Machine intelligence
Modeling and control overview

Dr. Ahmad Al-Mahasneh

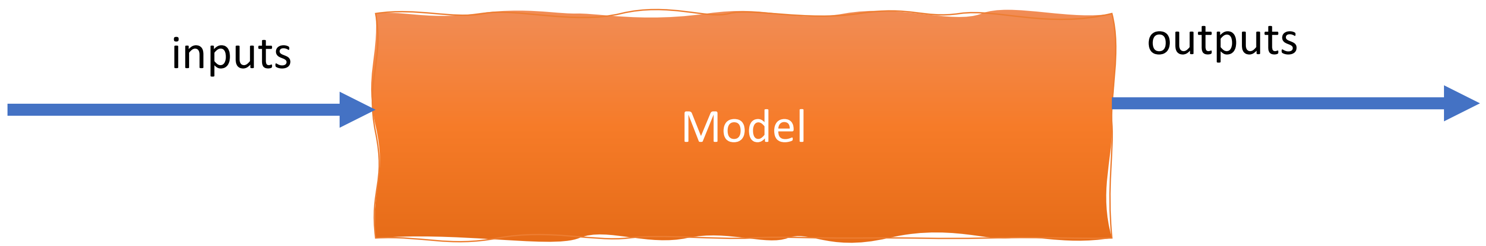
Outline

- Modelling overview
- Control Overview

Modelling

- Modelling of system or a process is the representation of the behavior of a real system by a collection of mathematical equations and/or logic.

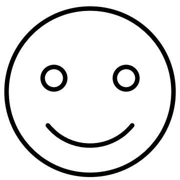
- Models are cause-and-effect structures—they accept external information and process it with their logic and equations to produce one or more outputs.
    - Parameter is a fixed-value unit of information
    - Signal is a changing-unit of information

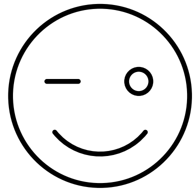


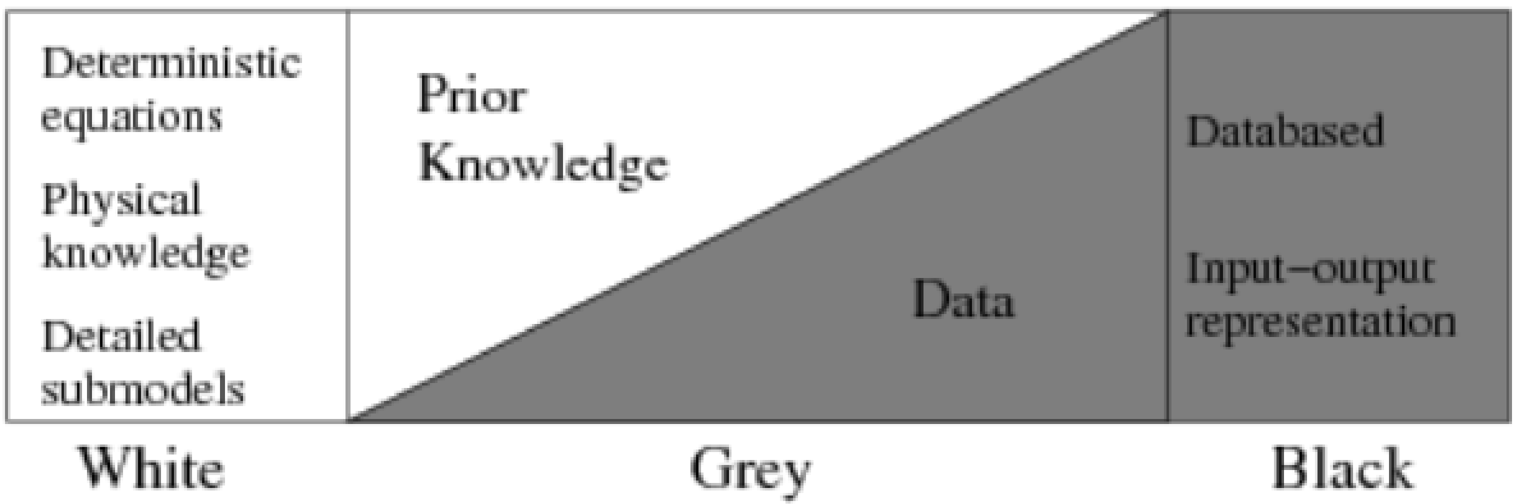
Modelling types

White-box/theoretical	Grey-box	Black-box
• Based on physical principles to find a model of the system. • The model structure is known and deterministic. • Uncertainty is discarded and the model tends to be over specified.	• Model structure is known but its parameters are not. measurements are used to find these parameters. • Between white and black box models. • Combines prior physical knowledge with information in data. • The model is not completely described by physical equations, but equations and the parameters are physically interpretable.	• Model structure and parameters are unknown. • Once we decided the structure, measurements are used to estimate the parameters of this structure. • Data based/driven models - input/output models. • The model and its parameters have little physical significance.
Example: using Newton laws to derive the model of a pendulum system	Example: using measurements of input/output data to find the optimal set of parameters of a given ODE.	Example: using NNs/Fuzzy logic to model a given system from the input/output data.

Modelling types







Good when there is no uncertainty in the system(ideal)

The best combination of physics and data-driven approaches but requires much effort

The best when nothing is known about the process/system.

Modelling Categories

- Linear vs. nonlinear
- Time-invariant vs. time-variant
- SISO vs. MIMO
- Continuous vs. discrete

Linear vs. Nonlinear

- **Linear** models follow the **superposition principle**
  - The summation outputs from individual inputs will be equal to the output of the combined inputs

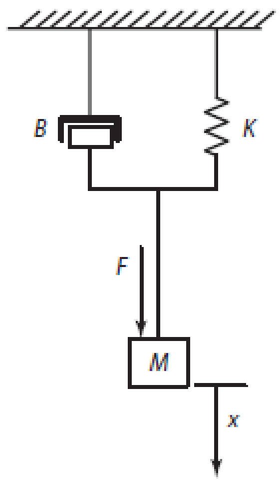
A system represented by $\mathcal{S}$ is said to be *linear* if for inputs $x(t)$ and $v(t)$, and any constants α and β, superposition holds—that is,

$$\begin{aligned}\mathcal{S}[\alpha x(t) + \beta v(t)] &= \mathcal{S}[\alpha x(t)] + \mathcal{S}[\beta v(t)] \\ &= \alpha\mathcal{S}[x(t)] + \beta\mathcal{S}[v(t)]\end{aligned}$$

- Most systems are nonlinear in nature, but linear models can be used to approximate the nonlinear models at certain point.

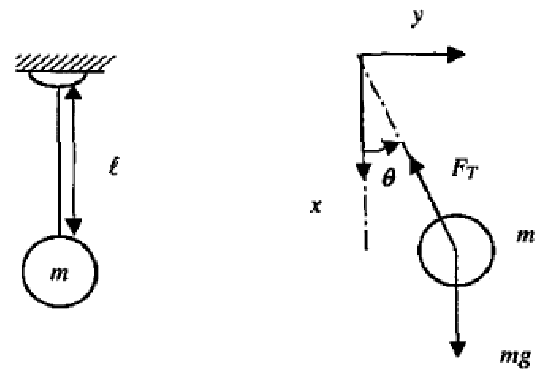
Linear vs. Nonlinear Models

- Linear Systems



$$\ddot{x}(t) = -\frac{B}{M}\dot{x}(t) - \frac{K}{M}(x(t) - x_0) + \frac{1}{M}F(t)$$

- Nonlinear Systems

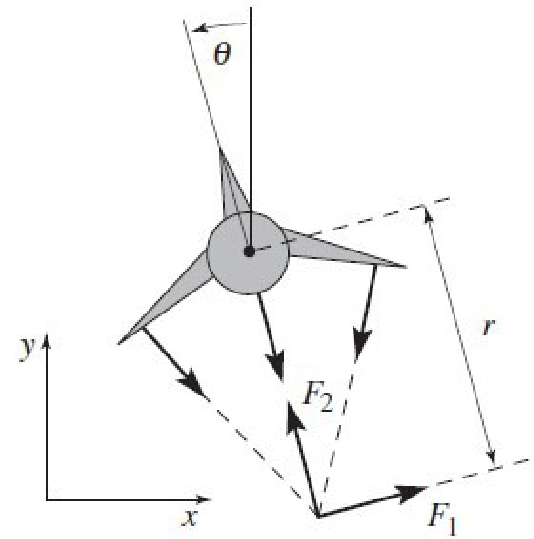


$$-F_T\cos\theta + mg = m\left(-\ell\ddot{\theta}\sin\theta - \ell\dot{\theta}^2\sin\theta\right)$$

$$-F_T\sin\theta = m\left(\ell\ddot{\theta}\cos\theta - \ell\dot{\theta}^2\sin\theta\right)$$

Nonlinear Model: Motion of Aircraft

(a) Harrier "jump jet"



(b) Simplified model

(x, y, θ) denote the position and orientation of the center of mass

$$m\ddot{x} = F_1 \cos\theta - F_2 \sin\theta - c\dot{x},$$
$$m\ddot{y} = F_1 \sin\theta + F_2 \cos\theta - mg - c\dot{y},$$
$$J\ddot{\theta} = rF_1.$$

Linearization using Taylor Series

$$f(x(t)) = f(x_0(t)) + \frac{d f(x_0(t))}{dt}(x(t) - x_0(t)) + \frac{1}{2}\frac{d^2 f(x_0(t))}{dt^2}(x(t) - x_0(t))^2 + \frac{1}{6}\frac{d^3 f(x_0(t))}{dt^3}(x(t) - x_0(t))^3 + \cdots + \frac{1}{n!}\frac{d^n f(x_0(t))}{dt^n}(x(t) - x_0(t))^n$$

Linearization is done at a particular point. For example, linearizing the nonlinear function, *sin* **at *x=0***, using the **first two terms** becomes

$$\sin(x) = \sin(0) + \cos(0)\,(x - 0) = x$$

Time-invariant vs. Time-variant

- The model parameters do not change in time-invariant models

- The model parameters change in time-variant models
  - Example: Mass in rockets vary with time as the fuel is consumed.

If the system parameters change with time, the system is time varying.

Time-invariant vs. variant

- Time-invariant

$$F(t) = ma(t) - g$$

Where m is the mass, a is the acceleration, and g is the gravity

- Time-variant

$$F = m(t)\, a(t) - g$$

Here, the mass varies with time. Therefore the **model is time-varying**

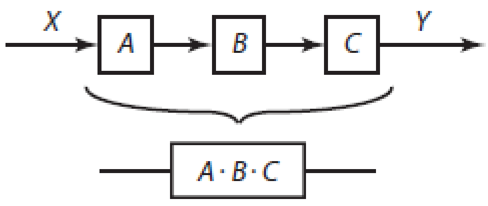
Linear Time-Invariant (LTI)

- **LTI** models are of great use in representing systems in many engineering applications.
  - The appeal is its simplicity and mathematical structure.

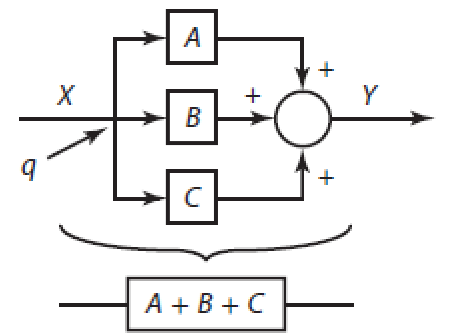
- Although most actual systems are nonlinear and time varying
  - Linear models are used to approximate around an operating point the nonlinear behavior
  - Time-invariant models are used to approximate in short segments the system's time-varying behavior.

LTI Models: Block Diagrams Manipulation

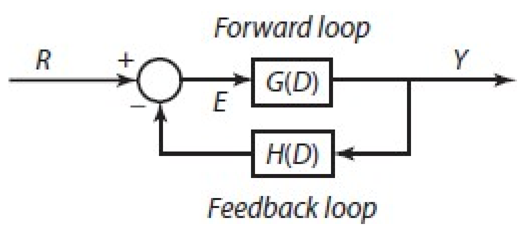
SERIES MANIPULATION—SERIES BLOCKS MULTIPLY



PARALLEL MANIPULATION—PARALLEL BLOCKS ADD



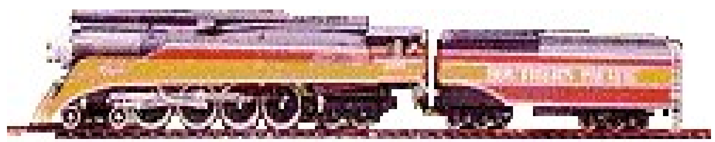
BASIC FEEDBACK SYSTEM (BFS) BLOCK DIAGRAM

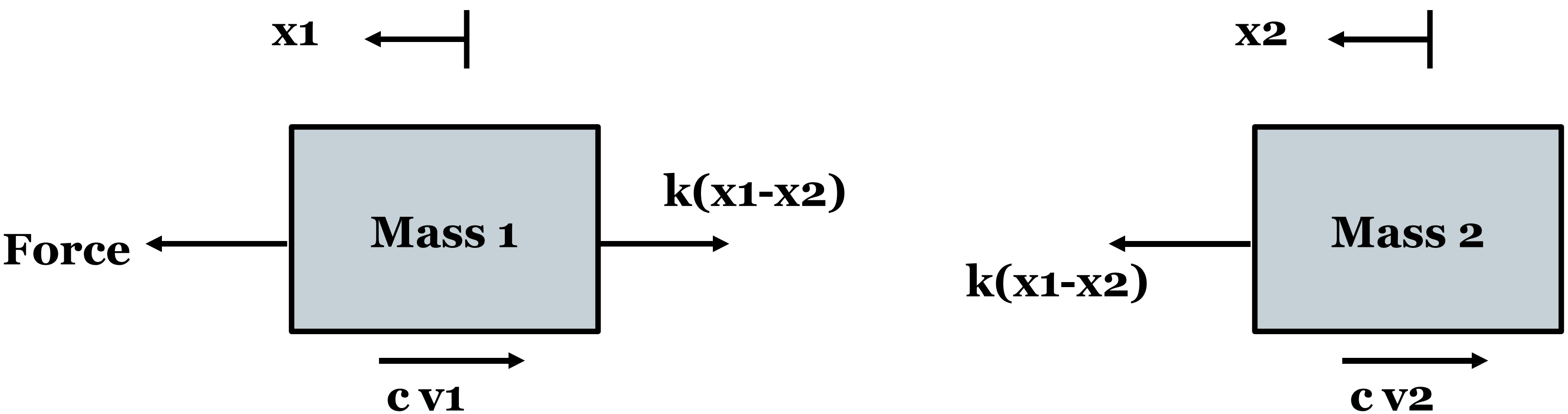


$$\frac{Y}{R} = \frac{G(D)}{1 + G(D) \cdot H(D)}$$

LTI Example: Mechanical Model

- Consider a two carriage train system





$$m_1 \ddot{x}_1 = f - k(x_1 - x_2) - c\dot{x}_1$$

$$m_2 \ddot{x}_2 = k(x_1 - x_2) - c\dot{x}_2$$

Example continued

- Taking the Laplace transform of the equations gives

$$m_1 s^2 X_1(s) = F(s) - k(X_1(s) - X_2(s)) - csX_1(s)$$

$$m_2 s^2 X_2(s) = k(X_1(s) - X_2(s)) - csX_2(s)$$

- Note: Laplace transforms the time domain problem into s-domain (i.e. frequency)

