 ver 2017-02-19

Inbyggda system och signaler

Styr- och reglerteknik

##### *Labbinlämning 1504e*

Utlämning: 20 febr 2017

Deadline inlämning: 14 mars 2017, kl. 16:00

Namn: Yurdaer Dalkic

Namn: Louay Khalil

Gion Koch Svedberg februari 2017

Klassisk reglerteknik och bestämning av tidsdiskreta överföringsfunktioner

Syftet med denna laboration är att praktiskt tillämpa olika klassiska regleralgoritmer på en klassisk vattentankmodell. Genom att få stegsvaret på det fysikaliska systemet kan man använda tumregler för inställningen av regulatorparameter samt genomföra en ”black-box” systemidentifikation.

Vi använder oss av en vattentankmodell 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.

|  |  |
| --- | --- |
| bild vm.JPG | 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 vattentankmodellen som används i denna inlämningsuppgift. | |

I dokumentet ”Överblick över reglerteknikdelen” visas vilka vattentankar (övre eller undre) som ska användas till vilka uppgifter. Där finns också referenser till kursboken till varje uppgift. Det är upp till varje grupp hur de planerar genomförandet av uppgifterna. Antal vattentankmodeller är begränsat och man behöver boka tider för att genomföra de praktiska experimenten. Det kräver en viss förberedning för att hinna färdig i tid. Uppgifterna delas upp i olika delar ”A-D” : teoretiska förberedande inkl. Matlabprogrammering (A), praktiska och experimentella (B), analyserande (C), och självreflekterande (D) delar.

Innehållsförteckning och översikt

[A.1 Vattentankmodell som enkel reglerkrets 6](#_Toc475283071)

[A.1.1 Rita upp den klassiska reglerkretsen för vattentankmodellen som ska regleras med 6](#_Toc475283072)

[A.1.2 Analysera processens (vattentankmodellens) in- och utgångssignalerna. 6](#_Toc475283073)

[A.1.3 Exempel av tillämpningsområden som kan beskrivas med vattentankmodellen 6](#_Toc475283074)

[B.1 Labbutrustningen och allmänna anvisningar beträffande experimentens genomförande 7](#_Toc475283075)

[B.1.1 Förberedning av Arduino Due 7](#_Toc475283076)

[B.1.2 Anslutningen av vattentankmodellen till Arduino Due 8](#_Toc475283077)

[B.1.2 Test av utgångsspänning mellan 0V-ca.2,5V 8](#_Toc475283078)

[A.2 Stegsvar, processtyp och typsiffra 8](#_Toc475283079)

[A.2.1 Stegsvar i öppna regelkretsen 9](#_Toc475283080)

[A.2.2 Processtypen och typsiffror för nivån i första eller andra behållaren 9](#_Toc475283081)

[A.2.3 Programmeringen av stegsvarsexperiment 9](#_Toc475283082)

[B.2 Stegsvarsexperiment 11](#_Toc475283083)

[B.2.1. Stegsvarsexperiment 12](#_Toc475283084)

[C.1 Jämförelse av stegsvaren 12](#_Toc475283085)

[C.1.2 Jämförelse mellan ritning och resultat 12](#_Toc475283086)

[C.2 Filtrering av mätvärden 12](#_Toc475283087)

[C.2.1 Filtrera mätningarna från stegsvaren, spara dem i globala variabler och klistra in grafen (plotts) här: 13](#_Toc475283088)

[A.3 Klassiska reglerprinciper 13](#_Toc475283089)

[A.3.1 Programmering av en tidsdiskret tvålägesreglering 14](#_Toc475283090)

[A.3.2 Programmering av en tidsdiskret flerstegsreglering 14](#_Toc475283091)

[A.3.3 Programmering av en tidsdiskret P-reglering 14](#_Toc475283092)

[A.3.4 Programmering av en tidsdiskret PID-reglering 14](#_Toc475283093)

[A.3.5 Programmering av en tidsdiskret PD-reglering 14](#_Toc475283094)

[B.3 Reglering med tidsdiskreta klassiska reglerprinciper 15](#_Toc475283095)

[B.3.1. Kopiera in grafen från ”tvålägesregleringen” här: 16](#_Toc475283096)

[B.3.2 Kopiera in grafen från ” flerstegsregleringen” här: 16](#_Toc475283097)

