

Dynamiska system och reglering

Kurskompendium

23 januari 2020



Reglerteknik, Institutionen för systemteknik,
Linköpings universitet , 581 83 Linköping
www.control.isy.liu.se

Kapitel 1

Sensorer och mätteknik

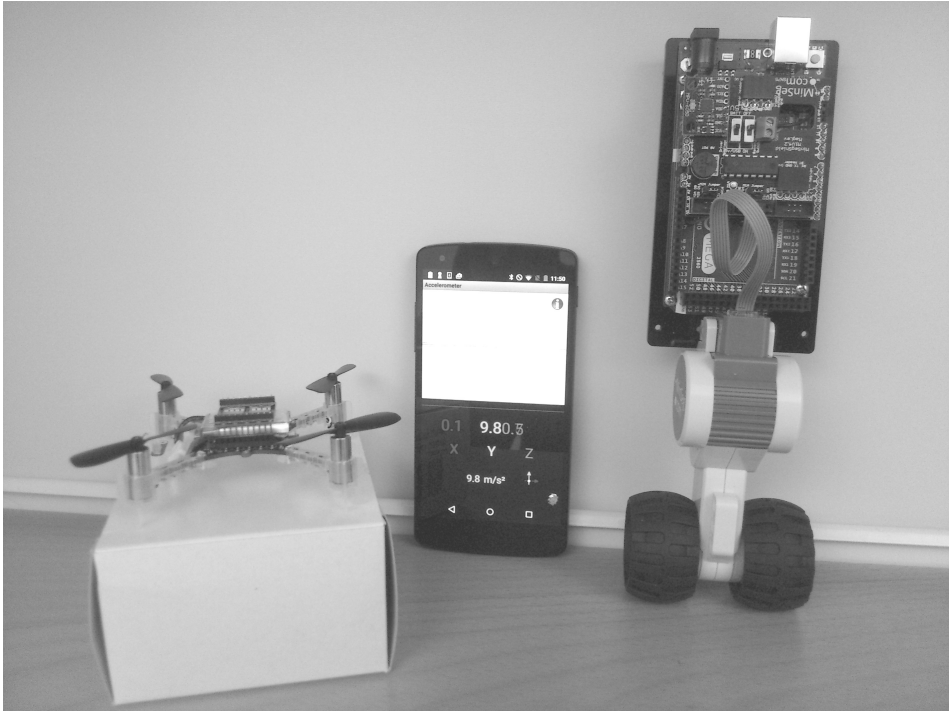
För att kunna agera i ett mekatroniskt system, eller vilket reglertekniskt system som helst, så måste vi ha tillgång till information om vad som händer. Vi behöver mätningar av fysikaliska storheter, och detta får man via sensorer. När vi designar reglertekniska system så abstraherar vi ofta bort själva mät- och sensortekniken, men det är naturligtvis en väldigt viktig aspekt i praktiken, så det kan vara bra att känna till grunderna, samt typiska sensorer som man kan stöta på.

Mättekniskt handlar det ofta om att via en enkelt uppmätt fysikalisk storhet via fysikaliska samband härleda en annan storhet som man är intresserad av. Det kan till exempel handla om att ta fram en acceleration genom att studera hur en fjäderupphängd massa rör sig, ett avstånd genom att mäta tid för att få en reflektion av en laserpuls, eller vattendjup genom att mäta spänning över en tryckkänslig resistor. Med hjälp av någon sådan slags mätteknisk princip bygger man en sensor. Värdet från denna sensor samplas av en dator, och i datorn kan man arbeta vidare med mätningarna mjukvarumässigt för att extrahera fram mer och bättre information och kanske andra fysikaliska storheter. Ytterligare ett steg är att man använder mätningar från ett flertal sensorer för att förbättra mätprestanda mjukvarumässigt genom så kallad sensorfusion.

Om du har en modern telefon kan du ladda ner en sensor-app för att testa de olika sensorer som din telefon har, och relatera detta till materialet här. Om du har en Android-telefon så kan du använda appen *LiU Sensor Fusion App* som är utvecklad på Avdelningen för Reglerteknik.

1.1 A/D-omvandlare och sampling

Innan vi diskuterar sensorer så kan det vara lämpligt att notera att allt kommer att användas tillsammans med datorer, vilket betyder att vi måste arbeta med mätningar digitalt och i diskret tid. Detta sker via så kallad *A/D-omvandling* och *sampling*.



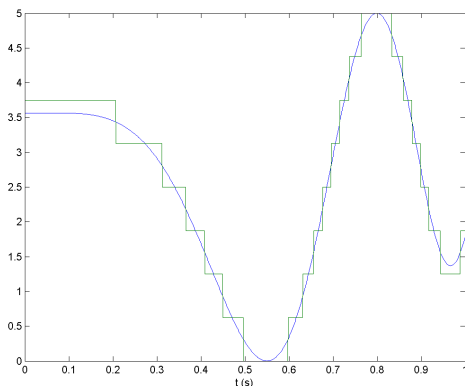
Figur 1.1: Exempel på system med sensorer. Quadcoptern till vänster (CrazyFlie) har en 3-axlig accelerometer, ett 3-axligt gyro samt en barometer. Telefonen i mitten innehåller en 3-axlig accelerometer, ett 3-axligt gyro, en magnetometer samt en GPS. Arduinodatorn till höger (som används i kursens laborationer) är utrustad med en 3-axlig accelerometer samt ett 3-axligt gyro.

En A/D-omvandlare (Analog-Digital) konverterar en analog fysisk spännings-signal till en kvantiserad digital signal som kan läsas in i en dator. Beroende på hur många bitar som används i talrepresentationen så får man olika upplösning. En binär representation med n bitar, $b_{n-1}b_{n-2} \dots b_0$, kan representera 2^n olika tal. Om A/D-omvandlaren kan hantera analoga insignaler från 0 V till U_{max} V, och man använder n bitar, så kommer man ha en upplösning på $U_{max}/2^n$ V på den kvantiserade signalen. Ju fler bitar desto bättre blir upplösningen på mätningarna, men desto mer komplicerad blir elektroniken som behövs för att göra A/D-omvandlingen. Om man mäter både negativa och positiva spänningar så måste en av bitarna reserveras för att definiera tecknet på mätsignalen, och de återstående bitarna används som tidigare för att beskriva absolutbeloppet på mätningen.

Som ett exempel kan vi i Tabell 1.1 se hur ett fall skulle se ut med 3 bitar då vi mäter en positiv spänning med en maximal signalnivå på 5 V, och i Figur 1.2 ser vi hur en signal mätt med denna kvantisering skulle se ut.

Digitalt värde	Binär representation	Motsvarar mätvärde
0	000	0.000 – 0.625
1	001	0.625 – 1.250
2	010	1.250 – 1.875
3	011	1.875 – 2.500
4	100	2.500 – 3.125
5	101	3.125 – 3.750
6	110	3.75 – 4.375
7	111	4.375 – 5.000

Tabell 1.1: Signelnivåer för en mätsignal kvantiserad med 3 bitar.



Figur 1.2: Exempel på en signal samt dess kvantiserade mätning via en A/D-omvandlare med 3 bitar, det vill säga 8 möjliga signelnivåer.

Före själva A/D-omvandlaren har man den faktiska sensorn som från en fysikalisk enhet såsom en höjd, ett tryck, eller en ljusstyrka, genererar en spänning. För att i datorn räkna ut den relevanta enheten givet den kvantiserade spänningen så måste vi känna till hur sensorn fungerar. Det kan t.ex handla om att det råder ett direkt proportionellt förhållande mellan spänning och signalen som sensorn mäter, dvs att en ljusstyrka på 0 Lumen genererar 0 V och 10000 Lumen ger 5 V. Om vi använder A/D-omvandlaren i exemplet ovan vet vi då att värdet 4 betyder att ljusstyrkan ligger mellan $2.5 \cdot \frac{10000}{5}$ och $3.125 \cdot \frac{10000}{5}$ Lumen. Vi har även att upplösningen i ljusstyrka kommer att vara $10000/8$ Lumen.

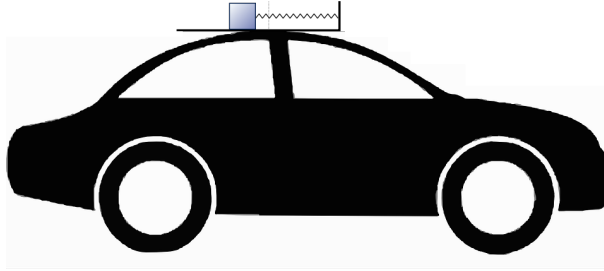
Den A/D-omvandlade signalen tas in (samplas) av datorn med en viss frekvens, till exempel 100 ggr per sekund. Effekter av sampling, och matematiken kring signalbehandling i diskret tid, är i fokus i nästa kapitel. Effekter av kvantisering

på grund av A/D-omvandling är inget vi analyserar i denna kurs, utan vi tänker oss att vi har så hög noggrannhet (antal bitar) att skillnaden jämfört med ideala mätningar är försumbar.

1.2 Accelerometer

En accelerometer är en fundamental sensor som till exempel finns i alla moderna mobiltelefoner och bilar, och som är användbar i tillämpningar där man vill mäta och agera på rörelser. Principen för en accelerometer är väldigt enkel, och bygger nästan alltid på att man registrerar hur en massa rör sig under acceleration.

Som ett exempel kan vi studera den hemmabyggsda takmonterade accelerometern i Figur 1.3. Då bilen accelererar kommer den friktionsfritt glidande massan



Figur 1.3: Hemmabyggsd accelerometer. När bilen accelererar rör sig den fjäderupphängda massan bakåt, och hur långt den rör sig är direkt proportionellt mot bilens acceleration.

med vikt M kg att påverkas dels av accelerationskraften, och dels av kraften från den utdragna fjädern. Om vi låter bilens acceleration vara konstant a m/s, och fjäderns utdragning vara d m, så måste det vid jämvikt gälla att

$$Ma = kd$$

där k är fjäderkonstanten (N/m). Genom att mäta hur mycket fjädern dras ut kan vi få fram accelerationen genom $a = kd/M$. I en riktig accelerometer i till exempel en mobiltelefon är naturligtvis denna konstruktion förbättrad, förädlad och miniaturiserad och kallas en MEMS-accelerometer, där MEMS står för micro electro-mechanical system. Principen är dock den samma. Genom att dessutom placera 3 st accelerometrar ortogonalt mot varandra, så kan man mäta accelerationer i alla riktningar, en så kallad 3-axlig accelerometer, eller 3D-accelerometer.

Dödräkning av hastigheter och positioner

I många tillämpningar är vi inte intresserade av accelerationen $a(t)$, utan mer intresserade av hastigheter $v(t)$ och positioner $x(t)$. En accelerometer kan utgöra

grunden för att ta fram dessa, genom sambanden

$$\dot{v}(t) = a(t), \quad \dot{x}(t) = v(t)$$

eller alternativt skrivet

$$\ddot{x}(t) = a(t)$$

Vi kan alltså räkna ut *förändringen* av hastigheten genom att integrera accelerations-signalen

$$v(t) - v(0) = \int_0^t a(\tau) d\tau$$

och positionsändringen genom att integrera vår beräknade hastighet

$$x(t) - x(0) = \int_0^t v(\tau) d\tau$$

Ett uppenbart problem med dessa definitioner är att de kräver kännedom om initial hastighet och position. I vissa tillämpningar kan det vara rimligt att anta att initial hastighet är 0, och att vi bara är intresserade av den relativa förflyttningen sedan start. Man använder dock typiskt inte accelerometrar direkt på detta naiva sätt (kallat dödräkning), utan om man vill ha hastighet och position så används accelerometern tillsammans med andra sensorer som ger någorlunda bra hastighets- och/eller positionsuppskattningar (kanske med långsam uppdateringshastighet), och accelerometern används därutöver för att extrahera vad som händer under snabba förlopp på korta tidshorisonter.

Notera även att beräkningarna av integralerna aldrig kan göras exakt, eftersom vår dator endast har tillgång till samplade mätvärden. I efterföljande kapitel beskrivs hur man praktiskt approximerar beräkningarna genom att ersätta integralerna med rektangelapproximationer baserade på de samplade värden man har.

Mätfel

Ett annat problem med dödräkningsstrategin ovan är att accelerationsmätningar aldrig är exakta. Om vi betecknar mätningen av accelerationen med $a_m(t)$ och den sanna accelerationen $a(t)$, så har vi typiskt ett konstant så kallat biasfel (kalibreringsfel och liknande) samt ett snabbt varierande mätbrus $e(t)$

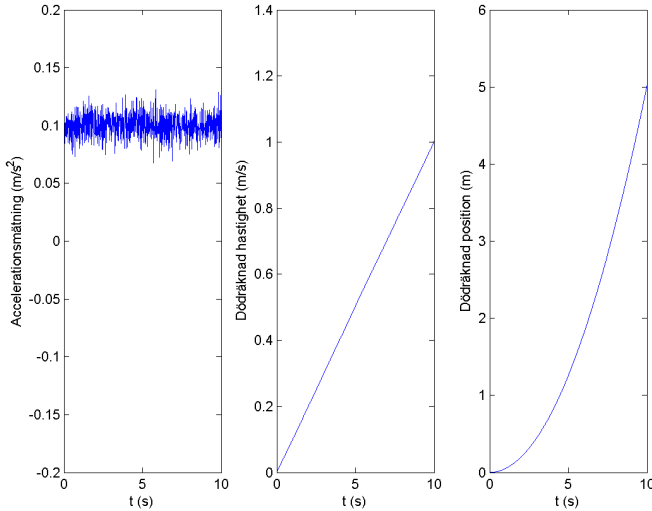
$$a_m(t) = a(t) + b + e(t)$$

När denna signal används för att uppskatta hastigheten $v_m(t)$ genom integration, så får vi (antar initial hastighet 0)

$$v_m(t) = \int_0^t a_m(\tau) d\tau = \int_0^t (a(\tau) + b + e(\tau)) d\tau = v(t) + bt + \int_0^t e(\tau) d\tau$$

Om mätbruset $e(t)$ är varierande kring 0 med medelvärde 0 så kommer dess integral typiskt bli liten, den medelvärdesbildas bort säger man. Biasfelet är dock högst

problematiskt. Vi ser att felet i skattningen av hastigheten kommer bli allt större med tiden. Man säger att det *driver*. Om man nu integrerar denna signal ytterligare en gång för att få fram ett positionsestimat, så kommer feltermen bt integreras till $\frac{1}{2}bt^2$, det vill säga positionsestimatet kommer väldigt snabbt bli helt oanvändbart.



