Realisering och inställning av PID-regulatorer

Författare:
Jonas Wahlfrid, jonas_wahlfrid@hotmail.com
Student på programmet Programvaruteknik, Malmö Högskola Teknik och Samhälle
2007-02-19

Handledare:
Ola Dahl
Doktor i reglerteknik, Malmö Högskola Teknik och Samhälle

Abstract

This 5 point report discusses realisation and tuning of software based PID-controllers. The intended reader is an engineer that is interested in how a PID-controller is implemented and tuned with the AMIGO (Approximate M-constrained integral gain optimization) method. All mathematical reasoning is explained in detail step by step. The main purpose with this report is to gather knowledge from the reference sources, and clearly present this knowledge, so that it is accessible for engineers that work with PID-controllers.

- How does a PID-controller function?
- How can a PID-controller be implemented?
- How is a PID-controller tuned?

Sammanfattning
Denna 5 poängs rapport behandlar realisering och inställning av mjukvarubaserade PID-regulatorer. Den tänkte läsaren är en ingenjör som är intresserad av hur en PID-regulator implementeras och trimmas med AMIGO (Approximate M-constrained integral gain optimization) metoden. Alla matematiska resonemang förklaras stegvis och utförligt. Huvudsyftet med rapporten är att inhämta kunskap från referenserna, och formulera denna kunskap så att den blir lättillgänglig för ingenjörer som praktiskt arbetar med PID-regulatorer.
Målet med denna rapport är att svara på frågorna:

- Hur fungerar en PID-regulator?
- Hur kan en PID-regulator implementeras?
- Hur trimmas en PID-regulator med AMIGO-metoden?

Innehållsförteckning

1 Introduktion till reglerteknik

Enligt [1] är reglerteknik ”läran om hur man använder on-line mätningar för automatiska korrektioner i processen”. Processens mätvärde uppdateras hela tiden dvs. on-line. Regulatorn gör hela tiden automatiska korrektioner av processen för att börvärdet ska hållas.

Ett exempel på reglering är en farthållare till en elbil. Hastigheten mäts med en givare som känner hjulets hastighet, detta kallas återkoppling. Bilens verkliga hastighet jämförs med den önskade hastigheten och bilens dator beräknar en signal som ökar eller minskar gaspådraget. Varför måste hastigheten mätas, räcker det inte men att ställa gaspådraget i ett visst läge som motsvarar en viss hastighet? Nej det räcker inte eftersom det finns flera störningar som stör processen att omvandla kemisk energi i batteriet till rörelseenergi. Exempel på störningar är ändrad batterispänning på grund av laddningsnivån eller ålder, nedförsbacke eller uppförsbacke och förändrad däckfriktion och lagerfriktion. Eftersom nivån på dessa störningar varierar går det inte att förvänta sig att ett visst läge på gasreglaget ska motsvara en bestämd hastighet. För att kompensera för störningarna återkopplas bilens hastighet genom on-line mätning, och regulatorn korrigerar gaspådragets läge för att börvärdet ska följas.

Enligt [1] är idén om att tillverka maskiner som automatiskt varierar en variabel för att bibehålla nivån på en annan variabel gammal. Flera anordningar från antiken (700 f.Kr.-400 e.Kr.) användes för att reglera vattenflöde med hjälp av en flottör, hävarm och en ventil. Ett exempel på en mekanisk proportionell regulator är James Watts centrifugalregulator för ångmaskiner från slutet av 1700-talet. Denna regulator reglerar ångflödet genom att två kulor som drivs av ångmaskinens huvudaxel snurrar runt. När varvtalet ökar lyfts kulorna av centrifugalkraften, vilket medför att en ventil stryper ångflödet varvid varvtalet minskar.

2 On-off regulatorn

För enkla regleringar räcker det ibland med en regulator som bara har två lägen, till eller ifrån [1]. Det vanligaste exemplet på en on-off regulator är termostaten till ett elektrisktvärmeelement. En termostat är vanligen en mekanisk brytare som växlar läge med temperaturen. On-off reglering är inte någon noggrann reglering eftersom styrsignalen antar maxvärdet eller minvärdet. Det är t.ex. inte acceptabelt att realisera en farthållare med hjälp av en on-off regulator. Visserligen skulle kanske medelhastigheten hållas runt t.ex. börvärdet 50km/h, men bilens momentana hastighet skulle pendla mellan t.ex. 0 och 200km/h, vilket inte är acceptabelt. Styrsignalen u antar det maximala värdet om reglerfelet $e > 0$ och det minimala värdet om $e < 0$. Reglerfelet e = börvärdet y_{sp} – ärvärdet y. Vanligen används hysteres eller en dödzon för att inte frekvensen på tillslag/frånslag ska bli för stor [5, sid. 4].

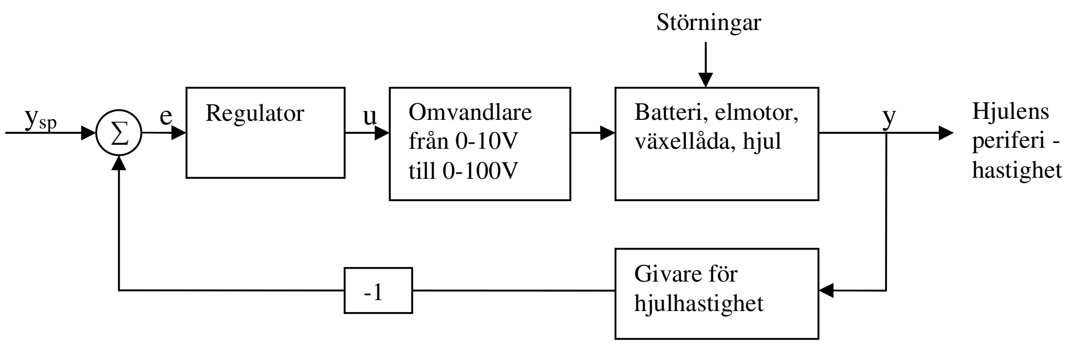
3 P, PI och PID-regulatorn

En mera noggrann regulator än On-off regulator är PID-regulatorn eftersom styrsignalen kan anta kontinuerliga värden [1]. PID är en förkortning av proportionell, integrerande och deriverande verkan. Vanligen kombineras dessa olika delar till regulatorer av typen PI eller PID [1]. En regulators uppgift är att hålla ärvärdet nära börvärdet genom att ändra styrsignalen. I farthållareexemplet i figur 3.1 är ärvärdet bilens hastighet i km/h, börvärdet är den önskade hastigheten i km/h och styrsignalen varierar mellan 0 Volt och 10 Volt. Styrsignalen motsvarar 0-100 Volt till likströmsmotorn som driver elbilen. Reglerfelet eller regleravvikelsen är skillnaden mellan börvärdet och ärvärdet. I tabell 3.1 listas beteckningar på variabler enligt [5].

Variabel	Beskrivning
y	ärvärde, mätvärde, processvärde
u	styrsignal
y_{sp}	börvärde, sp är en förkortning för engelskans ”set point”
e	reglerfel, regleravvikelse $e = y_{sp} - y$

Tabell 3.1 Variabeltabell [1].

För att visa en abstraktion av en reglerkrets används ett blockschema. Ett exempel på farthållarens blockschema visas i figur 3.1.



Figur 3.1 Blockschema som visar farthållaren schematiskt, princip från [3, sid. 10][5, sid. 3].

3.1 P-regulatorns funktion

Anta att vi försöker använda en P-regulator (proportionell regulator) för att bygga en farthållare. Följande ekvation visar ett exempel på beroendet mellan spänningen från regulatorn och uppmätt hastighet y. Styrsignalen kan inte anta värden högre än 10 Volt och inte lägre än 0 Volt.

$u = 0{,}1(90 - y) + 9$ V

90km/h är den önskade hastigheten och 9 V är den spänning som krävs för att hålla 90km/h på en plan vägsträcka och normalt luftmotstånd. Faktorn 0,1 kallas förstärkning och anger hur kraftigt regulatorn ska agera på en avvikelse. Om farten minskar till 85km/h på grund av t.ex. en uppförsbacke kommer regulatorn att öka spänningen till

$0{,}1(90 - 85) + 9 = 9{,}5$ V

Formeln för P-regulatorn är [1]
$u = K(y_{sp}-y) + u_b$

Eftersom e = y_{sp} - y kan formeln skrivas om till [1]
$u = K\,e + u_b$

Där K är regulatorförstärkningen och u_b en offset för att styrsignalen ska hålla en viss nivå då reglerfelet är noll. Förstärkningen K anger hur mycket styrsignalen ska reagera på en ändring av reglerfelet. Om en process har låg förstärkning väljs ett högre värde på K än om processen har en hög förstärkning, då ett lågt värde på K väljs. K väljes alltså omvänt proportionellt mot processens förstärkning. Om processförstärkningen är negativ vilket den är vid t.ex. kylning anges ett negativt värde på K. K sätts alltså till ett positivt värde om mätvärdet går åt samma håll som styrsignalen, annars negativt.

Dessvärre fungerar inte P-regulatorn särskilt bra eftersom den ger ett kvarstående reglerfel såvida inte offseten ändras vid varje börvärdesändring. Anta att vi ändrar börvärdet till 50km/h utan att ändra offset
$0{,}1(50-85)+9=5{,}5\ \text{V}$

Styrsignalen minskar till 5,5 V vid ett reglerfel på –35km/h. Men vid ett reglerfel på –15km/h ökar styrsignalen/spänningen i förhållande till ett reglerfel på –35km/h, enligt
$0{,}1(50-65)+9=7{,}5\ \text{V}$

Efterhand som reglerfelet minskar ökar styrsignalen och därmed hastigheten. Vid reglerfelet 0 fås en styrsignal/spänning av 9 V vilket motsvarar ca 90km/h vilket är långt ifrån börvärdet på 50km/h. Detta innebär att det blir omöjligt att komma fram till börvärdet.

I praktiken kommer ärvärdet att stabilisera sig på t.ex. 70km/h dvs. ett reglerfel på 50-70=-20km/h, eftersom kraften som driver bilen framåt kommer i jämvikt med luftmotståndet. Det är alltså omöjligt att få reglerfelet att minska till 0. Om förstärkningen ökas mycket går det att komma nära noll i reglerfel, men hög förstärkning kommer att leda till instabilitet i form av självsvängning. Även om börvärdet och offseten alltid är den samma fås en regleravvikelse om en störning i form av t.ex. en uppförsbacke uppstår.

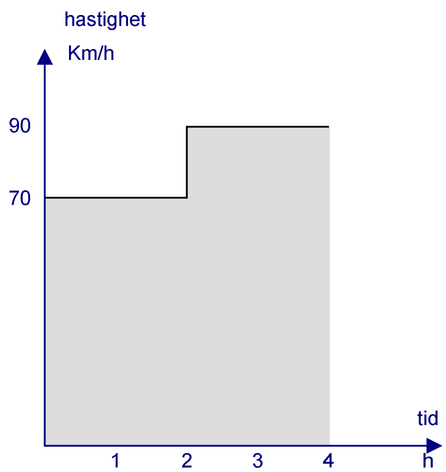
P-regulatorn klarar inte att hålla börvärdet (börvärdesföljning) eller störningsundertryckning utan bestående reglerfel, såvida inte värdet på offseten ändras efter omständigheterna [1]. Att ändra offset efter börvärde och omständigheter manuellt är inte praktiskt möjligt, vilket gör P-regulatorn nästan oanvändbar [1]. Ett exempel på en mekanisk P-regulator är termostatventilen på ett vattenelement i en byggnad.

3.2 PI-regulatorns funktion

PI-regulatorn har en proportionell och en integrerande del. Integraldelen ersätter offsetkonstanten i P-regulatorn så att justeringen går automatiskt. Integraldelen är proportionell mot integralen av reglerfelet [1].

3.2.1 Vad menas med en integral?

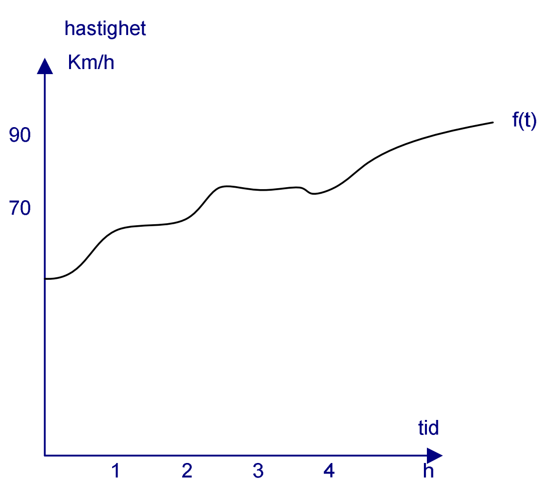
Figur 3.2 visar hur en bils hastighet ändras med tiden.



Figur 3.2 En bils hastighet under fyra timmar.

Hur långt färdas bilen i figur 3.2 under de två första timmarna?
Sträckan = Farten * Tiden
Sträckan = 70 * 2 = 140km

Arean av det skuggade området i figuren 3.2 mellan 0 och två timmar är
Arean = 70 * 2 = 140



Figur 3.3 En bils hastighet under fyra timmar.

Arean mellan kurvan och tidsaxeln i figur 3.3 är ett mått på hur långt bilen har färdats. Hastigheten ändras enligt funktionen f(t) i figur 3.3.

Om sträckan dvs. arean önskas mellan tiden 1 och 4, kan detta betecknas med symbolen

$$\int_1^4 f(\tau)d\tau$$

och utläses integralen av f(x) från 1 till 4 [4].

3.2.2 PI-regulatorns ekvation

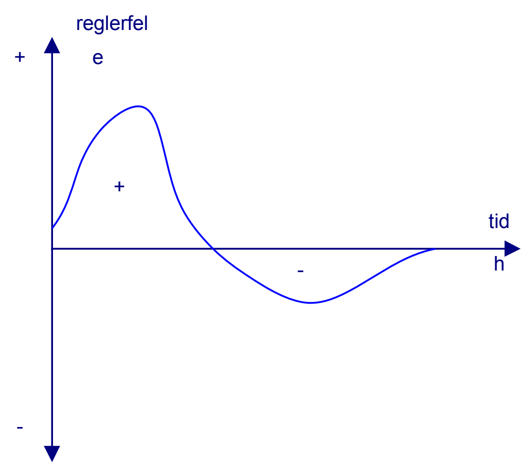
Den allmänna formeln för PI-regulatorn är [1][2][3][5]

$$\mathrm{u(t)} = \mathrm{K}\left(\mathrm{e}(\mathrm{t}) + \frac{1}{\mathrm{T_i}}\int_0^{\mathrm{t}} \mathrm{e}(\tau)d\tau\right)$$

Om K multipliceras in i parentesen kan formeln skrivas om till [1]

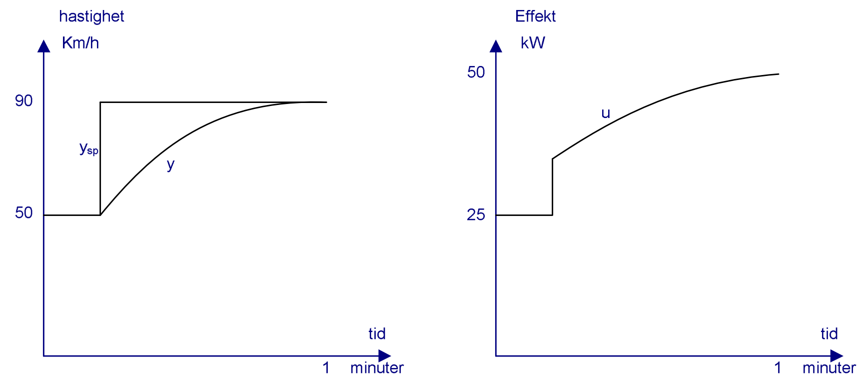
$$u(t) = K\,e(t) + K\frac{1}{T_i}\int_0^t e(\tau)d\tau$$

Denna formel läses: Regulatorns styrsignal u vid tiden t är summan av två delar. Den första är P-delen, förstärkningen K multiplicerat med reglerfelet vid tidpunkten t. Den andra delen är integraldelen, förstärkningen gånger inversen av integraltiden T_i multiplicerat med integralen av reglerfelet från tidpunkten 0 till tidpunkten t. Om integrationstiden T_i är stor blir justeringen långsam eftersom integralen multipliceras med $1/T_i$. Storheten för T_i är tid och anges vanligen i enheten sekunder eller minuter [1].



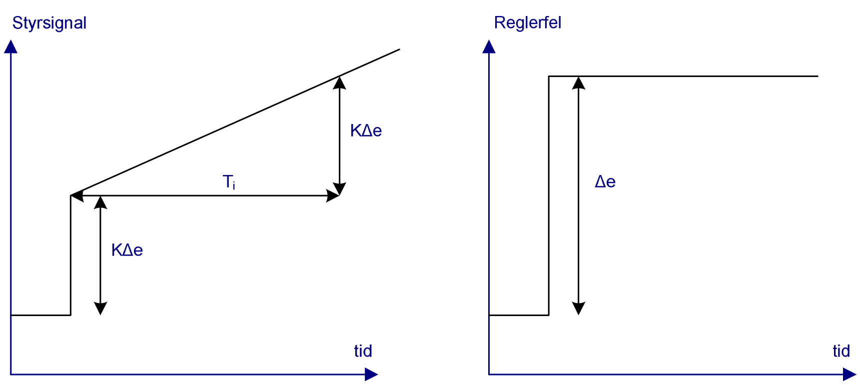
Figur 3.4 Integralen av reglerfelet e med positivt och negativt bidrag [2].

