Lecture 1

- Introduction - Course mechanics
- History
- Modern control engineering

Introduction - Course Mechanics

- What this course is about?
- Prerequisites & course place in the curriculum
- Course mechanics
- Outline and topics
- Your instructor

What this course is about?

- Embedded computing is becoming ubiquitous
- Need to process sensor data and influence physical world. This is control and knowing its main concepts is important.
- Much of control theory is esoteric and difficult
- 90% of the real world applications are based on 10% of the existing control methods and theory
- The course is about these 10%

Prerequisites and course place

- Prerequisites:
  - Linear algebra: EE263, Math 103
  - Systems and control: EE102, ENGR 105, ENGR 205
- Helpful
  - Matlab
  - Modeling and simulation
  - Optimization
  - Application fields
  - Some control theory good, but not assumed.
- Learn more advanced control theory in :
  - ENGR 207, ENGR 209, and ENGR 210

Course Mechanics

- Descriptive in addition to math and theory
- Grading
    - 25% Homework Assignments (4 at all)
    - 35% Midterm Project
    - 40% Final Project
- Notes at www.stanford.edu/class/ee392M/
- Reference texts
    - *Control System Design*, Astrom, posted as PDF
    - *Feedback Control of Dynamic Systems*, Fourth Edition, Franklin, Powell, Emami-Naeini, Prentice Hall, 2002
    - *Control System Design*, Goodwin, Graebe, Salgado, Prentice Hall, 2001

Outline and topics

Lectures - Mondays & Fridays

Assignments - Fridays, due on Friday

Lecture topics

Basic

1. Introduction and history
2. Modeling and simulation
3. Control engineering problems
4. PID control
5. Feedforward
6. SISO loop analysis
7. SISO system design

Advanced

8. Model identification
9. Processes with deadtime, IMC
10. Controller tuning
11. Multivariable control - optimization
12. Multivariable optimal program
13. MPC - receding horizon control

Breadth

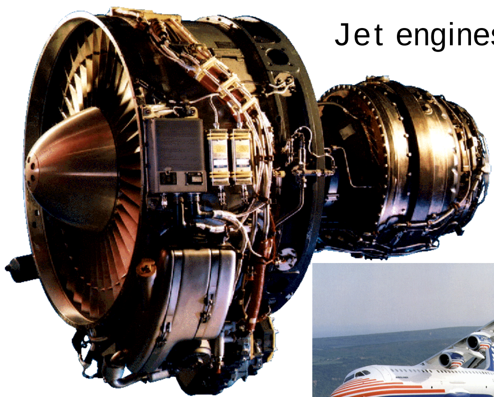
14. Handling nonlinearity
15. System health management
16. Overview of advanced topics

Who is your instructor?

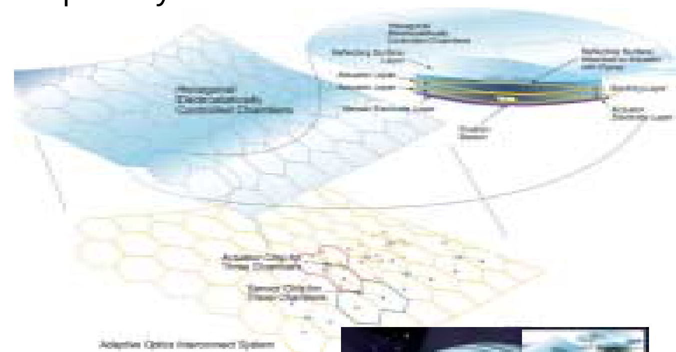
- Dimitry Gorinevsky
- Consulting faculty (EE)
- Honeywell Labs
  - Minneapolis
  - Cupertino
- Control applications across many industries
- PhD from Moscow University
  - Moscow $\rightarrow$ Munich $\rightarrow$ Toronto $\rightarrow$ Vancouver $\rightarrow$ Palo Alto

Some stuff I worked on

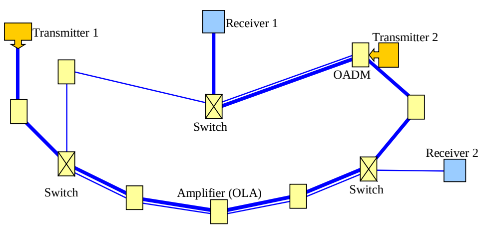
Jet engines



Space systems



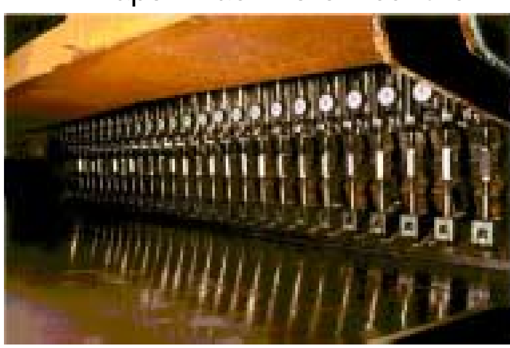
WDM networks

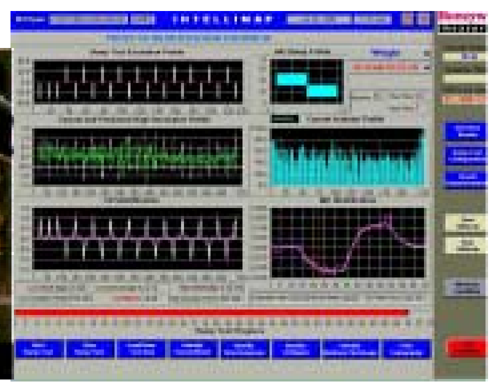


Powertrain control

Walking machine

Paper machine CD control





Lecture 1 - Control History

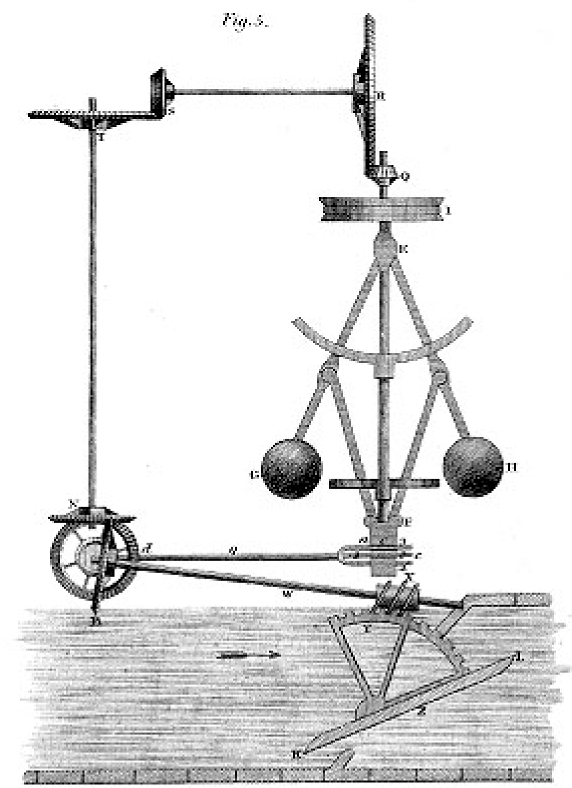
- Watt’s governor
- Thermostat
- Feedback Amplifier
- Missile range control
- TCP/IP
- DCS

Why bother about the history?

- Trying to guess, where the trend goes
- Many of the control techniques that are talked about are there for historical reasons mostly. Need to understand that.

1788 Watt’s Flyball Governor

- Watt’s Steam Engine
- Newcomen’s steam engine (1712) had limited success
- Beginning of systems engineering
- Watt’s systems engineering add-on started the Industrial Revolution
- Analysis of James Clark Maxwell (1868)
- Vyshnegradsky (1877)



From the 1832 *Edinburgh Encyclopaedia*

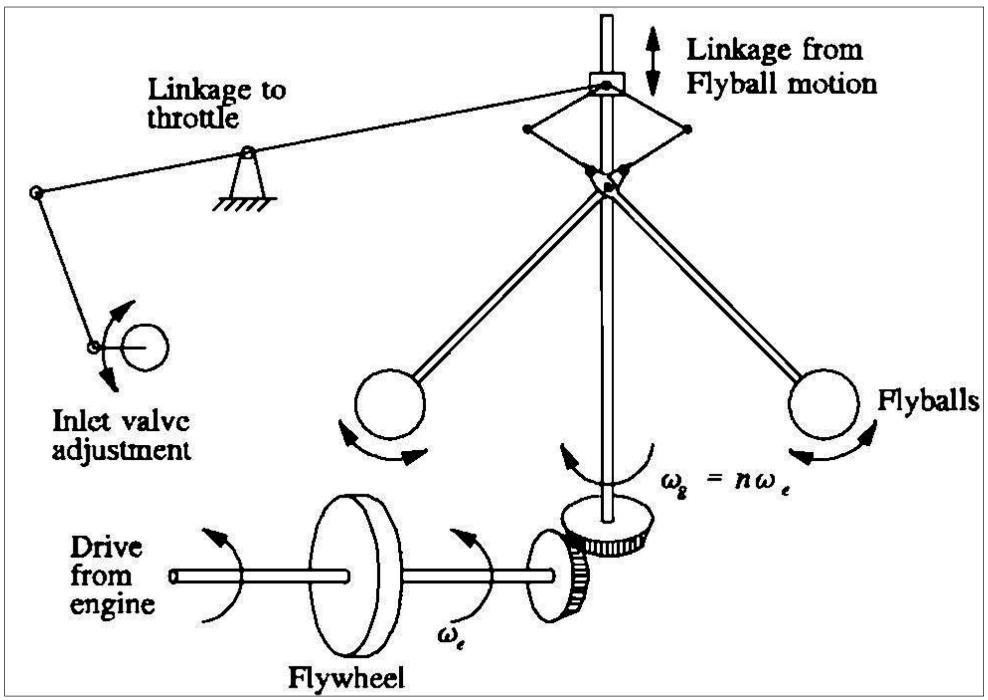
Rubs

- Mechanical technology use was extended from power to regulation
- It worked and improved reliability of steam engines significantly by automating operator's function
- Analysis was done much later (some 100 years) - this is typical!
- Parallel discovery of major theoretical approaches

Watt’s governor

- Analysis of James Clark Maxwell (1868)

$$ml\ddot{\phi} = l\left(m\omega_G^2 l \sin\phi\cos\phi - mg\sin\phi - b\dot{\phi}\right)$$



$$J\dot{\omega}_E = k\cos\phi - T_L$$

$$\omega_G = n\omega_E$$

- Linearization

$$\phi = \phi_0 + x \qquad x << 1$$

$$\omega_E = \omega_0 + y \qquad y << 1$$

$$\dddot{y} + a_1\ddot{y} + a_2\dot{y} + a_3 y = 0$$

Watt's governor

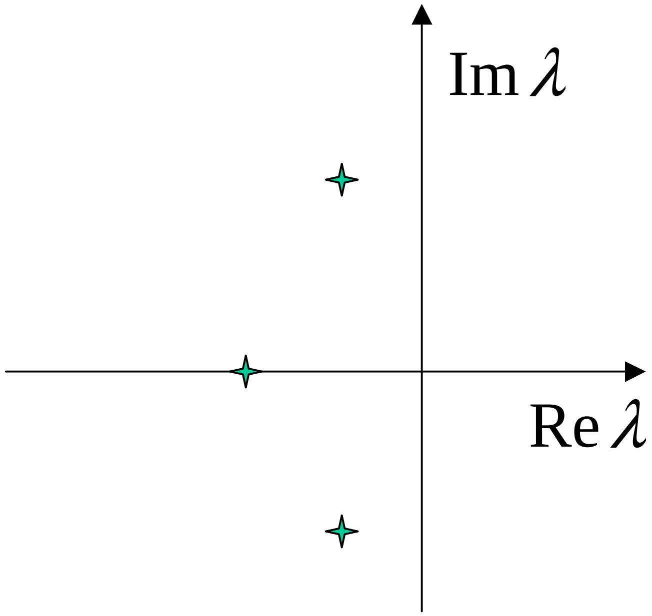
$$\dddot{y} + a_1\ddot{y} + a_2\dot{y} + a_3 y = 0$$

Characteristic equation: $y = e^{\lambda t}$

$$\lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 = 0$$

Stability condition:

$$\operatorname{Re}\lambda_k < 0, \quad (k = 1,2,3)$$



- Gist:
  - Model; P feedback control; linearization; LHP poles
- All still valid

1885 Thermostat

- 1885 Al Butz invented damper-flapper
  - bimetal plate (sensor/control)
  - motor to move the furnace damper)
- Started a company that became Honeywell in 1927

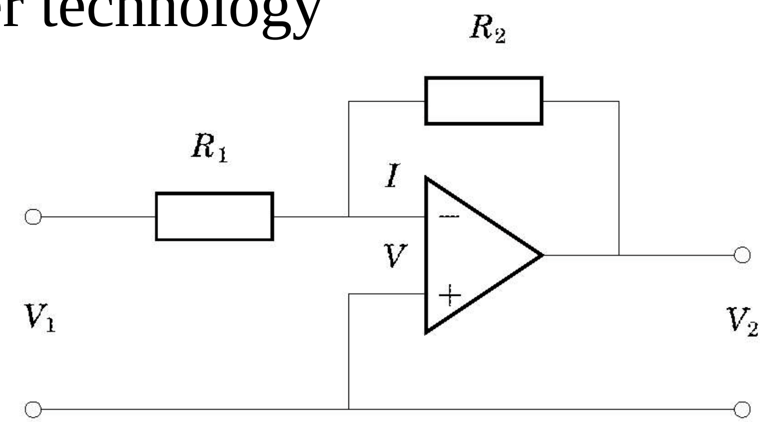
- Thermostat switching on makes the main motor shaft to turn one-half revolution opening the furnace's air damper.
- Thermostat switching off makes the motor to turn another half revolution, closing the damper and damping the fire.
- On-off control based on threshold

Rubs

- Use of emerging electrical system technology
- Significant market for heating regulation (especially in Minnesota and Wisconsin)
- Increased comfort and fuel savings passed to the customer - customer value proposition
- Integrated control device with an actuator. Add-on device installed with existing heating systems

1930s Feedback Amplifier

- Signal amplification in first telecom systems (telephone) Analog vacuum tube amplifier technology
- Feedback concept



$$\frac{V_1 - V}{R_1} = \frac{V - V_2}{R_2}$$

$$V_2 = GV$$

$$\frac{V_1}{V_2} = R_1\left[\frac{1}{R_2} - \frac{1}{G}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)\right] = -\frac{R_1}{R_2}\left[1 - \frac{1}{G}\left(1 + \frac{R_2}{R_1}\right)\right]$$

- Bode’s analysis of the transients in the amplifiers (1940)

Feedback Amplifier - Rubs

- Electronic systems technology
- Large communication market
- Useful properties of large gain feedback realized: linearization, error insensitivity
- Conceptual step. It was initially unclear why the feedback loop would work dynamically, why would it not grow unstable.

1940s WWII Military Applications

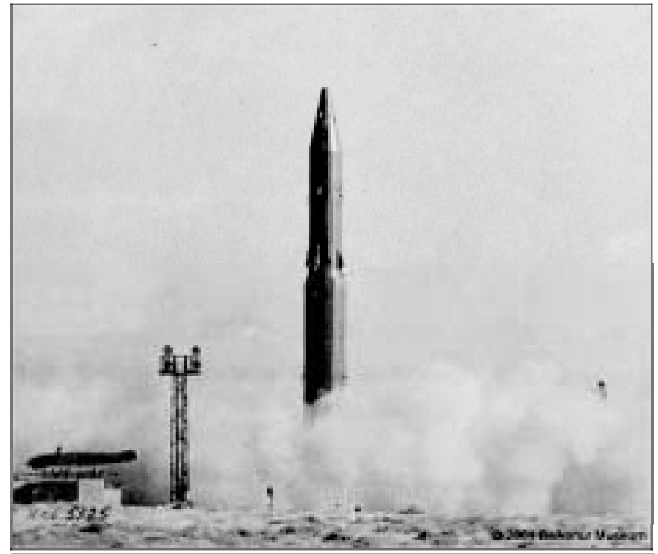
- Sperry Gyroscope Company – flight instruments – later bought by Honeywell to become Honeywell aerospace control business.
- Servosystem – gun pointing, ship steering, using gyro
- Norden bombsight – Honeywell C-1 autopilot - over 110,000 manufactured.
- Concepts – electromechanical feedback, PID control.
- Nyquist, servomechanism, transfer function analysis,

Autopilot - Rubs

- Enabled by the navigation technology - Sperry gyro
- Honeywell got the autopilot contract because of its control system expertise – in thermostats
- Emergence of cross-application control engineering technology and control business specialization.

1960s - Rocket science

- SS-7 missile range control
  - through the main engine cutoff time.



USSR R-16/8K64/SS-7/Saddler
Copyright © 2001 RussianSpaceWeb.com
http://www.russianspaceweb.com/r16.html

- Range

$$r = F\left(\Delta V_x, \Delta V_y, \Delta X, \Delta Y\right)$$

- Range Error

$$\delta r(t) = f_1 \Delta V_x(t) + f_2 \Delta V_y(t) + f_3 \Delta X(t) + f_4 \Delta Y(t)$$

- Algorithm:
  - track $\delta r(t)$, cut the engine off at T when $\delta r(T) = 0$

Missile range control - Rubs

- Nominal trajectory needs to be pre-computed and optimized
- Need to have an accurate inertial navigation system to estimate the speed and coordinates
- Need to have feedback control that keeps the missile close to the nominal trajectory (guidance and flight control system)
- $f_1,\ f_2,\ f_3,\ f_4,$ and f_T must be pre-computed
- Need to have an on-board device continuously computing

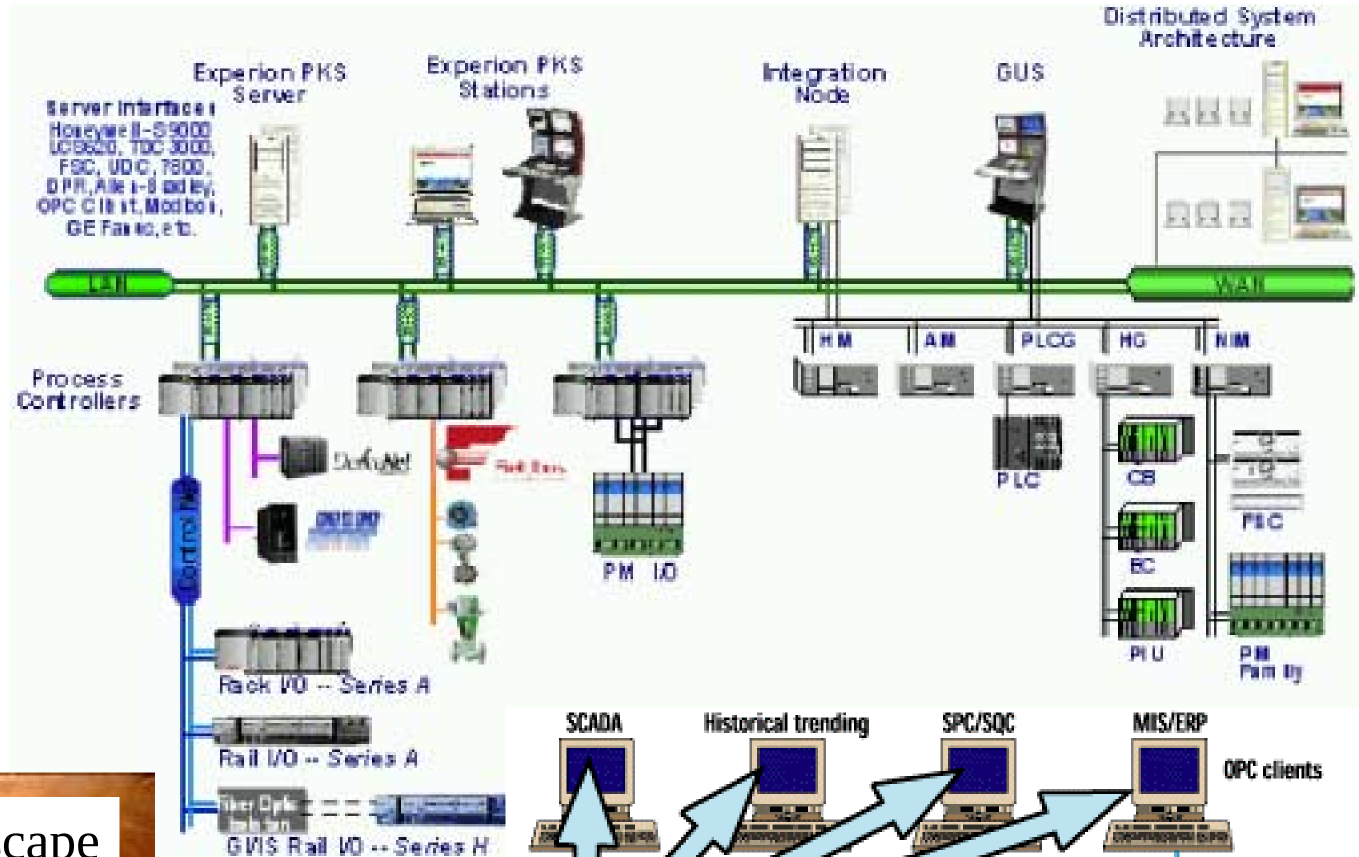
$$\delta r(t) = f_1 \Delta V_x(t) + f_2 \Delta V_y(t) + f_3 \Delta X(t) + f_4 \Delta Y(t)$$

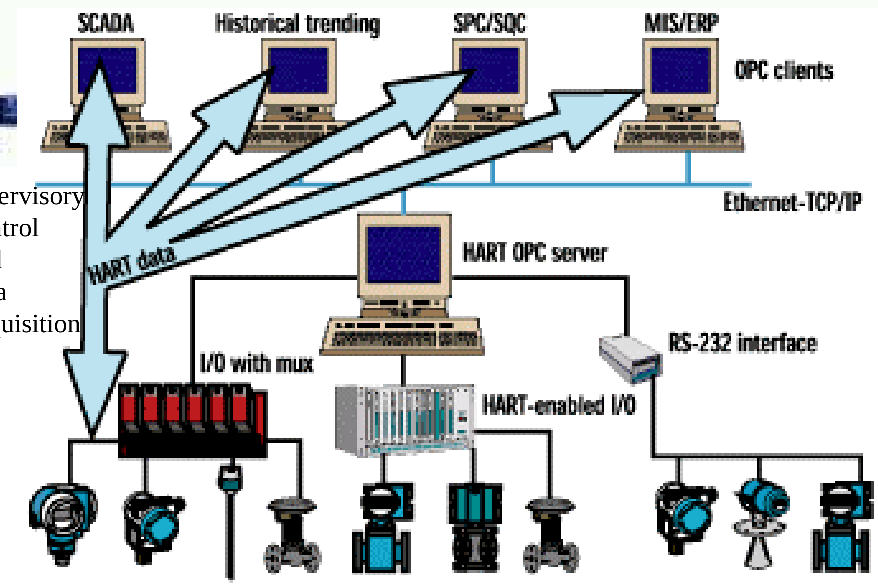
1975 - Distributed Control System

- 1963 - Direct digital control was introduced at a petrochemical plant. (Texaco)
- 1970 - PLC's were introduced on the market.
- 1975 - First DCS was introduced by Honeywell
- PID control, flexible software
- Networked control system, configuration tuning and access from one UI station
- Auto-tuning technology

DCS example

Honeywell Experion PKS



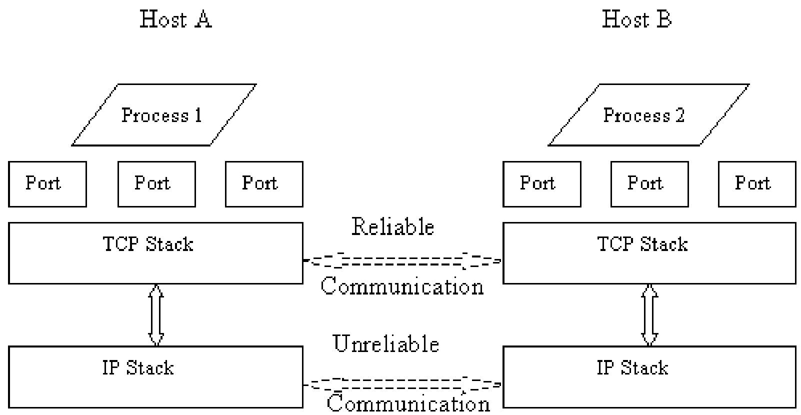


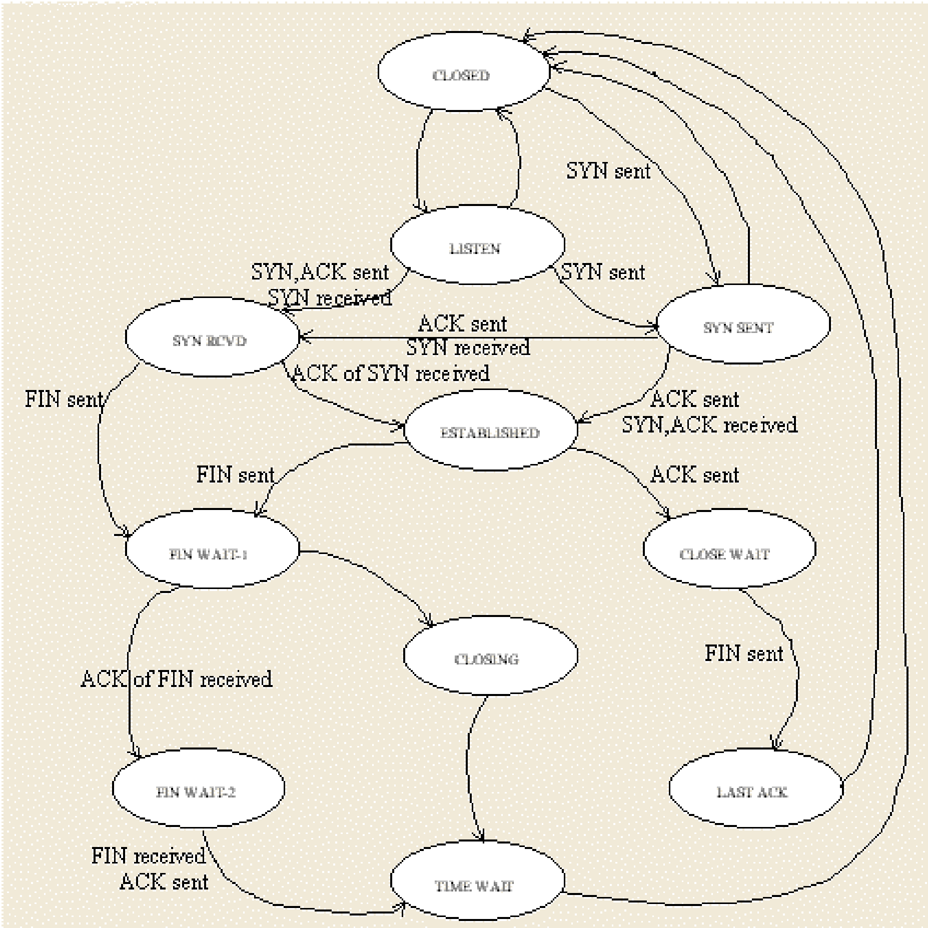
Supervisory
Control
And
Data
Acquisition

DCS - Rubs

- Digital technology + networking
- Rapid pace of the process industry automation
- The same PID control algorithms
- Deployment, support and maintenance cost reduction for massive amount of loops
- Autotuning technology
- Industrial digital control is becoming a commodity
- Facilitates deployment of supervisory control and monitoring

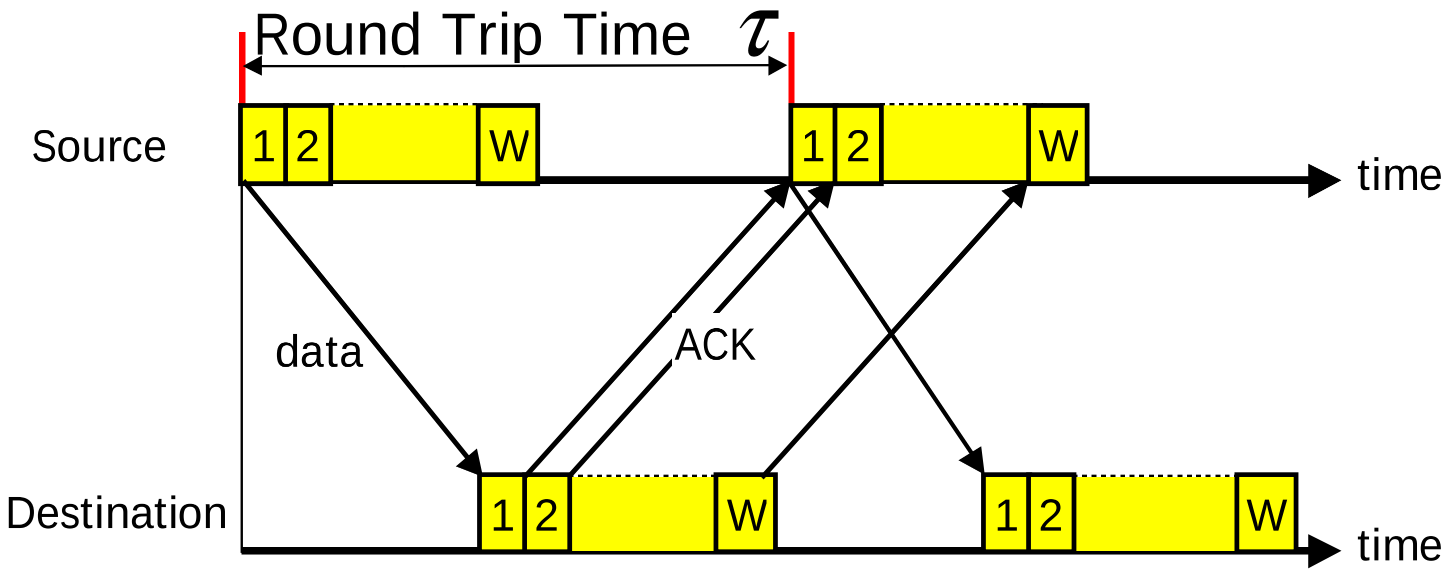
1974 - TCP/IP





- TCP/IP - Cerf/Kahn, 1974
- Berkeley-LLNL network crash, 1984
- Congestion control -Van Jacobson, 1986

TCP flow control



Transmission rate: $x = \dfrac{W}{\tau}$ packets/sec

Here:

- Flow control dynamics near the maximal transmission rate
- From S.Low, F.Paganini, J.Doyle, 2000

TCP Reno congestion avoidance

```
for every loss {
      W = W/2
      }
for every ACK {
      W += 1/W
      }
```

- packet acknowledgment rate: x
- lost packets: with probability q

$$\Delta x_{lost} = -xW/2$$

- transmitted: with probability (1-q)

$$\Delta x_{sent} = x/W$$

$$\dot{x} = q\frac{\Delta x_{lost}}{\tau} + (1-q)\frac{\Delta x_{sent}}{\tau} \qquad x = \frac{W}{\tau}$$

$$\dot{x} = \frac{1-q}{\tau^2} - \frac{1}{2}qx^2$$

- x - transmission rate
- τ - round trip time
- q - loss probability

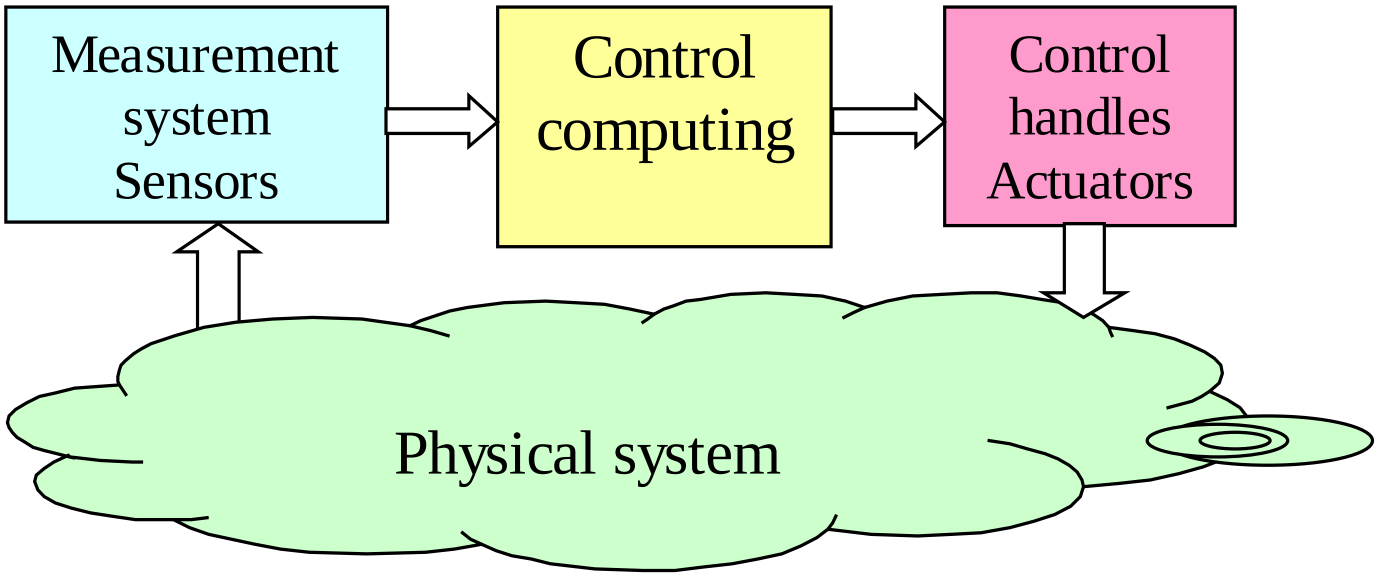
TCP flow control - Rubs

- Flow control enables stable operation of the Internet
- Developed by CS folks - no ‘controls’ analysis
- Ubiquitous, TCP stack is on ‘every’ piece of silicon
- Analysis and systematic design is being developed some 20 years later
- The behavior of the network is important. We looked at a single transmission.
- Most of analysis and systematic design activity in 4-5 last years and this is not over yet ...

Modern Control Engineering

- What BIG control application is coming next?
- Where and how control technology will be used?
- What do we need to know about controls to get by?

Modern Control Engineering

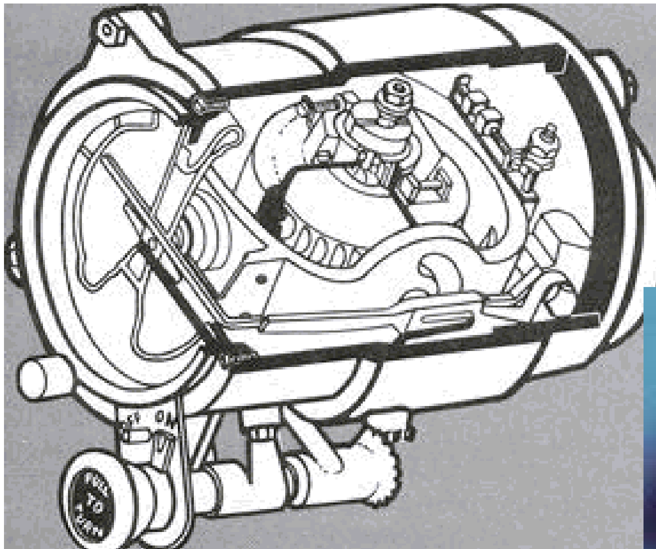


- This course is focused on **control computing** algorithms and their relationship with the overall system design.

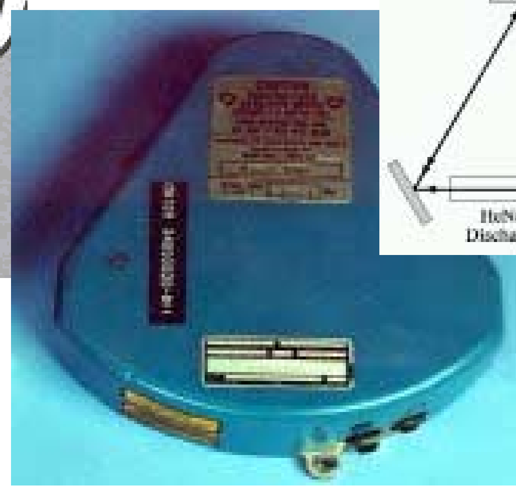
Modern control systems

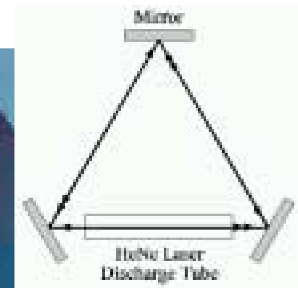
- Why this is relevant and important at present?
- Computing is becoming ubiquitous
- Sensors are becoming miniaturized, cheap, and pervasive. MEMS sensors
- Actuator technology developments include:
  - evolution of existing types
  - previously hidden in the system, not actively controlled
  - micro-actuators (piezo, MEMS)
  - control handles other than mechanical actuators, e.g., in telecom

Measurement system evolution.
Navigation system example

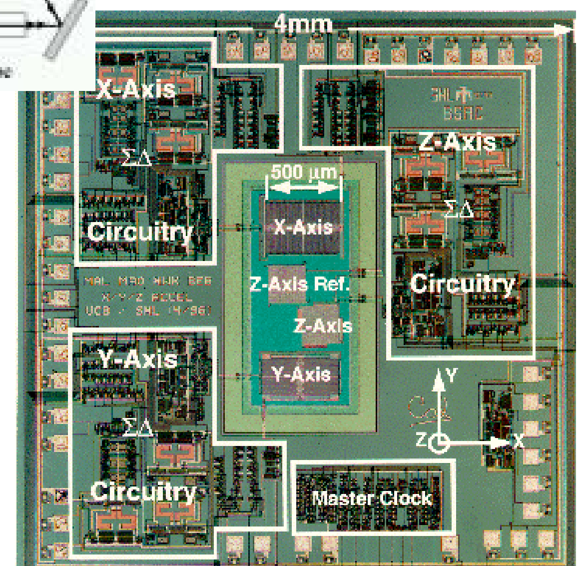


•Mechanical gyro by Sperry – for ships, aircraft. Honeywell acquired Sperry Aerospace in 1986 - avionics, space.





• Laser ring gyro, used in aerospace presently.



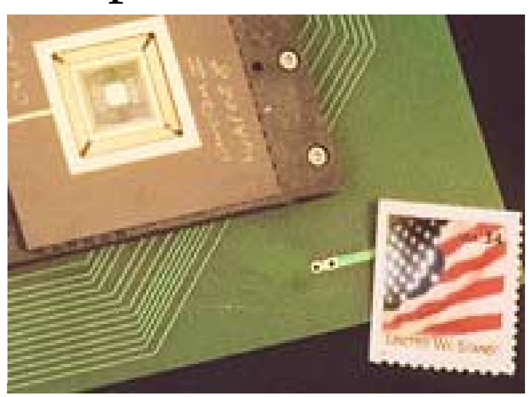
- MEMS gyro – good for any vehicle/mobile appliance.
  - (1")^3 integrated navigation unit

Actuator evolution

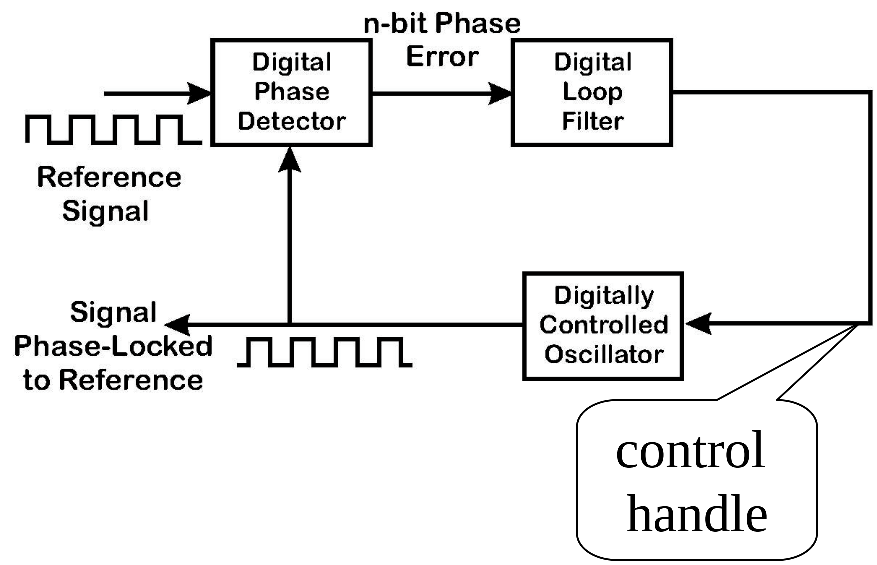
- Electromechanical actuators: car power everything

- Adaptive optics, MEMS

μDM 140, 3.3 mm



- Communication - digital PLL



Control computing

- Computing grows much faster than the sensors and actuators
- CAD tools, such as Matlab/Simulink, allow focusing on algorithm design. Implementation is automated
- Past: control was done by dedicated and highly specialized experts. Still the case for some very advanced systems in aerospace, military, automotive, etc.
- Present: control and signal-processing technology are standard technologies associated with computing.
- Embedded systems are often designed by system/software engineers.
- This course emphasizes practically important issues of control computing

Lecture 2 - Modeling and Simulation

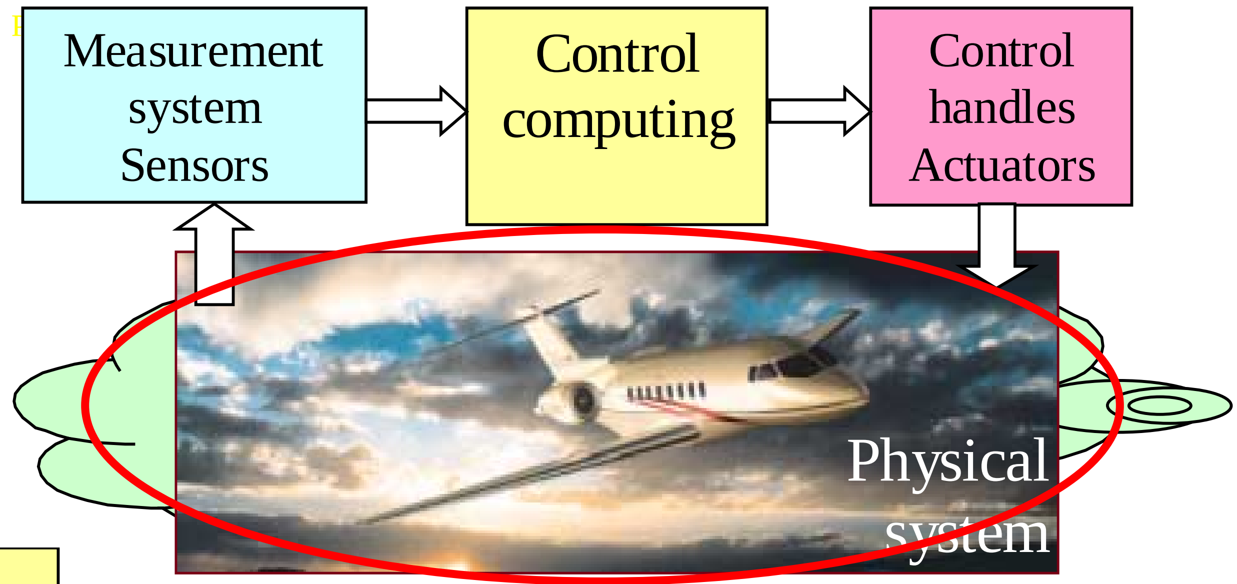
- Model types: ODE, PDE, State Machines, Hybrid
- Modeling approaches:
  - physics based (white box)
  - input-output models (black box)
- Linear systems
- Simulation
- Modeling uncertainty

Goals

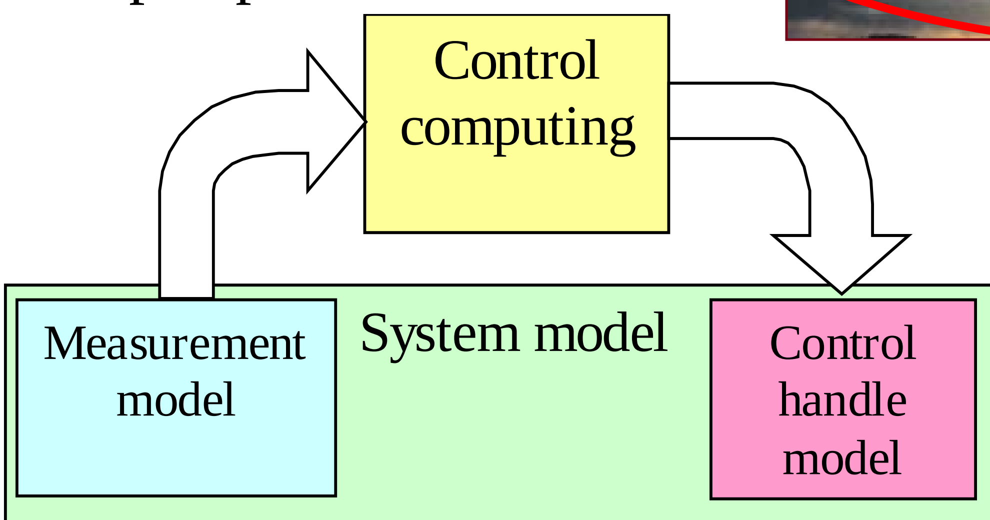
- Review dynamical modeling approaches used for control analysis and simulation
- Most of the material us assumed to be known
- Target audience
  - people specializing in controls - practical

Modeling in Control Engineering

- Control in a system perspective



- Control analysis perspective



Models

- Model is a mathematical representations of a system
  - Models allow simulating and analyzing the system
  - Models are never exact
- Modeling depends on your goal
  - A single system may have many models
  - *Always* understand what is the *purpose* of the model
  - Large ‘libraries’ of standard model templates exist
  - A conceptually new model is a big deal
- Main goals of modeling in control engineering
  - conceptual analysis
  - detailed simulation

Modeling approaches

- Controls analysis uses deterministic models. Randomness and uncertainty are usually not dominant.
- White box models: physics described by ODE and/or PDE
- Dynamics, Newton mechanics

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

- Space flight: add control inputs u and measured outputs y

$$\dot{x} = f(x,u,t)$$

$$y = g(x,u,t)$$

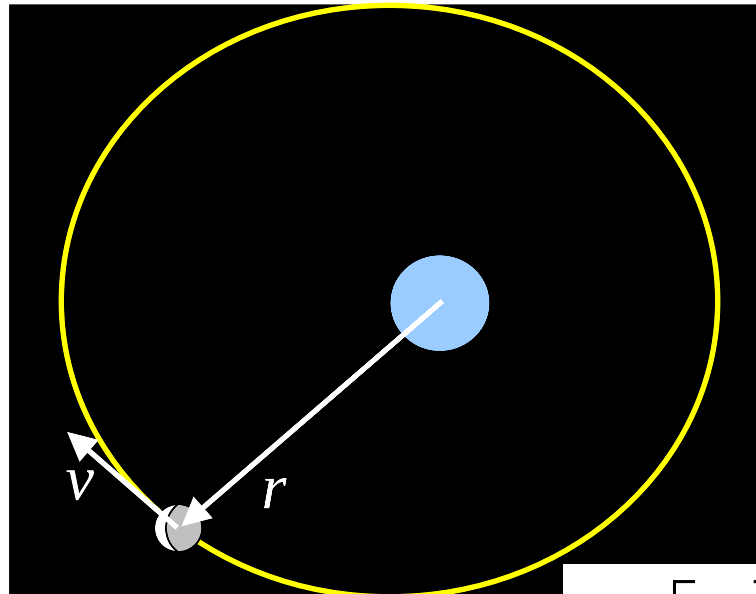
Orbital mechanics example


1643-1736

- Newton's mechanics
  - fundamental laws
  - dynamics

$$\dot{v} = -\gamma m \cdot \frac{r}{|r|^3} + F_{pert}(t)$$

$$\dot{r} = v$$




1749-1827

- Laplace
  - computational dynamics (pencil & paper computations)
  - deterministic model-based prediction

$$\dot{x} = f(x,t) \qquad x = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Orbital mechanics example

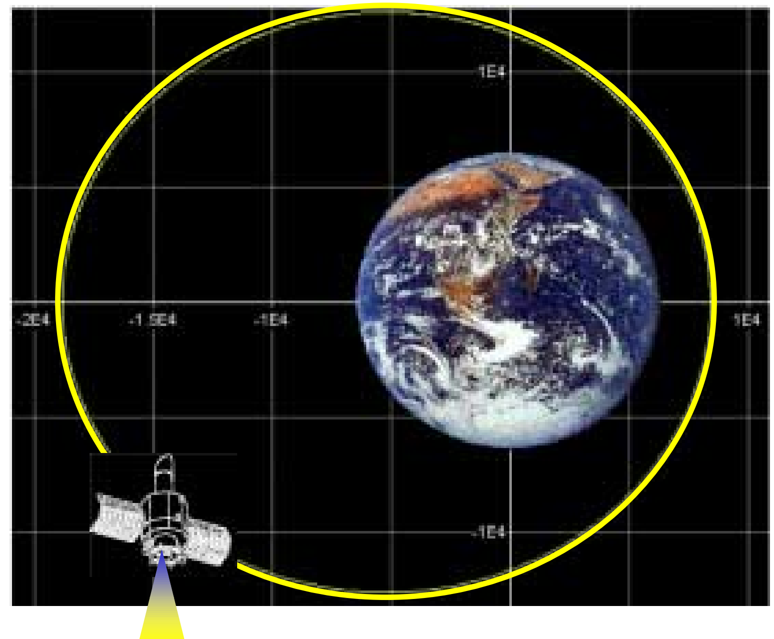
- Space flight mechanics

Thrust

$$\dot{v} = -\gamma m \cdot \frac{r}{|r|^3} + F_{pert}(t) + u(t)$$

$$\dot{r} = v$$

- Control problems: *u* - ?



state

$$x = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

observations / measurements

$$y = \begin{bmatrix} \theta(r) \\ \varphi(r) \end{bmatrix}$$

model

control

$$\dot{x} = f(x, u, t)$$

$$y = g(x, u, t)$$

Gene expression model

System variables and symbols

O	Total operon concentration
O_F	Free operon concentration
M_F	Free mRNA concentration
E	Total anthranilate synthase concentration
E_A	Active anthranilate synthase concentration
T	Tryptophan concentration
R	Total repressor concentration
R_A	Active repressor concentration
P	mRNA polymerase concentration
ρ	Ribosomal concentration
D	mRNA destroying enzyme concentration

Santillán-Mackey Model Equations

$$\dot{O}_F = \frac{K_r}{K_r + R_A(T)}\{\mu O - k_p P[O_F(t) - O_F(t-\tau_p)e^{-\mu\tau_p}]\} - \mu O_F(t)$$

$$\dot{M}_F = k_p P O_F(t-\tau_m)e^{-\mu\tau_m}[1 - A(T)] - k_\rho \rho[M_F(t) - M_F(t-\tau_\rho)e^{-\mu\tau_\rho}] - (k_d D + \mu)M_F(t)$$

$$\dot{E} = \frac{1}{2}k_\rho \rho M_F(t-\tau_e)e^{-\mu\tau_e} - (\gamma+\mu)E(t)$$

$$\dot{T} = K E_A(E,T) - G(T) + F(T,T_{ext}) - \mu T(t)$$

$$R_A(t) := R\frac{T(t)}{T(t)+K_t}$$

$$A(T) := b(1 - e^{-T(t)/c})$$

$$E_A(E,T) := \frac{K_i^{n_H}}{K_i^{n_H} + T^{n_H}(t)}E(t)$$

$$G(T) := g\frac{T(t)}{T(t)+K_g}$$

$$F(T,T_{ext}) := d\frac{T_{ext}}{e + T_{ext}[1 + T(t)/f]}$$

Sampled Time Models

- Time is often sampled because of the digital computer use
  - computations, numerical integration of continuous-time ODE

$$x(t+d) \approx x(t) + d \cdot f(x,u,t), \qquad t = kd$$

  - digital (sampled time) control system

$$x(t+d) = f(x,u,t)$$

$$y = g(x,u,t)$$

- Time can be sampled because this is how a system works
- Example: bank account balance
  - $x(t)$ - balance in the end of day t
  - $u(t)$ - total of deposits and withdrawals that day
  - $y(t)$ - displayed in a daily statement

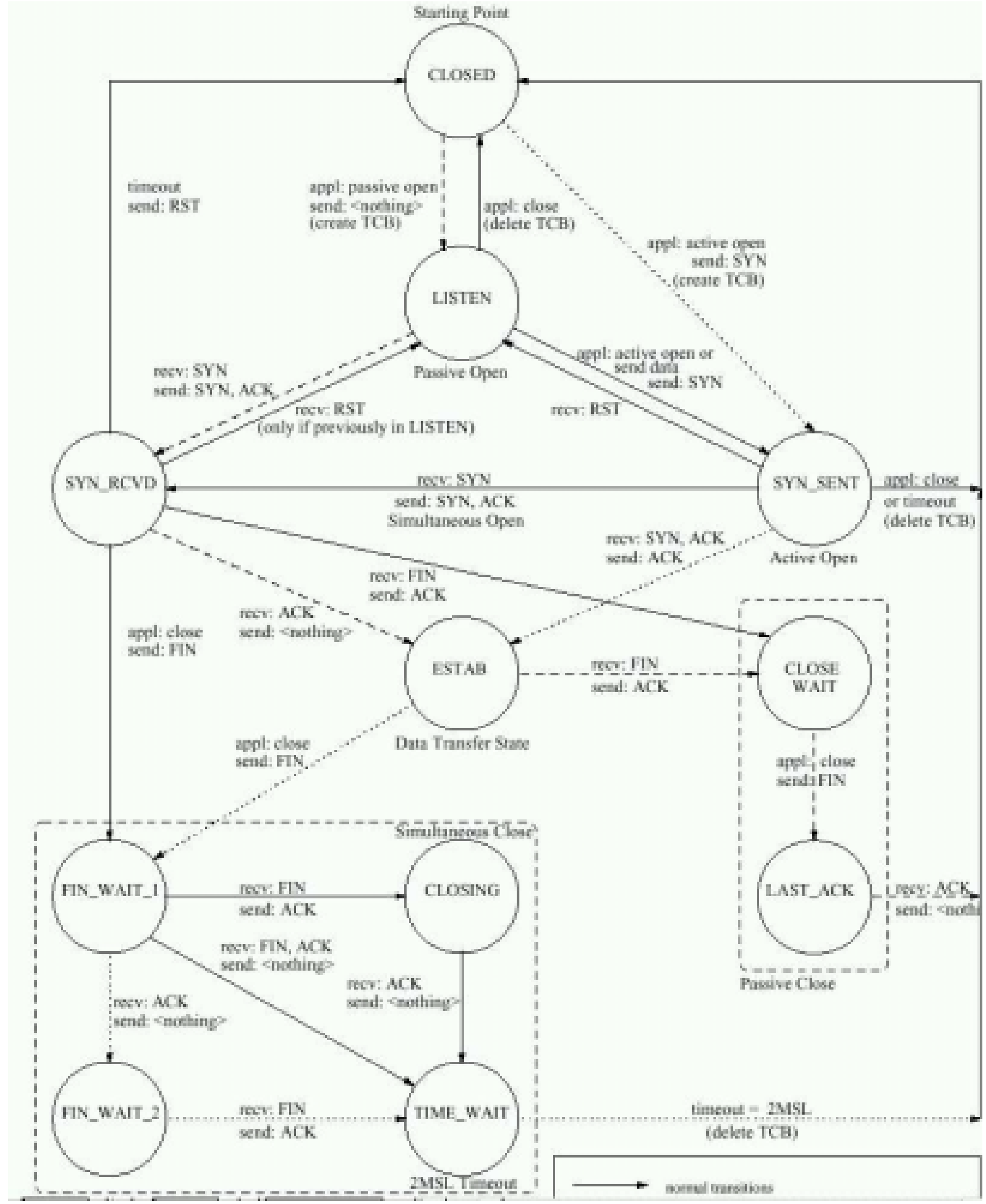
$$x(t+1) = x(t) + u(t)$$

$$y = x$$

- Unit delay operator z^{-1}: $z^{-1} x(t) = x(t-1)$

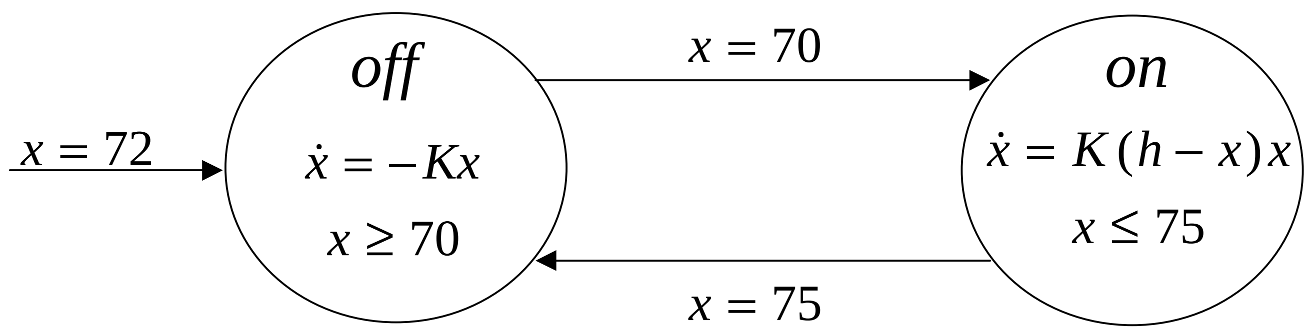
Finite state machines

- TCP/IP State Machine



Hybrid systems

- Combination of continuous-time dynamics and a state machine
- Thermostat example
- Tools are not fully established yet



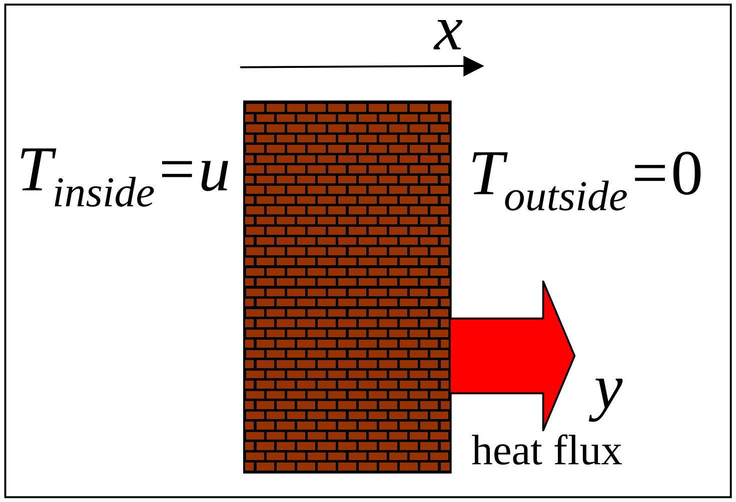
PDE models

- Include functions of spatial variables
  - electromagnetic fields
  - mass and heat transfer
  - fluid dynamics
  - structural deformations
- Example: sideways heat equation

$$\frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$$

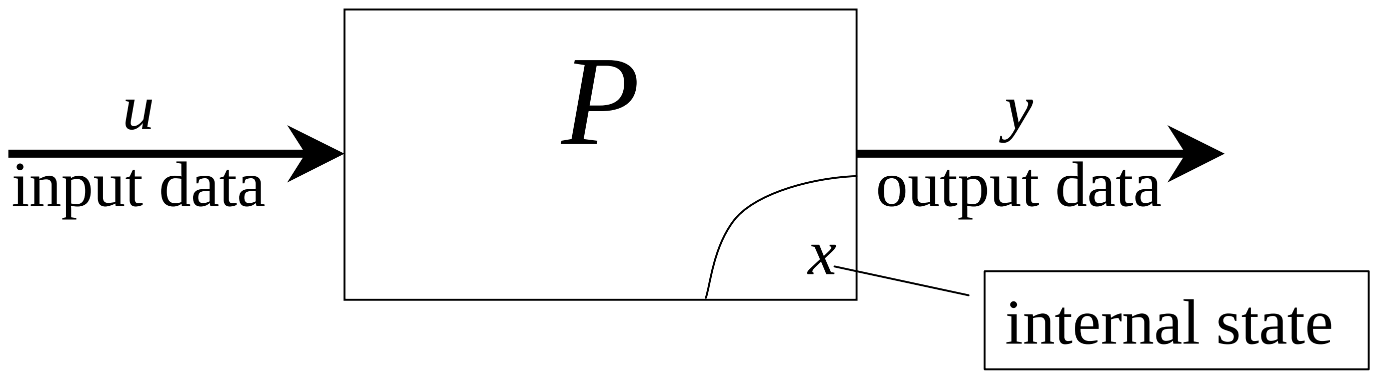
$$T(0) = u; \qquad T(1) = 0$$

$$y = \left. \frac{\partial T}{\partial x} \right|_{x=1}$$



Black-box models

- Black-box models - describe *P* as an operator



  - AA, ME, Physics - state space, ODE and PDE
  - EE - black-box,
  - ChE - use anything
  - CS - state machines, probablistic models, neural networks

Linear Systems

- Impulse response
- FIR model
- IIR model
- State space model
- Frequency domain
- Transfer functions
- Sampled vs. continuous time
- Linearization

Linear System (black-box)

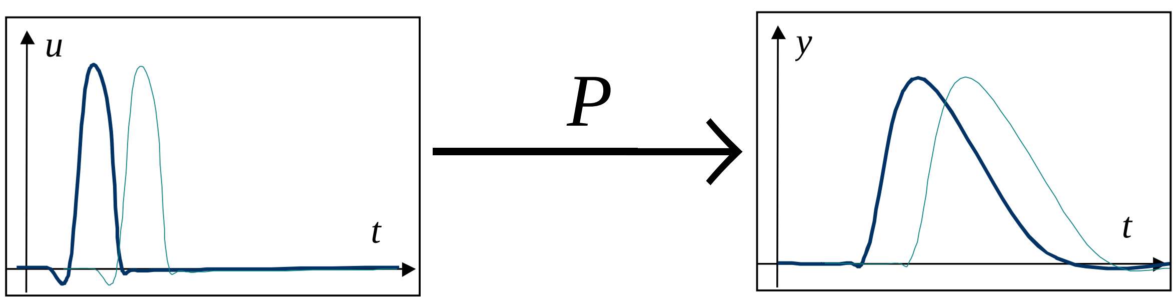
- Linearity

$$u_1(\cdot) \xrightarrow{P} y_1(\cdot) \qquad u_2(\cdot) \xrightarrow{P} y_2(\cdot)$$

$$au_1(\cdot) + bu_2(\cdot) \xrightarrow{P} ay_1(\cdot) + by_2(\cdot)$$

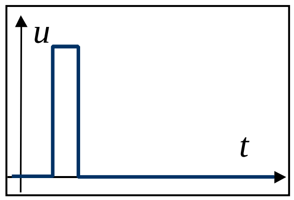
- Linear Time-Invariant systems - LTI

$$u(\cdot - T) \xrightarrow{P} y(\cdot - T)$$

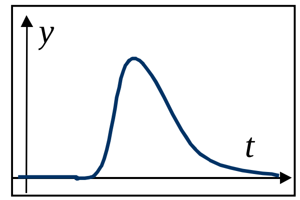


Impulse response

- Response to an input impulse



$$\delta(\cdot) \xrightarrow{P} h(\cdot)$$



- Sampled time: t = 1, 2, ...
- Control history = linear combination of the impulses $\Rightarrow$ system response = linear combination of the impulse responses

$$u(t) = \sum_{k=0}^{\infty} \delta(t-k)u(k)$$

$$y(t) = \sum_{k=0}^{\infty} h(t-k)u(k) = (h*u)(t)$$

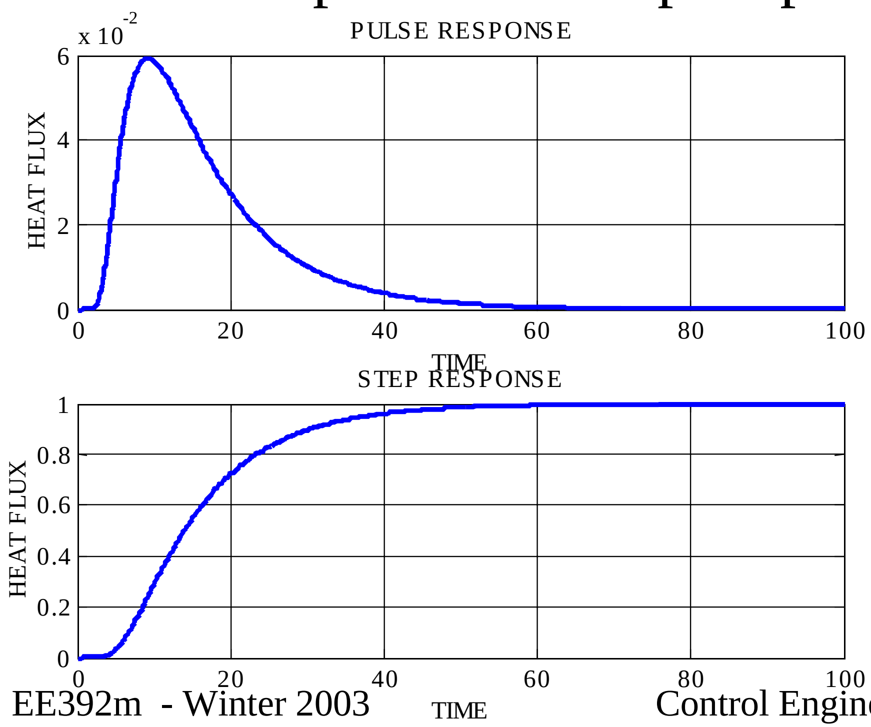
Linear PDE System Example

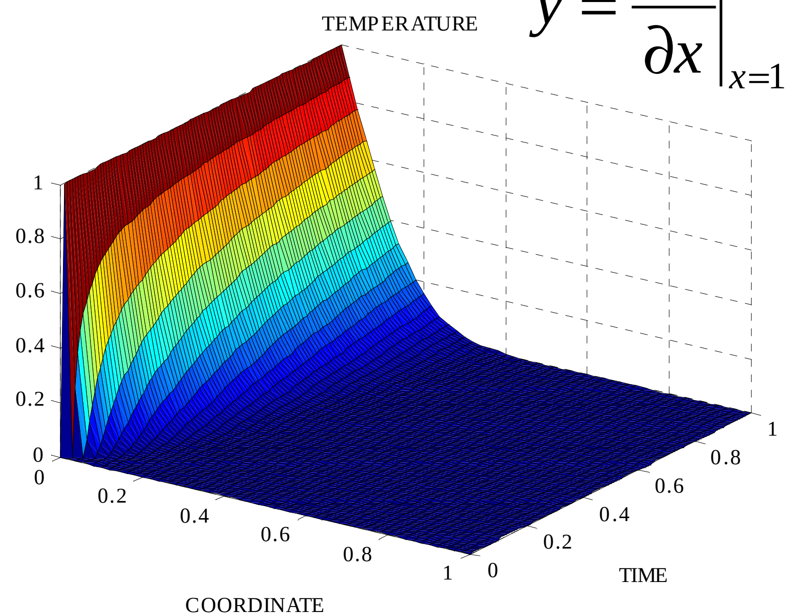
- Heat transfer equation,
  - boundary temperature input u
  - heat flux output y
- Pulse response and step response

$$\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2}$$

$$u = T(0) \qquad T(1) = 0$$

$$y = \left.\frac{\partial T}{\partial x}\right|_{x=1}$$



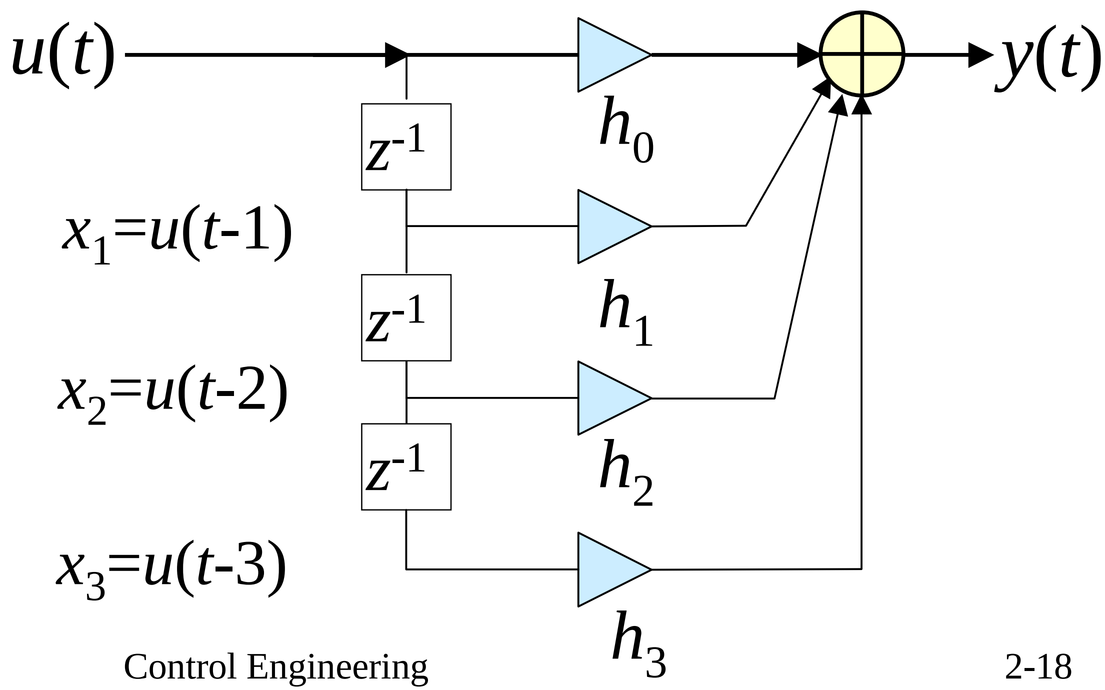


FIR model

$$y(t) = \sum_{k=0}^{N} h_{FIR}(t-k)u(k) = (h_{FIR} * u)(t)$$

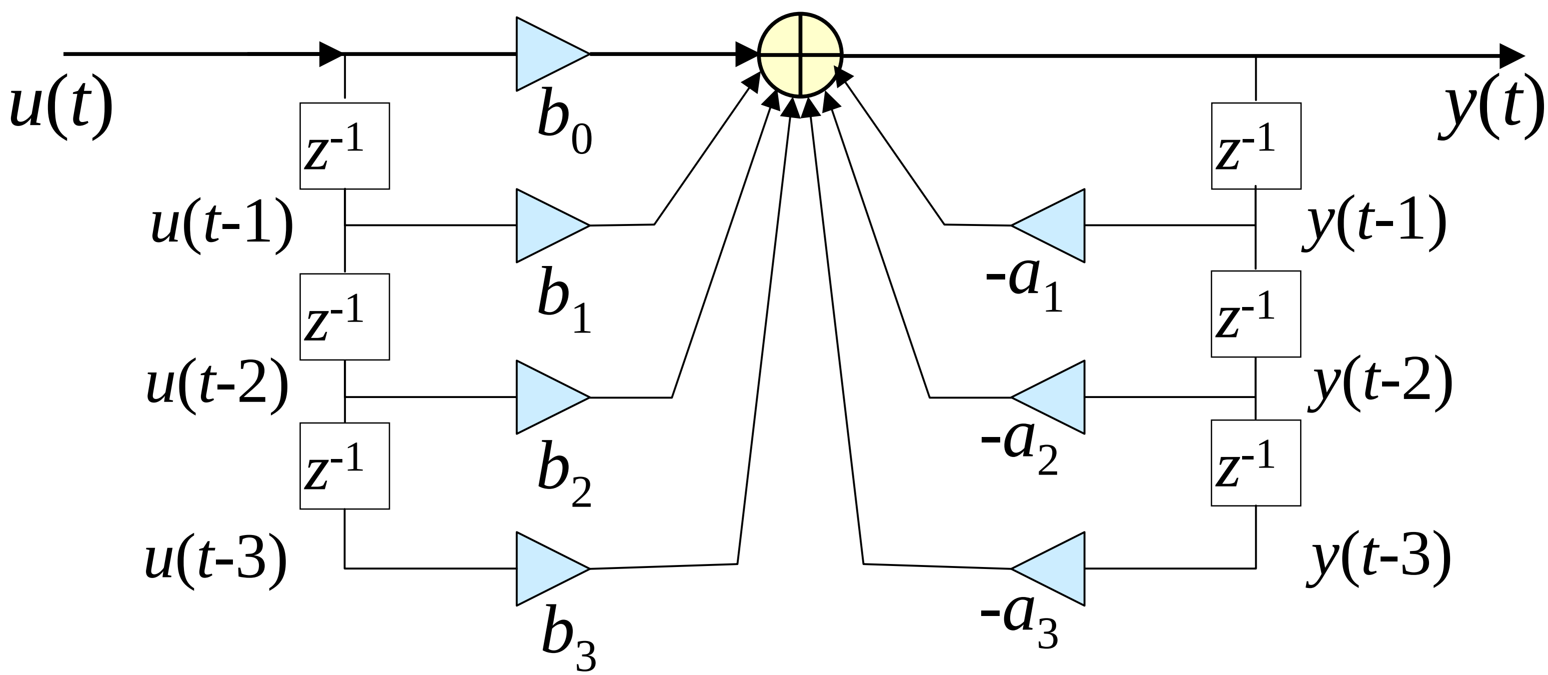
- FIR = Finite Impulse Response
- Cut off the trailing part of the pulse response to obtain FIR
- FIR filter state *x*. Shift register

$$x(t+1) = f(x,u)$$
$$y = g(x,u)$$



IIR model

- IIR model: $y(t) = -\sum_{k=1}^{n_a} a_k y(t-k) + \sum_{k=0}^{n_b} b_k u(t-k)$
- Filter states: $y(t\text{-}1), \ldots, y(t\text{-}n_a), u(t\text{-}1), \ldots, u(t\text{-}n_b)$



