Implementation of Digital PID Controller

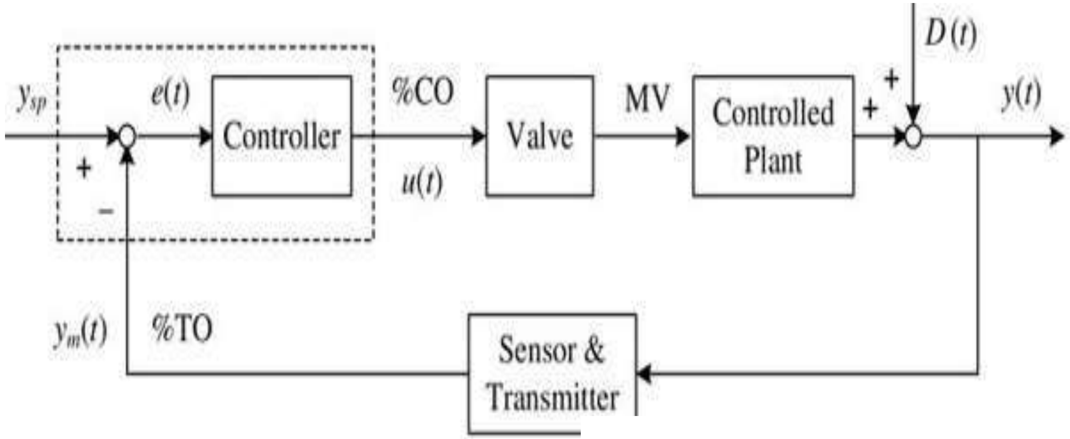
PRESENTED BY B KOTI REDDY

DEPARTMENT OF ATOMIC ENERGY

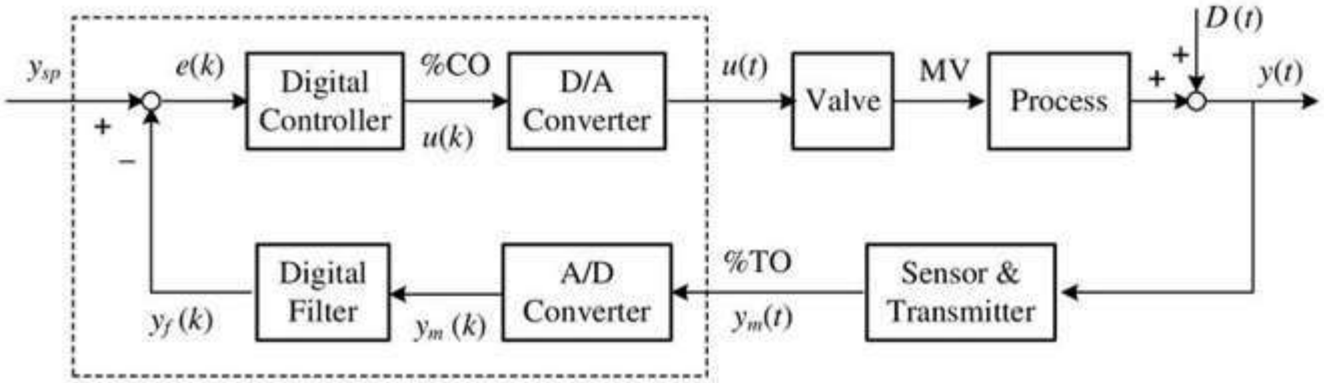
HEAVY WATER BOARD

Contents:

Construction of Digital Control System:

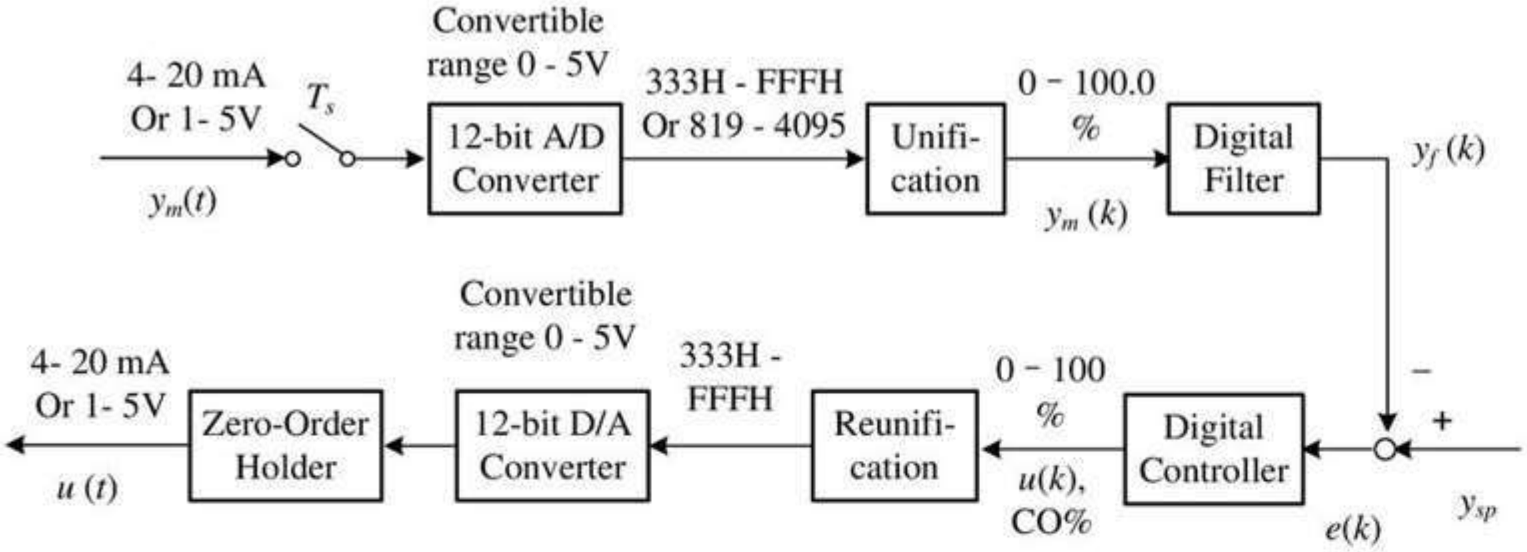


Analog Control System

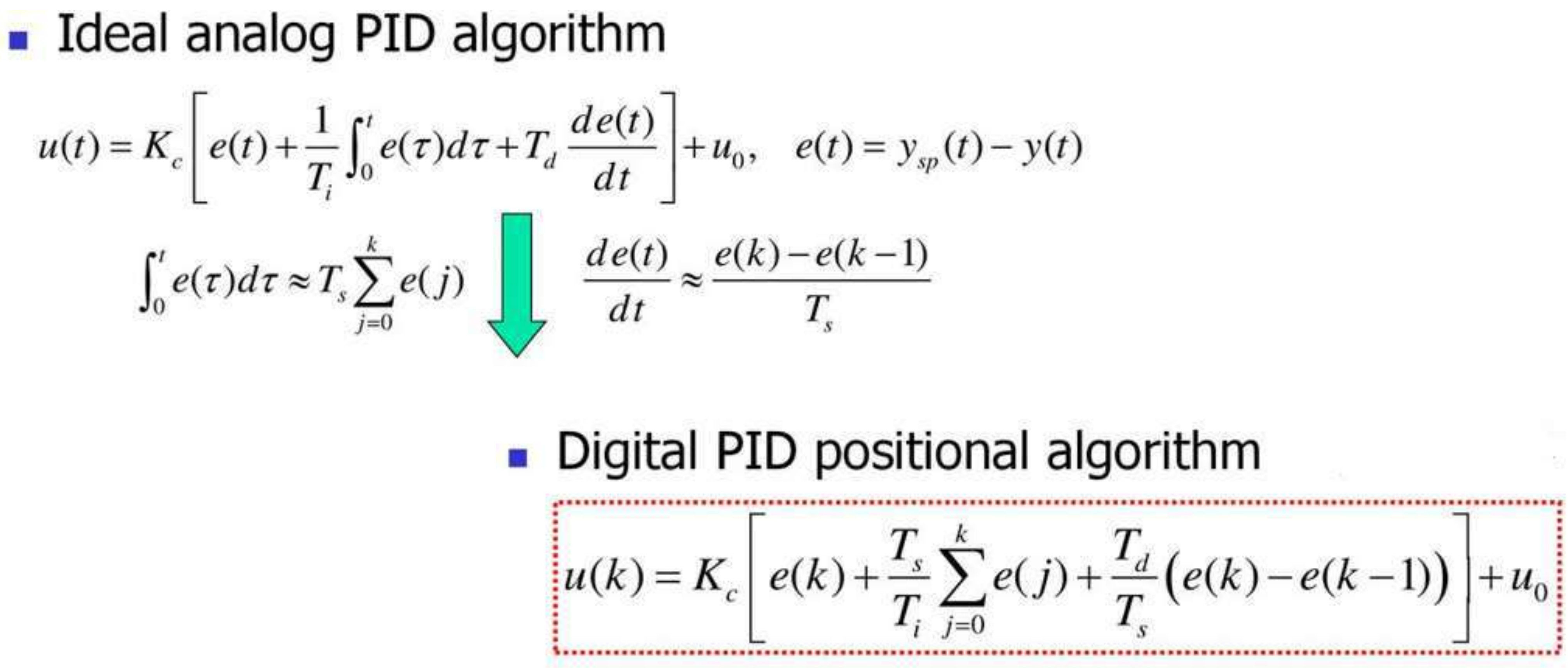


Digital Control System

General Digital Controller:



Digital PID Positioning Algorithm:



- Ideal analog PID algorithm

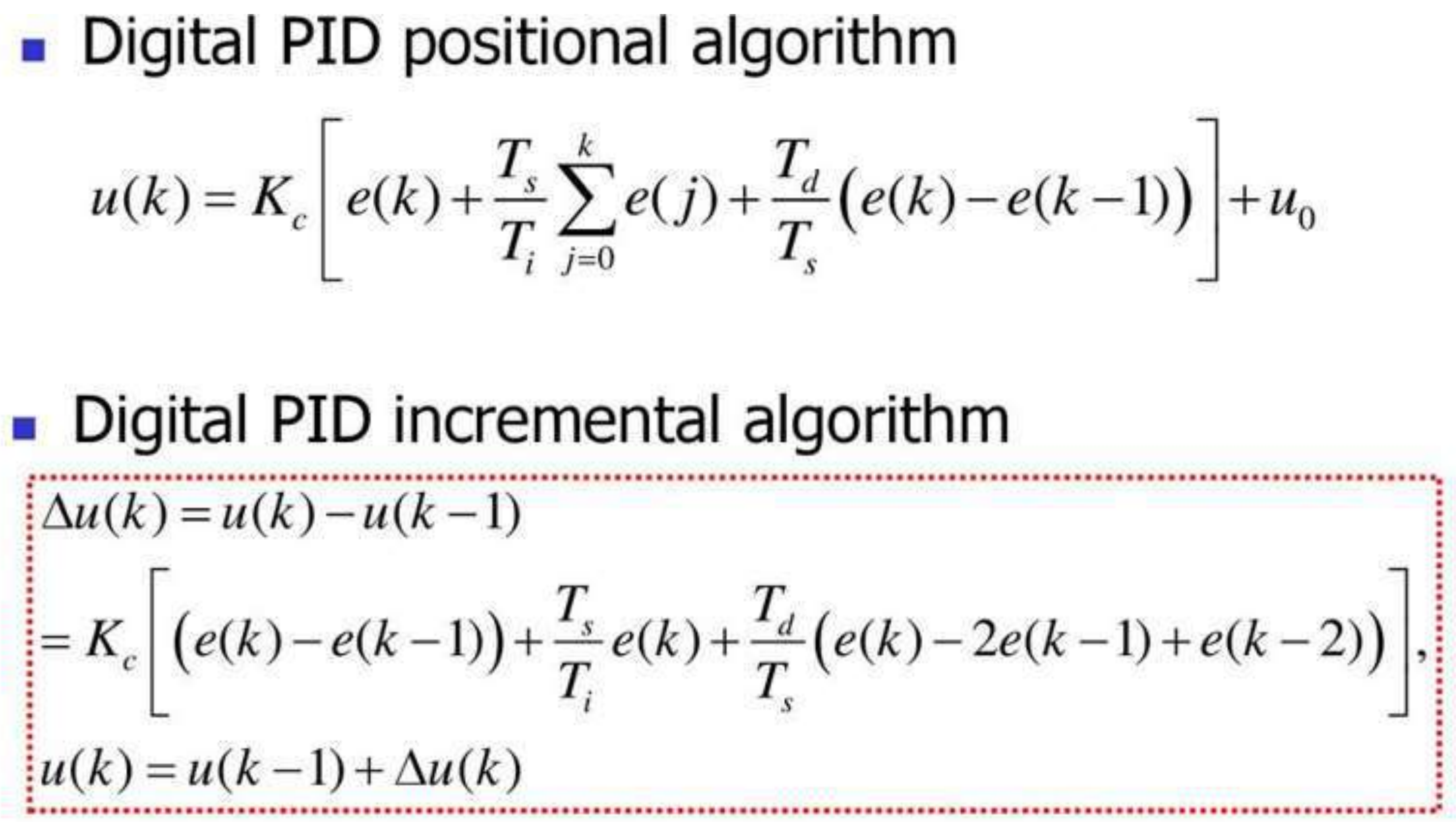
$$u(t) = K_c\left[e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right] + u_0, \quad e(t) = y_{sp}(t) - y(t)$$

$$\int_0^t e(\tau)d\tau \approx T_s\sum_{j=0}^{k} e(j) \qquad \frac{de(t)}{dt} \approx \frac{e(k)-e(k-1)}{T_s}$$

- Digital PID positional algorithm

$$\boxed{u(k) = K_c\left[e(k) + \frac{T_s}{T_i}\sum_{j=0}^{k} e(j) + \frac{T_d}{T_s}\big(e(k)-e(k-1)\big)\right] + u_0}$$

Digital PID Incremental Algorithm:



- Digital PID positional algorithm

$$u(k) = K_c\left[e(k) + \frac{T_s}{T_i}\sum_{j=0}^{k} e(j) + \frac{T_d}{T_s}\big(e(k) - e(k-1)\big)\right] + u_0$$

- Digital PID incremental algorithm

$$\begin{aligned}
&\Delta u(k) = u(k) - u(k-1)\\
&= K_c\left[\big(e(k) - e(k-1)\big) + \frac{T_s}{T_i}e(k) + \frac{T_d}{T_s}\big(e(k) - 2e(k-1) + e(k-2)\big)\right],\\
&u(k) = u(k-1) + \Delta u(k)
\end{aligned}$$

Digital PID Incremental Algorithm with Derivative :

- Digital PID incremental algorithm

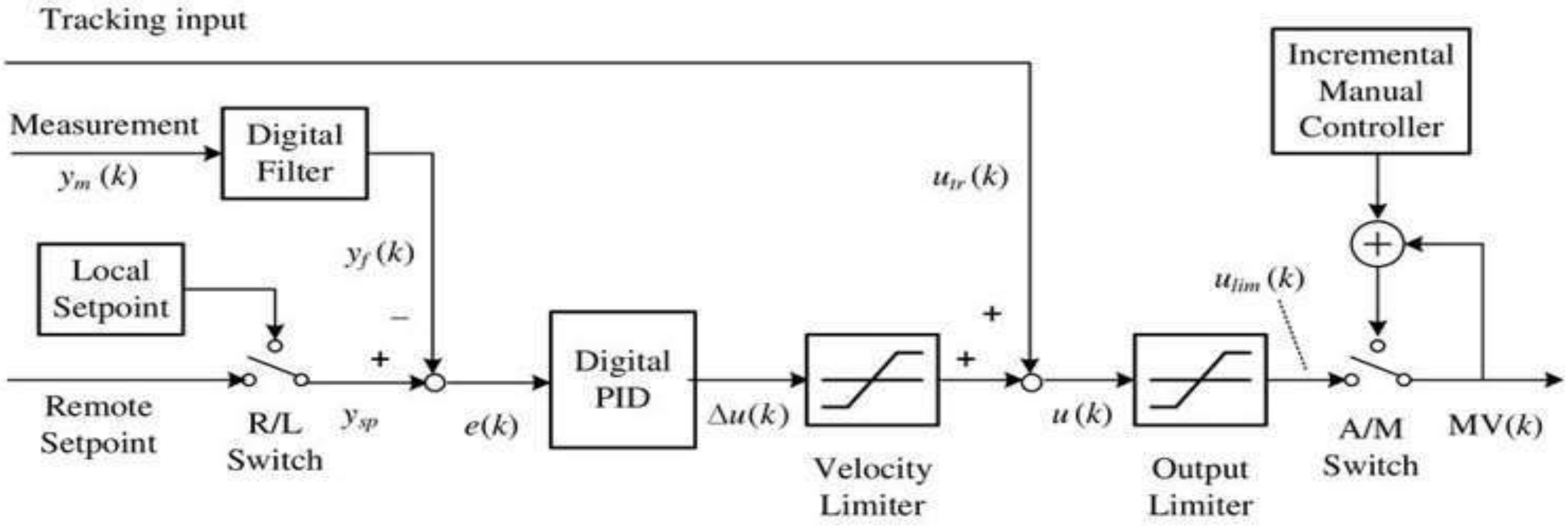
$$\Delta u(k) = K_c\left[\left(e(k)-e(k-1)\right)+\frac{T_s}{T_i}e(k)+\frac{T_d}{T_s}\left(e(k)-2e(k-1)+e(k-2)\right)\right]$$

- Digital PID incremental algorithm with derivative action first

$$\Delta u(k) = K_c\left[\left(e(k)-e(k-1)\right)+\frac{T_s}{T_i}e(k)+\frac{T_d}{T_s}\left(-y_f(k)+2y_f(k-1)-y_f(k-2)\right)\right],$$

$$u(k) = u(k-1)+\Delta u(k)$$

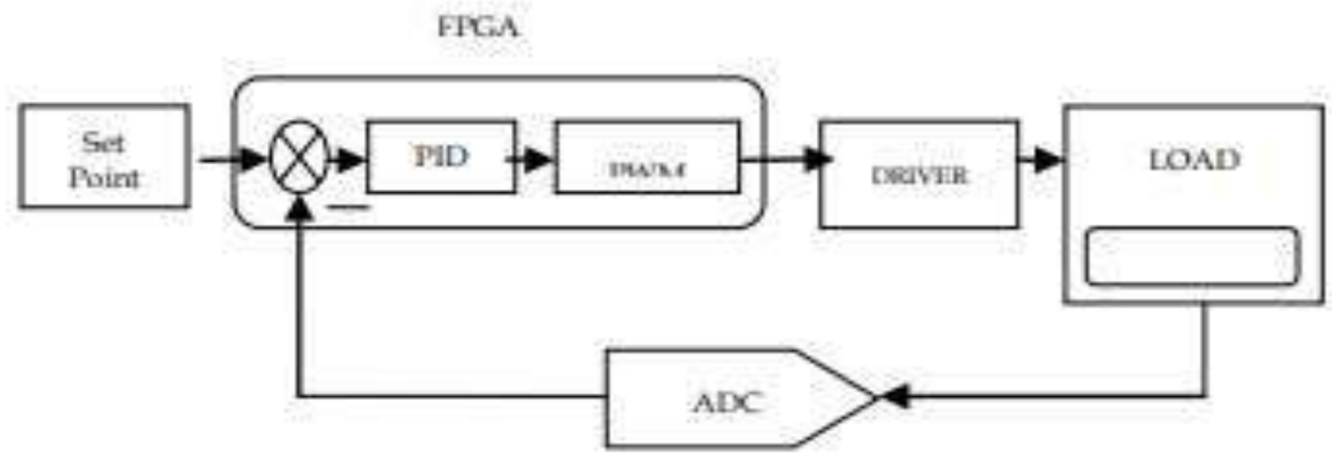
Industrial Digital PID Module:



Problem: （1）Programming； （2）Smooth A/M switching； （3）Reset windup prevention

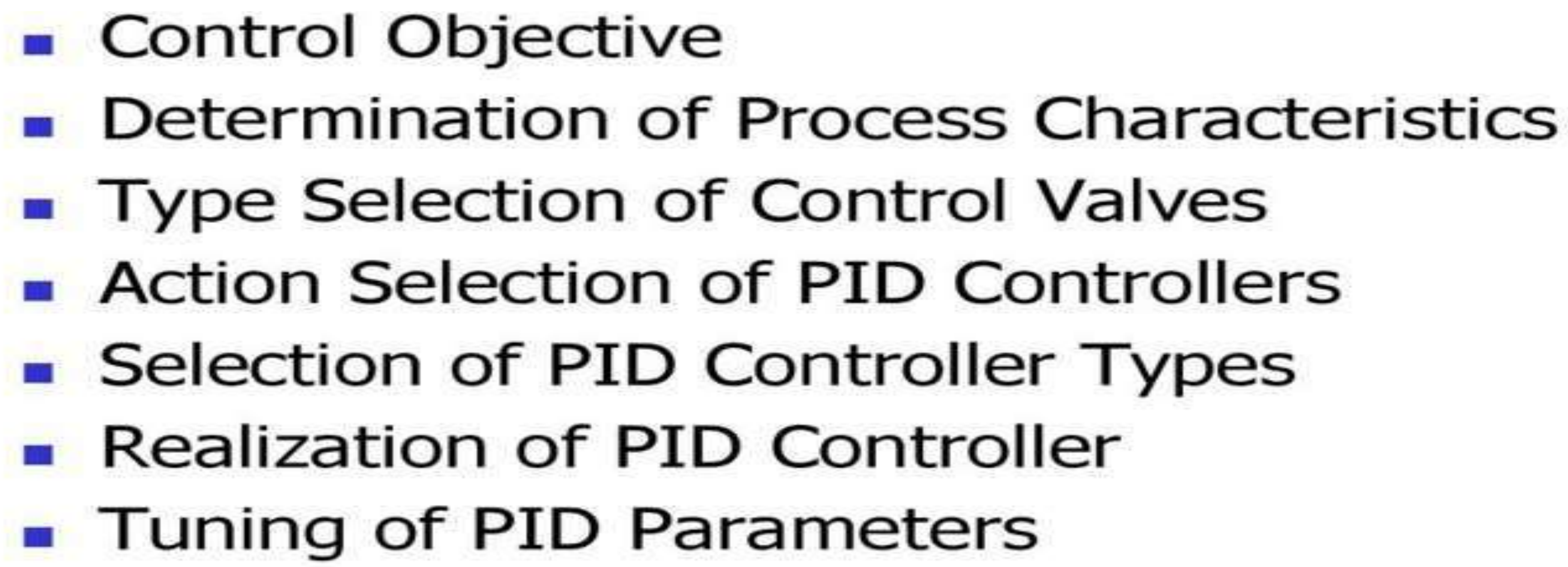
Digital PID Controller using UC &FPGA:

- Digital PID Controller can also be implemented Using Micro-controller and FPGA.



FPGA Based System Block Diagram

Summary For Single Loop PID Controller:



- Control Objective
- Determination of Process Characteristics
- Type Selection of Control Valves
- Action Selection of PID Controllers
- Selection of PID Controller Types
- Realization of PID Controller
- Tuning of PID Parameters

Reference:

1. “Digital Implementation of PID Controller for Temperature Control “, Prachi Rusia, International Journal of Scientific & Engineering Research Volume 8, Issue 5, May-2017 1774 ISSN 2229-5518.

2. “A Methodology to Design FPGA-based PID Controllers” João Lima, Ricardo Menotti, João M. P. Cardoso, and Eduardo Marques, IEEE International Conference on Systems, Man and Cybernetics, October8- 11, 2006, Taiwan

Thank You

MODEL ADAPTIVE REFERENCE CONTROL

Presented By B Koti Reddy

Department Of Atomic Energy

Heavy Water Board

CONTENTS:

Adaptive Control :

- Adaptive Control is the control method used by a controller which must adapt to a controlled system with parameters which vary or are initially uncertain. For example, as an aircraft flies, its mass will slowly decrease as a result of fuel consumption.

Classification of Adaptive control Techniques:

1. **Direct Method.**

Estimate the Controller Parameters.

2. Indirect Method.

Estimate the System Parameters.

3. Hybrid Method.

WHY ADAPTIVE CONTROL:

1. Systems To be Controlled have parameters Un-certinity.

2. System Dynamics experience unpredictable parameter variation as the control operation goes on.

Examples:

1. Robot Manipulation.

2. Ship Steering.

3. Aircraft Control.

- An Adaptive controller is a controller with adjustable parameters and a Mechanism of adjusting parameters.

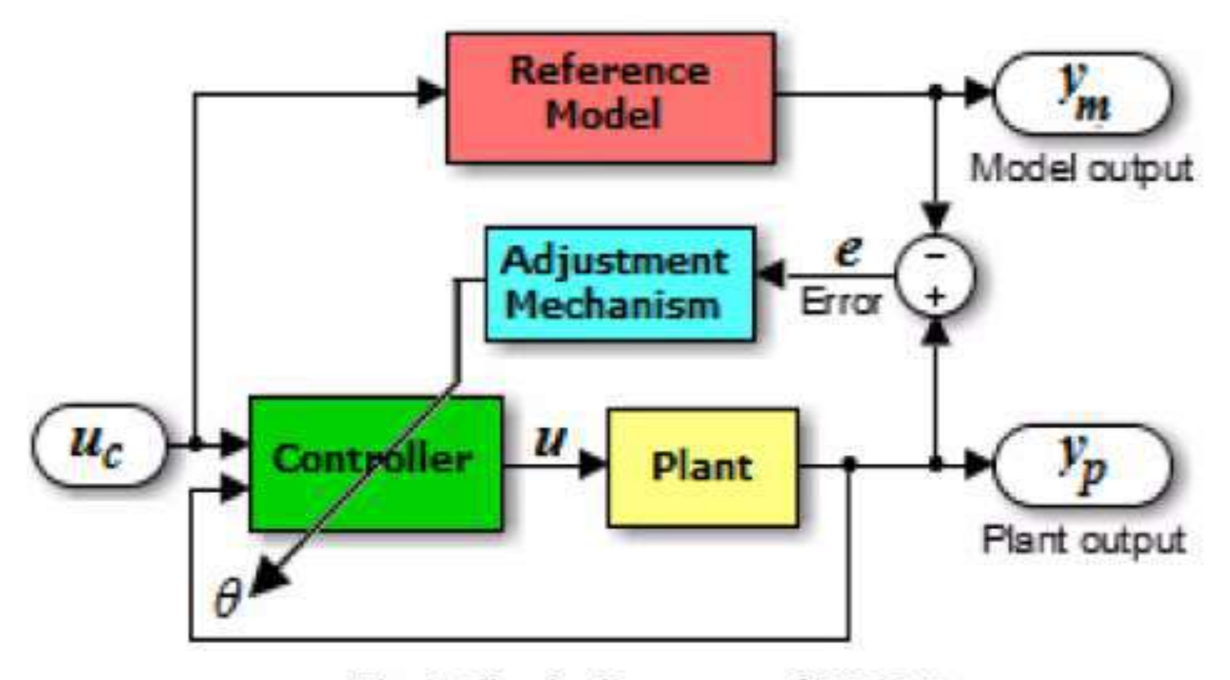


Fig.1 Block diagram of MRAC

MRAC Is Composed of :

1. Plant Containing Unknown Parameters.
2. Reference Model.
3. Adjustable parameters containing control Law.
4. Ordinary Feedback Loop.

