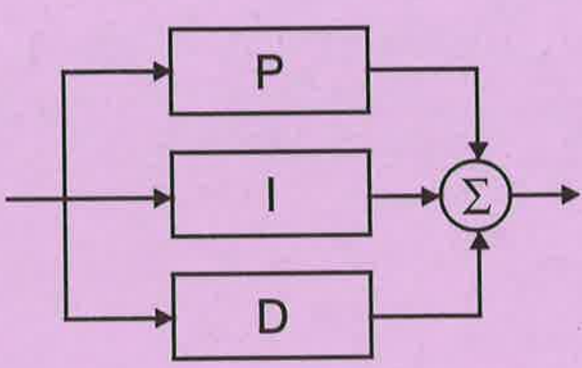
PID och Fuzzy

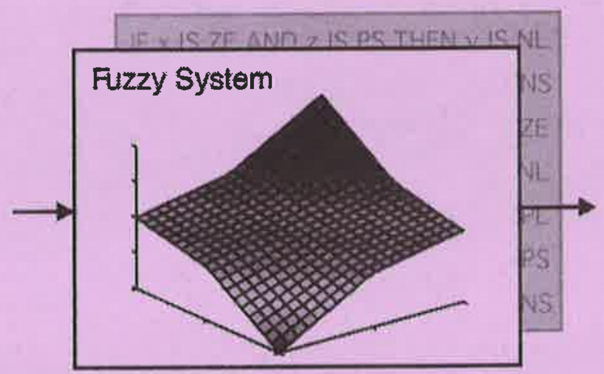
Industrikurs i Lund 10 juni 1998

Åström, Karl Johan; Hägglund, Tore

PID



och



Fuzzy

Industrikurs i Lund

10 juni 1998

Institutionen för Reglerteknik
Lunds Tekniska Högskola

PID och Fuzzy

Industrikurs i Lund 10 juni 1998

Program

09.00	Samling och kaffe
09.30	Inledning
09.45	PID-regulatorn
10.45	PID-regulatorns inställning
11.45	PID-regulatorns användning
12.30	Lunch
13.30	Fuzzy-logik och fuzzy-system
14.30	Fuzzy-reglering
15.15	Kaffe
15.45	Fuzzy-olinjäriteter
16.30	Diskussioner, sammanfattning
17.00	Kursen avslutas

Inledning

I samband med Reglermöte '98 ger vi i år en industrikurs med titeln "PID- och Fuzzy-reglering". Kursens innehåll täcker både den välkända PID-regulatorn och den relativt nya Fuzzy-tekniken.

PID-regulatorn har funnits i över femtio år. På senare år har den fått ny uppmärksamhet i samband med utvecklingen av automatinställning. Detta har lett till större insikt i regulatorns funktion, fördelar och begränsningar. Det har också lett till nya och bättre inställningsmetoder.

I kursen kommer vi att beskriva PID-regulatorns funktion och arbetssätt. Relativt stor vikt kommer att läggas vid PID-regulatorns inställning, både manuell och automatisk. Slutligen kommer vi att ta upp PID-regulatorns användning som byggblock vid instrumentering. Detta inkluderar grundläggande kopplingar som kaskadkoppling, framkoppling, kvotreglering, dödtidskompensering och väljarlogik.

Fuzzy-reglering är en metod för att implementera olinjära och multivariabla regulatorer som har tilldragit sig mycket intresse under de senaste åren. Metoden bygger på att regulator och/eller processmodell beskrivs av en uppsättning "If … then …"-regler, där varje regel ger en kvalitativ beskrivning av systemets lokala beteende runt en viss arbetspunkt.

Kursen behandlar fuzzy-logik och mängdlära, olika typer av fuzzysystem, fuzzy-reglering och modellering, fuzzy-system för funktionsapproximering samt relationer till neuronnät, s.k. "neuro-fuzzy"-system.

Föreläsare:

Karl Johan Åström, Tore Hägglund, Karl-Erik Årzén

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Box 118
221 00 LUND

046 - 222 87 80
control@control.lth.se
http://www.control.lth.se

Department of Automatic Control
Lund Institute of Technology

"PID och Fuzzy"

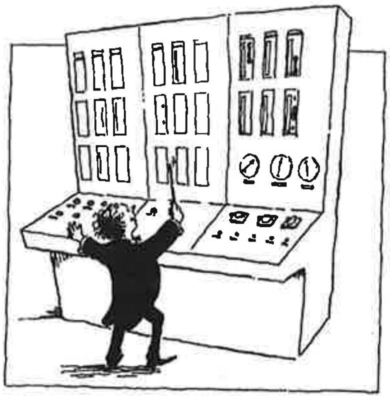
Inledning

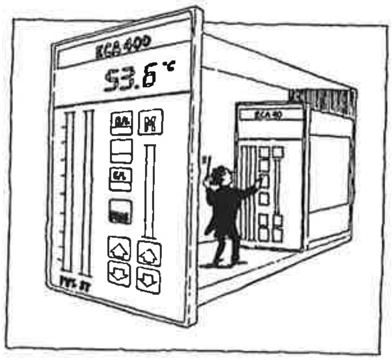
Mål:

- Du skall veta hur regulatorerna fungerar
- Du skall kännna till möjligher och begränsningar
- Du skall ha fått några ideer om förbättring av produkter och anläggningar

Utnyttja tillfället att bygga ut ditt kontaktnät, deltagarna har stor erfarenhet och de representerar ett brett spektrum av industrin.

Högre Automationsnivåer





Kvalitetskraven driver utvecklingen

PID

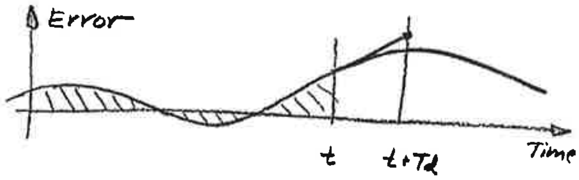
- Den äldsta regulatorn
- Den vanligaste regulatorn
- Många saker bortglömda
- Industriell praxis kan förbättras
  - SSG initiativet
  - Industriella styrsystem
- Automatinställning

Fuzzy (Oskarp reglering)

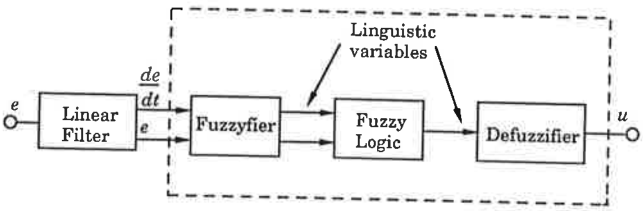
- "The new kid on the block"
- Nya begrepp
- Kontroversiell
- Många missförstånd
- Nyansering behövlig
- En skarp bild av oskarp reglering

PID

$$u(t) = k\Big(e + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de}{dt}\Big)$$



Fuzzy PD

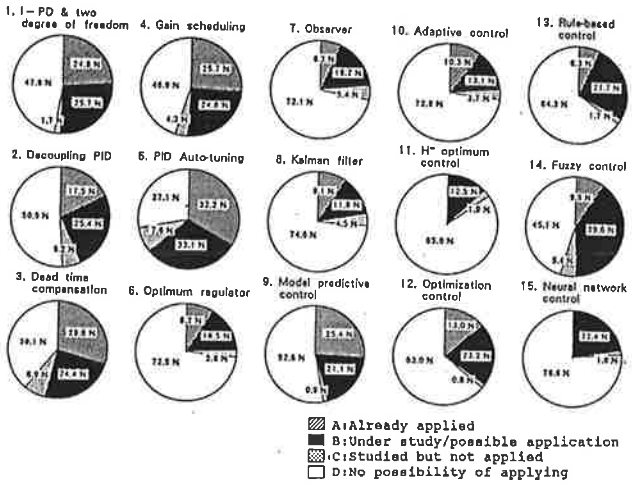


	P	Z	N
N	Z	NM	NL
Z	PM	Z	NM
P	PL	PM	Z

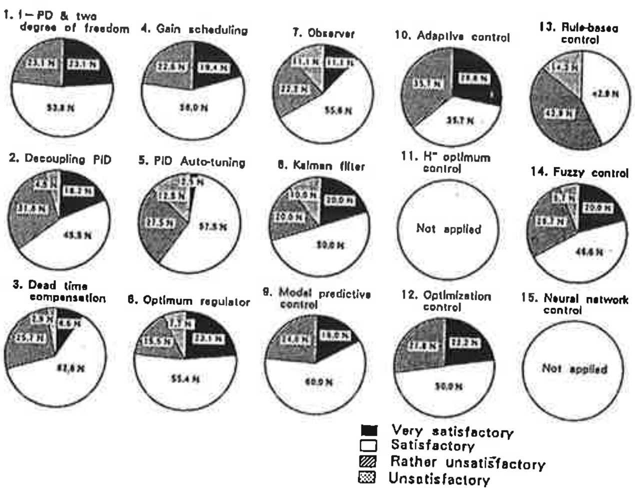
JEMIMA-Studien

- Systematisk undersökning av Japansk industri
- Regulatortyper
  - PID
  - Avancerad PID
  - "Modern control type"
  - AI typ (fuzzy, neural etc)
  - Manuell reglering
- Användning av PID i olika branscher
  - Petrokemisk PID 89% (M 8%)
  - Kemi, fibrer, film PID 93% (M 6%)
  - Rafinaderier PID 93% (M 3%)
  - Papper och massa PID 86% (M 11%)
  - Järn &Stål PID 93% (M 3%)
- Ungefär 80% av kretsarna fungerar bra

JEMIMA Studien
Metoder som prövats



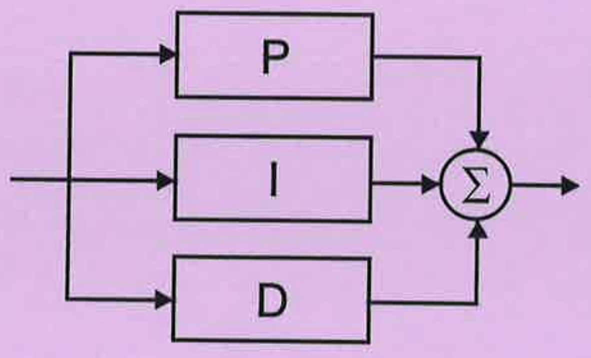
JEMIMA Studien
Samlad bedömning



Program

- PID
  - Regulatorn
  - Inställningar
  - Användning
- Fuzzy
  - Fuzzy logik och system
  - Fuzzy reglering
  - Olinjära aspekter

PID



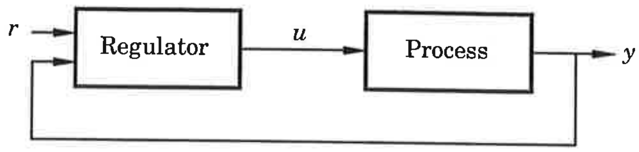
Department of Automatic Control
Lund Institute of Technology

"PID och Fuzzy"

PID-regulatorn

- Regulatorns struktur
- Regulatorns parametrar
- Varianter
- Praktiska modifieringar
- När ska P, I och D användas?

Den enkla reglerkretsen

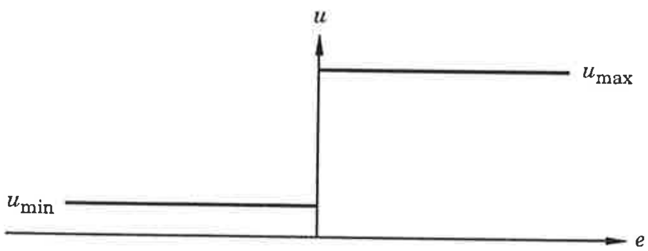


Reglerproblem:

Välj styrsignalen u så att mätsignalen y följer börvärdet r så bra som möjligt.

On/Off-regulatorn

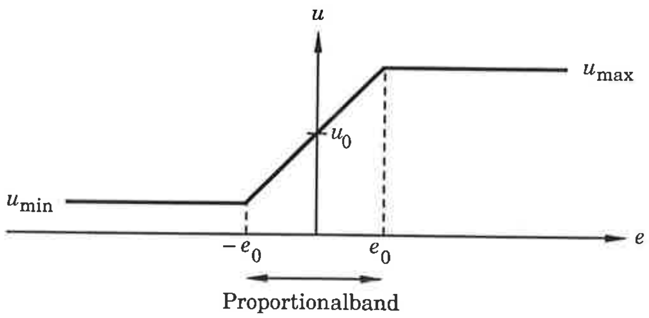
Reglerfelet: $e = r - y$



$$u = \begin{cases} u_{max} & e > 0 \\ u_{min} & e < 0 \end{cases}$$

Nackdel: On/Off-regulatorn orsakar svängningar

P-delen



$$u = \begin{cases} u_{max} & e > 0 \\ u_0 + Ke & -e_0 < e < e_0 \\ u_{min} & e < 0 \end{cases}$$

$$PB = \frac{100}{K} [\%]$$

Nackdel: P-regulatorn ger oftast ett kvarstående reglerfel

Reglerfelet vid P-reglering

Styrsignalen vid P-reglering:

$$u = u_0 + Ke$$

Reglerfelet vid P-reglering:

$$e = \frac{u - u_0}{K}$$

Möjligheter att uppnå $e = 0$:

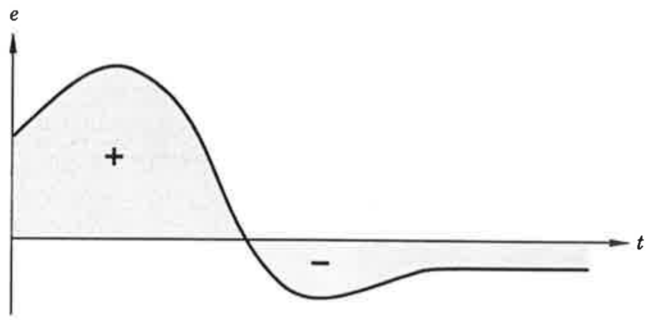
1. K är oändligt stor

2. $u_0 = u$

Lösning: Försök hitta u_0 automatiskt

I-delen

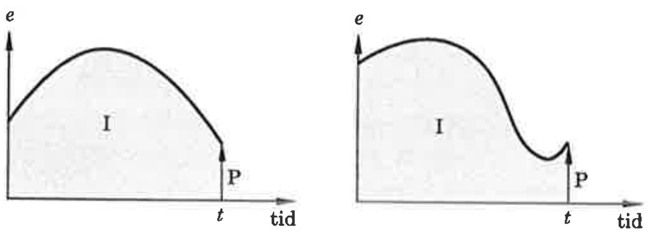
$$u = u_0 + Ke$$
$$u = K\left(\frac{1}{T_i}\int e(t)dt + e\right)$$



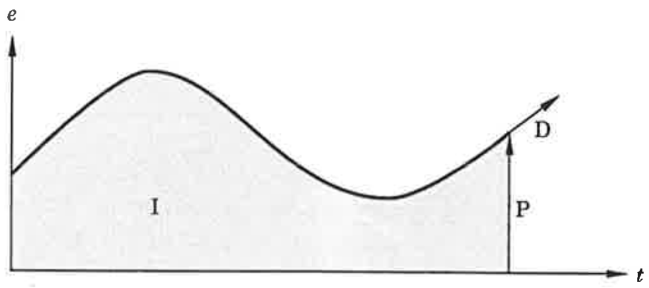
Ett stationärt fel innebär att $\int edt$ växer.
Detta innebär att u växer.
Om u växer måste också y växa.
Om y växer är inte felet stationärt.
Alltså kan vi inte ha något stationärt regler-fel när vi har en I-del i regulatorn.

PI-regulatorn predikterar inte framtiden

En PI-regulator ger samma styrsignal vid tiden t för dessa båda reglerfall:

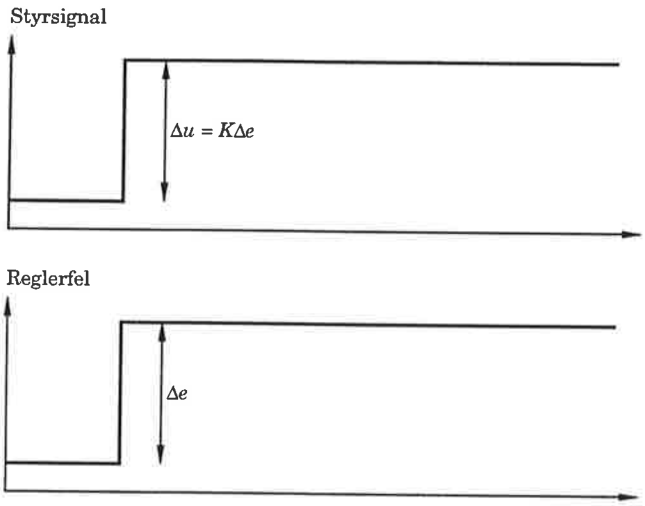


D-delen

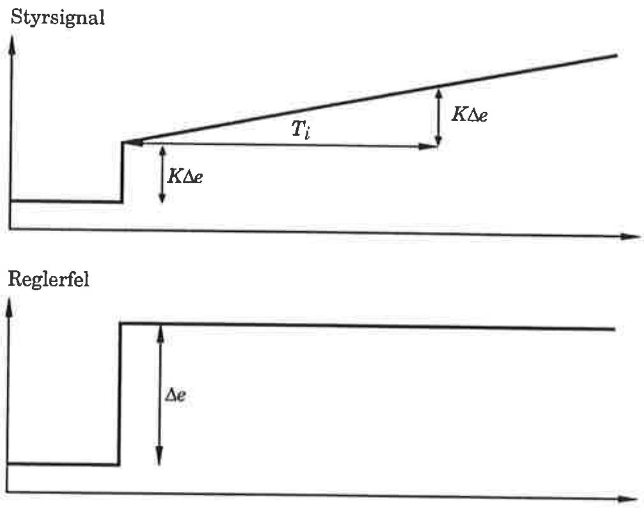


$$u = K\left(e + \frac{1}{T_i}\int e(t)dt + T_d\frac{de}{dt}\right)$$

Regulatorns förstärkning

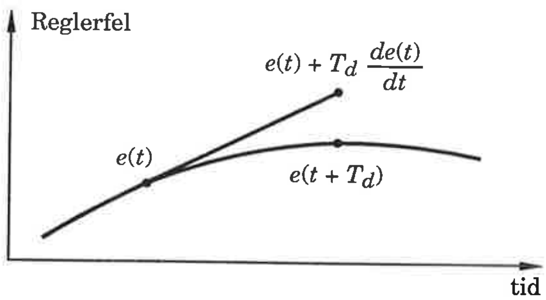


Regulatorns integraltid



T_i = Den tid det tar för regulatorn att upprepa P-steget

Regulatorns derivatatid



P:

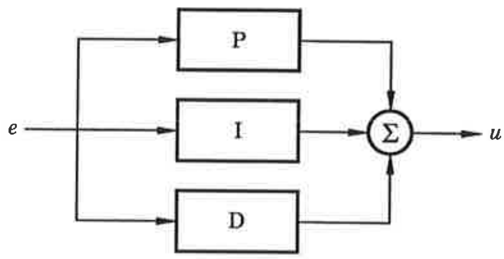
$$u(t) = Ke(t)$$

PD:

$$u(t) = K\left(e(t) + T_d\frac{de(t)}{dt}\right) \approx Ke(t + T_d)$$

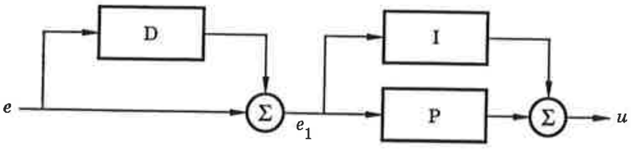
T_d = Regulatorns prediktionshorisont

Parallell form



$$u = K\left(e + \frac{1}{T_i}\int e(t)dt + T_d\frac{de}{dt}\right)$$

Serieform



$$e_1 = e + T'_d \frac{de}{dt}$$

$$u = K' \left(e_1 + \frac{1}{T'_i} \int e_1(t) dt \right)$$

Samband mellan parallell- och serieform

Serieform → parallellform:

$$K = K' \frac{T'_i + T'_d}{T'_i}$$

$$T_i = T'_i + T'_d$$

$$T_d = \frac{T'_i T'_d}{T'_i + T'_d}$$

Parallellform → serieform:

Villkor: $T_i > 4T_d$

$$K' = \frac{K}{2} \left(1 + \sqrt{1 - \frac{4T_d}{T_i}} \right)$$

$$T'_i = \frac{T_i}{2} \left(1 + \sqrt{1 - \frac{4T_d}{T_i}} \right)$$

$$T'_d = \frac{T_i}{2} \left(1 - \sqrt{1 - \frac{4T_d}{T_i}} \right)$$

I verkligheten

Algoritmen

$$u = K\left(e + \frac{1}{T_i} \int^t e(s) ds + T_d \frac{de}{dt}\right)$$

$$e = r - y$$

är en "läroboksalgoritm". Flera förändringar måste göras för att få en praktiskt användbar regulator. Vi måste tänka på

- Derivering av signaler kräver aktsamhet
- Det kan vara oklokt att derivera börvärdet
- Det kan vara klokt att proportionaldelen endast verkar på en del av börvärdet
- Åtgärder måste vidtagas om styrsignalen begränsas

Problemet med att derivera

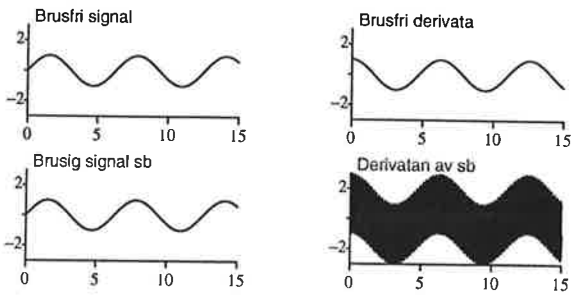
Betrakta signalen

$$y(t) = \sin t + a_n \sin \omega t$$

Derivatan är

$$\frac{dy(t)}{dt} = \cos t + a_n \omega \cos \omega t$$

Exempel $\omega = 100$, $a_n = 0.01$



Filtrering

Derivatadelen kräver filtrering.

Filtertidskonstanten ska relateras till processens dynamik.

Detta görs oftast automatiskt.

Tumregel: $T_f \approx T_d/10$

Börvärden – D-delen

Börvärdet är ofta konstant under långa tider.

Börvärdesändringar är ofta stegvisa.

Undvik därför derivering av börvärdet.

$$u = K\left(e + \frac{1}{T_i}\int e(t)dt + T_d\frac{de}{dt}\right)$$

ändras till

$$u = K\left(e + \frac{1}{T_i}\int e(t)dt - T_d\frac{dy}{dt}\right)$$

Börvärden – P-delen

Även P-delen modifieras ofta:

$$u = K\left(e + \frac{1}{T_i}\int e(t)dt - T_d\frac{dy}{dt}\right)$$

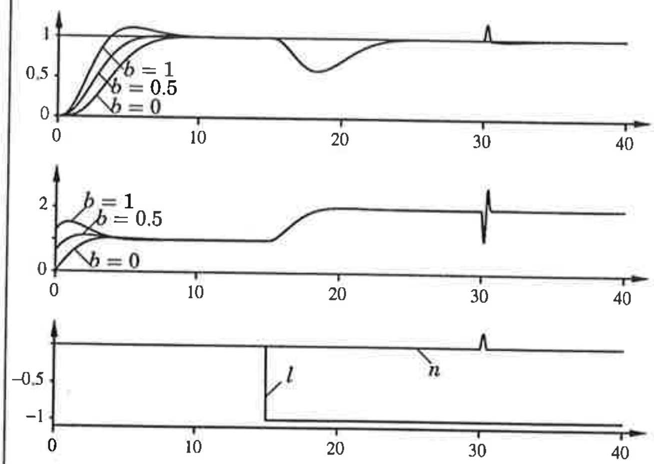
ändras till

$$u = K\left(br - y + \frac{1}{T_i}\int e(t)dt - T_d\frac{dy}{dt}\right)$$

Industristandard: b = 0 eller 1

Ofta fördel att välja $0 < b < 1$

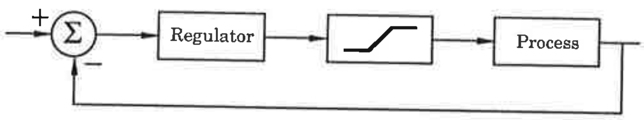
Verkan av börvärdesviktning



Vi återkopplar ej från reglerfelet utan vi har olika signalvägar $y \rightarrow u$ och $r \rightarrow u$. Regulatorn har två frihetsgrader.

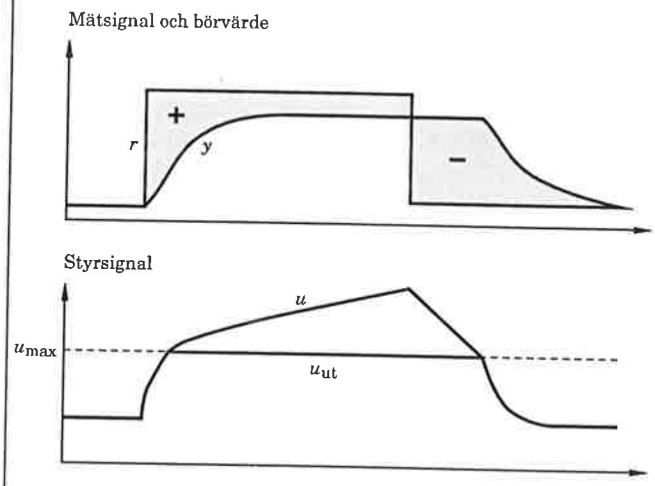
Problemen att reducera laststörningar och att få bra svar på börvärdesändringar har delvis separerats.

Problemet med mättning



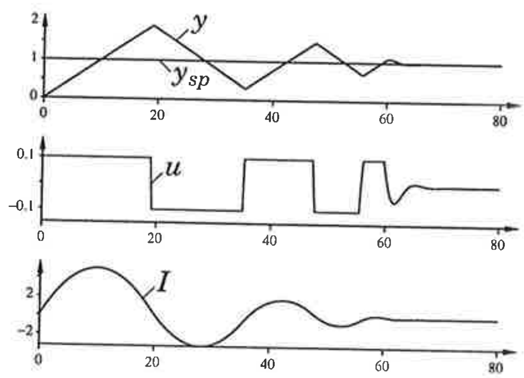
Om det finns en begränsare i en reglerkrets och om signalen blir så stor att begränsningen träder i funktion brytes återkopplingen. Detta kan ge förödande konsekvenser om regulatorn eller processen är instabil. JAS!

Integratoruppvridning



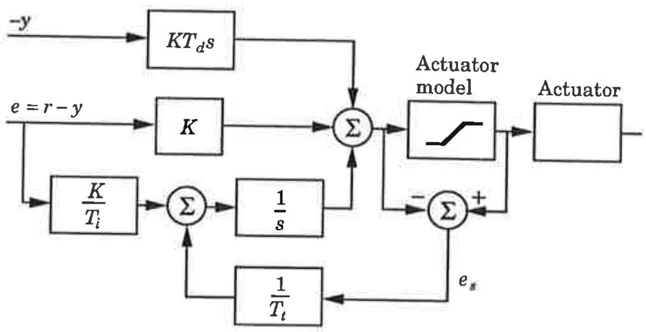
Integratoruppvridning

Ytterligare en simulering:



Hur undviker man integratoruppvridning?

Se till att integratorn inte skenar iväg när signalen mättar!

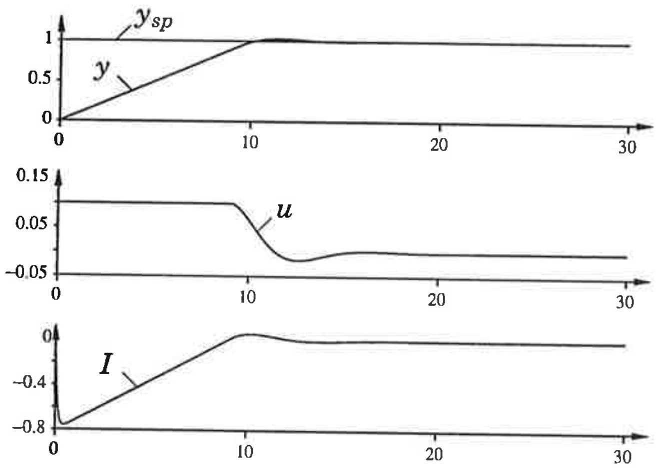


Hur fungerar kretsen?