IIR model

- Matlab implementation of an IIR model: **`filter`**
- Transfer function realization: unit delay operator z^{-1}

$$y(t) = H(z)u(t)$$

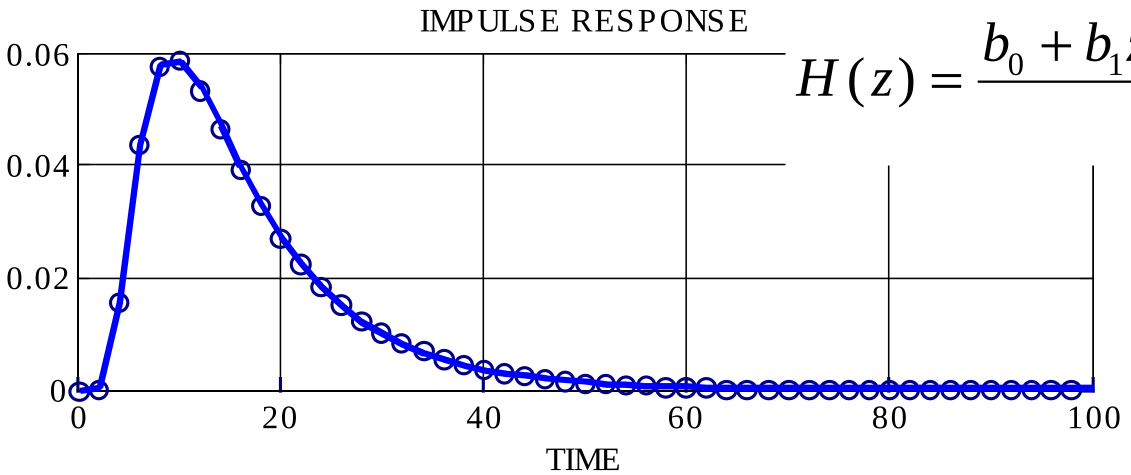
$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + ... + b_N z^{-N}}{1 + a_1 z^{-1} + ... + a_N z^{-N}} = \frac{b_0 z^N + b_1 z^{N-1} + ... + b_N}{z^N + a_1 z^{N-1} + ... + a_N}$$

$$\underbrace{\left(1 + a_1 z^{-1} + ... + a_N z^{-N}\right)}_{A(z)} y(t) = \underbrace{\left(b_0 + b_1 z^{-1} + ... + b_N z^{-N}\right)}_{B(z)} u(t)$$

- FIR model is a special case of an IIR with $A(z) = 1$ (or z^N)

IIR approximation example

- Low order IIR approximation of impulse response: (`prony` in Matlab Signal Processing Toolbox)
- Fewer parameters than a FIR model
- Example: sideways heat transfer
  - pulse response $h(t)$
  - approximation with IIR filter $a = [a_1 \; a_2]$, $b=[b_0 \; b_1 \; b_2 \; b_3 \; b_4]$



$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Linear state space model

- Generic state space model:

$$x(t+1) = f(x,u,t)$$
$$y = g(x,u,t)$$

- LTI state space model
  - another form of IIR model
  - physics-based linear system model

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

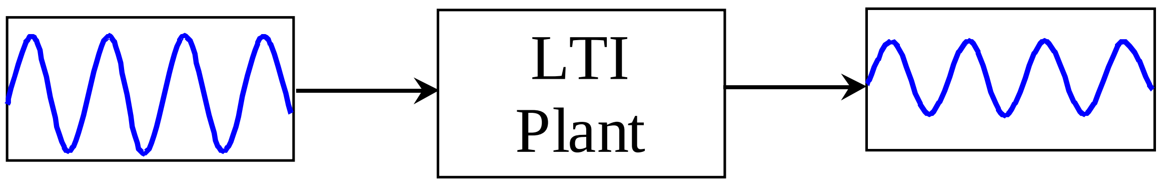
- Transfer function of an LTI model
  - defines an IIR representation

$$y = \left[(Iz - A)^{-1} B + D \right] \cdot u$$
$$H(z) = (Iz - A)^{-1} B + D$$

- Matlab commands for model conversion: **`help ltimodels`**

Frequency domain description

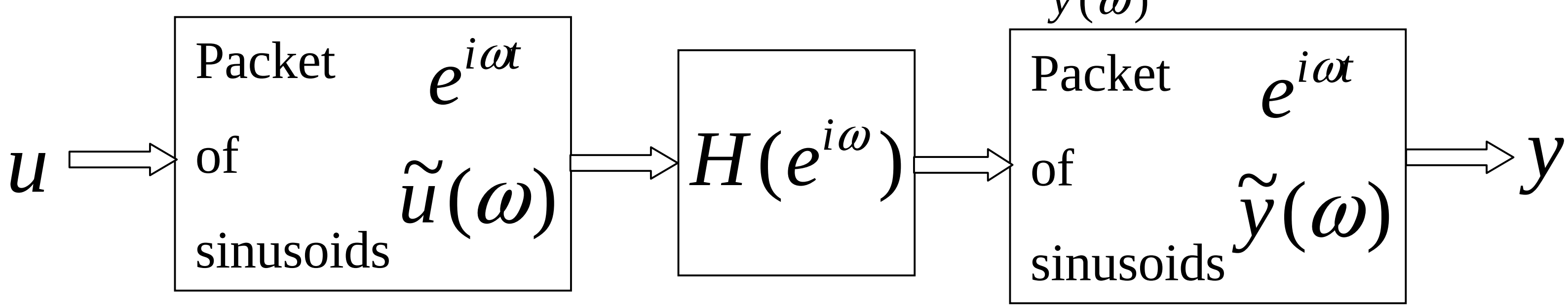
- Sinusoids are eigenfunctions of an LTI system: $y = H(z)u$



$$z^{-1}e^{i\omega t} = e^{i\omega(t-1)} = e^{-i\omega}e^{i\omega t}$$

- Frequency domain analysis

$$u = \int \tilde{u}(\omega)e^{i\omega t}d\omega \Rightarrow y = \int \underbrace{H(e^{i\omega})\tilde{u}(\omega)}_{\tilde{y}(\omega)}e^{i\omega t}d\omega$$



Frequency domain description

$$M(\omega) = \left| H(e^{i\omega}) \right|$$

$$\varphi(\omega) = \arg H(e^{i\omega})$$

- Bode plots:

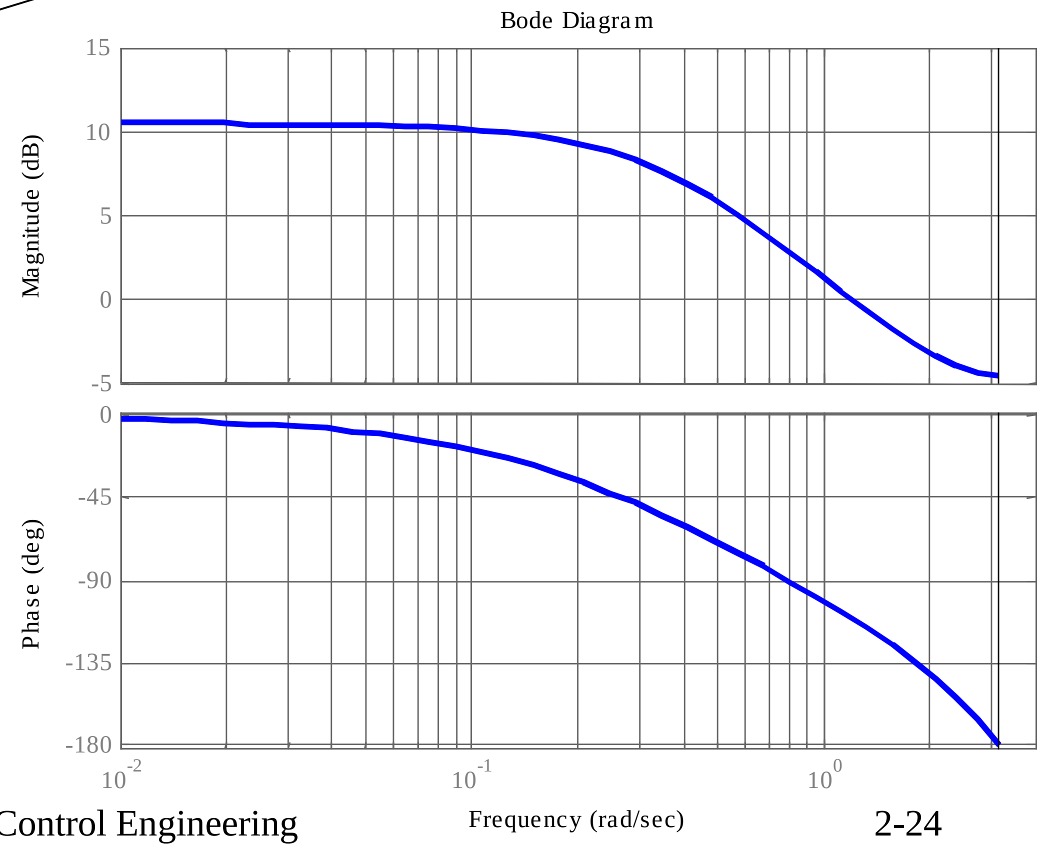
$$u = e^{i\omega t}$$

$$y = H(e^{i\omega})e^{i\omega t}$$

- Example:

$$H(z) = \frac{1}{z - 0.7}$$

- $|H|$ is often measured in dB



Black-box model from data

- Linear black-box model can be determined from the data, e.g., step response data
- This is called model identification
- Lecture 8

z-transform, Laplace transform

- Formal description of the transfer function:
  - function of complex variable z
  - analytical outside the circle $|z| \geq r$
  - for a stable system $r \leq 1$

$$H(z) = \sum_{k=0}^{\infty} h(k) z^{-k}$$

- Laplace transform:
  - function of complex variable s
  - analytical in a half plane $\mathrm{Re}\, s \leq a$
  - for a stable system $a \leq 1$

$$H(s) = \int_{-\infty}^{\infty} h(t) e^{st} dt$$

$$\hat{y}(s) = H(s)\hat{u}(s)$$

Stability analysis

- Transfer function poles tell you everything about stability
- Model-based analysis for a simple feedback example:

$$y = H(z)u$$
$$u = -K(y - y_d)$$

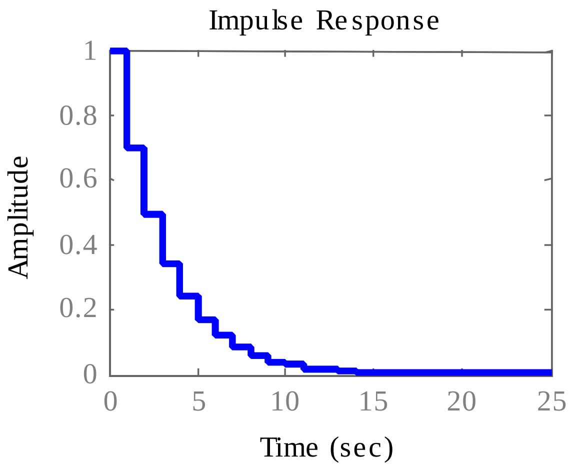
$$\Longrightarrow \quad y = \frac{H(z)K}{1 + H(z)K} y_d = L(z) y_d$$

- If $H(z)$ is a rational transfer function describing an IIR model
- Then $L(z)$ also is a rational transfer function describing an IIR model

Poles and Zeros <=> System

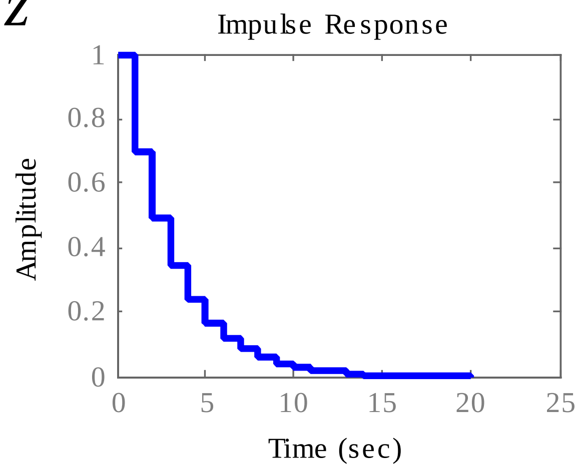
- **…not quite so!**
- Example:

$$y = H(z)u = \frac{z}{z-0.7}$$



- FIR model - truncated IIR

$$y = H_{FIR}(z)u = \frac{z^{19} + 0.7z^{18} + 0.49z^{17} + \ldots + 0.001628\, z + 0.00114}{z^{19}}$$



IIR/FIR example - cont'd

- Feedback control:

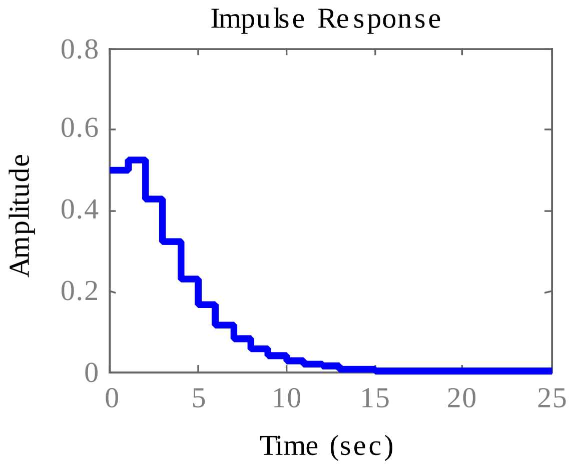
$$y = H(z)u = \frac{z}{z-0.7}$$

$$u = -K(y - y_d) = -(y - y_d)$$

- Closed loop:

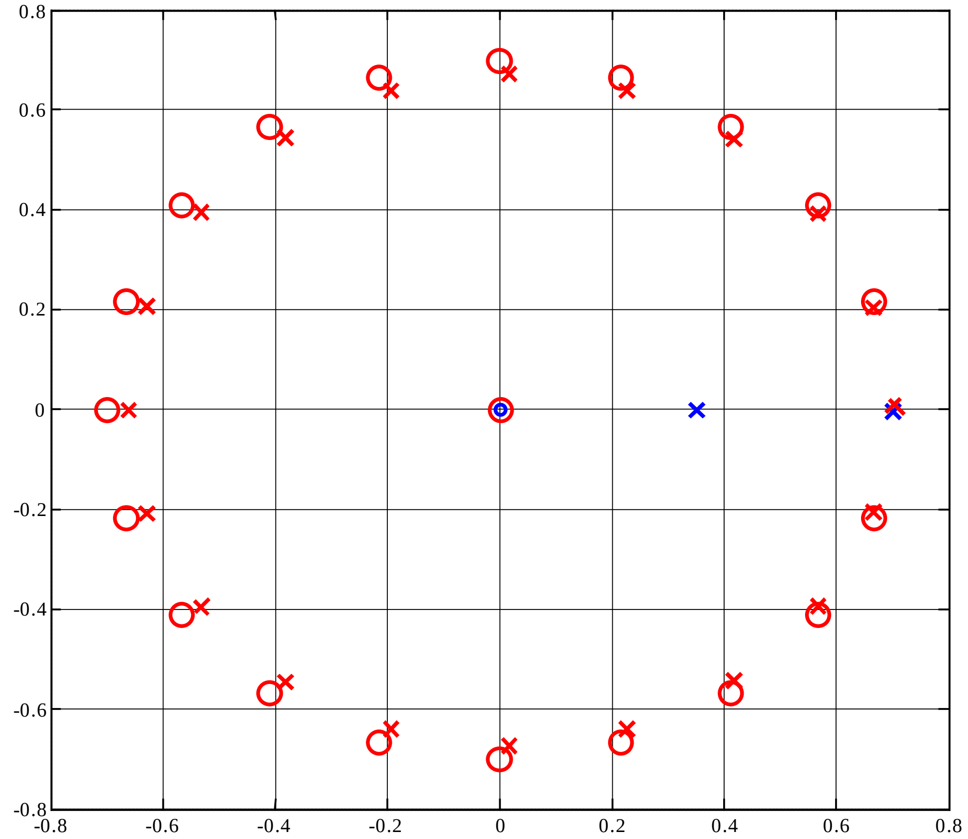
$$y = \frac{H(z)}{1+H(z)}u = L(z)u$$

$$y = \frac{H_{FIR}(z)}{1+H_{FIR}(z)}u = L_{FIR}(z)u$$



IIR/FIR example - cont’d
Poles and zeros

- **Blue:** Loop with IIR model poles **x** and zeros **o**
- **Red**: Loop with FIR model poles **x** and zeros **o**



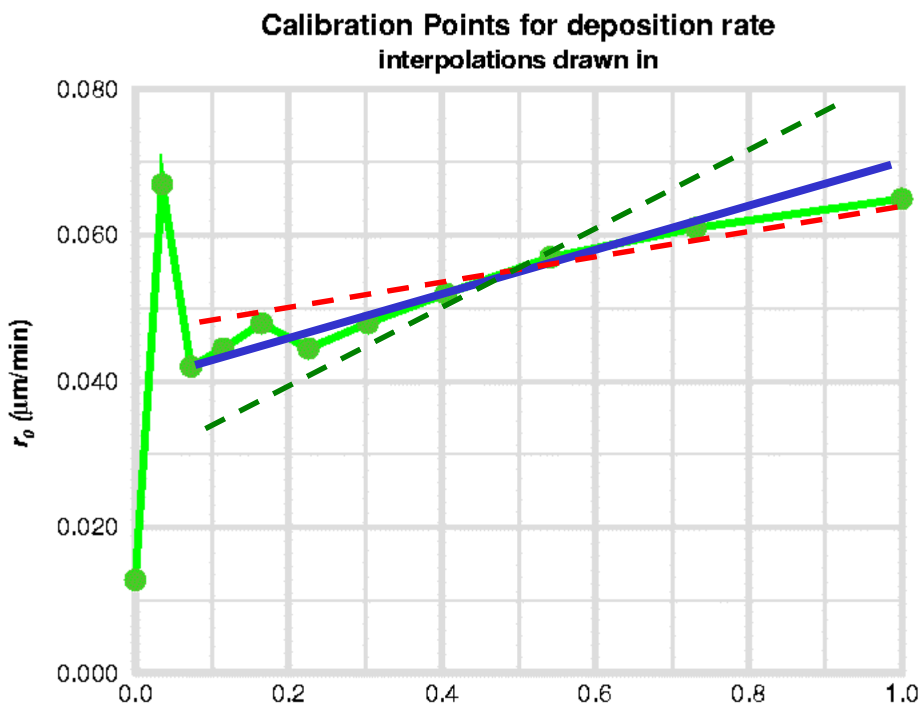
LTI models - summary

- Linear system can be described by impulse response
- Linear system can be described by frequency response = Fourier transform of the impulse response
- FIR, IIR, State-space models can be used to obtain close approximations of a linear system
- A pattern of poles and zeros can be very different for a small change in approximation error.
- Approximation error <=> model uncertainty

Nonlinear map linearization

- Nonlinear - detailed model
- Linear - conceptual design model
- Static map, gain range, sector linearity
- Differentiation, secant method

$$y = f(u) \approx \frac{\Delta f}{\Delta u}(u - u_0)$$



Nonlinear state space model linearization

- Linearize the r.h.s. map

$$\dot{x} = f(x,u) \approx \frac{\Delta f}{\Delta x}\underbrace{(x - x_0)}_{q} + \frac{\Delta f}{\Delta u}\underbrace{(u - u_0)}_{v}$$

$$\dot{q} = Aq + Bv$$

- Secant method

$$\left[\frac{\Delta f}{\Delta x}\right]^j = \frac{f(x + s_j)}{s_j}$$

$$s_j = [0 \quad \ldots \quad \underbrace{1}_{\# j} \quad \ldots \quad 0]$$

- Or … capture a response to small step and build an impulse response model

Sampled time vs. continuous time

- Continuous time analysis (Digital implementation of continuous time controller)
  - Tustin's method = trapezoidal rule of integration for $H(s) = \frac{1}{s}$

$$H(s) \rightarrow H_s(z) = H\left(s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

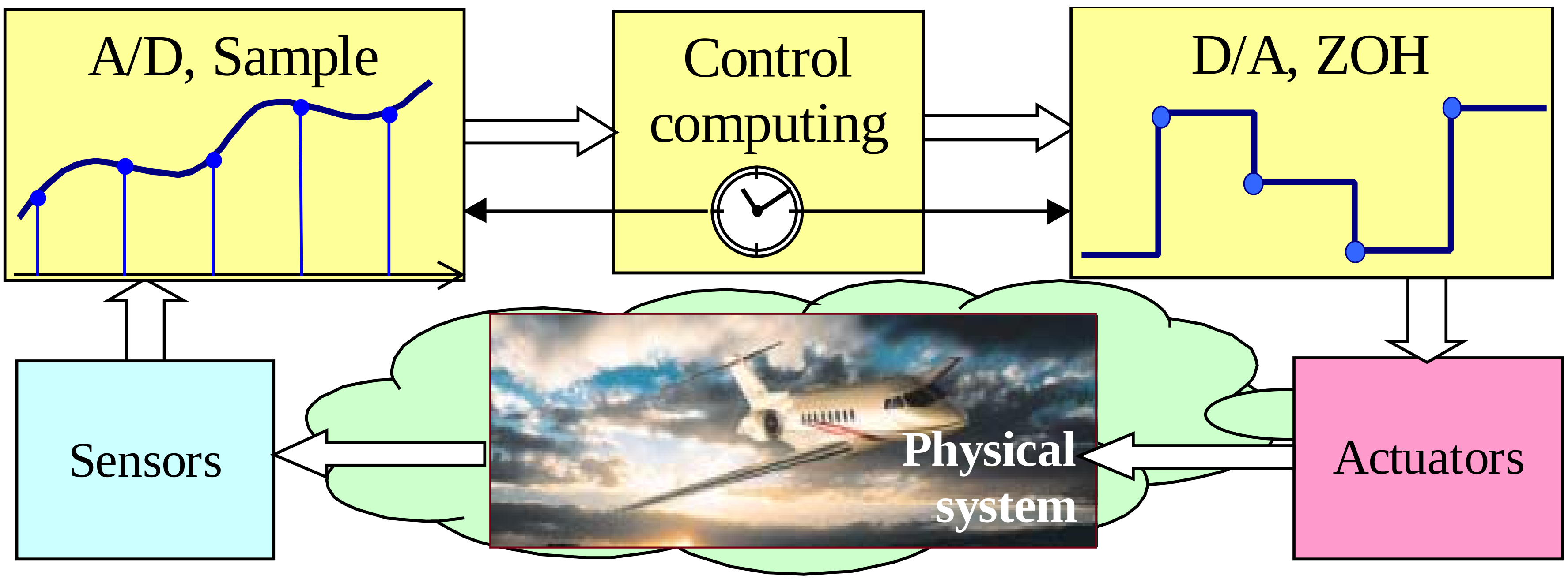
  - Matched Zero Pole: map each zero and a pole in accordance with

$$s = e^{sT}$$

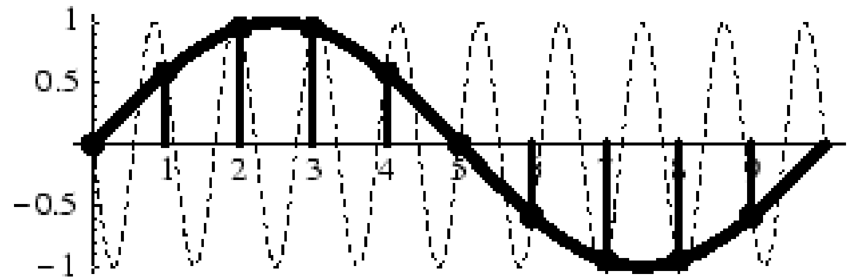
- Sampled time analysis (Sampling of continuous signals and system)

Sampled and continuous time

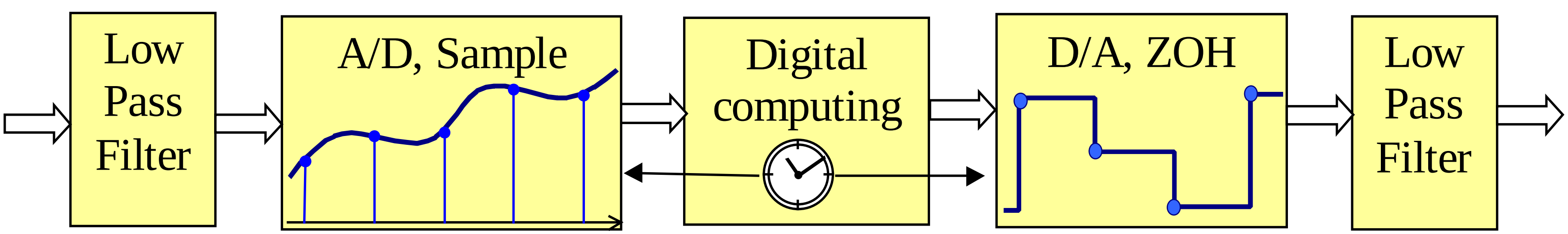
- Sampled and continuous time together
- Continuous time physical system + digital controller
  – ZOH = Zero Order Hold



Signal sampling, aliasing



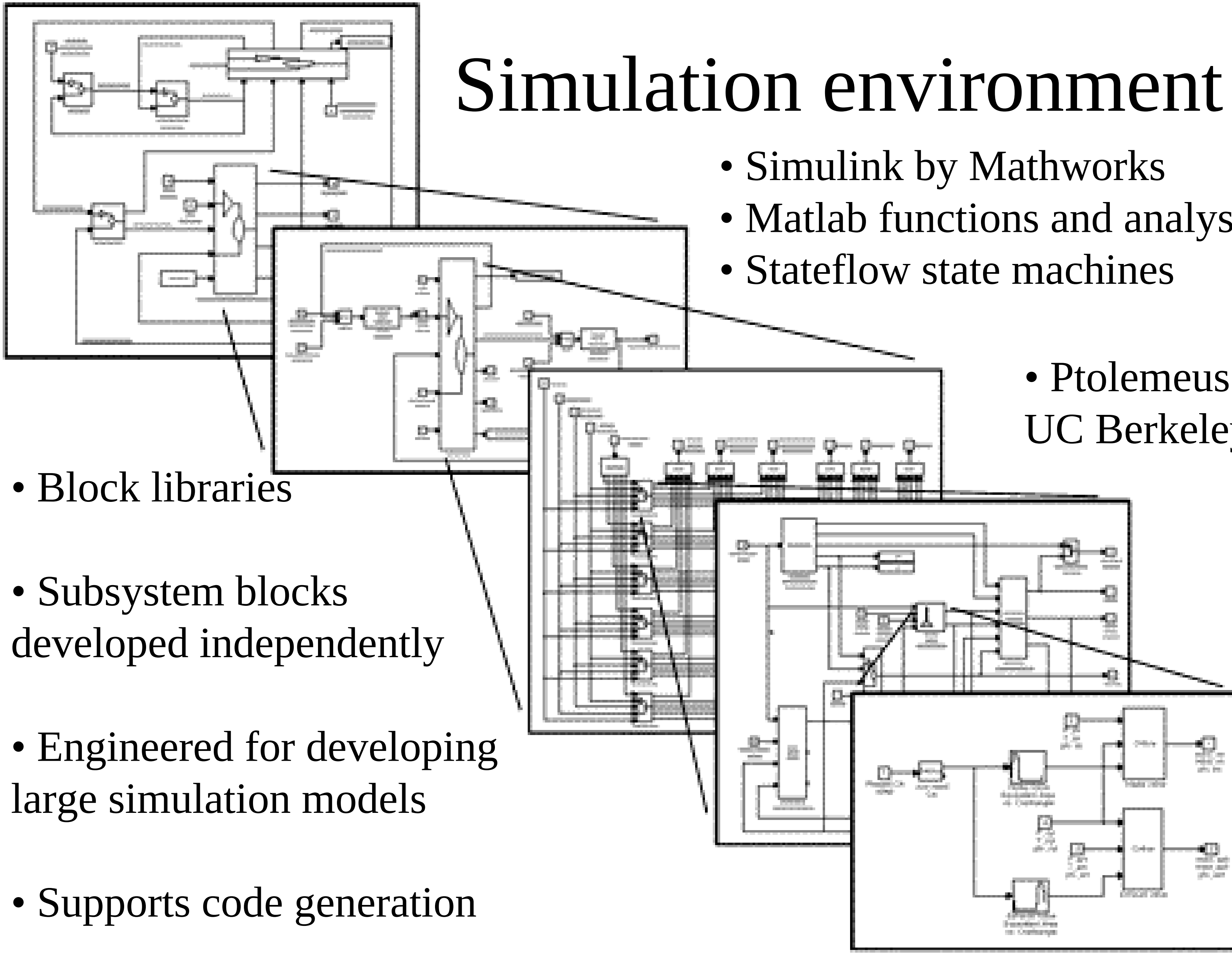
- Nyquist frequency:
  $\omega_N = \frac{1}{2}\omega_S$; $\omega_S = 2\pi/T$
- Frequency folding: $k\omega_S \pm \omega$ map to the same frequency ω
- Sampling Theorem: sampling is OK if there are no frequency components above ω_N
- Practical approach to anti-aliasing: low pass filter (LPF)
- Sampled→continuous: impostoring



Simulation

- ODE solution
  - dynamical model: $\dot{x} = f(x,t)$
  - Euler integration method: $x(t+d) = x(t) + d \cdot f\big(x(t),t\big)$
  - Runge-Kutta: **ode45** in Matlab
- Can do simple problems by integrating ODEs
- Issues:
  - mixture of continuous and sampled time
  - hybrid logic (conditions)
  - state machines
  - stiff systems, algebraic loops
  - systems integrated out of many subsystems
  - large projects, many people contribute different subsystems

Simulation environment



- Simulink by Mathworks
- Matlab functions and analysis
- Stateflow state machines

- Ptolemeus - UC Berkeley

- Block libraries

- Subsystem blocks developed independently

- Engineered for developing large simulation models

- Supports code generation

Model block development

- Look up around for available conceptual models
- Physics - conceptual modeling
- Science (analysis, simple conceptual abstraction) vs. engineering (design, detailed models - out of simple blocks)

Modeling uncertainty

- Modeling uncertainty:
  - unknown signals
  - model errors
- Controllers work with real systems:
  - Signal processing: data $\rightarrow$ algorithm $\rightarrow$ data
  - Control: algorithms in a feedback loop with ***a real*** system
- BIG question: Why controller designed for a model would *ever* work with a *real* system?
  - Robustness, gain and phase margins,
  - Control design model, vs. control analysis model
  - Monte-Carlo analysis - a fancy name for a desperate approach

Lecture 3 - Model-based Control Engineering

- Control application and a platform
- Systems platform: hardware, systems software. Development steps
- Model-based design
- Control solution deployment and support
- Control application areas

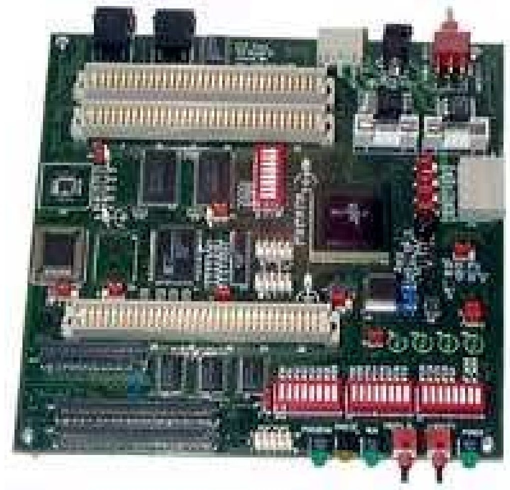
Generality of control

- Modeling abstraction
- Computing element - software
- System, actuator, and sensor physics might be very different
- Control and system engineering is used across many applications
  - similar principles
  - transferable skills
  - mind the application!

System platform for control computing

- Workstations
  - advanced process control
  - enterprise optimizers
  - computing servers (QoS/admission control)
- Specialized controllers:
  - PLC, DCS, motion controllers, hybrid controllers

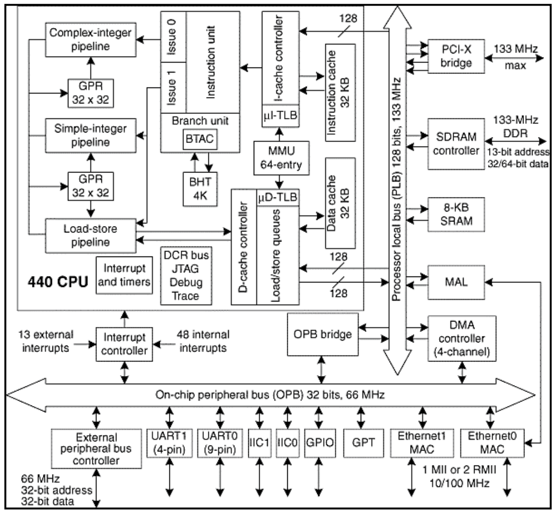
System platform for control computing



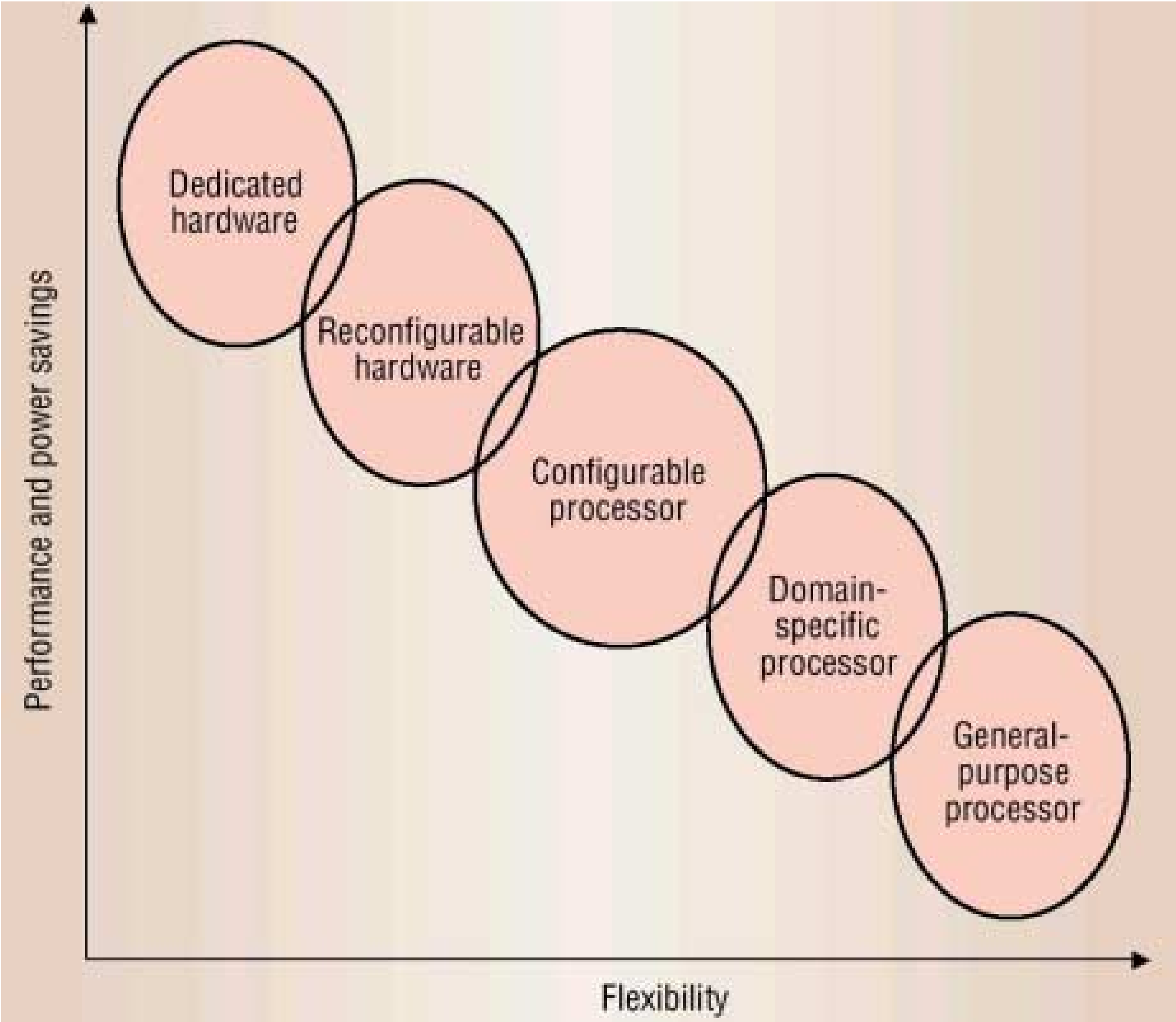
MPC555

- Embedded: μP + software
- DSP
- FPGA

- ASIC / SoC

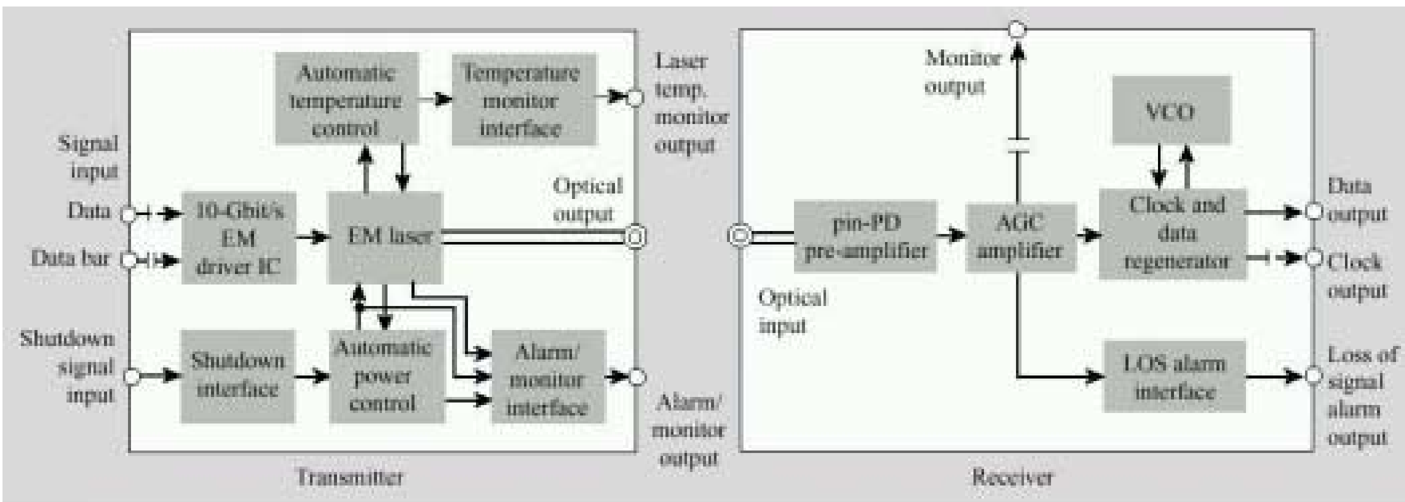


Embedded processor range



System platform, cont'd

- Analog/mixed electric circuits
  - power controllers
  - RF circuits
- Analog/mixed other
  - Gbs optical networks



EM = Electr-opt Modulator

Functional Block Diagram of 10-Gbit/s Optical Transmitter/Receiver.

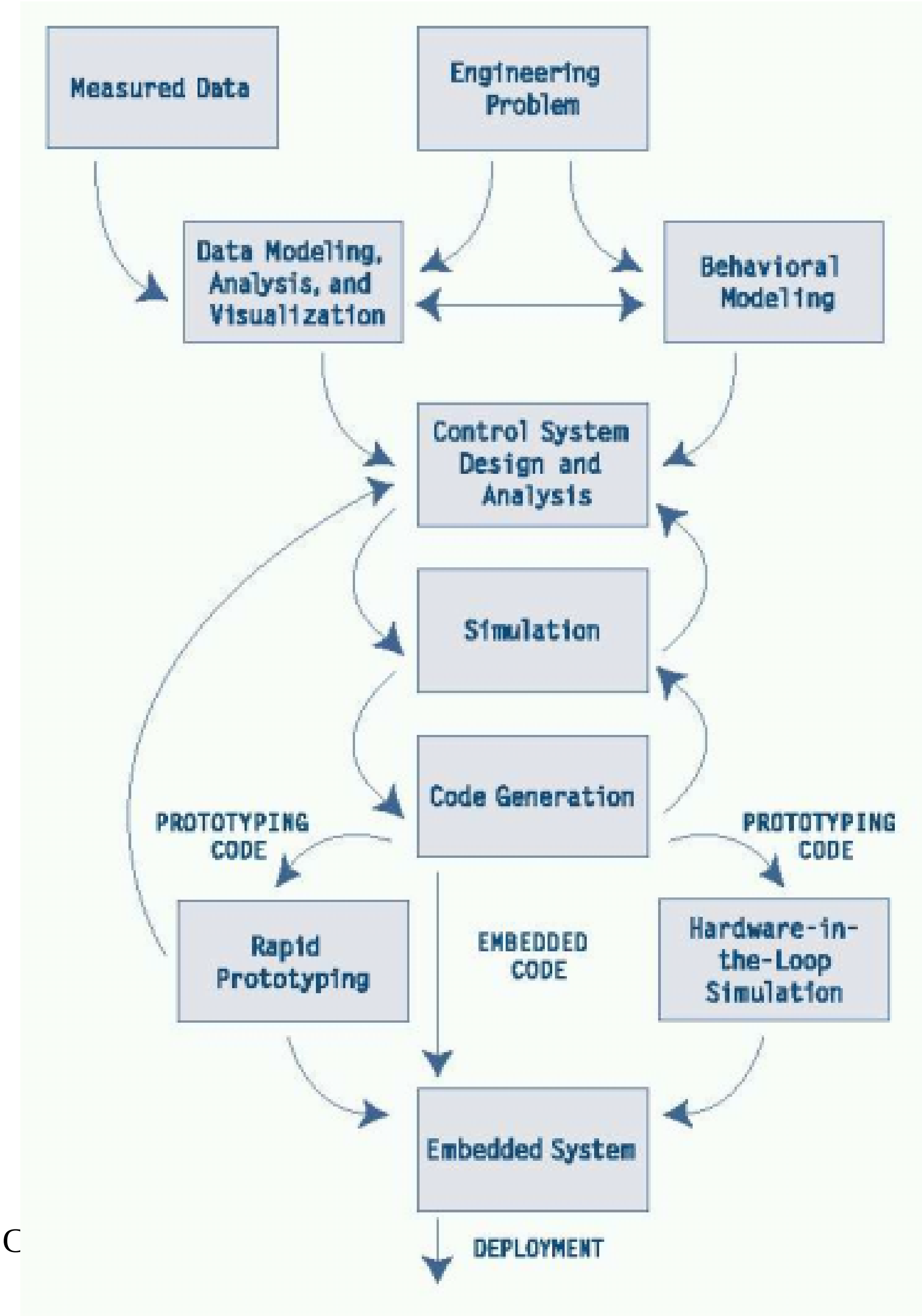
AGC = Auto Gain Control

Controls development cycle

- Analysis and modeling
  - physical model, or empirical, or data driven
  - use a simplified design model
  - system trade study - defines system design
- Heavy use of CAD tools
- Simulation
  - design validation using detailed performance model
- System development
  - control application, software platform, hardware platform
- Validation and verification
  - against initial specs
- Certification/commissioning

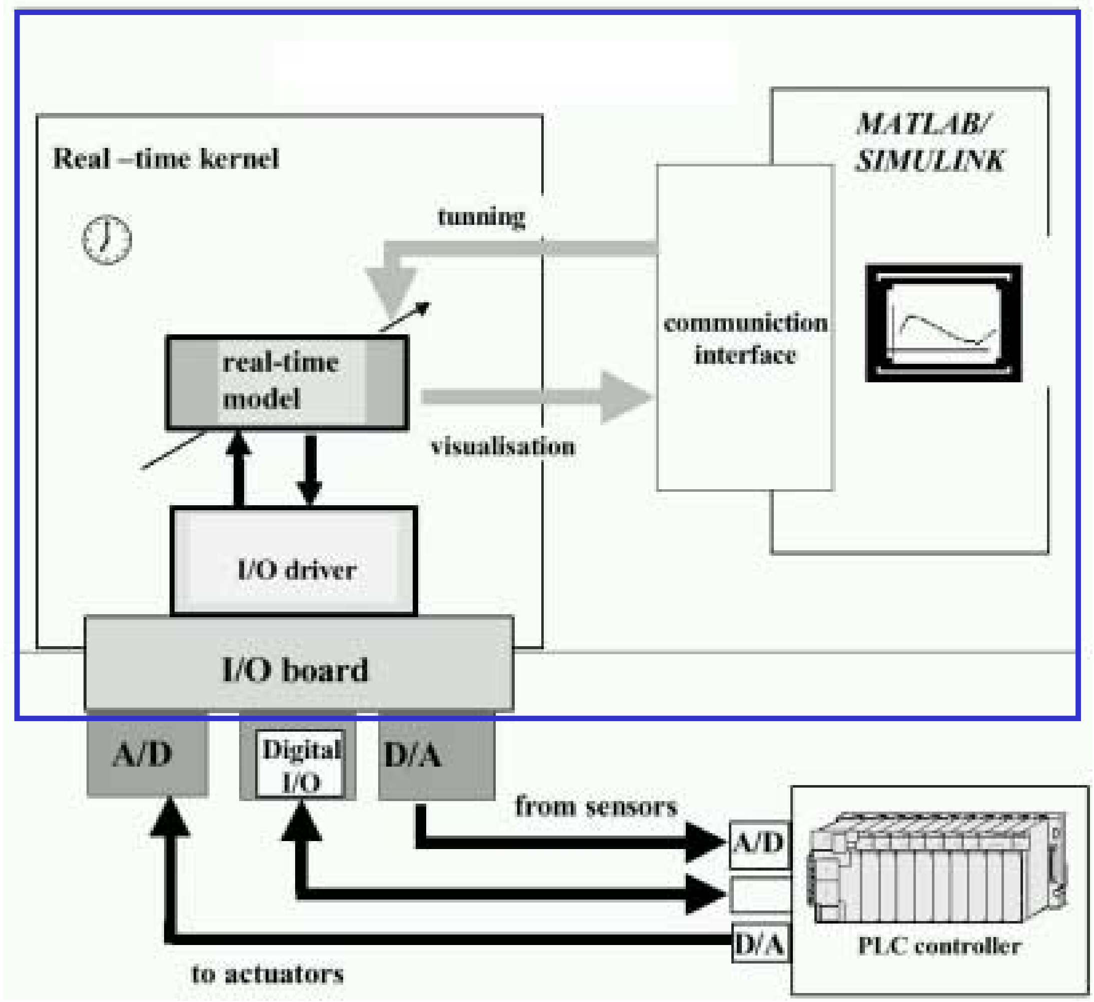
Control application software development cycle

- Matlab+toolboxes
- Simulink
- Stateflow
- Real-time Workshop

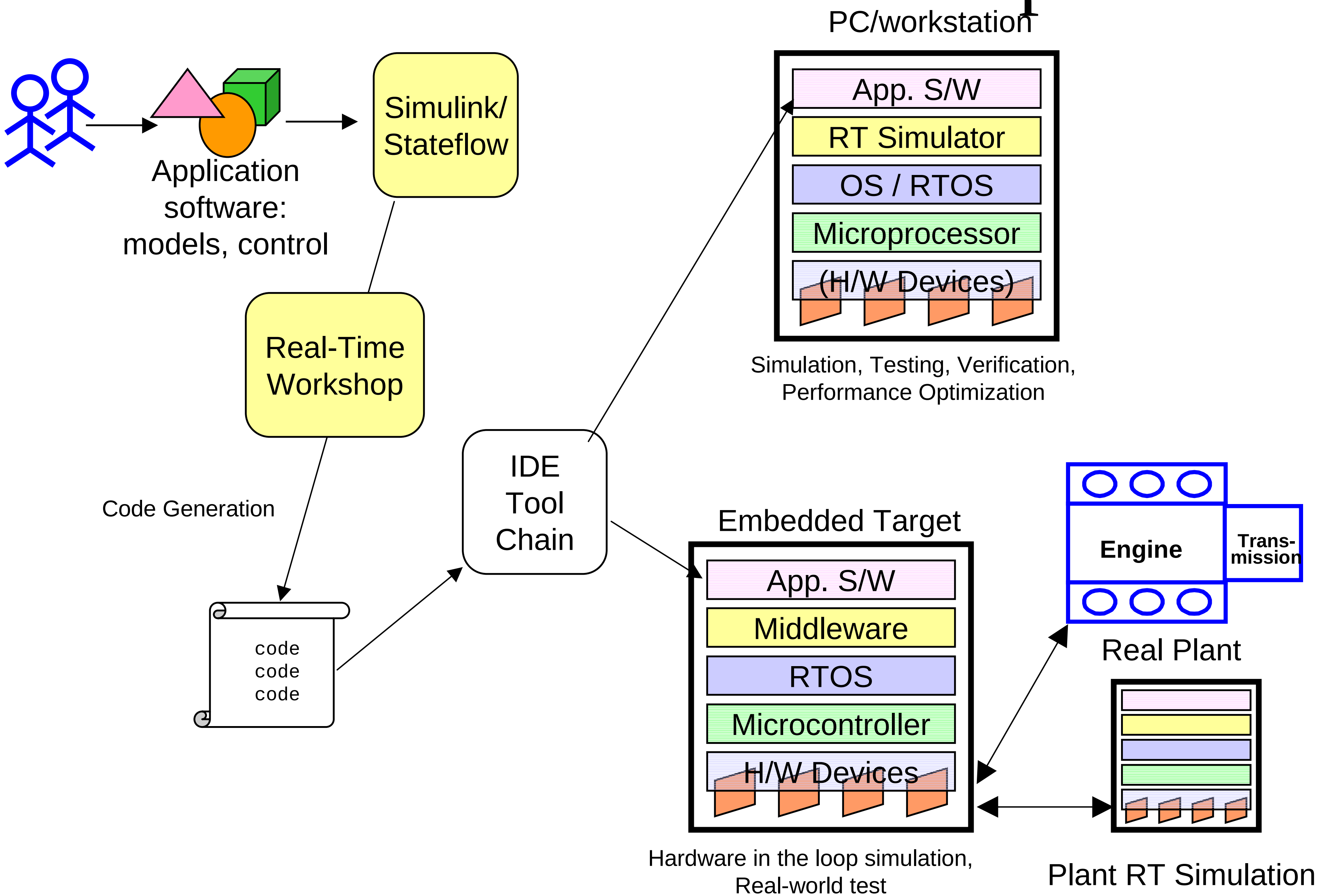


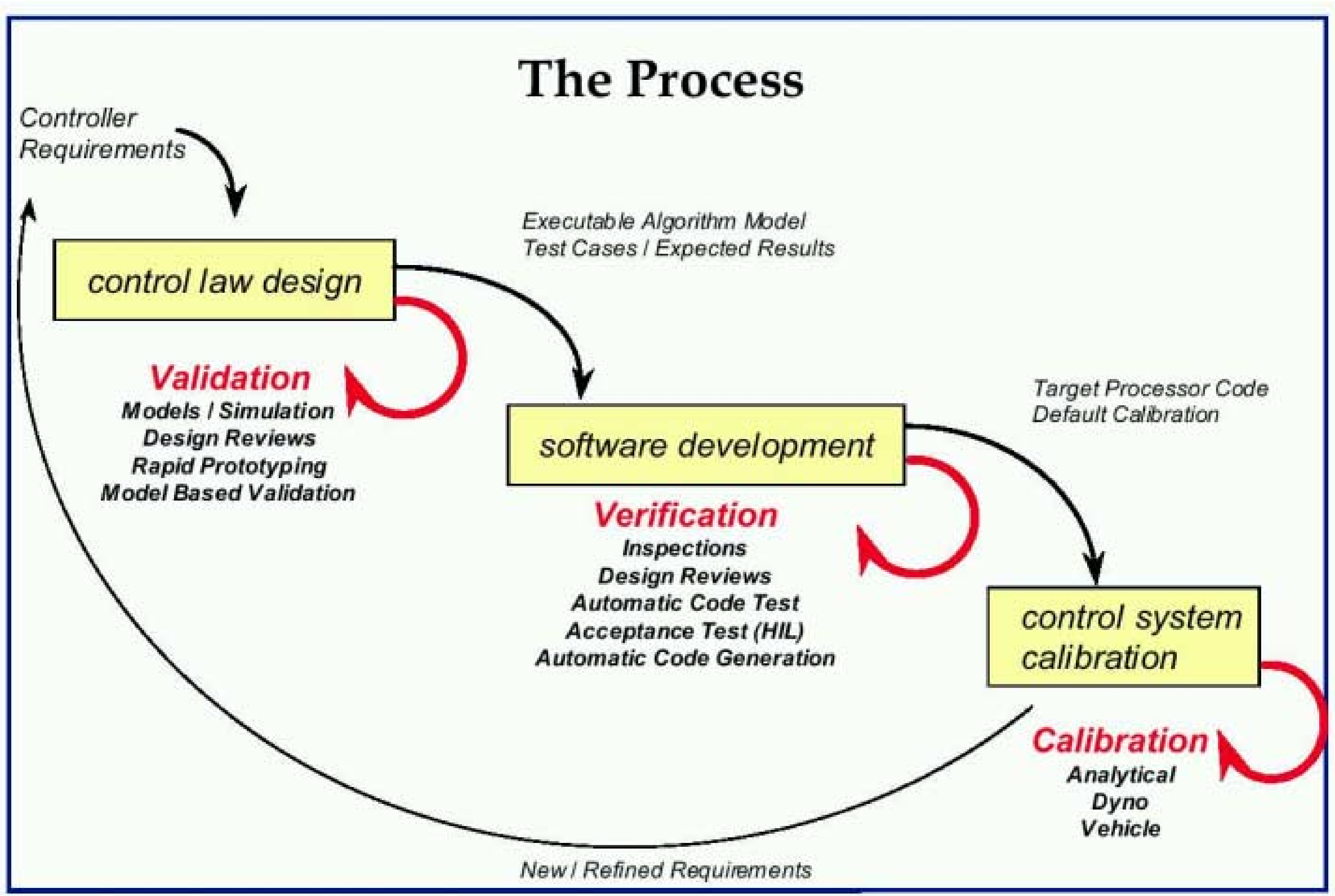
Hardware-in-the-loop simulation

- Aerospace
- Process control
- Automotive

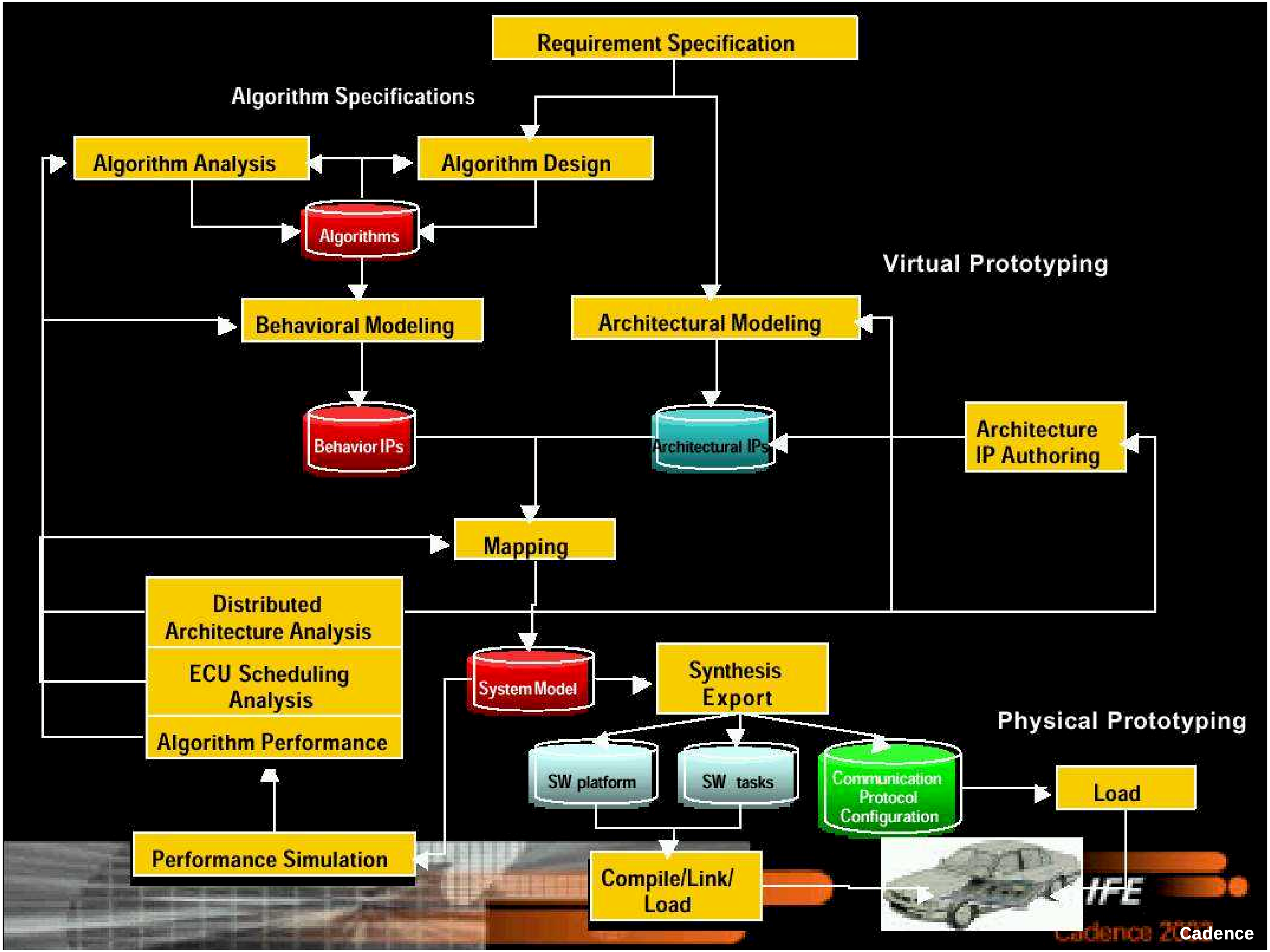


Embedded Software Development





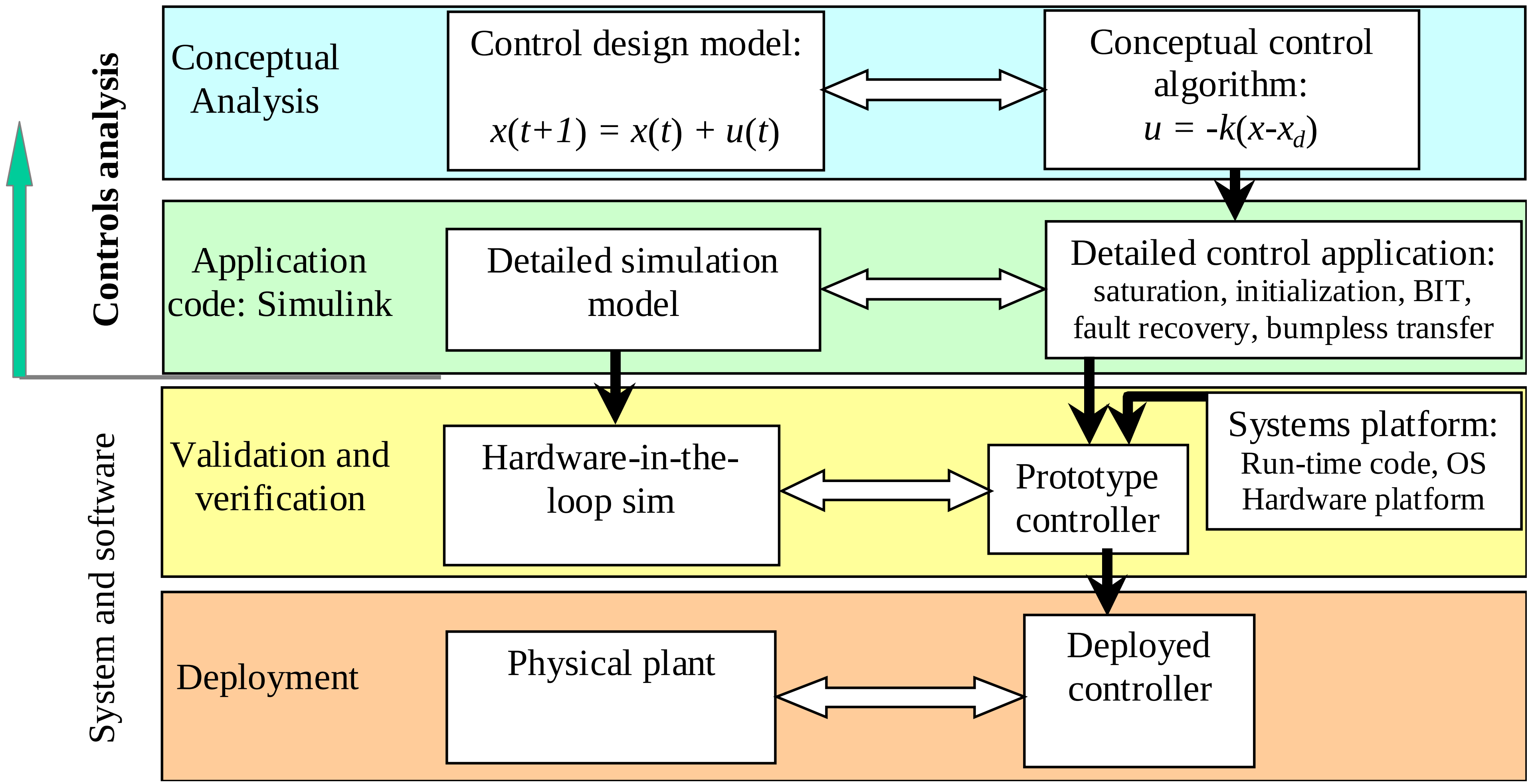
Ford Motor Company


Requirement Specification
Algorithm Specifications
Algorithm Analysis
Algorithm Design
Algorithms
Virtual Prototyping
Behavioral Modeling
Architectural Modeling
Behavior IPs
Architectural IPs
Architecture
IP Authoring
Mapping
Distributed
Architecture Analysis
ECU Scheduling
Analysis
Algorithm Performance
System Model
Synthesis
Export
Physical Prototyping
SW platform
SW tasks
Communication
Protocol
Configuration
Load
Performance Simulation
Compile/Link/
Load
Cadence

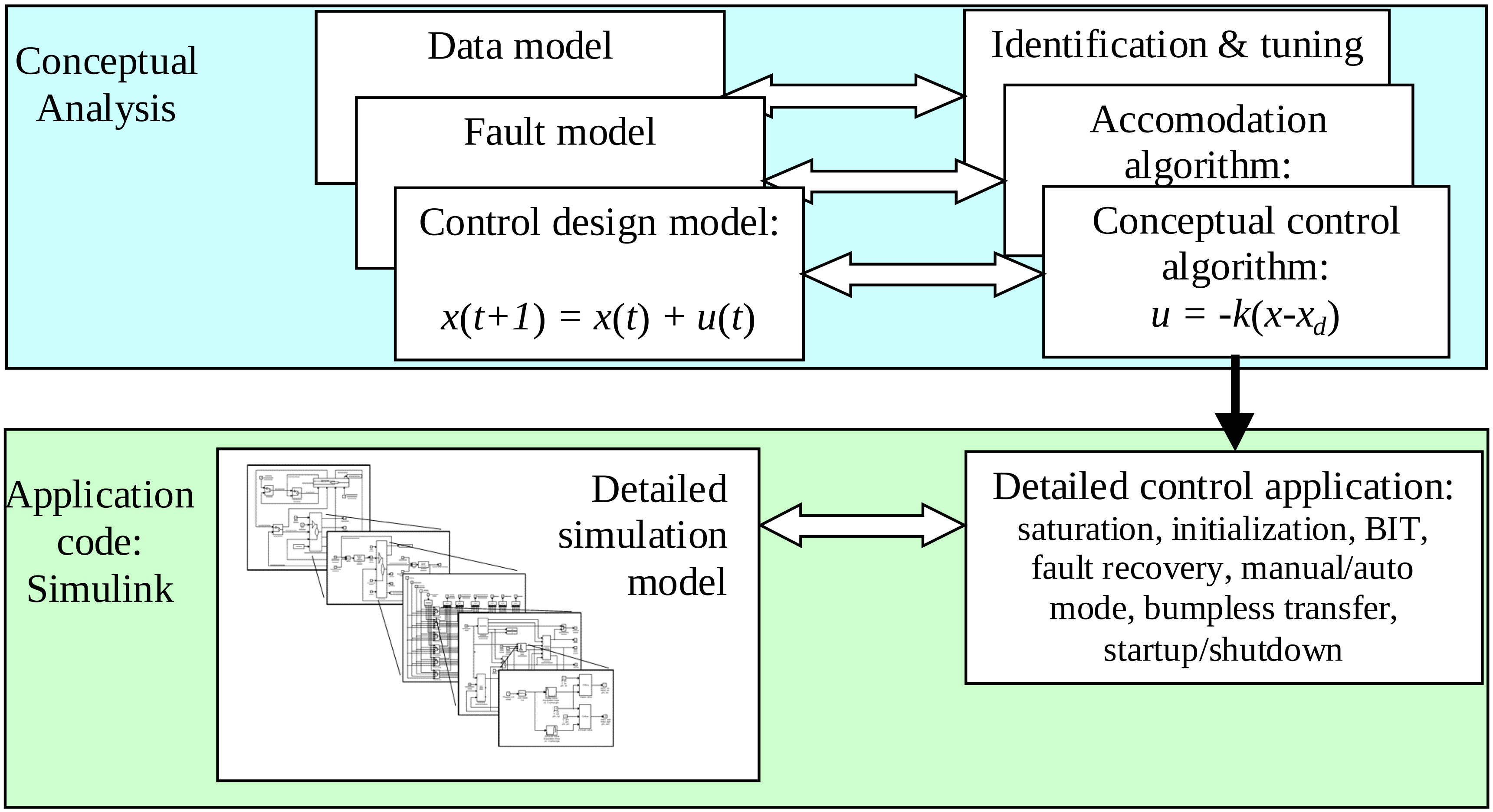
Control Technology

- Science
  - abstraction
  - concepts
  - simplified models
- Engineering
  - building new things
  - constrained resources: time, money,
- Technology
  - repeatable processes
  - control platform technology
  - control engineering technology

Controls development cycle



Controls analysis



Algorithms/Analysis

Much more than real-time control feedback computations

- modeling
- identification
- tuning
- optimization
- feedforward
- feedback
- estimation and navigation
- user interface
- diagnostics and system self-test
- system level logic, mode change

Practical Issues of Control Design

- Technical requirements
- Economics: value added, # of replications
  - automotive, telecom, disk drives - millions of copies produced
  - space, aviation - unique to dozens to several hundreds
  - process control - each process is unique, hundreds of the same type
- Developer interests
- Integration with existing system features
- Skill set in engineering development and support
- Field service/support requirements
- Marketing/competition, creation of unique IP
- Regulation/certification: FAA/FDA

Major control applications

Specialized control groups, formal development processes

- Aviation
  - avionics: Guidance, Navigation, & Control
  - propulsion - engines
  - vehicle power and environmental control
- Automotive
  - powertrain
  - suspension, traction, braking, steering
- Disk drives
- Industrial automation and process control
  - process industries: refineries, pulp and paper, chemical
  - semiconductor manufacturing processes
  - home and buildings

Commercial applications

Advanced design - commercial

- Embedded mechanical
  - mechatronics/drive control
- Robotics
  - lab automation
  - manufacturing plant robots (e.g., automotive)
  - semiconductors
- Power
  - generation and transmission
- Transportation
  - locomotives, elevators
  - marine
- Nuclear engineering

High-performance applications

Advanced design

- Defense and space
  - aero, ground, space vehicles - piloted and unmanned
  - missiles/munitions
  - comm and radar: ground, aero, space
  - campaign control: C4ISR
  - directed energy
- Science instruments
  - astronomy
  - accelerators
  - fusion: TOKAMAKs, LLNL ignition

Embedded applications

No specialized control groups

- Embedded controllers
  - consumer
  - test and measurement
  - power/current
  - thermal control
- Telecom
  - PLLs, equalizers
  - antennas, wireless, las comm
  - flow/congestion control
  - optical networks - analog, physics

Emerging control applications

A few selected cases

- Biomedical
  - life support: pacemakers anesthesia
  - diagnostics: MRI scanners, etc
  - ophthalmology
  - bio-informatics equipment
  - robotics surgery
- Computing
  - task/load balancing
- Finance and economics
  - trading

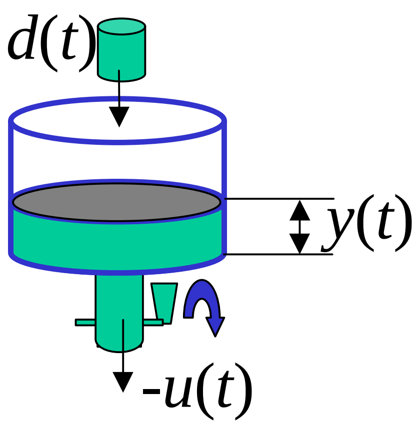
Lecture 4 - PID Control

- 90% (or more) of control loops in industry are PID
- Simple control design model $\rightarrow$ simple controller

P control

- Integrator plant:

  $$\dot{y} = u + d$$



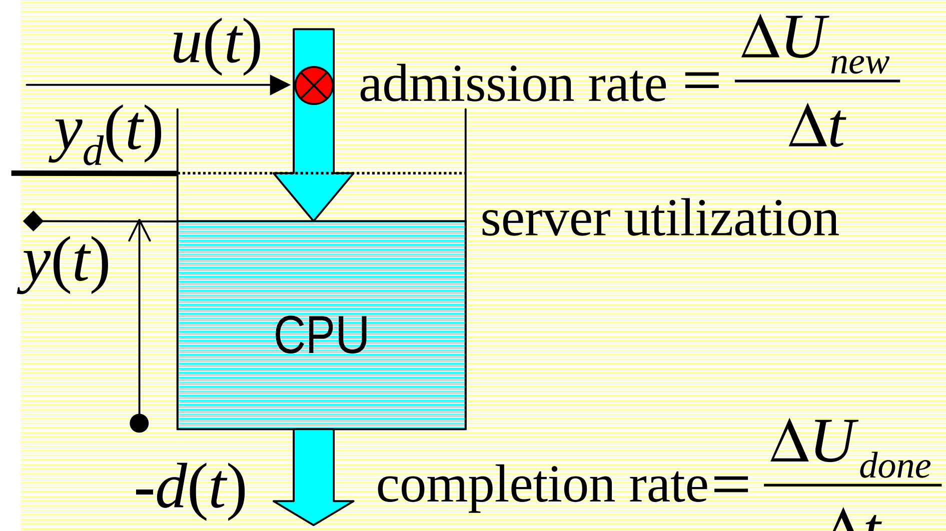
- P controller:

  $$u = -k_P (y - y_d)$$

Example:
Utilization control in a video server

Video stream *i*

- processing time $c[i]$, period $p[i]$
- CPU utilization: $U[i]=c[i]/p[i]$



P control

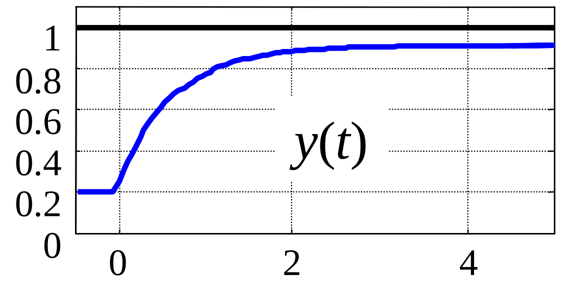
- Closed-loop dynamics

$$\dot{y} + k_P y = k_P y_d + d \qquad\qquad y = \frac{k_P}{s + k_P} y_d + \frac{1}{s + k_P} d$$

- Steady-state ($s = 0$) $y_{SS} = y_d + \frac{1}{k_P} d_{SS}$

- Transient $y(t) = y(0) e^{-t/T} + \left(y_d + \frac{1}{k_P} d_{SS} \right) \cdot \left(1 - e^{-t/T} \right)$

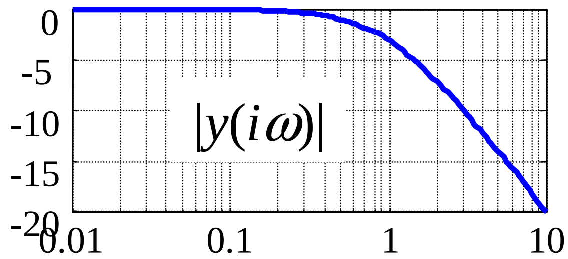
$$T = 1/k_P$$



- Frequency-domain (bandwidth)

$$y_d(t) = \hat{y}_d(i\omega) e^{i\omega t}$$
$$d(t) = \hat{d}(i\omega) e^{i\omega t}$$

$$|\hat{y}(i\omega)| = \frac{\left| \hat{y}_d(i\omega) + \hat{d}(i\omega)/k_P \right|}{\sqrt{(\omega/k_P)^2 + 1}}$$



I control

$$y = g \cdot u + d,$$

- Introduce integrator into control

$$\dot{u} = v,$$

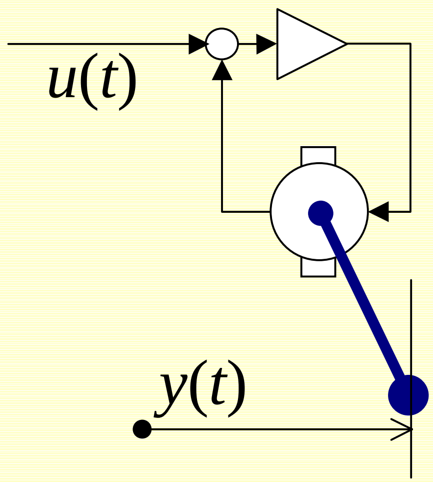
$$v = -k_I (y - y_d)$$

- Closed-loop dynamics

$$y = \frac{gk_I}{s + gk_I} y_d + \frac{s}{s + gk_I} d$$

Example:

- Servosystem command



- More:
  - stepper motor
  - flow through a valve
  - motor torque ...