[B.3.3 Kopiera in grafen från ”P-regleringen” här: 16](#_Toc475283098)

[B.3.4 Kopiera in grafen från ”PID-regleringen” här: 16](#_Toc475283099)

[B.3.5. Kopiera in grafen från ”PD-regleringen” här: 16](#_Toc475283100)

[C.3 Dimensionering av reglersystem med tumregelmetoder 16](#_Toc475283101)

[C.3.1 Lambda- och Amigo-metoden 16](#_Toc475283102)

[C.3.1.1 Vilka värden får ni från Lambda-metoden för: 16](#_Toc475283103)

[C.3.1.2 Vilka värden får ni från Amigo-metoden för: 16](#_Toc475283104)

[B.4. Ziegler-Nichols svängningsmetod 16](#_Toc475283105)

[B.4.1 Klistra in en graf som visar hur ni fick nivån i undre vattentanken att självsvänga enligt svängningsmetoden: 16](#_Toc475283106)

[B.4.2 Vilka kritiska förstärkning K0 och periodtid T0 får ni? 16](#_Toc475283107)

[B.4.3 Vilka blir PID-regulatorparametrarna enligt Ziegler-Nichols tabell? 17](#_Toc475283108)

[B.4.4 Vilka blir PID-regulatorparametrarna enligt Åström & Hägglunds föreslag? 17](#_Toc475283109)

[B.5 PID-regulator med olika parametrar enligt tumregler 17](#_Toc475283110)

[B.5.1 Kopiera in grafen från PID-regulatorn med Amigo-metoden-parametrarna 17](#_Toc475283111)

[B.5.2 Kopiera in grafen från PID-regulatorn med Ziegler-Nichols-metoden-parametrarna 17](#_Toc475283112)

[C.4 Jämförelse mellan olika reglersystem 17](#_Toc475283113)

[C.4.1 Jämförelse av stystemens egenskaper. Fyll i tabellen: 17](#_Toc475283114)

[A.4 Bestämning av tidsdiskreta överföringsfunktionen 18](#_Toc475283115)

[A.4.1 Beskriv och visa steg för steg hur ni bestämmer överföringsfunktionen för den övre 18](#_Toc475283116)

[A.4.2 Beskriv och visa steg för steg hur ni bestämmer överföringsfunktionen för den övre och 18](#_Toc475283117)

[A.5 Simulation av stegsvaren med Matlab Control-Toolbox 18](#_Toc475283118)

[A.5.1 Simulera stegsvaren av övre vattentank med Matlab och spara resultatet i en variabel. 18](#_Toc475283119)

[A.5.2 Simulera stegsvaren av övre och nedre vattentank tillsammans med Matlab och spara 18](#_Toc475283120)

[C.5 Jämförelse mellan verkliga och simulerade stegsvaren 18](#_Toc475283121)

[C.5.1 Jämförelse mellan verkliga och simulerade stegsvaren för övre vattentank. Klistra in 18](#_Toc475283122)

[C.5.2 Jämförelse mellan verkliga och simulerade stegsvaren för övre och nedre vattentank 18](#_Toc475283123)

[D. Reflektion och utvärdering över det egna lärandet 19](#_Toc475283124)

[Bilaga 20](#_Toc475283125)

[Översikt över Matlab instruktioner 20](#_Toc475283126)

[Exempel av nyckelfrågor i samband med reflektioner och utvärdering av det eget lärande 20](#_Toc475283127)

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 ”Modern Reglerteknik” av Bertil Thomas, Matlabguiden, Matlabs ”help” och dokumentation samt material som finns upplagda på its learning.

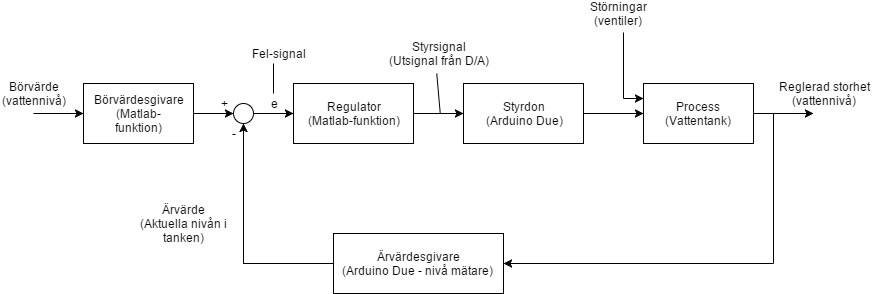
**Krav för godkänd**

* Fullständigt ifyllt dokument (inkl namn på titelsida) med korrekta svar till alla frågor, uppladdad till its learning som word eller pdf-fil, (okomprimerad).
* Binär fil ”labb1504e.mat” med alla variabler från reglerexperimenten, uppladdad på its learning.

