Bottom-Up Architectures

Bo Bernhardsson and K. J. Åström

Department of Automatic Control LTH,
Lund University

Bottom-Up Architectures

1. Introduction
2. Basic Architectures
3. Large Parameter Variations
4. Otto J. M. Smith's Specials
5. Miscellaneous
6. Soft Computing
7. Summary

Theme: Brick by brick.

Design

- Design is a part of all engineering disciplines: A, ME, EE, CE, CS, AA, ...
- Building complex systems from standard parts have been a standard procedure. Nuts, bolts standard assemblys. Transistors, boards, cabinetts. VLSI, graphs, design rules, libraries. Subroutines, programs, component software.
- Many attempts to make *design theories* and *design methods* - not too successful
- CS has been an interesting proving ground because it is easy to experiment, but also easy to include realistic settings - Abelson Sussman Structure and Interpretation of Computer Programs
- Chip design is the role model

  Abstractions, Layering, Design rules, Testing
- Can we imitate it?

Views on Control System Design

Holistic

- Requirements
- Specifications
- Modeling
- Analysis
- Simulation
- Design
- Implementation
- Commissioning
- Operation
- Upgrading

Reductionistic

- Stability
- Robustness
- Passivity
- Optimization

Components and Architectures

Build a system by combining a collection of building blocks. Key ingredients are building blocks and combination principles.

- What are the components (blocks, algorithms,...)?
- What are the rules for combining components? - Design principles.

Buiding blocks

- Linear Systems
  - Controllers
  - Estimators
- Nonlinearities
  - Limiters
  - Selectors
  - Logic
- Estimators
- Optimizers

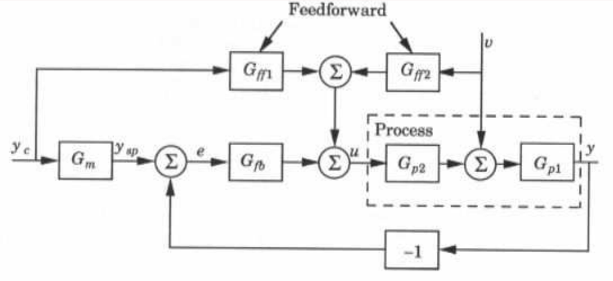
System principles

- Feedback
- Feedforward
- Cascade
- Midranging
- Selector control
- Model following
- Gain scheduling
- Adaptation

Bottom-Up Architectures

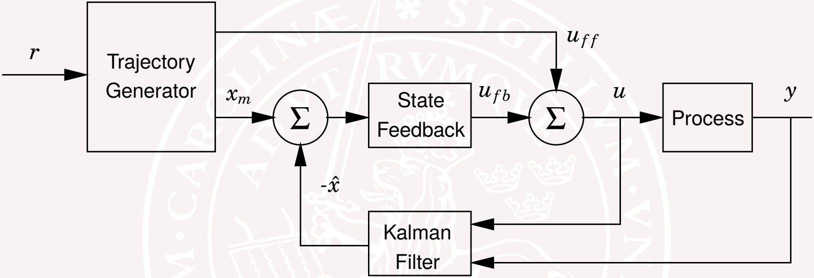
Theme: Brick by brick.

Feedback and Feedforward



- Reduce effects of measured disturbances and improve command signal response
- Feedback and feedforward have nice complementary properties
- Feedback does not require accurate models (sensitivity can be less than 1), feedforward does (sensitivity is always one)
- Feedback can lead to instability, feedforward can never destabilize
- Windup protection

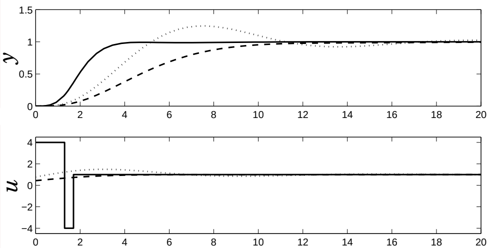
Command Signal Following in State Feedback



- A nice separation of the different functions
- The signals x_m and u_{ff} can be generated from r in real time or from stored tables (robotics)
- Integral action and windup protection

Improved Command Signal Response

- Essentially a problem of computing inverses or approximate inverses of systems $PM_u = M_y$ with constraints
- Make reasonable demands, time delays and RHP zeros of P must be included in M_y
- The feedforward parts M_u and M_y can be nonlinear. Modelica can deliver inverse models

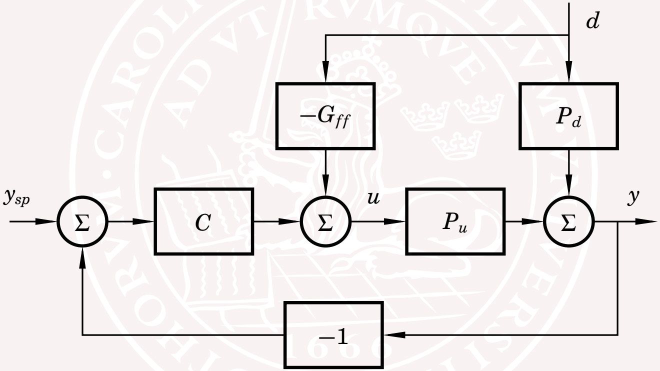


PI control with $M_s = 1.4$ (dashed) and $M_s = 2.0$ (dotted) for $P(s) = 1/(s+1)^4$ andnonlinear feedforward (solid)

Feedforward - Measured Disturbance

Basic scheme for reducing effects of measured load disturbance

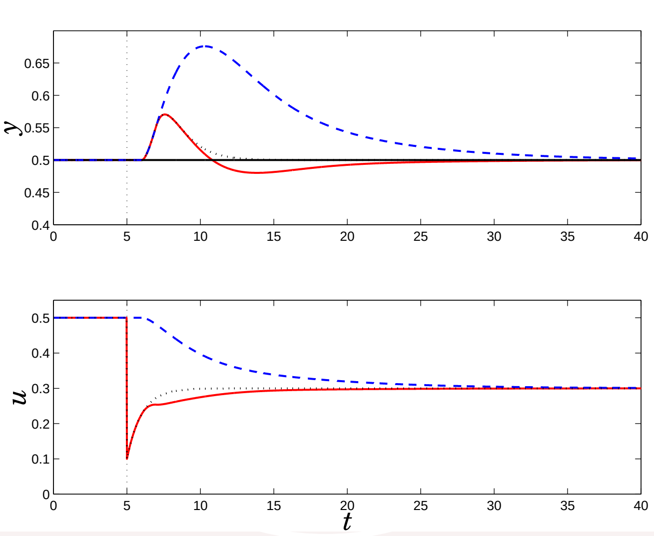
$$G_{ff} = P_u^{-1} P_d = P_u^{\dagger} P_d$$



$$G_{yd} = \frac{P_d - P_u G_{ff}}{1 + P_u C}$$

TAT: Can you see some drawbacks if perfect cancellation is not possible?

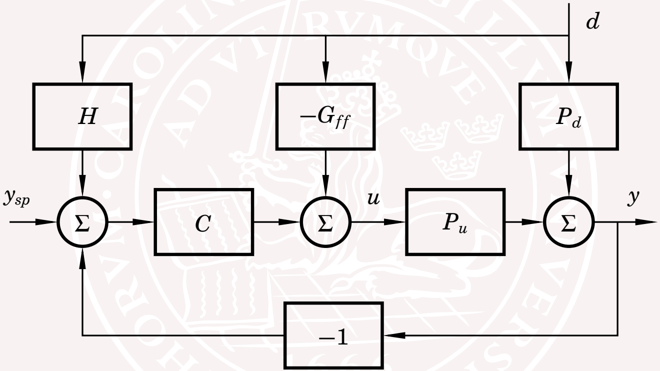
Simulation



Only feedback dashed line FB+FF red full line dotted Brosilov's version

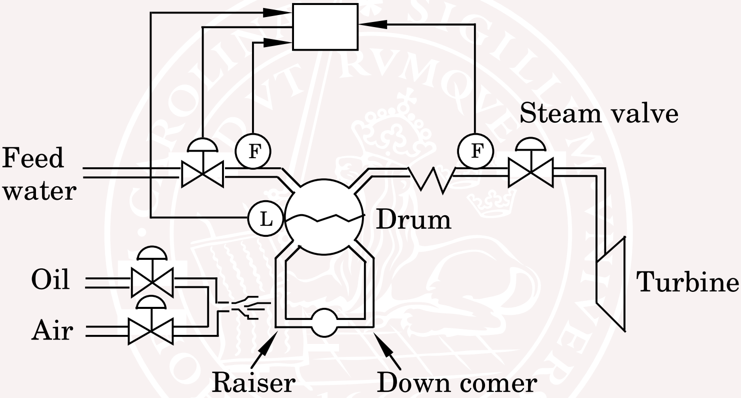
Feedforward - Brosilov's 2DOF

A 2DOF scheme where controller is told what the feedforward is doing



$$G_{yd} = \frac{P_d - P_u G_{ff}}{1 + P_u C}, \qquad H = P_d - P_u G_{ff}, \qquad G_{yd} = P_d - P_u G_{ff}$$

An Example - Drum Level Control



- The shrink and swell effect

Flatness M. Fliess

- System inversion is to compute the input that gives the output for a given system
- The concept of flat output
- A flat output y is an output signal such that the state x and the input u can be generated from y and its derivatives.