Sampled time I control

- Step to step update:

$$y(t) = g \cdot u(t) + d(t)$$
$$u(t) = u(t-1) + v(t-1)$$
$$v(t) = k_I\left[y(t) - y_d\right]$$

sampled time integrator

- Closed-loop dynamics

$$y = g \cdot u + d$$
$$u = \frac{k_I}{z-1}\left[y - y_d\right]$$
$$\Longrightarrow \quad y = \frac{gk_I}{z-1+gk_I} y_d + \frac{z-1}{z-1+gk_I} d$$

- Deadbeat control: $gk_I = 1 \quad \Longrightarrow \quad y = z^{-1} y_d + \left(1 - z^{-1}\right) d$

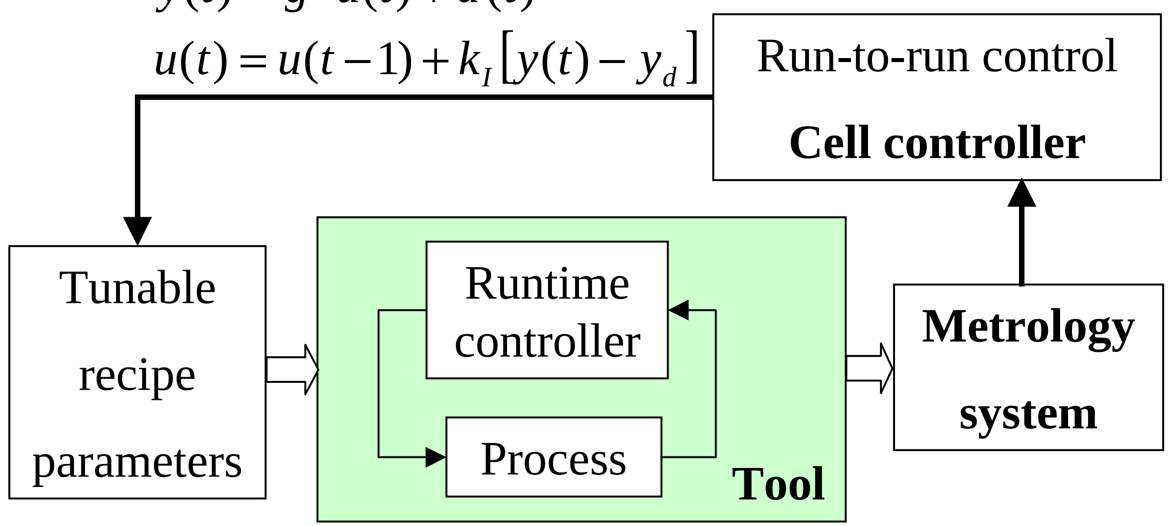
Run-to-run (R2R) control

- Main APC (Advanced Process Control) approach in semiconductor processes
- Modification of a product recipe between tool "runs"

- Processes:
  - vapor phase epitaxy
  - lithography
  - chemical mechanical planarization (CMP)
  - plasma etch

$$y(t) = g \cdot u(t) + d(t)$$

$$u(t) = u(t-1) + k_I \left[y(t) - y_d \right]$$



PI control

- First-order system:

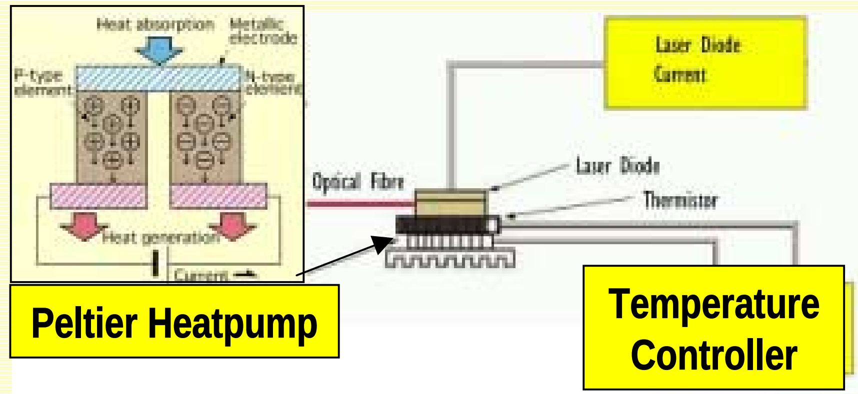
$$\tau\dot{y} = -y + u + d$$

- P control + integrator for cancelling steady state error

$$e = y - y_d;$$
$$\dot{v} = e$$
$$u = -k_I v - k_P e$$

Example:

- WDM laser-diode temperature control



$y(t)$ = temperature - ambient temperature

heat capacity; heat loss to environment; pumped heat; produced heat

$$\tau\dot{y} = -y + u + d$$

- Other applications
  - ATE
  - EDFA optical amplifiers
  - Fiber optic laser modules
  - Fiber optic network equipment

PI control

- P Control + Integrator for cancelling steady state error

$$e = y - y_d;$$

$$\dot{v} = e$$

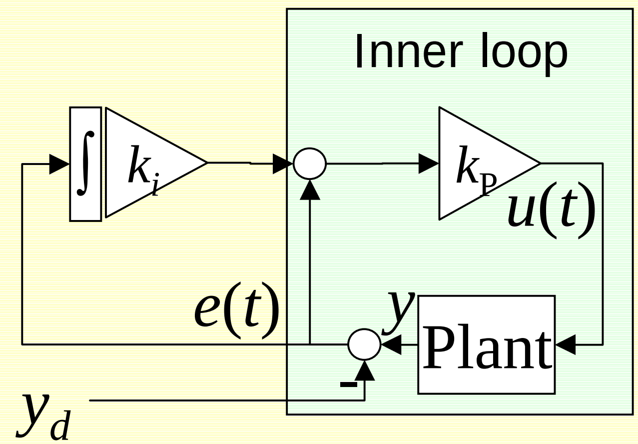
$$u = -k_I v - k_P e = k_P (e - \underbrace{k_i}_{\frac{k_I}{k_P}} v)$$

- Velocity form of the controller

$$\dot{u} = -k_I e - k_P \dot{e}$$

$$u(t+1) = u(t) - k_I e(t) - k_P \left[e(t) - e(t-1) \right]$$

Cascade loop interpretation:



$$k_I = k_i \cdot k_P$$

PI control

- Closed-loop dynamics

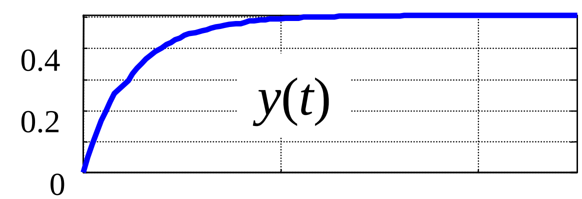
$$y = \frac{sk_P + k_I}{s(\tau s + 1) + sk_P + k_I} y_d + \frac{s}{s(\tau s + 1) + sk_P + k_I} d$$

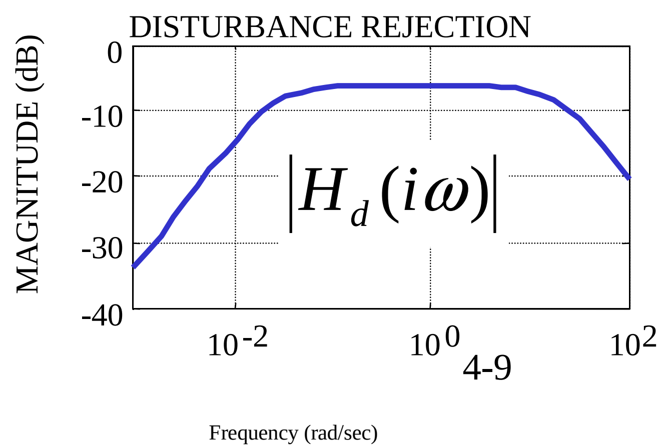
- Steady state (s = 0): $y_{SS} = y_d$.
  No steady-state error!
- Transient dynamics: look at the characteristic equation

$$\tau\lambda^2 + (1 + k_P)\lambda + k_I = 0$$

- Disturbance rejection

$$\left|\hat{y}(i\omega)\right| = \left|H_d(i\omega)\right| \cdot \left|\hat{d}(i\omega)\right|$$





PLL Example

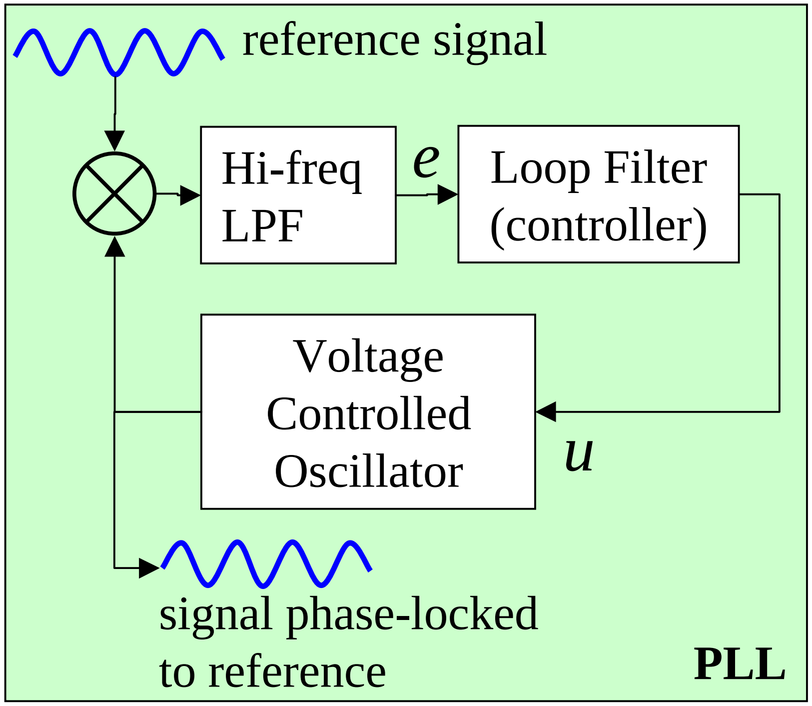
- Phase-locked loop is arguably a most prolific feedback system

$$e = 2K_m \mathrm{LPF}\langle r \times v \rangle$$

$$= 2K_m \mathrm{LPF}\langle A\sin(\omega t + \theta_d) \times \cos(\omega_o t + \theta_o) \rangle$$

$$e \approx AK_m \sin(\omega t - \omega_o t + \theta_d - \theta_o)$$

$$\dot{\theta}_o = \Delta\omega_o = K_o u$$



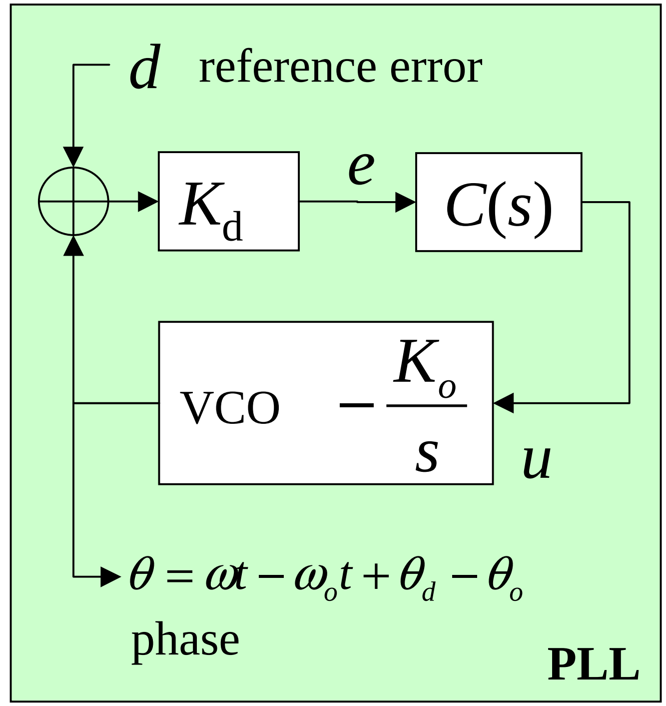
PLL Loop Model

- Small-signal model:

$$\theta = \omega t - \omega_o t + \theta_d - \theta_o << 1$$

$$e = K_d \sin(\theta) \approx K_d \theta$$

$$\dot{\theta} = \underbrace{\omega - \omega_o + \dot{\theta}_d}_{d} - \underbrace{K_o u}_{\theta_o}$$



- Loop dynamics:

$$\dot{\theta} = d - K_o u$$
$$e = K_d \theta$$
$$u = k_P e + k_I \int e \cdot dt$$

$$\Rightarrow \quad \theta = \frac{s}{s^2 + K_o K_d k_P s + k_I} d$$

PD control

- 2-nd order dynamics

$$\ddot{y} = u + d$$

- PD control

$$e = y - y_d$$

$$u = -k_D \dot{e} - k_P e$$

- Closed-loop dynamics

$$\ddot{e} + k_D \dot{e} + k_P e = d$$

$$e = \frac{1}{s^2 + k_D s + k_P} d$$

- Optimal gains (critical damping)

$$k_D = 2\tau; \qquad k_P = \tau^2$$

Example:

- Disk read-write control

$$J\ddot{\varphi} = T_{VCM} + T_{DISTURB}$$

Voice Coil Motor

PD control

- Derivative (rate of *e*) can be obtained
  - speed sensor (tachometer)
  - low-level estimation logic
- Signal differentiation
  - is noncausal
  - amplifies high-frequency noise
- Causal (low-pass filtered) estimate of the derivative

$$\dot{e} \approx \frac{s}{\tau_D s + 1} e = \frac{1}{\tau_D} e + \frac{1/\tau_D}{\tau_D s + 1} e$$

- Modified PD controller:

$$u = -k_D \frac{s}{\tau_D s + 1} e - k_P e$$

PD control performance

- The performance seems to be infinitely improving for

$$k_D = 2\tau; k_P = \tau^2; \qquad \tau \to \infty$$

- This was a simple design model, remember?
- Performance is limited by
  - system being different from the model
    - flexible modes, friction, VCM inductance
  - sampling in a digital controller
  - rate estimation would amplify noise if too aggressive
  - actuator saturation
  - you might really find *after* you have tried to push the performance
- If high performance is really that important, careful application of more advanced control approaches might help

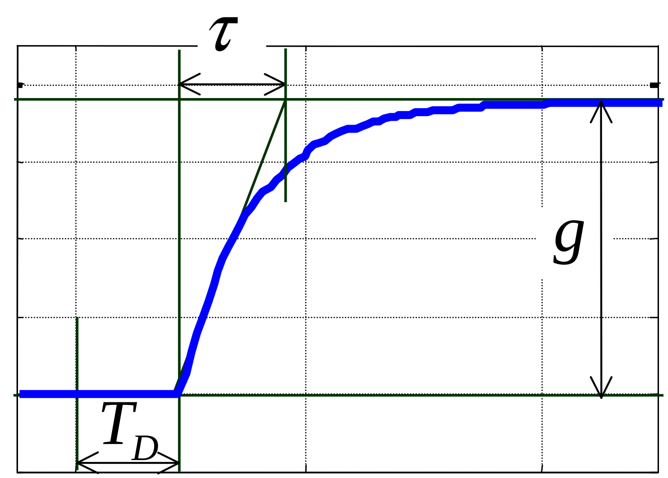
Plant Type

- Constant gain - I control
- Integrator - P control
- Double integrator - PD control
- Generic second order dynamics - PID control

PID Control

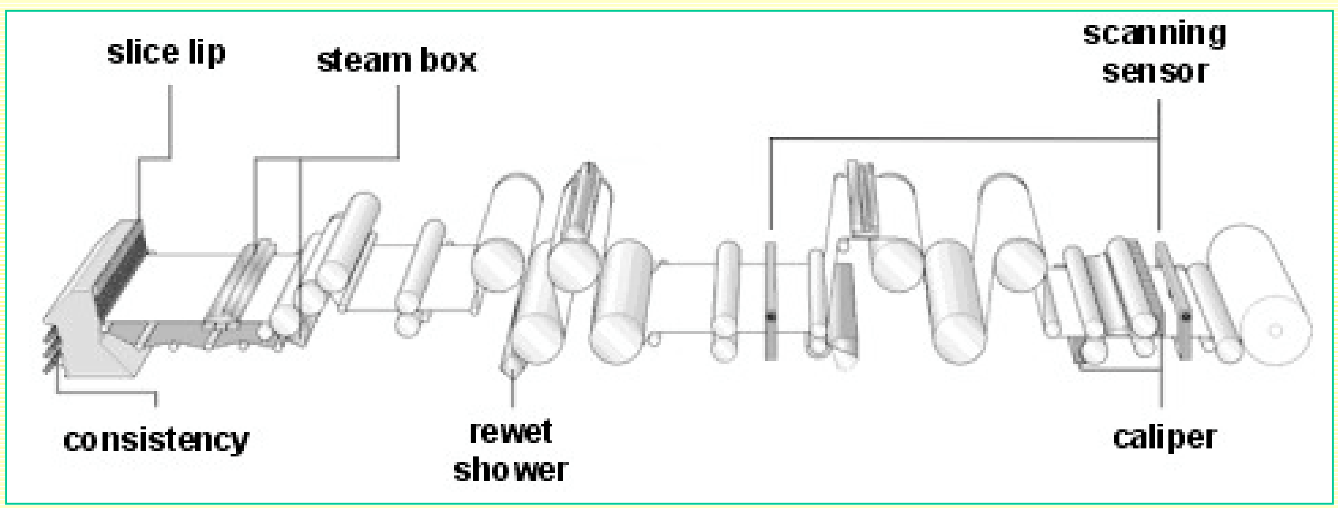
- Generalization of P, PI, PD
- Early motivation: control of first order processes with deadtime

$$y = \frac{g e^{-T_D s}}{\tau s + 1} u$$



Example:

- Paper machine control



PID Control

- PID: three-term control

$$e = y - y_d$$

$$u = -k_D \dot{e} - k_P e - k_I \int e \cdot dt$$

(independent sensor or an estimate)

- Sampled-time PID

$$u = -k_D\left(1 - z^{-1}\right)e + k_P e + k_I \frac{1}{1 - z^{-1}} e$$

(future) (present) (past)

- Velocity form
  - bumpless transfer between manual and automatic

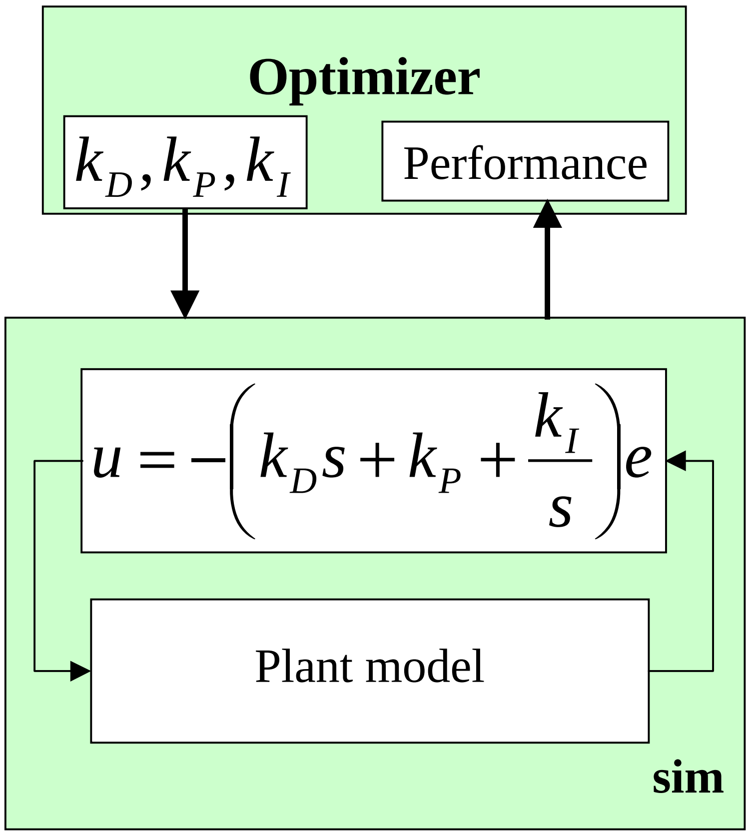
$$\Delta u = -k_D \Delta^2 e - k_P \Delta e - k_I e$$

$$\Delta = 1 - z^{-1}$$

$$u(t+1) = u(t) - k_I e(t) - k_P\left(e(t) - e(t-1)\right) - k_D\left(e(t) - 2e(t-1) + e(t-2)\right)$$

Tuning PID Control

- Model-based tuning
- Look at the closed-loop poles
- Numerical optimization
  - For given parameters run a sim, compute performance parameters and a performance index
  - Optimize the performance index over the three PID gains using grid search or Nelder method.



Zeigler-Nichols tuning rule

- Explore the plant:
  - set the plant under P control and start increasing the gain till the loop oscillates
  - note the critical gain k_C and oscillation period T_C
- Tune the controller:

	k_P	k_I	k_D
P	$0.5k_C$	——	——
PI	$0.45k_C$	$1.2k_P/T_C$	——
PID	$0.5k_C$	$2k_P/T_C$	$k_P T_C/8$

- Z and N used a Monte Carlo method to develop the rule
- Z-N rule enables tuning if a model and a computer are both unavailable, only the controller and the plant are.

Integrator anti wind-up

- In practice, control authority is always limited:
  - $u_{\text{MIN}} \le u(t) \le u_{\text{MAX}}$
- Wind up of the integrator:
  - if $|u_c| > u_{MAX}$ the integral v will keep growing while the control is constant. This results in a heavy overshoot later
- Anti wind-up:
  - switch the integrator off if the control has saturated

$$\dot{v} = e$$

$$u_c = -k_I v - k_P e$$

$$u = \begin{cases} u_{MAX}, & u_c > u_{MAX} \\ u_c, & u_{MIN} \le u_c \le u_{MAX} \\ u_{MIN}, & u_c < u_{MIN} \end{cases}$$

$$\dot{v} = \begin{cases} e, & \text{for } u_{MIN} \le u_c \le u_{MAX} \\ 0, & \text{if } u_c > u_{MAX} \text{ or } u_c < u_{MIN} \end{cases}$$

Industrial PID Controller

- A box, not an algorithm
- Auto-tuning functionality:
  - pre-tune
  - self-tune
- Manual/cascade mode switch
- Bumpless transfer between different modes, setpoint ramp
- Loop alarms
- Networked or serial port

Lecture 5 - Feedforward

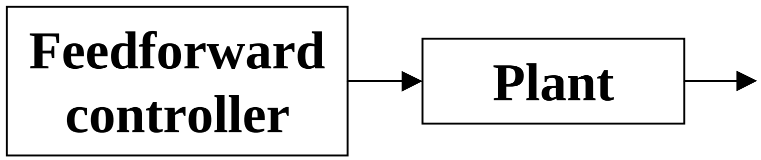
- Programmed control
- Path planning and nominal trajectory feedforward
- Feedforward of the disturbance
- Reference feedforward, 2-DOF architecture
- Non-causal inversion
- Input shaping, flexible system control
- Iterative update of feedforward

Why Feedforward?

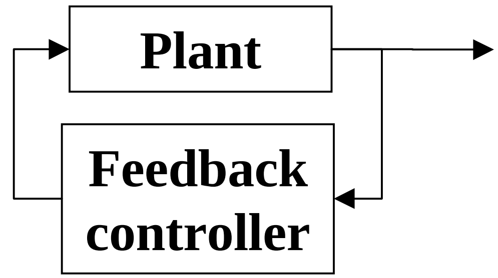
- Feedback works even if we know little about the plant dynamics and disturbances
- Was the case in many of the first control systems
- Much attention to feedback - for historical reasons

- Open-loop control/feedforward is increasingly used
- Model-based design means we know something
- The performance can be greatly improved by adding open-loop control based on our system knowledge (models)

Feedforward

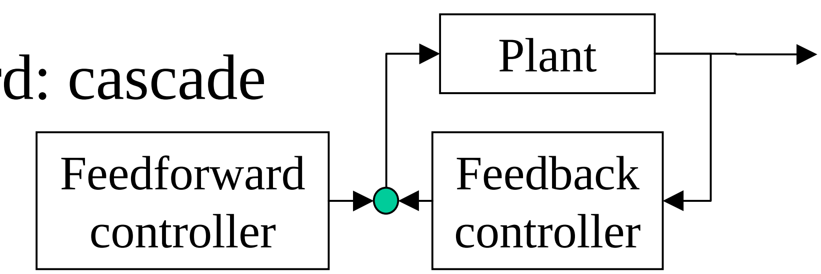


– this Lecture 5



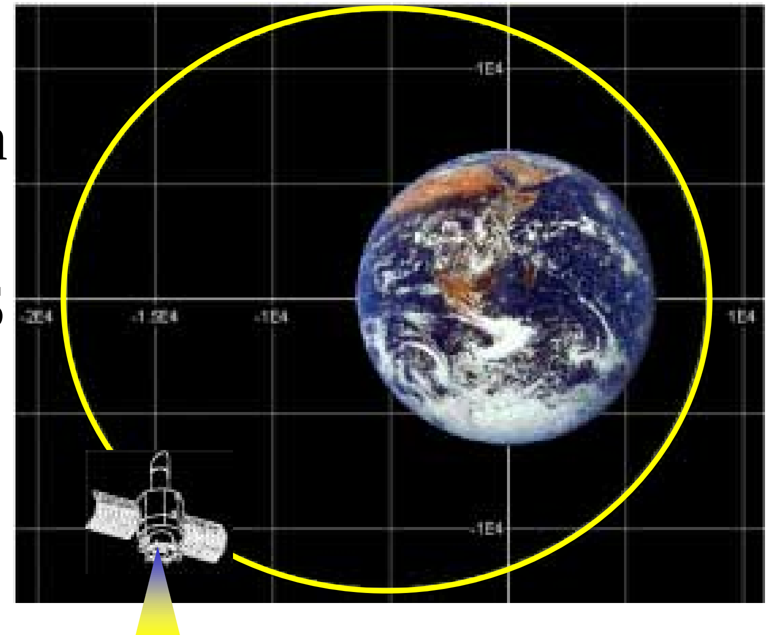
– Lecture 4 PID
– Lecture 6 Analysis
– Lecture 7 Design

- Main premise of the feedforward control: a model of the plant is known
- Model-based design of feedback control - the same premise
- The difference: feedback control is less sensitive to modeling error
- Common use of the feedforward: cascade with feedback



Open-loop (programmed) control

- Control $u(t)$ found by solving an optimization problem. Constraints on control and state variables.
- Used in space, missiles, aircraft FMS
  - Mission planning
  - Complemented by feedback corrections
- Sophisticated mathematical methods were developed in the 60s to overcome computing limitations.
- Lecture 12 will get into more detail of control program optimization.



$$\dot{x} = f(x,u,t)$$

$$J(x,u,t) \rightarrow \min$$

$$x \in \mathbf{X}, u \in \mathbf{U}$$

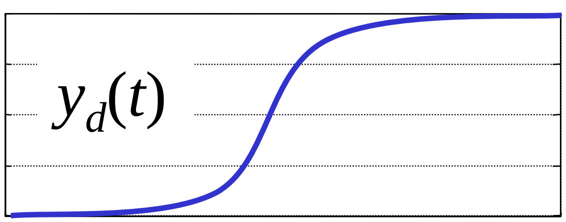
Optimal control: $u = u_*(t)$

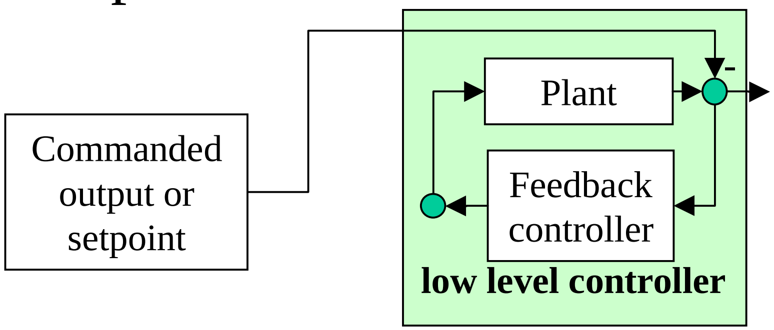
Optimal control

- Performance index and constraints
- Programmed control
  - compute optimal control as a time function for particular initial (and final) conditions
- Optimal control synthesis
  - find optimal control for *any* initial conditions
  - at any point in time apply control that is optimal now, based on the current state. This is *feedback* control!
  - example: LQG for linear systems, gaussian noise, quadratic performance index. Analytically solvable problem.
  - simplified model, toy problems, conceptual building block
- MPC - will discuss in Lecture 12

Path/trajectory planning

- The disturbance caused by the change of the command *r* influences the feedback loop.
- The error sensitivity to the reference $R(s)$ is bandpass: $|R(i\omega)| << 1$ for ω small
- A practical approach: choose the setpoint command (path) as a smooth function that has no/little high-frequency components. No feedforward is used.
- The smooth function can be a spline function etc

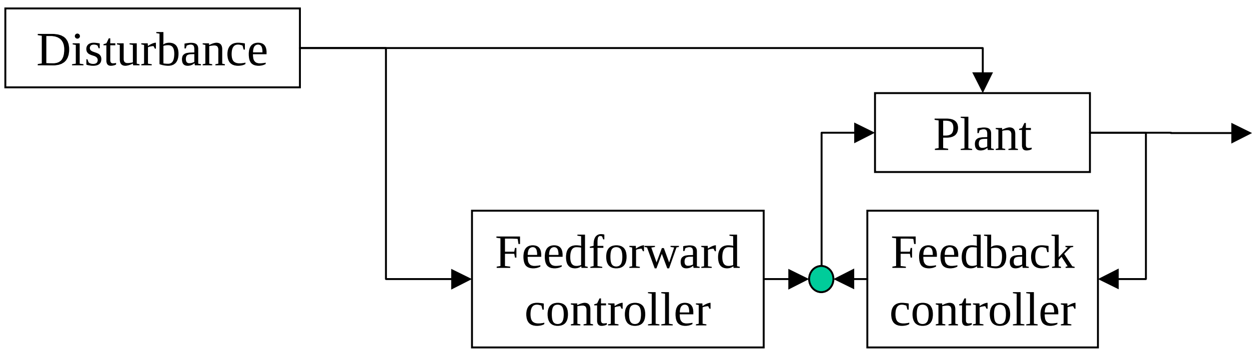




Disturbance feedforward

- Disturbance acting on the plant is measured
- Feedforward controller can react *before* the effect of the disturbance shows up in the plant output

Example:
Temperature control. Measure ambient temperature and adjust heating/cooling
• homes and buildings
• district heating
• industrial processes - crystallization
• electronic or optical components

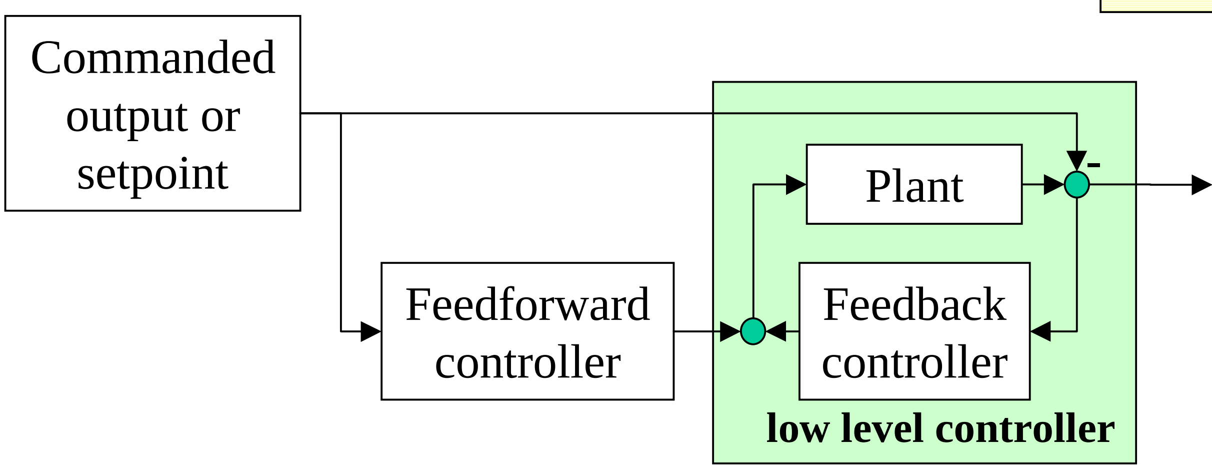


Command/setpoint feedforward

- The setpoint change acts as disturbance on the feedback loop.
- This disturbance can be measured
- 2-DOF controller

Examples:

•Servosystems
– robotics
•Process control
– RTP
•Automotive
– engine torque demand



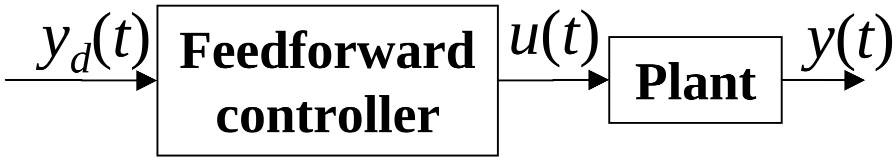
Feedforward as system inversion

$$y = P(s)u$$

$$y = y_d \Rightarrow u = [P(s)]^{-1} y_d$$

$$e = P(s)u + D(s)d$$

$$y_d \equiv -D(s)d$$



- Simple example:

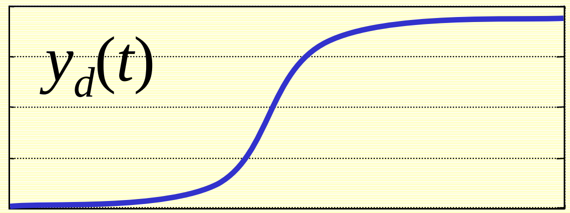
$$P(s) = \frac{1+2s}{1+s}$$

$$[P(s)]^{-1} = \frac{1+s}{1+2s}$$

More examples:

•Disk drive long seek

•Robotics: tracking a trajectory



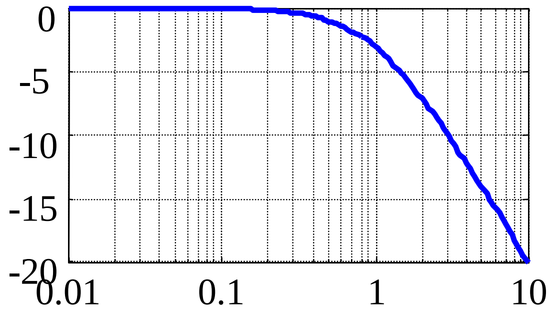
Feedforward as system inversion

$$y = P(s)u$$

$$y = y_d \Rightarrow u = \left[P(s)\right]^{-1} y_d$$

$$\tilde{u}(i\omega) = \frac{\tilde{y}_d(i\omega)}{P(i\omega)}$$

- Issue
  - high-frequency roll-off



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

$$\left[P(s)\right]^{-1} = 1 + s$$ non-proper

- Approximate inverse solution:
  - ignore high frequency in some way

Proper transfer functions

- Proper means deg(Denominator) ≥ deg(Numerator)
- Strictly proper <=> high-frequency roll-off, all physical dynamical systems are like that
- Proper = strictly proper + feedthrough
- State space models are always proper
- Exact differentiation is noncausal, non-proper
- Acceleration measurement example

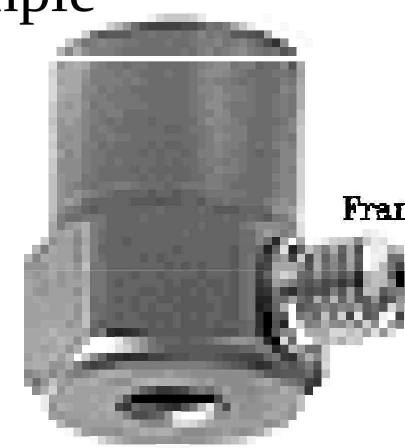
$$m\ddot{x} = u \qquad a = \ddot{x}$$

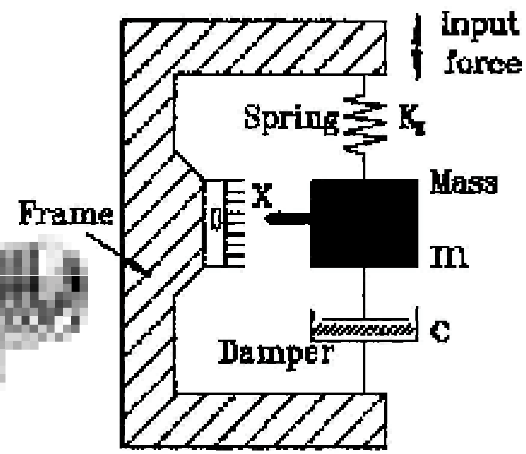
$$u = ma - k(x - x_d)$$

$$\Rightarrow x = x_d$$

this is wrong!

accelerometer





Differentiation

- Path/trajectory planning - mechanical servosystems
- The derivative can be computed if $y_d(t)$ is known ahead of time (no need to be causal then).

$$P^{-1}(s)y_d = \frac{1}{P(s)} \cdot \frac{1}{s^n} y_d^{[n]}, \qquad y_d^{[n]}(t) = \frac{d^n y}{dt^n}(t)$$

$$P(s) = \frac{1}{1+s} \qquad P^{-1}(s)y_d = \frac{1+s}{s}\dot{y}_d = \left(1+\frac{1}{s}\right)\dot{y}_d = \dot{y}_d + y_d$$

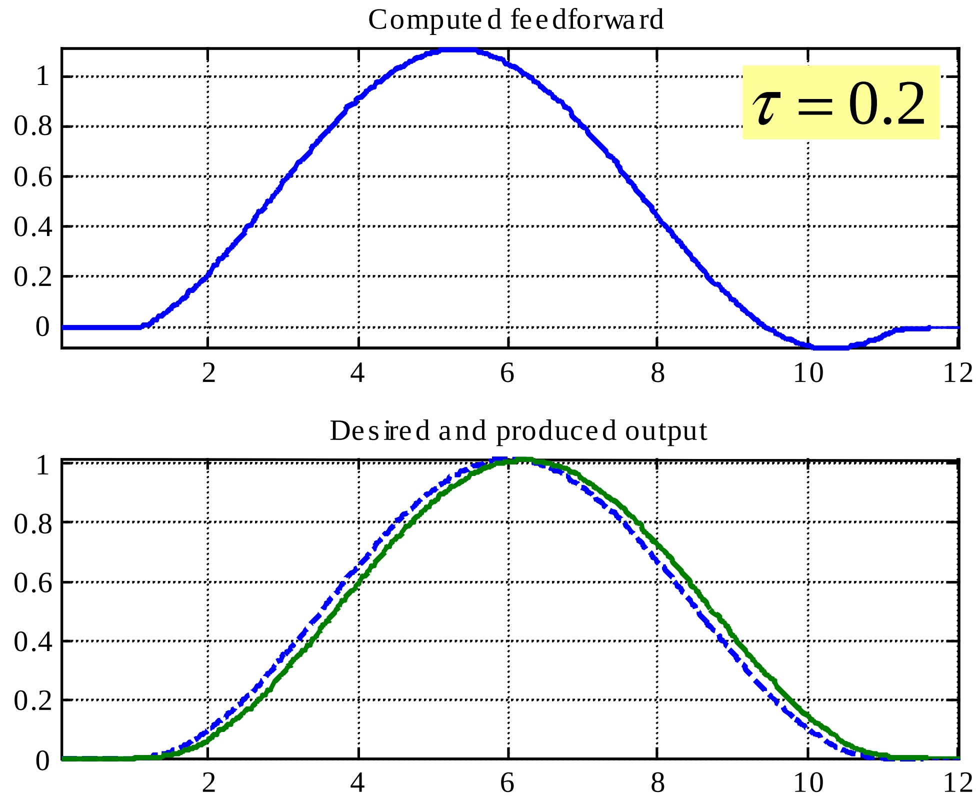
Approximate Differentiation

- Add low pass filtering:

$$P^{\dagger}(s) = \frac{1}{(1+\tau s)^n} \cdot \frac{1}{P(s)}$$

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

$$P^{\dagger}(s) = \frac{1}{1+\tau s} \cdot (1+s)$$



‘Unstable’ zeros

- Nonminimum phase system
  - r.h.p. zeros $\rightarrow$ r.h.p. poles
  - approximate solution: replace r.h.p. zeros by l.h.p. zeros

$$P(s) = \frac{1-s}{1+0.25s}, \qquad P^{\dagger}(s) = \frac{1+0.25s}{1+s}$$

- RHP zeros might be used to approximate dead time
  - exact causal inversion impossible

$$P(s) = e^{-2Ts} \approx \frac{1-sT}{1+sT}$$

- If preview is available, use a lead to compensate for the deadtime

Two sided *z*-transform, non-causal system

- Linear system is defined by a pulse response. Do not constrain ourselves with a causal pulse response anymore

$$y(x) = \sum_{k=-\infty}^{\infty} h(x-k)u(k)$$

- 2-sided z-transform gives a “transfer function”

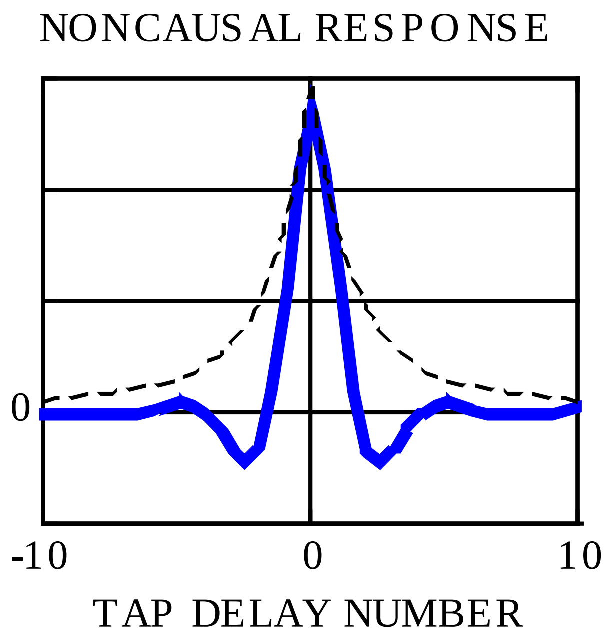
$$P(z) = \sum_{k=-\infty}^{\infty} h(k)z^{-k}$$

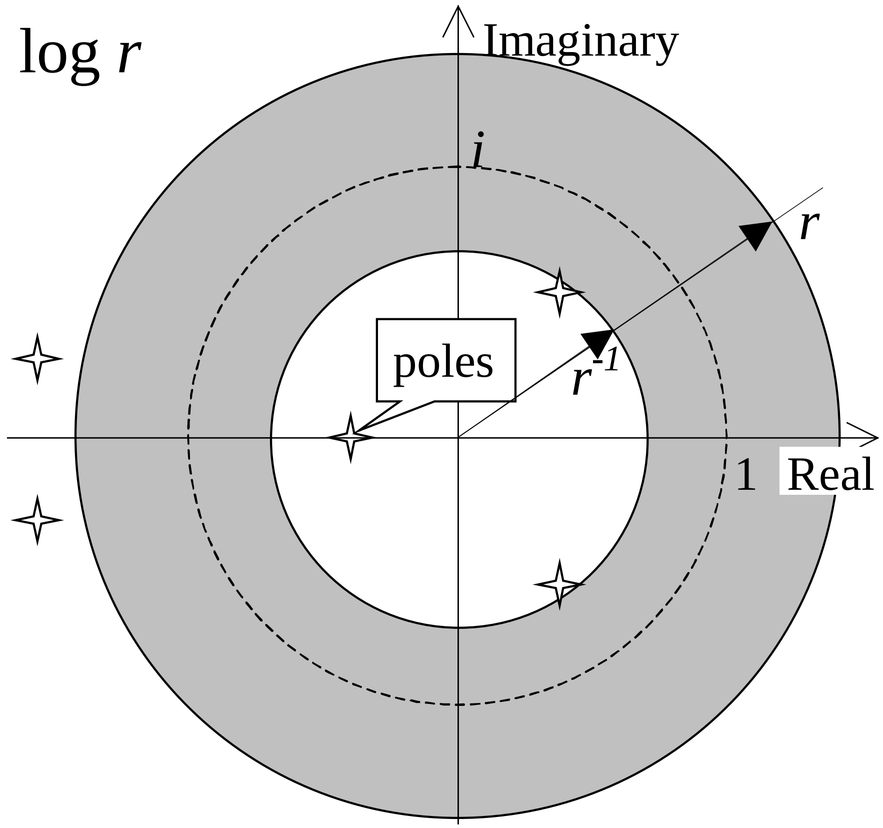
- Fourier transform/Inverse Fourier transform are two-sided

• Oppenheim, Schafer, and Buck, *Discrete-Time Signal Processing*, 2nd Edition, Prentice Hall, 1999.

Impulse response decay

- Decay rate from the center = $\log r$





Non-causal inversion

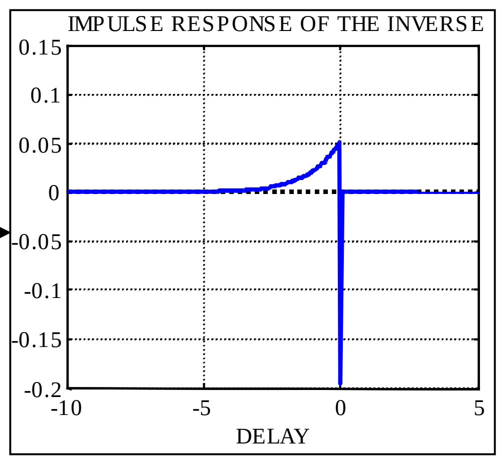
- Causal/anti-causal decomposition
  - 2-sided Laplace-transform

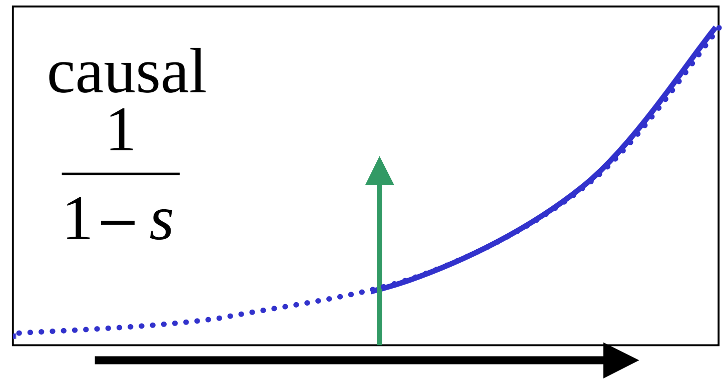
$$P(s) = \frac{1-s}{1+0.25s}$$

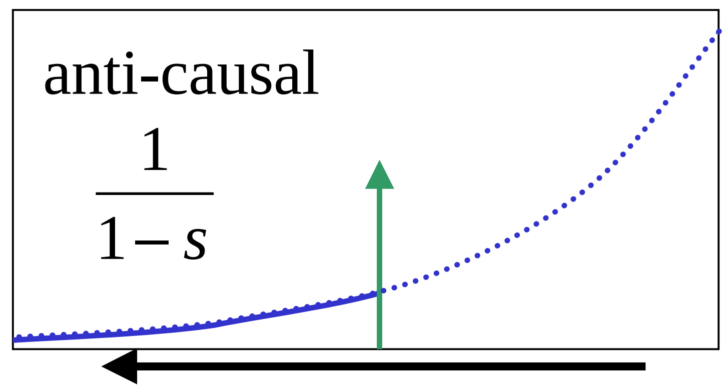
$$P^{-1}(s) = \frac{1+0.25s}{1-s} = -0.25 + \underbrace{\frac{1.25}{1-s}}_{\leftarrow}$$

$$P^{-1}(i\omega) = \frac{1}{P(i\omega)}$$

→ iFFT →







Frequency domain inversion

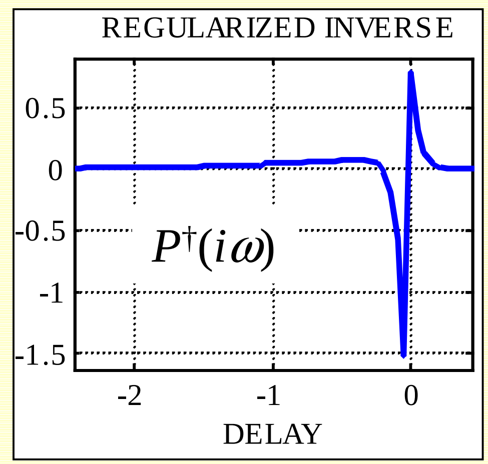
- Regularized inversion: $\left\| y_d - Pu \right\|_2^2 + \rho \left\| u \right\|_2^2 \to \min$

$$\int \left(\left| y_d(i\omega) - P(i\omega)u(i\omega) \right|^2 + \rho \left| u(i\omega) \right|^2 \right) d\omega \to \min$$

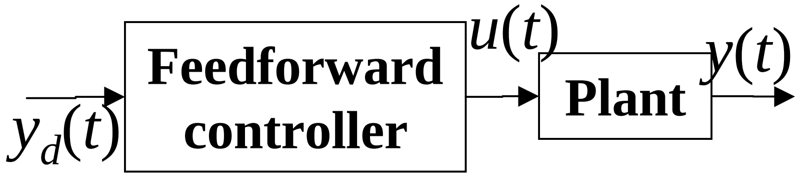
$$u(i\omega) = \frac{P^*(i\omega)}{P^*(i\omega)P(i\omega) + \rho} y_d(i\omega) = P^\dagger(i\omega) y_d(i\omega)$$

- Systematic solution
  - simple, use FFT
  - takes care of everything
  - noncausal inverse
  - high-frequency roll-off
  - Paden & Bayo, 1985(?)

$$P(s) = \frac{1 - s}{(1 + 0.25s)(1 + s)}$$

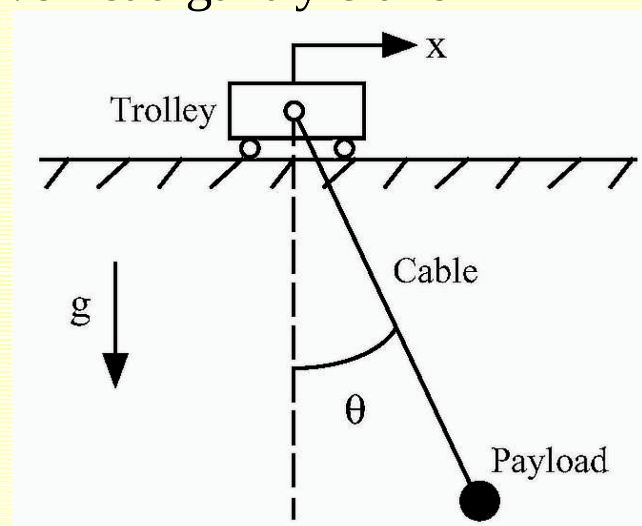


Input Shaping: point-to-point control



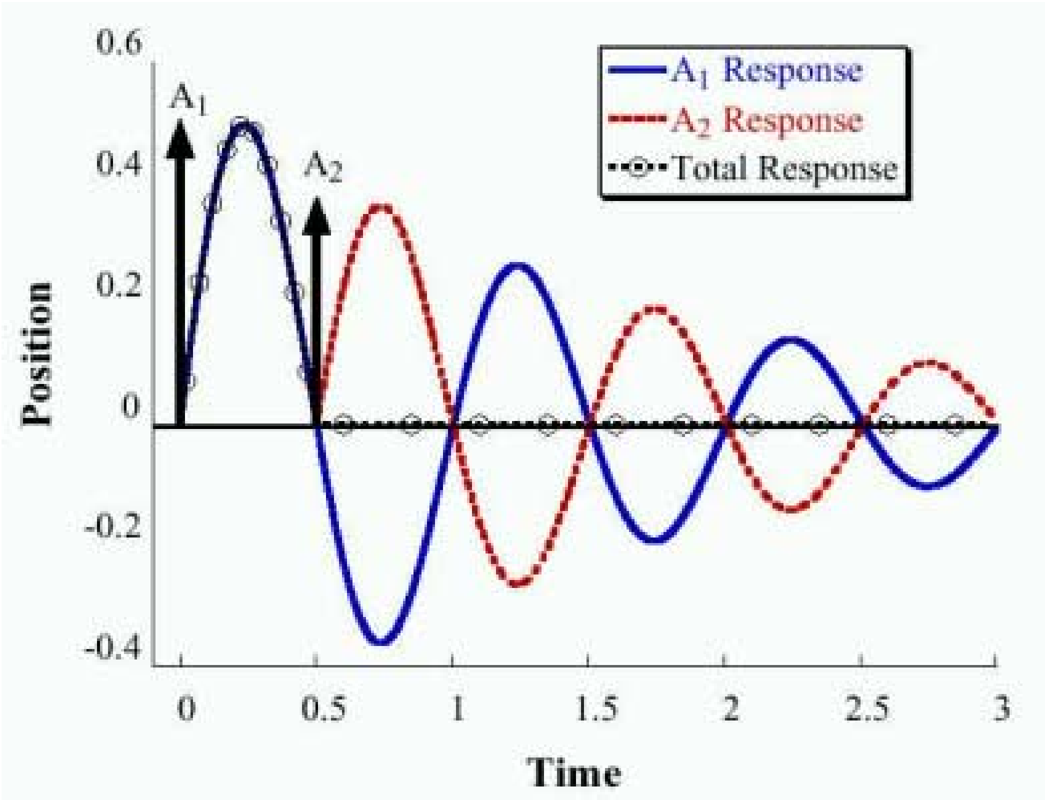
- Given initial and final conditions find control input
- No intermediate trajectory constraints
- Lightly damped, imaginary axis poles
  - preview control does not work
  - other inversion methods do not work well
- FIR notch fliter
  - Seering and Singer, MIT
  - Convolve Inc.

Examples:
- Disk drive long seek
- Flexible space structures
- Overhead gantry crane



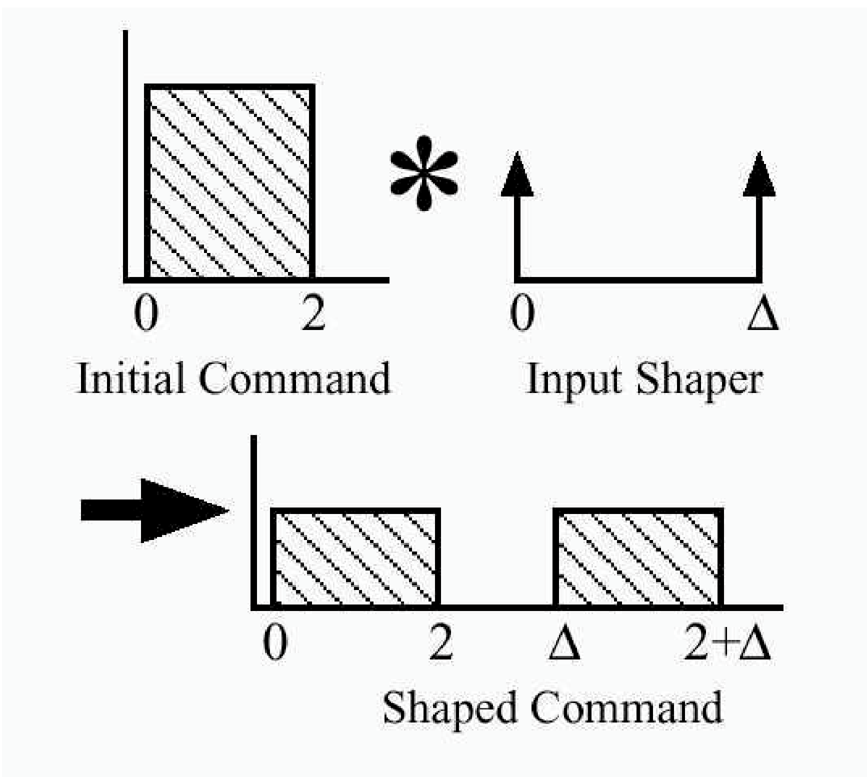
Pulse Inputs

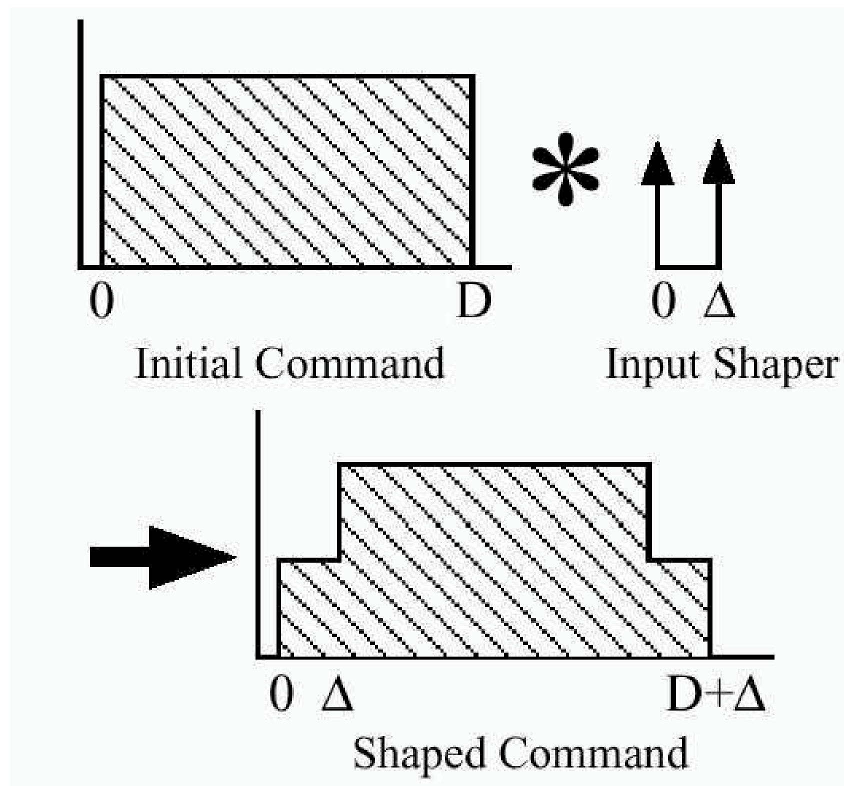
- Compute pulse inputs such that there is no vibration.
- Works for a pulse sequence input
- Can be generalized to *any* input



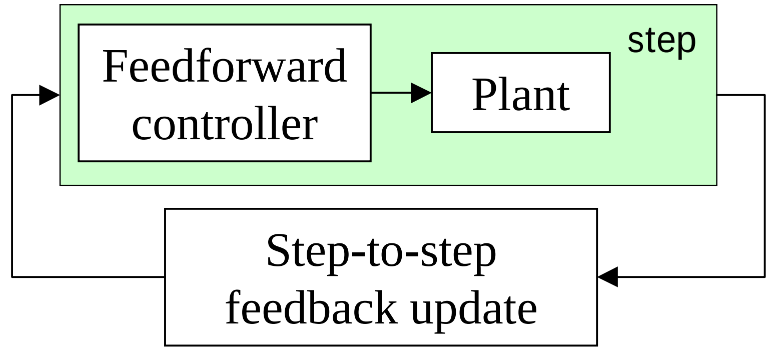
Input Shaping as signal convolution

- Convolution: $f(t) * \left(\sum A_i \delta(t - t_i)\right) = \sum A_i f(t - t_i)$





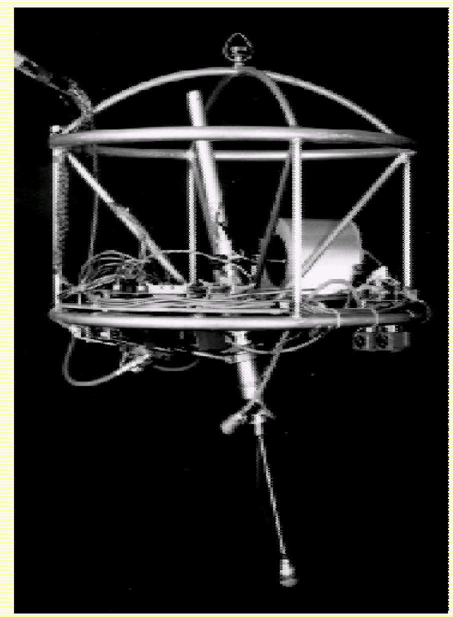
Iterative update of feedforward



- Repetition of control tasks
- Robotics
  - Trajectory control tasks: Iterative Learning Control
  - Locomotion: steps
- Batch process control
  - Run-to-run control in semiconductor manufacturing
  - Iterative Learning Control (*IEEE Control System Magazine*, Dec. 2002)

Example:
One-legged hopping machine (M.Raibert)

Height control:
$y_d = y_d(t-T_n;a)$
$h(n+1)=h(n)+Ga$



Feedforward Implementation

- Constraints and optimality conditions known ahead of time
  - programmed control
- Disturbance feedforward in process control
  - has to be causal, system inversion
- Setpoint change, trajectory tracking
  - smooth trajectory, do not excite the output error
  - in some cases have to use causal ‘system inversion’
  - preview might be available from higher layers of control system, noncausal inverse
- Only final state is important, special case of inputs
  - input shaping - notch filter
  - noncausal parameter optimization

Feedforward Implementation

- Iterative update
  - ILC
  - run-to-run
  - repetitive dynamics

- Replay pre-computed sequences
  - look-up tables, maps
- Not discussed, but used in practice
  - Servomechanism, disturbance model
  - Sinusoidal disturbance tracking - PLL
  - Adaptive feedforward, LMS update

Lecture 6 - SISO Loop Analysis

SISO = Single Input Single Output

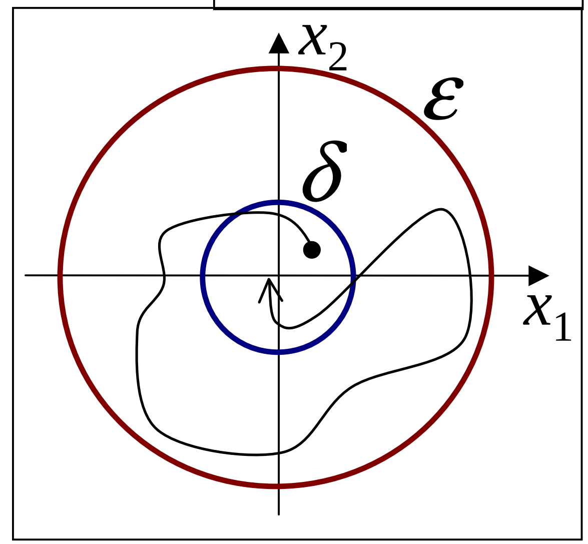
Analysis:

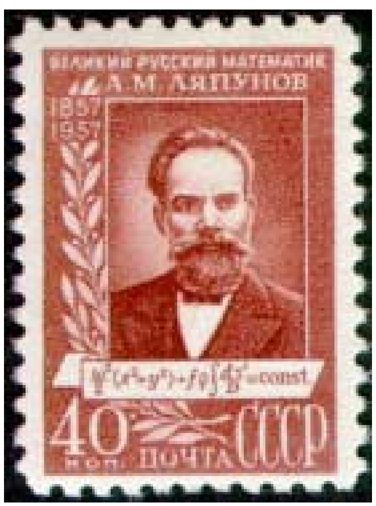
- Stability
- Performance
- Robustness

ODE stability

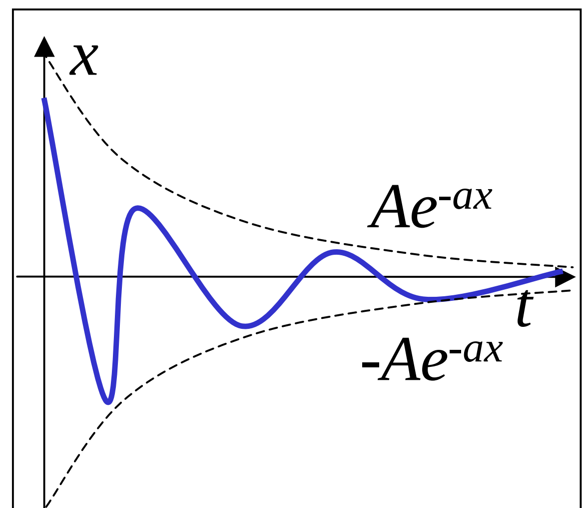
$$\dot{x} = f(x,t)$$

- Lyapunov's stability theory - nonlinear systems
  - stability definition
  - first (direct) method
    - exponential convergence
  - second method: Lyapunov function
    - generalization of energy dissipation





- Lyapunov's exponent
  - dominant exponent of the convergence
  - for a nonlinear system
  - for a linear system defined by the poles

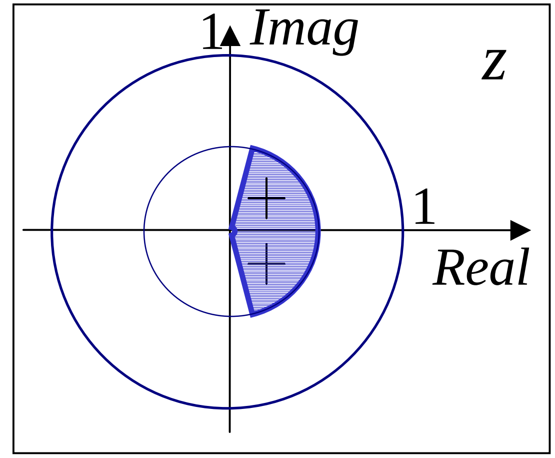


Stability: poles

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

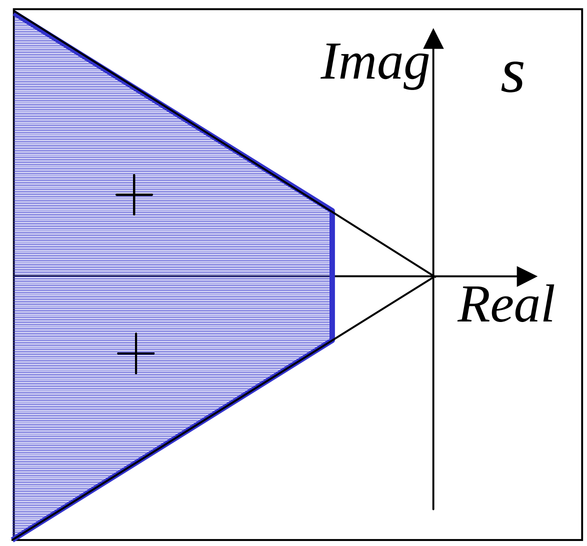
$$y = H(s) \cdot u$$

$$H(s) = C(Is - A)^{-1} B + D$$

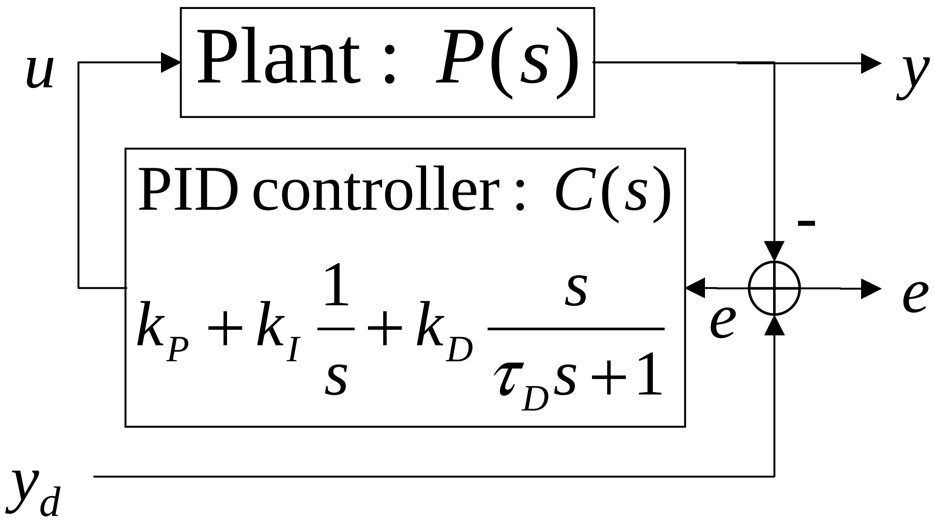


- Characteristic values = transfer function poles
  - l.h.p. for continuous time
  - unit circle for sampled time
- I/O model vs. internal dynamics

$$H(s) = \frac{N(s)}{D(s)} = \frac{g_1}{s - p_1} + \ldots + \frac{g_n}{s - p_n} + g_0$$



Stability: closed loop



$$y = P(s)u$$

$$u = -C(s)e; \qquad e = y - y_d$$

$$e = \underbrace{\left[1 + P(s)C(s)\right]^{-1}}_{S(s)} y_d$$

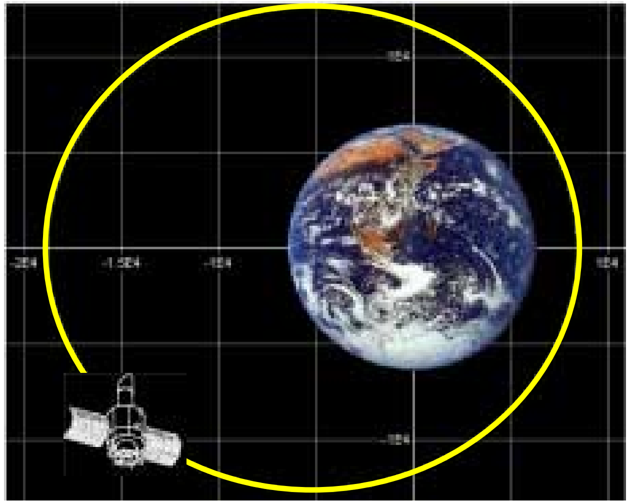
- The transfer function poles are the zeros of

$$1 + P(s)C(s)$$

- Watch for pole-zero cancellations!
- Poles define the closed-loop dynamics (including stability)
- Algebraic problem, easier than state space sim