Figur 1.4: Längst till vänster visas en accelerometersignal från mobiltelefonen i Figur 1.1. Under 10 sekunders datainsamlande i 100Hz (det vill säga 1000 datapunkter) ligger telefonen helt still på ett golv, men trots detta så visar accelerometern en varierande acceleration kring 0.1 m/s^2 . Accelerometern har både ett biasfel och högfrekvent mätfel. Figuren i mitten visar den integrerade accelerationsmätningen (som påstår att telefonen rör sig med 1 m/s efter 10 sekunder, samt den integrerade hastigheten (som säger att telefonen rört sig 5 meter under datainsamlandet). Notera att det högfrekventa bruset ej syns särskilt mycket i de integrerade mätningarna, utan det är främst biasfelet som ger upphov till felaktigt uppskattad hastighet och position.

1.3 Gyroskop

Även gyroskop, eller gyro, är vanliga i mobiltelefoner och bilar. Med ett gyroskop kan man mäta ett objekts *rotationshastighet*, och precis som en accelerometer så bygger grundprincipen ofta på en massas rörelse under rotation. Den fysikaliska effekt som utnyttjas för att detektera och mäta rotationshastigheten är i detta fall corioliseffekten.

Gyrot ger oss en mätning av rotationshastighet $\omega(t)$, och om man vill komma till rotationsvinkel $\theta(t)$ kan man precis som i accelerometerfallet använda sig av definitionen $\dot{\theta}(t) = \omega(t)$, det vill säga rotationen kan beräknas genom integration av uppmätt rotationshastighet

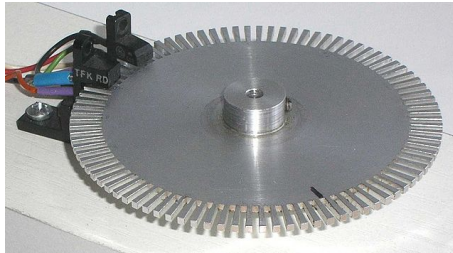
$$\theta(t) - \theta_0 = \int_0^t \omega(\tau) d\tau$$

En sådan här användning av ett gyro lider av precis samma problem som en accelerometer, i det att den endast ger oss en relativ vinkelförändring, samt att dödräkningen blir dålig (drift) om det finns biasfel. Av samma anledning så använder man därför ofta gyro i kombination med andra sensorer om man vill mäta vinklar.

1.4 Enkoder

Rotationsenkodrar används för att mäta *rotationshastigheter* på hjul och axlar. Mer avancerade så kallade absoluta rotationsenkodrar kan dessutom mäta absoluta vinklar. Principen för enkodrar är typiskt väldigt enkla, och vi skall här enbart diskutera så kallade optiska enkodrar.

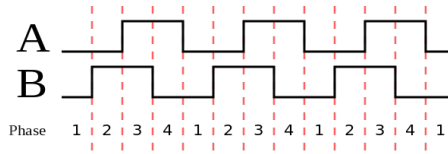
En skiva med en mängd jämnt fördelade spaltöppningar fästs på den axel vi vill göra mätningar på. På den ena sidan av skivan sitter en ljuskälla, och bakom skivan i linje med ljuskällan monteras en elektrisk krets med, till exempel, en ljuskänslig resistor. Då skivan roterar kommer spänningen över den ljuskänsliga



Figur 1.5: Enkoder för mätning av rotationshastighet. Bild från Wikimedia Commons.

resistorn att variera då ljuskällan blockeras respektive släpps genom. Mätning av denna spänning ger upphov till ett så kallat pulståg, och genom att studera hur ofta signalen växlar värde kan man enkelt, givet skivans geometri, räkna ut hur snabbt skivan roterar.

Genom att använda sig av två stycken ljuskällor och ljussensorer, placerade på ett sådant sätt att de ligger ur fas relativt varandra, kan man förutom rotationshastighet dessutom ta fram rotationsriktning, genom att studera i vilken ordning de två signalerna slås på respektive av.



Figur 1.6: Pulståg från enkoder med möjlighet till rotationsdetektion med hjälp av två ljussensorer A och B. Denna enkoder är byggd så att en övergång från $A = 0, B = 0$ till $A = 0, B = 1$ indikerar rotation medurs. Hade vi istället fått en övergång till $A = 1, B = 0$ hade vi detekterat en motsols rotation. Bild från Wikimedia Commons.

Prestanda på en enkoder definieras av upplösning och noggrannhet. Genom att använda fler spalter förbättrar man upplösningen, men samtidigt ställer detta större krav på elektroniken och samplingshastigheten så att noggrannheten som erhålls faktiskt avspeglar upplösningen. Om man har n spalter (hål) så kommer man få $2n$ pulser per varv, och ha en upplösning på $360^\circ/(2n)$. Om man använder sig av två ljussensorer för att även detektera rotationsriktning, ökar man samtidigt upplösningen till $360^\circ/(4n)$. På grund av faktorn 4 här kallas dessa enkodrar ibland kvadraturenkodar.

Till sist nämner vi även att det finns liknande enkodrar för att mäta linjära (translativa) rörelser till skillnad från rotationsrörelser. Dessa enkodrar kallas linjärenkodar.

1.5 GPS och radio

För positionsbestämning räcker det oftast inte med accelerometrar. Dödräkning för att få fram absoluta positioner och hastigheter blir ofta dåliga då man har drift på grund av biasfel i mätningarna.

Ett numera vanligt sätt att få en absolut position är via en GPS-mätning, det vill säga triangulering till satelliter. Nackdelen med en GPS är, förutom kostnad och komplexitet, att den kräver fri radiosikt till satelliterna, och således inte kan användas inomhus. Ett alternativt sätt att få en absolut position är att använda sig av triangulering till kända radiomaster, till exempel mobilbasstationer. Även här kan man naturligtvis drabbas av radioskugga så att inga mätningar erhålls. I en känd miljö inomhus kan man skapa sina egna radiofyror och använda sig av triangulering till dessa. Detta kan vara ett alternativ för att göra positionering i industriella sammanhang, till exempel för mobila robotar som skall röra sig i stora lagerlokaler.

Dessa absoluta positionsmätningar är naturligtvis inte perfekta, utan har mätfel. På grund av varierande propagering av radiovågor och diverse effekter av reflektioner och blockering av signaler, så får man variationer i positionsmätningarna även om man håller sig helt stilla. Som i accelerometerfallet så kan vi tänka oss en

mätning av en position $x_m(t)$ av en sann position $x(t)$, med ett konstant mätfel b samt varierande mätfel $e(t)$.

$$x_m(t) = x(t) + b + e(t)$$

Ett konstant mätfel b kan vi inte göra något åt. Om vår GPS-sensor säger att vi står utanför C-huset när vi i själva verket står utanför B-huset, så finns det inget att göra åt det. De snabba variationerna i $e(t)$ kan vi däremot försöka kompensera för. Om GPS-sensorn säger att vi står utanför B-huset men mätningen fluktuerar några meter fram och tillbaka, så kan vi medelvärdesbilda bort detta. Som vi senare kommer kalla det, så lågpasfilterar vi signalen.

En absolutmätning med mätfel har komplementära egenskaper jämfört med en accelerometer. För en accelerometer såg vi att vi inte var så känsliga för det högfrekventa mätfelet, men att en feljusterad nollnivå (bias) ger upphov till att hastighetsestimatet snabbt driver iväg och blir oanvändbart. Om vi nu istället försöker få fram hastigheten från positionsmätningen så har vi

$$\dot{x}_m(t) = \dot{x}(t) + 0 + \dot{e}(t)$$

Det konstanta mätfelet påverkar inte hastighetsestimatet $x_m(t)$ alls, men om $e(t)$ varierar snabbt, så kommer dess derivata att vara stor, och hastighetsestimatet blir helt fel. Går man ett steg längre och försöker få fram accelerationen så blir det än värre.

$$\ddot{x}_m(t) = \ddot{x}(t) + 0 + \ddot{e}(t)$$

Tänk dig till exempel att $e(t) = \sin(100t)$. Vi har då $\dot{e}(t) = 100 \cos(100t)$ och $\ddot{e}(t) = -10000 \sin(100t)$. Vi kommer endast se det deriverade mätfelet, de intressanta signalerna $\dot{x}(t)$ och $\ddot{x}(t)$ drunknar helt i bruset. För att kunna få fram ett bra estimat av derivatan måste man först filtrera bort de högfrekventa komponenterna i mätsignalen med hjälp av lågpasfilter (se nästa kapitel) innan man deriverar signalen.

1.6 Laser, radar och sonar

I många tillämpningar är man inte intresserad av en absolut position, men behöver känna till avstånd. Det kan till exempel handla om mobila robotar som rör sig inomhus där man vill känna till avstånd till väggar.

En generell strategi för att erhålla en sådan avståndsmätning är att man skickar ut en energipuls (radiovåg, ljus, ljud) och sedan mäter den tid det tar innan man får ett eko eller en reflex. Med kännedom om energipulsens utbredningshastighet kan man därigenom räkna ut avståndet till närmsta föremål.

Genom att använda en roterande laser eller radar kan man skapa sig en god bild av hur omgivningen ser ut.

1.7 Magnetometer

Genom att utnyttja att jorden har ett magnetfält med bestämda riktningar så kan man bygga sensorer som detekterar ett objekts vinkel relativt jordens magnetfält, det vill säga effektivt en 3-axlig kompass. En sådan magnetometer kan därigenom användas som en bas för att mäta ett objekts rotation. Ett problem med rotationsuppskattningar med hjälp av en magnetometer är att den är känslig för störningar av magnetfältet som kan uppkomma i närheten av till exempel elektrisk utrustning och magnetiska föremål.

1.8 Barometer

Quadcoptern i figur 1.1 är utrustad med en barometer, det vill säga en mätare av atmosfärstrycket. En sådan kan man använda för att mäta relativa höjdflyttningar. Trycket är förvisso beroende på vädret, men på en kort tidsskala (minuter till timmar) så kan en enkel barometer ge en god uppskattning av höjd (eller höjdförändring sedan start, eftersom man inte vet vilket tryck som motsvarar nollnivå) ner till decimeternivå, vilket kan användas som mätsignal för att hålla en konstant hovringshöjd på en enkel quadcopter. Genom att använda barometersignalen tillsammans med integrerade accelerometersignaler kan precisionen förbättras.

Kapitel 2

Signaler och signalbehandling

2.1 Signaler

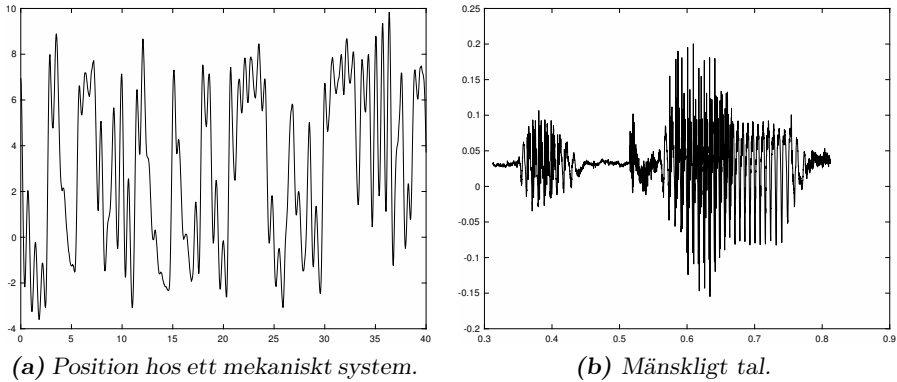
En *signal* är en variabel eller storhet som ändras som en funktion av tiden. Vissa signaler kan beskrivas matematiskt med elementära funktioner medan andra måste definieras i relation till ett fysiskt objekt. Några exempel på signaler med enkla matematiska beskrivningar är $v(t) = e^{-3t}$, $w(t) = \sin(2t)$ och $x(t) = \text{sign}(t)$ medan hastigheten hos en viss bil, ljudet från en viss fågel och den elektriska aktiviteten hos en viss persons hjärta är exempel på fysiska signaler som är svåra att ge en exakt matematisk definition. Man kan dock karakterisera dessa signaler på andra sätt, till exempel genom att plotta dem som funktioner av tiden. I figur 2.1 visas två signaler, dels den mätta positionen hos ett oscillativt mekaniskt system och dels uppmätt mänskligt tal.

