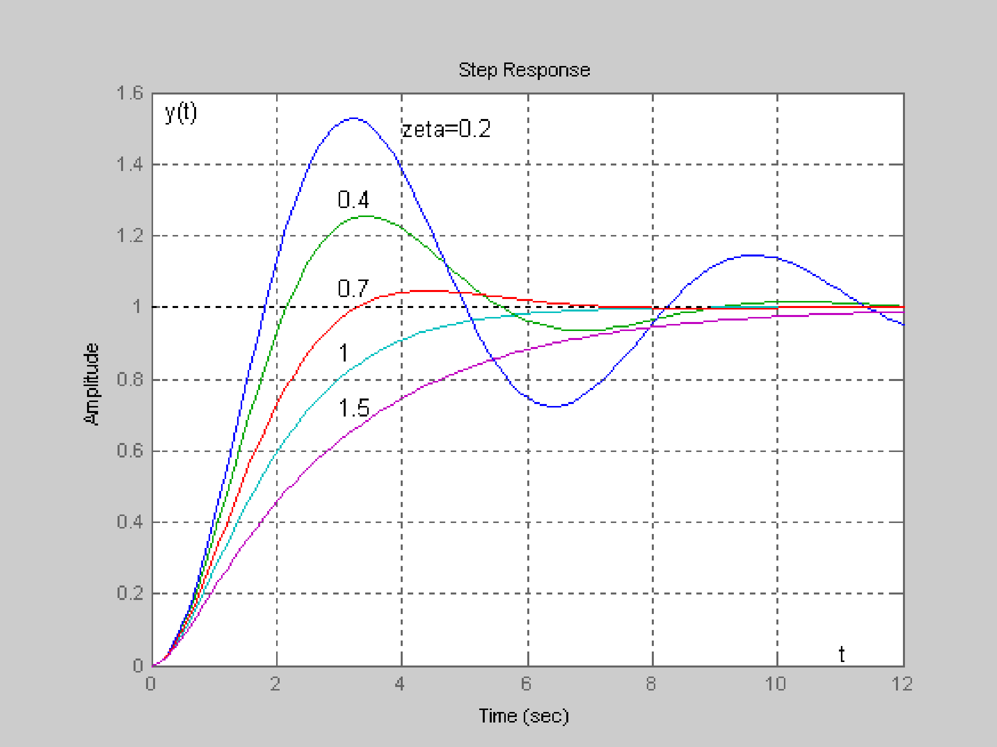
$$\frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (Transfer\ \ function)$$

$$y(t) = 1 - e^{-\zeta\omega_n t}\cos(\omega_d t + \phi) \quad (Step\ response)$$

$$\omega_n\ is\ undamped\ natural\ \ frequency$$

$$\zeta\ \ is\ damping\ ratio$$

# Time domain analysis

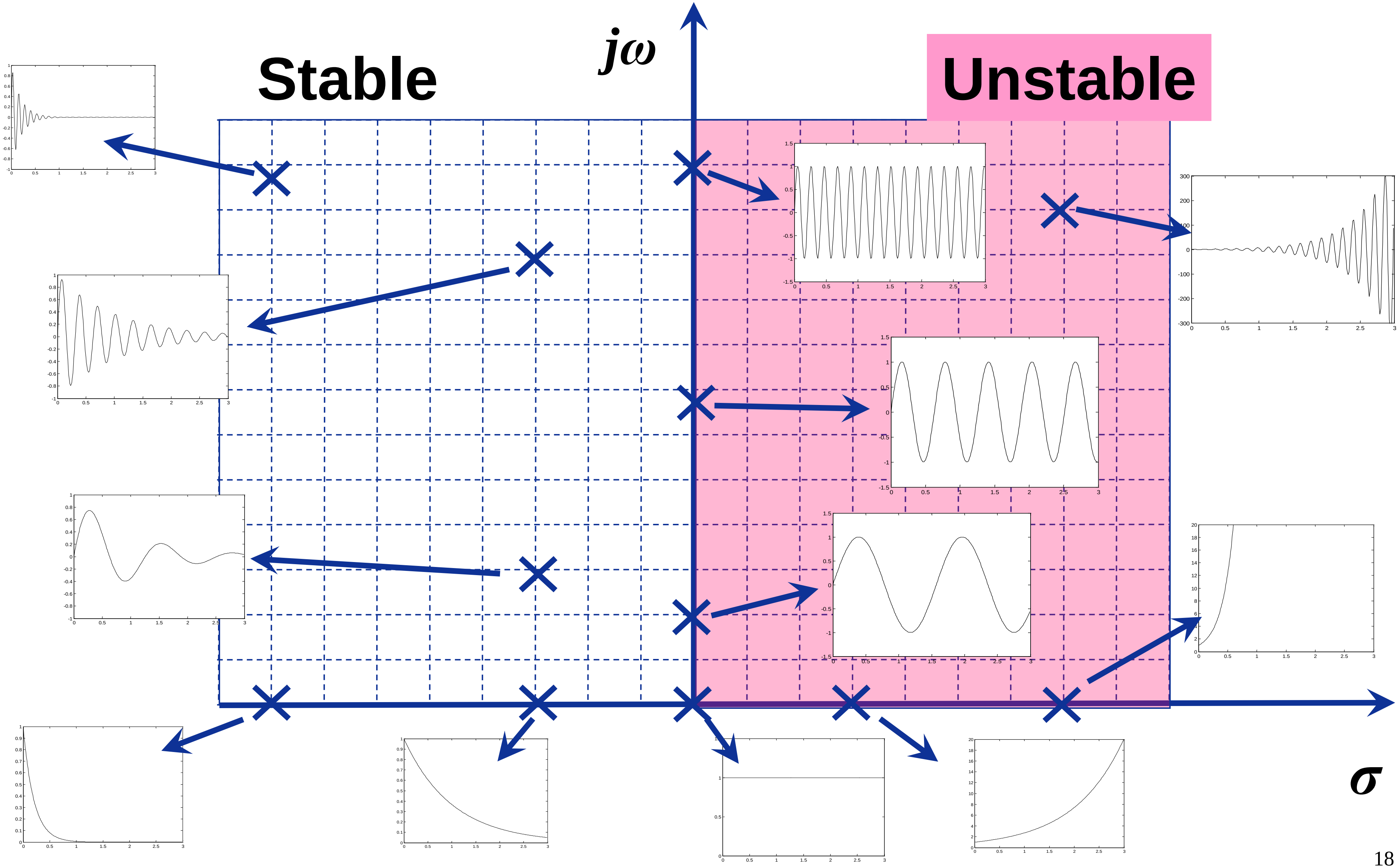


$$\frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (Transfer\ \ function)$$

$$y(t) = 1 - e^{-\zeta\omega_n t}\cos(\omega_d t + \phi) \quad (Step\ response)$$

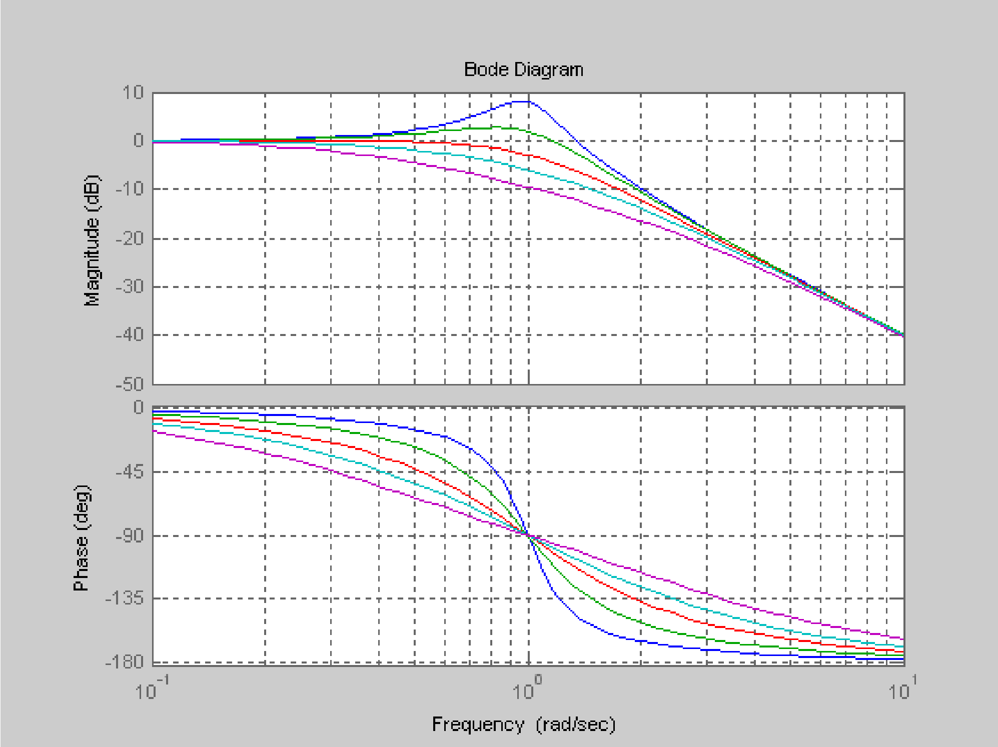
- Study system responses to standard inputs, eg. step, impulse, ramp, etc.
- Behaviour is characterised by overshoot, damping ratio, settling time, rise time, etc
- Responses based on 1st order system response or,
- Based on 2nd order system responses, dominant roots, etc