Om reglerfelet är positivt ökar integralen av e(t). Om reglerfelet är negativt minskar integralen av e(t). Integraldelen är en viktad summa av alla gamla reglerfel [2], se figur 3.4. Figur 3.5 visar resultatet av en börvärdesändring om vi använder en PI-regulator för att reglera t.ex. en bils hastighet.



Figur 3.5 Diagram som visar börvärdesändring från 50km/h till 90km/h.

Med en PI-regulatorn fås inget bestående reglerfel eftersom integralen av e(t) minskar eller ökar så länge reglerfelet inte är noll. Integraldelen påverkar styrsignalen, vilket medför att ärvärdet närmar sig börvärdet. Detta gäller inte bara vid börvärdesändringar utan även vid störningar.



Figur 3.6 Grafisk tolkning av integrationstiden T_i [2].

Styrsignalens ändring vid en ändring av reglerfelet ges enligt [2] av
$\Delta u(t) = K(1 + 1/T_i)\Delta e$.
Symbolen Δ är den grekiska bokstaven delta som i detta fall betecknar skillnad. Δe är alltså skillnaden på reglerfelet från tidigare värde till nuvarande värde. Integrationstiden kan enligt [2] illustreras enligt figur 3.6. Styrsignalen ändras först med P-delens bidrag dvs. K Δe. Därefter växer styrsignalen linjärt med en hastighet som bestäms av T_i. Styrsignalen växer till 2 K Δe efter tiden t = T_i, detta medför att P-steget har fördubblats. Integrationstiden kan tolkas som den tid det tar att fördubbla P-steget då reglerfelet är konstant. Om processen är snabb då väljs T_i kort, medan en långsam process kräver längre T_i tid eftersom styrsignalen ska förändras långsammare. Integrationstiden väljes alltså proportionellt mot processens tider [2].

3.2.2.1 Begränsning av styrsignalen

Styrsignalen har en undre gräns och en övre gräns. För att undvika att integralen av e(t) växer fast styrsignalen har nått en gräns, måste integraldelen begränsas om styrsignalen når en gräns. Om integraldelen tillåts växa trots att styrsignalen antagit den maximala eller den minimala gränsen, fås så kallad integratoruppvridning vilket måste undvikas. Den funktion som begränsar integraldelen kallas anti-windup. Om anti-windup inte används växer integraldelen positivt eller negativt, trots att styrsignalen inte kan påverka processen, eftersom styrsignalen nått det maximala eller minimala värdet. Det är inte önskvärt att integraldelen växer okontrollerat eftersom det då tar lång tid att åter minska integraldelen när styrsignalen vänder, vilket kan leda till dålig stabilitet. En anti-windup metod är att sluta uppdatera integraldelen om u når max eller min, en annan är att man låter skillnaden mellan den begränsade styrsignalen och den obegränsade styrsignalen påverka integralens värde. Mera information om metoder för anti-windup ges i avsnitt 4.

3.2.2.2 Startvärde på I-delen

För att minska tiden på insvängningsförloppet efter start av en regulator med integralverkan kan integraldelen ges ett startvärde [3, 56].

3.2.3 Börvärdesviktningsfaktorn b

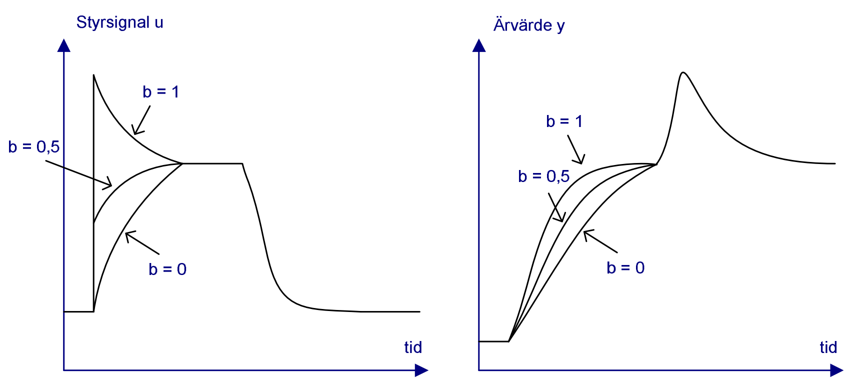
En regulator har enligt [1] två huvudfunktioner, börvärdesföljning och störningsundertryckning. För att en regulator inte ska reagera så häftigt vid en börvärdesändring kan börvärdesviktning användas. Formeln för en PI-regulator med börvärdesviktning ges nedan [1][5].

$$u(t) = K\left(b\, y_{sp}(t) - y(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau \right)$$

Om K multipliceras in i parentesen kan formeln skrivas om till

$$u(t) = K\left(b\, y_{sp}(t) - y(t)\right) + K\frac{1}{T_i}\int_0^t e(\tau)d\tau$$

Börvärdesviktningsfaktorn b ges värden mellan 0 och 1. Figur 3.7 visar att styrsignalen inte reagerar lika kraftfullt på en börvärdesändring om b < 1. Regulatorns förmåga att undertrycka störningar påverkas inte av värdet på b.



Figur 3.7 PI-regulator, börvärdessteg och stegformad störning med olika b värden [1][5].

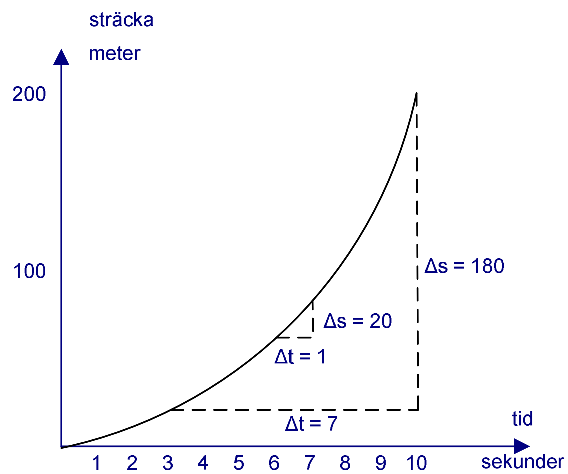
3.3 *PID-regulatorns funktion*

En egenskap enligt [2] som begränsar PI-regulatorn är att den inte försöker prediktera vad som kommer att hända med reglerfelet i den närmaste framtiden. D-delen i en PID-regulator är proportionell mot derivatan av reglerfelet. D-et står för deriverande verkan [1][2]. Idén med att använda derivatan av reglerfelet är att regulatorn då får en prediktiv förmåga att uppskatta vart reglerfelet är på väg. PID-regulators olika delar kan sammanfattas som att P-delen representerar nutid, I-delen förfluten tid, och D-delen framtid [1][5].

3.3.1 Vad menas med derivata?

3.3.1.1 Ändringskvot

Figur 3.8 visar hur en bil förflyttas från startpositionen med tiden.



Figur 3.8 En bils förflyttning under 10 sekunder.

Hur stor är bilens medelhastighet mellan tiden 0 och tiden 10?
Sträckan s = Farten v * Tiden t
$v = s / t$
$v = 200 / 10 = 20$ meter/sekund

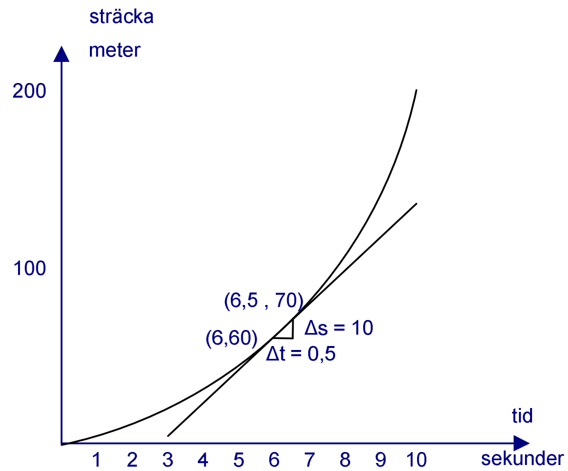
Hur stor är bilens medelhastighet under tiden 3 till tiden 10?
$v = 180 / 7 = 25{,}7$ m/s

Hur stor är bilens medelhastighet under tiden 6 till tiden 7?
$v = 20 / 1 = 20$ m/s

Hur stor är bilens hastighet vid tiden 6?
Om Δt minskar eller med andra ord går mot noll då fås momentanhastigheten till 20 m/s. Hastigheten 20 m/s är ett så kallat gränsvärde. Hastigheten i ett visst ögonblick är alltså det gränsvärde som ändringskvoten närmar sig då tidsintervallet närmar sig noll. Detta skrivs $\Delta t \rightarrow 0$.
Hastigheten = gränsvärdet av $\Delta s / \Delta t$ då $\Delta t \rightarrow 0$.

3.3.1.2 En kurvas lutning

En rät linjes riktningskoefficient k bestäms av k = Δy / Δx. I figur 3.9 ritas en rät linje genom punkterna (6,60) och (6,5 , 70).

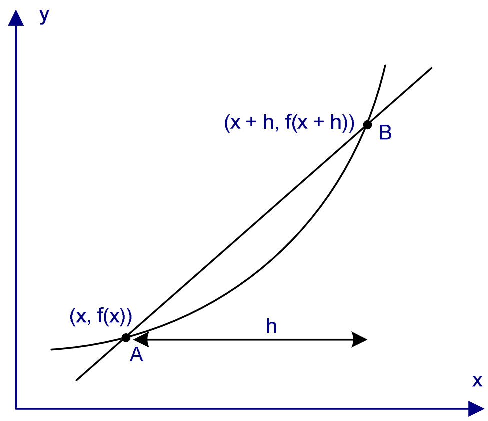


Figur 3.9 En bils förflyttning under 10 sekunder samt en rät linje inritad.

Den räta linjens riktningskoefficient blir k = Δs / Δt = 10 / 0,5 = 20 m/s. Linjens riktningskoefficient motsvarar alltså medelhastigheten mellan punkterna.
Antag att punkten (6,5 , 70) är en rörlig punkt på kurvan. När (6,5 , 70) närmar sig punkten (6,60) blir Δt mindre och mindre. Hastigheten = gränsvärdet av Δs / Δt då Δt → 0. Linjen genom punkten (6,60) och (6,5 , 70) kommer att närma sig ett gränsläge där den tangerar kurvan i punkten (6,60). Linjen är då en tangent till kurvan i punkten (6,60). Derivatan = tangentens k-värde [4].

3.3.1.3 Derivatans definition

Hur bestäms k-värdet/derivatan för en tangent till en godtycklig funktion y = f(x)?



Figur 3.10 Rät linje genom punkterna A och B [4].

Linjen genom punkterna A och B i figur 3.9 har k-värdet

$$k = \frac{\Delta y}{\Delta x} = \frac{f(x+h) - f(x)}{x+h-x} = \frac{f(x+h) - f(x)}{h}$$

Nu låter vi punkten B närma sig punkten A. Detta medför att sträckan h blir mindre och mindre. Detta kallas att h går mot noll, h → 0.

Tangenten i punkten A har k-värdet

$$k = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Detta kallas derivatans definition [4]. Limes förkortas lim som betyder gräns på latin.

Derivatan = tangentens k-värde [4].

Derivatan av en funktion f(x) anger hur snabbt den ändras när den oberoende variabeln x ökar. Derivatan betecknas vanligen f´(x) (uttalas 'f-prim av x') eller df/dx (uttalas 'd-f, d-x').
$f(x) = x^2$ har derivatan $f'(x) = 2x$
$f(x) = x^2$ har derivatan $df/dx = 2x$ (annan beteckning)

3.3.2 PID-regulatorns ekvation

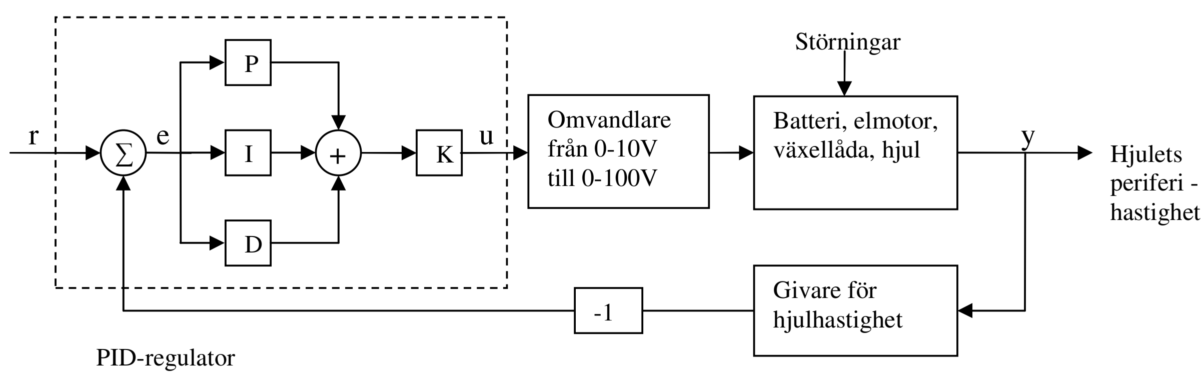
Den allmänna formeln för en PID-regulator är [1]

$$u(t) = K\left(b\ y_{sp}(t) - y(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\, e'(t)\right)$$

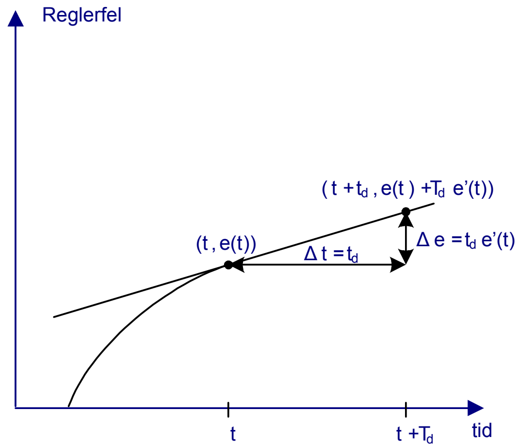
Där T_d är derivatatiden. Om K multipliceras in i parentesen kan formeln skrivas om till

$$u(t) = K\big(b\ y_{sp}(t) - y(t)\big) + K\frac{1}{T_i}\int_0^t e(\tau)d\tau + K\ T_d\, e'(t)$$

Denna formel läses: Regulatorns styrsignal u vid tiden t är summan av tre delar. Den första är P-delen, börvärdet y_{sp} multiplicerat med börvärdesviktningsfaktorn b, minus ärvärdet y vid tidpunkten t, som multipliceras med förstärkningen K. Den andra delen är integraldelen, förstärkningen gånger inversen av integraltiden T_i multiplicerat med integralen av reglerfelet från tidpunkten 0 till tidpunkten t. Om integrationstiden T_i är stor blir justeringen långsam eftersom integralen multipliceras med $1/T_i$. Storhet för T_i är tid och anges vanligen i enheten sekunder eller minuter [1]. Den tredje delen är derivatadelen, förstärkningen multiplicerat med derivatatiden T_d gånger derivatan av reglerfelet vid tidpunkten t. Om derivatatiden T_d är stor fås en uppskattning av reglerfelet en längre tid in i den närmaste framtiden. Storheten för T_d är tid och anges vanligen i enheten sekunder eller minuter. Ett exempel på en farthållares blockschema visas i figur 3.11.



Figur 3.11 Ett exempel på en farthållares blockschema, realiserad med en PID-regulator. Utsignalen från en PID-regulator är summan av tre delar. Princip från [1][3][3, sid. 10][5, sid. 3].



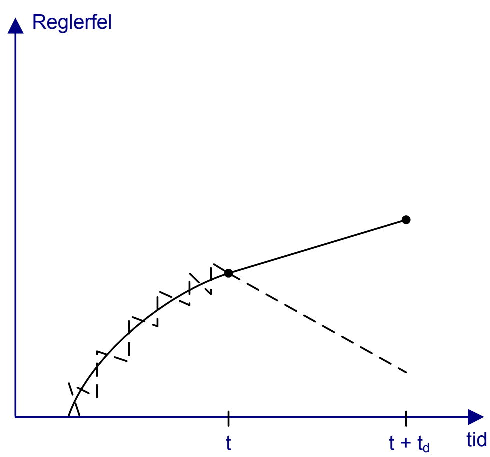
Figur 3.12 Tolkning av derivatatiden T_d. Principer från [2] och [4].

Figur 3.12 visar att regulatorns derivatadel ger möjlighet att prediktera reglerfelet tiden T_d framåt i tiden. En rät linje används för att extrapolera funktionen e(t) fram till tidpunkten t + T_d. En rät linjes riktningskoefficient k bestäms av k = Δy / Δx. Den räta linjen är en tangent till funktionen e(t). Derivatan av e(t) = e′(t) vilket är det samma som tangentens k-värde.

$$\Delta e = T_d e'(t)$$

$$e(t + T_d) = e(t) + T_d e'(t)$$

För att prediktionen ska vara användbar kan inte derivatatiden T_d vara för stor, eftersom prediktionen bara är tillförlitlig en begränsad tid in i den närmaste framtiden. Anledningen till detta är att tillförlitligheten på en prediktion försämras, om tidsspannet in i framtiden ökas. I snabba processer förändras reglerfelet snabbt, vilket medför att T_d måste väljas till ett mindre tal än om processen är långsam. Derivatatiden ska väljas proportionell mot processens tider [2].