Signaler kan vara *tidskontinuerliga* eller *tidsdiskreta*. En tidskontinuerlig signal är definierad för alla reella tidpunkter t och signalerna $v(t)$, $w(t)$ och $x(t)$ som nämndes ovan är exempel av denna typ. En tidsdiskret signal är däremot bara definierad för en sekvens av diskreta tidpunkter t_1, t_2, t_3, \dots . Ett exempel på en sådan signal är

$$p(t_k) = 2^{-t_k}, \quad t_k = 0, 1, 2, \dots$$

I detta fall är signalen inte definierad vid tidpunkter som inte är icke-negativa heltal och det är därför meningslöst att prata om till exempel $p(0.5)$.

När man har definierat en signal kan det också vara intressant att prata om olika egenskaper som till exempel maximalt och minimalt värde, medelvärde och gränsvärde när tiden går mot oändligheten. Man kan också definiera nya signaler genom att till exempel derivera, integrera eller tidsförskjuta den ursprungliga signalen eller genom att kombinera flera signaler på olika sätt. Vissa signaloperationer går enkelt att beskriva och utföra direkt på signalernas tidsbeskrivningar



Figur 2.1: Två exempel på signaler som plottats som funktioner av tiden. I den vänstra figuren visas positionen hos ett oscillerande mekaniskt system och i den högra visas uppmätt mänskligt tal.

men i andra fall kan man få enklare räkningar och samband genom att istället använda en så kallad *transform*.

Det finns flera typer av transformeringar som lämpar sig för olika typer av signaler. Generellt kan man dock säga att en transform av en signal är en funktion av en annan variabel än tiden som utgör en ekvivalent beskrivning av signalen. En operation på den ursprungliga signalen i *tidsdomänen* svarar mot en viss annan operation på signalens transform i *transformdomänen* och ibland är den ena operationen betydligt enklare än den andra. Man kan byta mellan de båda domänerna åt båda hållen och till exempel först transformera en viss signal, sedan manipulera den i transformdomänen, för att till sist *inverstransformera* resultatet för att återigen få en signal i tidsdomänen.

Ett exempel på en transform som kan användas för tidskontinuerliga signaler är *laplacetransformen* som definieras som

$$U(s) = \mathcal{L}\{u(t)\}(s) = \int_{t=0}^{\infty} e^{-st} u(t) dt,$$

där $s = \sigma + i\omega$ är ett komplext tal sådant att integralen konvergerar. (Denna definition gäller för signaler som endast är nollskilda för $t \geq 0$, vilket är generellare än det först låter eftersom man ofta kan välja själv vad som ska svara mot $t = 0$.) En fördel med laplacetransformen är att den underlättar många beräkningar som derivering och integrering. Några av de mest använda egenskaperna

hos laplacetransformen är:

$$\mathcal{L}\{af(t) + bg(t)\} = aF(s) + bG(s) \quad (2.1)$$

$$\mathcal{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0) \quad (2.2)$$

$$\mathcal{L}\left\{\int_0^t f(\tau) d\tau\right\} = \frac{1}{s}F(s) \quad (2.3)$$

$$\mathcal{L}\{f(t - L)\} = e^{-sL}F(s) \quad (2.4)$$

$$\mathcal{L}\left\{\int_0^t f(t - \tau)g(\tau) d\tau\right\} = F(s)G(s) \quad (2.5)$$

Exempel: Laplacetransformen för en signal

Den tidskontinuerliga signalen

$$u(t) = \begin{cases} 0 & t < 0 \\ e^{-at} & t \geq 0 \end{cases}$$

har laplacetransformen

$$U(s) = \int_{t=0}^{\infty} e^{-(s+a)t} dt = \left[-\frac{1}{s+a} e^{-(s+a)t} \right]_{t=0}^{\infty} = \frac{1}{s+a}$$

om $\text{Re}(s) > -a$.

I föregående exempel räknade vi ut laplacetransformen direkt med hjälp av dess definition. Det finns dock gott om tabeller och datorhjälpmedel som man kan använda istället för att underlätta användningen av transformeringar.

Anledningen till att laplacetransformen är så användbar är att dess egenskaper gör den smidig att använda när man vill förstå *linjära tidskontinuerliga system*. Ett system är någon typ av fysiskt eller begreppsmässigt avgränsad enhet som omvandlar en *insignal* till en *utsignal*. Ett system är linjärt om det har egenskapen att en linjärkombination av insignaler ska ge motsvarande linjärkombination av utsignaler. Detta innebär att om insignalen $u_1(t)$ ger utsignalen $y_1(t)$ och $u_2(t)$ ger $y_2(t)$ så ska insignalen $\alpha_1 u_1(t) + \alpha_2 u_2(t)$ ge utsignalen $\alpha_1 y_1(t) + \alpha_2 y_2(t)$ för godtyckliga konstanter α_1 och α_2 .

Ett linjärt tidskontinuerligt system kan beskrivas av en differentialekvation

$$\frac{d^n}{dt^n}y(t) + a_1 \frac{d^{n-1}}{dt^{n-1}}y(t) + \dots + a_n y(t) = b_0 \frac{d^m}{dt^m}u(t) + \dots + b_m u(t). \quad (2.6)$$

Med hjälp av laplacetransformering och upprepad användning av räkneregeln att derivering svarar mot multiplikation med s i transformdomänen om alla initialvillkor är noll kan sambandet mellan in- och utsignal skrivas på den alternativa

formen

$$Y(s) = \frac{b_0 s^m + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} U(s) \quad (2.7)$$

där

$$G(s) = \frac{b_0 s^m + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} \quad (2.8)$$

är systemets *överföringsfunktion*. Genom att studera överföringsfunktionens egenskaper kan man därför förklara hur systemet förändrar insignalens laplacetransform och även beräkna utsignalens laplacetransform.

Om man har en tidsdiskret signal som man vill översätta till en transformdomän så måste man använda *z-transformen* istället för laplacetransformen. Z-transformen $U(z)$ av en tidsdiskret signal $u(k)$, där k är ett heltal, definieras som

$$U(z) = \mathcal{Z}\{u(k)\}(z) = \sum_{k=0}^{\infty} u(k)z^{-k} \quad (2.9)$$

där z är ett komplext tal sådant att serien konvergerar. (Även denna definition gäller för signaler som endast är nollskilda för $k \geq 0$.) Här är det värt att notera att om den tidsdiskreta signal som man egentligen studerar inte har heltalsvärda argument så kan man alltid definiera en ny signal $\tilde{u}(k) = u(t_k)$ som har samma signalvärden som den ursprungliga men nu med för ett heltalsvärt argument. Definitionen av z-transformen kan sedan tillämpas direkt på den nya signalen.

Precis som laplacetransformen har z-transformen egenskapen att den underlättar många beräkningar. Några av z-transformens mest användbara egenskaper är:

$$\mathcal{Z}\{ay(k) + bv(k)\} = aY(z) + bV(z) \quad (2.10)$$

$$\mathcal{Z}\{y(k-1)\} = z^{-1}Y(z) + y(-1) \quad (2.11)$$

$$\mathcal{Z}\{y(k+1)\} = zY(z) - zy(0) \quad (2.12)$$

$$\mathcal{Z}\left\{\sum_{m=0}^k y(k-m)v(m)\right\} = Y(z)V(z) \quad (2.13)$$

Z-transformen är användbar när man ska analysera *linjära tidsdiskreta system*. Sådana system kan beskrivas med *differensekvationer*, som är en slags tidsdiskret motsvarighet till differentialekvationer. En linjär differensekvation

$$y(k+n) + a_1 y(k+n-1) + \dots + a_n y(k) = b_0 u(k+n) + \dots + b_n u(k) \quad (2.14)$$

kan, genom upprepade användning av räkneregeln $\mathcal{Z}\{y(k+1)\} = zY(z)$ om alla initialvillkor är noll, skrivas på den alternativa formen

$$Y(z) = \frac{b_0 z^n + \dots + b_n}{z^n + a_1 z^{n-1} + \dots + a_n} U(z) \quad (2.15)$$

där

$$G_{T_s}(z) = \frac{B(z)}{A(z)} = \frac{b_0 z^n + \dots + b_n}{z^n + a_1 z^{n-1} + \dots + a_n} \quad (2.16)$$

är det tidsdiskreta systemets överföringsfunktion. Även här kan man förstå hur systemet påverkar en insignal genom att studera överföringsfunktionens egenskaper.

2.2 Signalers frekvensinnehåll

En intressant egenskap hos transformerna som vi införde i föregående avsnitt är att de kan användas för att förstå en signals *frekvensinnehåll*. Betrakta till att börja med signalen $u(t) = \cos(2t)$, $t \geq 0$ ($u(t) = 0$ då $t < 0$). Det här är för $t \geq 0$ en periodisk signal med periodtid $T_0 = \pi$ s eftersom cosinusfunktionen har period 2π . Ett annat sätt att beskriva signalen är att säga att den är periodisk med *frekvensen* $f_0 = \frac{1}{T_0} = \frac{1}{\pi}$ Hz, vilket betyder att signalen genomlöper $\frac{1}{\pi}$ perioder under en sekund. Genom att multiplicera f_0 med 2π får vi *vinkelfrekvensen* $\omega_0 = 2\pi f_0 = 2$ rad/s. Vinkelfrekvensen är alltså den konstant som står framför t i argumentet till cosinusfunktionen.

Laplacetransformen för signalen $u(t) = \cos(2t)$ visar sig vara

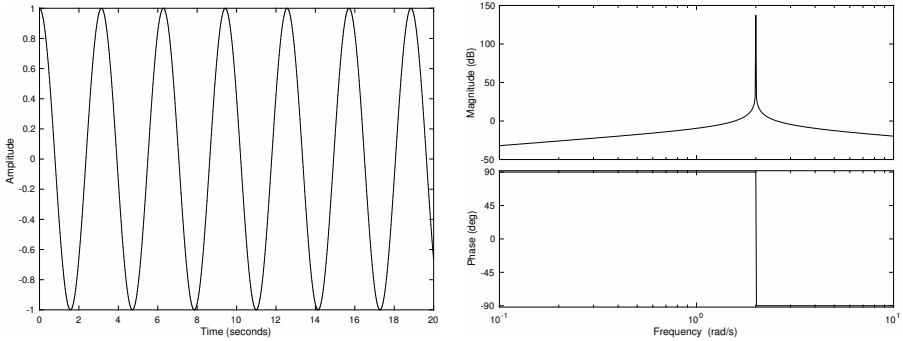
$$U(s) = \frac{s}{s^2 + 4}.$$

Argumentet s i detta uttryck är ett komplext tal och genom att göra det specifika valet $s = i\omega$ och sedan beräkna och plotta *amplituden* $|U(i\omega)|$ och *fasen* $\arg(U(i\omega))$ för olika reella och positiva värden på ω kan vi göra en intressant observation. Det visar sig nämligen att $|U(i\omega)|$ blir mycket stort när vi väljer ω nära $\omega_0 = 2$ och värdet på $|U(i\omega)|$ går mot oändligheten när $\omega \rightarrow 2$. Ett sätt att visualisera detta är att plotta $|U(i\omega)|$ och $\arg(U(i\omega))$ som funktioner av ω och en sådan plott visas i figur 2.2. Två av axlarna i denna figur är logaritmiska och amplituden är av tradition angiven i enheten decibel (dB), som beror på den vanliga amplituden enligt

$$|U(i\omega)|_{\text{dB}} = 20 \log_{10}(|U(i\omega)|).$$

Laplacetransformen i specialfallet $s = i\omega$ brukar kallas för *fouriertransformen* av den signal som studeras. Det faktum att fouriertransformens amplitud är som störst vid $\omega = 2$ indikerar att signalen har den mesta effekten vid denna vinkelfrekvens.

Genom att studera fouriertransformen kan vi alltså upptäcka om signalen innehåller en ren cosinus- (eller sinus-)term eftersom en sådan ger upphov till en tydlig topp i amplitudkurvan. Om man går vidare och undersöker en godtycklig periodisk signal så kommer man att se att dess amplitudkurva innehåller flera toppar, vilket indikerar att den kan uttryckas som en linjärkombination av flera cosinussignaler, något som också går att visa teoretiskt. Man kan också studera



(a) Signalen $u(t) = \cos(2t)$ som funktion av tiden t .

(b) Fouriertransformen $U(i\omega)$ som funktion av vinkelfrekvensen ω .

Figur 2.2: En periodisk signal (a) och dess fouriertransform (b). Fouriertransformens amplitud är som störst vid $\omega = 2$, vilket indikerar att signalen har den mesta effekten vid denna vinkelfrekvens.

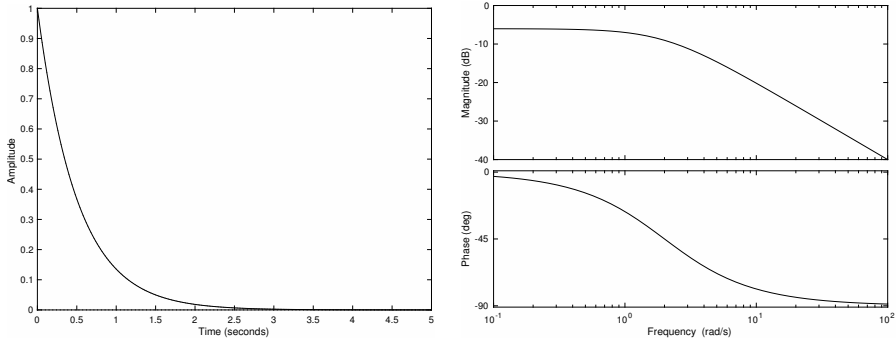
godtyckliga icke-periodiska signaler på det här sättet och man ser då att de normalt innehåller ett kontinuum av frekvenskomponenter. Ett exempel på en sådan signal visas i figur 2.3.