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n y = b_n u$$

- Useful for feedforward design by making the flat output behave in the desired way
- Can be applied to nonlinear systems

Generalized Integral Control

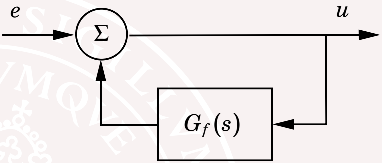
- Integral control was a real break through

  Maxwell (1868) mentioned that Siemens (1866) had distinguished between governors (PI) and moderators (P)
- Automatic removal of steady state errors (automatic reset)
- Integral control eliminates a disturbance that is constant with unknown amplitude
- Can it be generalized to other types of disturbances?

  Ramps $v(t) = a_0 + a_1 t$ where a_0, a_1 are unknown
  Jerks $v(t) = a_0 + a_1 t + a_2 t^2$ where $a_0, a_1\ a_2$ are unknown
  Sinusoidal disturbances with known frequency but unknown amplitude and phase
  Sinusoidal disturbances with unknown frequency, amplitude and phase
  Periodic disturbances with known period
- Idea: Build a model of the disturbance in the controller!

Generalized Integral Control

- Constant but unknown
- Ramps with unknown levels and rates
- Sinusoidal with known frequency but unknown amplitude and phase
- Periodic with known period but unknown shape



$$C(s) = \frac{k}{1 - G_f(s)}$$

$$G_f(s) = \frac{1}{1 + sT_f} \qquad C_{const}(s) = 1 + \frac{1}{sT}$$

$$G_f(s) = \frac{2\zeta\omega_0 s}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \qquad C_{sine}(s) = \frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{s^2 + \omega_0^2}$$

$$G_f(s) = e^{-sT} \qquad C_{periodic}(s) = \frac{1}{1 - e^{-sT}}$$

Elimination of Periodic Disturbances

$$G_f(s) = e^{-sT} \qquad C_{periodic}(s) = \frac{1}{1 - e^{-sT}}$$

Control law

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

Transfer function from disturbance to output

$$G_{yd}(s) = \frac{P(s)}{1 + P(s)C(s)} = \frac{P(s)(1 - e^{-sT})}{1 - e^{-sT} + P(s)}.$$

The relation between load disturbance and output

$$(1 - e^{-sT} + P(s))Y(s) = P(s)\big(1 - e^{-sT}\big)D(s).$$

Notice that the time function corresponding to $\big(1 - e^{-sT}\big)D(s)$ is $d(t) - d(t - T)$, which is zero if d has period T. Compare with PI control.

Recovering Robustness

- No difficulties with infinite gain for a PI controller
- Difficulties with controller that have infinite gain at high frequencies
- The remedy is to lower the gain and introduce high frequency roll-off

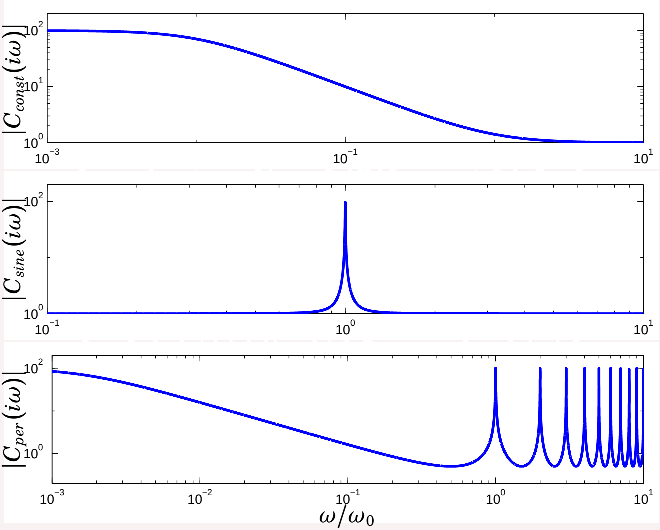
Replacing G_f by αG_f gives $C = \frac{1}{1-\alpha G_f}$, with α close to 1

$$C_{\text{const}}(s) = \frac{1+sT}{1-\alpha+sT}$$

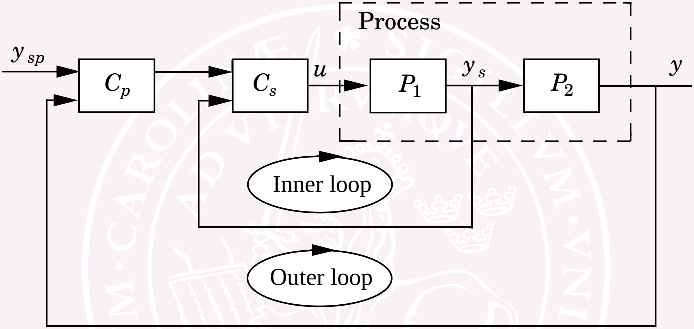
$$C_{\text{sine}}(s) = \frac{s^2+2\zeta\omega_0 s+\omega_0^2}{s^2+2(1-\alpha)\zeta\omega_0 s+\omega_0^2}$$

$$C_{\text{per}}(s) = \frac{1}{1-\alpha e^{-sT}}$$

Bode Plots for $\alpha = 0.99$

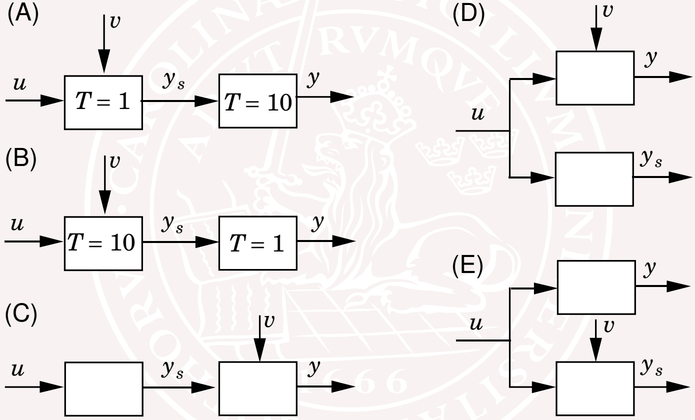


Cascade Control - Several Sensors



- Using several sensors - state feedback is the ultimate
- State feedback ultimate case
- Tight feedback around disturbances and uncertainty
- Linearize a nonlinear actuator
- Integral action and windup

When is Cascade Control Useful?

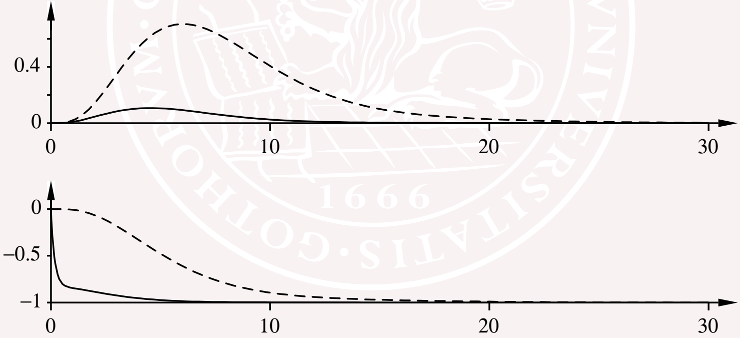


Cascade Control - Example

Process dynamics

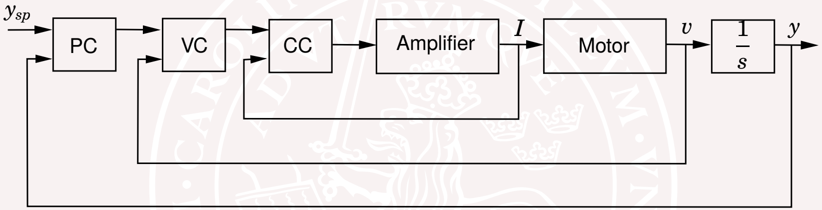
$$P_1(s) = \frac{1}{s+1} \qquad P_2(s) = \frac{1}{(s+1)^3} \qquad P(s) = P_1 P_2(s) = \frac{1}{(s+1)^4}$$

- PI Controller outer loop $K = 0.37$, $T_i = 2.2$
- P Controller inner loop $K = 5$, PI outer $K = 0.55$, $T_i = 1.9$



Examples of Cascade Control

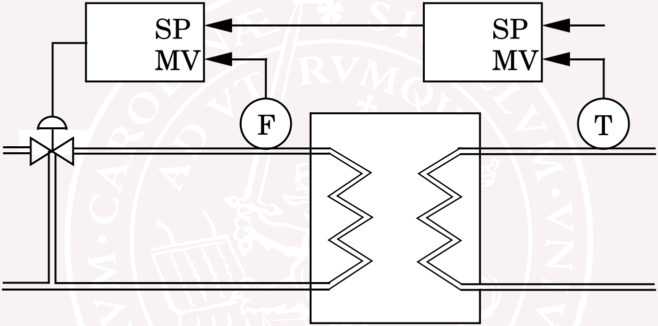
Motordrive



Three cascaded loops

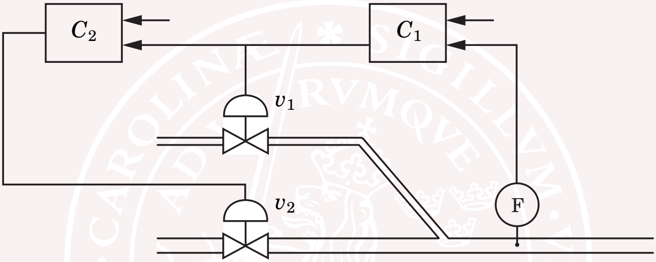
- Current loop
- Velocity loop
- Position loop