Figur 3.13: Den streckade linjen visar en signal med brus som gör prediktionen t_d tidsenheter in i framtiden otillförlitlig. Den heldragna linjen visar en signal som inte innehåller brus.

Om mätsignalen y innehåller brus är det olämpligt att använda derivatadelen eftersom prediktionen inte är tillförlitlig, se figur 3.13. Derivatadelen förstärker bruset om en brusig signal inte filtreras. Om en brusig mätsignal filtreras genom ett lågpassfilter som dämpar ut höga frekvenser kan derivatadelens funktionalitet förbättras.

Nackdelen med filtrering enligt [1, sid. 60] är att filtret ger upphov till en viss fördröjning/fasförskjutning av signalen. En fördröjning försämrar derivatadelens prediktions tillförlitlighet, och kan även försämra reglersystemets stabilitet. Det gäller att hitta en avvägning mellan hur hårt man vill filtrera mätvärdet och hur stor fördröjning man kan tillåta [1, sid. 61].

Om börvärdet ändras, då ändras e lika mycket vilket medför att derivatadelen blir mycket stor. Förklaringen till detta är att, om storleken på e vid tidpunkten t skiljer sig mycket från e vid tiden t + T_d, då blir den räta linjen i figur 3.12 brant. För att undvika detta låter man D-delen verka på mätvärdet y istället för reglerfelet e = börvärdet y_{sp} – ärvärdet y. Då fås följande ekvation där y_f är den filtrerade mätsignalen [1, sid. 58]

$$u(t) = K\left(b\, y_{sp}(t) - y(t) + \frac{1}{Ti}\int_0^t e(\tau)d\tau - T_d\, y_f'(t)\right)$$

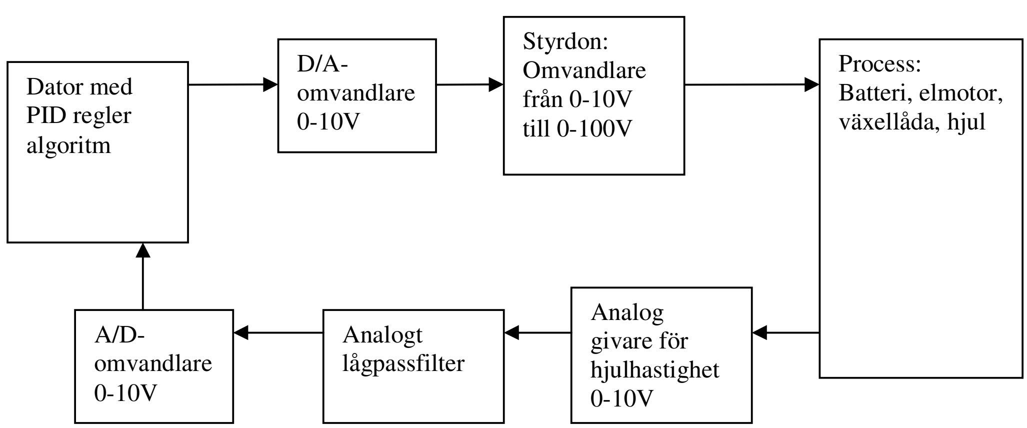
vilket även kan skrivas

$$u(t) = K\left(b\, y_{sp}(t) - y(t)\right) + K\frac{1}{T_i}\int_0^t e(\tau)d\tau - K\, T_d\, y_f'(t)$$

4 Implementering av en PID-regulator i mjukvara

4.1 Komponenter i tidsdiskreta reglersystem

En PID-regulator kan realiseras på flera sätt t.ex. med hjälp av analog teknik eller mjukvara i en dator. En regulator som implementeras i mjukvara arbetar i diskret tid medan det fysikaliska system som regleras arbetar i kontinuerlig tid. Med diskret tid menas att styrsignalen endast ändras vid diskreta tidpunkter, t.ex. en gång per sekund eller oftare [3]. Mätsignalen är även den diskret eftersom den samplas vid bestämda tidpunkter.
För att bygga ett tidsdiskret reglersystem behövs komponenterna i figur 4.1 [3, sid. 263].

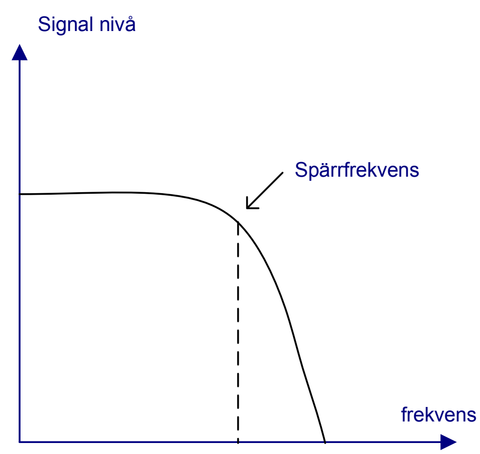


Figur 4.1 Datorbaserad farthållare till en bil, princip från [3, sid. 264].

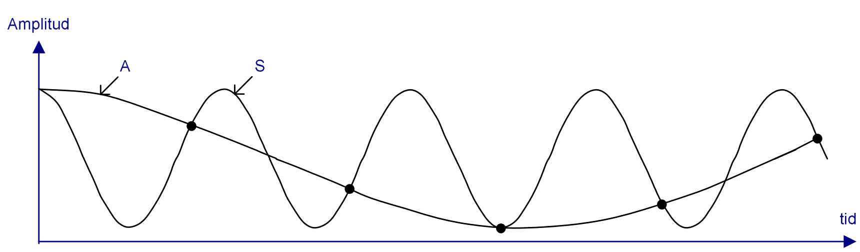
De olika delarna i figur 4.1 beskrivs nedan.

Givare: Givaren mäter processens ärvärde vilket i detta fall är hjulhastigheten.

Analogt lågpassfilter: Alias-effekten innebär att höga frekvenser misstolkas som låga frekvenser på grund av att mätvärdet samplas, se figur 4.3. För att undvika aliaseffekten måste det finnas ett lågpassfilter innan A/D omvandlaren. Ett lågpassfilter tillåter signaler med frekvenser som är lägre än spärrfrekvensen att passera, medan högre frekvenser dämpas, se figur 4.2. Filtret måste eliminera alla frekvenser som är högre än halva samplingsfrekvensen [5, sid. 414]. Om inte ett lågpassfilter används kommer höga frekvenser att misstolkas som låga frekvenser, vilket medför att regulatorn kommer att reglera på felaktiga mätvärden. Ett exempel på när det kan uppstå frekvenser, som är högre än halva samplingsfrekvensen är, mätningen av nivån i en tank vars innehåll skvalpar.

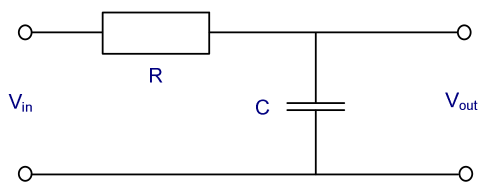


Figur 4.2 Lågpassfilter [6].



Figur 4.3 Signalen S uppfattas efter sampling som aliassignalen A [5, sid. 414].

Ett enkelt lågpassfilter kan skapas med ett motstånd R och en kondensator C [6, sid. 45], se figur 4.4.



Figur 4.4 Första ordningens lågpass RC filter [6, sid. 45].

En kondensator kan lagra elektrisk energi i ett elektriskt fält. Strömmen I genom kondensatorn C är lika med derivatan av spänningen V_{out} över kondensatorn multiplicerat med kapacitansen C som mäts i coulomb/volt eller Farad.

$$I = C\ V'_{out}(t)$$

Spänningen V_{out} över kondensatorn ges av

$$V'_{out}(t) = \frac{I}{C}$$

Om frekvensen över kondensatorn i figur 4.4 ökas, ökar strömmen genom kondensatorn vilket medför att spänningsfallet över resistorn ökar. Ett ökat spänningsfall över resistorn leder i sin tur till att V_{out} blir lägre än V_{in}. Det går att beskriva hur V_{out} beror på V_{in} med hjälp av Kirchhoffs spänningslag. Kirchhoffs spänningslag (även kallad Kirchhoffs 2:a lag) gäller för spänningar i ett elektriskt nät och lyder enligt följande: Summan av samtliga grenspänningar som ingår i en slinga är noll. Ohms lag: U = R I. Kirchhoff spänningslag applicerad på figur 4.4 ger

$$V_{in} - V_r - V_{out} = 0$$

där V_r är spänningsfallet över resistorn. Genom att använda Ohms lag kan V_r ersättas med R I, vilket ger

$$V_{in} - R\ I - V_{out} = 0$$

Vilket kan skrivas om till

$$V_{in} - V_{out} = R\ I$$

$$R\ I = -V_{out} + V_{in}$$

I kan ersättas med $C\ V'_{out}(t)$ vilket ger följande differentialekvation

$$R\ C\ V'_{out}(t) = -V_{out}(t) + V_{in}(t)$$

Eftersom enheten för R är volt / ampere och enheten för C är ampere / (volt / s) har $T_f = R\ C$ enheten sekunder s.

Lågpassfiltret i figur 4.4 får då följande ekvation

$$T_f\ V_{out}'(t) = -V_{out}(t) + V_{in}(t)$$

Sambandet mellan ett lågpassfilters tidskonstant T_f och dess spärrfrekvens är T_f = 1 / spärrfrekvensen.

A/D-omvandlare: Analog till digitalomvandlarens uppgift är att omvandla den analoga signalen till en binärsignal som kan läsas av datorn. Upplösningen hos omvandlaren är beroende av hur många bitar som används för att representera mätvärdet.

Dator: I datorn exekveras en PID regleralgoritm som läser av värdet från A/D omvandlaren och beräknar ett värde på styrsignalen. Styrsignalens värde läggs ut som ett binärt tal till D/A omvandlaren.

D/A-omvandlare: Digital till analogomvandlaren används för att omvandla den binära signalen till en analog signal.

Styrdon: Styrdonets uppgift är att påverka den styrda processvariabeln. I figur 4.1 är det spänningen som ändras för att få elmotorn att rotera med en viss hastighet så att hjulhastigheten kan påverkas.

4.2 Tidsdiskret PID-regulator

En tidskontinuerlig PID-regulator består av tre delar P, I och D del enligt ekvationen

$$u(t) = K\left(b\, y_{sp}(t) - y(t)\right) + K \frac{1}{T_i} \int_0^t e(\tau) d\tau - K\, T_d\, y_f'(t)$$

En förteckning över de variabler som kommer att användas i detta kapitel listas i tabell 4.1.

Variabel	Beskrivning
u	Styrsignal (begränsad styrsignal)
v	Obegränsad styrsignal
ulow	Minimalt värde på u
uhigh	Maximalt värde på u
K	Förstärkning K
b	Börvärdesviktningsfaktorn, värde 0-1
y_{sp}	Börvärde, sp är en förkortning för engelskans ”set point”
y	Ärvärde
T_i	Integrationstiden i t.ex. sekunder
T_t	Tidskonstant i t.ex. sekunder som styr hur snabbt integraldelen ska återställas efter en begränsning av styrsignalen. T_t benämns ”tracking time constant” på engelska.
e	Reglerfel, regleravvikelse $e = y_{sp} - y$
T_d	Derivatatiden i t.ex. sekunder
y_f	Filtrerat ärvärde
h	Samplingstid i t.ex. sekunder
P	Proportionaldelen
I	Integraldelen
t_k	Beteckningen t_k används istället för absolut tid t. Varje sampling numreras löpande där k betecknar löpnumret. Följande gäller då $t_k - t_{k-1} = h$
t	Tidpunkt i t.ex. sekunder
D	Derivatadelen
T_f	Lågpassfilterts tidskonstant i t.ex. sekunder
yfold	Samma som $y_f(t_{k-1})$, det filtrerade mätvärdet från den förra exekveringen
D_f	Filtrerad derivatadel

Tabell 4.1 Variabeltabell [1].

4.2.1 P-delen

Den tidskontinuerliga proportionaldelen ges av

$$P(t) = K\left(b\, y_{sp}(t) - y(t)\right)$$

Proportionaldelen kan implementeras direkt i tidsdiskret form genom att ersätta de kontinuerliga variablerna med samplade variabler [5, sid. 415].

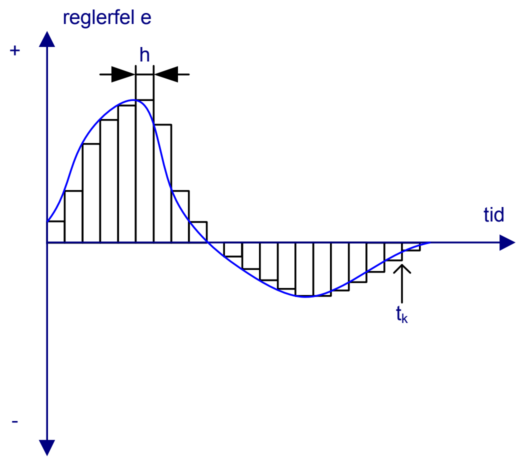
$$P(t_k) = K\left(b\, y_{sp}(t_k) - y(t_k)\right) \quad (4.1)$$

4.2.2 I-delen

Den tidskontinuerliga integraldelen ges av [5, sid. 415]

$$I(t) = K\frac{1}{T_i}\int_0^t e(\tau)d\tau$$

Denna ekvation kan approximeras med summan av alla de rektangelareor som fås om varje reglerfelsvärde $e(t_k)$ fram till den senast samplingen t_k multipliceras med längden på samplingstiden h. Ju kortare samplingstid h desto bättre approximation [3, sid. 269]. Se figur 4.5.



Figur 4.5 Princip för approximering av reglerfelets integral [3].

Om areafunktionen med tecken i figur 4.5 betecknas med A(t) då ges arean av

$$A(t) = \int_0^t e(\tau)d\tau$$

Areafunktionen A(t) är en primitiv funktion till e(t) vilket medför att

$$A'(t) = e(t)$$

Enligt derivatans definition kan derivatan approximeras med nedanstående framåt-differens

$$A'(t) \approx \frac{e(t+h)\text{-}e(t)}{h}$$

Den tidskontinuerliga integraldelen ges av [5, sid. 415]

$$I(t) = K\frac{1}{T_i}\int_0^t e(\tau)d\tau$$

En derivering av ovanstående ekvation ger [5, sid. 415]

$$I'(t) = K\frac{1}{T_i}e(t)$$

Detta kan skrivas om med hjälp av derivatans approximation till nedanstående. Beteckningen t_k används ovan istället för absolut tid t. Varje sampling numreras löpande där k betecknar löpnumret följande gäller då $t_k - t_{k-1} = h$.

$$\frac{I(t_{k+1}) - I(t_k)}{h} = K \frac{1}{T_i} e(t_k)$$

Ekvationen ovan kan skrivas som en rekursiv ekvation [5, sid. 415]

$$I(t_{k+1}) = I(t_k) + K \frac{1}{T_i} h\, e(t_k)$$

Vilket kan tolkas som att, h $e(t_k)$ ger en rektangelarea som multipliceras med K och $1/T_i$. $I(t_k)$ innehåller resultatet från tidigare beräkningar.

Den tidsdiskreta formen kan skrivas som [5, sid. 415]

$$I(t_{k+1}) = I(t_k) + \frac{K\,h}{T_i} e(t_k)$$

4.2.3 I-delen med anti-windup

Styrsignalen u har ett minimalt värde och ett maximalt värde eftersom ett styrdon som exempelvis omvandlaren i figur 4.1 inte kan öka spänningen till mer än 100 volt, inte heller kan spänningen sänkas till mindre än 0 volt. Att låta integraldelen växa positivt eller negativt trots att styrsignalen har begränsats är inte önskvärt. Anledningen till detta är att när reglerfelet senare byter tecken tar det lång tid för integraldelen att ändras så att styrsignalen ändras. För att t.ex. hastighetsbörvärdet i exemplet från figur 4.1 ska kunna hållas, kan inte styrsignalen tillåtas att fastna på det minimala eller maximala värdet, i väntan på att integraldelen ska återställas. Integraldelen måste begränsas så att integraldelen inte växer om minimal eller maximal styrsignal uppnås, detta kallas integratoruppvridning. För att förhindra integratoruppvridning (engelska: windup) måste en anti-windup mekanism införas.

