MALMÖ HÖGSKOLA

Inbyggda system och signaler

Styr- och reglerteknik

Labbinlämning 1504e

Utlämning:	23 febr 2016
Deadline inlämning:	11 mars 2016

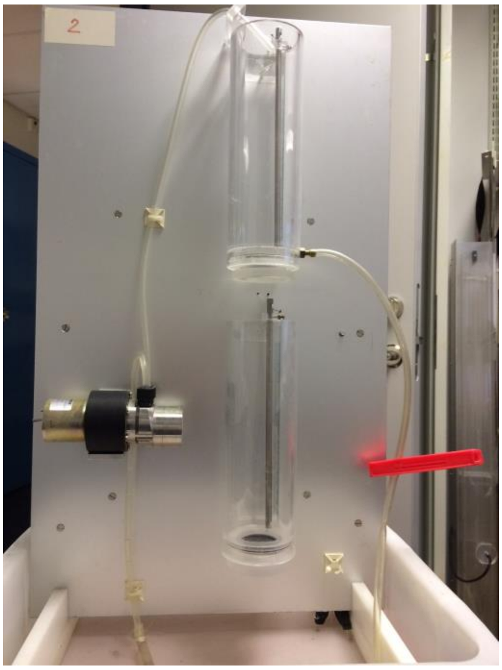
Namn: Marcus Sandberg

Namn: Daniel Petersén

Gion Koch Svedberg februari 2016

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.
I ”Kompendium om reglerteknikutrustningen” på its learning finns en ingående beskrivning av vattentankmodellen.

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 grafer 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

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_kaskad.m
- Binär fil ”labb1504e.mat” med alla variabler från reglerexperimenten, uppladdad 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.

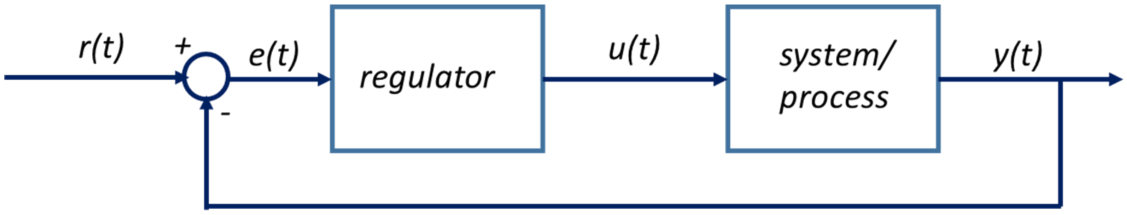


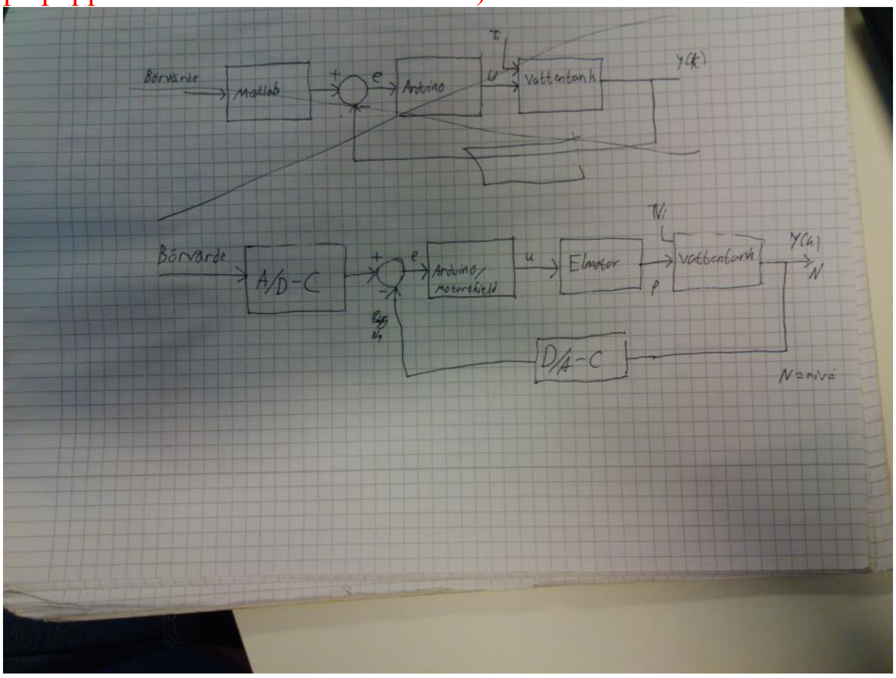
Fig A.1: blockdiagram av en enkel regelkrets

A.1.1 Begrepp i klassiska reglerkretsen
Ange begreppen till de olika signalerna i en klassisk reglerkrets enligt fig A.1:

r(t): _Börvärde____
e(t): _Felvärde____
u(t): _Styrsignal____
y(t):_Reglerad storhet____

Beskriv med egna ord vad som händer i blockdiagrammet genom att följa pilarna ett helt varv: Ett börvärde tas in(värdet som skall uppnås) och då räknas en differans ut mellan börvärdet och Y(t). Detta felvärde skickas till regulatorn som sedan skickar en styrsignal till processen. Sedan korrigeras systemet utifrån detta värde och sedan matas den reglerade storheten ut som används till differans med börvärdet.

A.1.2 Blockdiagram av vattenmodellsregleringen
Rita det fullständiga blockschemat eller blockdiagrammet av vattenmodellen med regleringen samt störningar med korrekta benämningar och begrepp, som exemplet s. 10 i kursboken. Ta med alla delar som ingår (såsom A/D och D/A-omvandlare inklusive skyddskretserna och/eller motorstyrnings-shielden). (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: Vattenreningsverk

2. Exempel: Element

3. Exempel: Ugn

Översikt och jämförelse

	Exempel 1:	Exempel 2:	Exempel 3:
Motorstyrning	Används för att styra pump	Styrning av vattenpump. Ventil.	Strömkälla
Pumpen	Pumpar in vatten till rening.	Pumpa varmvatten in i element.	Värmeelement
inflöde till första behållare	Inflöde av smutsigt vatten.	Flöde av vatten till element.	Strålad värme från värmeelement
utflöde ur första behållaren	Utflöde mot rening.	Tömning av vatten.	Inflöde till termostat
inflöde till andra behållaren	Inflöde till reningsmekanism.	Utstrålad värme till omgivning.	Inflöde till termostat
utflöde ur andra behållaren	Rent vatten, flöde till befolkning.	Fönster eller annat värmespill i omgivningen.	Krets
störningar	Föremål i vattnet.	Luftbubblor, temperaturer i omgivning.	Läckage i ugn
nivån i första behållaren	Mängden smutsigt vatten.	Mängd varmvatten i element.	Spänning i element
nivån i andra behållaren	Mängden rent vatten.	Temperatur i omgivning.	Temperatur på termostat
Börvärde	Rent vatten.	Önskad temperatur i omgivningen.	Önskad temperatur

A.2.2 Gemensamma (system-) egenskaper av tillämpningsområdena

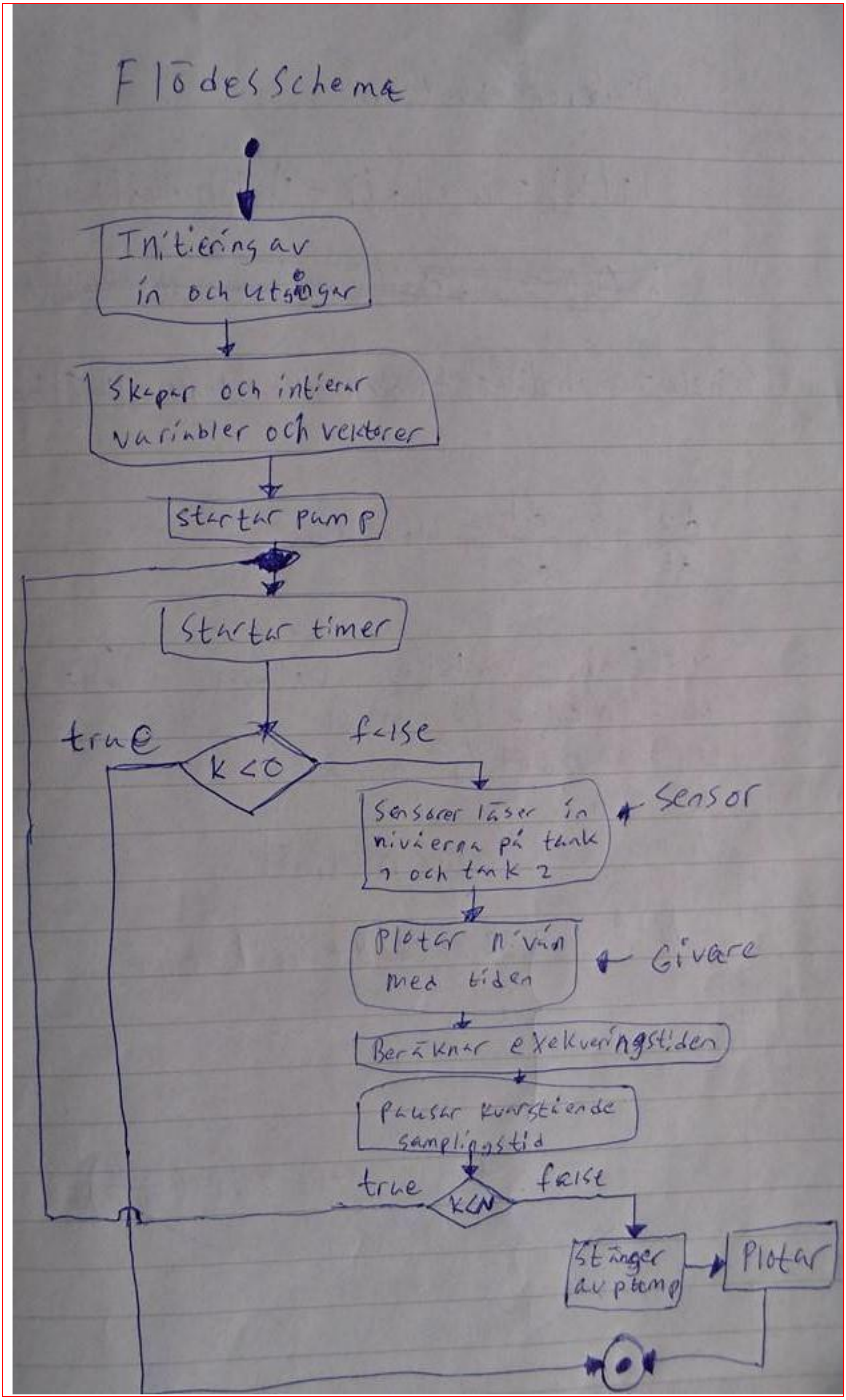
På vilket sätt liknar dina tre tillämpningsområden varandra? Finns det gemensamma typiska systemegenskaper?

Alla områden har gemensamt att de har någon typ av flöde- och nivåreglering. Alla områden använder vid något tillfälle vätskor i sin reglering också.

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. Använd informationen som finns sammanställd i dokumentet ”Kompendium om reglerteknikutrustningen”.

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.


Flödesschema
Initiering av in och utgångar
Skapar och intierar variabler och vektorer
startar pump
Startar timer
true
false
k<0
Sensorer läser in nivåerna på tank 1 och tank 2
Sensor
Plotar nivån med tiden
Givare
Beräknar exekveringstiden
pausar kvarstående samplingstid
true
false
k<N
Stänger av pump
Plotar

A.3.2 A/D-omvandling

a) Med vilken upplösning sker A/D-omvandlingen i den utrustning som används?
A/D-omvandlingen sker med en upplösning på 1024 steg. Dock finns möjlighet för 4096.

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?
Vattenmodellen kommer att ligga i ett intervall mellan 0-3.0 V. Om vi gör en beräkning som liknande: $\frac{3}{1024}$ = 2.93 mV. Alltså motsvarar i detta fall ett binärd steg 2.93 mV.