Adjustable of System parameters in MRAC can be obtained in Two ways

1. Gradient Method.(MIT Rule)
2. Lypnov Stability Theory.

MIT RULE :

Tracking Error = e = Y-Ym

Introduce the Cost Function J:

$$J(\theta)=\frac{1}{2}e^2$$

Where ɵ is a vector of controller parameters

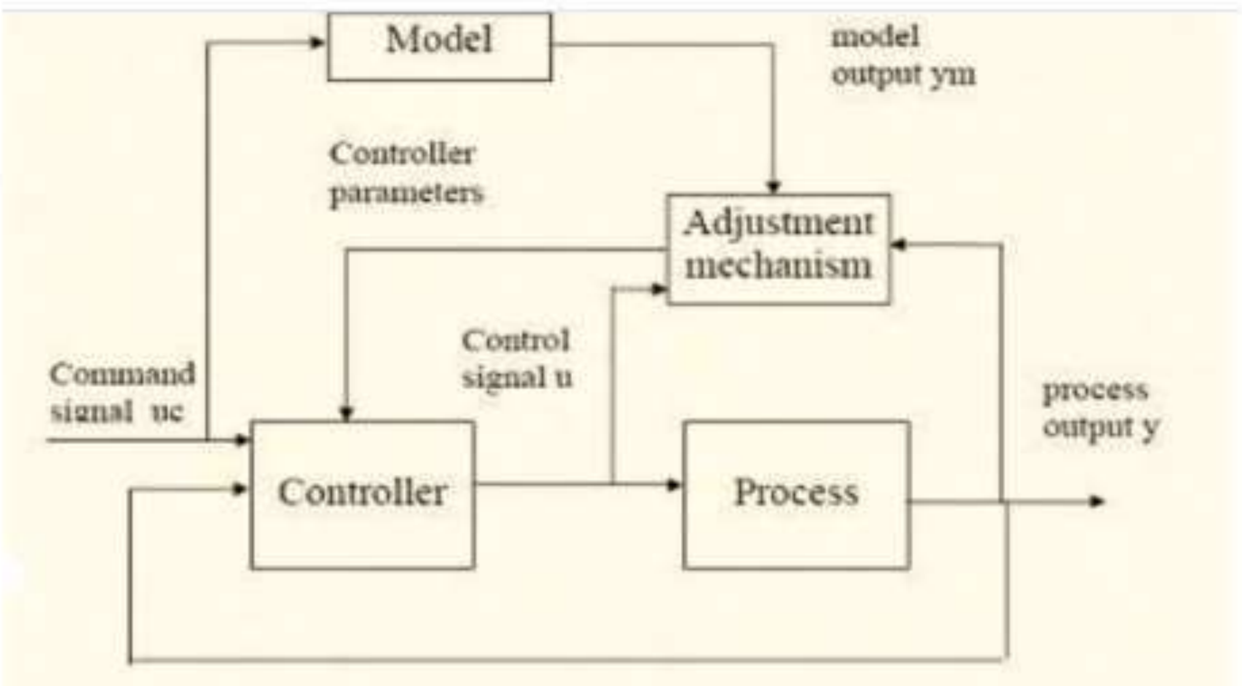


Fig.2 :Model Reference Adaptive Control Scheme

DEFINE THE MIT RULE OF $\frac{d\theta}{dt} = -\gamma\frac{\partial J}{\partial\theta} = -\gamma\, e\frac{\partial e}{\partial\theta}$

$\frac{\partial e}{\partial\theta}$ Is called the Sensitivity Derivative. It indicates how the error is influenced by the adjustable param

Example 1:

Process : $\frac{dy}{dt} = -a\,y + bu$

Model : $\frac{dy_m}{dt} = -a_m\,y_m + b_m u_c$

Controller : $u = \theta_1 u_c - \theta_2 y$

Closed Loop System :

$$\frac{dy}{dt} = -a\,y + bu = -ay + b(\theta_1 u_c - \theta_2 y) = -(a + b\theta_2)y + b\theta_1 u_c$$

Ideal Controller parameters for perfect model-Following :

$$\theta_1^0 = \frac{b_m}{b} \quad ; \quad \theta_2^0 = \frac{a_m - a}{b}$$

$$\frac{dy}{dt} = -\left(a + b\left(\frac{a_m - a}{b}\right)\right)y + b\frac{b_m}{b}u_c = -a_m y + b_m u_c$$

DERIVATION OF ADAPTIVE LAW:

Approximate $(s+a+b\theta_2)\approx(s+a_m)$

Error: $e = y - y_m$

where
$$s\,y = -(a+b\theta_2)sy + b\theta_1 u_c$$
$$(s+a+b\theta_2)y = b\theta_1 u_c$$
$$y = \frac{b\theta_1}{s+a+b\theta_2}u_c$$

$$\frac{dy}{dt} = sy$$

Sensitivity derivatives :

$$\frac{\partial e}{\partial \theta_1} = \frac{b}{s+a+b\theta_2}u_c \qquad \frac{\partial e}{\partial \theta_2} = -\frac{b^2\theta_1}{(s+a+b\theta_2)^2}u_c = -\frac{b}{s+a+b\theta_2}y$$

$$\frac{\partial \theta_1}{\partial t} = -\gamma'\left(\frac{b}{s+a_m}u_c\right)e \Rightarrow \theta_1 = \left(\frac{-\gamma}{s}\right)\left(\frac{a_m}{s+a_m}u_c\right)e$$

$$\frac{\partial \theta_2}{\partial t} = -\gamma'\left(\frac{b}{s+a_m}y\right)e \Rightarrow \theta_2 = \left(\frac{\gamma}{s}\right)\left(\frac{a_m}{s+a_m}y\right)e$$

Where $\gamma = \frac{\gamma' b}{a_m}$

Thank You

Control System Design

Presented By B Koti Reddy
Department Of Atomic Energy
Heavy Water Board

Contents :

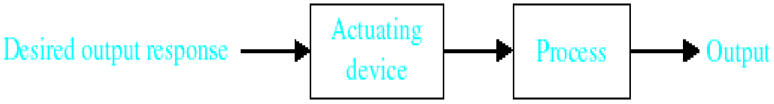
- Introduction.
- **Examples of Modern Control Systems.**
- Control System Design.

Introduction :

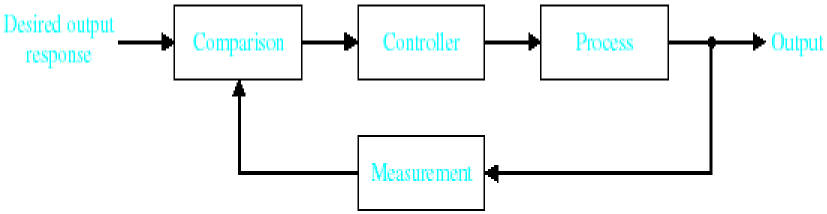
- **System** – An interconnection of elements and devices for a desired purpose.
- **Control System** – An interconnection of components forming a system configuration that will provide a desired response.
- **Process** – The device, plant, or system under control. The input and output relationship represents the cause-and-effect relationship of the process.

Introduction :

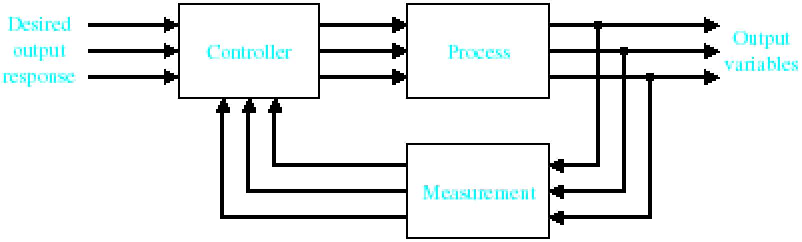
- **Open-Loop Control Systems** utilize a controller or control actuator to obtain the desired response.
- **Closed-Loop Control Systems** utilizes feedback to compare the actual output to the desired output response.
- Multivariable Control System



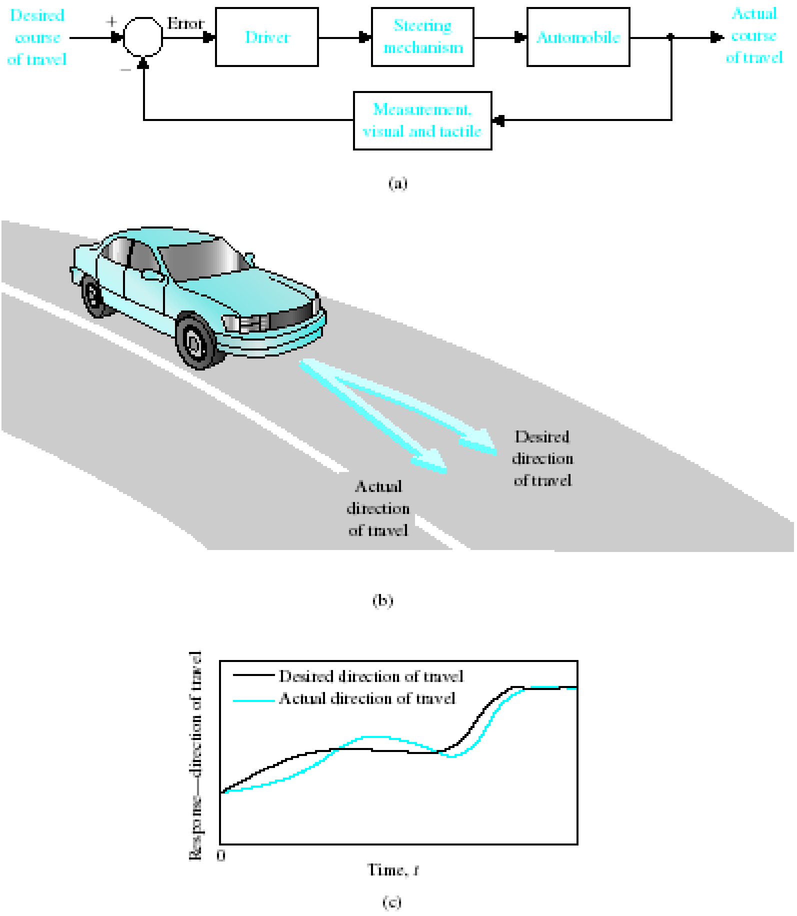
Open-loop control system (without feedback).



Closed-loop feedback control system (with feedback).



Examples of Modern Control Systems :



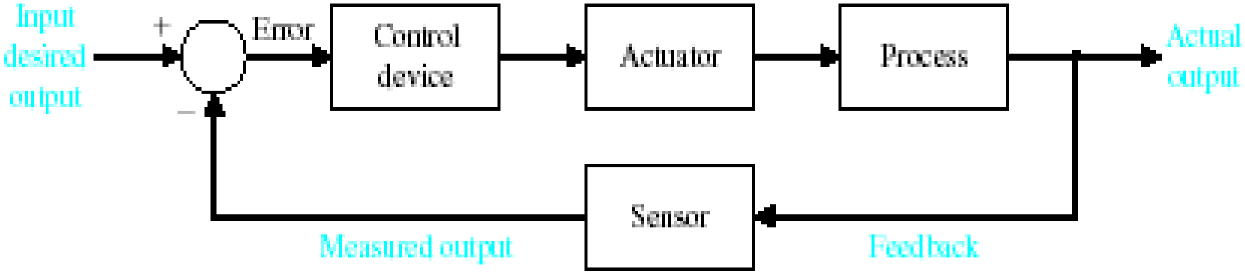
(a) Automobile steering control system.

(b) The driver uses the difference between the actual and the desired direction of travel

to generate a controlled adjustment of the steering wheel.

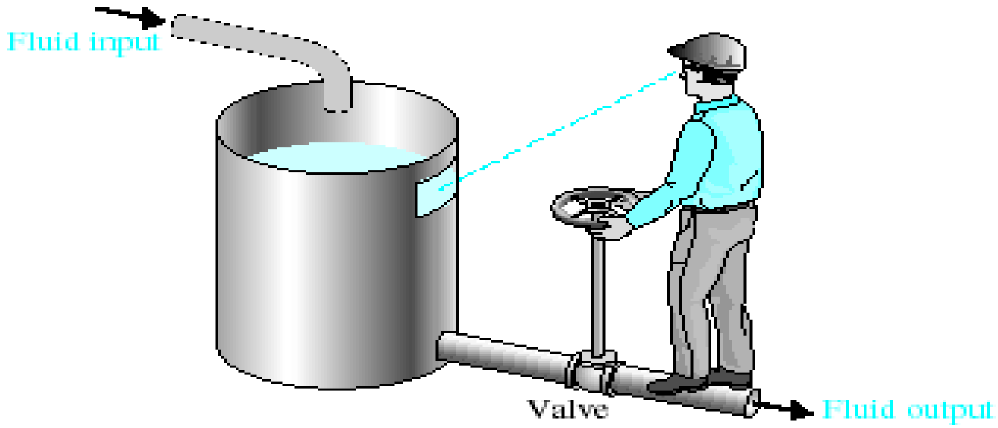
(c) Typical direction-of-travel response.

Examples of Modern Control Systems :



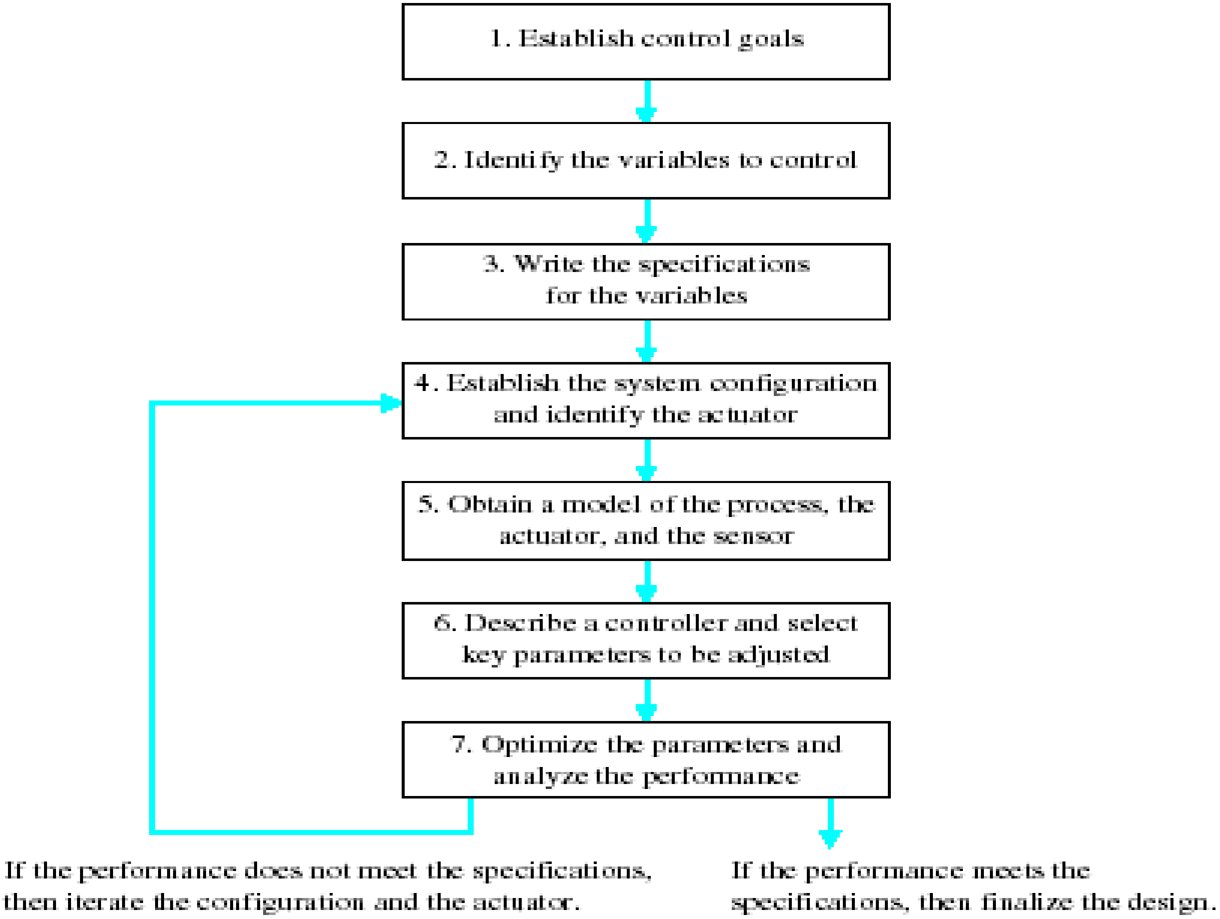
A negative feedback system block diagram depicting a basic closed-loop control system. The control device is often called a "controller."

Examples of Modern Control Systems :



A manual control system for regulating the level of fluid in a tank by adjusting the output valve. The operator views the level of fluid through a port in the side of the tank.

Control System Design :



1. Establish control goals

2. Identify the variables to control

3. Write the specifications for the variables

4. Establish the system configuration and identify the actuator

5. Obtain a model of the process, the actuator, and the sensor

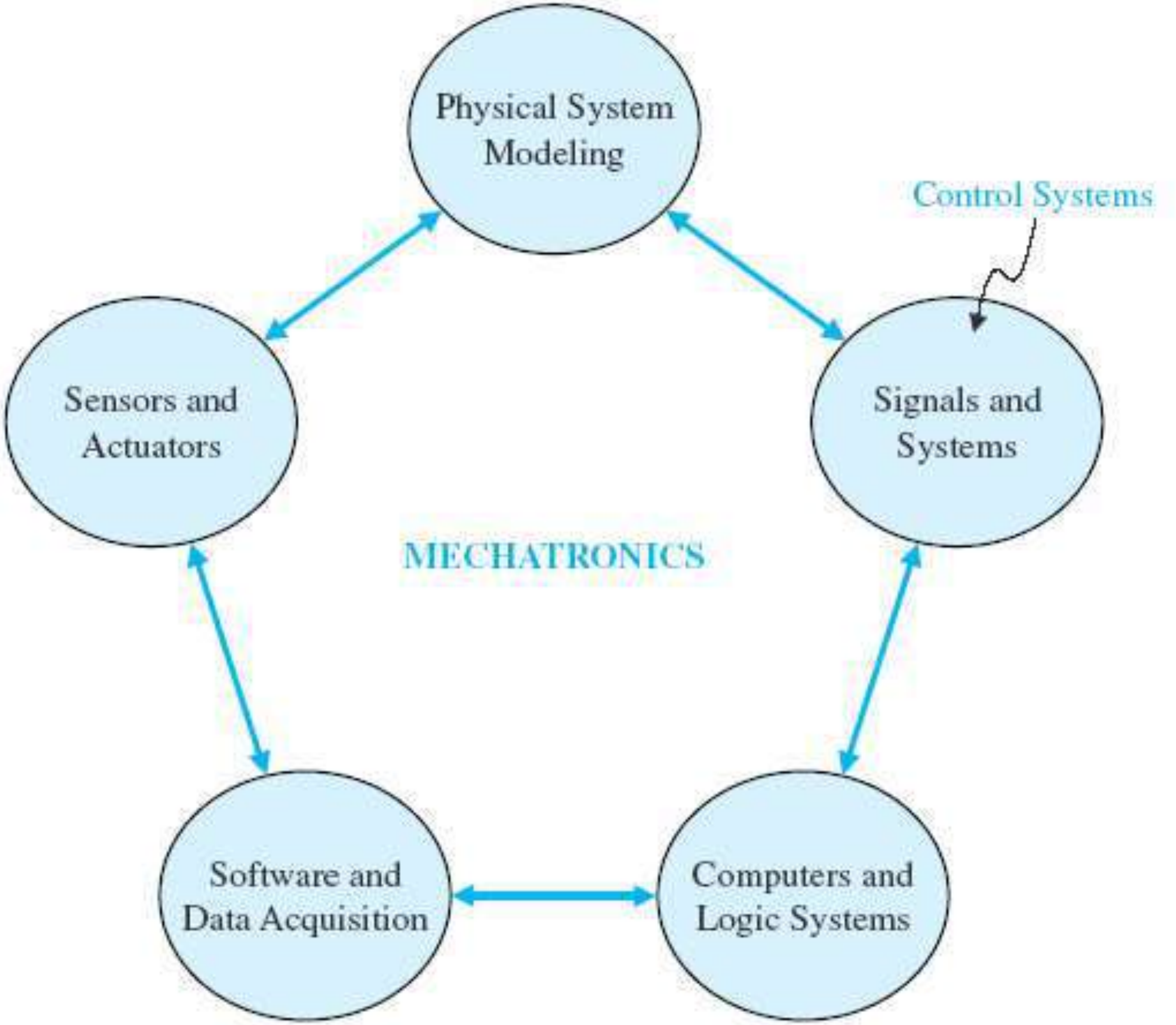
6. Describe a controller and select key parameters to be adjusted

7. Optimize the parameters and analyze the performance

If the performance does not meet the specifications, then iterate the configuration and the actuator.

If the performance meets the specifications, then finalize the design.

The control system design process.


Physical System Modeling
Control Systems
Sensors and Actuators
Signals and Systems
MECHATRONICS
Software and Data Acquisition
Computers and Logic Systems

Thank You

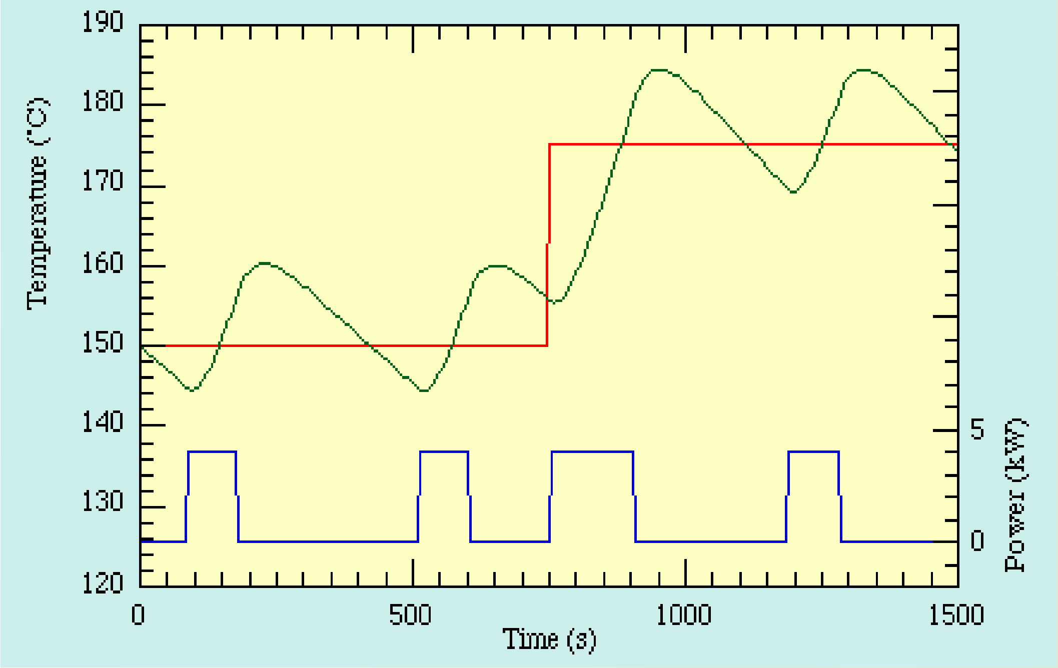
PID and Advanced Process Control

Presented By B Koti Reddy
Department Of Atomic Energy
Heavy Water Board

Basic Process Control: 1. Feedback Control.
2. Feed Forward Control.

Different Types of Feedback Control:

On-Off Control
This is the simplest form of control.



Proportional Control:

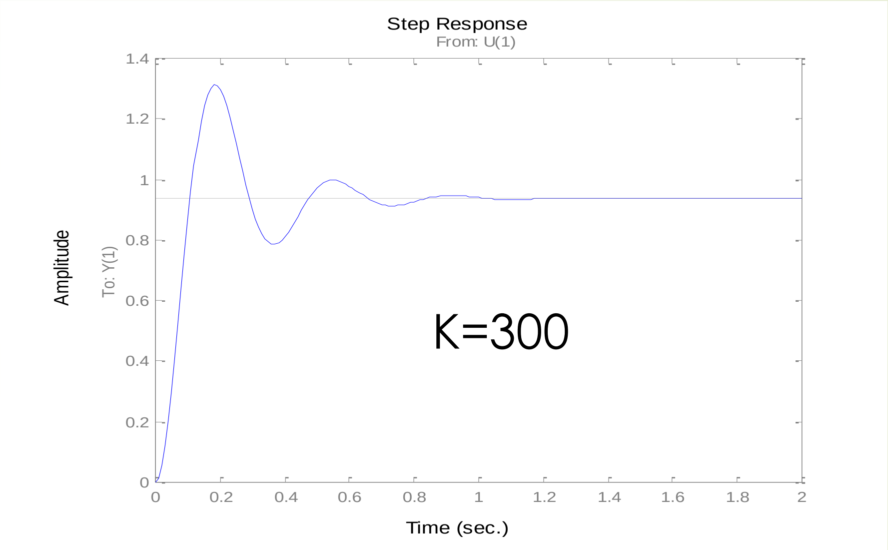
A proportional controller attempts to perform better than the On-off type by applying power in proportion to the difference in temperature between the measured and the set-point. As the gain is increased the system responds faster to changes in set-point but becomes progressively underdamped and eventually unstable. The final temperature lies below the set-point for this system because some difference is required to keep the heater supplying power.

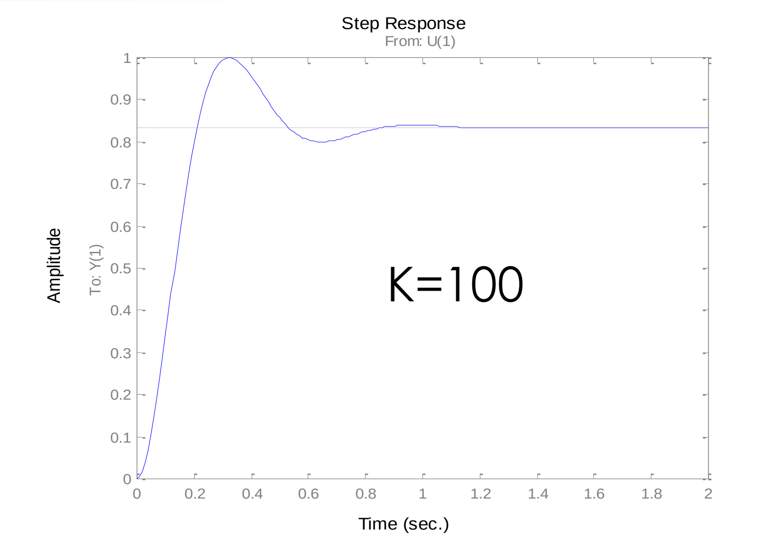
The proportional controller (Kp) reduces the rise time, increases the overshoot, and reduces the steady-state error.

<u>MATLAB Example:</u>

```
Kp=300;

num=[Kp];

den=[1 10 20+Kp];

t=0:0.01:2;

step(num,den,t)
```

$$T(s) = \frac{Kp}{s^{2} + 10 \cdot s + (20 + Kp)}$$





Proportional, Derivative Control:

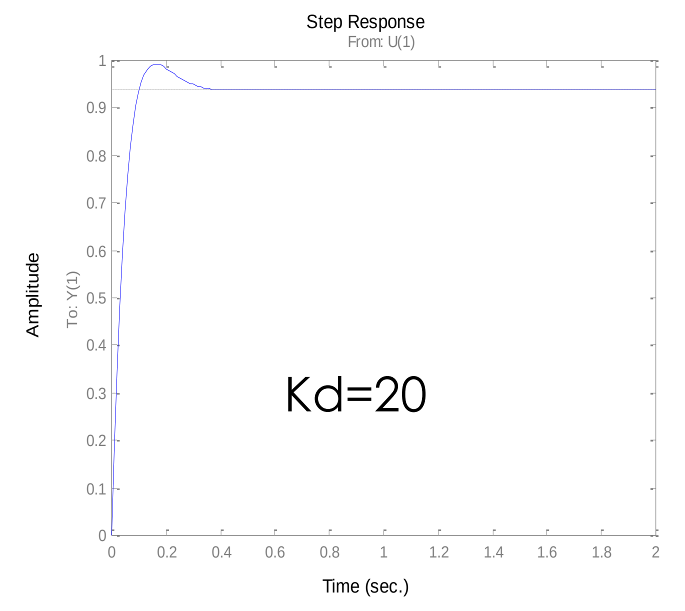
The stability and overshoot problems that arise when a proportional controller is used at high gain can be mitigated by adding a term proportional to the time-derivative of the error signal. The value of the damping can be adjusted to achieve a critically damped response.

