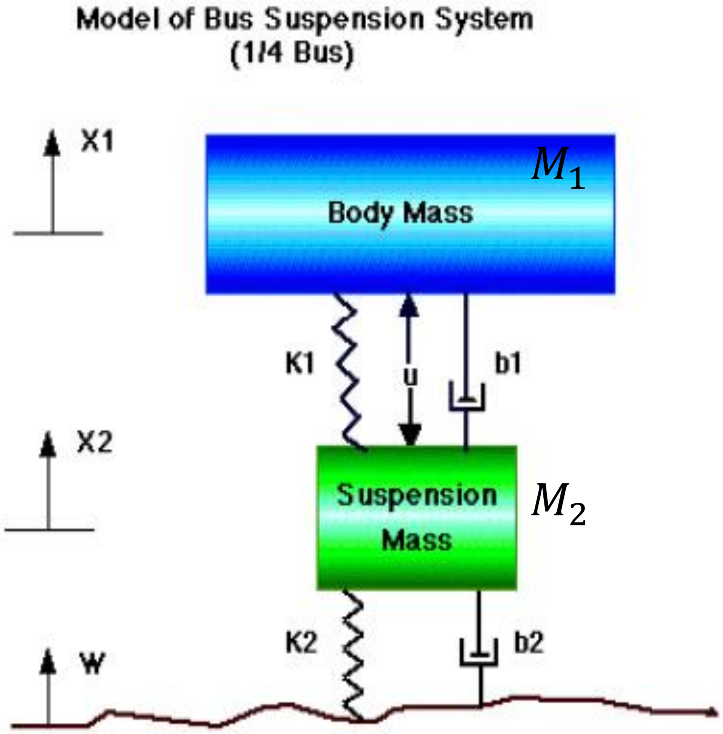
Classical and State Space Control

Prof. Bishakh Bhattacharya

Assignment: Design of a Bus Suspension System



In the range of

Design a Controller such that the over shoot is less
5% and the settling time is less than 2 seconds

Governing equations and Parameters

$$M_1 \ddot{x}_1 = -b_1(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2) + U$$

$$M_2 \ddot{x}_2 = b_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2) + b_2(\dot{w} - \dot{x}_2) + k_2(w - x_2) - U$$

* body mass (m1) = 2500 kg,

* suspension mass (m2) = 320 kg,

* spring constant of suspension system(k1) = 80,000 N/m,

* spring constant of wheel and tire(k2) = 500,000 N/m,

* damping constant of suspension system(b1) = 350 Ns/m.

* damping constant of wheel and tire(b2) = 15,020 Ns/m.

* control force (u) = force from the controller we are going to design.

Time Delay

  - *Time delay always reduces the stability of a system !*
  - *Important to be able to analyze its effect*
  - *In the s-domain a time delay is given by e^{-ls}*
  - *Most applications contain delays (sampled systems)*
- Root locus analysis
  - *The original method does only handle polynomials*
- Solutions
  - *Approximation (Padé) of e^{-ls}*
  - *Modifying the root locus method (direct application)*

More on Time Delay

First approximation

(1,1) Padé approximant

$$e^{-s} \approx \frac{b_0 s + b_1}{a_0 s + 1}$$

McLauren series

$$e^{-s} = 1 - s + \frac{s^2}{2} - \frac{s^3}{3!} + \frac{s^4}{4!} - \ldots$$

$$\frac{b_0 s + b_1}{a_0 s + 1} = b_1 + (b_0 - a_0 b_1)s$$

$$- a_0(b_0 - a_0 b_1)s^2 + a_0^2(b_0 - a_0 b_1)s^3$$

$$b_1 = 1$$

$$b_0 - a_0 b_1 = -1$$

$$- a_0(b_0 - a_0 b_1) = \frac{1}{2}$$

$$a_0^2(b_0 - a_0 b_1) = -\frac{1}{6}$$

$\Downarrow$ with $s \equiv T_d s$

$$e^{-T_d s} \cong \frac{1 - (T_d s / 2)}{1 + (T_d s / 2)}$$

Time Delay Implementation

Direct approach (exact calculation)

Process

$$G(s) = e^{-T_d s}\, G_0(s)$$

Notice, as $s = \sigma + j\omega$

$$\angle e^{-T_d s} = \angle (e^{-T_d \sigma} e^{-jT_d \omega}) = -T_d \omega$$

Modified root locus condition

$$\angle D(s)\, G(s) = \angle \left(D(s)\, G_0(s) \right) + \angle \left(e^{-T_d s} \right) = 180^\circ$$

$$\Downarrow$$

$$\underline{\angle D(s)\, G(s) = 180^\circ + T_d \omega}$$

However, Matlab does not support this approach...

How to build a transfer function in MATLAB?

- Suppose we consider the following equation:

$$Form1: G(s) = \frac{K(s+z_1)}{(s+p_1)(s+p_2)}$$

$$Form2: G(s) = \frac{Ks+Kz_1}{s^2+(p_1+p_2)s+p_1p_2} = \frac{s+k_2}{s^2+as+b}$$

- Now, we can develop this transfer function in MATLAB by the following ways:

- Form 1: tf1=zpk([z1],[p1,p2],k)

- Form 2: tf1=tf([1,k2],[1,a,b])

- You can draw the root-locus by the simple command: rlocus(tf1)

Some more useful commands

- For finding the response of the system – you can use the following commands as and when required:

- For step response: step(tf1)

- For impulse response: impulse(tf1)

- For other responses: lsim(tf1,U,T), where T is the time vector and U is the corresponding excitation vector.