Stability

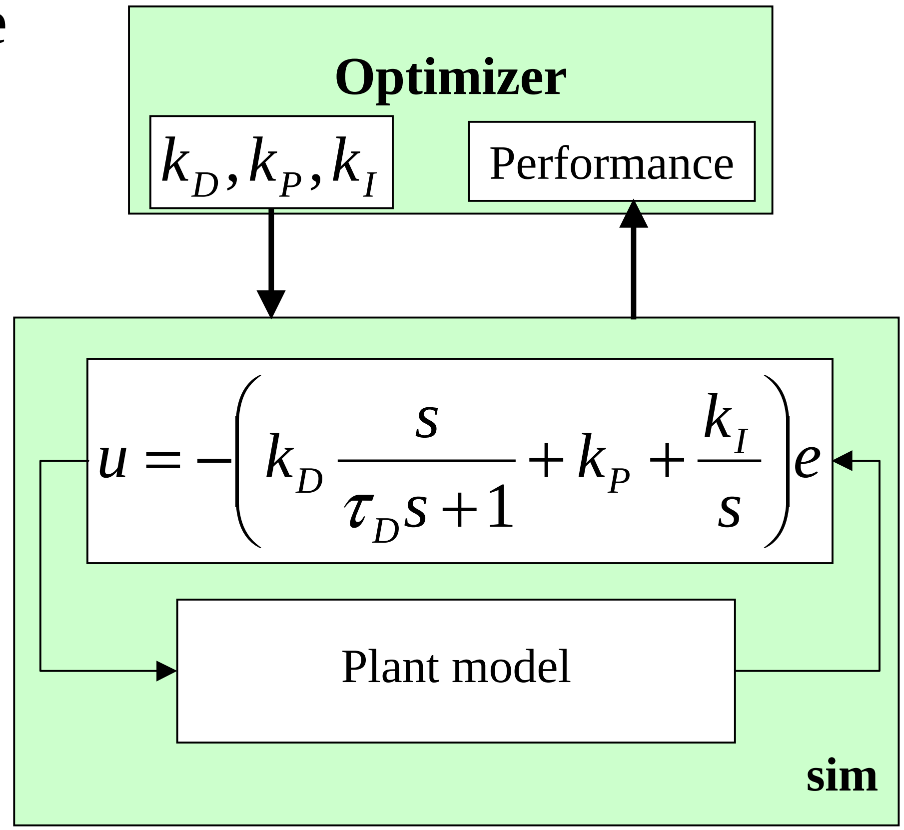
- **For linear system poles describe stability**
- … almost, except the critical stability
- For nonlinear systems
  - linearize around the equilibrium
  - might have to look at the stability theory - Lyapunov
- Orbital stability:
  - trajectory converges to the desired
  - the state does not - the timing is off
    - spacecraft
    - FMS, aircraft arrival



Performance

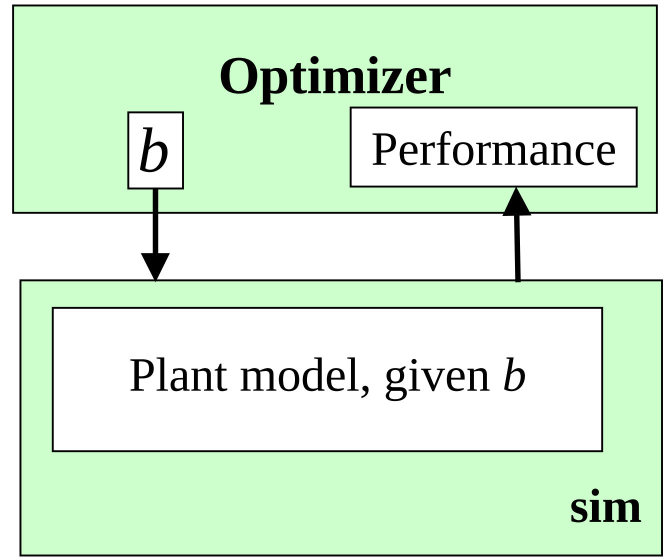
- Need to describe and analyze performance so that we can design systems and tune controllers

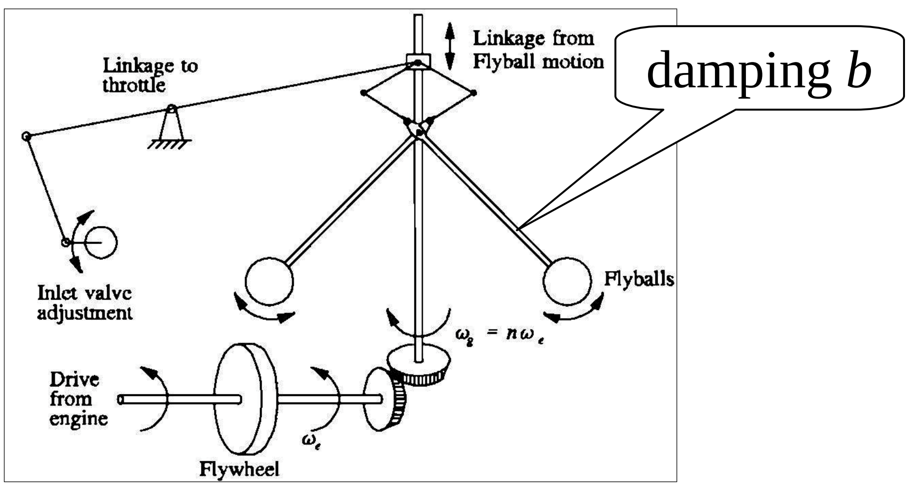
- There are usually many conflicting requirements
- Engineers look for a reasonable trade-off

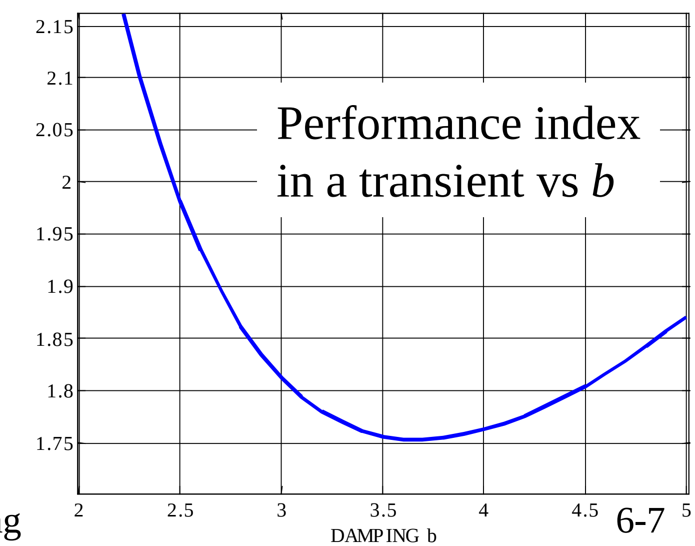


Performance: Example

- Selecting optimal *b* in the Watt's governor - HW Assignment 1



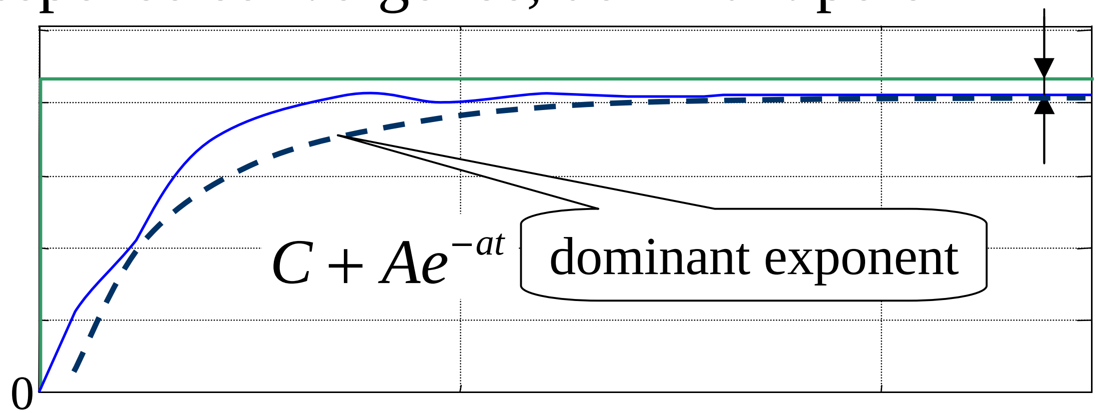




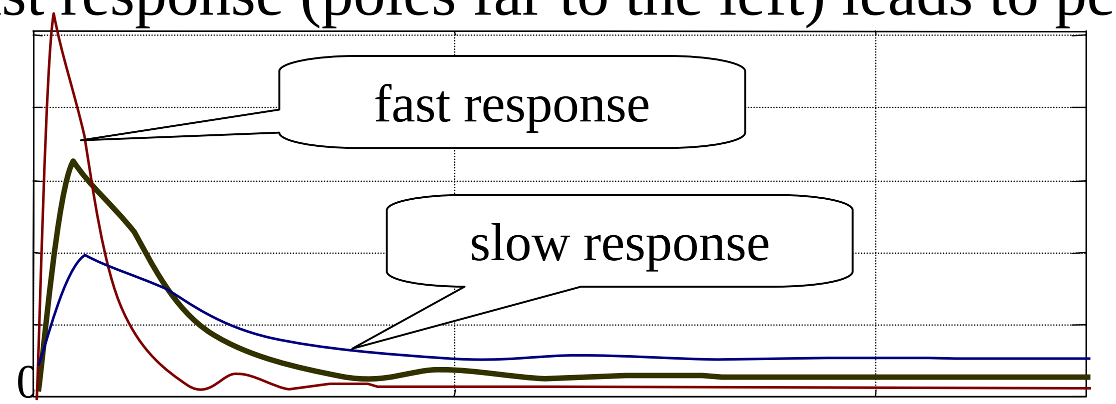
Performance - poles

- Steady state error: study transfer functions at s=0.
- Step/pulse response convergence, dominant pole

$$a = \min\{\mathrm{Re}\, p_j\}_{j=1}^{n}$$

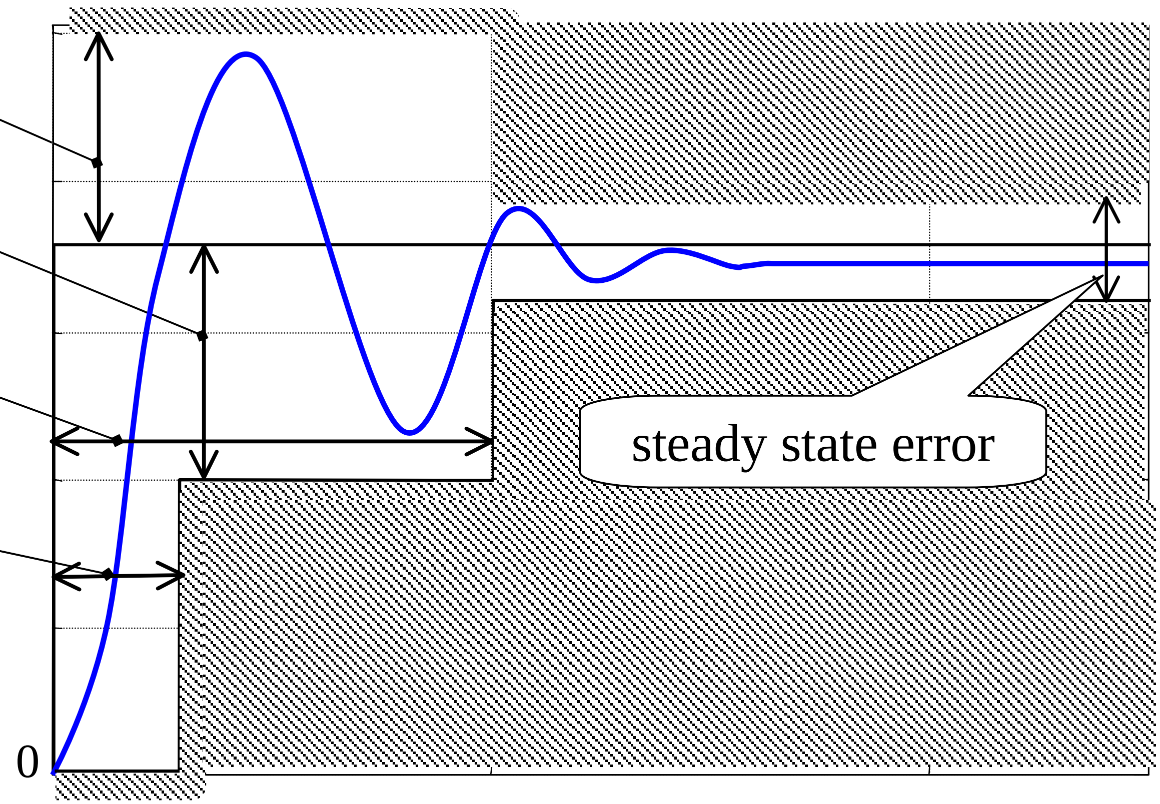


- Caution! Fast response (poles far to the left) leads to peaking



Performance - step response

- Step response shape characterization:
  - overshoot
  - undershoot
  - settling time
  - rise time

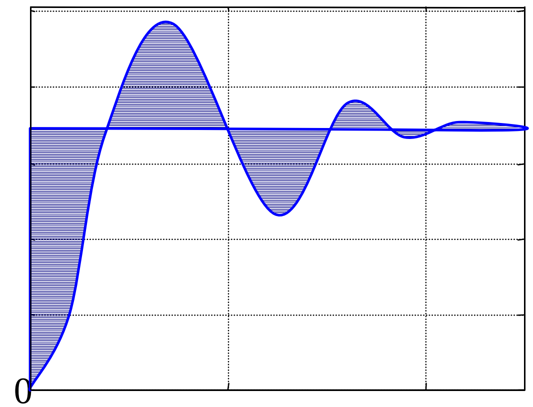


Performance - quadratic index

- Quadratic performance
  - response, in frequency domain

$$J = \int_{t=0}^{\infty} \left| y(t) - y_d(t) \right|^2 dt = \frac{1}{2\pi} \int_{\infty}^{\infty} \left| \tilde{e}(i\omega) \right|^2 d\omega =$$

$$\frac{1}{2\pi} \int \left| S(i\omega) \tilde{y}_d(i\omega) \right|^2 d\omega = \frac{1}{2\pi} \int \left| S(i\omega) \right|^2 \underbrace{\frac{1}{\omega^2}}_{\text{STEP}} d\omega$$

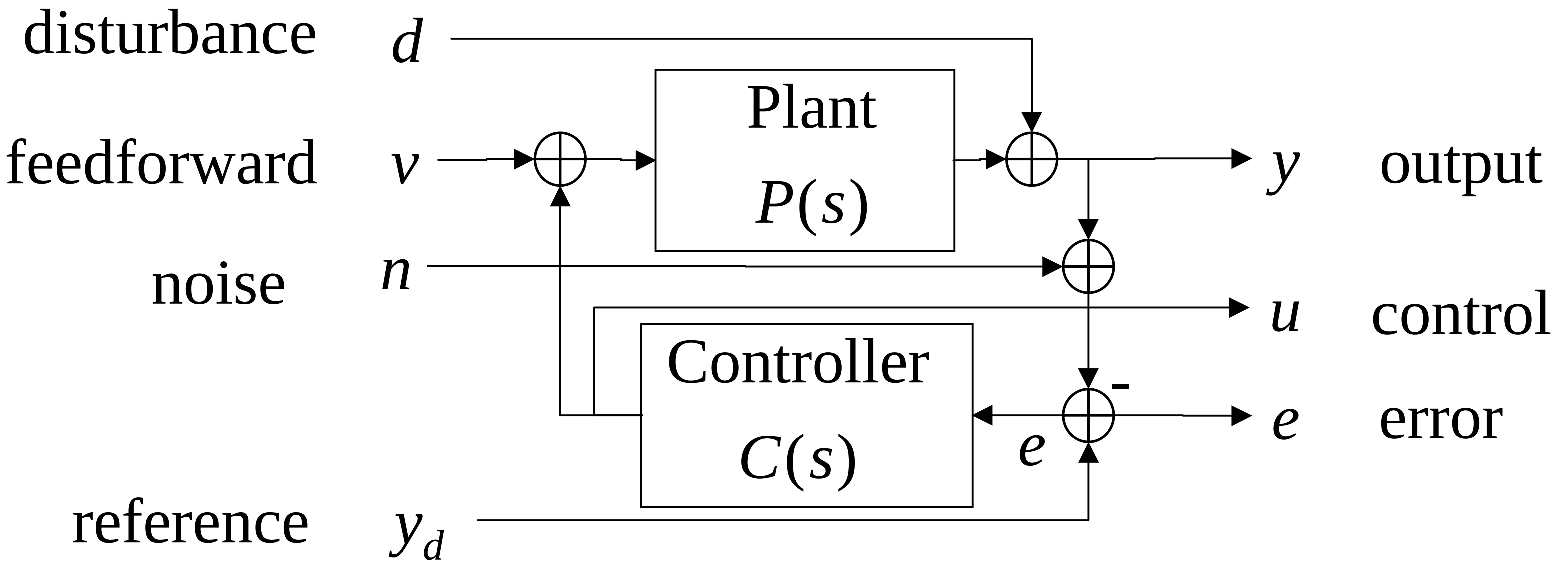


$$S(s) = \left[1 + P(s)C(s) \right]^{-1}$$

- If $y_d(t)$ is a zero mean random process with the spectral power $Q(i\omega)$

$$J = E\left(\int_{t=0}^{\infty} \left| y(t) - y_d(t) \right|^2 dt \right) = \frac{1}{2\pi} \int \left| S(i\omega) \right|^2 Q(i\omega) d\omega$$

Transfer functions in control loop



$$e = S(s)d - S(s)y_d + T(s)n + S_y(s)v$$

$$y = S(s)d + T(s)y_d + T(s)n + S_y(s)v$$

$$u = -S_u(s)d + S_u(s)y_d + S_u(s)n + T(s)v$$

Transfer functions in control loop

$$e = y - y_d + n$$
$$y = P(s)(u+v) + d$$
$$u = -C(s)e$$

$\Rightarrow$

$$e = S(s)d - S(s)y_d + T(s)n + S_y(s)v$$
$$y = S(s)d + T(s)y_d + T(s)n + S_y(s)v$$
$$u = -S_u(s)d + S_u(s)y_d + S_u(s)n + T(s)v$$

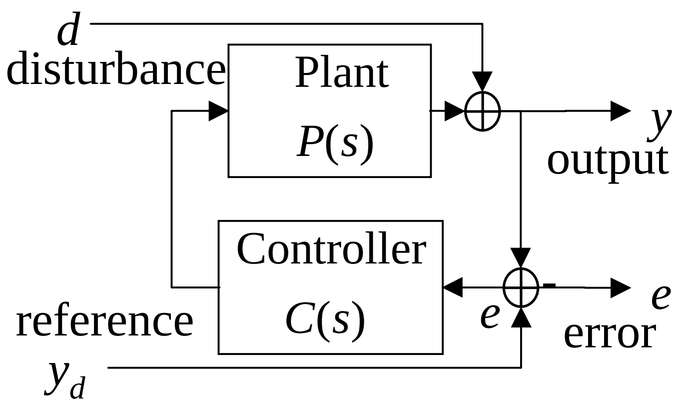
Sensitivity $S(s) = [1 + P(s)C(s)]^{-1}$

Complementary sensitivity $T(s) = [1 + P(s)C(s)]^{-1} P(s)C(s)$

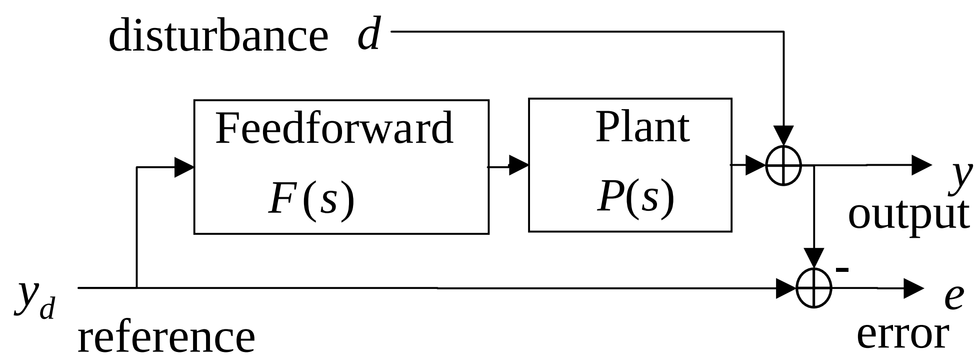
Noise sensitivity $S_u(s) = [1 + P(s)C(s)]^{-1} C(s)$

Load sensitivity $S_y(s) = [1 + P(s)C(s)]^{-1} P(s)$

Sensitivities



$$y = S(s)d + T(s)y_d$$



$$y = d + F(s)P(s)y_d$$

$$S(i\omega) = \frac{1}{1+L(i\omega)}, \qquad L(s) = P(s)C(s)$$

- Feedback sensitivity
  - $|S(i\omega)| \ll 1$ for $|L(i\omega)| \gg 1$
  - $|S(i\omega)| \approx 1$ for $|L(i\omega)| \ll 1$
  - can be bad for $|L(i\omega)| \approx 1$ - ringing, instability

$$S_{FF}(i\omega) = 1$$

- Feedforward sensitivity
  - good for any frequency
  - never unstable

Sensitivity requirements

$$e = S(s)d - S(s)y_d + T(s)n + S_y(s)v$$

$$y = S(s)d + T(s)y_d + T(s)n + S_y(s)v$$

$$u = -S_u(s)d + S_u(s)y_d + S_u(s)n + T(s)v$$

$$S(i\omega) = \frac{1}{1 + P(i\omega)C(i\omega)}$$

$$S_y(i\omega) = \frac{P(i\omega)}{1 + P(i\omega)C(i\omega)}$$

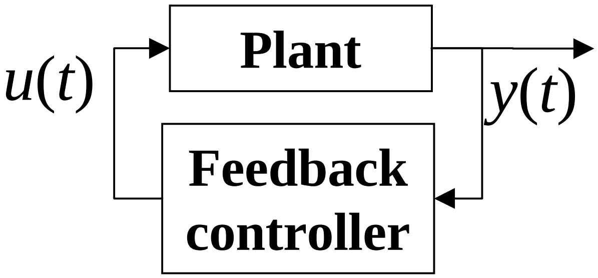
$$S_u(i\omega) = \frac{C(i\omega)}{1 + P(i\omega)C(i\omega)}$$

- Disturbance rejection and reference tracking
  - $|S(i\omega)| << 1$ for the disturbance d ; $|S_y(i\omega)| << 1$ for the input ‘noise’ v
- Limited control effort
  - $|S_u(i\omega)| << 1$ conflicts with disturbance rejection where $|P(i\omega)| < 1$
- Noise rejection
  - $|T(i\omega)| << 1$ for the noise n, conflicts with disturbance rejection

Robustness

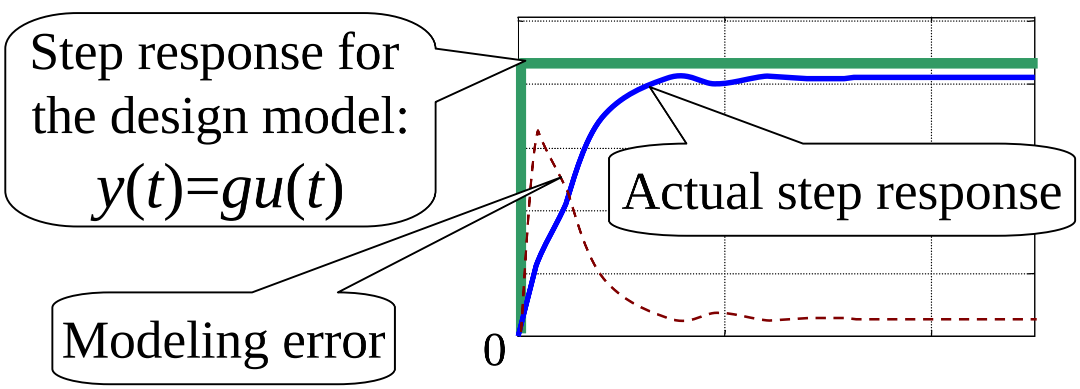
- Ok, we have a controller that works for a *nominal* model.
- Why would it ever would work for *real system*?
  - Will know for sure only when we try - V&V - similar to debugging process in software
- Can check that controller works for a *range* of different models and hope that the real system is covered by this range
  - This is called robustness analysis, robust design
  - Was an implicit part of the classical control design - Nyquist, Bode
  - Multivariable robust control - Honeywell: G.Stein, G.Hartmann, ‘81
  - Doyle, Zames, Glover - robust control theory

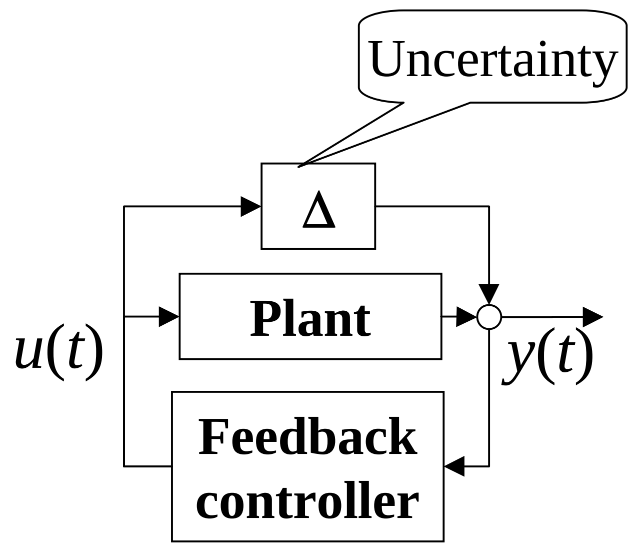
Control loop analysis



$$y = gu$$
$$\dot{u} = -k_I (y - y_d)$$
$$\Rightarrow \quad \dot{y} + gk_I (y - y_d) = 0$$

- Why control might work if the process differs from the model?
- Key factors
  - modeling error (uncertainty) characterization
  - time scale (bandwidth) of the control loop





Robustness - Small gain theorem

- Nonlinear uncertainty!

- Operator gain

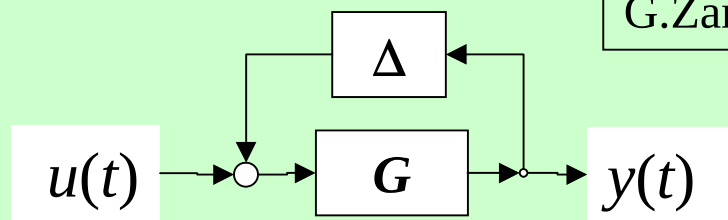
  $\|Gu\| \le \|G\| \cdot \|u\|$

  – *G* **can be a nonlinear operator**

- L_2 norm

$$\|u\|^2 = \int u^2(t)dt = \frac{1}{2\pi}\int \left|\tilde{u}(i\omega)\right|^2 d\omega$$

G.Zames



The loop is guaranteed stable if

$$\|G\| \cdot \|\Delta\| < 1$$

(Open-loop stability assumed)

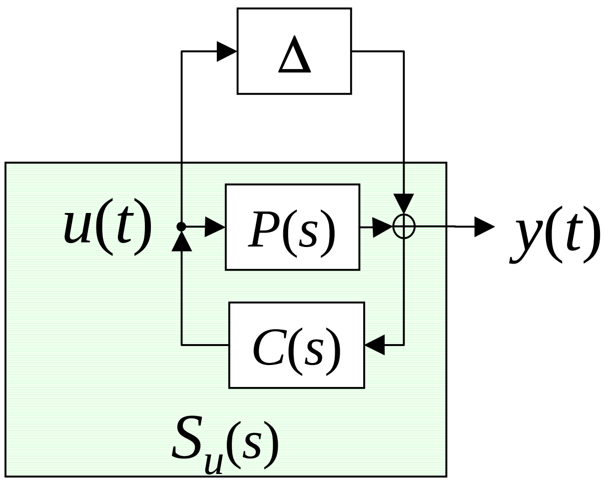
Desoer and Vidyasagar, *Feedback Systems: Input-Output Properties*, 1975

- L_2 gain of a linear operator

$$\|Gu\|^2 = \frac{1}{2\pi}\int \left|G(i\omega)\tilde{u}(i\omega)\right|^2 d\omega \le \underbrace{\sup\left(\left|G(i\omega)\right|^2\right)}_{\|G\|^2} \cdot \underbrace{\frac{1}{2\pi}\int \left|\tilde{u}(i\omega)\right|^2 d\omega}_{\|u\|^2}$$

Robustness

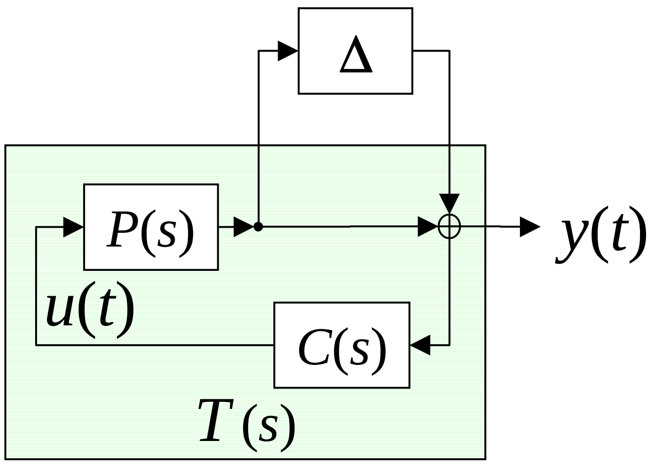
- Additive uncertainty



Condition of robust stability

$$\underbrace{\left|\frac{C(i\omega)}{1+P(i\omega)C(i\omega)}\right|}_{\|S_u\|} \cdot \underbrace{\left|\Delta(i\omega)\right|}_{\|\Delta\|} < 1$$

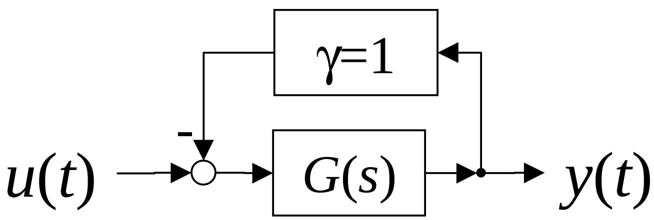
- Multiplicative uncertainty

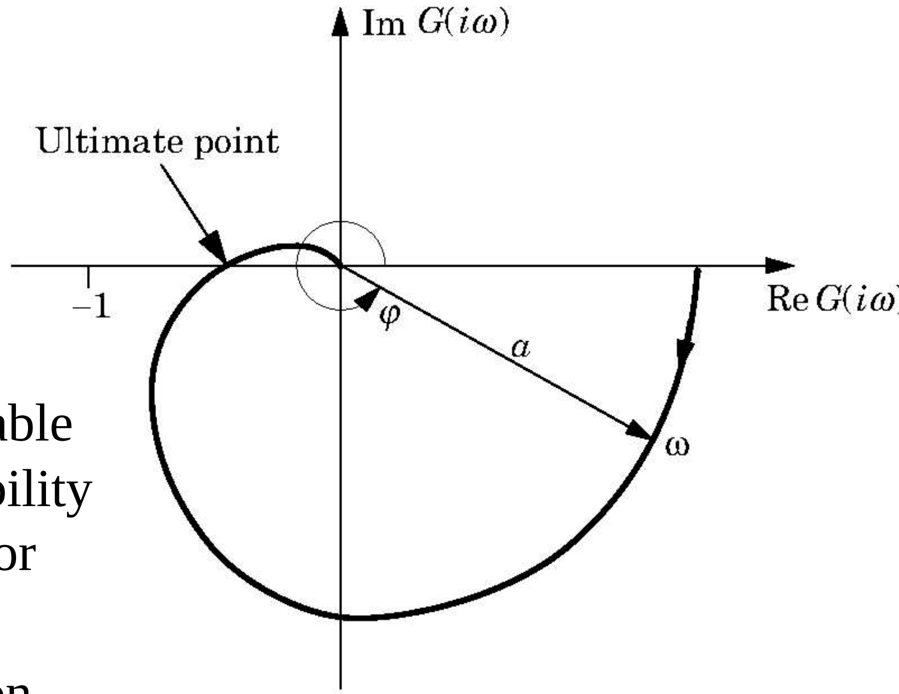


Condition of robust stability

$$\underbrace{\left|\frac{P(i\omega)C(i\omega)}{1+P(i\omega)C(i\omega)}\right|}_{\|T\|} \cdot \underbrace{\left|\Delta(i\omega)\right|}_{\|\Delta\|} < 1$$

Nyquist stability criterion

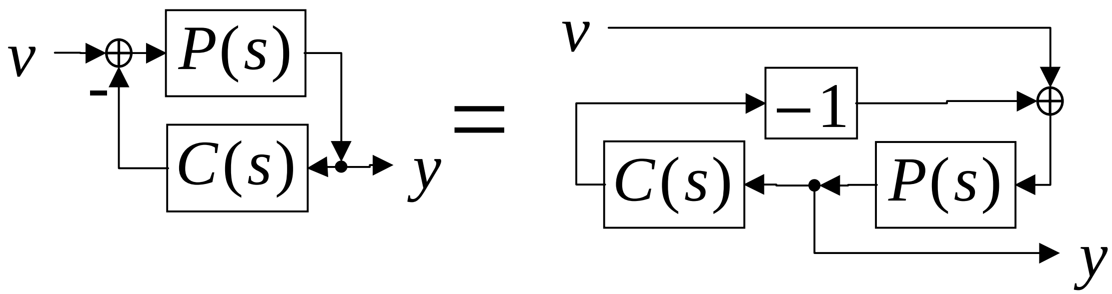




- Homotopy "Proof"
  - $G(s)$ is stable, hence the loop is stable for γ=0. Increase γ to 1. The instability cannot occur unless $\gamma G(iw)+1=0$ for some $0 \le \gamma \le 1$.
  - $|G(i\omega_{180})|<1$ is a *sufficient* condition
- Subtleties: r.h.p. poles and zeros
  - Formulation and real proof using the agrument principle, encirclements of -1
  - stable → unstable → stable as $0 \to \gamma \to 1$

Compare against Small Gain Theorem:

Gain and phase margins

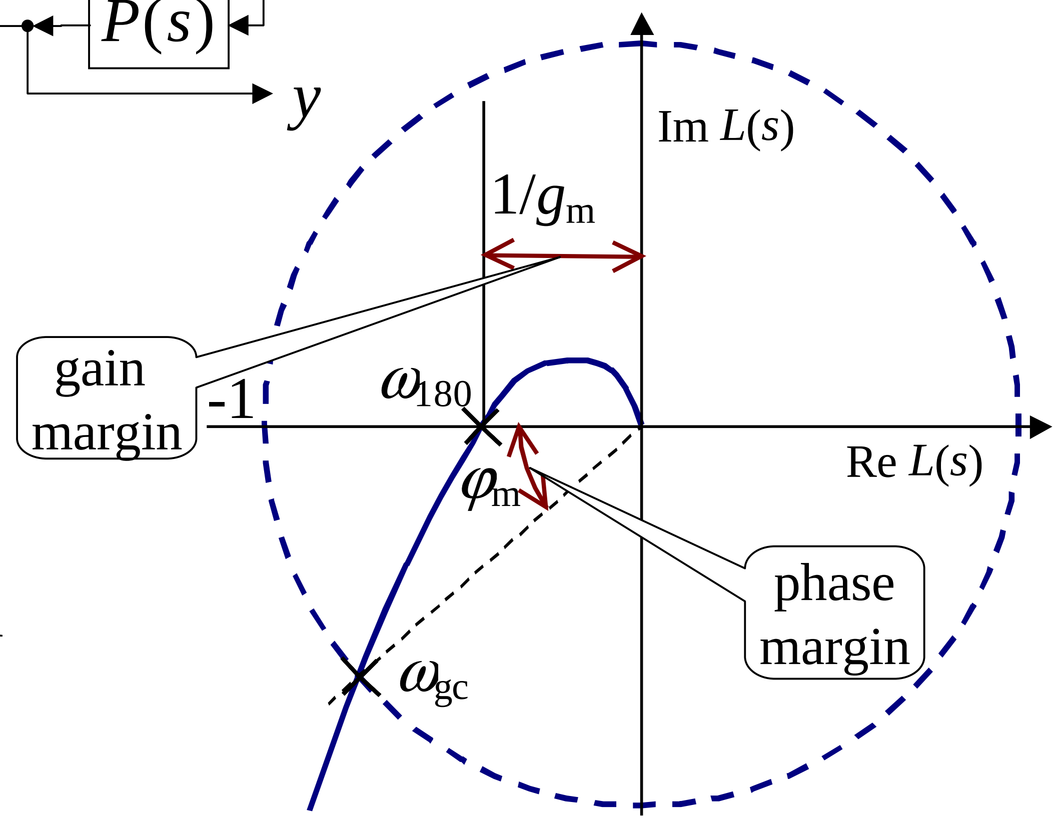


- Loop gain

$$L(s) = P(s)C(s)$$

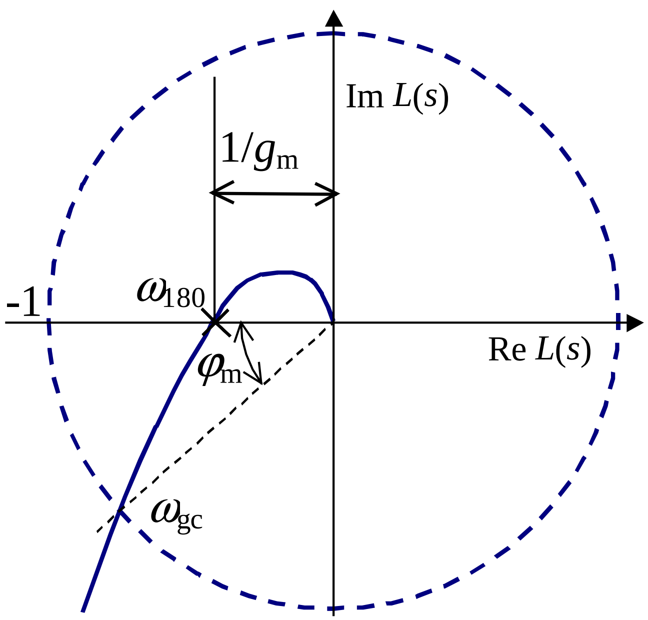
$$S(s) = [1 + L(s)]^{-1}$$

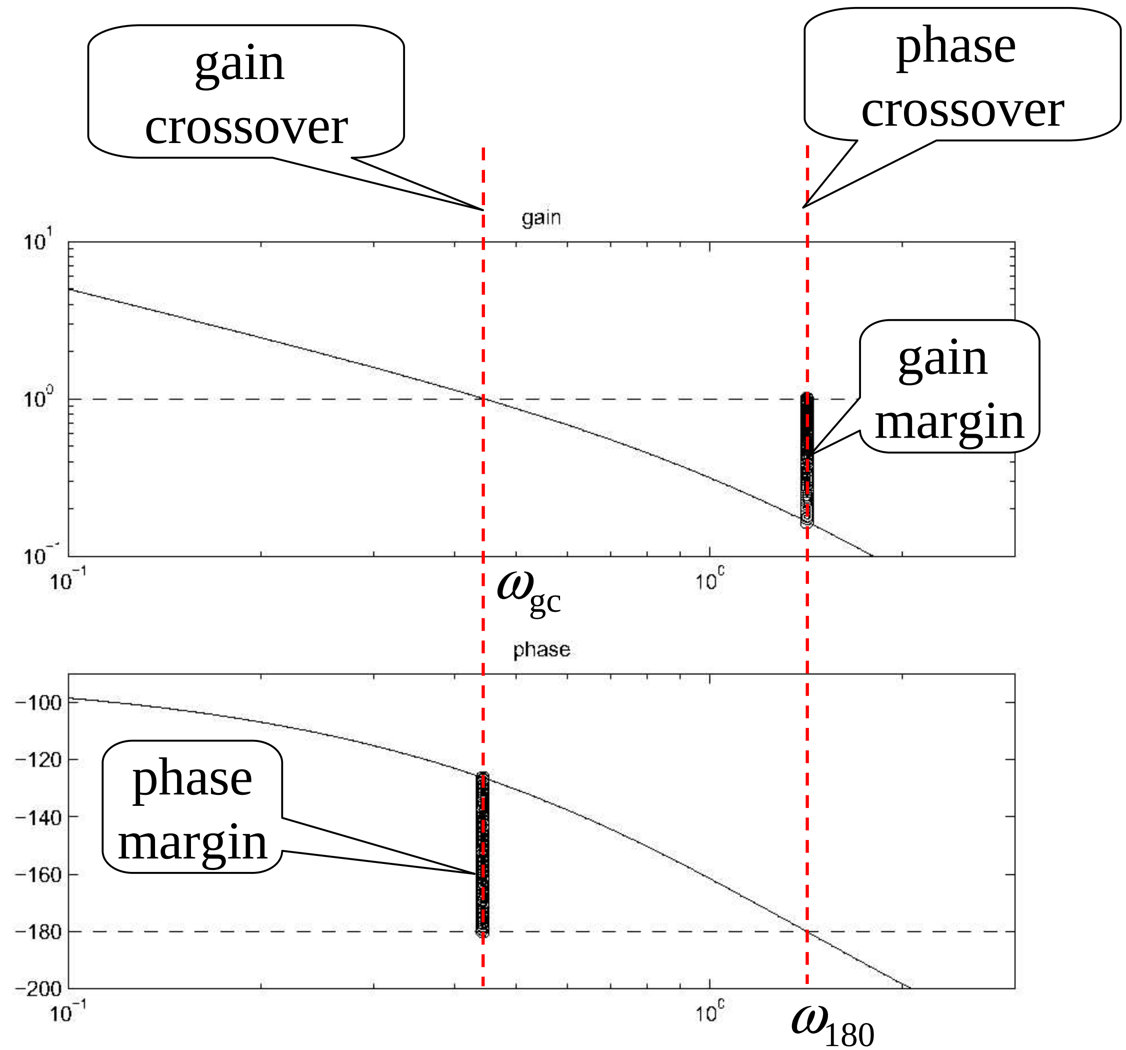
- Nyquist plot for L
  - at high frequency $|L(i\omega)| \le 1$



Gain and phase margins

- Bode plots





Advanced Control

- Observable and controllable system
  - Can put poles anywhere
  - Can drive state anywhere
- Why cannot we just do this?
  - Large control
  - Error peaking
  - Poor robustness, margins
    - Observability and controllability = matrix rank
    - Accuracy of solution is defined by condition number
- Analysis of this lecture is valid for *any* LTI control, including advanced

Lecture 7 - SISO Loop Design

- Design approaches, given specs
- Loopshaping: in-band and out-of-band specs
- Design example
- Fundamental design limitations for the loop
  - Frequency domain limitations
  - Structural design limitations
  - Engineering design limitations

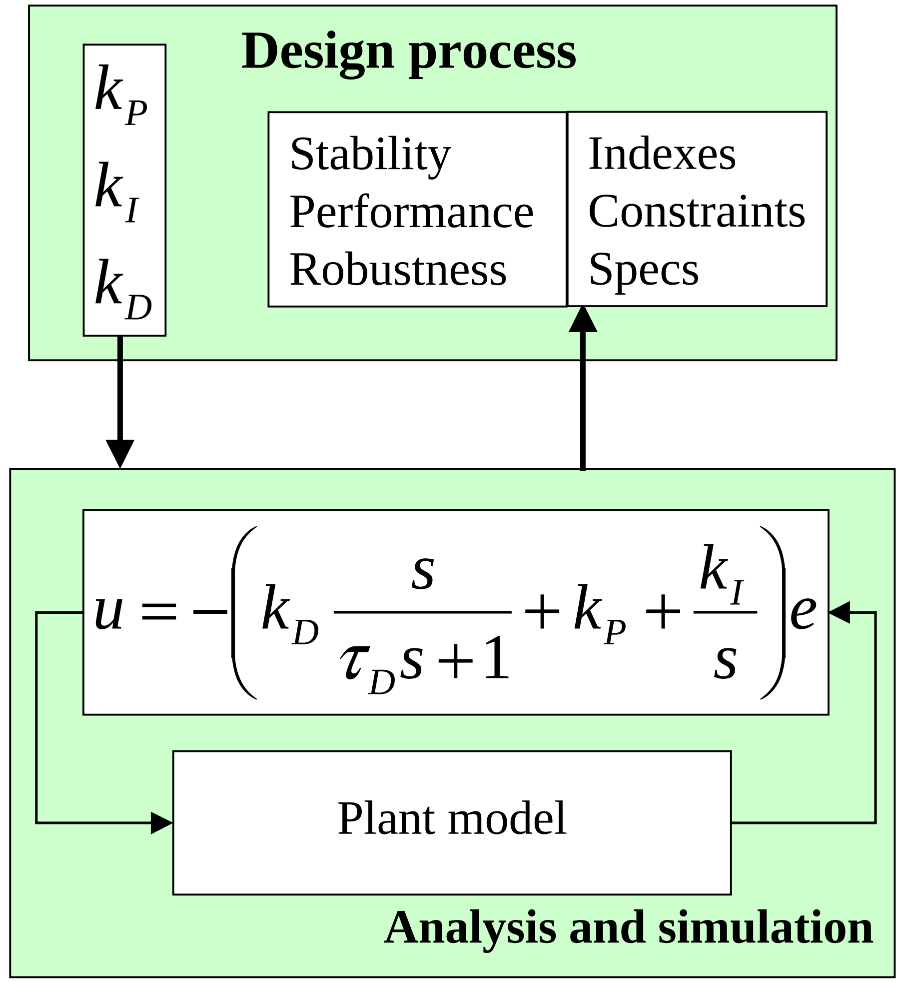
Modern control design

- Observable and controllable system
  - Can put poles anywhere
  - Can drive state anywhere
  - Can design ‘optimal control’

- Issues
  - Large control
  - Error peaking in the transient
  - Noise amplification
  - Poor robustness, margins
  - Engineering trade off vs. a single optimality index

Feedback controller design

- Conflicting requirements
- Engineers look for a reasonable trade-off
  - Educated guess, trial and error controller parameter choice
  - Optimization, if the performance is really important
    - optimality parameters are used as tuning handles



Loopshape requirements

$$L(i\omega) = P(i\omega)C(i\omega)$$

Performance

$$S(i\omega) = [1 + L(i\omega)]^{-1}$$

- Disturbance rejection and reference tracking
  - $|S(i\omega)| << 1$ for the disturbance d; $|P(i\omega)S(i\omega)| << 1$ for the load v
  - **satisfied for $|L(i\omega)| >> 1$**
- Noise rejection
  - $|T(i\omega)| = |S(i\omega)L(i\omega)| < 1$ is Ok unless $|1 + L(i\omega)|$ is small
- Limited control effort
  - $|C(i\omega)\, S(i\omega)| < 1$
  - works out with large $|C(i\omega)|$ for low frequency, where $|P(i\omega)| > 1$

Loopshape requirements

Robustness

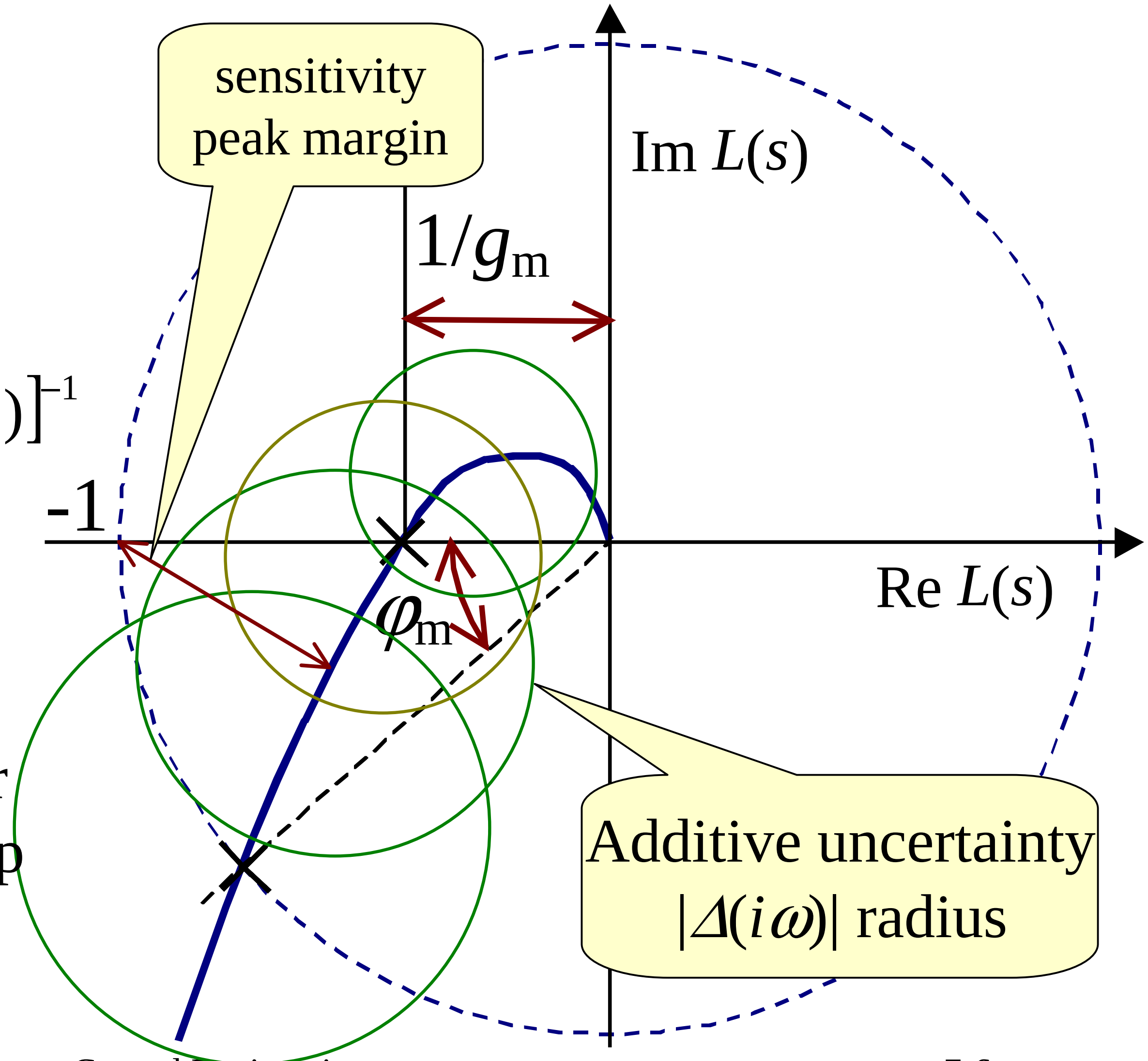
- Multiplicative uncertainty
  - $|T(i\omega)| < 1/\delta(\omega)$, where $\delta(\omega)$ is the uncertainty magnitude
  - at high frequencies, relative uncertainty can be large, hence, $|T(i\omega)|$ must be kept small
  - **must have $|L(i\omega)| << 1$ for high frequency, where $\delta(\omega)$ is large**
- Additive uncertainty
  - $|C(i\omega)\, S(i\omega)| < 1/\delta(\omega)$, where $\delta(\omega)$ is the uncertainty magnitude
- Gain margin of 10-12db and phase margin of 45-50 deg
  - this corresponds to the relative uncertainty of the plant transfer function in the 60-80% range around the crossover

Gain and phase margins

- Are less informative than the noise sensitivity

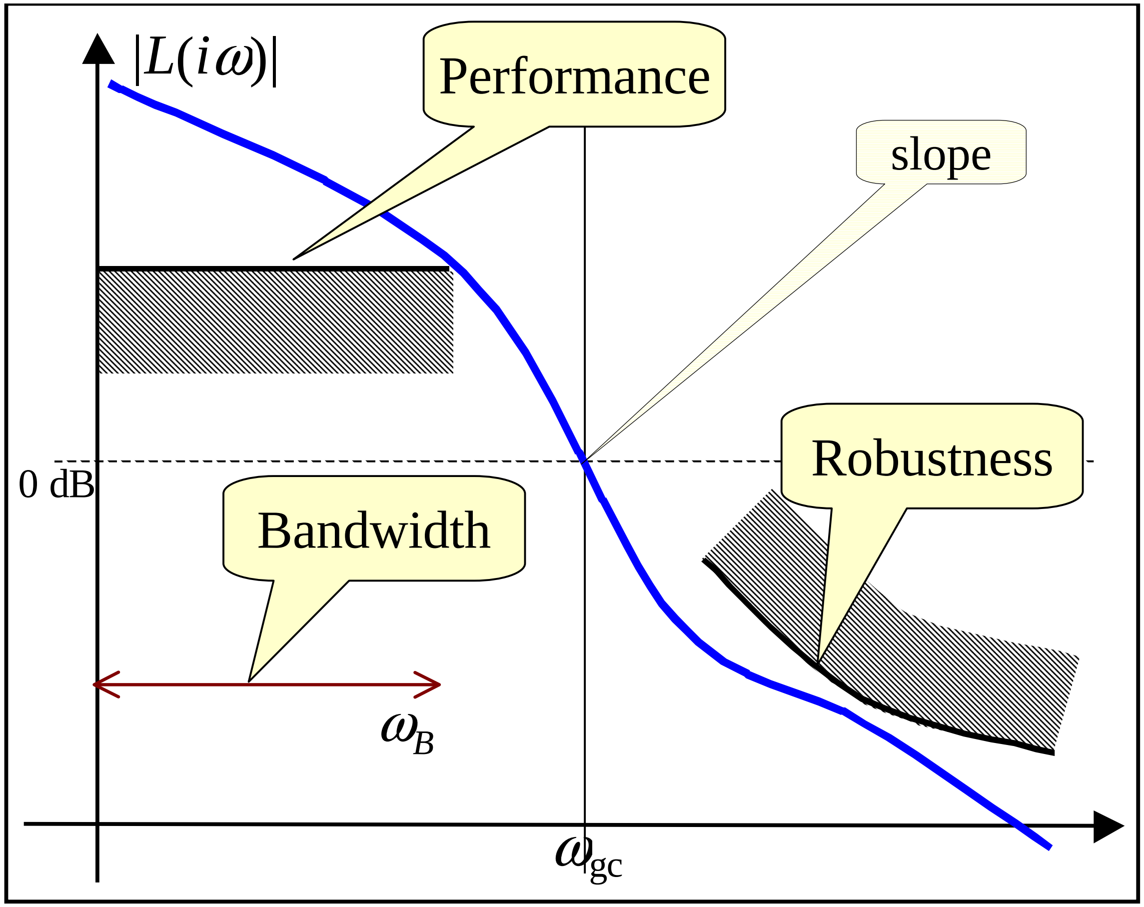
$$S_u(s) = C(s)\left[1 + P(s)C(s)\right]^{-1}$$

- Can use uncertainty characterization and the sensitivity instead
- Margins are useful for deciding upon the loop shape modifications



Loop Shape Requirements

- Low frequency:
  - high gain L
    = small S
- High frequency:
  - small gain L
    = small $T \cdot$ large δ
- Bandwidth
  - performance can be only achieved in a limited frequency band: $\omega \leq \omega_B$
  - ω_B is the bandwidth



Fundamental tradeoff: performance vs. robustness

Loopshaping design

- Loop design
  - Use P,I, and D feedback to shape the loop gain
- Loop modification and bandwidth
  - Low-pass filter - get rid of high-frequency stuff - robustness
  - Notch filter - get rid of oscillatory stuff - robustness
  - Lead-lag to improve phase around the crossover - bandwidth
    - P+D in the PID together have a lead-lag effect
- Need to maintain stability while shaping the magnitude of the loop gain
- Formal design tools H_2, H_∞, LMI, H_∞ loopshaping
  - cannot go past the fundamental limitations

Example - disk drive servo

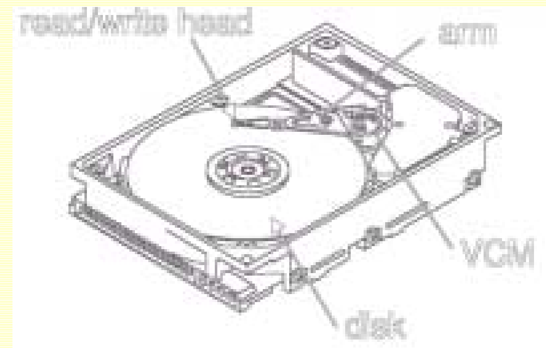
- The problem from HW Assignment 2
  - data in `diskPID.m, diskdata.mat`
- Design model: $\Delta P(s)$ is an uncertainty

$$P(s) = \frac{g_0}{s^2} + \Delta P(s)$$

- Analysis model: description for $\Delta P(s)$
- Design approach: PID control based on the simplified model

$$C(s) = k_P + \frac{k_I}{s} + k_D \frac{s}{\tau_D s + 1}$$

Disk servo control



$$J\ddot{\varphi} = T_{VCM} + T_{DISTURBANCE}$$

Voice Coil Motor

Disk drive servo controller

- Start from designing a PD controller
  - poles, characteristic equation

$$1 + C(s)P(s) = 0 \Rightarrow \left(k_P + sk_D\right) \cdot \frac{g_0}{s^2} + 1 = 0$$

$$s^2 + sg_0k_D + g_0k_P = 0$$

- Critically damped system

$$k_D = 2w_0 / g_0; \qquad k_P = w_0^{\,2} / g_0$$

where frequency w_0 is the closed-loop bandwidth

- In the derivative term make dynamics faster than w_0. Select $\tau_D = 0.25 / w_0$

$$k_D \frac{s}{\tau_D s + 1}$$

Disk drive servo

- Step up from PD to PID control

$$1+\left(k_P + sk_D + \frac{1}{s}k_I\right)\cdot\frac{g_0}{s^2} = 0$$

$$s^3 + s^2 g_0 k_D + s g_0 k_P + g_0 k_I = 0$$

- Keep the system close to the critically damped, add integrator term to correct the steady state error, keep the scaling

$$k_P = w_0^2 / g_0; \qquad k_D = a w_0 / g_0; \qquad k_I = b w_0^3 / g_0 \qquad \tau_D = c / w_0$$

where *a*, *b*, and *c* are the tuning parameters

- Initial guess: w_0 =2000; *a*=2; *b*=0.1; *c* =0.25
- Tune *a*, *b*, *c* and w_0 by watching performance and robustness

Disk drive - controller tuning

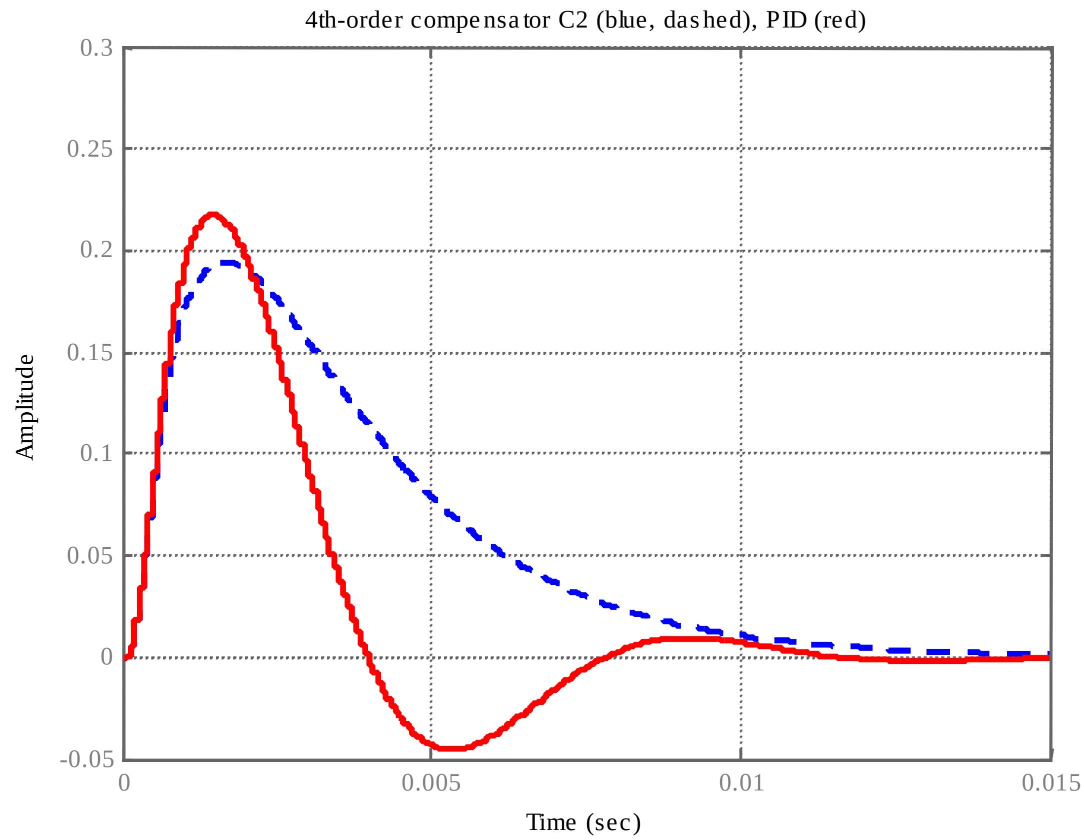
- Tune a, b, w_0 , and τ_D by trial and error
- Find a trade off taking into the account
  - Closed loop step response
  - Loop gain - performance
  - Robustness - sensitivity
  - Gain and phase margins
- Try to match the characteristics of C2 controller (demo)

- The final tuned values:

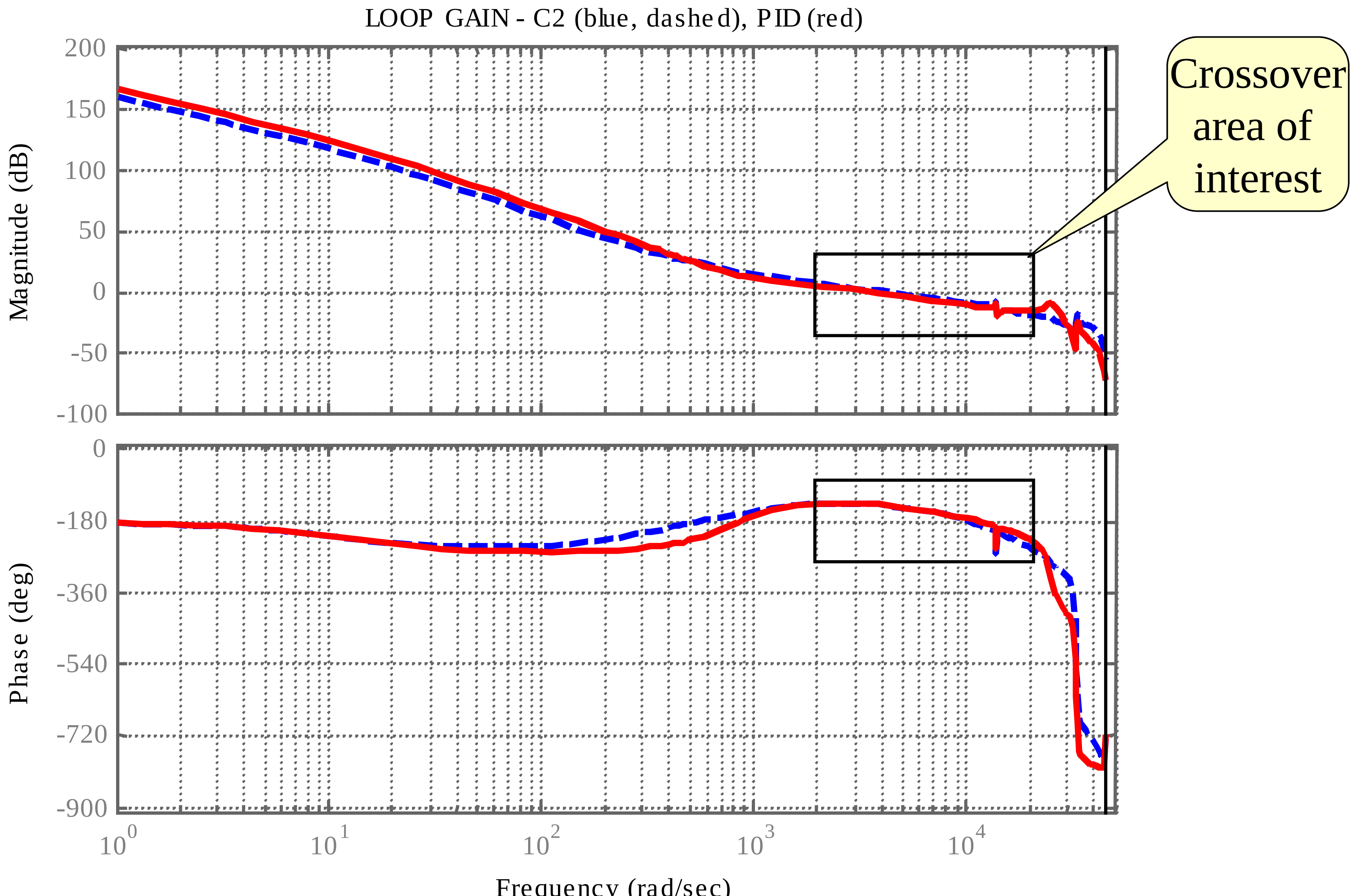
$$w_0 = 1700;\ a = 1.5;\ b = 0.5;\ c = 0.2$$

Disk servo - controller comparison

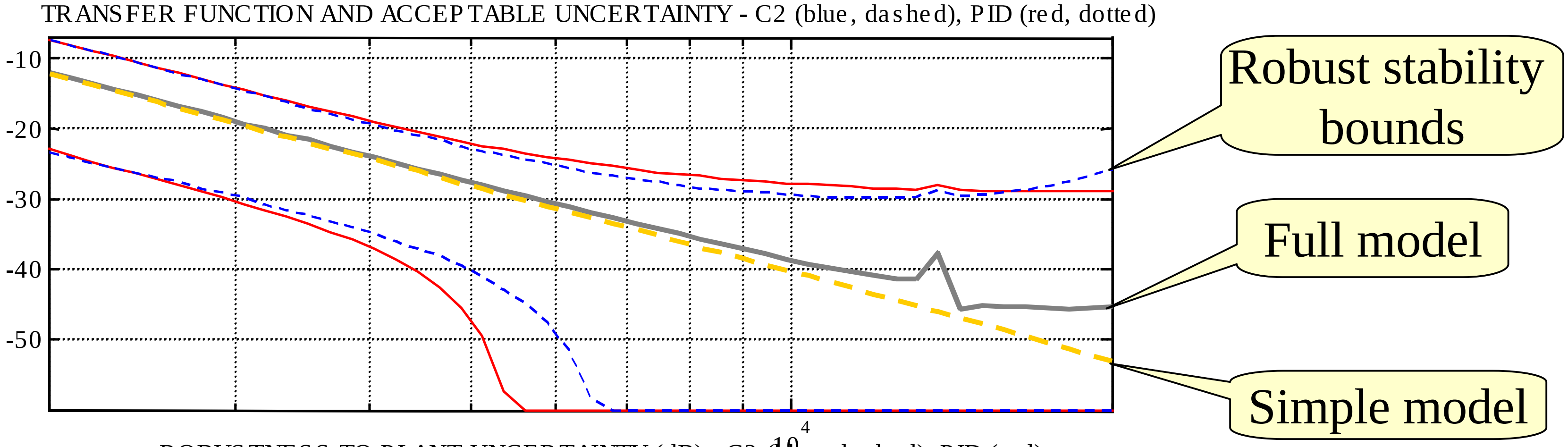
- PID is compared against a reference design
- Reference design: 4-th order controller: lead-lag + notch filter
  - Matlab `diskdemo`
  - Data in `diskPID.m`, `diskdata.mat`

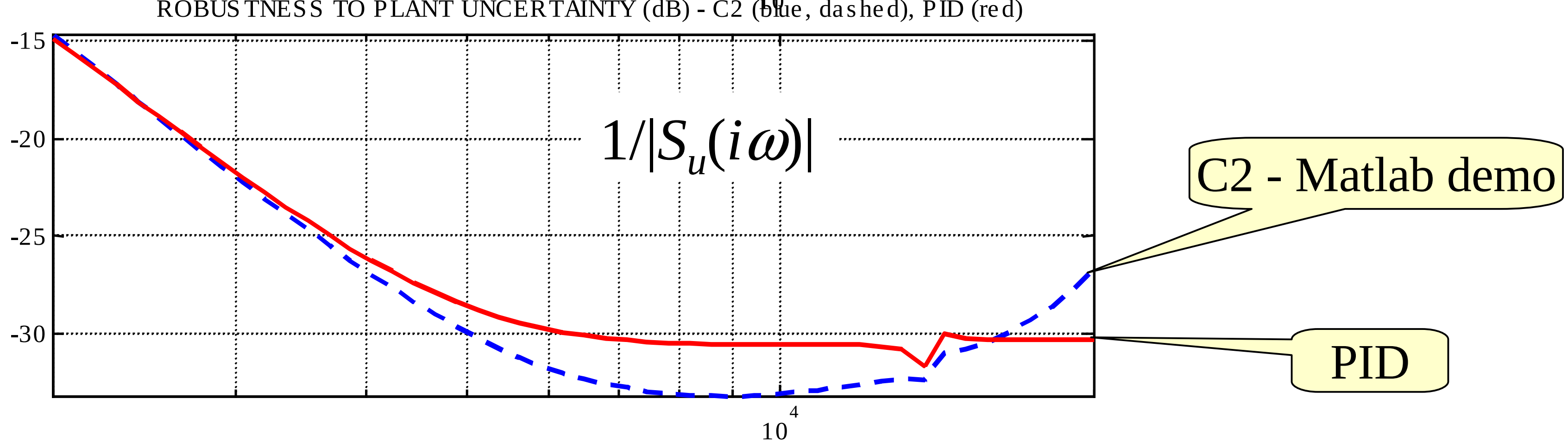


Loop shape, margins



Disk drive servo - robustness






```
[m2,ph2]=bode(feedback(C2,Gd),w))
[mP,phP]=bode(feedback(PIDd,Gd),w))    plot(w,1./mP,w,1/m2)
```

Fundamental design limitations

- If we do not have a reference design - how do we know if we are doing well. May be there is a much better controller?