# A.1 Vattentankmodell som enkel reglerkrets

Börja med att läsa igenom ”Kompendium om reglerteknikutrustningen”, åtminstone kap. 1 & 2. Du behöver få en överblick över processen som ska regleras. Gör därför följande uppgifter:

A.1.1 Rita upp den klassiska reglerkretsen för vattentankmodellen som ska regleras med Matlab och Arduino. Ange de olika begrepp till de olika signalerna.



### A.1.2 Analysera processens (vattentankmodellens) in- och utgångssignalerna.

a) I vilket spänningsområde befinner sig signalerna?

0-3,3 v

Ingångsspänningarna (styrsignal) som fås från D/A omvandlare genom anpassning krets ligger mellan 0-3 V.

Utgångsignalerna (nivåmätare) är signaler som fås från vattentank som ligger mellan 0 till 10 V.

b) Hur behöver dem anpassas för att kunna hanteras av Arduinon?

Vi kan anpassa utsignalerna genom att använda en spänningsdelare (seriekoppling av 27kOhm och 10kOhm) får man ner signalnivån till 0-2.7V över 10kOhm-resistorn. Vi kan använda CA3240 för att skydda arduinos analoga ingångarna som vi gjorde i tidigare labbarna.

Vi kan anpassa insignalen (styrsignal) genom att använda en förstärkningskrets med diod och op-amp (TS921). Utsignalen blir mellan 0-ca 2.5V.

c) Vad är sambandet mellan den fysikaliska enheten och Arduinos motsvarande digitala representation?

1024/20=51,2 bit per centimeter

### A.1.3 Exempel av tillämpningsområden som kan beskrivas med vattentankmodellen

Vattentankmodellen är en mycket användbar modell i helt olika sammanhang. Testa detta genom att ange tre olika exempel, en från området ”ekonomi”, en från ”samhälle” och en från ”industri” som kan beskrivas med vattentankmodellen. Beskriv vad i dina exempel som motsvarar motorstyrningen, pumpen, in- och utflöden, störningar, nivån i första och andra behållaren samt börvärden.

1. ”Ekonomi”-exempel:

En person jobbar kontinuerligt för att få in en lön kontinuerligt också. Lönen kommer in som pengar på ett bankkonto och regleras med hjälp av personens utgifter samt nästa månads löninsättning.

1. ”Samhälle”-exempel:

Population brukar oftast förknippats med den pågående politiken i ett samhälle. Det regleras exempelvis med hjälp av bra förhållanden för ny-föräldrar i form av föräldraledighet, barnbidrag etc.

1. ”Industri”-exempel:

En kund vill ha en mängd varor från en fabrik. Denna mängd reglerar antalet tillverkade, paketerade och pallsatta varor med undandrag från störningar såsom felaktigheter i tillverkningen eller paketeringen.

Översikt och jämförelse

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ekonomi-exempel: | Samhälle-exempel: | Industri-exempel: |
| motorstyrning | Arbetsmarknad | Samhällspolitisk agenda | Kund-order |
| pumpen | Arbete | Människor som skaffar barn | Tillverkning |
| inflöde till första behållare | Pengar som man tjänar | Antal barn som föds | Färdiga varor |
| utflöde ur första behållaren | Vanliga utgifter som mat, hyra och bränsle | Barn som blir vuxna | Paketerade varor |
| inflöde till andra behållaren | Maten i kylskåpet, bränsle i bilen | Människor som blir äldre | Paketerade varor |
| utflöde ur andra behållaren | Maten man äter, bränslet som bilen förbrukar, kläder som man slänger… | Människor som dör | Färdiga pallar |
| störningar | Oförväntade utgifter som renovering eller reparation av bilen, etc. | Oförväntade dödsfall | Felaktiga varor |
| nivån i första behållaren | Pengarna som finns i bankkonto | Antal barn | Antal varor som ska paketeras |
| nivån i andra behållaren | Värdet av varorna som man har köpt | Antal vuxna | Antal paket som ska läggas på pall |
| börvärde | Pengarna som man vill ha på bankkonto. | Ett genomsnitts värde på antal ungdomar enligt den politiska agendan. | Den mängd av varor som kunden har beställt |

# B.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 B.1 nedan ger en överblick över de olika delarna som man behöver ha koll på innan man kan påbörja experimenten.