# Stability in the s-Plane



InnoTecUK™

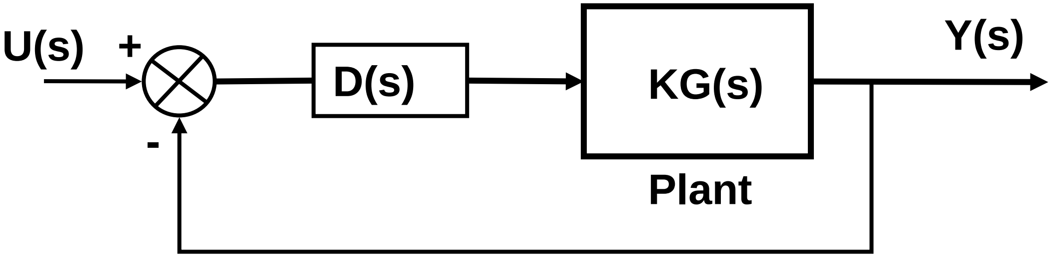
# Frequency domain analysis



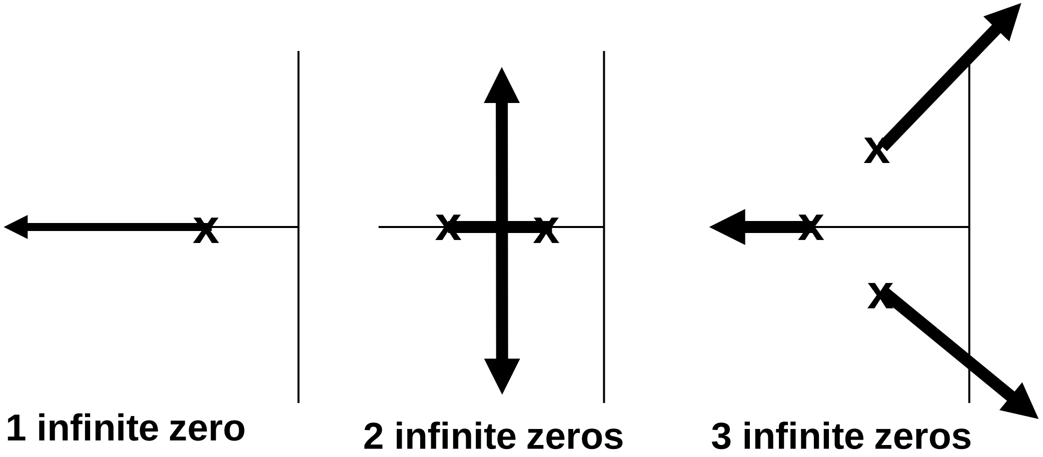
$$\textit{Bode plot of } \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \quad \textit{for } \zeta = 0.2-1.5$$

- **Study system's steady state responses to sinusoidal inputs**
- **Behaviour is characterised by gain margin, phase margin, bandwidth, type number, etc**
- **Use Laplace transfer function and replace s by jω.**
- **Bode, Nyquist, Polar**

# Root locus analysis



$$CE: \quad 1+KD(s)G(s)=0$$

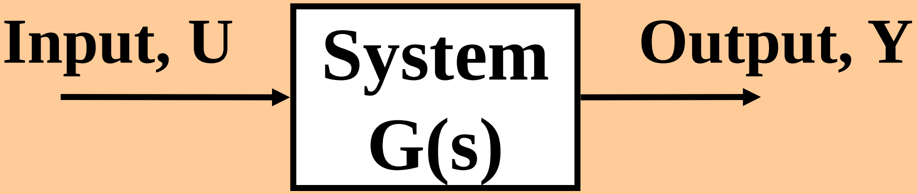


- Study variations of pole locations of characteristic equation as K: 0 → ∞
- Simple rules for drawing root locus in s-plane
  - Locus starts at open loop (OL) poles
  - End at OL zeros (finite and infinite)
  - Standard patterns for no of infinite zeros
- Same rules for drawing root locus in z-plane but pole locations have different meaning

# Transfer functions OL & CL

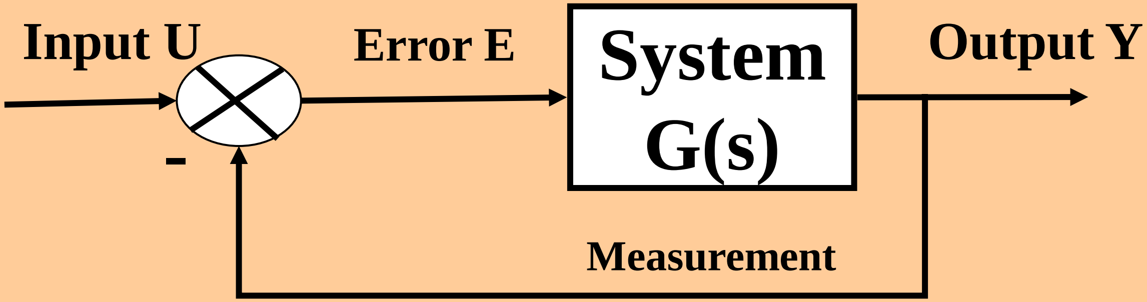
## Open Loop Transfer function

$$Y(s) = G(s)U(s)$$



## Closed Loop Transfer function: Unity feedback

$$Y(s) = G(s)E(s) \ and$$
$$E(s) = U(s) - Y(s)$$
$$\Rightarrow Y(s) = G(s)\{U(s) - Y(s)\}$$
$$\Rightarrow (1 + G(s))Y(s) = G(s)U(s)$$
$$\Rightarrow Y(s) = \frac{G(s)U(s)}{1 + G(s)}$$



# Controller design general aims

- Stability
- Low steady state errors
- Good transient response in time domain
  - Fast behaviour
  - Fast settling time
  - Low overshoot – damping ratio, $\zeta \sim 0.5\text{-}0.7$
- Design in frequency domain
  - Phase margin~50º
  - Gain margin~9dB
- Noise ignored

# Controller design methods

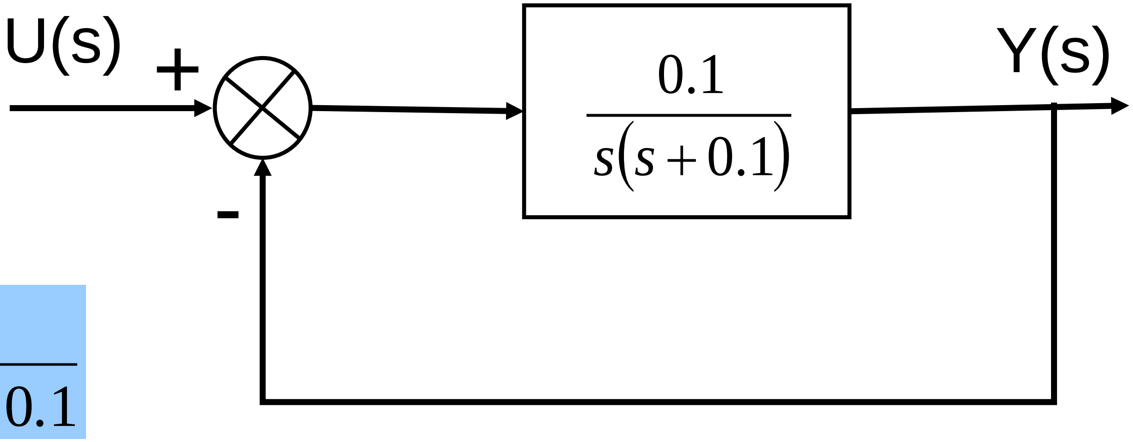
- Root locus – pole and zeros in s-plane adjusted for desired steady state error and transient response
- Frequency domain – Bode, Nyquist – phase margin related to overshoot, bandwidth to damping ratio and settling or peak time
- Pole placement via State feedback
- PID (P,I,D, PI,PD) controllers
- Lead, lag and lead-lag controllers