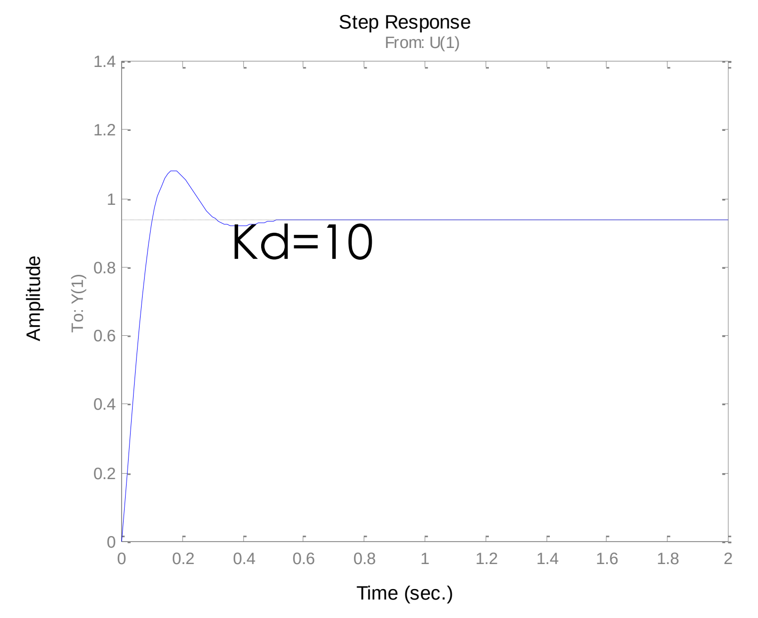
The derivative controller (Kd) reduces both the overshoot and the settling time.

MATLAB Example

$$T(s) = \frac{Kd \cdot s + Kp}{s^2 + (10 + Kd)\cdot s + (20 + Kp)}$$

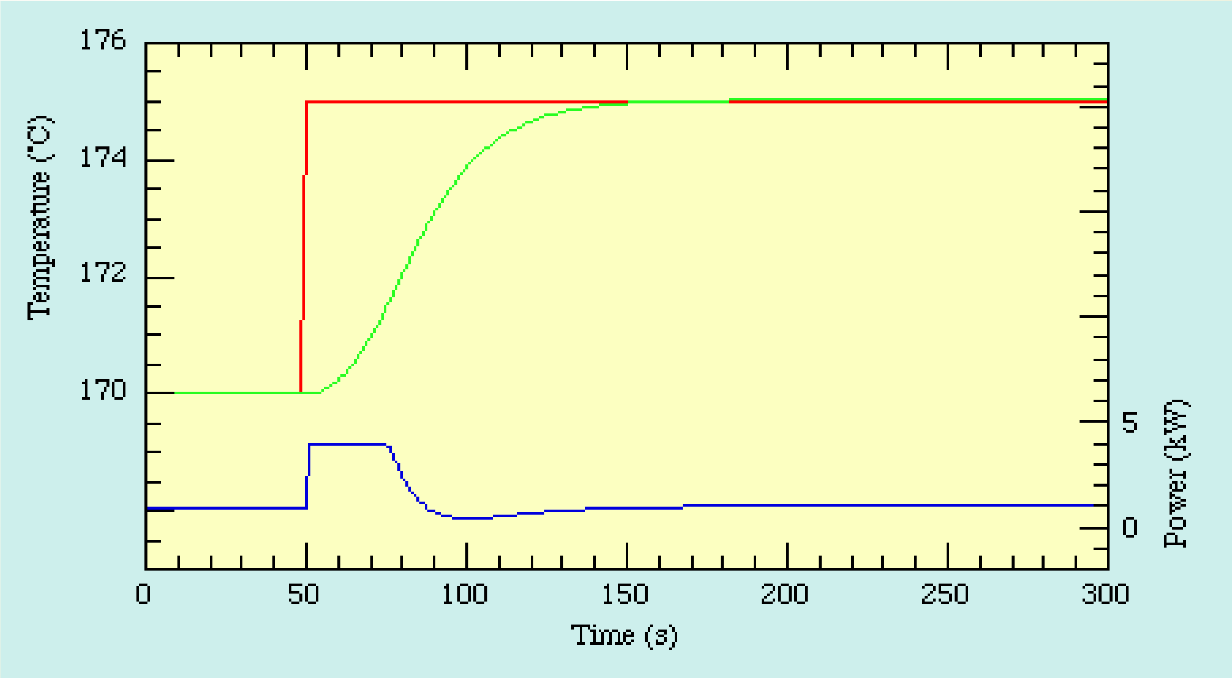
```
Kp=300;
Kd=10;
num=[Kd Kp];
den=[1 10+Kd 20+Kp];
t=0:0.01:2;
step(num,den,t)
```





Proportional +Integral + Derivative Control:

Although PD control deals neatly with the overshoot and problems associated with proportional control it does not cure the problem with the steady-state error. Fortunately it is possible to eliminate this while using relatively low gain by adding an integral term to the control function which becomes



The Characteristics of P, I, and D controllers:

A proportional controller (Kp) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error.

An integral control (Ki) will have the effect of eliminating the steady-state error, but it may make the transient response worse.

A derivative control (Kd) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.

Proportional Control

By only employing proportional control, a steady state error occurs.

Proportional and Integral Control

The response becomes more oscillatory and needs longer to settle, the error disappears.

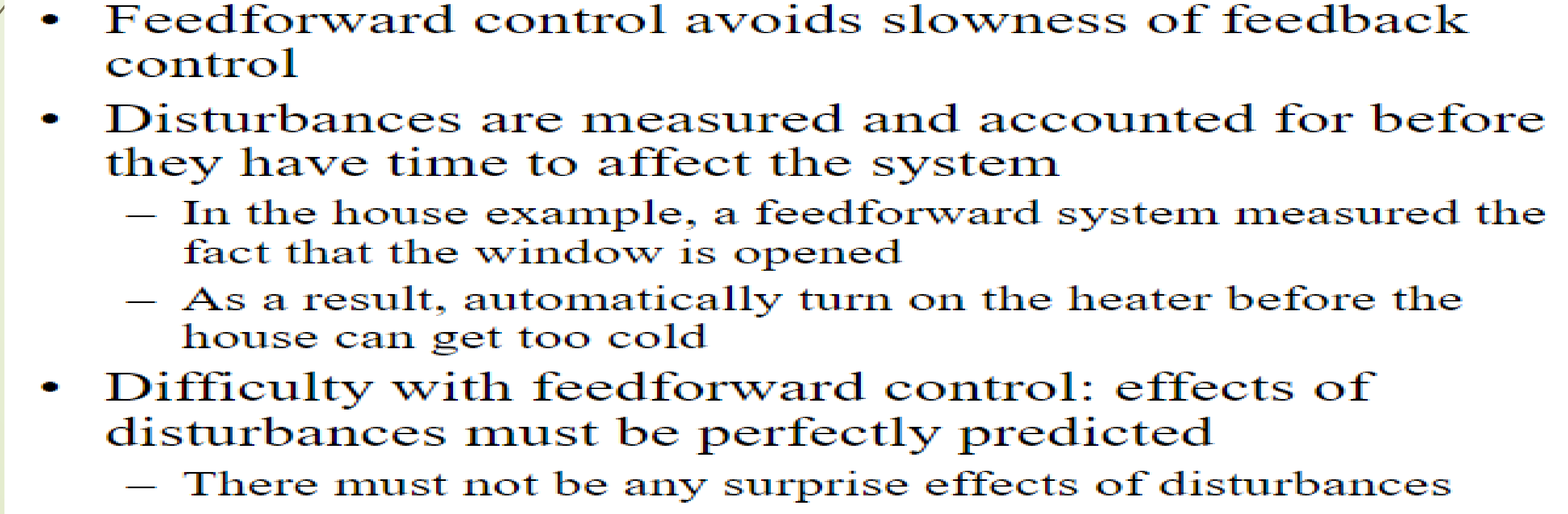
Proportional, Integral and Derivative Control

All design specifications can be reached.

Tips for Designing a PID Controller

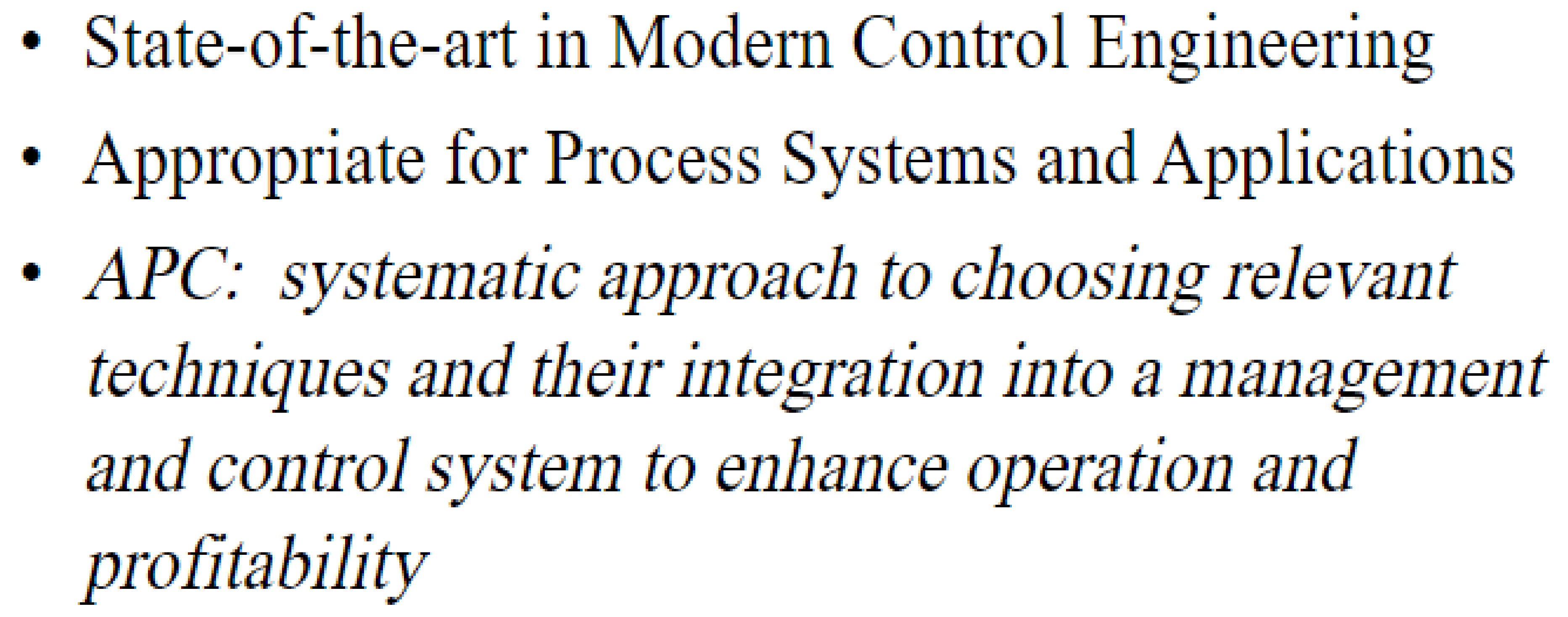
1. Obtain an open-loop response and determine what needs to be improved
2. Add a proportional control to improve the rise time
3. Add a derivative control to improve the overshoot
4. Add an integral control to eliminate the steady-state error
5. Adjust each of Kp, Ki, and Kd until you obtain a desired overall response.

Feed Forward Control:

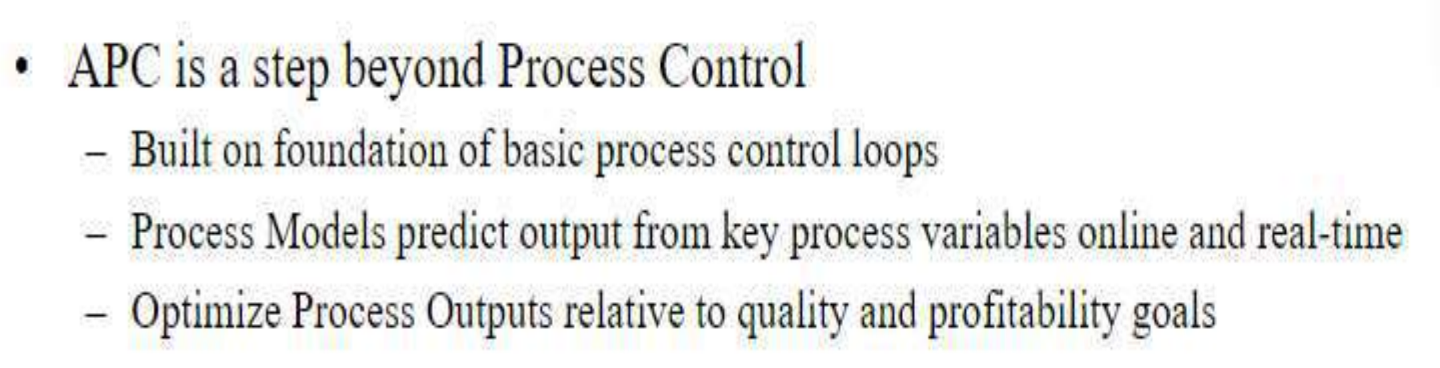


- Feedforward control avoids slowness of feedback control
- Disturbances are measured and accounted for before they have time to affect the system
  - In the house example, a feedforward system measured the fact that the window is opened
  - As a result, automatically turn on the heater before the house can get too cold
- Difficulty with feedforward control: effects of disturbances must be perfectly predicted
  - There must not be any surprise effects of disturbances

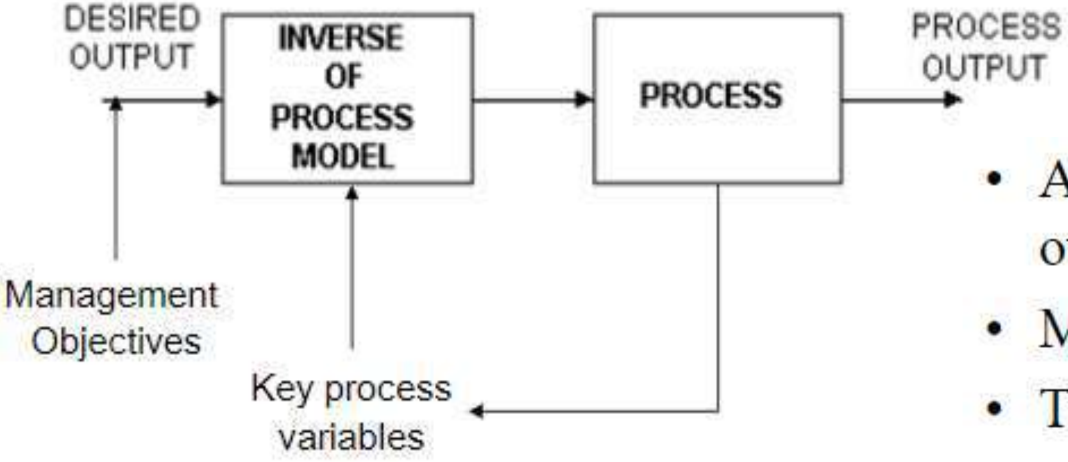
Advanced Process Control:



- State-of-the-art in Modern Control Engineering
- Appropriate for Process Systems and Applications
- *APC: systematic approach to choosing relevant techniques and their integration into a management and control system to enhance operation and profitability*



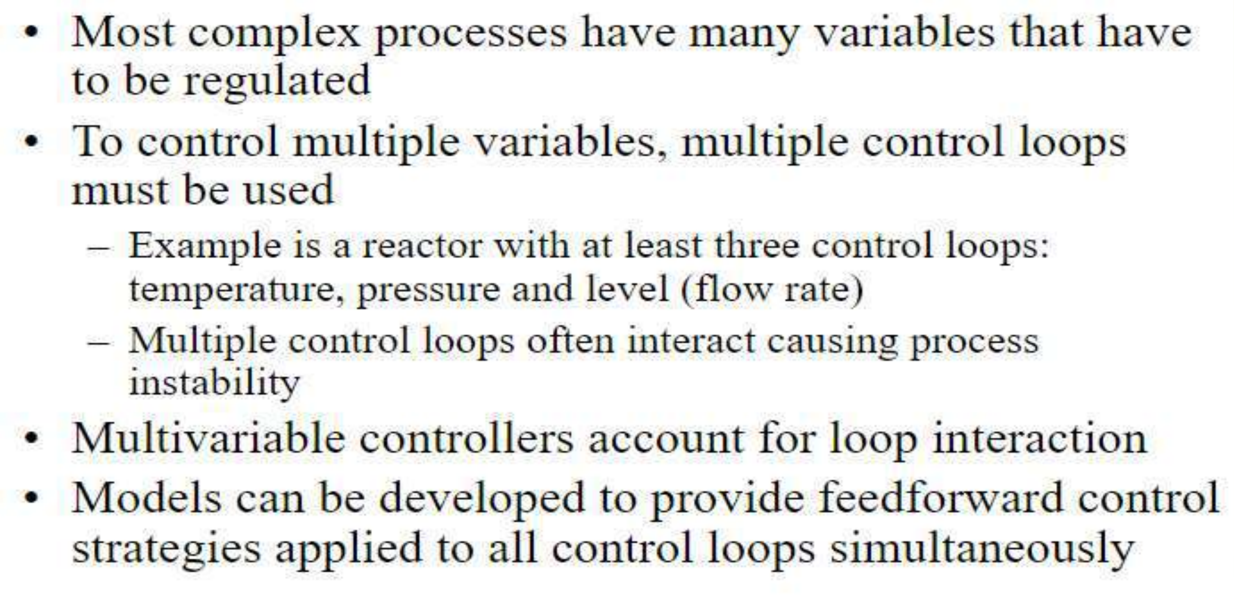
- APC is a step beyond Process Control
  - Built on foundation of basic process control loops
  - Process Models predict output from key process variables online and real-time
  - Optimize Process Outputs relative to quality and profitability goals



How Can APC Be Used?

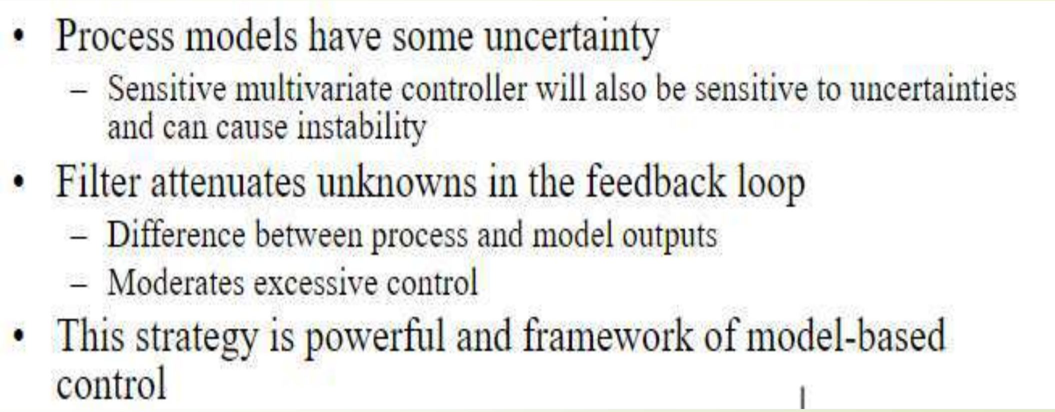
- APC can be applied to any system or process where outputs can be optimized on-line and in real-time
- Model of process or system exist or can be developed
- Typical applications:
  - Petrochemical plants and processes
  - Semiconductor wafer manufacturing processes
  - Also applicable to a wide variety of other systems including aerospace, robotics, radar tracking, vehicle guidance systems, etc.

Multi- Variable Control:

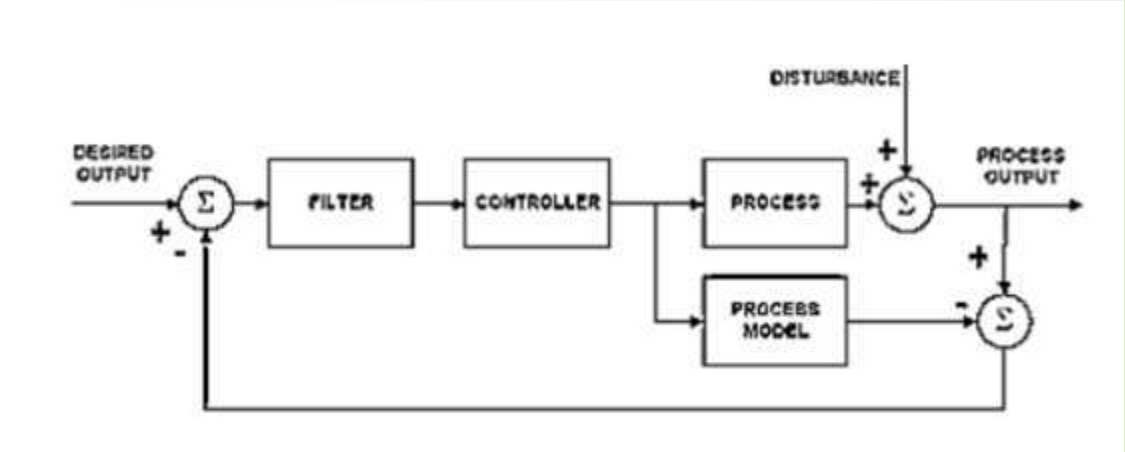


- Most complex processes have many variables that have to be regulated
- To control multiple variables, multiple control loops must be used
  - Example is a reactor with at least three control loops: temperature, pressure and level (flow rate)
  - Multiple control loops often interact causing process instability
- Multivariable controllers account for loop interaction
- Models can be developed to provide feedforward control strategies applied to all control loops simultaneously

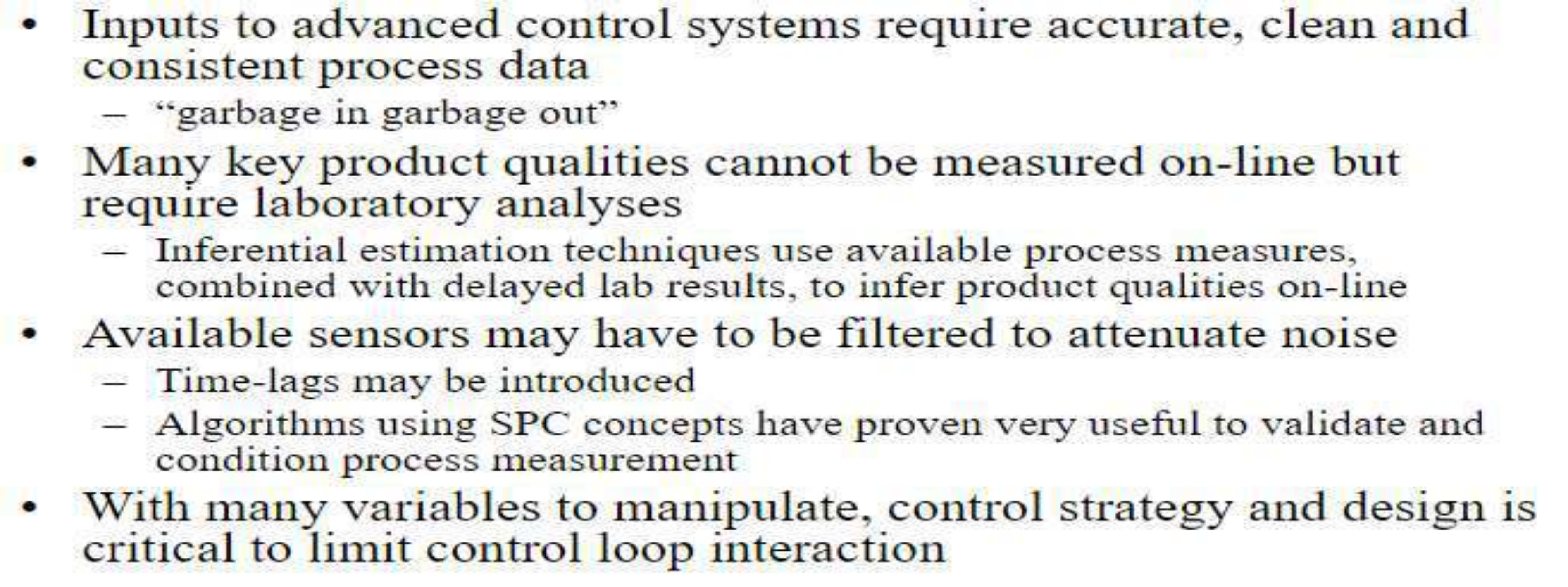
Internal Model Based Control:



- Process models have some uncertainty
  - Sensitive multivariate controller will also be sensitive to uncertainties and can cause instability
- Filter attenuates unknowns in the feedback loop
  - Difference between process and model outputs
  - Moderates excessive control
- This strategy is powerful and framework of model-based control

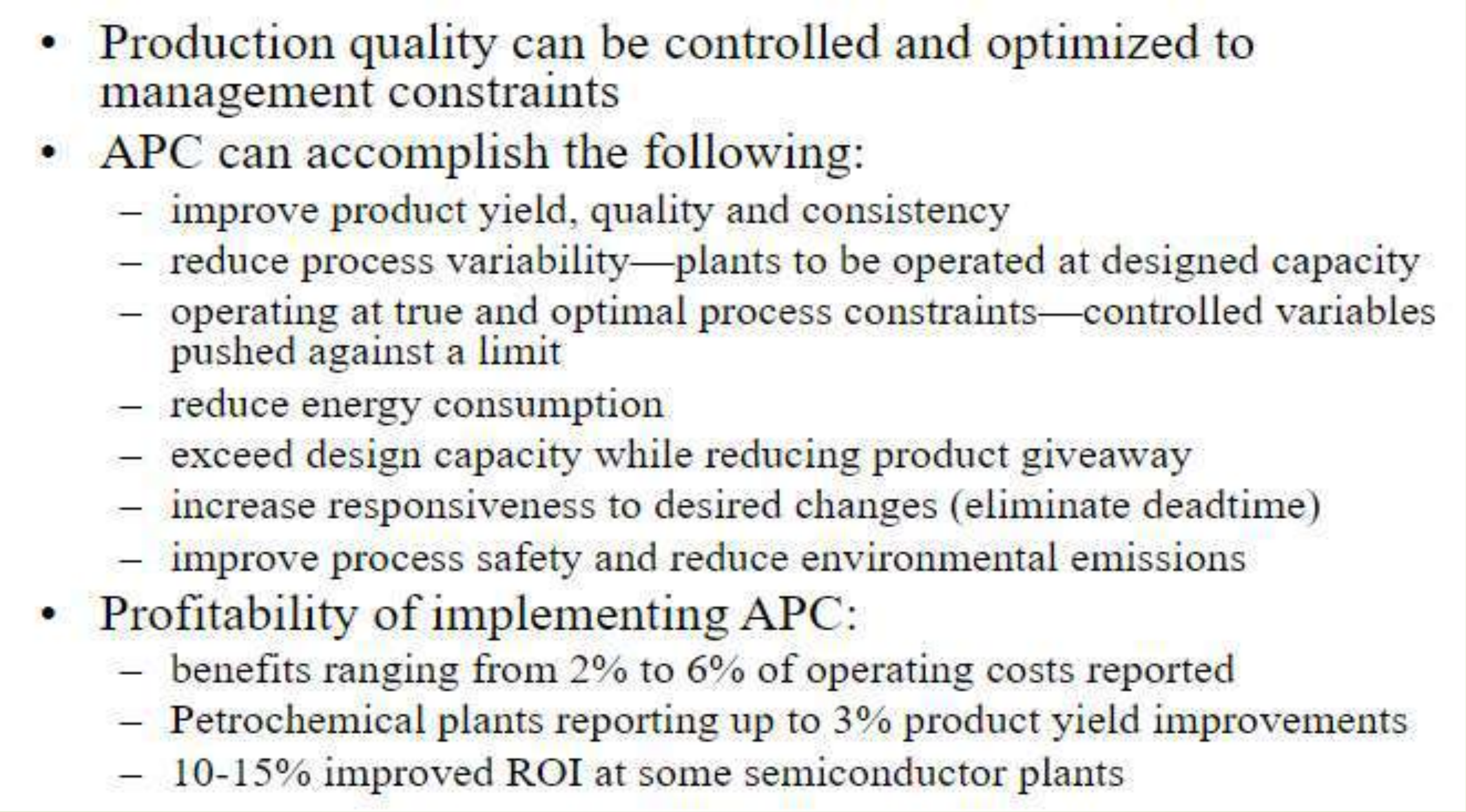


Important Data Issues:



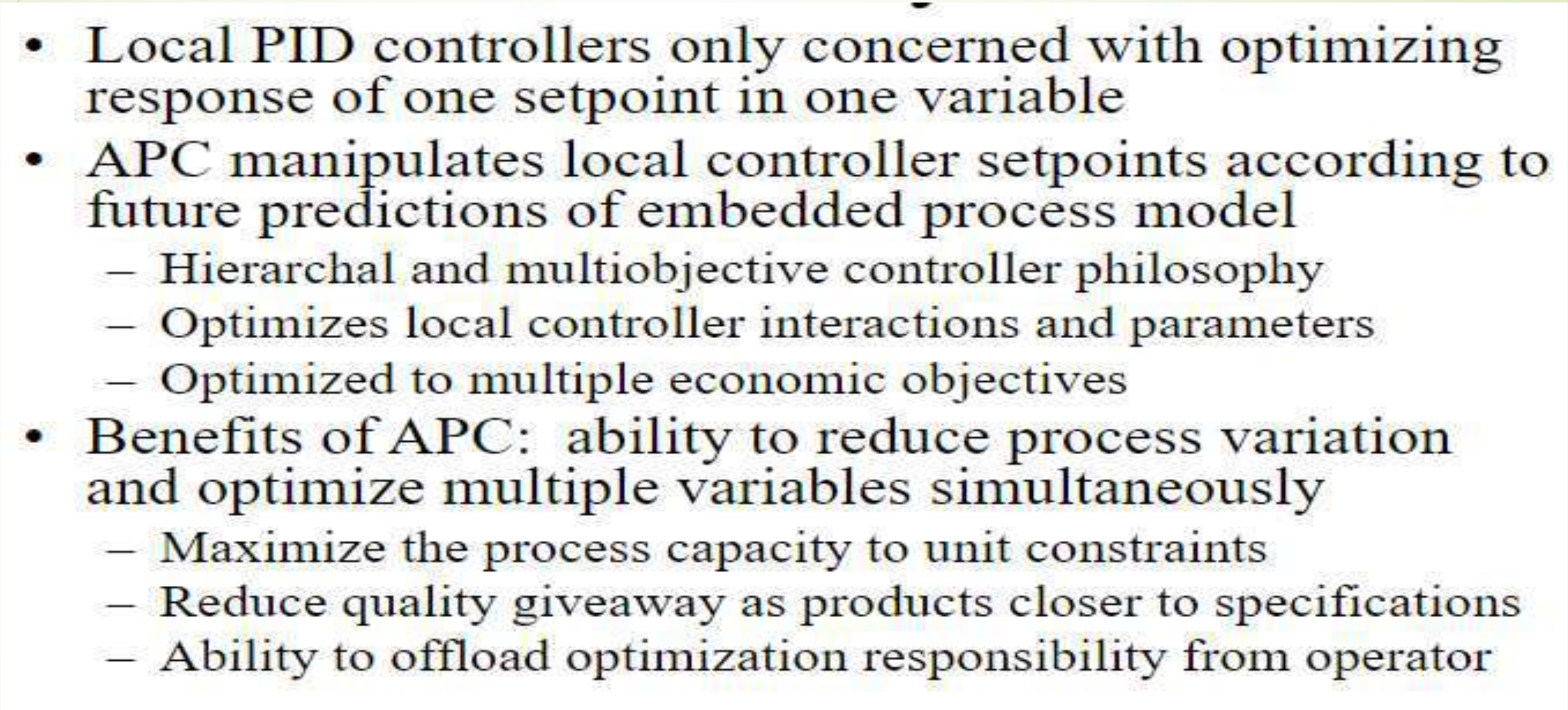
- Inputs to advanced control systems require accurate, clean and consistent process data
  - "garbage in garbage out"
- Many key product qualities cannot be measured on-line but require laboratory analyses
  - Inferential estimation techniques use available process measures, combined with delayed lab results, to infer product qualities on-line
- Available sensors may have to be filtered to attenuate noise
  - Time-lags may be introduced
  - Algorithms using SPC concepts have proven very useful to validate and condition process measurement
- With many variables to manipulate, control strategy and design is critical to limit control loop interaction

Advantages and Disadvantages:



- Production quality can be controlled and optimized to management constraints
- APC can accomplish the following:
  - improve product yield, quality and consistency
  - reduce process variability—plants to be operated at designed capacity
  - operating at true and optimal process constraints—controlled variables pushed against a limit
  - reduce energy consumption
  - exceed design capacity while reducing product giveaway
  - increase responsiveness to desired changes (eliminate deadtime)
  - improve process safety and reduce environmental emissions
- Profitability of implementing APC:
  - benefits ranging from 2% to 6% of operating costs reported
  - Petrochemical plants reporting up to 3% product yield improvements
  - 10-15% improved ROI at some semiconductor plants

Summary:



- Local PID controllers only concerned with optimizing response of one setpoint in one variable
- APC manipulates local controller setpoints according to future predictions of embedded process model
  - Hierarchal and multiobjective controller philosophy
  - Optimizes local controller interactions and parameters
  - Optimized to multiple economic objectives
- Benefits of APC: ability to reduce process variation and optimize multiple variables simultaneously
  - Maximize the process capacity to unit constraints
  - Reduce quality giveaway as products closer to specifications
  - Ability to offload optimization responsibility from operator

Thank You

Programmable Logic Controllers(PLC)

Presented By B Koti Reddy

Department Of Atomic Energy

Heavy Water Board

Contents :

What is PLC?

- A programmable logic controller (PLC) is a special form of microprocessor-based controller that uses programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting, and arithmetic in order to control machines and processes.

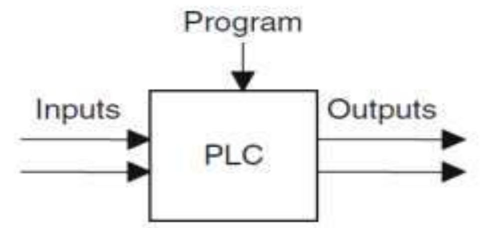


Fig 1. A Programmable Logic Controller

History of PLC:

- PLC was introduced in late 1960's.
- First commercial and Successful PLC was designed and developed by MODICON as a relay Replacer for General Motors.
- Earlier, It was a machine with thousands of Electronic Parts.
- Later, In Late 1970's The Microprocessor became reality and Greatly Enhanced the Role of PLC permitting to Evolve simple Relay to Sophisticated system as it is Today.

Major Parts of PLC

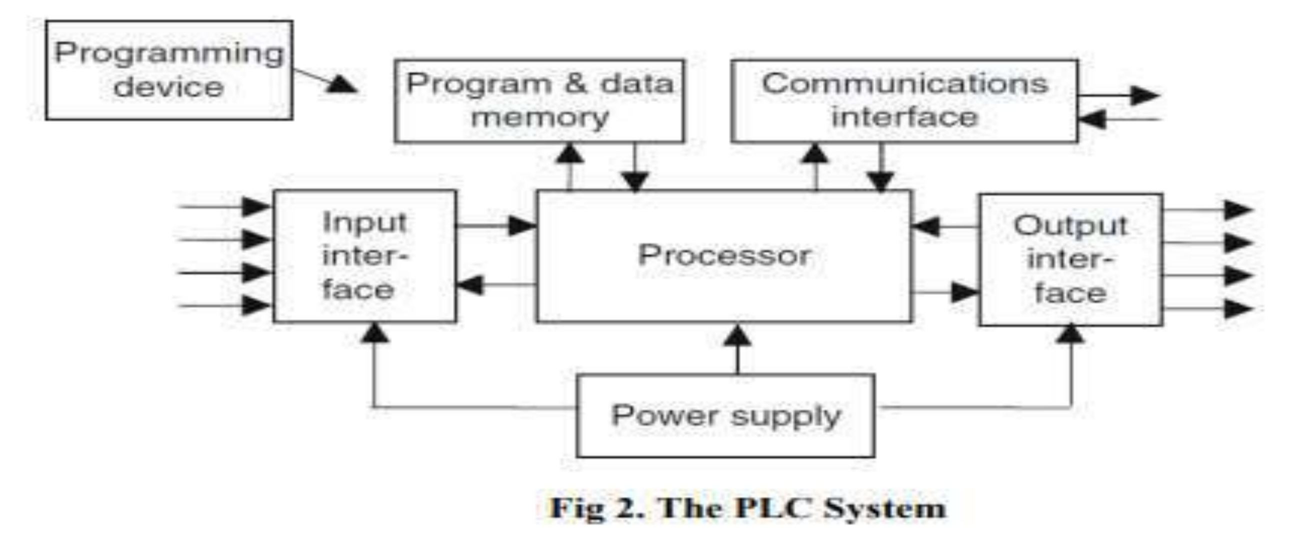


Fig 2. The PLC System

Typically a PLC system has the basic functional components of processor unit, memory, power supply unit, input/output interface section, communications interface, and the programming device in the basic arrangement.

The processor unit or central processing unit (CPU) is the unit containing the microprocessor. This unit interprets the input signals and carries out the control actions according to the program stored in its memory, communicating the decisions as action signals to the outputs.

The power supply unit is needed to convert the mains AC voltage to the low DC voltage (5V) necessary for the processor and the circuits in the input and output interface modules.

- The programming device is used to enter the required program into the memory of the processor. The program is developed in the device and then transferred to the memory unit of the PLC.
- The memory unit is where the program containing the control actions to be exercised by the microprocessor is stored and where the data is stored from the input for processing and for the output.
- The input and output sections are where the processor receives information from external devices and communicates information to external devices.
- The inputs might thus be from switches, sensors such as photoelectric cells, temperature sensors, flow sensors, or the like.
- The outputs might be to motor starter coils, solenoid valves, or similar things.
- The input/output channels provide isolation and signal conditioning functions so that sensors and actuators can often be directly connected to them without the need for other circuitry.
- Electrical isolation from the external world is usually by means of opto-isolators (the term opto-coupler is also often used). Figure shows the principle of an opto-isolator.

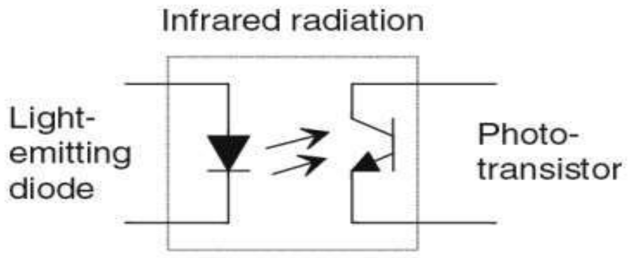


Fig 6. An Opto-isolator

Input And Output Devices: Input Devices

- Mechanical Switches:
- A mechanical switch generates an on/off signal or signals as a result of some mechanical input causing the switch to open or close.

Mechanical switch

- Push Button:
- A push button is a momentary or non-latching switch which causes a temporary change in the state of an electrical circuit only while the switch is physically actuated.
- An automatic mechanism (i.e. a spring) returns the switch to its default position immediately afterwards, restoring the initial circuit condition.

Push button

Selector switch:

- **A manually operated multi-position switch, which is usually adjusted by a knob or handle, and may have detents to hold in a given position. Used for instance, in devices or instruments with multiple functions, ranges, or modes of operation. Such a switch is usually rotary also called selector.**

Selector switch

- Contactor:
- A contactor is an electrically controlled switch used for switching an electrical power circuit, similar to a relay except with higher current ratings.
- A contactor is controlled by a circuit which has a much lower power level than the switched circuit.

Contactor

- Illuminated push-button:
- A push-button or simply button is a simple switch mechanism for controlling some aspect of a machine or a process.

Illuminated push-button

- Photoelectric Sensors and Switches: A photoelectric sensor, or photo eye, is an equipment used to discover the distance, absence, or presence of an object by using a light transmitter, often infrared, and a photoelectric receiver. They are largely used in industrial manufacturing

Photoelectric sensor

Encoders :

- The term encoder is used for a device that provides a digital output as a result of angular or linear displacement.

Encoder

Position sensor:

A position sensor is any device that permits position measurement.

It can either be an absolute position sensor or a relative one (displacement sensor).

Position sensors can be linear, angular, or multi-axis.

Position sensor

Temperature Sensors :

- A simple form of temperature sensor that can be used to provide an on/off signal when a particular temperature is reached is the bimetal element.

Temperature sensor

Pressure Sensors :

Pressure sensors can be designed to give outputs that are proportional to the difference in pressure between two input ports.

Pressure sensor

Output Devices:

The output ports of a PLC are relay or opto-isolator with transistor or triac, depending on the devices that are to be switched on or off. Generally, the digital signal from an output channel of a PLC is used to control an actuator, which in turn controls some process.

- Relay: A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays.
- Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal.

Relay

- Contactor relays :
- Contactor relays are often used in control and regulating functions. They are used in large quantities for the indirect control of motors, valves, clutches and heating equipment.

Contactor

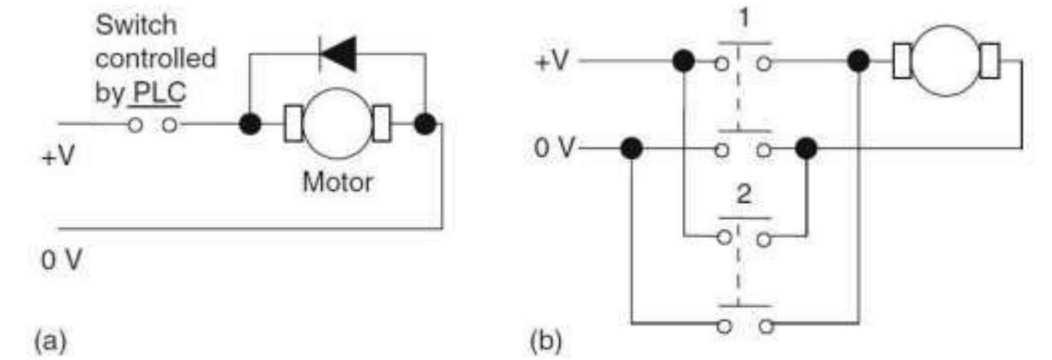
Directional control valve

Directional Control Valve :These valves are one of the most fundamental parts in hydraulic machinery as well as pneumatic machinery. They allow fluid flow into different paths from one or more sources.

- **Motors:** Directional control valve A DC motor has coils of wire mounted in slots on a cylinder of ferromagnetic material, which is termed the armature. The armature is mounted on bearings and is free to rotate. It is mounted in the magnetic field produced by permanent magnets or current passing through coils of wire, which are called the field coils. When a current passes through the armature coil, forces act on the coil and result in rotation.

DC Motor

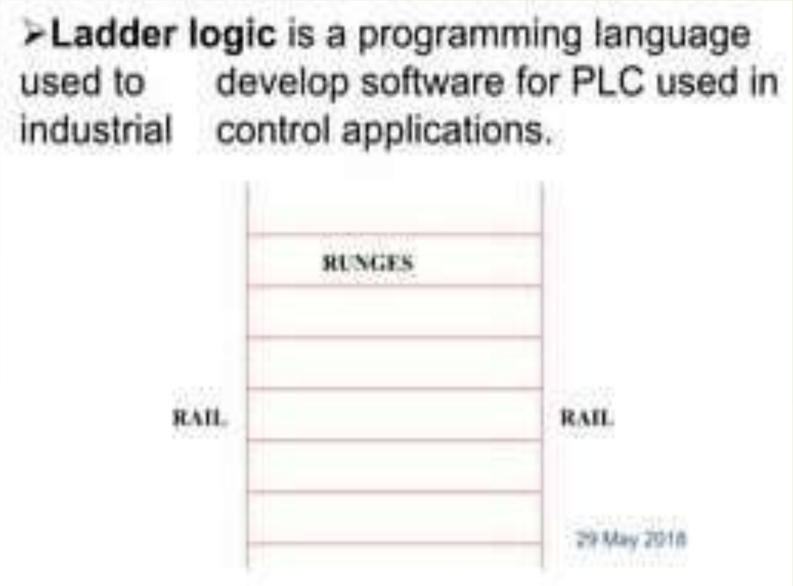
AC Motor



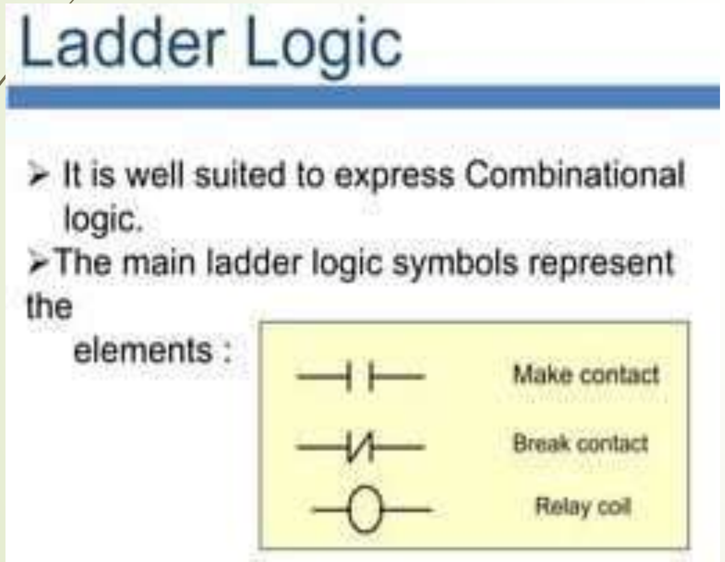
DC motor: (a) on/off control, and (b) directional control.

Programming Languages of PLC:

- Most common Languages used in PLC Programming are
- 1. Ladder Logic.
- 2. Functional Block Diagrams.
- 3. Sequential Function Chart.
- 4. Boolean Mnemonics.



➢**Ladder logic** is a programming language used to develop software for PLC used in industrial control applications.

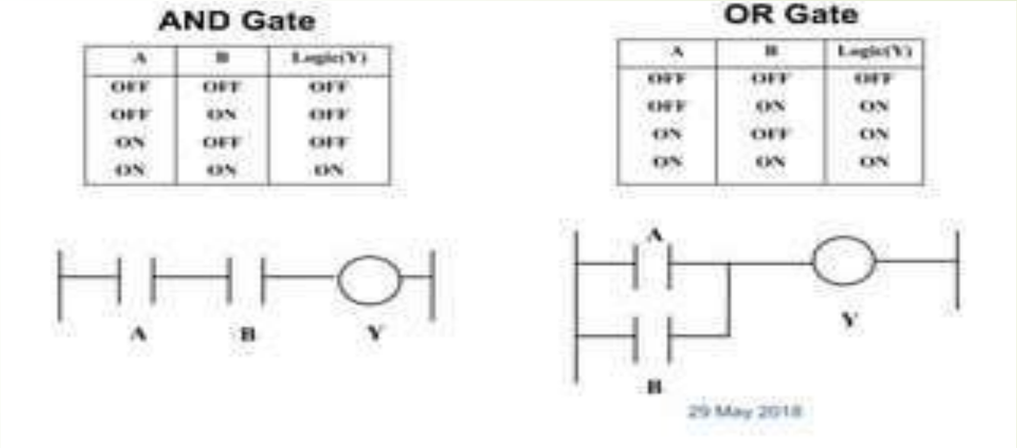


Ladder Logic

➢ It is well suited to express Combinational logic.

➢The main ladder logic symbols represent the elements :

Make contact
Break contact
Relay coil



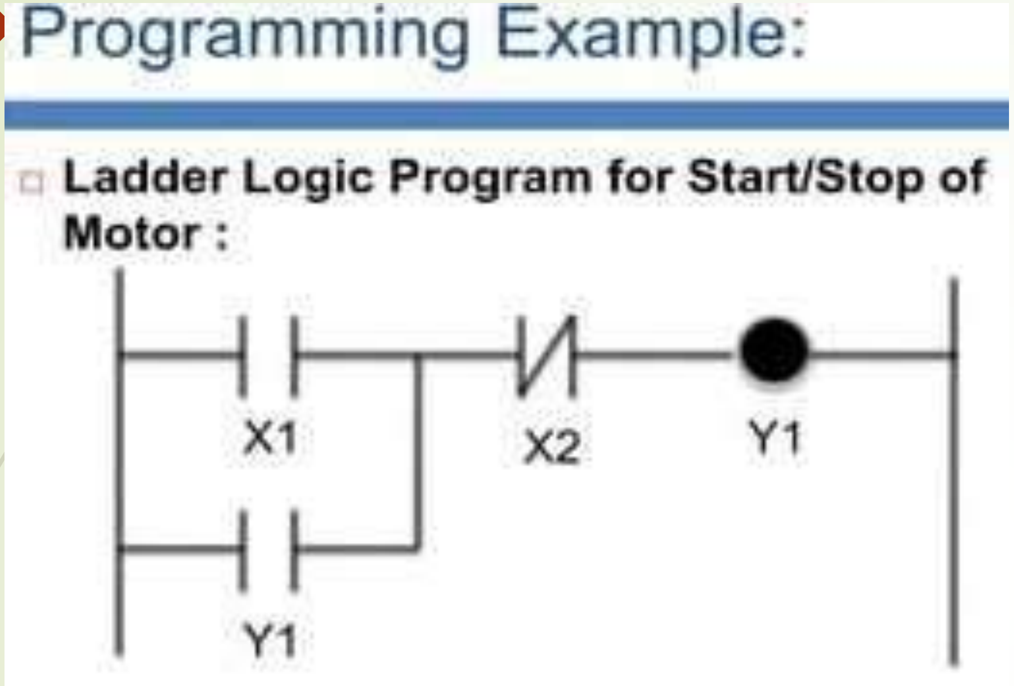
AND Gate

A	B	Logic(Y)
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

OR Gate

A	B	Logic(Y)
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

Ladder Diagram of AND & OR Gates



Programming Example:

☐ **Ladder Logic Program for Start/Stop of Motor :**

Applications:

PLCs are used in

1. Robots Manufacturing and control.
2. Car park control
3. Train Control Station System.
4. Food Processing.
5. Materials Handling.
6. Machine Tools.
7. Conveyors System.

Thank You

Root locus and frequency analysis of compensators

Presented By B Koti Reddy
Department Of Atomic Energy
Heavy Water Board

Lead or Phase-Lead Compensator Using Root Locus

A first-order lead compensator can be designed using the root locus. A lead compensator in root locus form is given by

$$G_c(s) = \frac{(s + z)}{(s + p)}$$

where the magnitude of z is less than the magnitude of p. A phase-lead compensator tends to shift the root locus toward the left half plane. This results in an improvement in the system's stability and an increase in the response speed.

When a lead compensator is added to a system, the value of this intersection will be a larger negative number than it was before. The net number of zeros and poles will be the same (one zero and one pole are added), but the added pole is a larger negative number than the added zero. Thus, the result of a lead compensator is that the asymptotes' intersection is moved further into the left half plane, and the entire root locus will be shifted to the left. This can increase the region of stability as well as the response speed.

Lead or Phase-Lead Compensator Using Root Locus

In Matlab a phase lead compensator in root locus form is implemented by using the transfer function in the form

```
numlead=kc*[1 z];
denlead=[1 p];
```

and using the `conv()` function to implement it with the numerator and denominator of the plant

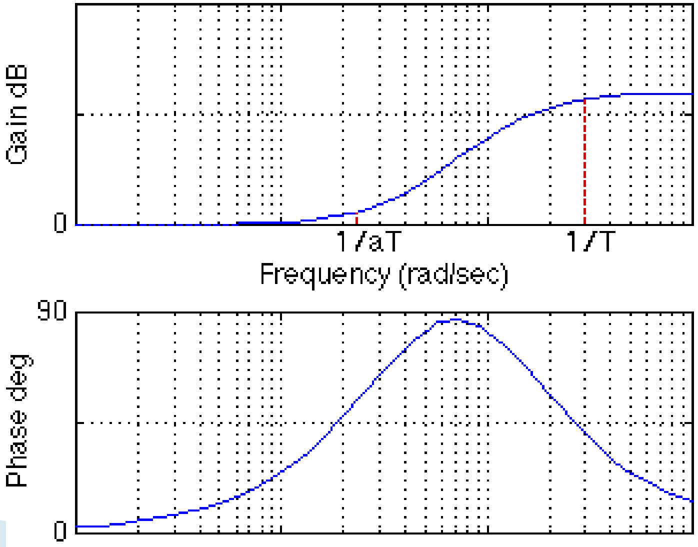
```
newnum=conv(num,numlead);
newden=conv(den,denlead);
```

Lead or Phase-Lead Compensator Using Frequency Response

A first-order phase-lead compensator can be designed using the frequency response. A lead compensator in frequency response form is given by

$$G_c(s) = \frac{(1 + \alpha\cdot\tau\cdot s)}{\alpha\cdot(1 + \tau\cdot s)} \qquad p = \frac{1}{\tau} \qquad z = \frac{1}{\alpha\tau} \qquad \omega_m = \sqrt{z\cdot p} \qquad \sin(\phi_m) = \frac{\alpha - 1}{\alpha + 1}$$

In frequency response design, the phase-lead compensator adds positive phase to the system over the frequency range. A bode plot of a phase-lead compensator looks like the following



Lead or Phase-Lead Compensator Using Frequency Response

Additional positive phase increases the phase margin and thus increases the stability of the system. This type of compensator is designed by determining **alfa** from the amount of phase needed to satisfy the phase margin requirements, and determining **tal** to place the added phase at the new gain-crossover frequency.

Another effect of the lead compensator can be seen in the magnitude plot. The lead compensator increases the gain of the system at high frequencies (the amount of this gain is equal to alfa. This can increase the crossover frequency, which will help to decrease the rise time and settling time of the system.

Lead or Phase-Lead Compensator Using Frequency Response

In Matlab, a phase lead compensator in frequency response form is implemented by using the transfer function in the form

```
numlead=[aT 1];
denlead=[T 1];
```

and using the `conv()` function to multiply it by the numerator and denominator of the plant

```
newnum=conv(num,numlead);
newden=conv(den,denlead);
```

Lag or Phase-Lag Compensator Using Root Locus

A first-order lag compensator can be designed using the root locus. A lag compensator in root locus form is given by

$$G_c(s) = \frac{(s + z)}{(s + p)}$$

where the magnitude of z is greater than the magnitude of p. A phase-lag compensator tends to shift the root locus to the right, which is undesirable. For this reason, the pole and zero of a lag compensator must be placed close together (usually near the origin) so they do not appreciably change the transient response or stability characteristics of the system.

When a lag compensator is added to a system, the value of this intersection will be a smaller negative number than it was before. The net number of zeros and poles will be the same (one zero and one pole are added), but the added pole is a smaller negative number than the added zero. Thus, the result of a lag compensator is that the asymptotes' intersection is moved closer to the right half plane, and the entire root locus will be shifted to the right.