Control of Heat Exchanger



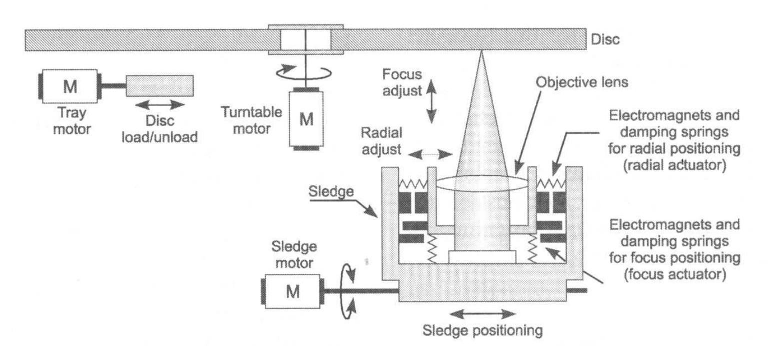
- Output temperature is the primary variable
- Three-way valve
- Flow measurement is used to mix primary water

Mid-Range Control - Two Valves



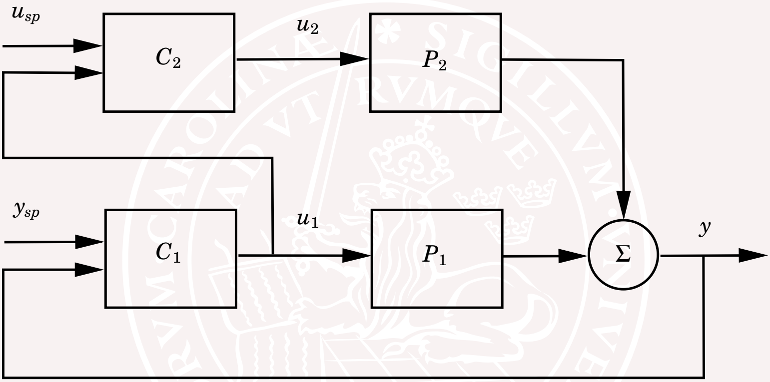
- Use parallel actuators to obtains high actuation precision and wide actuation range
- Fine actuation through v_1, course actuation through v_2
- Try to keep the valve v_1 in the mid range
- Course actuation can also be discrete (chillers)
- Separate time scales

CD Player - Two Actuators



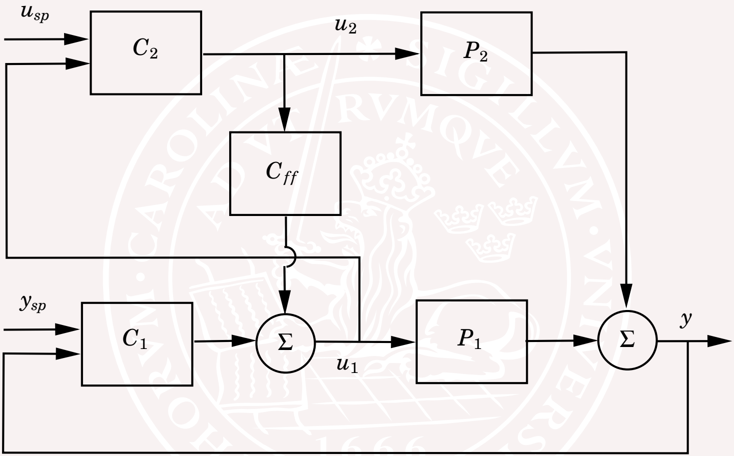
- The optical package is light with a voice coil drive
- The sledge drive is slower and provides the coarse motion

Midranging – Basic



- P_1 provides precise control of y but the range of u_1 is limited
- P_2 attempts to control the output y so that the control signal u_1 is in mid range
- Windup protection

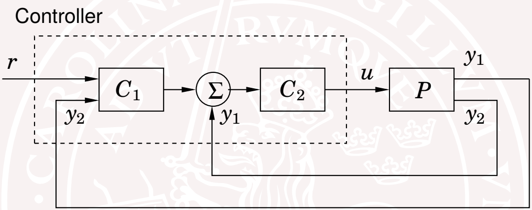
Midranging with Coupling



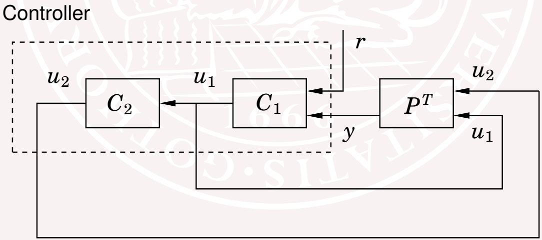
- Similar to basic scheme but with coupling that tells first loop what second loop is doing
- Windup protection

Duality

Cascade Control

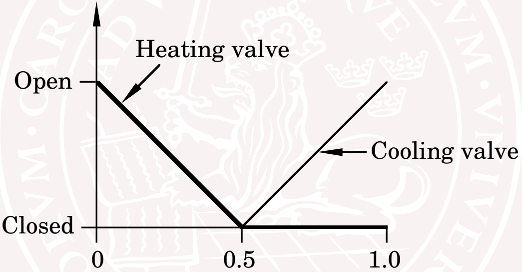


Midrange Control



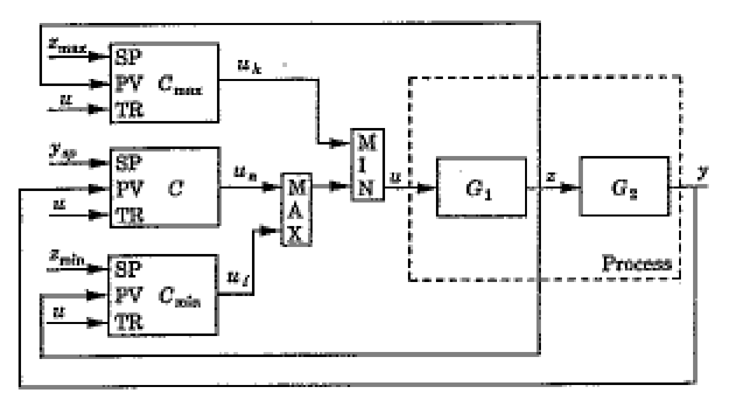
Split Range Control

Usign one controller for two actuators, typically heating and cooling.



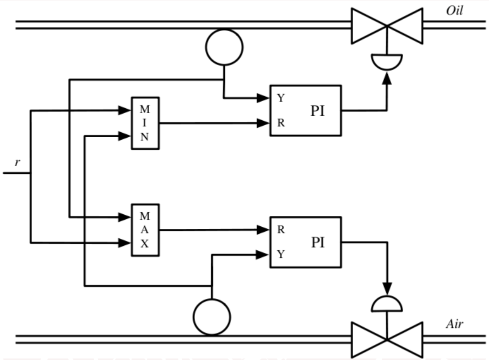
Nonlinearity can be asymmetric

Selector Control



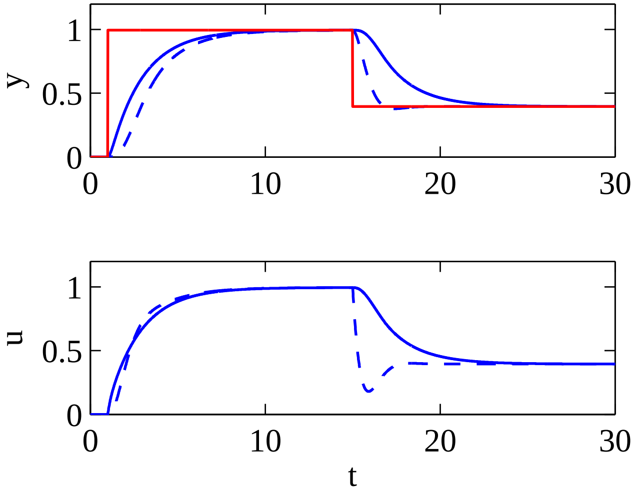
- Control with constraints
- Mixing objectives
- Elegant way do handle logic
- Other selectors: median 2 of 3
- Windup protection via tracking input!!
- Stability analysis

Air-fuel Controller



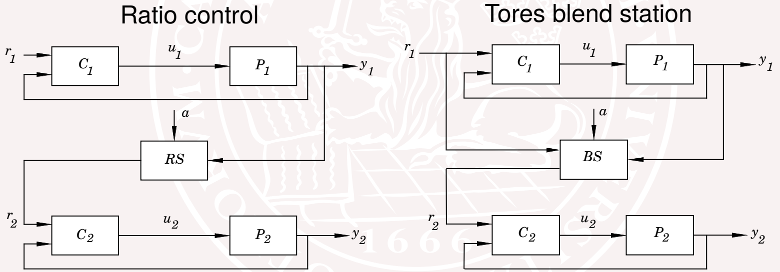
- Make sure the boiler always runs with excess air
- Discuss increase and decrease in power demand
- Compare with use of logic
- Windup protection

Air-fuel Controller - Always air excess



Ratio Control

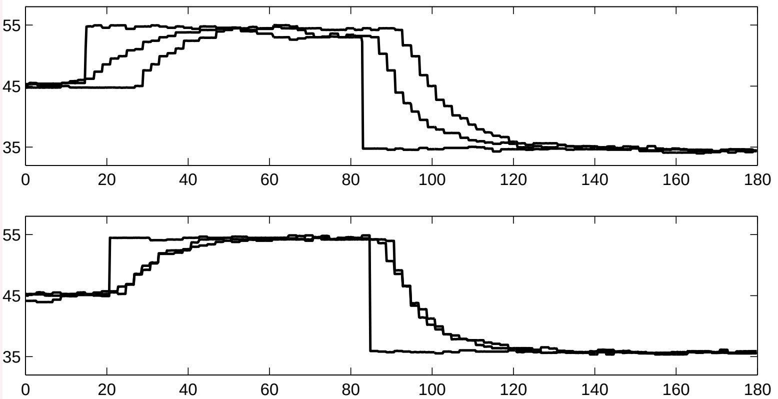
A common problem is to mix flows in given proportions. Ratio controllers is one way to do this selector control is an alternative (see Air-Fuel control later)



$$r_2 = a y_1$$

$$r_2(t) = a\left(\gamma r_1(t) + (1-\gamma) y_1(t)\right)$$

Tores Blend Station (real data)



Top curves ratio control bottom curves blend station

Bottom-Up Architectures

Theme: Brick by brick.