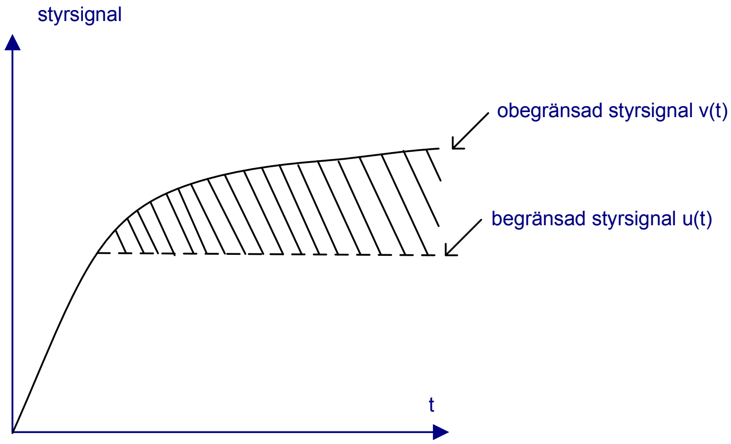
En anti-windup metod är att använda sig av ”Back-calculation and tracking”, denna metod beskrivs nedan [5, sid. 80]. En I-regulator utan anti-windup ges av

$$I(t) = K \frac{1}{Ti} \int_0^t e(\tau) d\tau$$

Följande funktion saturateOutput införs som en hjälpfunktion för att begränsa den obegränsade styrsignalen så att den begränsade styrsignalen u inte antar värden utanför minimum ulow eller maximum uhigh [5, sid. 428].

$$u(t_k) = saturateOutput(v(t_k),\ uLow,\ uHigh) \quad (4.2)$$

Givet parametrarna v, uLow och uHigh returnerar funktionen saturateOutput ett begränsat värde u enligt nedan.
Om uLow ≤ v ≤ uHigh returneras u = v
Om v > uHigh returneras u = uHigh
Om v < uLow returneras u = uLow



Figur 4.6 Obegränsade styrsignalen v(t) och den begränsade styrsignalen u(t).

Den negativa arean av det streckade området i figur 4.6 ges av

$$A_{streck} = \int_0^t [u(\tau) - v(\tau)]\, d\tau$$

Detta är den area med tecken som adderas till I-delen för att I-delen inte ska överstiga eller understiga begränsningarna ulow och uhigh. Man skulle kunna tänka sig att direkt justera I-delen med hela den streckade arean i figur 4.5, men detta är inte önskvärt, eftersom en tillfällig förändring av mätsignalen kan få derivatadelen och proportionaldelen att bidra med ett värde som mättar u, och därmed oavsiktligt nollställer integraldelen.
För att undvika detta införs tidskonstanten T_t. Tidskonstanten T_t kallas ”tracking time constant” på engelska [5, sid. 80].

$$A_{streck} = \frac{1}{T_t}\int_0^t [u(\tau) - v(\tau)]\, dt$$

Tidskonstanten T_t styr hur snabbt integraldelen ska återställas efter en begränsning av styrsignalen. Om tidskonstanten T_t väljes för liten, kan en tillfällig förändring av mätsignalen få derivatadelen och proportionaldelen att bidra med ett värde som mättar u, och därmed oavsiktligt nollställer integraldelen. Ett exempel på när detta kan inträffa är när en omrörare i en tank startas, varvid det bildas vågor i tanken som påverkar mätsignalen.
Tidskonstanten T_t ska vara större än T_d men mindre än Ti. En tumregel är att välja T_t enligt $T_t = \sqrt{T_i T_d}$ [5, sid. 80] för en PID-regulator och för en PI-regulator $T_t = 0{,}5\ T_i$ [7, sid. 5].

Den tidskontinuerliga integraldelen med anti-windup ges av följande ekvation, princip från [6, sid. 289]

$$I(t) = K\frac{1}{T_i}\int_0^t e(\tau)d\tau + \frac{1}{T_t}\int_0^t [u(\tau) - v(\tau)]\, d\tau$$

Ovanstående ekvation kan approximeras enligt samma princip som i avsnitt 4.2.2.
Efter en derivering av ovanstående ekvation erhålls

$$I'(t) = K\frac{1}{T_i}e(t) + \frac{1}{T_t}(u(t) - v(t))$$

Detta kan skrivas om med hjälp av derivatans approximation till

$$\frac{I(t_{k+1})\text{-}I(t_k)}{h} = K\frac{1}{T_i}e(t_k) + \frac{1}{T_t}(u(t_k) - v(t_k))$$

Vilket kan skrivas som en rekursiv ekvation

$$I(t_{k+1}) = I(t_k) + K\frac{1}{T_i}h\,e(t_k) + \frac{1}{T_t}h(u(t_k) - v(t_k))$$

Vilket kan tolkas som att, h $e(t_k)$ ger en rektangelarea som multipliceras med K och $1/T_i$. $I(t_k)$ innehåller resultatet från tidigare beräkningar. h $(u(t_k)\text{-}v(t_k))$ ger en rektangelarea för den del som överstiger den begränsade signalen $u(t_k)$ multiplicerat med $1/T_t$.

Den tidsdiskreta ekvationen ovan kan skrivas om, med regleravvikelse e ersatt med $y_{sp} - y$ till

$$I(t_{k+1}) = I(t_k) + \frac{K\,h}{T_i}(y_{sp} - y) + \frac{h}{T_t}(u(t_k) - v(t_k))$$

För att spara CPU-kraft kan ekvationen ovan skrivas om så att tre delar bildas. Parameterekvationerna 4.3 och 4.4 exekveras bara om någon variabel i ekvationerna ändras [5, sid. 428, F13.19].

$$p3 = \frac{K\,h}{T_i} \quad (4.3)$$

$$p4 = \frac{h}{T_t} \quad (4.4)$$

$$I(t_{k+1}) = I(t_k) + p3(y_{sp} - y) + p4(u(t_k) - v(t_k)) \quad (4.5)$$

4.2.3.1 Villkorlig integrering

Risken med anti-windup funktionen i avsnitt 4.2.3 är att, en tillfällig ändring av mätsignalen y kan få D-delen att generera ett stort värde, som nollställer I-delen om tiden T_t värjs fel. Enligt [9] finns det en metod som kallas villkorlig integrering (conditional integration) som inte har denna risk. Denna metod är en enkel och lätt att förklara. Enligt [9] är villkorlig integrering att rekommendera i de flesta sammanhang, se figur 4.7.

```
while(true) {
   y=getProcessVariable()      // get process variable
   e=ysp-y                     // compute error
   P=K*e                       // proportional part
   D=K*Td*(e-eold)/h           // derivative part
   v=P+I+D                     // compute nominal output
   u=saturateOutput(v, uLow, uHigh) // saturate output
   setManipulatedVariable(u)   // set manipulated variable
   if u==v {
      I=I+K*h/Ti*e             // integral part
   }
   eold=e                      // update e old
   wait(h)                     // wait h seconds
}
```

Figur 4.7 Regulator med villkorlig integrering. Derivering på e och inget filter på D-delen.

4.2.4 D-delen

Den tidskontinuerliga derivatadelen ges av

$$D(t) = -K\, T_d\, y_f'(t)$$

En approximation av derivatan i ekvationen ovan fås genom att använda bakåt-differens enligt nedan.

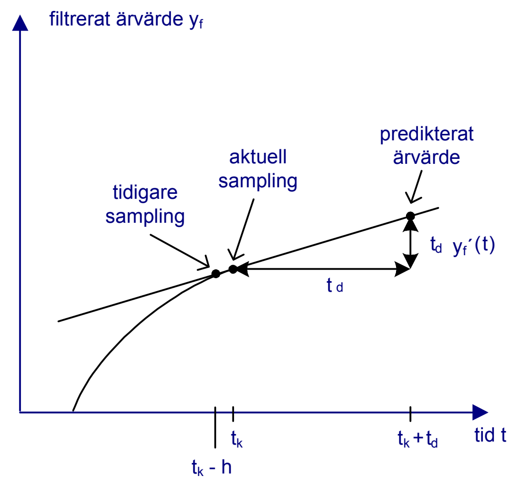
$$y_f'(t_k) \approx \frac{y_f(t_k) \text{-} y_f(t_k - 1)}{h}$$

Derivatan av det filtrerade ärvärdet y_f vid tidpunkten t_k approximeras genom att beräkna skillnaden mellan den aktuella samplingen $y_f(t_k)$ och den tidigare samplingen $y_f(t_{k-1})$ delat med samplingsintervallet h. Ju kortare samplingstid h desto bättre approximation [3, sid. 268].

Den tidsdiskreta formen av den tidskontinuerliga ekvationen ovan kan enligt [3, sid. 269] skrivas som

$$D(t_k) = -K\, T_d \frac{y_f(t_k) \text{-} y_f(t_{k-1})}{h}$$

Samplingspunkterna och det predikterade ärvärdet illustreras av figur 4.8.



Figur 4.8 Illustration av samplings punkter och det predikterade ärvärde.

För att spara CPU-kraft kan ekvationen ovan skrivas om så att två delar bildas. Parameter ekvationen 4.6 exekveras bara om någon variabel i ekvationen ändras, delvis från [5, sid. 428, F13.19].

$$p5 = -K \frac{T_d}{h} \quad (4.6)$$

$$D(t_k) = p5\left(y_f(t_k) \text{-} y_f(t_{k-1})\right) \quad (4.7)$$

4.2.4.1 Digitalt lågpassfilter

Som tidigare nämnts i avsnitt 3.3.2, måste mätvärdet filtreras för att D-delens prediktering ska vara tillförlitlig om mätsignalen innehåller brus, se även figur 3.13. En D-del utan filter bör inte implementeras eftersom förstärkningen av bruset blir för stor. Nackdelen med filtrering enligt [1, sid. 60] är att filtret ger upphov till en fördröjning eller mera exakt en fasförskjutning av signalen. En fördröjning påverkar prediktionen negativt och även stabiliteten, vilket inte är önskvärt. Det gäller att hitta en avvägning mellan hur hårt man vill filtrera mätvärdet och hur stor fördröjning man kan tillåta [1, 61].

Filtrering kan åstadkommas med ett första ordningens lågpassfilter. Om det behövs kraftigare filtrering än ett första ordningens filter, kan två första ordningens filter seriekopplas så att ett filter av andra ordningen skapas.

Lågpass RC filtret av första ordningen i figur 4.4 avsnitt 4.1 beskrivs av

$$T_f\, V'_{out}(t) = -V_{out}(t) + V_{in}(t)$$

Om beteckningarna ändras fås

$$T_f\, y_f'(t) = -y_f(t) + y(t)$$

Där y är den ofiltrerade mätsignalen och y_f den filtrerade mätsignalen. T_f är filtrets tidskonstant.

Eftersom datorn inte kan derivera måste deriveringen approximeras för att kunna programmeras. En approximation av derivatan i ekvationen ovan fås genom att använda bakåt-differens enligt nedan.

$$T_f \frac{y_f(t_k) - y_f(t_{k-1})}{h} \approx -y_f(t_k) + y(t_k)$$

Ovan kan skrivas om till nedanstående genom att dividera med T_f

$$\frac{y_f(t_k) - y_f(t_{k-1})}{h} = -\frac{1}{T_f}\, y_f(t_k) + \frac{1}{T_f}\, y(t_k)$$

Multiplicera med h och förenkla bort h i vänsterledet

$$y_f(t_k) - y_f(t_{k-1}) = -\frac{h}{T_f}\, y_f(t_k) + \frac{h}{T_f}\, y(t_k)$$

Samla $y_f(t_k)$ i vänsterledet

$$y_f(t_k) + \frac{h}{T_f}\, y_f(t_k) = y_f(t_{k-1}) + \frac{h}{T_f}\, y(t_k)$$

Vilket kan skrivas om till

$$\left(1 + \frac{h}{T_f}\right) y_f(t_k) = y_f(t_{k-1}) + \frac{h}{T_f}\, y(t_k)$$

Dividera med $1 + \frac{h}{T_f}$ för att få nedanstående filterekvation av första ordningen

$$y_f(t_k) = \frac{y_f(t_{k-1})}{1 + \frac{h}{T_f}} + \frac{\frac{h}{T_f}}{1 + \frac{h}{T_f}}\, y(t_k)$$

För att spara CPU-kraft kan ekvationen ovan skrivas om så att två delar bildas. Parameter ekvationen 4.8 exekveras bara om någon variabel i p1 ändras [6, sid. 144].

$$p1 = \frac{1}{1 + \frac{h}{T_f}} \quad (4.8)$$

$$y_f(t_k) = p1\, y_f(t_{k-1}) + (1 - p1)y(t_k) \quad (4.9)$$

4.2.5 Hoppfri övergång mellan automatik och manuellt

En regulator kan befinna sig i antingen från, automatik eller manuellt läge. Med automatik menas att PID algoritmen kontinuerligt beräknar och uppdaterar den automatiska styrsignalen u_{auto}. I manuellt läge kan en operatör välja värdet på den manuella styrsignalen u_{man}. Vid övergångarna från automatik till manuellt och från manuellt till automatik är det önskvärt att utgående styrsignal u_{out} inte hoppar till från ett värde till ett annat.

Vid övergången från automatik till manuellt eller kontinuerligt i automatik läget, tilldelas det manuella värdet på styrsignalen u_{man} det värde som styrsignalen u_{auto} hade strax innan övergången, på så vis fås ingen ändring av utgående styrsignal u_{out}. Denna lösning är en förenklad variant av vad som beskrivs i [5, sid. 425, F13.15].

I manuellt läge exekveras PID algoritmen som om regulatorn kördes i automatik men styrsignalen u_{auto} läggs inte ut till styrdonet. Det är det värde på u_{man} som operatören anger som u_{out} antar, som i sin tur manipulerar styrdonet, dvs. u_{out} tilldelas u_{man}. För att få PID algoritmen att följa det manuella värdet som läggs ut på u_{out}, kan anti-windup funktionen som beskrivits i avsnittet 4.2.3 användas. Genom att u_{auto} tilldelas u_{out} kommer den obegränsade styrsignalen v att följa $u_{out.}$ Vilket medför att ingen ändring av utgående styrsignal u_{out} fås vid övergången från manuell till automatik [5, sid. 425, F13.15].

4.2.6 Hoppfri parameterändring

Om en regulatorparameter ändras t.ex. förstärkningen K, är det önskvärt att utgående styrsignal u_{out} inte hoppar till från ett värde till ett annat. Detta kan enligt [5, sid. 425] åstadkommas genom att I-delens tillstånd ändras enligt nedanstående ekvation, vid en parameterändring.

$$I_{new} = I_{old} + K_{old}(b_{old}\, y_{sp} - y) - K_{new}(b_{new}\, y_{sp} - y) \quad (4.10)$$

4.2.7 Principkod

Genom att sätta samman de olika delarna i figur 4.9 fås en principkod enligt principer från [5, sid. 428, F13.19] och [8, sid. 19]. För att enkelt kunna härleda ekvationerna anges ekvationernas nummer Ex.x och figurnummer Fx.x.

Koden i Figur 4.13 är hämtad från [5, sid. 428, F13.19]. Författaren av denna rapports bidrag är att härleda och stegvis förklara bakgrunden till koden i figur 4.13.

I figur 4.9 separeras filtret och D-delen eftersom det är önskvärt att kunna studera/logga det filtrerade mätvärdet. För att regulatorns svarstid ska vara så kort som möjligt är det viktigt att styrsignalen ställs ut så fort den är beräknad [8, sid. 16].

```
// calculate parameters only when settings are changed
p11=1/(1+h/Tf)      // filter constant E4.8
p13=K*h/Ti          // integral gain E4.3
p14=h/Tt            // anti-windup gain E4.4
p15=-K*Td/h         // derivative gain E4.6

// Bumpless parameter changes
I=I+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y) E4.10

// control algorithm
while(true) {
   y=getProcessVariable()     // get process variable
   P=K*(b*ysp-y)              // proportional part E4.1
   yf=p11*yfold+(1-p11)*y     // filter process variable E4.9
   D=p15*(yf-yfold)           // derivative part E4.7
   v=P+I+D                    // compute nominal output F3.10
   u=saturateOutput(v, uLow, uHigh) // saturate output E4.2
   setManipulatedVariable(u) // set manipulated variable
   I=I+p13*(ysp-y)+p14*(u-v)  // integral part E4.5
   yfold=yf                   // update yfold
   wait(h)                    // wait h seconds
}
```

Figur 4.9 Principkod för en PID-regulator där mätvärdet filtreras i ett första ordningens filter innan D-delen. Mellan `getProcessVariable()` och `setManipulatedVariable(u)` används 5 multiplikationer, 3 subtraktioner och 3 additioner.

Det går att minska tiden mellan `getProcessVariable()` och `setManipulatedVariable(u)` funktionerna något genom att dela filtret i två delar och flytta `p11*yfold` beräkningen till efter `setManipulatedVariable(u)`.
Filtret `yf=p11*yfold+(1-p11)*y` delas alltså i två delar

```
yf=Ftemp+(1-p11)*y
Ftemp=p11*yfold
```

Parentesen i D-delen `D=p15*(yf-yfold)` kan tas bort och då fås

```
D=p15*yf-yfold*p15
```

D-delen kan delas i två delar

```
D=p15*yf-Dtemp
Dtemp=yfold*p15
```

Omskrivningen av D-delen i figur 4.10 medför att variabeln `yfold` inte längre behövs.

```
// calculate parameters only when settings are changed
p11=1/(1+h/Tf)      // filter constant E4.8
p13=K*h/Ti          // integral gain E4.3
p14=h/Tt            // anti-windup gain E4.4
p15=-K*Td/h         // derivative gain E4.6

// Bumpless parameter changes
I=I+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y) E4.10

// control algorithm
while(true) {
   y=getProcessVariable()     // get process variable
   P=K*(b*ysp-y)              // proportional part E4.1
   yf=Ftemp+(1-p11)*y         // filter process variable
   D=p15*yf-Dtemp             // derivative part
   v=P+I+D                    // compute nominal output F3.10
   u=saturateOutput(v, uLow, uHigh) // saturate output E4.2
   setManipulatedVariable(u) // set manipulated variable
   I=I+p13*(ysp-y)+p14*(u-v)  // integral part E4.5
   ~~yfold=yf                 // update yfold~~
   Ftemp=p11*yf               // update part of the filter
   Dtemp=yf*p15               // update part of derivative part
   wait(h)                    // wait h seconds
}
```

Figur 4.10 Principkod för en PID-regulator där mätvärdet filtreras i ett första ordningens filter innan D-delen. Mellan `getProcessVariable()` och `setManipulatedVariable(u)` funktionerna används 4 multiplikationer, 3 subtraktioner och 3 additioner.