Lag or Phase-Lag Compensator Using Root Locus

It was previously stated that that lag controller should only minimally change the transient response because of its negative effect. If the phase-lag compensator is not supposed to change the transient response noticeably, what is it good for? The answer is that a phase-lag compensator can improve the system's steady-state response. It works in the following manner. At high frequencies, the lag controller will have unity gain. At low frequencies, the gain will be z0/p0 which is greater than 1. This factor z/p will multiply the position, velocity, or acceleration constant (Kp, Kv, or Ka), and the steady-state error will thus decrease by the factor z0/p0. In Matlab, a phase lead compensator in root locus form is implemented by using the transfer function in the form

```
numlag=[1 z];
denlag=[1 p];
```

and using the `conv()` function to implement it with the numerator and denominator of the plant

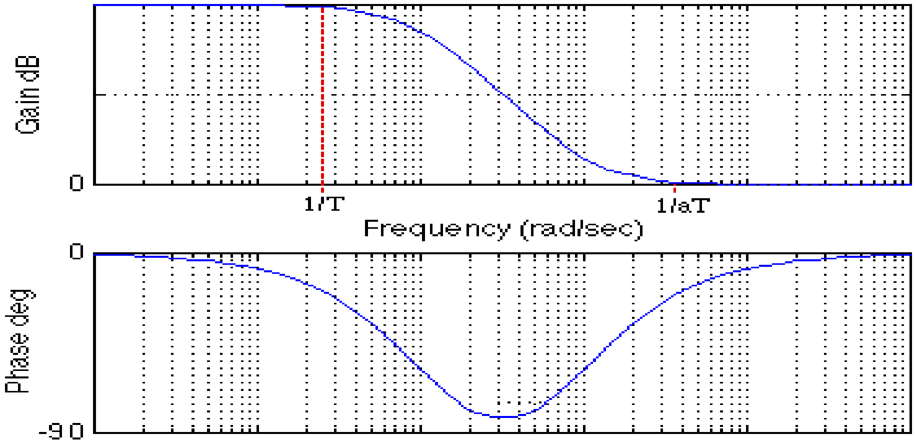
```
newnum=conv(num,numlag);
newden=conv(den,denlag);
```

Lag or Phase-Lag Compensator using Frequency Response

A first-order phase-lag compensator can be designed using the frequency response. A lag compensator in frequency response form is given by

$$G_c(s) = \frac{\left(1 + \alpha\cdot\tau\cdot s\right)}{\alpha\cdot\left(1 + \tau\cdot s\right)}$$

The phase-lag compensator looks similar to a phase-lead compensator, except that a is now less than 1. The main difference is that the lag compensator adds negative phase to the system over the specified frequency range, while a lead compensator adds positive phase over the specified frequency. A bode plot of a phase-lag compensator looks like the following



Lag or Phase-Lag Compensator using Frequency Response

In Matlab, a phase-lag compensator in frequency response form is implemented by using the transfer function in the form

```
numlead=[a*T 1];
denlead=a*[T 1];
```

and using the `conv()` function to implement it with the numerator and denominator of the plant

```
newnum=conv(num,numlead);
newden=conv(den,denlead);
```

Lead-lag Compensator using either Root Locus or Frequency Response

A lead-lag compensator combines the effects of a lead compensator with those of a lag compensator. The result is a system with improved transient response, stability and steady-state error. To implement a lead-lag compensator, first design the lead compensator to achieve the desired transient response and stability, and then add on a lag compensator to improve the steady-state response

Exercise - Dominant Pole-Zero Approximations and Compensations

The influence of a particular pole (or pair of complex poles) on the response is mainly determined by two factors: the real part of the pole and the relative magnitude of the residue at the pole. The real part determines the rate at which the transient term due to the pole decays; the larger the real part, the faster the decay. The relative magnitude of the residue determines the percentage of the total response due to a particular pole.

Investigate (using Simulink) the impact of a closed-loop negative real pole on the overshoot of a system having complex poles.

$$T(s) = \frac{pr \cdot \omega n^2}{(s + pr) \cdot \left[s^2 + (2 \cdot \zeta \cdot \omega n \cdot s) + \omega n^2\right]}$$

Make pr to vary (2, 3, 5) times the real part of the complex pole for different values of ζ (0.3, 0.5, 0.7).

Investigate (using Simulink) the impact of a closed-loop negative real zero on the overshoot of a system having complex poles.

$$T(s) = \frac{(s + zr)}{\left[s^2 + (2 \cdot \zeta \cdot \omega n \cdot s) + \omega n^2\right]}$$

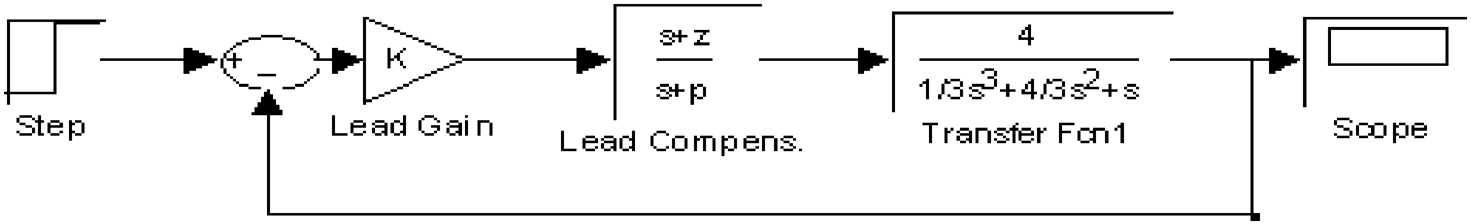
Make zr to vary (2, 3, 5) times the real part of the complex pole for different values of ζ (0.3, 0.5, 0.7).

Exercise - Lead and Lag Compensation

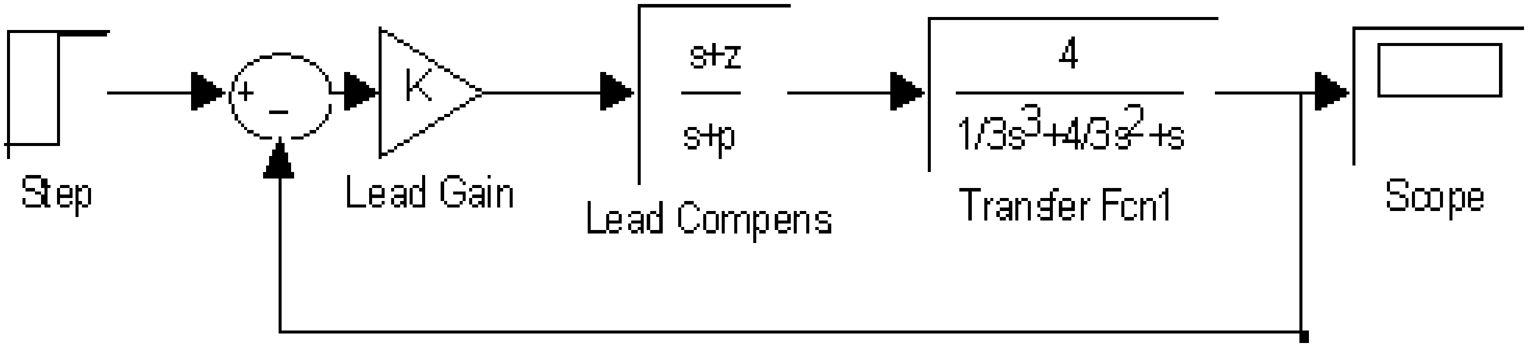
Investigate (using Matlab and Simulink) the effect of lead and lag compensations on the two systems indicated below. Summarize your observations. Plot the root-locus, bode diagram and output for a step input before and after the compensations.

Remember
lead compensation: z<p (place zero below the desired root location or to the left of the first two real poles)
lag compensation: z>p (locate the pole and zero near the origin of the s-plane)

Lead Compensation (use z=1.33, p=20 and K =15).



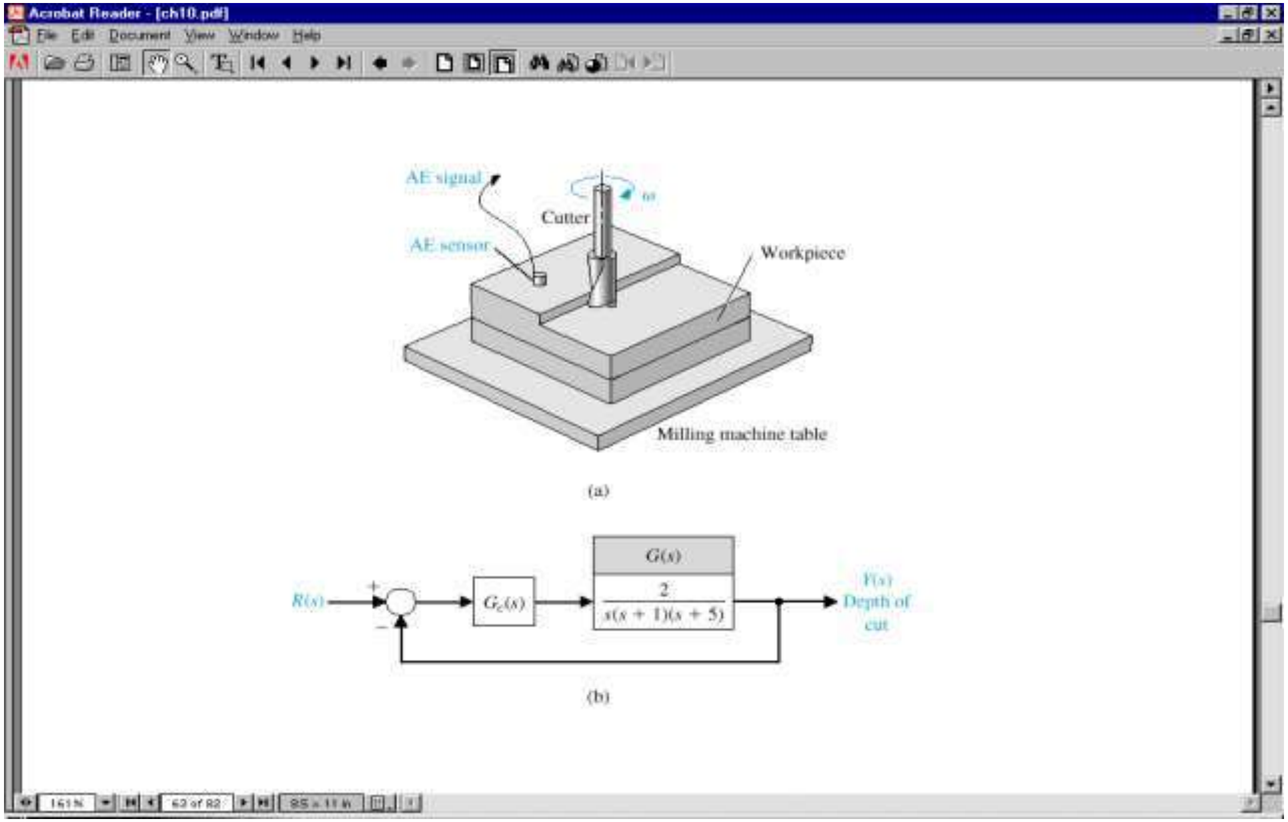
Lag Compensation (use z=0.09 , and p=0.015, K=1/6)

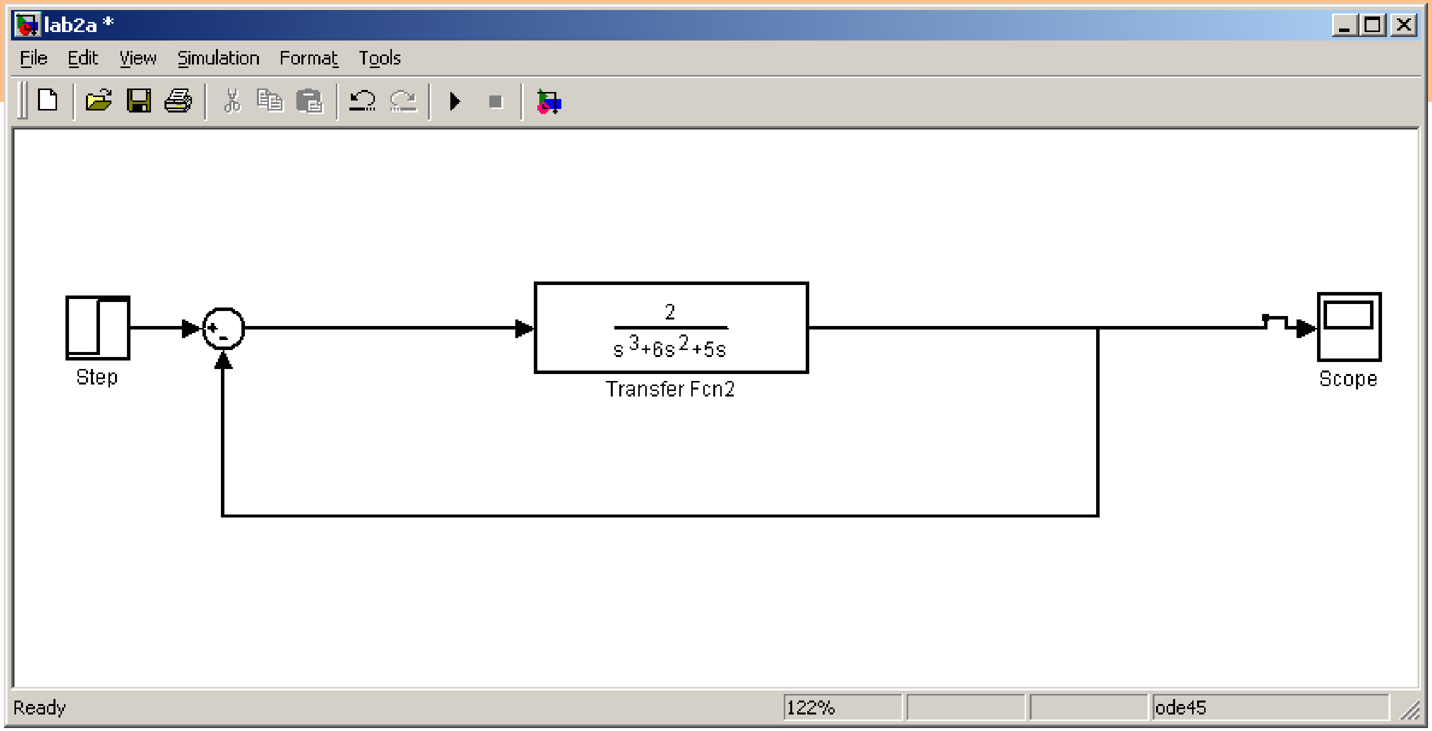


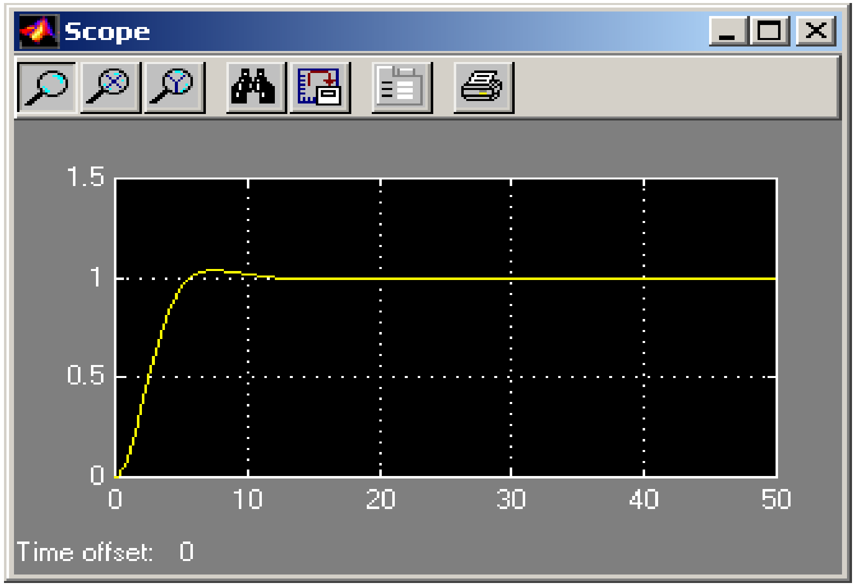
Summarize your findings

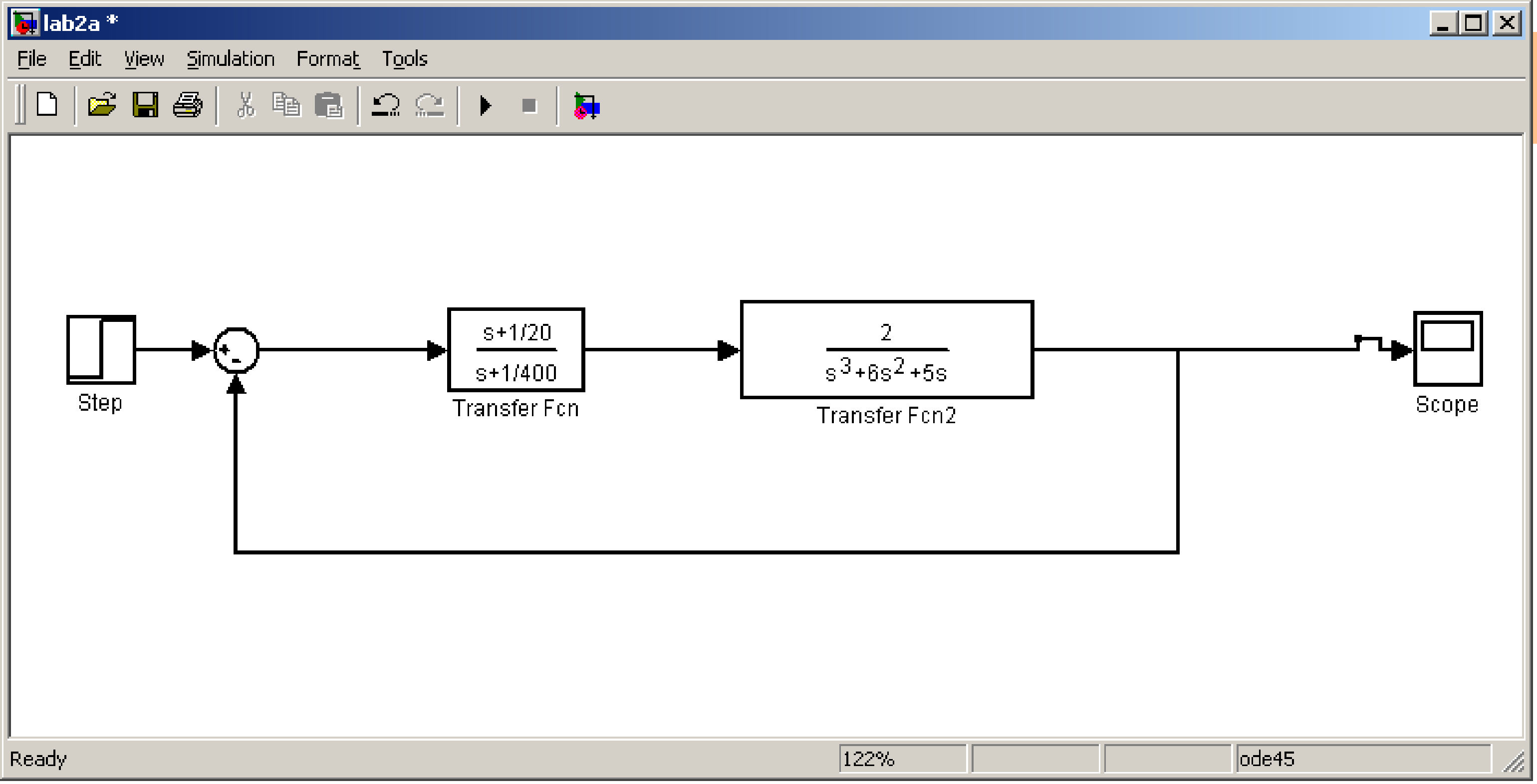
Problem 10.36

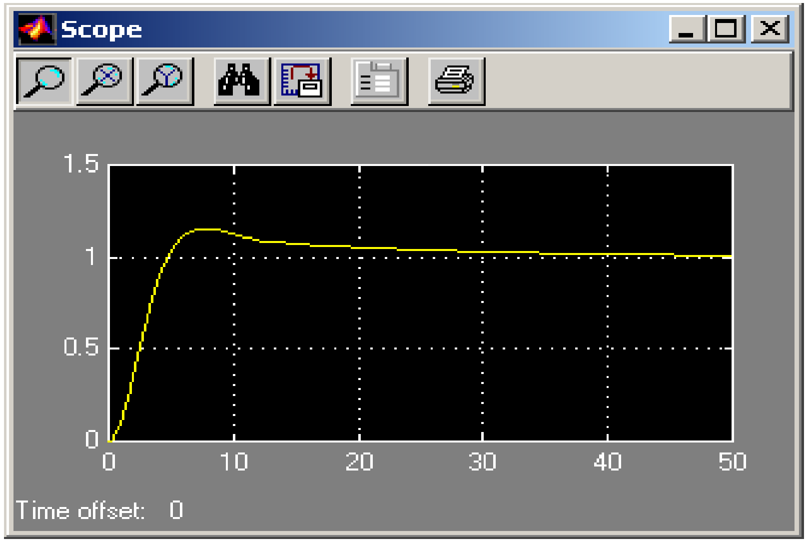
Determine a compensator so that the percent overshoot is less than 20% and Kv (velocity constant) is greater than 8.




lab2a *
File Edit View Simulation Format Tools
Step
+
-
2
s 3+6s 2+5s
Transfer Fcn2
Scope
Ready
122%
ode45


Scope
1.5
1
0.5
0
0
10
20
30
40
50
Time offset: 0


lab2a *
File Edit View Simulation Format Tools
Step
s+1/20
s+1/400
Transfer Fcn
2
s3+6s2+5s
Transfer Fcn2
Scope
Ready
122%
ode45


Scope
1.5
1
0.5
0
0
10
20
30
40
50
Time offset: 0

ROOT LOCUS DESIGN OF PID CONTROOLER:

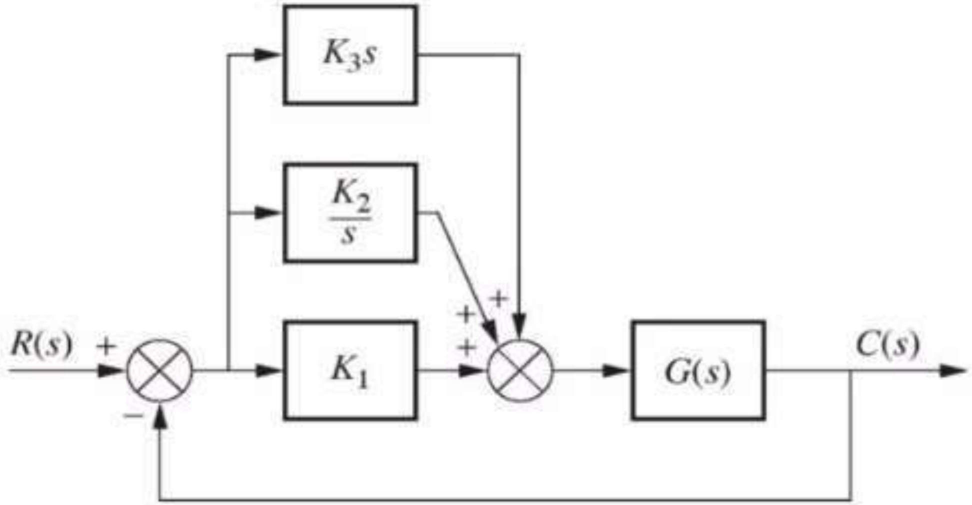
Presented By B Koti Reddy
Department Of Atomic Energy
Heavy Water Board

Contents:

- PID Controller.
- Design Steps.
- Example.

Design - Root Locus – PID Controller

- PID Controller : Compensation with two Zeros with one pole at origin. One Zero can be first designed as the derivative compensator then other Zero and one pole at the origin can be designed as ideal integrator.



$$G_c(s) = \frac{K_3(s^2+\frac{K_1}{K_3}s+\frac{K_2}{K_3})}{s}$$

Design Steps:

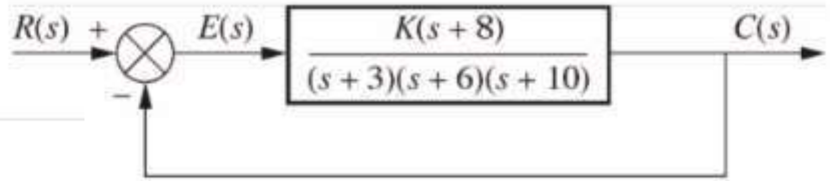
- Evaluate Performance of uncompensated system to determine how much improvement in Transient Response is required.
- Design PD controller, include Zero location and loop Gain.
- Simulate the system shows the requirements have been met.
- Redesign if the simulation shows that requirements have not been met.
- Design PI controller to yield the required steady state Error.
- Determine the Gains K1, K2, and K3.
- Simulate the system shows the requirements have been met.
- Redesign if the simulation shows that requirements have not been met.

Design - Root Locus – PID Controller

Example:

- Design a PID Controller so that the system can operate with a peak time that is two thirds of uncompensated system at 20% overshoot with zero Steady State Error.

Sol: Step1: Evaluate Performance of uncompensated system to determine how much improvement in Transient Response is required.



- Find equivalent damping ration line.

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \qquad \theta = 180^o - cos^{-1}\zeta$$

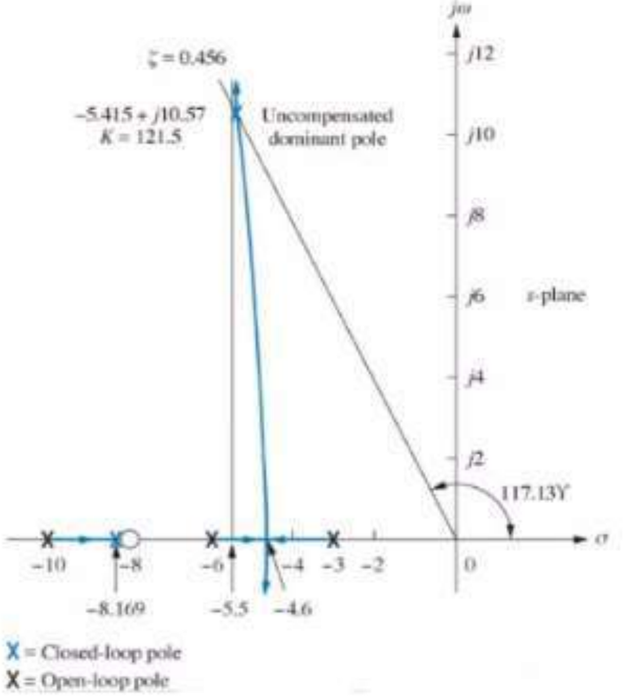
- Find the dominant pole location:

$$\angle KG(s)H(s) = (2k+1)180^o$$

$$K = \frac{1}{|G(s)||H(s}$$

- Find uncompensated peak time.

$$T_p = \frac{\pi}{\omega_d}$$



Step2 : Design PD Controller, Include Zero Location and Loop Gain.