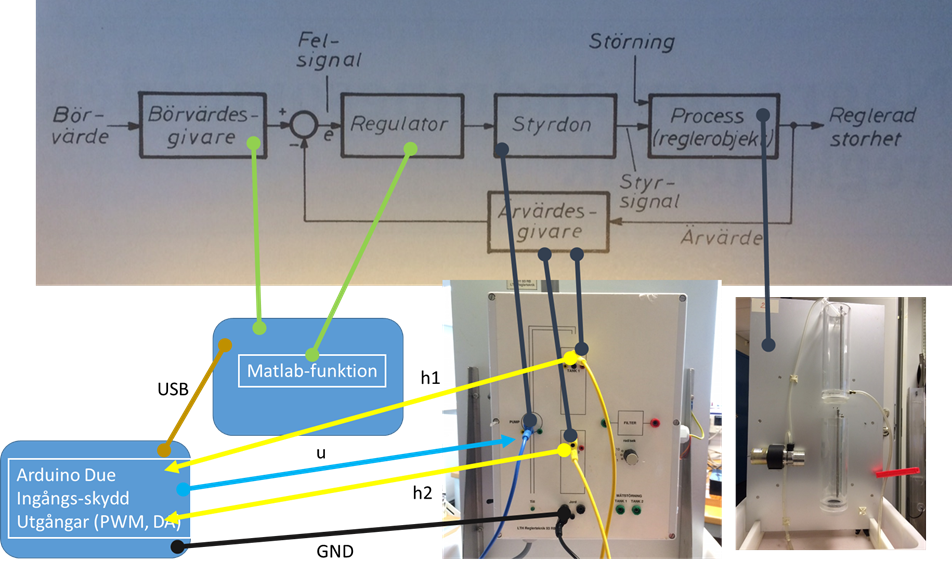


Bild B.1: Översikt över de olika delarna som ingår i utrustningen.

OBS: Det är Matlab 2015b som ska användas!

## B.1.1 Förberedning av Arduino Due

Börja med att anpassa arduino due till reglertekniklabben. 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.

## B.1.2 Anslutningen av vattentankmodellen till Arduino Due

Anslut nu kablarna från vattentankmodellen till Arduino Due:

1. Jordning från vattentankmodellen till Arduinos GND
2. Nivåmätning av övre tanken (h1) till Arduino, t.ex. ’A0’
3. Nivåmätning av nedre tanken (h2) till Arduino, t.ex. ’A1’
4. 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!

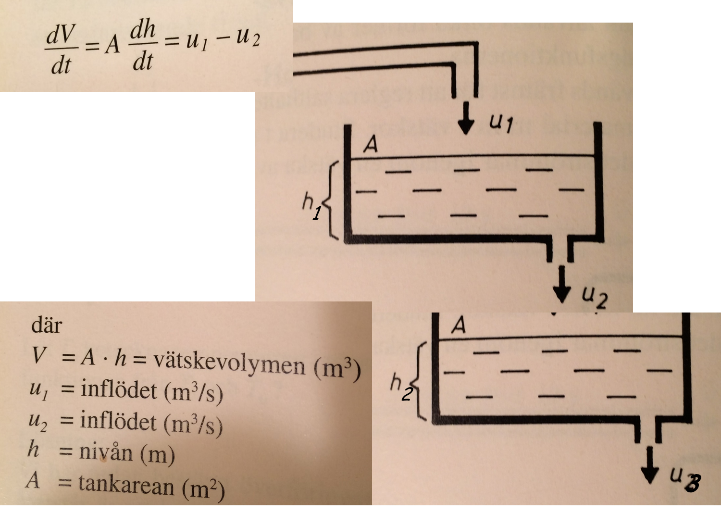
## B.1.2 Test av utgångsspänning mellan 0V-ca.2,5V

