MALMÖ HÖGSKOLA

Inbyggda system och signaler

Styr- och reglerteknik

Labbinlämning 1404f

Utlämning:	2 mars 2015
Deadline inlämning:	20 mars 2015, kl. 18.00

Namn: Matko Scapec-Kukina

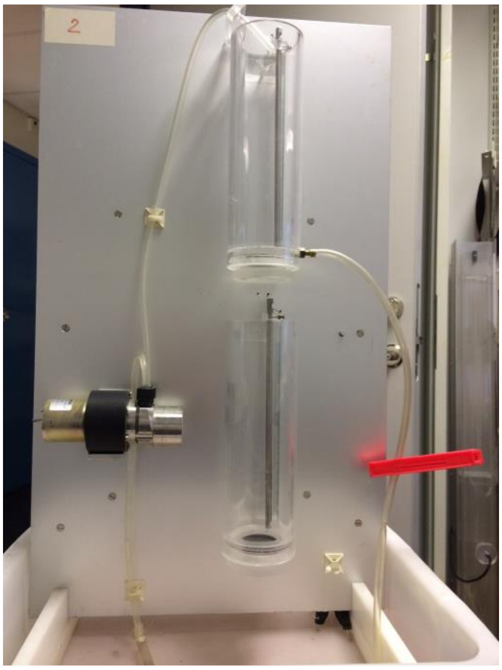
Namn: Ali Rama

Gion Koch Svedberg

februari 2015

Klassisk reglerteknik

Syftet med denna laboration är att praktiskt tillämpa stegsvar samt olika klassiska och avancerade regleralgoritmer på ett fysikaliskt system och att matematisk analysera resultat enligt teorin. Vi använder oss av en vattenmodell med två behållare där vattnet pumpas in i den första tank och därifrån rinner genom ett hål i botten in i den andra tanken, se bild 1.



Samma typ av vattenmodell är en klassisk process som används flitigt inom utbildning och forskning inom reglerteknik.

Bild 1: Kort av vattenmodellen som används i denna inlämningsuppgift.

Den teoretiska delen består av en förberedande och en analyserande del. Den förberedande delen ska göras innan den praktiska delen och består i en repetition av teorin från kursboken och framtagning av Matlabfunktionerna för de olika regulatorer som sedan ska köras för att reglera vattenmodellen.
Den analyserande teoridelen ska göras efter den praktiska delen då den använder sig av resultaten från era mätningar. Systemegenskaper som stigtid, insvängningstid och kvarstående fel som resultat av de olika regulatorerna ska diskuteras och jämföras. En enkel blackbox-systemidentifikation med hjälp av minsta kvadratmetoden ska också genomföras.

I den praktiska delen handlar uppgifterna om att experimentera med olika regleralgoritmer samt att tillämpa olika tumregel för inställningen av reglerparametrarna.
Resultat i form av plotts och mätserier ska tas fram för att sedan kunna analyseras och jämföras med varandra samt att diskutera för- och nackdelar av de olika regleralgoritmerna.

Innehållsförteckning och översikt

Table of Contents

Skriv inte ut detta dokument utan ha det öppet på datorn under laborationen och besvara frågorna direkt i dokumentet. Efter laborationen laddas dokumentet och utvalda filer upp på Its learning.

Läs hela uppgiften innan handledningstillfällen (och undervisningstillfällen). Lös de förberedande teoretiska uppgifterna innan du påbörjar med den praktiska delen i labbsalen! Läraren kan få vilja se att du har studerat teoridelen och ställa frågor om den som del av en effektiv handledning!

Inlämningen av detta fullständigt ifyllda dokument samt andra filer som ni ska generera för att dokumentera vissa delar av er lösning ska ske på its learning. **Ladda upp varje fil för sig, dvs inte komprimerade.** För videodokumentering kan länkar anges t.ex. till youtube eller andra lämpliga videotjänster.

Laborationen genomförs som vanligt i par dvs. ni jobbar två och två eller ensam. Vid inlämningen på Its learning anges vem som jobbat ihop. Forskningen visar att den mest effektiva inlärningen sker när man förklarar något till någon annan! Tillämpa det gärna på varandra i gruppen och i hela klassen för att få hjälp i att förstå vad som ska göras och varför. Själva laborationen blir dock meningslös om ni fuskar och bara kopierar varandras resultat eller formuleringar utan att själv har förstått vad ni skriver! Alla svar och alla programkod och mätresultat ska vara gruppens egen!! Labbinlämningsuppgifterna dokumenterar er inlärning i ämnet och om de genomförs seriöst har man uppnått lärandemålen och kommer att klara sluttentamen!

Dokument som ni behöver för att kunna lösa uppgifterna är kursboken, Matlabs ”help” och dokumentation samt material som finns upplagda på its learning.

Krav för godkänd

- Fullständigt ifyllt dokument (inkl namn på titelsida) med korrekta svar till alla frågor, uppladdad till its learning som word eller pdf-fil, (okomprimerad).
- Olika Matlabfunktioner (okomprimerade) uppladdad till its learning:
  - vm_openstep.m
  - vm_twostep.m
  - vm_P.m
  - vm_PI.m
  - vm_PID.m
  - vm_fuzzy.m
  - vm_kaskad.m
- Binär fil ”labb1404f.mat” med alla variabler från reglerexperimenten, uppladdad på its learning.
- Tiff-bilderna av plotten för stegsvaret, tvålägesreglering, P-reglering, PI-reglering, PID-reglering, (fuzzy- och/eller kaskadreglering) uppladdade på its learning.

A) TEORIDEL SOM FÖRBEREDNING

A.1 Vattenmodellen som enkel reglerkrets

Fördelen med den så kallade enkla reglerkretsen är att den är allmänt användbart, dvs mer eller mindre oberoende av själva processen som ska regleras.

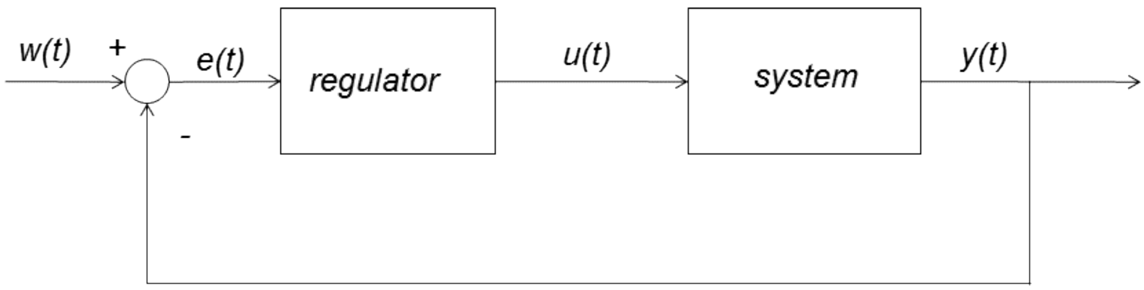


Fig A.1: blockdiagram av en enkel regelkrets

A.1.1 Begrepp i klassiska reglerkretsen

Ange begreppen till de olika signalerna i en klassisk reglerkrets enligt fig A.1:

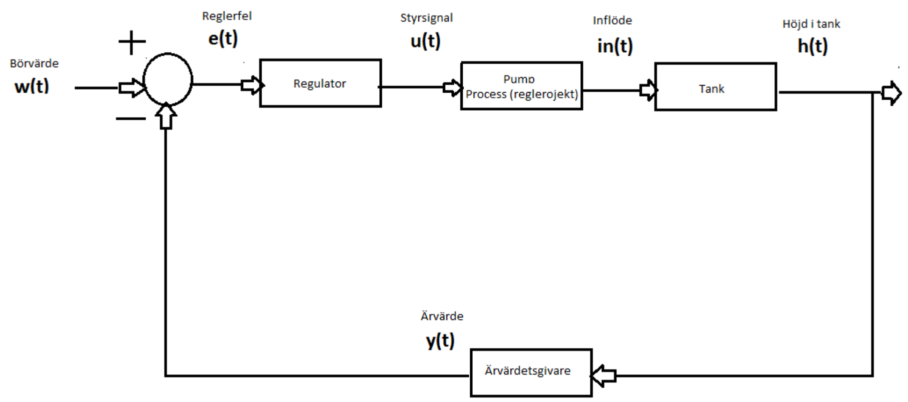
w(t): Börvärde – den reglerade storhetens önskade värde.
e(t): Reglerfel/Störning – en storhet som ger en oönskad påverkan på den reglerade storheten i ett reglersystem.
u(t): Styrsignal. – storhet som används för att påverka den process som ska regleras.
y(t): Ärvärde – den reglerade storhetens verkliga (aktuella) värde.
(Thomas B, 1997, s. 8)
Beskriv med egna ord vad som händer i blockdiagrammet:

Börvärdet w(t) och Ärvärdet y(t) är de värden vars differens bidrar till reglerfelet/störning vid differenspunkten i blockdiagrammet. Reglerfelet/störningen e(t) är den insignal som påverkar utsignalen u(t) d.v.s. styrsignalen för att justera för aktuella avvikelser. Styrsignalen u(t) är i sin tur den insignal som påverkar utsignalen y(t) d.v.s. anger det aktuella värdet efter justering.

1. Reglersystemet har i uppgift att kompensera för störningar e(t) så att dessa inte får för stor inverkan på den reglerade storheten (systemets utsignal) d.v.s. ärvärdet y(t). Regulatorn ska snabbt upptäcka avvikelser mellan börvärdet w(t) och reglerade storheten y(t) och vid behov justera styrsignalen så att avvikelsen försvinner.
2. Reglersystemet har i uppgift att följa ändringar i börvädet. Om börvärdet ändras ska regulatorn se till att systemets reglerade storhet svänger in sig till detta nya värde. (Thomas B, 1997, s. 11)

A.1.2 Blockdiagram av vattenmodellsregleringen

Rita det fullständiga blockschemat eller blockdiagrammet av vattenmodellen med regleringen samt störningar med korrekta benämningar och begrepp. (Det går bra att först rita på tavlan eller på pappret och sedan klistra in kortet).



A.2 Tillämpningsområden av vattenmodellen

Vattenmodellen används i undervisningen och i forskningen inom reglerteknik. Försök att ta reda på vad som gör vattenmodellen så användbar. (Se också sammanställningen av vetenskapliga artiklar på its learning som använder sig av vattenmodellen).

A.2.1 Exempel av tillämpningsområden som kan beskrivas med vattenmodellen
Ange tre olika tillämpningsområden som kan beskrivas med vattenmodellen. Beskriv vad i dina exempel som motsvarar motorstyrningen, pumpen, in- och utflöden, störningar, nivån i första och andra behållaren samt börvärden.

1. Exempel: Fukthaltsregleringen i pappersmaskin

I torkpartiet i en pappersmaskin kommer ett papper in med en fukthalt på 30-60 %. Pappret får löper runt två torkcylindrar som värms med ånga. Det är önskvärt att det utgående pappret har konstant fukthalt.

2. Exempel: Lagerhållning

Ett mellanlager tar emot varor från ett huvudförråd som sedan levererar varorna till kunder. Problemet för lagerinnehavaren är att alltid hålla ett bestämt antal varor i lagret och därmed att alltid kunna leverera till kunderna. Lagerinnehavaren anser sig att mellanlagret bör innehåll varor motsvarande minst 25 % vilket kan tolkas som ett *börvärde*. *Ärvärdet* är den aktuella lagerstatusen d.v.s. nuvarande volym av varor i lagret. Differensen mellan ovanstående värden är det värdet som regleras och vars resultat anger hur många varor (styrsignal) som bör beställas från en extern köpare till huvudförrådet för att sedan skickas vidare till mellanlagret. Den externa köparen kan tydas som *styrdon* . *In- och utflödet* kan beskrivas som volymen av varor som tas emot från den externa köparen till huvudförrådet och sedan vidare från huvudförrådet till mellanlagret samt det varor som tas emot från huvudförrådet till mellan lagret och det varor som säljs vidare från mellanlagret till kunderna.

3. Exempel: Det mänskliga ögat

Människans ögon fungerar genom att projicera bilder på en ljuskänslig näthinna (retina). Signalerna skickas därifrån till hjärnan via synnerven. Ögat har i syfte att släppa in lämplig ljusmängd in till näthinnan och de ljuskänsliga syncellerna, detta sker genom reglering av ljusmängden som passerar. Men ögat reglerar även brytningsförmågan som ger upphov till att vi kan kvarhålla ett skarpt seende på olika avstånd från ett objekt. (Thomas B, 1997, s. 23)

Översikt och jämförelse

	Exempel 1:	Exempel 2:	Exempel 3:
motorstyrning	Värmejustering av cylindrar	Antal beställda varor	Hjärnan
pumpen	Volymenhet av ånga	Antal skickade varor	Nervtrådarna till ögat
inflöde till första behållare	Volymenhet av ånga som når första cylindern	Antal varor som tas emot i huvudförrådet	Ljusinstrålning till näthinnan

utflöde ur första behållaren	Volymenhet av ånga som lämnar första cylindern och når andra cylindern	Antal varor från huvudförrådet till mellanlagret	Information från näthinnan till hjärnan
inflöde till andra behållaren	Volymenhet av ånga som når den andra cylindern	Antal varor som tas emot i mellanlagret	Ljusinstrålning till andra ögat
utflöde ur andra behållaren	Volymenhet av ånga som lämnar den andra cylindern	Antal varor från mellanlagret till kunder	Informationen från andra ögat till hjärnan
störningar	Värme som går förlorad i den andra cylindern	Antal varor sålda till kunder	Nervskador, partiklar i luften, svagt/stark ljus m.m.
nivån i första behållaren	Temperaturen i första cylindern	Antal varor i procentenheter i huvudförrådet	Pupillens anpassig sig efter ljusinstrålningen
nivån i andra behållaren	Temperaturen i andra cylindern	Antal varor i procentenheter i mellanlagret	Pupillen anpassar sig efter ljusinstrålningen
börvärde	Önskat temperatur i syfte att bibehålla konstant fuktighet	Önskat antal varor i procentenheter i mellanlagret	Ljusinstrålning

A.2.2 Gemensamma (system-) egenskaper av tillämpningsområdena

På vilket sätt liknar dina tre tillämpningsområden varandra? Finns det gemensamma typiska systemegenskaper?

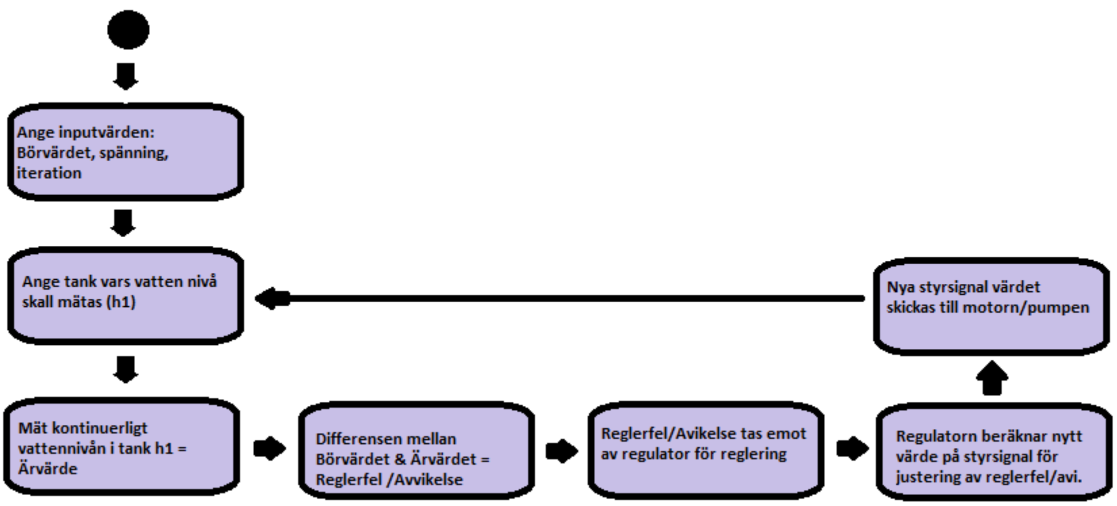
Samtliga exempel kan beskrivas med följande element. *System, omgivning och syfte.* Systemet karakteriseras av styrvariabler eller signaler som vi bestämmer men även mätsignaler eller utsignaler som bidrar med information om systemet.

A.3 Tidsdiskret reglering

I denna kurs fokuserar vi oss uteslutande på tidsdiskreta system. Det är därför viktigt att ha koll på de olika komponenter som ingår i ett digitalt reglersystem.

A.3.1 Flödesschema för datorns reglerprogram

Rita upp flödesscheman som Matlabprogrammet kommer att behöva följa för att reglera vattenmodellen. Ange vilka sensorer och givare det handlar om i de olika stegen inom flödet.



A.3.2 A/D-omvandling

a) Beskriv de karakteristiska, för vattenmodellen relevanta, egenskaperna av en A/D-omvandling på en Arduino Uno.

Atmega 238 styrenheten som används på Arduino Uno brädan innehåller 6 avsedda kanaler för A/D omvandling. A/D omvandlaren har en 10 bitars upplösning, returnerar alltså värden mellan 0-1024. Spänningen för denna styrenhet ligger mellan 0-5 V.

b) I vilken intervall kommer programmet att läsa in värden från vattenmodellen och vad motsvarar en enhet av den binära signalen för ett värde av den analoga mätsignalen?

Eftersom spänngen ligger mellan 0-5 V och den 10 bitars upplösning mellan 0-1023 motsvarar 1 enhet av den binära signalen:

$$\frac{5}{1023} = 0.0048$$

A.3.3 D/A-omvandling

a) Beskriv de karakteristiska, för vattenmodellen relevanta, egenskaperna av en D/A-omvandling på en Arduino Uno, (pin 3).

D/A omvandlaren på Arduino Uno har en 8-bitars DAC, som producerar värden 256 olika spänningsnivåer mellan 0-10 V.

b) I vilken intervall kommer styrvärden att vara begränsat genom programmet som körs via en Arduino Uno? Om vi antar att pumpen ska styras mellan 0 och 10V vad kommer man att få för en upplösning av styrsignalen?

$$\frac{10}{256} = 0{,}0391 \approx 0{,}4\ \%$$

c) Vilka andra möjligheter till D/A omvandling finns till exempel på en Arduino Due?

Två stycken pinnar med 12-bitar DAC med värden mellan 0-4095 för en full DAC upplösning.

A.4 Klassiska reglerprinciper

I praktiska delen ska vi testa olika reglerprinciper som används i ”regulatorblocken” inom den klassiska, enkla reglerkretsen. Regulatorn är den del i blockdiagrammet som har felsignalen e(t) som ingång och styrsignalen u(t) som utgång.

A.4.1 Tidsdiskret tvålägesreglering
Ange Matlabkoden för regulatorblocken som implementerar en ”tvålägesreglering”.

a) Beskriv relationen mellan styrsignalen och felvärdet i egna ord:

Tvålägesreglering som är en olinjär reglerprincip sägs vara den enklaste formen av reglering. Vanligt för tvålägesregleringen är att styrsignalen (u) enbart kan anta två olika värden u1 och u2. Det aktuella värdet beror på om felsignalen (e) d.v.s. skillnaden mellan börvädet och ärvärdet är positivt eller negativt (Thomas B, 1997, s. 46).

b) hur ser (pseudo-) koden för regulatordelen ut?