$$L\{x(t)\} = X(s) = \int_0^\infty e^{-st} x(t)dt$$

$$L\{x^{\cdot}(t)\} = sX(s)$$

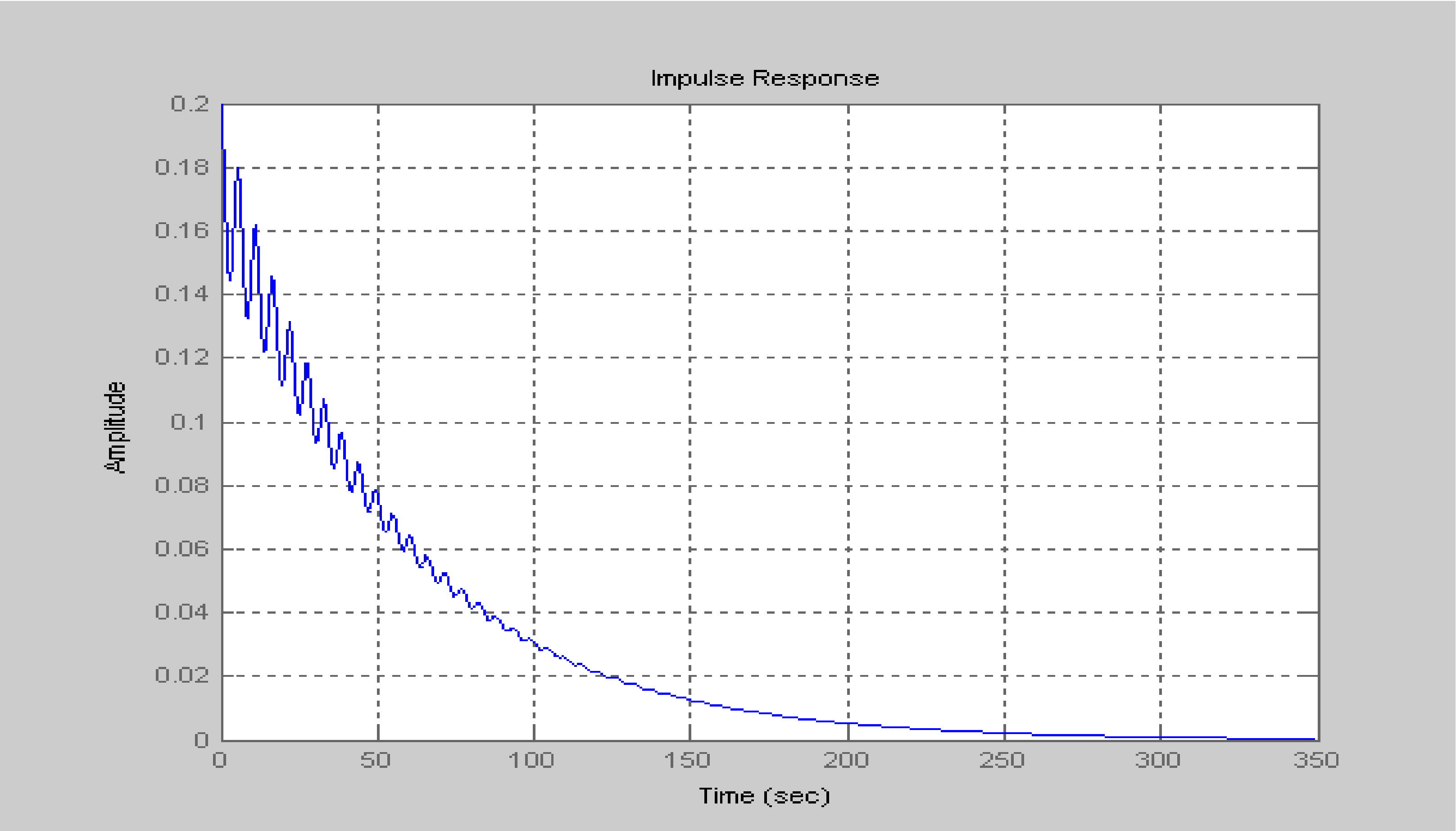
Example continued

- Manipulating the previous two equations, gives the following transfer function (with F as input and V1 as output)

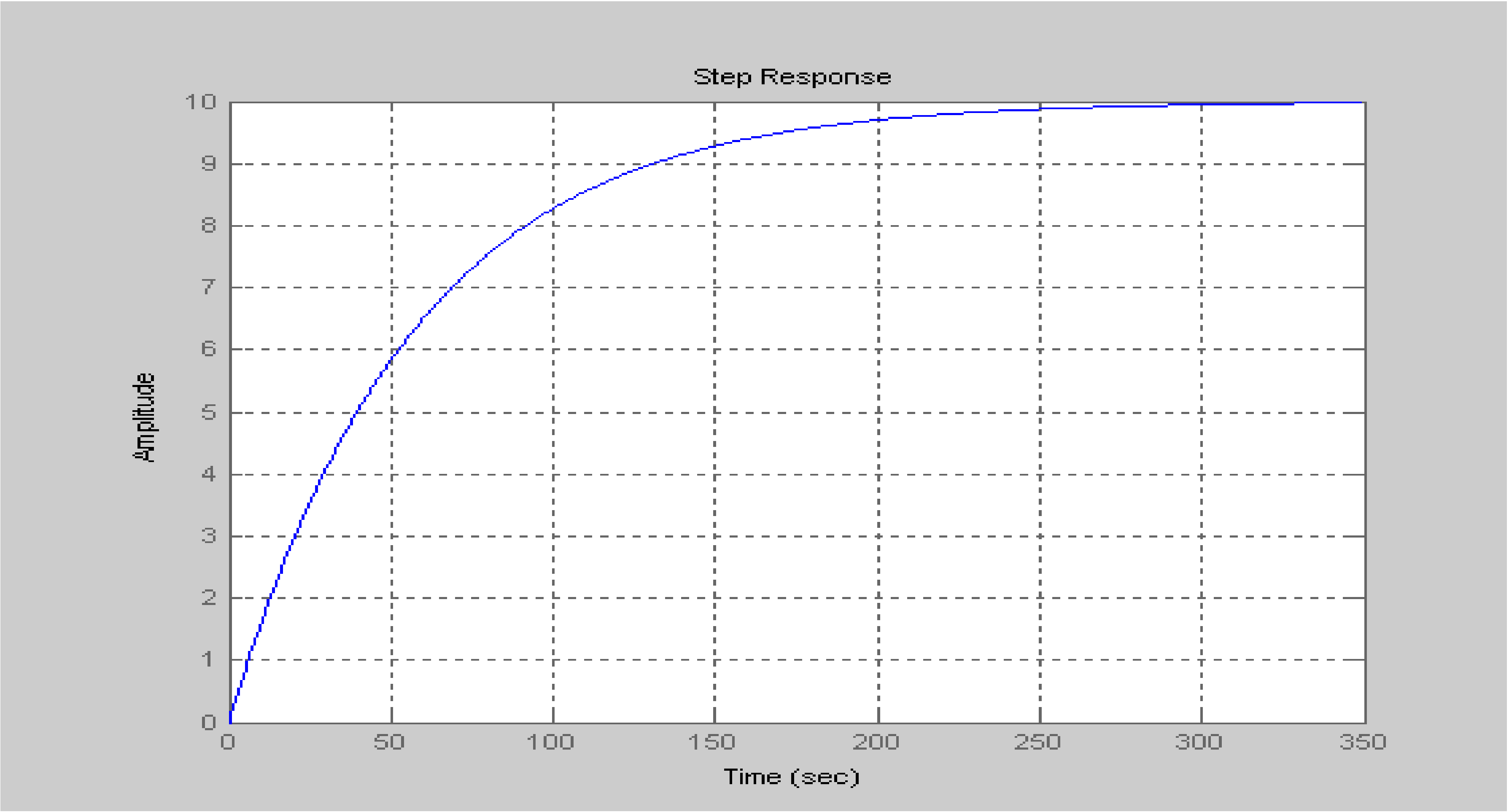
$$\frac{V_1(s)}{F(s)} = \frac{m_2 s^2 + cs + k}{m_1 m_2 s^3 + c(m_1 + m_2)s^2 + \left(km_1 + km_2 + c^2\right)s + 2kc}$$

- Note: Transfer function is an equation that gives the relationship between a specific input to a specific output

Example continued: Impulse response



Example continued: Step response



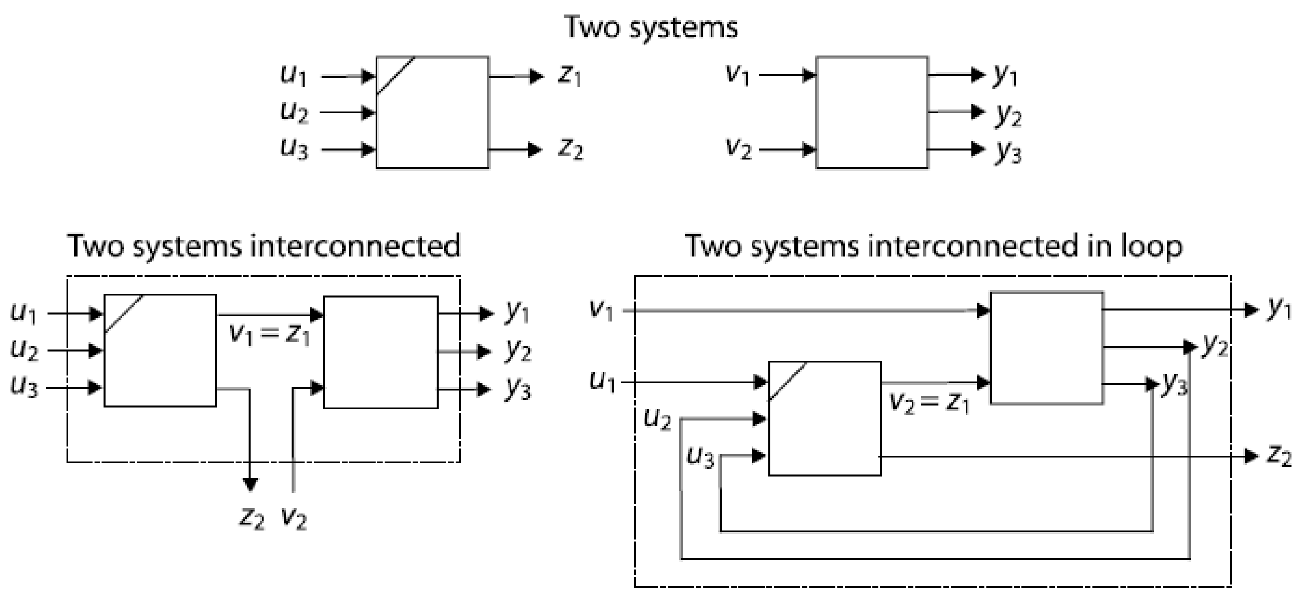
SISO vs. MIMO

- **Single-Input Single-Output** (SISO) models are somewhat easy to use. Transfer functions can be used to relate input to output.

- **Multiple-Input Multiple-Output** (MIMO) models involve combinations of inputs and outputs and are difficult to represent using transfer functions. ***MIMO models use State-Space equations***

What is a System?

- A **system** can be considered as a mathematical transformation of inputs into outputs.
- A **system** should have clear boundaries

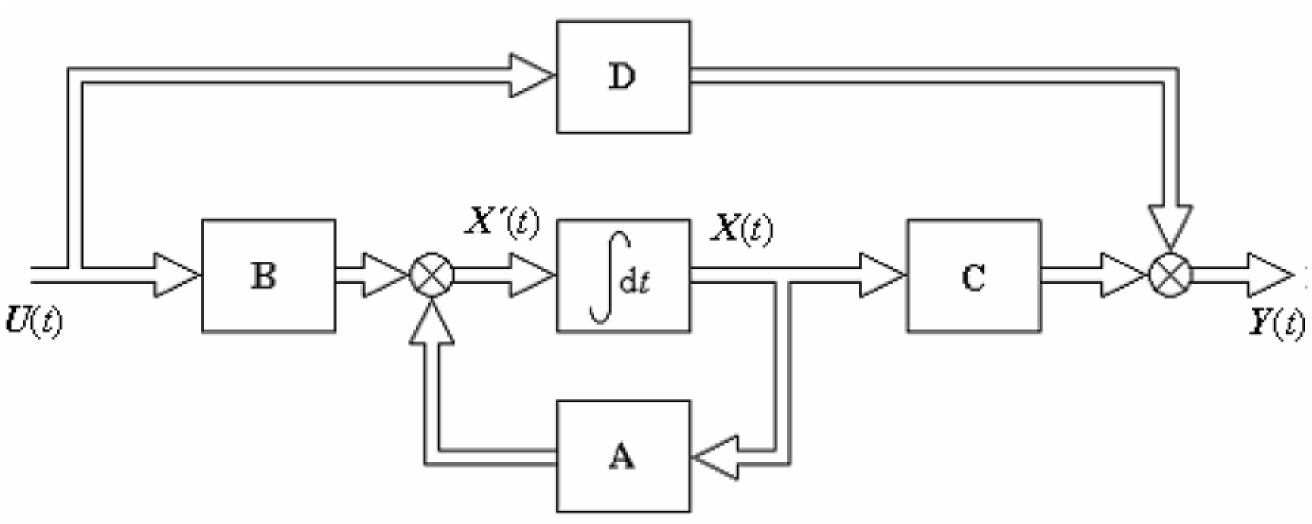


System States

- **Transfer functions**
  - Concentrates on the input-output relationship only.
  - Relates output-input to one-output only **SISO**
  - It hides the details of the inner workings.