- Find Compensated Peak Time $T_{p_c} = \frac{2}{3} T_{p_u}$

- Find Compensated Dominant Pole :

$$\omega_c = \frac{\pi}{T_{p_c}} \qquad \sigma_c = \frac{\omega_c}{tan(180^o - 117.13^o)}$$

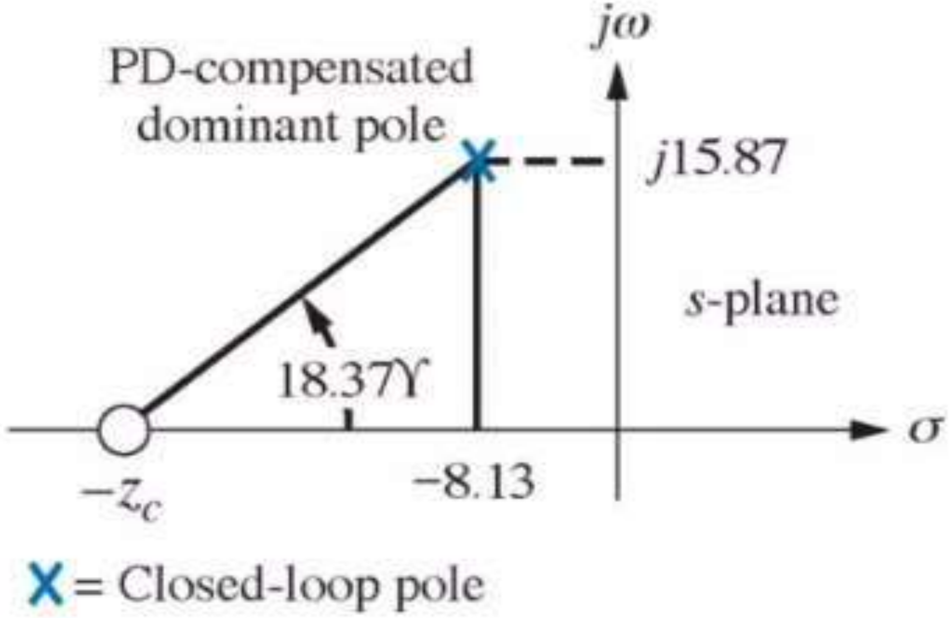
- Get PD Location and Angle

$$\sum \theta_{uc_{zero}} - \sum \theta_{uc_{pole}} - \theta_{PD_{pole}} = (2k+1)180^o$$

- Get PD zero location – Real Axis

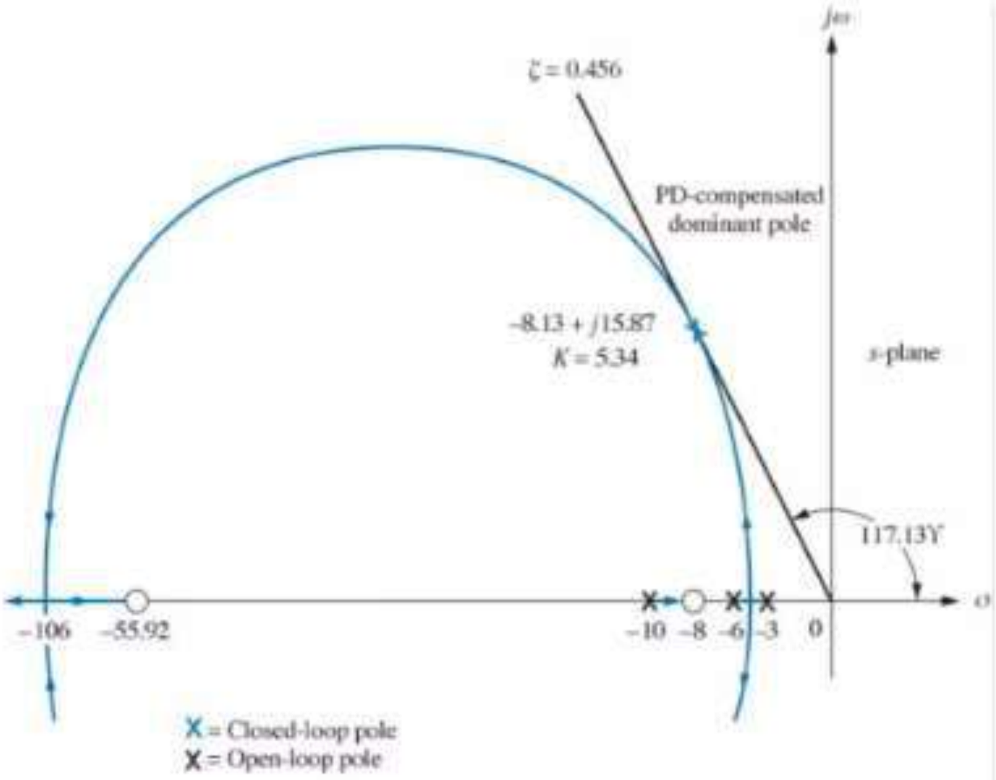
$$\frac{15.87}{z_c - 8.13} = tan\ 18.37^o$$

$$z_c = 55.92$$



- Get Resulting Gain:

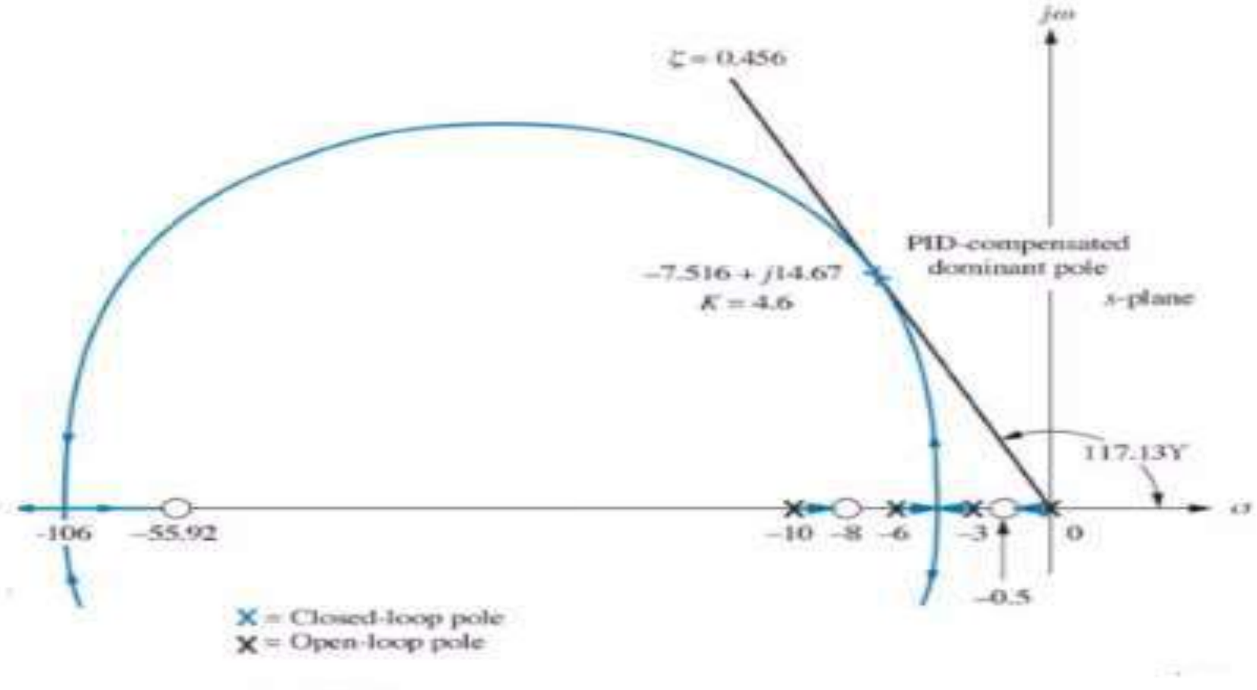
$$K = \frac{1}{|G(s)||H(s)|}$$
$$= 5.34$$



- Step-3,4 Validation:
- Step-5: Design PI controller to yield the required steady state Error. Any ideal integral compensator will work as long as the zero is placed close to the origin.

$$G_{PI}(s) = \frac{s+0.5}{s}$$

- Get Dominant pole location of Damping ratio line, and get the Gain.

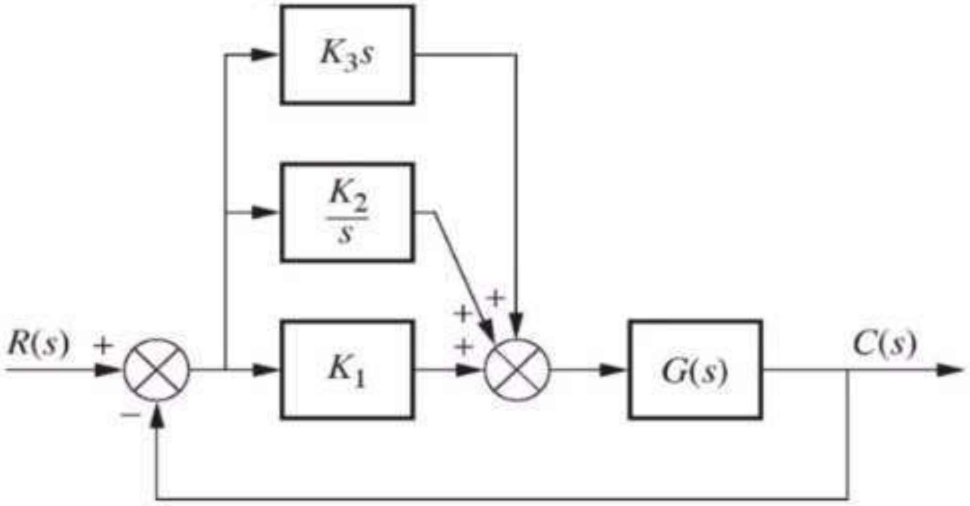


Step-6:

- Determine the Gains K1, K2, and K3:

$$
\begin{aligned}
G_{PID}(s) &= \frac{K_3(s^2 + \frac{K_1}{K_3}s + \frac{K_2}{K_3})}{s} \\
&= \frac{K(s+55.92)(s+0.5)}{s} \\
&= \frac{4.6(s^2+56.42s+27.96)}{s}
\end{aligned}
$$

$K_1 = 259.5,\ K_2 = 128.6,\ K_3 = 4.6$



Thank You

Sliding Mode control and Its Applications

PRESENTED BY B KOTI REDDY
DEPARTMENT OF ATOMIC ENERGY
HEAVY WATER BOARD

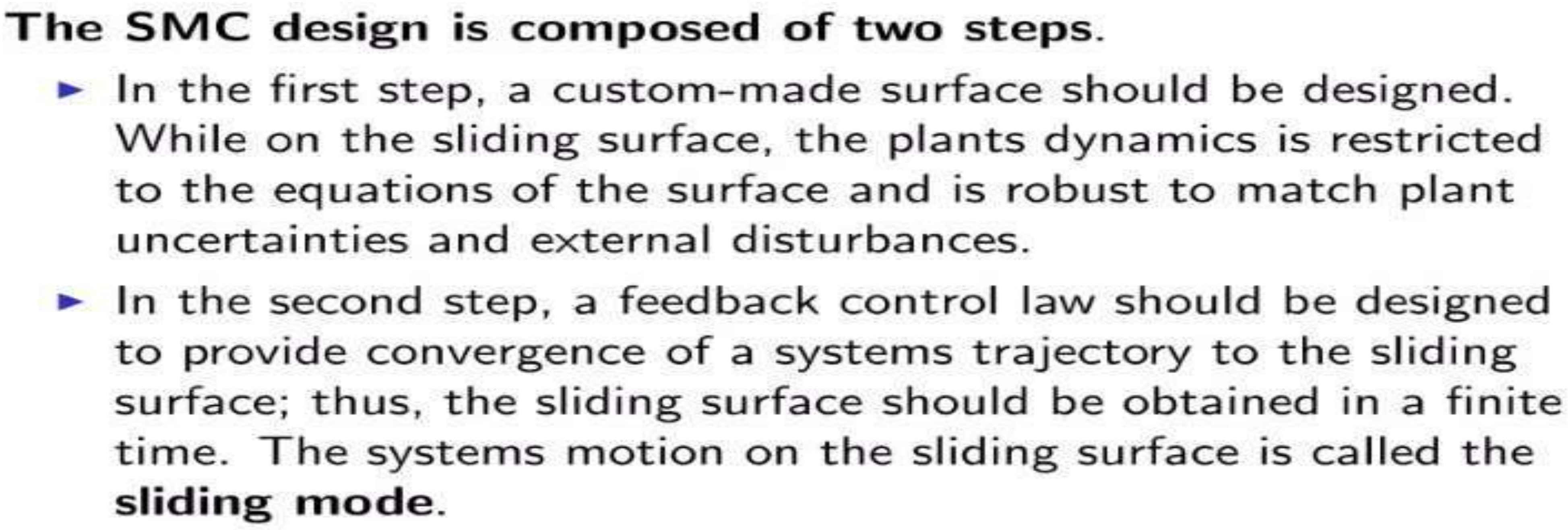
Contents:

Introduction :

- An ideal sliding mode exists only when the system state satisfies the dynamic equation that governs the sliding Mode for all time. This requires an infinite switching in general to ensure the sliding motion.
- The sliding mode approach is recognized as one of the efficient tools to design Robust controllers for complex high-order Non-liner Dynamic Plant operating under Uncertainty. Condition.
- Sliding Controller Design provides a systematic approach to the problem of maintaining stability and Consistent performance.

Concept of Sliding Mode Control :

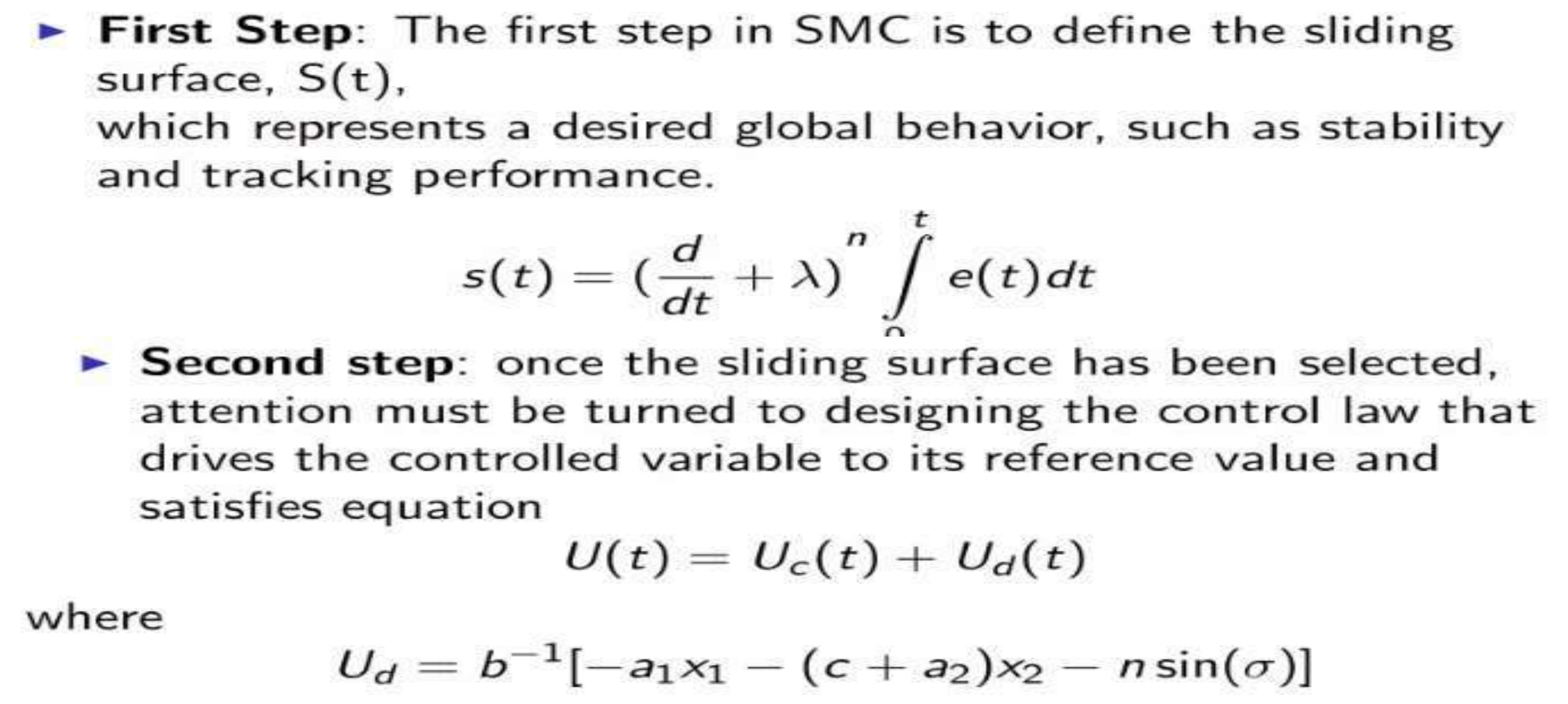
Advantage of Sliding Mode Controllers is their insensitivity to parameters variations and Disturbances once in the Sliding Mode, Their by Eliminating the Necessity of Exact Modelling.



The SMC design is composed of two steps.

- In the first step, a custom-made surface should be designed. While on the sliding surface, the plants dynamics is restricted to the equations of the surface and is robust to match plant uncertainties and external disturbances.
- In the second step, a feedback control law should be designed to provide convergence of a systems trajectory to the sliding surface; thus, the sliding surface should be obtained in a finite time. The systems motion on the sliding surface is called the **sliding mode**.

Concept of Sliding Mode Control :



- **First Step**: The first step in SMC is to define the sliding surface, S(t),
  which represents a desired global behavior, such as stability and tracking performance.

$$s(t) = (\frac{d}{dt} + \lambda)^n \int_0^t e(t)dt$$

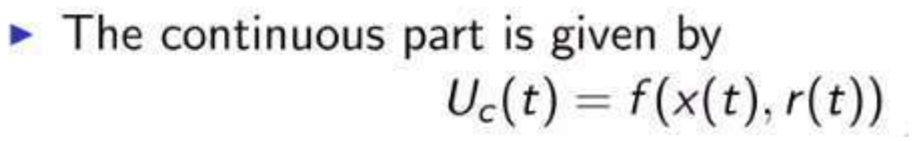
- **Second step**: once the sliding surface has been selected, attention must be turned to designing the control law that drives the controlled variable to its reference value and satisfies equation

$$U(t) = U_c(t) + U_d(t)$$

where

$$U_d = b^{-1}[-a_1x_1 - (c + a_2)x_2 - n\sin(\sigma)]$$

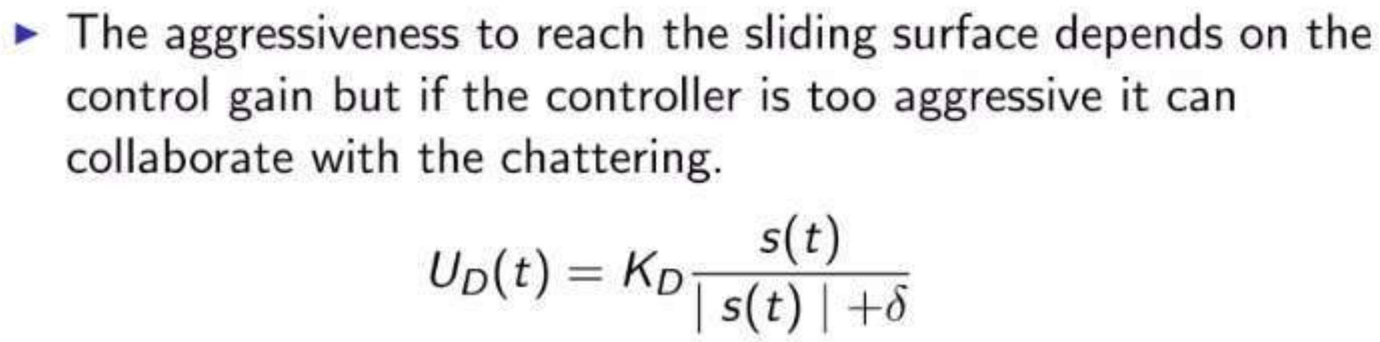
Concept of Sliding Mode Control



- The continuous part is given by

$$U_c(t) = f(x(t), r(t))$$

The Continuous part of the Controller is obtained by combining the process model and Sliding Condition. The Discontinuous part is Non-Linear and Represents the Switching element of the control law.

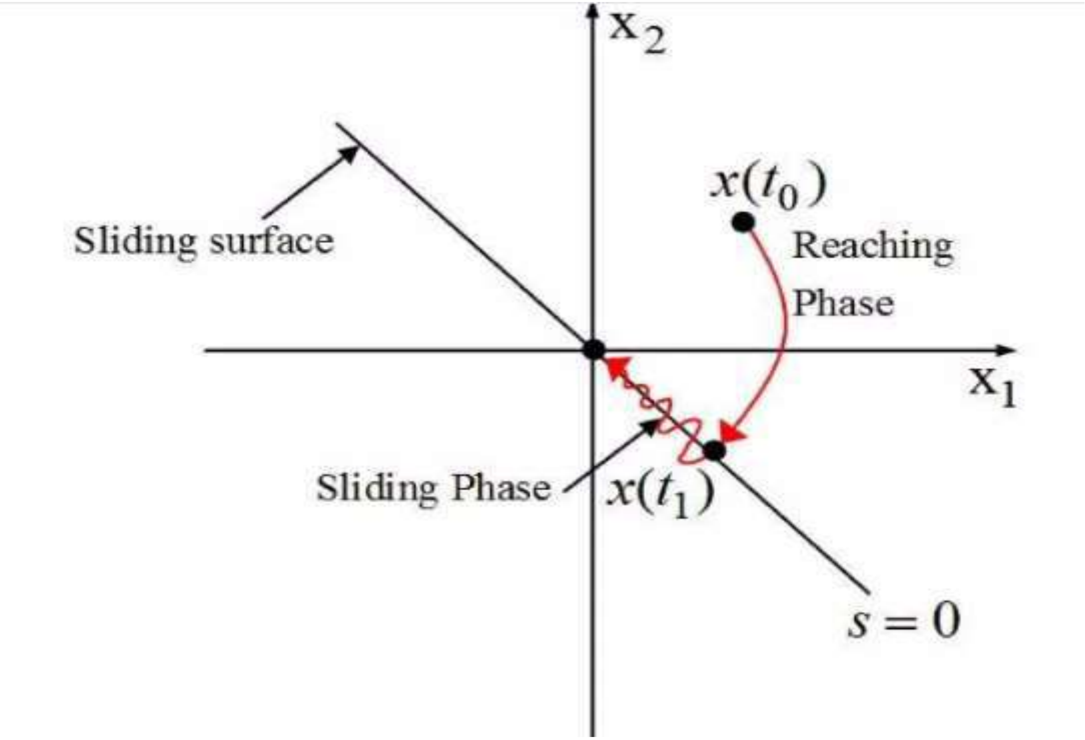


- The aggressiveness to reach the sliding surface depends on the control gain but if the controller is too aggressive it can collaborate with the chattering.

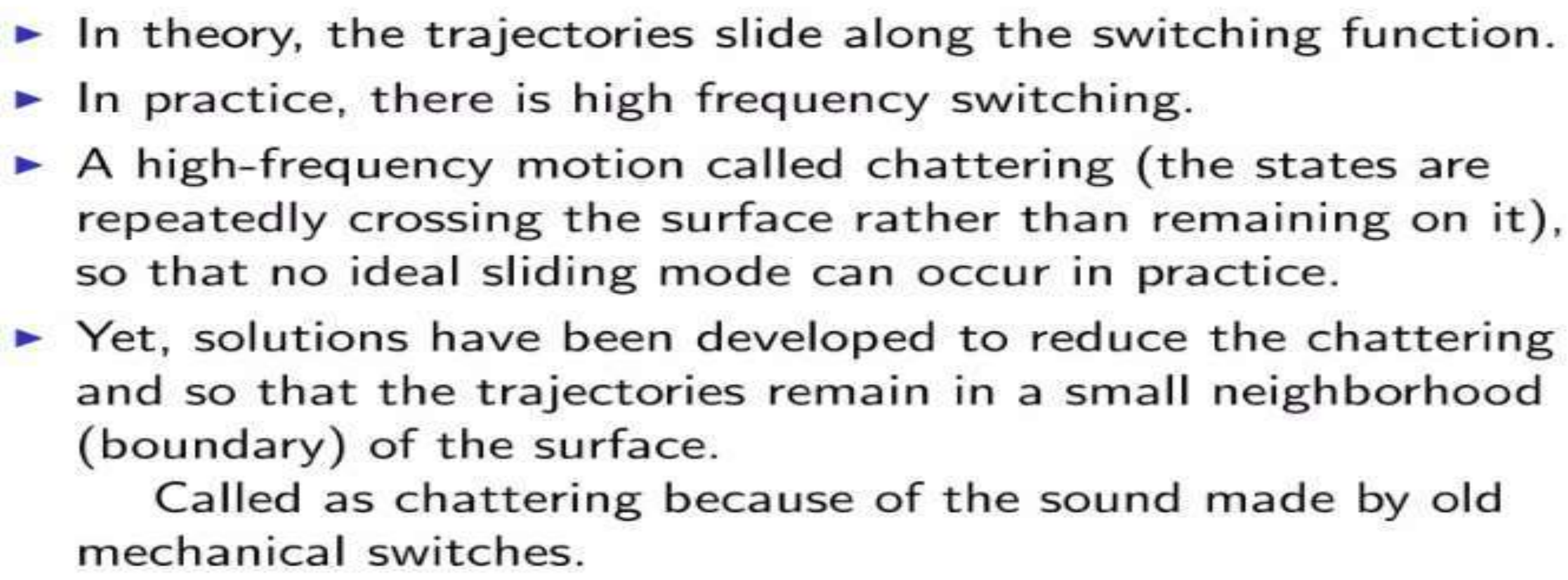
$$U_D(t) = K_D \frac{s(t)}{| s(t) | + \delta}$$

Concept of Sliding Mode Control

Sliding can be clearly understood by the below figure



Chattering Main Drawback :



- In theory, the trajectories slide along the switching function.
- In practice, there is high frequency switching.
- A high-frequency motion called chattering (the states are repeatedly crossing the surface rather than remaining on it), so that no ideal sliding mode can occur in practice.
- Yet, solutions have been developed to reduce the chattering and so that the trajectories remain in a small neighborhood (boundary) of the surface.

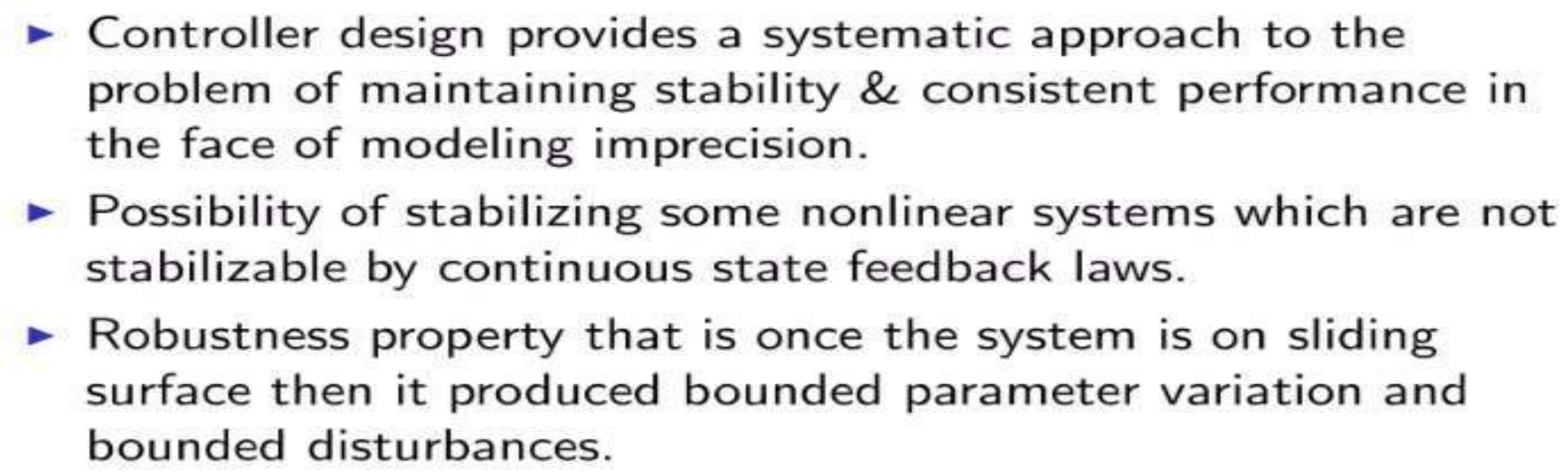
  Called as chattering because of the sound made by old mechanical switches.

Chattering Reduction methods :

Chattering could be Reduced or Suppressed using different Techniques such as :

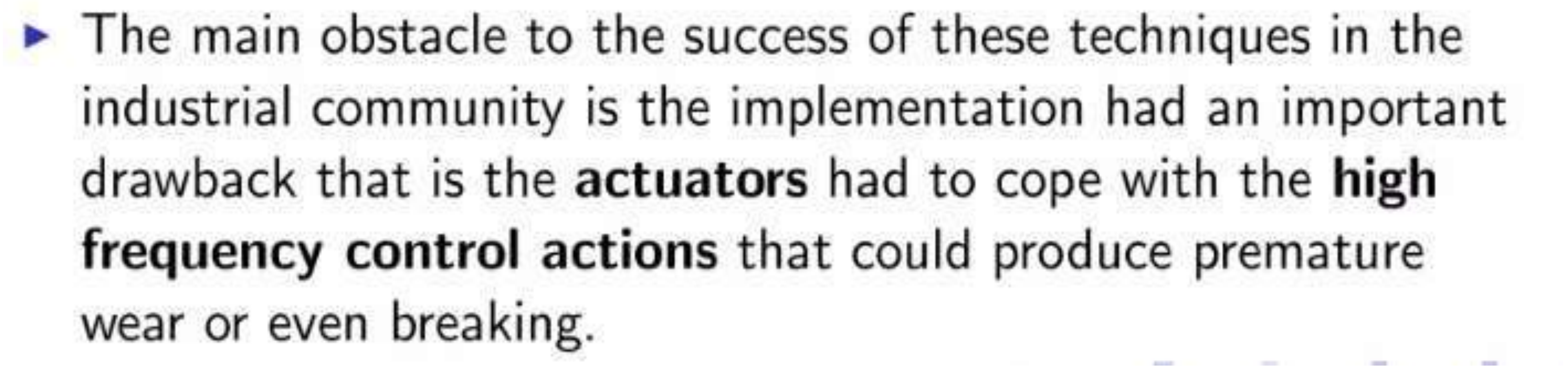
1. Non-Linear Gains.
2. Dynamic Extension.
3. Higher order Sliding Mode Control.