# System considered

- **Open-loop Continuous TF:**

$$G_{OL}(s) = \frac{1}{s(10s+1)} = \frac{0.1}{s(s+0.1)} = \frac{0.1}{s^2+0.1s}$$

- **Closed-loop Continuous TF**



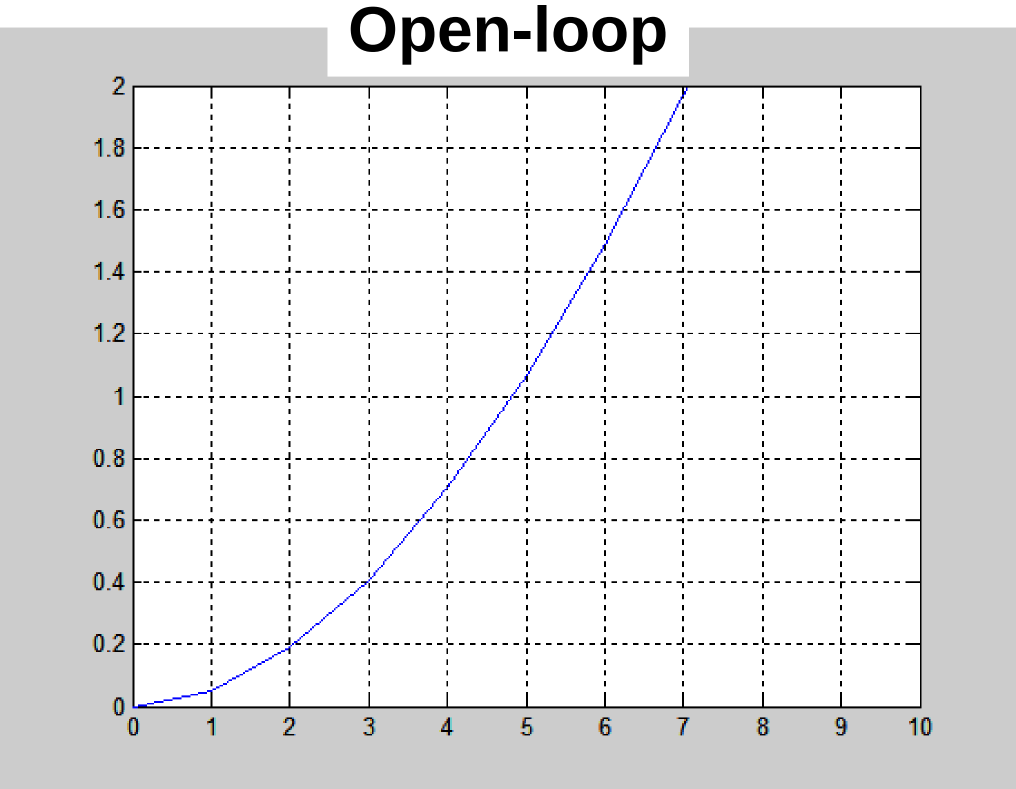
$$G_{CL}(s) = \frac{0.1}{s^2+0.1s+0.1}$$

$$\omega_n^2 = 0.1 \Rightarrow \quad \omega_n = 0.32 rad/\sec = 0.0509 Hz \quad (slow)$$

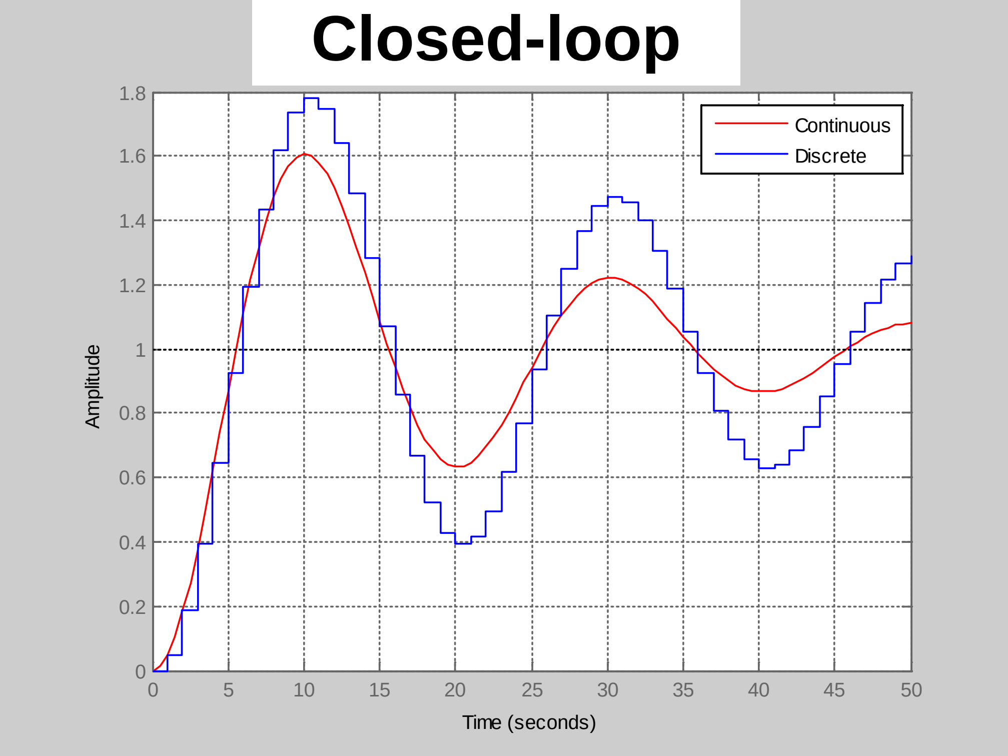
$$2\omega_n \zeta = 0.1 \Rightarrow \quad \zeta = 0.16 \quad (low)$$

# Uncompensated system response

Open-loop



Closed-loop



- System speed is low (settling time >60sec)
- Damping is low (large overshoot ≈60%)
- Will illustrate various controller design approaches using continuous and discrete control systems