- You can use nyquist(tf1) for obtaining the nyquist plot of the system and freqresp(tf1) for obtaining the frequency response.

Assignment

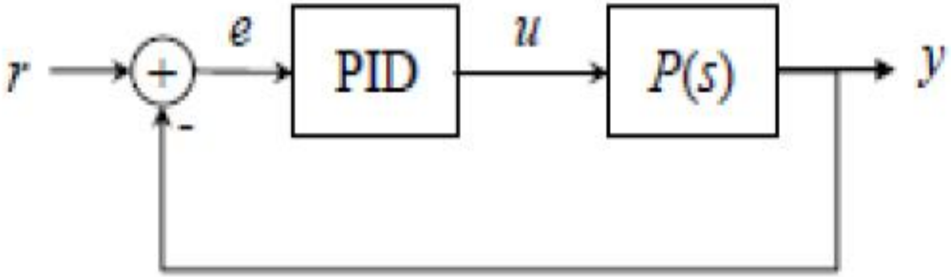
- Consider a unity feedback system with a plant transfer function:

$$G(s) = \frac{(s+6)}{(s+2)(s+4)(s+7)(s+8)}$$

- Input the transfer function in MATLAB and sketch the root locus.
- Find out the co-ordinates of the dominant poles for damping coefficient 0.707.
- Find the corresponding Gain.
- Find out the validity of assuming the system to be of second-order.

What is PID Control?

- Dynamic Systems are often controlled with the help of a three term compensator known as PID; P – stands for Proportional Control, I stands for Integral Control and D stands for Derivative Control.
- Consider a closed loop system with unity feedback as follows:



- The plant *P*, is controlled by a control input *u(t),* which can be expressed as follows:

$$u(t)=K_p e(t)+K_i \int e(t)\,dt+K_d \dot{e}(t)$$

Frequency Domain Representation of Controller

- **The control input could be represented in frequency domain as follows:**

$$U(s)=[K_p+\frac{K_i}{s}+K_d s]E(s)$$

- **The closed loop transfer function could be written as follows:**

$$\frac{Y(s)}{R(s)}=\frac{C(s)P(s)}{1+C(s)P(s)}$$

- **This tells us that the poles of the closed loop transfer functions are actually the zeros of *1+C(s)P(s)*. By considering the three terms as three parameters, you can study the effect of changing each one of these parameters on the root locus.**

Advantage of Different Parameters of PID Controller

Parameters	Advantage	Limitation
K_p	Adjustment of Controller output	May cause instability
K_i	Produces zero steady state error	Slow dynamic Response and Instability
K_d	Provides rapid system response	Sensitive to Noise and non-zero offset

Application of PI Controller on a First Order System

- Consider a first order plant as follows:

$$P(s)=\frac{K}{1+\tau s}$$

- If we apply a PI controller then *C(s)* becomes:

$$C(s)=K_p+\frac{K_i}{s}=K_p\frac{s+\frac{K_i}{K_p}}{s}$$

- The closed loop transfer function may be written as:

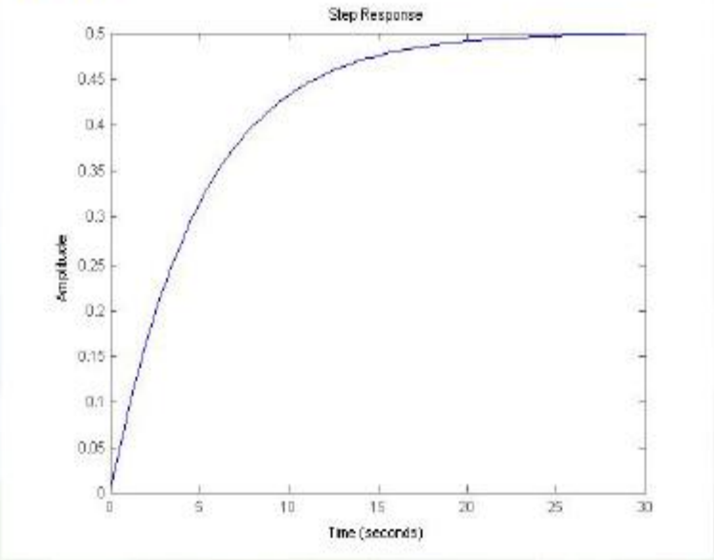
$$\frac{Y(s)}{R(s)}=(KK_p)\frac{s+K_i/K_p}{\tau s^2+(1+KK_p)s+KK_i}$$

Numerical Simulation of the system

- Let us consider the first order system with K=1 and τ = 10 s, hence the open loop system transfer function may be written as

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

- Let us look at the unit step response of this system – the controller has miserably failed to follow!



First Order System - Numerical Simulation

- Now let us consider a PI controller with the following parameters:

$$K = 1$$

$$K_p = 1$$

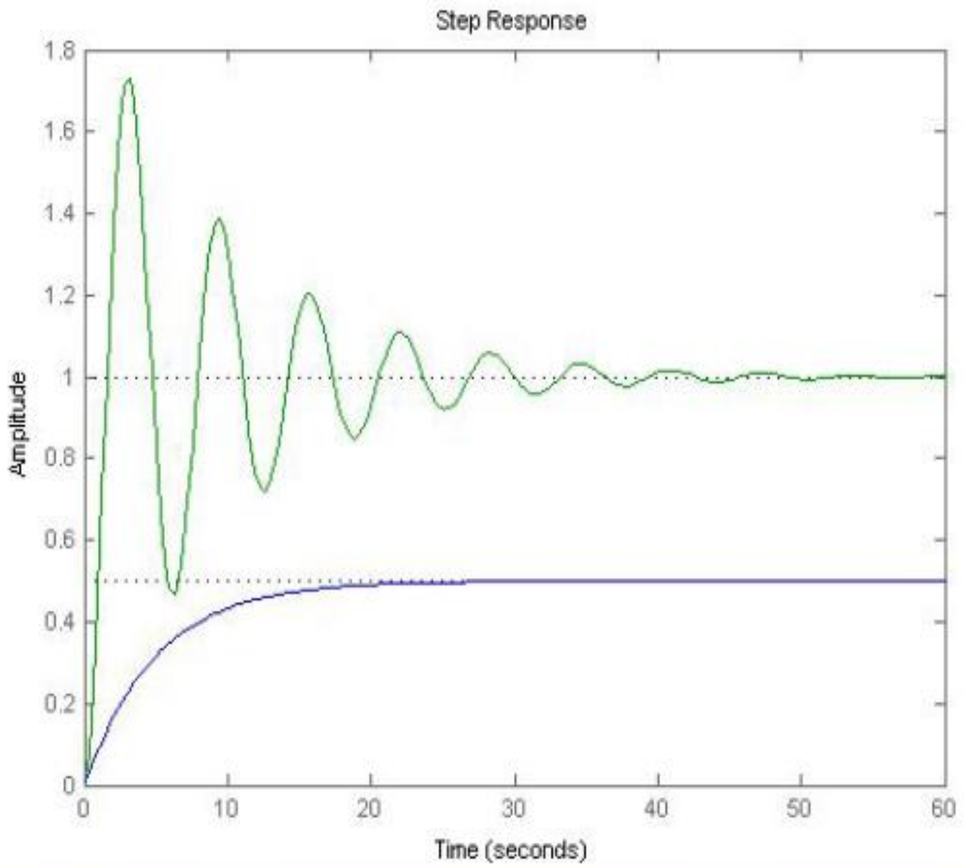
$$K_i = 10$$

- The new closed loop transfer function may be written as:

$$T = \frac{s + 10}{10s^2 + 2s + 10}$$

Response of the new system

- The unit step response of the new system is shown below vis a vis the old system:



Application of PD Controller on a First Order System

- Consider a first order plant as follows:

$$P(s)=\frac{K}{1+\tau s}$$

- If we apply a PD controller then *C(s)* becomes:

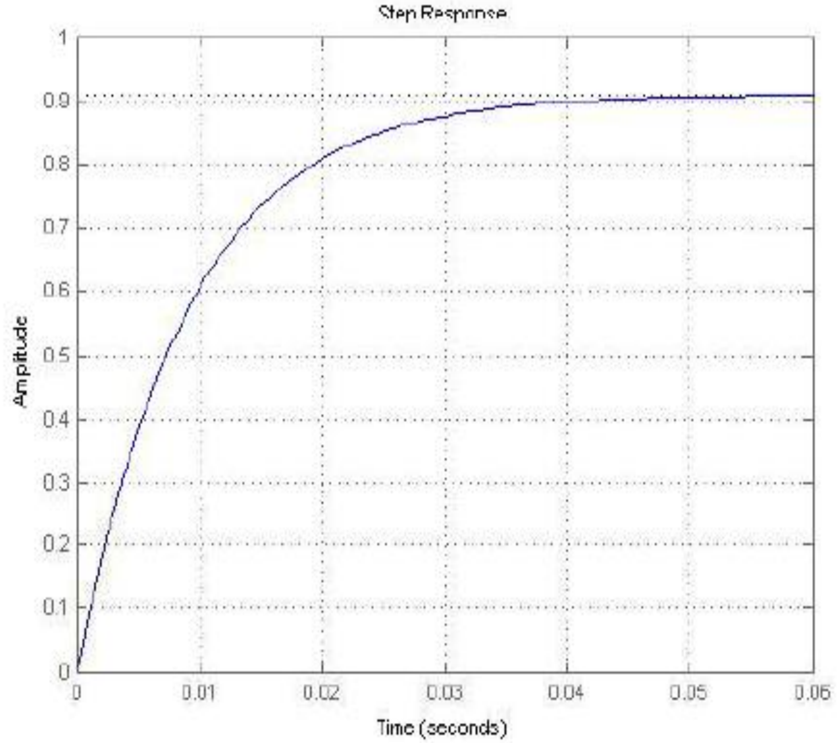
$$C(s)=K_p+K_d s=K_d\,(s+K_p/K_d)$$

- The closed loop transfer function may be written as:

$$\frac{Y(s)}{R(s)}=\frac{KK_d s+KK_p}{(\tau+KK_d)s\ \ +(1+K\,K_p)}$$

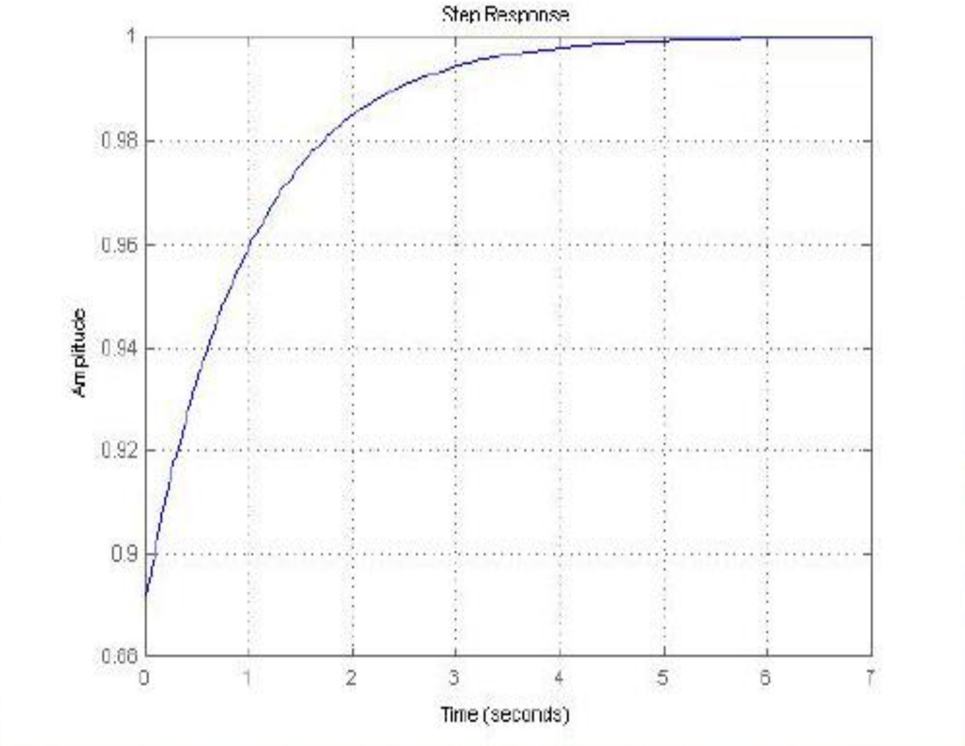
A numerical example

- Let us consider a numerical example where K=10 and τ = 0.1. The response of the system without any compensator is as follows: Note that it took about 0.05 seconds for the system to reach the steady state which is 0.9.



A Modified Response

- If you now add a compensator with K_p = 9 and K_d = 1, you can see the change in step response as follows. You may observe that the same value (0.9) is obtained in less than 0.2 seconds.



Assignment

Consider a second order system with natural frequency 10 Hz and Damping coefficient = 0.1. Find out a PID controller which will improve the steady state performance ten times and also improve the peak-time 5 times.

Hints: Consider the standard second order models discussed earlier and cascade it with a PID controller as shown in this lecture. Take trial values of $K_{p,}$ K_i and K_d and find out the response – continue till the performance is satisfactory.

How do we tune a PID Controller?

- In the last lecture, we have shown how by choosing the three different tuning constants of the PID compensator you can control the nature of the response. However, the question remains how can we obtain the actual values of these constants. Is it only through trial and error or are there any rules to obtain them?

- Sometimes in Industry people try to tune compensators intuitively. For example, we do know that the derivative constant helps to remove sluggish response of a system or the integral constant helps to remove offset errors. However, quite often it is found that these constants are interrelated. For example, increasing integral constant may help to improve the steady state response but it may increase the system overshoot.

- A popular rule for tuning which is being used since last century is known as Zeigler-Nichols rule. This was developed way back in 1942 and is still popular today.

Zeigler-Nichols Rule

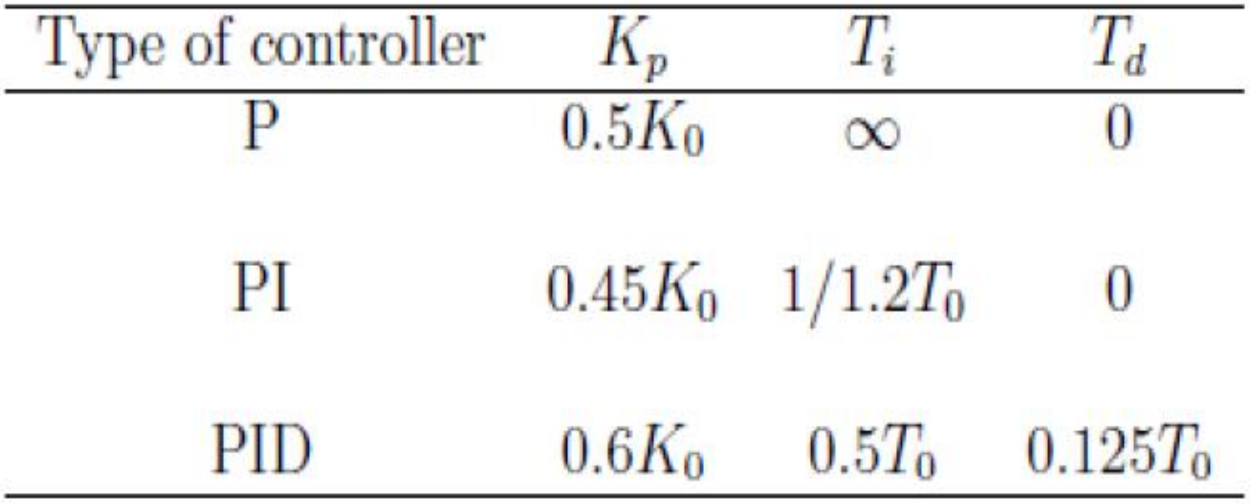
- There are two sets of rules. These rules are based on transient response of the system. The system dynamics may or may not be known to us. Let us now consider the control output *U(s)* to be defined in terms of the tuning constants as follows:

$$U(s) = K_p [1 + \frac{1}{T_i s} + T_d s] E(s)$$

- The first rule is for systems for which the exact dynamics is unknown.
- Using only a proportional controller first increase the gain so much that the response of the system starts to show oscillatory behavior. The corresponding gain could be termed as K_0 and the corresponding time period as $T_{0.}$

- Now the constants are determined by using the following table-

The Tuning Table for PID Controller



Type of controller	K_p	T_i	T_d
P	$0.5K_0$	∞	0
PI	$0.45K_0$	$1/1.2T_0$	0
PID	$0.6K_0$	$0.5T_0$	$0.125T_0$

The transfer function corresponding to the PID controller may be written as

$$T(s) = 0.07K_0T_0 \frac{(s + 4/T_0)^2}{s}$$

Zeigler-Nichol's tuning for Known Systems

- Now consider a dynamic system for which the system parameters are known and hence mathematical modeling is available.