- **State-Space Models**
  - ***States*** are introduced to get better insight into the systems' behavior. These states are a collection of variables that summarize the present and past of a system
  - Models can be used for **MIMO** models

SISO vs. MIMO Systems



Continuous vs. discrete models

- **Continuous models** have continuous-time as the dependent variable and therefore inputs-outputs take all possible values in a range

- **Discrete models** have discrete-time as the dependent variable and therefore inputs-outputs take on values at specified times only in a range

Continuous vs. discrete models

- **Continuous Models**

- Differential equations
- Integration
- Laplace transforms

- **Discrete Models**

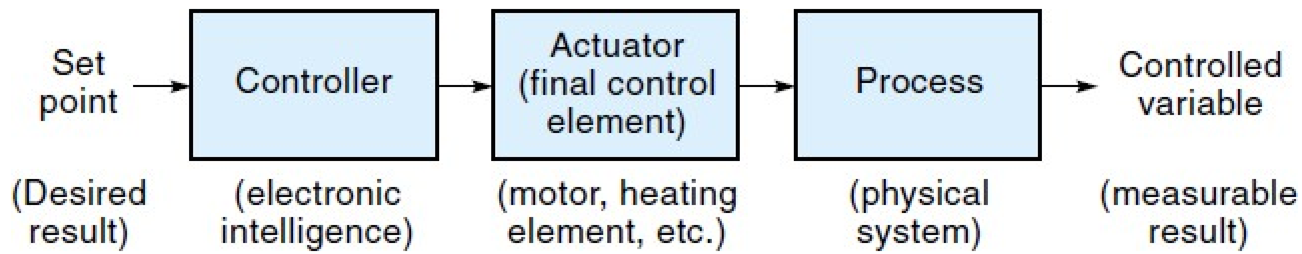
- Difference equations
- Summation
- Z-transforms

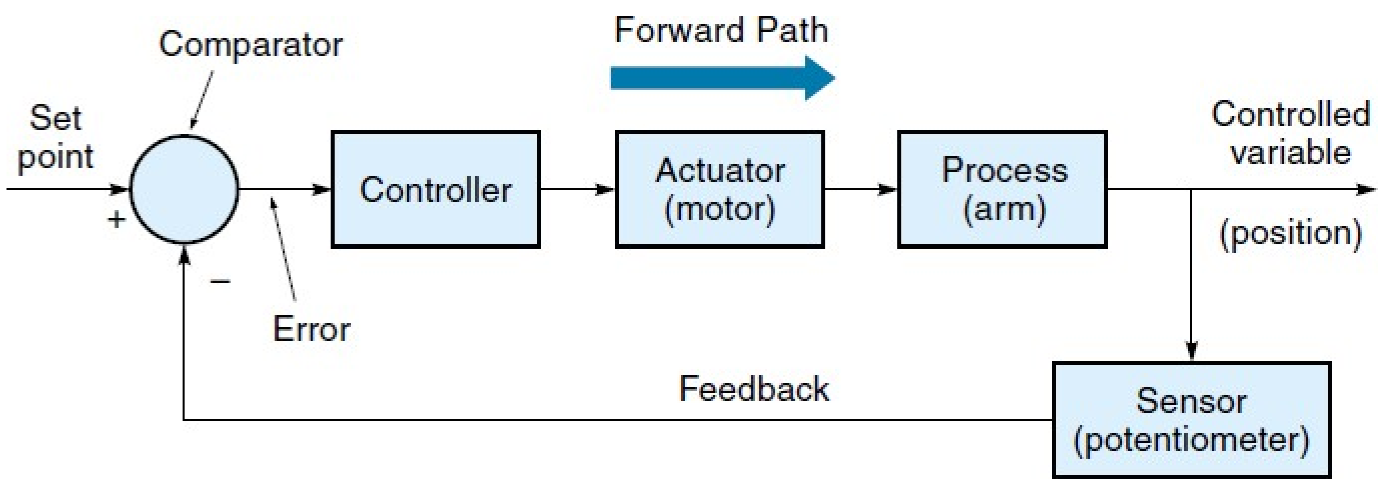
Control Systems

- The control system is at the **heart of mechatronic systems** and its selection is arguably the most **critical decision in the design process**.

- The controller selection involves two inter-dependent parts:
  - The control method (i.e. software)
  - The physical controller (i.e. hardware)

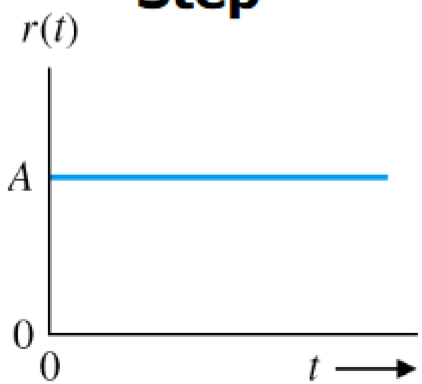
Closed-Loop vs. Open-Loop Control





Test input signals

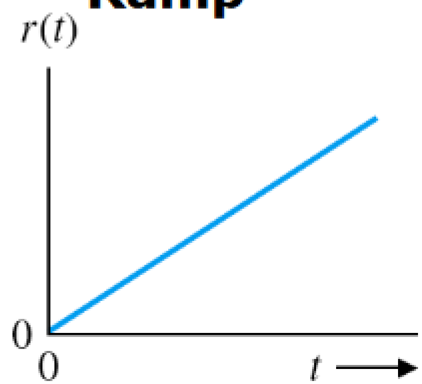
Step



$$r(t) = \begin{cases} A & \text{for } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$R(s) = \frac{A}{s}$$

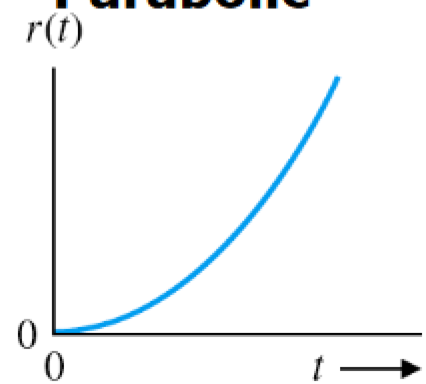
Ramp



$$r(t) = \begin{cases} At & \text{for } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$R(s) = \frac{A}{s^2}$$

Parabolic



$$r(t) = \begin{cases} \frac{1}{2}At^2 & \text{for } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

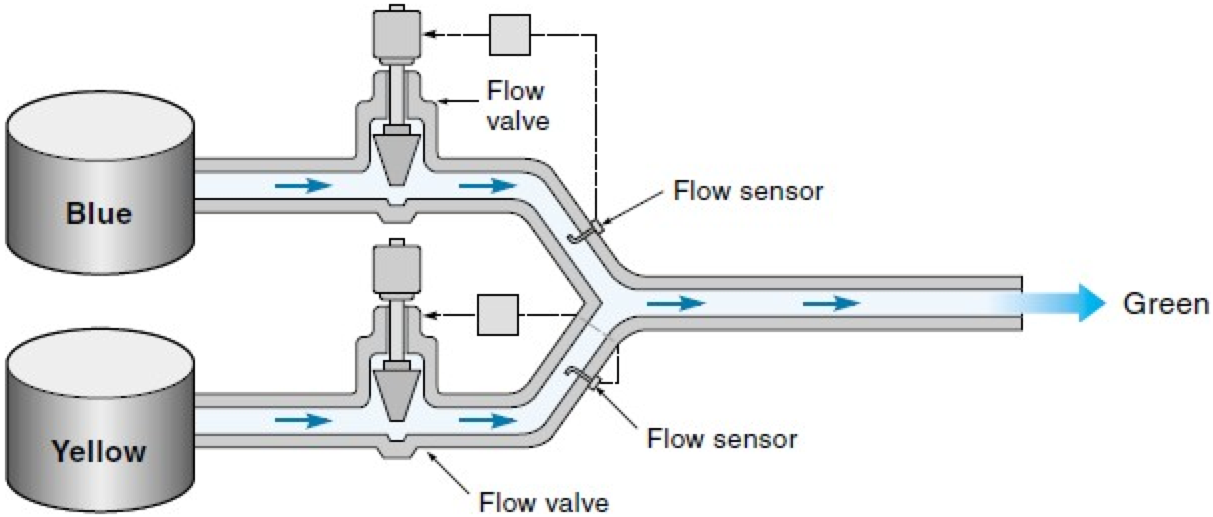
$$R(s) = \frac{A}{s^3}$$

- “Base-case” used to evaluate system response.

Control Systems Classification

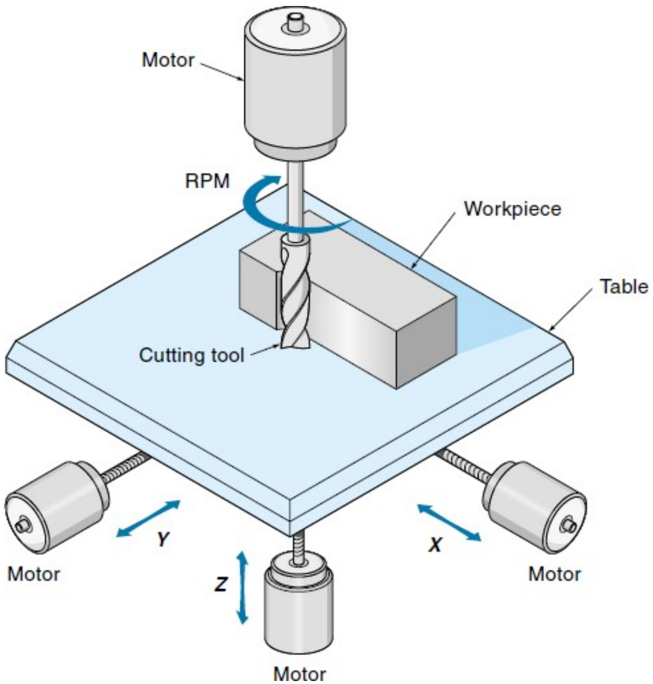
- Control systems are classified by application.
  - **Process control** usually refers to an industrial process being electronically controlled for the purpose of maintaining a uniform correct output.
  - **Motion control** refers to a system wherein things move. A servomechanism is a feedback control system that provides remote control motion of some object, such as a robot arm or a radar antenna.