Parameter Variations

- Robust control

  Find a control law that is insensitive to parameter variations

- Gain scheduling

  Measure variable that is well correlated with the parameter variations and change controller parameters

- Adaptive control

  Design a controller that can adapt to parameter variations

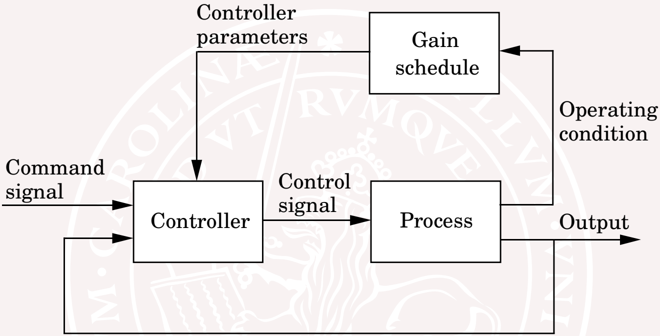
- Many different schemes

  Model reference adaptive control
  The self-tuning regulator
  L_1 adaptive control (later in LCCC)

- Dual control

  Control should be directing as well as investigating!

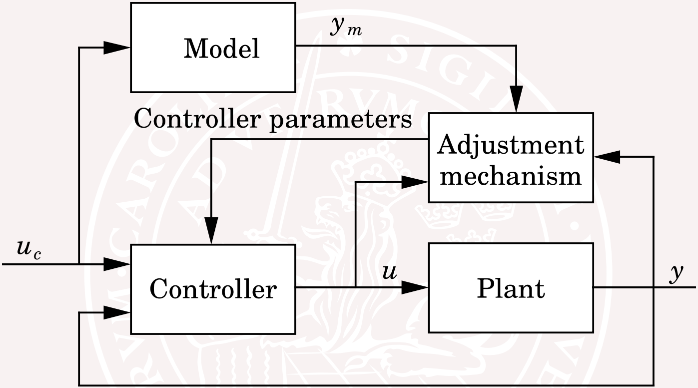
Gain Scheduling



Example of scheduling variables

- Production rate
- Machine speed
- Mach number and dynamic pressure
- Room occupancy

Model Reference Adaptive Control



The MIT rule

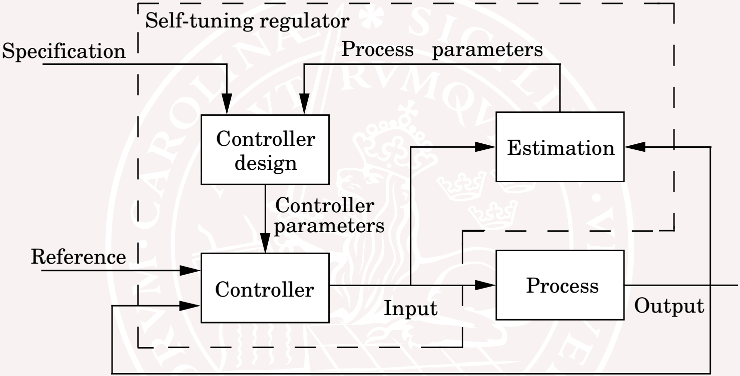
- Idea of model following
- MIT rule: $\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}$
- Many other rules

Recursive Least Squares

$$
\begin{aligned}
y_{t+1} &= -a_1 y_t - a_2 y_{t-1} + \cdots + b_1 u_t + \cdots + e_{t+1} \\
&= \varphi_t^T \theta + e_{t+1} \\
\hat{\theta}_t &= \hat{\theta}_{t-1} + K_t (y_t - \varphi_t \hat{\theta}_{t-1}) \\
K_t &= P_{t-1} \varphi_t (\lambda + \varphi_t^T P_{t-1} \varphi_t)^{-1} = P_t \varphi_t
\end{aligned}
$$

- Many versions: directional forgetting, resetting, ...
- Square-root filtering (good numerics!)

The Self-Tuning Regulator (STR)

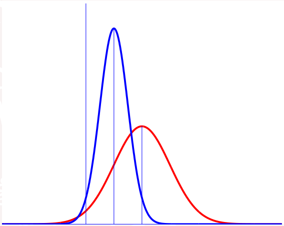


- Certainty equivalence
- Many estimation and control design methods

Minimum Variance Control and the STR

$$y_{t+1} + a y_t = b u_t + e_{t+1} + c e_t$$

$$u_t = \frac{1}{b}(a y_t - c e_t) = \frac{a-c}{b} y_t$$



In general $A(q)_t y = B(q) u_t + C(q) e_t$

$$u_t = \frac{a(q)-c(q)}{b(q)}(y_t - r_t)$$

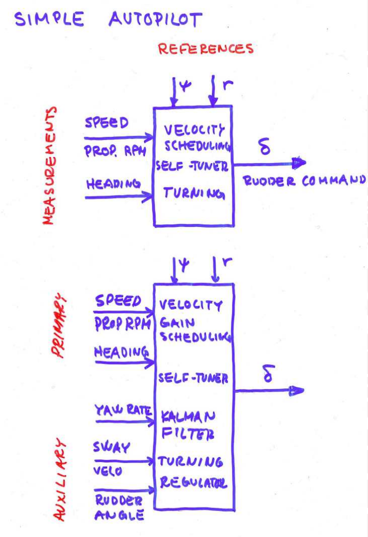
The self-tuning controller

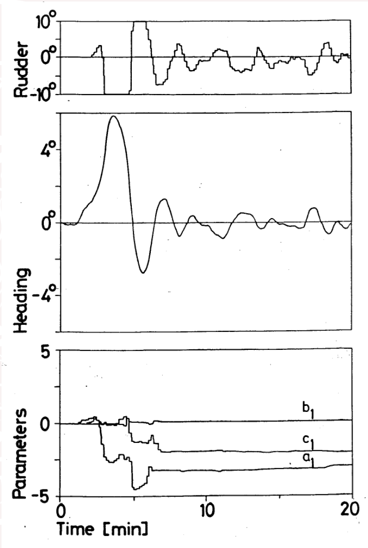
$$y_t = \beta(u_t - \theta y_t) + \epsilon$$

$$u_t = \mathrm{sat}\Big(\hat{\theta}(y_t - r_t)\Big)$$

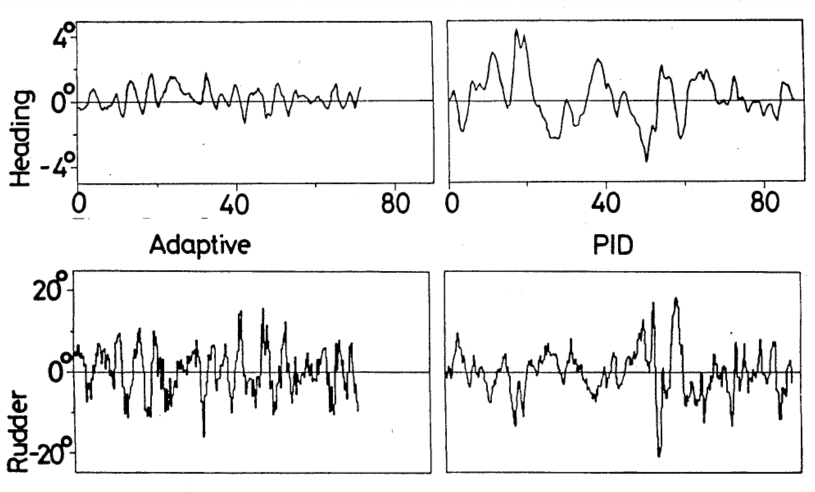
Estimate θ by least squares for fixed β, $0.5 < \beta/b < \infty$, $B(z)$ stable + order conditions. Local stability: real part of $C(z)$ positive for all zeros of $B(z)$

Ship Steering - Clas Källström





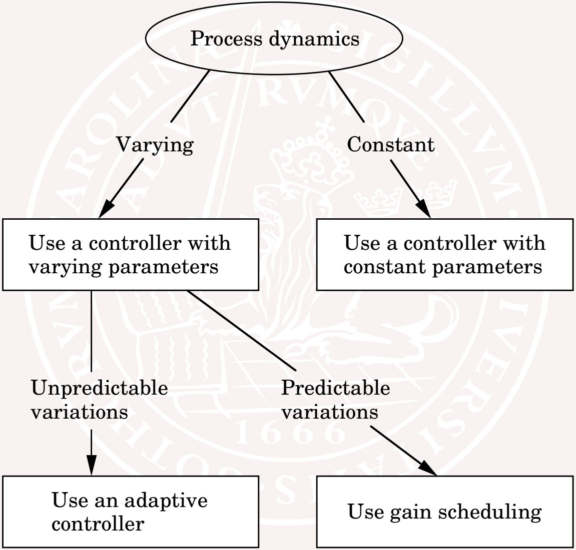
Ship Steering - Performance



Steermaster

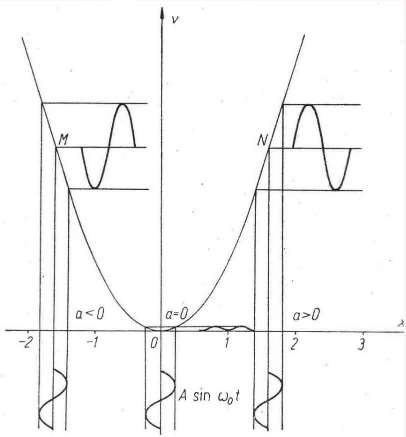
NORTHROP GRUMMAN

Using Different Methods



Extremal Seeking - Optimization

- Draper-Lee Optimize jet engine performance
- Bacteria searching for light or food
- Optimize production rate or quality
- Many perturbation and optimization techniques can be used