- Cannot get around the fundamental design limitations
  - frequency domain limitations on the loop shape
  - system structure limitations
  - engineering design limitations

Frequency domain limitation

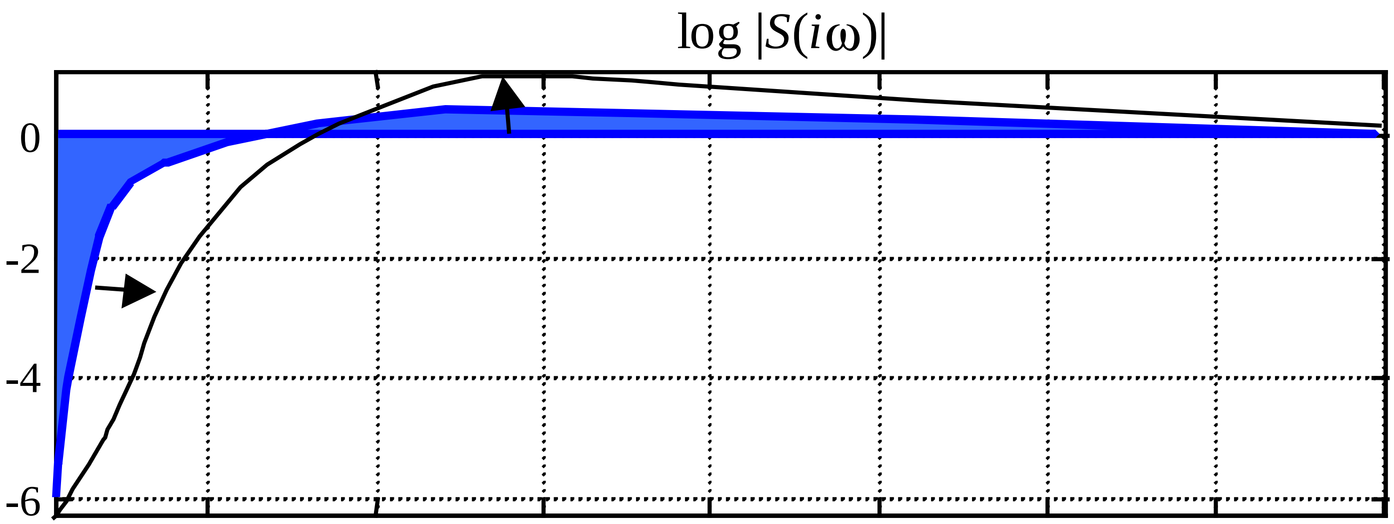
$S(i\omega) + T(i\omega) = 1$

Performance: $|S(i\omega)| << 1$

Robustness: $|T(i\omega)| << 1$

- Bode's integral constraint - waterbed effect

$$\int_0^\infty \log|S(i\omega)|\,d\omega = 0$$ (for most real-life stable system, or worse for the rest)



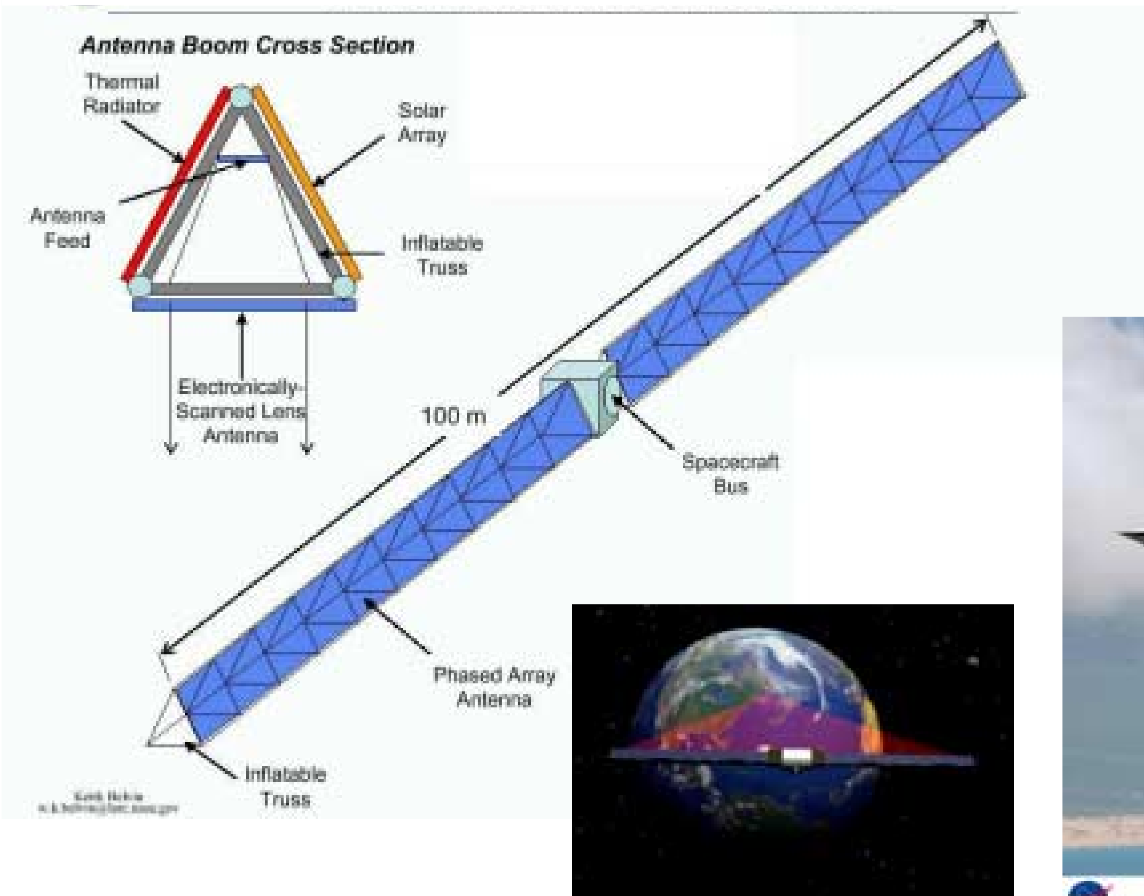
Structural design limitations

- Delays and non-minimum phase (r.h.s. zeros)
  - cannot make the response faster than delay, set bandwidth smaller
- Unstable dynamics
  - makes Bode's integral constraint worse
  - re-design system to make it stable or use advanced control design
- Flexible dynamics
  - cannot go faster than the oscillation frequency
  - practical approach:
    - filter out and use low-bandwidth control (wait till it settles)
    - use input shaping feedforward

Unstable dynamics

- Very advanced applications
  - need advanced feedback control design

Flexible dynamics



- Very advanced applications
  - really need control of 1-3 flexible modes

NASA Dryden Flight Research Center Photo Collection
http://www.dfrc.nasa.gov/gallery/photo/index.html
NASA Photo: ED02-0161-1 Date: June 24, 2002 Photo by: Nick Galante

Pathfinder-Plus flight in Hawaii

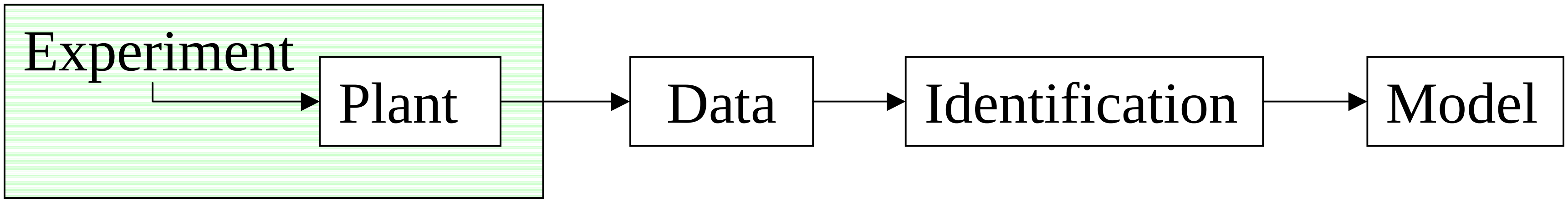
Engineering design limitations

- Sensors
  - noise - have to reduce $|T(i\omega)|$ - reduced performance
  - quantization - same effect as noise
  - bandwidth (estimators) - cannot make the loop faster
- Actuators
  - range/saturation - limit the load sensitivity $|C(i\omega)\ S(i\omega)|$
  - actuator bandwidth - cannot make the loop faster
  - actuation increment - sticktion, quantization - effect of a load variation
  - other control handles
- Modeling errors
  - have to increase robustness, decrease performance
- Computing, sampling time
  - Nyquist sampling frequency limits the bandwidth

Lecture 8 - Model Identification

- What is system identification?
- Direct pulse response identification
- Linear regression
- Regularization
- Parametric model ID, nonlinear LS

What is System Identification?



- White-box identification
  - estimate parameters of a physical model from data
  - Example: aircraft flight model
- Gray-box identification
  - given generic model structure estimate parameters from data
  - Example: neural network model of an engine
- Black-box identification
  - determine model structure and estimate parameters from data
  - Example: security pricing models for stock market

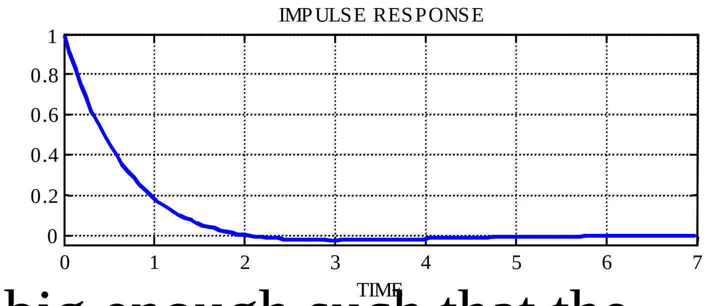
Rarely used in real-life control

Industrial Use of System ID

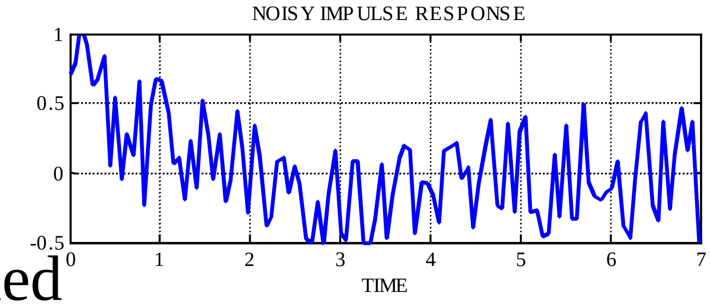
- Process control - most developed ID approaches
  - all plants and processes are different
  - need to do identification, cannot spend too much time on each
  - industrial identification tools
- Aerospace
  - white-box identification, specially designed programs of tests
- Automotive
  - white-box, significant effort on model development and calibration
- Disk drives
  - used to do thorough identification, shorter cycle time
- Embedded systems
  - simplified models, short cycle time

Impulse response identification

- Simplest approach: apply control impulse and collect the data



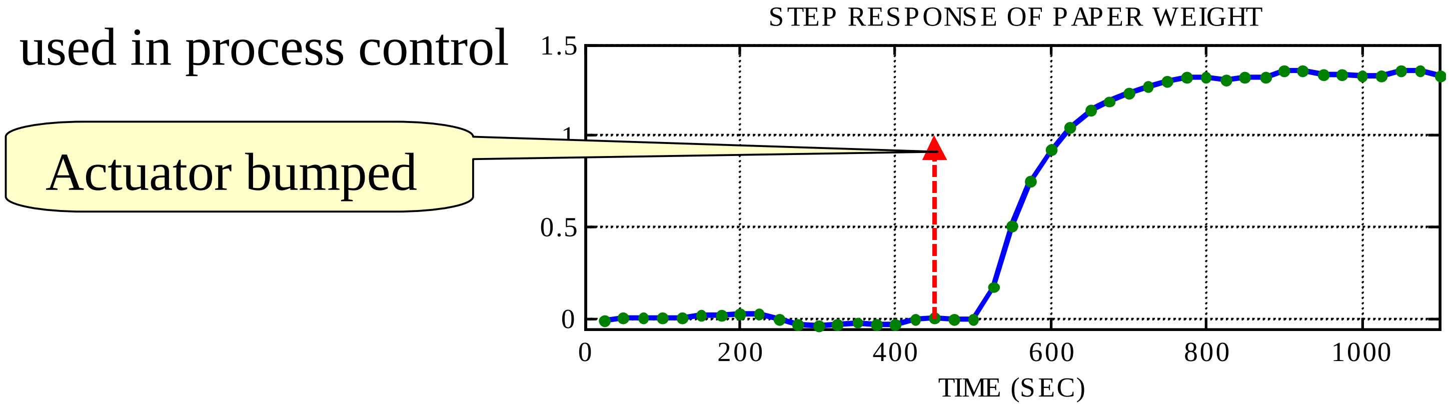
- Difficult to apply a short impulse big enough such that the response is much larger than the noise



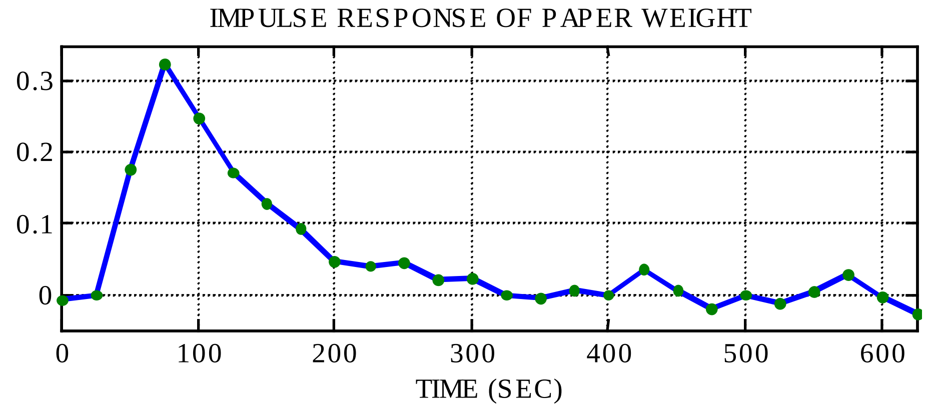
- Can be used for building simplified control design models from complex sims

Step response identification

- Step (bump) control input and collect the data
  - used in process control



- Impulse estimate still noisy: impulse(t) = step(t)-step(t-1)



Noise reduction

Noise can be reduced by statistical averaging:

- Collect data for mutiple steps and do more averaging to estimate the step/pulse response
- Use a parametric model of the system and estimate a few model parameters describing the response: dead time, rise time, gain
- Do both in a sequence
  - done in real process control ID packages
- Pre-filter data

Linear regression

- Mathematical aside
  - linear regression is one of the main System ID tools

Data — Regression weights — Regressor — Error

$$y(t) = \sum_{j=1}^{N} \theta_j \varphi_j(t) + e(t)$$

$$y = \Phi \theta + e$$

$$y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \Phi = \begin{bmatrix} \varphi_1(1) & \dots & \varphi_K(1) \\ \vdots & \ddots & \vdots \\ \varphi_1(N) & \dots & \varphi_K(N) \end{bmatrix}, \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_K \end{bmatrix}, e = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix}$$

Linear regression

- Makes sense only when matrix Φ is tall, $N > K$, more data available than the number of unknown parameters.
  - Statistical averaging
- Least square solution: $\|e\|^2 \rightarrow \min$
  - Matlab `pinv` or left matrix division `\`
- Correlation interpretation:

$$y = \Phi\theta + e$$

$$\hat{\theta} = \left(\Phi^T \Phi\right)^{-1} \Phi^T y$$

$$\hat{\theta} = R^{-1} c$$

$$R = \frac{1}{N}\begin{bmatrix} \sum_{t=1}^{N} \varphi_1^2(t) & \dots & \sum_{t=1}^{N} \varphi_K(t)\varphi_1(t) \\ \vdots & \ddots & \vdots \\ \sum_{t=1}^{N} \varphi_1(t)\varphi_K(t) & \dots & \sum_{t=1}^{N} \varphi_K^2(t) \end{bmatrix}, \qquad c = \frac{1}{N}\begin{bmatrix} \sum_{t=1}^{N} \varphi_1(t) y(t) \\ \vdots \\ \sum_{t=1}^{N} \varphi_K(t) y(t) \end{bmatrix}$$

Example: linear first-order model

$$y(t) = ay(t-1) + gu(t-1) + e(t)$$

- Linear regression representation

$$\varphi_1(t) = y(t-1) \qquad \theta = \begin{bmatrix} a \\ g \end{bmatrix} \qquad \hat{\theta} = \left(\Phi^T \Phi\right)^{-1} \Phi^T y$$
$$\varphi_2(t) = u(t-1)$$

- This approach is considered in most of the technical literature on identification

Lennart Ljung, *System Identification: Theory for the User*, 2nd Ed, 1999

- Matlab Identification Toolbox
  - Industrial use in aerospace mostly
  - Not really used much in industrial process control
- Main issue:
  - small error in *a* might mean large change in response

Regularization

- Linear regression, where $\Phi^T\Phi$ is ill-conditioned
- Instead of $\|e\|^2 \to \min$ solve a regularized problem

$$\|e\|^2 + r\|\theta\|^2 \to \min$$

$$\boxed{y = \Phi\theta + e}$$

r is a small regularization parameter

- Regularized solution

$$\hat{\theta} = \left(\Phi^T\Phi + rI\right)^{-1}\Phi^T y$$

- Cut off the singular values of Φ that are smaller than r

Regularization

- Analysis through SVD (singular value decomposition)

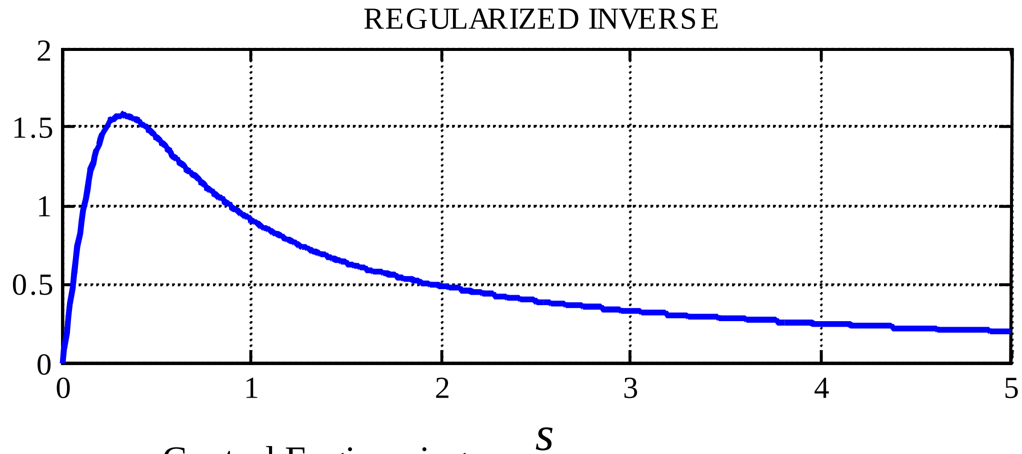
$$\Phi = USV^T; \qquad V \in R^{n,n}; U \in R^{m,m}; S = \mathrm{diag}\{s_j\}_{j=1}^n$$

- Regularized solution

$$\hat{\theta} = \left(\Phi^T \Phi + rI\right)^{-1} \Phi^T y = V \left[\mathrm{diag}\left\{ \frac{s_j}{s_j^2 + r} \right\}_{j=1}^n \right] U^T y$$

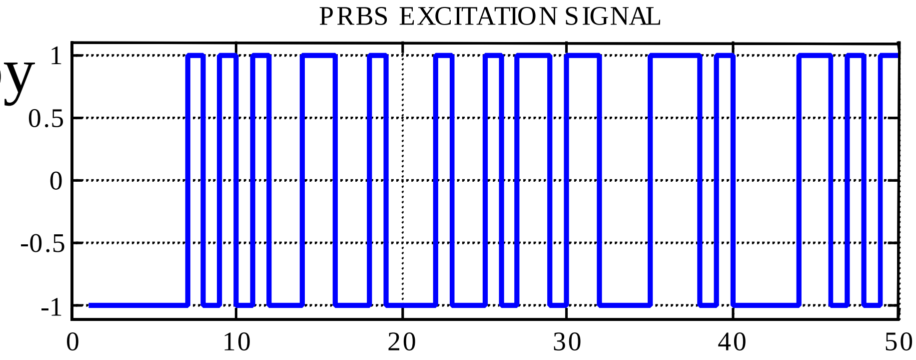
- Cut off the singular values of Φ that are smaller than r

$$\frac{s}{s^2 + 0.1}$$



Linear regression for FIR model

- Identifying impulse response by applying multiple steps
- PRBS excitation signal
- FIR (impulse response) model



PRBS =
Pseudo-Random Binary Sequence,
see `IDINPUT` in Matlab

$$y(t) = \sum_{k=1}^{K} h(k)u(t-k) + e(t)$$

- Linear regression representation

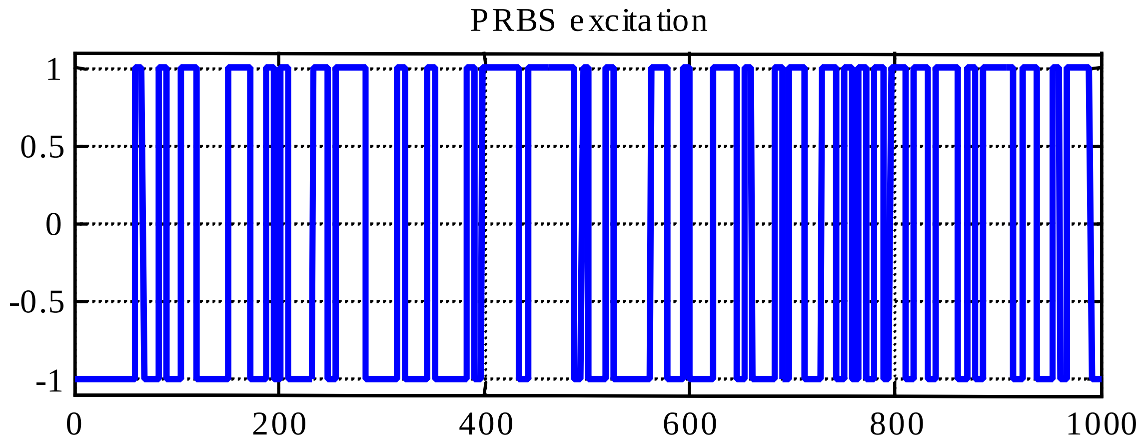
$$\varphi_1(t) = u(t-1)$$
$$\vdots$$
$$\varphi_K(t) = u(t-K)$$

$$\theta = \begin{bmatrix} h(1) \\ \vdots \\ h(K) \end{bmatrix}$$

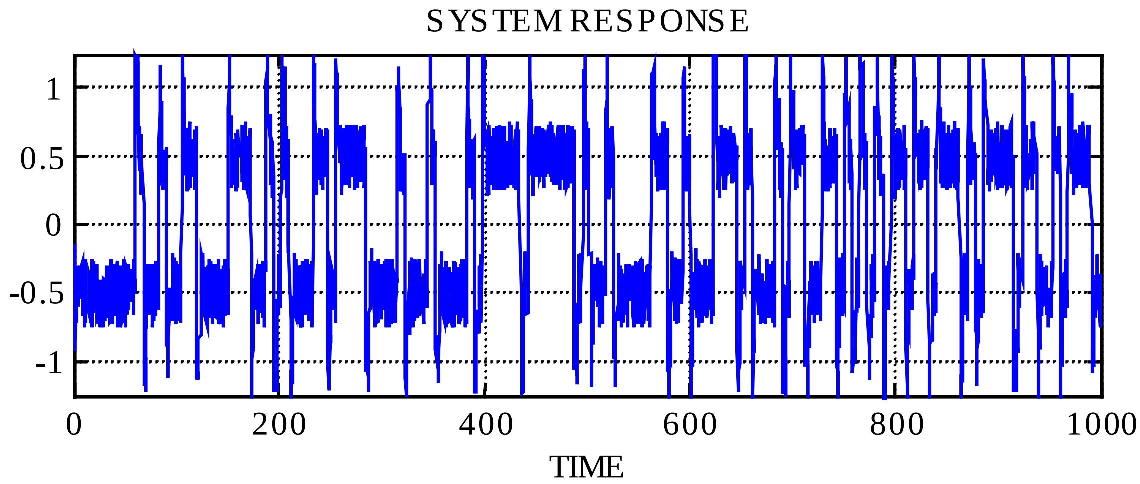
$$\hat{\theta} = \left(\Phi^T \Phi + rI\right)^{-1} \Phi^T y$$

Example: FIR model ID

- PRBS excitation input



- Simulated system output: 4000 samples, random noise of the amplitude 0.5

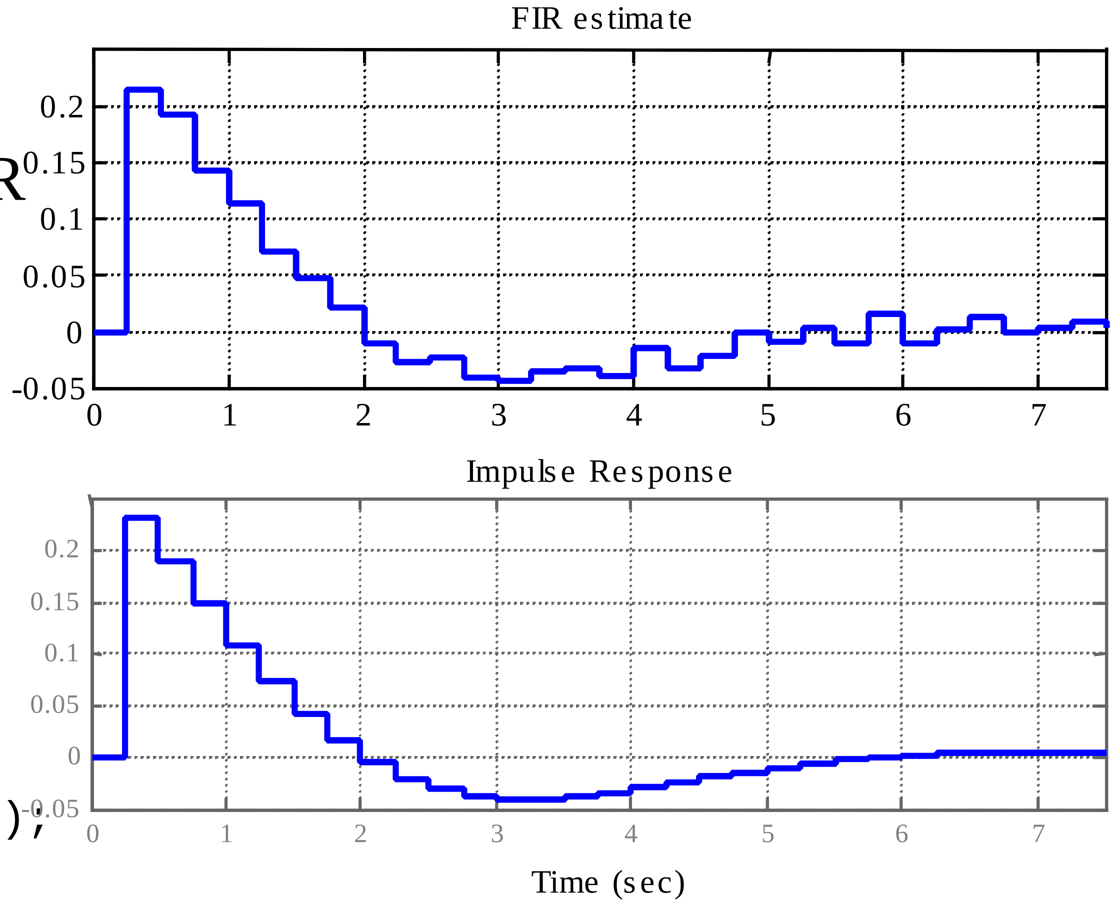


Example: FIR model ID

- Linear regression estimate of the FIR model

- Impulse response for the simulated system:

```
T=tf([1 .5],[1 1.1 1]);
P=c2d(T,0.25);
```



Nonlinear parametric model ID

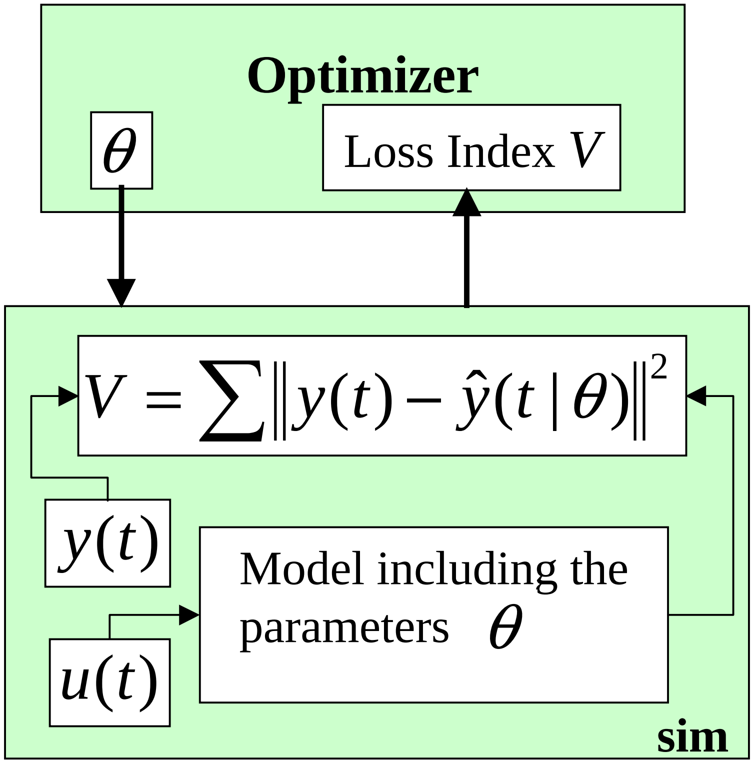
- Prediction model depending on the unknown parameter vector θ

$$u(t) \rightarrow \text{MODEL}(\theta) \rightarrow \hat{y}(t \mid \theta)$$

- Loss index

$$J = \sum \left\| y(t) - \hat{y}(t \mid \theta) \right\|^2$$

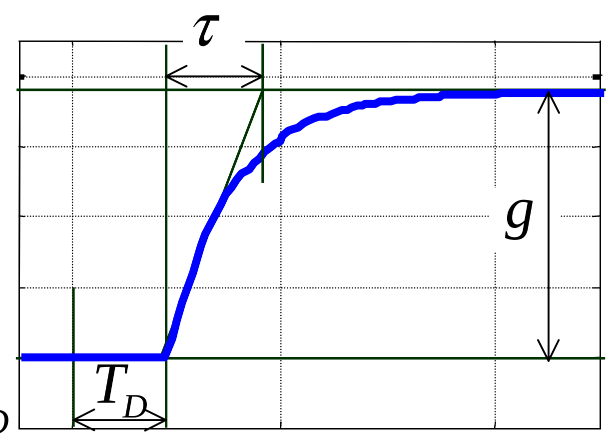
- Iterative numerical optimization. Computation of V as a subroutine



Lennart Ljung, "Identification for Control: Simple Process Models," *IEEE Conf. on Decision and Control*, Las Vegas, NV, 2002

Parametric ID of step response

- First order process with deadtime
- Most common industrial process model
- Response to a control step applied at t_B

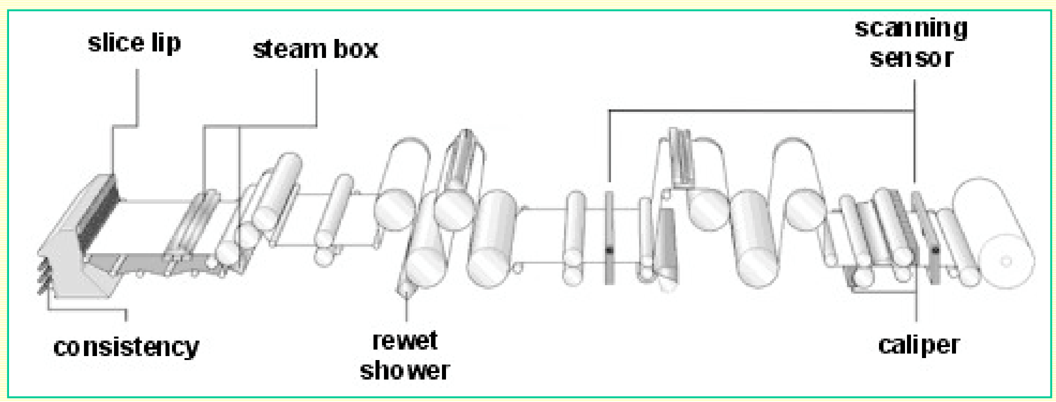


$$y(t\,|\,\theta) = \gamma + \begin{cases} g\left(1 - e^{(t - t_B - T_D)/\tau}\right), & \text{for } t > t_B - T_D \\ 0, & \text{for } t \le t_B - T_D \end{cases}$$

$$\theta = \begin{bmatrix} \gamma \\ g \\ \tau \\ T_D \end{bmatrix}$$

Example:

Paper machine process



Gain estimation

- For given τ, T_D, the modeled step response can be presented in the form

$$y(t\,|\,\theta) = \gamma + g \cdot y_1(t\,|\,\tau, T_D)$$

- This is a linear regression

$$y(t\,|\,\theta) = \sum_{k=1}^{2} w_k \varphi_k(t) \qquad \begin{aligned} w_1 &= g \\ w_2 &= \gamma \end{aligned} \qquad \begin{aligned} \varphi_1(t) &= y_1(t\,|\,\tau, T_D) \\ \varphi_2(t) &= 1 \end{aligned}$$

- Parameter estimate and prediction for given τ, T_D

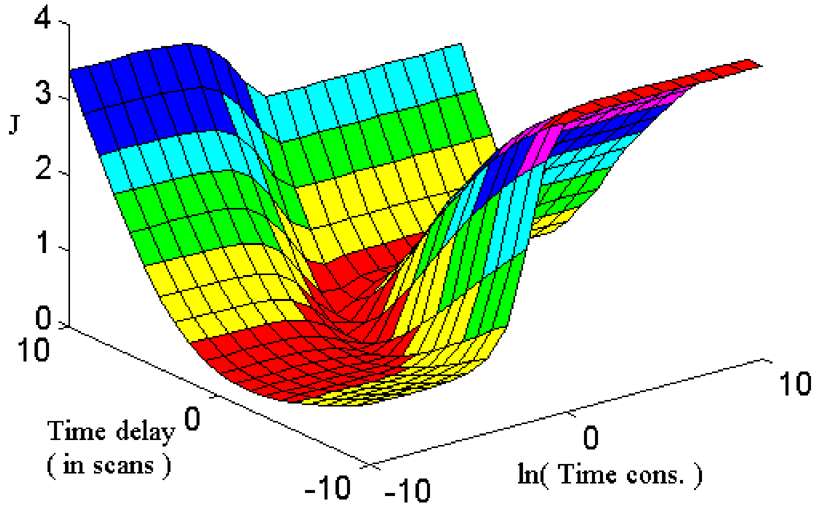
$$\hat{w}(\tau, T_D) = \left(\Phi^T \Phi\right)^{-1} \Phi^T y \qquad \hat{y}(t\,|\,\tau, T_D) = \hat{\gamma} + \hat{g} \cdot y_1(t\,|\,\tau, T_D)$$

Rise time/dead time estimation

- For given τ, T_D, the loss index is

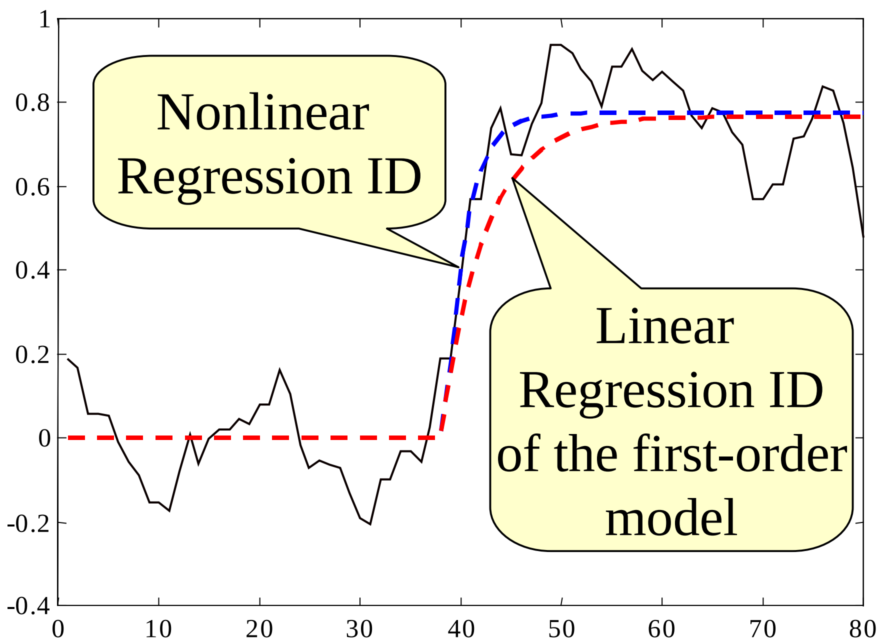
$$V = \sum_{t=1}^{N} \left| y(t) - \hat{y}(t \mid \tau, T_D) \right|^2$$

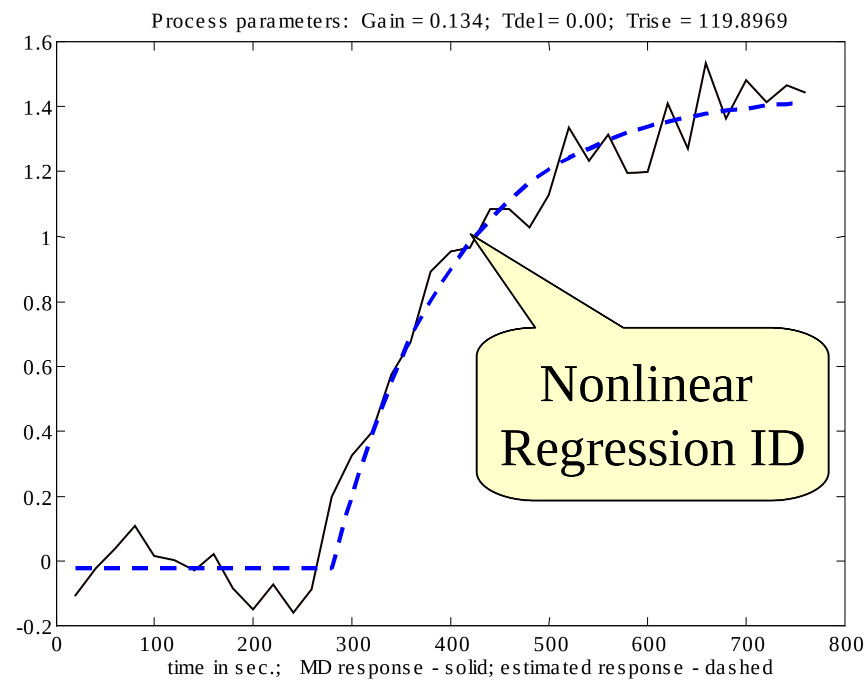
- Grid τ, T_D and find the minimum of $V = V(\tau, T_D)$



Examples: Step response ID

- Identification results for real industrial process data
- This algorithm works in an industrial tool used in 500+ industrial plants, many processes each





Linear filtering

- A trick that helps: pre-filter data
- Consider data model

$$y = h * u + e$$

- L is a linear filtering operator, usually LPF

$$\underbrace{Ly}_{y_f} = L(h * u) + \underbrace{Le}_{e_f}$$

$$L(h * u) = (Lh) * u = h * (Lu)$$

- Can estimate h from filtered y and filtered u
- Or can estimate filtered h from filtered y and 'raw' u
- Pre-filter bandwidth will limit the estimation bandwidth

Multivariable ID

- Apply SISO ID to various input/output pairs
- Need *n* tests - excite each input in turn

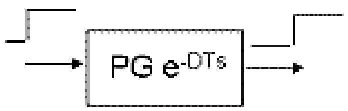
- Step/pulse response identification is a key part of the industrial Multivariable Predictive Control packages.

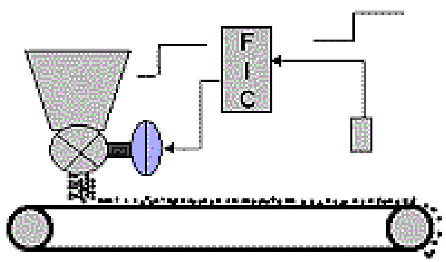
Lecture 9 - Processes with Deadtime, IMC

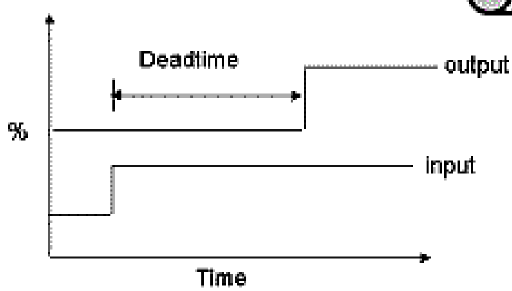
- Processes with deadtime
- Model-reference control
- Deadtime compensation: Dahlin controller
- IMC
- Youla parametrization of all stabilizing controllers
- Nonlinear IMC
  - Dynamic inversion - Lecture 13
  - Receding Horizon - MPC - Lecture 12

Processes with deadtime

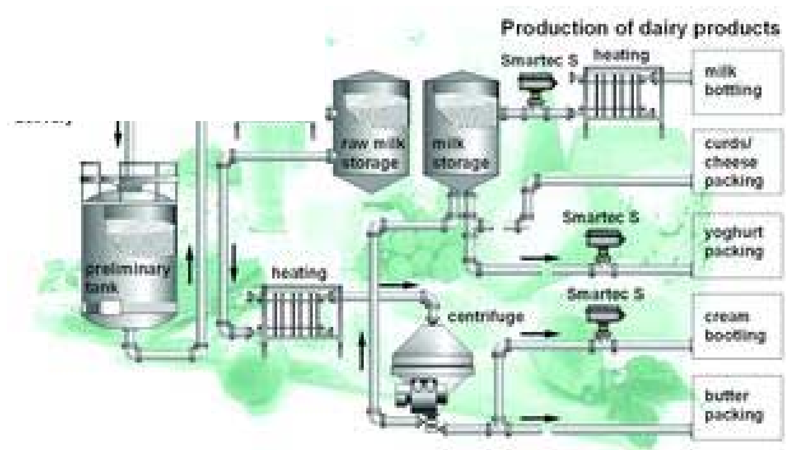
- Examples: transport deadtime in mining, paper, oil, food





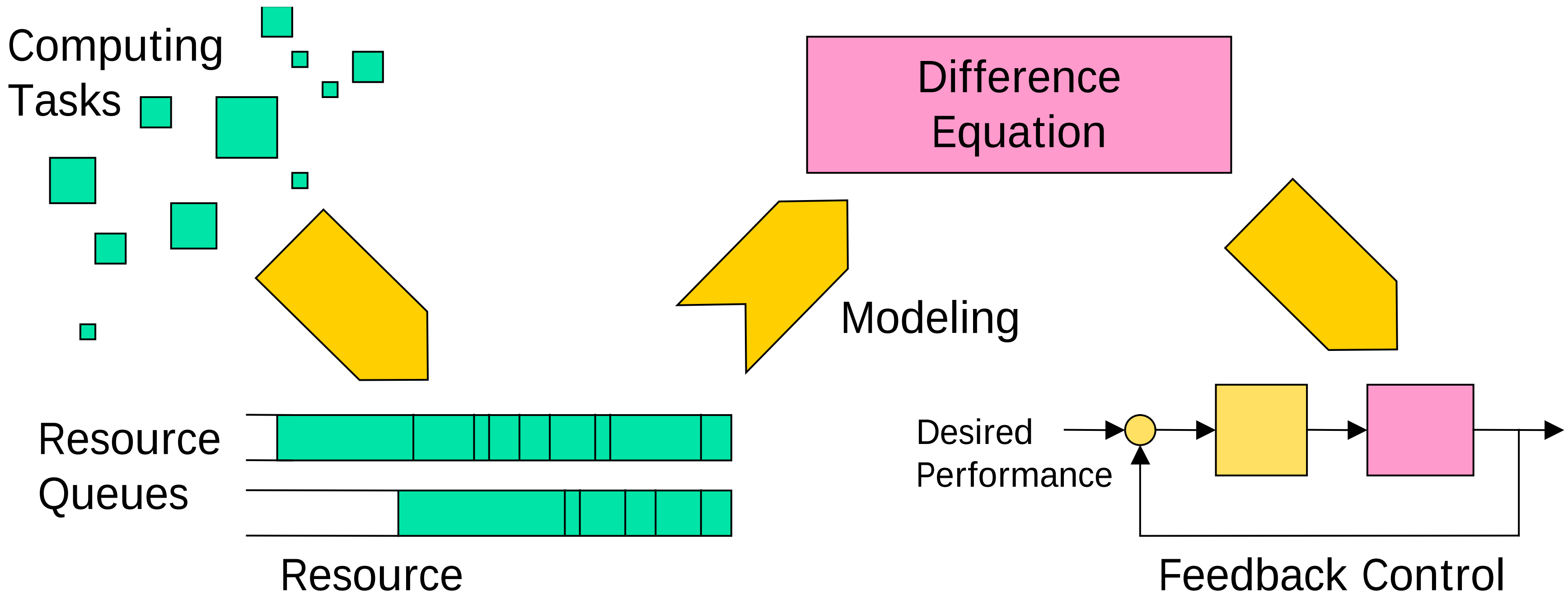


Deadtime = Transportation Time



Processes with deadtime

- Example: resource allocation in computing

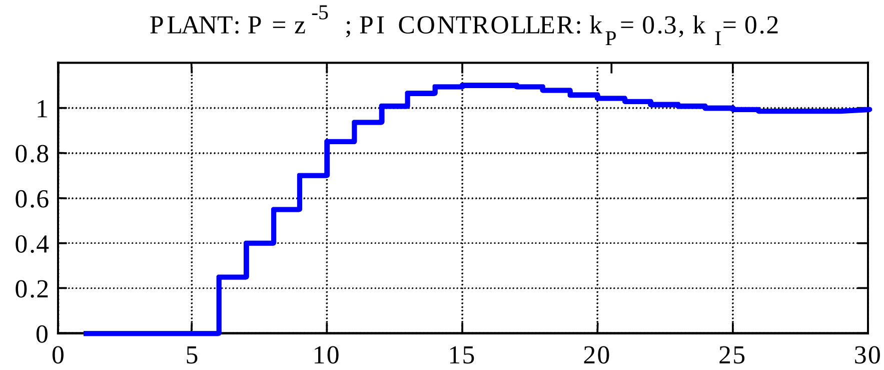


Control of process with deadtime

- PI control of a deadtime process

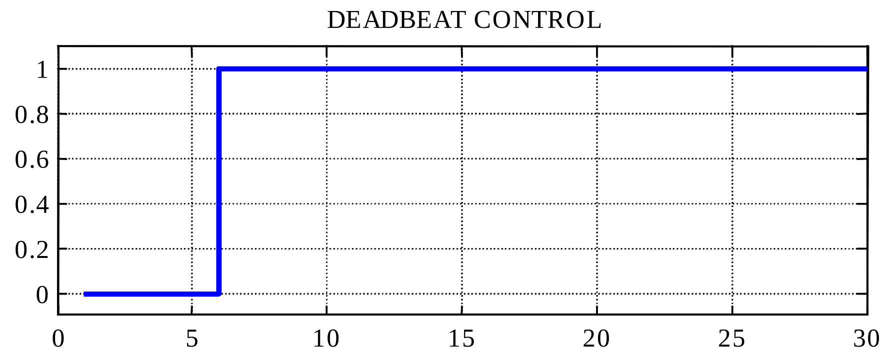
$$P = e^{-sT_D} \quad \text{continuous time}$$

$$P = z^{-d} \quad \text{discrete time}$$



- Can we do better?
  - Make $\dfrac{PC}{1+PC} = z^{-d}$
  - Deadbeat controller

$$PC = \frac{z^{-d}}{1-z^{-d}} \Rightarrow C = \frac{1}{1-z^{-d}}$$



$$u(t) = u(t-d) + e(t)$$

Model-reference control

- Deadbeat control has bad robustness, especially w.r.t. deadtime
- More general model-reference control approach
  - make the closed-loop transfer function as desired

$$\frac{P(z)C(z)}{1+P(z)C(z)} = Q(z)$$

$$C(z) = \frac{1}{P(z)} \cdot \frac{Q(z)}{1-Q(z)}$$

- Works if $Q(z)$ includes a deadtime, at least as large as in $P(z)$

Dahlin's controller

- Eric Dahlin worked for IBM in San Jose (?) then for Measurex in Cupertino.

$$C(z) = \frac{1}{P(z)} \cdot \frac{Q(z)}{1-Q(z)}$$

- Dahlin's controller, 1968

$$P(z) = \frac{g(1-b)}{1-bz^{-1}} z^{-d}$$

• plant, generic first order response with deadtime

$$Q(z) = \frac{1-\alpha}{1-\alpha z^{-1}} z^{-d}$$

• reference model

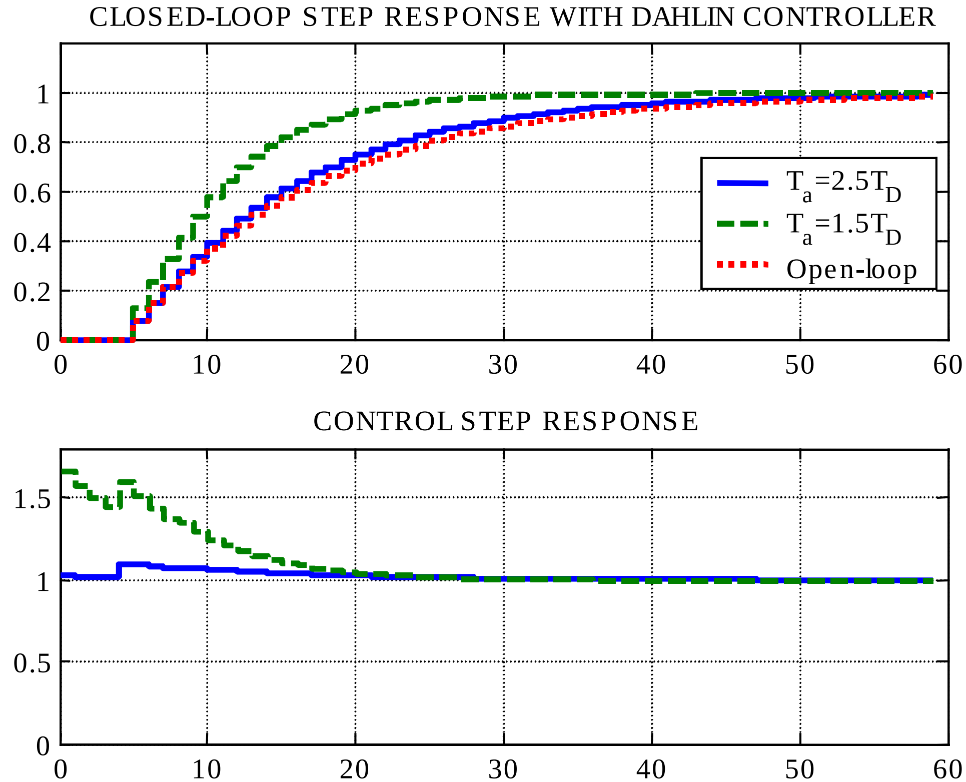
$$C(z) = \frac{1-bz^{-1}}{g(1-b)} \cdot \frac{1-\alpha}{1-\alpha z^{-1} - (1-\alpha) z^{-d}}$$

• Dahlin's controller

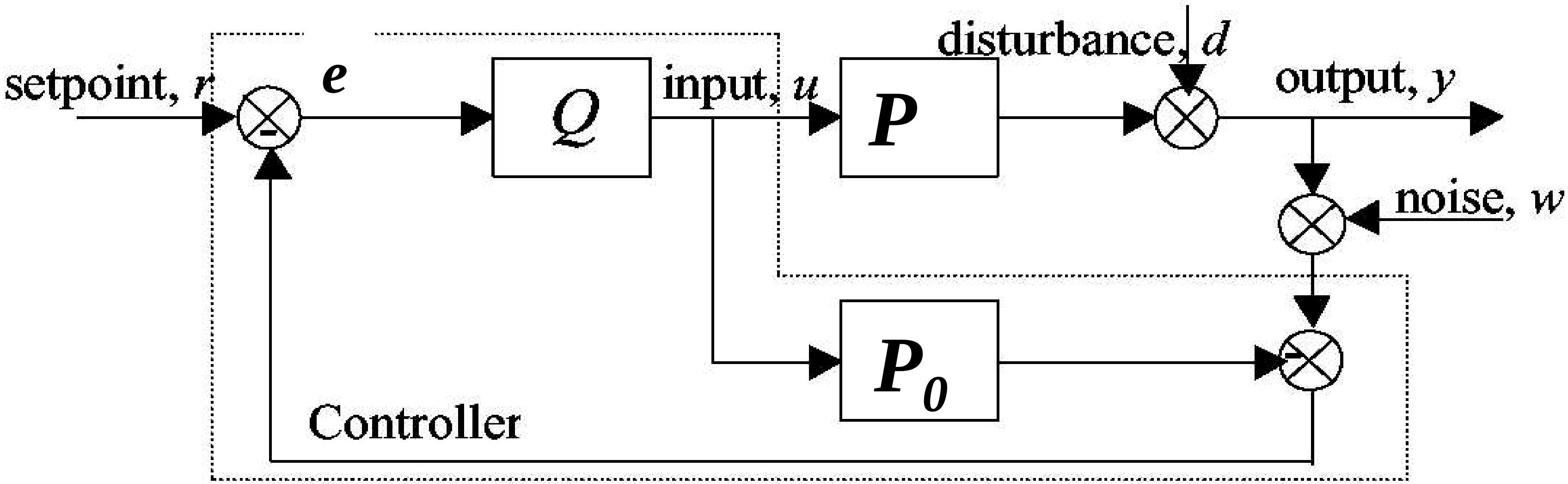
- Single tuning parameter: α - tuned controller

Dahlin's controller

- Dahlin's controller is broadly used through paper industry in supervisory control loops - Honeywell-Measurex, 60%.
- Direct use of the identified model parameters.
- Industrial tuning guidelines: Closed loop time constant = 1.5-2.5 deadtime.



Internal Model Control - IMC



$$e = r - (y - Pu)$$
$$u = Qe$$

$$\Rightarrow \quad C = \frac{Q}{1 - QP_0}$$

- continuous time *s*
- discrete time *z*

IMC and Youla parametrization

- Sensitivities

$$C = \frac{Q}{1 - QP_0}$$

$$S = 1 - QP_0 \qquad d \to y$$

$$T = QP_0 \qquad r \to y$$

$$S_u = Q \qquad d \to u$$

$$Q = \frac{C}{1 + CP_0}$$

- If Q is stable, then S, T, and the loop are stable
- If loop is stable, then Q is stable

- Choosing various stable Q parameterizes all stabilizing controllers
- This is called Youla parameterization
- Youla parameterization is valid for unstable systems as well

Q-loopshaping

- Systematic controller design: select Q to achieve the tradeoff
- The approach used in modern advanced control design: H_2/H_∞, LMI, H_∞ loopshaping

- Q-based loopshaping:

$$S = 1 - QP_0 \qquad S << 1 \Rightarrow Q \approx \left(P_0\right)^{-1}$$

• in band

- Recall system inversion

Inversion

Q-loopshaping

- Loopshaping

$$S = 1 - QP_0 \qquad S << 1 \Rightarrow Q \approx \left(P_0\right)^{-1}$$ • in band

$$T = QP_0 \qquad T << 1 \Rightarrow QP_0 << 1$$ • out of band

- Lambda-tuned IMC †

$$Q = FP_0^{\dagger}, \qquad S = 1 - QP_0 \approx 1 - F$$

$$F = \frac{1}{\left(1 + \lambda s\right)^n}$$

Loopshaping

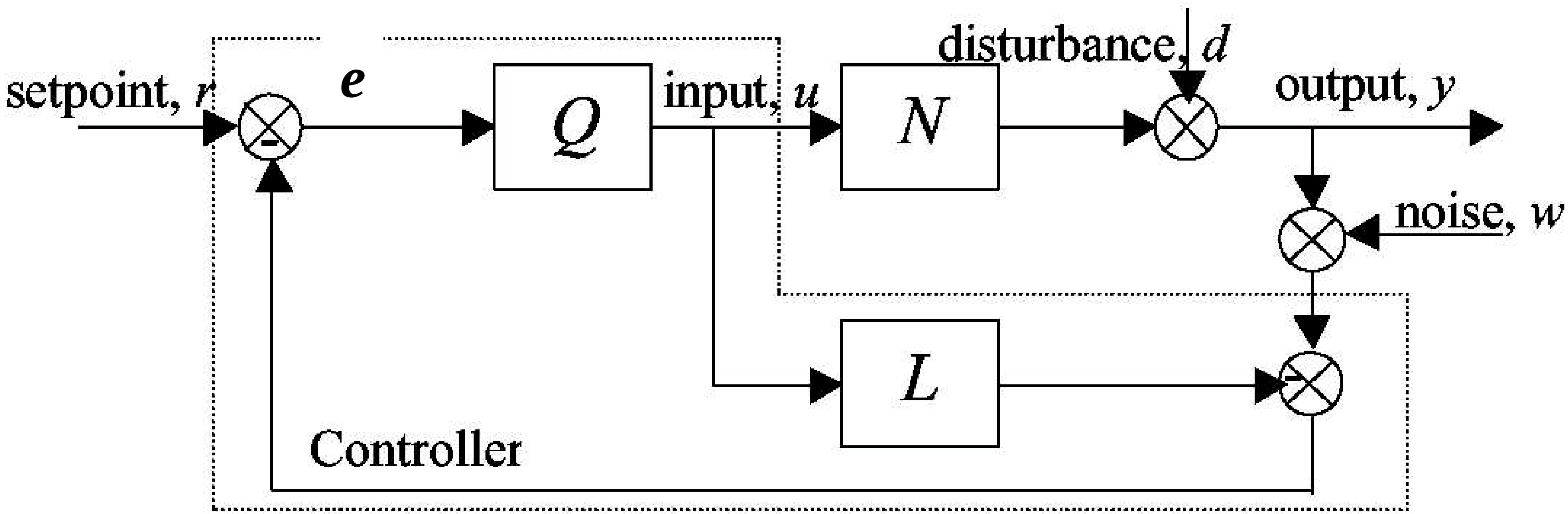
- F is called IMC filter, $F \approx T$, reference model for the output
- For minimum phase plant $Q = FP_0^{\dagger} = F\left(P_0\right)^{-1}, \qquad T = F$

IMC extensions

- Multivariable processes
- Nonlinear process IMC
- Dynamic inversion in flight control - Lecture 13 - ?
- Multivariable predictive control - Lecture 12

Nonlinear process IMC

- Can be used for nonlinear processes
  - linear Q
  - nonlinear model P_0
  - linearized model L



Industrial applications of IMC

- Multivariable processes with complex dynamics
- Demonstrated and implemented in process control by academics and research groups in very large corporations.
- Not used commonly in process control (except Dahlin controller)
  - detailed analytical models are difficult to obtain
  - field support and maintenance
    - process changes, need to change the model
    - actuators/sensors off
    - add-on equipment

Dynamic inversion in flight control

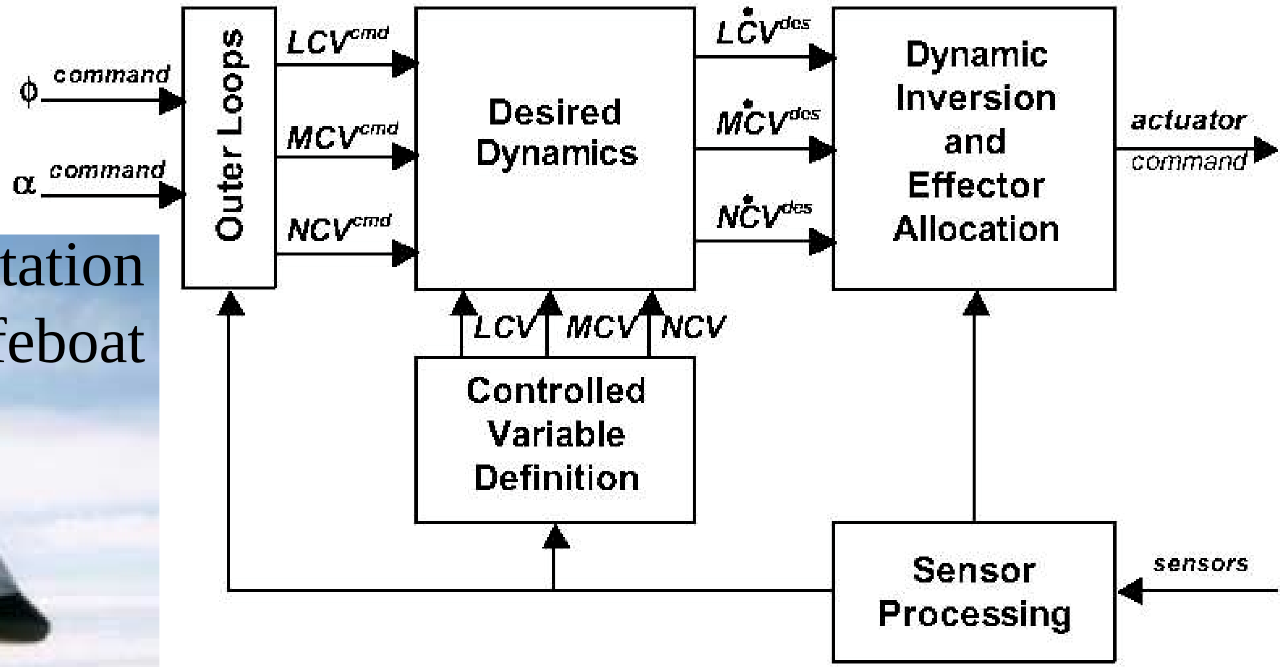
$$\dot{v} = F(x,v) + G(x,v)u$$

$$u = G^{-1}(\dot{v}^{des} - F)$$

$$v = \begin{bmatrix} LCV \\ MCV \\ NCV \end{bmatrix}$$

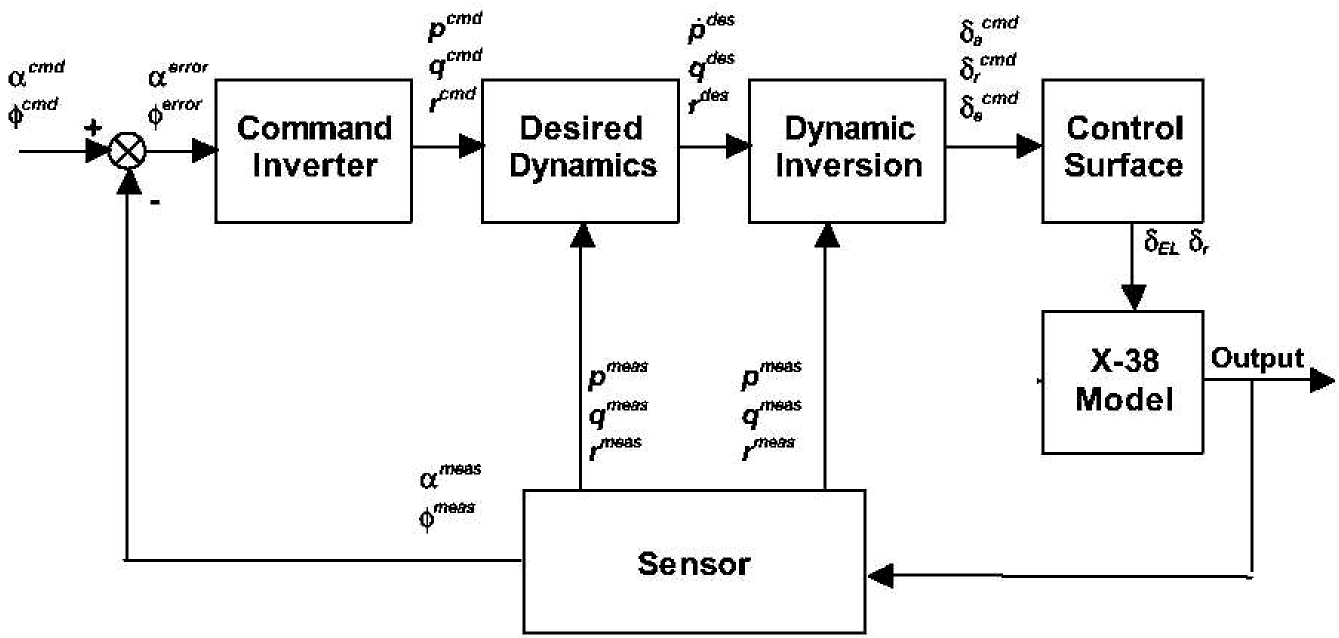
- Honeywell MACH

X-38 - Space Station Lifeboat



Dynamic inversion in flight control

- NASA JSC study for X-38
- Actuator allocation to get desired forces/moments
- Reference model (filter): vehicle handling and pilot ‘feel’
- Formal robust design/analysis (μ-analysis etc)



Summary

- Dahlin controller is used in practice
  - easy to understand and apply
- IMC is not really used much
  - maintenance and support issues
- Youla parameterization is used as a basis of modern advanced control design methods.
  - Industrial use is very limited.
- Dynamic inversion is used for high performance control of air and space vehicles
  - this was presented for breadth, the basic concept is simple
  - need to know more of advanced control theory to apply in practice

Lecture 10 - Optimization

- LP
  - Process plants - Refineries
  - Actuator allocation for flight control
  - More interesting examples
- Introduce QP problem

- More technical depth
  - E62 - Introduction to Optimization - basic
  - EE364 - Convex Optimization - more advanced

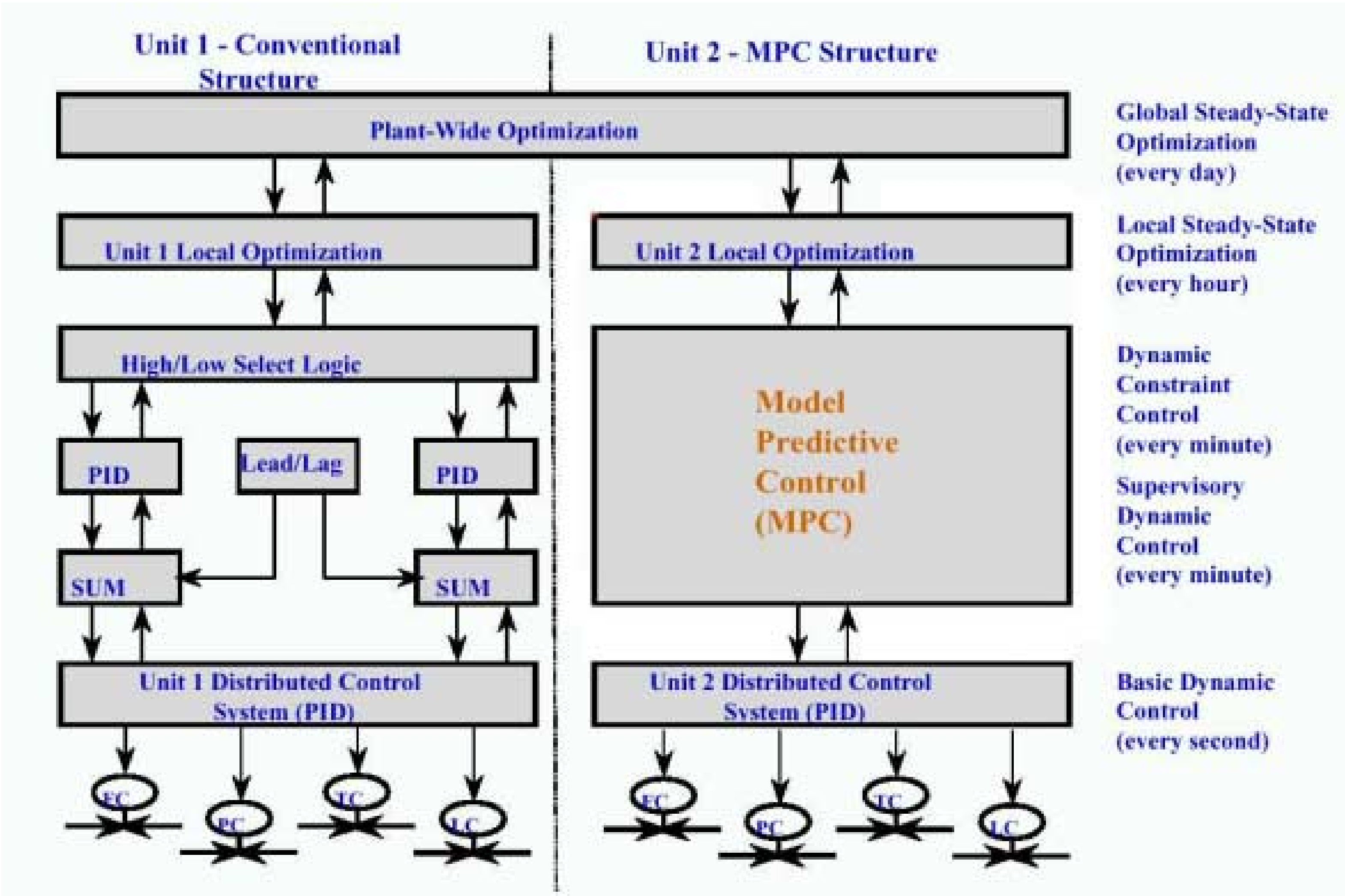
Real-time Optimization in Control

- Important part of multivariable control systems
- Many actuators, control handles
- Quasistatic control, dynamics are not important
  - slow process
  - low-level fast control loops
  - fast actuators

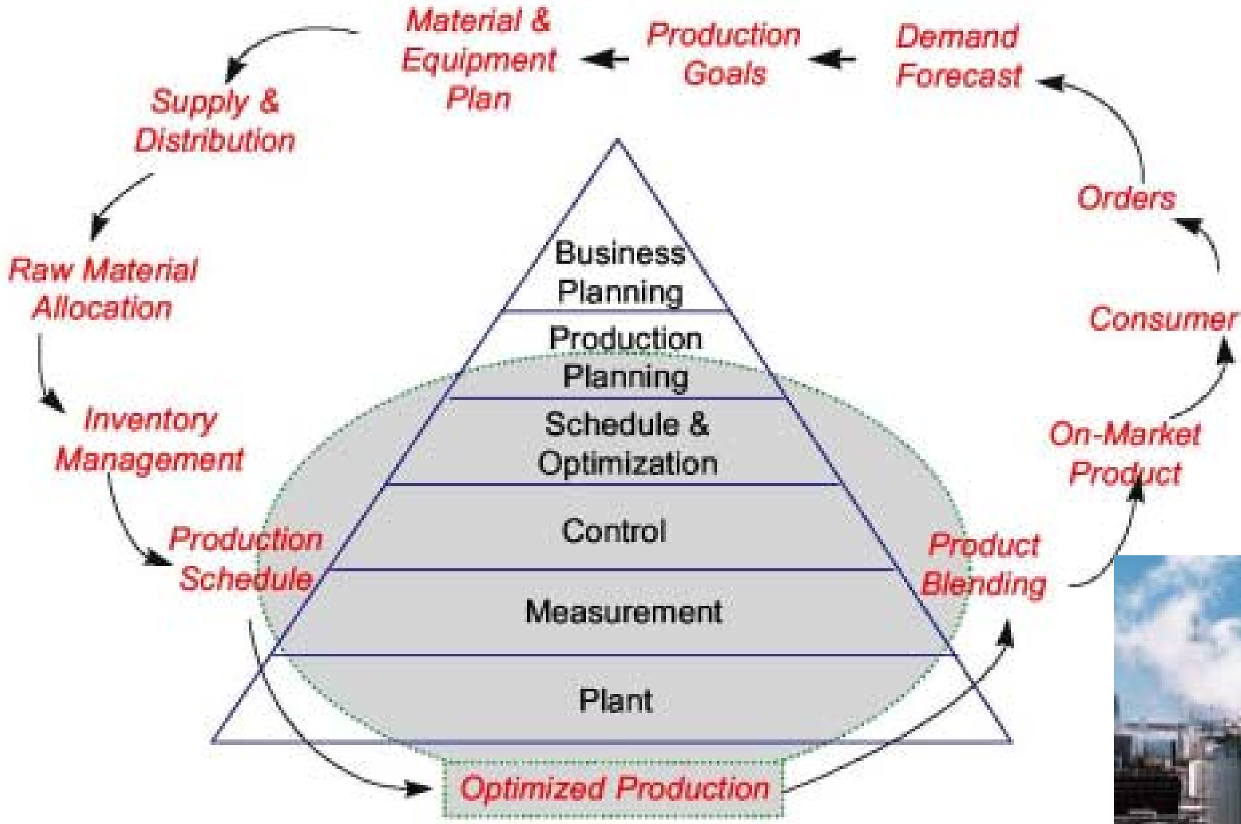
Optimization methods

- Need to state problem such that a solution can be computed quickly, efficiently, reliably
- Least squares - linear quadratic problems
  - analytical closed form, matrix multiplication and inversion
- Linear Programming
  - simplex method
- Quadratic Programming
  - interior point
- Convex optimization: includes LP, QP, and more

Optimization in Process Plants



Optimization in Process Plants



Linear programming

- LP Problem:

$$Ax \le b$$
$$Gx = h$$
$$J = f^T x \to \min$$

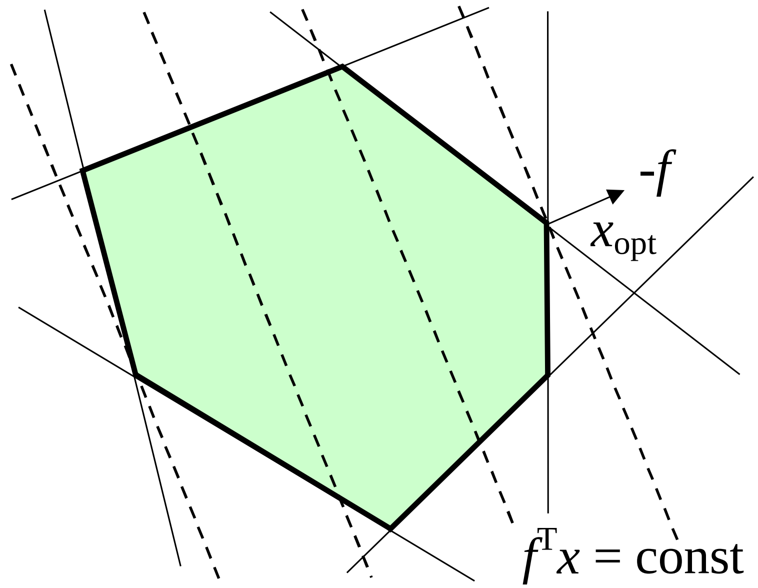
$$x \le y \quad \Leftrightarrow \quad \begin{bmatrix} x_1 \le y_1 \\ \vdots \\ x_n \le y_n \end{bmatrix}$$

- Might be infeasible! … no solution satisfies all constraints

- Matlab Optimization Toolbox: **`LINPROG`**

Linear programming

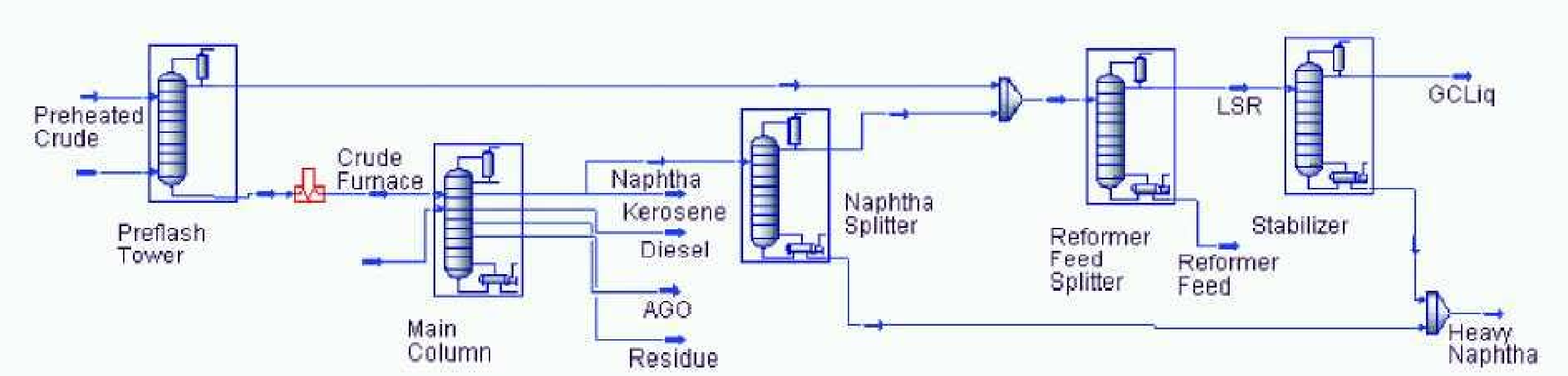
$$Ax \leq b$$
$$Gx = h$$
$$J = f^T x \rightarrow \min$$