# 1. Simple design

**Cancel slow pole in OLTF and add a faster pole. Can use file GSVDesign1.m to help with design.**

$$G_{OL}(s)D(s)=\frac{0.1}{s(s+0.1)}\frac{10(s+0.1)}{(s+1)}=\frac{1}{s(s+1)}$$

$$CLTF=\frac{1}{s^2+s+1}$$

**Continuous Controller**

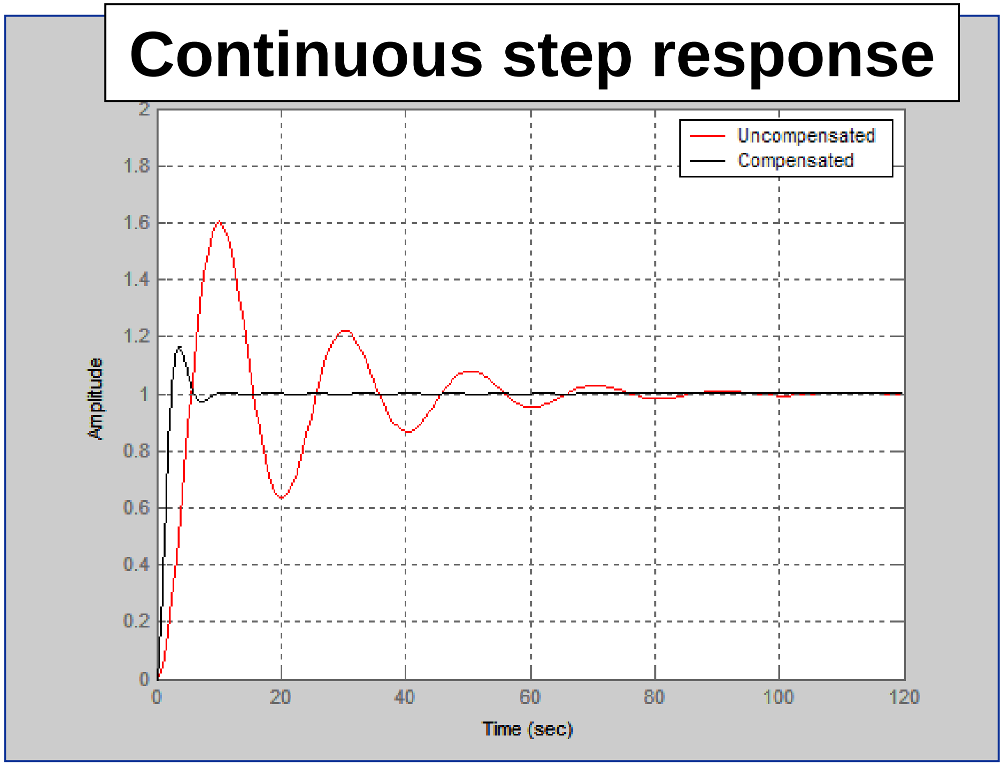
$$D(s)=\frac{10(s+0.1)}{(s+1)}$$

$$\omega_n^2=1\Rightarrow\quad\omega_n=1rad/\sec=0.159Hz\quad(3\times\ faster)$$

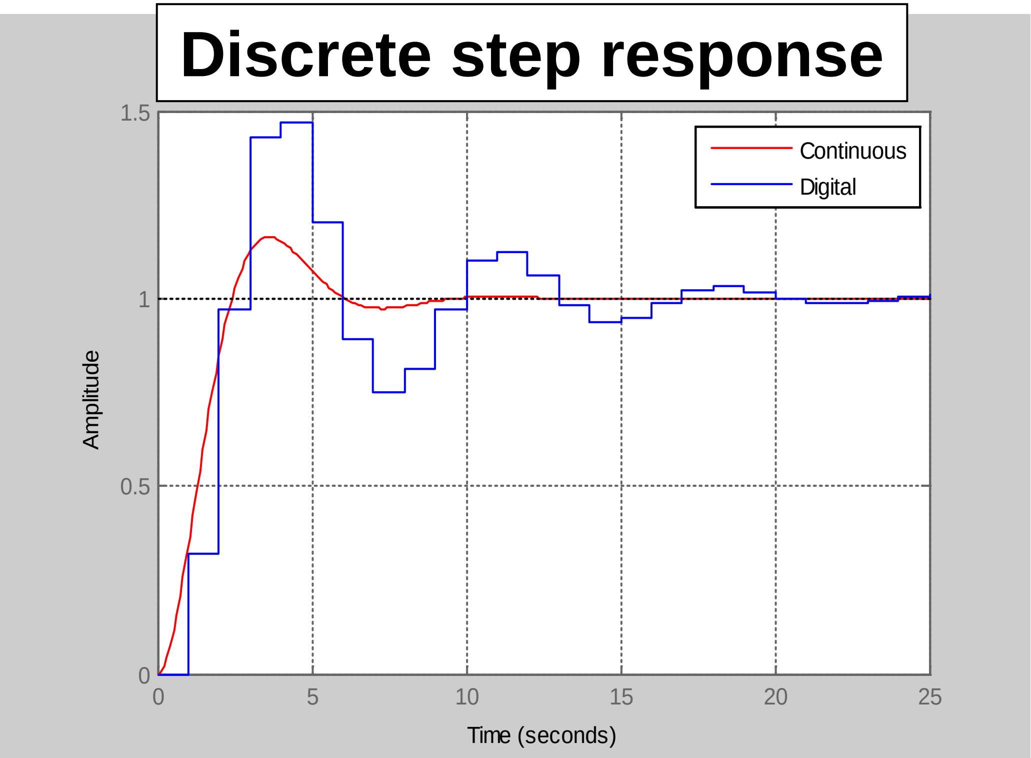
$$2\omega_n\zeta=1\Rightarrow\quad\zeta=0.5\quad(Increased)$$

# 1. Simple design's performance

Continuous step response



Discrete step response



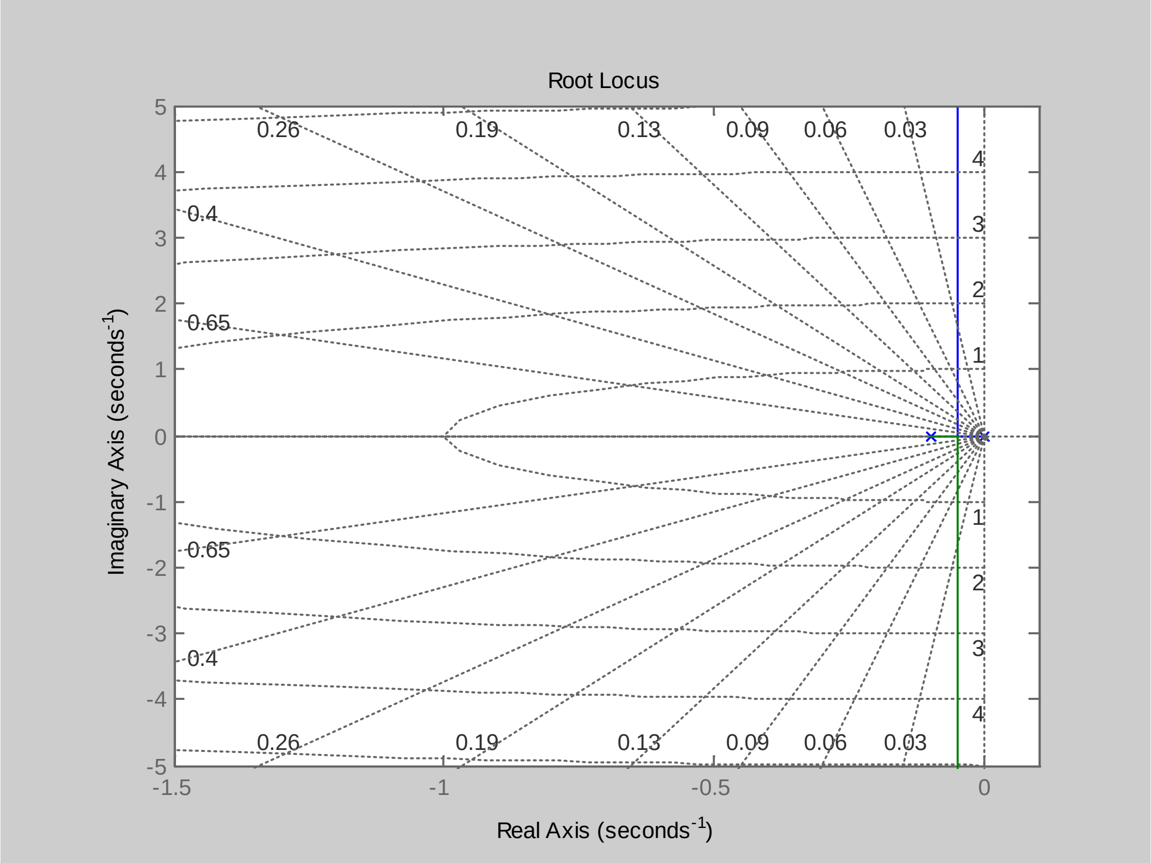
- Discretisation (with T=1sec) introduces de-stabilising effect that needs to be allowed for in continuous design
- Need for appropriate safety margin in the continuous design

# 2. Design via Root Locus

- **Design continuous controller *C(s)* via Root Locus**

$$G_{OL}(s) = \frac{1}{s(10s+1)}$$

- **Original root locus**
  - RL starts at OL poles
  - RL ends at OL zeros
  - System has
    - 2 slow poles
    - 2 infinite zeros



**System's Root locus is limited to lie near imaginary axis**

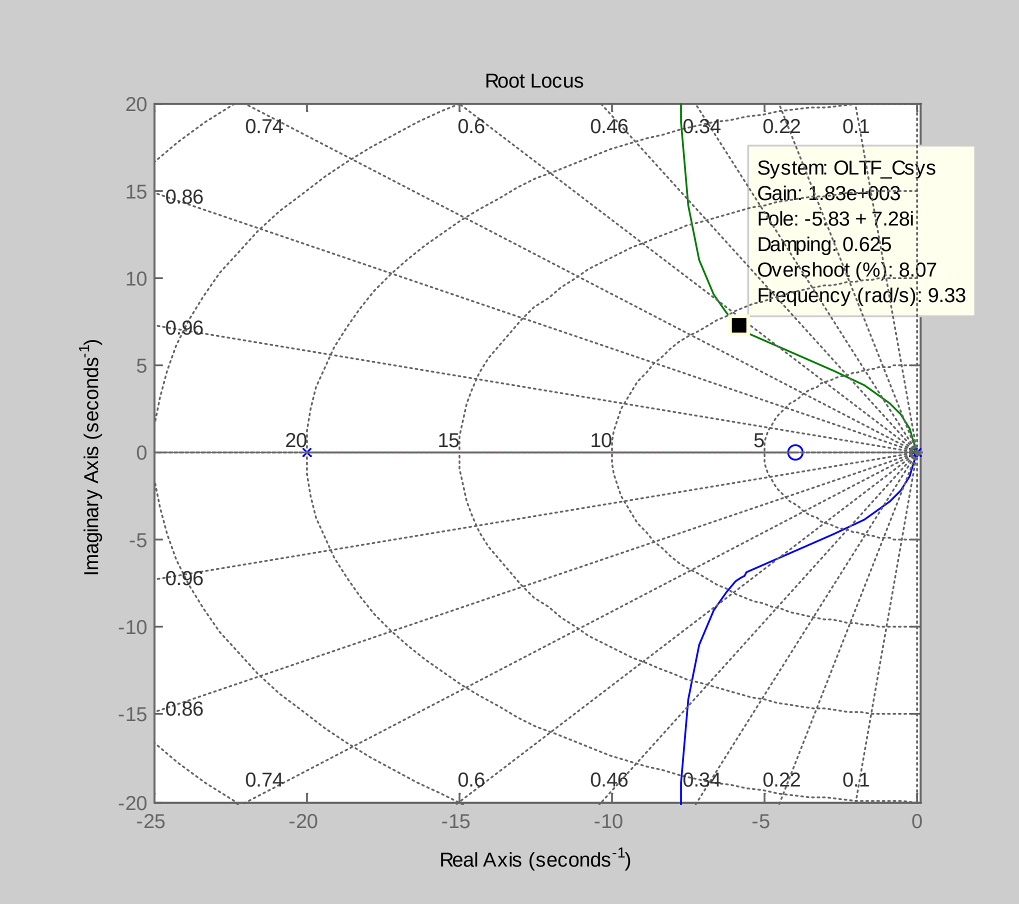
**Add a faster zero to "attract" the RL to the left to make it faster, plus a faster pole "out of the way" to make the controller proper**