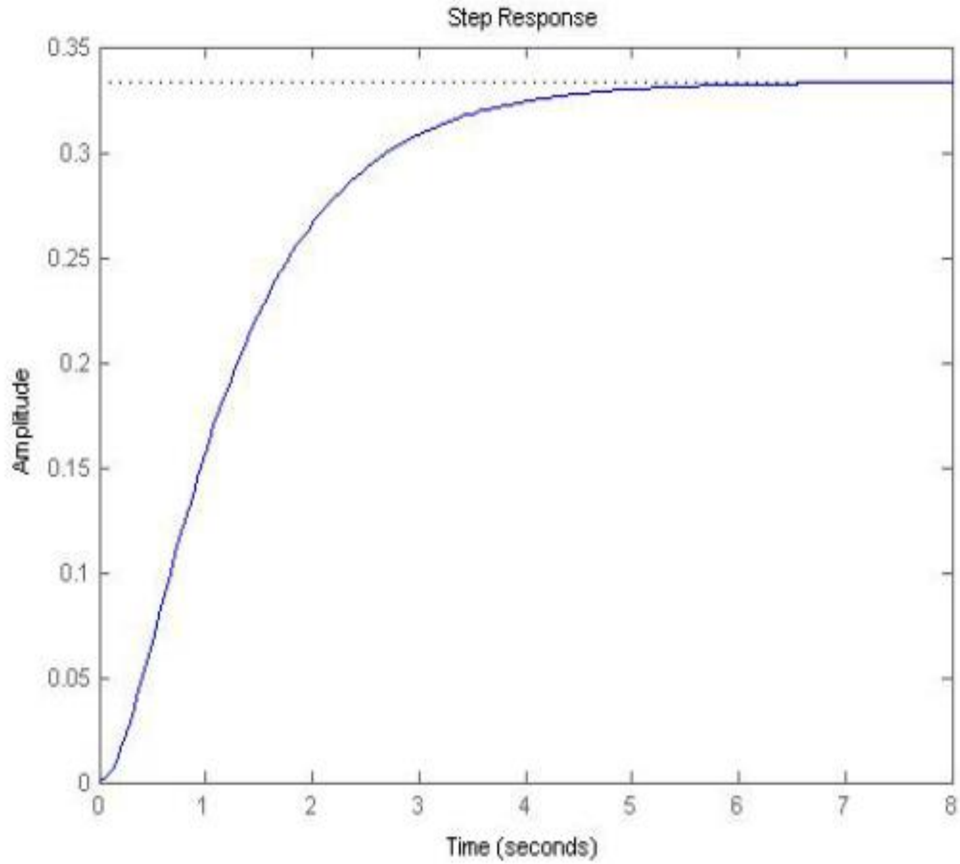
- Consider the system where $T_i = \infty$ and $T_d = 0$. Now for such a system, find out by using Routh's stability analysis the critical gain K_0 for which the roots cross the jω.

- Find out the frequency and the time period of the oscillation. Then, use the table mentioned earlier to determine the constants.

- These should be used as the starting point for tuning. Obtain the step response of the system and check whether the overshoot is less than 25%. If not, you should fine tune the system by moving the double zeros introduced by the PID controller.