Matematisk beskrivning:

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

Pseudokod:

Givet: $e = börvärde - ärvärde$

Om e > 0
U1; d.v.s. noll
Annars om e > 0
U2; d.v.s. ett

A.4.2 Tisdiskret P-reglering

Ange Matlabkoden för regulatorblocken som implementerar en P-reglering.

a) Beskriv relationen mellan styrsignalen och felvärdet i egna ord:

När det gäller proportionell reglering (p-reglering) är förändringarna i styrsignalen u proportionella mot reglerfelet e d.v.s. insignalen till regulatorn. Sambandet mellan felsignal e och styrsignalen u kan beskrivas med följande: $u = u_0 + K \cdot e$. Styrsignalens normalvärde u_0 är det värde på styrsignalen har då felet = 0. Förstärkningen K bestämmer hur mycket regulatorn ska ge för att justera för felet som uppkommit, d.v.s. hur mycket styrsignalen u ska förändras när felet e ökar med en enhet. När det gäller konstanten u_o bör man utgå från att den motsvarar normalt börvärdet. Konstanten k bör väljas till så litet som möjligt för att ett lägre värde ger ett system en god stabilitet men inte lika god snabbhet och ett större värde på konstanten k ger vice versa. Sambandet som för felsignalen e och styrsignal u gäller för "ideal" P-regulatorer medan det i praktiken finns ett gränsvärde för hur höga respektive låga värden styrsignalen kan anta. Man brukar säg att styrsignalen är proportionell mot felsignalen endast inom ett specifikt område s.k. proportionella bandet (Thomas B, 1997, s. 51, 199).

b) hur ser (pseudo-) koden för regulatordelen ut?

Om e > 0
$u = K \cdot e$
Annars
$u = 0$

A.4.3 Tidsdiskret PI-reglering
Ange Matlabkoden för regulatorblocken som implementerar en PI-reglering.

a) Beskriv relationen mellan styrsignalen och felvärdet i egna ord:

När P- och I-delen kombineras kallas man det för PI-reglering. Detta för att skapa en kombination med fördelarna från varje regulator typ. Sambandet kan beskrivas enligt följande:

$$u(t) = K\left[e(t) + \frac{1}{T_I}\int_0^t e(t)dt\right]$$

Förstärkningen K påverkar båda termerna i PI-regulatorn. Integreringstiden T_I anger hur snabbt I-delens utsignal ändrar sig jämfört med P-delens vid ett bestämt reglerfel. Integreringstiden T_I väljs vanligtvis hyfsat stor vilket leder till att I-delens utsignal ändrar sig ganska långsamt jämfört med P-delen. Integreringstiden T_I anger därmed också tiden det tar för I-delens utsignal att bli lika stor som P-delens utsignal (Thomas B, 1997, s. 60, 203).

b) För att räkna ut I-andelen brukar man använda en rekursiv summa ”w”. Vad är skillnaden mellan följande (giltiga) två sätt att räkna ut w:

i) w(k)=w(k-1)+e(k)
ii) wk=wk+e(k)

I den första (i) satsen är variabeln för den beräknade summan uppdaterad för varje samlingsintervall vilket innebär att man inte har något behov av att spara alla värden och beräkna summan vid varje samplingsintervall. Man beräknar istället samtliga tidigare värden av felsignalen och adderar detta med nuvarande.

I den andra (ii) satsen väljer man att spara alla värden på gamla felsignaler och sedan beräkna summan av felsignalerna vid varje samplingsintervall.

c) Om man inte använder sig av rekursiva beräkningen via w(k), hur skulle man också kunna räkna ut summan av alla fel från första samplingstidspunkt fram till k-te samplingstidpunkten?

Ett sätt att räkna ut summa av alla fel är att iterera genom längden av samplingsintervallet och lagra samtliga värden för att slutligen beräkna summan.

d) Vad betyder K, dT och TI i formeln?

$D_T: samplingstid\ i\ sekunder$

$T_I: Integreringstiden$

$K: Förstärkning$

e) Hur ser (pseudo-) koden för regulatordelen ut?

Om 0 > e >0

$u = K(e + (\frac{dT}{Ti} \cdot (summan\ av\ felsignaler))$

Annars

$u = 0$

A.4.4 Tidsdiskret PID-reglering

Ange Matlabkoden för regulatorblocken som implementerar en P-reglering.

a) Beskriv relationen mellan styrsignalen och felvärdet i egna ord:

När det gäller D-blocket har den funktionen att dess utsignal är skild från noll endast när insignalens derivata är skild från noll. Detta innebär att desto större förändring av insignalen d.v.s. själva felet desto större utsignal. Detta innebär också att en konstant insignal ger en utsignal som är noll. En PID-regulator består av summan av resultatet från samtliga tre block. Sambandet mellan insignalen *e* och utsignalen *u* kan formuleras enligt följande (Thomas B, 1997, s. 61, 211).

$$u(t) = K\left[e(t) + \frac{1}{T_I}\int_0^t e(t)dt + T_D \cdot e'(t)\right]$$

b) hur ser (pseudo-) koden för regulatordelen ut?

Om 0 > e > 0

$$u(k) = K\left[e(k) + T_D\frac{e(k)-e(k-1)}{h} + \frac{h}{T_I}\sum_{i=1}^{k} e(i)\right]$$

Annars

$$u = 0$$

A.4.5 Tumreglermetoder

a) Vad är syftet med tumregelmetoder och varför behövs dem? Fungerar de alltid eller vad är möjliga begränsningar?

Tumreglermetoder kan beskrivas som enklare inställningsmetoder som inte kräver att man känner till den överföringsfunktion för processen som ska regleras. De används då främst i fall där kraven för regleringen inte är så höga. Fördelen med tumreglermetoder är just att man undankommer omfattande teoretiskt arbete. Nackdelen är att det endast leder till en grovinställning av en regulator. Resultatet blir en blandning av olika egenskaper som t.ex. stabilitet, snabbhet, dämpning av störningar m.m (Thomas B, 1997, s. 190).

b) Vad är skillnaden mellan de olika tumregelmetoderna som beskrivs i kursboken och hur heter dem? (OBS: Olika upplagor av kursboken anger olika tumregler! Ange det som stämmer för upplagan som du använder)

Det finns ett stortantal tumregelmetoder som t.ex. Ziegler-Nichols svängningsmetod, Chien, Hrones och Reswicks stegsvarsmetod.

Ziegler-Nichols svängningsmetod innebär att man först inte ställer PID-regulatorn som en ren P-regulator med låg förstärkning d.v.s. att *Td* och *Ti* sätts till noll respektive oändligheten. Därefter ökar man förstärkningen *K* gradvis fram till dess att reglersystem börja självsvänga vilket sker när *K=K0.* Därefter mäter man upp den periodtid *T0* för självsvängningen. Och därefter ställer man inte regulator parametrarna enligt angiven tabell för metoden.

En nackdel med ovanstående svängningsmetod är att det praktiska experiment som metoden utgår ifrån då det kan vara riskabelt att fortsätta en reglersystem i självsvängning. Det finns då andra metoder som kan används som t.ex. Chien, Hrones och Reswicks tumregelmetoder.

Man mäter då först upp stegsvaret för den process som ska regleras med styrdon och givare så noggrant som möjligt. Sedan drar man en tangent i den punkt där stegsvaret har en maximal lutning *R.* Därefter bestämmer man parametrarna *a, L* och *T* enligt angiven tabell för metoden. Slutligen ställer man in regulatorn enligt tabellen med givna värden (Thomas B, 1997, s. 190-191, 196).

c) Vilka av de förslagna tumregelmetoderna är enkla att använda på vattenmodellen. Beskriv hur du tänker använda dem var för sig, steg för steg:

Svaret på fråga c) finns även i svaret för fråga b).

För får man ställa om PID-regulatorn som en ren P-regulator med låg förstärkning d.v.s. att *Td* och *Ti* sätts till noll respektive oändligheten. Därefter ökar man förstärkningen *K* gradvis fram till dess att reglersystem börja självsvänga vilket sker när *K=K0.* Därefter mäter man upp den periodtid *T0* för självsvängningen genom att t.ex. beräkna självsvängningsfrekvensen d.v.s. den frekvens vars fasvridning är -180 grader. Och därefter ställer man inte regulator parametrarna enligt angiven tabell för metoden.

A.5 Fuzzy control

För att visa att det finns andra sätt att reglera än bara den klassiska enkel reglerkrets ska vi i kursen också titta på ”Fuzzy control”. En beskrivning av fuzzy control finns i kursboken och på its learning finns en samling av vetenskapliga artiklar om tillämpningen av fuzzy-regler på vattenmodeller (ladda ner den komprimerade mappen med intressanta vetenskapliga artiklar, titta sedan under ”Vattenmodellpapers”->”artiklar med fuzzy-nyckelordet i titel”.

A.5.1 Egenskaper av Fuzzy control

a) För vilka typer av processer lämpar sig fuzzy control bäst?

De viktigaste användningsområdena för fuzzy control är för sådana processer där tillförlitliga matematiska modeller inte finns tillgängliga men där det fortfarande i klarspråk finns möjlighet att formulera regler för hur reglering ska gå till. Ett exempel som nämns i kurslitteraturen är sådana processer inom industrin som tidigare reglerats manuellt men där det sedan misslyckats när man velat övergå till automatisk reglering baserad på traditionell reglerteori. Problemet till detta menar man kommer från svårigheten att ställa upp tillräckligt bra matematiska modeller för processerna då de kanske är mycket komplicerade som ett resultat av kraftigt olinjära och tidsvariabla. Andra sammanhang som fuzzy control används är vid sidan av reglertekniken, t.ex. signalbehandling, feldiagnos och beslutstödssystem(Thomas B, 1997, s. 343-344).

b) Vad menas med lingvistiska variabler och lingvistiska termer?

Ett exempel på lingvistisk variabel är temperatur och lingvistisk term kan då vara mycket kall, kall, ljummen, varm, mycket varm o.s.v(Thomas B, 1997, s. 345).

c) Vilka olika defuzzifieringsmetoder finns och hur fungerar dem?

COM (Center of Maximum) s.k. medelvärdesmetoden är en av de vanligaste metoderna. Denna innebär att om flera fuzzy-regler samtidigt har ett sanningsvärde som är skilt från noll så kommer utsignalen att fastställas efter det viktade medelvärdet av samtliga olika värden på utsignalen som det enskilda reglerna var för sig skulle inneburit.
CoA (Center of Area) s.k. tyngdpunktsmetoden följer samma villkor som COM vad gäller fastställningen av sanningsreglerna men sedan beräknas utsignalen som tyngdpunkten av arean som fås om utsignalens medlemsfunktioner trunkeras vid aktuella värdet på medlemsfunktionen och läggs samman till en enda area s.k. aggregering.
MoM (Mean of Maximum) s.k. medianvärdesmetoden där utsignalen endast bestäms av den regeln som har det nuvarande högsta sanningsvärdet (Thomas B, 1997, s. 348).

A.5.2 Fuzzy control av vattenmodellen

Föreslå en egen fuzzy-controller till vattenmodellen.

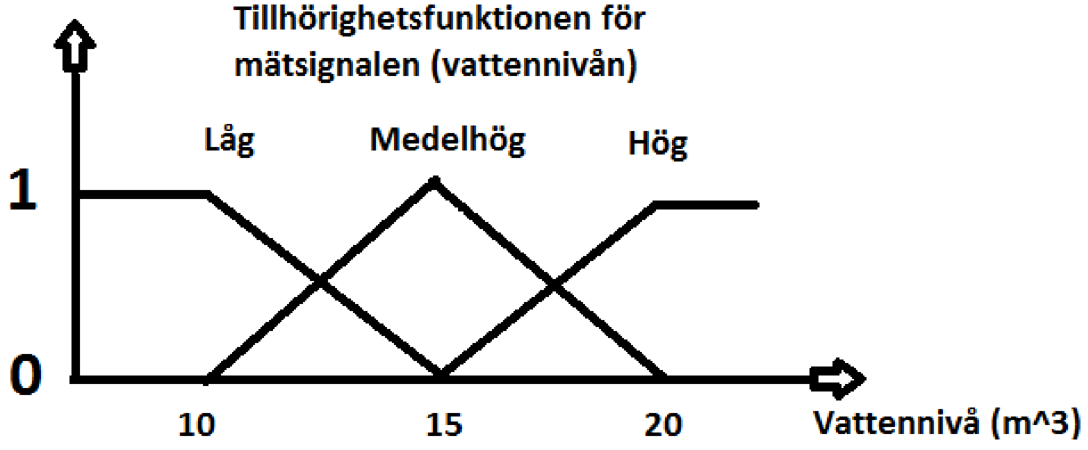
a) Vilka insignaler väljer du?

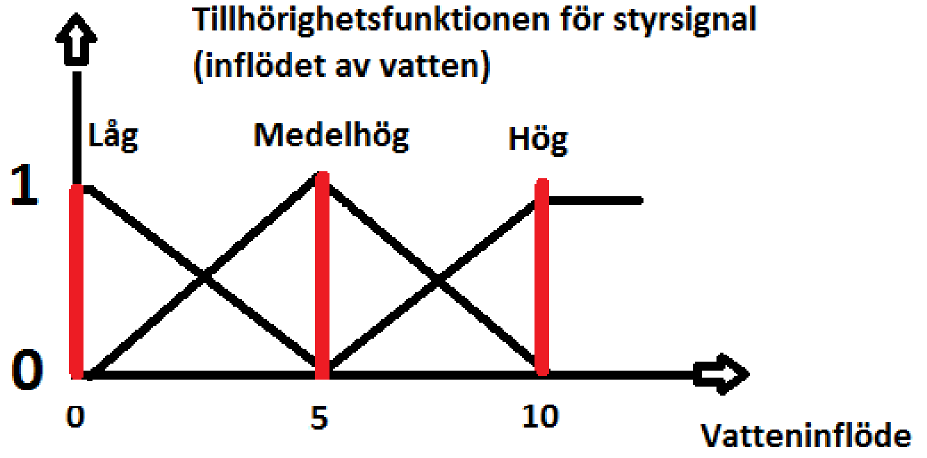
Vattennivån

b) Vad är det du vill reglera?

Reglera vattennivån i behållaren

c) Hur ser dina medlemsfunktioner ut för dina insignaler och för styrsignalen? (Rita på tavlan eller på pappret och klistra in kortet)





d) Vilka regler ska din fuzzyregulator använda?

”Om nivån i behållaren är låg ska inflödet vara stort”.
”Om nivån i behållaren är medelhög ska inflödet vara medelstort.”
”Om nivån i behållaren är hög ska inflödet vara litet.”

Exempel från kurslitteraturen Thomas.B, ”*Modern Reglerteknik”*
(Thomas B, 1997, s. 349)

e) Vilka defuzzifieringsmetod tänker du använda?

Center of Maximum – om flera fuzzy-regler samtidigt har en tillhörighetsgrad skild från noll bestäms utsignalen av det viktade medelvärdet.

f) Vilken styrvärd kommer din fuzzyregulator mest sannolikt att räkna ut när den sätts igång och både vattenbehållare är tumma?

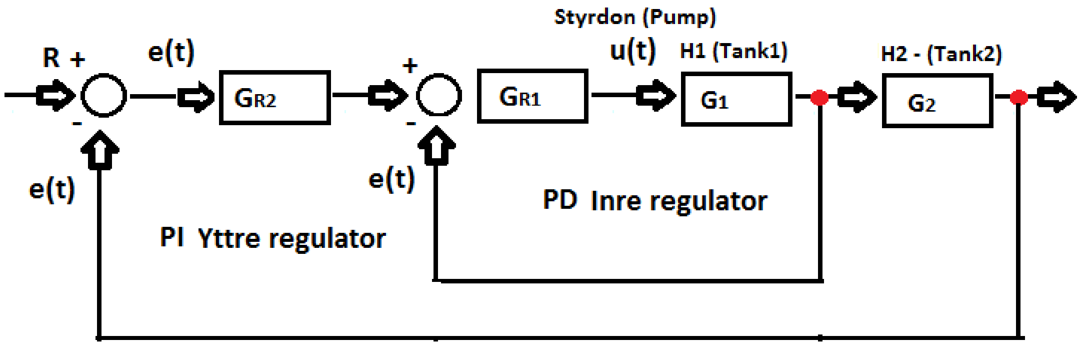
Om vatten behållaren är tomma vi start kan vi ur ovanstående tabell tyda att det motsvarar en ett för varje vatten behållare – med OCH-funktionen även s.k. minimum funktion innebär det att två ettor på ingångarna ger en etta ut.

A.6 Kaskadreglering

Vattenmodellen består av två seriekopplade vattenbehållare. Det gör systemet lämpligt för en kaskadreglering. Kaskadregleringen delar upp systemet i två reglerkretsar – en inre som brukar vara snabbare och en yttre som antas vara långsammare. Designen av kaskadregleringen blir enklare då man först ställer in en regler för inre kretsen och sedan en till yttre.

A.6.1 Scheman för en kaskadreglering av vattenmodellen
Rita scheman för en kaskadreglering av vattenmodellen. Vad väljer du som inre reglerkrets och vad som yttre? (Rita på tavlan eller papper och kopiera in bilden här).

A.6.2 Kod för kaskadreglering av vattenmodellen
a) Rita scheman för en kaskadreglering av vattenmodellen. Vad väljer du som inre reglerkrets och vad som yttre?



Den yttre reglerkretsen utgörs av den andra vattenbehållaren *(h2)* samt en PI-regulator medan den inre reglerkretsen utgörs av den första vattenbehållaren *(h1)* samt en PD-regulator.

b) Vilka regleralgoritmer väljer du för respektive reglerkrets? Förklara din val.