# 2. Design via RL (2)

- Use Matlab file GSVDesign2.m to assist design
- Try zero at -4 and a pole at -20

$$C(s)=\frac{s+4}{s+20}$$

- Gives Root Locus
- Find gain at “best location” on RL =1830

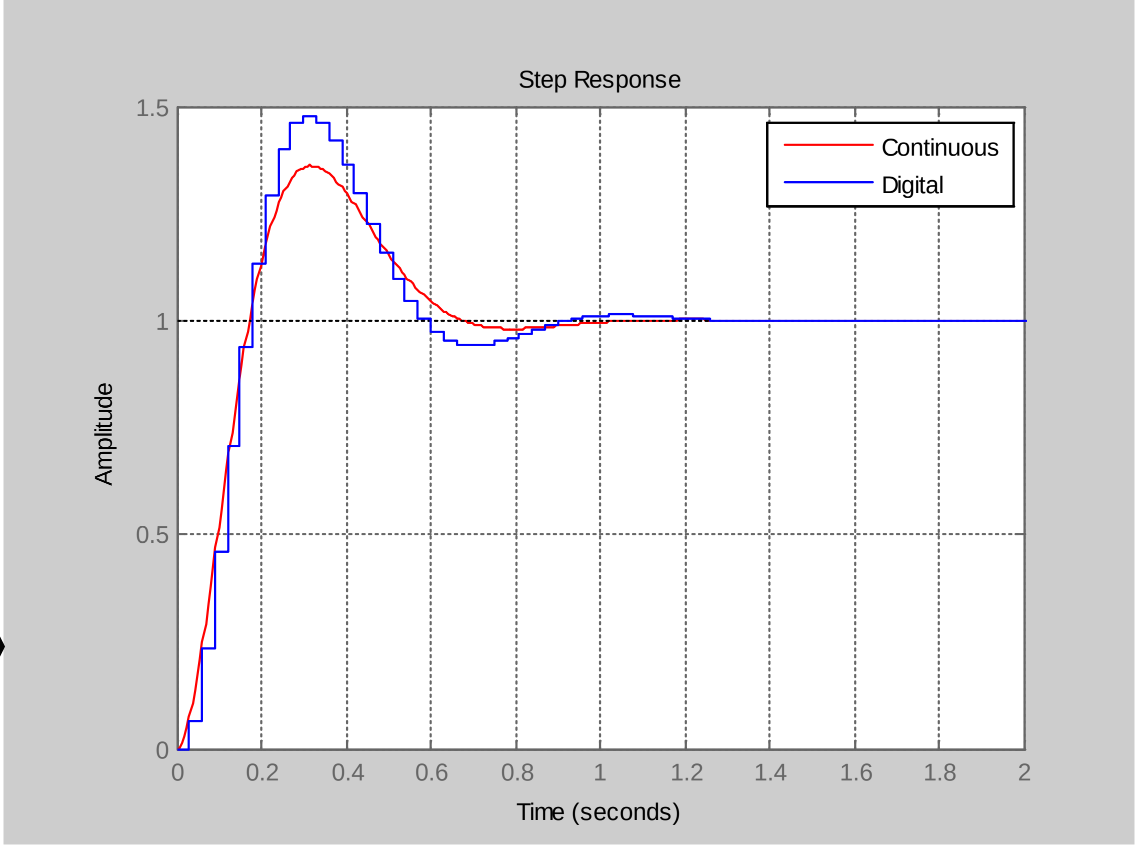


# 2. Design via RL (3)

- Enter gain=1830 into DesignGSV2.m file & RUN
- Note: System has been speeded up
- T needs reducing
  - T=0.05sec
- Controller is

$$C(s)=\frac{1830(s+4)}{s+20}$$

**Time response of designed controller**

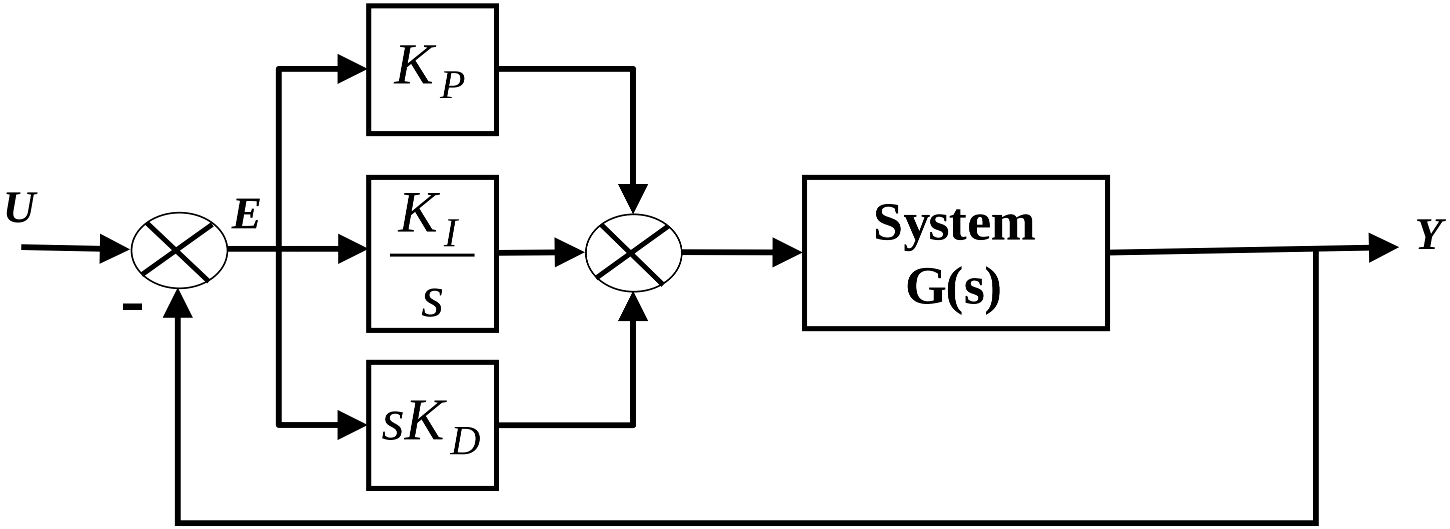


# 3. PID controller design

- PID controllers are the most widely used control strategy in industry. It consists of three different elements:
  - Proportional control: pure gain adjustment on the error signal to provide the driving input to the system, it is used to adjust the speed of the system; improves transient response, reduces steady-state error, may reduce stability
  - Integral control: implemented by including an integrator to provide the required accuracy for the control system; increases system type by 1, infinity steady-state gain which eliminates steady-state error for a unit step input. Need to avoid integral windup by switching integrator off when control has saturated
  - Derivative control: here derivative action is used to increase the damping in the system. The derivative term also amplifies the existing noise which can cause problems including instability
- Can be used as P, PI, PD and PID forms

# 3. PID controller design (2)

- **PID block diagram:**



$$PID = K_P + \frac{K_I}{s} + sK_D$$

$$PID = K_P\left\{1 + \frac{1}{T_I s} + T_D s\right\}$$

$$= \frac{K_P\left(T_I T_D s^2 + T_I s + 1\right)}{T_I s}$$

**where**
**$K_P$ is the proportional gain**
**$T_I$ is the integral time constant**
**$T_D$ is the derivative time constant**

**$K_P$, $T_I$, $T_D$ need to be tuned for tuning the PID controller**

InnoTecUK™

# 3. PID controller design (3)

- **Ziegler Nichols closed-loop tuning method:**
  - **Select proportional control only**
  - **Increase the value of $K_P$ until point of instability is reached (sustained oscillations), the critical value of gain $K_C$, is reached**
  - **Measure the period of oscillation to obtain the critical time constant TC**
- **Once $K_C$ and $T_C$ obtained, the PID parameters can be calculated from the following table:**

| Control | $K_P$ | $T_I$ | $T_D$ |
|---|---|---|---|
| P only | $0.5K_C$ | | |
| PI | $0.45K_C$ | $0.833T_C$ | |
| PID tight control | $0.6K_C$ | $0.5T_C$ | $0.125T_C$ |
| PID some overshoot | $0.33K_C$ | $0.5T_C$ | $0.33T_C$ |
| PID no overshoot | $0.2K_C$ | $0.3T_C$ | $0.5T_C$ |