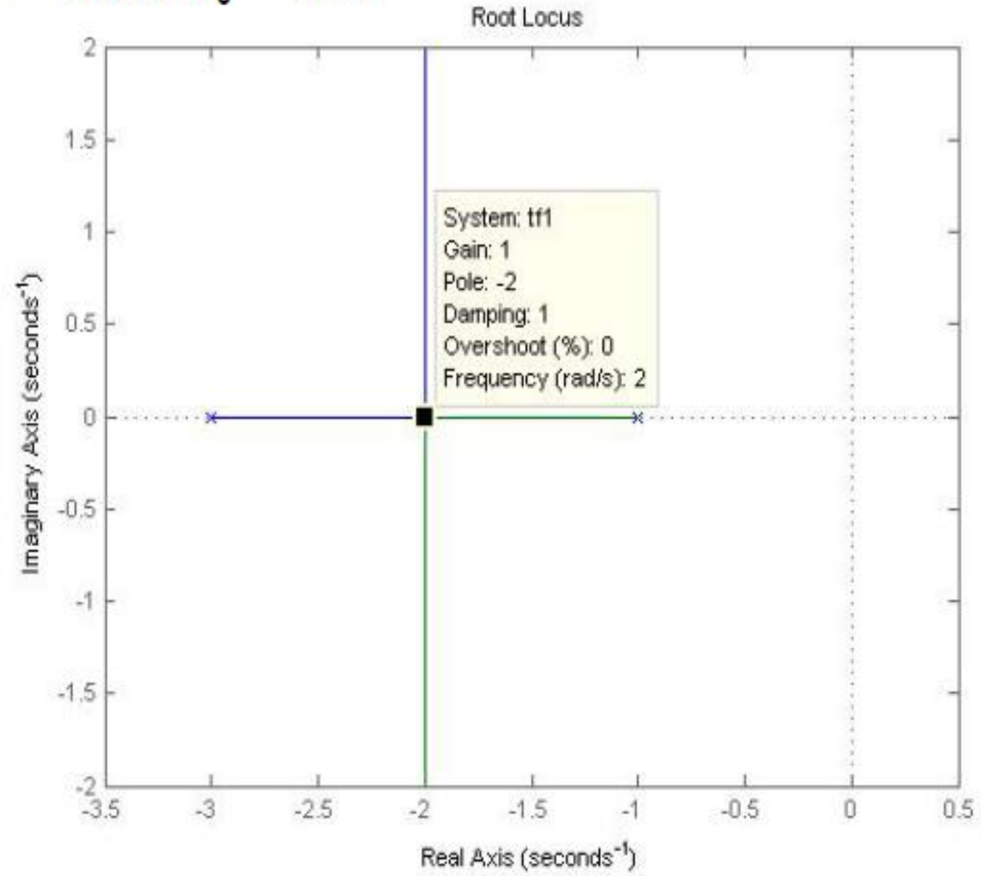
Simulation Example

Let us consider a 2nd-order system having two real poles at -1 and -3. The step response of the over-damped system is shown below:



Next step to find the Critical Gain:

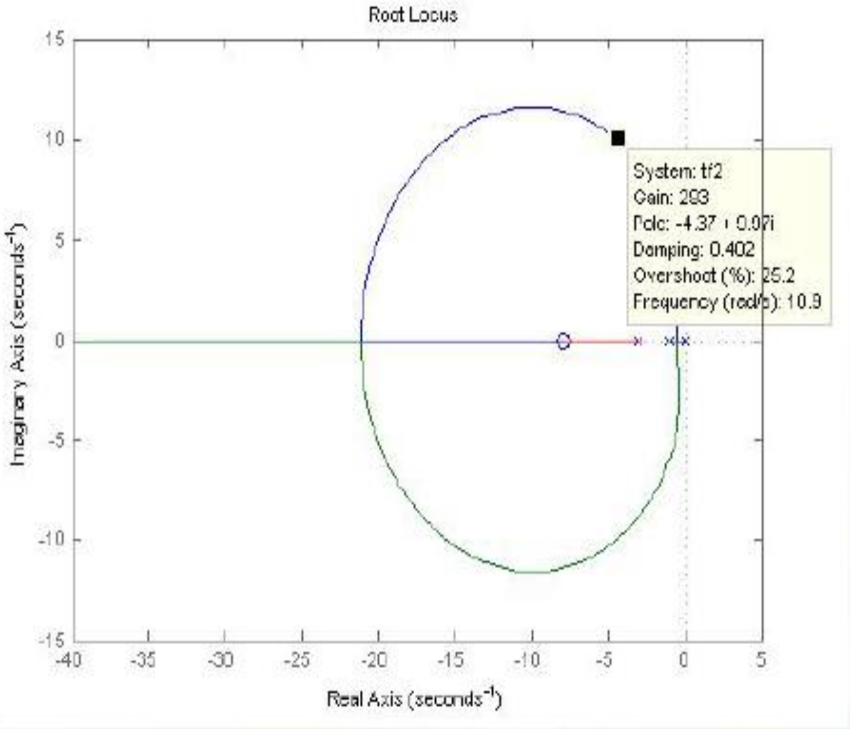
- Let us look at the root-locus of the system. At unit gain, the system will be just critically damped beyond which it will start to oscillate at 2 rad/s frequency. Hence, K_0 = 1 and T_0 = 0.5.



PID Controller following Z-N Rule:

- **Following the Z-N Table, the controller transfer function may be written as:**

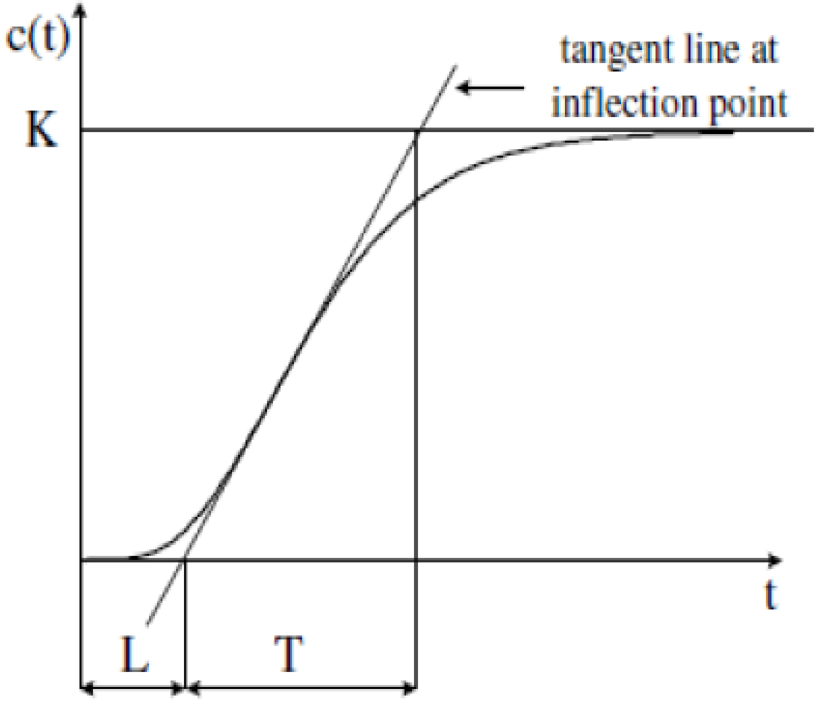
$$T(s) = 0.035\frac{(s+8)^2}{s}$$



The resultant system is found to have increased robustness and at gain 233, the overshoot is about 25% which is quite acceptable.