4.2.8 Kod med minimal svarstid

Om D-delen och filtret slås samma går det att minimera tiden mellan `getProcessVariable()` och `setManipulatedVariable(u)` och därmed minimera regulatorns svarstid till ett minimum [5, sid. 428, F13.18]. En härledning av koden i [5, sid. 428, F13.18] ges nedan.

Som tidigare nämnts i avsnitt 4.2.4 ges den tidskontinuerliga derivatadelen av $D(t) = -K\,T_d\,y'(t)$ och den tidsdiskreta formen av D-delen av

$$D(t_k) = -K\,T_d\,\frac{y(t_k)\text{-}y(t_{k-1})}{h}$$

I avsnitt 4.2.4.1 beskrevs ekvationen för ett digitalt lågpassfilter.
Om beteckningarna ändras fås

$$T_f\, D_f'(t) = -D_f(t) + D(t)$$

Där D är den ofiltrerade derivatadelen och D_f den filtrerade derivatadelen. T_f är filtrets tidskonstant. Att filtrera mätsignalen eller D-delens värde ger samma slutresultat.

En tidsdiskret ekvation av ovanstående filterekvation fås genom att använda bakåtdifferens

$$T_f\,\frac{D_f(t_k)\text{-}D_f(t_{k-1})}{h} \approx -D_f(t_k) + D(t_k)$$

Vilket kan skrivas om genom att addera $D_f(t_k)$ till båda sidorna om likhetstecknet och förenkla

$$D_f(t_k) + T_f\,\frac{D_f(t_k)\text{-}D_f(t_{k-1})}{h} = D(t_k)$$

Det går att skriva om ekvationerna till en ekvation där filter och D-del integreras. D-delens $D(t_k)$ substitueras med filterekvationen.

$$D_f(t_k) + T_f\,\frac{D_f(t_k)\text{-}D_f(t_{k-1})}{h} = -K\,T_d\,\frac{y(t_k)\text{-}y(t_{k-1})}{h}$$

Om bråken delas fås

$$D_f(t_k) + T_f\left(\frac{D_f(t_k)}{h} - \frac{D_f(t_{k-1})}{h}\right) = -K\,T_d\left(\frac{y(t_k)}{h} - \frac{y(t_{k-1})}{h}\right)$$

Parenteserna runt bråken tas bort genom att T_f respektive $-K\ T_d$ multipliceras med bråken innanför parenteserna.

$$D_f(t_k) + T_f\,\frac{D_f(t_k)}{h} - T_f\,\frac{D_f(t_{k-1})}{h} = -K\,T_d\,\frac{y(t_k)}{h} + KT_d\,\frac{y(t_{k-1})}{h}$$

Om $T_f\,\dfrac{D_f(t_{k-1})}{h}$ adderas till båda sidorna om likhetstecknet fås

$$D_f(t_k) + T_f\,\frac{D_f(t_k)}{h} = T_f\,\frac{D_f(t_{k-1})}{h} - K\,T_d\,\frac{y(t_k)}{h} + K\,T_d\,\frac{y(t_{k-1})}{h}$$

Eftersom $D_f(t_k) + T_f\,\dfrac{D_f(t_k)}{h} = \left(1 + \dfrac{T_f}{h}\right)D_f(t_k)$ så får man

$$\left(1 + \frac{T_f}{h}\right)D_f(t_k) = T_f\,\frac{D_f(t_{k-1})}{h} - K\,T_d\,\frac{y(t_k)}{h} + K\,T_d\,\frac{y(t_{k-1})}{h}$$

För att kunna förkorta bort h i nämnarna multipliceras båda sidorna med h.

$$h\left(1 + \frac{T_f}{h}\right)D_f(t_k) = T_f\,\frac{D_f(t_{k-1})}{h}h - K\,T_d\,\frac{y(t_k)}{h}h + K\,T_d\,\frac{y(t_{k-1})}{h}h$$

Efter att alla h som kan har förkortas bort och h multiplicerats in i vänsterledets parentes fås

$$(h + T_f)D_f(t_k) = T_f\,D_f(t_{k-1}) - K\,T_d\,y(t_k) + K\,T_d\,y(t_{k-1})$$

För att få $D_f(t_k)$ ensam i vänsterledet divideras båda leden med h + T_f och förkortas

$$D_f(t_k) = \frac{T_f\, D_f(t_{k-1})}{h + T_f} - \frac{K\, T_d y(t_k)}{h + T_f} + \frac{K\, T_d\, y(t_{k-1})}{h + T_f}$$

För att spara CPU-kraft kan ekvationen ovan skrivas om så att tre delar bildas. Parameterekvationerna 4.11 och 4.12 exekveras bara om någon variabel i ekvationerna ändras.

$$p3 = \frac{T_f}{h + T_f} \quad (4.11)$$

$$p24 = \frac{K\, T_d}{h + T_f} \quad (4.12)$$

$$D_f(t_k) = p3\, D_f(t_{k-1}) - p24\, y(t_k) + p24\, y(t_{k-1}) \quad (4.13)$$

I figur 4.11 ges ett första utkast till kod där ekvationerna ovan ingår.

```
// calculate parameters only when settings are changed
p3=Tf/(Tf+h)         // filter constant E4.11
p24=K*Td/(h+Tf)      // derivative gain E4.12
p5=K*h/Ti            // integral gain E4.3
p6=h/Tt              // anti-windup gain E4.4

// Bumpless parameter changes
I=I+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y) E4.10

// control algorithm
while(true) {
   y=getProcessVariable()      // get process variable
   P=K*(b*ysp-y)               // proportional part E4.1
   Df=p3*Dfold-p24*y+p24*yold  // D-part and filter E.13
   v=P+I+Df                    // compute nominal output F3.10
   u=saturateOutput(v, uLow, uHigh) // saturate output E4.2
   setManipulatedVariable(u)  // set manipulated variable
   I=I+p5*(ysp-y)+p6*(u-v)     // integral part E4.5
   Dfold=Df                    // update Dfold
   yold=y                      // update yold
   wait(h)                     // wait h seconds
}
```

Figur 4.11 Första utkastet till en PID-regulator där D-delen och ett första ordningens filter är sammanslagna. Mellan `getProcessVariable()` och `setManipulatedVariable(u)` används 5 multiplikationer, 2 subtraktioner och 3 additioner.

Ekvationen `Df=p3*Dfold-p24*y+p24*yold kan` delas i två delar, då fås

`Df=x-p24*y` (4.14)

`x=p3*Dfold+p24*yold` (4.15)

Omskrivningen av D-delen + filter i figur 4.12 medför att variabeln `Dfold` och `yold` inte längre behövs.

```
// calculate parameters only when settings are changed
p3=Tf/(Tf+h)        // filter constant E4.11
p24=K*Td/(h+Tf)     // derivative gain E4.12
p5=K*h/Ti           // integral gain E4.3
p6=h/Tt             // anti-windup gain E4.4

// Bumpless parameter changes
I=I+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y) E4.10

// control algorithm
while(true) {
   y=getProcessVariable()     // get process variable
   P=K*(b*ysp-y)              // proportional part E4.1
   Df=x-p24*y                 // part of D-part & filter E4.14
   v=P+I+Df                   // compute nominal output F3.10
   u=saturateOutput(v, uLow, uHigh) // saturate output E4.2
   setManipulatedVariable(u) // set manipulated variable
   Dfold=Df                   // update Dfold
   yold=y                     // update yold
   x=p3*Df+p24*y              // part of D-part & filter E4.14
   I=I+p5*(ysp-y)+p6*(u-v)    // integral part E4.5
   wait(h)                    // wait h seconds
}
```

Figur 4.12 Andra utkastet till en PID-regulator där D-delen och ett första ordningens filter är sammanslagna. Mellan `getProcessVariable()` och `setManipulatedVariable(u)` används 3 multiplikationer, 2 subtraktioner och 2 additioner.

Om variabeln Df i ekvationen 4.14 `x=p3*Df+p24*y` substitueras med `x-p24*y` fås
`x=p3*(x-p24*y)+p24*y`

Parentesen tas bort genom att multiplicera in p3
`x=p3*x-p3*p24*y+p24*y`

Genom utbrytning kan ovan stående ekvation skrivas om till
`x=p3*x+p24*(1-p3)*y`

Parametern p24 ersätts med p4.

p4

`x=p3*x+p24*(1-p3)*y`

Vilket kan skriva om till
`x=p3*x+p4*y` (4.16)

Parametern p4 ges av nedanstående där `p3` är substituerad av `Tf/(Tf+h)`
`p4=K*Td/(h+Tf)*(1-Tf/(Tf+h))`

Eftersom `1=(Tf+h)/(Tf+h)` kan ekvationen ovan skrivas om och sedan förenklas till

```
p4=K*Td/(h+Tf)*((Tf+h)/(Tf+h)-Tf/(Tf+h))
p4=K*Td/(h+Tf)*((~~Tf~~+h-~~Tf~~)/(Tf+h))
p4=K*Td*h/(h+Tf)*(Tf+h) (4.17)
```

Parentesen i ekvationen 4.1 `P=K*(b*ysp-y)` kan tas bort genom att `K` multipliceras in i parentesen, då fås

```
P=K*b*ysp-K*y
```

Variabeln `p24` i ekvationen `Df=x-p24*y` ersätts med `K*Td/(h+Tf)`

```
Df=x-K*Td/(h+Tf)*y
```

Variablerna `P` och `Df` i ekvationen `v=P+I+Df` från figur 4.11 kan ersättas med `K*b*ysp-K*y` respektive `x-K*Td/(h+Tf)`, vilket ger

```
v=K*b*ysp-K*y+I+x-K*Td/(h+Tf)*y
v=K*b*ysp-(K+K*Td/(Tf+h))*y+x+I
```

För att spara CPU-kraft kan ekvationen ovan skrivas om så att tre delar bildas. Parameterekvationerna 4.18 och 4.19 exekveras bara om någon variabel i ekvationerna ändras [5, sid. 428, F13.19].

```
p1=K*b (4.18)
p2=K+K*Td/(Tf+h) (4.19)
v=p1*ysp-p2*y+x+I (4.20)
```

I figur 4.13 används ovanstående ekvationer för att realisera en PID-regulator med kort svarstid.

```
// calculate parameters only when settings are changed
p1=K*b              // set-point gain E4.18
p2=K+K*Td/(Tf+h)    // PD gain E4.19
p3=Tf/(Tf+h)        // filter constant E4.11
p4=K*Td*h/(Tf+h)*(Tf+h) // derivative gain E4.17
p5=K*h/Ti           // integral gain E4.3
p6=h/Tt             // anti-windup gain E4.4

// Bumpless parameter changes
I=I+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y) E4.10

// control algorithm
while(true) {
   y=getProcessVariable()     // get process variable
   v=p1*ysp-p2*y+x+I          // compute nominal output 4.20
   u=saturateOutput(v, uLow, uHigh) // saturate output E4.2
   setManipulatedVariable(u) // set manipulated variable
   x=p3*x+p4*y                // part of D-part & filter E4.16
   I=I+p5*(ysp-y)+p6*(u-v)    // integral part E4.5
   wait(h)                    // wait h seconds
}
```

Figur 4.13 En PID-regulator där D-delen och ett första ordningens filter är sammanslagna. Regulatorn har kort svarstid eftersom antalet operationer mellan `getProcessVariable()` och `setManipulatedVariable(u)` är minimalt, 2 multiplikationer, 1 subtraktion och 2 additioner [5, sid. 428, F13.18].

5 Regulatorinställning

Syftet med detta kapitel är att ge läsaren kunskaper att använda AMIGO (Approximate M-constrained integral gain optimization) metoden för att hitta lämpliga regulatorparametrar. Att rätt regulatorparametrar väljs är viktigt för att regleringen ska uppfylla nedanstående krav [7]:

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

5.1 Regulatorparametrar

I tabell 5.1 listas de parametrar som måste anges innan en PID-regulator kan användas.

Parameter	Beskrivning
ulow	Minimalt värde på styrsignalen u
uhigh	Maximalt värde på styrsignalen u
y_{sp}	Börvärde, sp är en förkortning för engelskans ”set point”
T_t	Tidskonstant ”tracking time constant” i t.ex. sekunder
h	Samplingstid i t.ex. sekunder
$T_{analog\ f}$	Tidskonstant i t.ex. sekunder för det analoga mätvärdesfiltret
T_f	Derivatafiltrets tidskonstant i t.ex. sekunder
K	Förstärkning
T_i	Integrationstiden i t.ex. sekunder
T_d	Derivatatiden i t.ex. sekunder
b	Börvärdesviktningsfaktorn, värde 0-1

Tabell 5.1 Regulatorparameter.

ulow, uhigh: Anger det minsta respektive det största tillåtna värdet på styrsignalen.

y_{sp}: Börvärdet dvs. det värde som regulatorn ska följa.

T_t: Tidskonstant i t.ex. sekunder som styr hur snabbt integraldelen ska återställas efter en begränsning av styrsignalen. T_t benämns ”tracking time constant” på engelska. Tidskonstanten T_t ska vara större än T_d men mindre än Ti. En tumregel är att välja T_t enligt $T_t = \sqrt{T_i T_d}$ [5, sid. 80] för en PID-regulator och för en PI-regulator $T_t = 0,5\ T_i$ [7, sid. 5].

h: Samplingstiden är tiden i t.ex. sekunder mellan varje körning av regulator algoritmen. Samplingsfrekvensen = 1 / h. En tumregel är att välja samplingstiden kortare än en femtedel av processens tidskonstant T [1, sid. 75].

$T_{analog\ f}$: Det analoga mätvärdesfiltrets tidskonstant i t.ex. sekunder. För att undvika aliaseffekten måste filtret eliminera alla frekvenser som är högre än halva samplingsfrekvensen [5, sid. 414]. Tidskonstanten $T_{analog\ f}$ ska väljas så att frekvenserna ovanför halva samplingsfrekvensen dämpas t.ex. 16 gånger [5, sid. 414].

T_f: Derivatafiltrets tidskonstant i t.ex. sekunder. Beroende på implementation filtrerar detta filter mätvärdet innan D-delen eller D-delens utsignal. Filtret är nödvändigt för att D-delens prediktering ska vara tillförlitlig även om mätsignalen innehåller brus, se figur 3.12.

Enligt [1, sid. 58] är tumregeln för D filtret $T_f = T_d / 10$ en bra regel till en seriell PID-regulator men inte en parallell. En mera detaljerad diskussion finns enligt [1, sid. 59] i nedanstående:
A.J. Isaksson and S.F. Graebe, Derivative filter is an integral part of PID design, IEE Proceedings - Control Theory and Applications, January 2002, Volume 149, Issue 1, p. 41-45 och B. Lennartsson, Reglerteknikens grunder, 4:e upplagan, Studentlitteratur 2002.

Enligt [5, sid. 73] kan T_f väljas enligt $T_f = T_d / N$, där typiska värden på N är mellan 2 och 20.

Nackdelen med filtrering enligt [1, sid. 60] är att filtret ger upphov till en fördröjning eller mera exakt en fasförskjutning av signalen. En fördröjning påverkar D-delens prediktion funktion negativt, vilket inte är önskvärt. Det gäller att hitta en avvägning mellan hur hårt man vill filtrera och hur stor fördröjning man kan tillåta [1, 60].

Derivatafiltret reducerar en PID-regulators robusthet men det går att kompensera för filtrets dynamik se avsnitt 5.3.2.

K: Förstärkningen K sättes till ett positivt värde om mätvärdet går åt samma håll som styrsignalen, annars negativt. Värdet på K kan t.ex. fås genom att använda AMIGO-metoden.

T_i: Integrationstiden i t.ex. sekunder. Värdet på T_i kan t.ex. fås genom att använda AMIGO-metoden.

T_d: Derivatatiden i t.ex. sekunder. Värdet på T_d kan t.ex. fås genom att använda AMIGO-metoden.