**These values are not optimal and some additional tuning may be needed.**

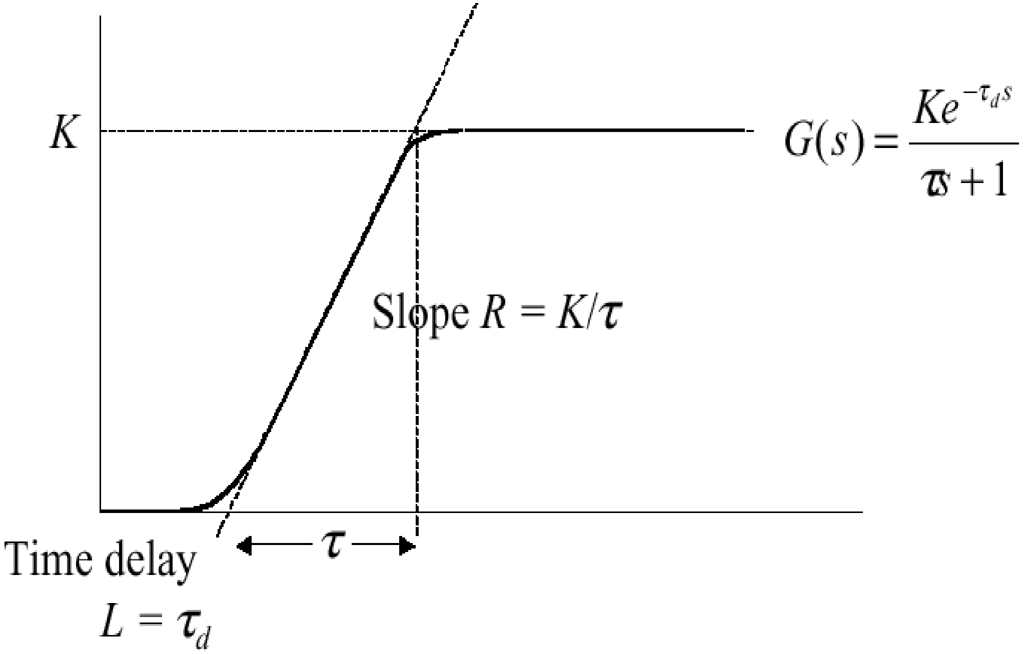
# 3. PID controller design (4)

**Closed-loop method cannot be applied in some cases. Can also use Ziegler-Nichols reaction curve method:**

- **With system in open loop, take it manually to the normal operating point and let the output settle at $y(t)=y_0$ for a constant input $u(t)=u_0$**
- **At initial time $t_0$, apply a step change to the input from $u_0$ to $u_1$ (≈ 10-20% of full scale)**
- **Record the system output until it settles to the new operating point. Assume you obtain the curve shown on the next slide. This curve is the process reaction curve**

# 3. PID controller design (5)

## Process reaction curve



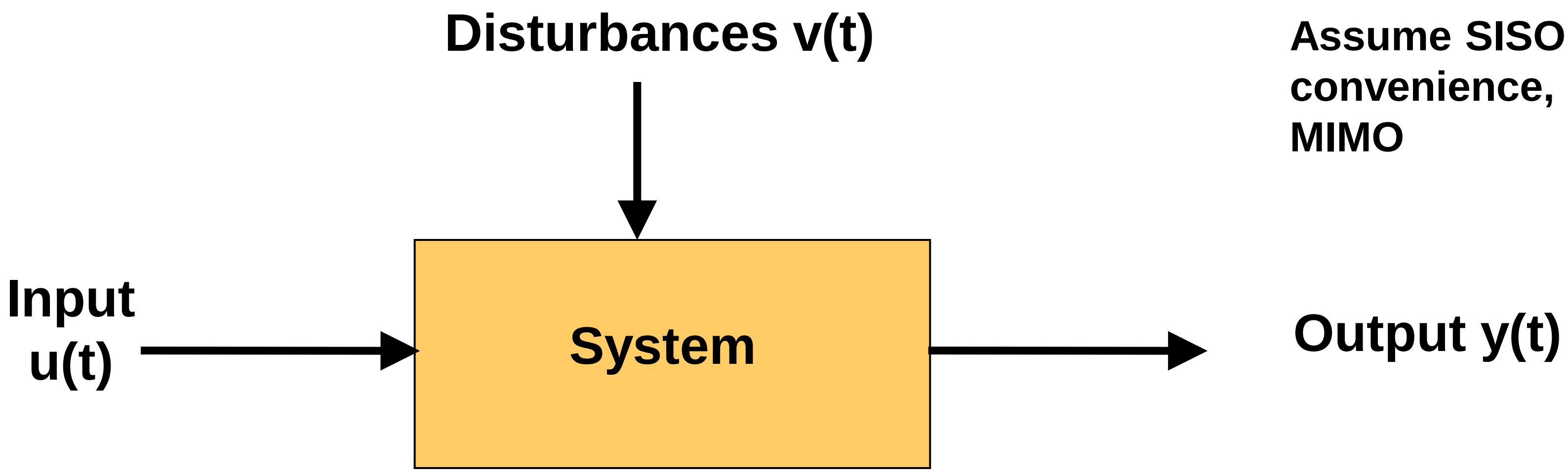
**Use values from reaction curve to tune PID controller parameters**

| Control | $K_P$ | $T_I$ | $T_D$ |
|---|---|---|---|
| P only | *1/RL* | | |
| PI | *0.9/RL* | *0.27/RL*$^2$ | |
| PID | *1.2/RL* | *0.6/RL*$^2$ | *0.6/R* |

**Other tuning methods also available**

# System identification

## Typical setup



Assume SISO for convenience, similarly for MIMO

System is driven by input u(t) and disturbances v(t)

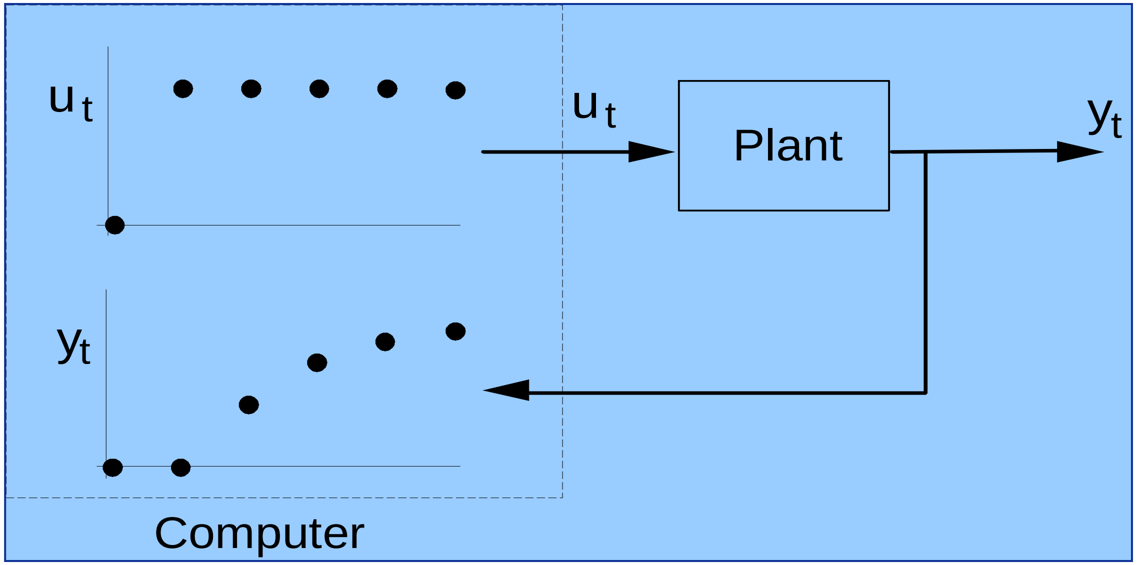
Can control

Can’t control (?)
Can’t measure (?)

u(t), v(t) → effects y(s) for s>t; causal system

# Identification procedure

1. Design experiment
2. Perform design experiment to collect data
3. Choose/ determine model structure
4. Determine model parameters (batch form and on-line)
5. Validate model
6. Good?
   - Yes – Stop
   - No – Repeat



**Many types of models exist; we will focus on discrete time invariant linear systems**

# Direct calculation: Example 1

- Step response data

| t | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $u_t$ | 0 | 1 | 1 | 1 | 1 | 1 |
| $y_t$ | 0 | 0 | 0.6 | 1.02 | 1.22 | 1.35 |

- The plant model is assumed to be of the form

$$y_t = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} u_t$$

- This gives a difference equation

$$y_t = -a_1 y_{t-1} + b_1 u_{t-1}$$

# Direct calculation: Example 1 (Cont)

| t | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $u_t$ | 0 | 1 | 1 | 1 | 1 | 1 |
| $y_t$ | 0 | 0 | 0.6 | 1.02 | 1.22 | 1.35 |