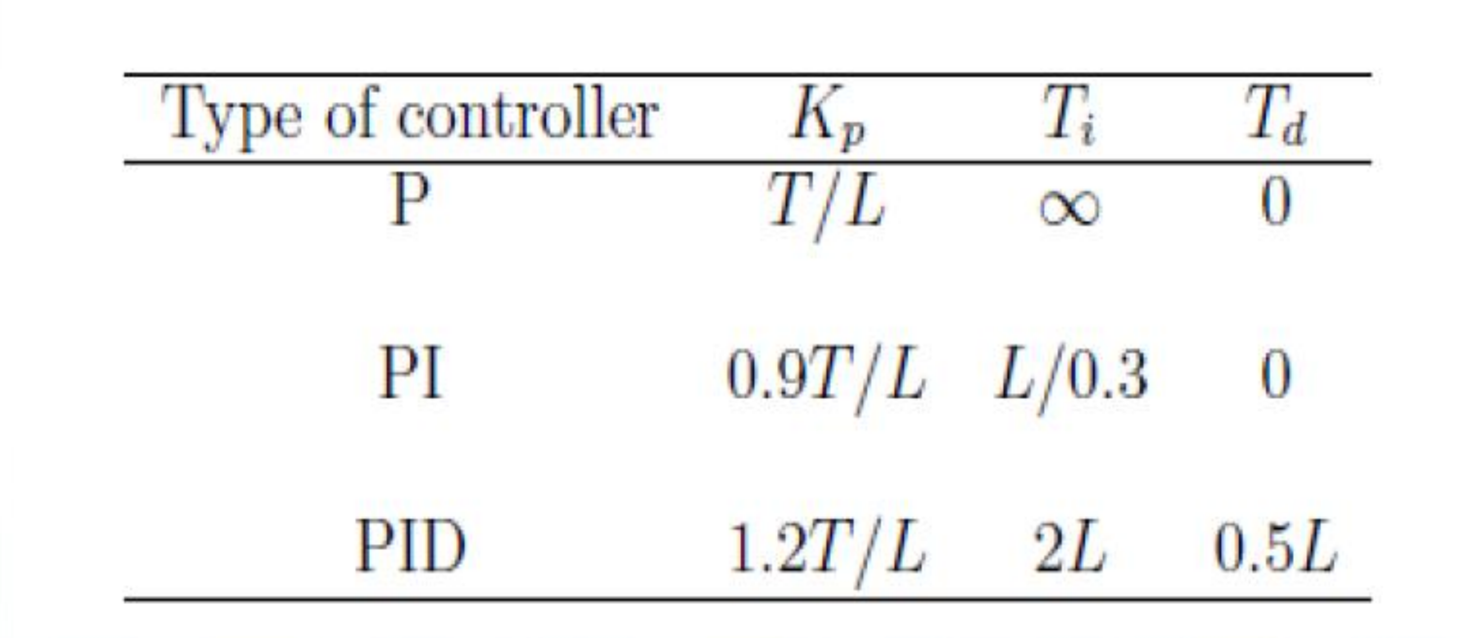
A Special Zeigler-Nichol's Rule for First Order System Behavior

- Let us consider a first order system which may be characterized as follows:



- Here, L is known as the time delay and T the time constant. The control gain could be evaluated in terms of these parameters as shown in the following table.

First Order System – A Special Table



Type of controller	K_p	T_i	T_d
P	T/L	∞	0
PI	$0.9T/L$	$L/0.3$	0
PID	$1.2T/L$	$2L$	$0.5L$

State Space Design

Dr. Bishakh Bhattacharya

Professor, Department of Mechanical Engineering

IIT Kanpur

Introduction to State Space Control

[Smallest, Linearly independent]

State – The states of a system refers to the property-set of the system that relates inputs to outputs such that the knowledge of the property set at any point of time along with inputs can completely define the output/response of the system in subsequent times.

Example: for a simple rotating pendulum the choice of states could be the position and angular velocity of the pendulum.

Note that for any system, the choice of states is not unique. However, the number of states needed to represent the system is unique.

Introduction (contd..)

State-space Control – Unlike frequency transform, in this technique, we preserve the time domain governing differential equations of a system. However, the differential equations describing the dynamics of the system are transformed into a set of first order differential equations in terms of the vectored representation of the states.

The solution of these first order ODEs yields a trajectory in space which Is spanned by the state vector. This space is known as State Space representation of the system.

Introduction (contd..)

The number of initial conditions required to solve the ODEs are unique and is equal to the number of states or the size of the state vector. This is also the same as the order of the system.

Consider the equation of motion of a single degree of freedom system described earlier:

$$M\ddot{x} + C\dot{x} + Kx = f(t)$$

This is a second order ODE, you need two initial conditions to solve the system, hence the order of the system is also two and you need to specify any two states to define the motion of the system. Let us Choose *x* and *dx/dt*.