Den inre reglerkretsen uppgift är att kompensera för störningar så att den inte hinner få för stor inverkan på den yttre processen (Thomas B, 1997, s. 237) *(G2)* d.v.s. den andra vattenbehållaren. PD- regulator bidrar till ökad snabbhet och stabilitet vilket är ett krav för den inre reglerkretsen. PI-regulator eliminerar kvarstående fel vilket gör den till en god kandidatt för den yttre reglerkretsen dock medför den också försämrad stabilitet (Thomas B. 1992, s. 203).

c) Visa med pseudo-kod hur en regulator med kaskadreglering skulle kunna se ut.
(Tips: programmera först regulatorn för den inre reglerkretsen)

Givet: *Sampelintervall, börvärde, förstärkning K*

Loop1{

Loop2{

PI-delen
Mät: ärvärde för block 1;
Beräkna: felvärde för block 1 (börvärde - ärvärde för block 1);
Beräkna: summan av alla felsignaler;
Beräkna: styrsignal PI (se formel i tidigare avsnitt);
}

PD-delen
Mät: ärvärde för block 2;
Beräkna: felvärde för block 2 (styrsignal P1 – ärvärde för block 2);
Sätt: börvärde till styrsignal för P1;

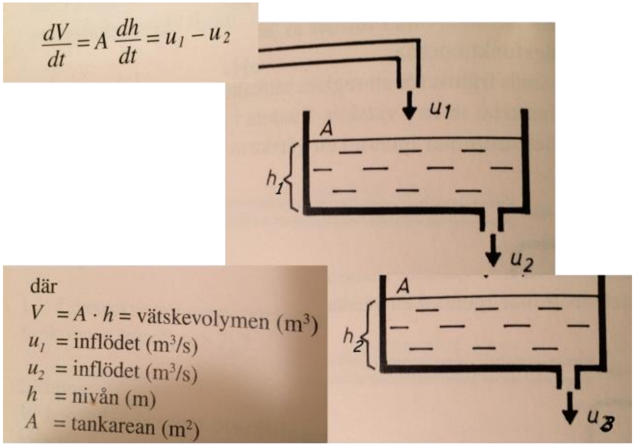
Sätt: felvärde för block 1 till felvärde för block 2;
Sätt: ärvärde för block 1 till ärvärde för block 2;

Beräkna: felvärde för block 2 (börvärde -ärvärde);
Beräkna styrsignal för PD-delen(se formel i tidigare avsnitt);

}

A.7 Egenskaper hos processer och reglersystem

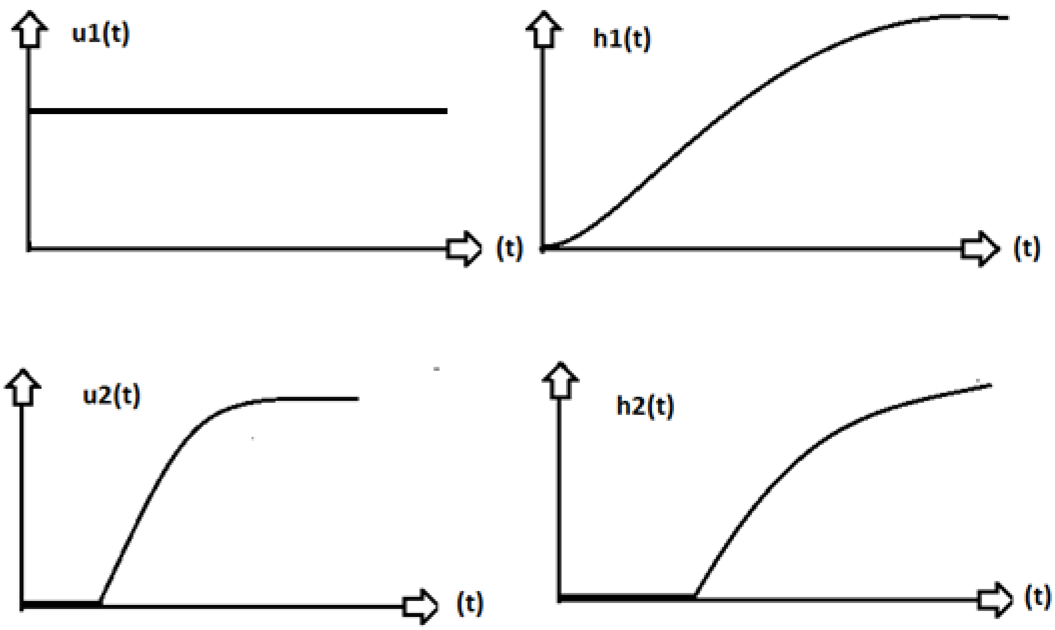
Olika system beter sig annorlunda. Beroende på deras statiska och dynamiska egenskaper kan man dela in dem i olika processtyper. Ett sätt att få systemet att avslöja sina egenskaper är att ta upp stegsvaret. Stegsvaret är förloppet av utgångsvärdet när ingångssignalen är ett steg.



Figur A.7.1 kopplade vattentankar

I följande uppgiften fokuserar vi oss på vattenmodellen enligt figur A.7.1 ovan.

A.7.1 Stegsvar i öppna regelkretsen
Rita upp hur vattennivåerna h1 och h2 förändrar sig som följd av ett stegsvar, dvs om man pumpar in ett konstant flöde u1 i den första vattentanken och vattnet därifrån flyter ut in i den andra tanken. Vi antar att tankarna är helt tomma i början och att tillflödet u1 är lagom så att vattnet inte rinner över, dvs det finns ett nivå h1 där tillflödet u1 är lika stor som utflödet u2.
(Rita på tavlan eller på pappret och tar kort och klistra in här:)



A.7.2 Processtypen för nivån i första eller andra behållaren
a) Utifrån dina funderingar om stegsvaret av h1(t) och h2(t) i förgående uppgiften: skiljer sig dynamiken åt av de två nivåförloppen?

b) Vilken processtyp har första behållaren?

Process med integration - Den första processtypen är process med integration som kännetecknas av att utsignalen (på något sätt) är integralen av insignalen. Där insignalen är inflödet u vatten nivån h är utsignalen (Thomas B. 1992, s. 33).

c) Vilken processtyp har andra behållaren?

Process med dödtid - Det som kännetecknar andra behållaren är att det tar en viss tid innan en ändring av insignalen syns på utsignalen. Dödtiden är i flest fall en transport tid då material förflyttas i detta fall vätskan (Thomas B. 1992, s. 33).

d) Hur förklarar du dina svar?

Se ovanstående svar i b) & c).

A.7.3 Egenskaper hos återkopplade system

I praktiska delen kommer du att reglera vattenmodellen med olika regulatorer. Fundera hur resultatet skulle kunna jämföras med varandra angående de relevanta egenskaperna hos återkopplade system, såsom

a) Störningsdämpning

Ett mål med återkoppling är att dämpa inverkan av störningar. I detta fall är huvudsyftet att med återkopplingen att skapa ett system som är självkorrigerande, med andra ord ett system som motverkar det störningar som verkar på en viss process. Störningsdämpning kan matematiskt beskrivas som funktion av frekvensen d.v.s. kvoten mellan utsignalens amplitud och störningens amplitud där målet är att finna en standardavvikelse mellan ovanstående amplituder som är så liten som möjligt. (Thomas B. 1992, s. 33, 37).

b) Stabilitet

Med stabilitet vill man skapa ett system som kan garantera att utsignalen inte växer okontrollerat eller varierar mellan amplituder p.g.a. av störningar. Ur *Modern Reglerteknik 1992: "Ett reglersystem för vilket man kan garantera att utsignalen inte börjar skena iväg eller börjar pendlar med tilltagande amplitud till följd av inkommande störningar sägs vara ett stabilt system"* (Thomas B. 1992, s. 38).

c) Snabbhet

Snabbhet syftar till hur snabbt utsignalen hos systemet svänger in sig efter börvärde ändringar men även hur fort systemet klarar av att undanröja reglerfel som kommer från störningar (Thomas B. 1992, s. 39).

d) Statisk noggrannhet

Statistisk noggrannhet syftar till hur väl ett system kan ställa in sig på varierande börvärden men även hur väl system kan undanröja reglerfel i statisktillstånd (Thomas B. 1992, s. 41).

(Beskriv vad man ska göra till punkterna a-d för att helst få fram kvantitativa mått. Anta att resultatet av utgångsvärdet finns som en vektor h1 eller h2 med alla samplade värden.)

a) Störningsdämpning

Eftersom det i stora drag endast är lågfrekventa processtörningar som går att dämpa och inte medel- och högfrekventa störningar. Ett stabilt reglersystems förmåga att dämpa processtörningar av lågfrekvent karaktär är djupt förankrad i regulatorns förstärkning vid låga frekvenser. Nedanstående kvot gäller som mått på en regulators förmåga att dämpa lågfrekventa processtörningar(Thomas B. 1992, s. 171-177).

$$J_v = \frac{1}{K_1}$$

Om värdet för detta kriterium är lågt anses dämpningen vara god och om den är hög anses den inte uppfylla ovanstående kriterium.

b) Stabilitet

När det gäller stabilitet finns det ett flertal olika metoder för att studera stabiliteten hos ett system. Dessa är bl.a. det förenklade Nyquistkriteriet, Polbestämningen och Rouths metod. Nedan beskrivs Nyquistkriteriet och polbestämning (Thomas B. 1992, s. 154-163).

.

- Instabilitet hos återkopplade system innebär oftast att utsignalen hos systemet börjar självsvänga med ökande amplitud. Förenklad Nyquistkriteriet innebär att ett återkopplat system är stabilt om amplitudförstärkningen hos kretsöverföringen G_k är mindre än 1 vid den frekvens där fasförskjutningen är – 180 grader annars är systemet instabilt.

- Polbestämning innebär att man skall finna ett maximalt värde på *K* innan system blir instabilt. Först gäller det att bestämma processens överföringsfunktion *H(z)* och finna ett *K* inom enhetscirkeln inom radien 1.

c) Snabbhet

Ett sätt att beräkna systemets snabbhet är via dess stigtid t_r d.v.s. den tid det tar för utsignalen att ändra sig från 10 % till 90 % av sitt slutvärde vid en stegformad ändring av börvärdet.

Ett annat mått på snabbhet kan finnas med Bodediagram och är överkorsningsfrekvensen ω_c för kretsöverföringen, vilket är den frekvens där amplituden är lika med 1. Detta då en hög överföringsfrekvens innebär att systemet kan förstärka höga frekvenser vilket leder till ökad snabbhet. (Thomas B. 1992, s. 171, 39).

Sambandet mellan överkorsningsfrekvensen ω_c och t_r kan beskrivas enligt följande:

$$t_r \approx \frac{1{,}4}{\omega_c}$$

d) Statistisk noggrannhet

Beräkning av kvarstående fel för tidsdiskreta system. Vid stegformade börvärdesändringar R:

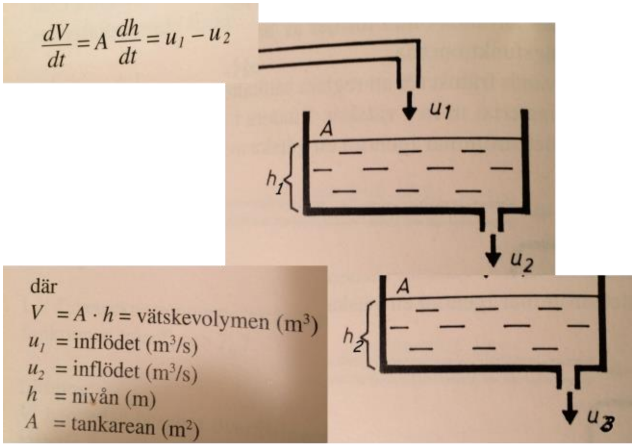
$$\mathrm{e}_0 = \lim_{z\to 1}\left(\frac{a}{1 + H_R + H_P}\right)$$

$d\ddot{a}r\ a\ \ddot{a}r\ sregets\ h\ddot{o}jd$

Formler för rampformade börvärdesändringar tar istället rampens lutning i beaktning. (*h*)

A.8 Bestämning av tidsdiskreta överföringsfunktionen utifrån stegsvar med z-transformationen

Teoretiskt ska dynamiken i vattenmodellen kunna beskrivas med en differensekvation första ordning för nivån i första behållaren (h1) och en differensekvation andra ordning för nivån i andra behållaren (h2). Resultatet man får från ett stegsvar ska därför uppfylla respektive ekvationer:

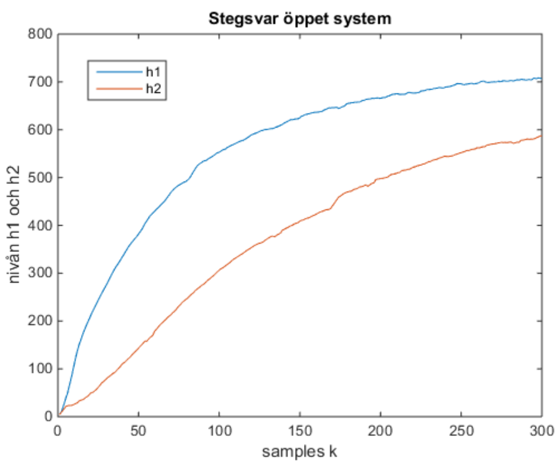


För nivån h1 (allmän differensekvation första ordning):
$h1(k) = a \cdot u1(k-1) + b \cdot u1(k-1)$

För nivån h2 (allmän differensekvation andra ordning):
$h2(k) = a_1 \cdot h2(k-1) + a_2 \cdot h2(k-2) + b_1 \cdot u1(k-1) + b_2 \cdot u1(k-2)$

I realitet påverkas själva vattenmodellen och mätvärden för h1 och h2 av yttre störningen. Dessutom är själva processen inte lineärt.
Genom att filtrera signalerna med exempelvis en ”moving average filter” kan störningarna dock dämpas något.

Utgå i uppgiften nedan från resultatet av följande stegsvar av den öppna vattenmodellen som har genomförts:



Tabell A.8: Mätresultat från stegsvaren

k	1	2	3	4	5	6	7
u1(k)	65	65	65	65	65	65	65
h1(k)	4.6	9.2	18.0	28.2	40.2	49.4	63.4
..h2(k)	4.2	8.4	12.8	17.2	21.8	22.8	23.4

A.8.1 Att bestämma differensekvationer utifrån mätvärden (systemidentifikation)

a) Bestäm parametrarna a och b i differensekvationen första ordning med hjälp av mätresultaten i tabellen A.8. Vilka värden får du för a och b om du använder dig av ekvationen för h1(k=1) och h1(k=2)?

```
>> y=[18;9.2];
>> A=[9.2 65;4.6 65];
>> A/y
Error using  /
Matrix dimensions must agree.

>> A\y

ans =

    1.9130
    0.0062
```

b) Vilka värden får du för a och b om du använder dig av ekvationen för h1(k=6) och h1(k=7)?

```
>> y=[63.4;49.4];
>> A=[49.4 65;40.2 65];
>> A\y

ans =

    1.5217
   -0.1811

>>
```

c) Bestäm också parametrarna a1, a2, b1 och b2 i differensekvationen andra ordning med hjälp av mätresultaten i tabellen A.8. Visa hur du har ställt upp och löst ekvationen. (Det går bra att använda papper eller Matlab, bara du visar och förklarar hur du har gjort).

```
>> y = [23.4; 22.8; 21.8; 17.2; 12.8];
>> A=[22.8 21.8 65 65; 21.8 17.2 65 65; 17.2 12.8 65 65; 12.8 8.4 65 65; 8.4 4.2 65 65];
>> A\y
Warning: Rank deficient, rank = 3, tol =
1.613647e-13.

ans =

    1.0973  // a1
   -0.3343  //a2
    0.0876  //b1
         0  //b2
```

A.8.2 z-Transformationen och tidsdiskreta överföringsfunktioner

a) Hur ser differensekvationen första ordningen ut för h1 och u1 med de parameter a och b som du har räknat ut i A8.1?

a = 1.9130
b = 0.0062

$$h1(k) = 1.9130 \cdot h1(k-1) + 0.0062 \cdot u1(k-1)$$

b) Hur ser z-Transformationen ut av denna differensekvation första ordningen?

$$H1(z) - 1.9130 \cdot H1(z) \cdot z^{-1} = 0.0062 \cdot U1(z) \cdot z^{-1}$$

c) Hur ser den tidsdiskreta överföringsfunktionen för första vattenbehållaren ut som beskriver relationen H1(z)/U1(z)?

$$\frac{b \cdot z^{-1}}{a \cdot z^{-1}}$$

$$(b)/(z-a)$$

d) Hur ser differensekvationen andra ordningen ut för h2 och u1 med de parameter a1, a2, b1 och b2 som du har räknat ut i A8.1?

a1 = 1.0973
a2 = -0.3343
b1 = 0.0876
b2 = 0

$$h2(k) = 1.0973 \cdot h2(k-1) - 0.3343 \cdot h2(k-2) + 0.0876 \cdot u1(k-1)$$

e) Hur ser z-Transformationen ut av denna differensekvation andra ordningen?

$$H2(z) - 1.0973 \cdot H2(z) \cdot z^{-1} + 0.3343 \cdot H2(z) \cdot z^{-2} = 0.0876 \cdot U(z) \cdot z^{-1}$$

f) Hur ser den tidsdiskreta överföringsfunktionen för andra vattenbehållaren ut som beskriver relationen H2(z)/U1(z)?

$$\frac{\frac{b1 \cdot z^{-1}}{1}}{1 - a1 \cdot z^{-1} + a2 \cdot z^{-2}}$$

$$b1 \cdot Z/(Z^2 - a \cdot Z + a2)$$

B) MATLABDEL SOM FÖRBEREDNING

Denna del kan utföras i en vanlig datasal på Mah eller på en dator med installerad Matlablicens. B.1 kräver Matlabs control-toolbox.och B.2.6 Matlabs fuzzy-toolbox. Om det saknas dessa toolboxen på datorn som du använder kan du också göra färdig respektive uppgifter senare i labbsalen.

B.1 Simulering av stegsvaret med Matlab med hjälp av tidsdiskreta överföringsfunktioner

Överföringsfunktioner som du räknade ut i uppgift A.8.2 kan användas i Matlab för att simulera stegsvaren.
Följande sammanställning av Matlabfunktioner (kräver control-toolboxen) kan vara bra att känna till:

Användbara Matlab-kommandon (5)
Följande Matlab-kommandon, som är mer utförligt beskrivna i kapitel 22, är lämpliga för att med dator lösa samma typer av problem som studerats i detta kapitel. Observera att de flesta kommandon som diskuterats i den analoga delen ser likadana ut i det tidsdiskreta fallet, det gäller t ex kommandon för beräkning av tidsförlopp, Bodediagram, blockschematransformering, beräkning av poler och nollställen m m.

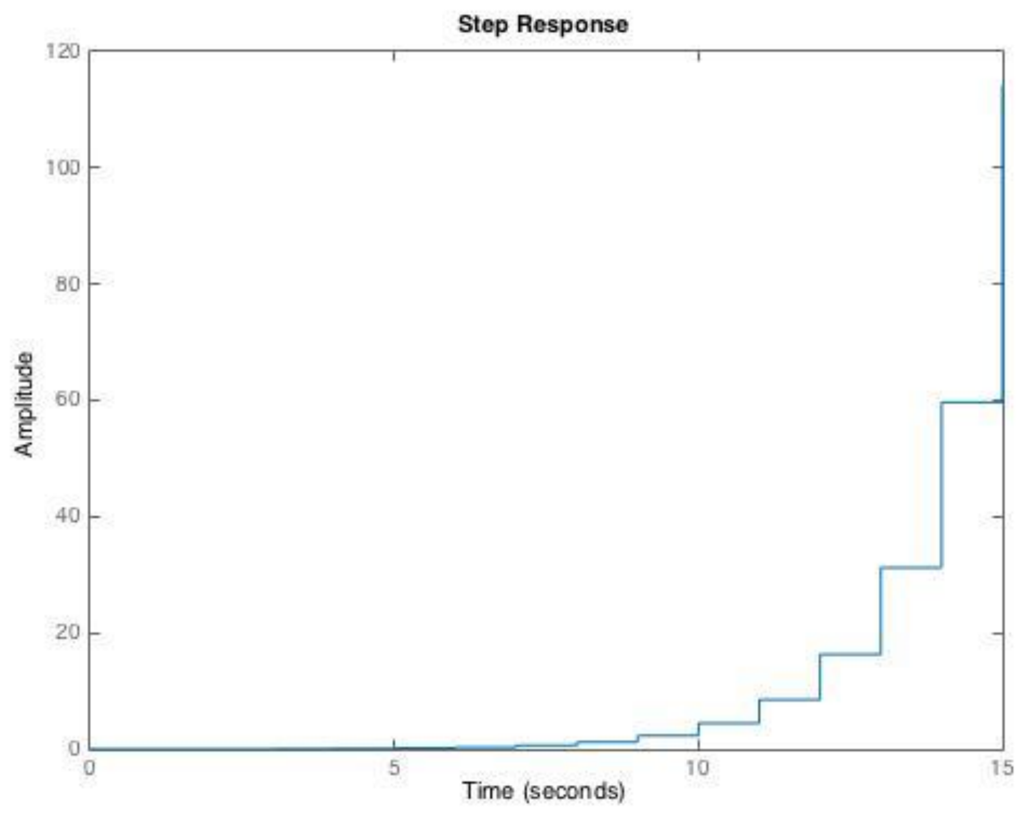
H=tf([…],[…],h)	kommando för inmatning av tidsdiskreta överföringsfunktioner. Det tredje argumentet anger aktuell samplingstid.
H=zpk([…],[…],k,h)	kommando för att skapa en tidsdiskret överföringsfunktion vars poler, nollställen och statiska förstärkning är givna. Det fjärde argumentet anger aktuell samplingstid.
impulse(H)	ger impulssvaret för det tidsdiskreta systemet H.
step(H,t)	ger stegsvaret för H upp till tiden t.
lsim(H,u)	simulering av ett system med godtycklig insignal specificerad i vektorn *u*. Samplingstiden förutsätts vara samma för processen H som i vektorn *u*.