- At t=1, apply unit step; $y_1 = 0$
- At t=2, $y_2 = -a_1 y_1 + b_1 u_1$ (1)
- At t=3, $y_3 = -a_1 y_2 + b_1 u_2$ (2)
- From (1): $0.6 = -a_1 \times 0 + b_1 \times 1 \quad \Rightarrow \quad b_1 = 0.6$
- From (2):

$$1.02 = -a_1 \times 0.6 + b_1 \times 1$$

$$1.02 = -a_1 \times 0.6 + 0.6 \quad \Rightarrow \quad a_1 = \frac{-0.42}{0.6} = -0.7$$

- Plant model is:

$$\frac{0.6z^{-1}}{1 - 0.7z^{-1}}$$

# Direct calculation: General

- **Substituting input (u) and output values for n+m different values of t gives n+m equations. Hence can solve for the n+m unknowns**

$$\begin{aligned}
y_i &= -a_1 y_{i-1} - a_2 y_{i-2} - \cdots - a_n y_{i-n} + b_1 u_{i-1} + b_2 u_{i-2} + \cdots + b_m u_{i-m} \\
y_{i+1} &= -a_1 y_i - a_2 y_{i-1} - \cdots - a_n y_{i+1-n} + b_1 u_i + b_2 u_{i-1} + \cdots + b_m u_{i+1-m} \\
y_{i+2} &= -a_1 y_{i+1} - a_2 y_i - \cdots - a_n y_{i+2-n} + b_1 u_{i+1} + b_2 u_i + \cdots + b_m u_{i+2-m} \\
&\vdots \\
y_{i+d-1} &= -a_1 y_{i+d-2} - a_2 y_{i+d-3} - \cdots - a_n y_{i+d-1-n} + b_1 u_{i+d-2} + b_2 u_{i+d-3} + \cdots + b_m u_{i+d-1-m}
\end{aligned}$$

- **where d=n+m**

# Direct calculation: General (Cont 1)

## Write in matrix form

$$\underline{y} = \underline{\Psi}\,\underline{\theta}$$

## where

$$\underline{y} = \begin{bmatrix} y_{i} \\ y_{i+1} \\ y_{i+2} \\ \vdots \\ y_{i+d-1} \end{bmatrix}$$

$$\underline{\Psi} = \begin{bmatrix} y_{i-1} & y_{i-2} & \cdots & y_{i-n} & u_{i-1} & u_{i-1} & \cdots & u_{i-m} \\ y_{i} & y_{i-1} & \cdots & y_{i+1-n} & u_{i} & u_{i-1} & \cdots & u_{i+1-m} \\ y_{i+1} & y_{i} & \cdots & y_{i+2-n} & u_{i+1} & u_{i} & \cdots & u_{i+2-m} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{i+d-2} & y_{i+d-3} & \cdots & y_{i+d-1-n} & u_{i+d-2} & u_{i+d-3} & \cdots & u_{i+d-1-m} \end{bmatrix}$$

$$\underline{\theta} = \begin{bmatrix} -a_{1} \\ -a_{2} \\ \vdots \\ -a_{n} \\ b_{1} \\ b_{2} \\ \vdots \\ b_{m} \end{bmatrix}$$

## Hence we have:

$$\underline{\theta} = \underline{\Psi^{-1}}\,\underline{y}$$

➡ **Parameters**

# Direct calculation: Drawbacks

- Data is normally corrupted by noise (sensor, system, measurement, etc). Hence plant parameters are not accurate
- Method relies on input being "persistently exciting" to ensure data contains the full dynamics of the system. When plant input is constant (eg step input), data is not usually good enough for system identification
- Need for much greater set of data so that
  - Effects of noise can be averaged (properties of noise is also important)
  - There is enough system dynamics information to capture all that is needed for the required accuracy
  - Parameter estimation via Least squares

# Least squares estimation

- Consider

$$\left(1+a_1z^{-1}+a_2z^{-2}+\cdots+a_{n_a}z^{-n_a}\right)y_t=\left(b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_{n_b}z^{-n_b}\right)u_t+e_t$$

$$\Rightarrow y_t=-(a_1y_{t-1}+a_2y_{t-2}+....+a_{n_a}y_{t-n_a})+b_0u_t+b_1u_{t-1}+b_2u_{t-2}+...+b_{n_b}u_{t-n_b}+e_t$$

- Assume input is zero for t<0 and start up values are known for the output

$$\underset{\boldsymbol{y}}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}=\underset{\boldsymbol{A}}{\begin{bmatrix} y_0 & y_{-1} & y_{-2} & \cdots & y_{-n_a+1} & u_1 & u_0 & u_{-1} & \cdots & u_{-n_b+1} \\ y_1 & y_0 & y_{-1} & \cdots & y_{-n_a+2} & u_2 & u_1 & u_0 & \cdots & u_{-n_b+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{N-1} & y_{N-2} & y_{N-3} & \cdots & y_{N-n_a} & u_N & u_{N-1} & u_{N-2} & \cdots & u_{N-n_b} \end{bmatrix}}\overset{\boldsymbol{\theta}}{\begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_{n_a} \\ b_0 \\ \vdots \\ b_{n_b} \end{bmatrix}}+\underset{\boldsymbol{e}}{\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}}$$

# Least squares (Cont)

- Hence we have system is governed by

$$y = A\theta + e \quad and \quad if \quad N > n_a + n_b + 1 \quad problem \quad is \quad solvable$$

- Assume we have a model $y = A\hat{\theta} + \hat{e}$ where $\hat{\theta}$ is the vector of system parameters and $\hat{e}$ is the modelling error
- What should $\hat{\theta}$ be to minimise $\hat{e}$ ?

$$\Rightarrow \hat{e} = y - A\hat{\theta}$$

# Least Squares (Cont 2)

- The least squares estimate chooses the set of parameters which minimises a cost function

$$J = \sum_{1}^{N} \hat{e}^2 = \hat{e}_N^T \hat{e}_N = \left( y - A\hat{\theta} \right)^T \left( y - A\hat{\theta} \right)$$

$$= y^T y - y^T A\hat{\theta} - \hat{\theta}^T A^T y + \hat{\theta}^T A^T A\hat{\theta}$$

- Hence for a minimum *J* wrt $\hat{\theta}$

$$\frac{\partial J}{\partial \hat{\theta}} = 0 = -y^T A - A^T y + 2A^T A\hat{\theta}$$

$$\Rightarrow 2A^T y = 2A^T A\hat{\theta} \quad \Rightarrow \hat{\theta} = \left[A^T A\right]^{-1} A^T y$$

$$\frac{\partial^2 J}{\partial \hat{\theta}^2} = 2A^T A$$

➡

$+ve\ definite$

$\Rightarrow \min$

**Excitation condition**

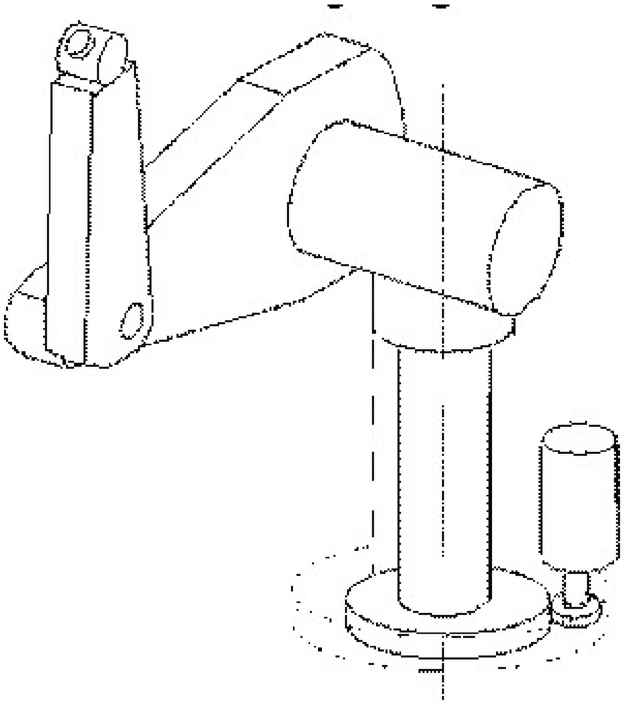
$\boldsymbol{A^T A \neq 0}$

- Basic equation upon which (BATCH form of) least squares algorithms are based

$$\hat{\theta} = \left[A^T A\right]^{-1} A^T y$$

# Motion control of a single joint

- In many industrial manipulators, each joint is driven using a separate, independent control system. For example, consider joint 1 of a Puma 560 manipulator, which is driven by a DC servomotor through a gear train



Puma 560 Joint 1 drive system

# Motion control: Analysis of drive system

**The joint torque $\tau$ will accelerate the joint and the manipulator above it; these have inertia *J* (second order system):** $\tau = J\ddot{\theta}$ **(1)**