Eftersom insignal-utsignalsambandet (2.7) gäller även för fouriertransformen är det ofta intressant att studera $G(i\omega)$ om man känner överföringsfunktionen $G(s)$. Även här har man infört ett speciellt namn och man brukar kalla $G(i\omega)$ för systemets *frekvenssvar*. Frekvenssvaret kan plottas på samma sätt som fouriertransformen och det diagram som man då får brukar kallas för ett *bodediagram*. I figur 2.4 visas bodediagrammet (över frekvenssvaret) för ett system som beskrivs av överföringsfunktionen

$$G(s) = \frac{100(2s + 10)}{(s + 10)^3}. \quad (2.17)$$

I det här fallet ser vi att amplitudkurvan är nära 0 dB (1 i vanlig skala) upp till ungefär $\omega = 10$ och vi kan därför dra slutsatsen att insignaler som har sin mesta effekt vid låga vinkelfrekvenser under 10 rad/s (till exempel en lågfrekvent cosinus-signal) kommer att passera systemet utan att deras amplitud förändras nämnvärt. Insignaler med signifikant frekvensinnehåll över 10 rad/s kommer däremot att påverkas mer av systemet och en ren cosinussignal med en vinkelfrekvens på 50 rad/s kommer att få en klart lägre amplitud efter att ha passerat systemet. Ett system av den här typen brukar därför sägas ha en *lågpasskaraktär* efter som den låter lågfrekventa signaler "passera" mer eller mindre obehindrat.

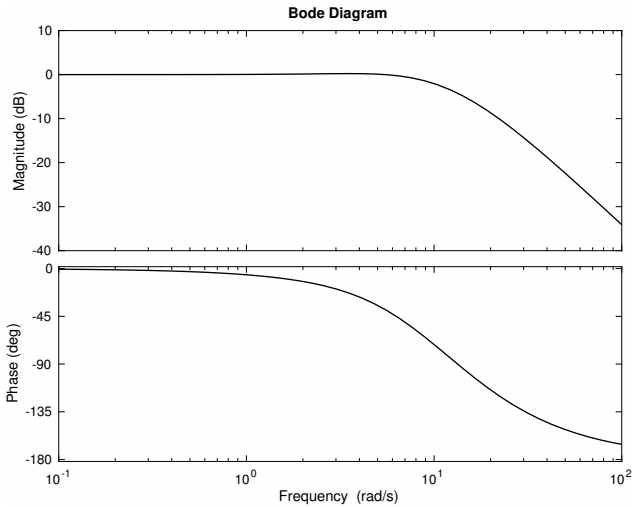
Man kan också definiera en *tidsdiskret fouriertransform* genom att sätta $z = e^{i\omega T_s}$ i z -transformen för en signal. Precis som i det tidskontinuerliga fallet ger detta ett verktyg för undersökning av en signals frekvensinnehåll. Substitutionen $z = e^{i\omega T_s}$ i en tidsdiskret överföringsfunktion ger också systemets frekvenssvar och



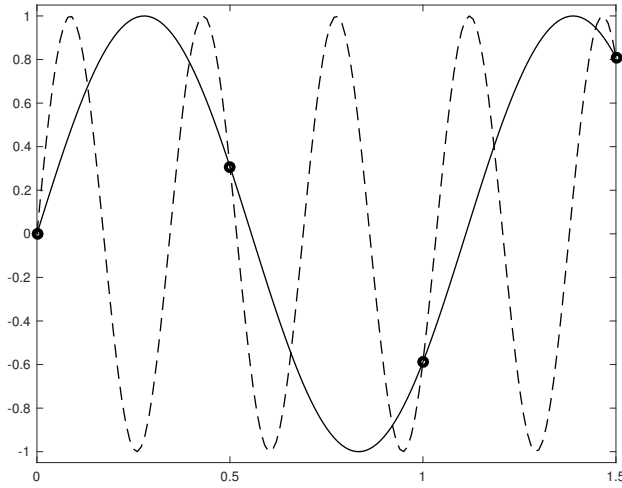
(a) Signalen $u(t) = e^{-2t}$ som funktion av tiden t .

(b) Fouriertransformen $U(i\omega)$ som funktion av vinkelfrekvensen ω .

Figur 2.3: En icke-periodisk signal (a) och dess fouriertransform (b). Fouriertransformens amplitud varierar men utan att uppvisa några skarpa toppar vilket är en följd av att signalen är icke-periodisk och har sin effekt fördelad över oändligt många olika frekvenser.



Figur 2.4: Bodediagram som visar frekvenssvaret för systemet (2.17). Amplitudkurvan visar att systemet är av lågpasskaraktär och att det kommer att dämpa frekvenskomponenter över 10 rad/s.



Figur 2.5: Aliaseffekten i samband med sampling av signaler. I detta fall är det inte möjligt att skilja de båda tidskontinuerliga signalerna åt om man bara har tillgång till de samplade värdena.

det kan plottas i ett bodediagram.

2.3 Sampling

Det vanligaste sättet att mäta en fysisk signal idag är att *sampla* den, det vill säga att mäta dess värden vid diskreta tidpunkter, ofta jämnt fördelade i tiden vid tidpunkter $t = kT_S$, där T_S är ett tidsintervall som kallas *sampleintervall*. Även om den ursprungliga signalen $y(t)$ var tidskontinuerlig så resulterar den här metoden i en tidsdiskret signal $y(kT_S)$. Anledningen till att metoden att sampla en signal är så vanlig är att resultatet lämpar sig väl för att spara och bearbeta med hjälp av en dator.

En utmaning med sampling av signaler är dock att det finns en risk för att signaler kan misstolkas och blandas ihop med andra signaler efter samplingen, något som brukar kallas för *aliaseffekten* eller *vinkningsdistortion*. Detta problem illustreras i figur 2.5 där det visas att två olika tidskontinuerliga sinussignaler kan resultera i samma sampelvärden. Om man bara har tillgång till de samplade signalvärdena kan man alltså inte avgöra vilken den ursprungliga signalen var.

I vissa fall kan man undvika aliaseffekten genom att se till att man samplar tillräckligt ofta, det vill säga genom att välja samplingstiden T_S tillräckligt liten eller samplingsvinkelfrekvensen

$$\omega_S = 2\pi f_S = \frac{2\pi}{T_S}$$

tillräckligt stor. (Här är f_S samplingsfrekvensen i hertz.) Förutsättningarna för att det ska gå specificeras av det så kallade *samplingsteoremet*.

Samplingsteoremet: En signal som inte innehåller några signalkomponenter över vinkelfrekvensen ω_0 kan exakt rekonstrueras från samplade värden om samplingsvinkelfrekvensen ω_S uppfyller olikheten

$$\omega_0 \leq \frac{\omega_S}{2}.$$

Vinkelfrekvensen $\omega_N = \frac{\omega_S}{2}$ brukar kallas *nyquistfrekvensen*.

2.4 Signalbehandling

Ordet *signalbehandling* är ett samlingsnamn på en mängd olika resultat om och metoder för att modifiera en signal. Signalbehandling kan göras både i kontinuerlig och diskret tid men eftersom de tidsdiskreta metoderna idag dominerar i många tillämpningsområden kommer vi att fokusera på dem här.

Användningen av så kallade *linjära filter* är en viktig klass av metoder för signalbehandling. Ett linjärt tidsdiskret filter är ett linjärt system (2.16) som man har designat och implementerat med det enda syftet att modifiera en viss signal. Tanken är att förstärka och förbättra vissa egenskaper hos signalen genom att låta den passera filtret, och man pratar därför ofta om *filtrering* av signaler. Precis som generella tidsdiskreta system svarar ett filter av typen (2.16) mot en differensekvation (2.14), och detta ger direkt en möjlighet att implementera filtret i programkod.

Ett allmänt filter av typen (2.16) har egenskapen att filtrets utsignal beror på *alla* gamla insignaler eftersom filtrets utsignal vid en viss tidpunkt beror både på den senaste insignalen och på en linjärkombination av gamla ut- och insignaler enligt

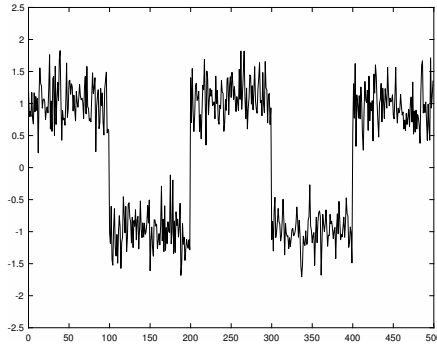
$$y(k) = -a_1y(k-1) - \dots - a_ny(k-n) + b_0u(k) + \dots + b_nu(k-n). \quad (2.18)$$

Man brukar säga att ett sådant här filter har ett oändligt *impulssvar* (eng.: infinite impulse response, IIR) eftersom effekten av en ensam insignalpuls vid tiden noll syns oändligt länge i utsignalen.

Ett viktigt specialfall av det allmänna linjära filtret fås om man sätter koefficienterna a_k , $k = 1, \dots, n$ till noll i (2.18). I detta fall blir filtrets utsignal helt enkelt en linjärkombination av ett ändligt antal insignalkomponenter, det vill säga

$$y(k) = b_0u(k) + \dots + b_nu(k-n). \quad (2.19)$$

I detta fall har filtret endast ett ändligt impulssvar (eng.: finite impulse response, FIR) eftersom beroendet på en viss insignalkomponent är borta efter $n+1$ samplintervall. Om man väljer alla nollskilda $b_k = \frac{1}{n+1}$ så blir FIR-filtret ett slags



Figur 2.6: En uppmätt fyrkantssignal som innehåller mätbrus.

medelvärdesfilter som beräknar medelvärdet av de $n + 1$ senaste insignalkomponenterna.

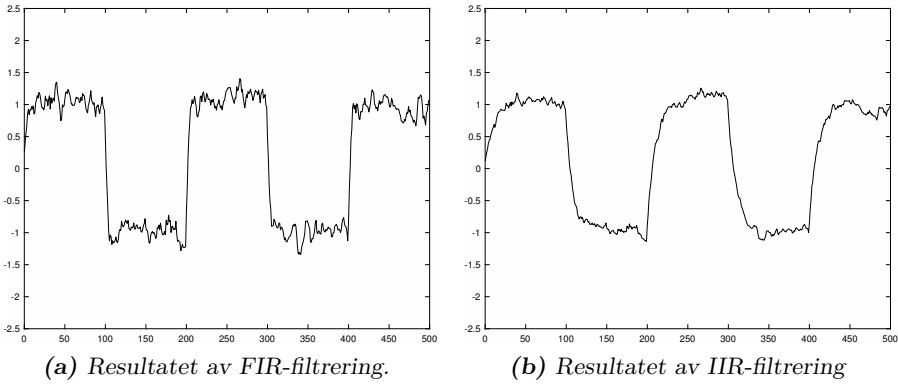
Det finns en utförlig litteratur om linjära filter och ett flertal principer för hur de kan designas. Det finns också hela klasser av färdiga filter med olika egenskaper. Några vanliga filterklasser är butterworthfilter, chebyshefilter, besselfilter och elliptiska filter och i många fall kan det vara värt att undersöka om man kan använda något standardfilter innan man sätter igång att designa ett eget.

Precis som för allmänna linjära system så kan man tolka ett filters egenskaper genom att plotta dess frekvenssvar i ett bodediagram. Frekvenssvarets utseende visar vilka signaler som filtret släpper igenom och några vanliga specialfall är filter av *lågpass*-, *bandpass*- eller *högpass*karaktär. Om man vet vilken karaktär filtret ska ha kan man som sagt ofta använda ett filter från en standardklass. Om man till exempel väljer att använda ett butterworthfilter av lågpasskaraktär är det då bara två parametrar som man måste välja, filtrets ordningstal och dess *brytfrekvens*, som är den frekvens där filtrets amplitud börjar avta. När man har valt dessa parametrar så kan filterkoefficienterna a_k och b_k enkelt beräknas med en färdig algoritm, till exempel i Matlab.

Vi ska nu avslutningsvis illustrera hur filtrering av en signal går till och vad resultatet blir i ett exempel.

Exempel: Filtrering av en brusig signal

Antag att man har mätt upp en fyrkantssignal med en enkel sensor som är känslig för störningar och som gör att den uppmätta signalen är ganska brusig. Ett exempel på en sådan signal visas i figur 2.6. Genom att använda sig av signalbehandling och ett par enkla filter kan man modifiera den mätta signalen så att bruset delvis undertrycks.



Figur 2.7: Resultatet av två typer av lågpasfiltrering av en brusig fyrkantssignal.