State-space representation of a SDOF system

Thus, we can formally write the state vector X as:

$$X = \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix}$$

The second order ODE could be now represented as a set of two first order ODEs such that

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

$$A = \begin{bmatrix} 0 & 1 \\ -K/M & -C/M \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ f/M \end{bmatrix}$$

A Standard State-Space Form

$$\left.\begin{array}{l} \dot{x} = A\,x + B\,u \\ y = C\,x + D\,u \end{array}\right\} \text{State-space equations}$$

$$\left.\begin{array}{l} x = n \times 1 \\ A = n \times n \\ B = n \times m \\ u = m \times 1 \\ C = r \times n \\ D = r \times m \end{array}\right\} \text{Order of the matrices with standard notations}$$

Controller Canonical Form

State space representation is not unique in nature. Some of the commonly used forms are mentioned here:

$$\frac{d^n y}{dt^n} + \alpha_1 \frac{dy^{n-1}}{dt^{n-1}} + .. + \alpha_n y = \beta_0 \frac{d^n u}{dt^n} + \beta_1 \frac{du^{n-1}}{dt^{n-1}} + .. + \beta_n u$$

$$A = \left[\begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ .. & .. & .. & .. & .. \\ 0 & 0 & 0 & .. & 1 \\ \hline -\alpha_n & -\alpha_{n-1} & -\alpha_{n-2} & & -\alpha_1 \end{array}\right] \qquad B = \begin{bmatrix} 0 \\ 0 \\ \\ 0 \\ 1 \end{bmatrix}$$

$$C = [\beta_n - \alpha_n \beta_0 \quad 0 \quad 0 \quad .. \quad \beta_1 - \alpha_1 \beta_0] \qquad D = \beta_0$$

Observer Canonical Form

$$\frac{d^{\,n}y}{dt^{\,n}} + \alpha_1 \frac{dy^{\,n-1}}{dt^{\,n-1}} + \ldots + \alpha_n y = \beta_0 \frac{d^{\,n}u}{dt^{\,n}} + \ldots + \beta_{n-1}\frac{du}{dt} + \beta_n u$$

$$A = \left[\begin{array}{c|cccc} -\alpha_1 & 1 & 0 & 0 & 0 \\ -\alpha_2 & 0 & 1 & 0 & 0 \\ .. & .. & .. & .. & .. \\ -\alpha_{n-1} & 0 & 0 & 0 & 1 \\ -\alpha_n & 0 & 0 & 0 & 0 \end{array}\right] \qquad B = \begin{bmatrix} \beta_1 - \alpha_1\beta_0 \\ \beta_2 - \alpha_2\beta_0 \\ \\ \beta_{n-1} - \alpha_{n-1}\beta_0 \\ \beta_n - \alpha_n\beta_0 \end{bmatrix}$$

$$C = [1 \quad 0 \quad 0 \quad .. \quad 0] \qquad D = \beta_0$$

Jordan Canonical Form
(for non-repeated eigenvalues)

$$\frac{d^{\,n}y}{dt^{\,n}} + \alpha_1 \frac{dy^{\,n-1}}{dt^{\,n-1}} + .. + \alpha_n y = \beta_0 \frac{d^{\,n}u}{dt^{\,n}} + \beta_1 \frac{du^{\,n-1}}{dt^{\,n-1}} + .. + \beta_n u$$

$$Y(s) = [\beta_0 + \frac{P_1}{(s - \lambda_1)} + .. + + \frac{P_n}{(s - \lambda_n)}]U(s)$$

$\lambda_1 \ldots \lambda_n$ roots of $s^n + \alpha_1 s^{n-1} + .. + \alpha_n = 0$

$$A = \left[\begin{array}{ccccc} \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ .. & .. & .. & .. & .. \\ 0 & 0 & 0 & \lambda_{n-1} & 0 \\ \hline 0 & 0 & .. & 0 & \lambda_n \end{array}\right] \qquad B = \begin{bmatrix} 1 \\ 1 \\ . \\ 1 \\ 1 \end{bmatrix}$$

$$C = [P_1 \quad P_2 \quad .. \quad ..P_n] \qquad D = \beta_0$$

Solution of State-Space Equation

State space equations could be solved by following simple ODE solving procedure. Thus, at any time t_f the response of the state space system in standard form could be expressed as:

$$x(t_f) = e^{At_f} x(0) + \int_0^{t_f} e^{A(t_f-\tau)} B u(\tau) d\tau$$

The expression e^{At} is also known as state transition matrix and could be solved by procedures like Eigen-vector representation, Cayley-Hamilton Method and Resolvent Matrix Method.

$$e^{At} = I + \sum_{k=1}^{\infty} (At)^k / k! = \sum_{i=1}^{n} t_i e^{\lambda_i t} q_i$$

t_i, q_i are left & right eigen - vectors

Invariance of Eigen Values

$$\overset{\square}{x} = Ax + Bu, \qquad Use \qquad x = Tz$$

$$T\overset{\square}{z} = ATz + Bu$$

$$\overset{\square}{z} = T^{-1}ATz + T^{-1}Bu$$

$$\left|\lambda I - T^{-1}AT\right| = \left|\lambda T^{-1}T - T^{-1}AT\right|$$

$$= \left|T^{-1}(\lambda I - A)T\right|$$

$$= \left|T^{-1}\right|\left|\lambda I - A\right|\left|T\right|$$

$$= \left|\lambda I - A\right|$$

How do we check the System Stability?

In state space form, the stability of a system depends on the Eigen Values of A which may be obtained from the characteristic equation as follows. If the real parts of the roots of this equation are strictly negative then the system is considered to be asymptotically stable

$$|\lambda I - A| = 0$$

$e.g.$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}$$

$$|\lambda I - A| = \begin{vmatrix} \lambda & -1 & 0 \\ 0 & \lambda & -1 \\ 6 & 11 & \lambda + 6 \end{vmatrix}$$

$$= \lambda^3 + 6\lambda^2 + 11\lambda + 6$$

$$\lambda = -1, -2, -3 : \quad Stable \quad System$$

Reference: Control system by Norman Nise, Chapter 12 (Design via State space)