Process Control Example

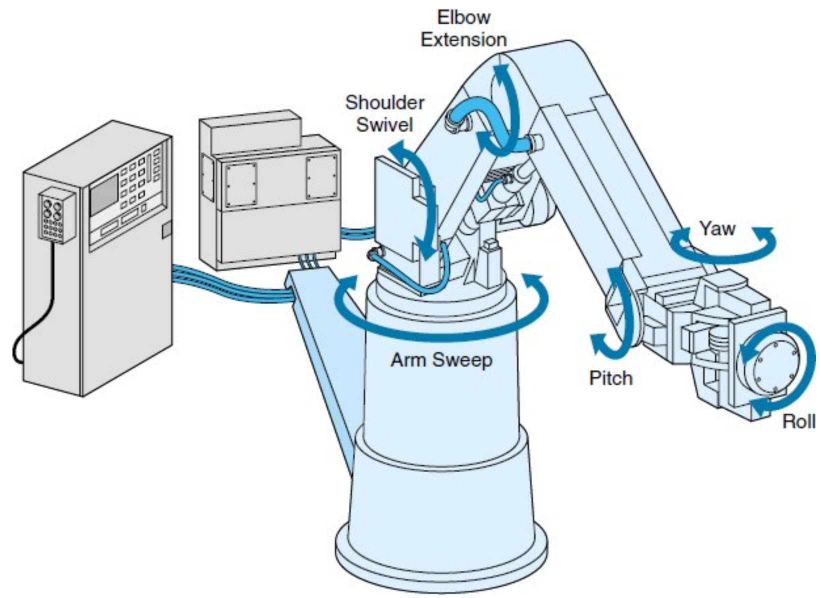


[Ref] Kilian

Motion Control Examples



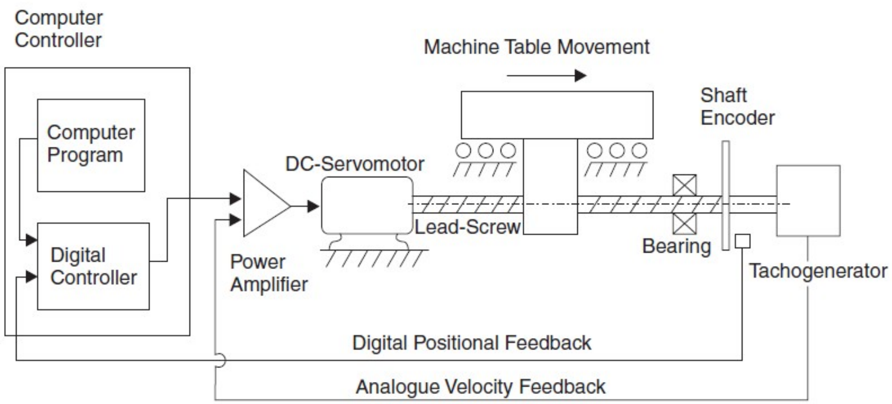
CNC Machine



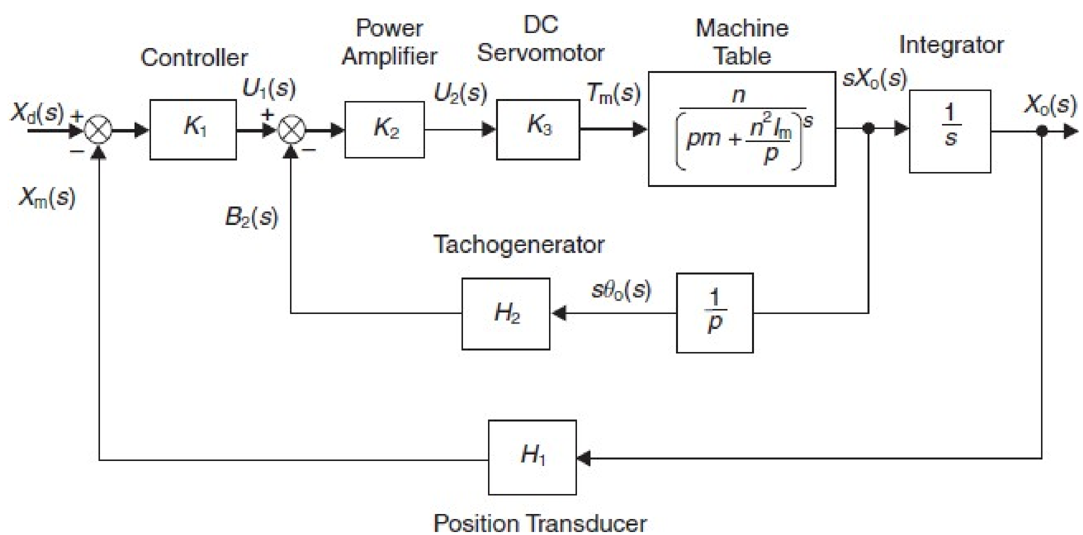
Robot Manipulator

[Ref] Kilian

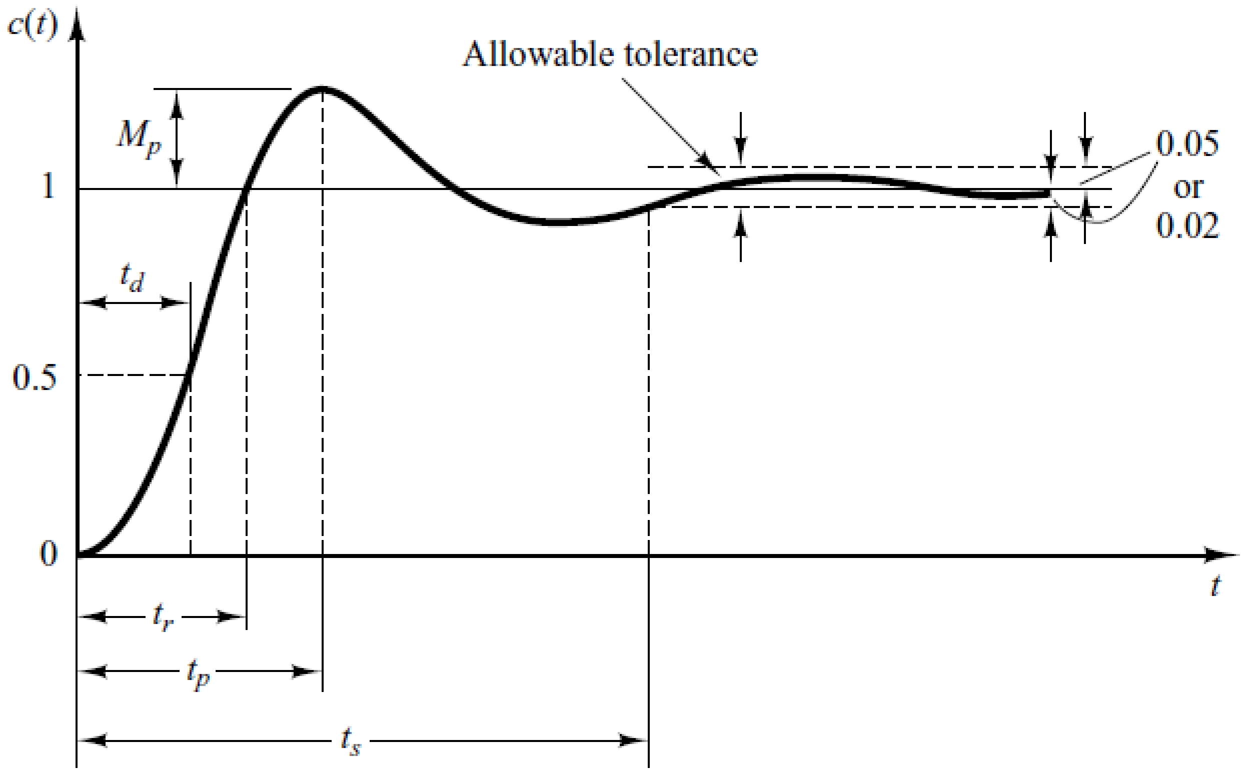
Example: CNC Positional Control



$$\frac{X_o}{X_d}(s) = \frac{\frac{1}{H_1}}{\left(\frac{p^2m+n^2I_m}{K_1K_2K_3npH_1}\right)s^2 + \left(\frac{H_2}{K_1pH_1}\right)s + 1}$$