Merits :



- Controller design provides a systematic approach to the problem of maintaining stability & consistent performance in the face of modeling imprecision.
- Possibility of stabilizing some nonlinear systems which are not stabilizable by continuous state feedback laws.
- Robustness property that is once the system is on sliding surface then it produced bounded parameter variation and bounded disturbances.

Demerits :



- The main obstacle to the success of these techniques in the industrial community is the implementation had an important drawback that is the **actuators** had to cope with the **high frequency control actions** that could produce premature wear or even breaking.

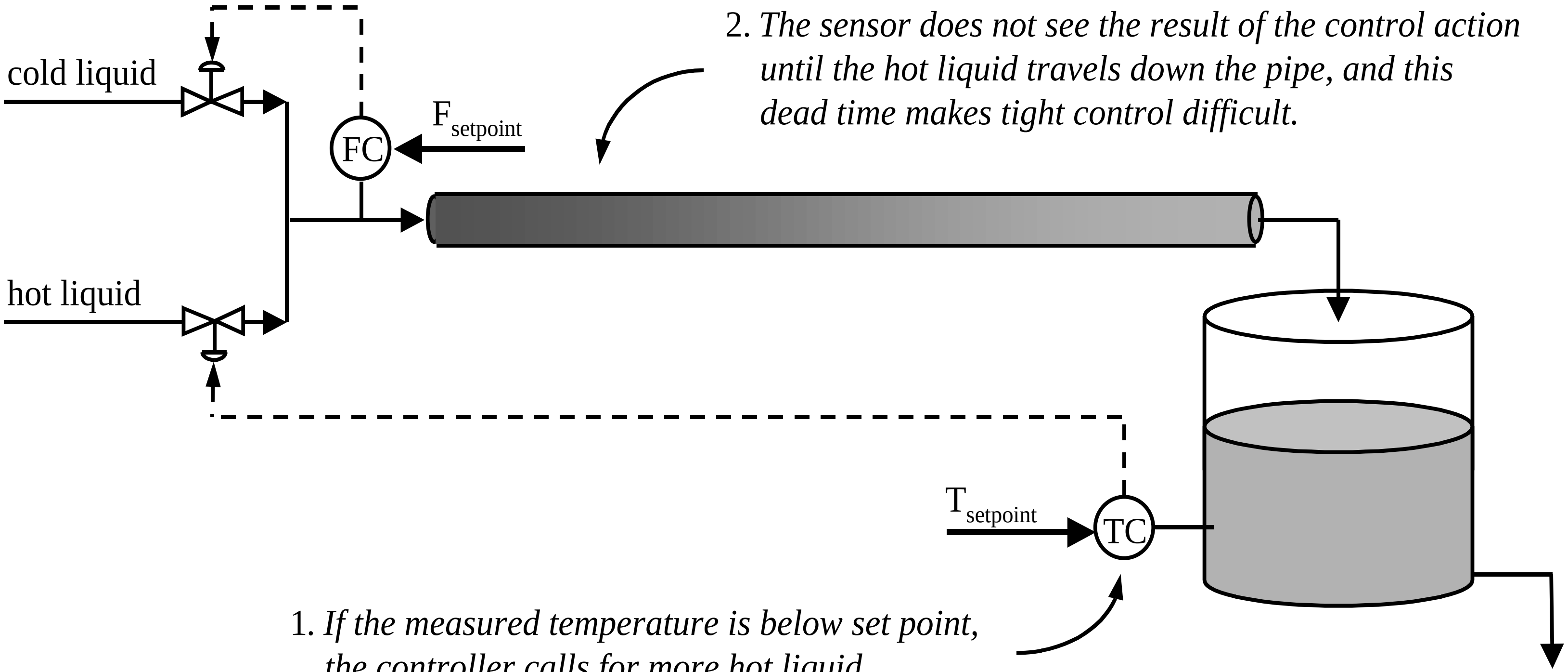
Thank You

Dead Time Compensation (Smith Predictor)

Presented By B Koti Reddy
Department of Atomic Energy
Heavy Water Board

Large Dead Time Impacts Controller Performance

Example of a "distant" temperature control



2. *The sensor does not see the result of the control action until the hot liquid travels down the pipe, and this dead time makes tight control difficult.*

1. *If the measured temperature is below set point, the controller calls for more hot liquid.*

- A hot and cold liquid stream combine at the entrance to a pipe, travel its length, and spill into a tank
- Control objective is to maintain temperature in the tank by adjusting the flow rate of hot liquid entering the pipe

Large Dead Time Impacts Controller Performance

Example of a “distant” temperature control

- If tank temperature is below set point, the hot liquid valve is opened and the temperature entering the pipe increases
- The sensor does not detect this, so the valve is opened more and more and the pipe fills with ever hotter liquid
- When hot liquid reaches the tank, the temperature rises to set point and the controller steadies the hot liquid flow rate
- But the full pipe continues pouring hot liquid into the tank, causing tank temperature to continue to rise
- Because of the delay, the controller will now fill the pipe with too-cold liquid resulting in large oscillations in temperature
- **Solutions** → 1) *detune* the PID controller

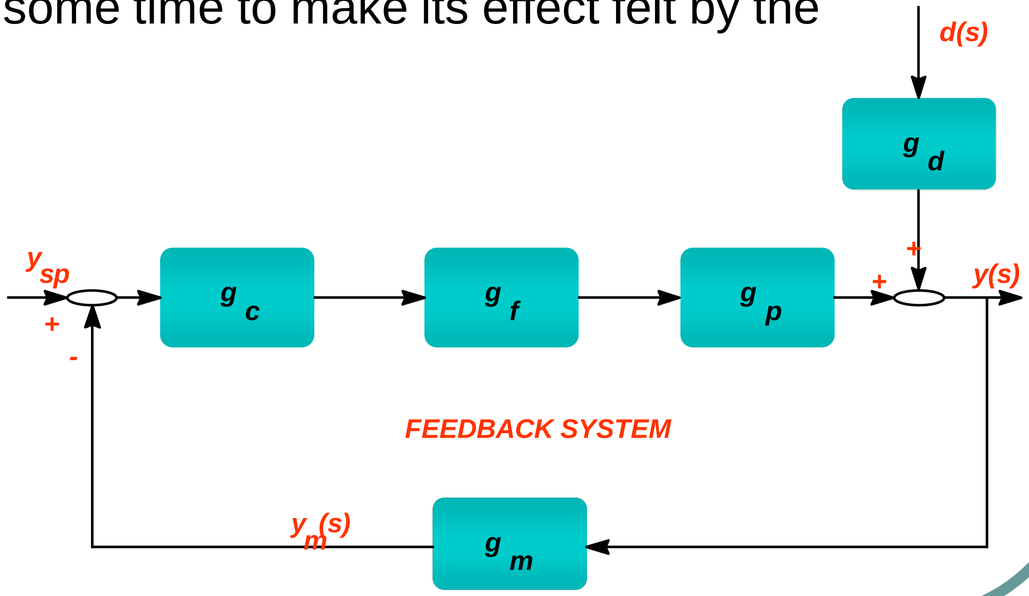
  2) switch to MPC (Model Predictive Control)

Delay Compensation

For processes with large time delays,

- A disturbance entering the process will not be detected until after some time period has passed,
- The control action based on the delayed information will be inadequate, and
- The control action may take some time to make its effect felt by the process.

The potential for instability increases!



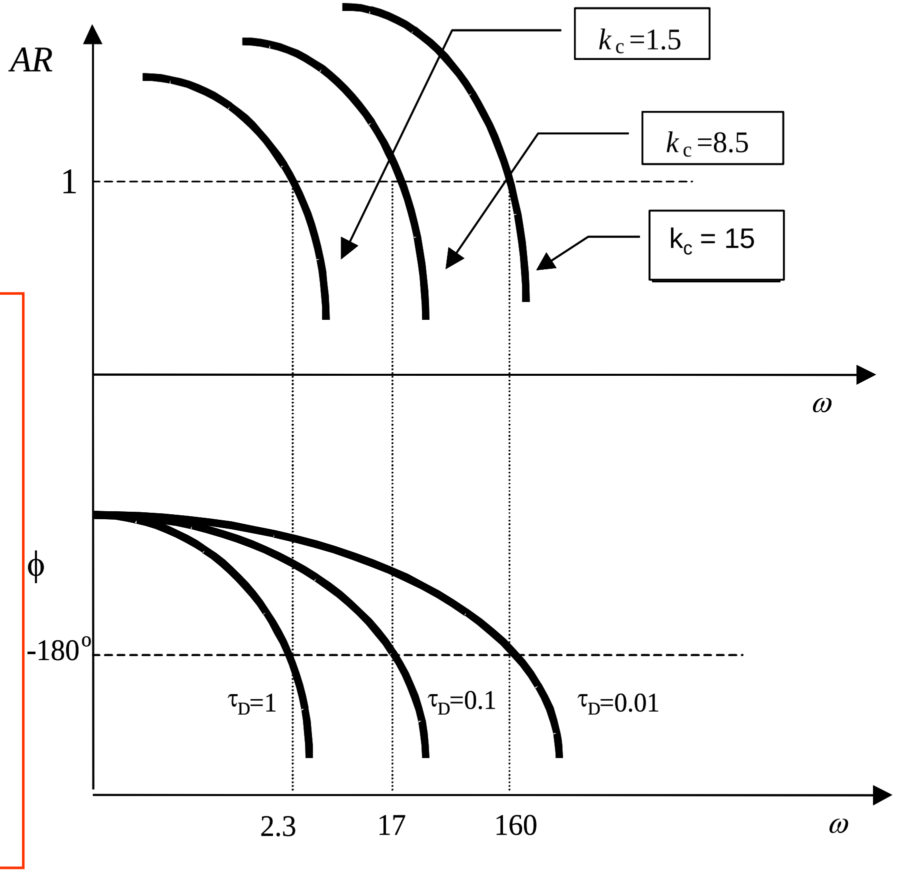
Delay Compensation

For processes with time delays, e.g.:

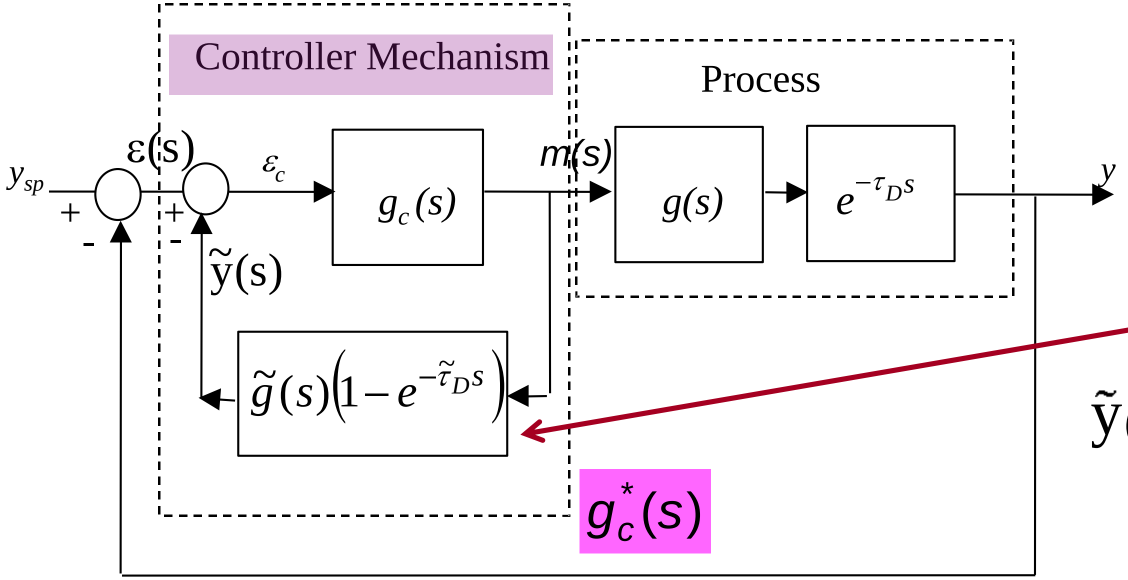
$$g_{OL}(s) = \frac{k_c e^{-\tau_D s}}{0.5s + 1}$$

Bode plots show that …when the process dead-time increases

- ➢ the crossover frequency decreases
- ➢ ..as a consequence, the ultimate gain decreases



Delay Compensation



$\tilde{g}(s)$ is the best estimate of the rational part of the process transfer function

$\tilde{t}_D$ is the best estimate of the actual process delay

Estimated process model:

$$\tilde{y}(s) = \tilde{g}(s)\cdot\left(1 - e^{-\tilde{t}_D s}\right)\cdot m(s)$$

- transfer function of the controller mechanism:

$$g_c^*(s) = \frac{m(s)}{e(s)} = \frac{g_c(s)}{1 + g_c(s)\tilde{g}(s)\left(1 - e^{-\tilde{t}_D s}\right)}$$

- closed loop transfer function for the “servo” problem:

$$G_{CL}(s) = \frac{y(s)}{y_{sp}(s)} = \frac{g_c^*(s)g(s)e^{-t_D s}}{1 + g_c^*(s)g(s)e^{-t_D s}}$$

Delay Compensation

- Now we assume a “**perfectly estimated**” model:

$$\tilde{g}(s)=g(s); \qquad \tilde{t}_D=t_D$$

- $g_c{}^*(s)$ is simplified to:

$$g_c^*(s)=\frac{m(s)}{e(s)}=\frac{g_c(s)}{1+g_c(s)g(s)\left(1-e^{-t_D s}\right)}$$

- the closed loop transfer function for the “servo” problem becomes:

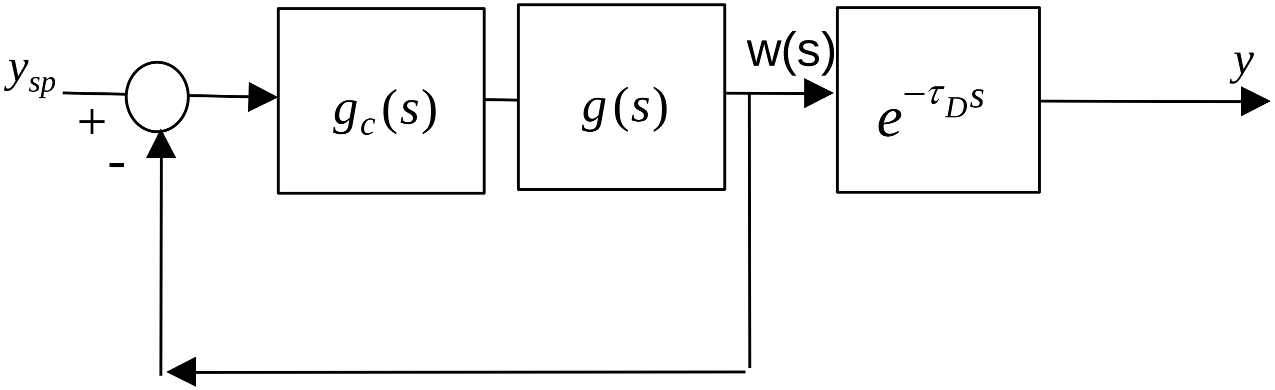
$$G_{CL}(s)=\frac{y(s)}{y_{sp}(s)}=\frac{\dfrac{g_c(s)}{1+g_c(s)g(s)\left(1-e^{-t_D s}\right)}g(s)e^{-t_D s}}{1+\dfrac{g_c(s)}{1+g_c(s)g(s)\left(1-e^{-t_D s}\right)}g(s)e^{-t_D s}}$$

Delay Compensation

- after algebraic manipulation, the closed loop transfer function for the “servo” problem is:

$$\boxed{G_{CL}(s) = \frac{y(s)}{y_{sp}(s)} = \frac{g_c(s) \times g(s)}{1 + g_c(s) \times g(s)} e^{-t_D s}}$$

- the closed-loop block diagram can be viewed as this:



- the closed loop transfer function can be viewed as follows:

$$G_{CL}(s) = \frac{y(s)}{y_{sp}(s)} = \frac{w(s)}{y_{sp}(s)} e^{-t_D s}$$

Delay Compensation

☺

- With Delay Compensator we can send **current** and not delayed information back to the controller
- If the **model is exact**, Delay Compensator moves the dead time out of the feedback loop
- The loop **stability is greatly improved**.
- **Much tighter control** can be achieved (e.g., gains can be increased manifold).

Delay Compensation

☹

- In most control problems

Real Process Model → ***Modeling error***

↓

compensation would not be complete!

...especially in the case of transportation delays that could change with process conditions.

- For uncertain processes (**inexact model**), the performance of the delay compensation can be arbitrarily poor.

Example Smith-1

Consider the following process:

$$g_p(s) = \frac{2e^{-2s}}{(10s+1)(5s+1)}$$

OBJECTIVES

- **Design a Delay Compensator**
- **Compare with a PI controller without Delay Compensator**

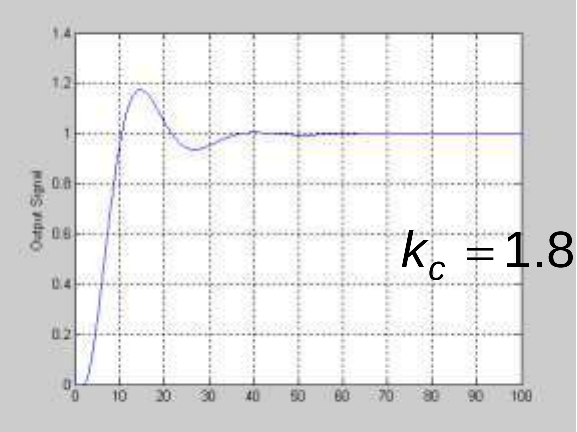
$$g_p(s) = g(s) \cdot g_{dt}(s) = \frac{2}{(10s+1)(5s+1)} e^{-2s}$$

Example Smith-1

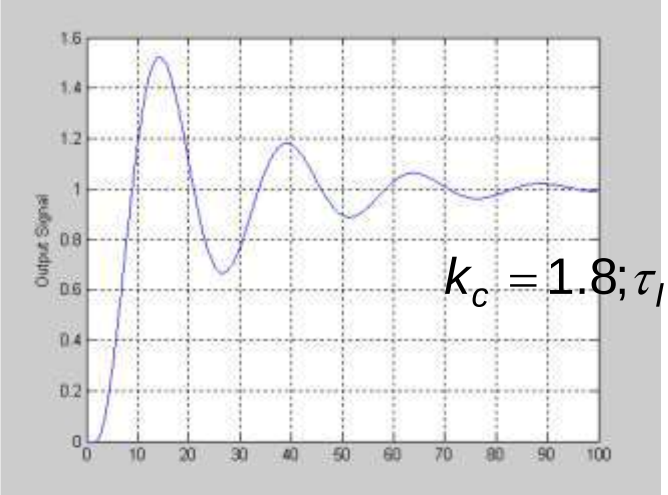
Closed loop response to a unit step in set point

PI Controller with Delay Compensator

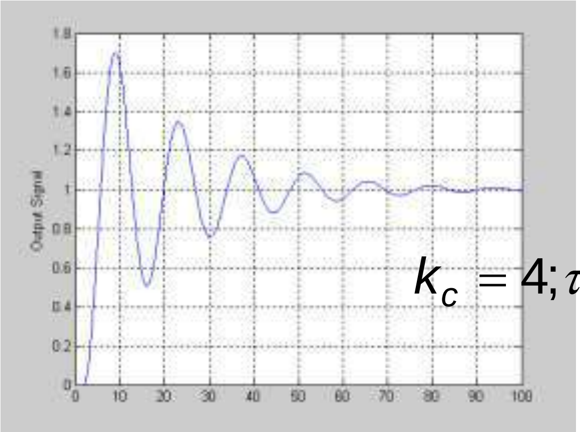
PI Controller only



$k_c = 1.8; \tau_I = 15s$

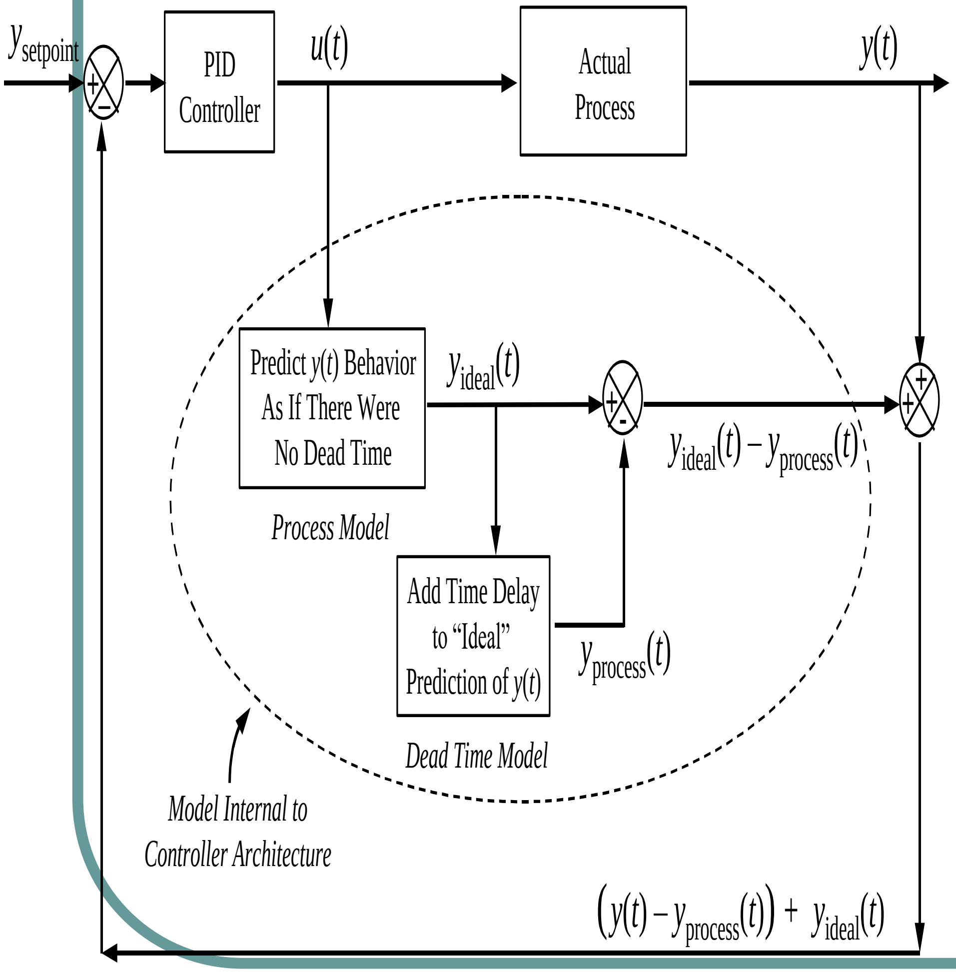


$k_c = 1.8; \tau_I = 15s$



$k_c = 4; \tau_I = 5s$

Smith Predictor
alternative block diagram representation



- The ideal process model receives u(t) and produces $y_{\text{ideal}}(t)$, a prediction of what $y(t)$ will be one dead time into the future
- this $y_{\text{ideal}}(t)$ is stored for one θ_P in the dead time model block. At the same time, a previously stored $y_{\text{process}}(t)$ is released that is the value of $y_{\text{ideal}}(t)$ stored one θ_P ago
- $y_{\text{process}}(t)$ is a prediction of the current value of $y(t)$

The Smith Predictor

The Smith controller error, $e^*(t)$, is thus:

$$e^*(t) = y_{setpoint}(t) - \mathbf{(}\, y(t) - y_{process}(t) + y_{ideal}(t)\mathbf{)}$$

If the model *exactly* describes the process dynamics

$$y(t) - y_{process}(t) = 0$$

so for a perfect model, the $e^*(t)$ going to the controller is:

$$e^*(t) = y_{setpoint}(t) - y_{ideal}(t)$$

If the model is exact, the Smith error is the set point minus a prediction of the process variable if there were no dead time

The model will never be exact, but:

- the better the model, the greater the benefit
- a bad model can make poor performance horrible

The Smith Predictor is the first and simplest example of MPC

Stability and convergence of ODEs

Presented By B Koti Reddy
Department Of Atomic Energy
Heavy Water Board

Contents:

Lyapunov stability of ODEs

- epsilon-delta and beta-function definitions
- Lyapunov’s stability theorem

Properties of hybrid systems

$\mathcal{X}_{\text{sig}} \equiv$ set of all piecewise continuous signals x:$[0,T) \to \mathbb{R}^n$, $T\in(0,\infty]$
$\mathcal{Q}_{\text{sig}} \equiv$ set of all piecewise constant signals q:$[0,T)\to \mathcal{Q}$, $T\in(0,\infty]$

Sequence property $\equiv p : \mathcal{Q}_{\text{sig}} \times \mathcal{X}_{\text{sig}} \to$ {false,true}

E.g.,

$$p(q,x) = \begin{cases} \text{true} & q(t) \in \{1,3\},\ x(t) \geq x(t+3),\ \forall t \\ \text{false} & \text{otherwise} \end{cases}$$

A pair of signals $(q, x) \in \mathcal{Q}_{\text{sig}} \times \mathcal{X}_{\text{sig}}$ ***satisfies*** p if $p(q, x)$ = true

A hybrid automaton H ***satisfies*** p (write $H \vDash p$) if
$p(q, x)$ = true, for every solution (q, x) of H

"***ensemble properties***" $\equiv$ property of the whole family of solutions
(cannot be checked just by looking at isolated solutions)
e.g., continuity with respect to initial conditions…

Lyapunov stability (ODEs)

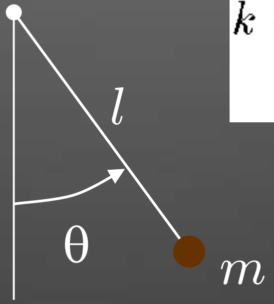
$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

equilibrium point $\equiv x_{\text{eq}} \in \mathbb{R}^n$ for which $f(x_{\text{eq}}) = 0$

thus $x(t) = x_{\text{eq}}\ \forall\ t \geq 0$ is a solution to the ODE

E.g., pendulum equation

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2$$



$k \equiv$ friction coefficient

$x_1 \equiv \theta \qquad x_2 \equiv \dot{\theta}$

two equilibrium points:

$x_1 = 0$, $x_2 = 0$ (down)

and $x_1 = \pi$, $x_2 = 0$ (up)

Lyapunov stability (ODEs)

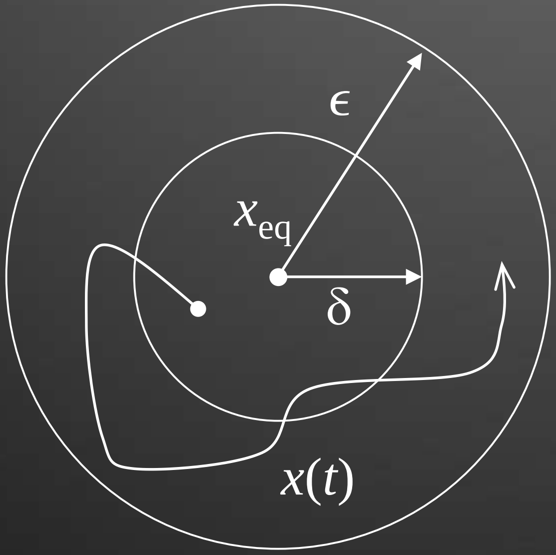
$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

equilibrium point $\equiv x_{\text{eq}} \in \mathbb{R}^n$ for which $f(x_{\text{eq}}) = 0$

thus $x(t) = x_{\text{eq}}\ \forall\, t \geq 0$ is a solution to the ODE

Definition (ϵ–δ definition):
The equilibrium point $x_{\text{eq}} \in \mathbb{R}^n$ is ***(Lyapunov) stable*** if

$$\forall\, \epsilon > 0\ \exists\, \delta > 0 : \|x(t_0) - x_{\text{eq}}\| \leq \delta \quad \Rightarrow \quad \|x(t) - x_{\text{eq}}\| \leq \epsilon\ \forall\, t \geq t_0 \geq 0$$