- Simplex method in a nutshell:
  - check the vertices for value of J, select optimal
  - issue: exponential growth of number of vertices with the problem size
  - Need to do 10000 variables and 500000 inequalities.
- Modern interior point methods are radically faster
  - no need to understand, standard solvers are available

Refinery Optimization

- Crude supply chain - multiple oil sources
- Distillation - separating fractions
- Blending - ready products, given octane ratings
- Objective function - profit
- LP works ideally:
  - linear equalities and inequalities, single linear objective function



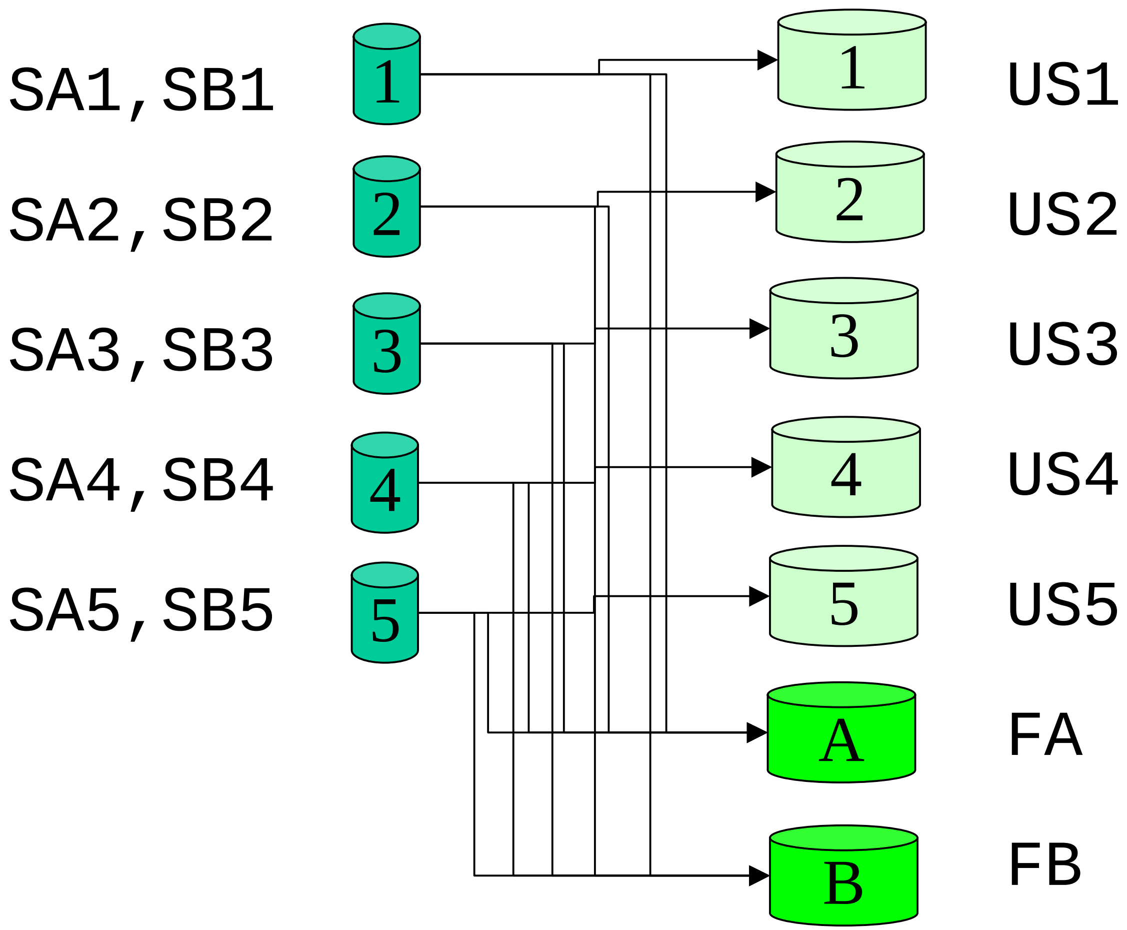
Blending Example

- A Blending Problem: A refinery produces two grades of fuel, A and B, which are made by blending five raw stocks of differing octane rating, cost and availability

Gasoline	Octane Rating	Price $/B
A	93	37.5
B	85	28.5

Stock	Octane Rating	Price $/B	Availability
1	70	9.0	2000
2	80	12.5	4000
3	85	12.5	4000
4	90	27.5	5000
5	99	27.5	3000

Blending Example



Blending Example

- LP problem formulation:

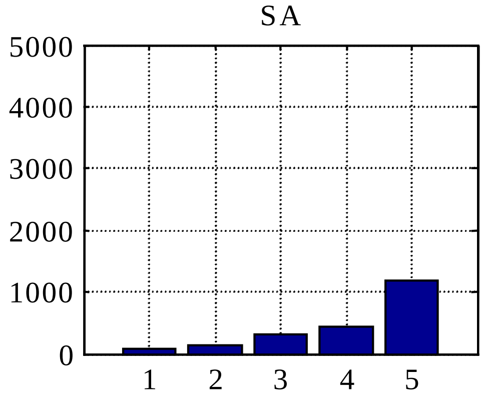
```
J = 9US1 + 12.5US2 + 12.5US3 + 27.5US4 + 27.5US5 + 37.5FA + 28.5FB -> MAX

[Stock Availability]
S1A                  +S1B                +US1               = 2000
    S2A              +    S2B            +   US2            = 4000
        S3A          +        S3B        +      US3         = 4000
            S4A      +           S4B     +         US4      = 5000
                S5A+                S5B +             US5 = 3000

[Fuel Quantity]
S1A+S2A+S3A+S4A+S5A                                        = FA
                    S1B+S2B+S4B+S5B                        = FB

[Fuel Quality]
70S1A + 80S2A + 85S3A + 90S4A + 99S5A                      ≥ 93FA [Quality A]
70S1B + 80S2B + 85S3B + 90S4B + 99S5B                      ≥ 85FB [Quality B]

[Nonnegativity]
S1A,S2A,S3A,S4A,S5A,S1B,S2B,S4B,S5B,US1,US2,US3,US4,US5,FA,FB ≥ 0
```

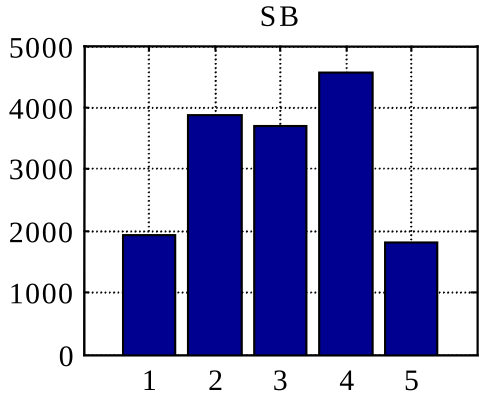
Matlab code for the example

```
%    OctRt  Price $/B
Gas = [93       37.5;
       85       28.5];
%Stock  OctRt    Price $/B    Availability
Stock = [70        12.5         2000;
         80        12.5         4000;
         85        12.5         4000;
         90        27.5         5000;
         99        27.5         3000];
% Revenue
f = [zeros(10,1); Stock(:,3);  Gas(:,2)];
% Equality constraint
G = [eye(5,5)    eye(5,5)    eye(5,5)    zeros(5,2);
     ones(1,5)   zeros(1,5)  zeros(1,5)  -1    0;
     zeros(1,5)  ones(1,5)   zeros(1,5)   0   -1];
h = [Stock(:,3); zeros(2,1)];
% Inequality (fuel quality) constraints
A = [-[Stock(:,1)' zeros(1,5) zeros(1,5);
       zeros(1,5)  Stock(:,1)' zeros(1,5)] diag(Gas(:,1))];
b = zeros(2,1);
% X=LINPROG(f,A,b,Aeq,beq,LB,UB)
x = linprog(-f,A,b,G,h,zeros(size(f)),[]);
Revenue = f'*x
```

Blending Example - Results

- Blending distribution:

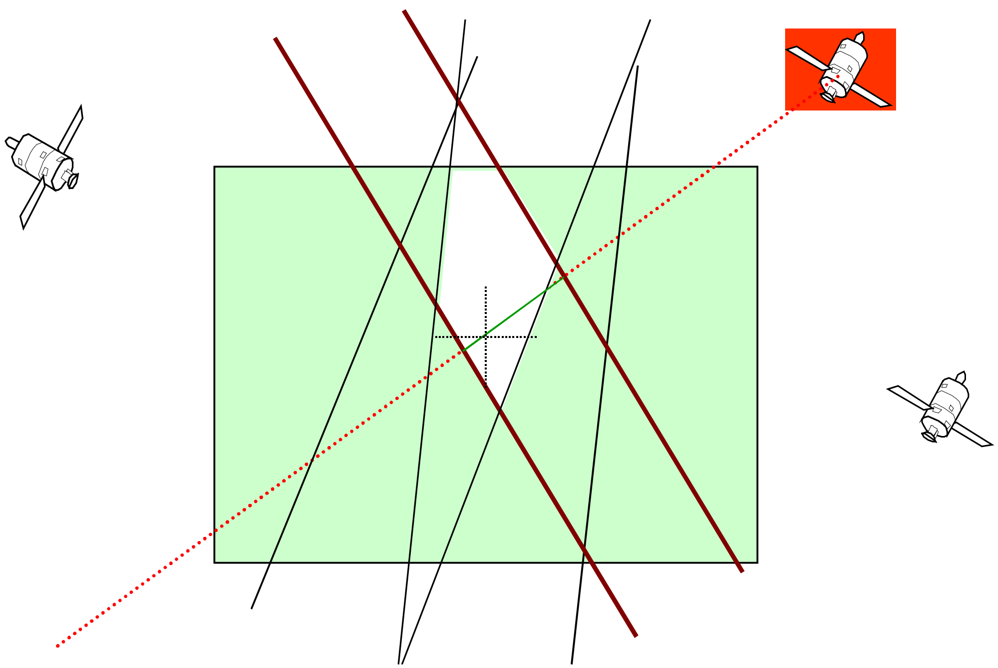




Produced Fuel:		Total Revenue:
A	2125	$532,125
B	15875	

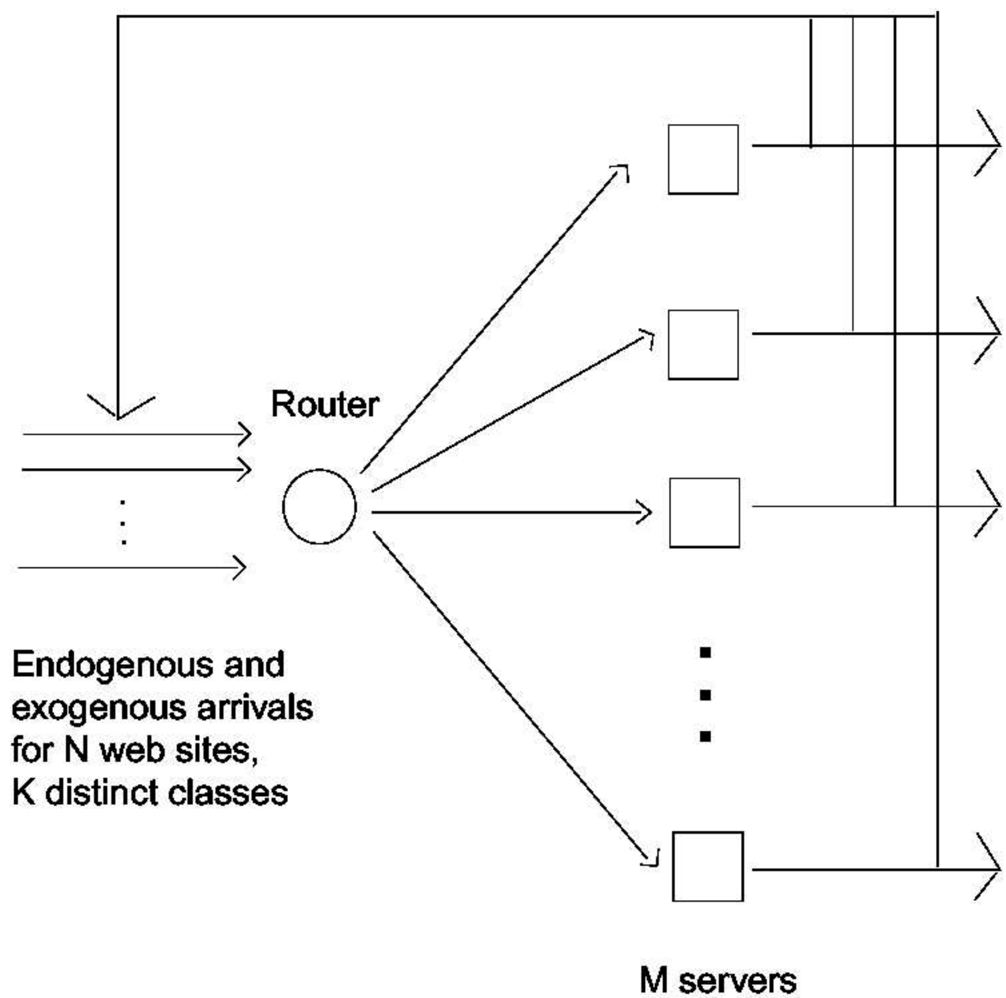
GPS

- Determining coordinates by comparing distances to several satellites with known positions
- See E62 website:
  `http://www.stanford.edu/class/engr62e/handouts/GPSandLP.ppt`

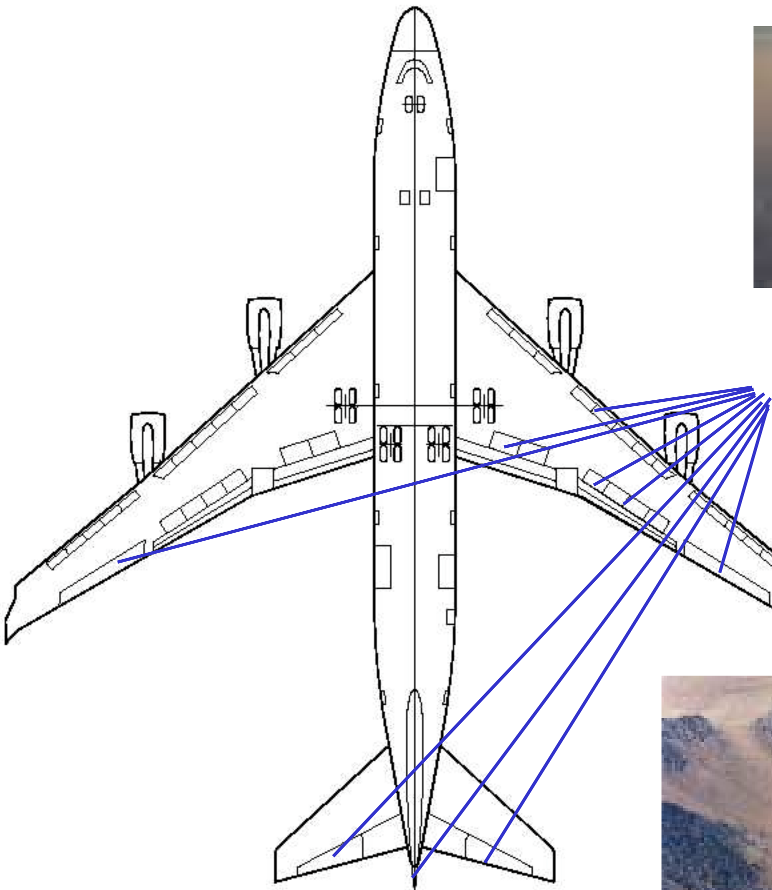


Computing Resource Allocation

- Web Server Farm
- LP formulation for the optimal load distribution



Aircraft actuator allocation



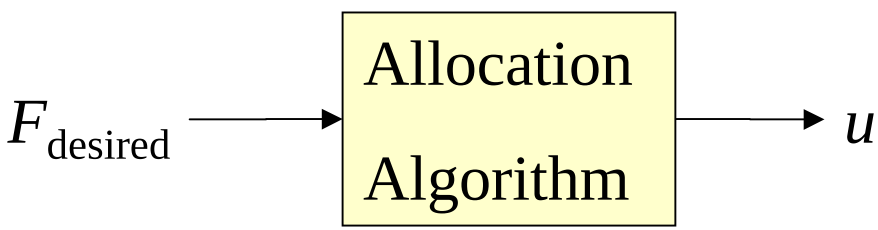
- Mutiple flight control surfaces

Aircraft actuator allocation

- Mutiple flight control surfaces: ailerons, elevons, canard foreplanes, trailing and leading edge flaps, airbrakes, etc

$$\begin{bmatrix} M_{roll} \\ M_{pitch} \\ M_{yaw} \end{bmatrix} = B(\alpha, \varphi, V)u$$

$$F = Bu$$



Actuator allocation

- Simplest approach - least squares

$$u = B^{\dagger} F$$

$$B^{\dagger} = \left(B^T B\right)^{-1} B^T \quad \text{solves} \quad Bu = F, \quad \|u\|_2^2 \to \min$$

- LP optimization approach

$$Bu = F, \quad \|w^T u\|_1 \to \min$$

$$\|w^T u\|_1 = \sum w_k \cdot |u_k|, \qquad w_k \ge 0$$

LP

$$w^T u^+ + w^T u^- \to \min$$
$$u^+ \ge 0$$
$$u^- \ge 0$$
$$Bu^+ - Bu^- = F$$

Solve the LP, get $u = u^+ - u^-$

Actuator allocation

- Need to handle actuator constrains (v - scale factor)

$$\|w^T u\|_1 - v \to \min \qquad u^l \le u \le u^u$$

$$Bu = vF \qquad 0 \le v \le 1$$

- LP can be extended to include actuator constrains

$$w^T u^+ + w^T u^- - v \to \min$$
$$Bu^+ - Bu^- - vF = 0$$
$$u^l \le u^+ \le u^u$$
$$u^l \le -u^- \le u^u$$
$$0 \le v \le 1$$

$$f^T = \begin{bmatrix} w^T & w^T & -1 \end{bmatrix}$$

$$A = \begin{bmatrix} I & 0 & 0 \\ -I & 0 & 0 \\ 0 & -I & 0 \\ 0 & I & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}, b = \begin{bmatrix} u^u \\ -u^l \\ u^u \\ -u^l \\ 1 \\ 0 \end{bmatrix}, \quad x = \begin{bmatrix} u^+ \\ u^- \\ v \end{bmatrix}$$

$$G = \begin{bmatrix} B & -B & -F \end{bmatrix}, h = 0$$

$$Ax \le b$$
$$Gx = h$$
$$f^T x \to \min$$

Actuator allocation example

- Problem:

$$\left\| w^T u \right\|_1 - v \rightarrow \min$$

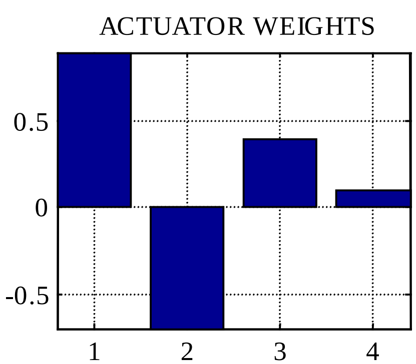
$$Bu = vF$$

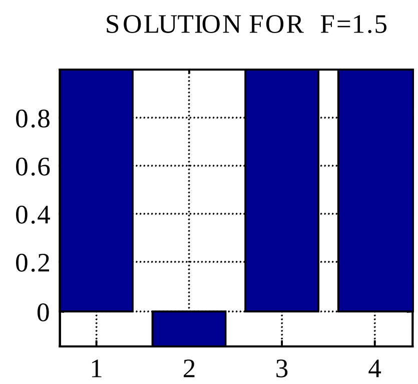
$$B = [0.9 \quad -0.7 \quad 0.4 \quad 0.1]$$

$$w = [0.1 \quad 0.1 \quad 0.02 \quad 0.001]$$

$$-1 \le u \le 1$$

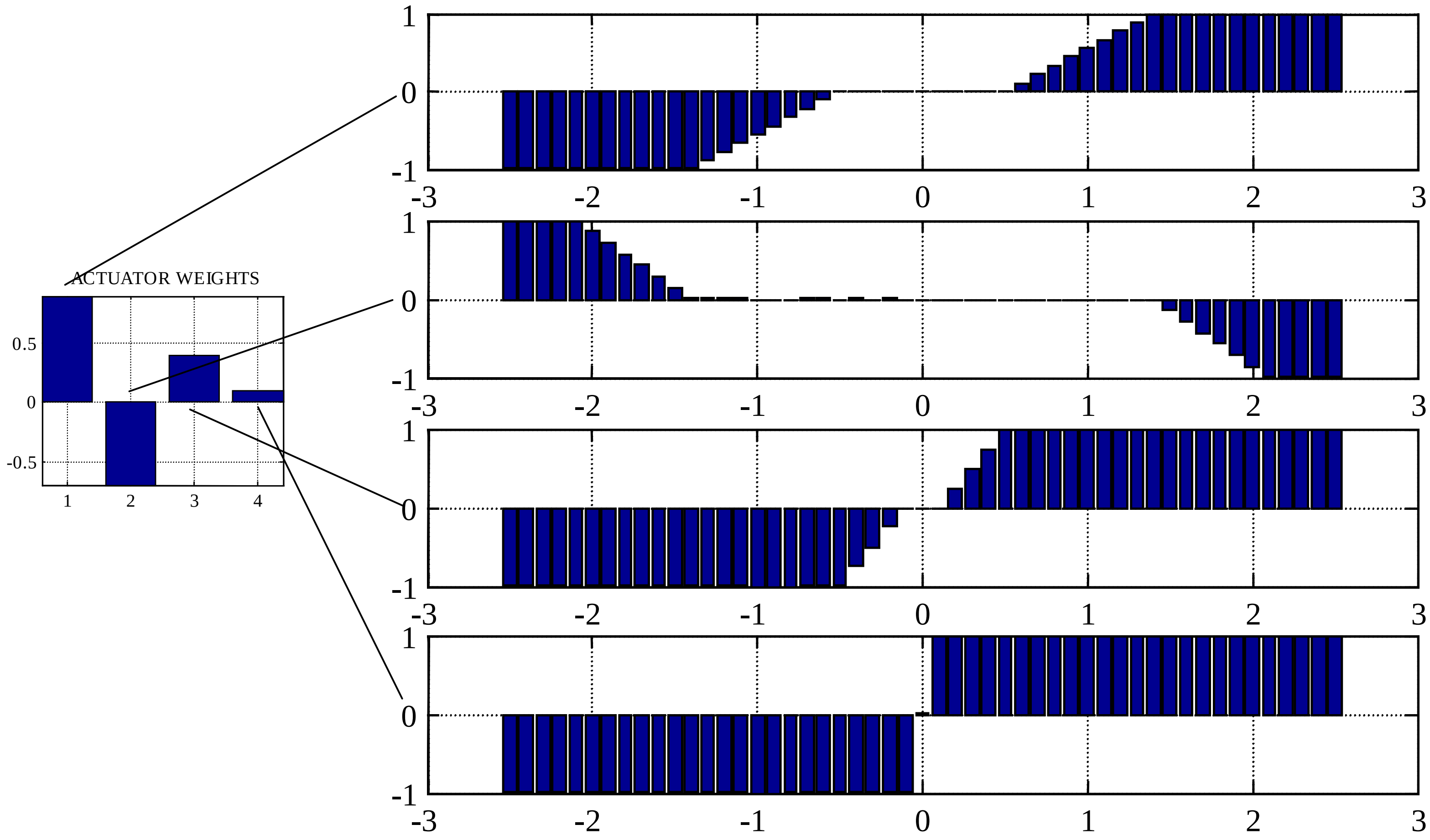
- LP problem solution for $F = 1.5$





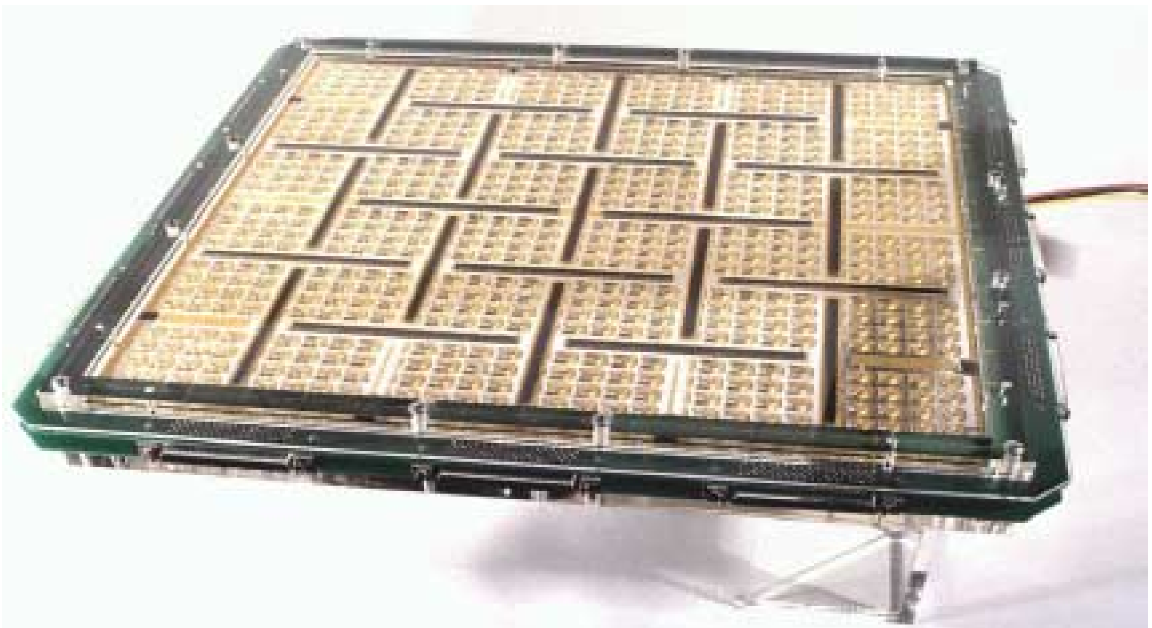
Actuator allocation example

- LP problem solution for *F* from -2.5 to 2.5



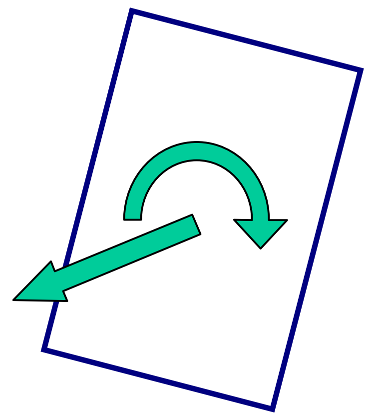
Extreme actuator allocation

- (Xerox) PARC jet array table
- Jets must be allocated to achieve commanded total force and torque acting on a paper sheet



$$\mathbf{F} = \sum \vec{f}_k$$

$$\mathbf{T} = \sum \vec{f}_k \times \vec{r}_k$$



Actuator allocation

- Least squares + actuator constraint

$$Bu = F,$$

$$\|u\|^2 \rightarrow \min$$

$$u^l \le u \le u^u$$

- This is a QP optimization problem

Quadratic Programming

- QP Problem:

$$Ax \le b$$
$$Gx = h$$
$$J = \frac{1}{2} x^T H x + f^T x \to \min$$

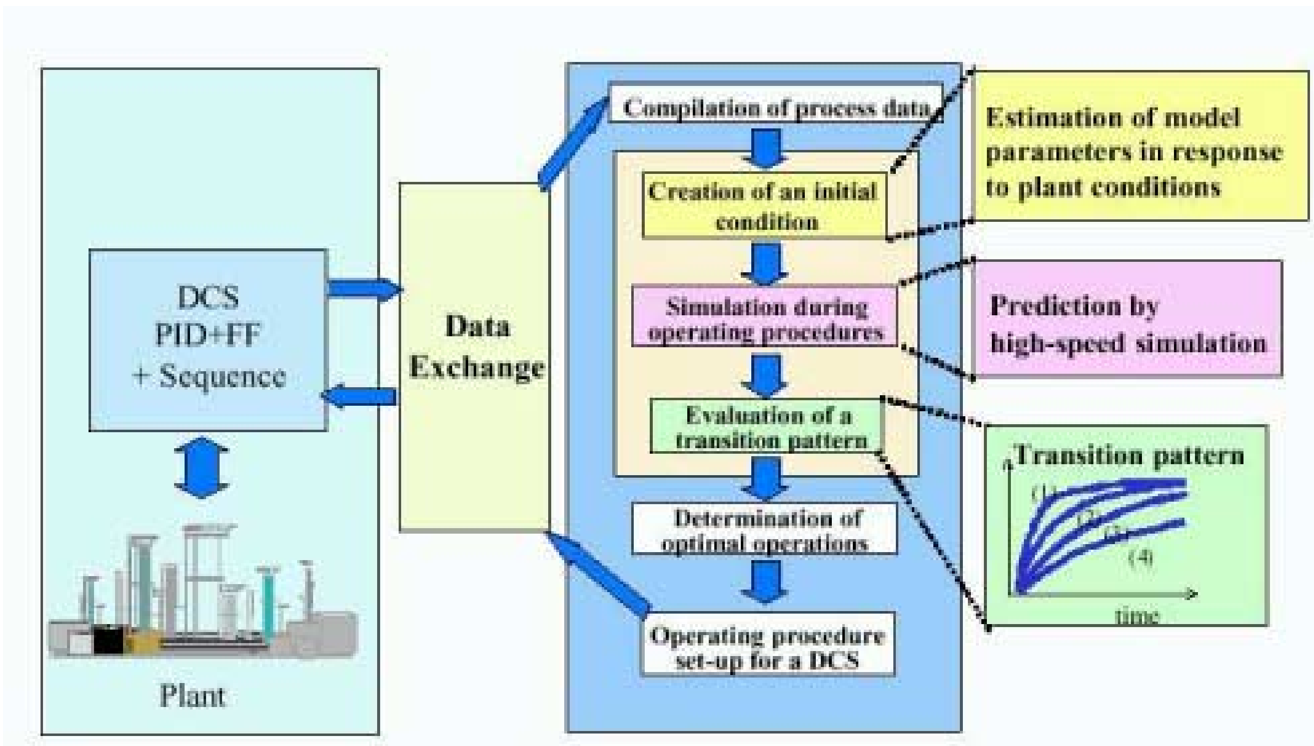
- Matlab Optimization Toolbox: **`QUADPROG`**
- Same feasibility issues as for LP
- Fast solvers available
- More in the next Lecture...

Lecture 11 - Optimal Program

- Grade change in process control
  - example
- QP optimization
- Flexible dynamics: input shaping, input trajectory
  - example
- Rocket, ascent
- Robotics

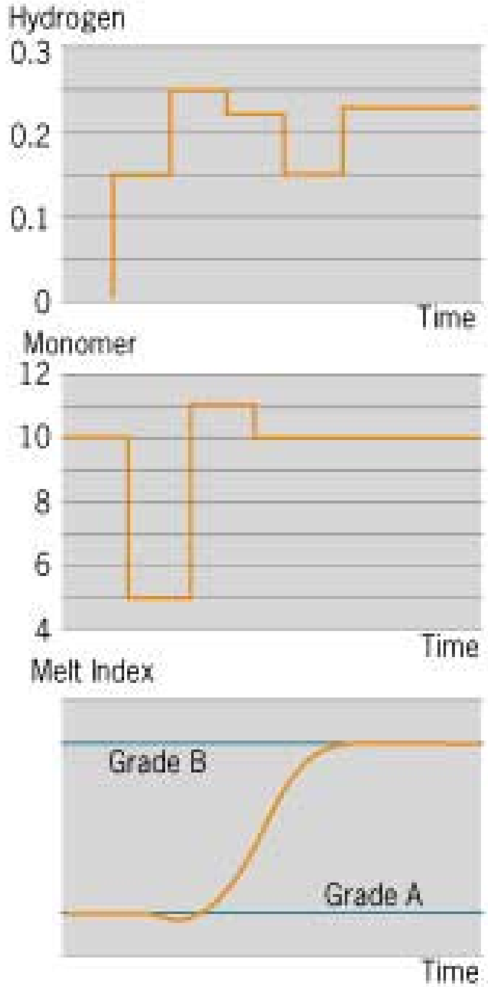
Optimization of process transitions

- Process plants manufacture different product varieties (grades)
- Need to optimize transitions from grade to grade



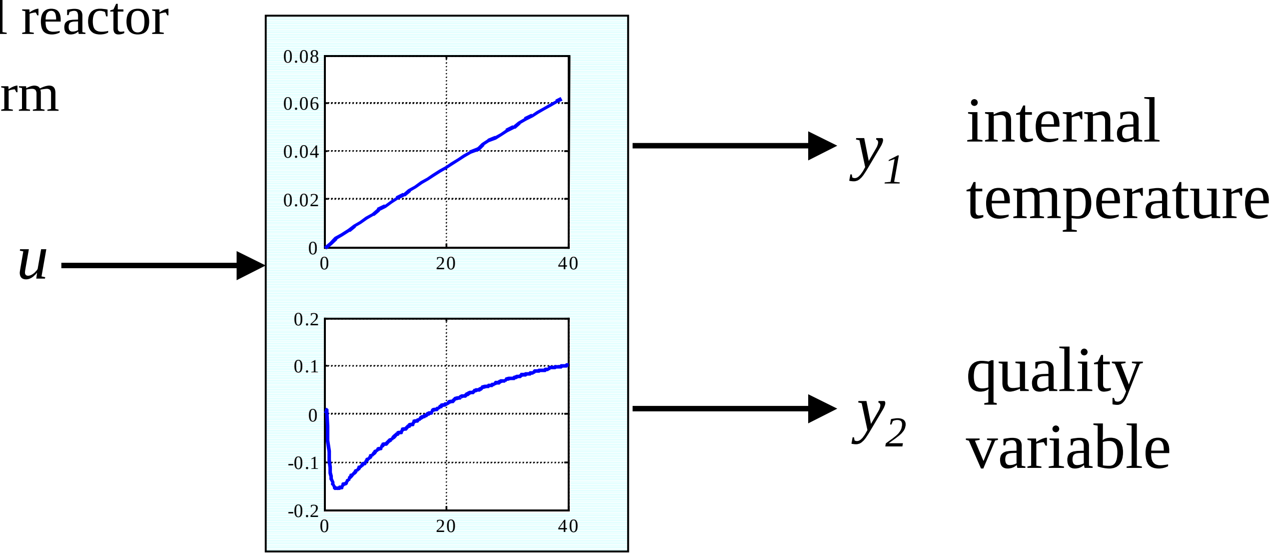
Product Grade Change

- The requirement: to change manufacture from grade A to grade B with the minimum off-spec production.
- The implementation: using detailed models of process and operating procedures.
- The results: optimum setpoint trajectories for key process controllers during the changeover, resulting in minimum lost revenue.



Grade change control example

- Simple process model:
  - chemical reactor
  - server farm



- The process is the initial steady state: $u = 0$; $y_1 = y_2 = 0$
- Need to transition, as quickly as possible, to other steady state: u = const; y_1= const; $y_2 = y_d$

Grade change control example

- Linear system model in the convolution form

$$y = h * u$$

- Quadratic-optimal control

$$\int \left(\left| y_2(t) - y_d \right|^2 + r \left| \dot{u}(t) \right|^2 \right) dt \to \min$$

- Equality constraint (process transitioning to the new grade)

$$\dot{y}_1(t) \equiv 0, \; y_2(t) \equiv y_d, \text{ for } T \le t \le T + T_f$$

- Inequality constraints
  - Control $\left| u(t) \right| \le u_*$
  - Temperature $\left| y_2(t) \right| \le d_*$

Grade change control example

- Sampled time: $t = k\tau$, (k=1,…,N);
  - Y is a $2N$ vector
  - H is a block-Toeplitz matrix

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_1 U \\ H_2 U \end{bmatrix} = HU$$

$$U = \begin{bmatrix} u(\tau) \\ \vdots \\ u(N\tau) \end{bmatrix}, Y_1 = \begin{bmatrix} y_1(\tau) \\ \vdots \\ y_1(N\tau) \end{bmatrix}, Y_2 = \ldots$$

$$H_{1,2}U = h_{1,2} * U$$

- Dynamics as an equality constraint:

$$HU - Y = 0$$

Grade change control example

- Quadratic-optimal control

optional

$$(Y_2 - Y_d)^T (Y_2 - Y_d) + rU^T D^T DU + w(Y_1 - Y_{d1})^T (Y_1 - Y_{d1}) \to \min$$

$$U^T D^T DU + Y_2^T Y_2 - 2Y_d^T Y_2 + \ldots \to \min$$

$$Y_d = y_d \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -1 \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

- Inequality constraints
  - Control $-u_* \le U \le u_*$
  - Temperature $0 \le Y_1 \le T_*$

Terminal constraint

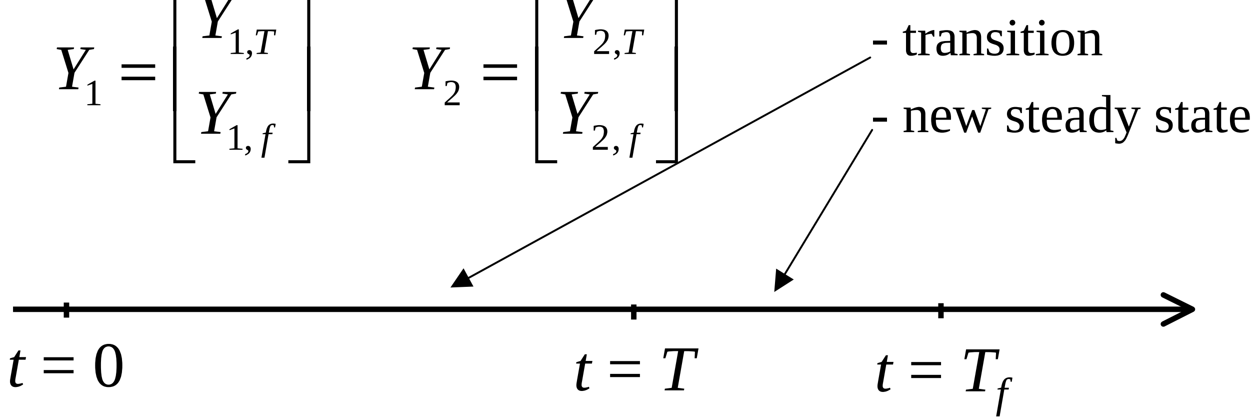
- Equality constraints (new grade steady state)

$$DY_{1,f} = 0$$ - steady in the end

$$Y_{2,f} = Y_{d,f}$$ - at target in the end

$$D = \begin{bmatrix} 1 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -1 \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

$$Y_1 = \begin{bmatrix} Y_{1,T} \\ Y_{1,f} \end{bmatrix} \qquad Y_2 = \begin{bmatrix} Y_{2,T} \\ Y_{2,f} \end{bmatrix}$$



Quadratic Programming

- QP Problem:

$$Ax \le b$$

$$Gx = h$$

$$J = \frac{1}{2} x^T H x + f^T x \to \min$$

- Matlab Optimization Toolbox: **QUADPROG**

Sim

QP Program for the grade change, no terminal constraint

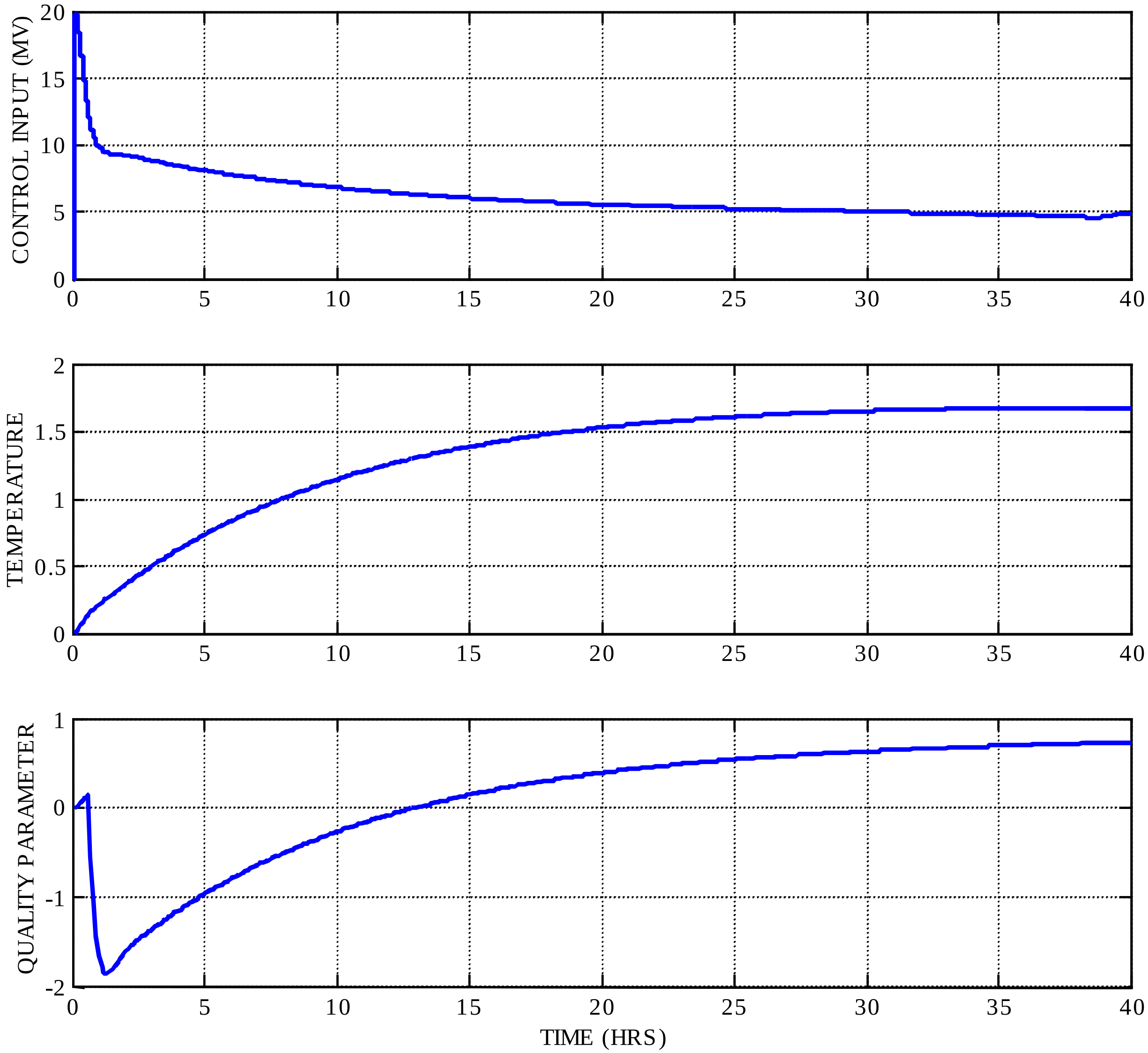
$y_d = 0.75$

$\tau = 0.1$

$r = 0.05$

$T_* = 2$

$u_* = 20$



Sim

QP Program for the grade change with a terminal constraint at $T = 8$

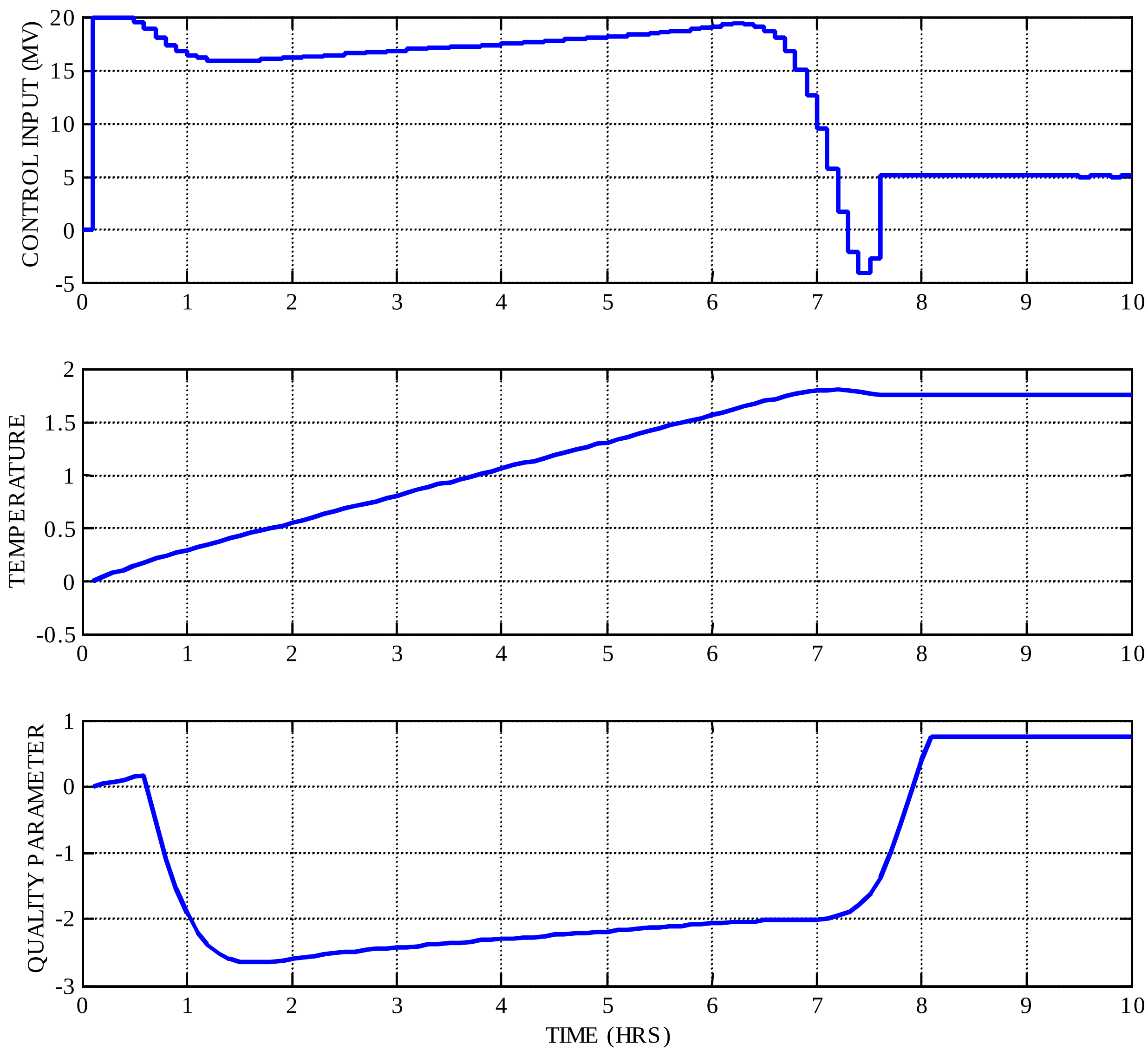
$y_d = 0.75$

$\tau = 0.1$

$r = 0.05$

$T_* = 2$

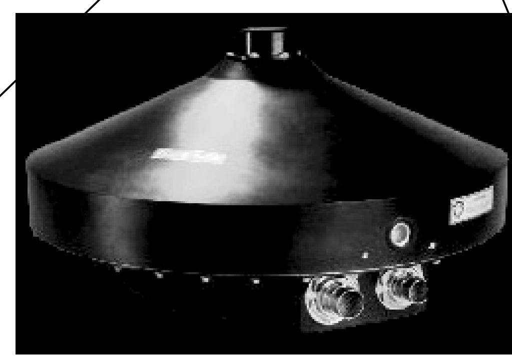
$u_* = 20$

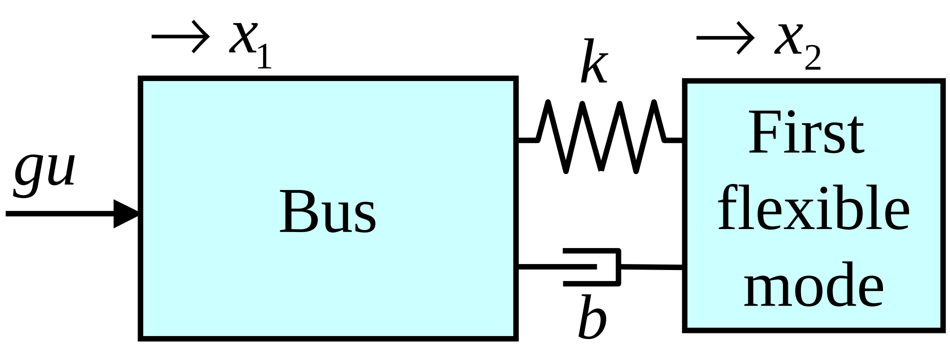


Flexible Satellite Slew Control

- Single flexible mode model
- Franklin, Section 9.2

$$J_1\ddot{x}_1 = -k(x_1 - x_2) - b(\dot{x}_1 - \dot{x}_2) + gu$$
$$J_2\ddot{x}_2 = k(x_1 - x_2) + b(\dot{x}_1 - \dot{x}_2)$$





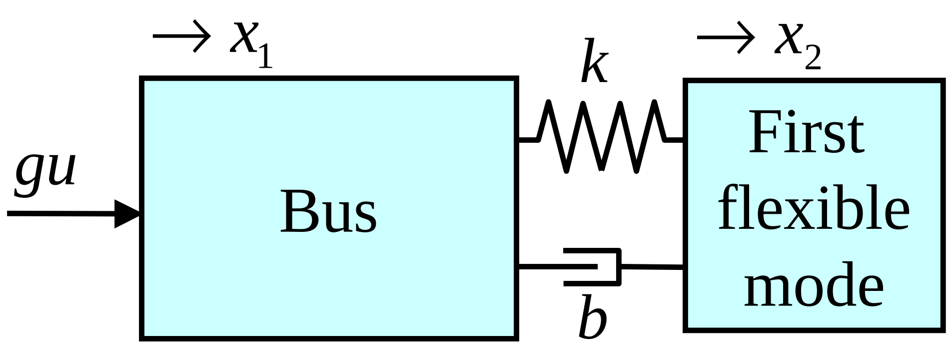
Flexible Satellite Slew Control

- Linear system model $\dot{x} = Ax + Bu$, $y = Cx$

$$x = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k/J_1 & -b/J_1 & k/J_1 & b/J_1 \\ 0 & 0 & 0 & 1 \\ k/J_2 & b/J_2 & -k/J_2 & -b/J_2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ g/J_1 \\ 0 \\ 0 \end{bmatrix}$$

$$y = \begin{bmatrix} x_1 \\ x_2 - x_1 \\ \dot{x}_1 \end{bmatrix}$$

- slew angle
- deformation
- slew rate



Flexible Satellite Slew Control

- Linear system model in the convolution form

$$y = h * u$$

- Quadratic-optimal control

$$\int \left| u(t) \right|^2 dt \to \min$$

- Equality constraint (system coming to at target slew angle)

$$y(t) \equiv y_d, \text{ for } T \le t \le T + T_f$$

- Inequality constraints
  - Control $\left| u(t) \right| \le 1$
  - Deformation $\left| y_2(t) \right| \le d_*$
  - Slew rate $\left| y_3(t) \right| \le v_*$

Flexible Satellite Slew Control

- Sampled time: $t = k\tau$, (k=1,…,N); Y is a 3N vector; H is a block-Toeplitz matrix

$$Y = HU$$

- Quadratic-optimal control

$$U^T U \rightarrow \min$$

- Equality constraint (system coming to at target slew angle)

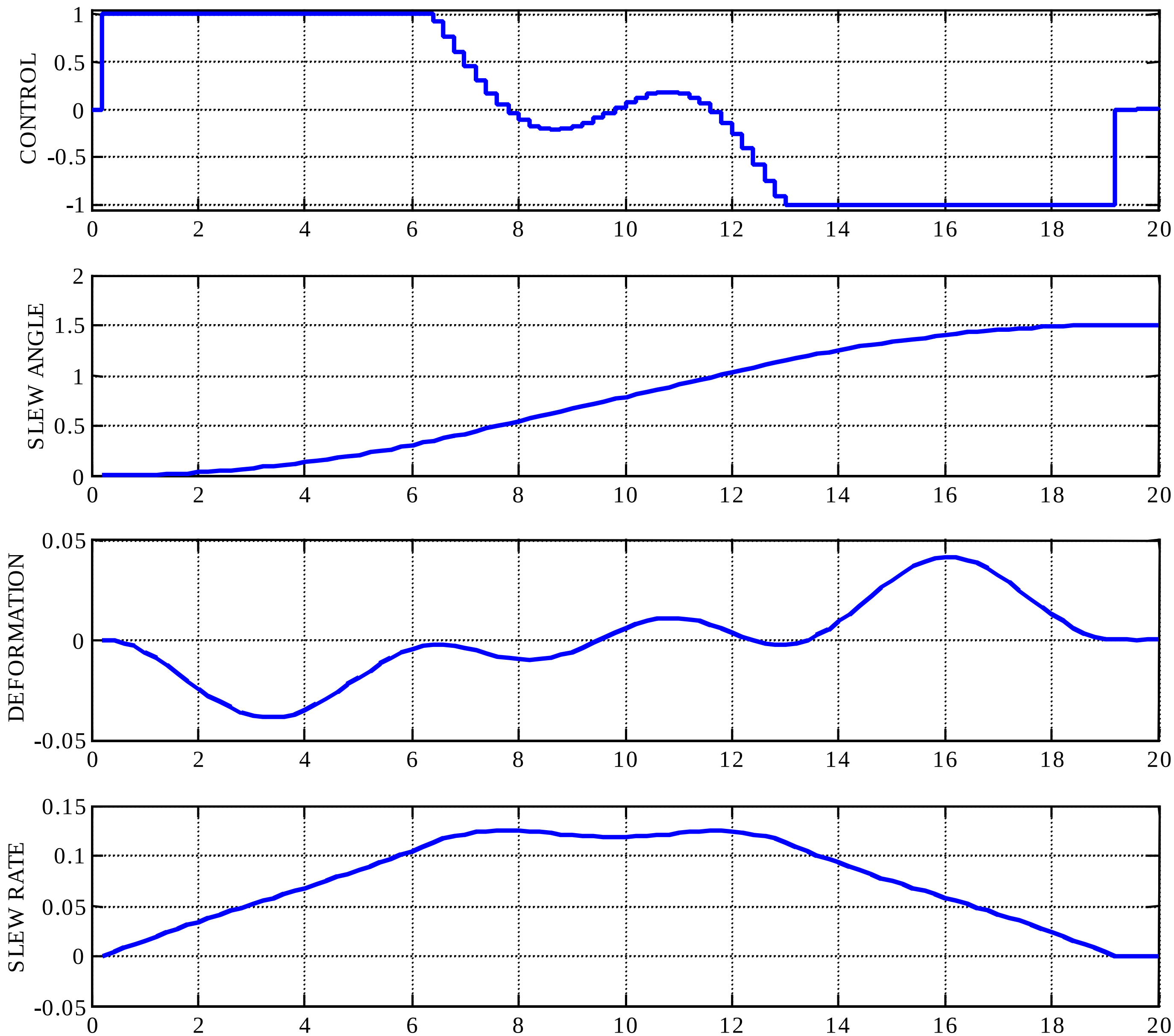
$$SY = Y_d$$

- Inequality constraints
  - Control $\quad -1 \le U \le 1$
  - Deformation $\quad d_* \le S_2 Y \le d_*$
  - Slew rate $\quad v_* \le S_3 Y \le v_*$

This is a QP problem

Sim

QP Program for the flexible satellite slew



$g = 0.02$

$J_1 = 1$

$J_2 = 0.1$

$k = 0.091$

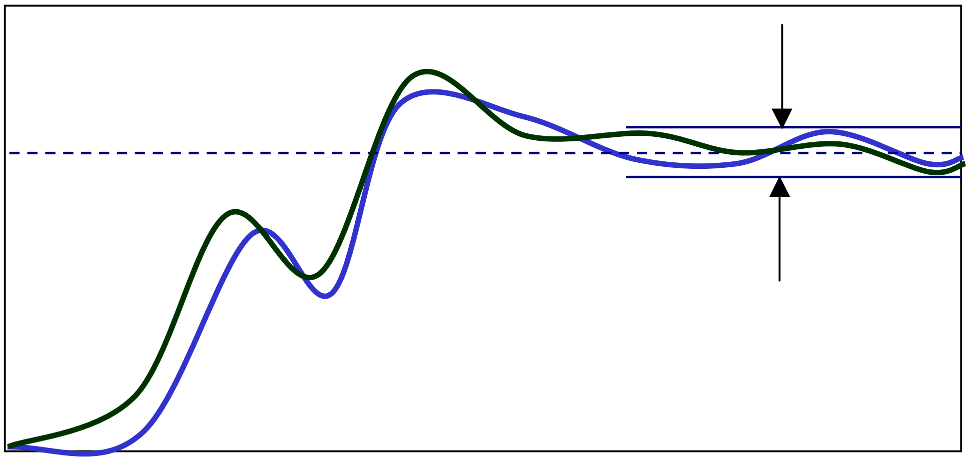
$b = 0.0036$

$d_* = 0.02$

$v_* = 0.2$

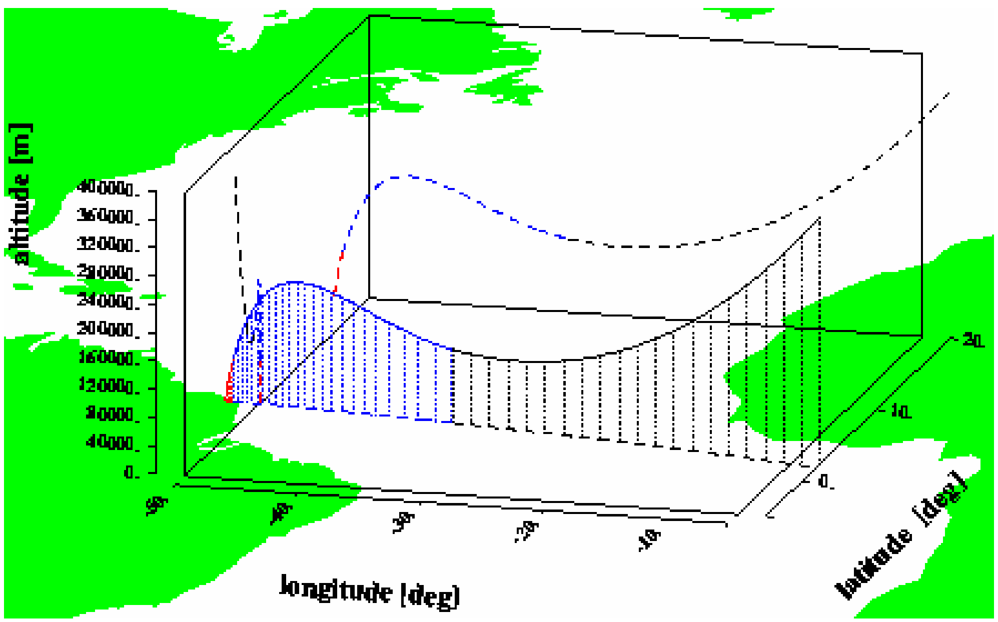
Robust design approach

- Replace exact terminal constraint by a given residual error
- Consider the system for several different values of parameters and group the results together
- As an optimality index, consider the average performance index or the worst residual error

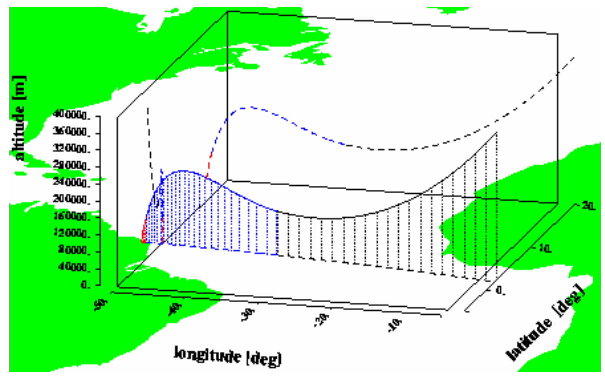


Ascend trajectory optimization

- Rocket launch vehicles
  - fuel (payload) optimality
  - orbital insertion constraint
  - flight envelope constraints
  - booster drop constraint



Ascend trajectory optimization



- Nonlinear constraint optimization problem
  - not QP, not LP
  - iterative optimization methods: Gradient, Newton, Levenberg-Marquardt, SQP, SSQP
  - can get results if supervised by a human
  - QP, LP are guaranteed always produce a solution if the problem is feasible - suitable for one-line use inside control loop

Mobile Robot Path Planning

$$F(\xi(\cdot),\eta(\cdot),t_f) \rightarrow \text{min}$$

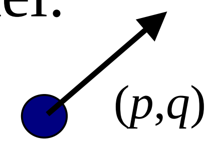
$$\ddot{\xi} = p, \;\; \|p\| \le p_{max},$$

$$\ddot{\eta} = q, \;\; \|q\| \le q_{max},$$

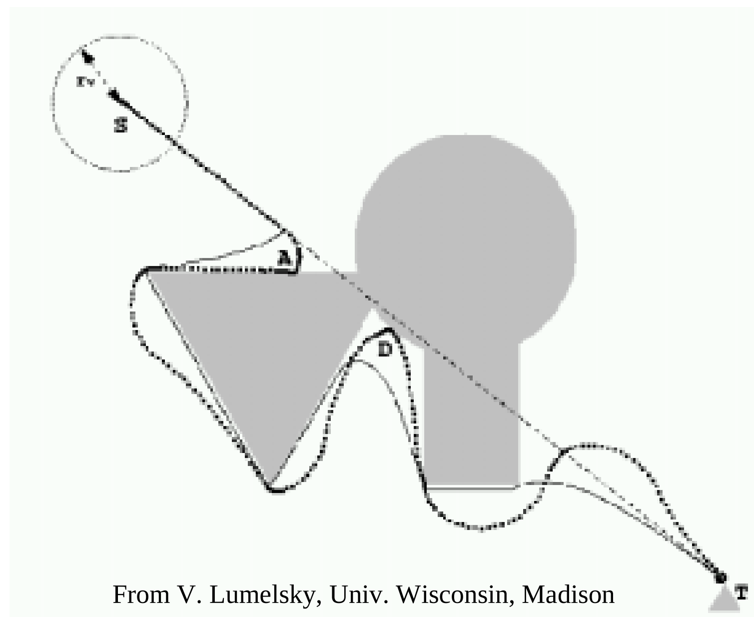
$$\xi(0) = \xi_0, \;\; \eta(0) = \eta_0, \;\; \dot{\xi}(0) = \dot{\xi}_0, \;\; \dot{\eta}(0) = \dot{\eta}_0,$$

$$\eta(t_f) = \eta(t_f) = \dot{\xi}(t_f) = \dot{\eta}(t_f) = 0$$

Point mass model:



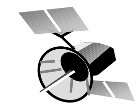
Constraint optimization problem of finding an optimal path

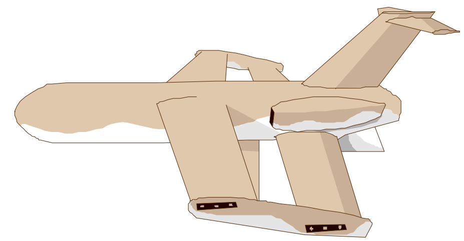


From V. Lumelsky, Univ. Wisconsin, Madison

Future Combat Systems (FCS)

- Ground and air robotics vehicles
- Potential application of robotics research
- Path planning and optimization are important

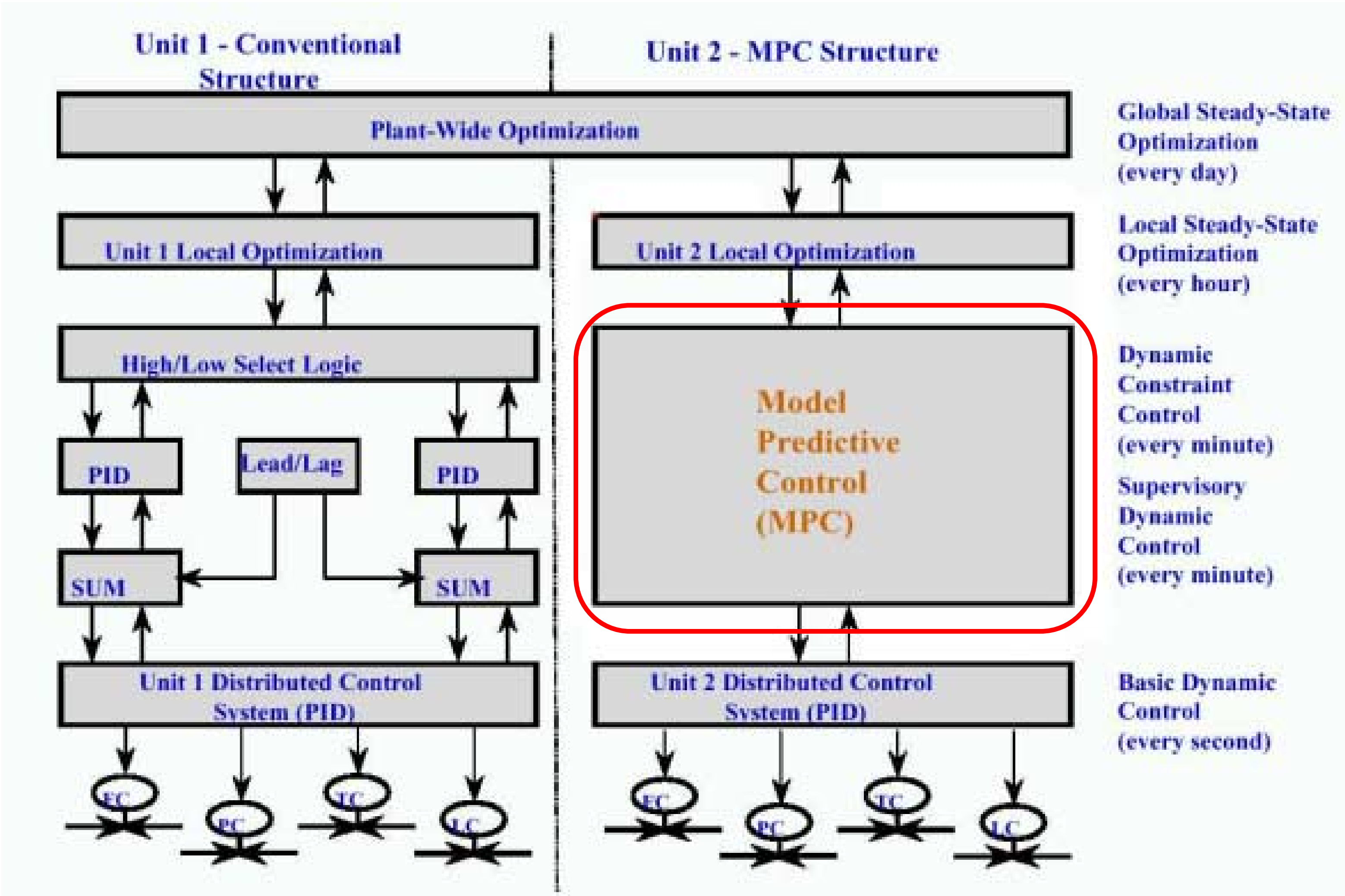




Lecture 12 - Model Predictive Control

- Prediction model
- Control optimization
- Receding horizon update
- Disturbance estimator - feedback
- IMC representation of MPC
- Resource:
  - Joe Qin, survey of industrial MPC algorithms
  - http://www.che.utexas.edu/~qin/cpcv/cpcv14.html

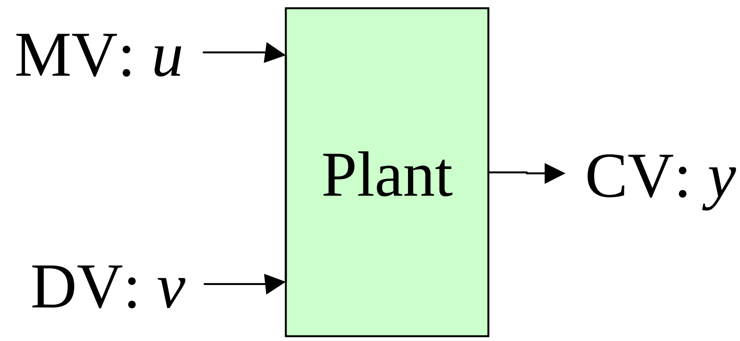
Control Hierarchy



Models for MPC

Plant structure:

- CV - controlled variables - *y*
- MV - manipulated variables - *u*
- DV - disturbance variables - *v*



- FSR - Finite Step Response model

$$y(t) = \sum_{k=1}^{N} S^U(k)\Delta u(t-k) + \sum_{k=1}^{N} S^D(k)\Delta v(t-k) + d$$

  – compact notation

$$y(t) = (s^U * \Delta u)(t) + (s^D * \Delta v)(t) + d$$

$$h = \Delta s;$$

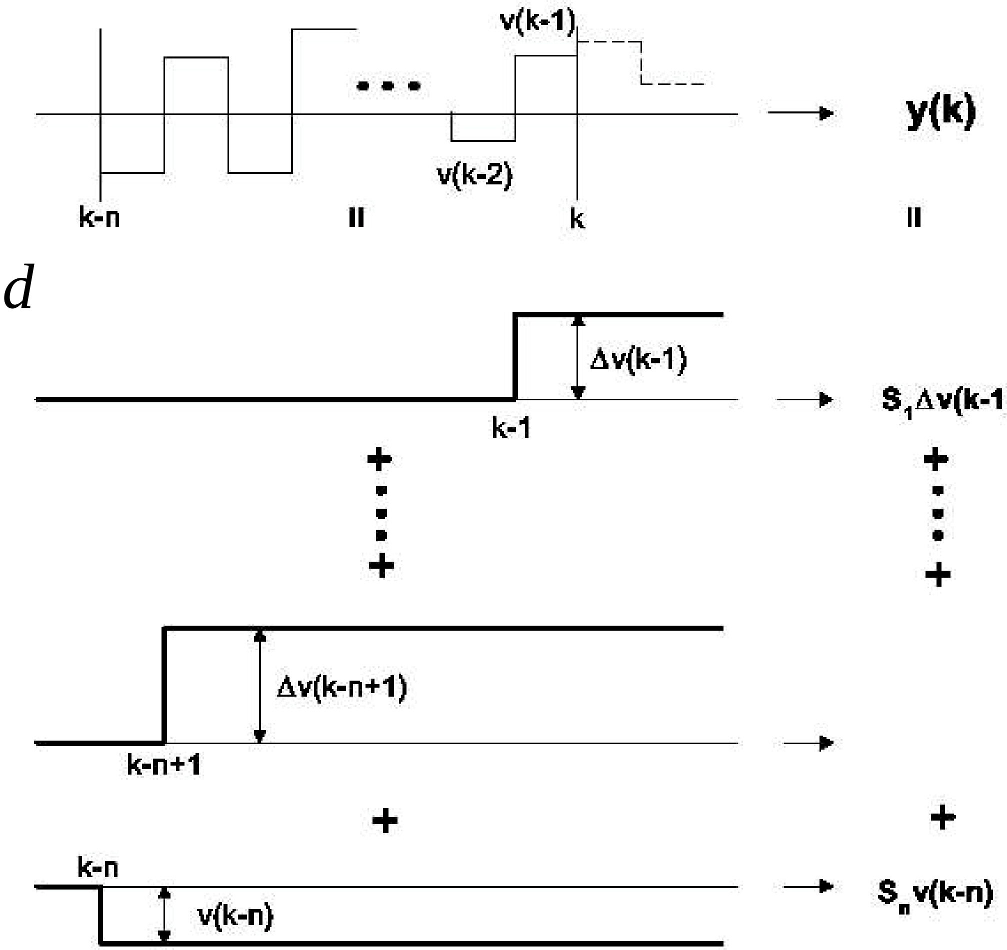
$$\Delta = 1 - z^{-1}$$

FSR Model

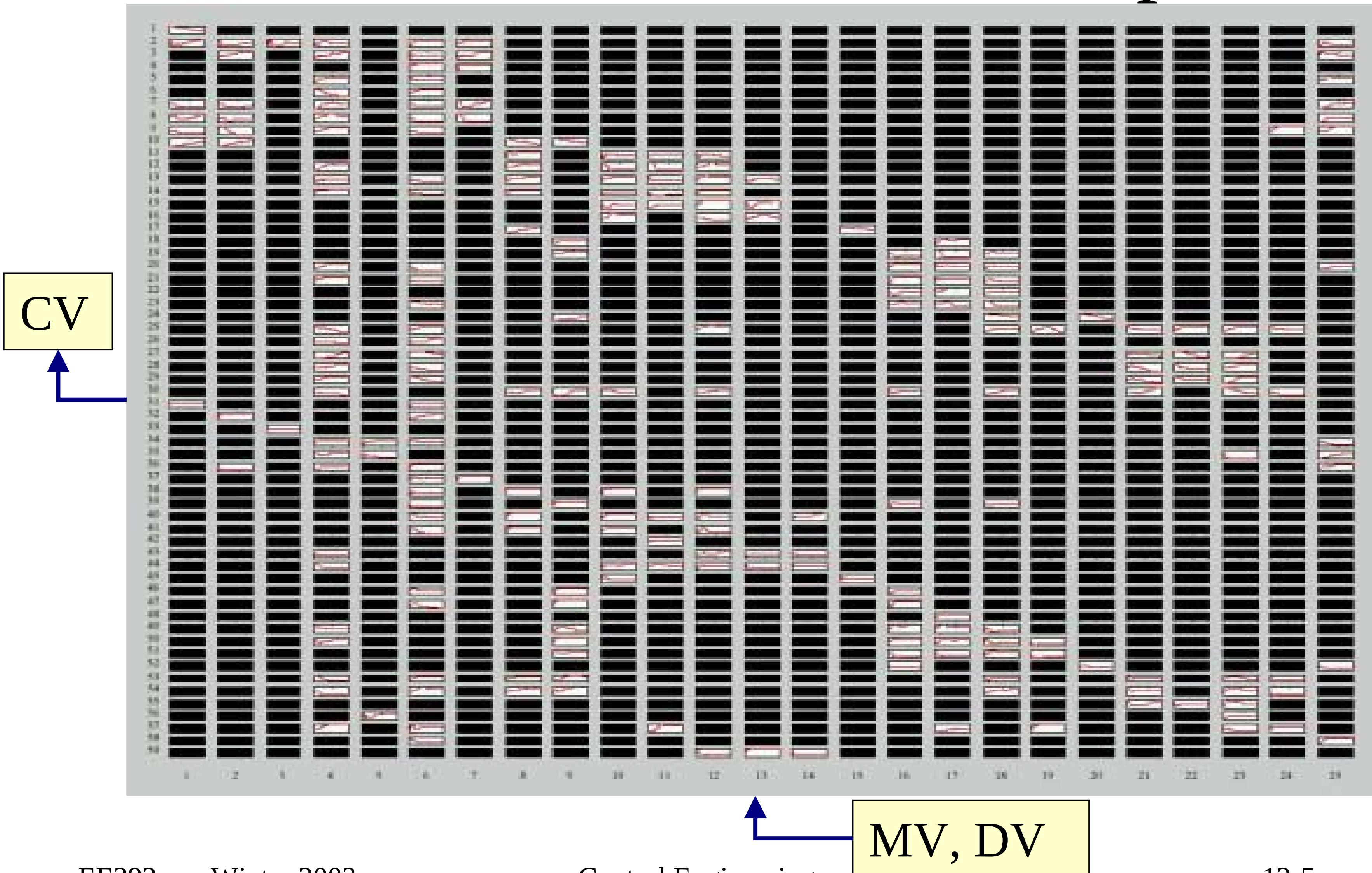
FSR model

$$y(t) = \sum_{k=1}^{n} S(k)\Delta v(t-k) + d$$

- Ignores anything that happened more than *n* steps in the past
- This is attributed to a constant disturbance *d*