FIR-filtret

$$H_{\text{LP-FIR}}(z) = 0.2134 \sum_{k=0}^5 0.9^k z^{-k}$$

och IIR-filtret

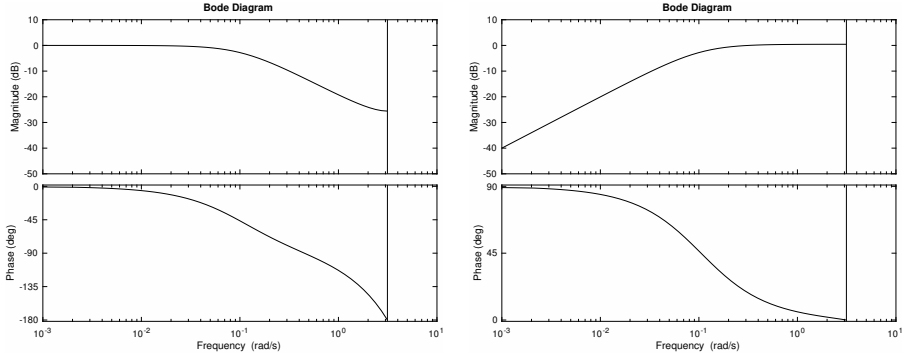
$$H_{\text{LP-IIR}}(z) = \frac{0.1}{z - 0.9}$$

är två lågpasfilter och om man skriver kod som svarar mot de differensekvationer som filtren representerar får man en filteralgoritm som kan användas på en uppmätt signal. Om man filtrerar den brusiga fyrkantssignalen med dessa båda filter får man resultatet som visas i figur 2.7. Det är enkelt att se att de filtrerade signalerna är mindre brusiga, men också att den underliggande fyrkantssignalen även den har påverkats lite av filtreringen.

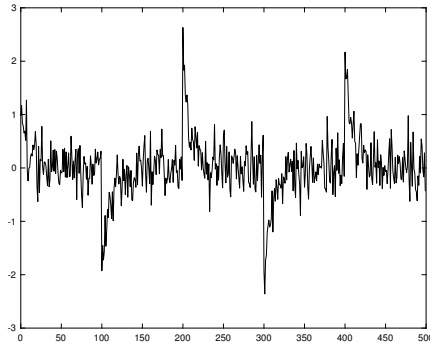
Om man istället skulle vara intresserad av det högfrekventa signalinnehållet kan man enkelt definiera ett högpasfilter som

$$H_{\text{HP-IIR}}(z) = 1 - H_{\text{LP-IIR}}(z) = \frac{z - 1}{z - 0.9}.$$

I figur 2.8 visas frekvenssvaren för de båda IIR-filtren och i figur 2.9 visas resultatet av att använda högpasfiltret på den brusiga fyrkantssignalen.

(a) Bodediagrammet för $H_{LP-IIR}(z)$.(b) Bodediagrammet för $H_{HP-IIR}(z)$.

Figur 2.8: Bodediagram som visar frekvenssvaren för ett lågpassfilter (a) och motsvarande högpasfilter (b).



Figur 2.9: Resultatet av högpasfiltrering av en brusig fyrkantssignal.

Kapitel 3

Implementering av regulatorer

Nuförtiden implementeras de allra flesta regulatorer med hjälp av datorer men trots det är det mycket vanligt att man gör själva reglerdesignen i kontinuerlig tid. Det behövs därför systematiska metoder för att ta fram tidsdiskreta approximationer av tidskontinuerliga regulatorer. Sådana metoder kommer att beskrivas i detta kapitel. En tidsdiskret implementering ger också möjlighet att med ganska enkla medel lösa ett problem som har med styrsignalmättning att göra. Detta kommer här att diskuteras för specialfallet PID-regulatorer.

3.1 Samplande regulatorer

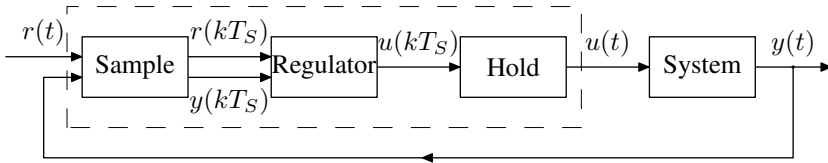
En regulator är i sig ett system som från de externa signalerna r och y skapar styrsignalen u . Detta system kan vara såväl tidskontinuerligt som tidsdiskret. Det finns en mångfald olika sätt att implementera det systemsamband som regulatorn representerar. Traditionell teknik har använt mekaniska, pneumatiska, hydrauliska och elektriska kopplingar för att realisera de differentialekvationer som regulatorsambandet egentligen utgör. Idag dominerar datorimplementerade regulatorer helt. Skälet är att det numera är enklare och billigare att realisera signalsamband i en dator, mikroprocessor eller med digital VLSI-teknik.

Datorimplementering av en regulator har dock en viktig konsekvens. En dator arbetar sekventiellt med ändligt antal mätdata. Den avläser värdena på $y(t)$ och $r(t)$ vid diskreta tidpunkter, t_k , $k = 1, 2, \dots$, och beräknar styrsignalen till processen baserat på dessa värden. En regulator som arbetar på ett sådant sätt ska vi kalla en *samplande* regulator (sampel = stickprov) eftersom dess styrsignal baserar sig på "stickprov" av processens utsignal och referenssignalen. Vi ska genomgående anta att samplingen sker *ekvidistant*, det vill säga med konstant tidsmellanrum

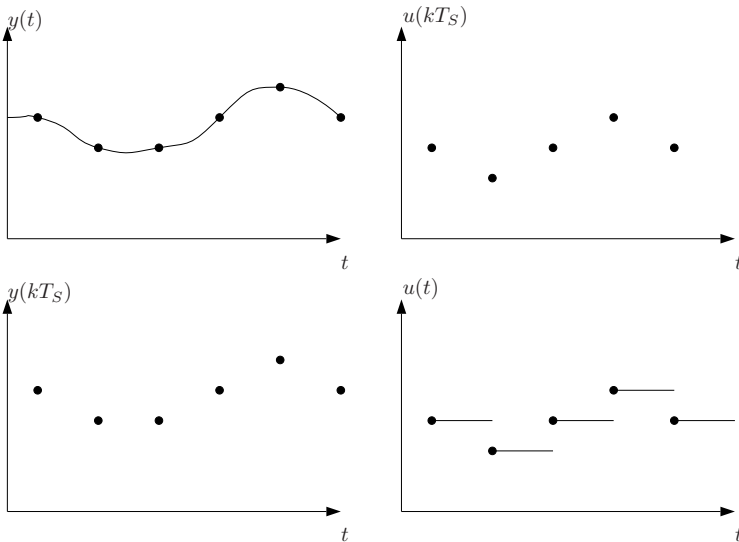
T_S , vilket medför att samplingsstidpunkterna kan uttryckas

$$t_k = kT_S, \quad k = 1, 2, \dots$$

Arbets sättet för en samplande regulator ges i princip av figurerna 3.1 och 3.2. Blocket "Sample" tar emot de tidskontinuerliga signalerna $y(t)$ och $r(t)$, samplar



Figur 3.1: En samplande regulator.



Figur 3.2: Signalerna i figur 3.1.

dem och skickar vidare en följd av tal, $y(kT_S)$ och $r(kT_S)$, som är signalernas värden i samplingsögonblicken. Blocket "Regulator" arbetar i diskret tid och beräknar från talföljderna $r(kT_S)$ och $y(kT_S)$ en annan följd av tal, $u(kT_S)$, enligt någon regel som vi tillsvidare låter vara ospecificerad. Blocket "Hold" tar emot talföljden $u(kT_S)$, och skickar vidare en tidskontinuerlig signal $u(t)$ vars värde är lika med $u(kT_S)$ för $kT_S \leq t < (k+1)T_S$. Denna signal $u(t)$ är insignal till det tidskontinuerliga systemet som i sin tur ger ifrån sig utsignalen $y(t)$.

Notera att blocket "Regulator" i figur 3.1 är ett *tidsdiskret system*. Däremot är hela den samplande regulatorn – det streckade blocket i figur 3.1 – ett tidskontinuerligt system.

Vid sidan av datorbaserade system finns det flera andra exempel på samplande regulatorer. Reglering av ekonomiska system är ett typiskt exempel. Här baseras reglerbesluten på data som till exempel samlas in månadsvis. Regleringreppen beslutas därefter vid regelbundet eller oregelbundet återkommande sammanträden. En av de viktigare reglervariablerna för Sveriges ekonomi, riksbankens reporänta, är ju typiskt en styckvis konstant funktion.

Ett annat fall av samplad reglering är då processens utsignal endast *kan* mätas vid tidsdiskreta ögonblick. Ett exempel på det är massakokning där en del viktiga mätvärden endast kan erhållas genom laboratorieanalys av stickprov från koket. Regleringreppen (till exempel kemikalietillsats eller temperatur) baseras därefter på analysresultaten.

I princip är det svårare att reglera ett system med en samplande regulator än med en tidskontinuerlig. Detta beror dels på att en samplande regulator bara förfogar över en delmängd av de signaler som en tidskontinuerlig regulator kan använda, nämligen styckvis konstanta signaler. Vidare har den samplande regulatorn inte fullständig information om systemets uppförande. Om samplingsintervallet (det vill säga tiden mellan samplingsögonblicken) är för långt, kan betydelsefulla saker hända mellan samplingsögonblicken som regulatorn inte ser och alltså inte kan åtgärda. Vid syntes av samplande regulatorer måste man också först bestämma samplingsintervallet. Detta innebär alltså en extra parameter att välja jämfört med kontinuerliga regulatorer. Instabilitet i återkopplade system beror på att man "litar för mycket på för gammal information". Samplande regulatorer använder sig av information där de nyaste inslagen kan vara upp till ett samplingsintervall gamla. Risken för instabilitet i system som regleras med samplande regulatorer kan alltså vara större, speciellt om samplingsintervallet är långt.

Den viktigaste fördelen med samplande regulatorer är att det är enklare och billigare med datorimplementering. Man har heller inga principiella inskränkningar i karaktären på reglermekanismerna. Olinjäriteter och villkor av olika slag kan lätt införas. En annan fördel är att system som innehåller tidsfördröjningar lättare hanteras med samplad reglering. Om effekten av en insignal inte syns i utsignalen förrän efter en viss tid bör man minnas vilken insignal som drivit systemet under denna tid för att få god reglering. Vid kontinuerlig reglering kan man inte exakt minnas ett tidsförlopp. Men med samplad reglering är ju signalen styckvis konstant, och man kan därmed minnas den genom att lagra ett ändligt antal tal.

Tekniska och fysikaliska system är oftast till sin natur tidskontinuerliga. Det ter sig därför i regel naturligtast att analysera dem som sådana och också beräkna tidskontinuerliga regulatorer för dem. Regulatorerna blir därmed typiskt differentialekvationer (linjära eller olinjära) som matas med y och r . Dessa kan inte exakt realiseras i en samplande regulator, utan differentialekvationerna diskretiseras på ett eller annat sätt, så att en tidsdiskret regleralgoritm erhålls.

Ibland sker samplingen av data i den samplande regulatorn så snabbt att den

tidsdiskreta algoritmen för allt praktiskt bruk beter sig som sin tidskontinuerliga förebild. Ibland är diskrepansen så stor att man måste ha speciell metodik för att analysera vad som händer då en samplande regulator används. Det finns också teknik för att direkt beräkna en samplande regulator som ger det slutna systemet önskade egenskaper.

3.2 Approximation av tidskontinuerliga regulatorer

Vi ska i detta kapitel studera hur man bestämmer en tidsdiskret regulator, det vill säga hur man "översätter" en tidigare designad tidskontinuerlig regulator till en tidsdiskret motsvarighet.

Utgångspunkten är att vi, ur datorns synvinkel, tar emot de tidsdiskreta signalerna

$$y(kT_S), \quad k = 0, 1, \dots$$

och

$$r(kT_S), \quad k = 0, 1, \dots$$

och med hjälp av dessa ska bestämma en sekvens av styrsignalvärden

$$u(kT_S), \quad k = 0, 1, \dots$$

För att möjliggöra en systematisk behandling av hur man överför tidskontinuerliga regulatorer till tidsdiskret form ska vi här använda derivations- respektive förskjutningsoperatorer för att beskriva de olika regulatorerna. Derivationsoperatoren p definieras enligt

$$pu(t) = \dot{u}(t) \tag{3.1}$$

medan förskjutningsoperatoren q^{-1} definieras enligt

$$q^{-1}u(kT_S) = u(kT_S - T_S) \tag{3.2}$$

Om vi använder en differenskvot som approximation av derivata får vi

$$\dot{u}(t) = pu(t) \approx \frac{1}{T_S}(u(t) - u(t - T_S)) = \frac{1}{T_S}(1 - q^{-1})u(t) \tag{3.3}$$

Vi ser alltså att en tidskontinuerlig regulator kan tidsdiskretiseras genom att göra variabelbytet

$$p = \frac{1}{T_S}(1 - q^{-1}) \tag{3.4}$$

Denna metod brukar kallas *Eulers metod* (bakåt). Vi noterar att samma notation och strategi fungerar på integraler. Om vi har

$$h(t) = \int_0^t e(\tau) d\tau \tag{3.5}$$

så kan vi alternativt skriva detta som $\dot{h}(t) = e(t)$, dvs. $ph(t) = e(t)$ eller som $h(t) = \frac{1}{p}e(t)$. Insättning av tidsdiskretiseringen ger

$$h(t) = \frac{T_s}{1 - q^{-1}}e(t) \quad (3.6)$$

vilket ger oss en enkel rektangelapproximation av integralen

$$h(kT_s) = h(kT_s - T_s) + T_s e(kT_s) \quad (3.7)$$

Exempel: Approx. av PI-regulator med Eulers metod

En PI-regulator beskrivs av

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right) = K \left(1 + \frac{1}{T_i p} \right) e(t) \quad (3.8)$$

Genomförs variabelbytet enligt (3.4) får vi

$$(1 - q^{-1})u(t) = K \left((1 - q^{-1}) + \frac{T_s}{T_i} \right) e(t),$$

det vill säga

$$u(kT_s) = u(kT_s - T_s) + K \left(1 + \frac{T_s}{T_i} \right) e(kT_s) - K e(kT_s - T_s) \quad (3.9)$$

Eulers metod (3.4) är ett enkelt och rättframt sätt att approximera en derivata men den är dock inte den ur stabilitetssynvinkel bästa metoden. En bättre approximation fås istället genom att göra variabelbytet

$$p = \frac{2}{T_s} \frac{(q-1)}{(q+1)} = \frac{2}{T_s} \frac{(1-q^{-1})}{(1+q^{-1})} \quad (3.10)$$

Denna metod brukar kallas *bilinjär transformation*, *trapetsregeln* eller *Tustins formel*.