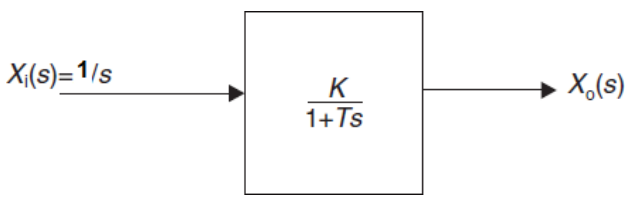


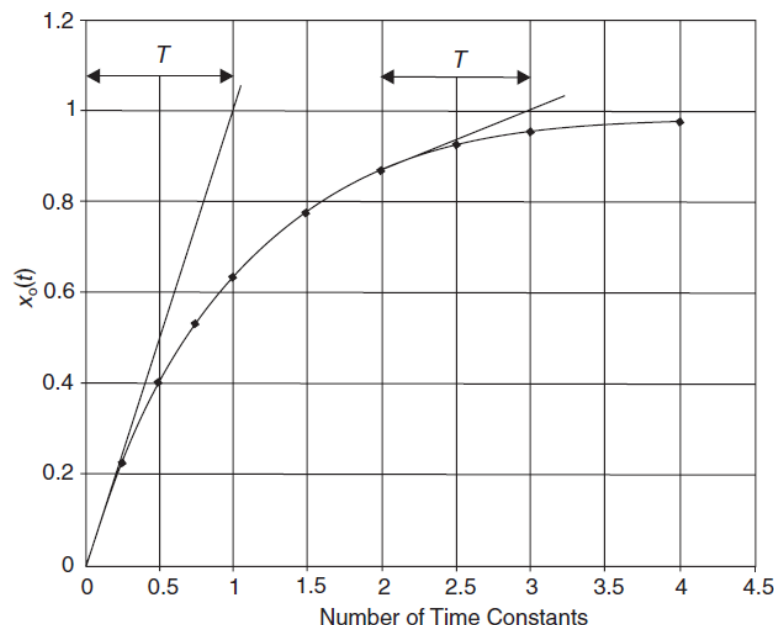
Transient and Steady State Response



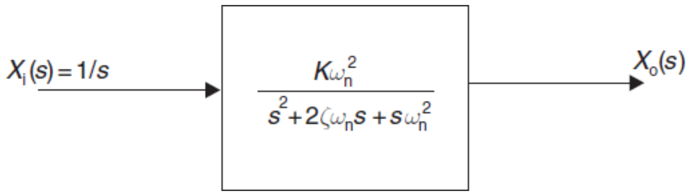
Step Response: First Order System

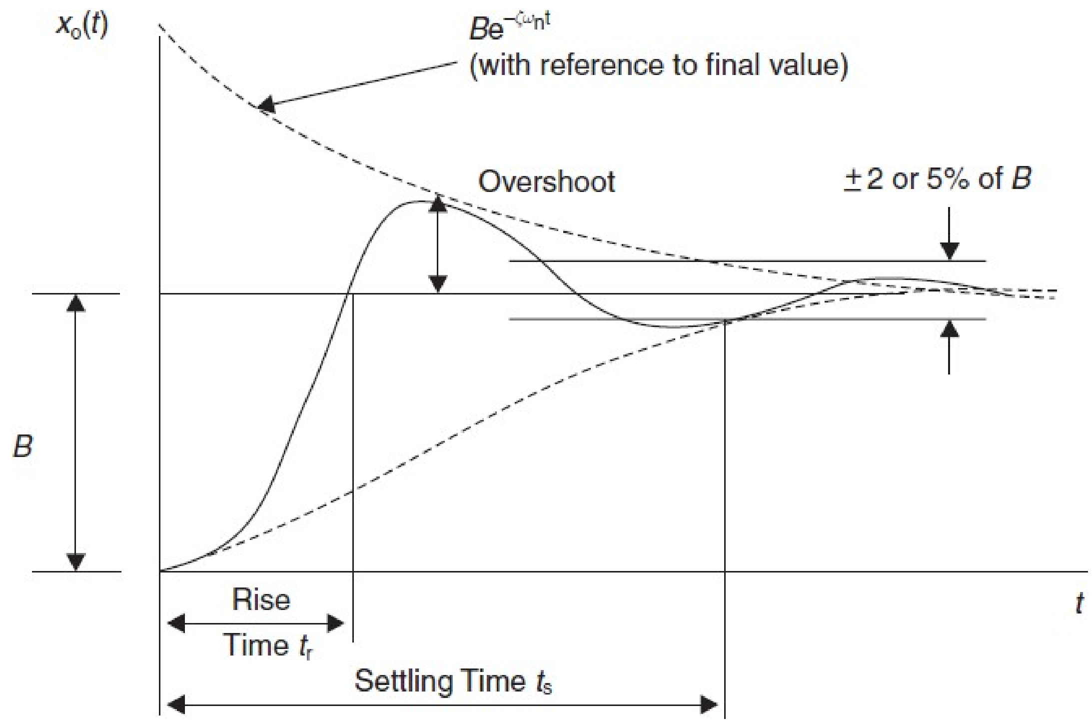
$$x_o(t) = \left(1 - e^{-t/T}\right)$$



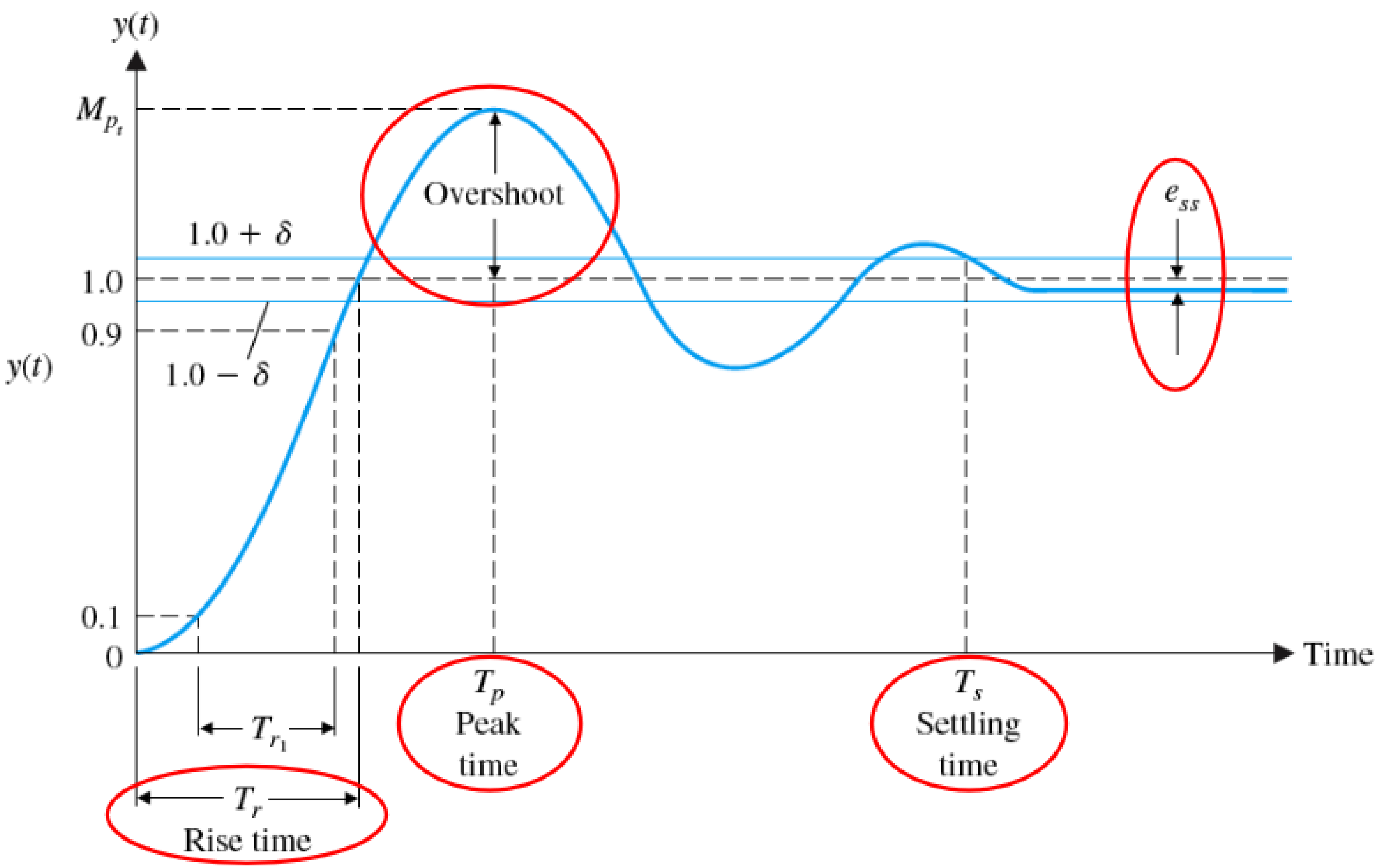


Step Response: Second Order System





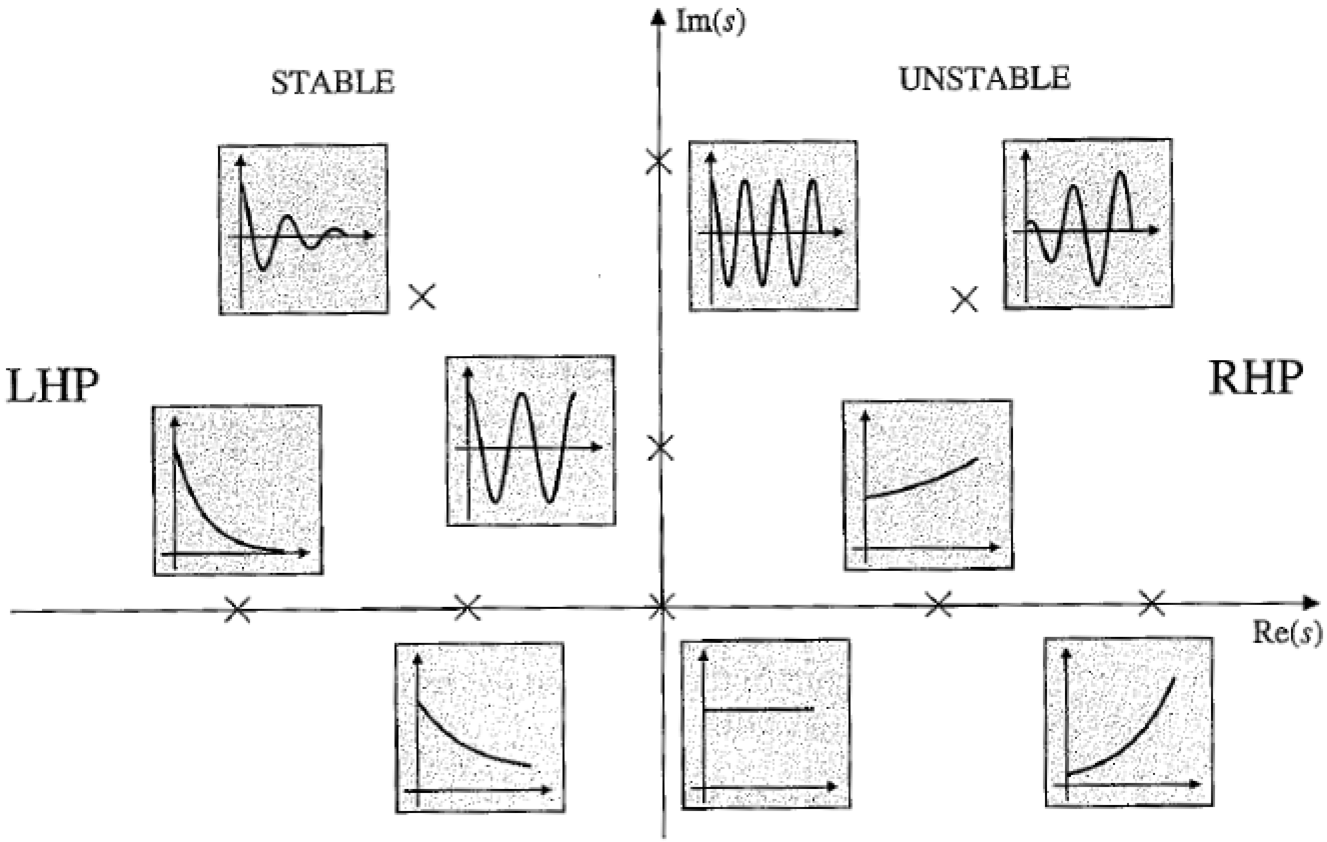
Step Response: Second Order System



Step Response: Second Order System

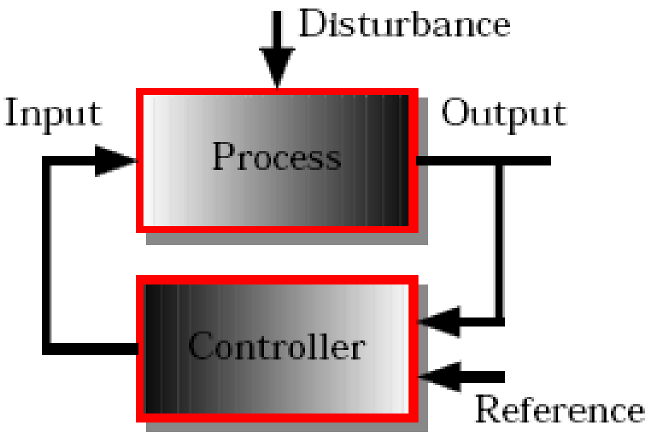
Rise time	$T_r \approx \frac{2.16\zeta + 0.60}{\omega_n}$
Peak time	$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}$
Overshoot	$M_p = 1 + e^{-\zeta\pi / \sqrt{1-\zeta^2}}$
Settling time	$T_s \approx \frac{4}{\zeta\omega_n}$

Root locus



Control Techniques / Strategies

- Classical Control
- Modern Control
- Optimal Control
- Robust Control
- Adaptive Control
- Variable Structure Control
- Intelligent Control



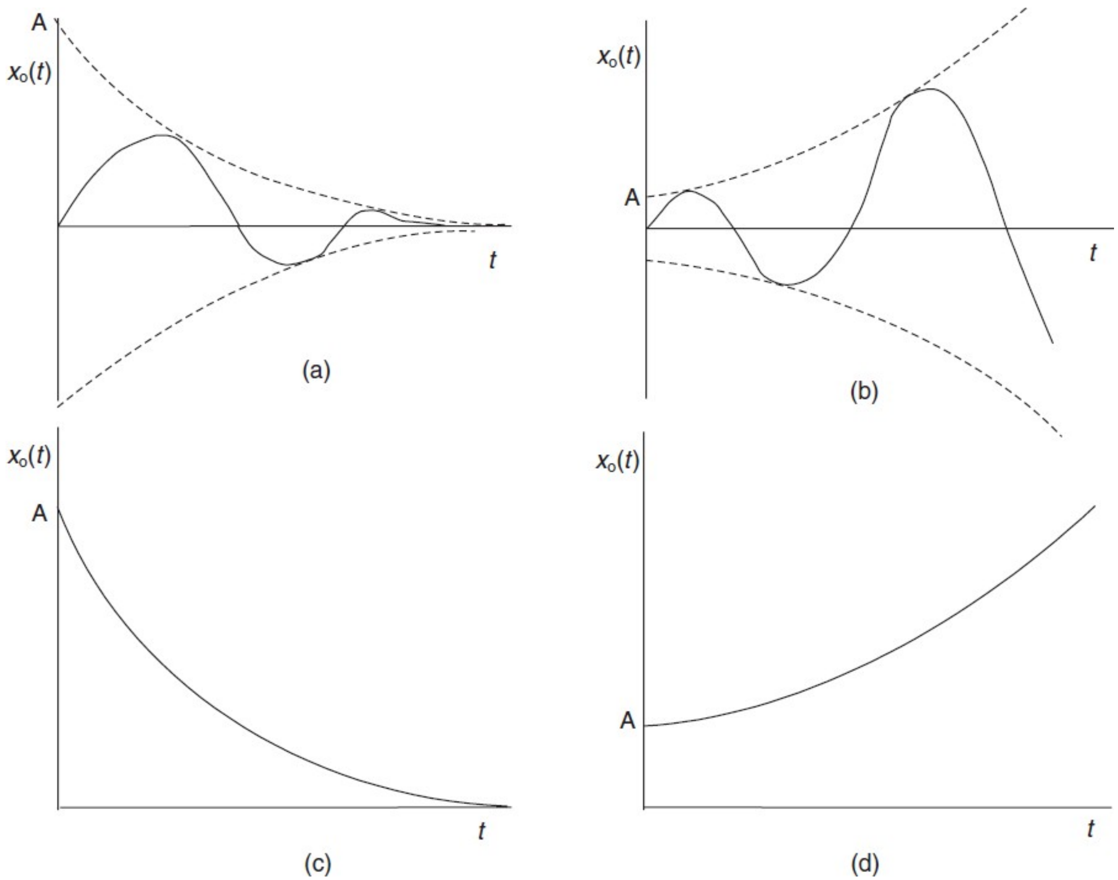
Classical Control

- Classical control design are used for SISO systems.
- Most popular concepts are:
  - Root Locus
  - Bode plots
  - Nyquist Stability
- PID is widely used in feedback systems.

Hendrik Bode (1905-1982)

Harry Nyquist (1889-1976)

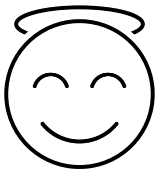
Stability



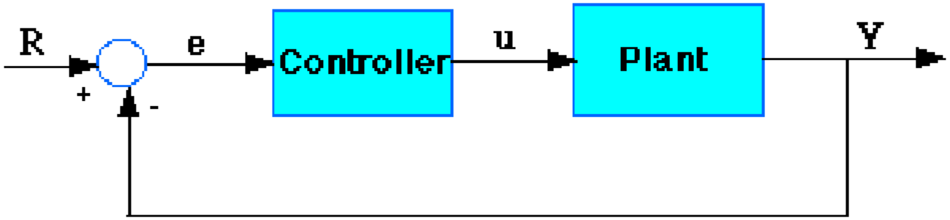
Stable **Unstable**

Classical Control: PID

- Proportional-Integral-Derivative (PID) is the most commonly used controller for SISO systems.



Engineers like PID because its simple, strong and works for most of the time.



$$e(t)=R-Y$$

$$u(t)=K_p e(t)+K_I \int e(t)dt+K_D \frac{de(t)}{dt}$$

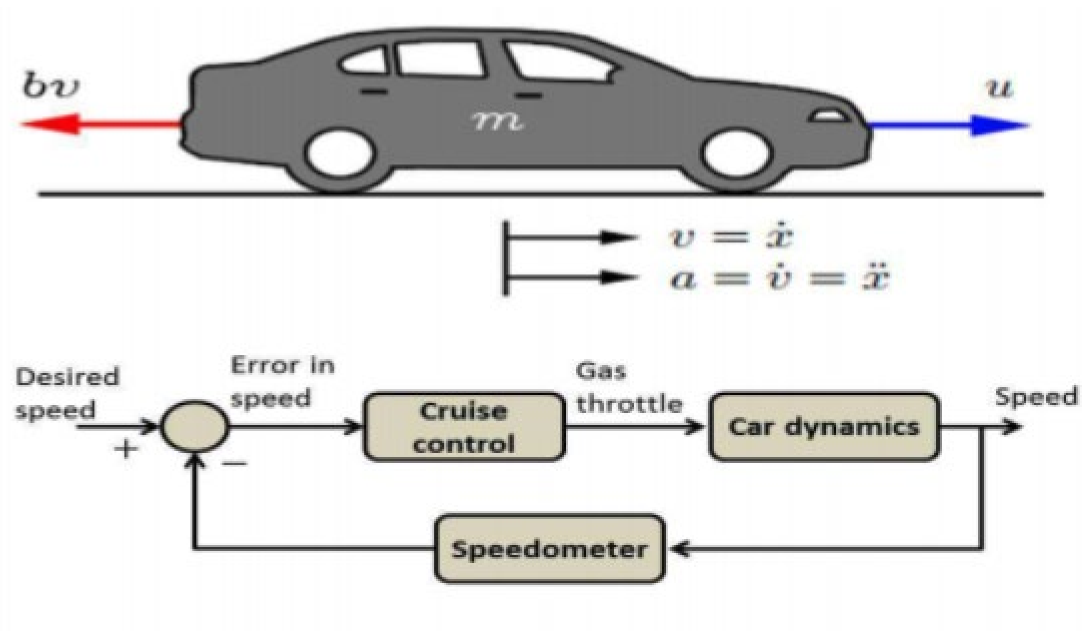
PID controller

Proportional tuning involves correcting a target proportional to the difference. Thus, the target value is never achieved because as the difference approaches zero, so too does the correction.
Integral tuning attempts to remedy this by effectively cumulating the error result from the "P" action to increase the correction factor.
Derivative tuning attempts to minimize this overshoot by slowing the correction factor applied as the target is approached.

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Decrease
Kd	Small Change	Decrease	Decrease	No Change

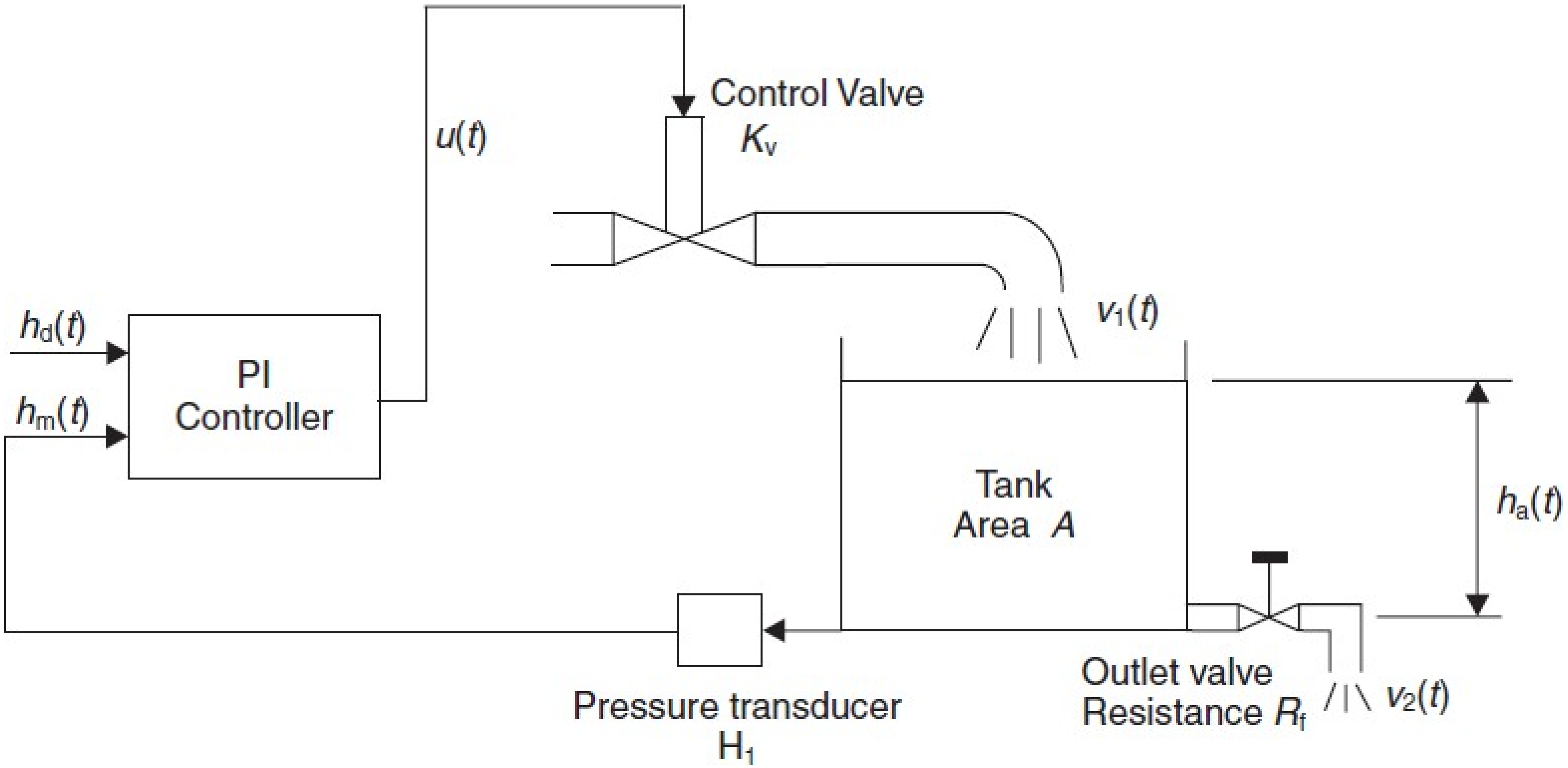
https://www.omega.co.uk/prodinfo/pid-controllers.html
https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID

Example



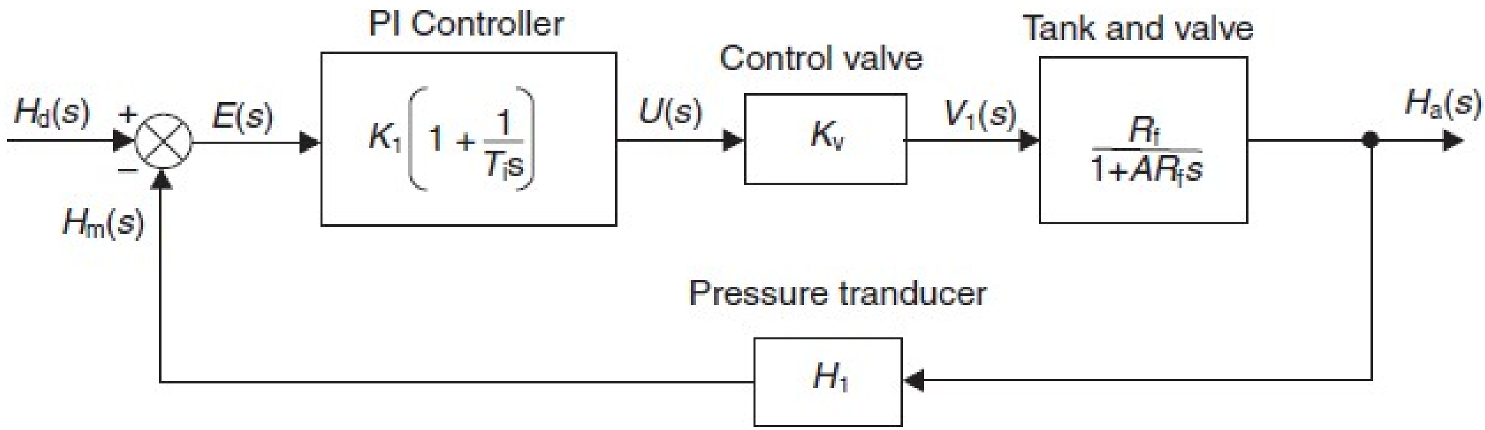
bv
m
u
$v = \dot{x}$
$a = \dot{v} = \ddot{x}$
Desired speed
Error in speed
Cruise control
Gas throttle
Car dynamics
Speed
Speedometer
+
−

Example

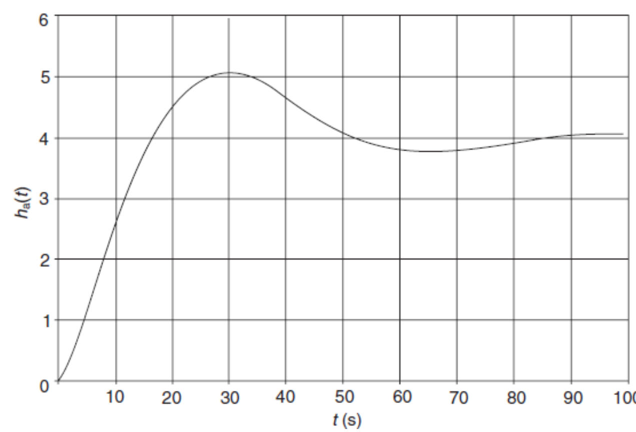


Control Valve
K_v
$u(t)$
$v_1(t)$
$h_d(t)$
$h_m(t)$
PI
Controller
Tank
Area A
$h_a(t)$
Pressure transducer
H_1
Outlet valve
Resistance R_f
$v_2(t)$

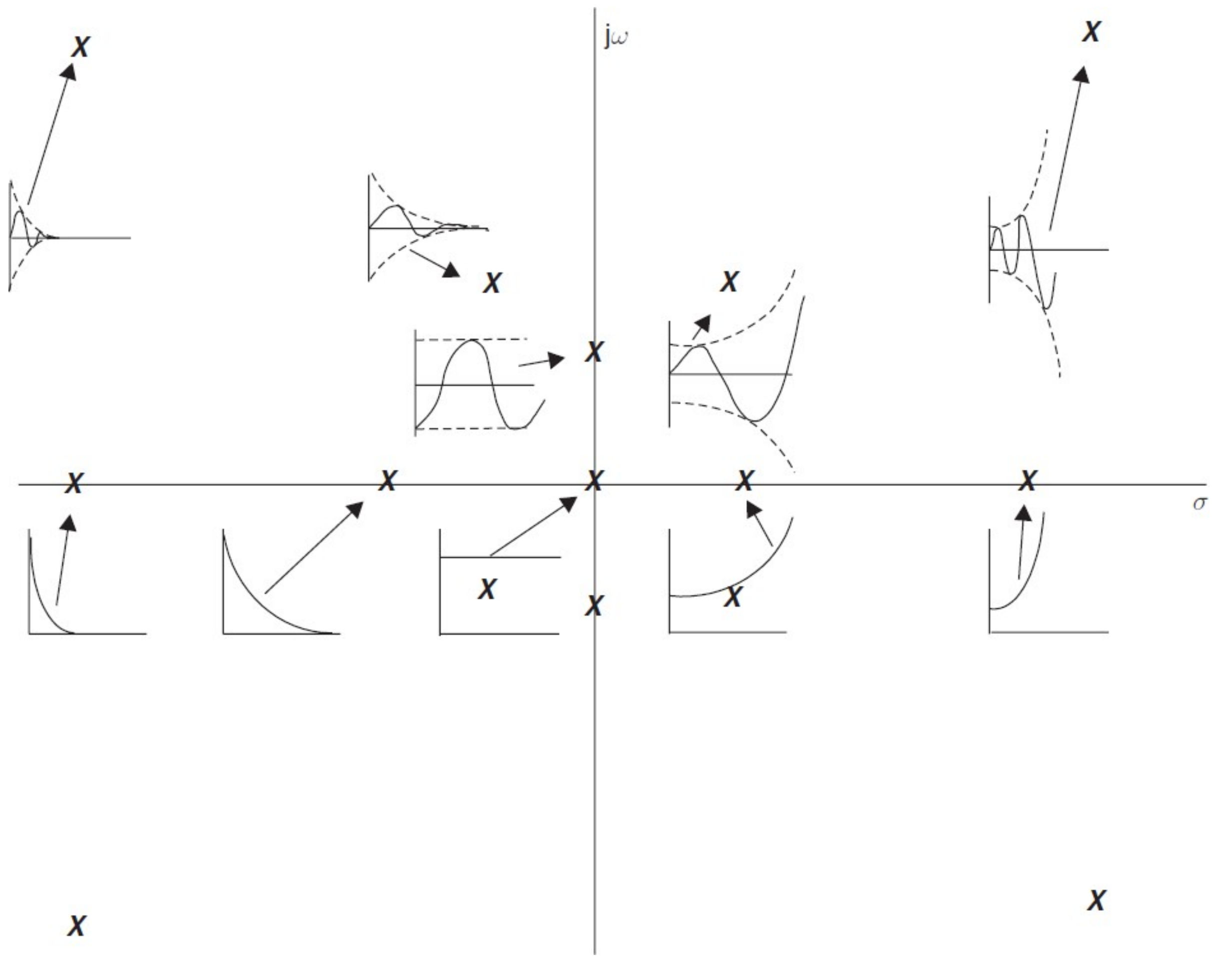
Example



$$\frac{H_a}{H_d}(s) = \frac{(1 + T_i s)}{\left(\frac{A T_i}{K_1 K_v}\right) s^2 + T_i \left(\frac{1}{K_1 K_v R_f} + 1\right) s + 1}$$

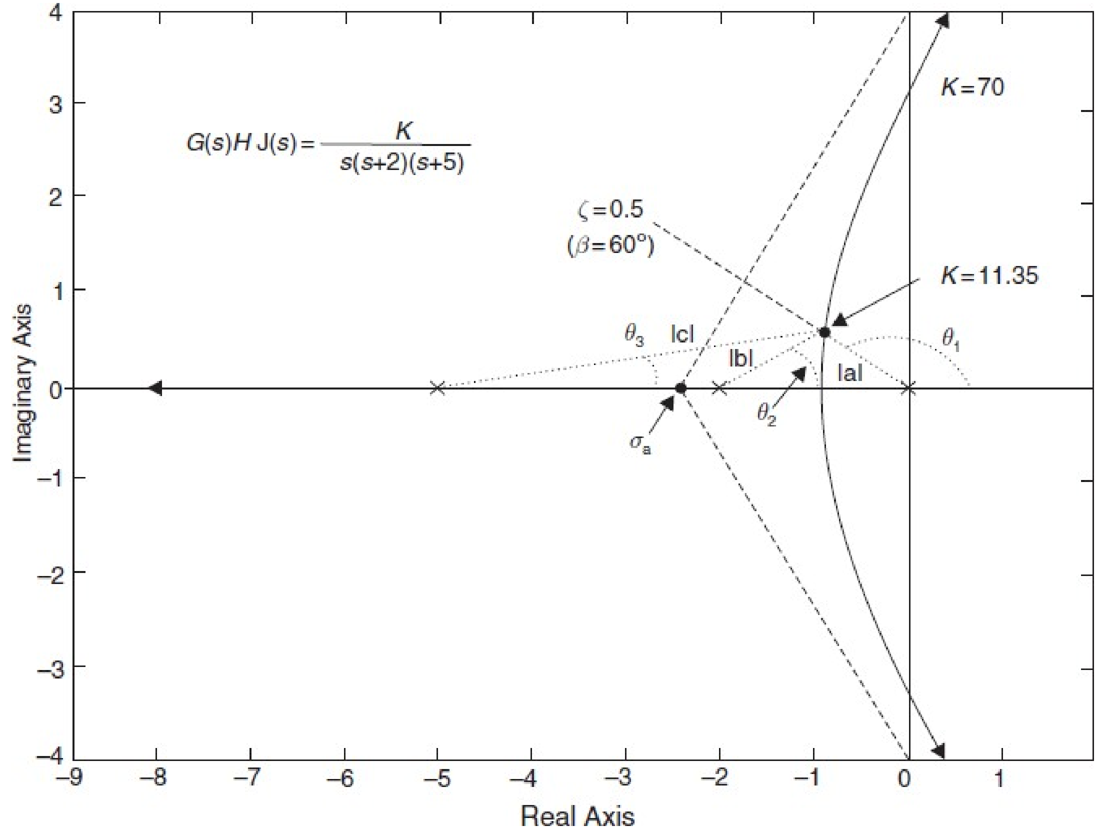


Pole location effects



Root Locus Example

$$G(s)H(s) = \frac{K}{s(s+2)(s+5)}$$



Modern Control

- **Modern control** theory utilizes the time-domain **state space** representation.
- A mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations.
- The variables are expressed as vectors and the differential and algebraic equations are written in matrix form.
- The state space representation provides a convenient and compact way to model and analyze systems with multiple inputs and outputs.

Optimal Control

- **Optimal control** is a set of differential equations describing the paths of the state and control variables that minimize a “cost function”

$$J = \int_{t_0}^{t_1} (\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u})\mathrm{d}t$$

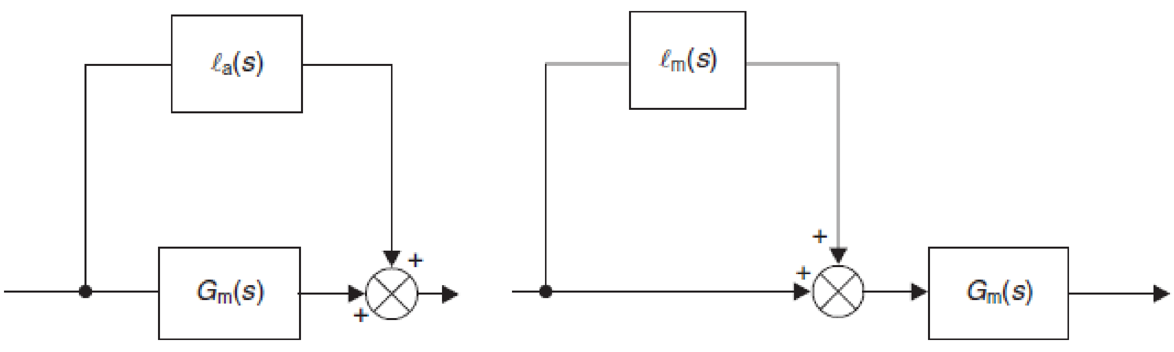
- For example, the jet thrusts of a satellite needed to bring it to desired trajectory that consume the least amount of fuel.