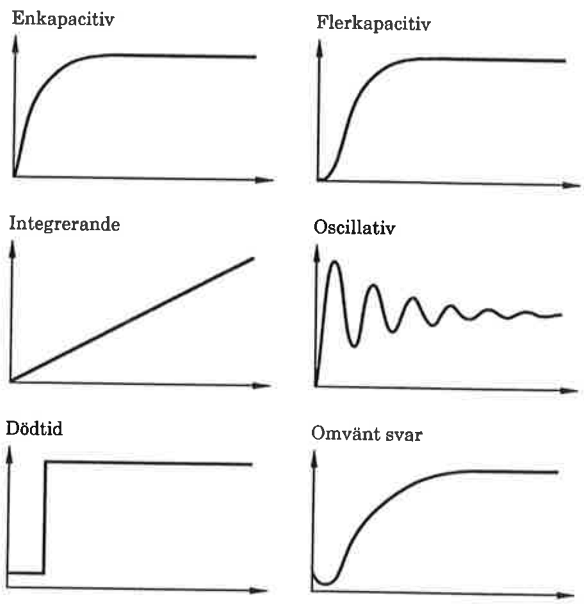
Utgå från att kretsen fungerar och att felet är noll! "Cherchez l'erreur!"

Drastisk förbättring!

Kompensering för regulatoruppvridning och upprepning av den tidigare simuleringen.



När ska P, I och D användas?



Sammanfattning

- PID naturlig utvidgning av On/Off
- Parametrarna har fysikalisk tolkning
- Det finns olika versioner
- Praktiska modifieringar
  - Börvärden
  - Brus
  - Integratoruppvridning
- Hur ska den användas?

Department of Automatic Control
Lund Institute of Technology

”PID och Fuzzy”

PID-regulatorns inställning

- Inledning
- Innan vi börjar ...
- Reglerkretsens egenskaper
- Manuell inställning
- Ziegler-Nichols metoder
- Specifikationer
- Moderna inställningsmetoder
- Automatisk inställning

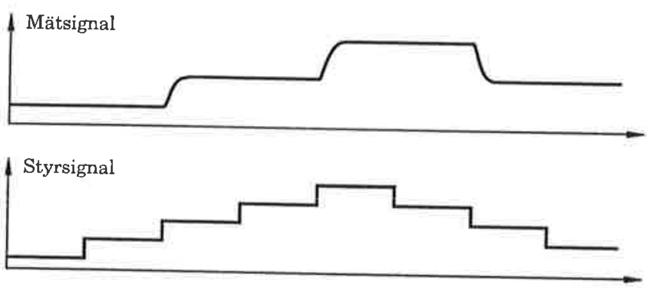
Inledning

- Många metoder
- Olika krav på prestanda
- Kunskap om process och reglerteknik
- Tid till förfogande
- Verktyg
  - Skrivare, penna, papper
  - Datalogger
  - Inställningsmaskiner

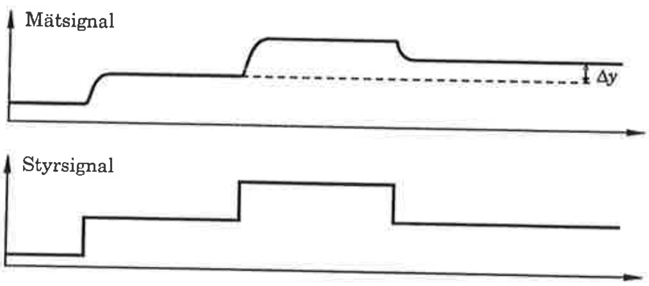
Innan vi börjar ...

Kontrollera funktion, glapp, friktion.

Friktionskontroll:

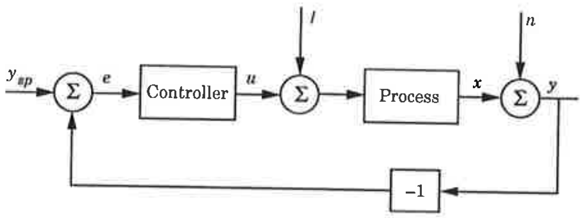


Kontroll av glapp:



Standardkretsen

Vi skall nu undersöka reglerkretsen



Önskvärda egenskaper

- Reducera inverkan av laststörningar
- Mätbrus skall ha liten inverkan
- Liten känslighet för variationer i processen
- Följa variationer i börvärdet (setpoint) väl

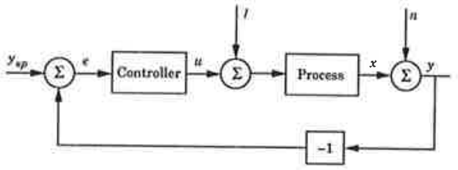
Hur skall detta uttryckas?

Finns det några begränsningar?

Hur skall vi gå tillväga?

Undersök hur x, y och u beror av y_{sp}, l och n

Grundläggande samband



Vi har tre insignaler $y_{sp} = r$, l och n och fyra intressanta x, y, e och u. Sambanden ges alltså av tolv överföringfunktioner.

$$G_{xr} = \frac{G_p G_c}{1+G_p G_c} \quad G_{xl} = \frac{G_p}{1+G_p G_c} \quad G_{xn} = -G_{xr}$$

$$G_{yr} = G_{xr} \quad G_{yl} = G_{xl} \quad G_{yn} = 1 - G_{xr}$$

$$G_{er} = 1 - G_{xr} \quad G_{el} = -G_{xl} \quad G_{en} = -1 + G_{xr}$$

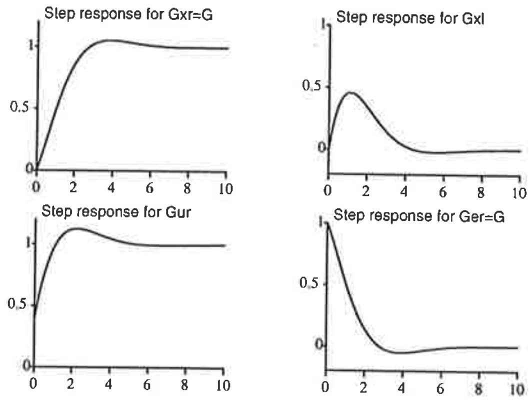
$$G_{ur} = \frac{G_c}{1+G_p G_c} \quad G_{ul} = -G_{xr} \quad G_{un} = -G_{ur}$$

Det räcker med fyra överföringfunktioner G_{xr}, G_{xl}, G_{yn} och G_{ur}.

Alla överföringsfunktioner behövs för att beskriva egenskaperna hos ett system. Det räcker inte med att visa en av dem!

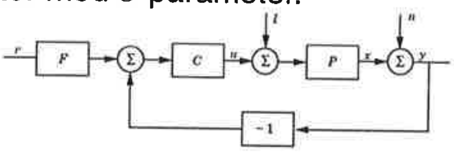
Hur man visar egenskaperna hos en enkel reglerkrets

För att bedöma en enkel reglerkrets är det nödvändigt att visa egenskaperna hos fyra linjära system. Systemen kan representeras med överföringsfunktionerna: G_{xr}, G_{xl}, G_{er} och G_{ur}. Vi kan t.ex. beräkna x och e vid ett stegsvar i referensvärdet r och utsignalen vid ett steg i laststörningen l.



Fler samband för system med framkoppling

Till exempel en PID regulator eller en PI regulator med b parameter.



System med två frihetsgrader (2DOF).

Tre insignaler r l och n och tre intressanta signaler u, x and y.

Nio samband!

$$G_{ur} = \frac{CF}{1+PC} \quad G_{xr} = \frac{PCF}{1+PC} \quad G_{yr} = G_{xr}$$

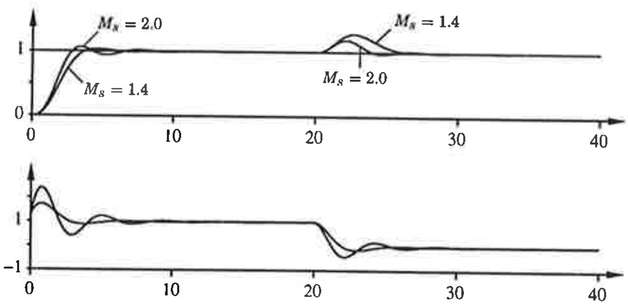
$$G_{ul} = -\frac{PC}{1+PC} \quad G_{xl} = \frac{P}{1+PC} \quad G_{yl} = G_{xl}$$

$$G_{un} = -\frac{C}{1+PC} \quad G_{xn} = -\frac{PC}{1+PC} \quad G_{yn} = \frac{1}{1+PC}$$

Bara 6 är olika!

Hur beskriva dynamiska samband?

Tidssvar

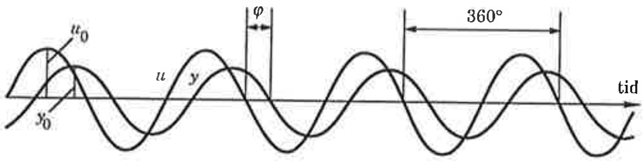


Hur man kan luras?

Frekvenssvar

Hur systemet svarar på sinussignaler.

Frekvenssvar



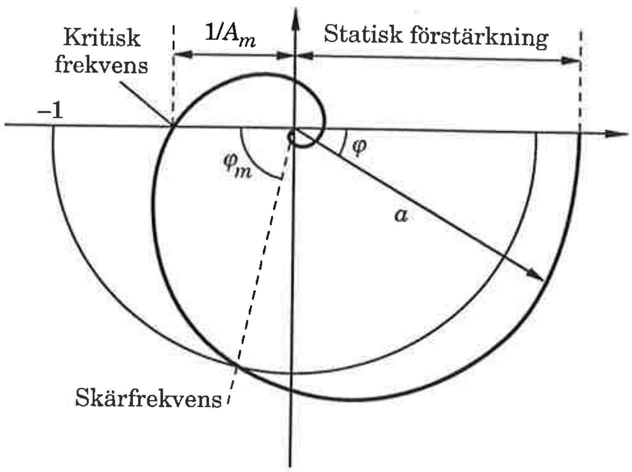
$$u(t) = u_0 sin(\omega t)$$
$$y(t) = y_0 sin(\omega t + \varphi)$$

Signalerna har samma frekvens men olika amplitud. De är fasförskjutna i förhållande till varanda.

Förstärkning: $a(\omega) = y_0/u_0$

Fasvridning: $\varphi(\omega)$

Nyquistdiagram

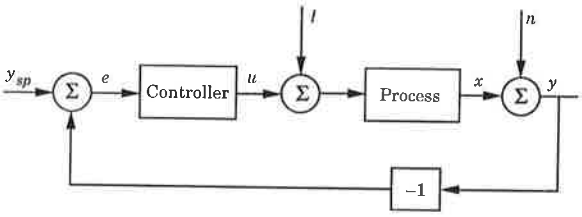


Nyquistkriteriet:

Det återkopplade systemet är stabilt om Nyquistkurvan inte omsluter punkten −1.

Intressanta samband

Betrakta det enkla återkopplade systemet



Vissa samband har speciella namn

Kretsöverföringfunktionen

$$G_o = G_p G_c$$

Känslighetsfunktionen

$$G_s = 1/(1 + G_o)$$

Komplementära känslighetsfunktionen

$$G_t = G_o/(1 + G_o)$$

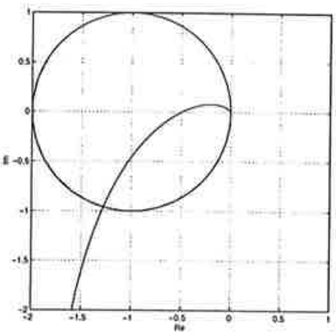
Fysikalisk tolkning!

Känslighetsfunktionen

Talar om hur störningar påverkas av regleringen!

$$\frac{Y_{cl}}{Y_{ol}} = \frac{1}{1 + G_p G_c} = G_s$$

Rita Nyquistdiagramet för kretsöverföringsfunktionen $G_p G_c$



Cirkeln $|G_s(i\omega)| = 1$ är intressant! Störningar med frekvenser som ligger innanför cirkeln förstärks. Störningar med frekvenser utanför cirkeln reduceras med regleringen.

Andra intressanta egenskaper

Känslighetsfunktionen

$$G_s = \frac{1}{1 + G_p G_c}$$

Komplementära känslighetsfunktionen

$$G_t = \frac{G_p G_c}{1 + G_p G_c} = 1 - G_s$$

har många intressanta egenskaper

$$G_s + G_t = 1$$

$$G_s = \frac{Y_{cl}(s)}{Y_{ol}(s)}$$

$$G_s = \frac{\partial \log G_t}{\partial \log G_p} = \frac{\partial \log G_t}{\partial \log G_c}$$

$$|\frac{\Delta G_p}{G_p}| < |\frac{1}{G_t}|$$

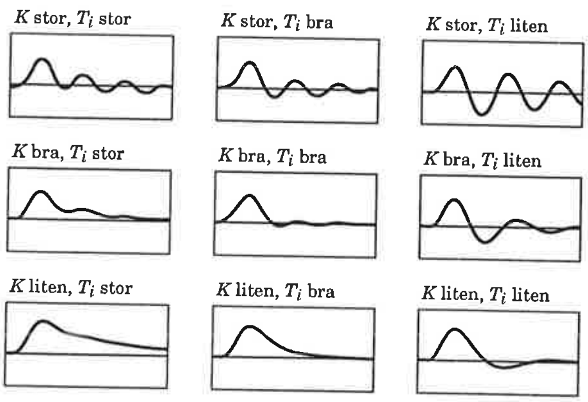
För att ha låg känslighet för processvariationer får G_s och G_t ej bli för stora (< 2 eller < 1.4).

Manuell inställning

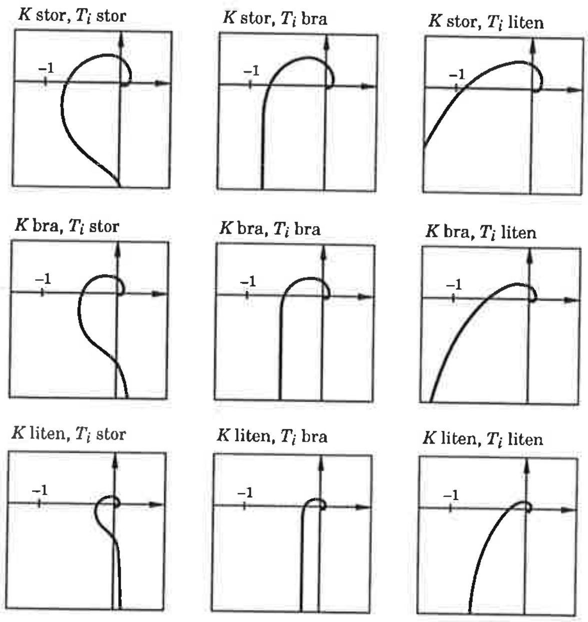
Tumregler:

	Snabbhet	Stabilitet
K ökar	ökar	minskar
T_i ökar	minskar	ökar
T_d ökar	ökar	ökar

Inställningsschema baserat på tidssvar

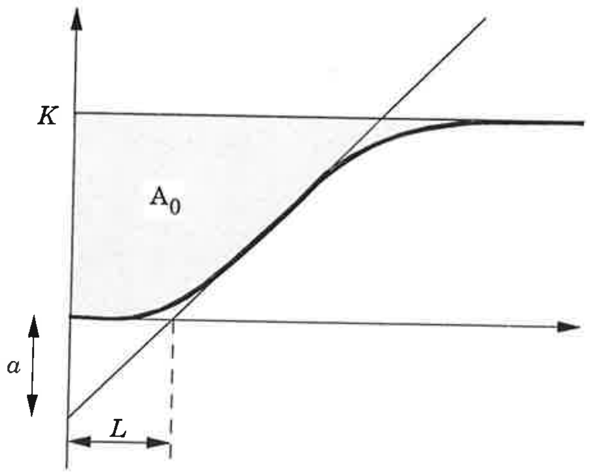


Inställningsschema baserat på frekvenssvar



Ziegler-Nichols' stegsvarsmetod

- Koppla regulatorn i manuell drift.
- Gör en stegändring i styrsignalen.
- Registrera processens utsignal. Normalisera kurvan så att den svarar mot ett steg med storleken 1.
- Bestäm parametrarna a och L.
- Lämpliga parametrar erhålles ur en tabell.



Ziegler-Nichols' stegsvarsmetod

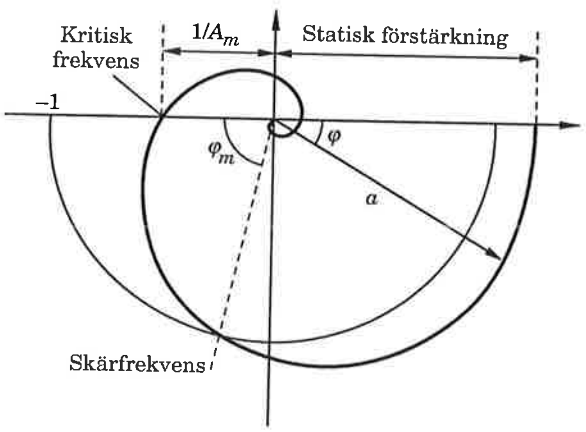
Parametrarna a och L bestäms ur (det normaliserade) stegsvaret

Regulator	K	T_i	T_d	T_p
P	$1/a$			$4L$
PI	$0.9/a$	$3L$		$5.7L$
PID	$1.2/a$	$2L$	$L/2$	$3.4L$

Parametern T_p anger det slutna systemets förväntade tidskonstant.

Ziegler-Nichols' frekvenssvarsmetod

- Låt regulatorn vara en ren P-regulator.
- Justera förstärkningen så att slutna systemet ligger på stabilitetsgränsen.
- Registrera förstärkningen K_u och perioden T_u hos den svängning som uppträder.
- Lämliga regulatorparametrar erhålles ur en tabell.



Ziegler-Nichols' frekvenssvarsmetod

Parametrarna K_u och T_u bestäms med det speciella experimentet

Reg.	K	T_i	T_d	T_p
P	$0.5K_u$			T_u
PI	$0.4K_u$	$0.8T_u$		$1.4T_u$
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$	$0.85T_u$

Parametern T_p anger det slutna systemets förväntade tidskonstant.

Sammanfattning

Ziegler och Nichols arbetade på Taylor Instruments (ingår numera i ABB)

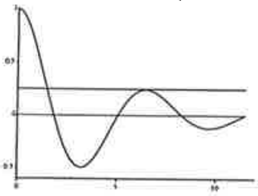
- Karakterisera processens dynamik med få (**två**) parametrar
- Nichols kom fram till parametrarna genom omfattande simuleringar

Egenskaper

\+ Lätt att förklara och använda

\+ Ofta använd

– Systemet alltför oscillativt. Inbyggt i metoden (quarter amplitude damping)



– Alltför stor översläng

– Känsligt för variationer i processen

Stort utrymme för förbättringar. Mer processinformation behövs.

Bättre inställningsregler

Ziegler-Nichols regler kan förbättras genom att utnyttja mer processinformation och ställa krav på bättre dämpning.

Många processer kan approximeras med följande modell som har tre parametrar

$$G_p(s) = K_p \frac{e^{-sL}}{1+sT}$$

Parametrarna K_p, L och T kan bestämmas från stegsvaret. Inför $a = K_p L/T$.

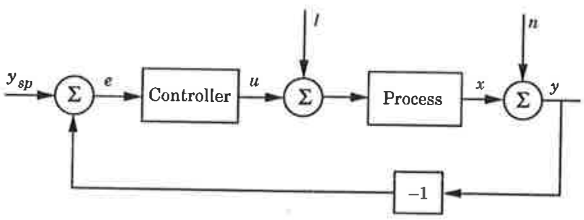
En bra inställning av en PI regulator ges av

$$K = \begin{cases} 0.4/a & \text{om } L \le 2T, \\ 0.2/K_p & \text{om } L > 2T \end{cases}$$

$$T_i = \begin{cases} 3.5L & \text{om } L \le 0.2T, \\ 0.7T & \text{om } 0.2T < L \le 2T \\ 0.35L & \text{om } L > 2T \end{cases}$$

Jämför med ZN!

Krav på en reglerkrets



- Reducera laststörningar
- Mätsignalen skall följa börvärdet
- Mätbrus skall ha liten inverkan
- Det slutna systemet skall inte vara känsligt för variationer i processens egenskaper

Kappa-Tau–metoden

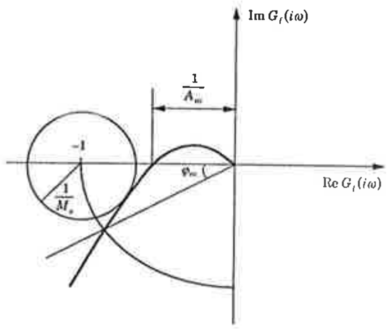
- Reglering av laststörningar

$$IE = \int_0^\infty e(t)\,dt = \frac{T_i}{K}$$

- Börvärdesföljning

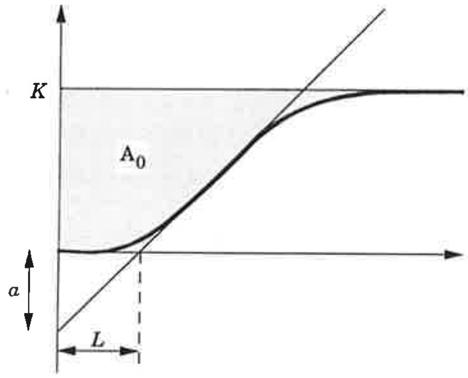
$$u = K\left(by_{sp} - y + \frac{1}{T_i}\int e dt - T_d \frac{dy}{dt}\right)$$

- Känslighet M_s



KT – Metoden

Stegsvarsmetoden

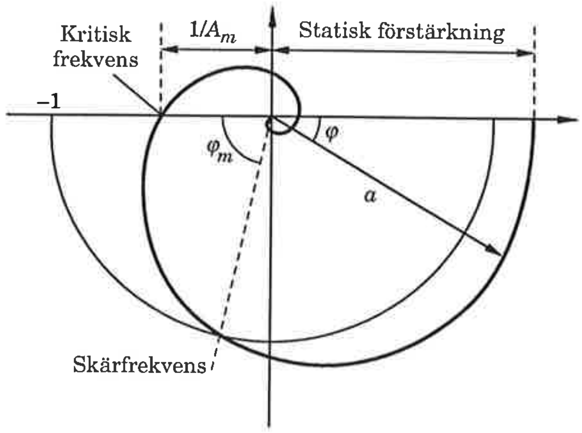


Tre parametrar: a, L, and K_p

Normaliserad dödtid: $\tau = \frac{L}{L+T} = \frac{a}{a+K_p}$

KT – Metoden

Frekvensmetoden

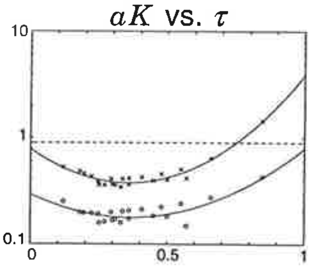


Tre parametrar: K_u, T_u, and K_p

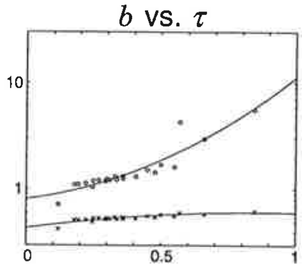
Förstärkningskvot: $\kappa = |\frac{G(i\omega_u)}{G(0)}| = \frac{1}{K_p K_u}$

PI – Stabila processer

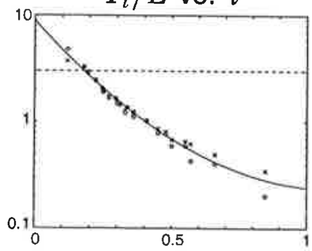
aK vs. τ



b vs. τ

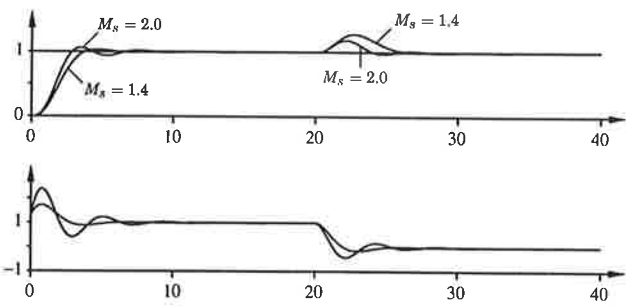


T_i/L vs. τ



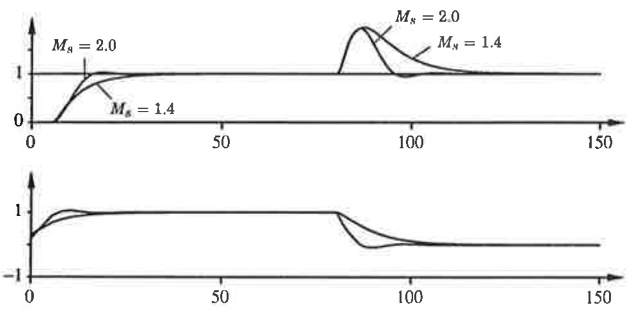
- Jämför med Ziegler-Nichols
- Vi behöver tre parametrar

Exempel 1



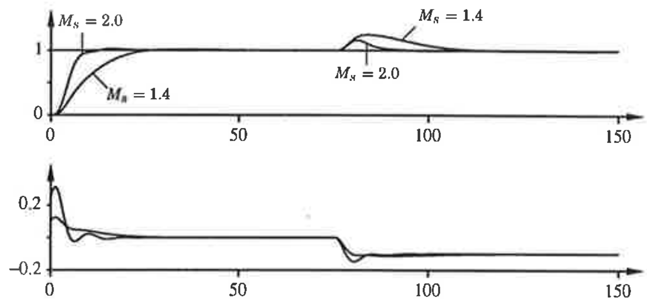
$$G(s) = \frac{1}{(s+1)^3}$$

Exempel 2



$$G(s) = \frac{e^{-5s}}{(s+1)^3}$$

Exempel 3



$$G(s) = \frac{1}{s(s+1)^3}$$

Lambda-tuning

- Hur uppkom metoden?
- Tidig digital reglering
  - Dahlin-Higham-Measurex
  - Förkorta processpoler
  - Slutna systemets tidskonstant $T_{cl} = \lambda T$
  - Fokus på dödtidskompensering
- PI(D)-reglering
  - Riviera-Morari
  - Specialfall av IMC
  - Approximationer ger PID
  - Bill Bialkowski
- Egenskaper
  - \+ Enkel
  - \+ Naturlig specifikation
  - \+ Basen för SSG arbetet
  - \- Farligt att förkorta långsamma poler
  - \- Fungerar ej för processer med integration

Vad är lambda-tuning?

Processmodell

$$G_p(s) = K_p \frac{e^{-sL}}{1+sT}$$

Iden:

- Förkorta processpol i $s = -1/T$ genom regulatorns nollställe, dvs $T_i = T$
- Placera en pol i $s = -1/T_{cl}$
- Approximera dödtid med nollställe: $e^{-sL} \approx 1 - sL$

Resultat:

- En mycket enkel inställningsregel

$$K_c K_p = \frac{T}{L + T_{cl}}$$
$$T_i = T$$

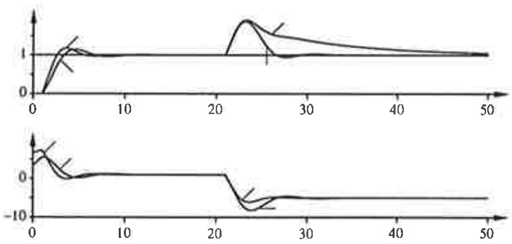
- Designvariabeln T_{cl} är slutna systemets tidskonstant

Förkortning

OK att förkorta pol om

- Den är stabil
- Fungerar inte på integrarande processer!
- $T < T_{cl}$

Förkortning av långsamma poler ger dålig reglering vid laststörningar



Enkelt att modifiera designmetoden.

Frågor

- Vad innebär förkortningen?
- Vad innebär approximationen?
- Vad händer med övriga poler?
- Uppnår vi specifikationen på T_{cl}?
- Känslighet för modellfel?
- Relation till andra designmetoder?
- Vägledning för val av T_{cl}?
- Designkriterium?