B.1.1 Simulering av stegsvaret för h1 med Matlab

a) Klura ut hur du ska använda dig av funktionen ”tf” (eng. transfer function = överföringsfunktion) för att beskriva din tidsdiskreta överföringsfunktion som du räknade ut i uppgift A.8.2.c). Anta att samplingstiden var h=1 sekund. Klistra in det du gjorde från Matlabs kommandofönster

```
>> H = tf([0.0062], [1 -1.9130], 1)

H =

   0.0062
  ---------
  z - 1.913

Sample time: 1 seconds
Discrete-time transfer function.

>>
```

b) Simulera nu stegsvaret och klistra in resultatet som plott här:



c) Jämför din plott från simulationen med grafen i uppgift A.8. Vad är skillnaden, är de lika?

Den största skillnaden uttrycker sig i att samplingstiden är betydligt längre. Vi har en samplingstid på 15 sekunder som är tjugo gånger mindre. För en mer tydlig bild plottar jag med högre sampling och kan då se att kurvorna är ganska lika. Som referens använde jag

```
>> step (H, 15)
>> step (H, 10)
>> step (H, 5)
>> step (H, 30)
>> step (H, 300)
>> step (H, 20)
>> step (H, 50)
>>
```

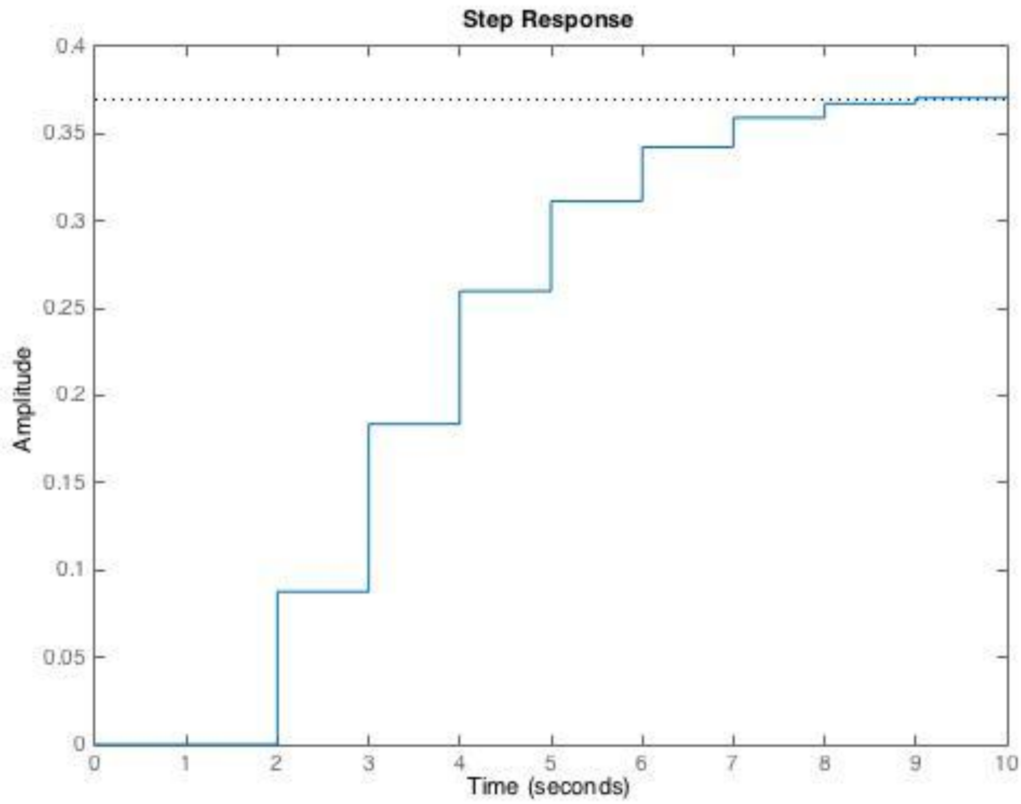
Och jämförde de olika plottarna.

B.1.2 Simulering av stegsvaret för h2 med Matlab

a) Använd dig igen av funktionen ”tf” för att beskriva din tidsdiskreta överföringsfunktion för andra behållaren som du räknade ut i uppgift A.8.2.f). Anta att samplingstiden var h=1 sekund. Klistra in det du gjorde från Matlabs kommandofönster

```
>> H = tf([0.00876], [1 -1.9130 -0.3343], 1)

H =

        0.00876
  ----------------------
  z^2 - 1.913 z - 0.3343

Sample time: 1 seconds
Discrete-time transfer function.
```

b) Simulera nu stegsvaret och klistra in resultatet som plott här:



c) Jämför din plott från simulationen med grafen i uppgift A.8. Vad är skillnaden, är de lika?

Återigen är det samplingstiden som är annorlunda. Vi använde samma metod för att klargöra skillnader och likheter och kom fram till att kurvorna är ganska lika.

B.2 Programmering av regulatorer i Matlab

För att kunna experimentera med olika regulatorer som reglerar vattenmodellen behöver du en rad Matlabfunktioner. Meningen med detta kapitel är att den som följer anvisningen och löser uppgifterna steg för steg bygger upp de nödvändiga Matlabfunktionerna.

I verktygslådan med Matlabfunktionerna som vi kommer ta fram och använda under den praktiska delen kommer att finnas följande funktioner:

- vm_openstep(): Sampling av stegsvaren (nivå h1 och h2) i den öppna reglerkretsen
- vm_twostep(): Tvålägesreglering (på/av-regulator) av vattennivån h2 i den nedra vattentanken.
- vm_P(): P-regleringen av vattennivån h1 i den övre vattentanken
- vm_PI(): PI-regleringen av vattennivån h1 i den övre vattentanken.
- vm_PID(): PID-regleringen av vattennivån h1 i den övre vattentanken.
- vm_fuzzy(): Fuzzy-regleringen av vattentanken.
- vm_kaskad(): Kaskad-regleringen av vattentanken.

B.2.1 Sampling av det öppna stegsvaret med Matlab (function vm_openstep)

Det finns en funktionsstomme ”vm_o.m” på its learning som ni kan ladda ner och använda att utgå ifrån. Spara det i så fall som ”vm_openstep” under samma filnamn.

I stommen finns följande delar:

Funktionsdeklaration med funktionens ingångar (argument) och utgångar (resultat):

DEL A: Beskrivning av de olika variablerna
Förklarar betydelsen av de olika variablerna som används som ingångar och utgångar till funktionen.

DEL B: Arduino-mapping i Ctrl-boxen
Definierar och initialiserar variabler som behövs för att få Ctrl-boxen att fungera. Vi återkommer till några detaljer i praktiska delen.

DEL C: Initialisering av in- och utgångar på Ctrl-Boxen
Borde kännas bekant. Ni fick redan göra likadant i tidigare uppgifter. Genom initialiseringen med a.pinMode() anger vi vilka anslutningar på ctrl-boxen som är ingångar eller utgångar.

DEL D: Skapa och initialisera olika variablerna för att kunna spara mätresultat.

DEL E: starta stegsvarsexperimentet
- Slår på pumpen med styrsignal ”v”
- Sampla N-gångar signalförloppet och sparar resultatet i vektorer
- Uppdatera plotten i varje samplingstillfälle

- I varje sampling: Räkna ut hur lång tid som mätningen och plottning har tagit och vänta resterande tid tills samplingstiden dT är uppnått

DEL F: avsluta experimentet
- stäng av pumpen
- plotta slutresultatet
-

Sätt dig in i programstommen och försök att förstå hur funktionen fungerar genom att svara på följande frågor:

B.2.1.1 Funktionsdeklaration av vm_openstep()

a) Vilka argument behöver funktionen och vilka är vettiga värden?

Funktionen behöver argumenten a, N, dT och v.

a: Blir arduino-objektet där vi väljer vilken comport som blivit tilldelad.
N: Antal sampling, 200
dT: Samplingstiden i sekunder, 1
v: Stegets höjd, börvärdet

b) Hur ser själva upprop av funktionen ut i kommandofönstret?

```
vm_openstep(a,N,dT,v)
```

>>

B.2.1.2 Längden av stegsvarsexperimentet

a) Hur länge, dvs hur många sekunder kommer experimentet att pågå med de variabler som du har valt under B.2.1.1 ?

200 sekunder

antalet samplingar multiplicerat med samplingstiden i sekunder (N*dT)

B.2.1.3 Garanterade samplingstider

a) Hur garanteras det i koden att varje samplingstid är precis lika lång?

Det görs genom en kontroll av en if-sats som granskar om koden hinner köras innan samplingstiden löper ut.

b) Vad skulle hända med samplingstiden om man inte kollade den och pausade alls? (Beskriv vad samplingstiden skulle bli beroende av och varför det är ett problem).

Detta innebär att körningstid för programmet skulle vara lika långt som samplingstiden. Detta hade medfört att samplingstiden hade kunnat variera mellan olika tidsintervall p.g.a. att körningstiden för programmet varierar för varje körning.

B.2.2 Tvålägesreglering (på/av-regulator) (function vm_twostep)

Första regulator som vi ska prova är en enkel på/av regulator för regleringen av vattennivån h2 i den nedra vattentanken.

Vi utgår från funktionen ”vm_openstep”, dvs öppna vm_openstep i Matlab och spara den som ”vm_twostep”!

För att sätta in en på/av-regulator och för att sluta den öppna reglerkretsen behöver vi lägga till följande kod:

1. Ändringar under ” % DEL E: starta stegsvarsexperimentet”
   - Ta bort hela raden som startar pumpen, dvs ta bort a.analogWrite(PWMA,v);
2. I själva for-loopen efter raden t(i)=i; och innan plotten ska vi lägga till följande rader för att:
   - Slutna reglerkretsen genom att räkna ut felvärdet e som differens mellan ärvärdet och börvärdet
     e(i)= bv-h2(i); %räknar ut felvärdet som differens mellan ärvärdet och börvärdet
3. Efter uträkningen av felvärdet kommer själva regulatordelen. En på/av-regulator kollar helt enkelt om felvärdet är större än noll och sätter på pumpen om så är fallet tills felvärdet blir lika eller mindre än noll. Då stängs pumpen av. Koden blir då följande:
   - ```
     % REGULATORN
     if e(i) > 0
         u(i) = 200; %pumpen kan styras med värden mellan 0..255
     else
         u(i) =0;
     end
     ```
4. I slutet av for-loopen, t.ex. i anslutningen till regulatorn-koden och innan %online-plot, behöver vi skriva ut vår styrvärdet u(i) till pumpen.
   - ```
     % Skriva till utgången
     a.analogWrite(PWMA, u(i));
     ```

B.2.2.1 Småändringar i funktionen

a) Kolla noggrant genom funktionen vm_twostep() och ändra och lägg till det som behövs. (Exempelvis anpassningar av argument, deklaration av variabler, osv) Ange vilka anpassningar och ändringar du genomförde:

I del A= `% e: felv%rde`

I del D= `e = zeros(1, N);% vektor för felv%rde med N antal värden`

I del E=
```
    % REGULATORN
if e(i) > 0
    u(i) = 200; %pumpen kan styras med v%rden mellan 0..255
else
    u(i) =0;
end


    % Skriva till utgÂngen
a.analogWrite(PWMA, u(i));
```

b) Lägg till felvärdet ”e” i klammer med variabler som funktionen ger tillbaka. Visa hur hela funktionsdeklaration ser ut:

```
function [h1,h2,t,u,e]
```

c) Vi vill gärna se sedan hur felvärdet ”e” ser ut. Lägg ”e” därför in i din plott-funktion. Anpassa kurvan och legenden att också beskriva ”e”. Kopiera in koden för din plott-funktion här:

```
 %online-plot
    plot(t,h1,'k-',t,h2,'r--',t,u,'m:', t,e,'b--');
 % plotta en fin slutbild,
  plot(t,h1,'k-',t,h2,'r--',t,u,'m:');
  xlabel('samples k')
  ylabel('Nivån i vattentank h1, h2, stegsvar u, felvärde e')
  title(' Stegsvar av tvålägesreglering')
  legend(' Nivån i tank ett - h1 ', ' Nivån i tank två -h2 ', 'u ', 'e')
```

Genomför ändringarna och spara hela funktionen som ”vm_twostep” under samma filnamn.

B.2.3 Reglering av vattennivån h1 med P-regulatorn (function vm_P)

”P” står för *proportional* och betyder att en P-regulator multiplicerar felvärdet e med en konstant faktor. Det är vanligt att man betecknar faktorn med variabelnamn ”*Kp*”.
Vi utgår från den senast skapade funktionen vm_twostep() och behöver i stort sätt bara ändra på följande:

- Ändra funktionsnamn och filnamn till ”vm_P” och beskriva att funktionen reglerar nivån i första behållaren med hjälp av en P-regulator.
- Lägga till variabeln Kp i funktionsupprop, och beskriva den i Del A
- Beräkna felvärdet e som differens mellan börvärdet och h1 (istället för h2)
- Ändra %Regulatorn –delen till att räkna ut u(i) enligt formeln som du tog fram i uppgift A.4.2. Tänk ocká på att ”u” måste vara heltal och får inte vara högre än 255. Ett enkelt sätt är att efter ha bestämt u(i) använda sig av följande kod:
  u(i)=min(255, round(u(i));

B.2.3.1 Småändringar i funktionen

a) Gör alla ändringarna i koden för funktionen ”vm_P” såsom det beskrivs i punkterna ovan. Kopiera in %Regulatorn-delen från din kod:

```
    % REGULATORN
if e(i) > 0
    u(i) = kp*e(i); %pumpen kan styras med v%rden mellan 0..255
else
    u(i) =0;
end

    u(i)=min(255, round(u(i)));
```

b) Förklara koden i raden med:
u(i)=min(255, round(u(i));

Värdet u begränsas till 255 samtidigt som round avrundar till ett heltal

Spara hela funktionen som ”vm_P” under samma filnamn.

B.2.4 Reglering av vattennivån h1 med PI-regulatorn (function vm_PI)

Vi fortsätter med att utveckla funktionerna även för de andra klassiska regleralgoritmerna. Efter P-reglering blir det PI-regleringen. ”I” står för *integral* och betyder att man summerar upp felvärdet och sedan multiplicerar summan med en konstant faktor som adderas till P-delen. Syftet med I-delen blir att få bort kvarstående fel. I uppgiften A.4.3 har du redan studerat formeln för PI-regulatorn och skrivit en pseudo-kod för den.

B.2.4.1 Småändringar i funktionen

a) Gör alla ändringarna i koden för funktionen ”vm_PI”. Kopiera in %Regulatorn-delen från din kod:

I P-regulator-delen har vi lagt till två variabler, dessa är förstärkningsvärdet K för PI-regleringen samt integreringstiden Ti som utgör ”I” i PI-regulatorn.

```
vm_pi(a,N,dT,v, kp, Ti)
```

För regulatordelen:

```
   % REGULATORN
if e(i) > 0
    errorsum= (e(i)+errorsum);

u(i) = kp*(e(i)+((dT/Ti)*errorsum)); %pumpen kan styras med v‰rden mellan
0..255
```

Spara hela funktionen som ”vm_PI” under samma filnamn.

B.2.5 Reglering av vattennivån h1 med PID-regulatorn (function vm_PID)

Sista klassiska regleralgoritmen som vi utvecklar en Matlabfunktion för är utvidgningen av PI-regleringen till en PID-reglering. ”D” står för *differential* och betyder att man ta med skillnaden eller differensen mellan det aktuella felvärdet e(k) och det förgående felvärdet e(k-1). Man räknar ut differensen och multiplicerar också den med en konstant faktor och resultatet adderas till PI-delen. Syftet med D-delen blir att kunna reagera snabbare på stora förändringar.
I uppgiften A.4.4 har du studerat formeln för PID-regulatorn.

B.2.5.1 Smååndringar i funktionen

a) Titta noggrant på D-delen och fundera hur du tänker programmera den. Det borde inte vara så svårt med ett undantag: Vad händer om for-loop variabeln "i" är lika med "1" när for-loopen börjar?

```
>> vm_pid(a,20,1,80,1,1,2)
Subscript indices must either be real positive integers or logicals.
Error in vm_pid (line 76)
 u(i)=kp*(e(i) + Td * ((e(i)-e(i-1))/dT) + (sumE * (dT/Ti))); %pumpen kan
styras med värden mellan 0..255
```

b) Hur gör du i programmet så att det inte blir ett problem när "i==1"?

```
    if (e(i) > 0 && i>1)
```

Vi utökar villkoren i if-satsen genom att inkludera i>1

c) Gör alla ändringarna i koden för funktionen "vm_PID". Kopiera in %Regulatorn-delen från din kod:

```
u(i)=kp*(e(i) + Td * ((e(i)-e(i-1))/dT) + (errorsum * (dT/Ti))); %pumpen
kan styras med värden mellan 0..255
disp('MATKO - ALI - KOD');

else

u(i) =0;

end
```

Spara hela funktionen som "vm_PID" under samma filnamn.

B.2.6 Fuzzy reglering av vattenmodellen (function vm_fuzzy)

Använd Matlabs fuzzy toolbox för att bygga en fuzzy regler som du sedan stoppar in istället för PID-regulatordelen i Matlabprogrammet.