Perturb input, correlate with output

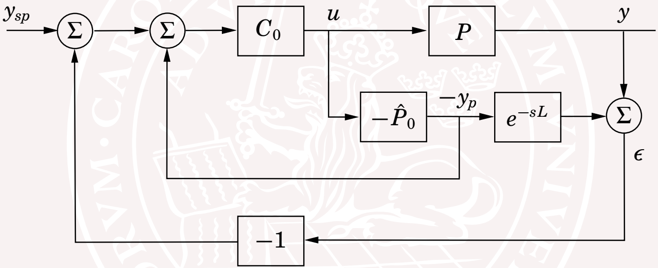
Bottom-Up Architectures

1. Introduction
2. Basic Architectures
3. Large Parameter Variations
4. Otto J. M. Smith's Specials
   - The Smith Predictor
   - Tore's PPI Controller
   - C_{pred} as a lead compensator
   - Posicast Control
5. Miscellaneous
6. Soft Computing
7. Summary

Theme: Brick by brick.

The Smith Predictor 1958

- O. J. M. Smith UC Berkeley 1958
- Idea: Use model to create output without delay
- $P(s) = P_0(s)e^{-sT}$



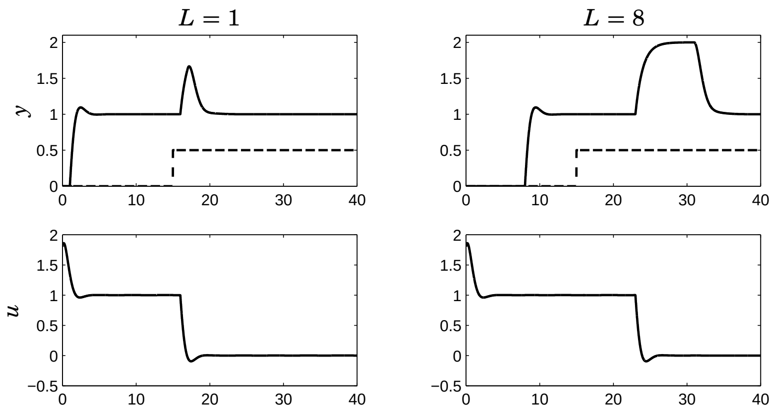
Controller and closed loop transfer function

$$C(s) = \frac{C_0}{1 + C_0 \bar{P}_0 (1 - e^{-sL})} \qquad T = \frac{P_0 C_0}{1 + P_0 C_0} e^{-sL}$$

TAT: When can you expect trouble?

https://www.youtube.com/watch?v=1zKLAkF9dwg start 7:30

Set Point and Disturbance Responses

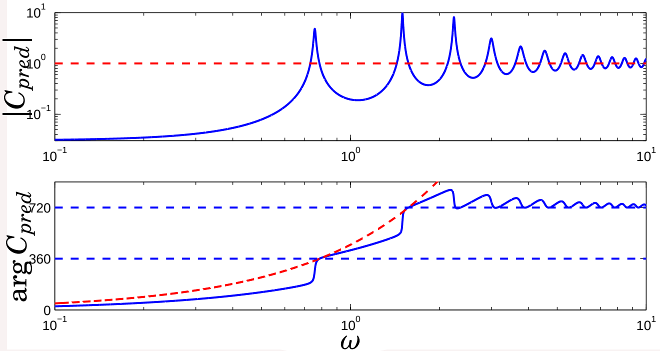


Process $P(s) = e^{-sL}/(s+1)$, $L = 1$ and 8
PI controller designed for $\omega_c = 2$ and $\zeta_c = 0.7$

Origin of Phase Advance

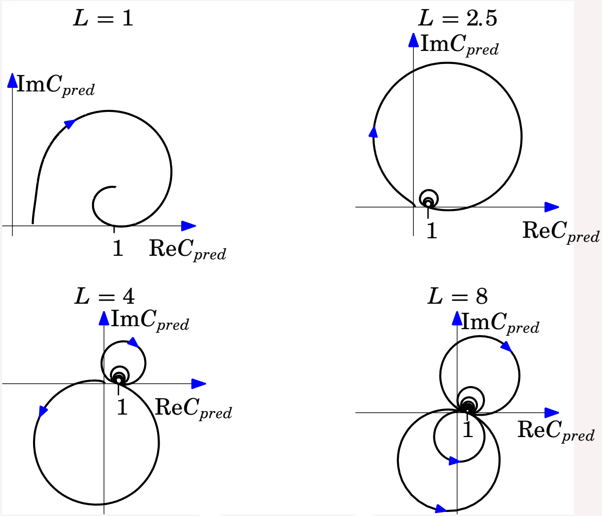
$$C = \frac{C_0}{1 + C_0 \bar{P}_0 (1 - e^{-sL})} = C_0 C_{pred}, \qquad C_{pred} = \frac{1}{1 + C_0 \bar{P}_0 (1 - e^{-sL})}$$

Near the crossover frequency $C_0 P_0 \approx -1$ and $C_{pred} \approx e^{sL}$



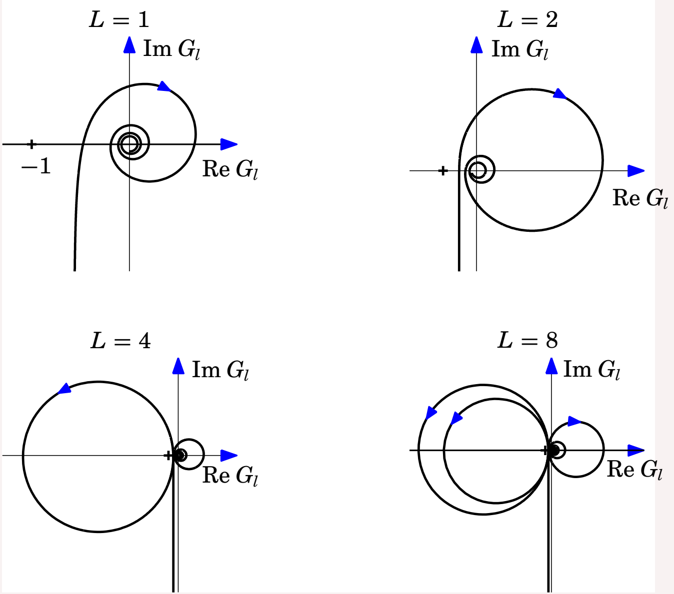
Bode plot of $C_{pred}(s)$ and e^{sL} for $L = 8$

Nyquist Plots of Predictor



Unstable oscillatory modes can give phase advance!

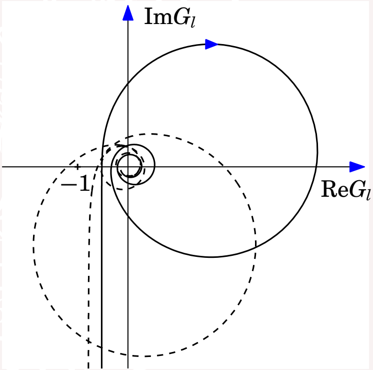
Nyquist Plot of Loop Transfer Function



The Danger of Phase Advance

Smith predictor of loop transfer function for $L = 2$ (full) and for an increase of the time delay by 30%

The delay margin is the percentage increase of the time delay that makes the system unstable. For $L = 2$ plots to the left the delay margin is 27%. Notice that it is the second peak that creates instability. The delay margin is only 7% for $L = 8$.