1. if the solution starts close to x_{eq} it will remain close to it forever
2. ϵ can be made arbitrarily small by choosing δ sufficiently small

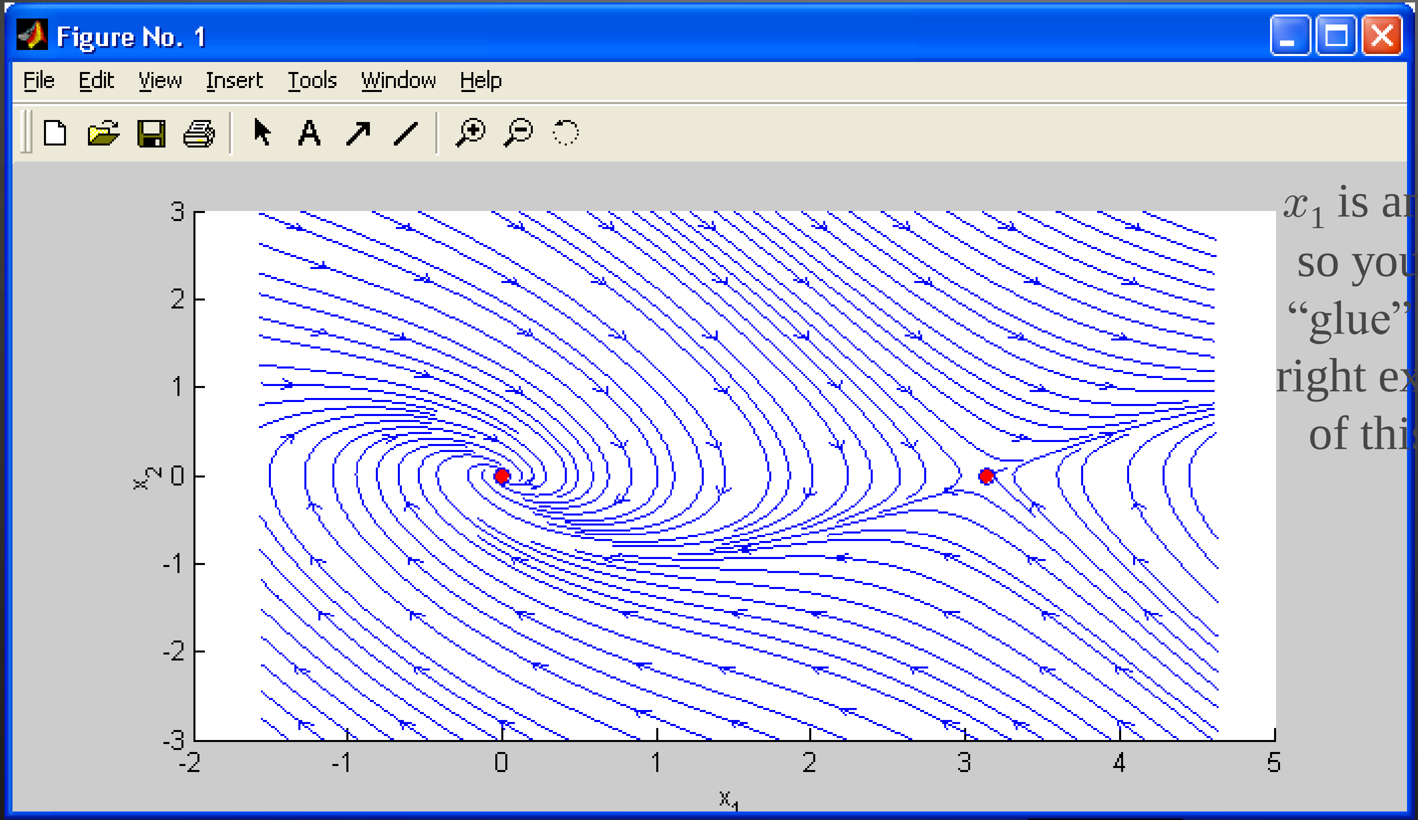
Example #1: Pendulum

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

$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2$$

$k \equiv$ friction coefficient

$x_1 \equiv \theta \qquad x_2 \equiv \dot{\theta}$



x_1 is an angle so you must "glue" left to right extremes of this plot

x_{eq}=(0,0) stable

x_{eq}=(π,0) unstable

pend.m

Lyapunov stability – continuity definition

$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$

$\mathcal{X}_{\text{sig}} \equiv$ set of all piecewise continuous signals taking values in $\mathbb{R}^n$

Given a signal $x \in \mathcal{X}_{\text{sig}}$, $\|x\|_{\text{sig}} := \sup_{t \geq 0} \|x(t)\|$

signal norm

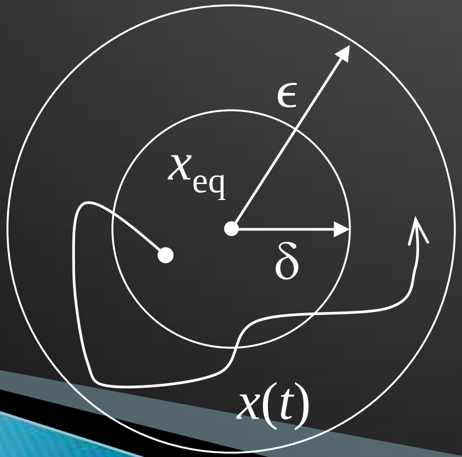
ODE can be seen as an operator

$$T : \mathbb{R}^n \to \mathcal{X}_{\text{sig}}$$

that maps $x_0 \in \mathbb{R}^n$ into the solution that starts at $x(0) = x_0$

Definition (continuity definition):
The equilibrium point $x_{\text{eq}} \in \mathbb{R}^n$ is ***(Lyapunov) stable*** if T is continuous at x_{eq}:

$$\forall \epsilon > 0 \ \exists \ \delta > 0 : \|x_0 - x_{\text{eq}}\| \leq \delta \quad \Rightarrow \quad \underbrace{\|T(x_0) - T(x_{\text{eq}})\|_{\text{sig}} \leq \epsilon}_{\sup_{t \geq 0} \|x(t) - x_{\text{eq}}\| \leq \epsilon}$$



can be extended to nonequilibrium solutions

Stability of arbitrary solutions

$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

$\mathcal{X}_{\text{sig}} \equiv$ set of all piecewise continuous signals taking values in $\mathbb{R}^n$

Given a signal $x \in \mathcal{X}_{\text{sig}}$, $\|x\|_{\text{sig}} := \sup_{t \geq 0} \|x(t)\|$

signal norm

ODE can be seen as an operator

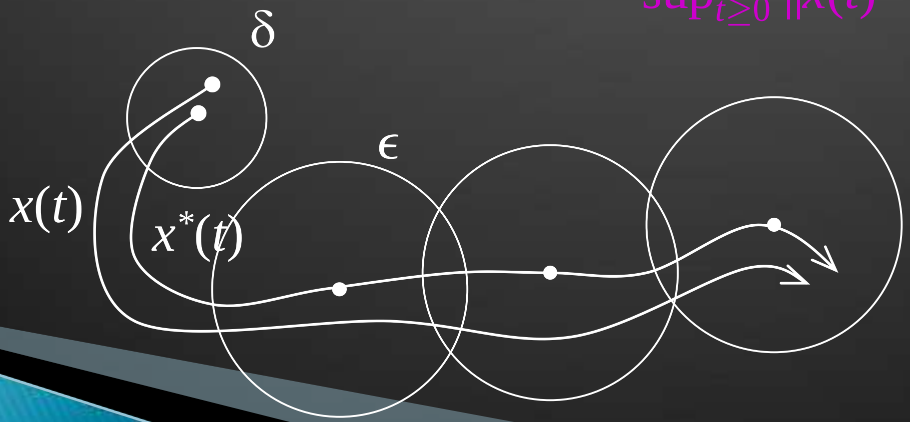
$$T : \mathbb{R}^n \to \mathcal{X}_{\text{sig}}$$

that maps $x_0 \in \mathbb{R}^n$ into the solution that starts at $x(0) = x_0$

Definition (continuity definition):

A solution $x^*:[0,T) \to \mathbb{R}^n$ is (***Lyapunov) stable*** if T is continuous at $x^*_0 := x^*(0)$, i.e.,

$$\forall\, \epsilon > 0\ \exists\, \delta > 0 : \|x_0 - x^*_0\| \leq \delta \quad \Rightarrow \quad \underbrace{\|T(x_0) - T(x^*_0)\|_{\text{sig}} \leq \epsilon}_{\sup_{t \geq 0} \|x(t) - x^*(t)\| \leq \epsilon}$$



pend.m

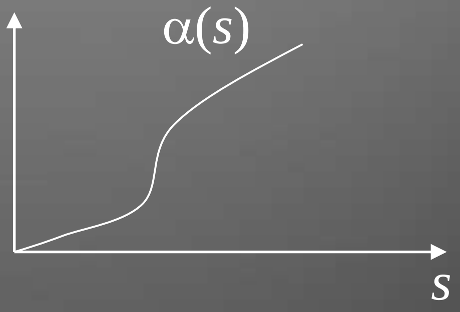
Lyapunov stability

$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

equilibrium point ≡ $x_{eq} \in \mathbb{R}^n$ for which $f(x_{eq}) = 0$

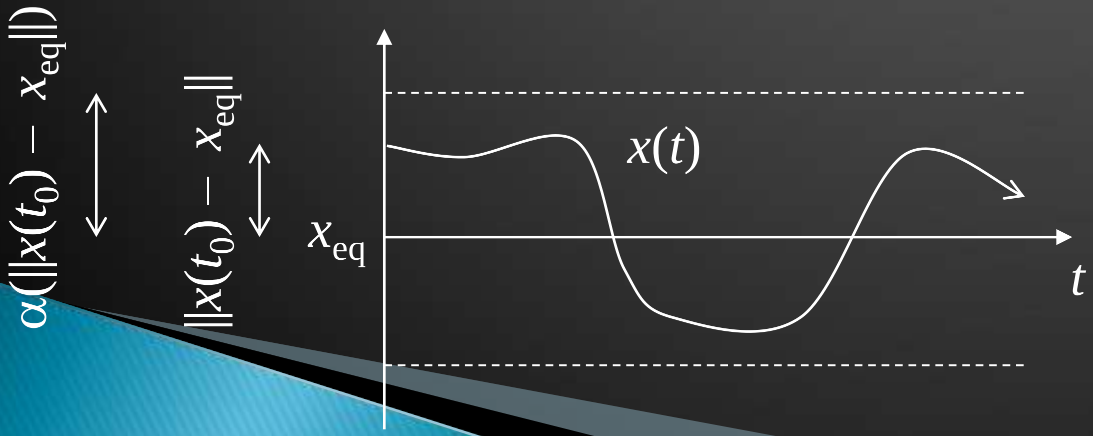
class $\mathcal{K}$ ≡ set of functions $\alpha:[0,\infty)\to[0,\infty)$ that are

1. continuous
2. strictly increasing
3. $\alpha(0)=0$



Definition (class $\mathcal{K}$ function definition):
The equilibrium point $x_{eq} \in \mathbb{R}^n$ is ***(Lyapunov) stable*** if $\exists\ \alpha \in \mathcal{K}$:

$$\|x(t) - x_{eq}\| \le \alpha(\|x(t_0) - x_{eq}\|)\ \ \forall\, t \ge t_0 \ge 0,\ \|x(t_0) - x_{eq}\| \le c$$



the function α can be constructed directly from the δ(ϵ) in the ϵ–δ (or continuity) definitions

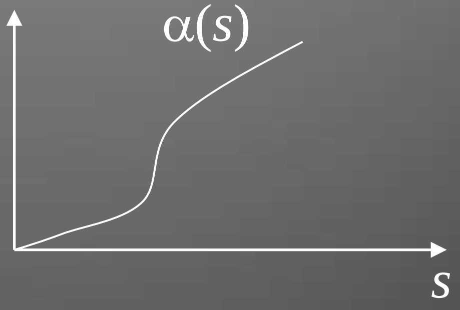
Asymptotic stability

$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

equilibrium point ≡ $x_{eq} \in \mathbb{R}^n$ for which $f(x_{eq}) = 0$

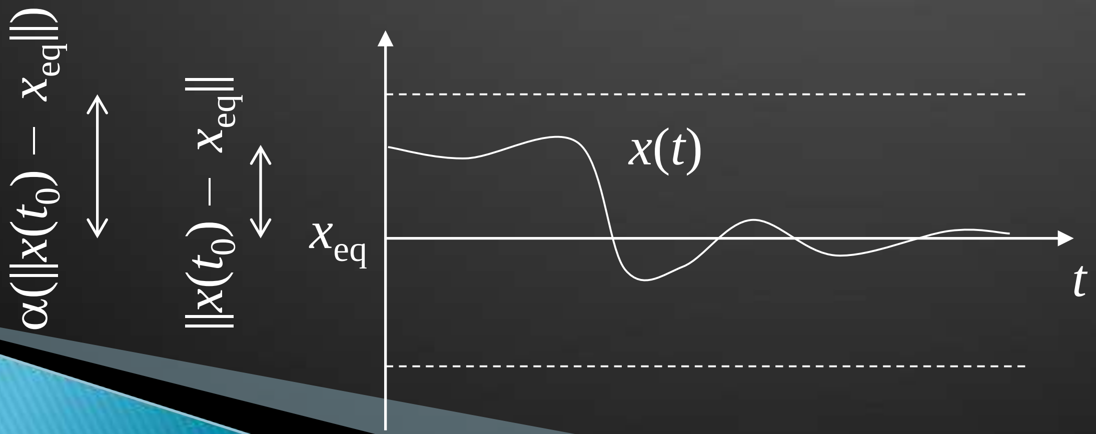
class $\mathcal{K}$ ≡ set of functions $\alpha:[0,\infty)\to[0,\infty)$ that are

1. continuous
2. strictly increasing
3. $\alpha(0)=0$



Definition:
The equilibrium point $x_{eq} \in \mathbb{R}^n$ is ***(globally) asymptotically stable*** if it is Lyapunov stable and for every initial state the solution exists on $[0,\infty)$ and

$$x(t) \to x_{eq} \text{ as } t\to\infty.$$



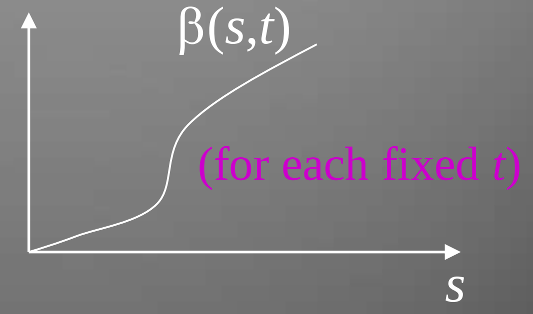
Asymptotic stability

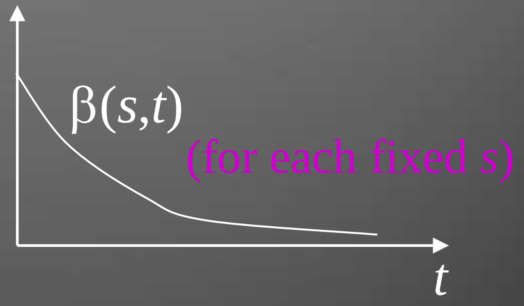
$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

equilibrium point ≡ $x_{eq} \in \mathbb{R}^n$ for which $f(x_{eq}) = 0$

class $\mathcal{KL}$ ≡ set of functions $\beta:[0,\infty)\times[0,\infty)\rightarrow[0,\infty)$ s.t.

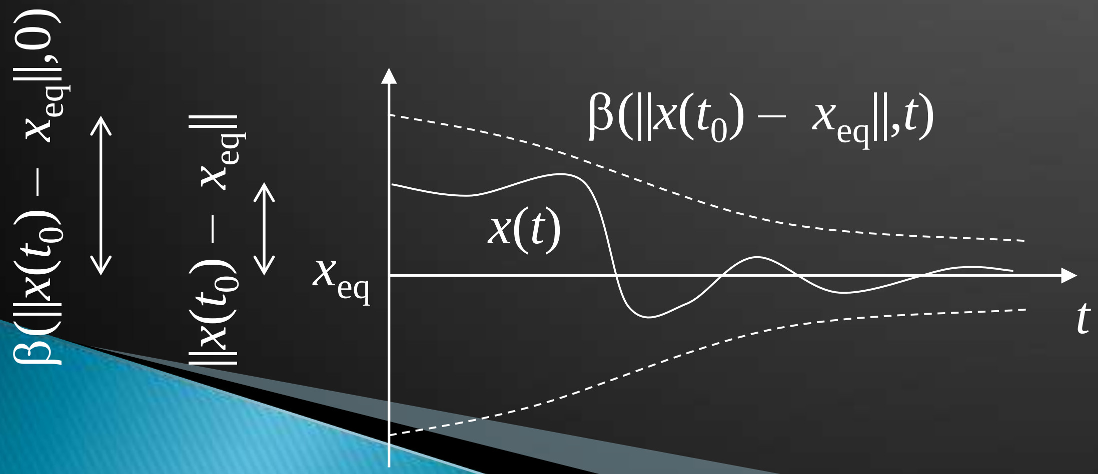
1. for each fixed t, $\beta(\cdot,t) \in \mathcal{K}$
2. for each fixed s, $\beta(s,\cdot)$ is monotone decreasing and $\beta(s,t) \rightarrow 0$ as $t\rightarrow\infty$





Definition (class $\mathcal{KL}$ function definition):
The equilibrium point $x_{eq} \in \mathbb{R}^n$ is ***(globally) asymptotically stable*** if $\exists\ \beta\in\mathcal{KL}$:

$$\|x(t) - x_{eq}\| \leq \beta(\|x(t_0) - x_{eq}\|, t - t_0) \quad \forall\ t \geq t_0 \geq 0$$



We have ***exponential stability*** when

$$\beta(s,t) = c\, e^{-\lambda t} s$$

with $c,\lambda > 0$

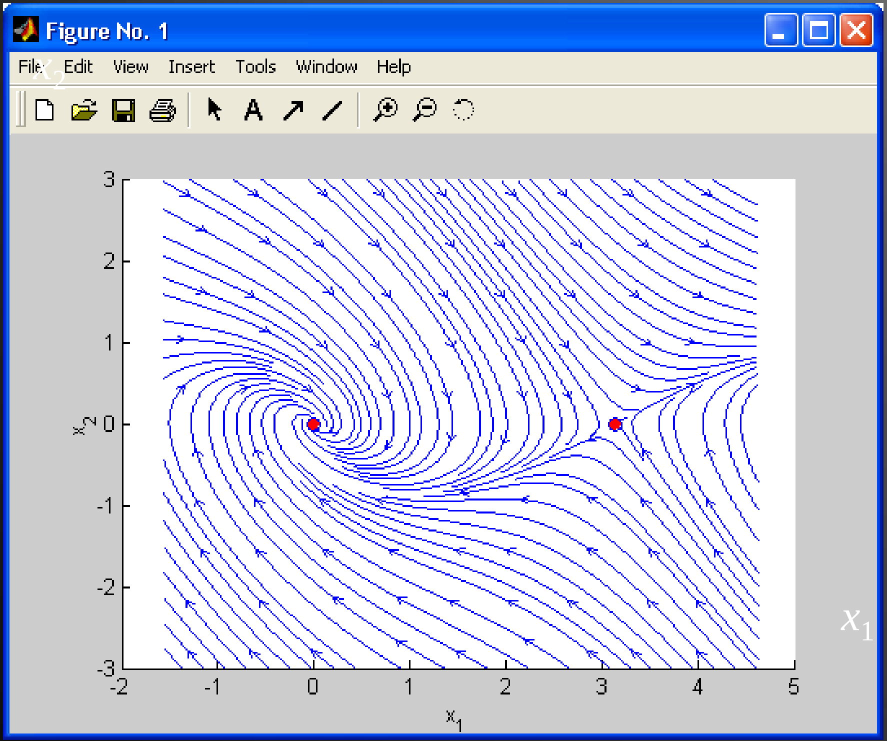
linear in s and negative exponential in t

Example #1: Pendulum

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

$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2$$

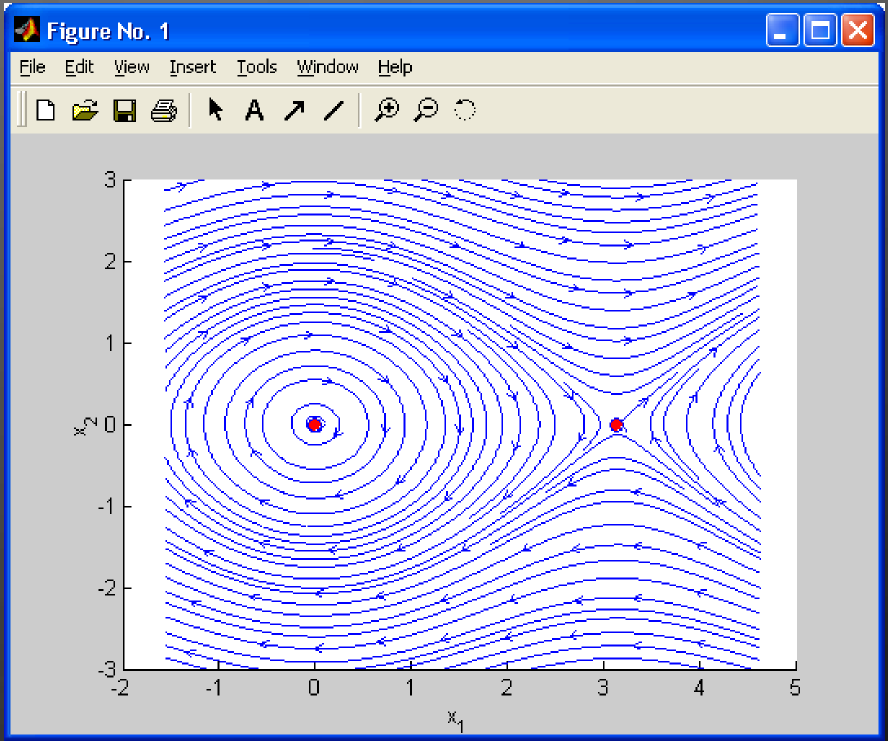
$k > 0$ (with friction)



x_{eq}=(0,0)
asymptotically
stable

x_{eq}=(π,0)
unstable

$k = 0$ (no friction)



x_{eq}=(0,0)
stable but not
asymptotically

x_{eq}=(π,0)
unstable

pend.m

Lyapunov's stability theorem

$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

Definition (class $\mathcal{K}$ function definition):
The equilibrium point $x_{\text{eq}} \in \mathbb{R}^n$ is ***(Lyapunov) stable*** if $\exists\ \alpha \in \mathcal{K}$:

$$\|x(t) - x_{\text{eq}}\| \leq \alpha(\|x(t_0) - x_{\text{eq}}\|) \ \forall\ t \geq t_0 \geq 0,\ \|x(t_0) - x_{\text{eq}}\| \leq c$$

Suppose we could show that $\|x(t) - x_{\text{eq}}\|$ always decreases along solutions to the ODE. Then

$$\|x(t) - x_{\text{eq}}\| \leq \|x(t_0) - x_{\text{eq}}\| \ \forall\ t \geq t_0 \geq 0$$

we could pick $\alpha(s) = s$ ⇒ Lyapunov stability

We can draw the same conclusion by using other measures of how far the solution is from x_{eq}:

$V: \mathbb{R}^n \to \mathbb{R}$ positive definite $\equiv V(x) \geq 0 \ \forall\ x \in \mathbb{R}^n$ with $= 0$ only for $x = 0$
$V: \mathbb{R}^n \to \mathbb{R}$ radially unbounded $\equiv x \to \infty \Rightarrow V(x) \to \infty$

$$V(x - x_{\text{eq}}) \quad \begin{cases} = 0 & x = x_{\text{eq}} \\ > 0 & x \neq x_{\text{eq}} \\ \to \infty & \|x - x_{\text{eq}}\| \to \infty \end{cases}$$

provides a measure of
how far x is from x_{eq}
(not necessarily a metric–may
not satisfy triangular inequality)

Lyapunov's stability theorem

$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

$V: \mathbb{R}^n \to \mathbb{R}$ positive definite $\equiv V(x) \geq 0 \;\; \forall\, x \in \mathbb{R}^n$ with $= 0$ only for $x = 0$

$$V(x - x_{\mathrm{eq}}) \quad \begin{cases} = 0 & x = x_{\mathrm{eq}} \\ > 0 & x \neq x_{\mathrm{eq}} \end{cases}$$

provides a measure of
how far x is from x_{eq}
(not necessarily a metric–may
not satisfy triangular inequality)

Q: How to check if $V(x(t) - x_{eq})$ decreases along solutions?

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} V\left(x(t) - x_{\mathrm{eq}}\right) &= \frac{\partial V}{\partial x}\left(x(t) - x_{\mathrm{eq}}\right)\dot{x}(t) \\ &= \frac{\partial V}{\partial x}\left(x(t) - x_{\mathrm{eq}}\right) f\left(x(t)\right) \end{aligned}$$

gradient of V

A: $V(x(t) - x_{eq})$ will decrease if

$$\frac{\partial V}{\partial x}\left(z - x_{\mathrm{eq}}\right) f(z) \leq 0 \qquad \forall z \in \mathbb{R}^n$$

can be computed without
actually computing $x(t)$
(i.e., solving the ODE)

Lyapunov's stability theorem

Definition (class $\mathcal{K}$ function definition):

$$\dot{x} = f(x) \qquad x \in \mathbb{R}^n$$

The equilibrium point $x_{eq} \in \mathbb{R}^n$ is (***Lyapunov) stable*** if $\exists\ \alpha \in \mathcal{K}$:

$$\|x(t) - x_{eq}\| \le \alpha(\|x(t_0) - x_{eq}\|)\ \ \forall\ t \ge t_0 \ge 0,\ \|x(t_0) - x_{eq}\| \le c$$

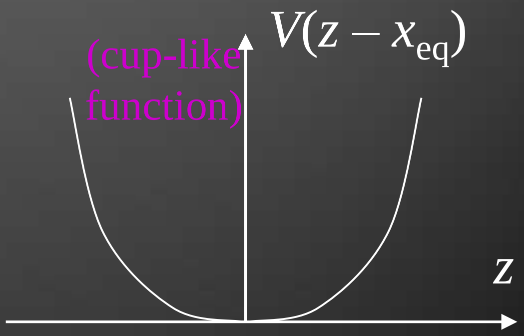
Lyapunov function

Theorem (Lyapunov):

Suppose there exists a continuously differentiable, positive definite function V: $\mathbb{R}^n \to \mathbb{R}$ such that

$$\frac{\partial V}{\partial x}(z - x_{\text{eq}})\, f(z) \le 0 \qquad \forall z \in \mathbb{R}^n$$

Then x_{eq} is a Lyapunov stable equilibrium.



Why?

V non increasing $\Rightarrow V(x(t) - x_{eq}) \le\ V(x(t_0) - x_{eq})\ \forall\ t \ge t_0$

Thus, by making $x(t_0) - x_{eq}$ small we can make $V(x(t) - x_{eq})$ arbitrarily small $\forall\ t \ge t_0$

So, by making $x(t_0) - x_{eq}$ small we can make $x(t) - x_{eq}$ arbitrarily small $\forall\ t \ge t_0$

(we can actually compute α from V explicitly and take $c = +\infty$).