Försök att realisera den fuzzy-regulator som du har beskrivit i uppgift A.5.2.
Följande är de olika steg som krävs för att bygga en fuzzy regler med användningen av fuzzy toolboxen, (se [2]) :

1. Starta toolboxen med >>fuzzy i Matlabs kommandofönster
2. Bestäm vilka ingångar fuzzyreglern ska ha och deras tillhörighetsfunktioner
3. Bestäm tillhörighetsfunktioner av reglerns utsignal (=pumpstyrning 0..255)
4. Ange reglarna som knyter ihop ingångarna med utgången
5. Kolla med "Rules"-funktionen att alla ingångsvärden täcks av en regel och att regulatorn gör det du vill den ska göra
6. Spara hela fuzzyregler som en fil med export menyn (till exempel som "vm_F")

Efter att du har skapat en fuzzyregler kan du bygga in det i ett matlabprogram på %Regulator-delen. Öppna till exempel funktionen ”vm_P” och gör följande anpassningar:

1. Använd funktionen ”readfis” i intialiseringsdelen för att ladda in fuzzyregulatorn som du har sparat som fil och tilldela ett fuzzy-objekt. Till exempel:
   fis=readfis('vm_F')
2. Istället för P-reglern i programmet bestämmer du styrvärde u med funktionen ”evalfis()”. Argumenter till funktionen är fuzzy-objektet som du skapade i punkt 1 ovan och en sammansatt vektor med ingångarna i den ordning som de förekommer i din fuzzyregulator. Om du till exempel använder ”e” och ”h1” i din regler så blir uppropet:
   u=evalfis([e h1],fis);
3. Du får sedan inte glömma att avrunda värdet för u, då Arduinos analoga utgångar bara kan hantera heltal.
4. Testa om det går att starta visualiseringsverktygen ”ruleview” och ”surview” i initialiseringsdelen direkt efter readfis-funktionen. Om det fungerar så visar det online hur fuzzyreglern fungerar. Funktionerna är:
   ruleview(fis); % Inference Viewer
   surfview(fis); % Surface Viewer

B.2.6.1 Matlabs fuzzy-toolbox

a) Kopiera in en beskrivning av din fuzzyregler genom att i Matlabs kommandofönster utföra följande funktioner och sedan klistra in resultatet:

>> fis=readfis(’filnamn av din fuzzyregler’);
>> getfis(fis);a)
Kopiera in resultatet här:

```
Name      = vm_F
    Type      = mamdani
    NumInputs = 1
    InLabels  =
        Tank2
    NumOutputs = 1
    OutLabels =
        styrsignal
    NumRules = 3
    AndMethod = min
    OrMethod = max
    ImpMethod = min
    AggMethod = max
    DefuzzMethod = centroid
```

a) Gör alla ändringarna i koden för funktionen ”vm_fuzzy”. Kopiera in %Regulatorn-delen från din kod:

```
u(i)=evalfis(h1(i), fis);
u(i)=min(255, round(u(i)));
```

Spara hela funktionen som ”vm_fuzzy” under samma filnamn.

B.2.7 Kaskadreglering av vattenmodellen (function vm_kaskad)

Utgå från en av de tidigare programmerade regulatorer, exempelvis ”vm_P” eller ”vm_PID”.

B.2.7.1 Programmering av kaskadregleringen

Programmera den inre och den yttre reglerkretsen enligt pseudokoden från uppgift A.6.2.c). Välj en samplingstid för den yttre reglerkrets som är 5 gånger större (dvs långsamare) än samplingstiden av den inre reglerkrets.

Gör alla ändringarna i koden för funktionen ”vm_kaskad”. Kopiera in %Regulatorn-delen från din kod:

```
    % PI- som yttre reglering i kaskad-reglering
    if ((mod(i,5)==0)|| 1==1)
       errorsum=(e(i)+errorsum);
        u(i)=kp*(e(i) + (errorsum * (dT/Ti)));
         pi_u = u(i);
    else
        u(i)=0; % Ingen förändring
         pi_u = u(i);
    end

    e(i) = pi_u - h1(i);
   % PD- som inre reglering i kaskad-reglering
   if(e(i)>0 && i > 1)
         u(i)=kp*(e(i) + Td * ((e(i)-e(i-1))/dT)); % ny styrsignal
    else
       u(i)=0; % ingen förändring
    end

    u(i)=min(255, round(u(i)));
```

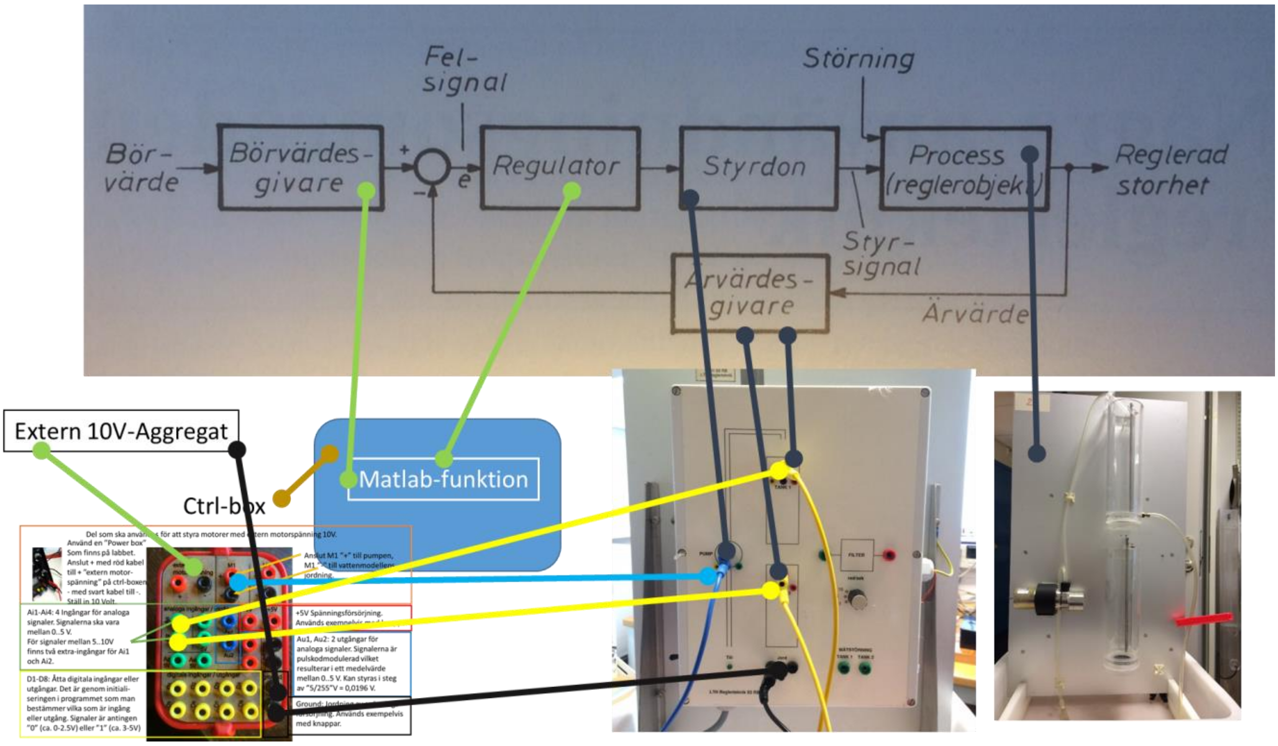
Spara hela funktionen som ”vm_kaskad” under samma filnamn.

C) PRAKTISK DEL: Matlab (R2013b), Arduino-Ctrl-box, vattenmodell

C.0 Labbutrustningen och allmänna anvisningar beträffande experimentens genomförande

C.0.1 Översikt över hela systemet

Bilden nedan försöker visa hur labbutrustningen relaterar till teorin om enkel reglerkretsen. Processen eller reglerobjektet som ska regleras är vattenmodellen med två behållare och en pump. Nivån i både behållare eller tank kan mätas i Matlab med hjälp av anslutningar till ctrl-boxen (analog-in 10V). Pumpen styrs via Matlab genom pumpens anslutning till ctrl-boxen och med hjälp av en extern spänningskällan med 10V som också den ska anslutas till ctrl-boxen. Anledningen att det krävs en extern spänningskälla är att ctrl-boxen inte har tillräckligt med effekt för att driva pumpen.



C.0.2 Kopplade vattentankar

Processen består som redan tidigare illustrerad av två vattentankar i serie till varandra med en övre och en nedre behållare. Utrustningen består av följande delar, se figur nedan:

- Två vattentankar som ligger i serie till varandra. Utflödet av första tanken är inflödet till andra tanken.

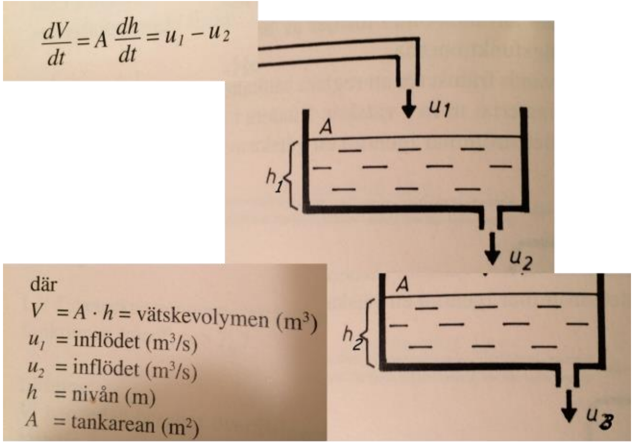


Fig C.0.2a schema av vattentankarna

- En pump som styrs mellan 0..10V tillför vatten till första tanken.
- Vattennivån i första tanken kan störas genom en separat utgång
- Vattennivåmätare i varje tank genererar en signal mellan 0V (tom tank) och 10V (full tank).
- Inbyggd elektronik säkrar ingångar samt anpassar och linjäriserar signalerna till spänningspotentialen mellan 0 och 10V.

Fig. C.0.2b: Bilder på vattenmodellens fram- och baksida

C.0.3 Anslutning till ctrl-boxen

Vattenmodellen kräver en signalanpassning mellan 0..10V. Arduino Uno som sitter i ctrl-boxen är dock bara utlagd för 0 .. 5V. Problemet lösas för *ingångarna* i själva ctrl-boxen genom en inbyggd spänningsdelare som delar upp 10V i 5V + 5V.

Anslut alltid signalerna från vattenmodellen till 10V-kontakten för analoga ingångar på ctrl-boxen!!

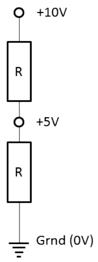
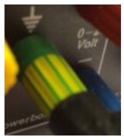


Bild C.0.3a med 10V-analoga ingångar på ctrl-boxen som ska användas

Analoga signalerna som ska anslutas från vattenmodellen till ctrl-boxen är nivåmätningar från tank 1 (h1) och tank 2 (h2).

För *utgångarna* som ska vara mellan 0..10V och kraftigt nog för att driva motorer (som t.ex. pumpar eller fläktar) används motor-shielden som sitter på själva Arduino Uno inne i crl-boxen. Det innebär att det krävs en extern ansluten spänningskälla (t.ex. genom en ”power box” i labbsalen). Den externa spänningskällan styrs sedan genom analogvärden (PWM) på pin 3 (mellan 0..255). Beroende på inställd riktning (pin 12) blir den styrda analoga utgång från motor shielden: antingen 0 .. +10V (pin 12 = 1) eller 0 .. -10V (pin 12 = 0).

För att förhindra problem med svävande potentialer krävs det att olika jordanslutningar (grnd till powerbox, ctrl-box och vattenmodell) kopplas ihop med varandra! Om motorns riktning inte är inställd på pin12=1 finns risk för kortslutningar!!



Analoga utgången för motorstyrningen ansluts från ctrl-boxen till motors ingång på vattenmodellen.

Glöm inte att koppla ihop jordningen på vattenmodellen med jordningen på ctrl-boxen. (Om ni undrar varför testa gärna skillnaden när ni kör uppgifterna).

C.0.3 Anvisningar beträffande experimentens genomförande

Vänligen läs noggrant igenom följande punkter innan du fortsätter med de olika experimenten med vattenmodellen och dess reglering:

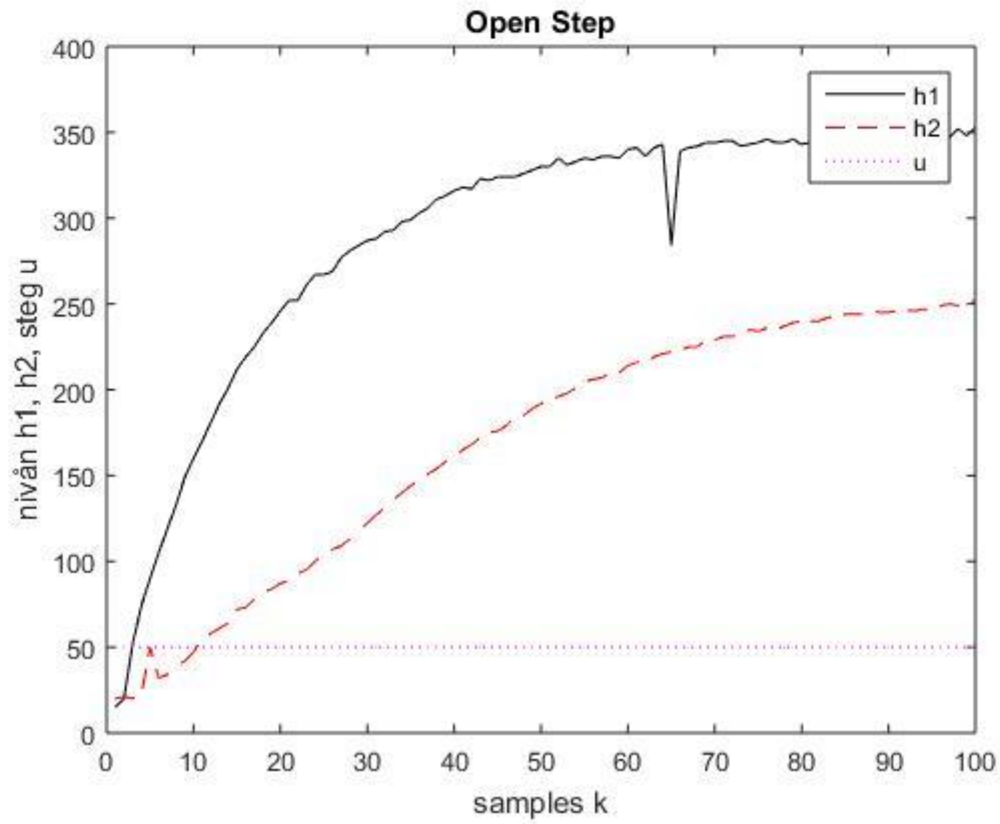
1. Resultaten av alla genomförda experiment ska sparas för att kunna analyseras och jämföras senare. Det enklaste sättet att göra det i Matlab är att ange specifika variabelnamn i samband med funktionens upprop i kommandofönster, exempelvis:
   ```
   >>[h1ost,h2ost,tost,uost] = vm_openstep(a,N,dT,v)
   >>[h1P,h2P,tP,uP,eP] = vm_P(a,N,dT,bv, Kp)
   ```
   Osv
   Det behövs inga ändringar i själva programmet, Matlab överger variablerna i samma ordning som står i funktionsdeklarationen efter nyckelordet ”function”.
2. Alla variabler som finns i arbetsminne kan sedan sparas tillsammans i en binär fil (extension ”.mat”). Föreslaget är att du använder filnamn ”labb1404f”. Kommandon i kommandofönster för att spara är:
   >> save labb1404f
3. Spara dina resultat på detta viset i samma fil efter varje experiment, så att ingenting tappas bort.
4. Om du genomför den praktiska delen i olika delar så kan du börja om där du slutade genom att ladda in alla variabler från filen genom kommandon:
   >> load labb1404f
   (Och sedan sparar du enligt pkt 2 och 3)
5. Angående de olika experimenten med olika regulatorer:
   - Testa dig fram angående olika börvärden. Om den är för hög kan det ta längre tid för att uppnå värdet. Om den är för låg kan fallhöjden av vattnet leda till för mycket turbulens i behållaren som gör det svårt att tolka resultatet.
   - När du har hittat ett bra värde för börvärdet ska du helst använda detsamma i de olika experimenten för att få en bättre jämförbarhet.
   - Minimalkrav för alla experiment är att de genomförs med ett börvärde och så länge i tid att man kan se hur systemet svänger in sig eller försöker svänga in sig. Se punkt 6 nedan om ni vill veta hur ni kan få ut ännu mer av experimenten.
6. För ambitiösa studenter: Syftet med regleringen är att kunna följa börvärdesändringar och att kompensera för störningar. I minimalkraven enligt pkt5 tester vi reglersystemets stegsvar från noll till börvärdet. Vad man därför dessutom skulle vilja testa är ändringar i stegsvaret (till olika nivåer eller att följa en ramp) och kompensation av störningar. Genom att skriva ett litet skript (te.ex. som egen funktion) skulle man för varje regulator kunna köra samma sekvens, exempelvis.
   - Vanlig stegsvar till börvärdet första n-samplingar, sedan
   - ändring i börvärdet (som steg eller ramp) för m-samplingar, sedan
   - uppmana användaren att initiera en störning (genom att öppna extra utlopp) och köra för m-samplingar
   - det vore också bra att i slutplotten ange var del 2 och 3 började

C.1 Stegsvar av det öppna systemet

I uppgiften A.7.1 ritade du upp förloppet av de förmodade stegsvaren för h1 och h2 av den öppna reglerkretsen. Vidare programmerade du funktionen ”vm_openstep” i uppgift B.2.1. Nu blir det dags att köra stegsvarexperimentet på vattenmodellen och att jämföra resultatet från experimentet med dina ritningar.
När du har fått fram resultatet från stegsvaret för h1 och h2 kan du använda mätvärden för att identifiera systemet i form av en differensekvation samt få fram parameterinställningar för en rad regulatorer med hjälp av tumreglerna.

C.1.1 Stegsvarsexperiment

Börja med funktionen ”vm_openstep”. Vi vill kunna skilja ur de olika mätvärdena för h1 och h2 från de olika funktionerna (se kommentarer i C.0.3). Spara därför resultatet som funktionen ger tillbaka i variabler med namn som gör att man vet att de tillhör stegsvaret till öppna reglerkretsen. Exempelvis: [h1-ost, h2-ost, t-ost, u-ost].
Genomför experimentet och klistra in grafen med stegsvaret av den öppna reglerkretsen här:



C.1.2 Jämförelse mellan ritning och resultat

Jämför nu resultatet från experimentet med din ritning i A.7.1:

a) Vad stämmer bra överens?

Börvärdet håller inflödet konstant, vilket stämmer överens med ritningen. Vattennivån hos tank 1 ökar avsevärt snabbare än tank 2, tills båda stabiliserat. Resultatet och ritningarna stämmer bra överens.

b) Vad är mest annorlunda och varför? Hur förklarar du skillnaden om det finns någon?

Dötiden hos tank 2 i resultatet är avsevärt kortare i praktiken än den är i ovanstående ritning. Även vattennivå accelererar betydligt fortare i praktiken än den gör i teorin tills den stabiliserats – alltså att inflödet = utflödet.

C.1.3 Filtrering av mätvärden

Mätvärden ”h1” (h1ost ?) och ”h2” (h2ost) från experimentet med stegsvaret av det öppna systemet behöver filtreras något för att dämpa störningarna innan vi kan använda dem vidare. Du kan använda vilket lågpassfilter som helst som du kanske redan har använt i Tommys del av kursen. Beskriv i så fall hur filtren ser ut och vad den gör:

Alternativet är att du bygger en enkel ”moving average filter”, se också utdraget ur Matlabs dokumentation nedan

Examples

Moving-Average Filter of Vector Data

Find the moving-average of a vector without using a `for` loop.