A.3.3 D/A-omvandling

a) Med vilken upplösning sker D/A-omvandlingen i den utrustningen som används?
Vår Dac har en 8-bits upplösning. Vilket är 2^8 = 256 bitar och ger värden mellan 0-256.

b) I vilken intervall kommer styrvärden att vara begränsat genom programmet? Om vi antar att pumpen ska styras mellan 0 och 2,5 V vad kommer man att få för en upplösning av styrsignalen, dvs vad motsvarar en enhet av den binära signalen för ett analogt utgångsvärde?

$$\frac{Analoga\ signalen}{D\ /\ A\ omvandling} = \frac{2.5}{256} = 9.76\ mv\ per\ bit$$

c) Vilka olika alternativ finns till D/A-omvandling?
Vi kan använda oss av arduinokortets inbyggda DAC, detta sätt tycker vi verkar mest lämpligt, vi kan använda oss av PWM(Pulse Width-modulator).

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:
Relationen mellan styrsignal och felvärde kan beskrivas på så sätt att styrsignalen alltid kommer ta ett av två lägen, av eller på. Då vi pratar om t.ex. en temperatursregulalror och då felvärdet är positivt kommer styrsignalen vara på(hög) och då felvärdet är negativt kommer styrsignalen att vara av(låg). Det ger tvålägesreglering egenskapen att det är en enkel konstruktion men att den samtidigt inte är flexibel.

b) hur ser (pseudo-) koden för regulatordelen ut?
Mata in ett börvärde (Ex. 35 grader celsius).
Läser in ett värde från sensor 1
Läser in ett värde från sensor 2
Räknar ut felsignalen e som är skillnaden mellan börvärde och ärvärde

```
if (felvärde > 0)
   sätter styrsignalens värde till högt (Positivt tecken)
if (felvärde < 0)
   sätter styrsignalens värde till lågt (Negativt tecken)
```

A.4.2 Tidsdiskret P-reglering

Ange Matlabkoden för regulatorblocken som implementerar en P-reglering.

a) Beskriv relationen mellan styrsignalen och felvärdet i egna ord:
Relationen mellan styrsignal och felvärde vid P-reglering(proportionell-reglering) kan beskrivas som att variationerna i styrsignalen kommer att vara proportionella mot felvärdet. Kan beskrivas matematiskt som följande $u = u_0 + K * e$ då u är styrsignal, u_0 är styrsignalens normalvärde, e är felvärde och K är regulatorns förstärkning.

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

```
StyrsignalNorm = u0;
e = börvärde - felvärde;
Förstärkning = k;
return u = StyrsignalNorm + Förstärkning * Felvärde;
```

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:

En PI-reglering är uppbyggd av två delar: Proportionell-regleringen och integrerande-regleringen. Hur en P-reglering funkar har vi redan nämnt tidigare. Båda delarna blir såklart påverkade av förstärkningen även kallad k. Beroende på hur stor fel integralen/felytan har varit tills aktuella tidpunkten är vad som styr den integrerande utsignalen vid en viss tidpunkt. Detta innebär att om systemet reglerfel är noll kommer styrsignalen inte ändras. När fel större än noll blir det strängt växande och antagande när fel mindre än noll.

En PI-reglering är snabbare än en I-reglering och det blir inget kvarstående fel vilket är väldigt viktigt inom reglerteknik.

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)

w(k) är en sampling och wk är ett variabelnamn.

Skillnaden mellan sätten är att w vid punkten k summeras som w-värdet för samplingen tidigare adderat med felvärdet vid punkten k(i) ooch det andra sättet beräknar variabeln wk som wk adderat med felvärdet vid punkten k(ii).

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?

$$\sum_{i=0}^{k} e(i)$$

Då man skulle programmera detta kan man fylla en array med k-antal felvärde med start från första samplingspunkt. Sedan adderar man detta i en for-loop.

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

K: Förstärkning

dT: Tidsderivatan/samplingstid

TI: Integreringstid

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

Skicka in börvärde, få ut felvärde.

Integrera felvärde.

Addera felvärde och dess integral.

Multiplicera dessa med K(förstärkning).

A.4.4 Tidsdiskret PID-reglering

Ange Matlabkoden för regulatorblocken som implementerar en PID-reglering.

```
e(k) = v - h1(k);
sum = sum + e(k);
if(k > 1)
   u(k) = kp * (e(k) + Td *((e(k)- e(k - 1)) / (dT))+ (dT / Ti) * sum);
   u(k) = max(0, min(255, round(u(k))));
   analogWrite(a, u(k), 'DAC0');
end
```

a) Beskriv effekten av D-andelen mellan styrsignalen och felvärdet i egna ord:
D-andelen resulterar att mindre störningar sållas ut. Det motverkar även översvängningar och risk för instabilitet. D-andelen representerar felvärdets derivata.

b) hur ser (pseudo-) koden för PID-regulatordelen ut?
Samma som PI-regulering fasst att man adderar även ett deriverat felvärde.

A.4.5 Tumreglermetoder

a) Vad är syftet med tumregelmetoder och varför behövs dem? Fungerar de alltid eller vad är möjliga begränsningar?
Syftet med tumregelmetoder är att sätta upp generella metoder för att få ut ett ungefärligt parametervärde för regleringeringsmetoder. Metoderna är inte optimala utan de ger värden som man kan utgå ifrån om man har okända parametrar.

b) Vad är skillnaden mellan de olika tumregelmetoderna som beskrivs i kursboken och vad heter dem? (OBS: Olika upplagor av kursboken anger olika tumregler! Ange det som stämmer för upplagan som du använder)
Zieger-Nichols svängnings- och stegsvarsmetoder går ut på att man har en PID-regulator som man ställer in som en P-regulator. D.v.s. att man sätter TD och TI till 0. Sedan ökar man förstärkningen succesivt tills att systemet börjar självsvänga. Då kan man mäta upp periodtiden(T0) och man kan få ut TI och TD.

Lambdametoden för PI-regulatorer ett sätt som är säkrare för känsliga system då självsvängningar kan vara dåligt för det system som man har att göra med.
Enligt lambdametoden skall man ändra processens insignal till stegformat och mäta upp insvängningsförloppet(US). Via denna metod kan man beräkna parametrarna genom att succesivt räkna ut de variabler som man kan räkna ut.

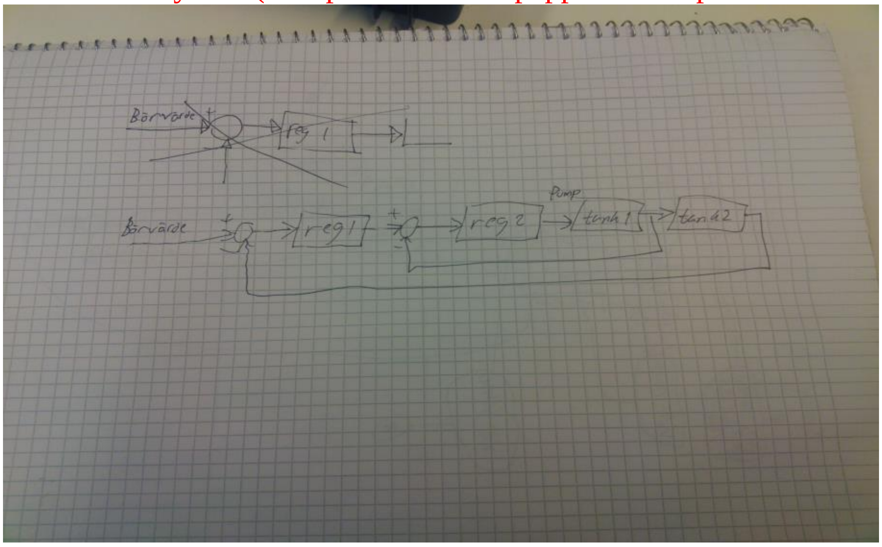
c) Vilka av de förslagna tumregelmetoderna är enkla att använda på vattenmodellen. Beskriv hur du tänker använda dem var för sig, steg för steg:
Ziegler-Nichols metoden är tydligt den metod som vi bör använda oss av. Eftersom att vårt system inte är känsligt för självsvängningar behöver vi inte oroa oss för detta samt att vi får ut de parametrar som vi behöver.

A.5 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.5.1 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.5.2 Kod för kaskadreglering av vattenmodellen

b) Vilka regleralgoritmer väljer du för respektive reglerkrets? Förklara din val.
För yttre reglerkrets hade vi valt PID-reglering för att kunna kontrollera vårt inflöde så noggrant som möjligt och eliminera kvarstående fel.

Då vårt inre system inte påverkas lika mycket av vår reglering som den yttre, kan vi välja en vanlig P-reglering i detta fall, det är snabbt och störningar kommer inte att vara lika mycket av ett problem.

c) Visa med pseudo-kod hur en regulator med kaskadreglering skulle kunna se ut.
(Tips: programmera först regulatorn för den inre reglerkretsen)
Ta in felvärde från bägge regleringar.
Beräkna värde på PID-reglering i inre krets.

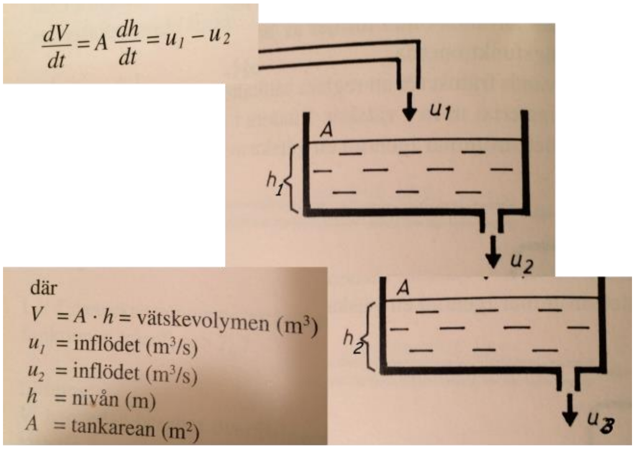
Börvärde sätts som start
Sätter I- och D värden utifrån ziegler-nichols metoden.
Beräknar felvärde utifrån felvärdet multiplicerat med börvärde-ärvärde)
Styrvärde beräknas från förstärkningen multiplicerad med felvärdet adderat med ett integrerat samt deriverat felvärde.

Beräkna P-reglering värde på yttre krets.

Styrvärde beräknas utifrån förstärkning multiplicerat med felvärdet.

A.6 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.



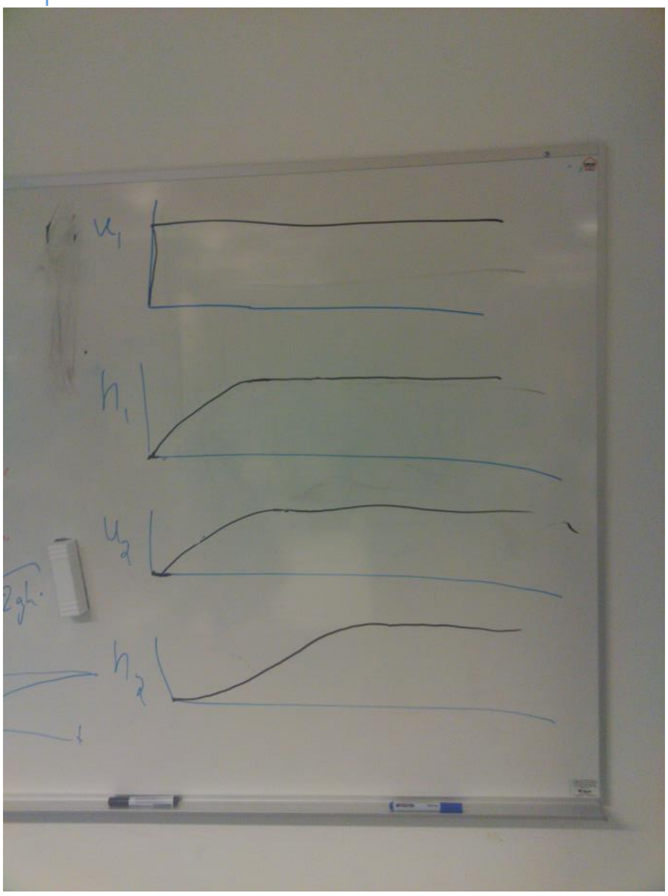
Figur A.7.1 kopplade vattentankar

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

A.6.1 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.6.2 Processtypen för nivån i första eller andra behållaren

a) Utifrån dina funderingar om stegsvaret av h1(t) och h2(t) i förgående uppgiften: hur skiljer sig dynamiken åt av de två nivåförloppen?

h1(t) och h2(t) har väldigt lika stegsvar, dock skiljs de åt genom en fördröjning mellan bägarna, detta beror på att det tar en liten tid innan vattnet i bägare 1 når vattnet i bägare 2.

b) Vilken processtyp har första behållaren?