Exempel: Approx. av PI-regulator med Tustins formel

Genomförs variabelbytet enligt (3.10) i ekvation (3.8) fås

$$\frac{2}{T_s} \frac{(1-q^{-1})}{(1+q^{-1})} u(kT_s) = K \left(\frac{2}{T_s} \frac{(1-q^{-1})}{(1+q^{-1})} e(kT_s) + \frac{1}{T_i} e(kT_s) \right)$$

det vill säga

$$(1 - q^{-1})u(kT_S) = K \left((1 - q^{-1})e(kT_S) + \frac{T_S}{2T_i}(1 + q^{-1})e(kT_S) \right)$$

vilket medför

$$u(kT_S) = u(kT_S - T_S) + K \left(1 + \frac{T_S}{2T_i} \right) e(kT_S) - K \left(1 - \frac{T_S}{2T_i} \right) e(kT_S - T_S) \quad (3.11)$$

Denna differensekvation kan jämföras med resultatet av Eulers metod (3.9). Differensekvationerna har samma ordningstal men koefficientvärdena blir olika beroende på vilken översättningsmetod som används.

Av åskådlighetsskäl studerade vi i exemplet översättning av en PI-regulator, men Eulers respektive Tustins metod är naturligtvis tillämpliga även på regulatorer av högre ordning. Den resulterande tidsdiskreta regulatorn ger i de flesta fall ett slutet system med egenskaper som ligger nära dem som skulle erhållits med den ursprungliga tidskontinuerliga regulatorn. Ett villkor är dock att samplingen är ”snabb”, typiskt cirka 10 gånger det slutna systemets bandbredd (den snabbaste frekvens som det slutna systemet kan följa).

Om den regulator som ska översättas är av högre ordning blir översättning med hjälp av Tustins regel tämligen arbetsam att genomföra. Man kan då med fördel använda något datorhjälpmedel, och vi ska här i ett exempel visa hur översättningen kan göras med MATLAB och CONTROL SYSTEM TOOLBOX.

Exempel: Översättning av PI-regulator i Matlab

Låt oss utgå från PI-regulatorn

$$u(t) = 5 \left(e(t) + \frac{1}{0.5} \int_0^t e(\tau) d\tau \right)$$

vilken har överföringsfunktionen

$$F(s) = \frac{5s + 10}{s},$$

det vill säga överföringsoperatoren

$$F(p) = \frac{5p + 10}{p}$$

PI-regulatorn representeras som ett så kallat LTI-objekt i MATLAB med funktionen `tf` enligt

```
>> F=tf([5 10],[1 0])
```

```
Transfer function:
```

```
5 s + 10
```

```
-----
```

```
    s
```

Antag nu att vi vill översätta PI-regulatorn till tidsdiskret form med Tustins regel och med samplingsintervallet $T_S = 0.1$ sekunder. Detta kan göras med funktionen `c2d` där vi som argument anger regulatorn, samplingstiden samt vilken översättningsmetod som ska användas, enligt

```
>> Fd=c2d(F,0.1,'tustin')
```

```
Transfer function:
```

```
5.5 z - 4.5
```

```
-----
```

```
    z - 1
```

```
Sampling time: 0.1
```

Den tidsdiskreta regulatorn ges alltså av den tidsdiskreta överföringsoperatorm (ersätt helt enkelt z i resultatet ovan med q)

$$F_d(q) = \frac{5.5q - 4.5}{q - 1} = \frac{5.5 - 4.5q^{-1}}{1 - q^{-1}}$$

Detta motsvarar differensekvationen

$$u(kT_S) = u(kT_S - T_S) + 5.5e(kT_S) - 4.5e(kT_S - T_S)$$

vilken kan jämföras med det allmänna uttrycket i ekvation (3.11).

3.3 Integratoruppvidning i PID-regulatorer

Vi ska nu detaljstudera tidsdiskret implementering av PID-regulatorer och peka på några av de möjligheter som en sådan implementation medför. Först ska vi betrakta vad som händer då styrsignalen mätts.

Om man tillämpar Eulers metod på en tidskontinuerlig PID-regulator innebär det att man använder sig av approximationerna

$$\int_{t_0}^t e(\tau) d\tau \approx T_S (e(t) + e(t - T_S) + e(t - 2T_S) + \dots) \quad (3.12)$$

och

$$\frac{de(t)}{dt} \approx \frac{e(t) - e(t - T_S)}{T_S} \quad (3.13)$$

där T_S är samplingsintervallet. Om vi låter e_k och u_k beteckna värdena vid den k :te samplingsstidpunkten får vi alltså följande ekvation för den tidsdiskreta PID-regulatorn,

$$u_k = K \left(e_k + \frac{T_S}{T_i} \sum_{j=0}^k e_j + T_d \frac{e_k - e_{k-1}}{T_S} \right). \quad (3.14)$$

(Verifiera gärna, till exempel genom att bilda $u_k - u_{k-1}$, att denna version av den tidsdiskreta PID-regulatorn är ekvivalent med den i (3.9) när $T_d = 0$.) Om vi låter S_k beteckna värdet på summan i (3.14) ser vi att PID-regulatorn kan implementeras genom att följande steg utförs i varje samplingsstidpunkt

$$S_k = S_{k-1} + \frac{T_S}{T_i} e_k \quad (3.15a)$$

$$v_k = K \left(e_k + S_k + \frac{T_d}{T_S} (e_k - e_{k-1}) \right) \quad (3.15b)$$

$$u_k = \begin{cases} u_{\max} & \text{om } v_k > u_{\max} \\ v_k & \text{om } u_{\min} \leq v_k \leq u_{\max} \\ u_{\min} & \text{om } v_k < u_{\min} \end{cases} \quad (3.15c)$$

Här har vi också tagit hänsyn till att den styrsignal som kan ställas ut i praktiken alltid är begränsad. I detta fall betecknar vi den maximala och minimala styrsignalen med u_{\max} respektive u_{\min} . Man ser att två värden, S_k och e_k , måste sparas från ett samplingsstillfälle till nästa.

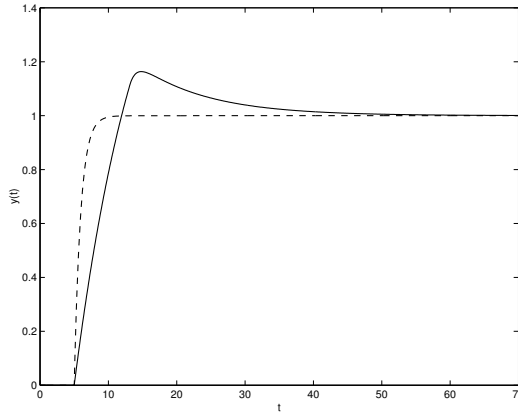
Det visar sig nödvändigt att modifiera den enkla PID-algoritmen (3.15) på flera sätt. En anledning till det är det besvärande uppförande I-delen uppvisar när styrsignalen mäts. Vi illustrerar detta med ett exempel.

Exempel: Integratoruppridning

Antag att vi reglerar systemet

$$G(s) = \frac{10}{10s + 1}$$

med en diskret PI-regulator enligt (3.15) med $T_S = 0.1$, $K = 1$, $T_i = 10$ och $T_d = 0$. Antag vidare att referenssignalen ändras plötsligt från 0 till 1.0 vid tiden $t = 5$ och att styrsignalen är begränsad så att den maximala amplituden är $u_{\max} = -u_{\min} = 0.2$. Man får då det utseende på utsignalen som visas i figur 3.3. Utsignalen får en stor översläng och det tar relativt lång tid innan den svänger in mot rätt värde. Orsaken till fenomenet är att det på grund av mättningen tar "onormalt" lång tid innan utsignalen nått referenssignalens nivå. Under denna tid integreras I-delen hela tiden upp och når en mycket hög nivå, se figur 3.4. Det tar lång tid efter det att reglerfelet skiftat tecken innan I-delen integrerats ner till ett lagom värde. Under denna tid kommer styrsignalen att ligga kvar på den maximala nivån, se figur 3.5



Figur 3.3: Effekten av integratoruppvridding: Den heldragna kurvan är den utsignal som erhålls då styrsignalen är begränsad till $|u(t)| \leq 0.2$. Den streckade kurvan är den utsignal som erhålls utan styrsignalbegränsning.

Fenomenet att I-delen under vissa betingelser summeras upp till ett mycket stort värde brukar kallas *integratoruppvridding* (eng. *reset¹ windup*).

Sätt att undvika integratoruppvridding

Det finns flera sätt att undvika integratoruppvridding. Vi ska här behandla två huvudmetoder:

- (i) Villkorlig integration.
- (ii) Justering av I-delen.

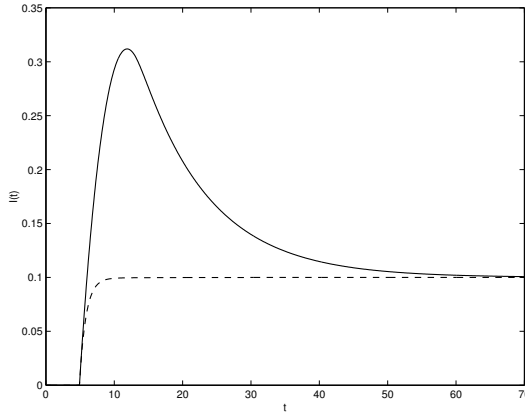
Som förberedelse inför de kommande metoderna skriver vi om vår tidsdiskreta PID-regulator (3.15) på följande sätt

$$I_k = I_{k-1} + K \frac{T_S}{T_i} e_k \quad (3.16a)$$

$$v_k = K e_k + I_k + K \frac{T_d}{T_S} (e_k - e_{k-1}) \quad (3.16b)$$

$$u_k = \begin{cases} u_{\max} & \text{om } v_k > u_{\max} \\ v_k & \text{om } u_{\min} \leq v_k \leq u_{\max} \\ u_{\min} & \text{om } v_k < u_{\min} \end{cases} \quad (3.16c)$$

¹Integraldelen brukar i engelsk litteratur ibland refereras till som "reset" eftersom den nollställer det stationära felet.



Figur 3.4: *I*-delen i regulatorn vid simuleringen i figur 3.3. Den streckade kurvan är den *I*-del som erhålls utan styrsignalbegränsning.

Som vi sett tidigare så leder denna implementering till integratoruppvridding då styrsignalen mättar. Ovanstående form är bättre än (3.15) men det kommer vi inte att undersöka närmare här.

Villkorlig integration

En lösning på problemet med integratoruppvridding är att stänga av uppdateringen av *I*-delen när styrsignalen mättar. Detta kan göras enligt följande algoritm

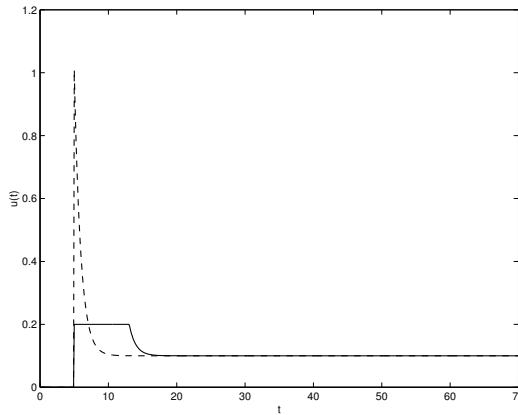
$$\text{if (VILLKOR) then } I_k = I_{k-1} + K \frac{T_S}{T_i} e_k \text{ else } I_k = I_{k-1}, \quad (3.17a)$$

$$v_k = K e_k + I_k + K \frac{T_d}{T_S} (e_k - e_{k-1}), \quad (3.17b)$$

$$u_k = \begin{cases} u_{\max}, & \text{om } v_k > u_{\max}, \\ v_k, & \text{om } u_{\min} \leq v_k \leq u_{\max}, \\ u_{\min}, & \text{om } v_k < u_{\min}, \end{cases} \quad (3.17c)$$

där $VILLKOR = u_{\min} < v_k < u_{\max}$. Figur 3.6 illustrerar hur detta förfarande påverkar uppförandet hos reglersystemet i det tidigare studerade exemplet. Vi ser att vi nu har eliminerat överslängen genom att inte uppdatera *I*-delen i tidsintervallet $5 \leq t \leq 10$ då styrsignalen mättas. Invängningsförloppet blir dock långsammare på grund av mättningen.

Villkorlig integration är dock även användbart sett ur en annan synvinkel. Vi såg i exemplet i figur 4.4 att *I*-delen tog bort det stationära felet till priset av ett mer oscillativt insvängningsförlopp. I figur 3.7 (a) visas insvängningen hos en



Figur 3.5: Styrsignalen vid simuleringen i figur 3.3 (heldragen). Den streckade kurvan är den styrsignal som erhålls utan styrsignalbegränsning.