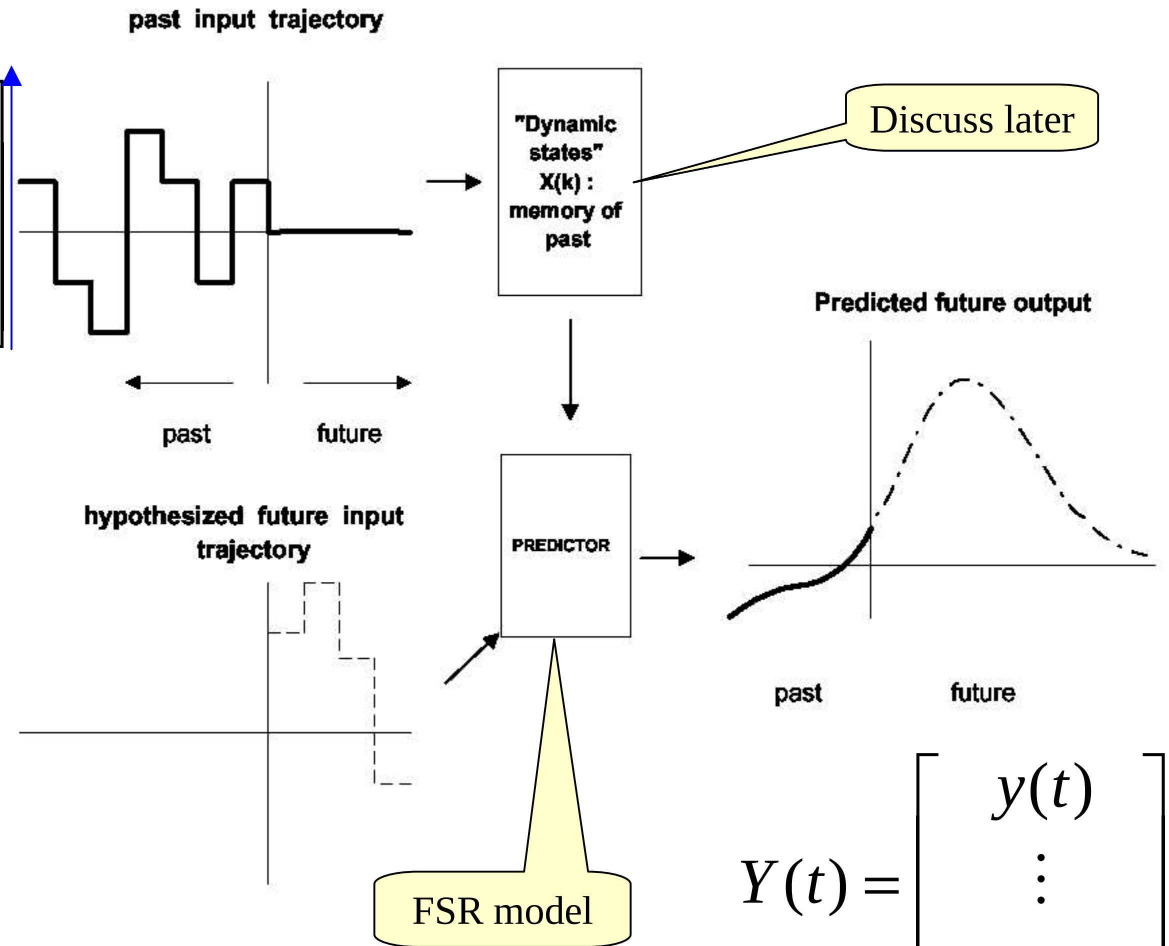


MPC Process Model Example



Prediction Model

$$\underline{U}(t) = \begin{bmatrix} \Delta u(t-1) \\ \vdots \\ \Delta u(t-N+1) \end{bmatrix}$$



$$U(t) = \begin{bmatrix} \Delta u(t) \\ \vdots \\ \Delta u(t+n) \end{bmatrix}$$

$$Y(t) = \begin{bmatrix} y(t) \\ \vdots \\ y(t+n) \end{bmatrix}$$

Prediction Model

$$y(t) = (s^U * \Delta u)(t) + (s^D * \Delta v)(t) + d$$

Future MV

Hankel matrix

Past MV

Hankel matrix

Past DV

Toeplitz matrix

Unmeasured disturbance

CV Prediction

$$Y(t) = \Psi U(t) + \Phi \cdot \underline{U}(t) + \Phi^D \underline{V}(t) + Dd$$

$$\Psi = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ S^U(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ S^U(n) & S^U(n-1) & \cdots & 0 \end{bmatrix} \qquad \Phi^D = \begin{bmatrix} S^D(1) & S^D(2) & \cdots & S^D(N) \\ S^D(2) & S^D(3) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ S^D(N) & 0 & \cdots & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Toeplitz matrix

Hankel matrix

Future impact of the disturbance

Optimization of future inputs

$$Y(t) = \Psi U(t) + \underbrace{\Phi \cdot \underline{U}(t) + \Phi^D \underline{V}(t) + Dd}_{Y^*(t)}$$

- Optimization problem

$$J = \left(Y(t) - Y_d(t)\right)^T Q \left(Y(t) - Y_d(t)\right) + U^T(t) R U(t) \to \min$$

$$Y_d(t) = \begin{bmatrix} y_d(t) \\ \vdots \\ y_d(t+N) \end{bmatrix}$$

$$Q = \begin{bmatrix} Q^y & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q^y \end{bmatrix}, R = \begin{bmatrix} R^u & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R^u \end{bmatrix}$$

Optimization constraints

- MV constraints

$$-\Delta u_{\max} \le \Delta u(t) \le \Delta u_{\max}$$

$$u_{\min} \le u(t) \le u_{\max}$$

$$\Longrightarrow$$

$$-\Delta u_{\max} \le U(t) \le \Delta u_{\max}$$

$$u_{\min} \le \Sigma U(t) - C \le u_{\max}$$

$$u(t+k) = u(t-1) + \sum_{j=1}^{k} \Delta u(t)$$

- CV constraints

$$y_{\min}(t) \le y(t) \le y_{\max}(t) \quad \Longrightarrow \quad Y_{\min} \le Y(t) \le Y_{\max}$$

- Terminal constraint:

$$y(t+k) = y_d; \Delta u(t+k) = 0 \ \text{ for } \ k \ge p$$

QP solution

- QP Problem:

$$Ax \le b$$

$$A_{eq}x = b_{eq}$$

$$J = \frac{1}{2}x^T Qx + f^T x \rightarrow \min$$

$$x = \begin{bmatrix} U(t) \\ Y(t) \end{bmatrix} \quad \text{Predicted MVs, CVs}$$

- Standard QP codes are available

Receding horizon control

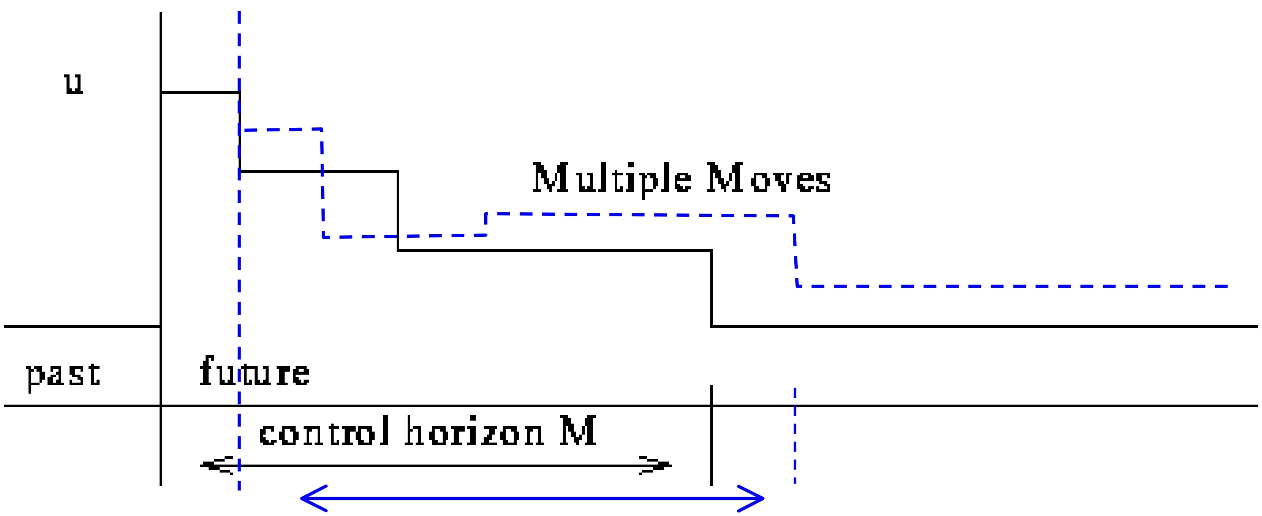
- Optimization problem solution at step t :

$$J(U) \to \min \qquad \Rightarrow \qquad U = U_{OPT}(t)$$

- Use the first computed control value only

$$u(t) = [1 \quad 0 \quad \ldots \quad 0] \cdot U_{OPT}(t)$$

- Repeat at each t



Control dynamics

- System dynamics as an equality constraint in optimization

$$Y(t) = \Psi U(t) + Y^*(t)$$

$$Y^*(t) = \left[\Phi \quad \Phi^D\right] \cdot X(t) + d(t)$$

- Update of the system state

$$X(t+1) = AX(t) + B\Delta u(t) + B^D \Delta v(t)$$

- Optimization problem solution at step t :

$$J(U; Y(U)) \rightarrow \min \qquad \Rightarrow \qquad U = U_{OPT}(t)$$

- Use the first of the computed control values

$$u(t) = [1 \quad 0 \quad \ldots \quad 0] \cdot U_{OPT}(t)$$

State update and estimation

- State update - shift register

$\Delta u(t)$ $\qquad$ $\Delta v(t)$

$$X(t) = \begin{bmatrix} \underline{U}(t) \\ \underline{V}(t) \end{bmatrix} \qquad \underline{U}(t) = \begin{bmatrix} \Delta u(t-1) \\ \vdots \\ \Delta u(t-N+1) \end{bmatrix} \qquad \underline{V}(t) = \begin{bmatrix} \Delta v(t-1) \\ \vdots \\ \Delta v(t-N+1) \end{bmatrix}$$

- Disturbance estimator (feedback)

$$d(t+1) = d(t) + \left(y_m(t) - y(t)\right)$$

Unmeasured disturbance

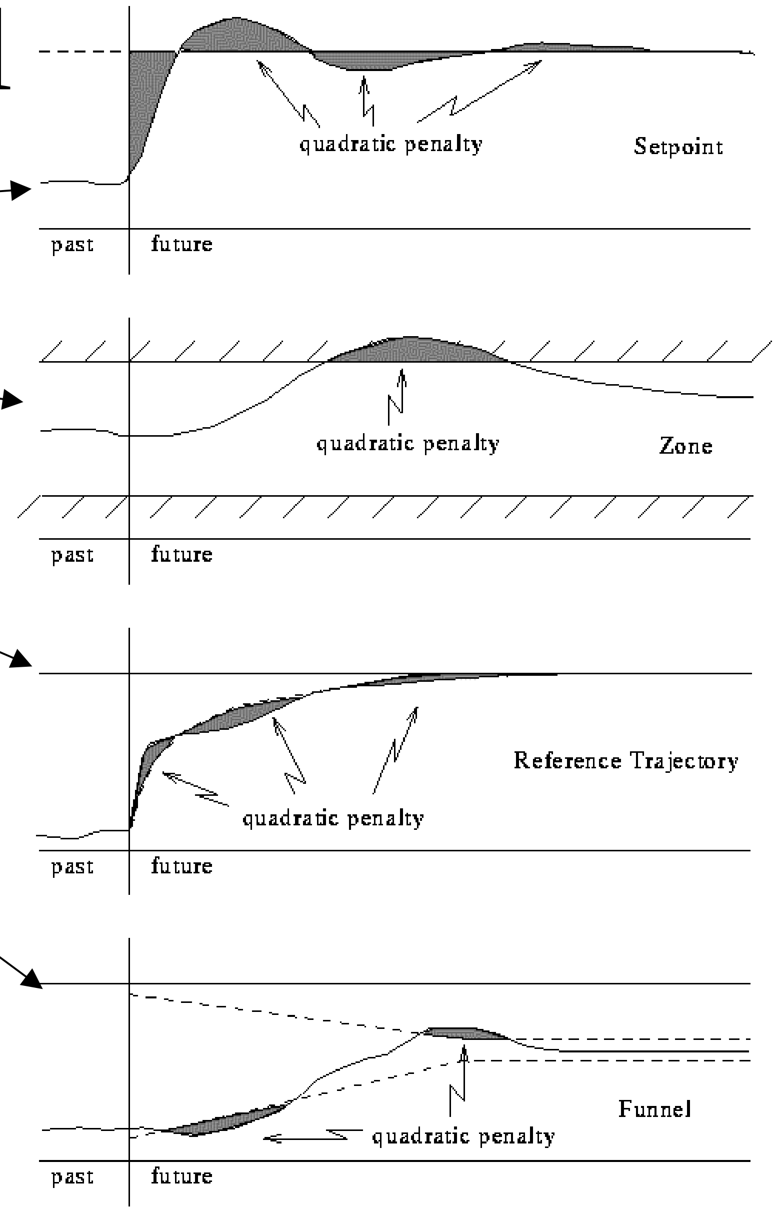
Actually measured CV

CV Prediction based on the current state X(t)

- Integrator feedback

Optimization detail

- Setpoint
- Zone
- Trajectory
- Funnels



- Soft constraints (quadratic penalties) and hard constraints for MV, CV
- Regularization
  - penalty
  - singular value thresholding

Advantages and Conveniences

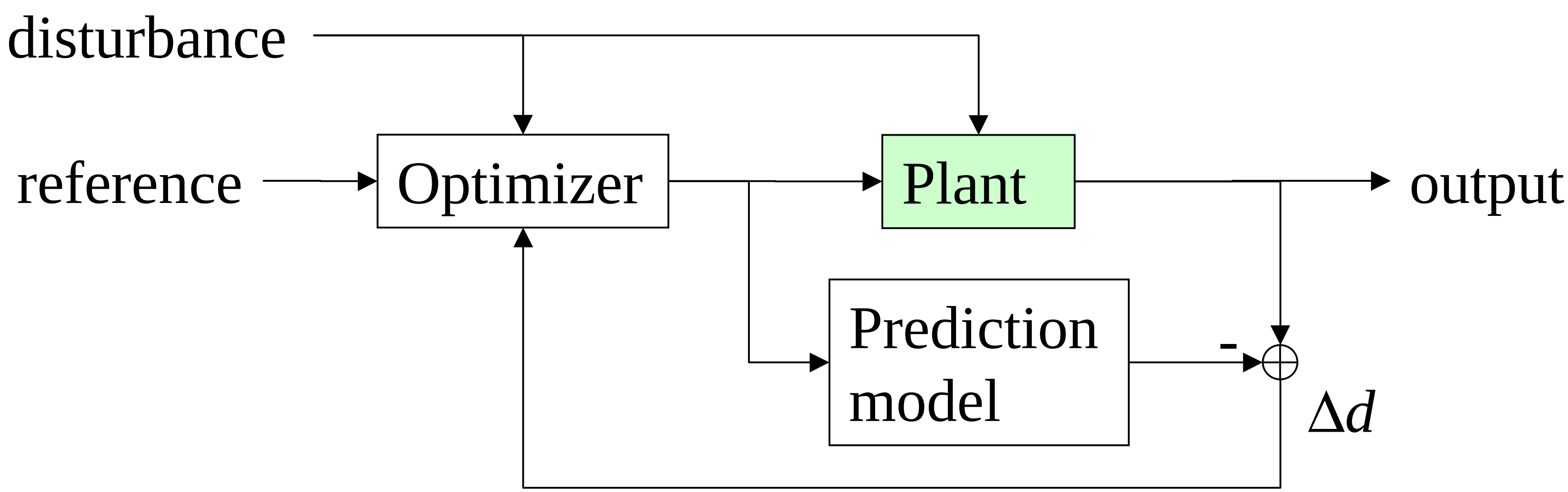
- Industrial strength products that can be used for a broad range of applications
- Flexibility to plant size, automated setup
- Based on step response/impulse response model
- On the fly reconfiguration if plant is changing
  - MV, CV, DV channels taken off control / returned into MPC
  - measurement problems, actuator failures
- Systematic handling of multi-rate measurements and missed measurement points
  - do not update d if no data

Technical detail

- Tuning of MPC feedback control performance is an issue.
  - Works in practice, without formal analysis
  - Theory requires
    - Large (infinite) prediction horizon
    - Terminal constraint
- Additional tricks for
  - a separate static optimization step
  - integrating and unstable dynamics
  - active constraints
  - regularization
  - shape functions for control
  - different control horizon and prediction horizon
  - ...

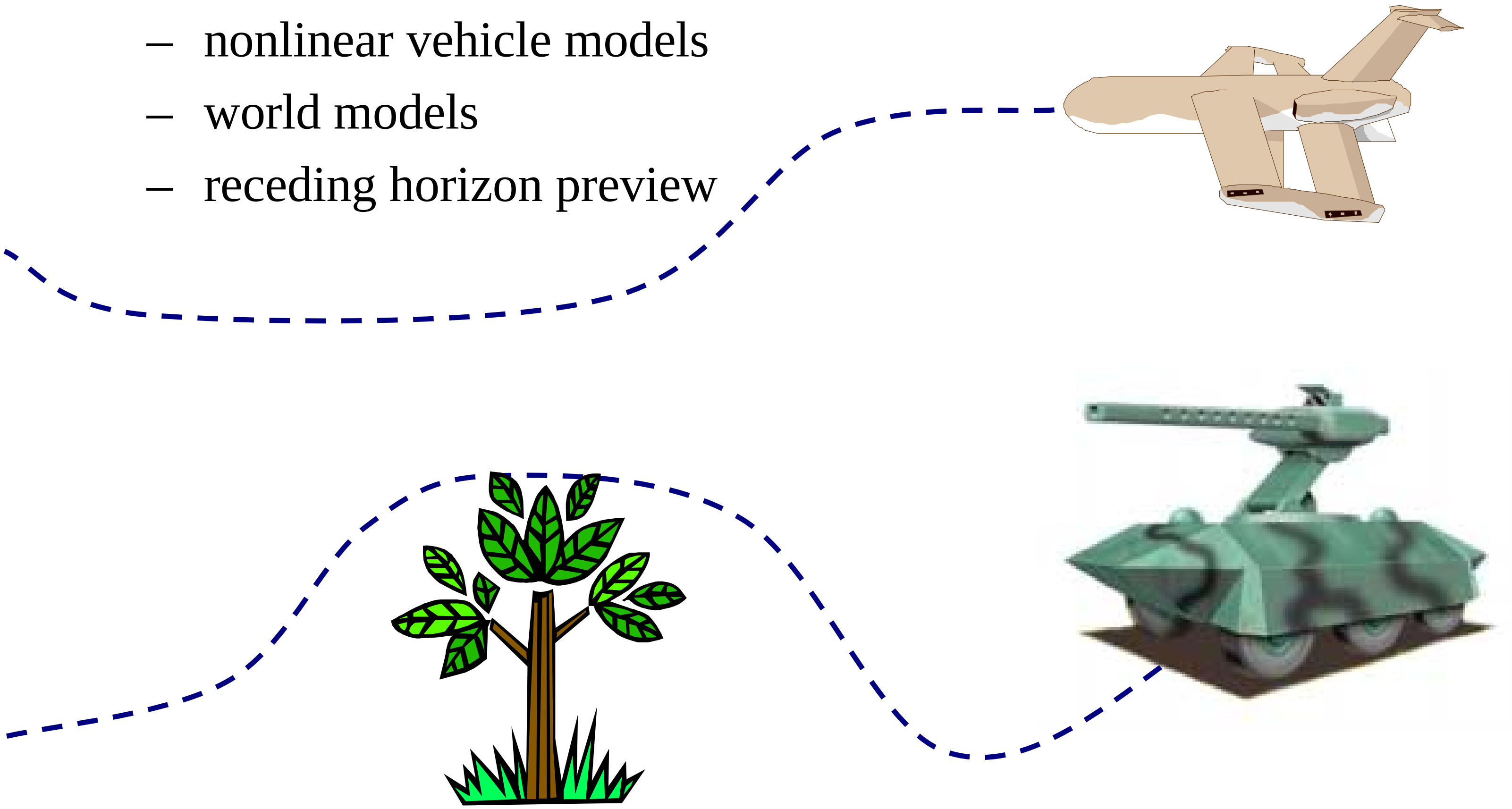
MPC as IMC

- MPC is a special case of IMC
- Closed-loop dynamics (filter dynamics)
  - integrator - in disturbance estimator
  - N poles z=0 - in the FSR model update



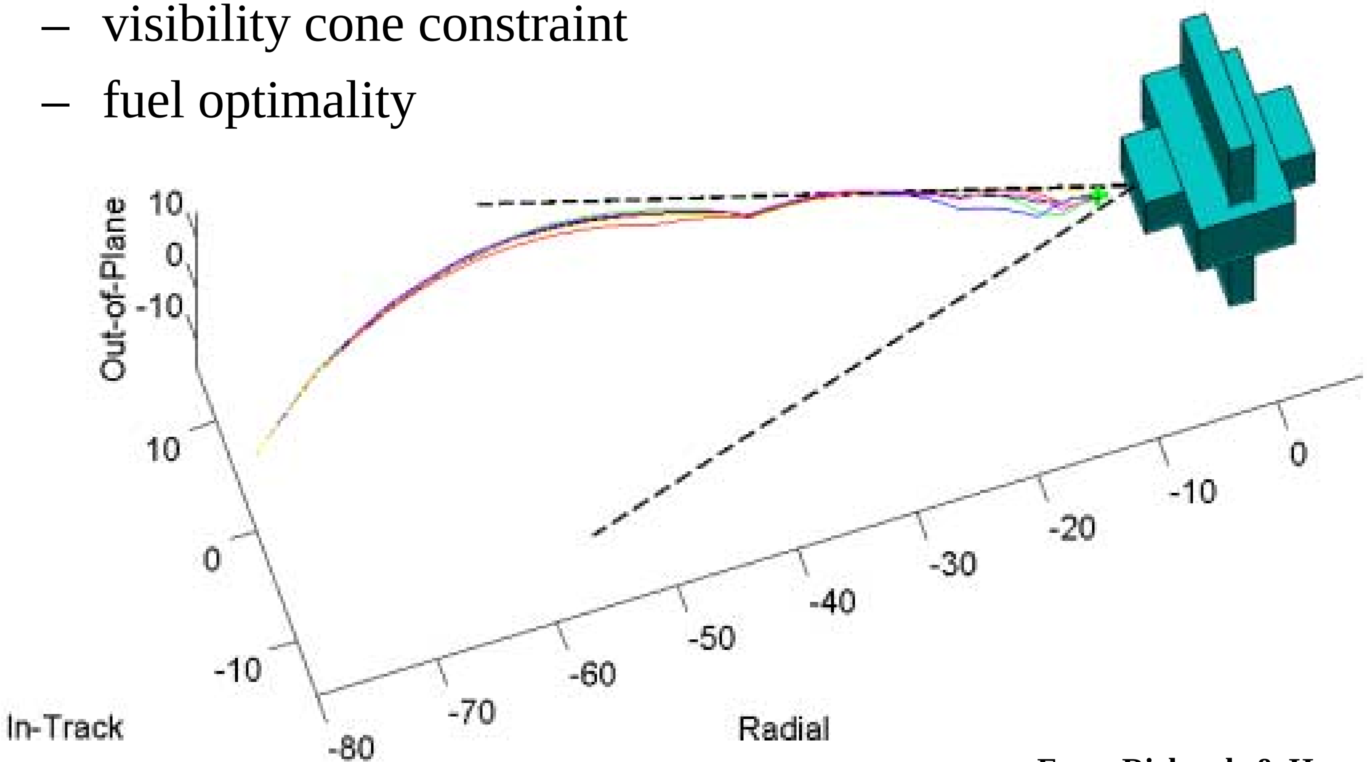
Emerging MPC applications

- Vehicle path planning and control
  - nonlinear vehicle models
  - world models
  - receding horizon preview



Emerging MPC applications

- Spacecraft rendezvous with space station
  - visibility cone constraint
  - fuel optimality



From Richards & How, MIT

Emerging MPC applications

- Nonlinear plants
  - just need a computable model (simulation)
- Hybrid plants
  - combination of dynamics and discrete mode change
- Engine control
- Large scale operation control problems
  - operations management
  - campaign control

Lecture 13 - Handling Nonlinearity

- Nonlinearity issues in control practice
- Setpoint scheduling/feedforward
  - path planning replay - linear interpolation
- Nonlinear maps
  - B-splines
  - Multivariable interpolation: polynomials/splines/RBF
  - Neural Networks
  - Fuzzy logic
- Gain scheduling
- Local modeling

Nonlinearity in control practice

Here are the nonlinearities we already looked into

- Constraints - saturation in control
  - anti-windup in PID control
  - MPC handles the constraints
- Control program, path planning
- Static optimization
- Nonlinear dynamics
  - dynamic inversion
  - nonlinear IMC
  - nonlinear MPC

One additional nonlinearity in this lecture

- Controller gain scheduling

Dealing with nonlinear functions

- Analytical expressions
  - models are given by analytical formulas, computable as required
  - rarely sufficient in practice
- Models are computable off line
  - pre-compute simple approximation
  - on-line approximation
- Models contain data identified in the experiments
  - nonlinear maps
  - interpolation or look-up tables
- Advanced approximation methods
  - neural networks

Path planning

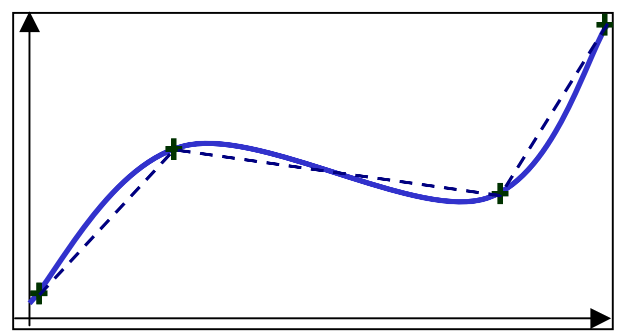
- Real-time replay of a pre-computed reference trajectory $y_d(t)$ or feedforward $v(t)$
- Reproduce a nonlinear function $y_d(t)$ in a control system

$$Y = \begin{bmatrix} Y_1 = y_d(\theta_1) \\ Y_2 = y_d(\theta_2) \\ \vdots \\ Y_n = y_d(\theta_n) \end{bmatrix}, \quad \Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

Code:

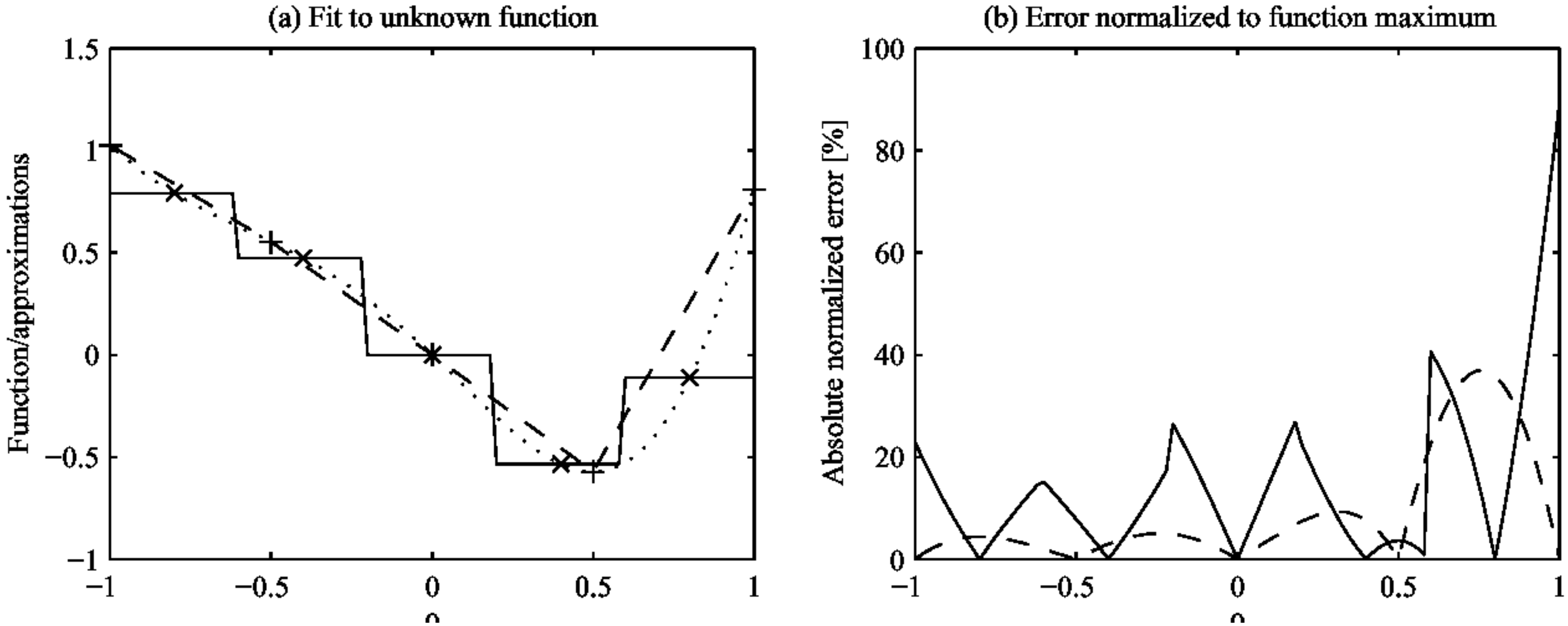
1. Find j, such that $\theta_j \le t \le \theta_{j+1}$
2. Compute

$$y_d(t) = Y_j \frac{\theta_{j+1} - t}{\theta_{j+1} - \theta_j} + Y_{j+1} \frac{t - \theta_j}{\theta_{j+1} - \theta_j}$$



Linear interpolation vs. table look-up

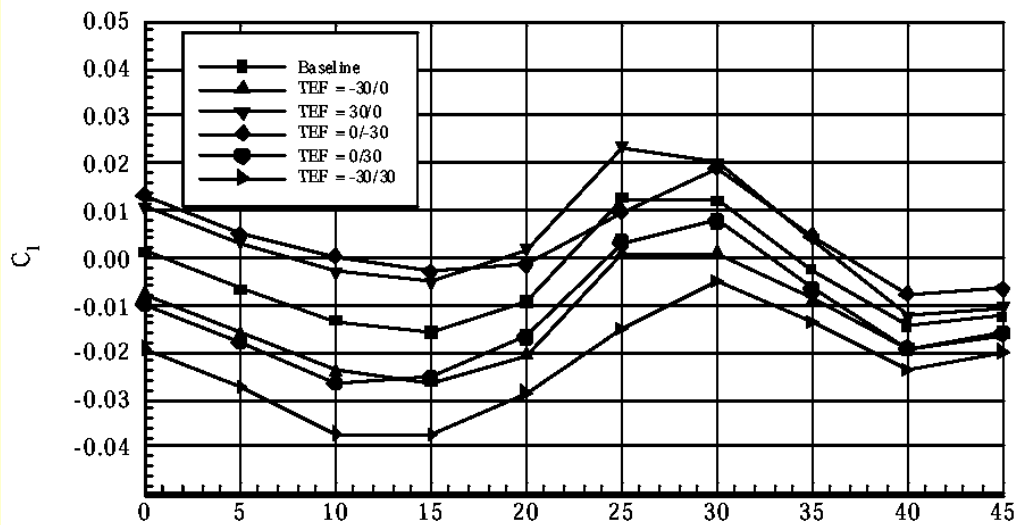
- linear interpolation is more accurate
- requires less data storage
- simple computation



Empirical models

- Aerospace - most developed nonlinear approaches
  - automotive and process control have second place
- Aerodynamic tables
- Engine maps
  - jet turbines
  - automotive
- Process maps, e.g., in semiconductor manufacturing
- Empirical map for a attenuation vs. temperature in an optical fiber

Example

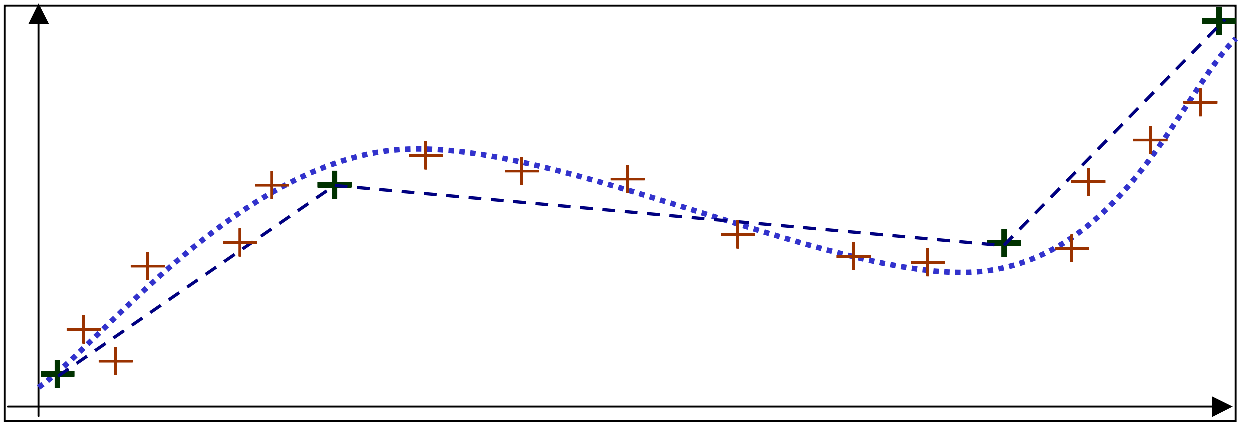


Rolling-Moment Coefficient as a Function of Angle of Attack with TEF Deflections, $\beta = 10°$.

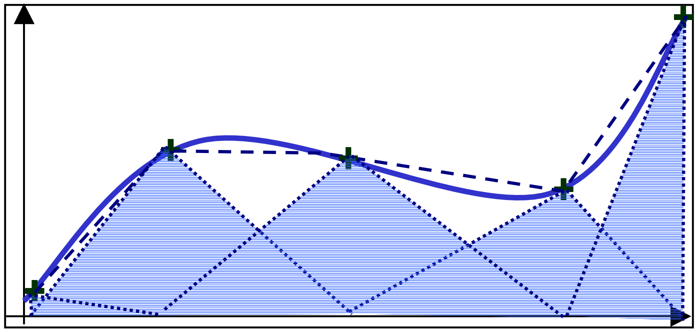
TEF=Trailing Edge Flap

Approximation

- Interpolation:
  – compute function that will provide given values Y_j in the nodes θ_j
  – not concerned with accuracy in-between the nodes
- Approximation
  – compute function that closely corresponds to given data, possibly with some error
  – might provide better accuracy throughout

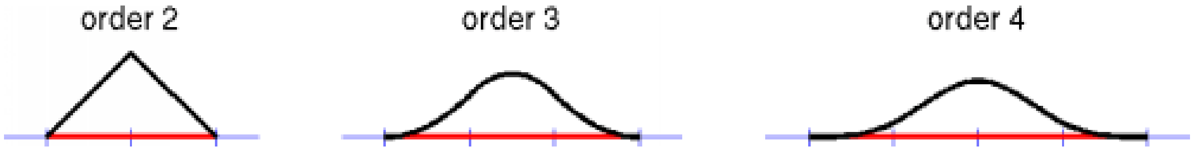


B-spline interpolation



- 1st-order
  - look-up table, nearest neighbor
- 2nd-order
  - linear interpolation

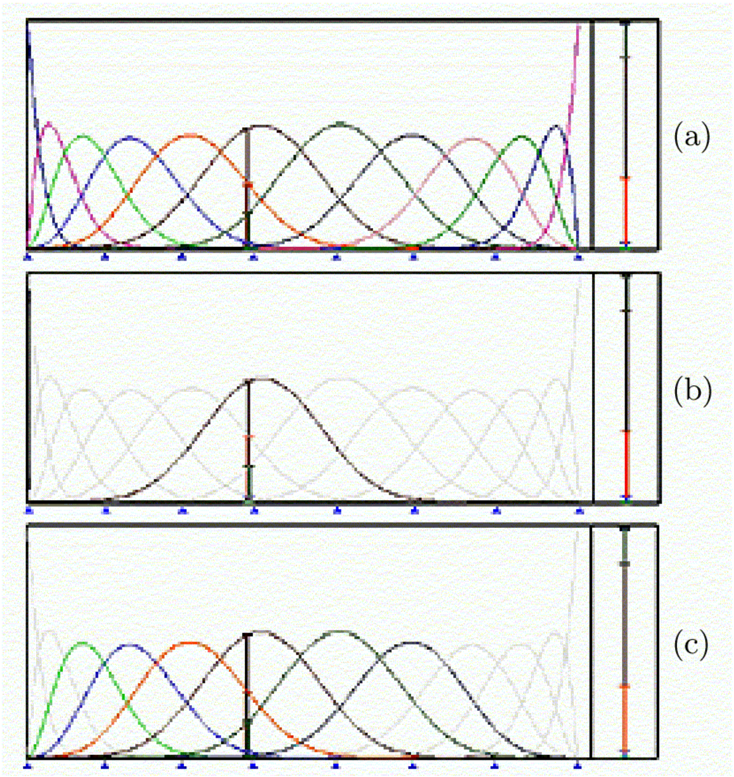
$$y_d(t) = \sum_j Y_j B_j(t)$$



- n-th order:
  - Piece-wise *n*-th order polynomials, matched *n*-2 derivatives
  - zero outside a local support interval
  - support interval extends to *n* nearest neighbors

B-splines

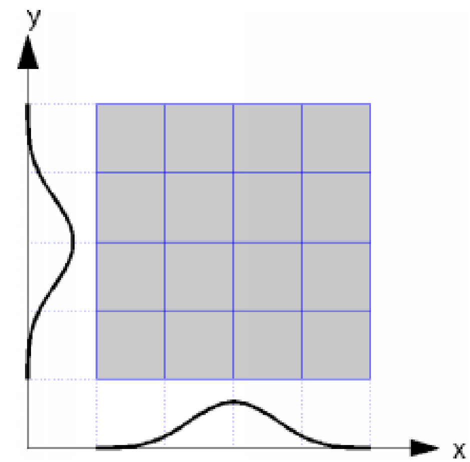
- Accurate interpolation of smooth functions with relative few nodes
- For 1-D function the gain from using high-order B-splines is not worth an added complexity
- Introduced and developed in CAD for 2-D and 3-D curve and surface data
- Are used for defining multidimensional nonlinear maps

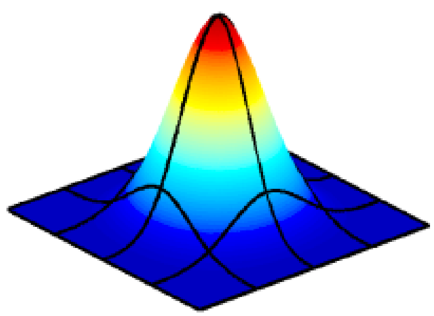


Multivariable B-splines

- Regular grid in multiple variables
- Tensor product B-splines
- Used as a basis of finite-element models

$$y(u,v) = \sum_{j,k} w_{j,k} B_j(u) B_k(v)$$





Linear regression for nonlinear map

- Linear regression

$$y(\overline{x}) = \sum_j \theta_j \varphi_j(\overline{x}) = \theta^T \cdot \phi(\overline{x}) \qquad \overline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- Multidimensional B-splines
- Multivariate polynomials

$$\varphi_j(x_1, \ldots, x_n) = (x_1)^{k_1} \cdot \ldots \cdot (x_n)^{k_n}$$

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 (x_1)^2 + \theta_4 x_1 x_2 + \ldots$$

- RBF - Radial Basis Functions

$$\varphi_j(\overline{x}) = R\left(\left\|\overline{x} - \overline{c}_j\right\|\right) = e^{-a\left\|\overline{x} - \overline{c}_j\right\|^2}$$

Linear regression approximation

- Nonlinear map data
  - available at scattered nodal points

$$Y = \left[y^{(1)} \quad \dots \quad y^{(N)} \right]$$

$$\bar{x}^{(1)} \quad \dots \quad \bar{x}^{(N)}$$

- Linear regression map

$$Y = \theta^T \cdot \left[\phi(\bar{x}^{(1)}) \quad \dots \quad \phi(\bar{x}^{(N)}) \right] = \theta^T \Phi$$

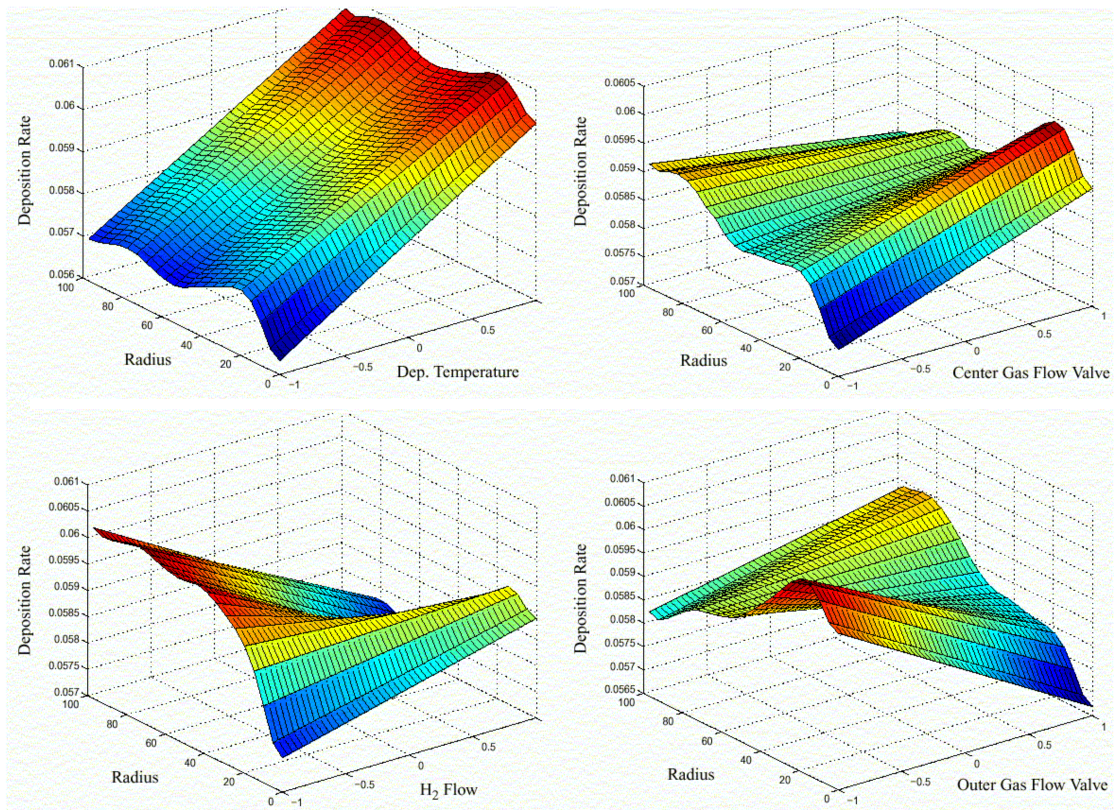
- Linear regression approximation
  - regularized least square estimate of the weight vector

$$\hat{\theta} = \left(\Phi \Phi^T + rI \right)^{-1} \Phi Y^T$$

- Works just the same for vector-valued data!

Nonlinear map example - Epi

- Epitaxial growth (semiconductor process)
  - process map for run-to-run control



Linear regression for Epi map

- Linear regression model for epitaxial grouth

$$y = c_0 x_1 p_1(x_2) + c_1(1 - x_1) p_2(x_2)$$

$$p_1 = w_0 + w_1 x_2 + w_3 (x_2)^2 + w_4 (x_2)^3$$

$$c_0 x_1 p_1 = \underbrace{w_0 c_0}_{\theta_1} x_1 + \underbrace{w_1 c_0}_{\theta_2} x_1 x_2 + \underbrace{w_3 c_0}_{\theta_3} x_1 (x_2)^2 + \underbrace{w_4 c_0}_{\theta_4} x_1 (x_2)^3$$

$$c_1(1 - x_1) p_2(x_2) =$$

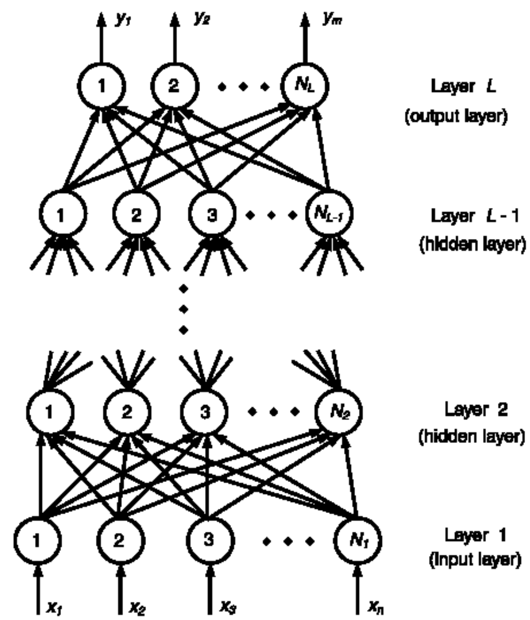
$$\underbrace{v_0 c_1}_{\theta_5}(1 - x_1) + \underbrace{v_1 c_1}_{\theta_6}(1 - x_1) x_2 + \underbrace{v_3 c_0}_{\theta_7}(1 - x_1)(x_2)^2 + \underbrace{w_4 c_0}_{\theta_8}(1 - x_1)(x_2)^3$$

$$y(x_1, x_2) = \sum_j \theta_j \varphi_j (x_1, x_2) = \theta^T \cdot \phi(x_1, x_2)$$

Neural Networks

- Any nonlinear approximator might be called a Neural Network
  - RBF Neural Network
  - Polynomial Neural Network
  - B-spline Neural Network
  - Wavelet Neural Network

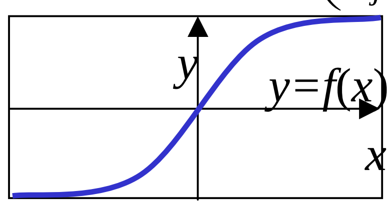
Linear in parameters



- MPL - Multilayered Perceptron
  - Nonlinear in parameters
  - Works for many inputs

$$y(\bar{x}) = w_{1,0} + f\left(\sum_j w_{1,j} y_j^1\right), \; y_j^1 = w_{2,0} + f\left(\sum_j w_{2,j} x_j\right)$$

$$f(x) = \frac{1}{1+e^{-x}}$$



Multi-Layered Perceptrons

- Network parameter computation
  - training data set
  - parameter identification

$$y(\overline{x}) = F(\overline{x};\theta)$$

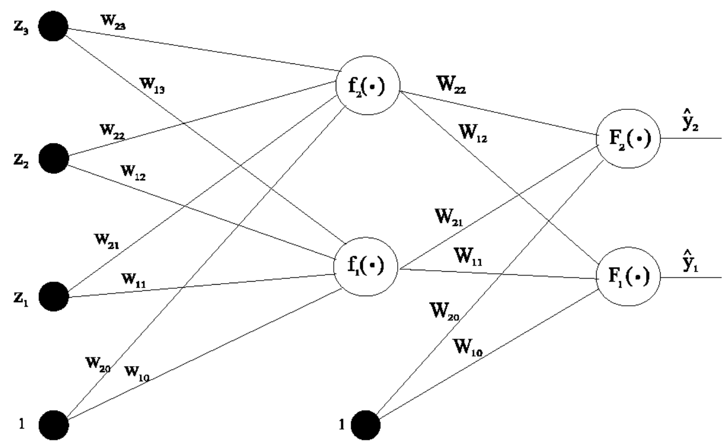
- Noninear LS problem

$$V = \sum_j \left\| y^{(j)} - F(\overline{x}^{(j)};\theta) \right\|^2 \to \min$$

- Iterative NLS optimization
  - Levenberg-Marquardt
- Backpropagation
  - variation of a gradient descent

$$Y = \left[y^{(1)} \quad \dots \quad y^{(N)} \right]$$

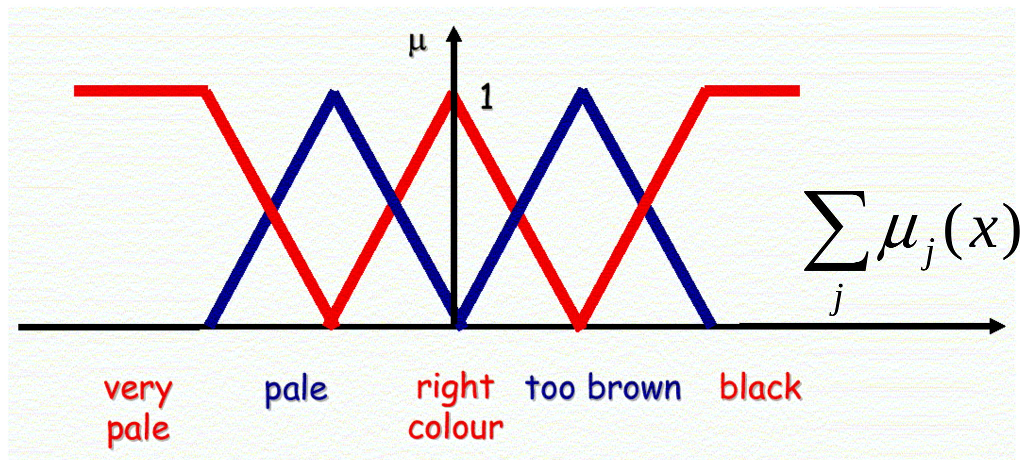
$$\overline{x}^{(1)} \quad \dots \quad \overline{x}^{(N)}$$



Fuzzy Logic

- Function defined at nodes. Interpolation scheme
- Fuzzyfication/de-fuzzyfication = interpolation
- Linear interpolation in 1-D

$$y(x) = \frac{\sum_j y_j \mu_j(x)}{\sum_j \mu_j(x)}$$

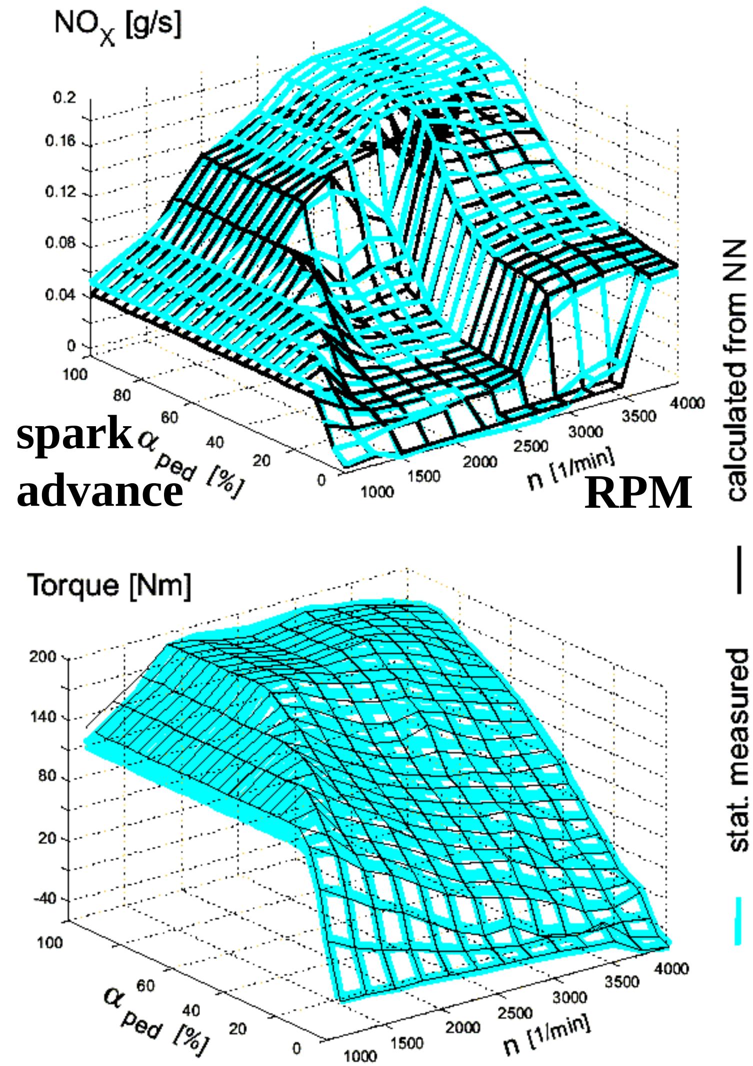


$$\sum_j \mu_j(x) = 1$$

- Marketing (communication) and social value
- Computer science: emphasis on interaction with a user
  – EE - emphasis on mathematical computations

Neural Net application

- Internal Combustion Engine maps
- Experimental map:
  - data collected in a steady state regime for various combination of parameters
  - 2-D table
- NN map
  - approximation of the experimental map
  - MLP was used in this example
  - works better for a smooth surface

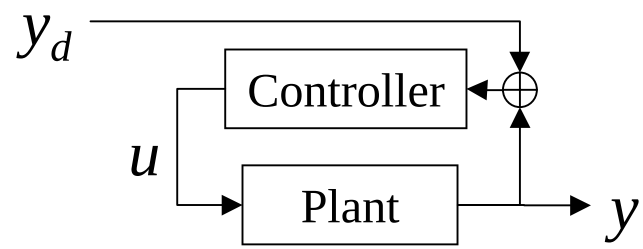


Linear feedback in a nonlinear plant

- Simple example

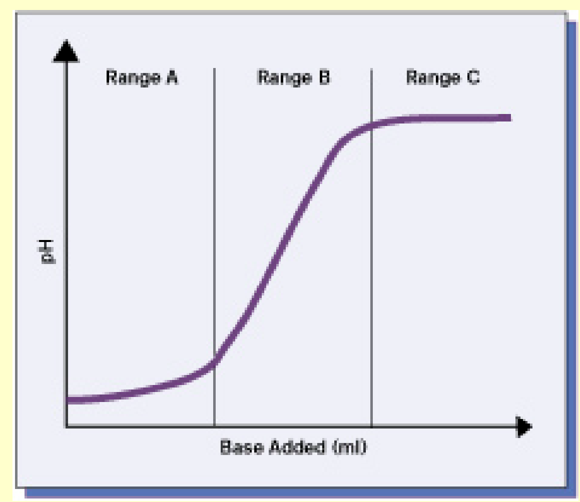
  $y = f(x) + g(x)u$

  $u = -k(x)(y - y_d) + u_{ff}(x)$



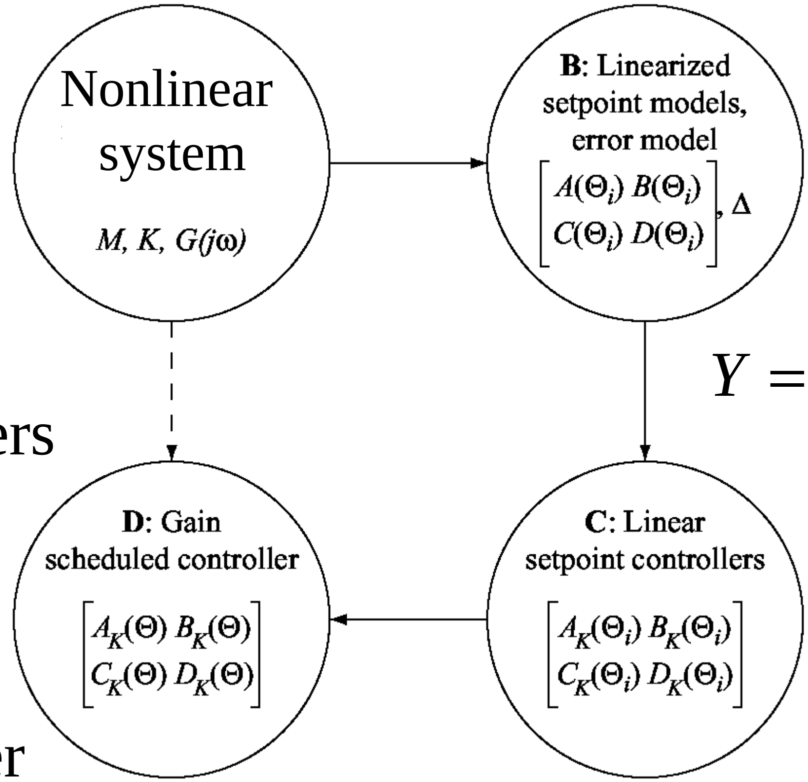
- Control design requires $k(x), u_{ff}(x), y_d(x)$
- These variables are *scheduled* on *x*

Example:
varying process gain



Gain scheduling

- Single out several regimes - model linearization or experiments
- Design linear controllers in these regimes: setpoint, feedback, feedforward
- Approximate controller dependence on the regime parameters



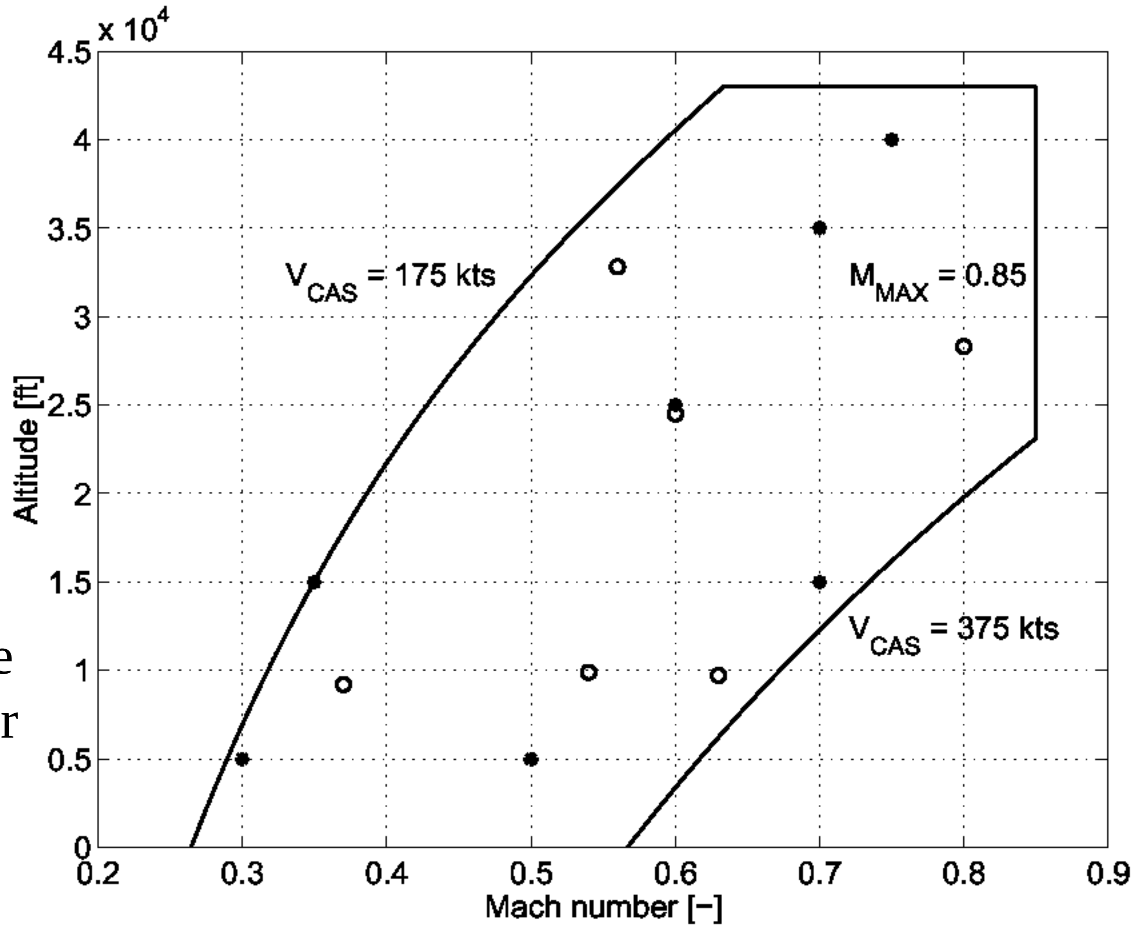
$$Y = \begin{bmatrix} \mathrm{vec}(A) \\ \mathrm{vec}(B) \\ \mathrm{vec}(C) \\ \mathrm{vec}(D) \end{bmatrix}$$

Linear interpolation:

$$Y(\Theta) = \sum_j Y_j \varphi_j(\Theta)$$

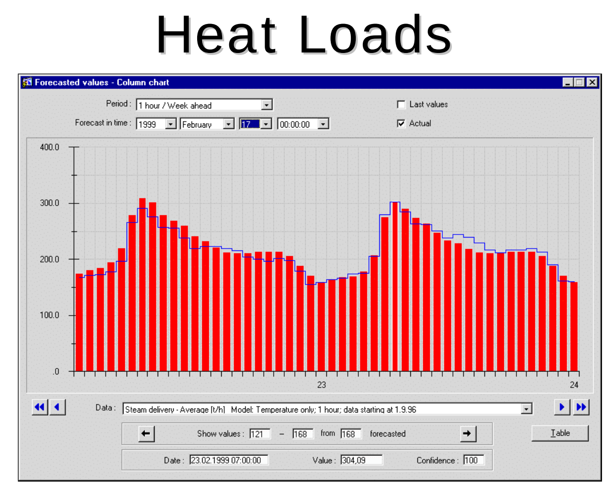
Gain scheduling - example

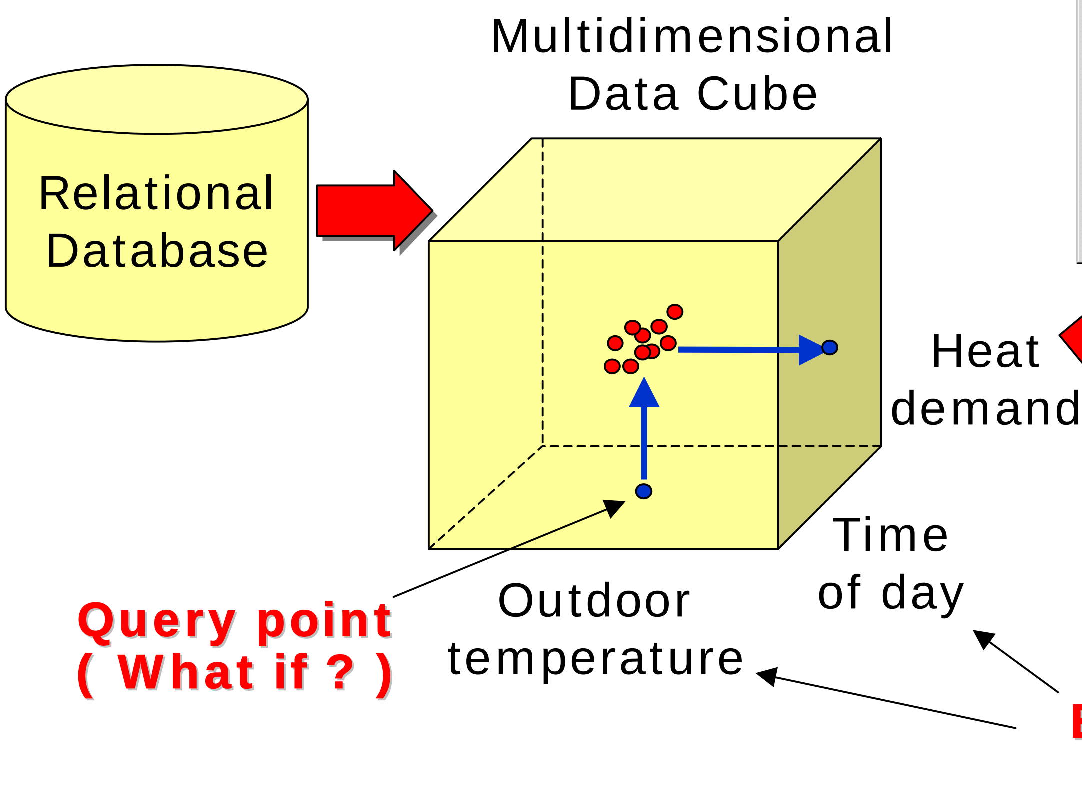
- Flight control
- Flight envelope parameters are used for scheduling
- Shown
  - Approximation nodes
  - Evaluation points
- Key assumption
  - Attitude and Mach are changing much slower than time constant of the flight control loop

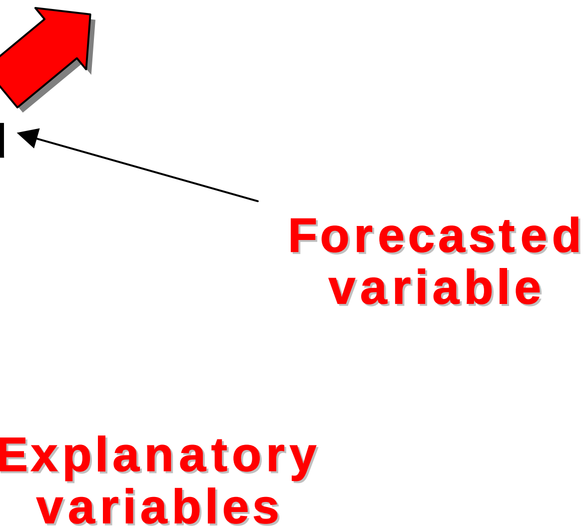


Local Modeling Based on Data

- Data mining in the loop
- Honeywell product







Lecture 14 - Health Management

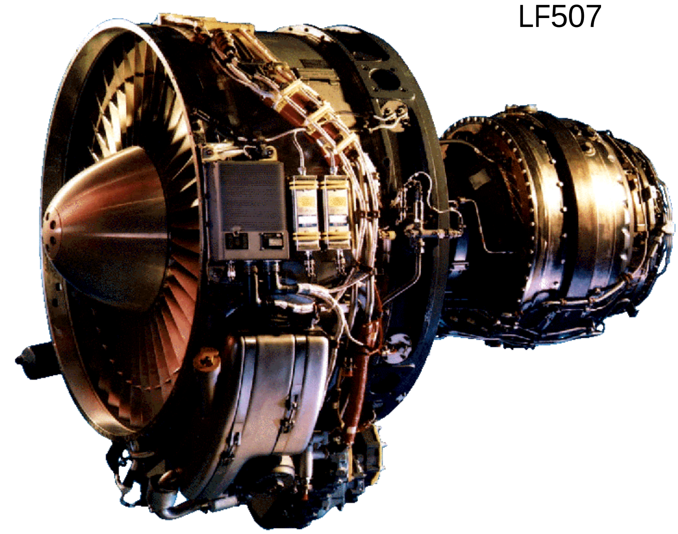
- Fault detection and accommodation
- Health management applications
  - Engines
  - Vehicles: space, air, ground, marine, rail
  - Industrial plants
  - Semiconductor manufacturing
  - Computing
- Abnormality detection - SPC
- Parameter estimation
- Fault tolerance - redundancy

Diagnostics in Control Systems

- Control algorithms are less than 20% of the embedded control application code in safety-critical systems
- 80% is dealing with special conditions, fault accomodation
  - BIT (Built-in Test - software)
  - BITE (Built-in Test Equipment - hardware)
  - Binary results
  - Messages
  - Used in development and in operation

Example

Honeywell
LF507



Health Management

- Emerging technology - recent several years
  - less established than most of what was discussed in the lectures
- Systems fault management functions
  - Abnormality detection and warning - something is wrong
  - Diagnostics - what is wrong
  - Prognostics - predictive maintenance
  - Accomodation - recover
- On-line functions - control system
  - Fault accommodation - FDIR
- Off-line functions - enterprise system
  - Maintenance automation
  - Logistics automation

Vehicle Health Management

- IVHM - Integrated Vehicle Health Management - On-board
- PHM - Prognostics and Health Management - On-ground
- Vehicles: space, air, ground, rail, marine
  - Integrated systems, many complex subsystems
  - Safety critical, on-going maintenance, on-board fault diagnostics