Första behållaren är en process med en tidskonstant då vattennivån varierar utifrån hur länge vi har haft vårt konstanta flöde till tanken.

c) Vilken processtyp har andra behållaren?

Processtypen för behållare två är en process med dödtid + två tidskonstanter.

d) Hur förklarar du dina svar?

Eftersom att vi måste vänta en tid innan vattnet når tank 2 är vi tvungna att vänta innan vi kan se ett stegsvar hos vår andra behållare, sedan kan man se att kurvtypen efterliknar en process med två tidskonstanter.

A.6.3 Egenskaper hos återkopplade system

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

a) Störningsdämpning

Då det gäller störningsdämpning ser vi PID-reglering som bäst. Detta beror på störningsförhindrandet i ID-delen av regleringen. T.ex. P-reglering förhindrar inte störningar utan det ger oss endast en mer flexibel styrsignal.

b) Stabilitet

Eftersom att störningar kan leda till instabila reglersystem anser vi att PID-reglering är en effektiv regleringsmetod. En regleringsmetod som tvåstegsmetoden ger oss en snabbhet men mycket självsvängning som kan leda till instabilitet.

c) Snabbhet

Våra snabbare regleringsmetoder är P- och tvåstegsreglering. Detta är fallet eftersom att minimalt antal beräkningar behövs för att få ut vår styrsignal.

d) Statisk noggrannhet

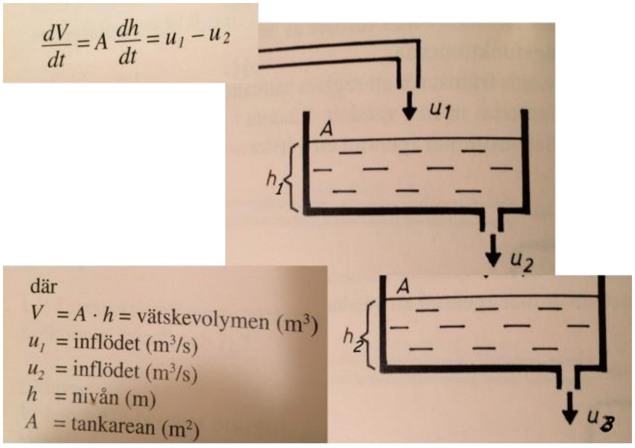
PID-reglering ger oss en bra statisk noggrannhet eftersom att stegsvaret kommer att avta då det närmar sig börvärdet. D.v.s. att vi kommer att slippa en översvängning. De andra tre regleringsmetoder som vi har nämnt, ger oss en återsvängning av något slag.

(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. Studera Matlab-tips i ”Kompendium till reglerteknikutrustningen”.)

Störningsdämpning studerar vi som en funktion av frekvensen då utsignalens amplitud genom störningens amplitud då vi har sinusformade störningar. Då vi har stabilitet kan vi beräkna graden av stabilitet med hjälp av en matematisk stabilitetsdefinition(BIBO), då ett system hålls stabilt om en begränsad insignal alltid leder till en begränsad utsignal. Snabbhetsmätning utförs vanligast genom att undersöka stigtiden. D.v.s. hur lång tid det tar för signalen att ta sig mellan 10%-90% av sitt önskade börvärde.

A.7 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:

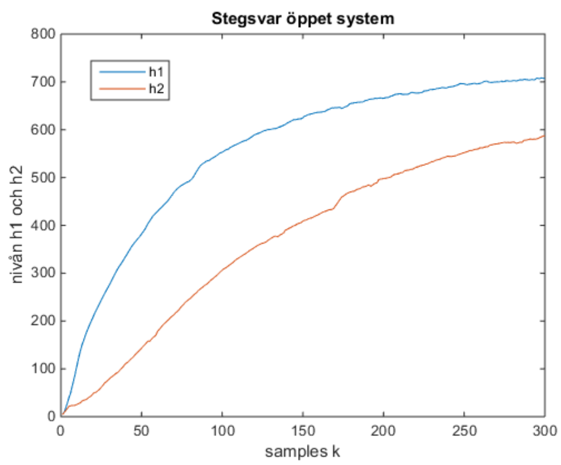


För nivån h1 (allmän differensekvation första ordning):
$h1(k) = a \cdot h1(k-1) + b \cdot u1(k-1)$

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.7: 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.7.1 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=2) och h1(k=3)?

$$A = \frac{44}{23} = 1.91$$

$$B = \frac{2}{325} = 0.006$$

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

$$A = \frac{35}{23} = 1.52$$

$$B = -\frac{-1354}{7475} = 0.18$$

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).

```
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];
B = [21.4 ; 22.8 ; 21.8 ; 17.2];
konst = A\B
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
3.416071e-18.
> In Koefficienter (line 3)

konst =

   1.0e+15 *

    0.0000
   -0.0000
    1.0791
   -1.0791
```

A.7.2 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 A7.1b?

$$h1(k) = \frac{32}{23} * h1(k-1) + \frac{-1354}{7475} * u(k-1)$$

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

$$Y(z) = \frac{(-0.18\, z^{-1}) * 65}{(1 - z^{-1})(1 - 1.52\, z^{-1})}$$

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

$$\mathrm{H(z)} = \frac{-0.18\ \mathrm{z}^{-1}}{1 - 1{,}52\ \mathrm{z}^{-1}}$$

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 A7.1c?
På grund av singuläritetsfel i matlab fick vi inte ut giltiga värden på a1, a2, b1, b2.

e) Hur ser z-Transformationen ut av denna differensekvation andra ordningen?
På grund av singuläritetsfel i matlab fick vi inte ut giltiga värden på a1, a2, b1, b2.

f) Hur ser den tidsdiskreta överföringsfunktionen för andra vattenbehållaren ut som beskriver relationen H2(z)/U1(z)?
På grund av singuläritetsfel i matlab fick vi inte ut giltiga värden på a1, a2, b1, b2.

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. Om det saknas toolboxen på datorn kan man 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.7.2 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.

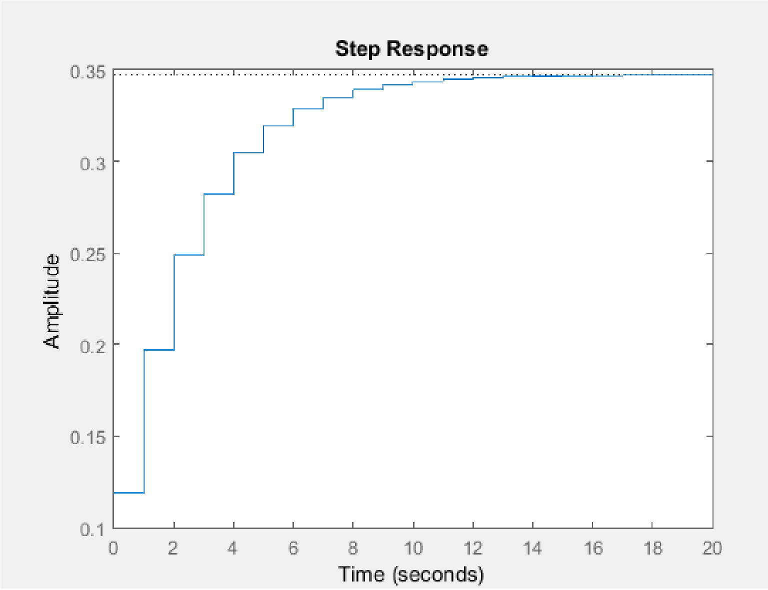
H=tf([…],[…],h)	kommando för inmatning av tidsdiskreta överföringsfunktioner. Det tredje argumentet anger aktuell samplingstid.
H=zpk([…],[…],k,h)	kommando för att skapa en tidsdiskret överföringsfunktion vars poler, nollställen och statiska förstärkning är givna. Det fjärde argumentet anger aktuell 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.7.2.c). Anta att samplingstiden var h=1 sekund. Klistra in det du gjorde från Matlabs kommandofönster

```
num = [-1354/7475 0];
dem = [-35/23 1];
H = tf(num, dem, 1)
```

b) Simulera nu stegsvaret med Matlabfunktionen ”step” och klistra in resultatet som plott här:



c) Jämför din plott från simulationen med grafen som du ritade från hand i uppgift A.6.1. Vad är skillnaden, är de lika?

Bilderna är lika på sätt och vis, de båda har liknande utseende, den plot som vi har fått från matlab har en snabbare utjämning och den är självklart trappliknande då den är av samplingar och inte ritad av oss människor för hand.

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.7.2.f). Anta att samplingstiden var h=1 sekund. Klistra in det du gjorde från Matlabs kommandofönster

Kan inte p.g.a. matlab säger att vi har singularitet i beräkningen av ekvationssystemet.

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

c) Jämför din plott från simulationen med grafen som du ritade från hand i uppgift A.6.1. Vad är skillnaden, är de 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 den egna 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_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. Ändra funktionsnamn till ”vm_openstep”och spara det som fil vm_openstep.m.

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: Initialisering av in- och utgångar
Borde kännas bekant. Ni fick redan göra likadant i tidigare uppgifter. De analoga in- och utgångar är fast tilldelade och behöver inte initialiseras här.

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

DEL D: 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 E: avsluta experimentet
- stäng av pumpen
- plotta slutresultatet
-

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 D: starta stegsvarsexperimentet”
   - Ta bort hela raden som startar pumpen, dvs ta bort a.analogWrite(PWMA,v);
2. I själva for-loopen efter inläsning av analoga ingångarna; 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(k)= v-h2(k); %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(k) > 0
         u(k) = 200; %pumpen kan styras med värden mellan 0..255
     else
         u(k) =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(k) till pumpen.
   - ```
     % Skriva till utgången
     analogWrite(a, u(k));
     ```

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:

```
function [h1,h2,t,u,e] = vm_twostep(a,N,dT,v)
```

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] = vm_twostep(a,N,dT,v)
```

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:

```
plot(t,h1,'k-',t,h2,'r--',t,u,'m:', e);
xlabel('samples k')
ylabel('nivån h1, h2, steg u', 'e')
title('öppet stegsvar vattenmodel')
legend('h1 ', '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(k) enligt formeln som du tog fram i uppgift A.4.2. Tänk också på att ”u” måste vara heltal, större än 0 och får inte vara högre än 255. Ett enkelt sätt är att efter ha bestämt u(k) använda sig av följande kod: u(k)=max(0, min(255, round(u(k)));

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:

```
e(k) = v - h1(k);
u(k) = kp * e(k);
u(k) = max(0, min(255, round(u(k))));
```

b) Förklara koden i raden med:
u(k)=max(0, min(255, round(u(k)));

Värdet u(k) ska ligga i intervallet 0-255. Med funktionen max så bestämmer vi intervallet och detta intervallet går från 0 till en min funktion som sätter att om något tal är större än 255 så blir dem 255 och såklart avrundas även svaret.

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:

```
e(k) = v - h1(k);
sum = sum + e(k);
u(k) = kp * (e(k) + (dT / Ti) * sum);
u(k)=max(0, min(255, round(u(k))));
```

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 Småändringar 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 ”k” är lika med ”1” , dvs när for-loopen börjar?
Det blir en error eftersom att k blir lika med noll.

b) Hur gör du i programmet så att det inte blir ett problem när ”k==1”?
```
if(k > 1)
```

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

```
e(k) = v - h1(k);
sum = sum + e(k);
if(k > 1)
   u(k) = Kp * (e(k) + Td *((e(k)- e(k - 1)) / (dT))+ (dT / Ti) * sum);
   u(k) = max(0, min(255, round(u(k))));
   analogWrite(a, u(k), 'DAC0');
end
```

Spara hela funktionen som ”vm_PID” 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.5.2.c). Välj en samplingstid för den yttre reglerkrets som är 5 gånger större (dvs långsamare) än samplingstiden av den inre reglerkrets. (Dvs att yttre reglerkrets behåller samma värdena under 5 samplingar av inre kretsen).

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

```
%yttre
count = count + 1;
if(count == 5)
  count = 0;
  sum1 = sum1 + ey(k);
  ey(k) = v - h2(k);
  temp = kpy * (ey(k) + (dT / Tiy) * sum1);
end

%Inre
if(k > 1)
  sum2 = sum2 + ei(k);
  ei(k) = temp - h1(k);
  u(k) = kpi * (ei(k) + Td *((ei(k)- ei(k - 1)) / (dT))+ 1 / Tii * sum2);
  u(k) = max(0, min(255, round(u(k))));
  analogWrite(a, u(k), 'DAC0');
end
```

Spara hela funktionen som ”vm_kaskad” under samma filnamn.

C) PRAKTISK DEL: Matlab (R2015b), Arduino-Due, vattentankmodell

C.1 Labbutrustningen och allmänna anvisningar beträffande experimentens genomförande

Vi har ställt ihop ett ”Kompendium om reglerteknikutrustningen” som finns på its learning. Där beskrivs de olika delarna som ingår i utrustningen. Vänligen läs igenom kompendiet först, innan ni fortsätter!! Fig C.1 nedan ger en överblick över de olika delarna som man behöver ha koll på innan man kan påbörja experimenten.

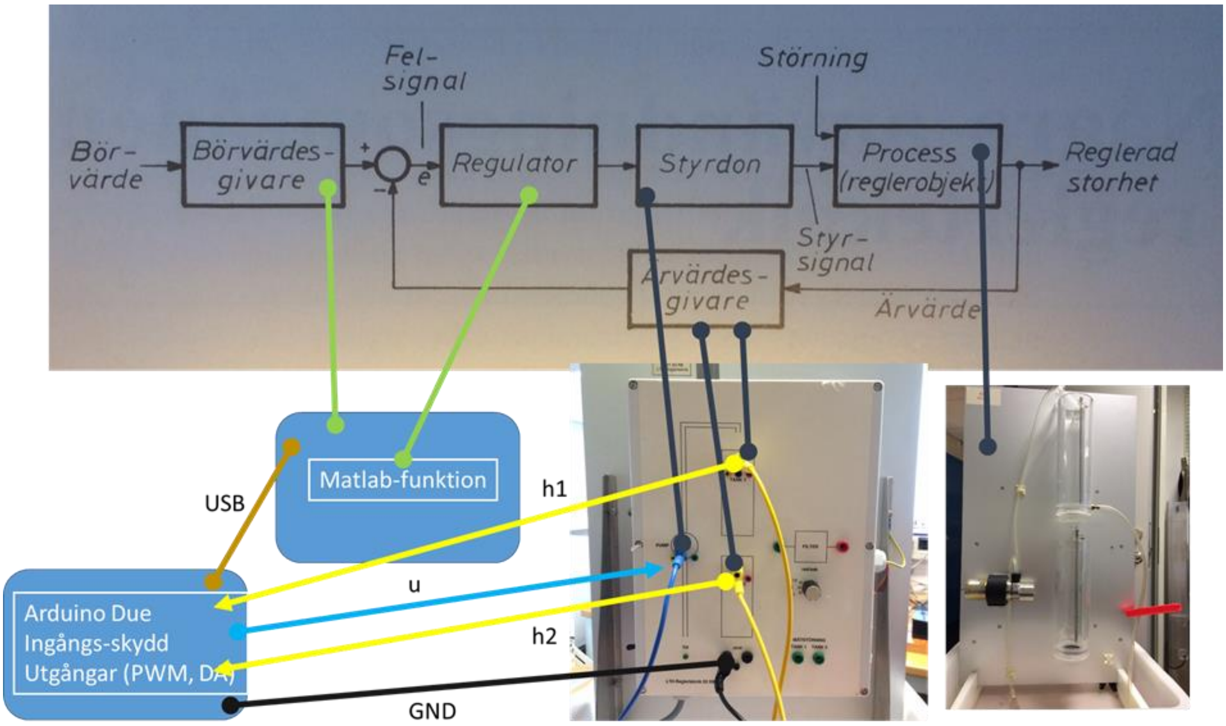


Bild C.1: Översikt över de olika delarna som ingår i utrustningen.

C.1.1 Förberedning av Arduino Due

Börja med att anpassa arduino due till reglerteknikláabben. Detta innebär, att enligt beskrivningen i kompendium:

- Skydda och anpassa analoga ingångarna ’A0’ och ’A1’, genom elektroniska kretsen med op-amp CA3240, beskriven i kompendiums kap. 2.1.
- Bygga upp en elektronisk krets med en op-amp TS921 för anpassningen av PWM-utgång (’DAC1’) och digital/analog utgången (’DAC0’), enligt beskrivningen i kompendiums kap. 2.2.

Innan ni ansluter elektroniska kretsar till arduino dues ingångar och utgångar ska ni först mäta med en extern spänningskälla och en voltmeter att kretsen beter sig som förväntat!

C.1.1.1 Test av ingångsspänning mellan 0V-10V

Välj ingångsspänningar med en extern powerbox mellan 0V-10V och mät spänningen vid utgångerna (pin 1 och pin 7).

Ingångsspänning För h1 och h2	CA3240-pin 1	CA3240-pin 7
0V	0 V	0 V
2V	0.5 V	0.6V
5V	1.2 V	1.3V
6V	1.5 V	1.55V
8V	2 V	2.2V
10V	2.5V	2.7V

Utgångssignalen ska vara linjärt mellan 0V-2,87V när ingångsspänningen varierar mellan 0V-10V. Eventuell behöver förhållanden mellan resistorerna i spänningsdelaren anpassas!

C.1.1.2 Test av utgångsspänning mellan 0V-ca.2,5V

Antingen A) eller B)!

A) Utgången ska producera 0-ca 2.2-2.7V för pumpens styrspänning. Välj ingångsspänningar med en extern powerbox mellan 0V-3,3V och mät spänningen vid utgången (pin 6).

Ingångsspänning TS921-krets	TS921-pin 6
0V	0V
0,5V	0.45V
1V	1.14V
1,5V	1.8V
2V	2.2V
3V	3.12V

B) Om motor shielden används och pumpen ska styras med 0V-10V (vissa äldre vattentankmodeller):

Vi kan ansluta arduino dues 'DAC1' direkt till motorshieldens pin 3 (PWMA). När PWM-signalen är 100% kommer det motsvara 3,3V vilket är 3,3/5=66% av det förväntade maxvärdet. Detta betyder att om pumpmotorn ska kunna styras med 10V när PWM-signalen setts till 100% så måste externa motorspänningen vara 152% av 10V, dvs 15,2V. Testa vad ni får för värden:

		Extern spänning?
DAC1 satt till 100%	Pumpstyrning ska vara 10V	
DAC1 satt till 50%	Pumpstyrning ska vara 5V Hur mycket är det?	

Om motor shield används så ska den kopplas rätt till Arduino genom:

- Arduino pin DAC1 ska förbindas med motorshielden pin 3 (PWMA)
- Arduino GND ska förbindas med motorshieldens GND

- Extern spänningskälla (power-box) ska anslutas till motor shield vid skruvkontaktar ”GND” och ”Vin”: minus ska till ground, plus till Vin!
- Styrningen av pumpmotorn ska anslutas till skruvkontakterna för kanal A på motor shielden: ”+” till pumpanslutningen, ”-” till GND på vattentankmodellen.
- Arduino +5V ska anslutas till motorshieldens +5V

C.1.2 Anslutningen av vattentankmodellen till Arduino Due

Anslut nu kablarna från vattentankmodellen till Arduino Due:

i) Jordning från vattentankmodellen till Arduinos GND
ii) Nivåmätning av övre tanken (h1) till Arduino, t.ex. ’A0’
iii) Nivåmätning av nedre tanken (h2) till Arduino, t.ex. ’A1’
iv) Pumpstyrning från Arduino (TS921 eller motor shield) till vattentanken

Om banankontaktanslutningarna har tagit slut igen i labbet så behöver ni eventuellt löda ihop några igen för att underlätta experimenterandet!

C.1.3 Anvisningar beträffande experimentens genomförande

Vänligen läs noggrant igenom följande punkter innan du fortsätter med de olika experimenten med vattentankmodellen 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:
   ```
   >>[h1o,h2o,to,uo] = 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 ni använder filnamn ”labb1504e”. Kommandon i kommandofönster för att spara är:
   >> save labb1504e
3. Spara era resultat på detta viset i samma fil efter varje experiment, så att ingenting tappas bort.
4. Om ni genomför den praktiska delen i olika omgångar så kan ni börja om där ni slutade genom att ladda in alla variabler från filen genom kommandon:
   >> load labb1504e
   (Och sedan sparar ni enligt pkt 2 och 3)
5. Angående de olika experimenten med olika regulatorer:
   - Testa er 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 ni har hittat ett bra värde för börvärdet ska ni helst använda detsamma i de olika experimenten för att få en bättre jämförbarhet.
   - Minimalkrav för alla experiment är att de genomförs så att man ser hur systemet svänger in sig eller försöker svänga in sig mot börvärdet eller ändringen i börvärdet. Välj därför inte en för kort tid för era experiment!