- Two optimal control design methods have been widely used in industrial applications, as it has been shown they can guarantee closed-loop stability.
  - **Model Predictive Control** (MPC)
  - Linear-Quadratic-Gaussian control (LQG).

Robust Control

- **Robust control** is a branch of control theory that deals with uncertainty in its approach to controller design.

- Robust control methods are designed to function properly so long as uncertain parameters or disturbances are within some set.

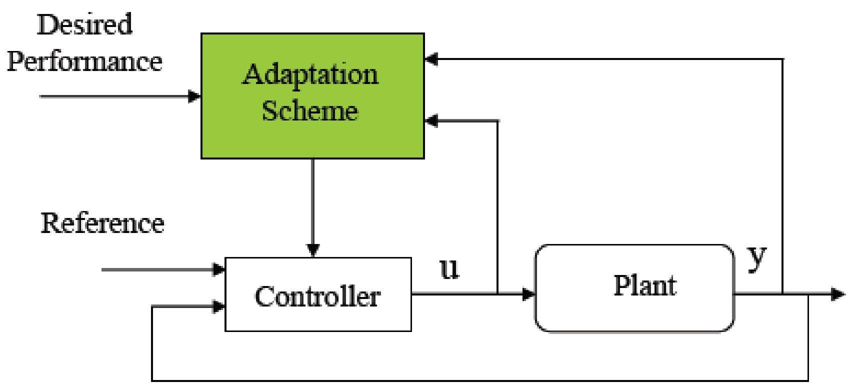


(a) Additive Mode Uncertainty (b) Multiplicative Uncertainty

Adaptive Control

- **Adaptive control** involves modifying the control law used by a controller to cope with changes in the systems' parameters

- For example, as an aircraft flies, its mass will slowly decrease as a result of fuel consumption; A control law can adapt to such changing conditions.



Variable Structure Control

- **Variable structure control**, or **VSC**, is a form of discontinuous nonlinear control.
- The method alters the dynamics of a nonlinear system by application of a high-frequency *switching control*.
- The main mode of VSC operation is sliding mode control (SMC).

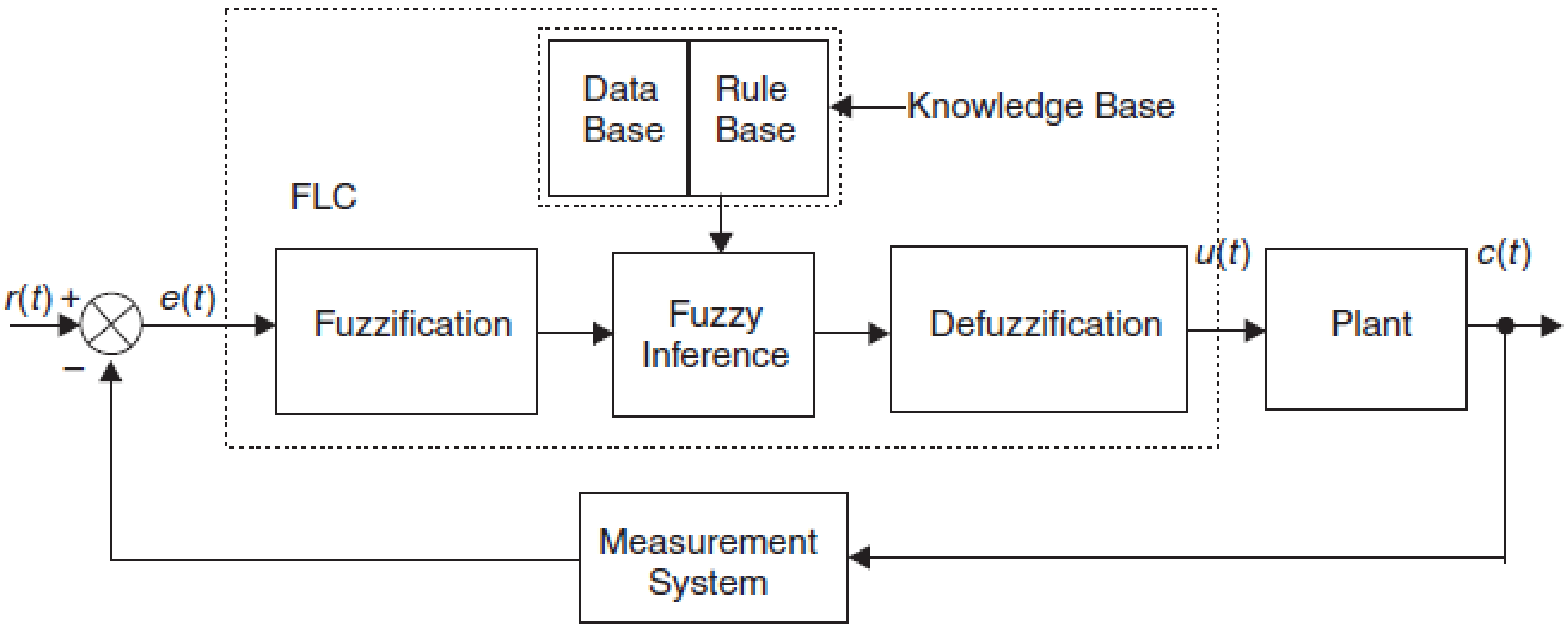
Intelligent Control

- **Intelligent Control** is usually used when the mathematical model for the plant is unavailable or highly complex.

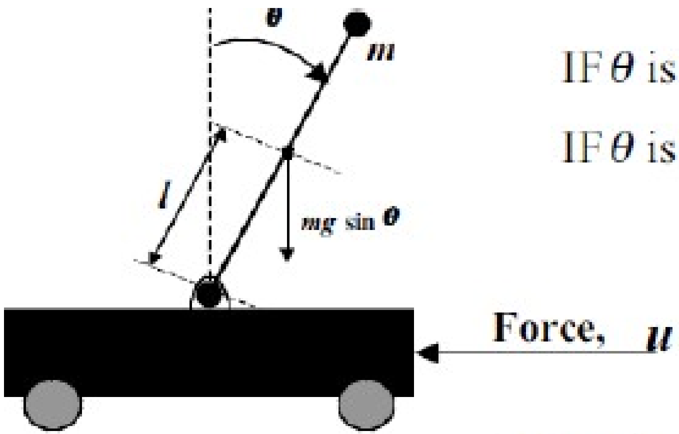
- The most commonly used intelligent controllers are:

  - Fuzzy Logic
  - Artificial Neural Networks
  - Neuro/fuzzy controllers

Fuzzy Control: Fuzzy

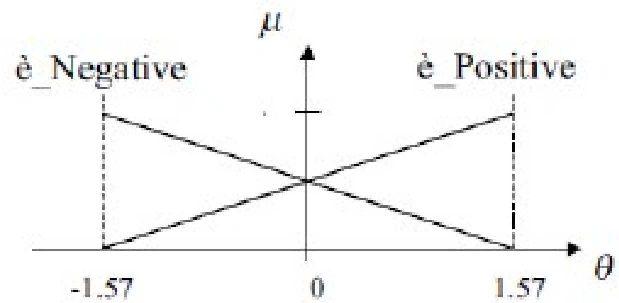


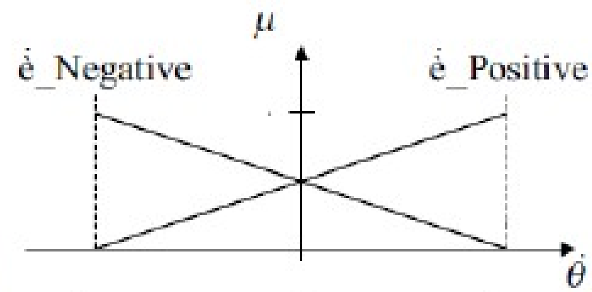
Fuzzy Control: Fuzzy

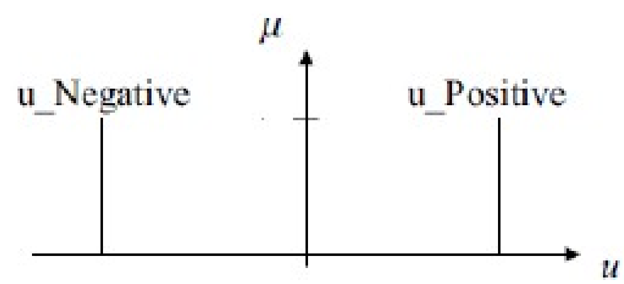


IF θ is è_Positive AND $\dot{\theta}$ is ė_Positive THEN u is u_Negative

IF θ is è_Negative AND $\dot{\theta}$ is ė_Negative THEN u is u_Positive







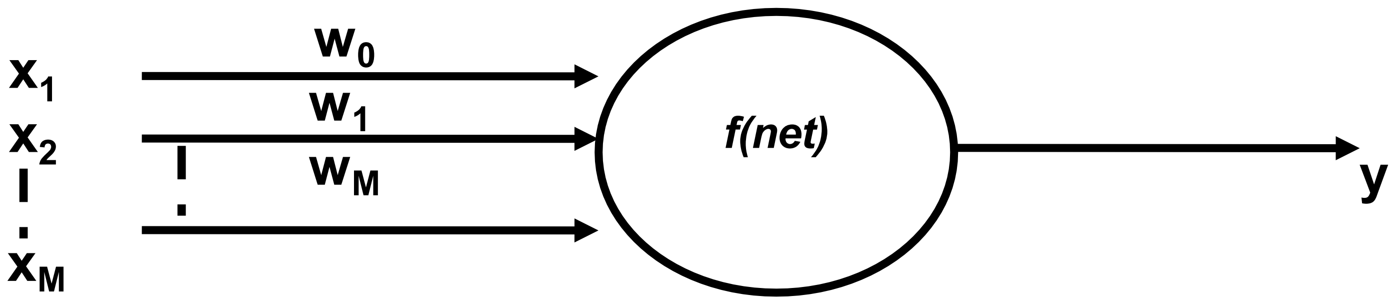
Intelligent Control: ANN

- Artificial Neural networks (ANN) are nonlinear mathematical models that are used to mimic the biological neurons in the brain.

- ANN are used as black box models to map unknown functions

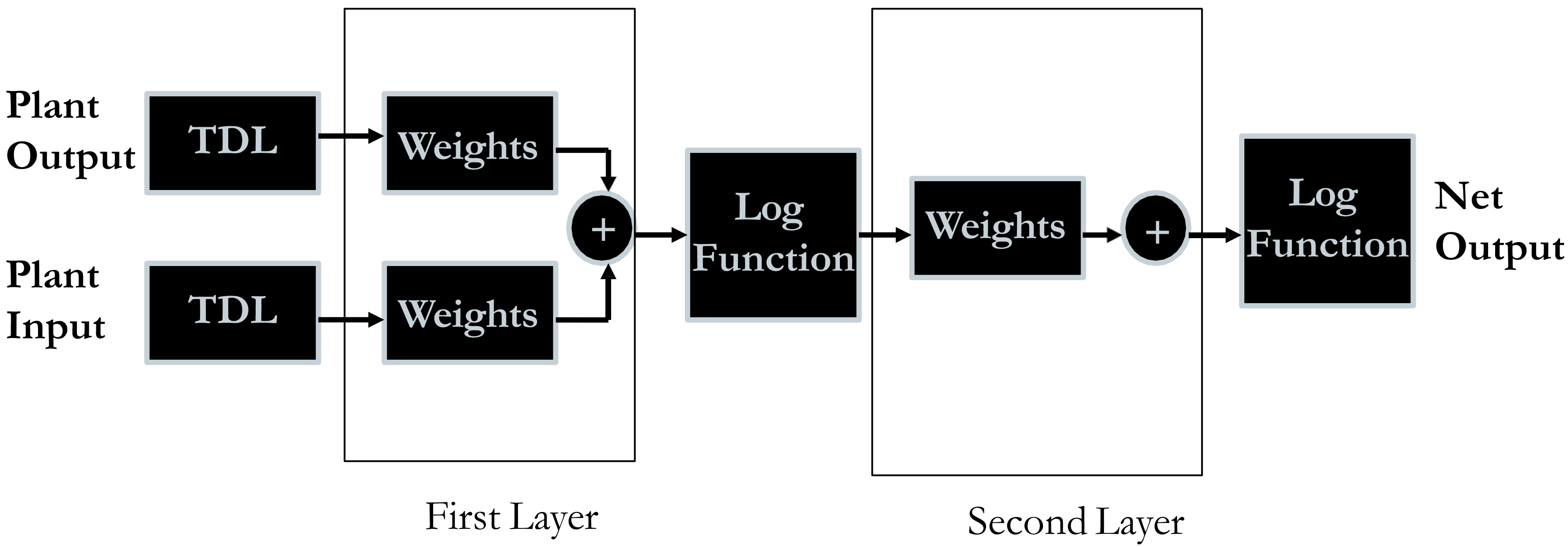
- ANN can be used for: Identification and Control