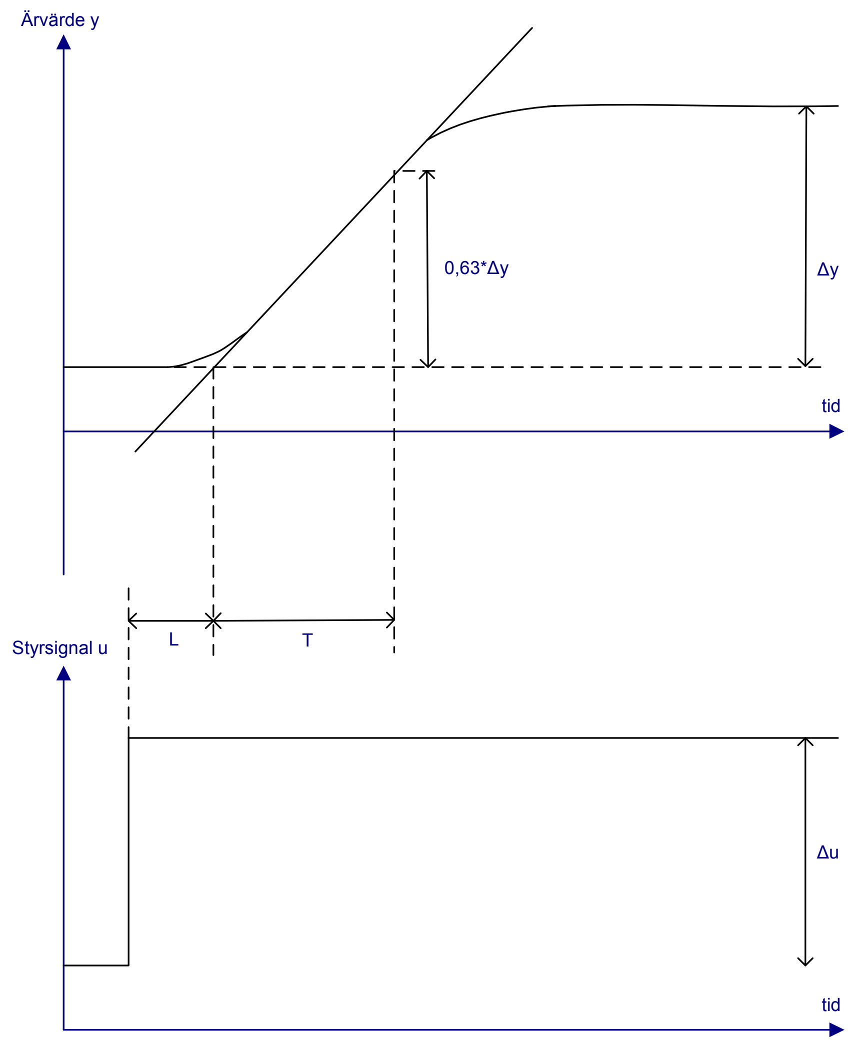
b: Börvärdesviktningsfaktorn, värde 0-1. Värdet på b kan t.ex. fås genom att använda AMIGO-metoden.

5.2 Processanalys

För att regulatorparametrarna K, Ti, Td och b ska kunna väljas rätt behövs information om processens dynamiska egenskaper, d.v.s. hur processens mätsignal reagerar då styrsignalen till processen varierar. En matematisk processmodell kan fås genom att ställa upp en differentialekvation baserat på fysikaliska lagar. Denna differentialekvation transformeras sedan till en överföringsfunktion med hjälp av Laplacetransformering. Om man känner processens överföringsfunktion kan en PID-regulators parametrar beräknas genom att använda flera olika metoder, t.ex. polplacering. Detta tillvägagångssätt kräver kunskaper i fysik och matematik på högskolenivå. Ett alternativ är att utföra experiment på en process för att få processdata och att applicera t.ex. AMIGO-metoden för att få fram regulatorparametrarna K, Ti, Td och b. Nackdelen med att enbart genomföra experiment är att processen måste finnas tillgänglig och att det inte går att simulera regleringen innan processen är byggd. Alla processer är inte lämpade för experiment eftersom de är säkerhetskritiska, t.ex. processen att flyga ett flygplan med autopilot. Inom verkstadsindustrin och processindustrin är det ofta möjligt att genomföra experiment på processen. I följande avsnitt redovisas flera olika typer av experiment.

5.2.1 Stegsvarsanalys på stabila processer

Genom att lägga regulatorn i manuellt och göra en stegändring på styrsignalen kan nedanstående figurer fås 5.1 och 5.2.

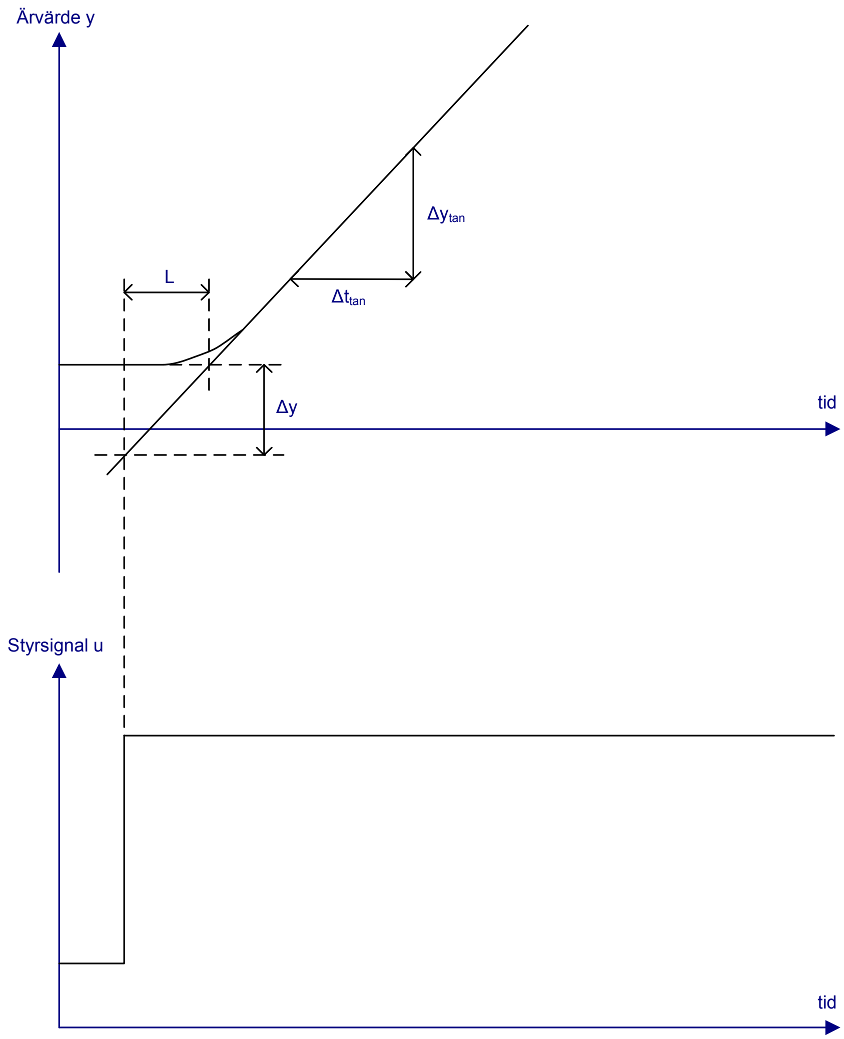


Figur 5.1 Stegsvar från en process som inte är av typen integrerande[5, sid. 49].

Den statiska förstärkningen K_p ges av $K_p = \frac{\Delta y}{\Delta u}$. En tangent som följer den brantaste lutningen på stegsvaret ritas in i figuren, se figur 5.1. Dödtiden L ges i punkten där tangenten skär det stabila mätvärdet innan styrsignalsteget genomfördes. Tidskonstanten T är den tid som det tar för mätsignalen att nå till 63 % av sitt delta värde. Dödtiden räknas dock inte med i denna tid, se figur 5.1 [5, sid. 48]. Tiden T_{63} definieras som $T_{63} = T + L$.

5.2.2 Stegsvarsanalys på integrerande processer

Om ett stegsvar utförs på en process som är av typen integrerande stabiliseras inte mätvärdet, se figur 5.2.



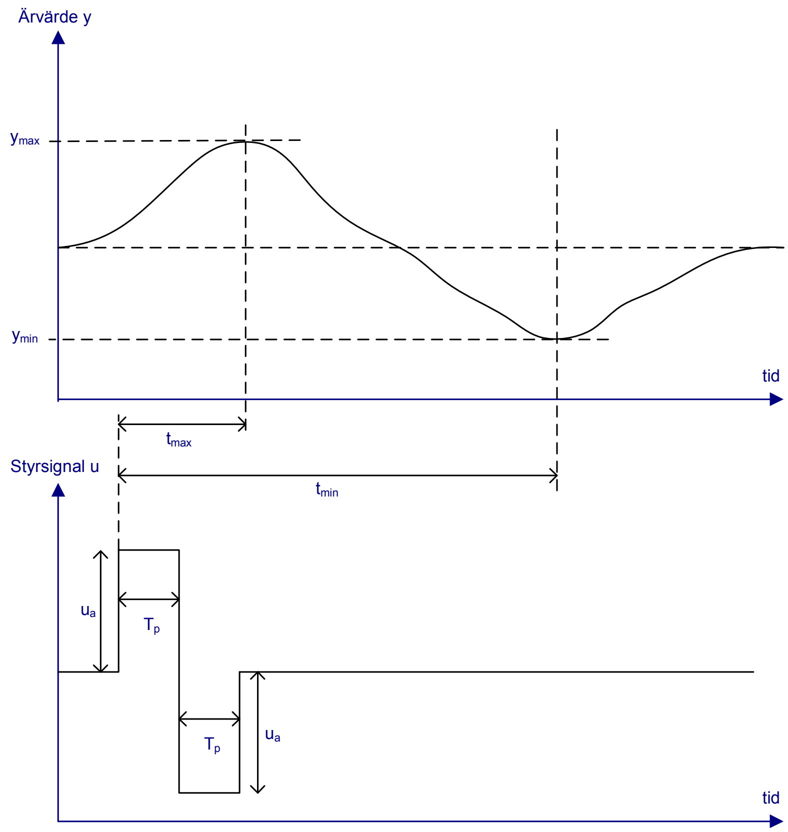
Figur 5.2 Stegsvar från en process som är av typen integrerande[5, sid. 49][2, sid. 62].

En tangent som följer den brantaste lutningen på stegsvaret ritas in i figuren, se figur 5.2. Hastighetsförstärkningen K_v ges av riktningskoefficienten på tangenten, dvs. $K_v = \dfrac{\Delta y}{L}$ eller $K_v = \dfrac{\Delta y_{\tan}}{\Delta t_{\tan}}$. Dödtiden L ges i punkten där tangenten skär det stabila mätvärdet innan styrsignalsteget genomfördes [5, sid. 49]. Observera att den statiska förstärkningen K_p inte kan mätas med hjälp av ett stegsvar från en process som är av typen integrerande.

Ett exempel på en integrerande process är en nivåreglering i en tank. Om tanken fylls på med lika mycket som det tappas ut kommer nivån att vara konstant. Fylls det på mer i tanken än vad som tappas ut kommer nivån att stiga. Om det fylls på mindre i tanken än vad som tappas ut kommer nivån att sjunka.

5.2.3 Dubbelpulsanalys

En variant på stegsvarsanalysen är att utsätta processen med en dubbelpuls vilket illustreras i figur 5.3.



Figur 5.3 Dubbelpulsanalys [5, sid. 50].

Pulsens amplitud u_a väljes så att den är betydligt större än bruset. Pulsvidden T_p väljs lite längre än processens tidsfördröjning. Mätvärdena y_{max}, y_{min}, och tidpunkterna t_{max} och t_{min} avläses, se figur 5.3. En fördel med metoden är att mätvärdet återgår till sitt ursprungliga värde, en annan att tidsåtgången är kortare än stegsvarsanalysen eftersom man inte behöver vänta på att mätvärdet ska stabiliseras. Nackdelarna med metoden är att det är svårt att fastställa min och max noggrant och att uppskattningen av den statiska förstärkningen K_p är dålig. Tiden T_{63} definieras som $T_{63} = T + L$.

Med hjälp av nedanstående ekvationer kan den statiska förstärkningen K_p, tidskonstanten T och dödtiden L beräknas. Dödtiden L kan beräknas på två sätt enligt nedan. Detta kan användas för att avgöra om dubbelpulsanalysen är tillämpbar på processen som det experimenteras med [5, sid. 50]

$$K_p = -\frac{y^2{}_{\max}}{u_a y_{\min}}$$

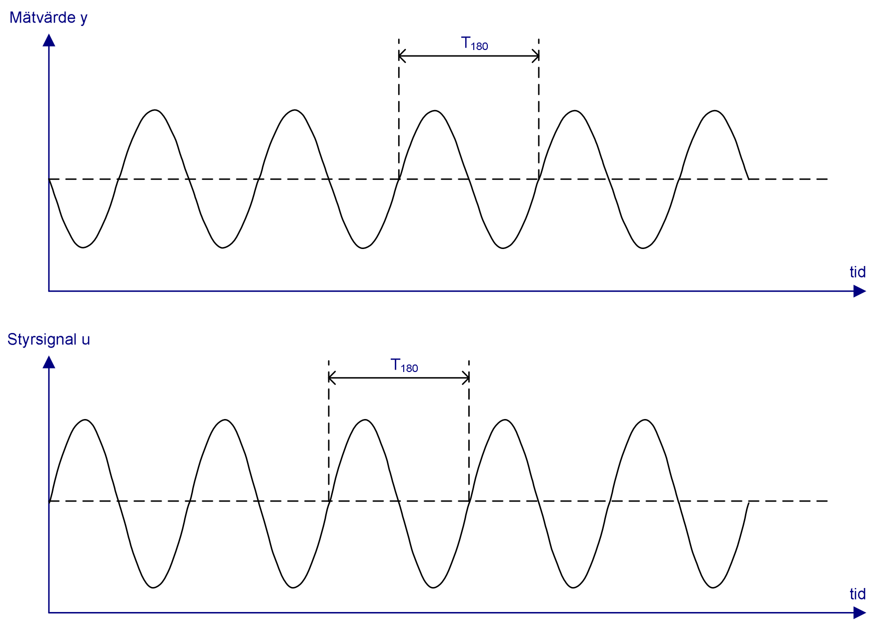
$$T = -\frac{T_p}{\log(1 + y_{\max}/y_{\min})}$$

$$L = t_{\max} - T_p$$

$$L = t_{\min} - 2T_p$$

5.2.4 Frekvenssvarsanalys genom självsvängning

Vid frekvenssvarsanalys genom självsvängning låter man en P-regulator styra processen. Förstärkningen K på denna regulator ökas tills reglerkretsen börjar självsvänga med den ultimata frekvensen, se figur 5.4.



Figur 5.4 Frekvenssvarsanalys genom självsvängning.

Vid den ultimata frekvensen där styrsignalen och mätsignalen är fasvridna 180° avläses periodtiden T_{180} och regulatorns förstärkning K. Regulatorförstärkningen vid den ultimata frekvensen kallas den ultimata förstärkningen och betecknas K_u. Processförstärkningen K_{180} vid den ultimata frekvensen kan beräknas enligt

$$K_{180} = \frac{1}{K_u}$$

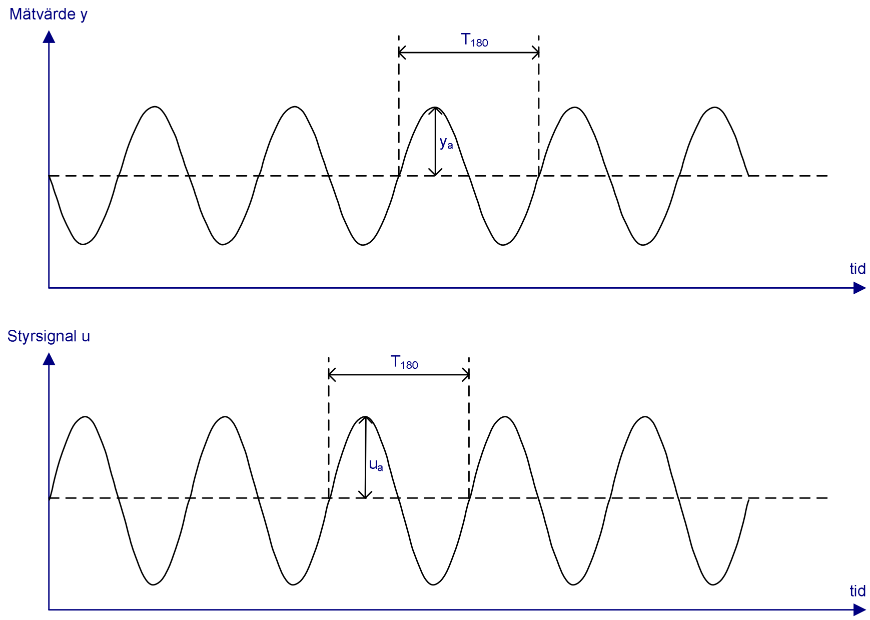
Den ultimata frekvensen när fasförskjutningen är 180° betecknas ω_{180} (grekiska bokstaven omega), mäts i radianer per sekund och ges av

$$\omega_{180} = \frac{2\pi}{T_{180}}$$

En nackdel med metoden är att den balanserar på stabilitetsgränsen. En annan stor nackdel är att processen påverkas mycket av försöket [5, sid. 26][5, sid. 52][5, sid. 161][2, sid. 66].

5.2.5 Frekvenssvarsanalys med sinussignal

Vid frekvenssvarsanalys med sinussignal låter man en sinussignal styra styrsignalen u, se figur 5.5.



Figur 5.5 Frekvenssvarsanalys med sinussignal.