Automatisk inställning

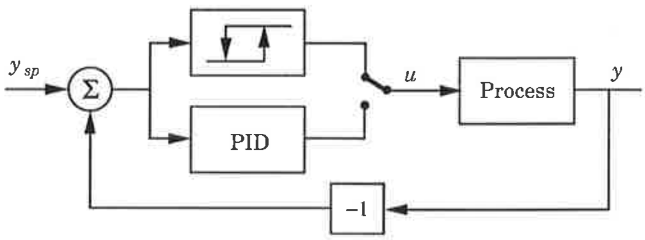
När en regulator skall ställas in går vi igenom följande tre faser:

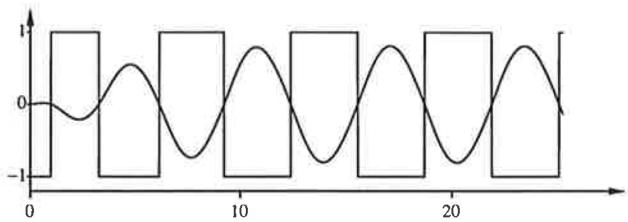
1. Stör processen.
2. Bilda en processmodell.
3. Bestäm regulatorparametrar baserat på modellen.

Automatisk inställning (Auto-tuning)

Ett hjälpmedel för inställning av regulatorer där dessa faser gås igenom automatiskt.

Relämetoden





- Lite apriorikunskap
- En knapp räcker
- Automatisk generering av testsignal
- Goda industriella erfarenheter

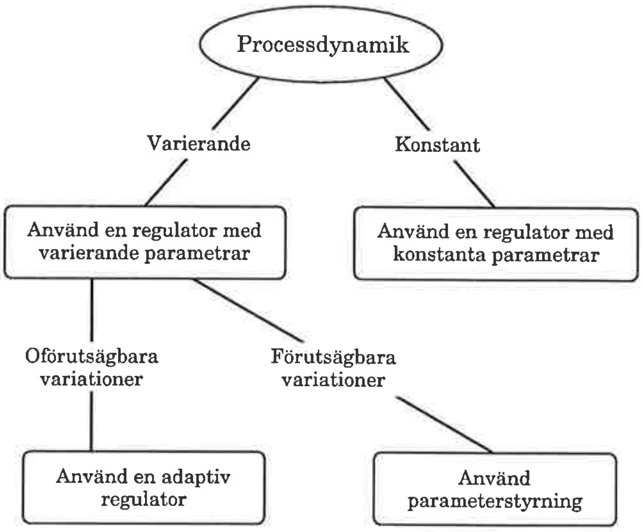
Department of Automatic Control
Lund Institute of Technology

"PID och Fuzzy"

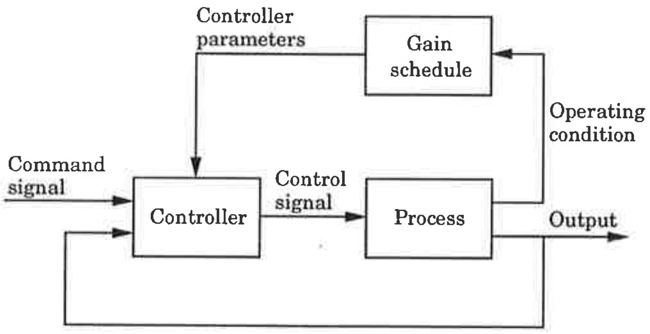
PID-regulatorns användning

- Varierande processdynamik
- Dödtider – PPI
- Kaskadkoppling
- Framkoppling
- Utvecklingstrender

Varierande processdynamik



Parameterstyrning



Exempel på parameterstyrningsvariabler

- Produktionsnivå
- Maskinhastighet
- Mach-tal och dynamiskt tryck (flygplan)
- Antal personer i rum

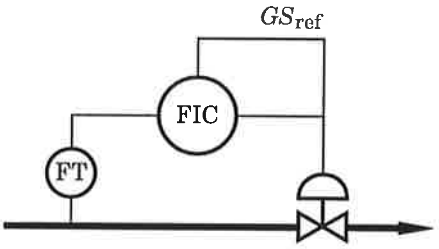
Parameterstyrning

En tabell med olika regulatorparametrar för olika arbetsområden.

GS_{ref}	0–33 %	33–67 %	67–100 %
K	0.43	0.28	0.20
T_i	1.81	1.89	1.71
T_d	0.45	0.47	0.43

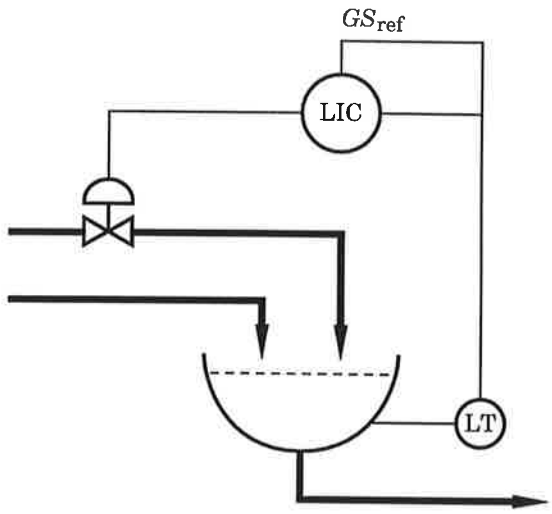
- Interpolation
- Fuzzy

Parameterstyrning



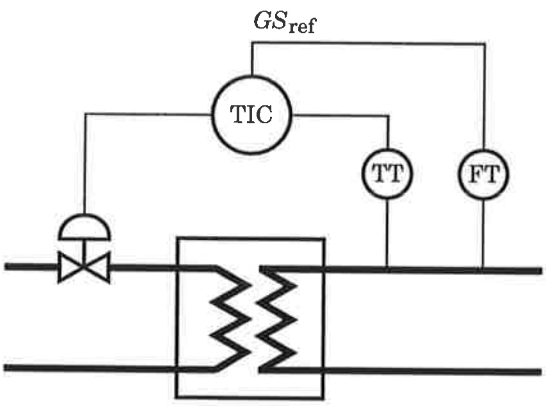
Parameterstyrningen baseras på styrsignalen.

Parameterstyrning



Parameterstyrningen baseras på mätsignalen.

Parameterstyrning

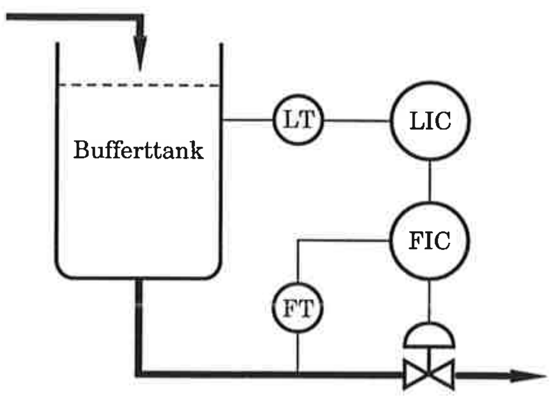


Parameterstyrningen baseras på en yttre signal (flödet).

Parameterstyrning

Parameterstyrning kan även användas för linjära processer då man har produktionsberoende specifikationer.

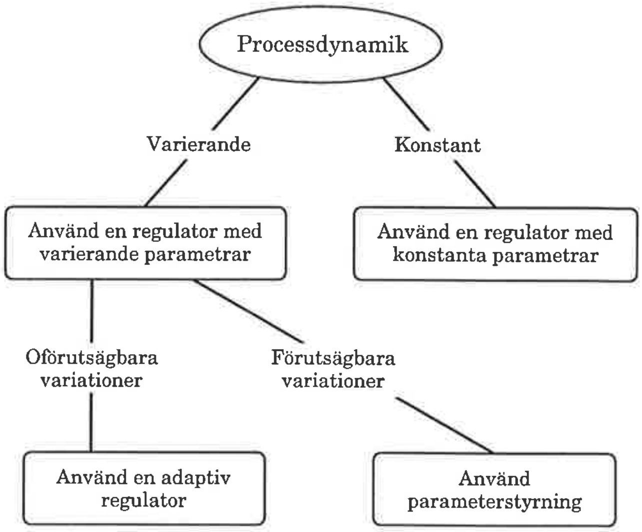
Exempel – Buffertreglering:



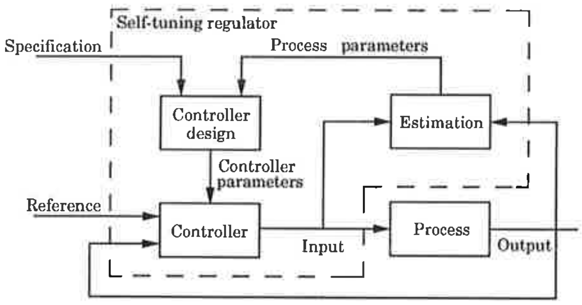
Parameterstyrning

- Många användningsområden
  - Reglering över stora arbetsområden
  - Linjärisering av ventiler
  - Reglering av bufferttankar
- Viktiga aspekter
  - Val av referenssignal
  - Områdenas indelning
  - Interpolering?
  - Stötfria övergångar vid parameter-byten
  - Människa-maskin-interface
- Automatinställning användbart

Varierande processdynamik

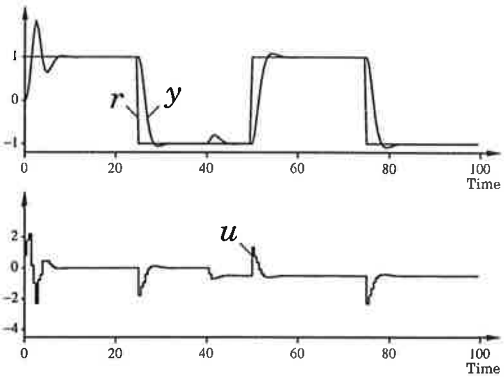


Adaptiv reglering

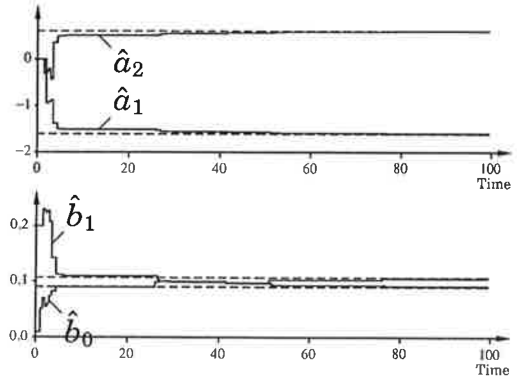


- Många varianter
  - Många regulatorstrukturer
  - Många estimeringsmetoder
  - Många designmetoder
- Dual reglering
  - Regulatorn ska både reglera och excitera!

Exempel - adaptiv reglering

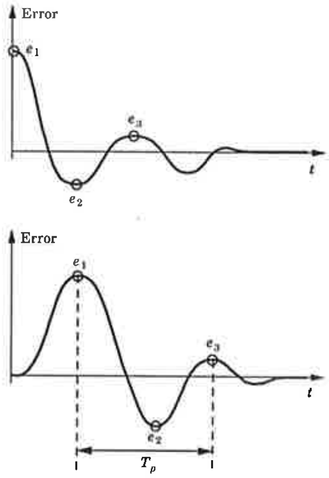


Estimerade processparametrar



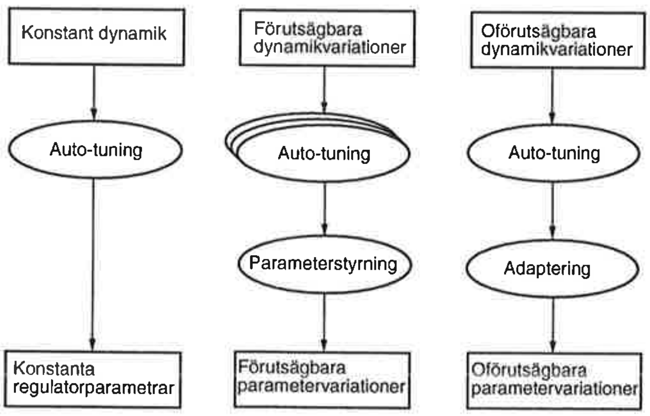
Foxboro EXACT

- Härma en bra instrumentingenjör
- Mönsterigenkänning



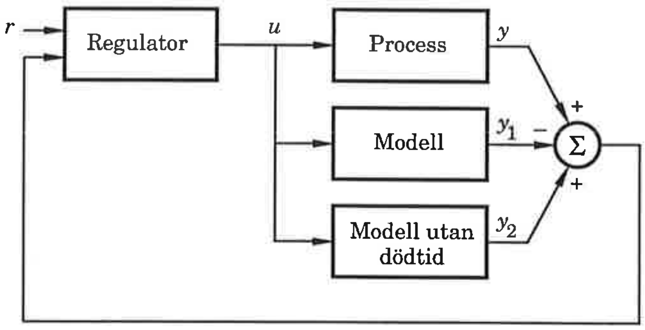
- Regelbaserad
- Huvudidée
  - Börja med rimliga parametrar och förbättra
- Kräver startvärden

Sammanfattning - Adaptiv teknik



Dödtidskompensering

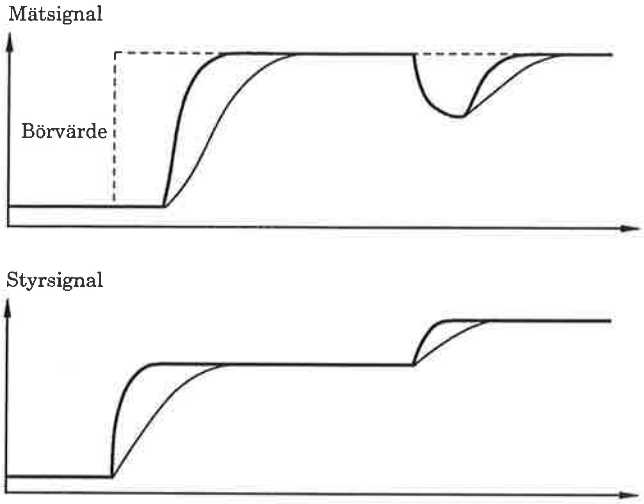
Smithprediktorn:



Om modellen stämmer exakt blir $y - y_1 = 0$

Då blir regleringen som om dödtiden ej fanns, bortsett från att ärvärdet blir fördröjt.

Exempel: Dödtidskompensering



Tunna linjer: PI
Tjocka linjer: Dödtidskompensering

PPI-regulatorn

PID-regulatorn:

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{de(t)}{dt}\right)$$

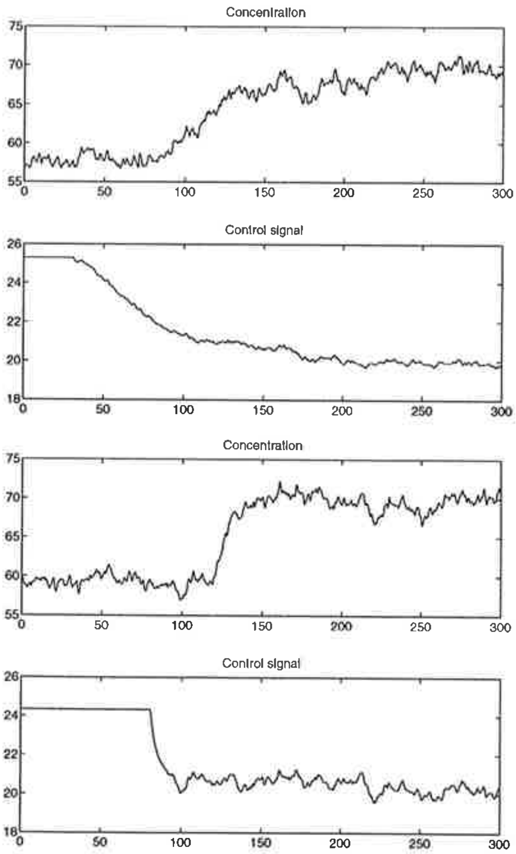
PPI-regulatorn:

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int e(t)dt\right) - \frac{1}{T_i}\int_{t-L}^{t} u(t)dt$$

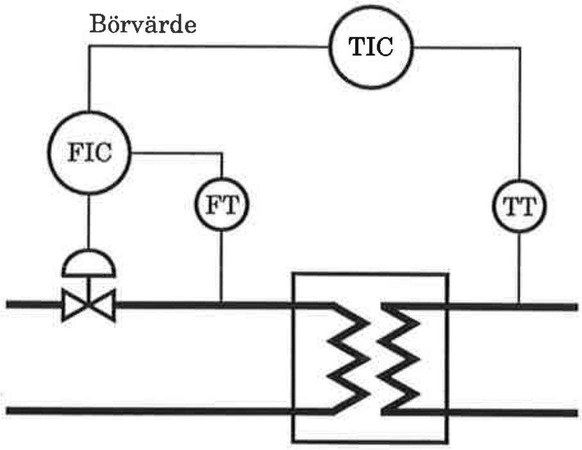
Prediktionen åstadkommen genom att lågpassfiltrera u i stället för att högpassfiltrera y.

Endast 3 parametrar att ställa in: K, T_i, L

Exempel: PI och PPI



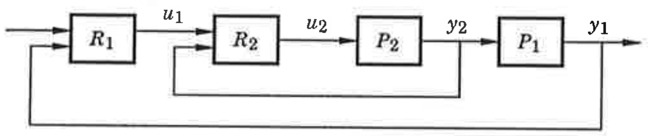
Kaskadreglering



Den inre kretsen kompenserar för tryckvariationer i ångledningen.
Den yttre regulatorn styr därför ångflödet i stället för ventilläget.
Konstruktionen kompenserar även för olinjäriteter i ventilen.

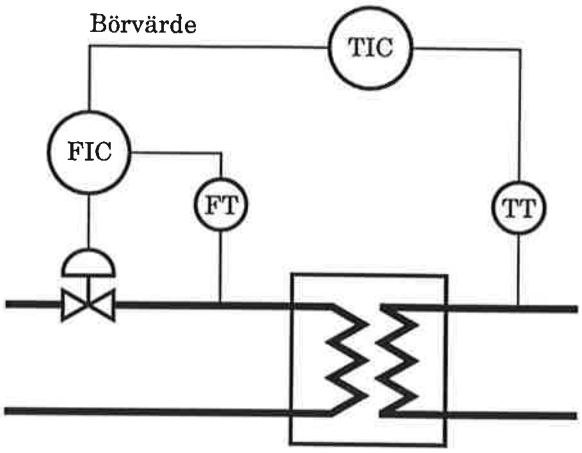
Principen för kaskadreglering

Kombination av två PID-regulatorer, där utsignalen från den ena regulatorn bildar börvärde till den andra.



Kompensera för variationer i y_2 innan de slår igenom i y_1.

Kaskadreglering



Vad händer när

- Styrsignalen från FIC begränsas?
- FIC arbetar i manuell reglering?
- FIC arbetar med lokalt börvärde?

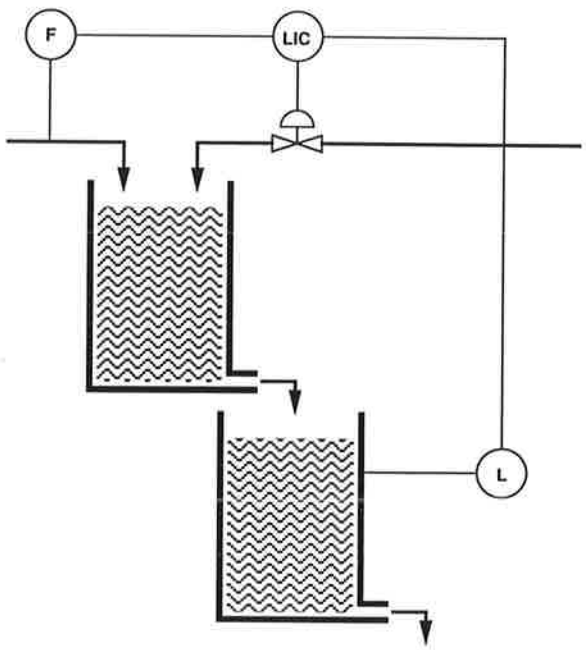
Framkoppling

Återkoppling är ett effektivt sätt att eliminera effekterna av störningar. Dock måste först ett reglerfel uppstå innan regulatorn reagerar.

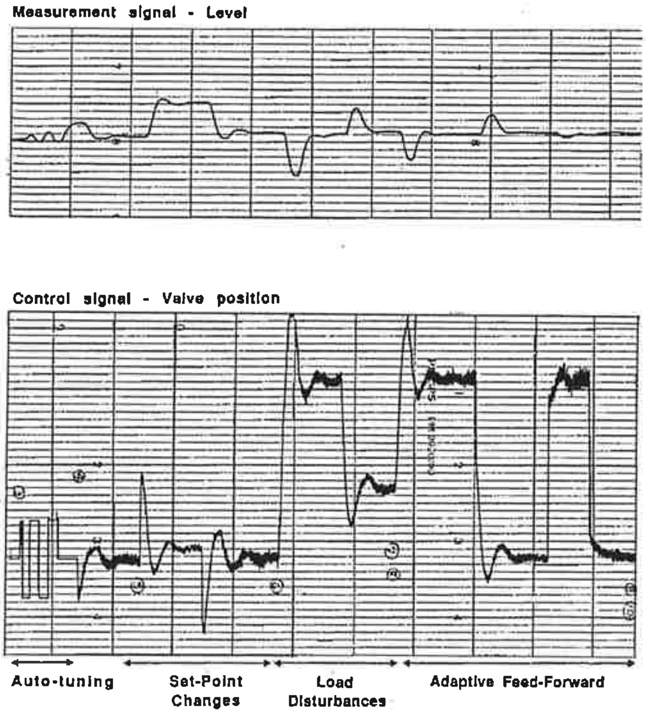
Framkoppling innebär att vi kompenserar för en mätbar störning innan den ger upphov till ett reglerfel.

Vanlig tillämpning: Temperaturreglering av bostadshus.

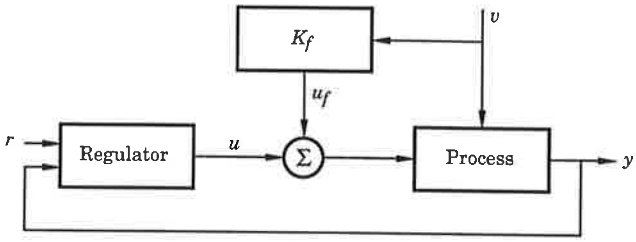
Framkoppling vid nivåreglering



Framkoppling vid nivåreglering

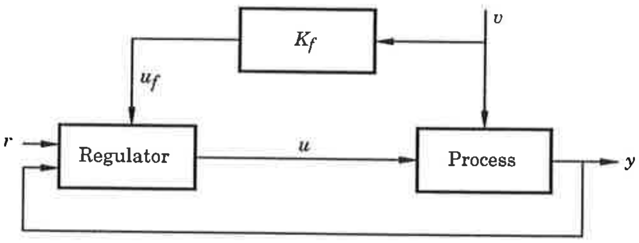


Principen för framkoppling



Så här ska den inte realiseras!

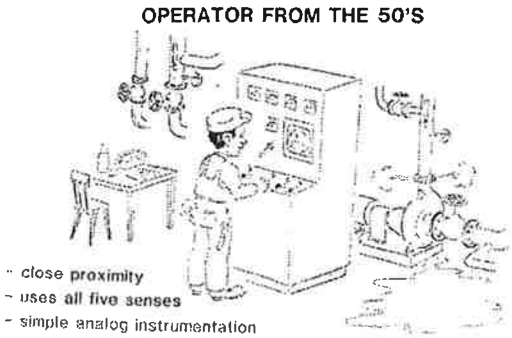
Realisering av framkoppling

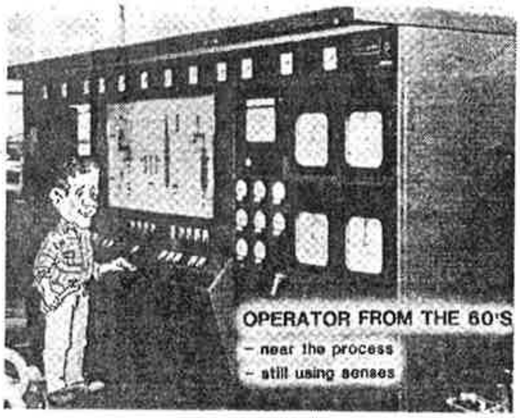


Andra strategier

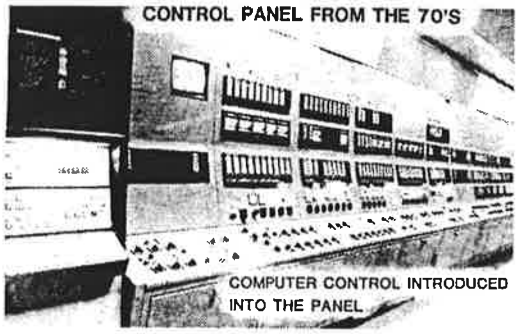
- Kvotreglering
- Väljare
- Begränsare
- Split range
- Modellföljning

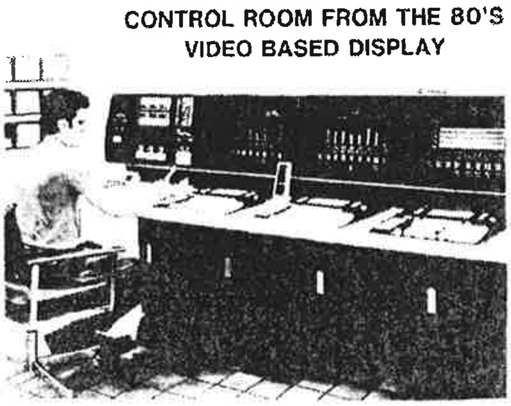
Utvecklingstrender





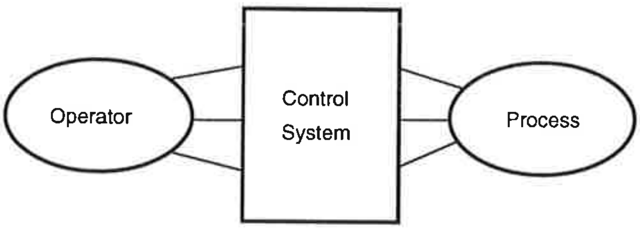
Utvecklingstrender





Utvecklingtrender

- Hårdvara – Revolution
- Operatörens roll – Revolution
- Hur har reglertekniken påverkats?
  - + Automatinställning, gain scheduling m.m.
  - – Många funktioner används fortfarande som på den gamla pneumatiska tiden
- Intresset måste flyttas från verktygen (systemen) till processen



Sammanfattning

- PID-regulatorn grundläggande byggblock
- Funktionen kan utvidgas
  - Parameterstyrning
  - Adaptivitet
  - Dödtidskompensering
- Grundkopplingar
- Den nya tekniken ger möjligheter

10.5 PID Control

Karl J. Åström, *Department of Automatic Control, Lund Institute of Technology, Lund, Sweden*
Tore Hägglund, *Department of Automatic Control, Lund Institute of Technology, Lund, Sweden*

10.5.1 Introduction

The proportional-integral-derivative (PID) controller is by far the most commonly used controller. About 90 to 95% of all control problems are solved by this controller, which comes in many forms. It is packaged in standard boxes for process control and in simpler versions for temperature control. It is a key component of all distributed systems for process control. Specialized controllers for many different applications are also based on PID control. The PID controller can thus be regarded as the "bread and butter" of control engineering. The PID controller has gone through many changes in technology. The early controllers were based on relays and synchronous electric motors or pneumatic or hydraulic systems. These systems were then replaced by electronics and, lately, microprocessors.

Much interest was devoted to PID control in the early development of automatic control. For a long time researchers paid very little attention to the PID controller. Lately there has been a resurgence of interest in PID control because of the possibility of making PID controllers with automatic tuning, automatic generation of gain schedules and continuous adaptation. See the chapter "Automatic Tuning of PID Controllers" in this handbook.

Even if PID controllers are very common, they are not always used in the best way. The controllers are often poorly tuned. It is quite common that derivative action is not used. The reason is that it is difficult to tune three parameters by trial and error.