ANN: Single Neuron



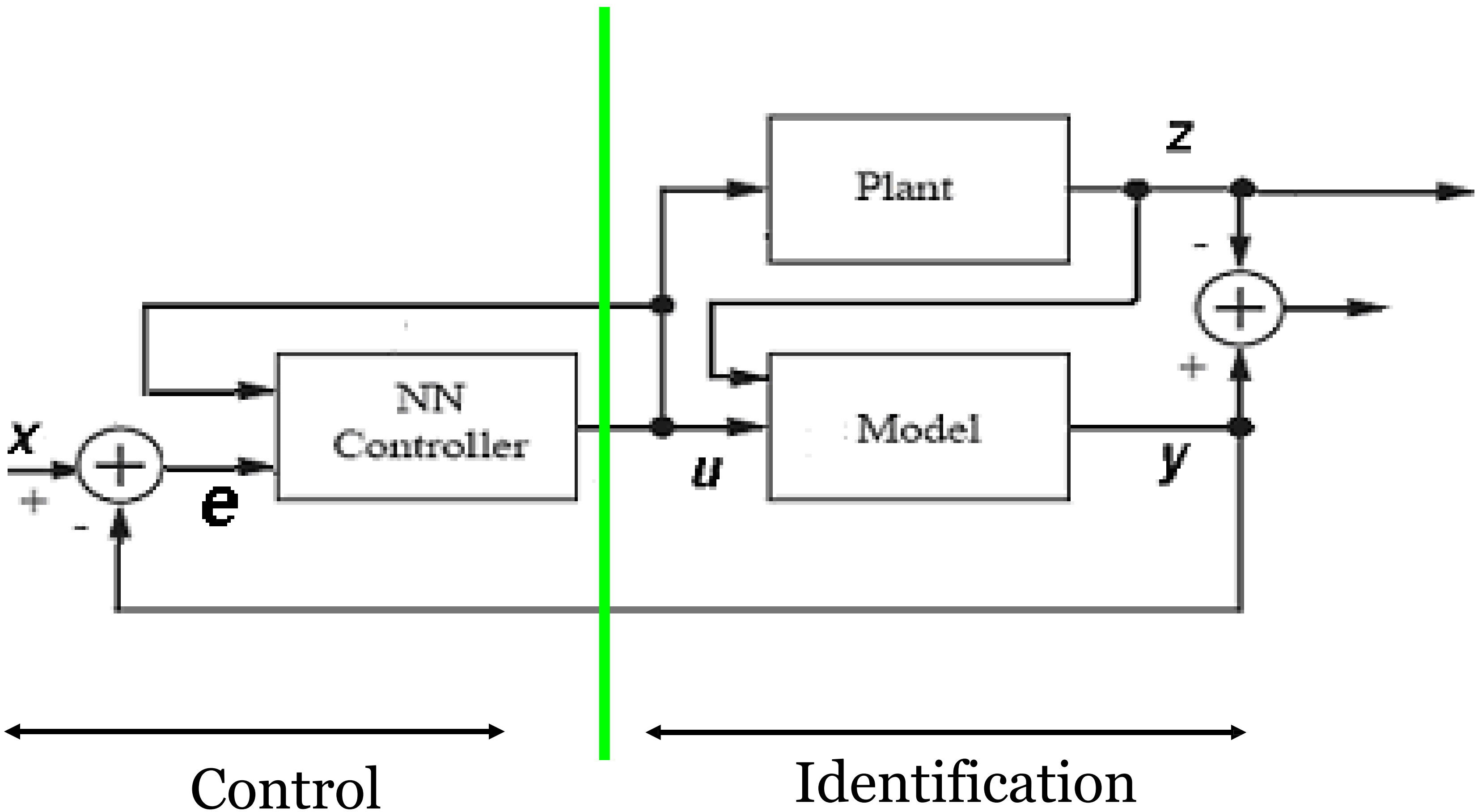
$$y = f\left(\sum_{m=1}^{M} x_m w_m \right)$$

Neural Nets



Plant Output
TDL
Weights
Plant Input
TDL
Weights
+
Log Function
Weights
+
Log Function
Net Output
First Layer
Second Layer

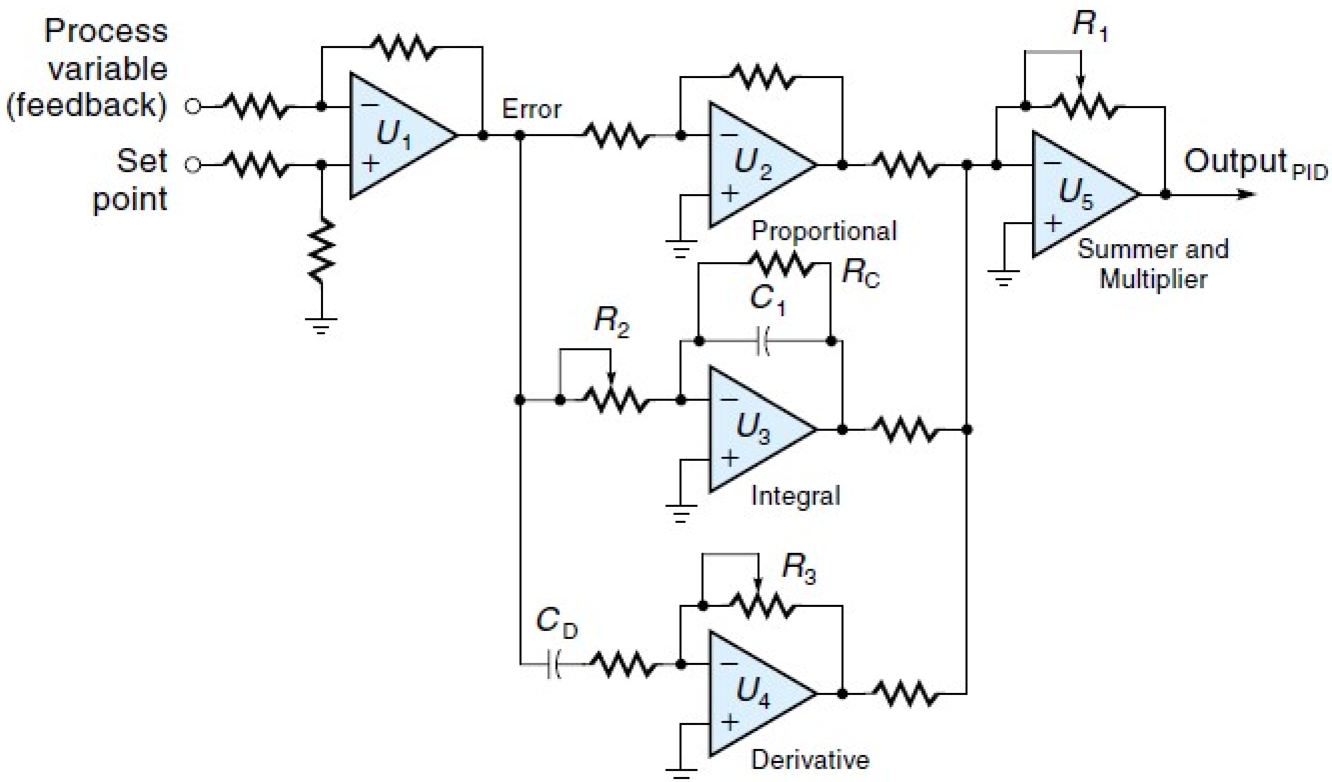
ANN: Identification and Control



Analog vs. Digital Control Systems

	Analog	Digital
Time variable	Continuous	Discrete
Time equations	Differential equations	Difference equations
Frequency transforms	Laplace	Z-Transform
Stability	Poles on LHS	Poles inside unit circle
Controller	Hardware: Op-Amps Software: None	Hardware: Microcontroller Software: Program

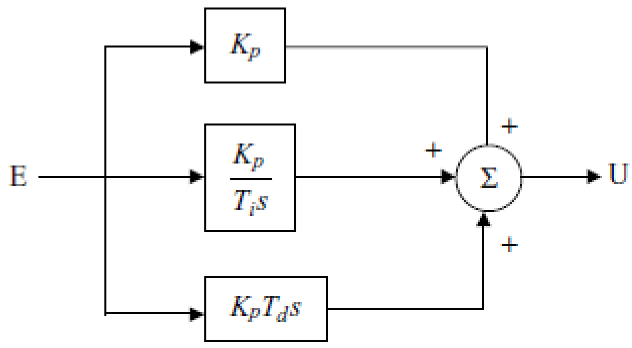
Analog PID Implementation



[Ref] Kilian

Digital PID Control

Analog



$$u(t) = K_p e(t) + K_i \int^{t} e(t)dt + K_d \frac{de(t)}{dt} + u_0,$$

$$\frac{U(s)}{E(s)} = K_p + \frac{K_p}{T_i s} + K_p T_d s.$$

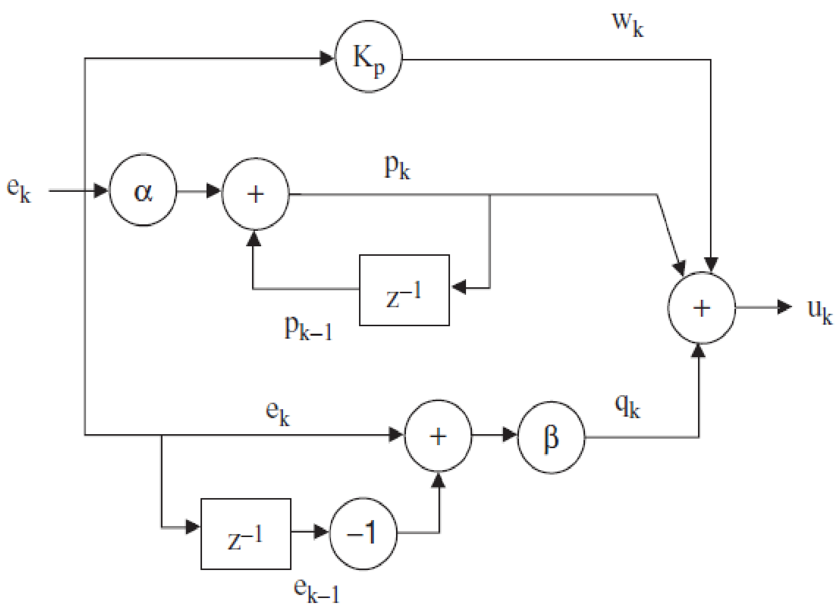
Digital

$$\frac{U(z)}{E(z)} = K_p \left[1 + \frac{T}{T_i(1 - z^{-1})} + T_d \frac{(1 - z^{-1})}{T} \right].$$

$$u(kT - T) = K_p \left[e(kT - T) + T_d \frac{e(kT - T) - e(kT - 2T)}{T} + \frac{T}{T_i} \sum_{k=1}^{n-1} e(kT) \right] + u_0.$$

Digital PID Realization

$$D(z) = K_p + \frac{K_p T}{T_i(1 - z^{-1})} + \frac{K_p T_d(1 - z^{-1})}{T}.$$

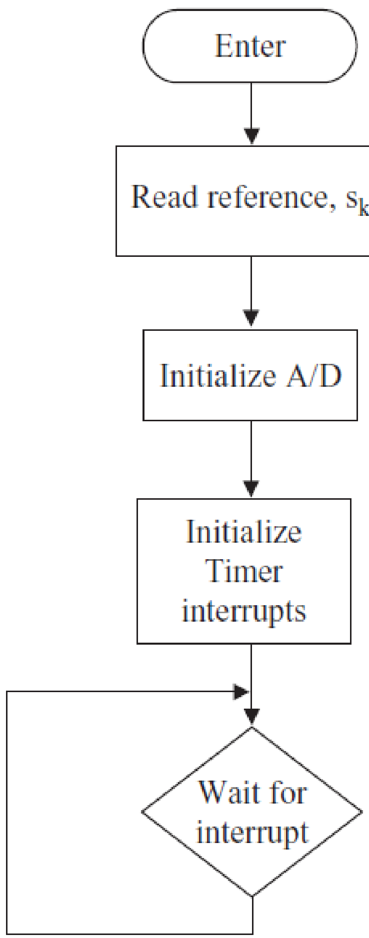


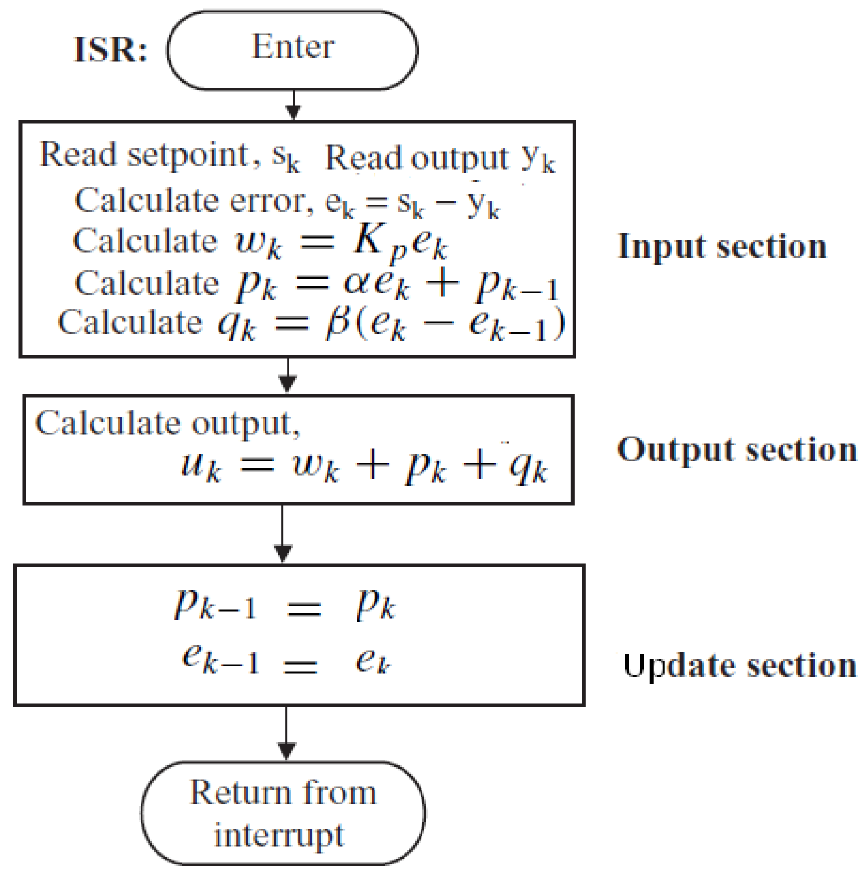
$\alpha = K_p T / T_i$
$\beta = K_p T_D / T$

$$w_k = K_p e_k.$$
$$p_k = \alpha e_k + p_{k-1}$$
$$q_k = \beta(e_k - e_{k-1})$$
$$u_k = w_k + p_k + q_k$$

Required Operations:
- **Multiplication**
- **Addition**
- **Delay**

Discrete PID Implementation





References

- **Advanced Control Engineering** (Chapter 8: State Space Methods for Control System Design) by Roland **Burns** 2001
- **Modern Control Engineering** (Chapter 10: State Space Design Methods) by **Paraskevopoulos** 2002
- **Modern Control Engineering** (Chapters 9 and 10 Control System Analysis and Design in State Space) by **Ogata** 5th edition 2010
- **Feedback Systems: An Introduction to Scientists and Engineers** (Chapter 8: System Models) by **Astrom** and **Muray** 2009
- **Modern Control Technology: Components and Systems** by **Kilian** 2nd edition