Vid den ultimata frekvensen där styrsignalen och mätsignalen är fasvridna 180° avläses periodtiden T_{180}, mätsignalens amplitud y_a och styrsignalens amplitud u_a, se figur 5.5. Processförstärkningen K_{180} vid den ultimata frekvensen kan beräknas enligt

$$K_{180} = \frac{y_a}{u_a}$$

Den ultimata frekvensen när fasförskjutningen är 180° betecknas ω_{180} (grekiska bokstaven omega), mäts i radianer och ges av

$$\omega_{180} = \frac{2\pi}{T_{180}}$$

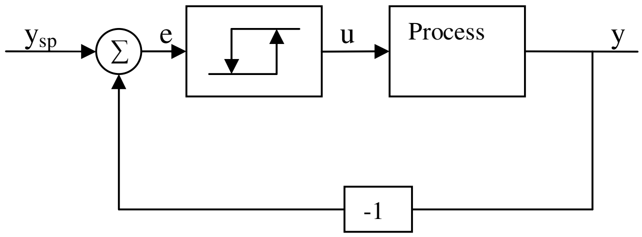
Regulatorförstärkningen vid den ultimata frekvensen kallas den ultimata förstärkningen och betecknas K_u. Den ultimata förstärkningen K_u beräknas enligt

$$K_u = \frac{1}{K_{180}}$$

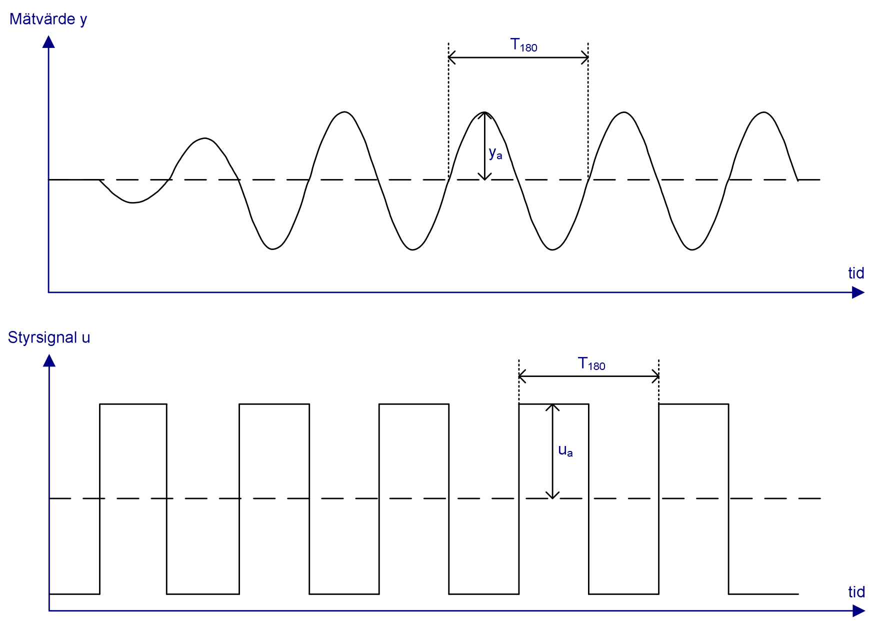
En nackdel med metoden är att man måste prova flera olika frekvenser för att hitta den ultimata frekvensen [5, sid. 21][5, sid. 52][5, sid. 161][2, sid. 17].

5.2.6 Frekvenssvarsanalys med relämetoden

Relämetoden innebär att processen regleras med en on-off regulator med hysteres, se figur 5.6 [2, sid. 124].



Figur 5.6 Principen för relämetoden [5, sid. 53].



Figur 5.7 Processanalys med relämetoden [5, sid. 53].

On-off regleringen genererar en styrsignal u som kommer att skifta mellan två nivåer och mätvärdet y kommer att svänga runt börvärdet, se figur 5.7. Svängningen har ungefär samma frekvens som i frekvenssvarsanalysen genom självsvängning, dvs. den ultimata frekvensen där styrsignalen och mätsignalen är fasvridna 180°. Periodtiden T_{180}, mätsignalens amplitud y_a och styrsignalens amplitud u_a vid den ultimata frekvensen kan utläsas enligt figur 5.7. Processförstärkningen K_{180} vid den ultimata frekvensen kan beräknas enligt

$$K_{180} = \frac{\pi y_a}{4 u_a}$$

Den ultimata frekvensen när fasförskjutningen är 180° betecknas ω_{180} (grekiska bokstaven omega), mäts i radianer och ges av

$$\omega_{180} = \frac{2\pi}{T_{180}}$$

Regulatorförstärkningen vid den ultimata frekvensen kallas den ultimata förstärkningen och betecknas K_u. Den ultimata förstärkningen K_u beräknas enligt

$$K_u = \frac{1}{K_{180}}$$

Relämetoden har två stora fördelar jämfört med frekvenssvarsanalys genom självsvängning. För det första balanserar man inte på stabilitetsgränsen eftersom on-off regulatorn reglerar processen. För det andra kan amplituden på styrsignalen väljas så att processens inte störs så mycket. Detta är en stor fördel eftersom regleravvikelsen från det önskade börvärdet kan minskas. Andra fördelar är att det är lätt att automatisera experimentet och att tidsåtgången är kort [2, sid. 124][5, sid. 53][5, sid. 161].

5.2.7 Kombinerad stegsvarsanalys och frekvenssvarsanalys

Resultaten från en stegsvarsanalys och en frekvensanalys kan kombineras. Indata till beräkningen från en stegsvarsanalys är den statiska förstärkningen K_p, tidskonstanten T och dödtiden L. Data från en frekvensanalys är periodtiden T_{180} och processförstärkningen K_{180}. Med hjälp av nedanstående ekvationer beräknas T1, T2 och L1 som sedan används i AMIGO-metoden för att hitta lämpliga regulatorparametrar [5, sid. 54]. Se även MATLAB-kod för att lösa dessa ekvationer i figur 5.12. Eftersom värdet på den statiska förstärkningen K_p krävs, kan inte regulatorparametrar till processer som är av typen integrerande, beräknas med nedanstående ekvationer.

Först beräknas tiden T_{63}, den ultimata frekvensen ω_{180} i radianer och den ultimata förstärkningen K_u.

$$T_{63} = T + L$$

$$\omega_{180} = \frac{2\pi}{T_{180}}$$

$$K_u = \frac{1}{K_{180}}$$

Enligt [5, sid. 51] kan det antas att $T_2 \leq T_1$. Om $T_2 = T_1$ är $\alpha = 1$ eftersom $\alpha = T_2 / T_1$. Om både T_2 och T_1 är positiva tal och $T_2 < T_1$ är $\alpha < 1$, men samtidigt gäller det att $\alpha \geq 0$. Alltså kan α variera mellan 0 och 1.

Variablerna e_1, e_2, och e_3 i ekvationerna nedan är lika med noll när ett perfekt värde på α hittats. För att väga samman e_1, e_2, och e_3 till ett värde kan ekvationen $e = \sqrt{e_1^2 + e_2^2 + e_3^2}$ användas. För att lösa ekvationerna nedan kan ett antal värden på α mellan 0 och 1 provas. När e är som minst har ett lämpligt värde på α hittats och därmed även T1, T2 och L1.

$$T_1 = \frac{1}{\left(\alpha\, \omega_{180} \sqrt{2}\right)} * \sqrt{\left(\sqrt{4\alpha^2 K_p^2 K_u^2 + \left(1-\alpha^2\right)^2}\right) - 1 - \alpha^2}$$

$$T_2 = \alpha\, T_1$$

$$L_1 = \frac{\left(\pi - \arctan(\omega_{180} T_1) - \arctan(\omega_{180} T_2)\right)}{\omega_{180}}$$

$$e_1 = 0{,}37 - \frac{T_1}{(T_1 - T_2)} e^{-(T_{63} - L_1)/T_1} - \frac{T_2}{(T_2 - T_1)} e^{-(T_{63} - L_1)/T_2} \text{ används om } \alpha \neq 1$$

$$e_1 = 1 - e^{-(T_{63} - L_1)/T_1} - \frac{T_{63}}{T_1} e^{-(T_{63} - L_1)/T_1} - 0{,}63 \text{ används om } \alpha = 1$$

$$e_2 = \left(1 + \omega_{180}^2\ T_1^2\right)\left(1 + \omega_{180}^2\ T_2^2\right) - K_p^2\ K_u^2$$

$$e_3 = \arctan(\omega_{180} T_1) + \arctan(\omega_{180} T_2) + \omega_{180}\, L_1 - \pi$$

5.3 AMIGO-metoden

AMIGO (Approximate M-constrained integral gain optimization) metoden är utvecklad som en ersättare till Ziegler-Nichols metoder från 1940 talet. Enligt [5, sid. 225] har Ziegler-Nichols metoder stora brister. Med hjälp av AMIGO-metoden går det att hitta lämpliga regulatorparametrar med stegsvarsanalys och/eller frekvensanalys som indata. Målet med utvecklingen av AMIGO har varit att hitta en metod som kan användas för både manuell inställning och automatisk inställning av regulatorparametrarna K, T_i, T_d och b. Metoden har utvecklats genom att först applicera tekniken för ”Robust loop shaping” och MIGO (M-constrained integral gain optimization) metoden på 134 testprocesser som är representativa för processindustrin. Optimala reglerparametrarna har tagits fram med hjälp av MIGO-metoden. Förhållandet mellan de optimala reglerparametrarna och data från stegsvarsanalyser och frekvensanalyser har analyserats och resultatet blev AMIGO-metoden.

AMIGO maximerar integralförstärkningen $k_i = K / T_i$ samtidigt som reglerparametrarna väljs så att regleringen blir robust, dvs. okänslig för processvariationer. Börvärdesföljning hanteras genom att använda börvärdesviktning. För att regleringen ska vara robust ger AMIGO även regler för hur regulatorparametrarna ska modifieras om mätsignalen innehåller brus som inte kan filtreras bort, för mera information om detta se avsnittet ”7.9 Detuning” i [5] på sidan 253.

För att grovt karakterisera processers dynamik används måttet relativ tidsfördröjning $\tau = L / (L+T)$ (grekiska bokstaven tau) som antar värden mellan 0 och 1. Processer med litet värde på τ kallas eftersläpnings-dominerande (lag dominated), processer med τ nära 1 kallas dödtids-dominerande (delay dominated) och processer med τ runt 0,5 benämns balanserad (balanced)[5, sid. 225][5, sid. 262].

Sammanfattningsvis gäller alltså följande, T > L för processer som är av typen eftersläpnings-dominerande, L > T för dödtids-dominerande, T = L för balanserad. För en beskrivning av dödtiden L och tidskonstanten T se avsnitt 5.2.1.

5.3.1 PI eller PID-regulator?

Analyser av regulatorparametrarna tillhörande de 134 processerna visar att derivatadelen bara ger en liten förbättring om processen är av typen balanserad eller dödtids-dominerande. Däremot ger derivatadelen stora förbättringar för processer av typen eftersläpnings-dominerande [5, sid. 226]. Anledningen till att derivatadelen ger bättre regulatorprestanda är att förstärkningen K och integralförstärkningen k_i kan ökas [2, sid. 35]. Genom att jämföra integralförstärkningen som ges av $k_i = K / T_i$ mellan en PI och en PID-regulator kan nyttan av derivatadelen bedömas [5, sid. 264].

I förstahand bör en PI-regulator väljas eftersom en PID-regulator är känslig för mätvärdesbrus [1, sid. 63].

5.3.2 Justering för derivatafiltrets dynamik

Derivatafiltret reducerar en PID-regulators robusthet men det går att kompensera för filtrets dynamik. AMIGO-metoden ger följande regler på hur data från processanalyserna ska modifieras för att kompensera för filtret.

Data från stegsvarsanalys ska justeras enligt

$K_{p\ \text{med filter}} = K_{p\ \text{utan filter}}$ (ingen ändring)

$T_{\text{med filter}} = T_{\text{utan filter}} + T_f / 2$

$L_{\text{med filter}} = L_{\text{utan filter}} + T_f / 2$

Data från en kombinerad stegsvarsanalys och frekvensanalys ska justeras enligt

$T_{1\ \text{med filter}} = T_{1\ \text{utan filter}}$ (ingen ändring)

$T_{2\ \text{med filter}} = T_{2\ \text{utan filter}} + T_f / 2$

$L_{1\ \text{med filter}} = L_{1\ \text{utan filter}} + T_f / 2$

Filtret dämpar mätvärdesbrus på bekostnad av sämre dämpning av lastörningar. Anledningen till detta är att reglerparametrarna måste justeras så att regulatorn blir långsammare för att förbli robust även med ett filter [5, sid. 251][5, sid. 265].

5.3.3 PI parametrar, stegsvarsanalys

AMIGO ekvationerna nedan ger parametrarna K och Ti som överensstämmer mycket bra för alla testprocesser med de värden som MIGO-metoden ger. Indata till ekvationerna är data från en stegsvarsanalys. Parametrar för processer som inte är integrerande ges av [5, sid. 228]

$$K = \frac{0{,}15}{K_p} + \left(0{,}35 - \frac{L\,T}{(L+T)^2}\right)\frac{T}{(K_p L)}$$

$$T_i = 0{,}35\,L + \frac{13\,L\,T^2}{\left(T^2 + 12\,L\,T + 7\,L^2\right)}$$

Parametrar för processer som är integrerande ges av [5, sid. 229]

$$K = \frac{0{,}35}{K_v L}$$

$$T_i = 13{,}4\,L$$

Börvärdesviktningsparametern b är beroende på processens relativatidsfördröjning τ enligt [5, sid. 230].

b = 0 för $\tau \le 0{,}3$

b = 1 för $\tau > 0{,}3$

5.3.4 PID parametrar, stegsvarsanalys

Indata till ekvationerna är data från en stegsvarsanalys. Parametrar för processer som inte är integrerande ges av [5, sid. 233]

$$K = \frac{1}{K_p}\left(0{,}2 + 0{,}45\frac{T}{L}\right)$$

$$T_i = \frac{(0{,}4\ L + 0{,}8\ T)}{(L + 0{,}1\ T)}L$$

$$T_d = \frac{0{,}5\ L\ T}{0{,}3\ L + T}$$

För processer där $\tau > 0{,}3$ ger AMIGO ekvationen ovan ett K som följer det K som MIGO ger bra. För mindre värden på τ underskattar AMIGO K för nästan alla testprocesser.

Integreringstiden beskrivs bra för processer där $\tau > 0{,}2$. För mindre värden på τ överskattar AMIGO T_i för nästan alla processer jämfört med MIGO.

Derivatatiden T_d beskrivs bra för processer där $\tau > 0{,}5$. I intervallet $0{,}3 < \tau < 0{,}5$ kan AMIGO derivatatiden vara halva värdet från MIGO. Om värden på derivatatiden används inom intervallet $0{,}3 < \tau < 0{,}5$ försämras robustheten. För värden $\tau < 0{,}3$ ger AMIGO en derivatatid som ibland är mindre än och ibland större än vad MIGO ger [5, sid. 233].

Ekvationerna ger konservativa parametrar om $\tau < 0{,}5$, tolkning av [5, sid. 231]. Anledningen till de konservativa parametrarna är framför allt minskad förstärkning K och ökad integreringstid T_i [5, 233]. Ekvationerna kan användas även för eftersläpnings-dominerande processer under förutsättningen att konservativa parametrar kan accepteras [5, 264].

En observation som författaren av denna rapport gör, är att ekvationerna ger bra resultat för processer av typen balanserad eller dödtids-dominerande vilka inte har så stor nytta av derivatadelen, och mindre bra värden för eftersläpnings-dominerande processer som har stor nytta av derivatadelen.

Parametrar för processer som är integrerande ges av

$$K = 0{,}45 / K_v$$

$$T_i = 8\ L$$

$$T_d = 0{,}5\ L$$

Sambandet mellan b och τ för de olika processerna är inte så bra, men en konservativ regel är att välja börvärdesviktningsparametern b enligt [5, sid. 235]

b = 0 för $\tau \leq 0{,}5$

b = 1 för $\tau > 0{,}5$

5.3.5 PI parametrar, frekvensanalys

Ekvationerna nedan ger parametrar som är hyfsat nära de optimala MIGO parametrarna, för processer där förstärkningsförhållandet κ > 0,2 (grekiska bokstaven kappa). Förstärkningsförhållandet ges av $\kappa = K_{180} / K_p$. Ekvationerna kan användas på processer som är av typen balanserad eller dödtids-dominerande, men är olämpliga till processer som är av typen eftersläpnings-dominerande. Indata till ekvationerna är data från en frekvenssvarsanalys för mätvärden på processförstärkningen K_{180} och periodtiden T_{180} [5, sid. 239].

$$K = 0{,}16 / K_{180}$$

$$T_i = \frac{1}{(1 + 4{,}5\,\kappa)} T_{180}$$

Eftersom data på den statiska förstärkningen K_p krävs måste även en stegsvarsanalys utförs för att få ett mätvärde på K_p [9]. Eftersom värdet på den statiska förstärkningen K_p krävs, kan inte regulatorparametrar till processer som är av typen integrerande, beräknas med ovanstående ekvationer.

5.3.6 PID parametrar, frekvensanalys

Ekvationerna nedan ger parametrar som är hyfsat nära de optimala MIGO parametrarna, för processer där förstärkningsförhållandet κ > 0,2 (grekiska bokstaven kappa). Förstärkningsförhållandet ges av $\kappa = K_{180} / K_p$. Ekvationerna kan användas på processer som är av typen balanserad eller dödtids-dominerande, men är olämpliga till processer som är av typen eftersläpnings-dominerande. Indata till ekvationerna är data från en frekvenssvarsanalys för mätvärden på processförstärkningen K_{180} och periodtiden T_{180} [5, sid. 241].

$$K = \frac{(0{,}3 - 0{,}1\,\kappa^4)}{K_{180}}$$

$$T_i = \frac{0{,}6}{(1 + 2\,\kappa)} T_{180}$$

$$T_d = \frac{0{,}15\,(1 - \kappa)}{(1 - 0{,}95\,\kappa)} T_{180}$$

Eftersom data på den statiska förstärkningen K_p krävs måste även en stegsvarsanalys utförs för att få ett mätvärde på K_p [9]. Eftersom värdet på den statiska förstärkningen K_p krävs, kan inte regulatorparametrar till processer som är av typen integrerande, beräknas med ovanstående ekvationer.