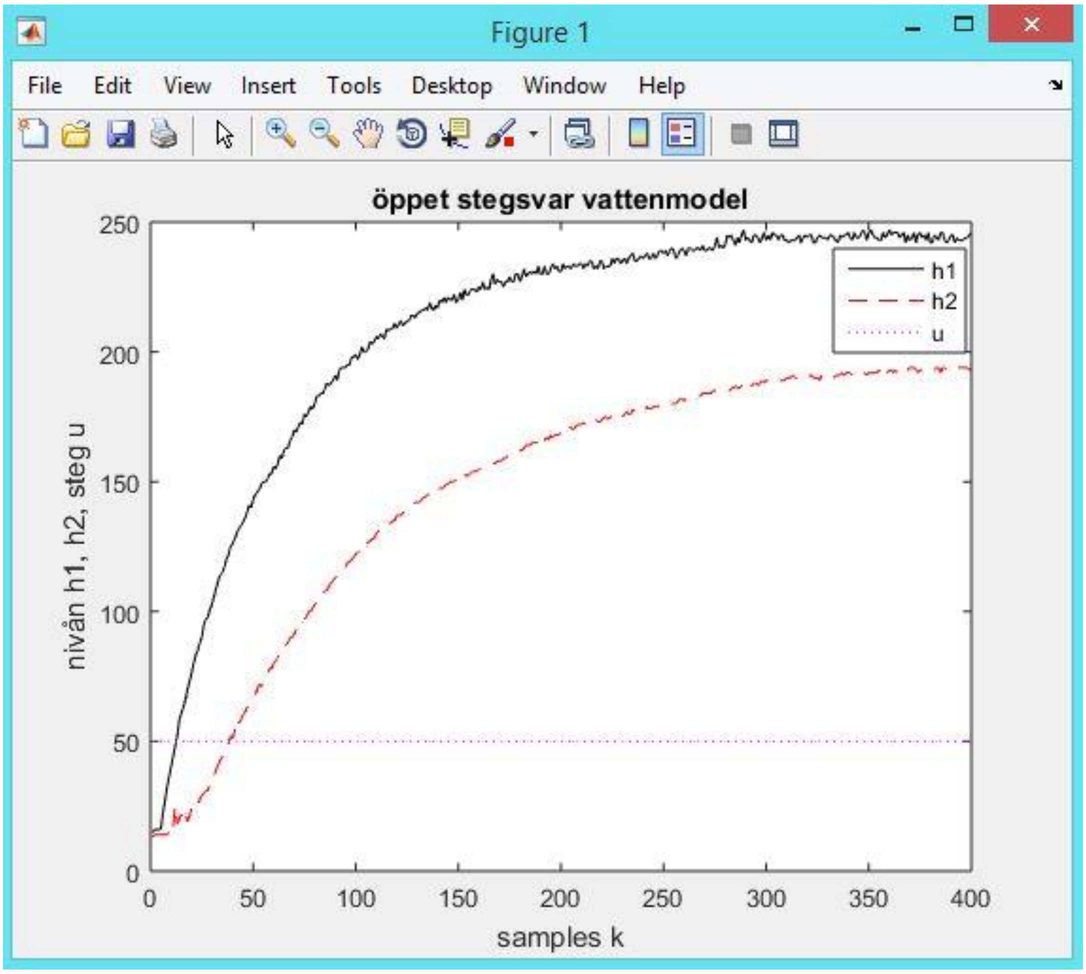
6. För ambitiösa studenter: Syftet med regleringen är att kunna följa börvärdesändringar och att kompensera för störningar. I minimalkraven enligt pkt5 tester vi reglersystemets stegsvar från noll till börvärdet. Vad man därför dessutom skulle vilja testa är ändringar i stegsvaret (till olika nivåer 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.2 Stegsvar av det öppna systemet

I uppgiften A.6.1 ritade ni upp förloppet av de förmodade stegsvaren för h1 och h2 av den öppna reglerkretsen. Vidare programmerade ni funktionen ”vm_openstep” i uppgift B.2.1. Nu blir det dags att köra stegsvarexperimentet på vattenmodellen och att jämföra resultatet från experimentet med era ritningar.
När ni har fått fram resultatet från stegsvaret för h1 och h2 kan ni använda mätvärden för att identifiera systemet i form av en differensekvation samt få fram parameterinställningar för en rad regulatorer med hjälp av tumreglerna.

C.2.1 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.1.3). 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-o, h2-o, t-o, u-o].
Genomför experimentet och klistra in grafen med stegsvaret av den öppna reglerkretsen här:



C.2.2 Jämförelse mellan ritning och resultat

Jämför nu resultatet från experimentet med din ritning i A.6.1:

a) Vad stämmer bra överens?

Kurvformen tycks stämma bra överens på h1, men h2 stämmer inte lika bra överens.

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

Skillnaden som är utseendet på graf 2, dess processtyp tycks vara annorlunda men detta kan bero på tiden vi lät experimentet gå.