In this chapter we will first present the basic PID controller in Section 10.5.2. When using PID control it is important to be aware of the fact that PID controllers are parameterized in several different ways. This means for example that "integral time" does not mean the same thing for different controllers. PID controllers cannot be understood from linear theory. Amplitude and rate limitations in the actuators are key elements that lead to the windup phenomena. This is discussed in Section 10.5.4 where different ways to avoid windup are also discussed. Mode switches also are discussed in the same section.

Most PID controllers are implemented as digital controllers. In Section 10.5.5 we discuss digital implementation. In Section 10.5.6 we discuss uses of PID control, and in Section 10.5.7 we describe how complex control systems are obtained in a "bottom up" fashion by combining PID controllers with other simple systems.

We also refer to the companion chapter "Automatic Tuning of PID Controllers" in this handbook, which treats design and tuning of PID controllers. Examples of industrial products are also given in that chapter.

10.5.2 The Control Law

In a PID controller the control action is generated as a sum of three terms. The control law is thus described as

$$u(t) = u_P(t) + u_I(t) + u_D(t) \tag{10.93}$$

where u_P is the proportional part, u_I the integral part and u_D the derivative part.

Proportional Control

The proportional part is a simple feedback

$$u_P(t) = Ke(t) \tag{10.94}$$

where e is the control error, and K is the controller gain. The error is defined as the difference between the set point y_{sp} and the process output y, i.e.,

$$e(t) = y_{sp}(t) - y(t) \tag{10.95}$$

The modified form,

$$u_P(t) = K(by_{sp}(t) - y(t)) \tag{10.96}$$

where b is called *set point weighting*, admits independent adjustment of set point response and load disturbance response.

Integral Control

Proportional control normally gives a system that has a steady-state error. Integral action is introduced to remove this. Integral action has the form

$$u_I(t) = k_i \int^t e(s)ds = \frac{K}{T_i} \int^t e(s)ds \qquad (10.97)$$

The idea is simply that control action is taken even if the error is very small provided that the average of the error has the same sign over a long period.

Automatic Reset

A proportional controller often gives a steady-state error. A manually adjustable reset term may be added to the control signal to eliminate the steady-state error. The proportional controller given by Equation 10.94 then becomes

$$u(t) = Ke(t) + u_b(t) \qquad (10.98)$$

where u_b is the reset term. Historically, integral action was the result of an attempt to obtain automatic adjustment of the reset term. One way to do this is shown in Figure 10.49.

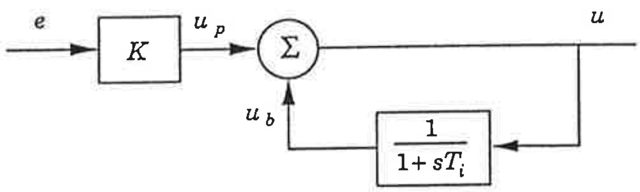


Figure 10.49 Controller with integral action implemented as automatic reset.

The idea is simply to filter out the low frequency part of the error signal and add it to the proportional part. Notice that the closed loop has positive feedback. Analyzing the system in the figure we find that

$$U(s) = K(1 + \frac{1}{sT_i})E(s)$$

which is the input-output relation of a proportional-integral (PI) controller. Furthermore, we have

$$u_b(t) = \frac{K}{T_i} \int^t e(s)ds = u_I(t)$$

The automatic reset is thus the same as integral action.

Notice, however, that set point weighting is not obtained when integral action is obtained as automatic reset.

Derivative Control

Derivative control is used to provide anticipative action. A simple form is

$$u_D(t) = k_d \frac{de(t)}{dt} = KT_d \frac{de(t)}{dt} \qquad (10.99)$$

The combination of proportional and derivative action is then

$$u_P(t) + u_D(t) = K[e(t) + T_d \frac{de(t)}{dt}]$$

This means that control action is based on linear extrapolation of the error T_d time units ahead. See Figure 10.50. Parameter T_d, which is called derivative time, thus has a good intuitive interpretation.

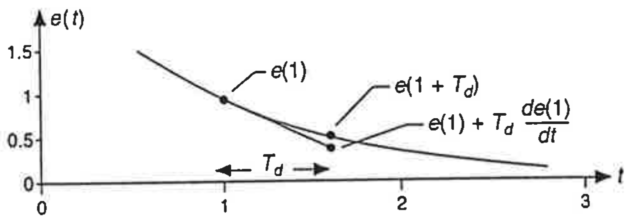


Figure 10.50 Interpretation of derivative action as prediction.

The main difference between a PID controller and a more complex controller is that a dynamic model admits better prediction than straight-line extrapolation.

In many practical applications the set point is piecewise constant. This means that the derivative of the set point is zero except for those time instances when the set point is changed. At these time instances the derivative becomes infinitely large. Linear extrapolation is not useful for predicting such signals. Also, linear extrapolation is inaccurate when the measurement signal changes rapidly compared to the prediction horizon T_d.

A better realization of derivative action is, therefore,

$$U_D(s) = \frac{KT_d s}{1+sT_d/N}(cY_{sp}(s) - Y(s)) \qquad (10.100)$$

The signals pass through a low-pass filter with time constant T_d/N. Parameter c is a set point weighting, which is often set to zero.

Filtering of Process Variable

The process output can sometimes be quite noisy. A first-order filter with the transfer function

$$G_f(s) = \frac{1}{1+sT_f} \qquad (10.101)$$

is often used to filter the signal. For PID controllers that are implemented digitally, the filter can be combined with the antialiasing filter as discussed in Section 10.5.5.

Set Point Weighting

The PID controller introduces extra zeros in the transmission from set point to output. From Equations 10.96, 10.97, and 10.99, the zeros of the PID controller can be determined as the roots of the equation

$$cT_iT_ds^2 + bT_is + 1 = 0 \qquad (10.102)$$

There are no extra zeros if $b = 0$ and $c = 0$. If only $c = 0$, then there is one extra zero at

$$s = -\frac{1}{bT_i} \qquad (10.103)$$

This zero can have a significant influence on the set point response. The overshoot is often too large with $b = 1$. It can be reduced substantially by using a smaller value of b. This is a much better solution than the traditional way of detuning the controller.

This is illustrated in Figure 10.51, which shows PI control of a system with the transfer function

$$G_p(s) = \frac{1}{s+a} \qquad (10.104)$$

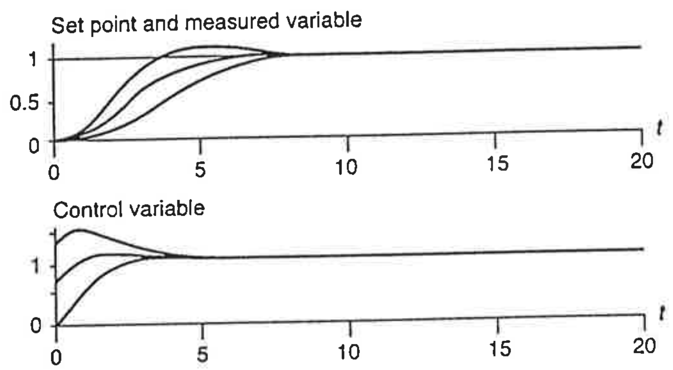


Figure 10.51 The usefulness of set point weighting. The values of the set point weighting parameter are 0, 0.5 and 1.

10.5.3 Different Representations

The PID controller discussed in the previous section can be described by

$$U(s) = G_{sp}(s)Y_{sp}(s) - G_c(s)Y(s) \qquad (10.105)$$

where

$$\begin{aligned} G_{sp}(s) &= K(b + \frac{1}{sT_i} + c\frac{sT_d}{1+sT_d/N}) \\ G_c(s) &= K(1 + \frac{1}{sT_i} + \frac{sT_d}{1+sT_d/N}) \end{aligned} \qquad (10.106)$$

The linear behavior of the controller is thus characterized by two transfer functions: $G_{sp}(s)$, which gives the signal transmission from the set point to the control variable, and $G_c(s)$, which describes the signal transmission from the process output to the control variable.

Notice that the signal transmission from the process output to the control signal is different from the signal transmission from the set point to the control signal if either set point weighting parameter $b \neq 1$ or $c \neq 1$. The PID controller then has two degrees of freedom.

Another way to express this is that the set point parameters make it possible to modify the zeros in the signal transmission from set point to control signal.

The PID controller is thus a simple control algorithm that has seven parameters: controller gain K, integral time T_i, derivative time T_d, maximum derivative gain N, set point weightings b and c, and filter time constant T_f. Parameters K, T_i and T_d are the primary parameters that are normally discussed. Parameter N is a constant, whose value typically is between 5 and 20. The set point weighting parameter b is often 0 or 1, although it is quite useful to use different values. Parameter c is mostly zero in commercial controllers.

The Standard Form

The controller given by Equations 10.105 and 10.106 is called the *standard form*, or the ISA (Instrument Society of America) form. The standard form admits complex zeros, which is useful when controlling systems with oscillatory poles. The parameterization given in Equation 10.106 is the normal one. There are, however, also other parameterizations.

The Parallel Form

A slight variation of the standard form is the *parallel form*, which is described by

$$\begin{aligned} U(s) &= k[bY_{sp}(s) - Y(s)] + \frac{k_i}{s}[Y_{sp}(s) - Y(s)] \\ &+ \frac{k_d s}{1+sT_{df}}[cY_{sp}(s) - Y(s)] \end{aligned} \qquad (10.107)$$

This form has the advantage that it is easy to obtain pure proportional, integral or derivative control simply by setting appropriate parameters to zero. The interpretation of T_i and T_d as integration time and prediction horizon is, however, lost in this representation. The parameters of the controllers given by Equations 10.105 and 10.107 are related by

$$\begin{aligned} k &= K \\ k_i &= \frac{K}{T_i} \\ k_d &= KT_d \end{aligned} \qquad (10.108)$$

Use of the different forms causes considerable confusion, particularly when parameter $1/k_i$ is called integral time and k_d derivative time.

The form given by Equation 10.107 is often useful in analytical calculations because the parameters appear linearly. However, the parameters do not have nice physical interpretations.

Series Forms

If $T_i > 4T_d$ the transfer function $G_c(s)$ can be written as

$$G_c'(s) = K'\left(1 + \frac{1}{sT_i'}\right)(1 + sT_d') \qquad (10.109)$$

This form is called the series form. If $N = 0$ the parameters are related to the parameters of the parallel form in the following way:

$$\begin{aligned} K &= K'\frac{T_i' + T_d'}{T_i'} \\ T_i &= T_i' + T_d' \end{aligned}$$

$$T_d = \frac{T_i' T_d'}{T_i' + T_d'} \qquad (10.110)$$

The inverse relation is

$$\begin{aligned} K' &= \frac{K}{2}\left(1 + \sqrt{1 - 4T_d/T_i}\right) \\ T_i' &= \frac{T_i}{2}\left(1 + \sqrt{1 - 4T_d/T_i}\right) \\ T_d' &= \frac{T_i}{2}\left(1 - \sqrt{1 - 4T_d/T_i}\right) \end{aligned} \qquad (10.111)$$

Similar, but more complicated, formulas are obtained for $N \neq 0$. Notice that the parallel form admits complex zeros while the series form has real zeros.

The parallel form given by Equations 10.105 and 10.106 is more general. The series form is also called the classical form because it is obtained naturally when a controller is implemented as automatic reset. The series form has an attractive interpretation in the frequency domain because the zeros of the feedback transfer function are the inverse values of T_i' and T_d'. Because of tradition, the form of the controller remained unchanged when technology changed from pneumatic via electric to digital.

It is important to keep in mind that different controllers may have different structures. This means that if a controller in a certain control loop is replaced by another type of controller, the controller parameters may have to be changed. Note, however, that the series and parallel forms differ only when both the integral and the derivative parts of the controller are used.

The parallel form is the most general form because pure proportional or integral action can be obtained with finite parameters. The controller can also have complex zeros. In this way it is the most flexible form. However, it is also the form where the parameters have little physical interpretation. The series form is the least general because it does not allow complex zeros in the feedback path.

Velocity Algorithms

The PID controllers given by Equations 10.105, 10.107 and 10.109 are called positional algorithms because the output of the algorithms is the control variable. In some cases it is more natural to let the control algorithm generate the rate of change of the control signal. Such a control law is called a velocity algorithm. In digital implementations, velocity algorithms are also called incremental algorithms.

Many early controllers that were built around motors used velocity algorithms. Algorithms and structure were often retained by the manufacturers when technology was changed in order to have products that were compatible with older equipment. Another reason is that many practical issues, like windup protection and bumpless parameter changes, are easy to implement using the velocity algorithm.

A velocity algorithm cannot be used directly for a controller without integral action because such a controller cannot keep the stationary value. The system will have an unstable mode, an integrator, that is canceled. Special care must therefore be exercised for velocity algorithms that allow the integral action to be switched off.

10.5.4 Nonlinear Issues

So far we have discussed only the linear behavior of the PID controller. There are several nonlinear issues that also must be considered. These include effects of actuator saturation, mode switches, and parameter changes.

Actuator Saturation and Windup

All actuators have physical limitations, a control valve cannot be more than fully open or fully closed, a motor has limited velocity, etc. This has severe consequences for control. Integral action in a PID controller is an unstable mode. This does not cause any difficulties when the loop is closed. The feedback loop will, however, be broken when the actuator saturates because the output of the saturating element is then not influenced by its input. The unstable mode in the controller may then drift to very large values. When the actuator desaturates it may then take a long time for the system to recover. It may also happen that the actuator bounces several times between high and low values before the system recovers.

Integrator windup is illustrated in Figure 10.52, which shows simulation of a system where the process dynamics is a saturation at a level of ± 0.1 followed by a linear system with the transfer function

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

The controller is a PI controller with gain $K = 0.27$ and $T_i = 7.5$. The set point is a unit step. Because of the saturation in the

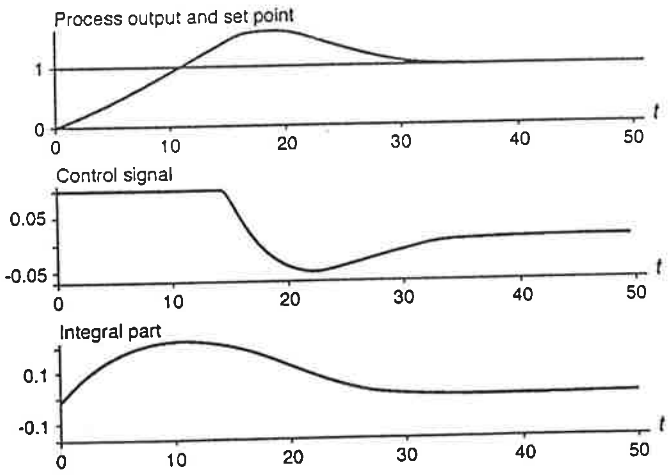


Figure 10.52 Simulation that illustrates integrator windup.

actuator, the control signal saturates immediately when the step is applied. The control signal then remains at the saturation level and the feedback is broken. The integral part continues to increase because the error is positive. The integral part starts to decrease when the output equals the set point, but the output remains saturated because of the large integral part. The output finally decreases around time $t = 14$ when the integral part has decreased sufficiently. The system then settles. The net effect is that there is a large overshoot. This phenomenon, which was observed experimentally very early, is called "integrator windup." Many so-called anti-windup schemes for avoiding windup have been developed; conditional integration and tracking are two common methods.

Conditional Integration

Integrator windup can be avoided by using integral action only when certain conditions are fulfilled. Integral action is thus switched off when the actuator saturates, and it is switched on again when it desaturates. This scheme is easy to implement, but it leads to controllers with discontinuities. Care must also be exercised when formulating the switching logic so that the system does not come to a state where integral action is never used.

Tracking

Tracking or back calculation is another way to avoid windup. The idea is to make sure that the integral is kept at a proper value when the actuator saturates so that the controller is ready to resume action as soon as the control error changes. This can be done as shown in Figure 10.53. The actuator output is measured

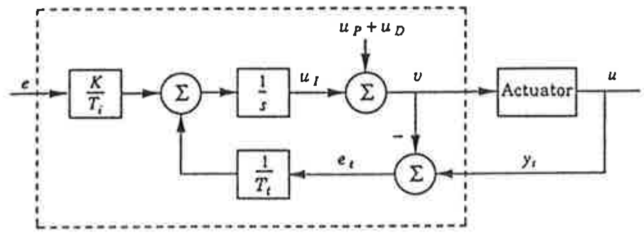


Figure 10.53 PID controller that avoids windup by tracking.

and the signal e_t, which is the difference between the input v and the output u of the actuator, is formed. The signal e_t is different from zero when the actuator saturates. The signal e_t is then fed back to the integrator. The feedback does not have any effect when the actuator does not saturate because the signal e_t is then zero. When the actuator saturates, the feedback drives the integrator output to a value such that the error e_t is zero.

Figure 10.54 illustrates the effect of using the anti-windup scheme. The simulation is identical to the one in Figure 10.52, and the curves from that figure are copied to illustrate the properties of the system. Notice the drastic difference in the behavior of the system. The control signal starts to decrease before the output reaches the set point. The integral part of the controller is also initially driven towards negative values.

The signal y_t may be regarded as an external signal to the controller. The PID controller can then be represented as a block with three inputs, y_{sp}, y and y_t, and one output v, and the anti-windup scheme can then be shown as in Figure 10.55. Notice that tracking is disabled when the signals y_t and v are the same.

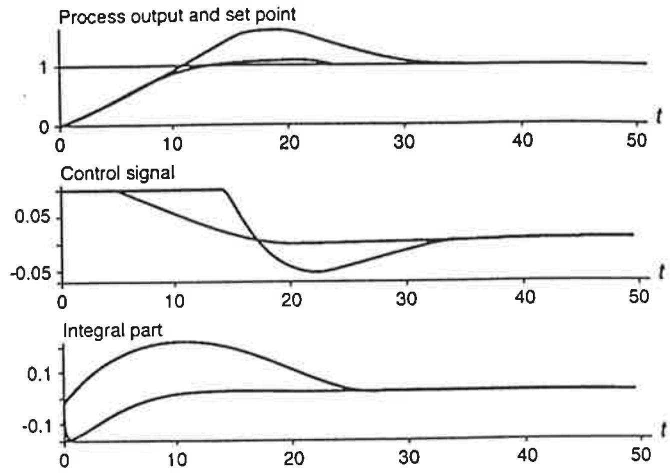


Figure 10.54 Simulation of PID controller with tracking. For comparison, the response for a system without windup protection is also shown. Compare with Figure 10.52.

The signal y_t is called the tracking signal because the output of

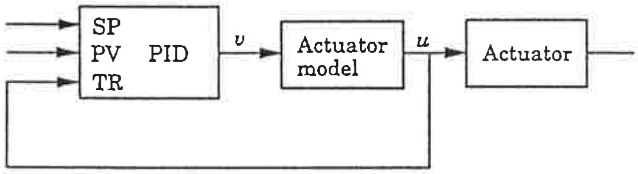


Figure 10.55 Anti-windup in PID controller with tracking input.

the controller tracks this signal. The time constant T_t is called the tracking time constant.

The configuration with a tracking input is very useful when several different controllers are combined to build complex systems. One example is when controllers are coupled in parallel or when selectors are used.

The tracking time constant influences the behavior of the system as shown in Figure 10.56. The values of the tracking constant are 1, 5, 20, and 100. The system recovers faster with smaller

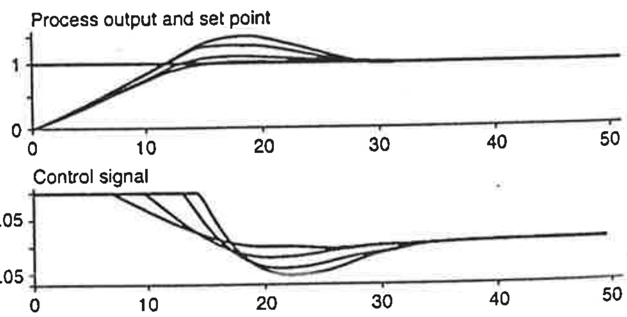


Figure 10.56 Effect of the tracking time constant on the anti-windup. The values of the tracking time constant are 1, 5, 20 and 100.

tracking constants. It is, however, not useful to make the time constant too small because tracking may then be introduced accidentally by noise. It is reasonable to choose $T_t < T_i$ for a PI controller and $T_d < T_t < T_i$ for a PID controller.

The Proportional Band

Let u_{max} and u_{min} denote the limits of the control variable. The proportional band K_p of the controller is then

$$K_p = \frac{u_{max} - u_{min}}{K}$$

This is sometimes used instead of the gain of the controller; the value is often expressed in percent (%).

For a PI controller, the values of the process output that correspond to the limits of the control signal are given by

$$\begin{aligned} y_{max} &= by_{sp} + \frac{u_I - u_{max}}{K} \\ y_{min} &= by_{sp} + \frac{u_I - u_{min}}{K} \end{aligned}$$

The controller operates linearly only if the process output is in the range (y_{min}, y_{max}). The controller output saturates when the predicted output is outside this band. Notice that the proportional band is strongly influenced by the integral term. A good insight into the windup problem and anti-windup schemes is obtained by investigating the proportional band. To illustrate this, Figure 10.57 shows the same simulation as Figure 10.52, but the proportional band is now also shown. The figure shows that

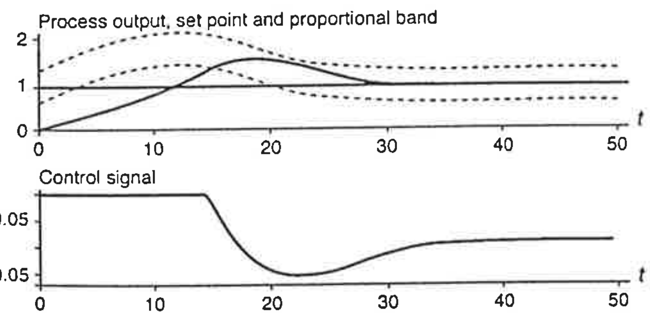


Figure 10.57 Proportional band for simulation in Figure 10.52.

the output is outside the proportional band initially. The control signal is thus saturated immediately. The signal desaturates as soon as the output leaves the proportional band. The large overshoot is caused by windup, which increases the integral when the output saturates.

Anti-Windup in Controller on Series Form

A special method is used to avoid windup in controllers with a series implementation. Figure 10.58 shows a block diagram of the system. The idea is to make sure that the integral term that represents the automatic reset is always inside the saturation limits. The proportional and derivative parts do, however, change the output directly. It is also possible to treat the input to the saturation as an external tracking signal.

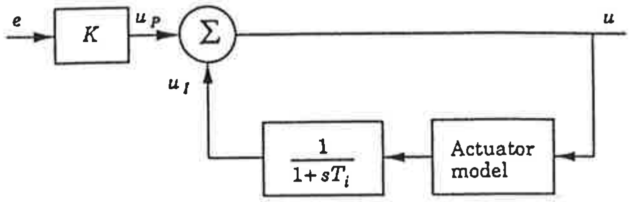


Figure 10.58 A scheme for avoiding windup in a controller with a series implementation.

Notice that the tracking time constant in the controller in Figure 10.58 is equal to the integration time. Better performance can often be obtained with smaller values. This is a limitation of the scheme in Figure 10.58.

Anti-Windup in Velocity Algorithms

In a controller that uses a velocity algorithm we can avoid windup simply by limiting the input to the integrator. The behavior of the system is then similar to a controller with conditional integration.

Mode Switches

Most PID controllers can be operated in one of two modes, manual or automatic. So far we have discussed the automatic mode. In the manual mode the controller output is manipulated directly. This is often done by two buttons labeled "increase" and "decrease". The output is changed with a given rate when a button is pushed. To obtain this function the buttons are connected to the output via an integrator.

It is important that the system be implemented in such a way that there are no transients when the modes are switched. This is very easy to arrange in a controller based on a velocity algorithm, where the same integrator is used in both modes.

It is more complicated to obtain bumpless parameter changes in the other implementations. It is often handled via the tracking mode.

Parameter Changes

Switching transients may also occur when parameters are changed. Some transients cannot be avoided, but others are implementation dependent. In a proportional controller it is unavoidable to have transients if the gain is changed when the control error is different from zero.

For controllers with integral action, it is possible to avoid switching transients even if the parameters are changed when the error is not zero, provided that the controller is implemented properly.

If integral action is implemented as

$$\begin{aligned} \frac{dx}{dt} &= e \\ I &= \frac{K}{T_i}x \end{aligned}$$

there will be a transient whenever K or T_i is changed when $x \neq 0$.

If the integral part is realized as

$$\begin{aligned}\frac{dx}{dt} &= \frac{K}{T_i}e \\ I &= x\end{aligned}$$

we find that the transient is avoided. This is a manifestation that linear time-varying systems do not commute.

10.5.5 Digital Implementation

Most controllers are implemented using digital controllers. In this handbook several chapters deal with these issues. Here we will summarize some issues of particular relevance to PID control. The following operations are performed when a controller is implemented digitally:

- Step 1. Wait for clock interrupt.
- Step 2. Read analog input.
- Step 3. Compute control signal.
- Step 4. Set analog output.
- Step 5. Update controller variables.
- Step 6. Go to 1.

To avoid unnecessary delays, it is useful to arrange the computations so that as many as possible of the calculations are performed in Step 5. In Step 3 it is then sufficient to do two multiplications and one addition.

When computations are based on sampled data, it is good practice to introduce a prefilter that effectively eliminates all frequencies above the Nyquist frequency, $f_N = \pi/h$, where h is the sampling period. If this is not done, high-frequency disturbances may be aliased so that they appear as low-frequency disturbances. In commercial PID controllers this is often done by a first-order system.

Discretization

So far we have characterized the PID controller as a continuous time system. To obtain a computer implementation we have to find a discrete time approximation. There are many ways to do this. Refer to the section on digital control for a general discussion; here we do approximations specifically for the PID controller. We will discuss discretization of the different terms separately. The sampling instants are denoted as t_k where $k = 0, 1, 2, \ldots$. It is assumed that the sampling instants are equally spaced. The sampling period is denoted by h. The proportional action, which is described by

$$u_p = K(by_{sp} - y)$$

is easily discretized by replacing the continuous variables with their sampled versions. This gives

$$u_p(t_k) = K\left(by_{sp}(t_k) - y(t_k)\right) \tag{10.112}$$

The integral term is given by

$$u_I(t) = \frac{K}{T_i}\int_0^t e(\tau)d\tau$$

Differentiation with respect to time gives

$$\frac{du_I}{dt} = \frac{K}{T_i}e$$

There are several ways to discretize this equation. Approximating the derivative by a forward difference gives

$$u_I(t_{k+1}) = u_I(t_k) + \frac{Kh}{T_i}e(t_k) \tag{10.113}$$

If the derivative is approximated by a backward difference we get instead

$$u_I(t_k) = u_I(t_{k-1}) + \frac{Kh}{T_i}e(t_k) \tag{10.114}$$

Another possibility is to approximate the integral by the trapezoidal rule, which gives

$$u_I(t_{k+1}) = u_I(t_k) + \frac{Kh}{T_i}\frac{e(t_{k+1})+e(t_k)}{2} \tag{10.115}$$

Yet another method is called ramp equivalence. This method gives exact outputs at the sampling instants if the input signal is continuous and piecewise linear between the sampling instants. In this particular case, the ramp equivalence method gives the same approximation of the integral term as the Tustin approximation. The derivative term is given by

$$\frac{T_d}{N}\frac{du_D}{dt} + u_D = -KT_d\frac{dy}{dt}$$

This equation can be approximated in the same way as the integral term.