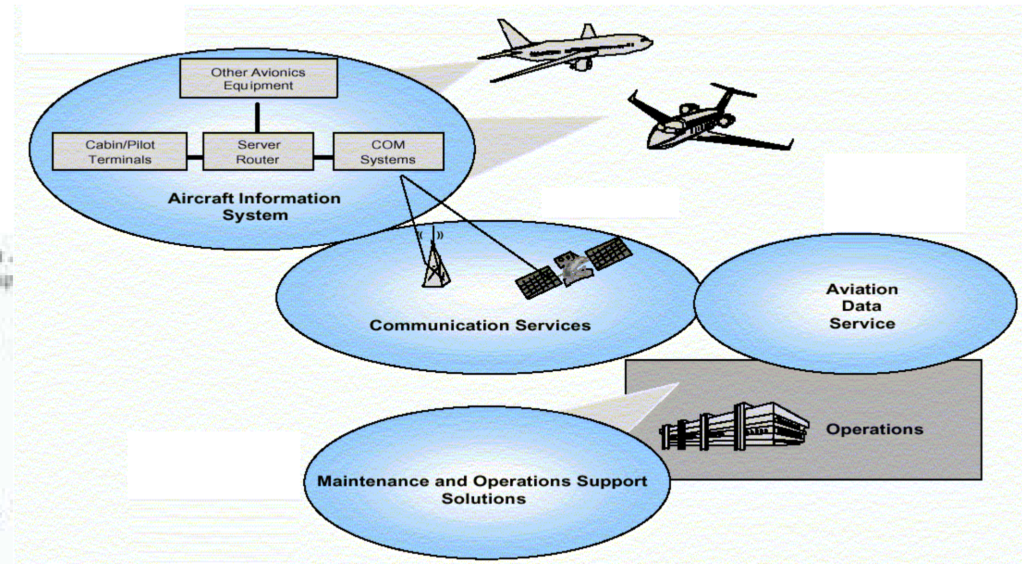


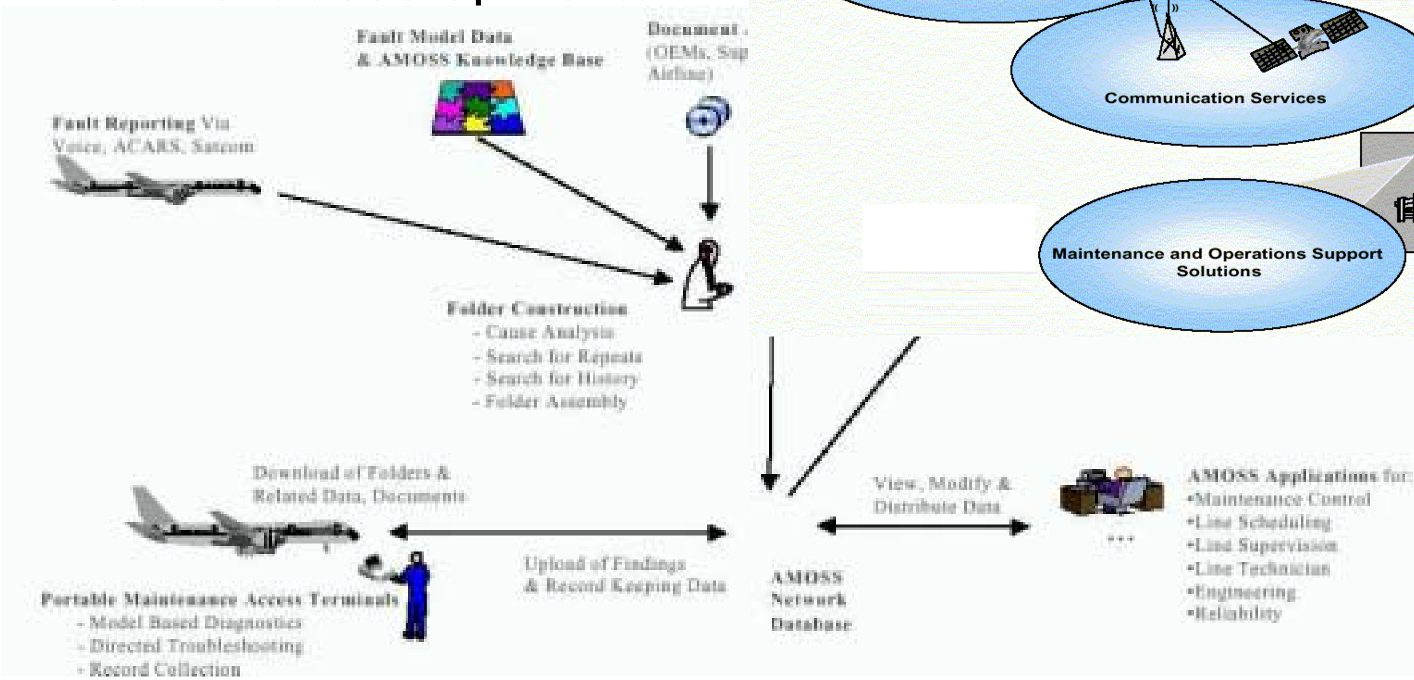


Airline enterprise - maintenance

Military

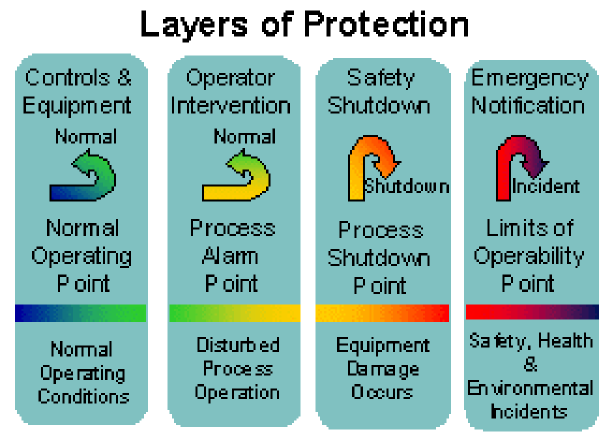
- Integrated on-board and on-ground system
- Maintenance automation
- Main expense/revenue
- Current developments

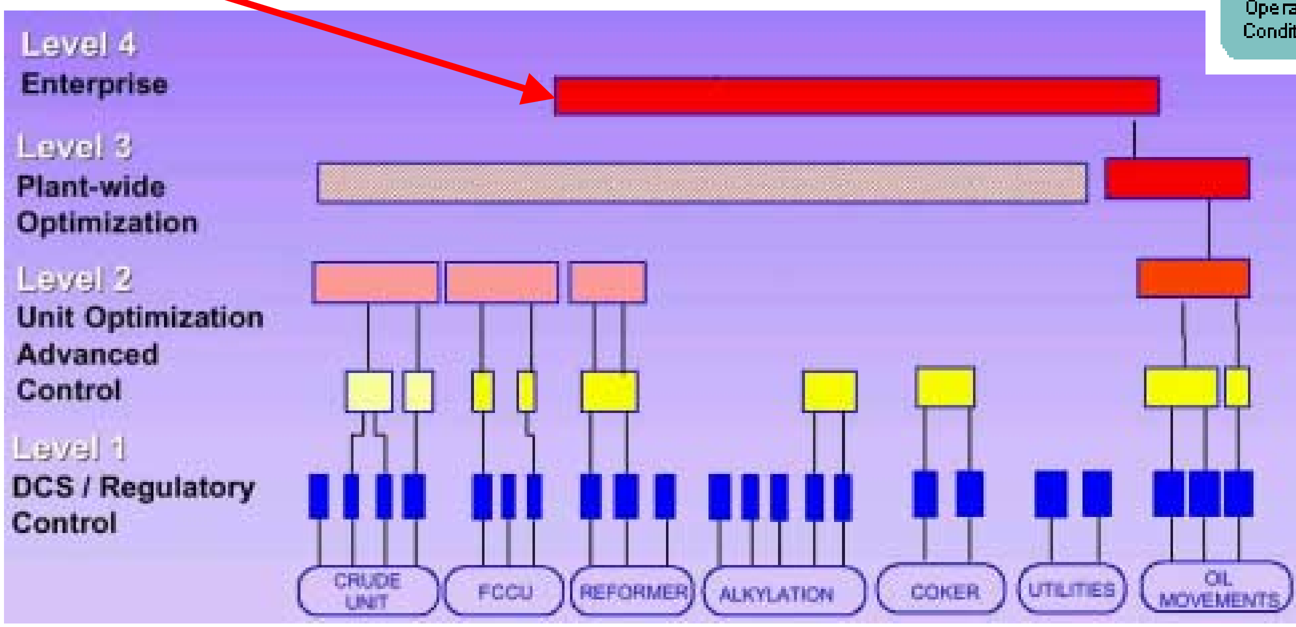




Industrial plants

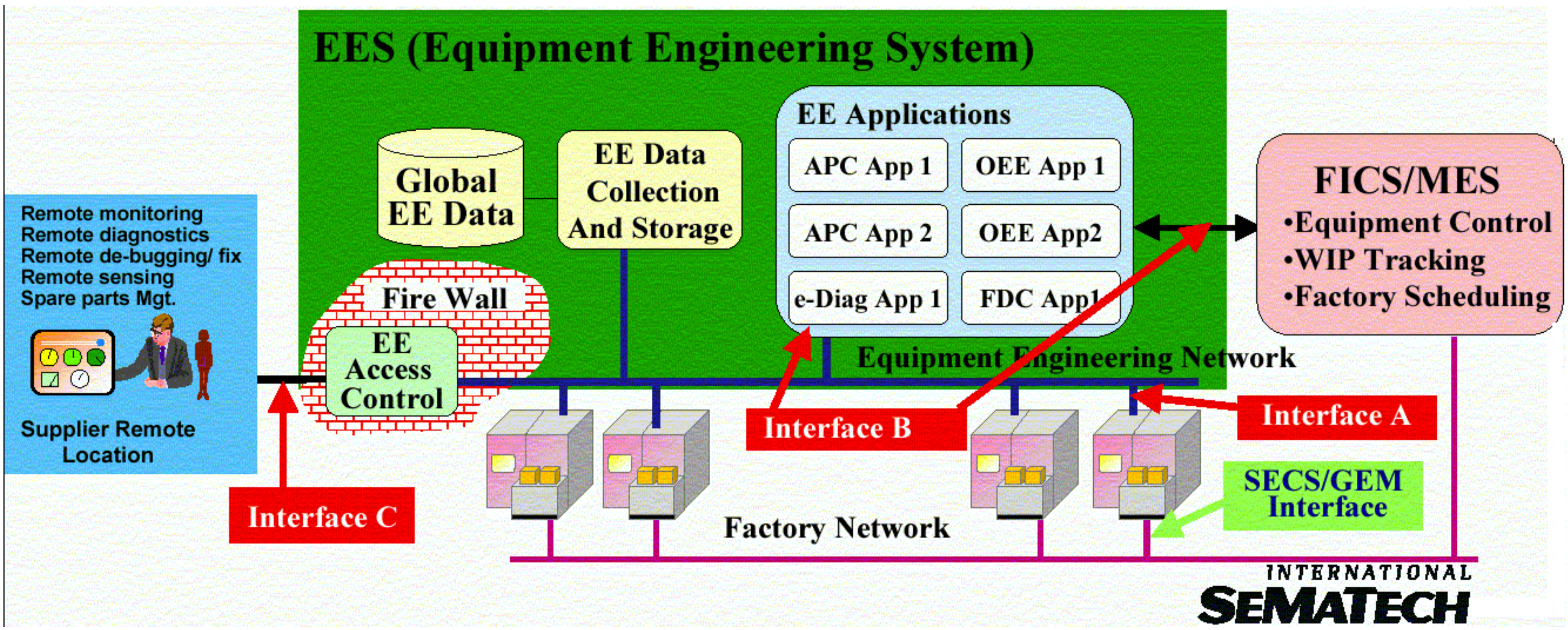
- Abnormal Situation Management
  - large cost associated with failures and production disruption
  - solutions are presently being deployed





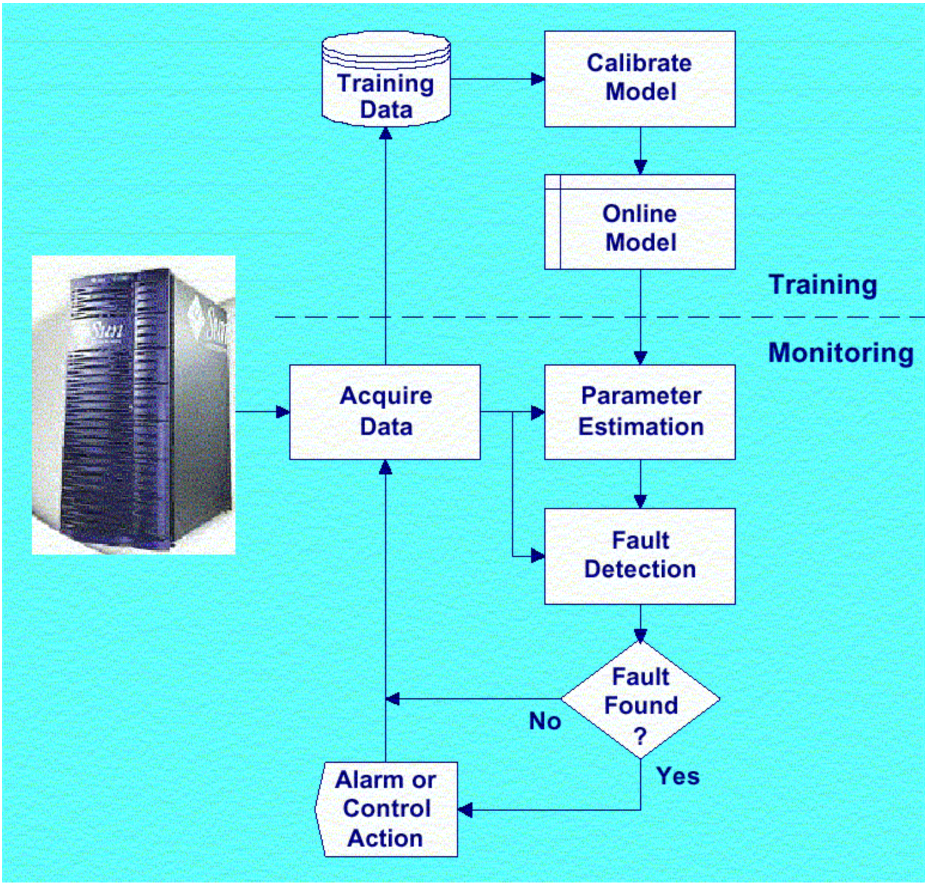
Semiconductor manufacturing

- E-diagnostics initiative by SEMATECH



Computing

- Autonomic computing
  - Fault tolerance
  - Automated management, support, security
  - IBM, Sun, HP - Scientific American, May 2002
- Sun Storage Automated Diagnostic Environment
  - Health Management and Diagnostic Services



K.Gross, Sun Microsystems

Abnormality detection - SPC

- SPC - Statistical Process Control (univariate)
  - discrete-time monitoring of manufacturing processes
  - early warning for an off-target quality parameter
- SPC vs EPC
  - EPC (Engineering Process Control) - ‘normal’ feedback control
  - SPC - operator warning of abnormal operation
- SPC has been around for 80 years
- Three main methods of SPC:
  - Shewhart chart (20s)
  - EWMA (40s)
  - CuSum (50s)

Abnormality detection - SPC

- Process model - SISO
  - quality variable randomly changes around a steady state value
  - the goal is to detect change of the steady state value

$$X(t) \approx \begin{cases} N(\mu_0, \sigma^2), & t \le T \\ N(\mu_1 \ne \mu_0, \sigma^2), & t > T \end{cases}$$

- Shewhart Chart

$$Y(t) = \frac{X(t) - \mu_0}{\sigma} \qquad \text{detection:} \quad Y(t) > Z = c_1$$

  - Simple thresholding for deviation from the nominal value μ_0
  - Typical threshold of 3σ <=> 0.27% probablity of false alarm

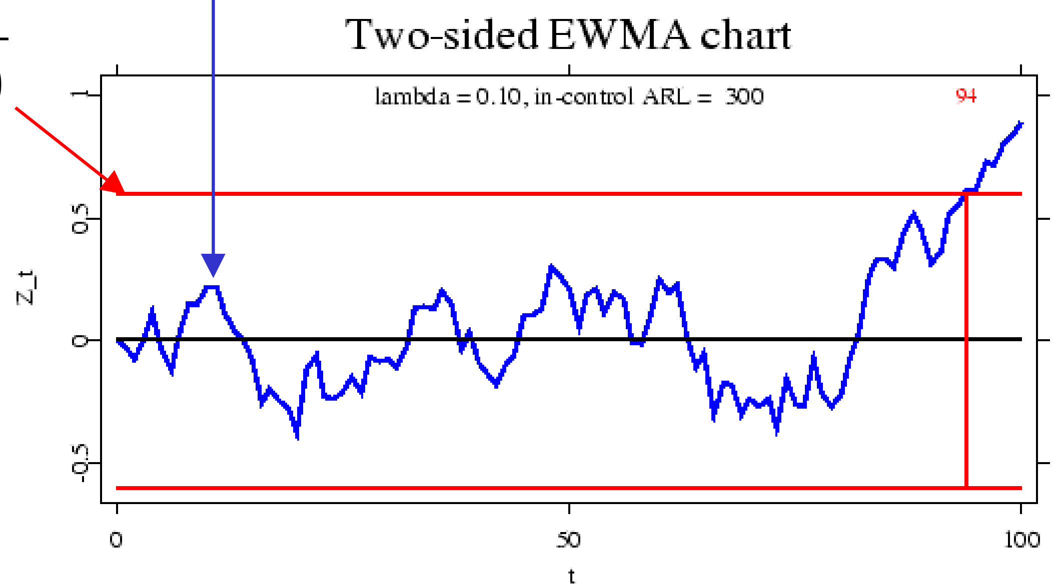
SPC - EWMA

- EWMA = Exponentially Weighted Moving Average
- First order low pass filter

$$Y(t+1) = (1-\lambda)Y(t-1) + \lambda X(t)$$

– Detection threshold

$$Z = c_2\sqrt{\lambda(2-\lambda)}$$



SPC - CuSum

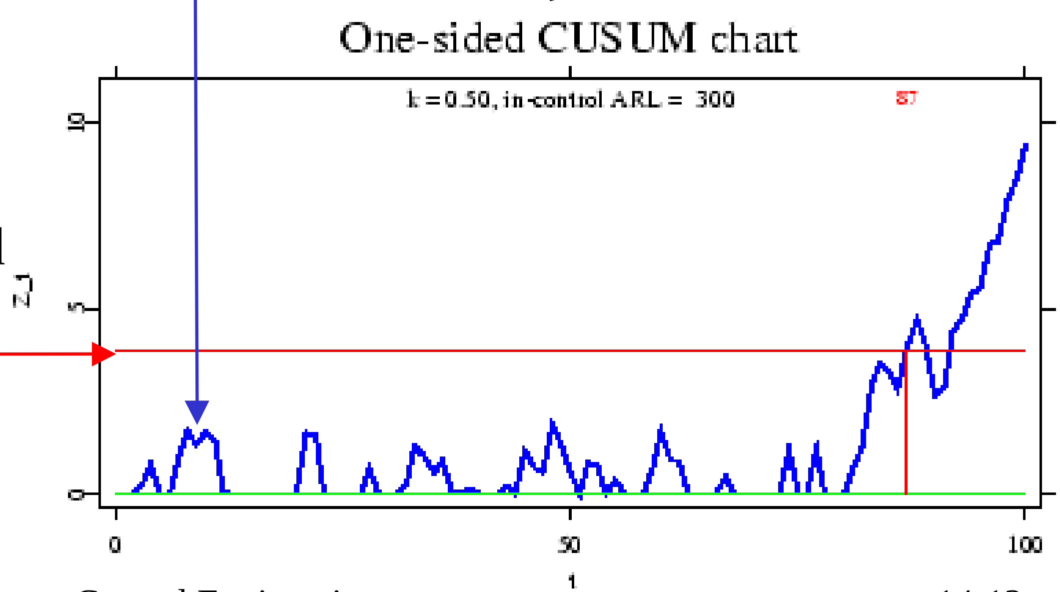
- CuSum = Cumulative Sum
  - a few modifications
  - one-sided CuSum most common

$$Y(t+1) = \max\left\{0, Y(t) + \frac{X(t) - \mu_0}{\sigma} - k\right\}$$

$$k = \frac{\mu_1 + \mu_0}{2\sigma}$$

  - Detection threshold

$$Z = c_3$$



Multivariate SPC - Hotelling's T^2

- The data follow multivariate normal distribution

$$X(t) \approx N(\mu, \Sigma)$$

$$X(t) = \Sigma^{1/2} Y(t) + \mu$$

Uncorrelated white noise

- Empirical parameter estimates

$$\mu = E(X) \approx \frac{1}{n} \sum_{t=1}^{n} X(t)$$

$$\Sigma = E\left((X-\mu)(X-\mu)^T\right) \approx \frac{1}{n} \sum_{t=1}^{n} (X(t)-\mu)(X^T(t)-\mu)$$

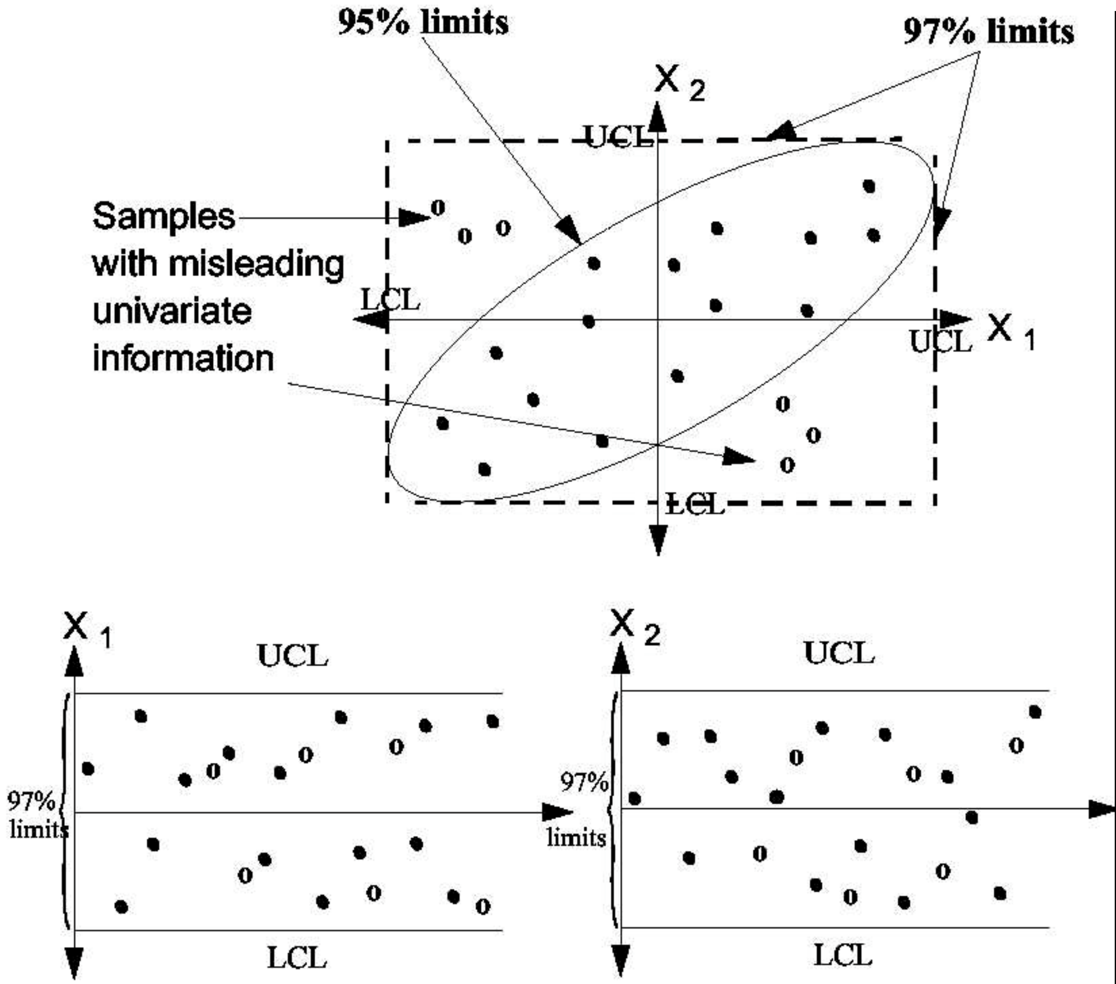
- The Hotelling's T^2 statistics is

$$T^2 = \left(X(t)-\mu\right)^T \Sigma^{-1} \left(X(t)-\mu\right)$$

$$T^2 = Y^T(t) Y(t)$$

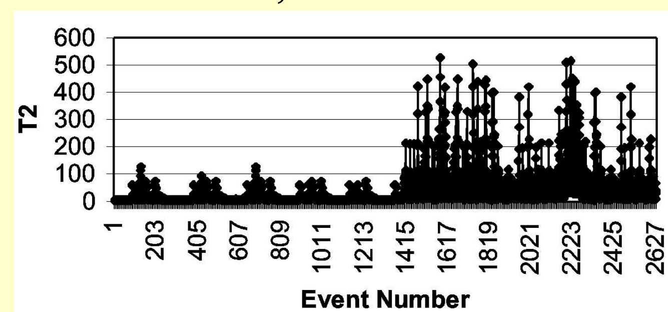
- T can be trended as a univariate SPC variable (almost)

Multivariate SPC

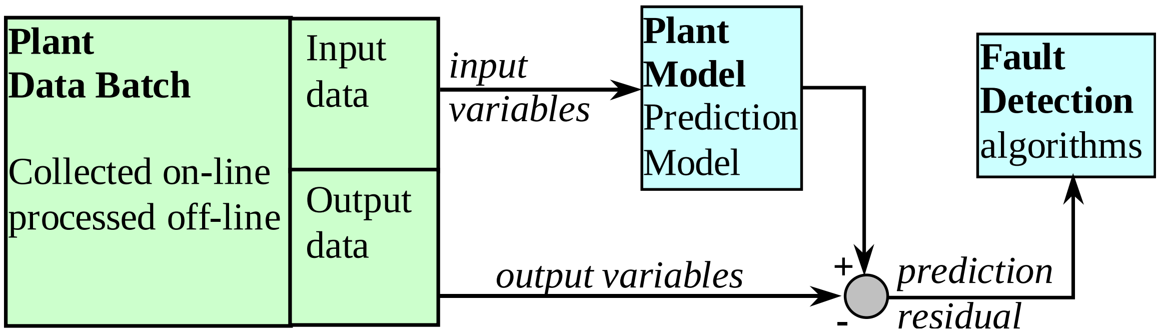


Example:

- MSPC for Cyber attack detection
- *X*(*t*) consists of 284 audit events in Sun's Solaris Basic Security Module
- Ye & Chen, Arizona State



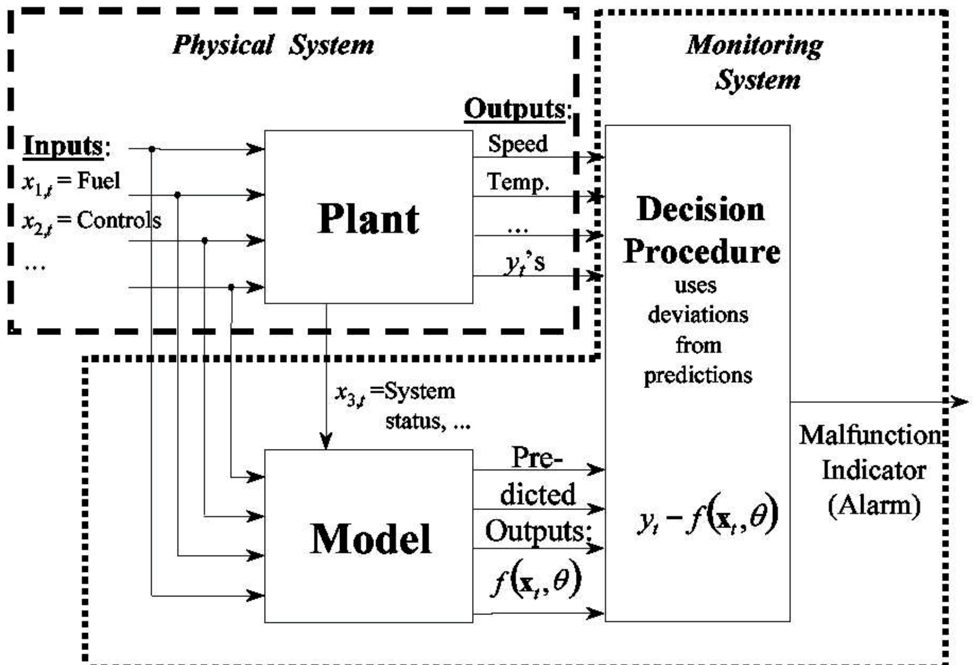
Model-based fault detection



- Compute model-based prediction residual
  - result of a simulation run

  $X = Y - f(U, \theta)$

- If $\theta = 0$ (nominal case) we should have $X = 0$.
- X reflects faults



Model-based fault detection

- Compute model-based prediction residual $X(t)$ at cycle t
  - flight/trip/maneuver for a vehicle
  - update time interval or a batch for a plant
  - semiconductor process run
- $X(t)$ reflects modeling error, process randomness, and fault
- Use MSPC for detecting abnormality through $X(t)$
  - Hotelling's T^2
  - CuSum
- Does not tell us what the fault might be (diagnostics)

Parameter estimation

- Residual model: $X = Y - f(U,\theta)$

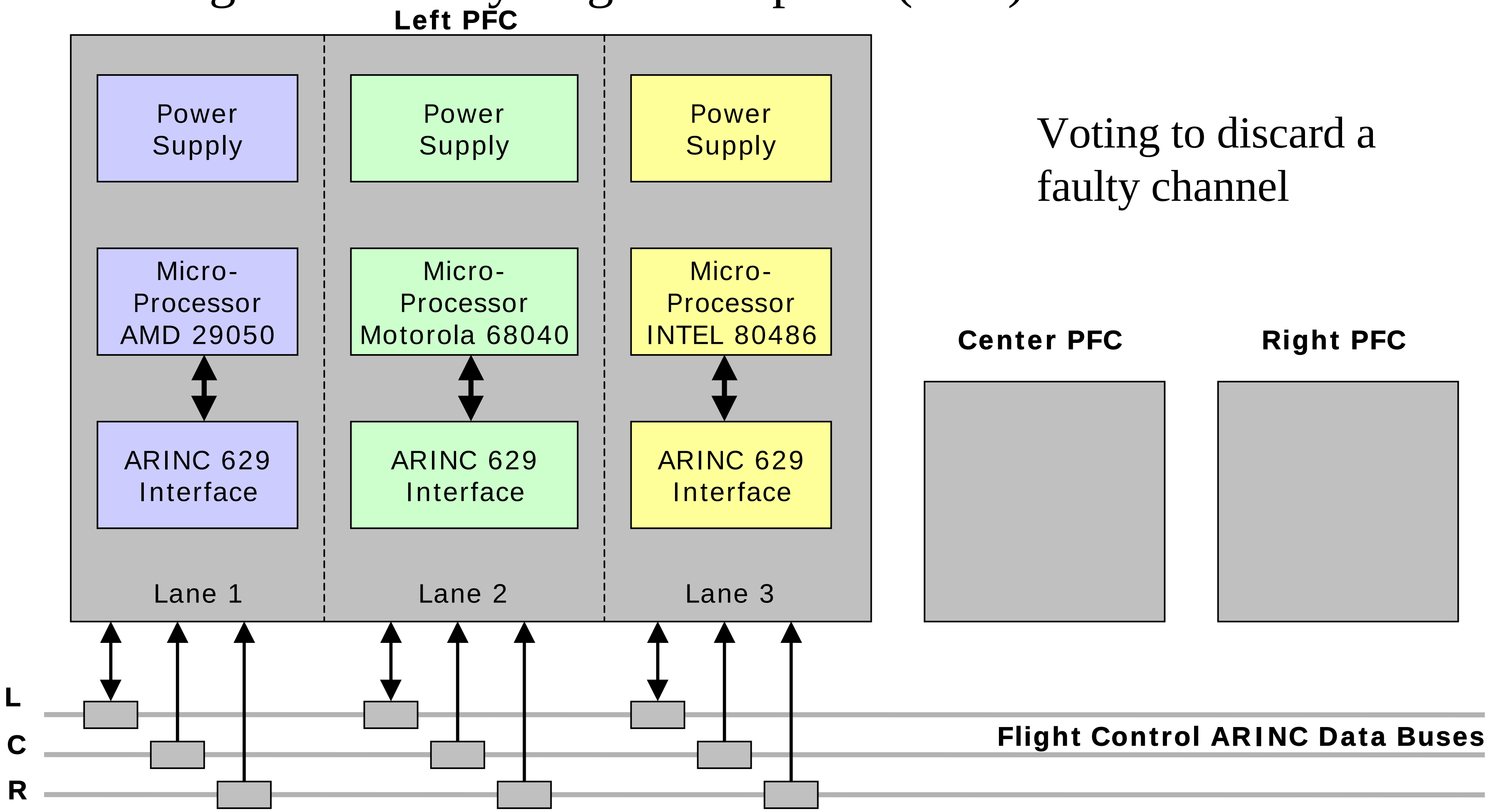
$$X = \Phi\theta + \xi \qquad\qquad \Phi = -\frac{\partial f(U,\theta)}{\partial\theta}$$

- Fault models - meaning of θ
  - Sensor fault model - additive output change
  - Actuator fault model - additive input change
- Estimation technique
  - Fault parameter estimation - regression

$$\hat{\theta} = \left(\Phi^T\Phi + rI\right)^{-1}\Phi^T X$$

Fault tolerance: Hardware redundancy

- Boeing 777 Primary Flight Computer (PFC) Architecture



Voting to discard a faulty channel

Analytical redundancy

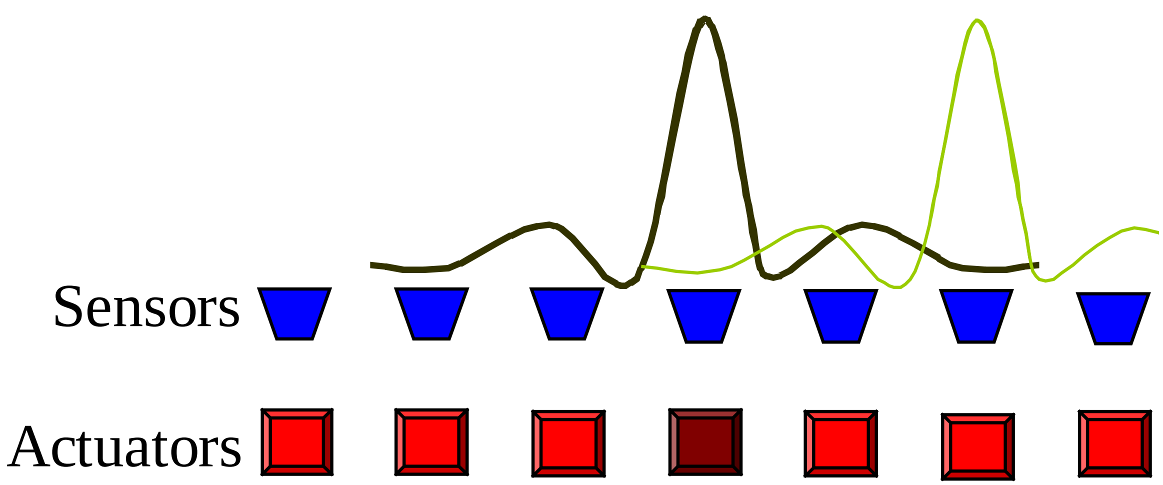
- Analytical Redundancy
  - correlate data from diverse measurements through an analytical model of the system
- Estimation techniques
  - KF observer
- Talked about in the literature
- Used only in much simplified form:
  - on loss of a sensor, use inferential estimate of the variable using other sensor measurements
  - on loss of an actuator, re-allocate control to other actuators

Lecture 15 - Distributed Control

- Spatially distributed systems
- Motivation
- Paper machine application
- Feedback control with regularization
- Optical network application
- Few words on good stuff that was left out

Distributed Array Control

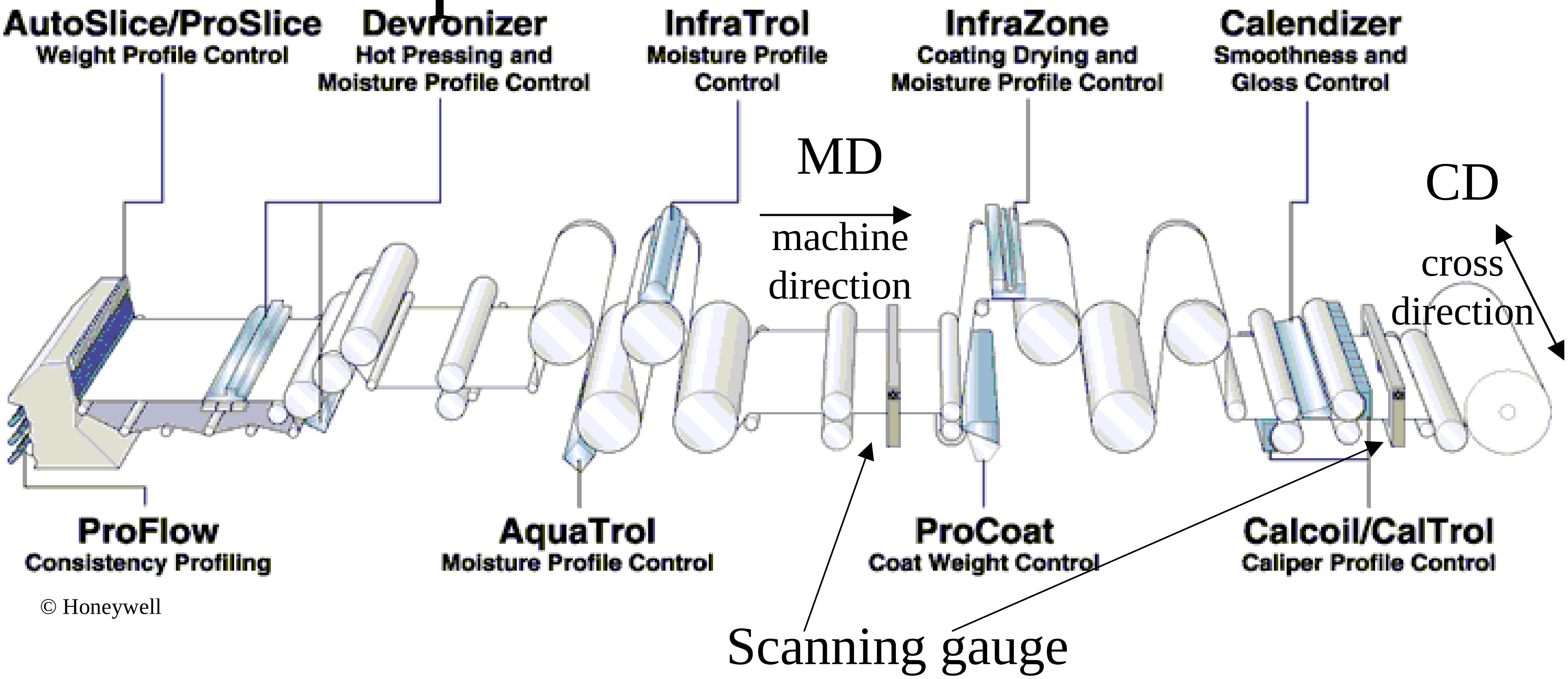
- Sensors and actuators are organized in large arrays distributed in space.
- Controlling spatial distributions of physical variables
- Problem simplification: the process and the arrays are uniform in spatial coordinate
- Problems:
  - modeling
  - identification
  - control



Distributed Control Motivation

- Sensors and actuators are becoming cheaper
  - electronics almost free
- Integration density increases
- MEMS sensors and actuators
- Control of spatially distributed systems increasingly common
- Applications:
  - paper machines
  - fiberoptic networks
  - adaptive and active optics
  - semiconductor processes
  - flow control
  - image processing

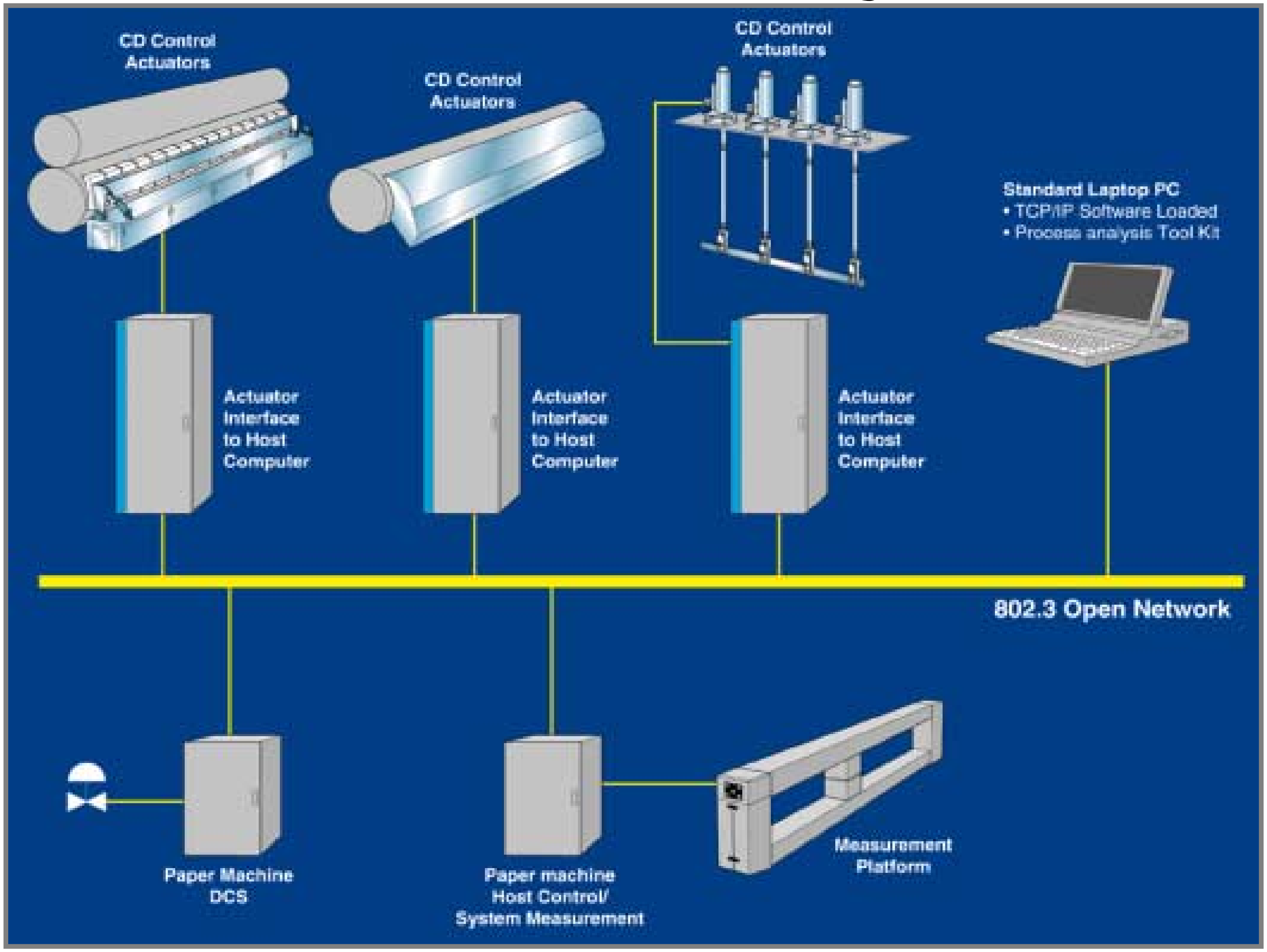
Paper Machine Process



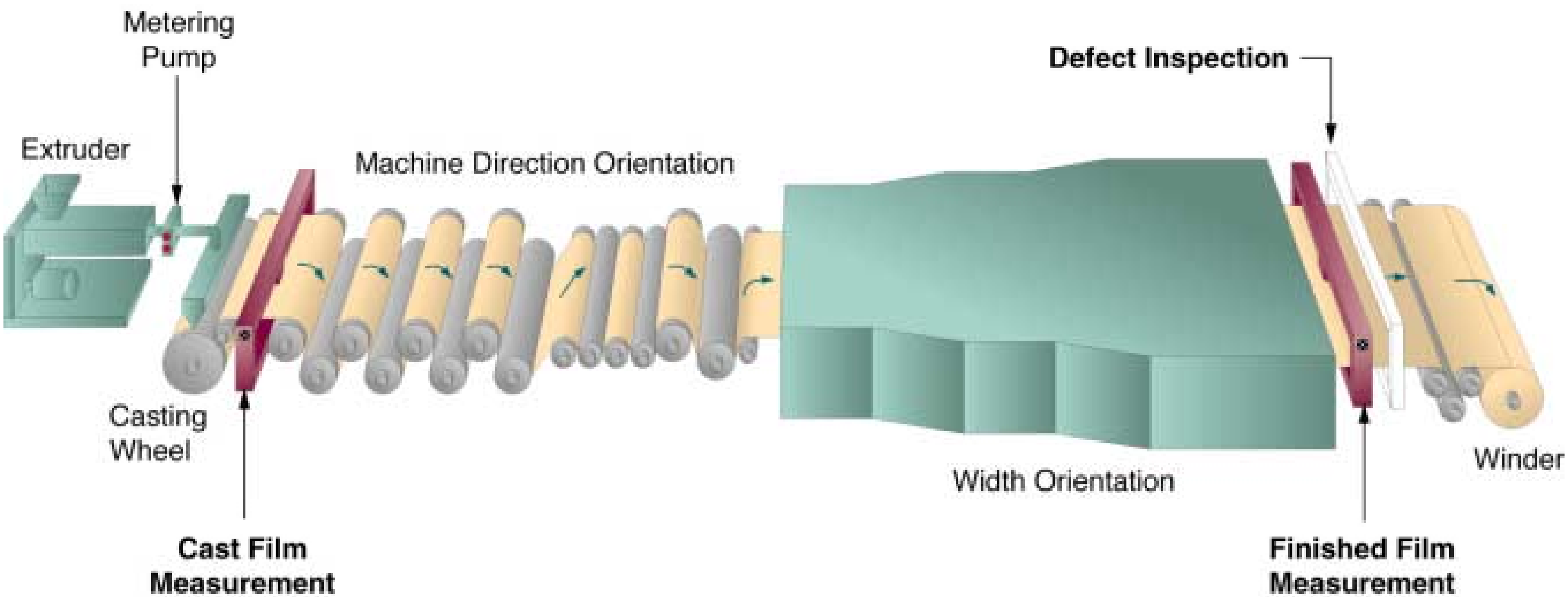
- Control objective: flat profiles in the cross-direction
- The same control technology for different actuator types: flow uniformity control, thermal control of deformations, and others

Headbox with Slice Lip CD Actuators

Profile Control System



Biaxial Plastic Line Control



© Honeywell

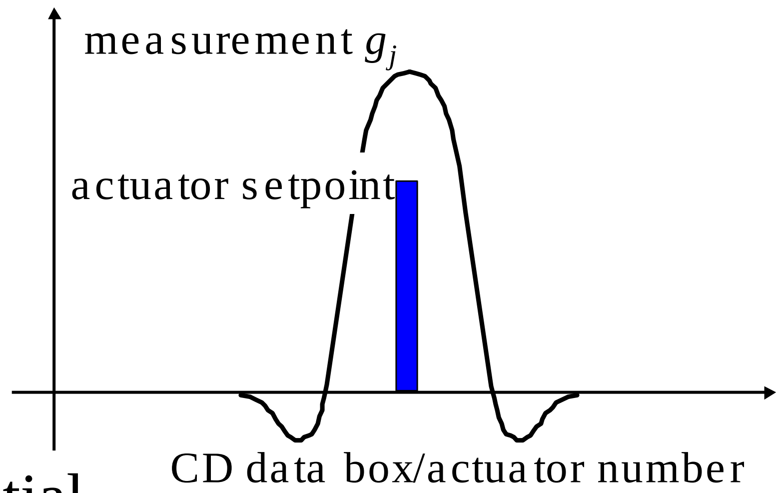
Model Structure

- Process-independent model structure

$$\Delta Y = G \Delta U$$

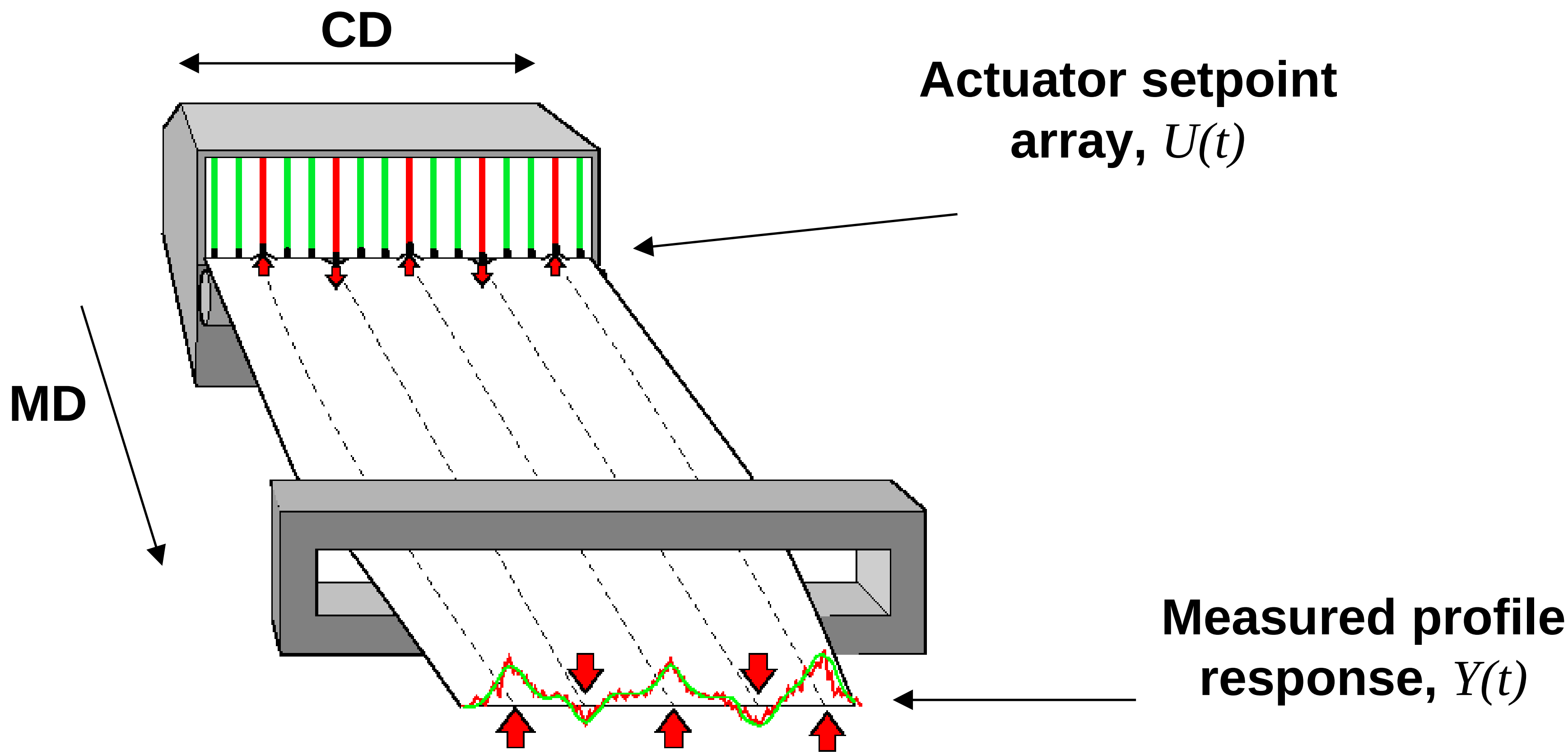
$$Y \in \Re^m, U \in \Re^n, G \in \Re^{m,n}$$

- G - spatial response matrix with columns g_j
- Known parametric form of the spatial response (noncausal FIR)
- Green Function of the distributed system

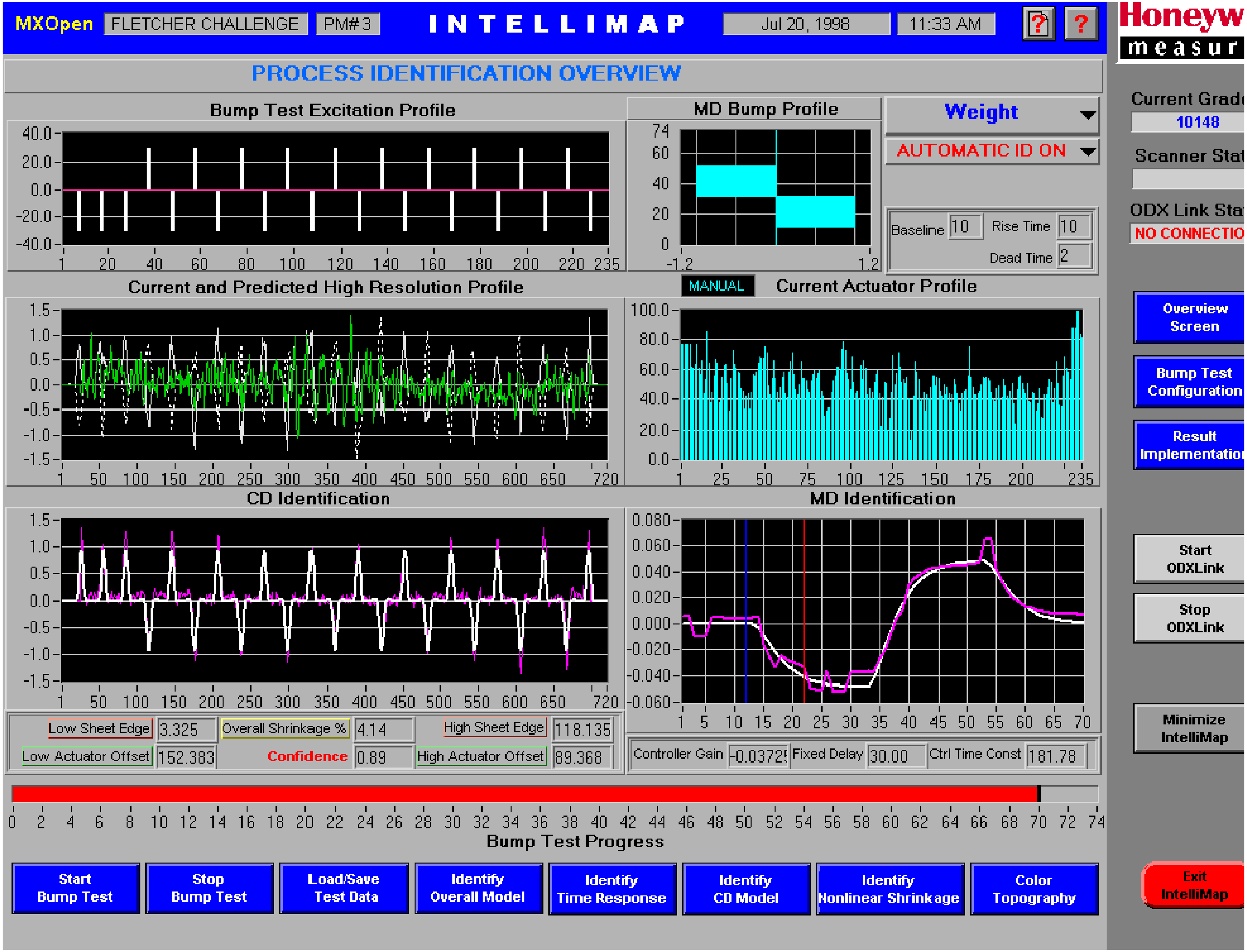


$$g_{j,k} = g \varphi(x_k - c_j)$$

Process Model Identification



- Extract noncausal FIR model
- Fit parameterized response shape



MXOpen
FLETCHER CHALLENGE
PM#3
INTELLIMAP
Jul 20, 1998
11:33 AM
Honeyw
measur
PROCESS IDENTIFICATION OVERVIEW
Bump Test Excitation Profile
MD Bump Profile
Weight
AUTOMATIC ID ON
Baseline 10
Rise Time 10
Dead Time 2
Current and Predicted High Resolution Profile
MANUAL
Current Actuator Profile
CD Identification
MD Identification
Low Sheet Edge 3.325
Overall Shrinkage % 4.14
High Sheet Edge 118.135
Low Actuator Offset 152.383
Confidence 0.89
High Actuator Offset 89.368
Controller Gain -0.0372
Fixed Delay 30.00
Ctrl Time Const 181.78
Bump Test Progress
Start Bump Test
Stop Bump Test
Load/Save Test Data
Identify Overall Model
Identify Time Response
Identify CD Model
Identify Nonlinear Shrinkage
Color Topography
Current Grade
10148
Scanner Stat
ODX Link Sta
NO CONNECTIO
Overview Screen
Bump Test Configuration
Result Implementation
Start ODXLink
Stop ODXLink
Minimize IntelliMap
Exit IntelliMap

Simple I control

- Compare to Lecture 4, Slide 5

I control

- Step to step update:

$$Y(t) = G \cdot U(t) + D(t)$$

$$U(t) = U(t-1) - k\left[Y(t-1) - Y_d\right]$$

- Closed-loop dynamics

$$Y = \left((z-1)I + kG\right)^{-1}\left[kGY_d + (z-1)D\right]$$

- Steady state: $z = 1$

$$Y = Y_d, \qquad U = G^{-1}(Y_d - D)$$

Simple I control

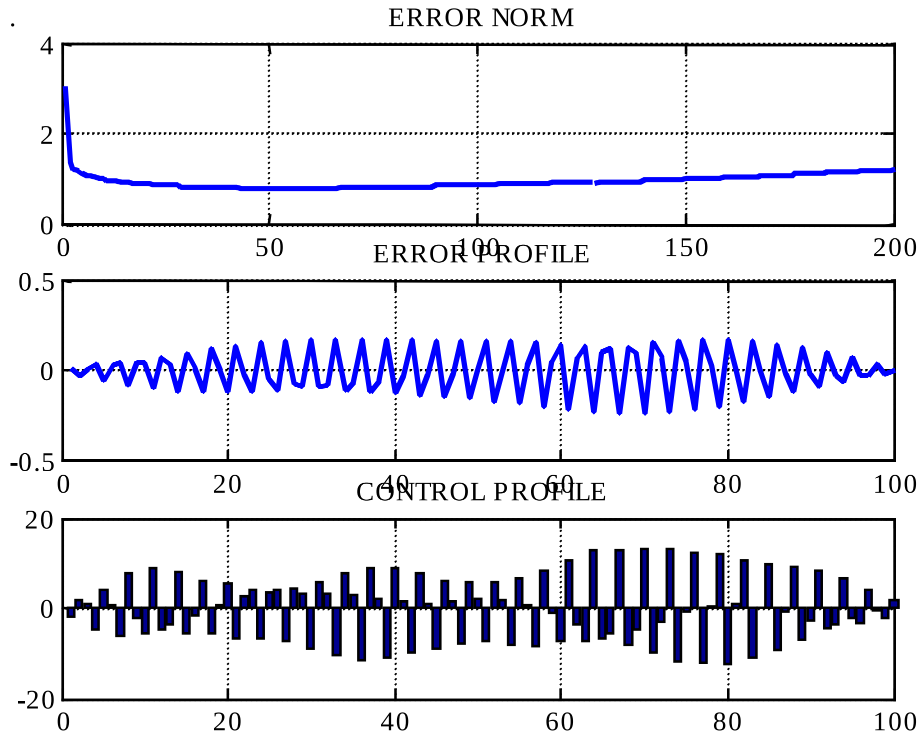
Issues with simple I control

- G not square positive definite
  - use G^T as a spatial pre-filter

$$Y_G(t) = G^T G \cdot U(t) + D_G(t)$$

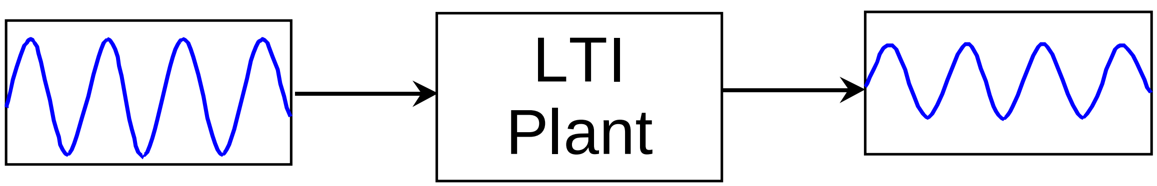
$$Y_G = G^T Y, \quad D_G = G^T D$$

- For ill-conditioned G get very large control, picketing
  - use regularized inverse
- Slowly growing instability
  - control not robust
  - regularization helps again



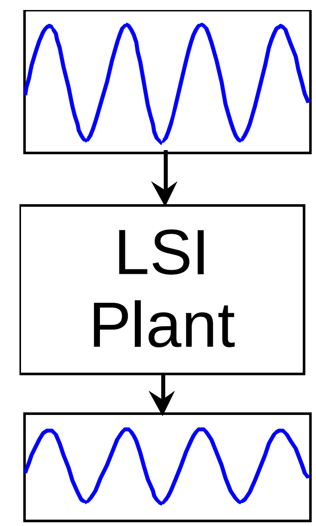
Frequency Domain - Time

- LTI system is a convenient engineering model
- LTI system as an input/output operator
- Causal
- Can be diagonalized by harmonic functions
- For each frequency, the response is defined by amplitude and phase



Frequency Domain - Space

- Linear Spatially Invariant (LSI) system
- LSI system is a convenient engineering model
- LSI system as an input/output operator
- Noncausal
- Can be diagonalized by harmonic functions
- Diagonalization = modal analysis; spatial modes are harmonic functions



Control with Regularization

- Add integrator leakage term

$$\Delta U(t) = -K\left(Y(t-1) - Y_d\right) - SU(t-1)$$

- Feedback operator K
  - spatial loopshaping
    - $KG \approx 1$ at low spatial frequencies
    - $KG \approx 0$ at high spatial frequencies
- Smoothing operator S
  - regularization
    - $S \approx 0$ at low spatial frequencies
    - $S \approx s_0$ at high spatial frequencies - regularization

Spatial Frequency Analysis

- Matrix G $\rightarrow$ convolution operator g (noncausal FIR) $\rightarrow$ spatial frequency domain (Fourier) $g(v)$
- Similarly: $K \rightarrow k(v)$ and $S \rightarrow s(v)$
- Each spatial frequency - mode - evolves independently

$$y(\nu) = \frac{g(\nu)k(\nu)}{z-1+s(\nu)+g(\nu)k(\nu)} y_d + \frac{z-1+s(\nu)}{z-1+g(\nu)k(\nu)} d$$

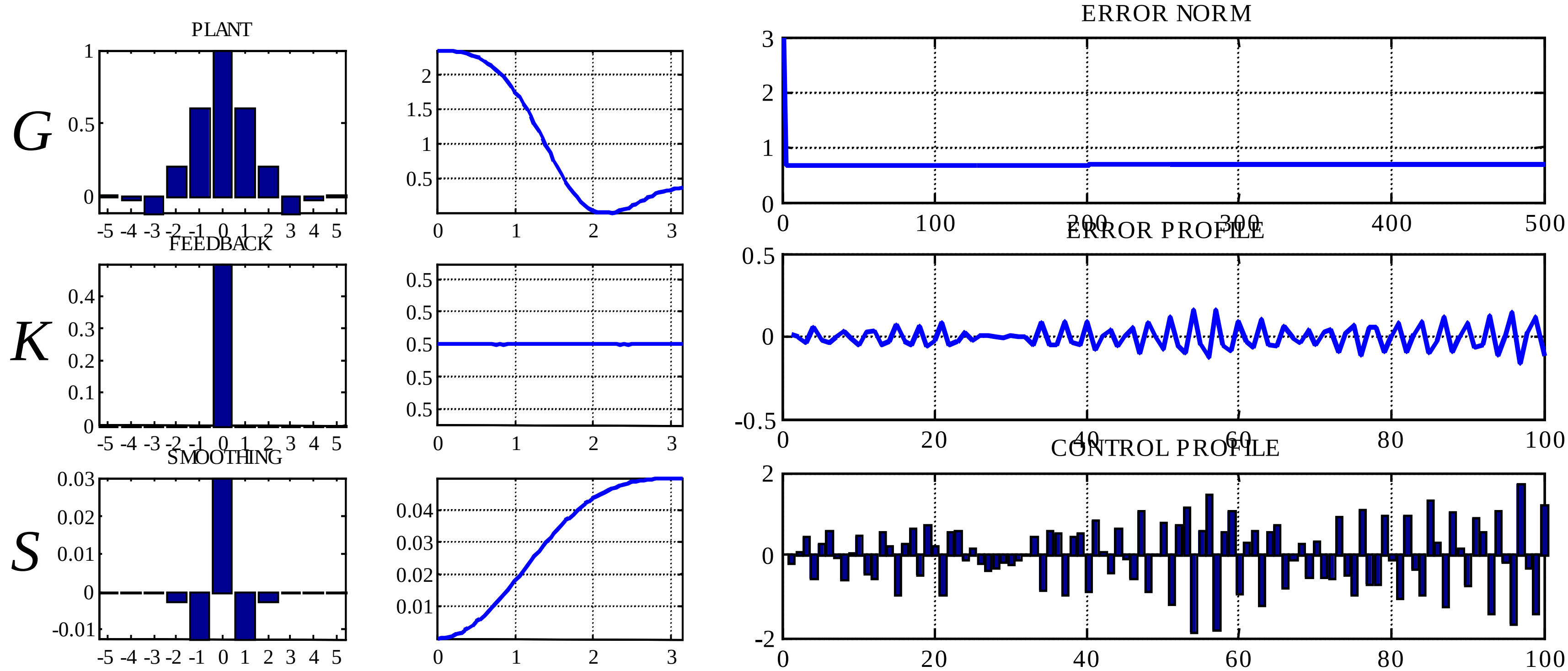
- Steady state

$$y(\nu) = \frac{g(\nu)k(\nu)}{s(\nu)+g(\nu)k(\nu)} y_d + \frac{s(\nu)}{s(\nu)+g(\nu)k(\nu)} d$$

$$u(\nu) = \frac{k(\nu)}{s(\nu)+g(\nu)k(\nu)} \left(y_d(\nu) - d(\nu) \right)$$

Sample Controller Design

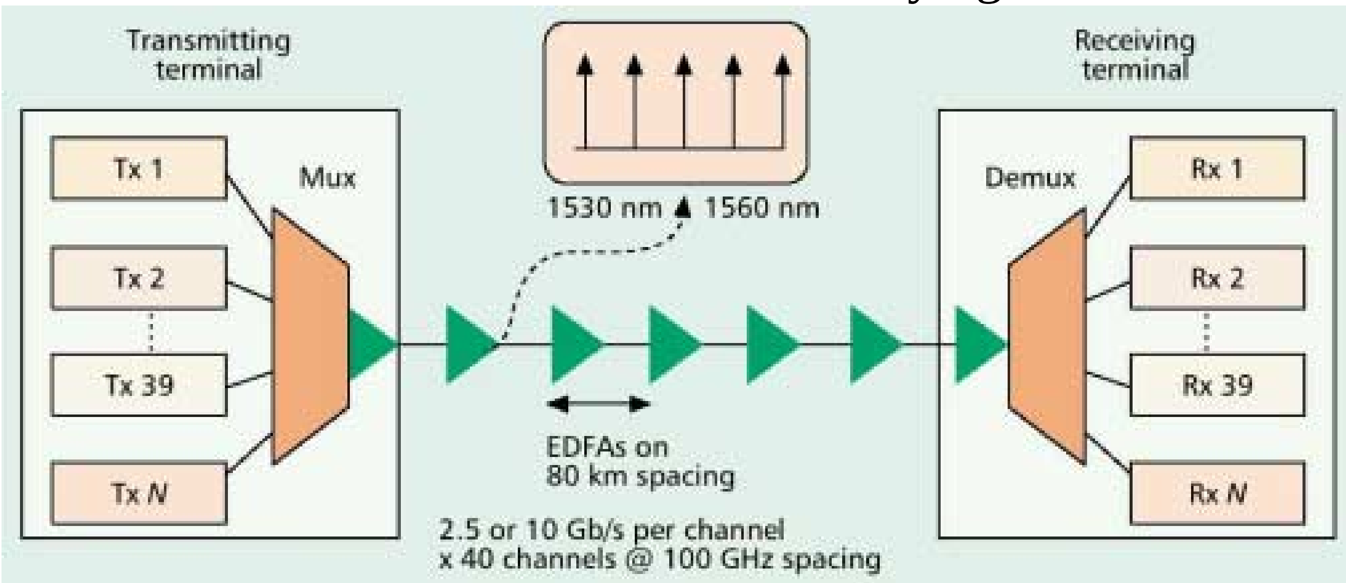
- Spatial domain loopshaping is easy - it is noncausal
- Example controller with regularization



For more depth and references, see: Gorinevsky, Boyd, Stein, ACC 2003

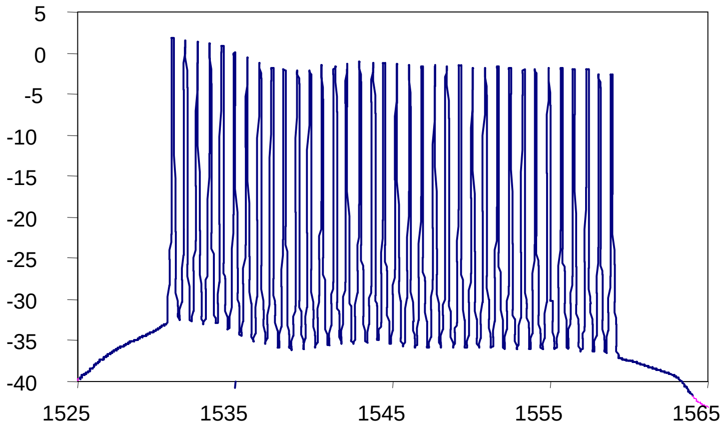
WDM network equalization

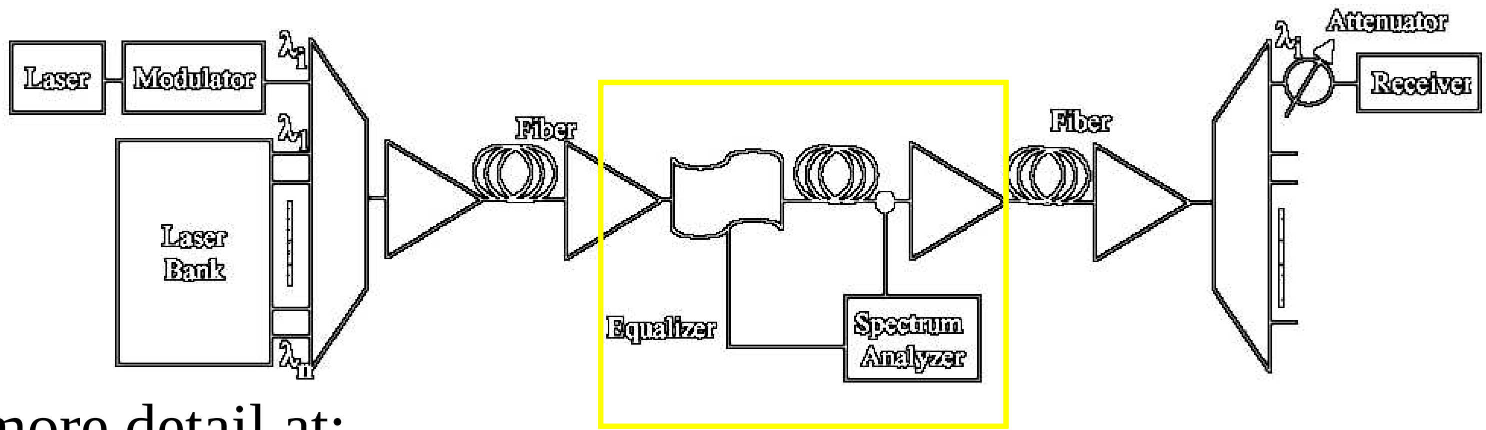
- WDM (Wave Division Multiplexing) networks
  - multiple (say 40) independent laser signals with closely space wavelength packed (multiplexed) into a single fiber
  - each wavelength is independently modulated
  - in the end the signals are unpacked (de-mux) and demodulated
  - increases bandwidth 40 times without laying new fiber



WDM network equalization

- Analog optical amplifiers (EDFA) amplify all channels
- Attenuation and amplification distort carrier intensity profile
- The profile can be flattened through active control





See more detail at:

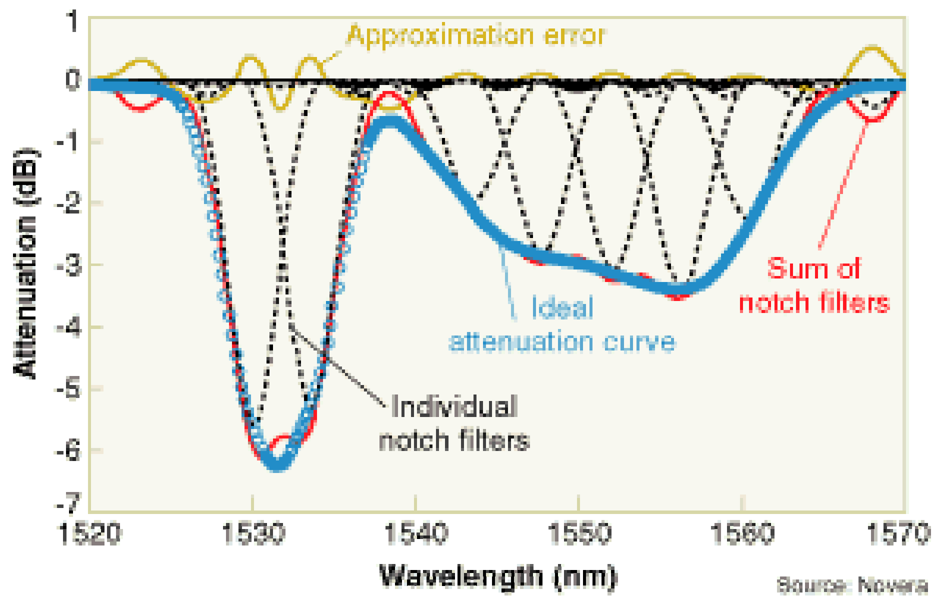
www126.nortelnetworks.com/news/papers_pdf/electronicast_1030011.pdf

WDM network equalization

- Logarithmic (dB) attenuation for a sequence of notch filters

$$A = A_1 \cdot \ldots \cdot A_N$$

$$\log A = \sum_{k=1}^{N} \log A_k$$



$$a(\lambda) = \sum_{k=1}^{N} w_k \varphi(\lambda - \lambda_0 - ck)$$

Attenuation gain - control handle

Notch filter shape

WDM

Good stuff that was left out

- Estimation and Kalman filtering
  - navigation systems
  - data fusion and inferential sensing in fault tolerant systems
- Adaptive control
  - adaptive feedforward, noise cancellation, LMS
  - industrial processes
  - thermostats
  - bio-med applications, anesthesia control
  - flight control
- System-level logic
- Integrated system/vehicle control