C.2.3 Filtrering av mätvärden

Mätvärden ”h1” (h1-o ?) och ”h2” (h2-o) från experimentet med stegsvaret av det öppna systemet behöver filtreras något för att dämpa störningarna innan vi kan använda dem vidare. Ni kan använda vilket lågpassfilter som helst som ni 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;
```

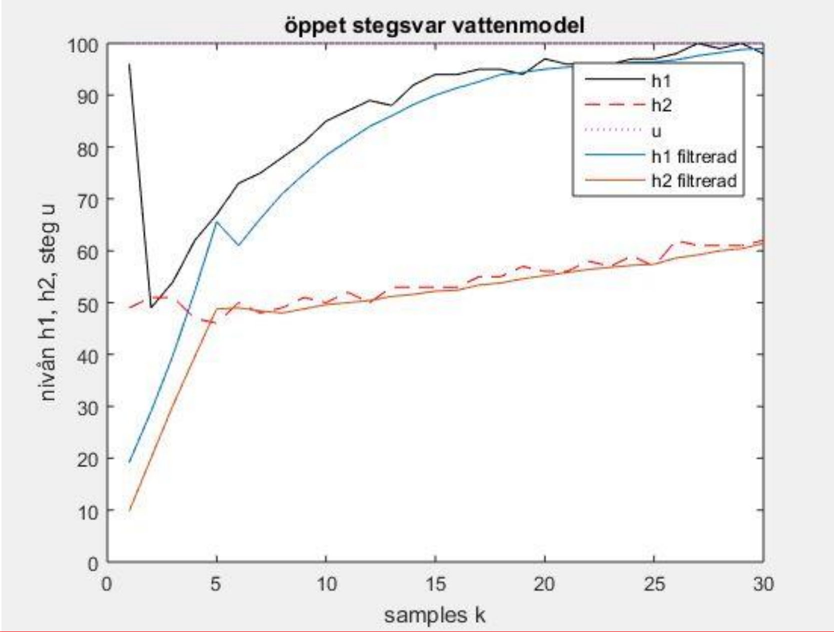
Find the moving-average of the data with a window size of 5.

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

Testa gärna med olika värden för windowSize. Istället för ”x” ska du ange h1-o eller h2-o, resultatet y blir då det filtrerade signal.

Klistra in en plot av både filtrerade mätserier (h1-o och h2-o) och ange värdet för ”windowSize” som du har använt:

windowSize = 5


öppet stegsvar vattenmodel
nivån h1, h2, steg u
samples k
h1
h2
u
h1 filtrerad
h2 filtrerad
0
10
20
30
40
50
60
70
80
90
100
0
5
10
15
20
25
30

C.2.4 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.7

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
h1(5-15): 44, 53, 60, 66, 76, 76, 79, 83, 84, 86, 88
h2(5-15): 29, 30, 29, 28, 26, 26, 28, 28, 25, 23, 20

b) Vilka värden får du för a och b?
A:0.8041
B:0.1656

c) Vilka värden får du för a1, a2, b1 och b2?
A1:0.0718
A2:0.4258
B1:0.1075
B2:0

C.2.5 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:

$$\frac{H1(z)}{U1(z)} = \frac{0.1656z^{-1}}{1-0.8051z^{-1}}$$

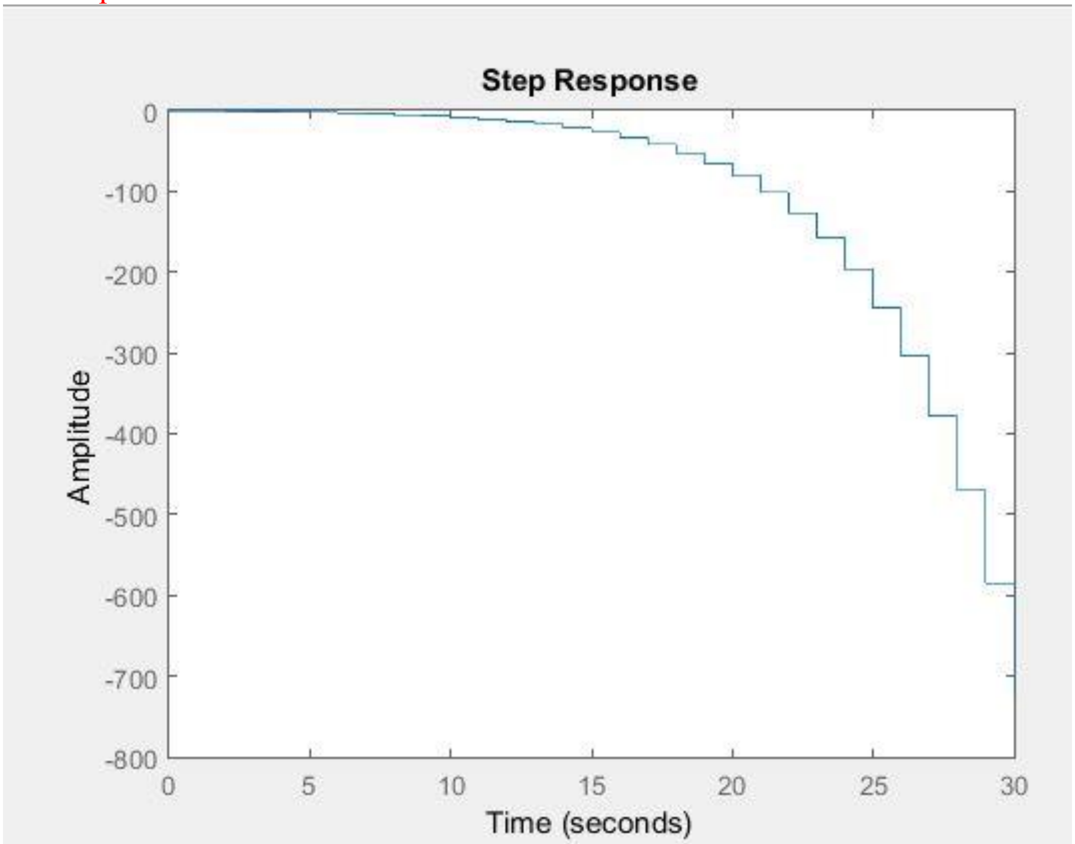
b) Tidsdiskreta överföringsfunktion mellan u1 och h2:

$$\frac{H2(z)}{U1(z)} = \frac{0.1057z^{-1}}{1-0.0718z^{-1}+0.4258z^{-2}}$$

C.2.6 Simulering av stegsvaret för h1 med Matlab

Gör samma som ni redan har gjort 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:



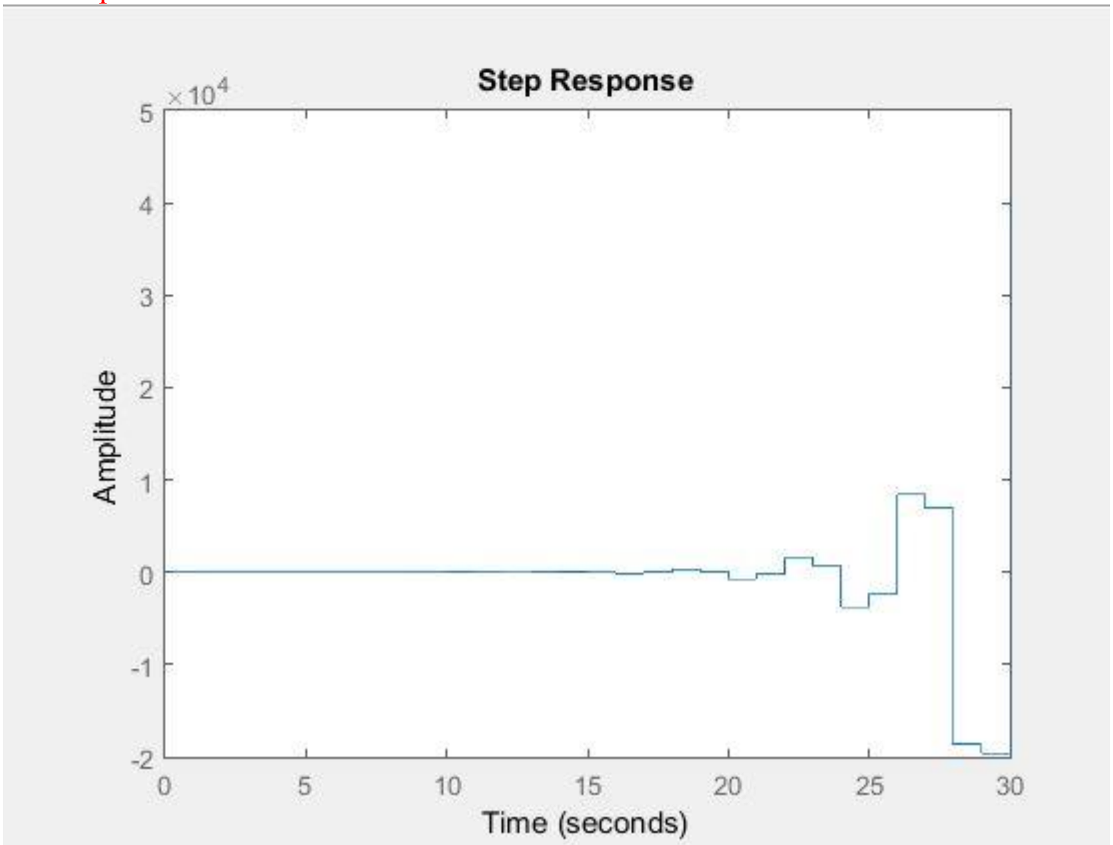
b) Jämför din plott från simulationen med resultatet från experimentet (C.2.1). Vad är skillnaden, är de lika?

Vi vet inte hur vi ska tolka vårt svar, vår plot stämmer inte överens med experimentet. Eftersom att vi har en fallande kurvform på c.2.6 och en stigande på 6.2.1.

C.2.7 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.2.1). Vad är skillnaden, är de lika?

Vi vet inte hur vi ska tolka vårt svar, vår plot stämmer inte överens med experimentet. Eftersom att vi har en fallande kurvform på c.2.7 och en stigande på 6.2.1.

C.2.8 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)

   Vi tänker använda zieger-nichols självsvängningsmetod.

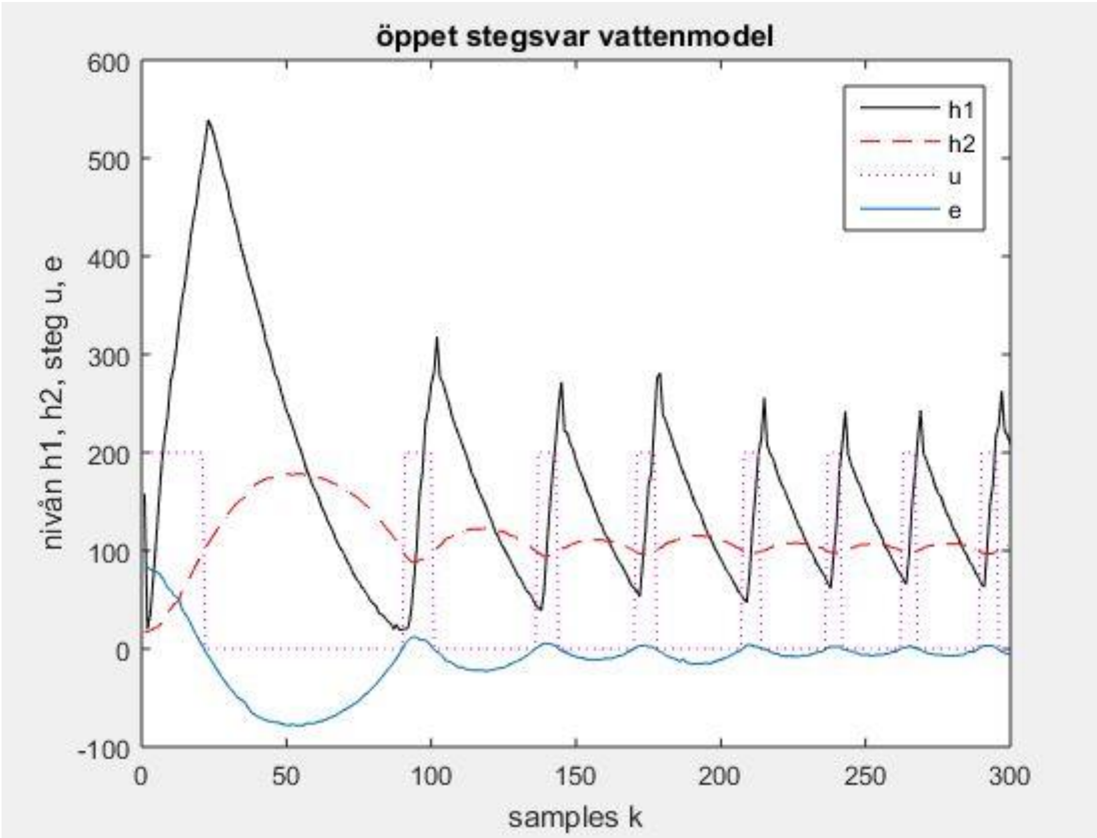
b) Beskriv hur ni gör och vilka värden du får, så att det du gör blir reproducerbar (dvs någon annan får samma resultat om den gör såsom ni beskriver):

   Vi tänker använda oss utav p-reglering, och sedan öka vårt k-värde tills den punkt då vårt system börjar självsvänga.

C.3 På/av- eller tvålägesreglering

C.3.1 Experiment med tvålägesreglering

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



C.4 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.4.1 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.

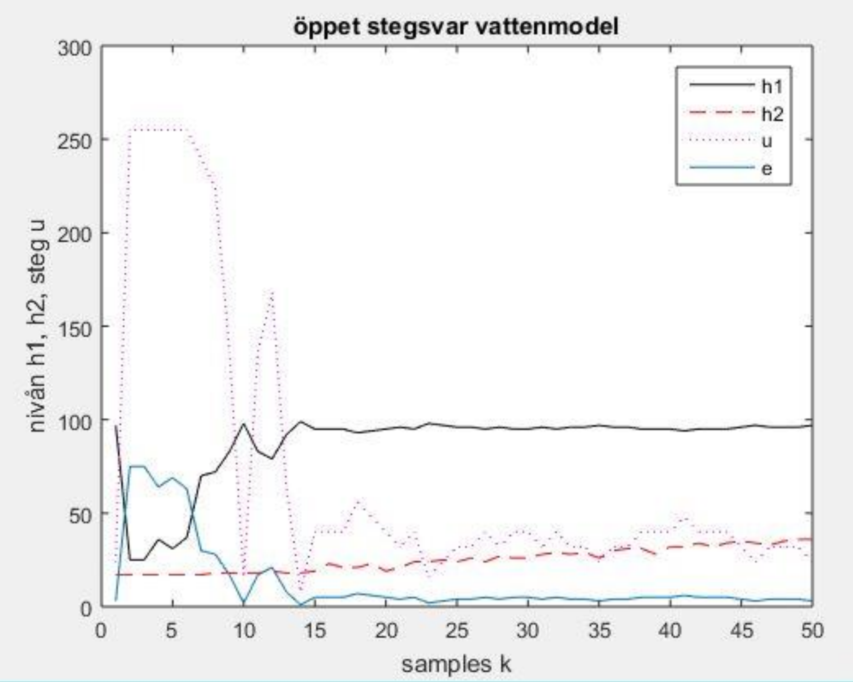
a) Vad händer om Kp blir större och större?

Det gör att vi får en självsvängning och ett instabilt system.

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

Vårt kvarstående fel blir Ca. 50%

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



d) Värdet för förstärkningen som du har valt?

Kp = 8.

C.4.2 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).

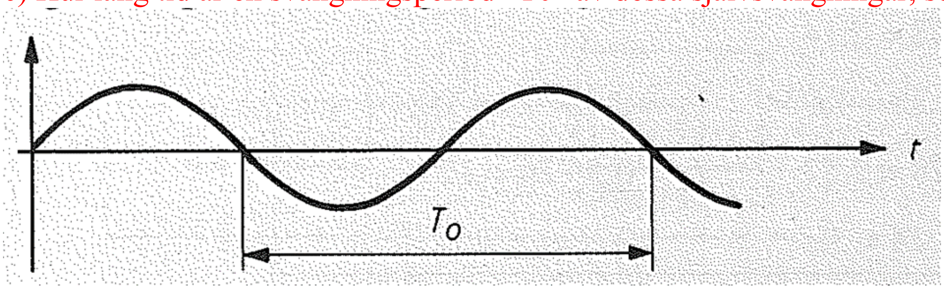
a) Valt Kp = 3.5

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 = 5