The forward difference approximation is

$$\begin{aligned}u_D(t_{k+1}) &= \left(1 - \frac{Nh}{T_d}\right)u_D(t_k) - KN\left(y(t_{k+1})\right. \\ &\quad \left.-y(t_k)\right)\end{aligned} \tag{10.116}$$

The backward difference approximation is

$$\begin{aligned}u_D(t_k) &= \frac{T_d}{T_d + Nh}u_D(t_{k-1}) \\ &\quad - \frac{KT_dN}{T_d + Nh}\left(y(t_k) - y(t_{k-1})\right)\end{aligned} \tag{10.117}$$

Tustin's approximation gives

$$\begin{aligned}u_D(t_k) &= \frac{2T_d - Nh}{2T_d + Nh}u_D(t_{k-1}) \\ &\quad - \frac{2KT_dN}{2T_d + Nh}\left(y(t_k) - y(t_{k-1})\right)\end{aligned} \tag{10.118}$$

The ramp equivalence approximation gives

$$\begin{aligned}u_D(t_k) &= e^{-Nh/T_d}u_D(t_{k-1}) \\ &\quad - \frac{KT_d(1 - e^{-Nh/T_d})}{h} \\ &\quad \left(y(t_k) - y(t_{k-1})\right)\end{aligned} \tag{10.119}$$

Unification

The approximations of the integral and derivative terms have the same form, namely

$$\begin{aligned} u_I(t_k) &= u_I(t_{k-1}) + b_{i1}e(t_k) + b_{i2}e(t_{k-1}) \\ u_D(t_k) &= a_d u_D(t_{k-1}) - b_d(y(t_k) - y(t_{k-1})) \end{aligned} \tag{10.120}$$

The parameters for the different approximations are given in Table 10.7.

TABLE 10.7 Parameters for the Different Approximations.

	Forward	Backward	Tustin	Ramp Equivalence
b_{i1}	0	$\frac{Kh}{T_i}$	$\frac{Kh}{2T_i}$	$\frac{Kh}{2T_i}$
b_{i2}	$\frac{Kh}{T_i}$	0	$\frac{Kh}{2T_i}$	$\frac{Kh}{2T_i}$
a_d	$1 - \frac{Nh}{T_d}$	$\frac{T_d}{T_d+Nh}$	$\frac{2T_d-Nh}{2T_d+Nh}$	e^{-Nh/T_d}
b_d	KN	$\frac{KT_dN}{T_d+Nh}$	$\frac{2KT_dN}{2T_d+Nh}$	$\frac{KT_d(1-e^{-Nh/T_d})}{h}$

The controllers obtained can be written as

$$\begin{aligned} u(t_k) = {} & t_0 y_{sp}(t_k) + t_1 y_{sp}(t_{k-1}) + t_2 y_{sp}(t_{k-2}) \\ & - s_0 y(t_k) - s_1 y(t_{k-1}) - s_2 y(t_{k-2}) \\ & + (1 + a_d)u(t_{k-1}) - a_d u(t_{k-2}) \end{aligned} \tag{10.121}$$

where

$$\begin{aligned} s_0 &= K + b_{i1} + b_d \\ s_1 &= -K(1 + a_d) - b_{i1}a_d + b_{i2} - 2b_d \\ s_2 &= Ka_d - b_{i2}a_d + b_d \\ t_0 &= Kb + b_{i1} \\ t_1 &= -Kb(1 + a_d) - b_{i1}a_d + b_{i2} \\ t_2 &= Kba_d - b_{i2}a_d \end{aligned}$$

Equation 10.121 gives the linear behavior of the controller. To obtain the complete controller we have to add the anti-windup feature and facilities for changing modes and parameters.

Discussion

There is no significant difference between the different approximations of the integral term. The approximations of the derivative term have, however, quite different properties.

The approximations are stable when $|a_d| < 1$. For the forward difference approximation, this implies that $T_d > Nh/2$. The approximation is thus unstable for small values of T_d. The other approximations are stable for all values of T_d. Tustin's approximation and the forward difference method give negative values of a_d if T_d is small. This is undesirable because the approximation then exhibits ringing. The backward difference approximation gives good results for all values of T_d.

Tustin's approximation and the ramp equivalence approximation give the best agreement with the continuous time case; the backward approximation gives less phase advance; and the forward approximation gives more phase advance. The forward approximation is seldom used because of the problems with instability for small values of derivative time T_d. Tustin's algorithm has the ringing problem for small T_d. Ramp equivalence requires evaluation of an exponential function. The backward difference approximation is used most commonly. The backward difference is well behaved.

Computer Code

As an illustration we give the computer code for a reasonably complete PID controller that has set point weighting, limitation of derivative gain, bumpless parameter changes and anti-windup protection by tracking.

```
Code
Compute controller coefficients
bi=K*h/Ti
ad=(2*Td-N*h)/(2*Td+N*h)
bd=2*K*N*Td/(2*Td+N*h)
a0=h/Tt
Bumpless parameter changes
uI=uI+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y)
Read set point and process output from AD converter
ysp=adin(ch1)
y=adin(ch2)
Compute proportional part
uP=K*(b*ysp-y)
Update derivative part
uD=ad*uD-bd*(y-yold)
Compute control variable
v=uP+uI+uD
u=sat(v,ulow,uhigh)
Command analog output
daout(ch1)
Update integral part with windup protection
uI=uI+bi*(ysp-y)+ao*(u-v)
yold=y
```

Precomputation of the controller coefficients *ad*, *ao*, *bd* and *bi* in Equation 10.121 saves computer time in the main loop. These computations are made only when the controller parameters are changed. The main program is called once every sampling period. The program has three states: *yold*, *uI*, and *uD*. One state variable can be eliminated at the cost of a less readable code.

PID controllers are implemented in many different computers, standard processors as well as dedicated machines. Word length is usually not a problem if general-purpose machines are used. For special-purpose systems, it may be possible to choose word length. It is necessary to have sufficiently long word length to properly represent the integral part.

Velocity Algorithms

The velocity algorithm is obtained simply by taking the difference of the position algorithm

$$\Delta u(t_k) = u(t_k) - u(t_{k-1}) = \Delta u_P(t_k) + \Delta I(t_k) + \Delta D(t_k)$$

The differences are then added to obtain the actual value of the control signal. Sometimes the integration is done externally. The differences of the proportional, derivative and integral terms are obtained from Equations 10.112 and 10.120.

$$\begin{aligned}
\Delta u_P(t_k) &= u_P(t_k) - u_P(t_{k-1}) \\
&= K\left(by_{sp}(t_k) - y(t_k) - by_{sp}(t_{k-1}) + y(t_{k-1})\right) \\
\Delta u_I(t_k) &= u_I(t_k) - u_I(t_{k-1}) \\
&= b_{i1}\, e(t_k) + b_{i2}\, e(t_{k-1}) \\
\Delta u_D(t_k) &= u_D(t_k) - u_D(t_{k-1}) \\
&= a_d \Delta u_D(t_{k-1}) \\
&\quad - b_d\left(y(t_k) - 2y(t_{k-1}) + y(t_{k-2})\right)
\end{aligned}$$

One advantage with the incremental algorithm is that most of the computations are done using increments only. Short word-length calculations can often be used. It is only in the final stage where the increments are added that precision is needed. Another advantage is that the controller output is driven directly from an integrator. This makes it very easy to deal with windup and mode switches. A problem with the incremental algorithm is that it cannot be used for controllers with P or proportional-derivative (PD) action only. Therefore, Δu_P has to be calculated in the following way when integral action is not used:

$$\Delta u_P(t_k) = K\left(by_{sp}(t_k) - y(t_k)\right) + u_b - u(t_{k-1})$$

where u_b is the bias term. When there is no integral action, it is necessary to adjust this term to obtain zero steady-state error.

10.5.6 Uses of PID Control

The PID controller is by far the control algorithm that is most commonly used. It is interesting to observe that in order to obtain a functional controller it is necessary to consider linear and nonlinear behavior of the controller as well as operational issues such as mode switches and tuning. For a discussion of tuning we refer to the chapter "Automatic Tuning of PID Controllers" in this handbook. These questions have been worked out quite well for PID controllers, and the issues involved are quite well understood.

The PID controller in many cases gives satisfactory performance. It can often be used on processes that are difficult to control provided that extreme performance is not required. There are, however, situations when it is possible to obtain better performance by other types of controllers. Typical examples are processes with long relative dead times and oscillatory systems.

There are also cases where PID controllers are clearly inadequate. If we consider the fact that a PI controller always has phase lag and that a PID controller can provide a phase lead of at most 90°, it is clear that neither will work for systems that require more phase advance. A typical example is stabilization of unstable systems with time delays.

A few examples are given as illustrations.

Systems with Long Time Delays

Processes with long time delays are difficult to control. The loop gain with proportional control is very small so integral action is necessary to get good control. Such processes can be controlled by PI controllers, but the performance can be increased by more sophisticated controllers. The reason derivative action is not so useful for processes of this type is that prediction by linear extrapolation of the output is not very effective. To make a proper prediction, it is necessary to take account of the past control signals that have not yet shown up in the output. To illustrate this we consider a process with the transfer function

$$G(s) = \frac{e^{-10s}}{(s+1)^3}$$

The dynamics of this process is dominated by the time delay. A good PI controller that gives a step response without overshoot has a gain $K = 0.27$ and $T_i = 4.8$. The response to set point changes and load disturbances of the system is shown in Figure 10.59. This figure shows the response to a step in the set point at time $t = 0$ and a step at the process input at time $t = 50$.

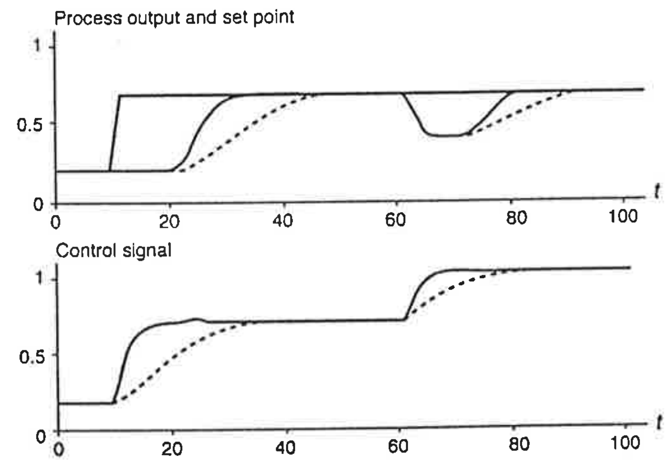


Figure 10.59 Control of a process with long time delays with a PI controller (dashed lines) and a Smith predictor (solid lines).

One way to obtain improved control is to use a controller with a Smith predictor. This controller requires a model of the process. If a model in the form of a first-order system with gain K_p, time constant T, and a time delay L is used, the controller becomes

$$\begin{aligned}
U(s) &= K\left(1 + \frac{1}{sT_i}\right)\Big(E(s) \\
&\quad - \frac{K_p}{1+sT}(1 - e^{-sL})U(s)\Big) \qquad (10.122)
\end{aligned}$$

The controller can predict the output better than a PID controller because of the internal process model. The last term in the right-hand side of Equation 10.122 can be interpreted as the effect on the output of control signals that have been applied in the time interval $(t - T, t)$. Because of the time delay the effect of these signals has not appeared in the output at time t. The improved performance is seen in the simulation in Figure 10.59.

If load disturbance response is evaluated with the integrated absolute error (IAE), we find that the Smith predictor is about 30% better than the PI controller. There are situations when the increased complexity is worth while.

Systems with Oscillatory Modes

Systems with poorly damped oscillatory modes are another case where more complex controllers can outperform PID control. The reason for this is that it pays to have a more complex model in the controller. To illustrate this we consider a system with the transfer function

$$G(s) = \frac{25}{(s+1)(s^2+25)}$$

This system has two complex undamped poles.

The system cannot be stabilized with a PI controller with positive coefficients. To stabilize the undamped poles with a PI controller, it is necessary to have controllers with a zero in the right half-plane. Some damping of the unstable poles can be provided in this way. It is advisable to choose set point weighting $b = 0$ in order to avoid unnecessary excitation of the modes. The response obtained with such a PID controller is shown in Figure 10.60. In this figure a step change in the set point has been introduced at time $t = 0$, and a step change in the load disturbance has been applied at time $t = 20$. The set point weighting b is zero. Because of this we avoid a right half-plane zero in the transfer function from set point to output, and the oscillatory modes are not excited much by changes in the set point. The oscillatory modes are, however, excited by the load disturbance.

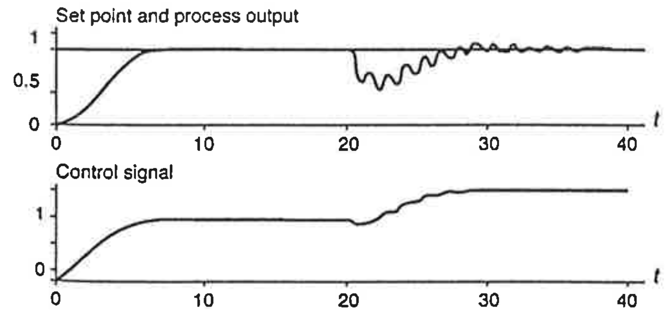


Figure 10.60 Control of an oscillatory system with PI control. The controller parameters are $K = -0.25$, $T_i = -1$ and $b = 0$.

By using a controller that is more complex than a PID controller it is possible to introduce damping in the system. This is illustrated by the simulation in Figure 10.61. The controller has

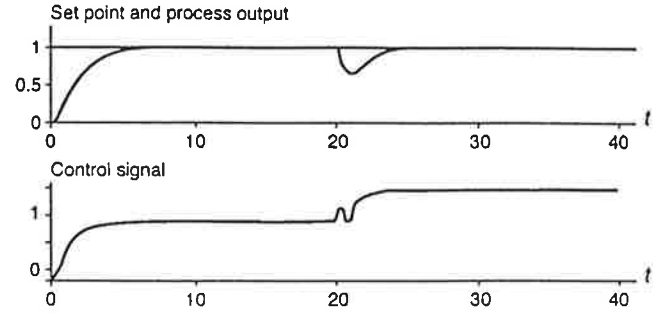


Figure 10.61 Control of the system in Figure 10.60 with a third-order controller.

the transfer functions

$$\begin{aligned} G_c(s) &= \frac{21s^3 - 14s^2 + 65s + 100}{s(s^2 + 16s + 165)} \\ G_{sp}(s) &= \frac{100}{s(s^2 + 16s + 165)} \end{aligned}$$

The transfer function $G_c(s)$ has poles at 0 and $-8 \pm 10.05i$ and zeros at -1 and $0.833 \pm 2.02i$. Notice that the controller has two complex zeros in the right half-plane. This is typical for controllers of oscillatory systems. The controller transfer function can be written as

$$\begin{aligned} G_c(s) &= 0.6061(1 + \frac{1}{s}) \frac{1 - 0.35s + 0.21s^2}{1 + 0.0970s + 0.00606s^2} \\ G_{sp} &= \frac{0.6061}{s} \cdot \frac{1}{1 + 0.0970s + 0.00606s^2} \end{aligned}$$

The controller can thus be interpreted as a PI controller with an additional compensation. Notice that the gain of the controller is 2.4 times larger than the gain of the PI controller used in the simulation in Figure 10.60. This gives faster set point response and a better rejection of load disturbances.

10.5.7 Bottom-Up Design of Complex Systems

Control problems are seldom solved by a single controller. Many control systems are designed using a "bottom up" approach where PID controllers are combined with other components, such as filters, selectors and others.

Cascade Control

Cascade control is used when there are several measured signals and one control variable. It is particularly useful when there are significant dynamics (e.g., long dead times or long time constants) between the control variable and the process variable. Tighter control can then be achieved by using an intermediate measured signal that responds faster to the control signal. Cascade control is built up by nesting the control loops, as shown in Figure 10.62. The system in this figure has two loops. The inner loop is called *the secondary loop*; the outer loop is called *the primary loop*. The reason for this terminology is that the outer loop controls the signal we are primarily interested in. It is also possible to have a cascade control with more nested loops.

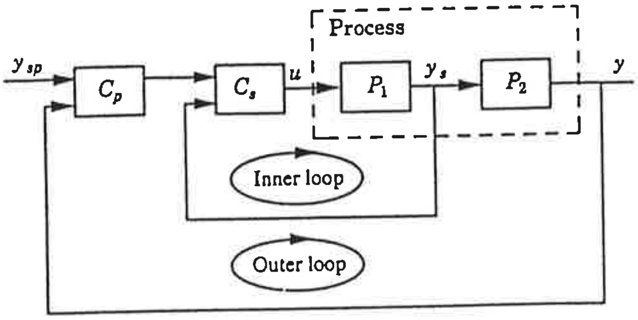


Figure 10.62 Block diagram of a system with cascade control.

The performance of a system can be improved with a number of measured signals, up to a certain limit. If all state variables are measured, it is often not worthwhile to introduce other measured variables. In such a case the cascade control is the same as state feedback.

Feedforward Control

Disturbances can be eliminated by feedback. With a feedback system it is, however, necessary that there be an error before the controller can take actions to eliminate disturbances. In some situations it is possible to measure disturbances before they have influenced the processes. It is then natural to try to eliminate the effects of the disturbances before they have created control errors. This control paradigm is called *feedforward*. The principle is illustrated simply in Figure 10.63. Feedforward can be used for

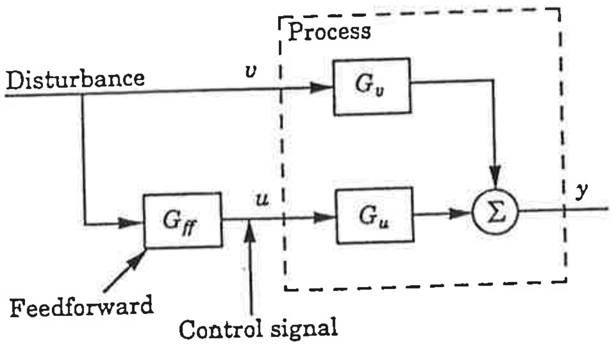


Figure 10.63 Block diagram of a system with feedforward control from a measurable disturbance.

both linear and nonlinear systems. It requires a mathematical model of the process.

As an illustration we consider a linear system that has two inputs, the control variable u and the disturbance v, and one output y. The transfer function from disturbance to output is G_v, and the transfer function from the control variable to the output is G_u. The process can be described by

$$Y(s) = G_u(s)U(s) + G_v(s)V(s)$$

where the Laplace transformed variables are denoted by capital letters. The feedforward control law

$$U(s) = -\frac{G_v(s)}{G_u(s)}V(s)$$

makes the output zero for all disturbances v. The feedforward transfer function thus should be chosen as

$$G_{ff}(s) = -\frac{G_v(s)}{G_u(s)}$$

10.5.8 Selector Control

Selector control can be viewed as the inverse of split range control. In split range, there is one measured signal and several actuators. In selector control, there are many measured signals and only one actuator. A selector is a static device with many inputs and one output. There are two types of selectors: *maximum* and *minimum*. For a maximum selector, the output is the largest of the input signals.

There are situations where several controlled process variables must be taken into account. One variable is the primary controlled variable, but it is also required that other process variables remain within given ranges. Selector control can be used to achieve this. The idea is to use several controllers and to have a selector that chooses the controller that is most appropriate. For example, selector control is used when the primary controlled variable is temperature and we must ensure that pressure does not exceed a certain range for safety reasons.

The principle of selector control is illustrated in Figure 10.64. The primary controlled variable is the process output y. There

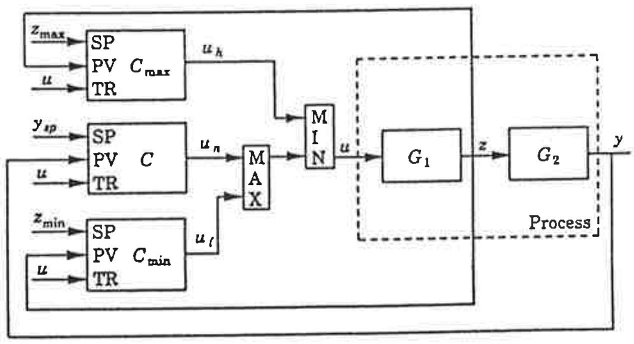


Figure 10.64 Control system with selector control.

is an auxiliary measured variable z that should be kept within the limits z_{min} and z_{max}. The primary controller C has process variable y, setpoint y_{sp} and output u_n. There are also secondary controllers with measured process variables that are the auxiliary variable z and with set points that are bounds of the variable z. The outputs of these controllers are u_h and u_l. The controller C is an ordinary PI or PID controller that gives good control under normal circumstances. The output of the minimum selector is the smallest of the input signals; the output of the maximum selector is the largest of the inputs.

Under normal circumstances the auxiliary variable is larger than the minimum value z_{min} and smaller than the maximum value z_{max}. This means that the output u_h is large and the output u_l is small. The maximum selector, therefore, selects u_n and the minimum selector also selects u_n. The system acts as if the maximum and minimum controller were not present. If the variable z reaches its upper limit, the variable u_h becomes small and is selected by the minimum selector. This means that the control system now attempts to control the variable z and drive it towards its limit. A similar situation occurs if the variable z becomes smaller than z_{min}. To avoid windup, the finally selected control u is used as a tracking signal for all controllers.

References

[1] Åström, K. J. and Hägglund, T., *PID Control—Theory, Design and Tuning*, 2nd ed., Instrument Society of America, Research Triangle Park, NC, 1995.

[2] Åström, K. J., Hägglund, T., Hang, C.C., and Ho, W. K., Automatic tuning and adaptation for PID controllers—a survey, *Control Eng. Pract.*, 1(4), 699–714, 1993.

[3] Åström, K. J., Hang, C.C., Persson, P., and Ho, W. K., Towards intelligent PID control, *Automatica*, 28(1), 1–9, 1992.

[4] Fertik, H. A. Tuning controllers for noisy processes, *ISA Trans.*, 14, 292–304, 1975.

[5] Fertik, H. A. and Ross, C.W., Direct digital control algorithms with anti-windup feature, *ISA Trans.*, 6(4), 317–328, 1967.

[6] Ross, C. W., Evaluation of controllers for deadtime processes, *ISA Trans.*, 16(3), 25–34, 1977.

[7] Seborg, D. E., Edgar, T.F., and Mellichamp, D.A., *Process Dynamics and Control*, Wiley, New York, 1989.

[8] Shinskey, F. G. *Process-Control Systems. Application, Design, and Tuning*, 3rd ed., McGraw-Hill, New York, 1988.

[9] Smith, C. L. and Murrill, P.W., A more precise method for tuning controllers, *ISA Journal*, May, 50–58, 1966.

52

Automatic Tuning of PID Controllers

Tore Hägglund
Department of Automatic Control, Lund Institute of Technology, Lund, Sweden

Karl J. Åström
Department of Automatic Control, Lund Institute of Technology, Lund, Sweden

52.1 Introduction

Methods for automatic tuning of PID controllers have been one of the results of the active research on adaptive control. A result of this development is that the design of PID controllers is going through a very interesting phase. Practically all PID controllers that are designed now have at least some features for automatic tuning. Automatic tuning has also made it possible to generate automatically gain schedules. Many controllers also have adaptation of feedback and feedforward gains.

The most important component of the adaptive controllers and automatic tuning procedures is the design method. The next section presents some of the most common design methods for PID controllers. These design methods are divided into three categories: (1) future based techniques, (2) analytical methods, (3) and methods that are based on optimization.

Section 52.3 treats adaptive techniques. An overview of different uses of these techniques is first presented, followed by a more detailed treatment of automatic tuning, gain scheduling, and adaptive control. Section 52.4 gives an overview of how the adaptive techniques have been used in commercial controllers. References are at the end of the chapter.

52.2 Design Methods

To obtain rational methods for designing controllers it is necessary to deal with specifications and models. In the classical Ziegler-Nichols methods, the process dynamics are characterized by two parameters, a gain and a time. Another approach is used in the analytical design methods, where the controller parameters are obtained from the specifications and the process transfer function by a direct calculation. Optimization methods allow for compromise between several different criteria. These approaches are discussed here.

52.2.1 Specifications

When solving a control problem it is necessary to understand the primary goal of control. Two common control objectives are to follow the setpoint and to reject disturbances. It is also important to have an assessment of the major limitations, which can be system dynamics, nonlinearities, disturbances, or process uncertainty. Typical specifications on a control system may include attenuation of load disturbances, setpoint following, robustness to model uncertainty, and lack of sensitivity to measurement noise.

Attenuation of Load Disturbances:

Attenuation of load disturbances is of primary concern for process control. The disturbances may enter the system in many different ways, but it is often assumed that they enter at the process input. Let e be the error caused by a unit step load disturbance at the process input. Typical quantities used to characterize the error are: maximum error, time to reach maximum, settling time, decay ratio, and the integrated absolute error, which is defined by

$$IAE = \int_0^\infty |e(t)|dt \qquad (52.1)$$

Setpoint Following:

Setpoint following is often less important than load disturbance attenuation for process control applications. Setpoint changes are often only made when the production rate is altered. Furthermore, the response to setpoint changes can be improved by feeding the setpoint through ramping functions or by adjust-

0-8493-8570-9/96/$0.00+$.50

ing the setpoint weightings. Specifications on setpoint following may include requirements on rise time, settling time, decay ratio, overshoot, and steady-state offset for step changes in setpoint.

Robustness to Model Uncertainty

It is important that the controller parameters are chosen in such a way that the closed-loop system is not too sensitive to changes in process dynamics. There are many ways to specify the sensitivity. Many different criteria are conveniently expressed in terms of the Nyquist plot of the loop transfer function, and its distance to the critical point -1. Maximum sensitivity M_s is a good measure since $1/M_s$ is the shortest distance between the Nyquist plot and the critical point. Gain and phase margins are other related measures.

Sensitivity to Measurement Noise:

Care should always be taken to reduce measurement noise by appropriate filtering, since it will be fed into the system through the feedback. It will generate control actions and control errors. Measurement noise is typically of high frequency. The high-frequency gain of a PID controller is

$$K_{hf} = K(1 + N)$$

where K is the controller gain and N is the derivative gain limitation factor. See Chapter 10. Notice that $N = 0$ corresponds to PI control. Multiplication of the measurement noise by K_{hf} gives the fluctuations in the control signal that are caused by the measurement noise. Also notice that there may be a significant difference in K_{hf} for PI and PID control. It is typically an order of magnitude larger for a PID controller, since the gain normally is higher for a PID controller than for a PI controller, and N is typically around 10.

52.2.2 Feature-Based Techniques

The simplest design methods are based on a few features of the process dynamics that are easy to obtain experimentally. Typical time-domain features are static gain K_p, dominant time constant T, and dominant dead time L. They can all be determined from a step response of the process, see Figure 52.1. Static gain K_p, dominant time constant T, and dominant dead time L can be used to obtain an approximate first order plus dead-time model for the process as given in Equation 52.2.

$$G_p(s) = \frac{K_p}{1 + sT} e^{-sL} \tag{52.2}$$

Typical frequency-domain features are static gain K_p, ultimate gain K_u, and ultimate period T_u. They are defined in Figure 52.2.

Ziegler-Nichols Methods

In 1942, Ziegler and Nichols presented two design methods for PID controllers, one time-domain and one frequency-domain method [10]. The methods are based on determination of process dynamics in terms of only two parameters, a gain and a time. The controller parameters are then expressed in terms of these parameters by simple formulas. In both methods, the design specification of quarter amplitude decay ratio was used. The decay ratio is the ratio between two consecutive maxima of the control error after a step change in setpoint or load.

The time-domain method is based on a registration of the open-loop step response of the process. Ziegler and Nichols have given PID parameters directly as functions of a and L, defined in Figure 52.1. These are given in Table 52.1.

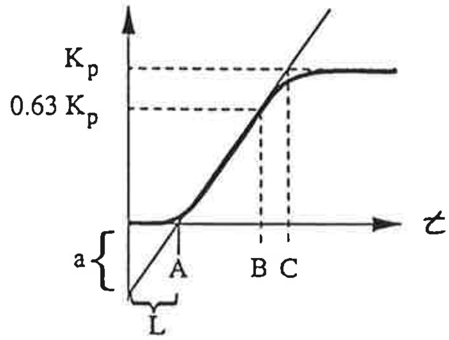


Figure 52.1 Determining a first-order plus dead-time model from a step response. Time constant T can be obtained either as the distance AB or the distance AC.

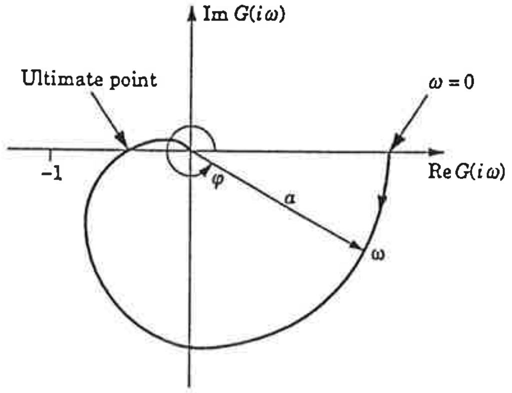


Figure 52.2 Static gain K_p, ultimate gain K_u, and ultimate period T_u defined in the Nyquist diagram. Static gain K_p is the point on the Nyquist plot at $\omega = 0$. Ultimate gain K_u is -1 divided by the ultimate point. Ultimate period T_u is 2π divided by the frequency corresponding to the ultimate point.