PI and Smith Predictor for Pure Delay

PI control full lines

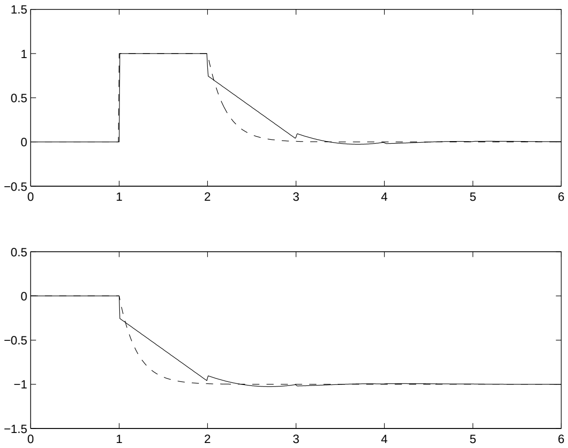
$$G = e^{-sL}$$

$$k = 0.25$$

Smith predictor dashed lines

$$G_p G_c = \frac{5}{sL}$$

$$\frac{T_{cl}}{L} = 0.2$$



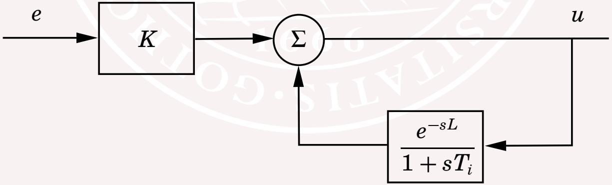
Tore's PPI Controller 1996

Design a PI controller for FOTD process by canceling the process pole and choosing the gain to give a closed loop time constant T_{cl}. The controller then becomes

$$C(s) = \frac{1+sT}{K_p s T_{cl}} \frac{1}{1 + \frac{1}{sT_{cl}}(1-e^{-sL})}$$

$$U(s) = \Big(k_p + \frac{k_i}{s}\Big)E(s) - \frac{1}{sT_{cl}}\big(1-e^{-sL}\big)U(s)$$

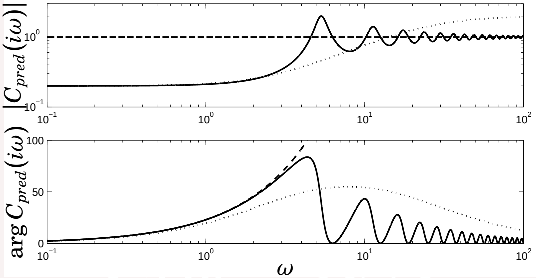
Predicts by accounting for past control actions that have not yet shown up in the output. Better than to predict by derivatives of the output! Particularly simple for $T_{cl} = T = T_i$ (Foxboros version)



Phase Advance of PPI Predictor

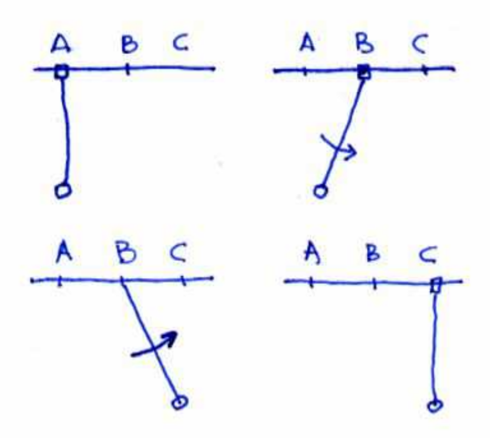
$$C_{pred}(s) = \frac{1}{1 + \frac{1}{sT}(1 - e^{-sL})} \approx \frac{T}{T + L - sL^2/2}$$

$$\approx \frac{T}{T+L}(1 + sT_{pred}), \qquad T_{pred} = \frac{1}{2}\frac{L^2}{T+L}$$



- C_{pred} is a nice phase advance network, better phase advance than with derivative action (dotted)!

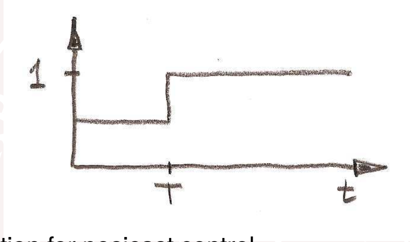
Moving a Hanging Container - Posicast Control



- O. J. M. Smith Posicast control of damped oscillatory systems, Proc. IRE. (45) 1957, 1249-1255
- Has been used succcessfully for cranes and micro systems
- What is the transfer function?

Transfer Function

Step response



- Transfer function for posicast control

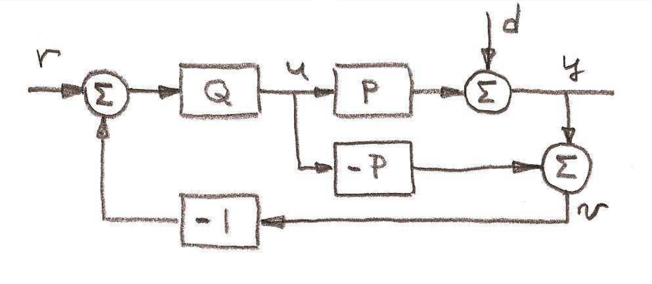
$$G_{ff}(s) = \frac{1}{2}\big(1 + e^{-sT}\big).$$

- Sinusoidal signals of frequencies $\omega = \omega_0, 3\omega_0, 5\omega_0 \ldots$, where $\omega_0 = 2\pi/T$.
- Nonrational transfer function
- Easy to implement posicast control using digital control.

Bottom-Up Architectures

Theme: Brick by brick.

Internal Model Control IMC



- The signal v does not depend on the control actions
- The signal v represents an output equivalence of all disturbances acting on the process
- Disturbance attenuation is done by design of Q
- TAT: What is the loop transfer function?
- Ideal response if $Q = P^{-1}$, or approximately $Q = P^{\dagger}$
- TAT: P must be stable! Why? Modifications?

A Pure Delay Process

$$P(s) = e^{-sL} \quad Q(s) = 1 \quad T(s) = e^{-sL} \quad L(s) = \frac{e^{-sL}}{1 - e^{-sL}}$$

Frequency response of loop transfer function

$$L(i\omega) = -\frac{1}{2} - i\frac{\sin \omega L}{2(1 - \cos \omega L)} = -\frac{1}{2} - i\frac{1}{\tan(\omega L/2)}$$

The Gang of Four

$$S(s) = 1 - e^{-sL} \qquad PS(s) = e^{-sL}(1 - e^{-sL})$$
$$CS(s) = e^{-sL} \qquad T(s) = e^{-sL}$$

- Nyquist plot (Discuss!)
- Gain margin $g_m = 2$ phase margin $\phi_m = 60°$
- Maximum sensitivities $M_s = 2$ and $M_t = 1$
- Looks OK BUT!!!

Delay Margin

Using standard criteria the system looks robust, but what about parametric changes. Assume that the time delay of the process changes from L to $L + \delta$.

$$
\begin{aligned}
e^{-s(L+\delta L)} &= e^{-sL}e^{-s\delta L} = e^{-sL} + e^{-sL}(e^{-s\delta L} - 1) \\
\Delta P(s) &= e^{-sL}(e^{-s\delta L} - 1)
\end{aligned}
$$

Hence $|\Delta P(i\omega)| = |e^{-i\omega\delta L} - 1|$. The stability criterion

$$
\frac{|\Delta P(i\omega)|}{|P(i\omega)|} = |e^{-i\omega\delta L} - 1| < \frac{1}{|T(i\omega)|} = 1.
$$

is not satisfied for any $\delta L > 0$ because the left-hand side is 2 for some ω and the right hand side is 1. Hence unstable for arbitrary small perturbation in the time delay. Sketch Nyquist plot.

Complementary Filtering

- A technique to combine information from several sensors
- A precursor to Kalman filtering
- Useful in its own right
- Requires only models of sensor systems

Signal model (y_1 slow but accurate, y_2 fast but drifting)

$$y_1 = x + n_1, \qquad y_2 = x + n_2$$

Filter for recovering the variable x

$$X_f(s) = \frac{1}{s+1} Y_1(s) + \frac{s}{s+1} Y_2(s)$$

Choose G_1 as low pass filter, G_2 then becomes high pass.

A Kalman Filter Solution

Model the measured value x_1 and the drift of the second sensor as unknown constants

$$y_1 = x_1 + n_1, \qquad y_2 = x_1 + x_2 + n_2, \qquad \dot{x}_1 = 0, \qquad \dot{x}_2 = 0$$

The Kalman filter

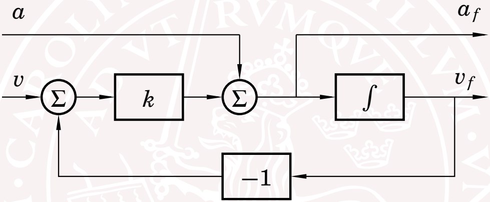
$$\frac{d}{dt}\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} k_{11}(y_1 - \hat{x}_1) + k_{12}(y_2 - \hat{x}_1 - \hat{x}_2) \\ k_{21}(y_1 - \hat{x}_1) + k_{22}(y_2 - \hat{x}_1 - \hat{x}_2) \end{pmatrix}$$

After some calculations

$$\hat{X}_1(s) = \frac{k_{11}s + k_{11}k_{22} - k_{12}k_{21}}{s^2 + (k_{11} + k_{22} + k_{12}) + k_{11}k_{22} - k_{12}k_{21}} Y_1(s) + \frac{k_{12}s}{s^2 + (k_{11} + k_{22} + k_{12}) + k_{11}k_{22} - k_{12}k_{21}} Y_1(s)$$

Velocity and Acceleration Measurements

Determine estimates of velocity and acceleration from measurements of the same quantities



$$V_f(s) = \frac{s}{s+k} V(s) + \frac{k}{s+k} V(s)$$

$$V_f(s) = \frac{1}{s+k} A(s) + \frac{k}{s+k} V(s)$$

$$A_f(s) = \frac{s}{s+k} A(s) + \frac{ks}{s+k} V(s)$$

Use of an accelerometer and a rate gyro to determine tilt for the Segway is a similar problem.

Bengt Sjöberg Saab

I tidigare projekt hade man ju stött på behovet av filter, speciellt för att ta hand om brusiga radarsignaler. Man upptäckte då att tex antennvinklarna från egen flygradar mot ett radarföljt mål på grund av målets och det egna flygplanets fart, accleration och rotation vairerade starkt på grund av grundläggande kinematiska samband. ... Jag lärde mig ju snart att inse att dessa praktiska åtgärder helt enkelt bottnade i att man måste tvinga sina filtrerade variabler att satisfiera en modell för sambanden mellan accelerationer, farter och positioner hos eget flygplan och mål. Dessa modellsamband sattes då upp i vektorform varvid det oftast visade sig praktiskt att arbeta i olika koordinatsystem som oftast roterade. ... På detta sätt uppstod vad jag då efter viss vånda valde att kalla “komplementära filter”.