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



T0= 6 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-reglering: K = 0.5 * 5 = 2.5

PI-reglering: K = 0.45 * 5 = 2.25. Ti = 0.85 * 6 = 5.1 S

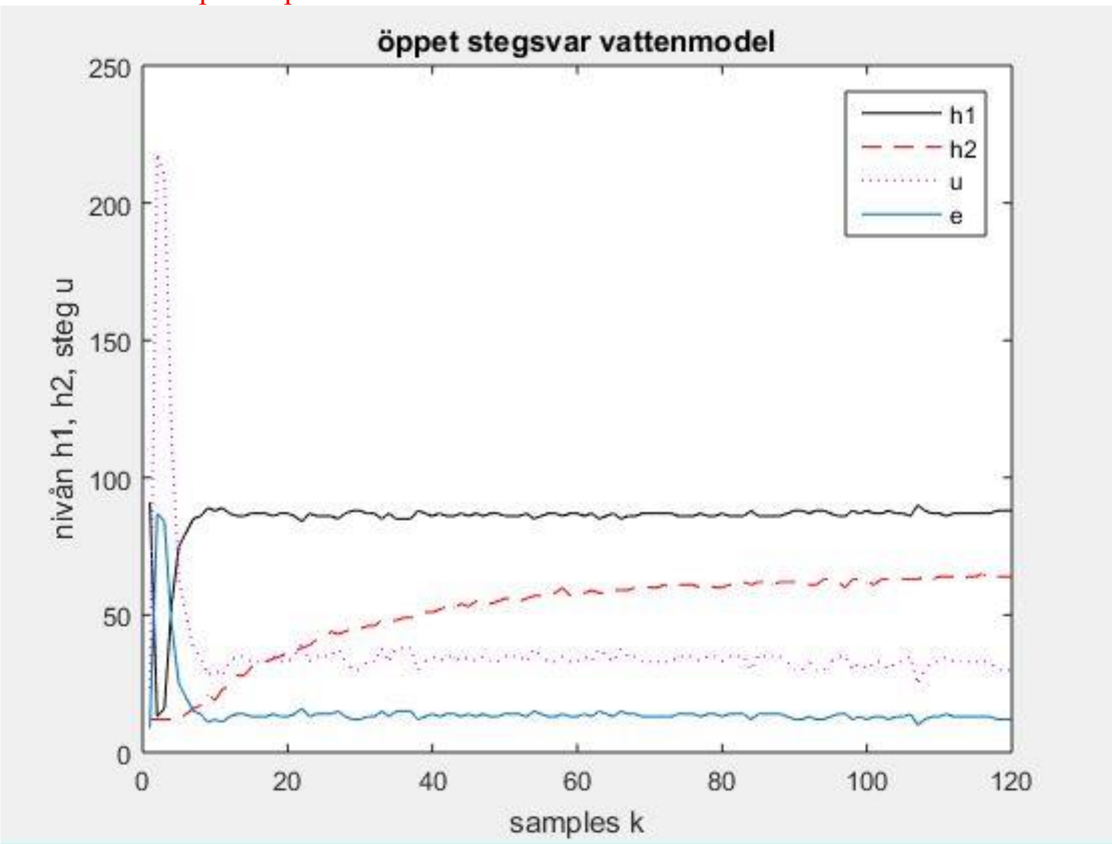
PID-reglering: K = 0.6*5 = 3. Ti = 0.5 * 6 = 3. TD = 0.125 * 6 = 0.75

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

C.4.3 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?

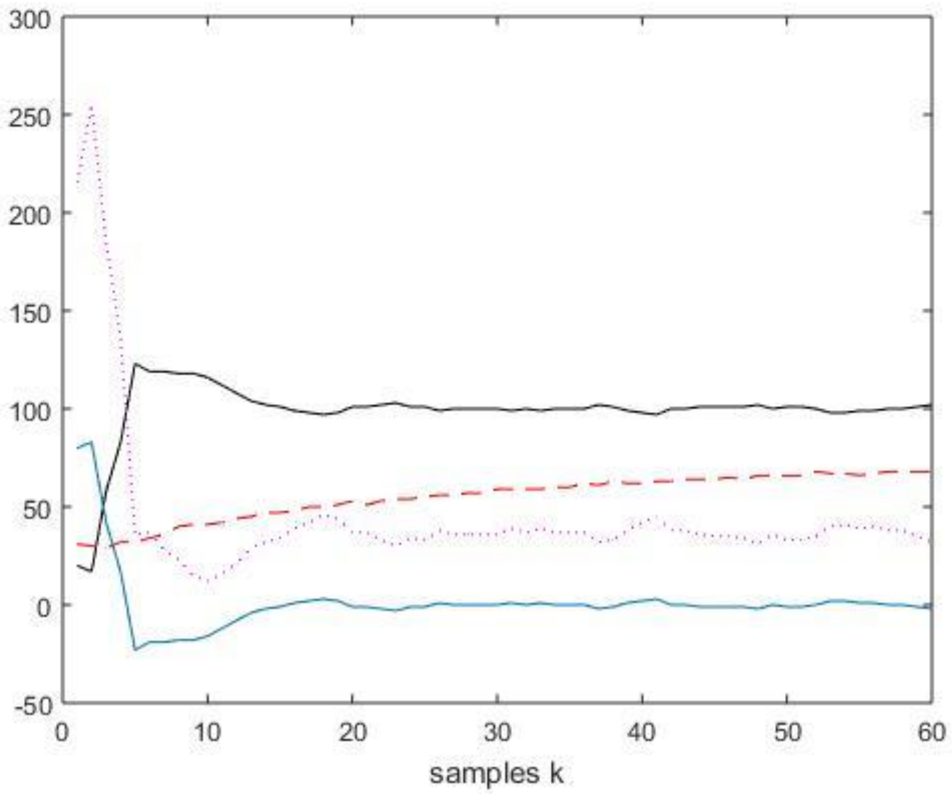
Vi är nöjda med vår p-regulator eftersom att vi inte får en översvängning, dock är vi bekymrade över det kvarstående felet som vi har kvar eftersom att det ligger på 10-15%.

C.5 PI-reglering

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

C.5.1 Experiment med PI-reglering- del I
Använd dig av parametervärdena för PI-regulatorn från tumregel-svängningsmetoden som du bestämde i C.4.2d).

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



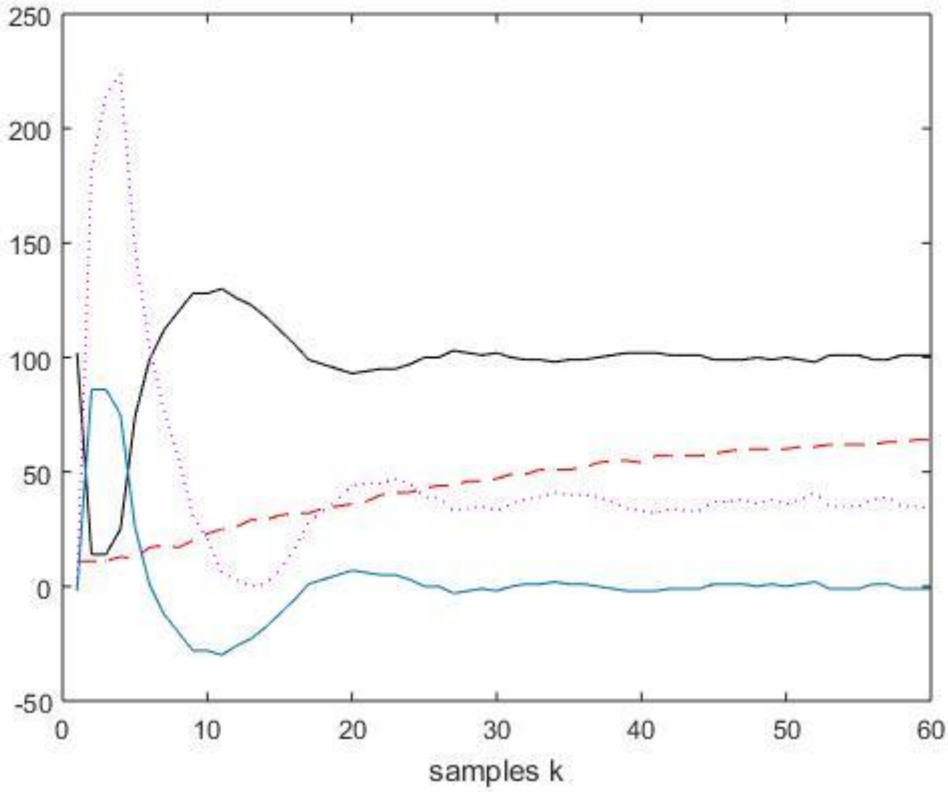
b) Hur stort är det kvarstående felet i procent av börvärdet?
Vi har ett kvarstående fel på 1-2% vilket är väldigt bra.

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 PI-regulatorn?
K = 0.35 * 5 = 1.75
TI = 0.77 * 6 = 4.62
TD = 0.19 * 6 = 1.14.

C.5.3 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?

Vi har inget kvarstående fel, h1 värdet pendlar mellan +1 till -1 %.

C.5.4 Jämförelse

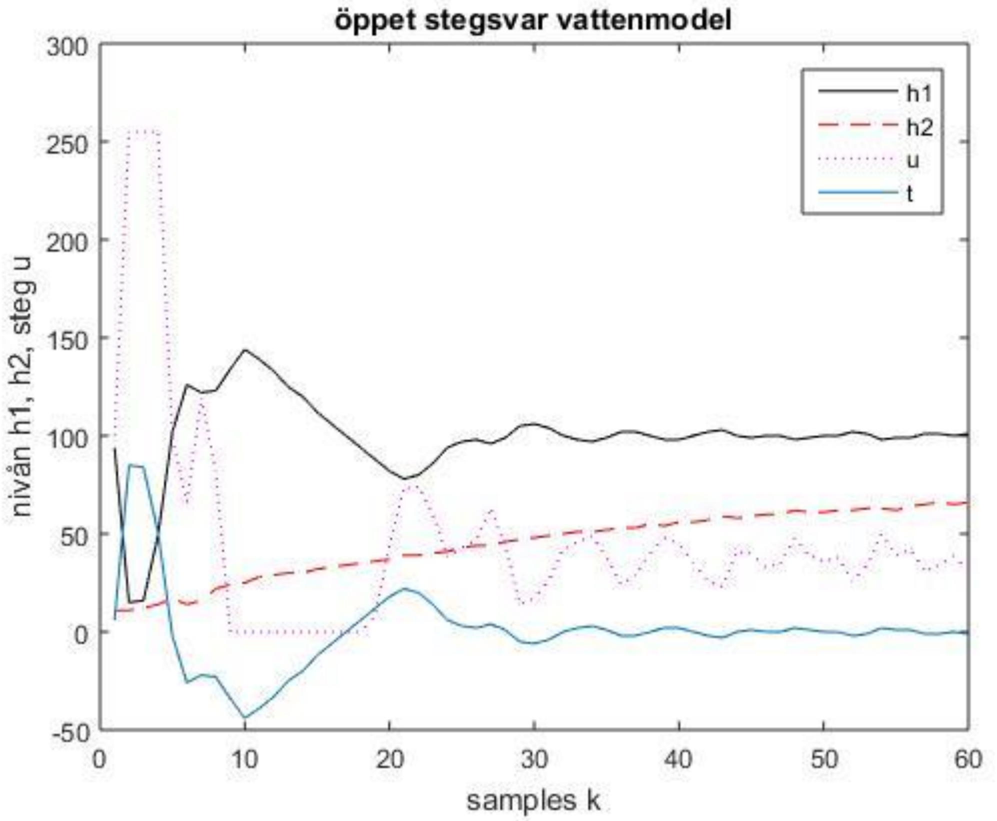
Vilken av de både PI-regulatorerna anser du är den bättre? Förklara varför.

Vi anser at ziegler-nichols metoden fungerade bättre, detta beror på att vi har liknande kurvor men ziegler-nichols metoden gav oss en översvängning som jämnades ut snabbare. Alltså att vår signal blev snyggare, snabbare.

C.6 PID-reglering

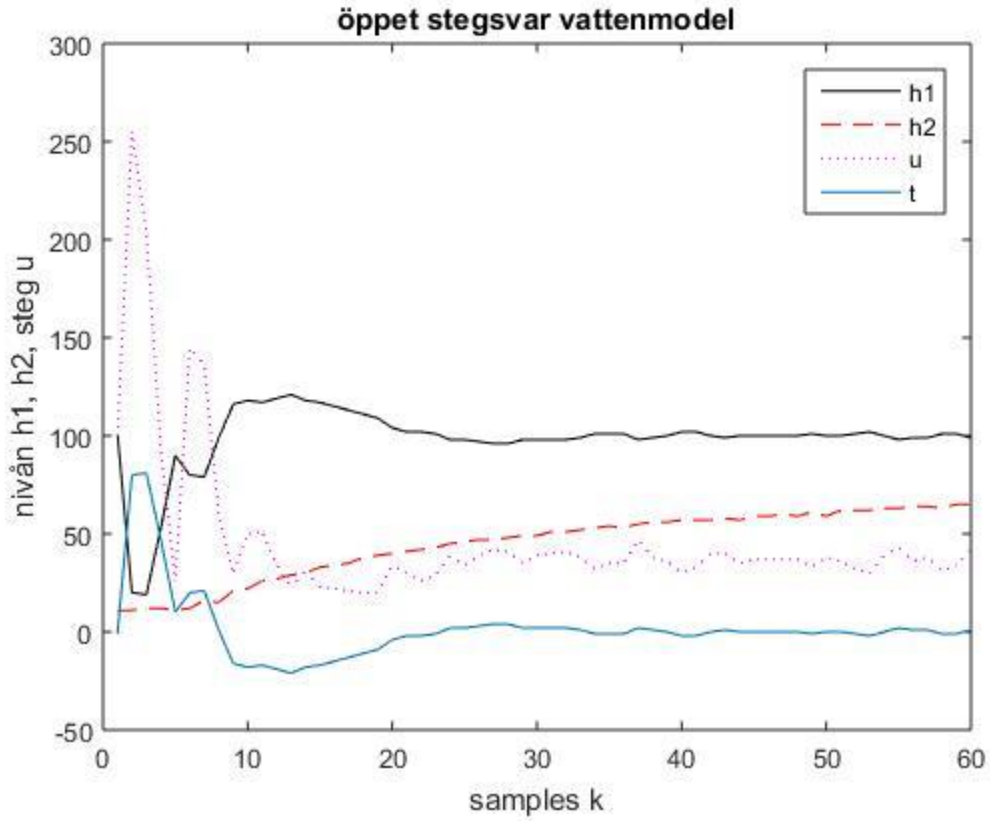
Nu ska vi också testa om vi kan förbättra de dynamiska egenskaperna av vårt regelsystem genom att tillföra PI-regulatorn ett derivat- eller ”D”-andel, så att det blir en PID-regulator (funktion ”vm_PID()”).

C.6.1 Experiment med PID-reglering- del I
Använd dig av parametervärdena för PID-regulatorn från tumregel-svängningsmetoden som du bestämde i C.4.2d).
Utför funktionen ”vm_PID()” och klistra in plottet av resultatet här:



C.6.2 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?
K = 0.35 * 5 = 1.75
TI = 0.77 * 6 = 4.62
TD = 0.19 * 6= 1.14

C.6.3 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:

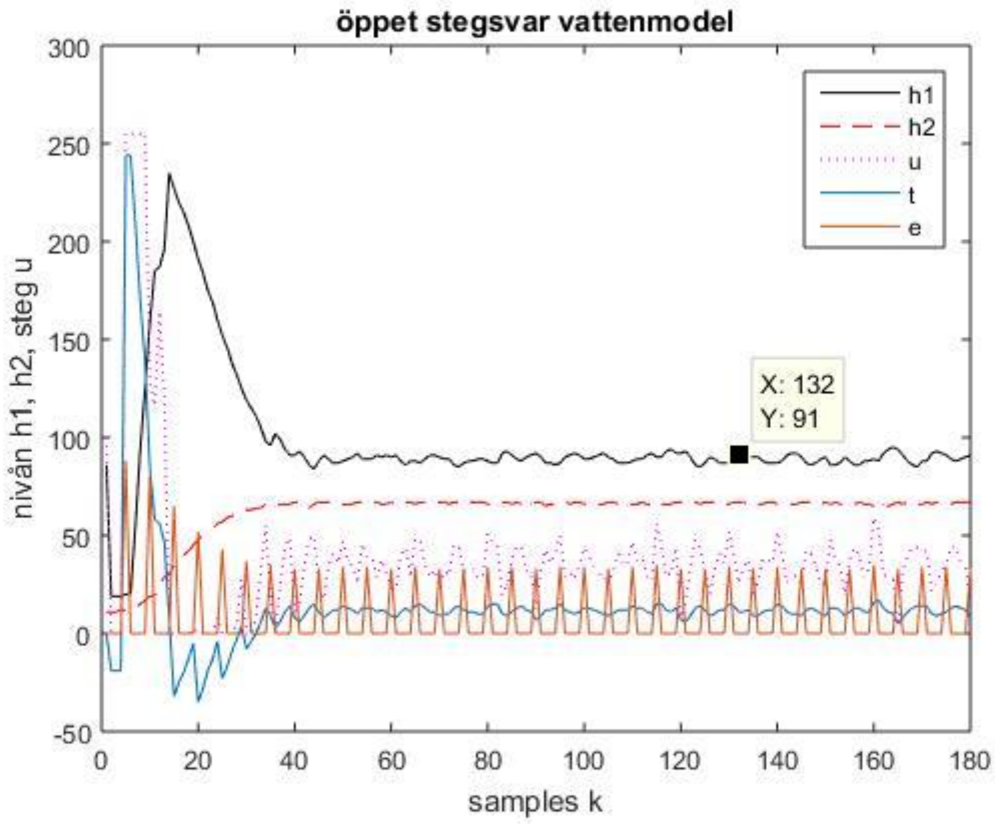


C.6.4 Jämförelse
Vilken av de både PID-regulatorerna anser du är den bättre? Förklara varför.
Vi kan se att vår andra regulator var solklart mycket bättre, eftersom att vi hade en stabilare och snabbare insvängning. Även en mindre översvängning.

C.7 Kaskadreglering

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:

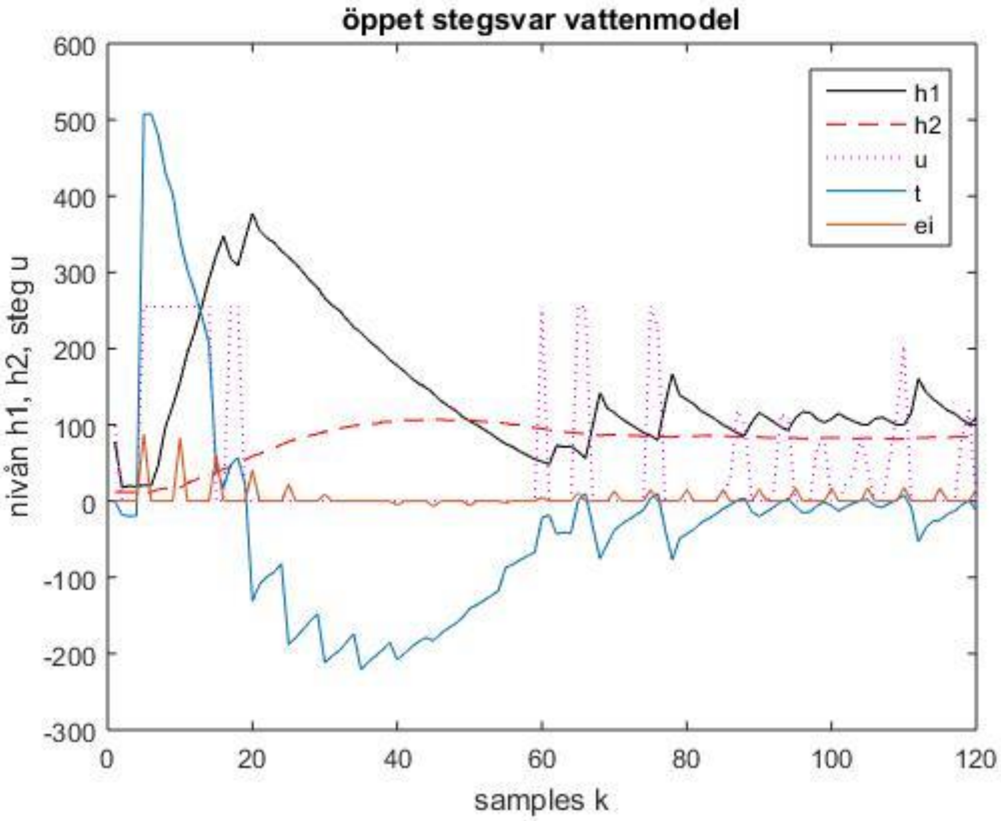


C.7.2 Förbättring av kaskadregleringen

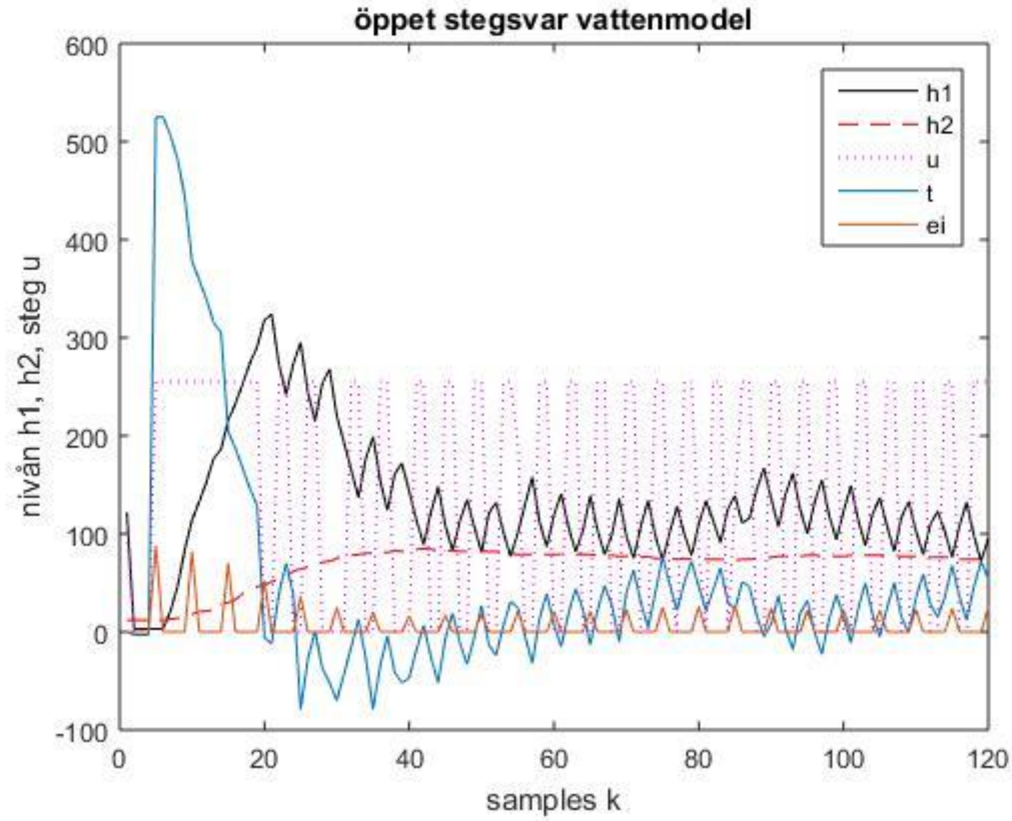
Försök att förbättra inställningarna av din inre- och yttre reglerkrets i din kaskadreglering.

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

Vi dubblade alla våra parametrar(k = 6, TI = 6, TD = 1.5). Då vi utförde denna ändring fick vi en stabilare kurva på h2.



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.6.3 funderade ni omkring hur olika regulatorers resultat skulle kunna jämföras med varandra angående de relevanta egenskaperna hos återkopplade system.
I detta sista avsnitt ska ni analysera och jämföra de olika resultat med varandra.

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?