En observation som författaren av denna rapport gör, är att ekvationerna är användbara för processen av typen balanserad eller dödtids-dominerande vilka inte har så stor nytta av derivatadelen, och inte användbara för eftersläpnings-dominerande processer som har stor nytta av derivatadelen.

5.3.7 PID parametrar, stegsvarsanalys och frekvensanalys

5.3.7.1 Processer som är stabila

Ekvationerna nedan ger liknande värden på processer av typen balanserad eller dödtids-dominerande som i ekvationerna från avsnitten 5.3.4 och 5.3.6, men för eftersläpnings-dominerande fås betydligt bättre parametrar. Parametrarna K_p, T_1, T_2 och L_1 är indata till ekvationerna, dessa parametrar är resultatet från en kombinerad stegsvarsanalys och frekvenssvarsanalys, se avsnitt 5.2.7.

Eftersom värdet på den statiska förstärkningen K_p krävs i avsnitt 5.2.7, kan inte regulatorparametrar till processer som är av typen integrerande, beräknas med nedanstående ekvationer.

Integralförstärkningen ges av $k_i = K / T_i$ och derivataförstärkningen ges av $k_d = K\ T_d$. Regulatorparametrar för processer som inte är integrerande ges av [5, sid. 245]

$$K = \frac{\left(\alpha_1 + \alpha_2 \frac{T_1}{L_1} + \alpha_3 \frac{T_2}{L_1} + \alpha_4 \frac{T_1\ T_2}{L_1^2}\right)}{K_p}$$

$$k_i = \frac{\left(\beta_1 \frac{1}{L_1} + \beta_2 \frac{T_1}{L_1^2} + \beta_3 \frac{T_2}{L_1^2} + \beta_4 \frac{T_1\ T_2}{L_1^3}\right)}{K_p}$$

$$k_d = \frac{\left(\left(\gamma_1 L_1 + \gamma_2 T_1 + \gamma_3 T_2 + \gamma_4 \frac{T_1\ T_2}{L_1}\right)\frac{(T_1 + T_2)}{(T_1 + T_2 + L_1)}\right)}{K_p}$$

Parametrar till ekvationerna ovan

$\alpha_1 = 0{,}19$	$\alpha_2 = 0{,}37$	$\alpha_3 = 0{,}18$	$\alpha_4 = 0{,}02$
$\beta_1 = 0{,}48$	$\beta_2 = 0{,}03$	$\beta_3 = -0{,}0007$	$\beta_4 = 0{,}0012$
$\gamma_1 = 0{,}29$	$\gamma_2 = 0{,}16$	$\gamma_3 = 0{,}20$	$\gamma_4 = 0{,}28$

De grekiska bokstäverna är α = alpha, β = beta, och γ = gamma.

Integrationstiden ges av

$$T_i = \frac{K}{k_i}$$

Derivatatiden ges av

$$T_d = \frac{k_d}{K}$$

I exemplen [5, sid. 247 – 250] anges ett värde på b för PID AMIGO step+frequency, men det framgår inte klart när b ska var 0 respektive 1. Enligt [9] ska samma regler som för PID AMIGO step användas enligt nedan.

b = 0 för $\tau \le 0{,}5$

b = 1 för $\tau > 0{,}5$

5.3.7.2 Processer som är av typen integrerande

Enligt [9] har författarna av [5] inte funderat närmare på hur man experimentellt ska ta fram parametrarna K_v, T2 och L1 som är indata till ekvationerna nedan.

Ett sätt enligt [9] är att generera ett pulssvar, d.v.s. att skicka in en kort puls med varaktigheten t och höjden 1/t, enligt samma principer som i avsnitt 5.2.1. Ju kortare t desto bättre. För att pulsen ska kunna betraktas som en puls måste den vara betydligt kortare är det L och T man skattar i modellen. Det finns ingen skarp gräns, men om t är mindre än en tiondel av L och T bör det fungera bra. Detta innebär att svaret ser ut som ett stegsvar från en stabil process. Ur detta kan de tre parametrarna bestämmas enligt $K_v=K_p$, T2=T, L1=L.

Om K_v och L skattas med en stegsvarsanalys enligt avsnitt 5.2.2, fås en modell där L sammanfattar både den äkta dödtiden och tidskonstanten, i en stegsvarsanalys enligt avsnitt 5.2.1. Relationen är alltså L(stegsvar) = L(pulssvar) + T(pulssvar) [9].

Enligt [9] går det eventuellt att använda dubbelpulsanalysen från avsnitt 5.2.3, men det måste utredas noggrannare.

Integralförstärkningen ges av $k_i = K / T_i$ och derivataförstärkningen ges av $k_d = K\,T_d$. Regulatorparametrar för processer som är integrerande ges av [5, sid. 246]

$$K = \frac{\left(\alpha_2 \frac{1}{L_1} + \alpha_4 \frac{T_2}{L_1^2}\right)}{K_v}$$

$$k_i = \frac{\left(\beta_2 \frac{1}{L_1^2} + \beta_4 \frac{T_2}{L_1^3}\right)}{K_v}$$

$$k_d = \frac{\left(\gamma_2 + \gamma_4 \frac{T_2}{L_1}\right)}{K_v}$$

Integrationstiden ges av

$$T_i = \frac{K}{k_i}$$

Derivatatiden ges av

$$T_d = \frac{k_d}{K}$$

Parametrar till ekvationerna ovan

$\alpha_1 = 0{,}19$	$\alpha_2 = 0{,}37$	$\alpha_3 = 0{,}18$	$\alpha_4 = 0{,}02$
$\beta_1 = 0{,}48$	$\beta_2 = 0{,}03$	$\beta_3 = -0{,}0007$	$\beta_4 = 0{,}0012$
$\gamma_1 = 0{,}29$	$\gamma_2 = 0{,}16$	$\gamma_3 = 0{,}20$	$\gamma_4 = 0{,}28$

De grekiska bokstäverna är α = alpha, β = beta, och γ = gamma.

I exemplen [5, sid. 247 – 250] anges ett värde på b för PID AMIGO step+frequency, men det framgår inte klart när b ska var 0 respektive 1. Enligt [9] ska samma regler som för PID AMIGO step användas enligt nedan.

b = 0 för $\tau \le 0{,}5$

b = 1 för $\tau > 0{,}5$

5.4 AMIGO exempel

I detta avsnitt ges ett exempel hur regulatorparametrar kan väljas med hjälp av AMIGO-metoden. Exemplet är hämtat ifrån [5, sid. 247]. MATLAB-koderna är skrivna av författaren till denna rapport.

5.4.1 Exempel på en eftersläpnings-dominerande process

Mätdata från en stegsvarsanalys på en eftersläpnings-dominerande process som inte är integrerande visar att dödtiden L = 0,075, tidskonstanten T = 1,04 och den statiska förstärkningen K_p = 1. En frekvensanalys visar att processförstärkningen K_{180} = 0,0091 och periodtiden T_{180} = 0,199.

5.4.1.1 PI - stegsvarsanalys

MATLAB-kod som beräknar PI parametrar med indata från en stegsvarsanalys ges i figur 5.8.

```
format short;
L=0.075;
T=1.04;
Kp=1;
K=0.15/Kp+(0.35-L*T/(L+T)^2)*T/(Kp*L)
Ti=0.35*L+13*L*T^2/(T^2+12*L*T+7*L^2)
tau=L/(L+T);
if tau > 0.3
  b=1
else
  b=0
end
ki=K/Ti

K = 4.1333
Ti = 0.5389
b = 0
ki = 7.6696
```

Figur 5.8 MATLAB-kod baserad på ekvationer från avsnitt 5.3.3.

5.4.1.2 PID - stegsvarsanalys

MATLAB-kod som beräknar PID parametrar med indata från en stegsvarsanalys ges i figur 5.9.

```
format short;
L=0.075;
T=1.04;
Kp=1;
K=1/Kp*(0.2+0.45*T/L)
Ti=(0.4*L+0.8*T)/(L+0.1*T)*L
Td=0.5*L*T/(0.3*L+T)
tau=L/(L+T);
if tau > 0.5
  b=1
else
  b=0
end
ki=K/Ti
if tau < 0.5
  info = 'Warning conservative parameters'
end

K = 6.4400
Ti = 0.3612
Td = 0.0367
b = 0
ki = 17.8308
info = Warning conservative parameters
```

Figur 5.9 MATLAB-kod baserad på ekvationer från avsnitt 5.3.4.

5.4.1.3 PI - frekvensanalys

MATLAB-kod som beräknar PID parametrar med indata från en frekvensanalys ges i figur 5.10.

```
format short;
K180=0.0091;
T180=0.199;
Kp=1;
kappa=K180/Kp
K=0.16/K180
Ti=1/(1+4.5*kappa)*T180
b='?'
ki=K/Ti
if kappa <= 0.2
  info = 'Warning parameters are not appropriate'
end

kappa = 0.0091
K = 17.5824
Ti = 0.1912
b = ?
ki = 91.9719
info = Warning parameters are not appropriate
```

Figur 5.10 MATLAB-kod baserad på ekvationer från avsnitt 5.3.5.

5.4.1.4 PID - frekvensanalys

MATLAB-kod som beräknar PID parametrar med indata från en frekvensanalys ges i figur 5.11.

```
format short;
K180=0.0091;
T180=0.199;
Kp=1;
kappa=K180/Kp;
K=(0.3-0.1*kappa^4)/K180
Ti=0.6/(1+2*kappa)*T180
Td=0.15*(1-kappa)/(1-0.95*kappa)*T180
b='?'
ki=K/Ti
if kappa <= 0.2
  info = 'Warning parameters are not appropriate'
end

K = 32.9670
Ti = 0.1173
Td = 0.0298
b = ?
ki = 281.1309
info = Warning parameters are not appropriate
```

Figur 5.11 MATLAB-kod baserad på ekvationer från avsnitt 5.3.6.

5.4.1.5 PID – kombinerad stegsvarsanalys och frekvensanalys

Parametrar från en kombinerad stegsvarsanalys och frekvensanalys beräknas i två steg.

5.4.1.5.1 T_1, T_2 och L_1

MATLAB-koden i figur 5.12 beräknar T_1, T_2 och L_1, som sedan används i AMIGO-metoden för att beräkna lämpliga regulatorparametrar. Indata till beräkningen är den statiska förstärkningen K_p, tidskonstanten T, dödtiden L, periodtiden T_{180} och processförstärkningen K_{180}.

```
format short;
Kp=1;
T=1.04;
L=0.075;
T63=T+L;
T180=0.199;
K180=0.0091;
omega180=2*pi/T180;
Ku=1/K180;
step=0.00001;
startAlpha=0;
endAlpha=1;
alpha=startAlpha+step;
bestE=100000000;
bestT1=0;
bestT2=0;
bestL1=0;

while alpha <= endAlpha
  T1=1/(alpha*omega180*sqrt(2))*sqrt((sqrt(4*alpha^2*Kp^2*Ku^2+(1-
alpha^2)^2))-1-alpha^2);
  T2=alpha*T1;
  L1=(pi-atan(omega180*T1)-atan(omega180*T2))/omega180;
  if alpha~=1
    e1=0.37-T1/(T1-T2)*exp(-(T63-L1)/T1)-T2/(T2-T1)*exp(-(T63-
L1)/T2);
  else
    e1=1-exp(-(T63-L1)/T1)-T63/T1*exp(-(T63-L1)/T1)-0.63;
  end
  e2=(1+omega180^2*T1^2)*(1+omega180^2*T2^2)-Kp^2*Ku^2;
  e3=atan(omega180*T1)+atan(omega180*T2)+omega180*L1-pi;
  e=sqrt(e1^2+e2^2+e3^2);
  if e < bestE
    bestE=e;
    bestT1=T1;
    bestT2=T2;
    bestL1=L1;
  end
  alpha=alpha+step;
end

bestE, bestT1, bestT2, bestL1
bestE = 1.5161e-007
bestT1 = 0.9988
bestT2 = 0.1057
bestL1 = 0.0102
```

Figur 5.12 MATLAB-kod baserad på ekvationer från avsnitt 5.2.7.

5.4.1.5.2 Regulatorparametrarna

MATLAB-kod som beräknar PID parametrar med indata från en kombinerad stegsvarsanalys och frekvensanalys ges i figur 5.13.

```
format short;
T1=0.9988;
T2=0.1057;
L1=0.0102;
Kp=1;
L=0.075;
T=1.04;
alpha1=0.19;
alpha2=0.37;
alpha3=0.18;
alpha4=0.02;
beta1=0.48;
beta2=0.03;
beta3=-0.0007;
beta4=0.0012;
gamma1=0.29;
gamma2=0.16;
gamma3=0.20;
gamma4=0.28;
K=(alpha1+alpha2*T1/L1+alpha3*T2/L1+alpha4*T1*T2/L1^2)/Kp
ki=(beta1*1/L1+beta2*T1/L1^2+beta3*T2/L1^2+beta4*T1*T2/L1^3)/K
p;
Ti=K/ki
kd=((gamma1*L1+gamma2*T1+gamma3*T2+gamma4*T1*T2/L1)*(T1+T2)/(T
1+T2+L1))/Kp;
Td=kd/K
tau=L/(L+T);
if tau > 0.5
  b=1
else
  b=0
end
ki

K = 58.5810
Ti = 0.1291
Td = 0.0521
b = 0
ki = 453.7330
```

Figur 5.13 MATLAB-kod baserad på ekvationer från avsnitt 5.3.7.

5.4.1.6 Sammanställning av exemplet

I tabell 5.2 sammanställs parametrarna från beräkningarna utförda med MATLAB-koden som pressenterades i avsnitt 5.4.1. Beräkningar från avsnitten 5.4.1.3 och 5.4.1.4 är inte införda i tabell 5.2, eftersom $\kappa \leq 0{,}2$ vilket medför att värdena inte är användbara.

Metod	K	T_i	T_d	b	k_i
PI MIGO	3,56	0,660		0	5,39
PI AMIGO stegsvarsanalys	4,13	0,539		0	7,67
PID MIGO	56,9	0,115	0,0605	0	495
PID AMIGO stegsvarsanalys	6,44	0,361	0,0367	0	17,8
PID AMIGO stegsvarsanalys + frekvenssvarsanalys	58.6	0,129	0,0521	0	454

Tabell 5.2 Regulatorparametrar från exemplet.

Värden från PI AMIGO stegsvarsanalys är lika värdena från PI MIGO. Eftersom processen är av typen eftersläpnings-dominerande fås konservativa parametrar för PID AMIGO stegsvarsanalys som är långt ifrån de optimala PID MIGO värdena. Värdena från PID AMIGO stegsvarsanalys + frekvenssvarsanalys stämmer bra med PID MIGO.
Eftersom processen i exemplet från avsnitt 5.4.1 är eftersläpnings-dominerande, är nyttan med derivatadelen stor eftersom k_i värdet kan maximeras.

6 Slutsatser

Målet med denna rapport var att svara på frågorna:

- Hur fungerar en PID-regulator?
- Hur kan en PID-regulator implementeras?
- Hur trimmas en PID-regulator med AMIGO-metoden?

Författaren av denna rapport anser att svaret på frågorna har besvarats i denna rapport. Vidare borde det vara möjligt för t.ex. en Programvaruteknik-student att förstå, designa, implementera och ställa in en PID-regulator med hjälp av denna rapport.

Eftersom PID-regulatorer är mycket vanliga är det viktigt att det finns tillförlitliga metoder för att välja optimala regulatorparametrar baserat på enkla experiment på processen. Forskarna Åström, Hägglund och deras studenters arbete med AMIGO-metoden är ett stort framsteg eftersom Ziegler-Nichols metoder har stora brister enligt [5, sid. 225]. Med hjälp av AMIGO-metoden går det att manuell eller automatisk att hitta lämpliga värden på regulatorparametrarna K, T_i, T_d, och b.

7 Referenser

[1] Krister Forsman, Reglerteknik för processindustrin, Studentlitteratur, 2005

[2] Tore Hägglund, Praktisk processreglering, Studentlitteratur, 1997

[3] Bertil Thomas, Modern Reglerteknik, Liber, 1992

[4] Holmström och Smedhamre, Komvux matematik etapp III, Almqvist & Wiksell Förlag, 1992

[5] Karl J Åström and Tore Hägglund, Advanced PID control, ISA, 2006

[6] Gustaf Olsson and Gianguido Piani, Computer systems for automation and control, Prentice Hall, 1992

[7] K. J. Åström, Lecture 9 - PID Control, October 2002

[8] Ola Dahl, Föreläsningsanteckningar Föreläsning 9 RT7010 – Reglerteknik, Malmö Högskola Teknik och samhälle

[9] Tore Hägglund, E-post korrespondens mellan Tore Hägglund och Jonas Wahlfrid, januari – februari 2007