**Torque delivered by the motor $\tau_m$ is used to provide the joint torque, to overcome any unmeasured disturbance torque $\tau_d$** (which could include friction) and to accelerate the motor itself:

$$\tau_m = \tfrac{\tau}{G} + \tau_d + J_m\left(G\ddot{\theta}\right) \qquad \textbf{(2)}$$

where *G* is the gear ratio (motor speed over joint speed) substituting equation (1) into equation (2):

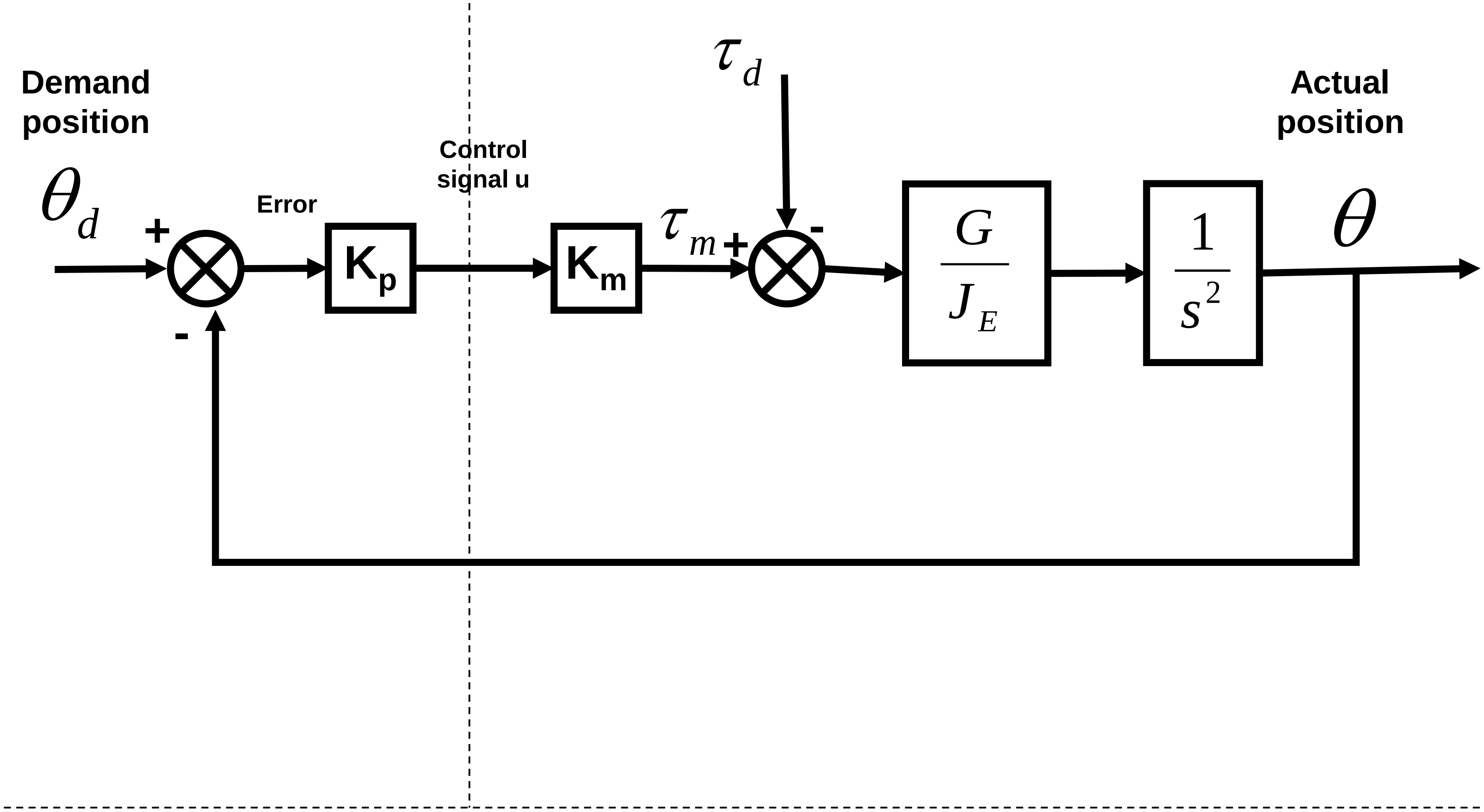
$$\tau_m = \tfrac{J}{G}\ddot{\theta} + \tau_d + J_m G\ddot{\theta} = \frac{J + J_m G^2}{G}\ddot{\theta} + \tau_d$$

$$\tau_m = \tfrac{J_E}{G}\ddot{\theta} + \tau_d \quad or \quad \ddot{\theta} = \tfrac{G}{J_E}\left(\tau_m - \tau_d\right) \qquad \textbf{(3)}$$

Motor torque is proportional to current. If a current amplifier is used, then the current is in turn proportional to the current signal *u*. Thus if $K_m$ is a constant: **(4)**

$$\tau_m = K_m u$$

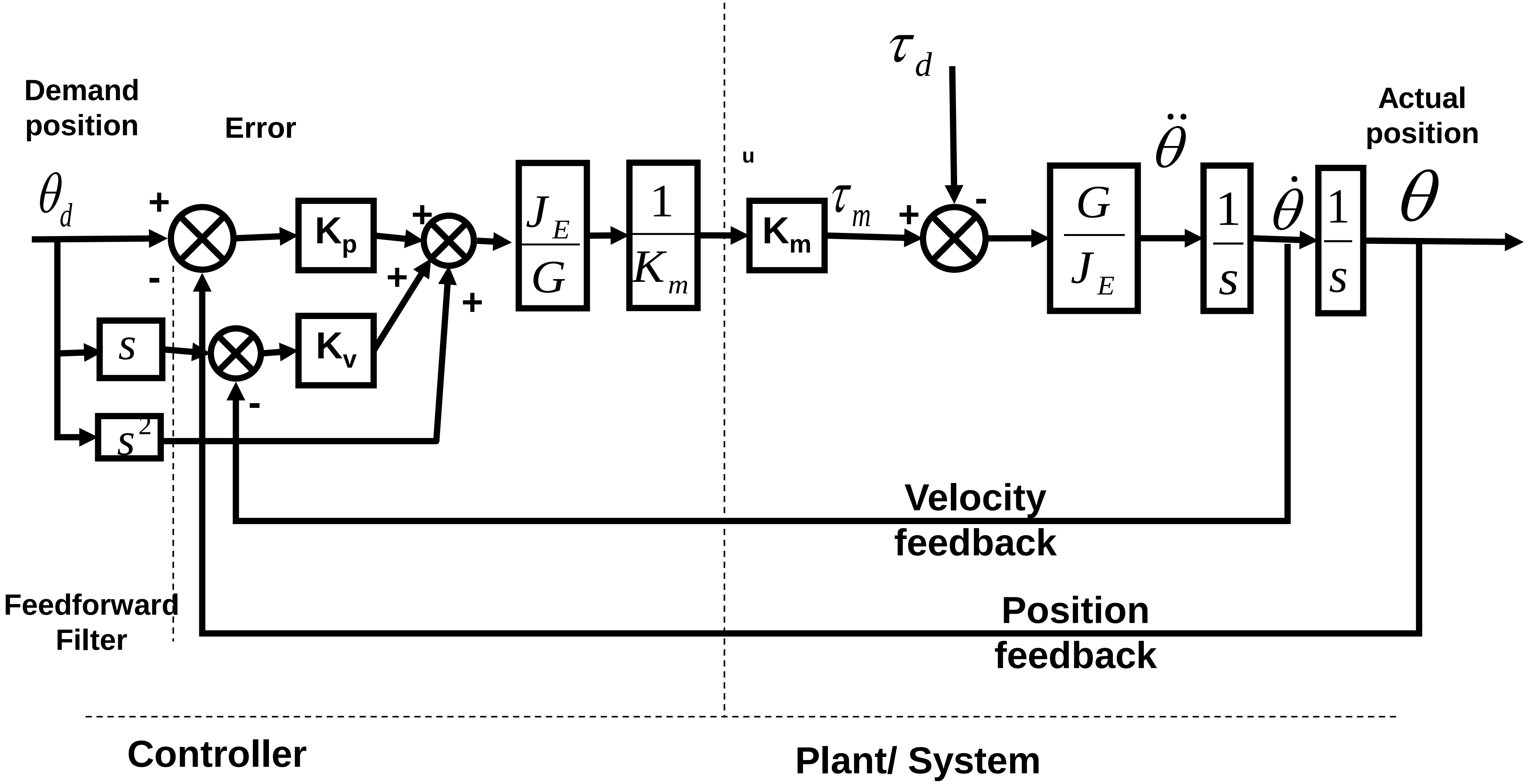
# Proportional control of a single joint



Controller

Plant/ System

# PD control of a single joint



InnoTecUK™

# Summary

- Overview of modelling and control in general systems engineering applications
- Introduction to controller design via several methods
- Examples described

Thanks are expressed to all the colleagues (too many to mention individually) across the world who have helped with the formulation of this lecture on Modelling and Control from material placed on the www