PI-regulator tillsammans med integraldelen S . Man ser att det är den kraftiga ökningen av S under första fasen som ger upphov till en stor del av svängigheten. För att mildra denna nackdel kan man låta I-delen vara urkopplad om felet ligger över en viss nivå, det vill säga låta villkoret ges av $VILLKOR = |e| \leq \epsilon$ med ϵ en positiv konstant. Det blir då bara den avslutande delen av insvängningsförloppet som påverkas, vilket ger ett lugnare uppförande, se figur 3.7 (b). Naturligtvis måste det område där I-delen är inkopplad vara tillräckligt stort för att utsignalen ska kunna föras dit med enbart P-reglering.

Justering av I-delen

I stället för att stänga av integratorn, det vill säga sätta $I_k = I_{k-1}$, kan man försöka låta den anpassa sig till den uppkomna situationen.

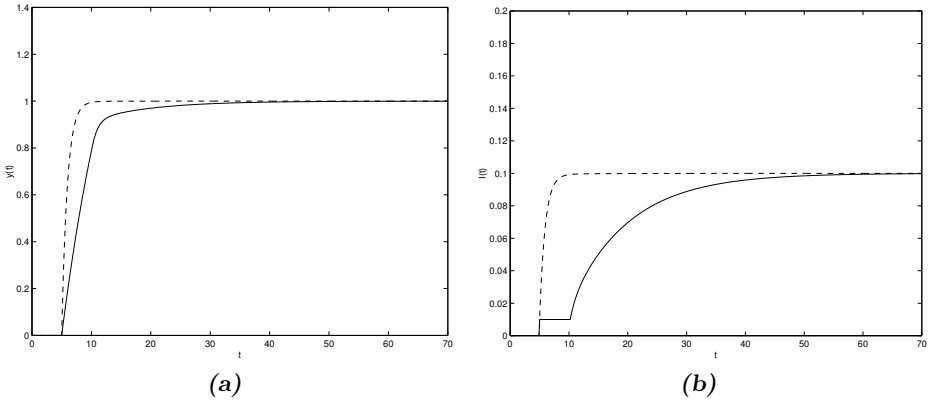
Om man vill använda en regulator på formen (3.15) men vill undvika integratoruppvridning, så kan man göra på följande sätt. När styrsignalen gått i en begränsning, det vill säga då u_k och v_k skiljer sig åt, så kan man reducera storleken på I_k genom tilldelningen

$$I_k := I_k + \frac{T_S}{T_t}(u_k - v_k). \quad (3.18)$$

Hela algoritmen blir således

$$I_k = I_{k-1} + K \frac{T_S}{T_i} e_k \quad (3.19a)$$

$$v_k = K e_k + I_k + K \frac{T_d}{T_S} (e_k - e_{k-1}) \quad (3.19b)$$



Figur 3.6: Användning av villkorlig integration: (a) Utsignalen. (b) I-delen.

$$u_k = \begin{cases} u_{\max}, & \text{om } v_k > u_{\max}, \\ v_k, & \text{om } u_{\min} \leq v_k \leq u_{\max}, \\ u_{\min}, & \text{om } v_k < u_{\min}, \end{cases} \quad (3.19c)$$

$$I_k := I_k + \frac{T_S}{T_t}(u_k - v_k) \quad (3.19d)$$

Parametern T_t i (3.18) och (3.19) kallas *tracking-konstant* och kan väljas på följande sätt:

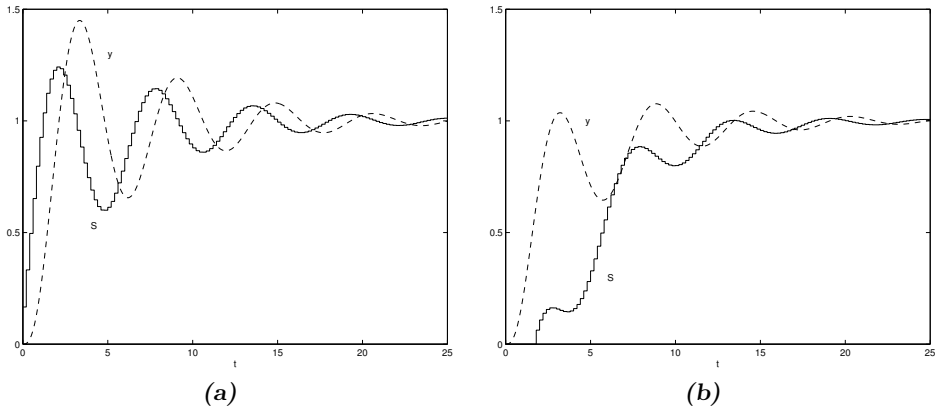
Momentan justering: Väljer man $T_t = T_S$ kommer I_k vid mättning att bero bara på u_{\max} eller u_{\min} samt på P- och D-delarnas bidrag till v_k (det vill säga på e_k och på e_{k-1}). Om v_k innehåller störningar kan det innebära att man överkompenserar.

Följemod (Tracking): Andra värden på T_t ($T_t > T_S$) ger ett mjukare beteende, och man talar då om följemod (tracking) för integratorn. En tumregel är att använda $T_t = T_i$, det vill säga att låta T_t vara lika med integraltiden. Denna tumregel kan vara fast inlagd som skönsvärde.

Figur 3.8 illustrerar hur modifiering av I-delen påverkar uppförandet hos regelsystemet i det tidigare studerade exemplet.

3.4 Modifiering av D-delen

Den rättframma differensapproximation av derivatadelen som gjordes i ekvation (3.13), har bland annat nackdelen att brus i mätsignalen förstärks kraftigt när



Figur 3.7: Reglering med PI-regulator: (a) Normal integration. (b) Villkorlig integration. ($I_k = I$ -delen)

samlingsintervallet T_S är litet. I figur 3.9 (a) visas en liknande simulering som i figur 4.4 (c), men med en diskret PID-regulator och med brus adderat till mätsignalen $y(t)$. Som figuren visar slår mätbruset igenom kraftigt, särskilt i styrsignalen $u(t)$. För tidskontinuerliga regulatorer löser man problemet genom att inte utföra en exakt derivation med överföringsfunktionen $T_d s$, utan i stället en approximativ med överföringsfunktionen

$$\frac{T_d s}{\mu T_d s + 1} \quad (3.20)$$

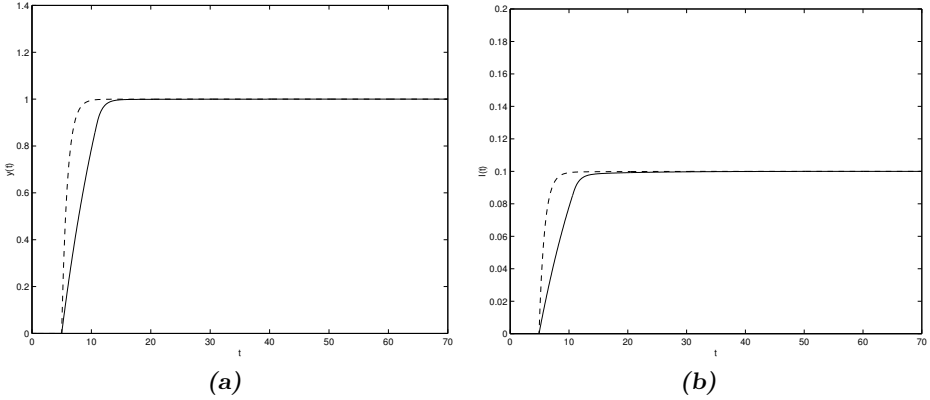
Detta uttryck är ungefär lika med s för små värden på s och ungefär lika med $1/\mu$ för stora värden på s . Följden blir att signaler som huvudsakligen innehåller låga frekvenser deriveras nästan exakt, medan höga frekvenser bara förstärks med faktorn $1/\mu$ (för den exakta derivationen går förstärkningen mot oändligheten när frekvensen går mot oändligheten). Genom att μ görs litet kan man få en god derivation högt upp i frekvens till priset av att högfrekvent brus inte längre undertrycks så bra.

Det enklaste sättet att åstadkomma motsvarande effekt i en samplande regulator är att göra differensapproximationer av alla derivator. Om d betecknar den approximativt deriverade signalen, så gäller

$$D(s) = \frac{s}{1 + s\mu T_d} E(s), \quad (3.21)$$

eller i tidsplanet

$$\mu T_d \dot{d} + d = \dot{e}. \quad (3.22)$$



Figur 3.8: Användning av modifiering av I-delen med $T_t = T_i = 10$: (a) Utsignalen. (b) I-delen.

Med differensapproximation fås

$$\mu T_d \frac{d_k - d_{k-1}}{T_S} + d_k = \frac{e_k - e_{k-1}}{T_S} \quad (3.23)$$

eller

$$d_k = \frac{\mu T_d}{\mu T_d + T_S} d_{k-1} + \frac{1}{\mu T_d + T_S} (e_k - e_{k-1}). \quad (3.24)$$

Detta värde sätts sedan in i uttrycket för v_k i algoritmen för PID-regulatorn (3.16), det vill säga

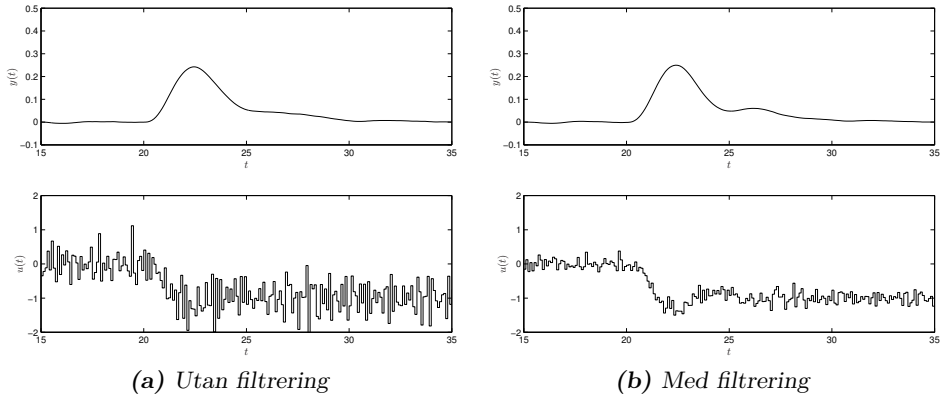
$$v_k = K e_k + I_k + K T_d d_k. \quad (3.25)$$

I figur 3.9 (b) visas samma simulering som i 3.9 (a) men med en regulator där D-delen filtrerats ($\mu T_d = 1/5$). Man ser att man får ungefär samma prestanda fast med lägre amplitud på styrsignalen.

3.5 Regulatorkod för PID-regulatorer

Det sista steget i en diskret PID-design är att implementera regulatorn i något lämpligt högnivåspråk såsom Java, C eller Ada. Det kan vara lämpligt att strukturera regulatorkoden enligt

1. Läs insignaler
2. Beräkna styrsignal
3. Ställ ut styrsignal



Figur 3.9: Inverkan av brus för PID-regulator med D -del. Överst mätsignal, underst styrsignal.

4. Uppdatera tillstånd
5. Vänta

Därvid lägger man så få beräkningar som möjligt i "Beräkna styrsignal" för att minimera beräkningsfördröjningar. Ytterligare en effektivitetshöjande åtgärd är att skriva om regulatorn så att allt som inte beror på senaste mätningen är förberäknat.

3.6 Allmänna regulatorer

Om man vill använda en tidsdiskret approximation av en mer generell regulatoralgoritm än PID så måste man ta hänsyn till åtminstone de krav som behandlats i de senaste avsnitten, det vill säga hantera eventuella problem med integratorupprivning och rena deriveringar. Emellertid kan det också tillkomma krav. Ett exempel är att det kan uppkomma numeriska problem.

Kapitel 4

Inställningsregler för PID-regulatorer

Inställning av regulatorer sker i praktiken ofta helt ”*ad hoc*”-mässigt. Detta gäller i synnerhet PID-regulatorer, som är den i processindustrin helt dominerande regulatortypen. Även om erfarna installatörer med god intuition för den reglerade processen ofta kan göra ett bra jobb utan någon mer systematisk inställningsmetod är det tveklöst så att en sådan metod ofta skulle vara önskvärd. Det är till exempel lätt att konstatera att det finns dåligt inställda PID-regulatorer på många håll.

Ett mer systematiskt sätt att ställa in regulatorer kan till exempel vara att göra något experiment med processen och sedan justera regulatorparametrarna baserat på experimentets utfall. Det är praktiskt att tänka sig att man bygger en enkel modell av systemet med utgångspunkt från experimentet och sedan gör en reglerdesign baserad på modellen i fråga. Vi ska först beskriva ett par modeller som används i dessa sammanhang och hur man kan skatta deras koefficienter från mätdata. I avsnitt 4.4 återkommer vi till hur modellparametrarna kan omräknas till PID-inställningar, men modellerna kan självklart även användas för inställning av andra typer av regulatorer.

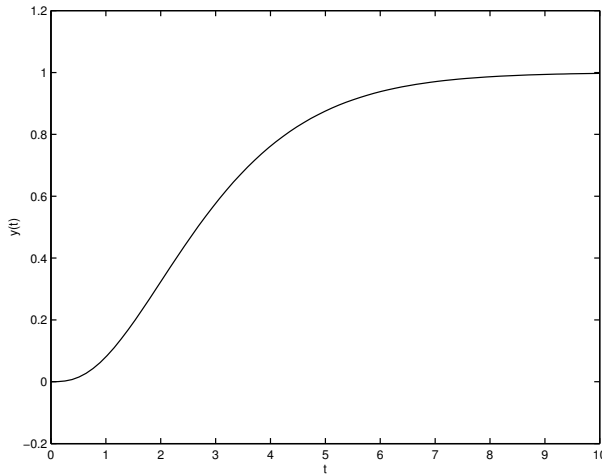
4.1 Processmodeller

Om signalen ändras som ett steg kallas processens utsignal *stegsvar*. Ett typiskt stegsvar från en industriell process visas i figur 4.1. Vi ser tre karakteristiska drag:

- Det tar en viss tid innan stegsvaret börjar ”röra på sig”. Detta kan bero på en ren tidsfördröjning (*dödtid*) i processen, till exempel orsakad av en transportfördröjning, eller på att utsignalens derivata är noll i början av steget. (Detta är alltid fallet om antalet poler överstiger antalet nollställen

med två eller mer.)