TABLE 52.1 Controller parameters obtained from the Ziegler-Nichols step response method.

Controller	K	T_i	T_d
P	$1/a$		
PI	$0.9/a$	$3L$	
PID	$1.2/a$	$2L$	$L/2$

The second method presented by Ziegler and Nichols is based on the frequency response of the process. They have given simple formulas for the parameters of the controller in terms of ultimate gain K_u and the ultimate period T_u. These parameters can be determined in the following way. Connect a controller to the process, set the parameters so that control action is proportional, i.e., $T_i = \infty$ and $T_d = 0$. Increase the gain slowly until the process starts to oscillate. The gain when this occurs is K_u and the period of the oscillation is T_u. The parameters can also be determined approximately by relay feedback as is discussed in Section 52.3. The controller parameters are given in Table 52.2.

TABLE 52.2 Controller parameters obtained from the Ziegler-Nichols frequency response method.

Controller	K	T_i	T_d
P	$0.5K_u$		
PI	$0.4K_u$	$0.8T_u$	
PID	$0.6K_u$	$0.5T_u$	$0.12T_u$

Modifications of the Ziegler-Nichols Methods

The Ziegler-Nichols methods do not give satisfactory control. The reason is that they give closed-loop systems with very poor damping. The design criterion "quarter amplitude decay ratio" corresponds to a relative damping of $\zeta \approx 0.2$ which is much too small for most applications. The maximum sensitivity is also much too large, which means that the closed-loop systems obtained are too sensitive to parameter variations.

The Ziegler-Nichols methods do, however, have the advantage of being very easy to use. Many efforts have therefore been made to obtain tuning methods that retain the simplicity of the Ziegler-Nichols methods but give improved robustness.

Chien, Hrones, and Reswick modified the coefficients in the Ziegler-Nichols methods [3]. In this way, they obtained tuning rules that give better damping. They developed different tuning rules for setpoint changes and load disturbances. Using the two-degrees of freedom structure of the PID controller, there is, however, normally no need to compromise between these two disturbances. If the tuning is performed to give good responses to load disturbances, the setpoint changes can be treated by, e.g., setpoint weighting (see Chapter 10). Table 52.3 gives the controller parameters based on load disturbances.

TABLE 52.3 Controller parameters obtained from the Chien, Hrones and Reswick load disturbance response method.

Overshoot	0%			20%		
Controller	K	T_i	T_d	K	T_i	T_d
P	$0.3/a$			$0.7/a$		
PI	$0.6/a$	$4L$		$0.7/a$	$2.3L$	
PID	$0.95/a$	$2.4L$	$0.42L$	$1.2/a$	$2L$	$0.42L$

Kappa-Tau Tuning

Significantly better tuning rules can be obtained if the process dynamics are described in terms of three parameters instead of two. An early step in this direction was made by Cohen and Coon, who assumed that the process was given by Equation 52.2, which has three parameters [4]. Their design did, however, also give very sensitive systems.

The Kappa-Tau method (see [1]) is a recent method that was developed for automatic tuning. It is given in two versions, one that is based on a step response experiment, and one that is based on the frequency response of the process. In both methods, maximum sensitivity M_s is used as a design variable. The Kappa-Tau method was obtained from extensive simulation investigations of the dominant pole design method applied on typical process control models [1].

In the step response method, the process is characterized by static gain K_p, gain a, and apparent deadtime L (see Figure 52.1). The controller parameters are given as functions of normalized dead time τ, which is defined as

$$\tau = \frac{L}{L+T} \tag{52.3}$$

In the frequency domain method, the process is characterized by static gain K_p, ultimate gain K_u, and ultimate period T_u. The controller parameters are given as functions of gain ratio κ, which is defined as

$$\kappa = \frac{1}{K_p K_u} \tag{52.4}$$

The Kappa-Tau method gives both PI and PID controller parameters. Figure 52.3 shows the PID controller parameters for the frequency response method.

The solid lines in the figure correspond to functions having the form

$$f(\tau) = a_0 e^{a_1\tau + a_2\tau^2}$$

that are fitted to the data. The function parameters are given in Table 52.4.

52.2.3 Analytical Methods

If the process can be described well by a simple model, the controller parameters can be obtained by a direct calculation. This approach is treated in this section.

Pole Placement

If the process is described by a low-order transfer function, a complete pole-placement design can be performed. Consider for example the process described by the second-order model

$$G(s) = \frac{K_p}{(1+sT_1)(1+sT_2)} \tag{52.5}$$

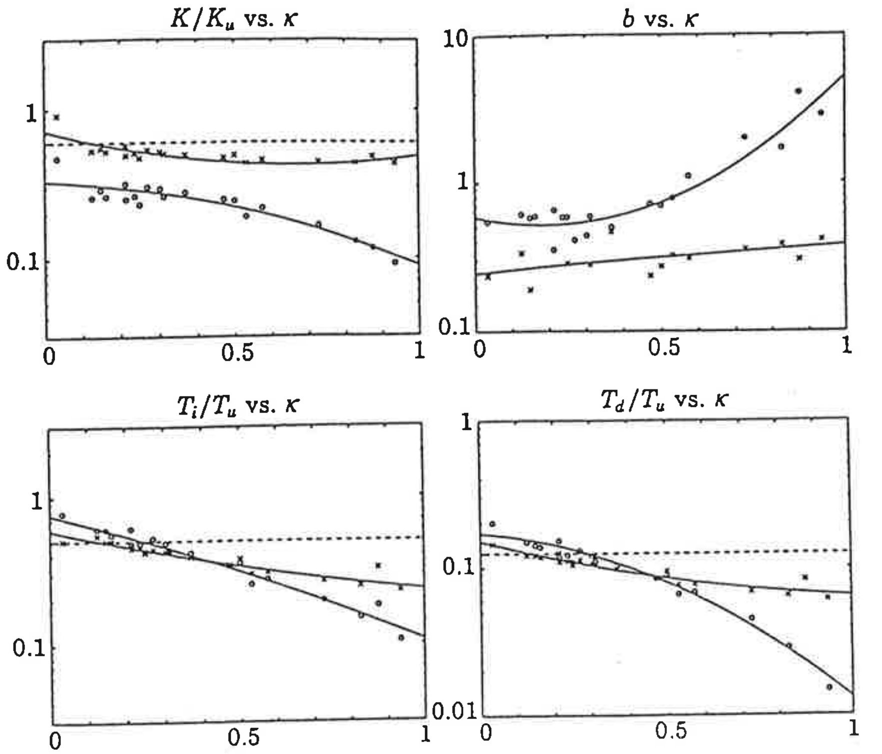


Figure 52.3 Tuning diagrams for PID control based on K_u, T_u and κ. Controller parameters are obtained by applying dominant pole design with $M_s = 1.4$, marked with o, and $M_s = 2$ marked with x, to processes in a test batch. The dashed lines correspond to the Ziegler-Nichols tuning rule.

TABLE 52.4 Tuning formula for PID control based on the Kappa-Tau method. The table gives parameters of functions of the form $f(\tau) = a_0 exp(a_1\tau + a_2\tau^2)$ for the normalized controller parameters.

	$M_s = 1.4$			$M_s = 2.0$		
	a_0	a_1	a_2	a_0	a_1	a_2
K/K_u	0.33	-0.31	-1.0	0.72	-1.6	1.2
T_i/T_u	0.76	-1.6	-0.36	0.59	-1.3	0.38
T_d/T_u	0.17	-0.46	-2.1	0.15	-1.4	0.56
b	0.58	-1.3	3.5	0.25	0.56	-0.12

This model has three parameters. By using a PID controller, which also has three parameters, it is possible to arbitrarily place the three poles of the closed-loop system. The transfer function of the PID controller in parallel form can be written as

$$G_c(s) = \frac{K(1 + sT_i + s^2T_iT_d)}{sT_i}$$

The characteristic equation of the closed-loop system becomes

$$s^3 + s^2\left(\frac{1}{T_1} + \frac{1}{T_2} + \frac{K_pKT_d}{T_1T_2}\right) + s\left(\frac{1}{T_1T_2} + \frac{K_pK}{T_1T_2}\right) + \frac{K_pK}{T_iT_1T_2} = 0$$

A suitable closed-loop characteristic equation of a third-order system is

$$(s + \alpha\omega)(s^2 + 2\zeta\omega s + \omega^2) = 0$$

which contains two dominant poles with relative damping (ζ) and frequency (ω), and a real pole located at $-\alpha\omega$. Identifying the coefficients in these two characteristic equations determines the PID parameters K, T_i and T_d. The solution is

$$K = \frac{T_1T_2\omega^2(1 + 2\zeta\alpha) - 1}{K_p}$$

$$T_i = \frac{T_1T_2\omega^2(1 + 2\zeta\alpha) - 1}{T_1T_2\alpha\omega^3}$$

$$T_d = \frac{T_1T_2\omega(\alpha + 2\zeta) - T_1 - T_2}{T_1T_2(1 + 2\zeta\alpha)\omega^2 - 1}$$

If the model is given by Equation 52.5, any value of ζ and ω can be chosen. There are, however, restrictions on reasonable choices of ω. Criteria like IAE will be smaller the larger the bandwidth is. Pure PI control is obtained for

$$\omega_c = \frac{T_1 + T_2}{T_1T_2(\alpha + 2\zeta)}$$

The choice of ω may be critical. The derivative time, T_d, is negative for $\omega < \omega_c$. The frequency ω_c thus gives a lower bound to the bandwidth. The gain increases rapidly with ω. The upper bound to the bandwidth is given by the validity of the simplified model. With this approach, it still remains to give a suitable value of ω.

λ-Tuning

Let G_p and G_c be the transfer functions of the process and the controller. The closed-loop transfer function obtained with error feedback is then

$$G_0 = \frac{G_p G_c}{1 + G_p G_c}$$

Solving this equation for G_c gives

$$G_c = \frac{1}{G_p} \cdot \frac{G_0}{1 - G_0} \tag{52.6}$$

If the closed-loop transfer function G_0 is specified and G_p is known, it is thus easy to compute G_c.

The method, called λ-tuning, was developed for processes with long dead time L [5]. Consider a process with the transfer function

$$G_p = \frac{K_p}{1 + sT} e^{-sL} \tag{52.7}$$

Assume that the desired closed-loop transfer function is specified as

$$G_0 = \frac{e^{-sL}}{1 + s\lambda T} \tag{52.8}$$

where λ is a tuning parameter. The time constants of the open- and closed-loop systems are the same when $\lambda = 1$. The closed-loop system responds faster than the open-loop system if $\lambda < 1$. It is slower when $\lambda > 1$.

It follows from Equation 52.6 that the controller transfer function becomes

$$G_c = \frac{1 + sT}{K_p(1 + \lambda sT - e^{-sL})}$$

When $L = 0$ this becomes a PI controller with gain $K = 1/(\lambda K_p)$ and integral time $T_i = T$. The sensitivity function obtained with λ-tuning is given by

$$S(s) = 1 - \frac{e^{-sL}}{1 + s\lambda T} = \frac{1 + s\lambda T - e^{-sL}}{1 + s\lambda T}$$

The maximum sensitivity M_s is always less than 2 if the model is correct. With unmodeled dynamics, the sensitivity may be larger. The parameter λ should be small to give a low IAE, but a small value of λ increases the sensitivity.

IMC

The internal model principle is a general method for design of control systems that can be applied to PID control. A block diagram of such a system is shown in Figure 52.4. It is assumed that all disturbances acting on the process are reduced to an equivalent disturbance d at the process output. In the figure, G_m denotes a model of the process, $G_m^\dagger$ is an approximate inverse of G_m, and G_f is a low-pass filter. The name internal model controller derives from the fact that the controller contains a model of the process. This model is connected in parallel with the process.

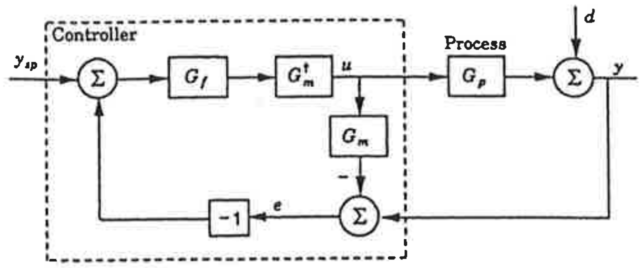


Figure 52.4 Block diagram of a closed-loop system with a controller based on the internal model principle.

If the model matches the process, i.e., $G_m = G_p$, the signal e is equal to the disturbance d for all control signals u. If $G_f = 1$ and $G_m^\dagger$ is an exact inverse of the process, then the disturbance d will be canceled perfectly. The filter G_f is introduced to obtain a system that is less sensitive to modeling errors, and to ensure that the system $G_f G_m^\dagger$ is realizable. A common choice is $G_f(s) = 1/(1 + sT_f)$, where T_f is a design parameter.

The controller obtained by the internal model principle can be represented as an ordinary series controller with the transfer function

$$G_c = \frac{G_f G_m^\dagger}{1 - G_f G_m^\dagger G_m} \tag{52.9}$$

From this expression it follows that controllers of this type cancel process poles and zeros. The controller is normally of high order. Using simple models it is, however, possible to obtain PI or PID controllers. To see this consider a process with the transfer function

$$G_p(s) = \frac{K_p}{1 + sT} e^{-sL}$$

An approximate inverse, where no attempt is made to find an inverse of the time delay, is given by

$$G_m^\dagger(s) = \frac{1 + sT}{K_p}$$

Choosing the filter

$$G_f(s) = \frac{1}{1 + sT_f}$$

and approximating the time delay by

$$e^{-sL} \approx 1 - sL$$

Equation 52.9 now gives

$$G_c(s) = \frac{1 + sT}{K_p s(L + T_f)}$$

which is a PI controller. If the time delay is approximated instead by a first-order Padé approximation

$$e^{-sL} \approx \frac{1 - sL/2}{1 + sL/2}$$

Equation 52.9 gives instead the PID controller

$$G_c(s) = \frac{(1+sL/2)(1+sT)}{K_p s(L+T_f+sT_f L/2)} \approx \frac{(1+sL/2)(1+sT)}{K_p s(L+T_f)}$$

An interesting feature of the internal model controller is that robustness is considered explicitly in the design. Robustness can be adjusted by selecting the filter G_f properly. A trade-off between performance and robustness can be made by using the filter constant as a design parameter.

The IMC method can be designed to give excellent responses to setpoint changes. Since the design method inherently implies that poles and zeros of the plant are canceled, the response to load disturbances may be poor if the canceled poles are slow in comparison with the dominant poles. This is discussed in the next section.

52.2.4 Optimization-Based Methods

A third category of design methods is based on optimization techniques.

Direct Criteria Optimization

Optimization is a powerful tool for design of controllers. The method is conceptually simple. A controller structure with a few parameters is specified. Specifications are expressed as inequalities of functions of the parameters. The specification that is most important is chosen as the function to optimize. The method is well suited for PID controllers where the controller structure and the parameterization are given. There are several pitfalls when using optimization. Care must be exercised when formulating criteria and constraints; otherwise, a criterion will indeed be optimal, but the controller may still be unsuitable because of a neglected constraint. Another difficulty is that the loss function may have many local minima. A third is that the computations required may be excessive. Numerical problems may also arise. Nevertheless, optimization is a good tool that has successfully been used to design PID controllers.

Popular optimization criteria are the integrated absolute error (IAE), the integrated time absolute error (ITAE), and the integrated square error (ISE). They are mostly done for the first-order plus dead-time model as given in Equation 52.2. Many tables that provide controller parameters based on optimization have been published [7].

Modulus Optimum (BO) and Symmetrical Optimum (SO)

Modulus Optimum (BO) and Symmetrical Optimum (SO) are two design methods that are based on optimization. The acronyms BO and SO are derived from the German words Betrags Optimum and Symmetrische Optimum. These methods are based on the idea of finding a controller that makes the frequency response from setpoint to plant output as close to one as possible for low frequencies. If $G(s)$ is the transfer function from the setpoint to the output, the controller is determined in such a way that $G(0) = 1$ and that $d^n|G(i\omega)|/d\omega^n = 0$ at $\omega = 0$ for as many n as possible.

If the closed-loop system is given by

$$G(s) = \frac{\omega_0^2}{s^2 + \sqrt{2}\omega_0 s + \omega_0^2}$$

the first three derivatives of $|G(i\omega)|$ will vanish at the origin. If the transfer function G in the example is obtained by error feedback of a system with the loop transfer function G_{BO}, the loop transfer function is

$$G_{BO}(s) = \frac{G(s)}{1-G(s)} = \frac{\omega_0^2}{s(s+\sqrt{2}\omega_0)}$$

which is the desired loop transfer function for the method called modulus optimum.

If the closed-loop transfer function is given by

$$G(s) = \frac{\omega_0^3}{(s+\omega_0)(s^2+\omega_0 s+\omega_0^2)} \qquad (52.10)$$

the first five derivatives of $|G(i\omega)|$ will vanish at the origin. A system with this closed-loop transfer function can be obtained with a system having error feedback and the loop transfer function

$$G_\ell(s) = \frac{\omega_0^3}{s(s^2+2\omega_0 s+2\omega_0^2)}$$

The closed-loop transfer function (52.10) can also be obtained from other loop transfer functions if a two-degree of freedom controller is used. For example, if a process with the transfer function

$$G_p(s) = \frac{\omega_0^2}{s(s+2\omega_0)}$$

is controlled by a PI controller having parameters $K = 2$, $T_i = 2/\omega_0$ and $b = 0$, the loop transfer function becomes

$$G_{SO} = \frac{\omega_0^2(2s+\omega_0)}{s^2(s+2\omega_0)} \qquad (52.11)$$

The symmetric optimum aims at obtaining the loop transfer function given by Equation 52.11. Notice that the Bode diagram of this transfer function is symmetrical around the frequency $\omega = \omega_0$. This is the motivation for the name symmetrical optimum.

The methods BO and SO can be called loop-shaping methods since both methods try to obtain a specific loop transfer function. The design methods can be described as follows. It is first established which of the transfer functions, G_{BO} or G_{SO}, is most appropriate. The transfer function of the controller $G_c(s)$ is then chosen so that $G_\ell(s) = G_c(s)G_p(s)$, where G_ℓ is the chosen loop transfer function.

52.2.5 Cancellation of Process Poles

The PID controller has two zeros. Many design methods choose these zeros so that they cancel one or two of the dominant process poles. This often results in a simple design as in IMC or SO optimization. The response to set-point changes is good, but the methods will often result in poor responses to load disturbances. An exception to this is in the case of large dead time, in which case the settling time is already quite long relative to the time constant being canceled.

Figure 52.5 illustrates the problem. A process with a time constant $T = 10$ is controlled with a PI controller with integral time $T_i = 10$ and a suitable controller gain. The response to setpoint changes is good, but the load disturbance response is very sluggish. The figure also shows a retuned controller, where pole cancellation is avoided.

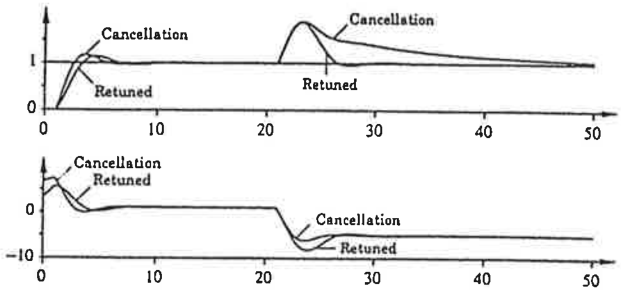


Figure 52.5 Simulation of a closed-loop system obtained by pole cancellation. The process transfer function is $G(s) = e^{-s}/(10s + 1)$, and the controller parameters are $K = 6.67$ and $T_i = 10$. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u. The figure also shows the responses to a retuned controller with $K = 6.67$, $T_i = 3$ and $b = 0.5$.

52.3 Adaptive Techniques

This section gives an overview of adaptive techniques. It starts with a discussion of uses of the different techniques, followed by a more detailed presentation of automatic tuning, gain scheduling, and adaptive control. The section ends with a presentation of how the adaptive techniques have been used in industrial controllers.

52.3.1 Use of the Adaptive Techniques

The word *adaptive techniques* is used to cover auto-tuning, gain scheduling and adaptation. Although research on adaptive techniques has almost exclusively focused on adaptation, experience has shown that auto-tuning and gain scheduling have much wider industrial applicability. Figure 52.6 illustrates the appropriate use of the different techniques.

Controller performance is the first issue to consider. If requirements are modest, a controller with constant parameters and conservative tuning can be used. Other solutions should be considered when higher performance is required.

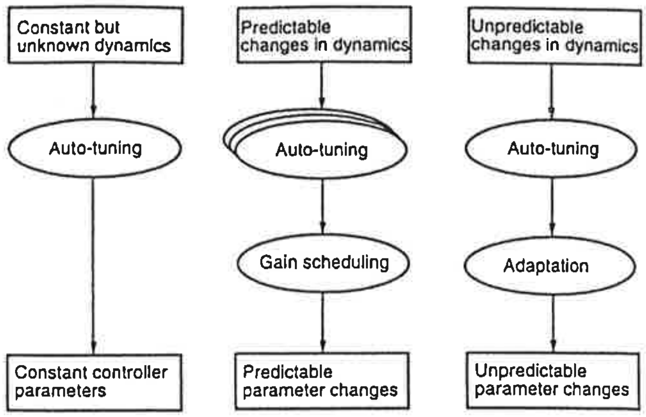


Figure 52.6 When to use different adaptive techniques.

If the process dynamics are constant, a controller with constant parameters should be used. The parameters of the controller can be obtained by auto-tuning.

If the process dynamics or the character of the disturbances are changing it is useful to compensate for these changes by changing the controller. If the variations can be predicted from measured signals, gain scheduling should be used since it is simpler and gives superior and more robust performance than continuous adaptation. Typical examples are variations caused by nonlinearities in the control loop. Auto-tuning can be used to build up the gain schedules automatically.

There are also cases where the variations in process dynamics are not predictable. Typical examples are changes due to unmeasurable variations in raw material, wear, fouling etc. These variations cannot be handled by gain scheduling but must be dealt with by adaptation. An auto-tuning procedure is often used to initialize the adaptive controller. It is then sometimes called pre-tuning or initial tuning.

52.3.2 Automatic Tuning

Automatic tuning (or auto-tuning) is a method where a controller is tuned automatically on demand from the user. Typically the user will either push a button or send a command to the controller. Automatic tuning is sometimes called tuning on demand or one-shot tuning.

Automatic tuning can also be performed by external devices which are connected to the control loop during the tuning phase. Since these devices are supposed to work together with controllers from different manufacturers, they must be provided with quite a lot of information about the controller structure and parameterization in order to provide appropriate controller parameters. Such information includes controller structure (series or parallel form), sampling rate, filter time constants, and units of the different controller parameters (gain or proportional band, minutes or seconds, time or repeats/time) (see Chapter 10).

The automatic tuning procedures can be divided into methods that are based on step response experiments, and methods based on frequency response experiments.

Step Response Methods

Most methods for automatic tuning of PID controllers are based on step response experiments. When the operator wishes to tune the controller, an open-loop step response experiment is performed. A process model is then obtained from the step response, and controller parameters are determined. This is usually done using simple look-up tables as in the Ziegler-Nichols method.

The greatest difficulty in carrying out the tuning automatically is in selecting the amplitude of the step. The user naturally wants the disturbance to be as small as possible so that the process is not disturbed more than necessary. On the other hand, it is easier to determine the process model if the disturbance is large. The result of this dilemma is usually that the operator himself has to decide how large the step in the control signal should be.

Controllers with automatic tuning which are based on this technique have become very common in the last few years. This is especially true of temperature controllers.

The Relay Autotuner

Frequency-domain characteristics of the process can be determined from experiments with relay feedback in the following way. When the controller is to be tuned, a relay with hysteresis is introduced in the loop, and the PID controller is temporarily disconnected, see Figure 52.7. For large classes of processes, relay feedback gives an oscillation with period close to the ultimate frequency ω_u. The gain of the transfer function at this frequency is also easy to obtain from amplitude measurements. Describing function analysis can be used to determine the process characteristics. The describing function of a relay with hysteresis is

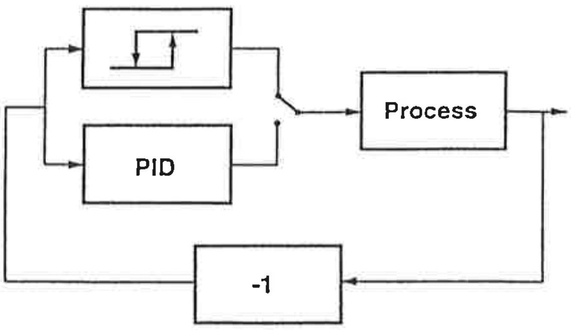


Figure 52.7 The relay autotuner. In the tuning mode the process is connected to relay feedback.

$$N(a) = \frac{4d}{\pi a}\left(\sqrt{1-\left(\frac{\epsilon}{a}\right)^2} - i\,\frac{\epsilon}{a}\right)$$

where d is the relay amplitude, ϵ the relay hysteresis and a is the amplitude of the input signal. The negative inverse of this describing function is a straight line parallel to the real axis, see Figure 52.8. The oscillation corresponds to the point where the negative inverse describing function crosses the Nyquist curve of the process, i.e., where

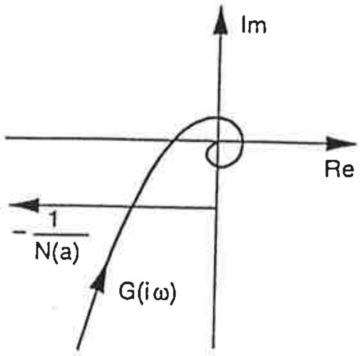


Figure 52.8 The negative inverse describing function of a relay with hysteresis $-1/N(a)$ and a Nyquist curve $G(i\omega)$.

$$G(i\omega) = -\frac{1}{N(a)}$$

Since $N(a)$ is known, $G(i\omega)$ can be determined from the amplitude a and the frequency ω of the oscillation.

52.3.3 Gain Scheduling

By gain scheduling we mean a system where controller parameters are changed depending on measured operating conditions. The scheduling variable can, for instance, be the measurement signal, controller output or an external signal. For historical reasons the word gain scheduling is used even if other parameters like derivative time or integral time are changed. Gain scheduling is a very effective way of controlling systems whose dynamics change with the operating conditions.

The notion of gain scheduling was originally used for flight control systems, but it is being used increasingly in process control. It is, in fact, a standard ingredient in some single-loop PID controllers. For process control applications, significant improvements can be obtained by using just a few sets of controller parameters.

52.3.4 Adaptive Control

An adaptive controller is a controller whose parameters are continuously adjusted to accommodate changes in process dynamics and disturbances. Parameters can be adjusted directly or indirectly via estimation of process parameters. There is a large number of both direct and indirect methods available. Adaptation can be applied both to feedback and feedforward control parameters. Adaptive controls can be described conveniently in terms of the methods used for modeling and control design.

Model-Based Methods

All indirect systems can be represented by the block diagram in Figure 52.9. There is a parameter estimator that determines the parameters of the model based on observations of process inputs and outputs. There is also a design block that

TABLE 52.5 Industrial adaptive process controllers.