A moving-average filter is represented by the following difference equation,

$$y(n) = \frac{1}{windowSize}\left(x(n) + x(n-1) + \ldots + x(n - (windowSize - 1))\right).$$

Define the numerator coefficients of the rational transfer function. Use a window size of 5.

```
windowSize = 5;

b = (1/windowSize)*ones(1,windowSize)
```

```
b =

    0.2000    0.2000    0.2000    0.2000    0.2000
```

Define the denominator coefficients of the rational transfer function.

```
a = 1;
```

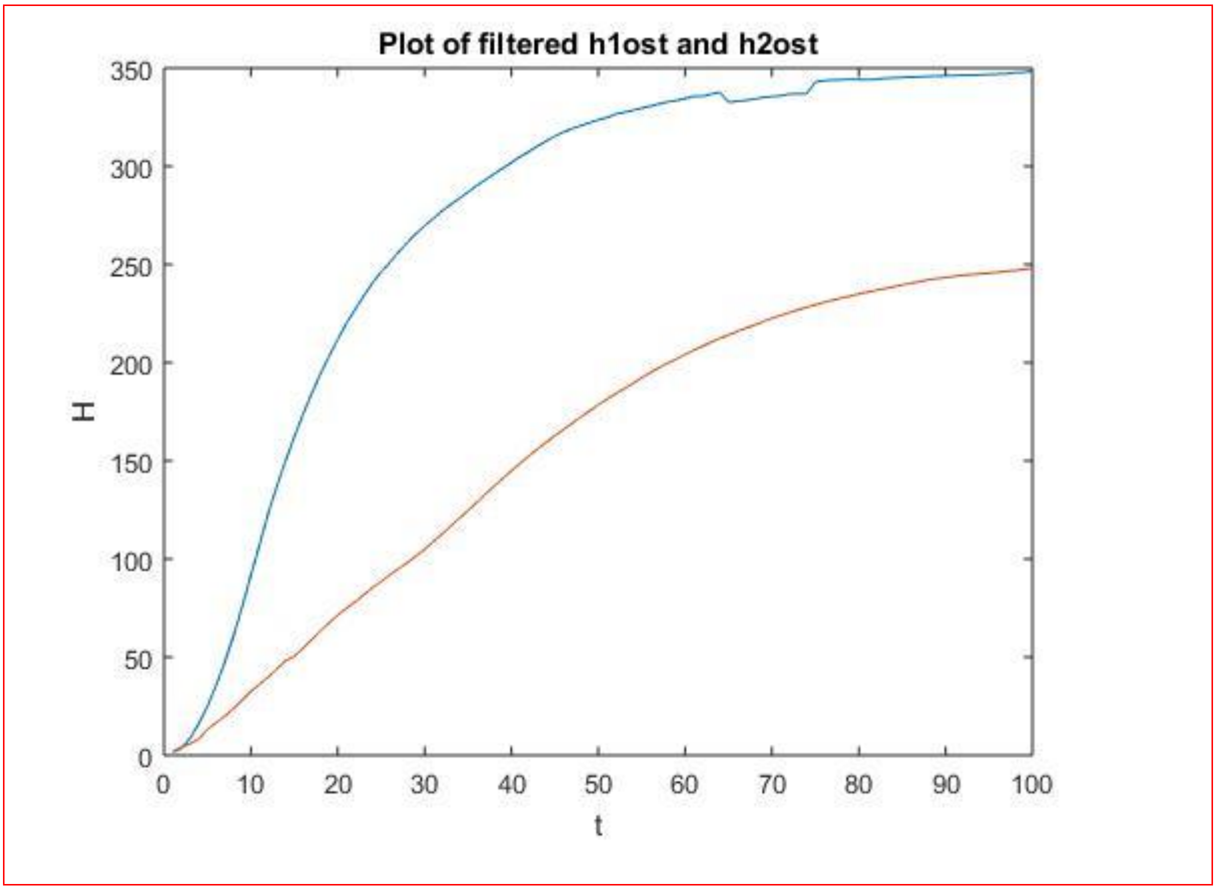
Find the moving-average of the data with a window size of 5.

```
y = filter(b,a,x);
```

Testa gärna med olika värden för windowSize. Istället för ”x” ska du ange h1ost eller h2ost, resultatet y blir då det filtrerade signal.

Klistra in en plot av både filtrerade mätserier (h1ost och h2ost) och ange värdet för ”windowSize” som du har använt:

I nedanstående filtrering använde vi en Windowsize av värdet 10 då den gav en slätare kurva än Windowsize av värdet 10.


Plot of filtered h1ost and h2ost
H
t
0
50
100
150
200
250
300
350
0
10
20
30
40
50
60
70
80
90
100

C.1.4 Identifiering av tidsdiskreta överföringsfunktioner med minsta kvadratmetoden
Följ exempel i kursboken för hur man kan räkna ut parametrarna a och b (för h1) eller a1,a2,b1,b2 (för h2), se också A.8

a) vilka (ange med index, till exempel 34:156) och hur många samplade mätvärden av h1 respektive h2 tar du för att ställa upp ekvationssystemet i matrisform

För att ställa upp ekvationssystemet använde vi index 20 till index 30 för h1(211.8:269.7), och index 30 till index 50 för h2(105:178.5). Vi använder oss av 10 samplingar för h1 och 19 samplingar för h2.

b) Vilka värd får du för a och b?
c)

```
>> y=[265.6; 261.1; 256.3; 251.1; 246.1; 240.6; 234; 227.1; 220; 211.8];

A=[261.1 50; 256.3 50; 251.1 50; 246.1 50; 240.6 50; 234 50; 227.1 50; 220 50;
211.8 50; 203.2 50];
>> A\y

ans =

   0.9263
   0.4716
```

d) Vilka värden får du för a1, a2, b1 och b2?

```
>>   yh2 =
[175.4;172.3;169.2;165.9;162.7;159.5;156;152.4;148.7;144.9;141;137;132.9;128.7;124.7;1
20.7;116.7;112.6;108.8;105];

>> A=
[172.3 169.2 50 50; 169.2 165.9 50 50; 165.9 162.7 50 50; 162.7 159.5 50 50; 159.5 156.0
50 50;156.0 152.4 50 50;152.4 148.7 50 50; 148.7 144.9 50 50; 144.9 141.0 50 50; 141.0
137.0 50 50; 137.0 132.9 50 50; 132.9 128.7 50 50; 128.7 124.7 50 50; 124.7 120.7 50 50;
120.7 116.7 50 50; 116.7 112.6 50 50; 112.6 108.8 50 50; 108.8 105.0 50 50; 105.0 101.5
50 50; 101.5 98.1 50 50];

>> A\yh2

ans =

    1.7771 // a1
   -0.7825 // a2
    0.0305 // b1
         0 // b2
```

C.1.5 Tidsdiskreta överföringsfunktionerna

Ange hela tidsdiskreta överföringsfunktionerna genom att stoppa in de identifierade parametrarna i ekvationen som man får genom z-transformationen. Ett utmärkt tillfälle att träna sig på användningen av Words inbyggda formeleditorn (tillgänglig via ”Insert”-menu).

a) Tidsdiskreta överföringsfunktion mellan u1 och h1:

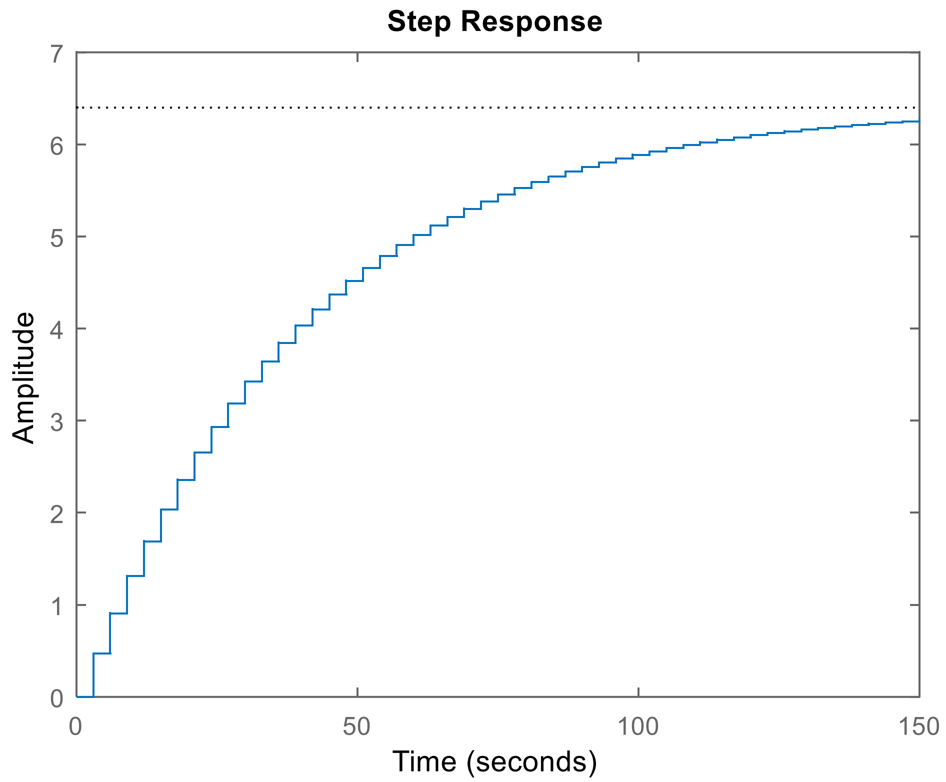
$$\frac{H1(z)}{U1(z)} = \frac{(0{,}4716 \cdot z^{-1})}{(1-0{,}9326 \cdot z^{-1})} = \frac{(0{,}4716\ )}{(Z-0{,}9326)}$$

b) Tidsdiskreta överföringsfunktion mellan u1 och h2:

$$\frac{H2(z)}{U1(z)} = \frac{(0.0305\ \cdot z^{-1})}{(1-1.7771 \cdot z^{-1}-(-0.7825 \cdot z^{-1}))} = \frac{(0.0305\ \cdot Z)}{(Z^{2}-1.7771 \cdot Z + 0.7825\ )}$$

C.1.6 Simulering av stegsvaret för h1 med Matlab
Gör samma som du redan gjorde under B.1.1 men med de aktuella värdena för a, b och h= samplingstid. (Dvs ange den samplingstiden som ni faktiskt använde för att genomföra stegsvarsexperimentet).

a) Simulera stegsvaret med Matlabfunktionen "step(H,t)" och klistra in resultatet som plott här:



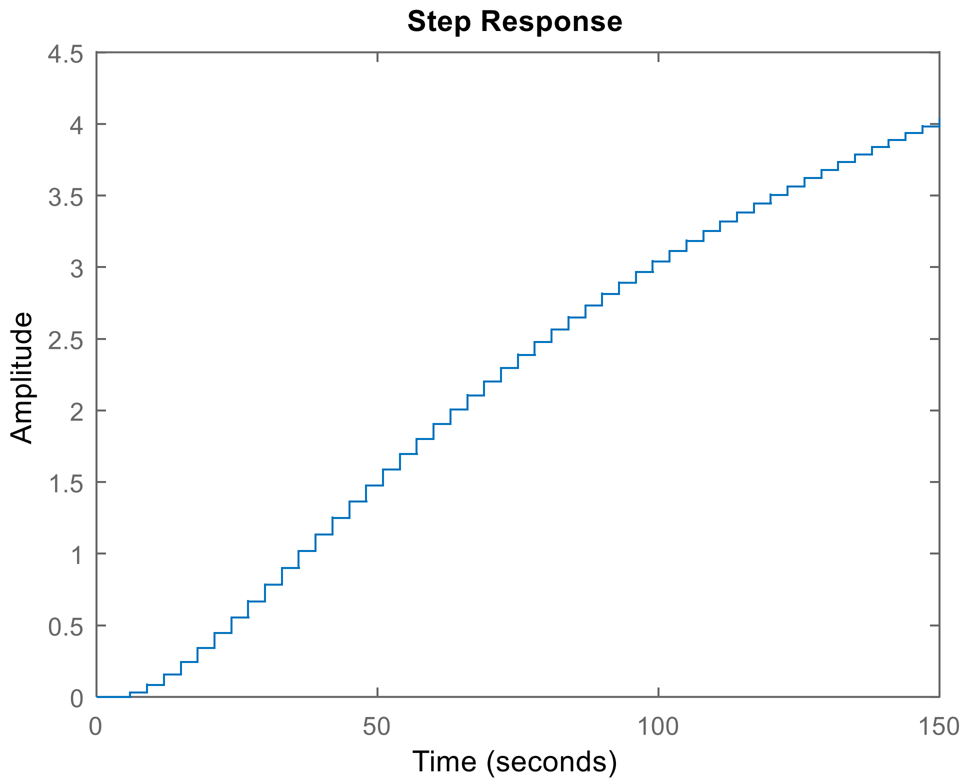
b) Jämför din plott från simulationen med resultatet från experimentet (C.1.1). Vad är skillnaden, är de lika?

Det är väldigt lika i formen dock kan C.1.6 anses något bättre än C.1.1 vilket säkerligen beror på att den genomgått filtreringen.

C.1.7 Simulering av stegsvaret för h2 med Matlab

Gör samma som du redan gjorde under B.1.2 men med de aktuella värdena för a1, a2, b1, b2 och h= samplingstid. (Dvs ange den samplingstiden som ni faktiskt använde för att genomföra stegsvarsexperimentet).

a) Simulera stegsvaret med Matlabfunktionen ”step(H,t)” och klistra in resultatet som plott här:



b) Jämför din plott från simulationen med resultatet från experimentet (C.1.1). Vad är skillnaden, är de lika?

Återigen kan man tyda en goda likheter mellan grafen i C.1.7 och den i C.1.1. Den ur C.1.7 kan även i detta fall anses mycket tydligare vilket kan vara ett resultat av filtrering.

C.1.8 Tumregel som baserar på stegsvaret

Beroende på kursbokens upplaga finns olika tumregler som beskriver hur man kan använda sig av stegsvaret för att räkna ut reglerparameter.

a) Vilka tumregel tänker du använda? (Ange namnet)

Lambdametoden

b) Beskriv hur du gör och vilka värden du får, så att det du gör blir reproducerbar (dvs någon annan får samma resultat om den gör såsom du beskriver):

Lambdametoden skiljer sig från andra metoden genom en parameter som användaren själv kan justera nämligen λ. Vid implementering av Lambda metoden behöver man känna till följande Regulatorparametrar:

T = tidskonstanten
L = dödtid
Ti = regulatorns I-tid
Kp = processförstärkning
Td = regulatorns D-tid

I första steg gäller det att finna Y_s som är den totala förändringen på utsignalen.
Y_s = utsignalens totala ändring
När Y_s är funnen kan man bestämma K_s genom:

$$K_s = \frac{Y_s}{U_s}$$

Man inför sedan e konstant Msom utgör det högsta dödtiden L eller tidskonstanten T.
λ, kan sedan finnas genom:

$$\lambda = p \cdot M$$

Där p utgör en parameter som anger snabbheten/stabiliteten på regleringen enligt:

Långsam reglering:
$\lambda = 3 \cdot T \; om \; T > L$
$\lambda = 3 \cdot L \; om \; L > T$

Medelsnabb reglering:
$\lambda = 2 \cdot T \; om \; T > L$
$\lambda = 2 \cdot L \; om \; L > T$

Snabb reglering:
$\lambda = T \; om \; T > L$

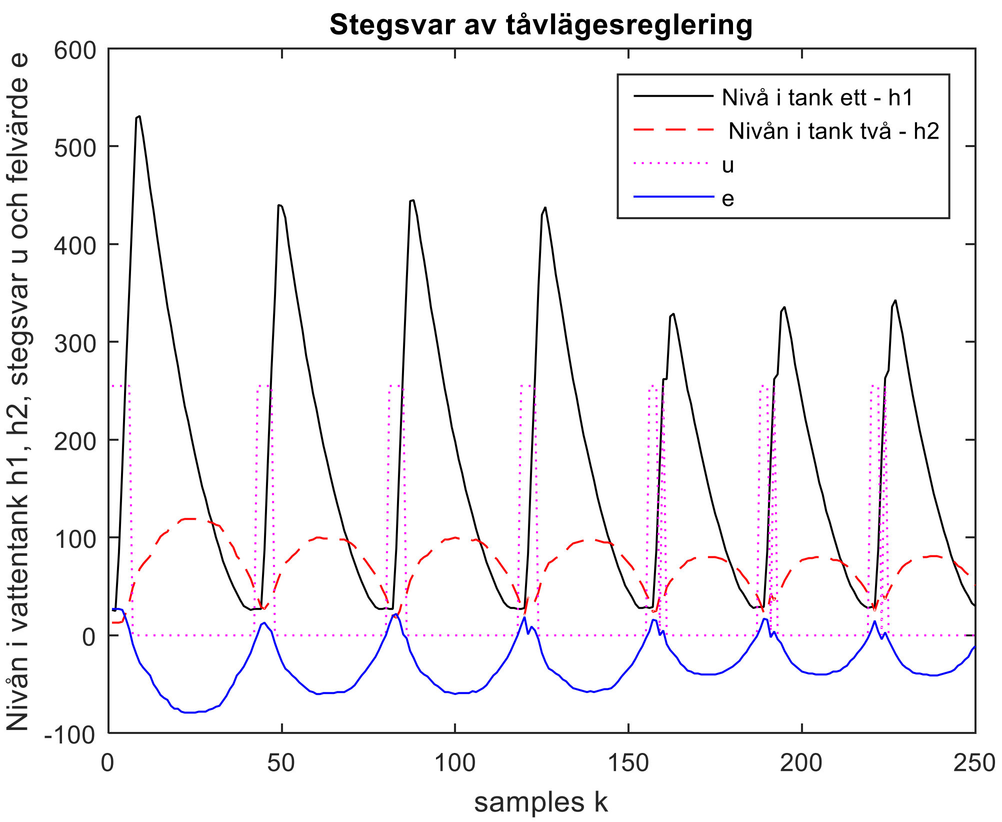
Avslutningsvis kan K enligt:

$$K = \frac{T}{K(\lambda + L)}, \text{ där } T = T_i$$

C.2 På/av- eller tvålägesreglering

C.2.1 Experiment med tvålägesreglering

Reglera vattennivån h2 i andra tanken genom att utföra funktionen ”vm_twostep()”.
Klistra in grafen här:



C.3 P-reglering

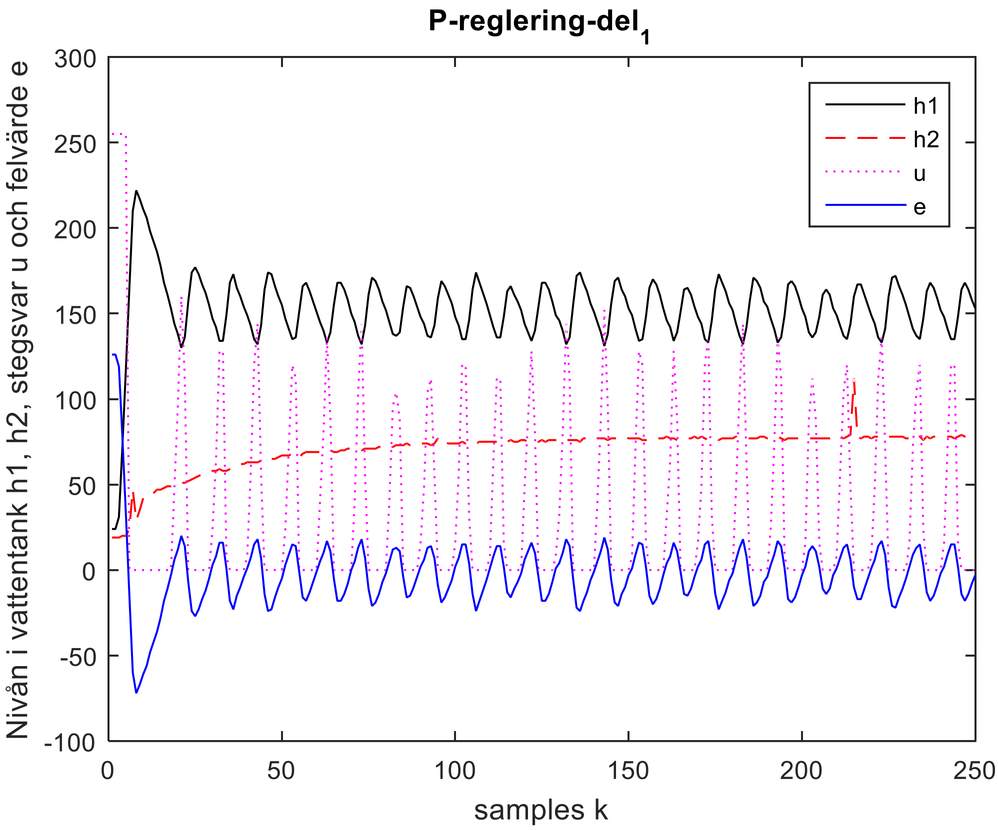
P-regleringen är den första och enklaste, klassiska reglerprincipen. Förstärkningen Kp är den enda parameter som man behöver ställa in. Om den är för låg så blir den kvarstående fel stor och om förstärkningen blir för stor så blir P-regleringen till en tvålägesreglering och systemet börja svänga eller bli till och med instabilt. Detta självsvängning används av vissa tumregelmetoder för att hitta bra inställningar för PI, PD och PID-regulatorer.