Den regulator som hade minst översvängning och samtidigt en stabil reglerad storhet var PID-regulatorn. PI-regulatorn hade även en mindre översvängning men en mindre stabil reglerad storhet. Den regleringsmetod med minst stabil reglerad storhet var P-reglering.

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

Vi kom fram till svaret genom att undersöka våra plottar, vi kontrollerade samt med den kvantitativa metoden BIBO. Alltså, att regleringarna skall ligga inom en viss procentuell felmarginal.

C.8.2 Snabbhet

En enkel definition av snabbhet är stigtiden och insvängningstiden.

a) Hur definieras det stigtiden och insvängningstiden?

Stigtid: Den tid det tar för signalen att gå från 10% till 90% av börvärdet.
Insvängningstid: Den tid det tar för ett systems utsignal att nå +/- 5% av börsignalen.

b) Vilken regulator har längsta stigtid, vilken kortaste?

PI-regulatorn hade kortast stigtid för oss. Den som hade längst var P-reglering.

c) Vilken regulatorer har längsta insvängningstid, vilken kortaste?

Den regulator med längst insvängningstid var PID-reglering. Och den med kortast var P-regulatorn.

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

Måttet på snabbhet får man genom att addera alla övervägningar. På detta vis får vi vårt snabbhetsmått.

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?

P-reglering hade störst kvarstående fel som låg på ungefär 10%. Den med minst var PID-reglering efter sin insvängningstid.

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

$$e_0 = \lim_{t\to\infty}(r - y) = 100 - 87 = 13$$

C.8.4 Diskussion

a) Blev resultatet av jämförelsen mellan de olika regulatorerna som du hade förväntat dig?
Nej det blev inte riktigt som vi hade tänkt oss.

b) Om inte, varför inte?
Vi förväntade oss bland annat att PID-regulatorn inte skulle ha en så stor översvängning, och att P-regulatorn inte skulle ha ett så felaktigt värde(mindre kvarstående fel).

D) Reflektion och utvärdering

D.1 Vad tycker du/ni var lärorik med uppgiften? (Minst 5 meningar och minst 75 ord!)
Vi tycker att teoridelen var otroligt givande även om den var lång. Kanske hade kunnat skippa A1 – A2. Vi har lärt oss väldigt mycket om reglering samt z-transformation. Att få se tillämpningarna fysiskt har även varit till stor nytta. Kopplingen var ingenting nytt men det var en skön omväxling att få koppla lite elektronik.

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!)
Nästan allting som vi har utfört i labben har varit nytt. Reglerteknik, z-transformation och överföringsfunktioner har varit nytt och intressant att lära sig. Att testa på alla olika reglertekniker har gett ett perspektiv på hur teknikerna skiljer sig åt. Att behöva använda sig mycket av en bok som faktiskt är bra har även underlättat(till skillnad från tidigare delkurs).

D.3 Vad var det svåraste med uppgiften? (Minst 5 meningar och minst 75 ord!)
Det som var svårast för oss var att lära oss z-transformation. Men då det var utfört har det mesta gått smidigt. Vi har stött på mycket problem med våra DAC kanaler, men efter arduinobyte runt uppgift C.6 har det gått mycket bättre.

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?
Vår ungefärliga tid skulle vi uppskatta till 20 timmar. Vi har lagt ungefär 4-5 timmar på att undersöka kretsfel och kodfel då våra DAC-kanaler inte har fungerat.

D.4 Synpunkter, förslag, kommentarer? (Minst 5 meningar och minst 75 ord!)
Vi föreslår att antalet laborationsändringar kan minskas och att labbändringar sker innan labben är påbörjad.