- Stegsvaret stabiliserar sig så småningom på en viss nivå, svarande mot processens *statiska förstärkning*.
- Det tar en viss tid för stegsvaret att efter den ursprungliga tidsfördröjningen komma nära fortfarighetsnivån. Denna tid kallas (den dominerande) *tidskonstanten*.



Figur 4.1: Ett typiskt stegsvar från en process.

I regel är inte förloppet oscillativt. Detta är sant för typiska system i processindustrin, men gäller naturligtvis inte generellt. Till exempel har ju mekaniska system med resonanser oscillativa stegsvar. De slutna systemen kan förstås också bli oscillativa på grund av återkopplingen.

Egenskaperna ovan kan naturligtvis beskrivas på flera sätt. Vi ska här studera två olika möjligheter:

Treparametermodellen: De mest typiska dragen i stegsvaret i figur 4.1 fångas av den enkla processmodellen

$$G(s) = \frac{K_p}{1 + sT} e^{-sL} \quad (4.1)$$

där L är tidsfördröjningen, T tidskonstanten och K_p den statiska förstärkningen. Denna treparametermodell har också fördelen att den har lika många parametrar/frihetsgrader som PID-regulatorn.

Ziegler-Nichols modell: En enklare modell, som bara innehåller två parametrar, och som ligger till grund för Ziegler-Nichols inställningsregler (se avsnitt 4.4.1), är

$$G(s) = \frac{b}{s} e^{-sL} \quad (4.2)$$

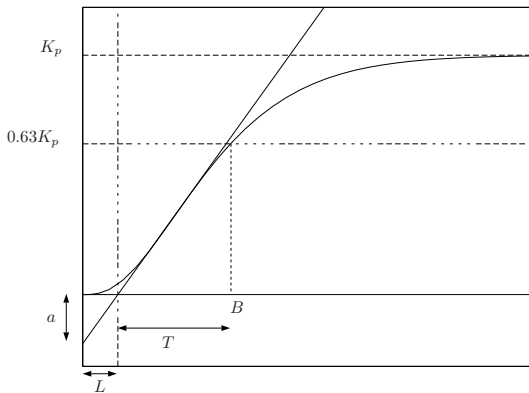
Detta är alltså en tidsfördröjning, följt av en ren integration. Enligt denna modell är den statiska förstärkningen oändlig (eller ointressant) och den intressanta informationen är lutningen hos stegsvaret.

4.2 Transientförsök

Ett rättframt sätt att anpassa treparametermodellen (4.1) är att genomföra ett stegsvarsexperiment: Låt u vara ett steg och mät y . Skala y så att det svarar mot ett enhetssteg i u , det vill säga

$$u(t) = \begin{cases} 0, & t < 0, \\ 1, & t \geq 0. \end{cases}$$

Om processen verkligen beskrivs av (4.1) blir då



Figur 4.2: Bestämning av modellparametrar ur ett stegsvar. Den räta linjen är en tangent som beskriver stegsvarets brantaste lutning.

$$y(t) = \begin{cases} 0, & t < L, \\ K_p(1 - e^{-(t-L)/T}), & t \geq L, \end{cases} \quad (4.3)$$

och parametrarna K_p och L kan bestämmas som i figur 4.2. Tiden $t = T + L$ är den tid då $y(t) = (1 - e^{-1})K_p = 0.6321K_p$, det vill säga punkten B i figuren. Alternativt, K_p/T är brantaste lutningen hos stegsvaret (som för (4.1) inträffar vid $t = L$). Kvoten a/L i figuren kan vara ett bra mått på denna lutning.

För *Ziegler-Nichols* modell (4.2) anpassar man liksom i figur 4.2 en tangent med brantaste lutningen hos stegsvaret och bestämmer L som i figuren. Lutningen b ges av

$$b = \frac{a}{L}. \quad (4.4)$$

4.3 PID-regulatorn

PID-regulatorn har använts och utvecklats under minst 250 år och är idag den vanligaste regulatorn i industriella tillämpningar¹.

En PID-regulator beskrivs i princip av uttrycket

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right), \quad (4.5)$$

vilket med hjälp av laplacetransform motsvarar

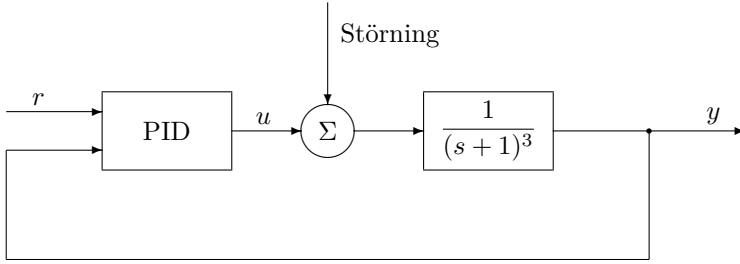
$$U(s) = K \left(1 + \frac{1}{T_i s} + T_d s \right) E(s) \quad (4.6)$$

Parametern K kallas *förstärkningen*, T_i och T_d har dimension tid och kallas *integraltid* respektive *derivatetid*. Grundprincipen för en PID-regulator illustreras i följande exempel.

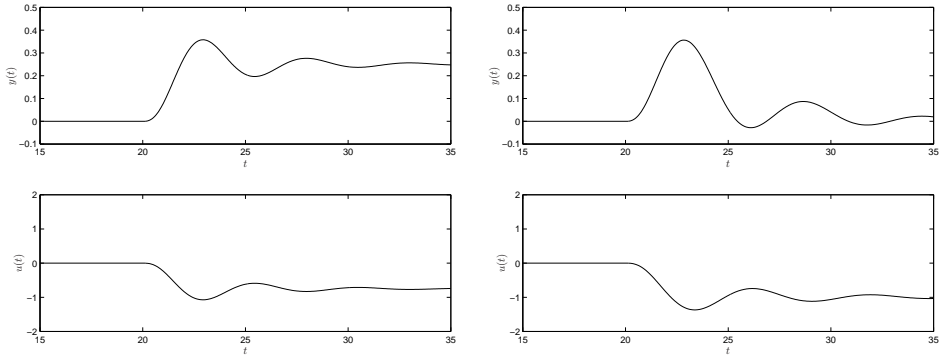
Exempel: PID-regulatorns funktion

För att repetera funktionen hos en PID-regulator ska vi titta på det exempel som visas i figur 4.3. Systemet utsätts för en stegstörning på ingången enligt figuren. Störningen kommer vid tidpunkten $t = 20$. I figur 4.4 (a) visas simuleringsresultatet vid P-reglering. Som synes får man ett stationärt fel. Detta kan minskas genom att K ökas, men då blir i gengäld överslängen större. Om I-delen kopplas in fås det svar på störningen som visas i figur 4.4 (b), det vill säga reglerfelet avtar mot noll. Man ser också att systemet visar ett mera oscillativt uppförande. Slutligen visas i figur 4.4 (c) vad som händer om man använder en fullständig PID-regulator. Den stabiliserande verkan som införandet av D-termen ger syns där tydligt genom att praktiskt taget alla svängningar i utsignalen har försvunnit.

¹Det sägs skämtsamt att PID egentligen står för "process industry default controller".

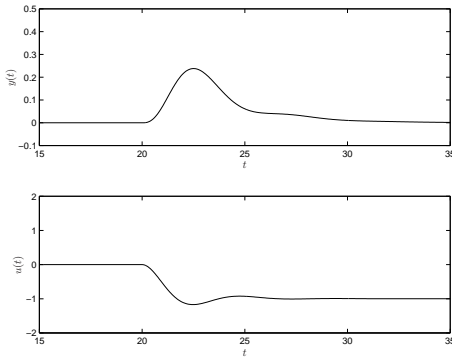


Figur 4.3: Reglerkrets med PID-regulator.



(a) $K = 3, T_i = \infty, T_d = 0$

(b) $K = 2.5, T_i = 3, T_d = 0$



(c) $K = 3.5, T_i = 3, T_d = 0.5$

Figur 4.4: Utsignalen (övre kurvorna) och insignalen (undre kurvorna) vid P-, PI-, och PID-reglering.

4.4 Inställningsmetoder

Att konstruera en regulator för ett system innebär många kompromisser, och det finns ingen ”bästa metod”. Man ska alltså inte vänta sig att det ska finnas ett unikt sätt att välja PID-koefficienter för ett system, vare sig det ges som en sofistikerad matematisk modell eller som en enkel approximativ beskrivning enligt avsnitt 4.1. I detta avsnitt ska vi anta att vi genom ett enkelt experiment tagit fram en grov modell av treparameter-typ,

$$G(s) = \frac{K_p}{1 + sT} e^{-sL}, \quad (4.7)$$

eller en modell av Ziegler-Nichols typ,

$$G(s) = \frac{b}{s} e^{-sL}. \quad (4.8)$$

Vi ska härnäst beskriva två olika föreslagna metoder att omsätta dessa modeller i P-, PI- eller PID-koefficienter.

4.4.1 Ziegler-Nichols regler

Ziegler & Nichols beskrev år 1942 en uppsättning inställningregler för PID-regulatorer som har kommit att bli mycket använda. Vi beskriver dem här inte i första hand för att de är de bästa reglerna, utan för deras stora spridning. De har dessutom kommit att stå modell för flera andra regler som vi behandlar senare i kapitlet. Reglerna togs fram genom ganska omfattande simuleringar av PID-reglering av olika system. Resultatet blev en enkel strategi: Genomför ett stegsvarsexperiment enligt beskrivningen i avsnitt 4.2 och bestäm b och L i (4.8). Välj sedan PID-koefficienter enligt tabell 4.1. Erfarenheten säger dock att Ziegler-Nichols regler ofta föreslår något för hög förstärkning.

Tabell 4.1: PID-inställning enligt Ziegler-Nichols för processmodellen (4.8).

Regulator	K	T_i	T_d
P	$1/(bL)$		
PI	$0.9/(bL)$	$3L$	
PID	$1.2/(bL)$	$2L$	$L/2$

4.4.2 Lambdatrimning

En vanligt förekommande regulatorinställningsmetod inom processindustrin är *lambdatrimning*. Idén är att specificera regulatorn i termer av parametrarna T_c och T , det vill säga det önskade slutna respektive det öppna systemens tidskonstanter.

Mer specifikt definierar man parametern λ som kvoten mellan dessa storheter, det vill säga

$$\lambda = \frac{T_c}{T}, \quad (4.9)$$

där $\lambda < 1$ ger ett snabbare slutet system och $\lambda > 1$ ger ett långsammare dito. Normalt väljer man λ i intervallet $0.5 \leq \lambda \leq 5$. Frågan är förstås hur man kan bestämma regulatorn från sin modell och λ ? Om man reglerar ett system $G(s)$ med en regulator $F(s)$ med en frihetsgrad blir det slutna systemet

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

Om man löser ut $F(s)$ som funktion av $G(s)$ och $G_c(s)$ erhålls

$$F(s) = \frac{G_c(s)}{G(s)(1 - G_c(s))}. \quad (4.10)$$

Vid lambdatrimning utgår man ofta från treparametermodellen (4.7) och vi tittar först på fallet $L = 0$. Vi vill (enligt (4.9)) att det slutna systemet ska uppfylla

$$G_c(s) = \frac{1}{1 + sT_c} = \frac{1}{1 + s\lambda T} \quad (4.11)$$

Om (4.7) och (4.11) stoppas in i (4.10) erhålls

$$F(s) = \frac{1 + sT}{K_p s \lambda T} = \frac{1}{K_p \lambda} \left(1 + \frac{1}{Ts} \right) \quad (4.12)$$

det vill säga en PI-regulator. När modellen innehåller en tidsfördröjning måste även det önskade återkopplade systemet innehålla denna fördröjning, det vill säga det återkopplade systemet specificeras som

$$G_c(s) = \frac{1}{1 + s\lambda T} e^{-sL} \quad (4.13)$$

Insatt i (4.10) ger detta

$$F(s) = \frac{1 + sT}{K_p (s\lambda T + 1 - e^{-sL})}. \quad (4.14)$$

Denna regulator kan även skrivas på insignal-utsignalform som

$$U(s) = \frac{1}{\lambda K_p} \left(1 + \frac{1}{sT} \right) \left(E(s) - \frac{K_p}{1 + sT} (1 - e^{-sL}) U(s) \right). \quad (4.15)$$

Med hjälp av approximationen

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

får man regulatoren

$$F(s) \approx \frac{1 + sT}{K_p(s\lambda T + 1 - (1 - sL))} = \frac{T}{K_p(\lambda T + L)} \left(1 + \frac{1}{Ts}\right), \quad (4.16)$$

Denna regulator ger direkt en metod för att välja regulatorparametrarna. Vid lambdatrimning av en PI-regulator brukar man därför använda inställningsregeln

$$\boxed{K = \frac{T}{K_p(\lambda T + L)}, \quad T_i = T} \quad (4.17)$$