C.3.1 Experiment med P-reglering-del I

Reglera vattennivån h1 i första tanken genom att utföra funktionen ”vm_P()”. Prova först med olika värden för Kp för att få en känsla för betydelsen av förstärkningsfaktorn.

a) Vad händer om Kp blir större och större?

När förstärkningen Kp ökar börjar system att svänga kraftigt vilket ger en instabil graf. Det resulterar till att ett för högt Kp i P-regleringen får egenskaperna av en tvålägesreglering.

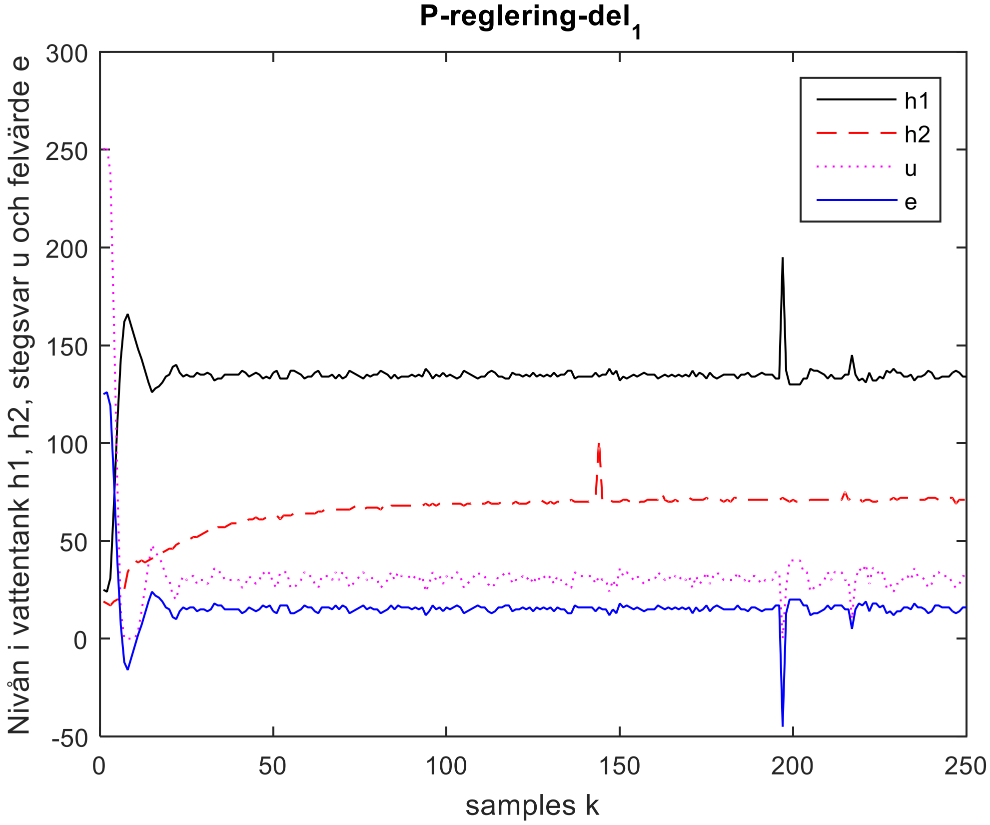


b) Hur stort blir det kvarstående felet i procent av börvärdet?

Det kvarståendefelet beräknas till:

$$\frac{Kvarståendefel}{Börvärdet}\ \frac{Med{:}\ 10}{150} = \text{ca } 6.5\ \%$$

c) Hitta en förstärkning som ger hyfsat bra resultat och klistra in grafen här:



d) Värdet för förstärkningen?

Kp = 2

C.3.2 Tumregel-svängningsmetoden av Ziegler och Nichols[1]

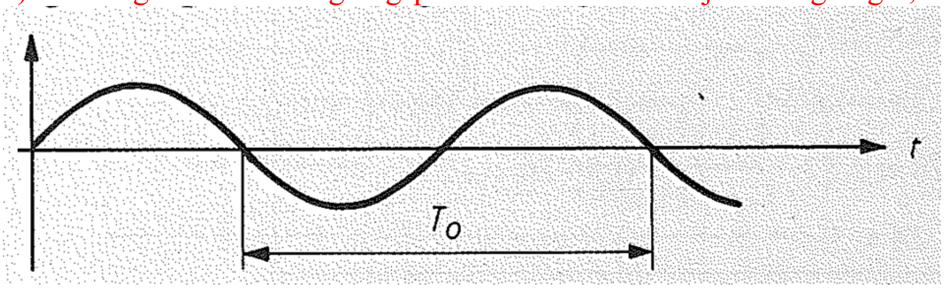
a) Börja nu med en låg förstärkning Kp där nivån h1 i tank 1 inte visar någon översvängning eller självsvängning. (Kan vara svårt att avgöra, välj ett värde för förstärkningen där största felet e (i början) resulterar i ett styrvärde för u som knappt ligger under umax (=255).

a) Valt Kp = 2

b) Öka nu förstärkningen Kp försiktigt i småsteg (svårt att säga hur stort steget ska vara). Korta ner "N" om det känns som ett stegsvar tar för mycket tid. Öka Kp tills regleringen av nivån i behållaren börjar nätt och jämt självsvänga med en tydlig svängning av flera perioder.
Hur stor är den kritiska förstärkningen Kp=K0 där systemet börjar självsvänga?

K0 = 6

c) Hur lång tid är en svängningsperiod "T0" av dessa självsvängningar, se figur nedan?



T0 = 9 sek

d) Använd tabellen nedan och dina värden för K0 och T0 för att bestämma parameter till P-, PI- och PID-regulatorn

Regulatortyp	Parametrar		
	K	T_I	T_D
P-regulator	0,5 K_0	–	–
PI-regulator	0,45 K_0	0,85 T_0	–
PID-regulator	0,6 K_0	0,5 T_0	0,125 T_0

Vad får du för värden?

$P - K: reglering: 0{,}5 \cdot 6 = 3$

$PI - K: reglering: 0{,}45 \cdot 6 = 2{,}7$
$PI - TI: reglering: 0{,}85 \cdot 9 = 7{,}65$

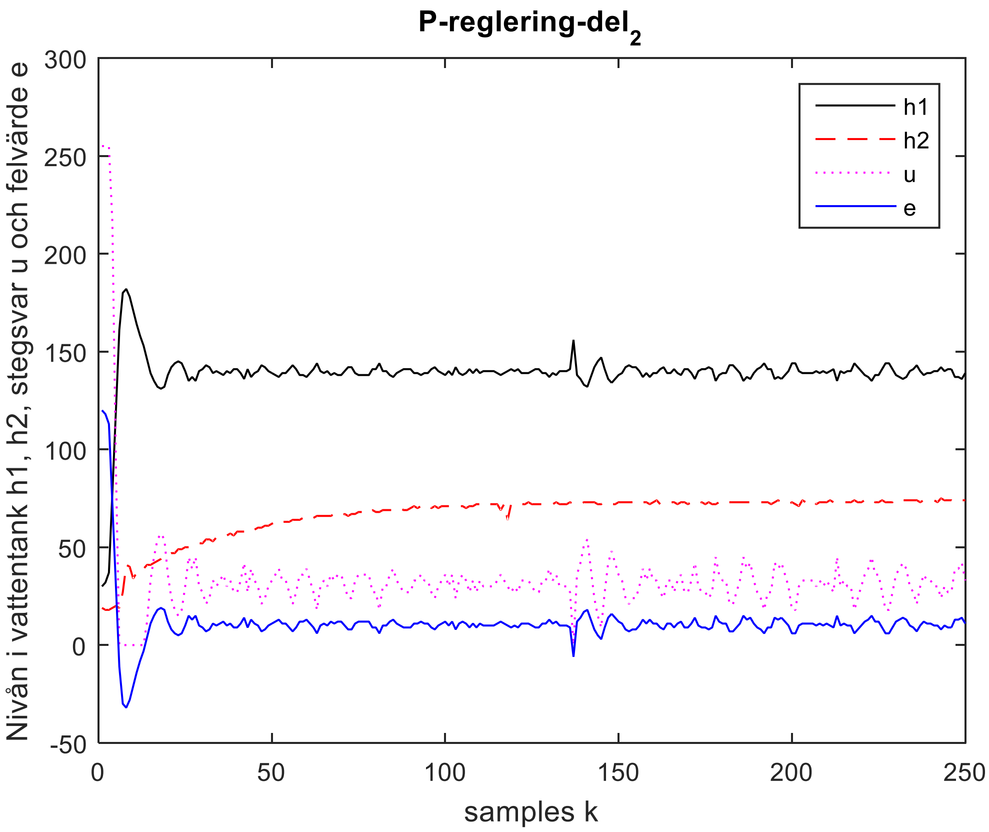
$PID - K: reglering: 0{,}6 \cdot 6 = 3{,}6$
$PID - TI: reglering: 0{,}5 \cdot 9 = 4{,}5$
$PID - TD: reglering: 0{,}125 \cdot 9 = 1{,}125$

Hur jämförbar är dessa värden med de som du fick som resultat i C.1.8b) ?

Det värden som räknades i C.1.8b (se bifogad fil labb1404f då endast teoretisk beskrivning illustreras i detta avsnitt) utgör ingen större skillnad.

C.3.3 Experiment med P-reglering-del II

a) Välj nu en förstärkning Kp som enligt tumregel är hälften av värdet K0, dvs Kp=0,5 K0. Öka "N" igen så att man kan få en uppfattning av kvarstående felet.
Klistra in en kopia av plottet här:



b) Är du nöjd med denna P-regulator? Förklara vad du anser är bra eller lyckat och vad inte?

Det är inte en generellt dålig P-regulator men visst förekommer det en del oregelbunda svängningar som inte avtar med tiden. En orsak kan vara ett alltför högt värde på förstärkningen.

[1] se också deras vetenskapliga publikation från 1942 på its learning

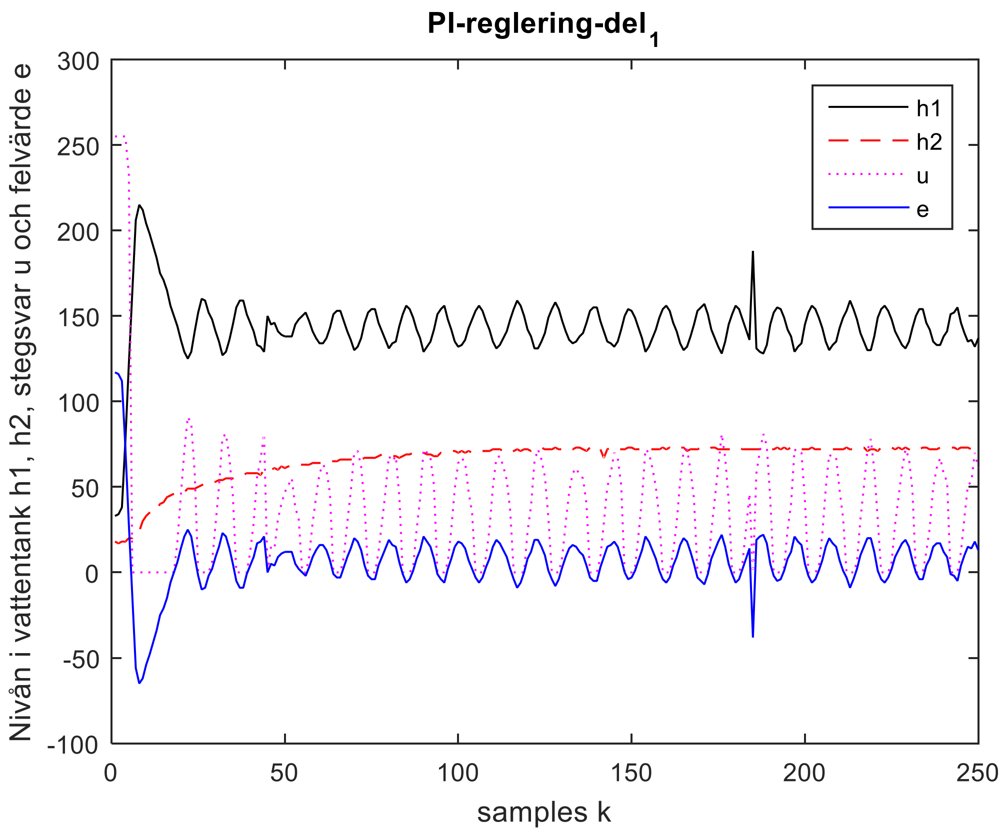
C.4 PI-reglering

Enligt teorin ska det kvarstående felet i en P-reglering kunna regleras bort med hjälp av en fel-integrerande I-del.

C.4.1 Experiment med PI-reglering- del I

Använd dig av parametervärdena för PI-regulatorn från tumregel-svängningsmetoden som du bestämde i C.3.2d).

a) Utför funktionen ”vm_PI()” och klistra in plottet av resultatet här:



b) Hur stort är det kvarstående felet i procent av börvärdet?

Vi har ett börvärde på 150 och ett kvarstående fel på ca 22 vilket ger ett värde på ungefär 14.6 %.

C.4.2 Alternativa parameterinställningar

Använd dig av alternativa parameterinställningar, antingen genom stegsvars-tumregelmetoder från C.1.8b) eller genom att prova Karl-Johan Åströms och Tore Hägglunds förslag att välja K=0,35K0, TI=0,77T0, TD=0,19T0.

Vad blir dina parametrar för PI-regulatorn?

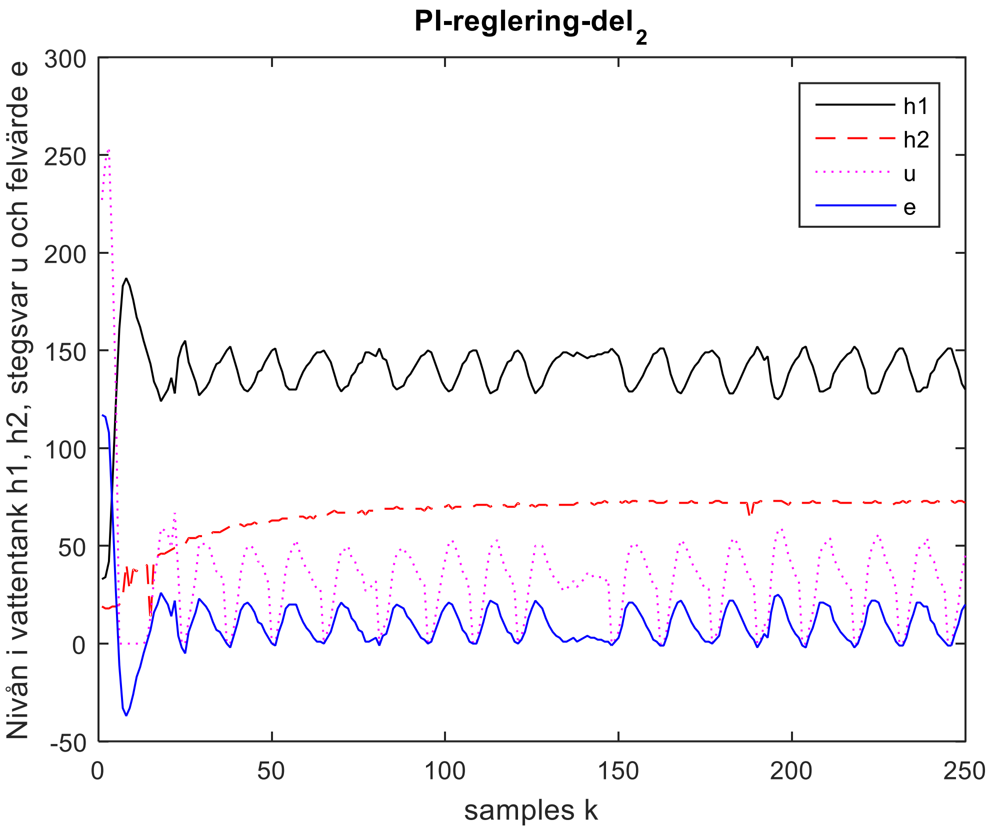
Här används Karl-Johan Åströms och Tore Hägglunds förslag med det nuvarande K0 =5 och T0 = 12.

$PI - K: reglering: 0{,}35\ \cdot 5 = 1{,}75$
$PI - TI: reglering: 0{,}77\ \cdot 12 = 9{,}24$

C.4.3 Experiment med PI-reglering- del II

Använd dig av de alternativa parametervärdena för PI-regulatorn

a) Utför funktionen ”vm_PI()” och klistra in plottet av resultatet här:



b) Hur stort är det kvarstående felet i procent av börvärdet?

Det kvarstående felet är ungefär detsamma som under del 1 d.v.s mellan 13-14 %.

C.4.4 Jämförelse

Vilken av de både PI-regulatorerna anser du är den bättre? Förklara varför.

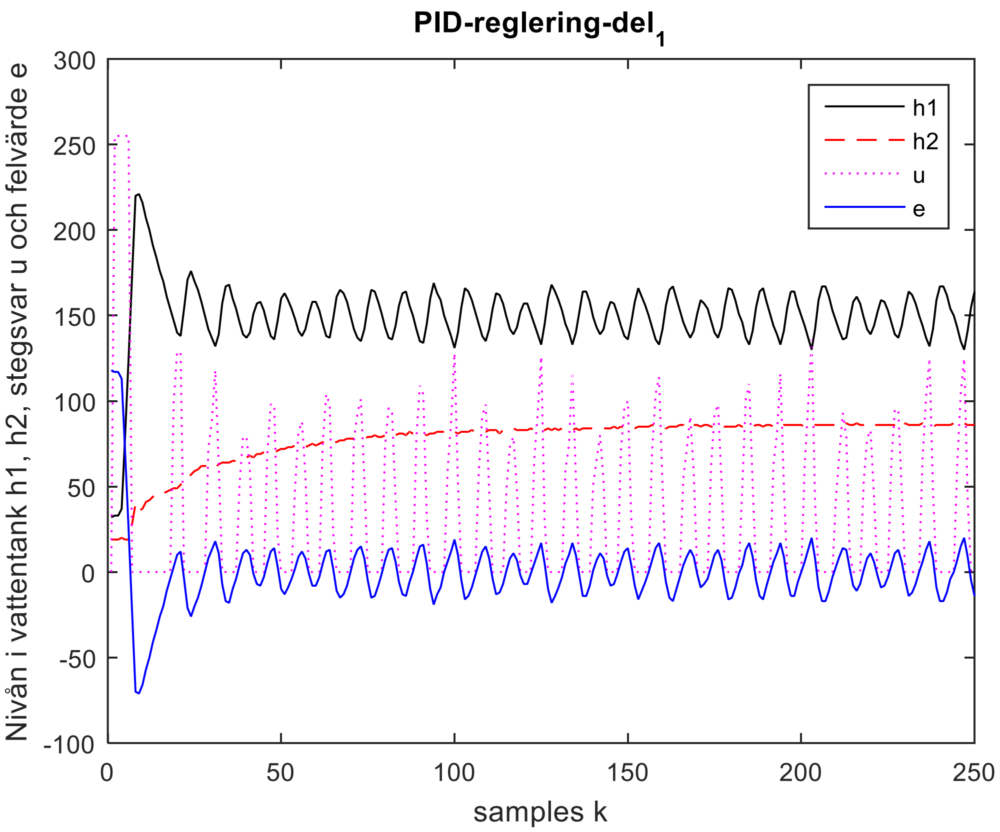
Ur graferna kan man tyda att det inte är någon större skillnad mellan del 1 och 2 även om del 1 kan anses något mer stabilare och mindre störningar som självklart kan beror på yttre störningar och kanske inte skall tas i beaktning under jämförelsen.