Complementary filtering is a well established field

Complementary Filters or Observers

Both

- Generate estimates of signals that are not measured directly
- Unify information from different sensors (sensor fusion)
- Can be optimized if noise information is available

Complementary filters

- Require models of sensor systems only not process dynamics

Observers

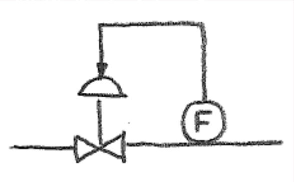
- Require models of process dynamics that typically involves command signals.
- Process inputs provide phase lead.

Linearization

- Both sensors and actuators can be linearized in open loop by feedforward
- Feedback can also be used effectively when sensors are available

Open loop

Feedback

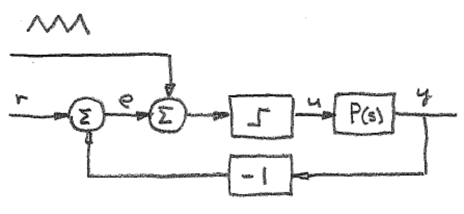


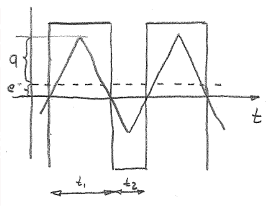
Process model $y = f(u)$
Feedfoward: $u = f^{-1}(u_c)$
Hence y=$f(f^{-1}(u_c)) = u_c$
Requires model
Sensitivity =1

Requires sensor
Less sensitive

Linearization by Using Jitter Signals

Mechanical and electrical jitter





When a triangular jitter signal is added to the error signal the average relay output is

$$\frac{T_p}{4}\left(1+2\frac{e}{a}\right)-\frac{T_p}{4}\left(1-2\frac{e}{a}\right)=\frac{e}{a}$$

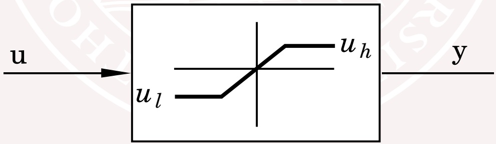
The combination of a relay with a jitter signal thus acts like a saturated linearity

Limiters

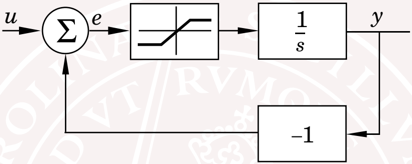
Limiters are used to avoid windup and to limit levels and rates for command signals (never ask the system to do more than it can). Kurt Nicolin Asea (legendary Swedish industrialist): “To add more workorders to an overloaded production unit increases confusion but not productivity.”

- Avoid actuator saturation
- Match demant to process capabilities
- Windup protection

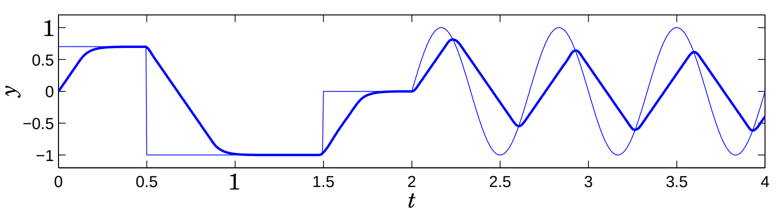
A simple limiter



Simple Rate Limiter

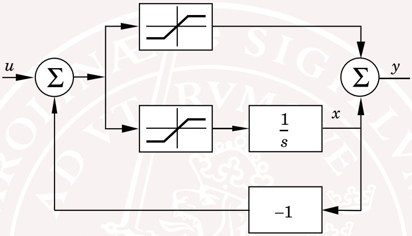


Notice that it creates phase lag

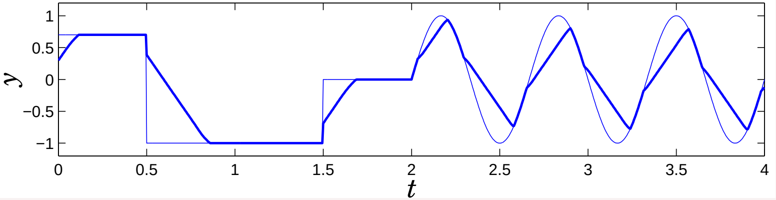


The JAS Gripen problem show video

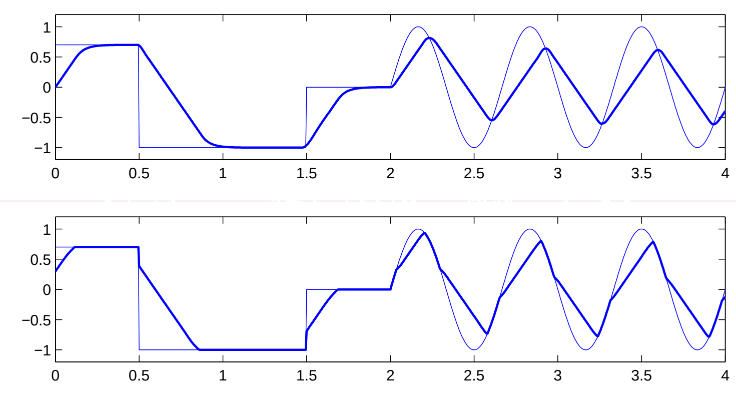
Jump and Rate Limiters



Less phase lag than with rate limiter



Rate and Jump-and Rate Limiters

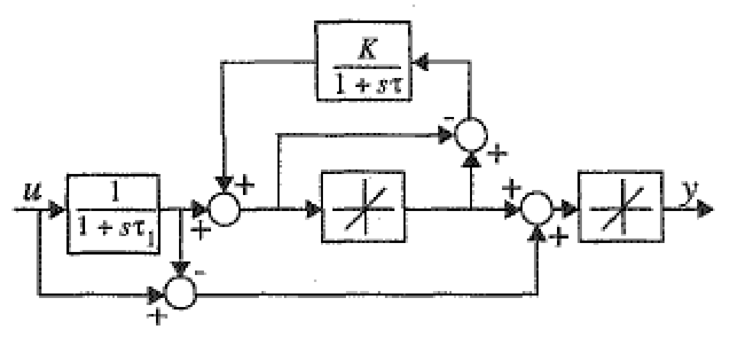


Rate limiters give phase lag, JAS Gripen

Jump-and-rate limiters are commonly used in power systems

Lars Rundqwist's JAS Gripen Fix

- Assignment of authority for manual and automatic control
- Rate saturation in hydraulic servos causes phase lag
- Commissioning of flight control systems
- Rundqwist's Rlim inspired by windup protection



Bengt Sjöberg Strikes Again

- Phone calls in the night
- Rlim can be improved!
- Project opportunities

Bottom-Up Architectures

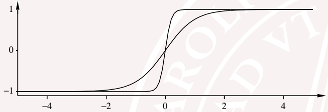
Theme: Brick by brick.

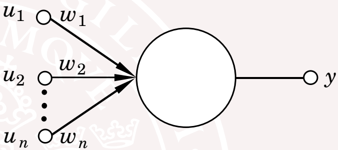
Soft Computing

- Technical and biological systems
- Cybernetics - Control and communication in the animal and the machine. Wiener 1948, Ashby 1956
- Neural systems and the Perceptron (Neural Network)
  McCulloch Pitts 1943
  Rosenblatt 1958, Widrow-Hoff 1961 (Addaline)
  Collapse due to Minsky and Papert 69
  Survivors: Anderson, Grossberg, Kohonen
- Emergence of Artificial Intelligence
  Dartmouth Conference 1956, Minsky,
- Revival of Neural Networks
  Hopfield 1982 and The Snowbird Conference
  The parallel distributed process group
- Neuro Fuzzy - Zadeh and Japan
- IBM Watson (Kasparov 1997, Jeopardy 2010)
- Autonomous Cars
- Machine Learning

Neural Networks

Artificial neuron: $y(t) = f\left(\sum a_i u_i(t)\right)$



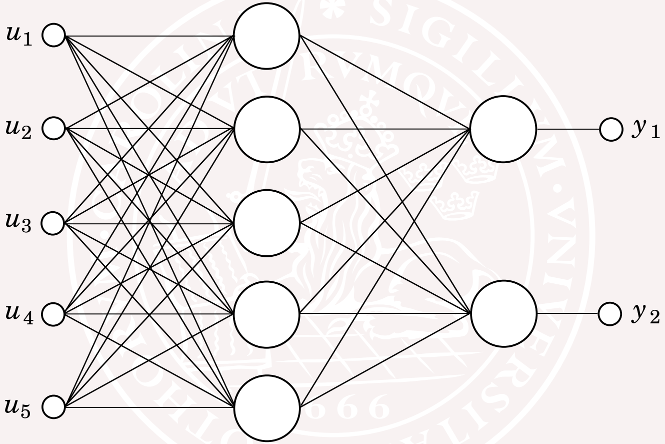


Kolmogorov's Theorem: There exist fixed continuous increasing functions $\phi_{ij}(x)$ so that any continuous function $f \in R^n$ can be written in the form

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} g_i\Bigl(\sum_{j=1}^{n} \phi_{ij}(x_j)\Bigr)$$