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 | 0V |
| 1V | 0.45V |
| 1,5V | 0.9V |
| 2V | 1.49V |
| 3V | 2.39V |

# A.2 Stegsvar, processtyp och typsiffra

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.2.1 kopplade vattentankar

I följande uppgift fokuserar vi oss på vattenmodellen enligt figur A.2.1 ovan.

|  |
| --- |
| A.2.1 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:)  C:\Users\yurdaer\Dropbox\IMG_20170228_150511.jpg |

|  |
| --- |
| A.2.2 Processtypen och typsiffror 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?  Andra behållaren är beroende av att första behållaren kommer upp till sin nivå av h1, för att trycket då har ökat så att utflödet också ökar. Dock kommer första behållaren alltid att fyllas i snabbare än andra behållaren.  b) Vilken processtyp och typsiffra har första behållaren? Förklara varför.  Det är en process med en tidskonstant därför att stegsvarets lutning är störst i början och sedan avtar den tills slutnivån uppnås. Det är ett system av typen noll därför att utsignalen (vattennivån) går mot ett ändligt värde vid en stegformad insignalsförändring.  c) Vilken processtyp och typsiffra har andra behållaren? Förklara varför.  Det är process med dödtid och har två tidskonstanter därför är det så att utflödet u2 tar längre tid för att komma igång med en ordentlig mängd som kan få upp nivån på höjden i andra behållaren. Det är ett system av typen noll därför att utsignalen (vattennivån) går mot ett ändligt värde vid en stegformad insignalsförändring. |

## A.2.3 Programmeringen av stegsvarsexperiment

I uppgiften A.2.1 ritade ni upp förloppet av de förmodade stegsvaren för h1 och h2 av den öppna reglerkretsen. Nu blir det dags att köra stegsvarexperimentet på vattenmodellen och att jämföra resultatet från experimentet med era ritningar.

För detta behöver ni först programmera en Matlabfunktion. Ni kan utgå från en gemensam stomme som ni sedan också kan använda för att programmera de olika regulatorerna. Stommen av ett matlabprogram visas i figuren nedan och finns förklarad i *Kompendium* *om reglerteknikutrustningen*:

|  |
| --- |
| function [e, u, y, t]=vm\_stomme(a, N, Ts, v)  % Stomme för regulator-block. Kan användas för att lägga till och anpassa till olika  % klassiska, tidsdiskreta regulatorer  % Argument (anpassas efter ändamål)  % a: arduino-objektet som Matlab använder för att kommunicera med Arduino  % N: antal samplingar  % Ts: samplingstiden mellan samplingar  % v: börvärde i digitala enheter (0..1023)  % Resultat (anpassas efter ändamål)  % e: vektor med N mätningar av felsignalen  % u: vektor med N mätningar av styrsignalen  % y: vektor med N mätningar av process-svaren == ärvärden  % t: tidsdiskret tidsvektor 1:N  % Initialisering av variablerna ---------------------------------------------------------------  e=zeros(1, N);  u=zeros(1, N);  y=zeros(1, N);  t=zeros(1,N);  start=0; elapsed=0; ok=0; % används för att upptäcka för korta samplingstider  k=0; % samplingsindex  % Konfigurering av in- och utgångar -----------------------------------------------------  % Arduino ingångar  % analoga ingångar ’A0’ och ’A1’ behöver inte konfigureras. 0..1023  % ’A0’: y  % analoga utgångar behöver inte heller konfigureras. DAC1-> PWM, DAC0 -> DAC,  % 0..255  % ’DAC0’: u  % cyklisk exekvering av samplingar  for k=1:N % slinga kommer att köras N-gångar, varje gång tar exakt Ts-sekunder    start = cputime; %startar en timer för att kunna mäta tiden för en loop  if ok <0 % testar om samplingen är för kort  k  disp('samplingstiden är för lite! Ök värdet för Ts');  return  end    % uppdatera tidsvektorn  t(k)=k;    % läs ingångsvärde sensorvärden  y(k)= analogRead(a, ’A0’); % mät ärvärdet  % beräkna felvärdet som skillnad mellan ärvärdet och börvärdet  e(k)=v-y(k);    % Regulatorblock  % beräkna styrvärdet, t.ex p-regulator med förstärkning Kp=1  u(k)=e(k); % p-regulator, Kp=1  % begränsa styrvärdet till lämpliga värden, vattenmodellen t.ex. u >=0 och u <255, samt  % heltal  u(k)=min(max(0, round(u(k)), 255);  % skriva ut styrvärdet  analogWrite(a,u(k), ’DAC0’); %DAC-utgång  %online-plot  plot(t,y,'k-',t,u,'m:',t,e,'b');    elapsed=cputime-start; % räknar åtgången tid i sekunder  ok=(Ts-elapsed); % sparar tidsmarginalen i ok    pause(ok); %pausar resterande samplingstid    end % slut av samplingarna ----------------------------------------------------------------------  % plotta en fin slutbild,  plot(t,y,'k-',t,u,'m:',t,e,'b');  xlabel('samples k')  ylabel('y, u ,e')  title('xxx-regulator')  legend('y ', 'u ', 'e ')  % ------------------------------------------------------------------------------------------- |