Manufacturer	Controller	Automatic tuning	Gain scheduling	Adaptive feedback	Adaptive feedforward
Bailey Controls	CLC04	Step	Yes	Model based	–
Control Techniques	Expert controller	Ramps	–	Model based	–
Fisher Controls	DPR900	Relay	Yes	–	–
	DPR910	Relay	Yes	Model based	Model based
Foxboro	Exact	Step	–	Rule based	–
Fuji	CC-S:PNA 3	Steps	Yes	–	–
Hartmann & Braun	Protronic P	Step	–	–	–
	Digitric P	Step	–	–	–
Honeywell	UDC 6000	Step	Yes	Rule based	–
Alfa Laval Automation	ECA40	Relay	Yes	–	–
	ECA400	Relay	Yes	Model based	Model based
Siemens	SIPART DR22	Step	Yes	–	–
Toshiba	TOSDIC-215D	PRBS	Yes	Model based	–
	EC300	PRBS	Yes	Model based	–
Turnbull Control Systems	TCS 6355	Steps	–	Model based	–
Yokogawa	SLPC-171,271	Step	Yes	Rule based	–
	SLPC-181,281	Step	Yes	Model based	–

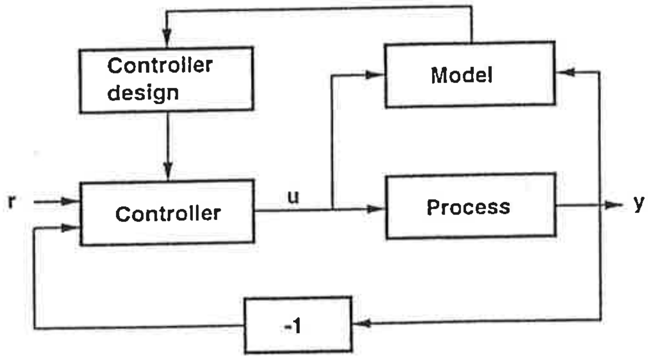


Figure 52.9 Block diagram of indirect systems.

computes controller parameters from the model parameters. The parameters can either be estimated recursively or batch-wise.

Rule-Based Methods

In the direct methods, the key issues are to find suitable features that characterize relevant properties of the closed-loop system and appropriate ways of changing the controller parameters so that the desired properties are obtained.

The majority of the PID controllers in industry are tuned manually by instrument engineers. The tuning is done based on past experience and heuristics. By observing the pattern of the closed-loop response to a set-point change, the instrument engineers use heuristics to directly adjust the controller parameters. The heuristics have been captured in tuning charts that show the responses of the system for different parameter values. A considerable insight into controller tuning can be developed by studying such charts and performing simulations. The heuristic rules have also been captured in knowledge bases in the form of crisp or fuzzy rules. Rules of this type are used in several commercial adaptive controllers. Most products will wait for set-point changes or major load disturbances. When these occur, properties like damping, overshoot, period of oscillations and static gains are estimated. Based on these properties, rules for changing the controller parameters to meet desired specifications are executed.

52.3.5 Adaptive Feed Forward

Feedforward control deserves special mention. It is a very powerful method for dealing with measurable disturbances. Use of feedforward control requires, however, good models of process dynamics. It is difficult to tune feedforward control loops automatically on demand, since the operator often cannot manipulate the disturbance used for the feedforward control. To tune the feedforward controller it is therefore necessary to wait for an appropriate disturbance. Adaptation is therefore particularly useful for the feedforward controller.

52.4 Some Commercial Products

Commercial PID controllers with adaptive techniques have been available since the beginning of the eighties.

There is a distinction between *temperature controllers* and *process controllers*. Temperature controllers are primarily designed for temperature control, whereas process controllers are supposed to work for a wide range of control loops in the process industry such as flow, pressure, level, and pH control loops. Automatic tuning and adaptation are easier to implement in temperature controllers, since most temperature control loops have several common features. This is the main reason why automatic tuning has been introduced more rapidly in these controllers.

The process controllers can be separate hardware boxes for single loops, or distributed control systems (DCS) where many loops are handled by one system.

Since the processes that are controlled with process controllers may have large differences in their dynamics, tuning and adaptation becomes more difficult compared to the pure temperature control loops. In Table 52.5, a collection of process controllers is presented together with information about their adaptive techniques.

Automatic tuning is the most common adaptive technique in the industrial products. The usefulness of this technique is also obvious from Figure 52.6, where it is shown that the auto-tuning procedures are used not only to tune the controller, but also to obtain a comfortable operation of the other adaptive techniques. Most auto-tuning procedures are based on step response analysis.

Gain scheduling is often not available in the controllers. This is surprising, since gain scheduling is found to be more useful than continuous adaptation in most situations. Furthermore, the technique is much simpler to implement than automatic tuning or adaptation.

It is interesting to see that many industrial adaptive controllers are rule based instead of model based. The research on adaptive control at universities has been almost exclusively focused on model-based adaptive control.

One of the earliest rule-based adaptive controllers is the Foxboro EXACT. It was released in 1984. In this controller, the user specifies a maximum damping and a maximum overshoot. Whenever the control loop is subjected to a larger load disturbance or setpoint change, the response is investigated and the controller parameters are adjusted according to certain rules to meet the specifications.

Adaptive feedforward control is seldom provided in the industrial controllers. This is surprising, since adaptive feedforward control is known to be of great value. Furthermore, it is easier to develop robust adaptive feedforward control than adaptive feedback control.

Alfa Laval Automation's ECA400 and Fisher Controls DPR910 are identical. This controller has automatic tuning, gain scheduling, adaptive feedback, and adaptive feedforward. The automatic tuning procedure is based on relay feedback. The automatic tuning procedure is also used to build the gain schedule automatically, and to initiate the adaptive feedback and feedforward. In this way, there is no need for the user to supply any information about the process dynamics. All adaptive features can be used automatically.

References

[1] Åström, K.J. and Hägglund, T., *PID Control— Theory, Design and Tuning,* Instrument Society of America, Research Triangle Park, NC, 2nd ed., 1995.

[2] Åström, K.J., Hägglund, T., Hang, C.C., and Ho, W.K., Automatic tuning and adaptation for PID controllers — A survey, *Control Eng. Prac.,* 1(4), 699, 1993.

[3] Chien, K.L., Hrones, J.A., and Reswick, J.B., On the automatic control of generalized passive systems, *Trans. ASME,* 74, 175, 1952.

[4] Cohen, G.H. and Coon, G.A., Theoretical consideration of retarded control, *Trans. ASME,* 75, 827, 1953.

[5] Dahlin, E.B., Designing and tuning digital controllers, *Instr. Control Syst.,* 42, 77, 1968.

[6] Hägglund, T. and Åström, K.J, Industrial adaptive controllers based on frequency response techniques, *Automatica,* 27, 599, 1991.

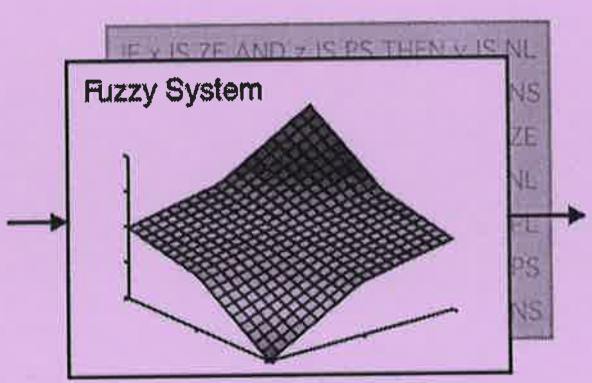
[7] Kaya, A. and Scheib, T.J., Tuning of PID controls of different structures, *Control Eng.,* July, 62, 1988.

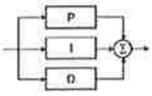
[8] Kraus, T.W. and Myron, T.J., Self-tuning PID controller uses pattern recognition approach, *Control Eng.,* June, 106, 1984.

[9] Morari, M. and Zafiriou, E., *Robust Process Control,* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

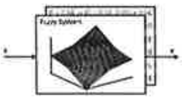
[10] Ziegler, J.G. and Nichols, N.B., Optimum settings for automatic controllers, *Trans. ASME,* 64, 759, 1942.

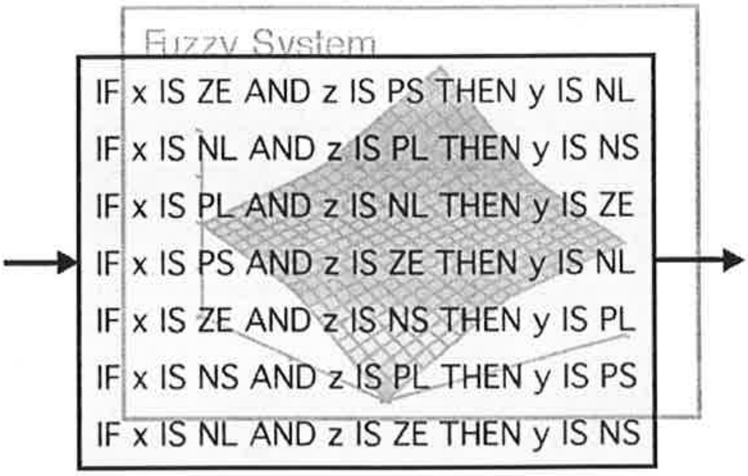
Fuzzy



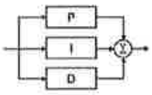


Views on Fuzzy Control

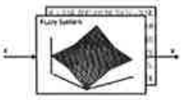


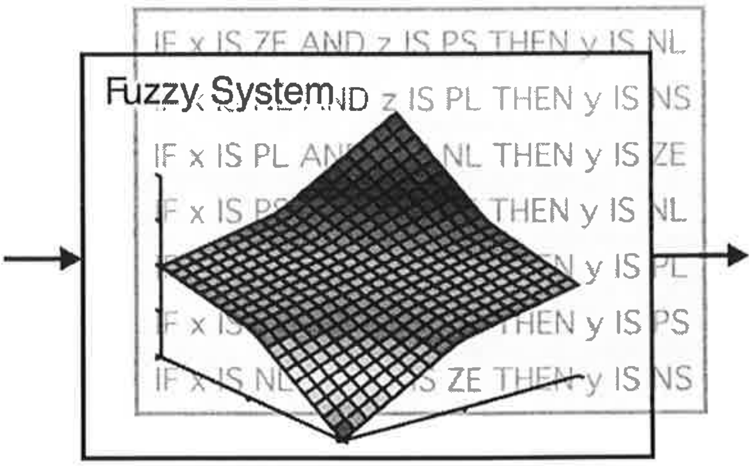


Rule-based Control



Views on Fuzzy Control





Non-Linear Control

Fuzzy Control

- Background & Motivation
- Fuzzy Sets & Fuzzy Logic
- Fuzzy Systems
- Fuzzy Control
  - Heuristic
  - Model-based
- Interpolation, Modeling, Function Approximation
- Summary

Background

The world is nonlinear.

Linear systems often a good local approximation (linearisation)

Linear control often sufficient, e.g. PID

Growing industrial demand for nonlinear control:

- increased functionality/complexity
- rapid production changes
- higher precision
- wider operating ranges

Reasons for Nonlinear Control

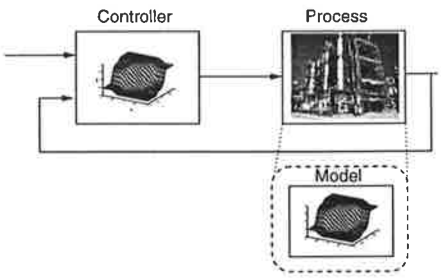
Nonlinear process:

- nonlinear dynamics
- linear dynamics + constraints on inputs, states, outputs

“Nonlinear" specifications:

- e.g. time-optimal control
- constraints
- small signal vs large signal behaviour

Nonlinearities in Control

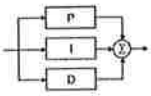


In the controller:

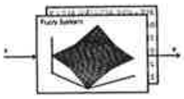
- feedback path
- feedforward path

In the process model:

- basis for model-based controller design
- as a part of a model-based controller



Nonlinear Mappings



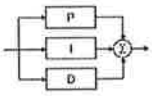
- Difficult to represent

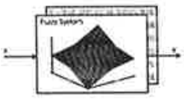
$$\mathbb{R}^n \to \mathbb{R}$$

  - Homogeneous discretization $N \Rightarrow N^n$ parameters.

Several approaches (parameterizations):

- Look-up Tables
- Splines
- Fuzzy Systems
- Neural Nets
- Wavelets
- Analytic functions
- Radial Basis Functions
- Logic & Selectors
- ...





Fuzzy Logic

Introduced by Lotfi Zadeh in early 1960s

Set theory/logic for non-probabilistic uncertainties

Developed and found applications in e.g., decision support, data classification, computer vision, etc

Major application – automatic control:

- Abe Mamdani, Queen Mary College, 1974
- steam engine and boiler

Motivation

	Originally		Currently
MAMDANI	Model operator's control actions	→	Represent process knowledge
	Obtain smooth interpolation	→	Define non–linear mappings

Cement Kilns

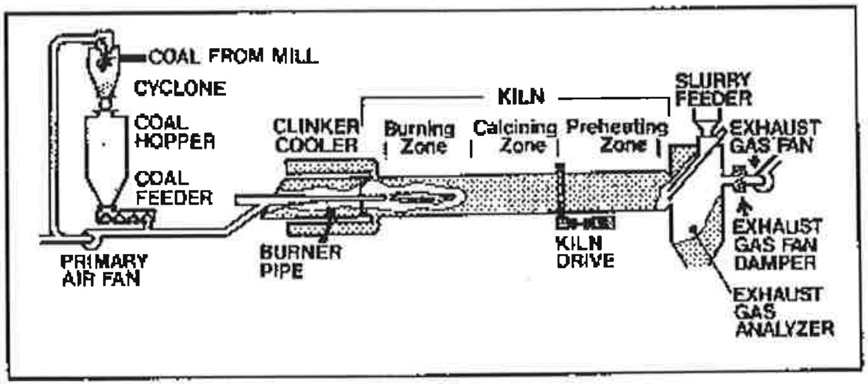
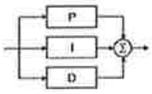
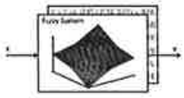


Fig. 17. The cement kiln process.

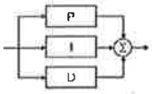
F.L Smidth (end 1970s), SIRA Linkman (now ABB)

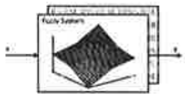




Case	Condition	Action to be taken	Reason
10	BZ OK	a. Increase I.D. fan speed	To raise back-end temperature and increase oxygen percentage for action 'b'
	OX low		
	BE low	b. Increase fuel rate	To maintain burning zone temperature
11	BZ OK	a. Decrease fuel rate slightly	To raise percentage of oxygen
	OX low		
	BE OK		
12	BZ OK	a. Reduce fuel rate	To increase percentage of oxygen for action 'b'
	OX low	b. Reduce I.D. fan speed	To lower back-end temperature and maintain burning zone temperature
	BE high		
13	BZ OK	a. Increase I.D. fan speed	To raise back-end temperature
	OX OK	b. Increase fuel rate	To maintain burning zone temperature
	BE low		
14	BZ OK	NONE. However, do not get overconfident,	
	OX OK	and keep all conditions under close observation.	
	BE OK		
15	BZ OK	When oxygen is in upper part of range	
	OX OK	a. Reduce I.D. fan speed	To reduce back-end temperature
	BE high	When oxygen is in lower part of range	
		b. Reduce fuel rate	To rise oxygen percentage for action 'c'
		c. Reduce I.D. fan speed	To lower back-end temperature and maintain burning zone temperature
16	BZ OK	a. Increase I.D. fan speed	To raise back-end temperature
	OX high	b. Increase fuel rate	To maintain burning zone temperature and reduce percentage of oxygen
	BE low		
17	BZ OK	a. Reduce I.D. fan speed slightly	To lower percentage of oxygen
	OX high		
	BE OK		

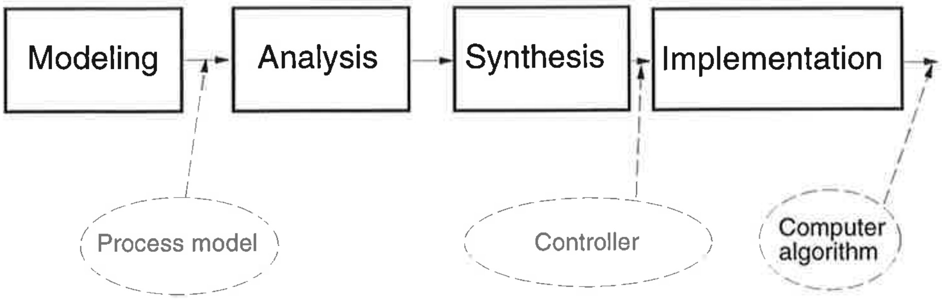
Fig. 1. Extract from textbook for cement kiln operators (ref. 1).





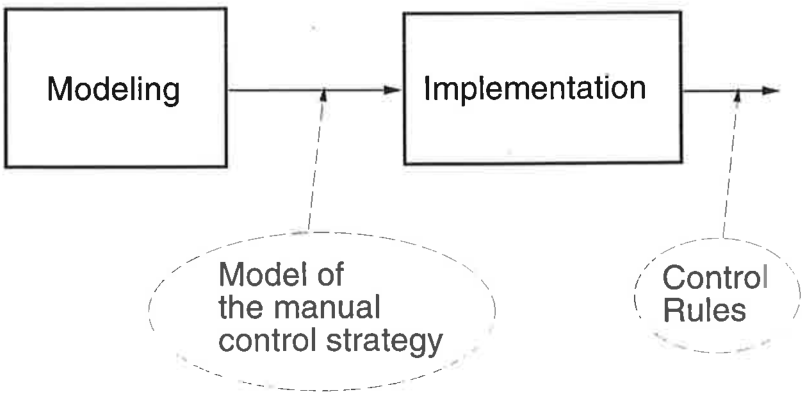
```
If torque is zero and
   free lime is low
then
   fuel rate change is medium-negative

If torque is zero and
   free lime is OK
then
   fuel rate change is zero

If torque is zero and
  free lime is high
then
   fuel rate change is medium-positive

If torque is negative and
   free lime is low
then
   fuel rate change is small-positive
```

Conventional Control



Also, Model-Based Fuzzy Control

Heuristic Fuzzy Control



Fuzzy Control

- Background & Motivation
- Fuzzy Sets & Fuzzy Logic
- Fuzzy Systems
- Fuzzy Control
  - Heuristic
  - Model-based
- Interpolation, Modeling, Function Approximation
- Summary

Ordinary ("crisp") sets

The set $A := \{x \mid x \in A\}$

Characteristic function

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & \text{otherwise} \end{cases}$$

Set operations

- intersection
- union
- complement
- subset

Fuzzy Sets

Membership function

$$\mu_A : x \rightarrow [0,1] \quad (1)$$

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (2)$$

$\mu_A(x)$ expresses to what degree "x is A"

Discrete or continuous universes

- **Discrete universes**

  Example: Expensive cars

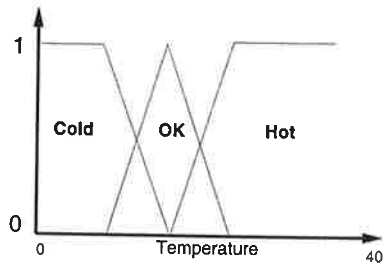
  $$U = \{\mathrm{BMW}, \mathrm{Ford}, \mathrm{Lada}\}$$

  Fuzzy set of expensive cars:

  $$\{(\mathrm{BMW}, 1), (\mathrm{Ford}, 0.6), (\mathrm{Lada}, 0)\}$$

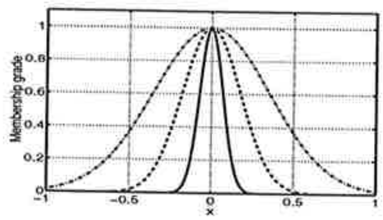
Discrete or continuous universes

- **Continuous universes**

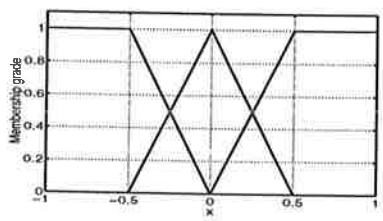


Typical Membership Functions

Gaussian:

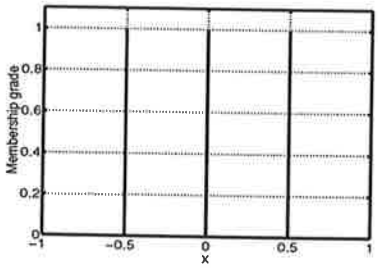


Triangular/Trapezoidal:



Typical Membership Functions

Singleton:



Set operations

Intersection – AND: $A \cap B$

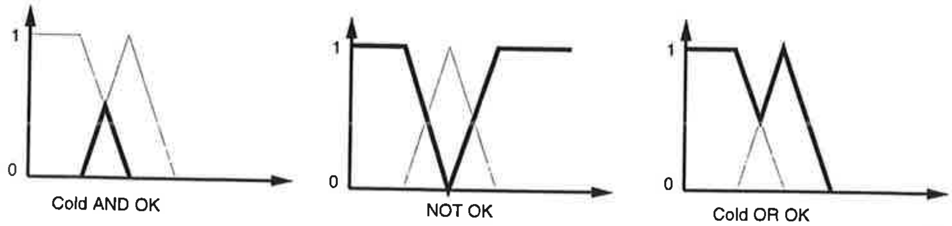
$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

Union – OR: $A \cup B$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

Complement – NOT: A'

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

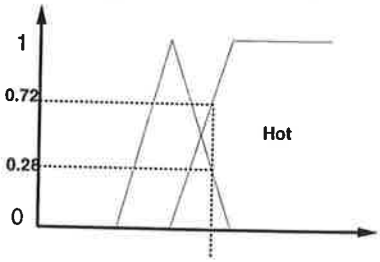


Fuzzy Logic

Generalization of ordinary boolean logic.

Propositions have truth values between 0 and 1.

Temperature is Hot



AND, OR and NOT connects simple propositions into compound propositions

x_1 is $A_1^{(i)}$ AND x_2 is $A_2^{(i)}$ OR...

Fuzzy Relations

R = "x is considerably larger than y"

x \ y	1	3	7	11
2	0.4	0	0	0
5	0.6	0.5	0	0
9	1	0.8	0.4	0

$\mu_R(x, y)$

Fuzzy Inference

`If <fuzzy proposition> Then <fuzzy proposition>`

Generalized Modus Ponens:

Modus Ponens:

A
A –> B

B

Generalized Modus Ponens:

x is A1
if x is A2 then u is B1

---

u is B2

Example:

tomato is very red
if tomato is red then tomato is ripe

---

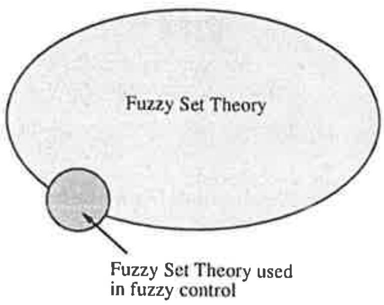
tomato is very ripe

Compositional rule of inference

Fuzzy Logic

A very large body of theory has been developed

Very little of this is needed to understand/use fuzzy logic for control



Fuzzy Control

- Background & Motivation
- Fuzzy Sets & Fuzzy Logic
- Fuzzy Systems
- Fuzzy Control
  - Heuristic
  - Model-based
- Interpolation, Modeling, Function Approximation
- Summary

Connecting to Physical Systems

Problem: Inputs and outputs have numeric values.

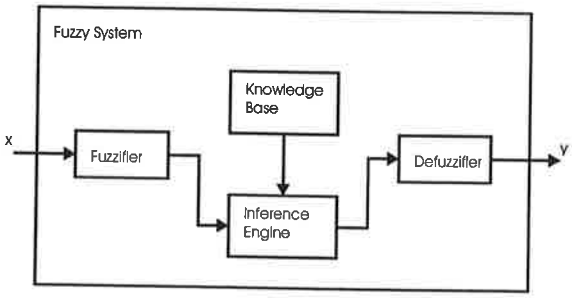
Solution: Add interfaces.

- Fuzzifier : Numbers $\rightarrow$ fuzzy sets.
- Defuzzifier: Fuzzy sets $\rightarrow$ numbers.

Architecture

Fuzzy System:

- Knowledge Base
- Logic and Inference
- Interfaces



Fuzzification

Fuzzifier : Number x' $\rightarrow$ fuzzy set A'.

– Common choice: Singleton fuzzifier

$$\mu_{A'}(x) = \begin{cases} 1, & x = x' \\ 0, & \text{otherwise} \end{cases}$$

Inference System

Two main types used in fuzzy control:

1. Mamdani Inference Systems
2. Takagi-Sugeno Inference Systems

Mamdani Systems

Rules:

$$\text{IF } x \text{ IS } A \text{ THEN } u \text{ IS } B$$

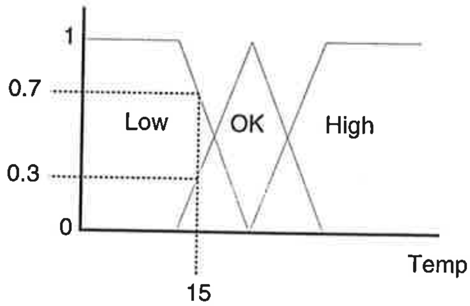
A and B fuzzy sets

Calculations:

1. Input fuzzy set evaluation
2. Calculation of degree-of-fulfillment of each rule
3. Calculation of the fuzzy output of each rule
4. Aggregation of the fuzzy outputs
5. Defuzzification

Mamdani Systems

1. Input fuzzy set evaluation



Temperature is Low to degree 0.7

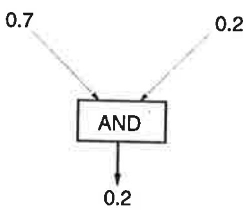
Temperature is OK to degree 0.3

Temperature is High to degree 0

Mamdani Systems

2. Calculation of degree-of-fulfillment of each rule

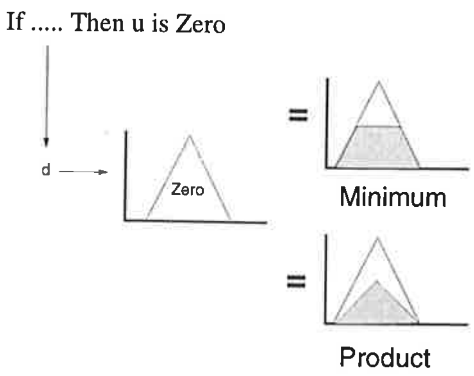
If Temp is Low AND Pressure is OK Then ...



AND	minimum	product	T–norm
OR	maximum	bounded sum	T–conorm
NOT	1 – a		

Mamdani Systems

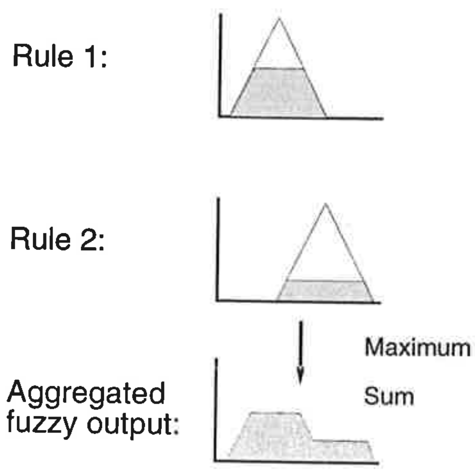
3. Calculation of the fuzzy output of each rule



35

Mamdani Systems

4. Aggregation of the fuzzy outputs



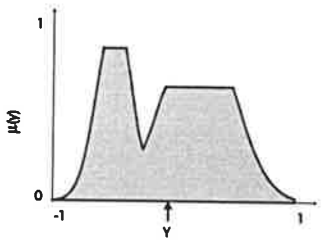
36

Mamdani Systems

5. Defuzzification

Defuzzifier: Fuzzy set B' → number y'.

- Common choice: Center of Gravity

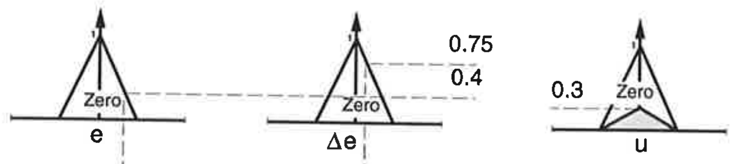


$$y = \frac{\int_Y w \cdot \mu_U(w)\,\mathrm{dw}}{\int_Y \mu_U(w)\,\mathrm{dw}}$$

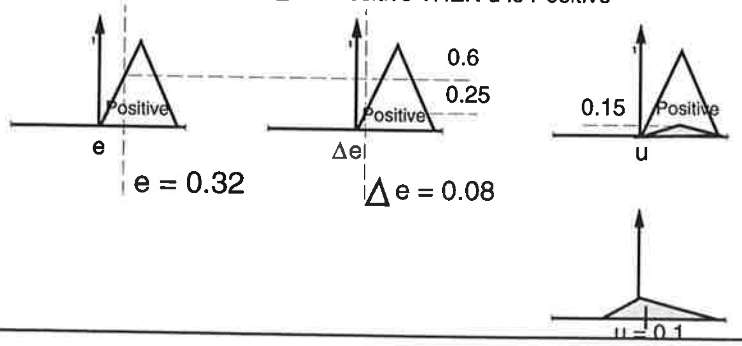
- Other choices: Center of Area, Mean of Maximum, ...