C.5 PID-reglering

Nu ska vi också testa om vi kan förbättra de dynamiska egenskaperna av vårt regelsystem genom att tillföra PI-regulatorn ett derivat- eller ”D”-andel, så att det blir en PID-regulator (funktion ”vm_PID()”).

C.5.1 Experiment med PID-reglering- del I
Använd dig av parametervärdena för PID-regulatorn från tumregel-svängningsmetoden som du bestämde i C.3.2d).
Utför funktionen ”vm_PID()” och klistra in plottet av resultatet här:

$PID - K: reglering: 0{,}6 \cdot 6 = 3{,}6$ (3,3 används p.g.a. av körningsfel: 0<u<255)
$PID - TI: reglering: 0{,}5 \cdot 9 = 4{,}5$
$PID - TD: reglering: 0{,}125 \cdot 9 = 1{,}125$

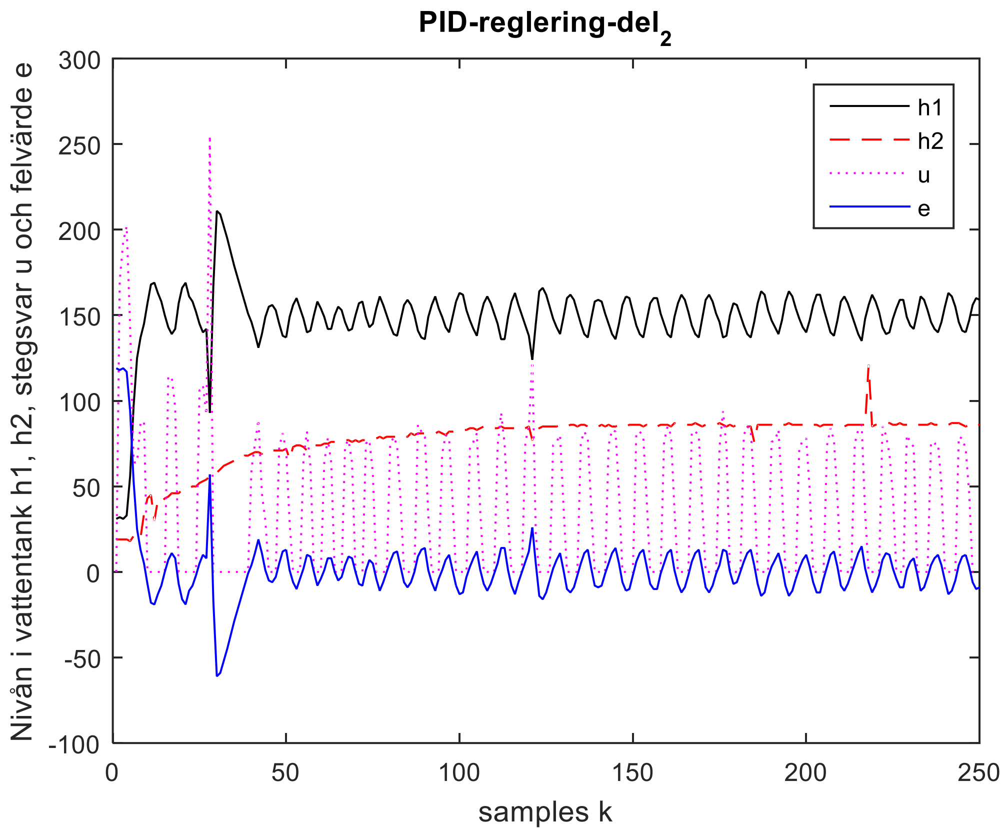


C.5.2 Alternativa parameterinställningar
Använd dig av alternativa parameterinställningar, antingen genom stegsvars-tumregelmetoder från C.1.8b) eller genom att prova Karl-Johan Åströms och Tore Hägglunds förslag att välja K=0,35K0, TI=0,77T0, TD=0,19T0.

Vad blir dina parametrar för PID-regulatorn?

$PID - K: reglering: 0{,}35 \cdot 3{,}3 = 1{,}155$
$PID - TI: reglering: 0{,}77 \cdot 10 = 7{,}7$
$PID - TD: reglering: 0{,}19 \cdot 10 = 1{,}9$

C.5.3 Experiment med PID-reglering- del II
Använd dig av de alternativa parametervärdena för PI-regulatorn
Utför funktionen ”vm_PID()” och klistra in plottet av resultatet här:



C.5.4 Jämförelse
Vilken av de både PID-regulatorerna anser du är den bättre? Förklara varför.

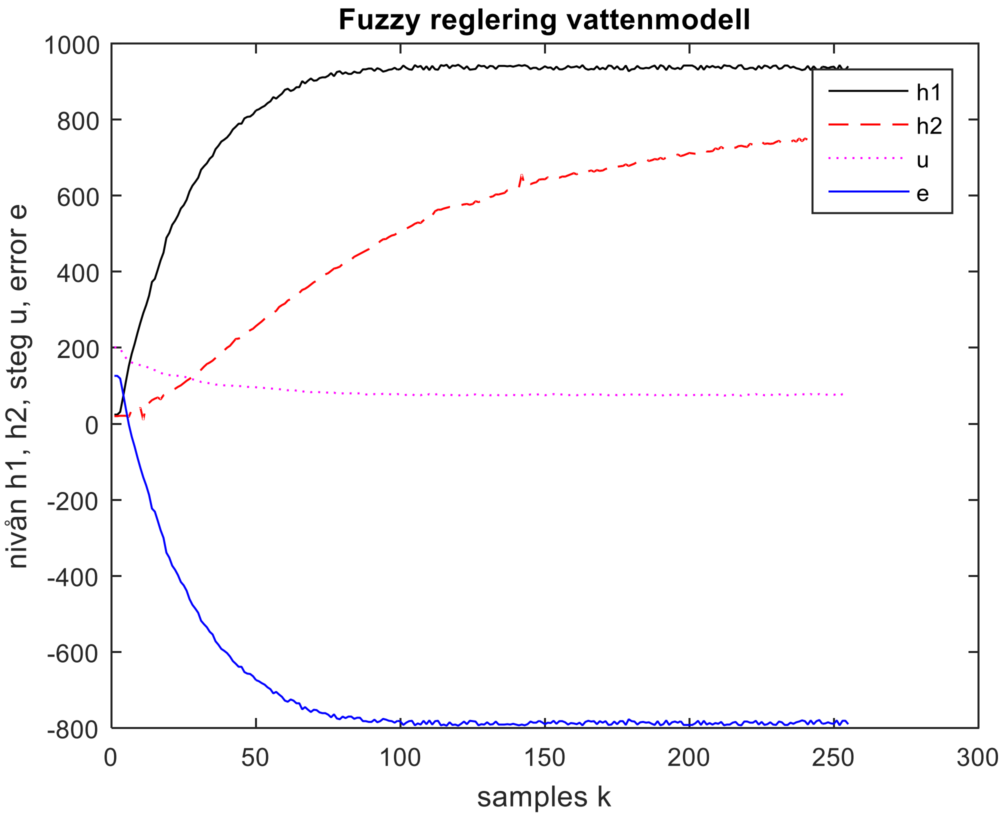
I teorin kan man tycka att PID-reglering del 2 bör visa bättre resultat då parametrarna är anpassade efter systemet, både förstärkningen och insvängningen tas med i beräkning.
I praktiken kan man tycka att skillnaderna är försumbara mellan del 1 och del 2 fastän än del 2 tyder på mindre svängningar vilket ger bättre resultat.

C.6 Fuzzy-reglering

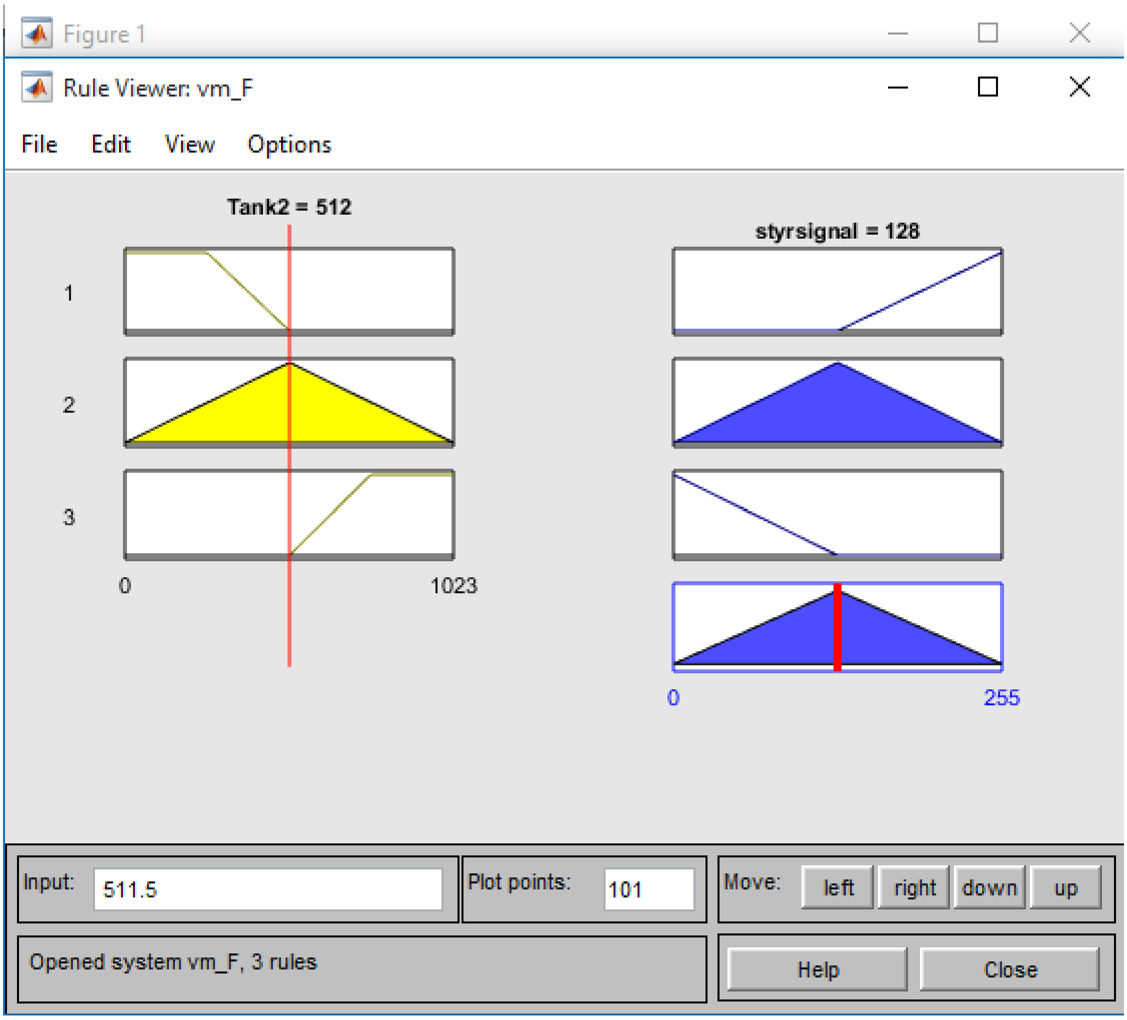
(Om du tröttnar på vattenmodellen kan du välja att antingen köra fuzzy-regleringen eller kaskadregleringen, dvs inte båda två!)

C.6.1 Experiment med Fuzzy-reglering
Kör din fuzzy-regler som du har programmerat i funktionen ”vm_fuzzy()”.

Klistra in plottet av resultatet här:



C.6.2 Ruleview
Försök att lyckas med att få upp ruleview under själva regleringen och klistra in ett prt screen av själva fönstret här:



C.6.3 Utvärdering
Beskriv vad som fungerar bra med din fuzzy-regler och vad som skulle behöva förbättras:

Vi anser att vi kan ha skapat regler som inte är tillräckligt goda och resulterar till en udda fuzzy-reglering. Detta hade möjligtvis kunnat finjusteras för att eftersträva den fuzzy-regleringen som enligt teorin anses god i många sammanhang. Men efter många om och men kan vi endå dra den slutsatsen att vi fick en god graf som illustrerar detta.

C.7 Kaskadreglering

(Om du tröttnar på vattenmodellen kan du välja att antingen köra fuzzy-regleringen eller kaskadregleringen, dvs inte båda två!)

C.7.1 Experiment med kaskadreglering-del I

Kör din kaskadreglering som du har programmerat i funktionen ”vm_kaskade()”.

Klistra in plottet av resultatet här:

C.7.2 Förbättring av kaskadregleringen

Försök att förbättra inställningarna av din inre- och yttre reglerkrets i din kaskadreglering.

a) Beskriv vad du har ändrat och vilka förbättringar du får.

b) Kör din kaskadregleringen igen. Öppna extra-ventilen i mitten av experimentet för att se hur regleringen kompenserar för störningar.
   Klistra in plottet med resultatet här:

C.8 Jämförelsen av resultaten

I uppgift A.7.3 funderade du omkring hur olika regulatorers resultat skulle kunna jämföras med varandra angående de relevanta egenskaperna hos återkopplade system.

C.8.1 Stabilitet

Diskutera stabilitets-egenskapen hos de olika regulatorerna:

a) Vilken regulatorer har minsta översvängningar och stabilast reglerad storhet? Vilken har största svängningar och är minst stabil?

Vi tyckte självklart att PID var bäst men det endast när vi kunde finjustera PID värdena. Vi lyckades inte få en god P reglering vi påverkade resten av PID regleringen. Två läges reglering hade mest svängningar och Fuzzy blev inte som vi hade hoppats på pga reglerna inte var helt rätt inställda.

b) Förklara hur du kom fram till ditt svar? Finns det en kvantitativ metod för att bevisa ditt påstående?

Stabiliteten är självklart godast i Fuzzy enligt teorin som lämnats som referens under labben. Men även PID och kan skapa goda förutsättningar för en god stabilitet.

C.8.2 Snabbhet

En enkel definition av snabbhet är stigtiden och insvängningstiden.

a) Hur definieras det stigtiden och insvängningstiden?

Stigtiden kan man säga är den tid det tar för utsignalen att förändras från tio procent till nittio procent av slutvärdet vid en stegformad börvärdesändring.

b) Vilken regulator har längsta stigtid, vilken kortaste?

Två läges-regleringen har längst tid och PID samt Fuzzy kan ha kortast enligt våra grafer.

c) Vilken regulatorer har längsta insvängningstid, vilken kortaste?

Fuzzy är den med kortast insvängningstid.

d) Finns det en kvantitativ metod för att bevisa ditt påstående? Använd den i så fall och förklara hur du gör.

Den bästa referens är självklart graferna där man kan tyda stigtiden och insvängningstiden för de olika regulatorerna. Kaskad har vi inte provat men kan mycket väl var en med kortast insvängningstid.

C.8.3 Statisk noggrannhet

Med statisk noggrannhet menas det kvarstående felet i pro cent av börvärdet.

a) Vilken regulator har minsta kvarstående felet, vilken störst?

Fuzzy är det enligt teorin (gäller inte vår då)medan två läges-regleringen har mest.

b) Finns det en kvantitativ metod för att bevisa ditt påstående? Använd den i så fall och förklara hur du gör.

Kvarstående fel beskriv i rapporten genom formeln med gränsvärde där z går mot 1 och där kvoten av höjden och Hr och Hp +1. Men detta kan man också tyda på graferna som är en god referenskälla.

C.8.4 Diskussion

a) Blev resultatet av jämförelsen mellan de olika regulatorerna som du hade förväntat dig?
Ja det tycker vi men man måste säga att skillnaden och likheterna kan vara väldig lika ibland vilket får en att undra om man begått ett fel t.e.x Fuzzy.
b) Om inte, varför inte?
P-I-D del regulatorerna var det inte stora skillnader på även om de var klara förbättring för varje implementation av en ytterligare del. Fuzzy var en besvikelse förmodligen p.g.a. av bristande regel inställning.

D) Reflektion och utvärdering

D.1 Vad tycker du/ni var lärorik med uppgiften? (Minst 5 meningar och minst 75 ord!)

Uppgiften var lärorik på många sätt, framförallt att man fick prova teoretiska värden för att sen köra dessa i praktiken. Den var pedagogisk vilket gjorde att man enkelt kunde följa de olika deluppgiften i laborationen och gå tillbaka till en tidigare uppgift om fel begåtts. Den var indelad i rätt delar med teori, praktik med praktik/teori och slutligen ren praktik sista delen som vävd in allt.

D.2 På vilket sätt har ni fördjupat er i något nytt? Vad kände ni från tidigare och på vilket sätt har ni lärt er något nytt utifrån det ni redan kunde? (Minst 5 meningar och minst 75 ord!)

Mycket kunde man redan om de olika regleringarna men man får inte riktigt en bild av vad det egentligen innebär om man endast gör det teoretiskt. Denna laboration satt svart på vitt och man fick en god helhetsbild av beteende för de olika regulatorerna och hur teori blir praktik. Mycket var också nytt och sådant man missat eller inte uppmärksammat under kursen.

D.3 Vad var det svåraste med uppgiften? (Minst 5 meningar och minst 75 ord!)

Det svåraste var nog Fuzzy reglen som krävde något som skiljde sig åt från tidigare regleringar och hur man ställde in den manuellt. En del av teorin var också svår att förstå sig på då den krävde en del matematik och fysik som inte direkt vidare användes i labben särskilt mycket.

Hur mycket tid totalt har ni lagt ner på att lösa uppgiften och hur mycket av denna tid har ni lagt på det som ni anser var det svåraste?

Vi har lagt ner 3 veckor effektiv tid. Svåraste uppgiften får man nog kalla teorin som krävde näste 70% av tiden då den var väldigt omfattande.

D.4 Synpunkter, förslag, kommentarer? (Minst 5 meningar och minst 75 ord!)

Väldigt pedagogisk och lätta att följa under hela labben. Laborationen var ett nöje då man var införstådd inför den partiska delen genom att arbete med teorin i början av labben. Den praktiska delen men graferna var väldigt kort vilket man har full förståelse för då mycket av arbetet gjordes i tidigare delar. Men det hade kunnat vara kul att inkludera en del av den praktiska delen i den teoretiska.