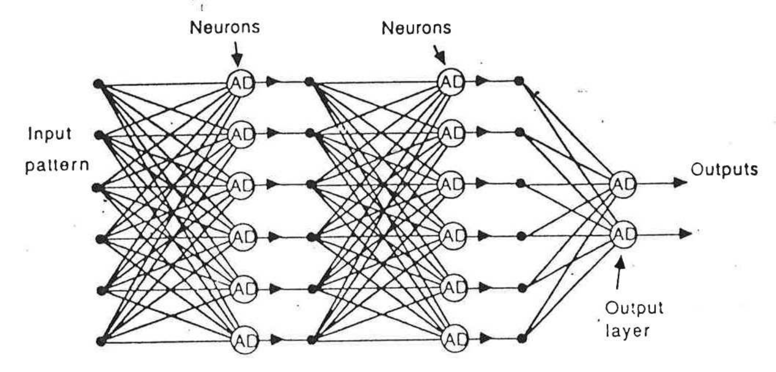
A function of many variables $f : R^n \to R^n$ can be represented as a combination of MISO functions $f : R^n \to R$

Neural Networks



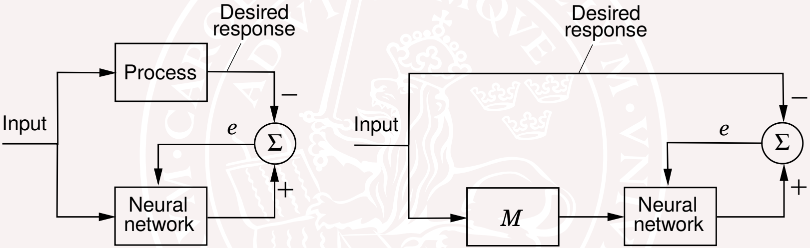
A nonlinear function with a learning mechanism!

Multilayer Networks



- Feedforward nets
- Nets with feedback (Kohonen)
- Nets with feedback and dynamics (Boltzmann)

Training Neural Networks



Both functions and inverse functions can be generated

Soft Computing and Control

- Assume that all states can be measured or estimated using simple estimators
- The control law is then a nonlinear function $f : R_x \to R_u$
- Difficult to represent functions of many variables make neural networks and fuzzy attractive
- Quantize the states N^n large for large n
- Use soft computing to construct or learn the control law
- Typical example: swinging up a pendulum or balancing a pole

$$u(x_1, x_2) = 2a \sin x_1 + b x_2 F(x_1, x_2) \cos x_1$$

$$F(x_1, x_2) = \frac{2a+1}{4a}(2a \cos x_1 - 1) + \frac{x_2^2}{2}.$$

- Michie Chambers Boxes 1968 - a never-ending story
- David Russel, The BOXES Methodology - Black Box Dynamic Control, Springer 2012

Michie - Chambers Boxes 1968

Michie, D., & Chambers, R. A. (1968). Boxes: An Experiment in Adaptive Control. In E. Dale. & D. Michie (Eds.), Machine Intelligence 2. Edinburgh: Oliver and Boyd.

- Measure $\theta, \dfrac{d\theta}{dt}, x, \dfrac{dx}{dt}$
- Quantize crudely: position and angle 5 levels, velocities 3 levels, gives 225 boxes
- On-off control
- Scoring method LL: left life, RL right life, LU left usage, RU right usage, ...
- Many followers

  Quantization and smoothing

  Neural networks, many versions

  Genetic optimization

  Fuzzy, neural, neuro-fuzzy

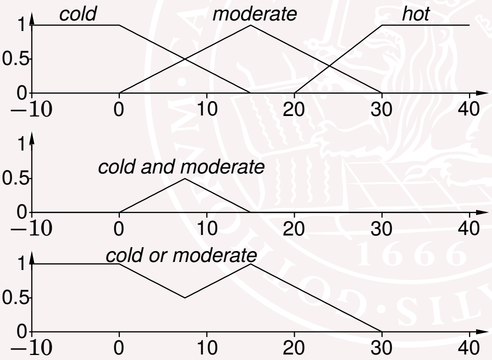
  Machine learning
- Tobias Glück https://www.youtube.com/watch?v=Lt-KLtkDlh8

Fuzzy Logic and Control

- Lotfi Zadeh Fuzzy Logic 1965
- Mamdani Fuzzy Control 1974
- F. L. Smith Fuzzy control of cement kilns 1981
- Blue Circle Cement Linkman
- Hitachi subway system 1987
- Laboratory for International Fuzzy Engineering Tokyo 1988
- Japan Society for Fuzzy Theory and Systems SOFT 1989
- Fuzzy Logic Systems Institute 1990
- Center for promotion of Fuzzy Engineering TIT 1991
- Lots of products from Japan with Fuzzy Omron, Sharp

Fuzzy Logic

- Rule based control - one way to describe nonlinearities
- Linguistic variables *high, low, medium* and membership functions
- If temperature *high* then increase flow *a little*



Fuzzy PD Controller – Rule Representation

```
Rule 1: If e is N and de/dt is P then u is Z
Rule 2: If e is N and de/dt is Z then u is NM
Rule 3: If e is N and de/dt is N then u is NL
Rule 4: If e is Z and de/dt is P then u is PM
Rule 5: If e is Z and de/dt is Z then u is Z
Rule 6: If e is Z and de/dt is N then u is NM
Rule 7: If e is P and de/dt is P then u is PL
Rule 8: If e is P and de/dt is Z then u is PM
Rule 9: If e is P and de/dt is N then u is Z
```

		de/dt		
		P	Z	N
	N	Z	NM	NL
e	Z	PM	Z	NM
	P	PL	PM	Z

Fuzzy Inference

The fuzzy statement

```
If x is A and y is B
```

is interpreted as the crisp variable

$$z^0 = \min(f_A(x_0), f_B(y_0))$$

and is equivalent to minimization of the membership functions. The linguistic variable u defined by

```
If x is A or y is B then u is C
```

`or` is interpreted as a linguistic variable with the membership function

$$f_u(x) = z^0 f_C(x).$$

Defuzzification

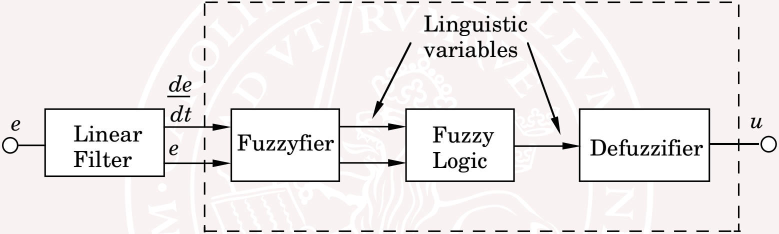
Consider a linguistic variable A with the membership function $f_A(x)$. Defuzzification by mean values gives the value

$$x_0 = \frac{\int x f_A(x) dx}{\int f_A(x) dx}.$$

Defuzzification by the centroid gives a the real variable x_0 that satisfies

$$\int_{-\infty}^{x_0} f_A(x) dx = \int_{x_0}^{\infty} f_A(x) dx$$

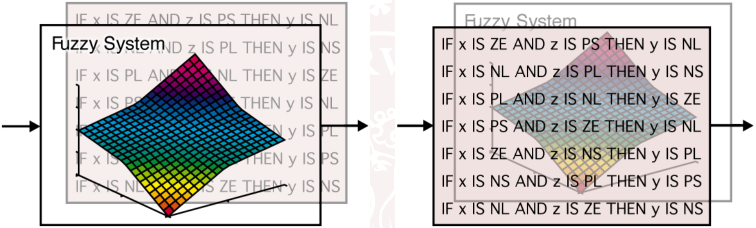
Fuzzy Control



If e large positive and de/dt large positive then u large

If e med positive and de/dt med negative then u zero

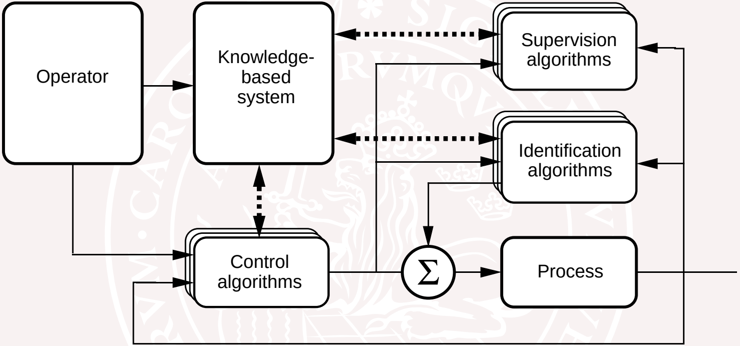
Two Views of Fuzzy Control



Picture from Karl-Erik Årzen

- Good language for translating manual operating practice (control of cement kilns)
- Snake-oil salesmen
- Software for generating nonlinearities

Intelligent Control



A knowledge bases system is used for monitoring, process supervision and switching of control and estimation algorithms.

ABB's notion of state-based control

Bottom-Up Architectures

1. Introduction
2. Basic Architectures
3. Large Parameter Variations
4. Otto J. M. Smith's Specials
5. Miscellaneous
6. Soft Computing
7. Summary

Theme: Brick by brick.

Some Useful Blocks

- PID
- Linearities
  - Low pass, band pass, high pass
  - Smith type of predictors
  - Notch filters
  - Delays
  - Posicast
- Static nonlinearities
  - Saturation
  - Selectors
  - Jump and rate
- State estimators
- Parameter estimators
- Optimizers
- Neural networks

  Nonlinear multivariable function with learning mechanism

Summary

- Layering, abstraction and formal definition of building blocks for control systems is a major research issue. It may bring control design to the level of VLSI design!
- A rich collection of methods and ideas
- Generalized integral control
- Cascade control and Midranging (duals)
- Selectors
- Delay related, Smith predictors, posicast
- Internal model control
- Special course on adaptive control