Mamdani Systems: Summary

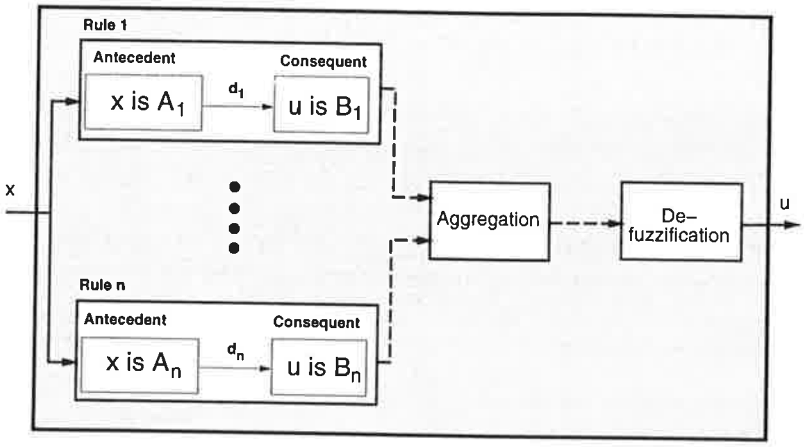
Rule 1: IF e is Zero and Δe is Zero THEN u is Zero



Rule 2: IF e is Positive and Δe is Positive THEN u is Positive



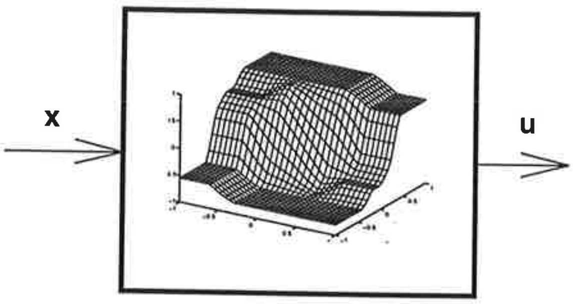
Internal View



Crisp value

Fuzzy value

External view



Sugeno Systems

Rules:

$$\text{IF } x \text{ IS } A \text{ THEN } u = f(x)$$

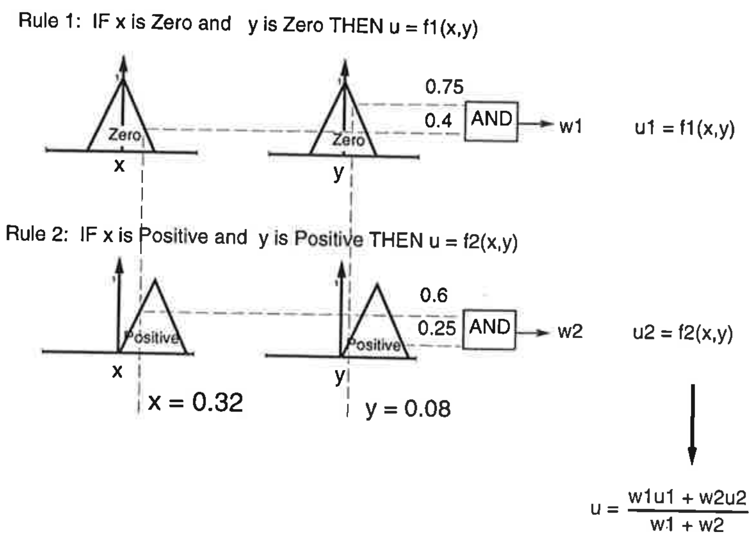
A fuzzy set, $f(x)$ function, often linear, i.e.,

$$f(x) = l_0 + l_1 x_1 + \cdots + l_n x_n$$

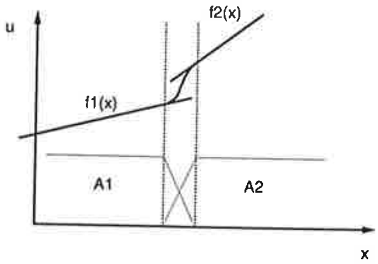
Calculations:

1. Input fuzzy set evaluation
2. Calculation of degree-of-fulfillment (weight) of each rule
3. Calculation of the output of each rule
4. Calculation of the output by weighted average

Sugeno Systems



Sugeno Systems



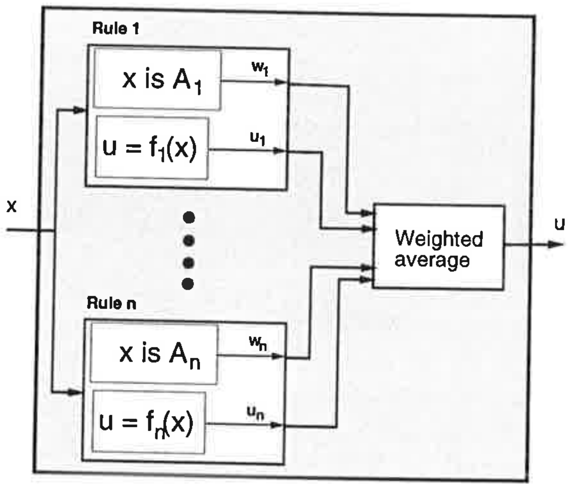
If x is A1 then u=f1(x)

If x is A2 then u=f2(x)

Compare:

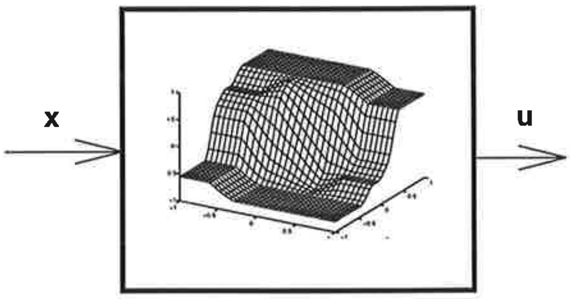
- Gain scheduling
- Heterogeneous controllers

Internal View



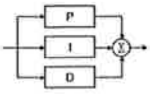
Crisp value

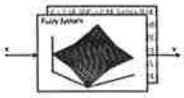
External view



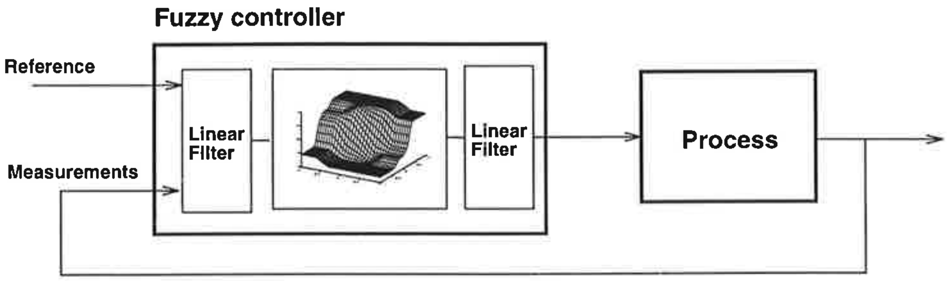
Fuzzy Control

- Background & Motivation
- Fuzzy Sets & Fuzzy Logic
- Fuzzy Systems
- Fuzzy Control
  - Heuristic
  - Model-based
- Interpolation, Modeling, Function Approximation
- Summary





Fuzzy Controller

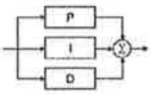


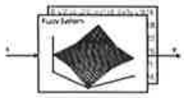
Inputs:

- Measured variables, reference variables, controller errors
- Filtered signals (low-pass, derivatives, lagged signals, ...)

Outputs:

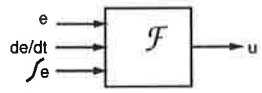
- Controller output
- Controller output increment



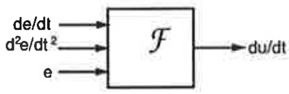


Controller Structures

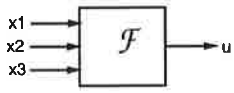
PID – position form



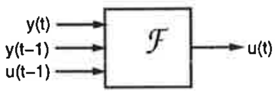
PID – velocity form



State feedback



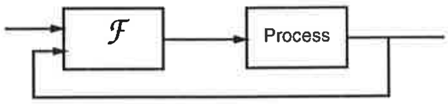
Polynomial controllers



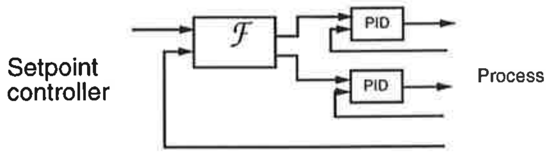
•••

Application Types

Direct feedback control schemes

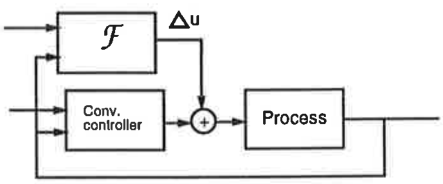


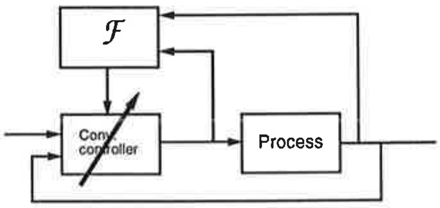
Supervisory control schemes



Application Types

Hybrid Control Schemes





Heuristic Fuzzy Control

Heuristic process & control knowledge

Often based on a PI, PD, PID scheme augmented with:

- external input signals (MISO)
- nonlinearities, e.g., deadbands, $|e|e$
- gainscheduling
- logic and constraint handling

Fuzzy PD-Controller

e \ $\triangle$e	NL	NS	Z	PS	PL
NL	PL	PL	PM	PS	Z
NS	PL	PM	PS	Z	NS
Z	PM	PS	Z	NS	NM
PS	PS	Z	NS	NM	NL
PL	Z	NS	NM	NL	NL

Certain choices of parameters give:

- exactly linear mappings
- piecewise linear mappings

Useful for:

- initial controller design
- linear behaviour in specific regions
- avoid large local gain variations
- simplify analysis

Model-Based Fuzzy Control

Fuzzy controller based on an explicit process model

The model often expressed as a fuzzy system, e.g.,

- fuzzy NARX-model – Mamdani form

IF $y(t-1)$ IS NL AND $u(t)$ IS PS THEN $y(t)$ IS NS.

- fuzzy NARX-model – Sugeno form

IF $y(t-1)$ IS NL AND $u(t)$ IS PS THEN
$$y(t) = a_1 y(t-1) + b_1 u(t).$$

- fuzzy state-space model

IF x_1 is F_1 AND $\cdots$ AND x_n is F_n AND u is E THEN
$$\dot{x} = Ax + Bu$$

Model-Based Design Methods

Fuzzy versions exists of most conventional linear/nonlinear design methods, e.g.,

- inverse control
- internal model control (IMC)
- model-predictive control (MPC)
- feedback linearisation
- sliding mode control
- ...

Outside the scope of this course

Quadratic Lyapunov Functions

Sugeno linear state-space model:

$$R_l: \text{ IF } x_1 \text{ IS } F_{l,1} \text{ AND } \ldots \text{ AND } x_n \text{ IS } F_{l,n}$$
$$\text{THEN } \dot{x} = A_l x,$$

Powerful methods for analysis and synthesis:

Stability analysis:

If there exists a matrix $P = P' > 0$ such that

$$A_l' P + P A_l < 0, \qquad l = 1, \ldots, L \tag{3}$$

then every trajectory $x(t)$ tends to zero exponentially.

$V(x) = x'Px$ is a Lyapunov function for the system.

Quadratic Lyapunov Functions

Synthesis of stabilizing state feedback control:

If one can find a common Lyapunov matrix P it is also possible to design stabilizing state feedback controllers:

- requirements on exponential decay rate
- constraints on input signals
- fuzzy observer design
- ...

Convex optimization in terms of Linear Matrix Inequalities (LMIs):

- efficient
- software available (LMI toolbox)

Piecewise Quadratic Lyapunov

Finding a common Lyapunov matrix can often be difficult or impossible

Conservative methods

Recent work in Lund:

- piecewise quadratic Lyapunov functions
- analysis and synthesis
- applies to fuzzy control
- formulated as LMIs
- Mikael Johansson & Anders Rantzer

Tuning Parameters

A lot of parameters:

- Number of fuzzy sets
- Shape of fuzzy sets
- Overlap of fuzzy sets
- Inference and defuzzification methods
- Normalization and Denormalization gains

Many of these only have minor impact on the nonlinearity generated.

Application survey

D.E. Thomas and B. Armstrong-Hélouvry: Fuzzy Logic Control – A Taxonomy of Demonstrated Benefits, Proc. of the IEEE, Vol 83, No.3 March 1995

Demonstrated benefits:

1. Capture operator knowledge
2. Exception handling – the possibility to express dynamic mode-dependent control policies
3. Fast set-point change responses through non-linear, "bang-bang" control.
4. Local Adaptation – adjust (or adapt) control actions in local regions of the input space.
5. Smooth interpolation of different control policies

Commercial Tools

Graphical development environments:

- user-friendly interfaces
- C code generation
- code for special processors
- simulation possibilities

Fuzzy VLSI chips

- digital or analog techniques
- high speed

User Interface Issues

Development environment:

- fuzzy sets: graphically, math. functions, point pairs, ...
- rules: text, tables, graphical blocks

Run-Time interface:

- rule degree of fulfillment
- output fuzzy set
- phase plane plots

Application Types

- direct control, "small" systems
- supervisory set-point control, "large" systems, quality control
- hybrid applications, fuzzy gain scheduling,

Application Examples

- automatic train brake systems
- elevator control
- automobile transmission and engine control
- heat exchangers
- cement kilns
- water purification
- power systems
- washing machines
- vacuum cleaners
- air conditioners
- CAMcorders

Fuzzy Control

- Background & Motivation
- Fuzzy Sets & Fuzzy Logic
- Fuzzy Systems
- Fuzzy Control
  - Heuristic
  - Model-based
- Interpolation, Modeling, Function Approximation
- Summary

A Fuzzy PD Rule Base

Rules:

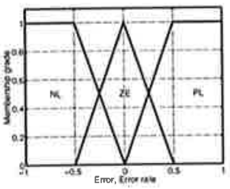
IF e is PL AND $\dot{e}$ is NL THEN u is NL

$\vdots$

Illustrated in a rule table:

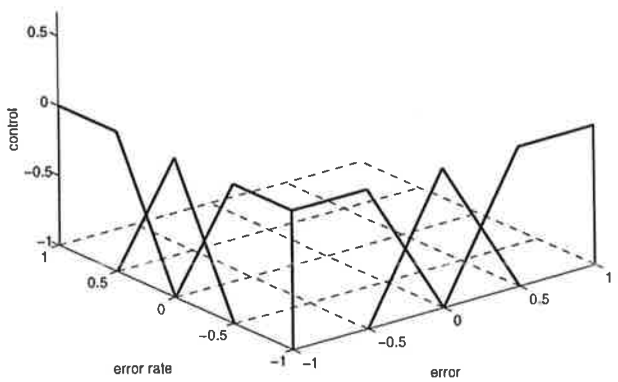
		e		
		NL	ZE	PL
	PL	ZE	PL	PL
$\dot{e}$	ZE	NL	ZE	PL
	NL	NL	NL	ZE

Typical membership functions:



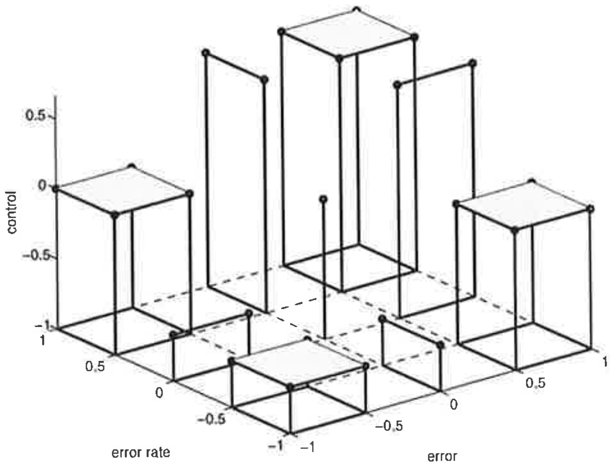
A Table Look-up Analogy

- Rule premises partition controller state space into a set of intervals



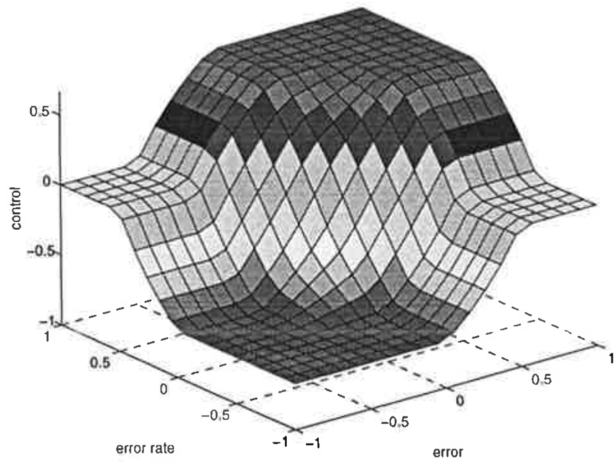
A Table Look-up Analogy

- Rule consequents specify nonlinearity at interval endpoints



A Table Look-up Analogy

- Inference process performs interpolation (also influenced by fuzzifier/defuzzifier)

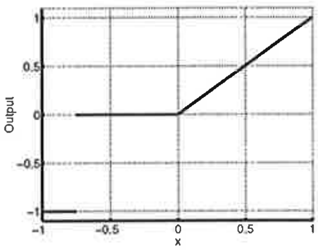


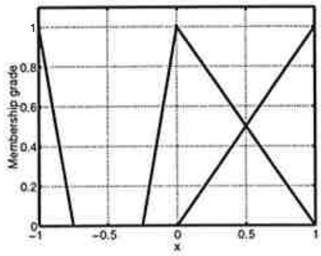
Why Overlapping Fuzzy Sets ?

Insight:

- Several active rules: interpolation.
- One valid rule: constant output – Mamdani, linear output – Sugeno .
- No valid rule : zero output.

Example:





Fuzzy Systems and Nonlinear Maps

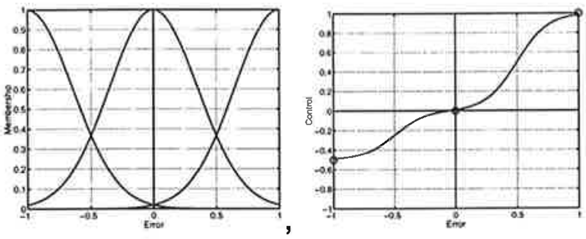
- Two representations
  - Nonlinear Function
  - Fuzzy system
- Closed forms sometimes possible

$$y = \sum_{i=1}^{M} \overbrace{g_i(x)}^{\text{IF-part}} \underbrace{w_i}_{\text{THEN-part}}$$

- One-to-One

Fuzzy System Nonlinearities I

Gaussian Membership Functions:



Formula

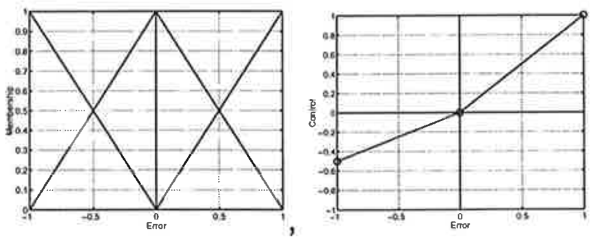
$$\hat{f}(x) = \sum_{i=1}^{M} \frac{\mu_i(x;\theta)}{\sum_{i=1}^{M} \mu_i(x;\theta)} w_i = \sum_{i=1}^{M} g_i(x) w_i$$

Remarks:

- "Normalized Radial Basis Functions."
- Global formula.

Fuzzy System Nonlinearities II

Triangular Membership Functions:



Formula:

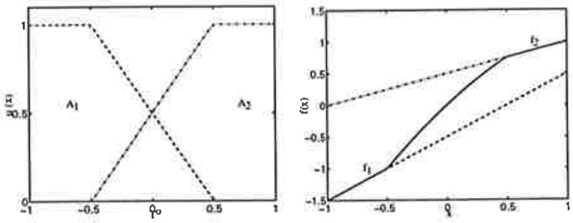
$$\hat{f}(x) = \sum_{i=1}^{M} \mu_i(x) w_i = \sum_{i=1}^{M} g_i(x) w_i$$

Remarks:

- "Linear B-Splines"
- Piecewise multilinear, can be made exactly linear

Fuzzy System Nonlinearities III

Sugeno-Type Models, Linear Consequents:



Formula:

$$\hat{f}(x) = \sum_{i=1}^{M} \frac{\mu_i(x;\theta)}{\sum_{i=1}^{M} \mu_i(x;\theta)} \left(L^{(i)}\right)^T x = \sum_{i=1}^{M} g_i(x) \left(L^{(i)}\right)^T x$$

Remarks:

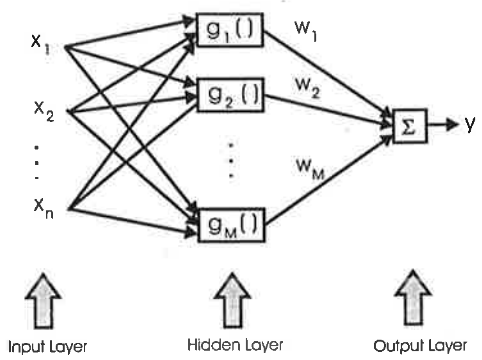
- Gain-scheduling: $\hat{f}(x) = L^T(x)x$
- Can be made exactly linear

Relation to Neural Nets

Evaluation of a fuzzy system mapping

$$f(x) = \sum_{i=1}^{M} g_i(x) w_i$$

can be illustrated as a "feedforward" net



Basis for "neuro-fuzzy" systems.

Universal Approximators

Fuzzy systems are universal approximators

Let

$$f : U \subseteq \mathbf{R}^n \rightarrow \mathbf{R}$$

be a continuous function defined on a compact set U. Then, for each $\varepsilon > 0$ there is a fuzzy system $\hat{f}_\varepsilon(x)$ such that

$$\sup_{x \in U} |f(x) - \hat{f}_\varepsilon(x)| \leq \varepsilon$$

Valid for Mamdani and Sugeno fuzzy systems.

Fuzzy Modeling & Identification

Modeling – A rough outline

- Determine relevant process variables
- Formulate heuristic knowledge as rules
- Transform rules into the equivalent nonlinear formula
- Adjust parameters to fit data
- Transform back to rules

Parameter Identification I

Fit fuzzy model to N measurements (x_k, y_k).

Fix $g_i(x; \theta)$, adjust w_i ($w_i \leftrightarrow$ consequents).

Writing

$$\hat{f}(x) = \sum_{i=1}^{M} g_i(x; \theta) w_i = \phi^T(x) w$$

we have

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \\ \phi^T(x_N) \end{bmatrix} w = \Phi w$$

Optimal parameters in LS sense:

$$w^* = \Phi^+ Y$$

Parameter Identification II

Adjusting Premises – Nonlinear Optimization

Several “off-the-shelf” methods:

- Gauss-Newton
- Levenberg-Marquardt
- Conjugate Gradient

Clustering based methods popular in fuzzy systems

“Iterative Hill Climbing” – Local Minima ?

The backpropagation method famous in the neural network community is simply a version of gradient based optimization applying the chain rule

Can also be applied to fuzzy systems

Neuro-Fuzzy Systems – Again

What's new with Fuzzy Systems ?

- Neural Networks:
  - Black Box Models
- Fuzzy Systems:
  - Rules $\Rightarrow$ "Grey-Box"
  - Initial Parameters & Learning.

Fuzzy Control

- Background & Motivation
- Fuzzy Sets & Fuzzy Logic
- Fuzzy Systems
- Fuzzy Control
  - Heuristic
  - Model-based
- Interpolation, Modeling, Function Approximation
- Summary

Summary

Why Fuzzy Control?

The populistic argument:

- for control of processes that are difficult to model and control with conventional control techniques

The control engineers argument:

- as a user-friendly way of designing non-linear low-order controllers.

Advantages

- User-friendly way to design low-order nonlinear controllers
- Allows explicit representation of process control knowledge
- Identification and adaptation possible
- Nicely packaged technique
- Intuitive, perhaps specially for those with limited knowledge of classical control

Disadvantages

- Limited analysis and synthesis methods applicable
  - not surprising – nonlinear control theory
- Validation through simulation
  - requires process model!
- Computationally intensive
  - unless correct parameter choices are made
- Many poor papers
  - irreproducible results
  - exaggerated claims
  - unfair comparisons
  - beginning to improve

What is best?

The question *Is fuzzy control better than X control?* is totally irrelevant if one only considers the control loop performance

From the process' point of view it is the nonlinearity that matters and not how it is parameterized.

How should then different nonlinear methods be compared?

Comparative Issues

Function approximation properties:

- which classes of functions can the method approximate

Approximation efficiency:

- how many adjustable parameters are needed in the approximation
- "bias-variance tradeoff"

Degree of locality:

- are local adjustments possible?

Comparative Issues

Support for estimation:

- does the parameterization allow the nonlinearity to be generated from input/output data

Computational efficiency:

- during estimation and during evaluation

Theoretical groundedness:

- the amount of theoretical results that are available for the parameterization

Transparency:

- how readable is it and how easy it is to express prior knowledge

Comparative Issues

Availability of computer tools:

- Matlab, CAD environments that generate PLC/C code

Subjective issues:

- how comfortable the designer/operator is with the formalism
- the level of training needed to be able to use/understand the method

Lund Activities

Heterogeneous control:

- K J Åström & B. Kuipers
- Sugeno fuzzy controller + qualitative simulation

Fuzzy anti-reset windup for PID and heater control:

- Anders Hansson together with Landis & Gyr

FALCON:

- Fuzzy ALgorithms for CONtrol
- ESPRIT III Working group project 93–95
- 11 European groups

Lund Activities

"Design och inställning av fuzzy-regulatorer baserat på olinjär reglerteknik":

- ITM (Institute of Applied Mathematics), Volvo, ABB
- 93–96
- Mikael Johansson
- car climate control, electric arc furnace control
- theory for piecewise linear systems

FAMIMO:

- Fuzzy Algorithms for Multi-Input Multi-Output Control
- Esprit IV Long Term Project, 97-99
- control of direct injection car engine
- four universities + Siemens Automotive

Learn More

Survey report:

- A Survey on Fuzzy Control, K-E Årzén, M. Johansson, R. Babuska

Books:

- Fuzzy Logic:
  - a large number of books available of varying quality
- Fuzzy Control:
  - D. Driankov, H. Hellendoorn, M. Reinfrank: An Introduction to Fuzzy Control, Springer Verlag, 1993
  - Li-Xin Wang: Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Prentice Hall, 1994
  - M. Brown, C. Harris: Neuro-Fuzzy Adaptive Modeling Control, Prentice Hall, 1995

Learn More

Articles:

- Proceedings of the IEEE, March 1995, Special Issue
  - J-S. R. Jang, C-T. Sun: Neuro-Fuzzy Modeling and Control
  - J. Mendel: Fuzzy Logic Systems for Engineering: A tutorial
- C.C. Lee: Fuzzy Logic in Control Systems: Fuzzy Logic Controller Part I and II, IEEE Transactions on Systems, Man & Cybernetics, Vol 20, No. 2, 1990
- A. Juditsky et al: Nonlinear black-box modeling in system identification: Mathematical foundations. Automatica, 31, 1995
- Sjöberg et al: Nonlinear black-box modeling in system identification: A unified overview, Automatica, 31, 1995
- R. Zbikowski et al: A review of advances in neural adaptive control systems. Technical report, Daimler-Benz AG and University of Glasgow, 1994

Learn More

Journals:

- Fuzzy Sets and Systems
- IEEE Transactions on Fuzzy Systems
- IEEE Transactions on Systems, Man & Cybernetics

Conferences

- FUZZ-IEEE
- IFSA World Congress
- (EUFIT)
- + several smaller workshops