Fig A.3.1: Stomme av matlabprogrammet som kan användas för programmeringen.

Anpassa stommen med de rader som krävs för att kunna ta upp stegsvaret av övre och undre vattentank. Sätt pumpen från noll till maxhastighet i tidpunkten ”t=0”!

Tänk på att resultatet, dvs vektorerna med mätvärden, ska senare analyseras, dvs de ska kunna sparas som globala matlabvariabler.

A.2.3.1 Klistra in Matlabprogrammen för stegsvaret här:

function [e, u, y1,y2, t]=vm\_stomme(a, N, Ts, v)

e=zeros(1, N);

u=zeros(1, N);

y1=zeros(1, N);

y2=zeros(1, N);

t=zeros(1,N);

start=0;

elapsed=0;

ok=0; % används för att upptäcka för korta samplingstider

k=0; % samplingsindex

analogRead(a, 'A0');

analogRead(a, 'A1');

% cyklisk exekvering av samplingar

for k=1:N % slinga kommer att köras N-gångar, varje gång tar exakt Ts-sekunder

start = cputime; %startar timer för att kunna mäta tiden för loop

if ok <0 % testar om samplingen är för kort

k;

disp('samplingstiden är för lite! Ök värdet för Ts');

return

end

% uppdatera tidsvektorn

t(k)=k;

% läs ingångsvärde sensorvärden

y1(k)= analogRead(a, 'A0'); % mät ärvärdet

y2(k)= analogRead(a, 'A1'); % mät ärvärdet

% beräkna felvärdet som skillnad mellan ärvärdet och börvärdet

e(k)=v-y1(k);

u(k) = 50;

if u(k)> 255

u(k)=255;

end

% skriva ut styrvärdet

a.analogWrite(u(k)); %DAC-utgång

%online-plot

plot(t,y1,'k-',t,u,'m',t,e,'b:');

elapsed=cputime-start; % räknar åtgången tid i sekunder

ok=(Ts-elapsed); % sparar tidsmarginalen i ok

pause(ok); %pausar resterande samplingstid

end % slut av samplingarna

% plotta en fin slutbild,

plot(t,y1,'k-',t,u,'m');

xlabel('samples k')

ylabel('y, u')

title('xxx-regulator')

legend('y ', 'u ')

# 

# B.2 Stegsvarsexperiment

Genomför stegsvarsexperiment med den övre och den nedre vattentanken. (För den nedre betyder att den fylls via utloppet av den övre men att man bara mäter nivån i den nedre tanken.)

|  |
| --- |
| B.2.1. Stegsvarsexperiment Genomför experimentet och klistra in grafen med stegsvaret av den öppna reglerkretsen här:   1. För övre vattentank   C:\Users\yurdaer\AppData\Local\Microsoft\Windows\INetCache\Content.Word\stegsvar_h1.jpgdT=0,8s   1. För nedre vattentank   C:\Users\yurdaer\AppData\Local\Microsoft\Windows\INetCache\Content.Word\stegsvar_h2.jpgdT=0,8s   1. Mätningen av stegsvaren har sparats i variablerna \_**y1**\_\_(övre tank) och \_**y2**\_\_\_\_ (nedre tank) |

# C.1 Jämförelse av stegsvaren

Analysera era förväntningar av stegsvaren med era experimentella mätningar.

|  |
| --- |
| C.1.2 Jämförelse mellan ritning och resultat Jämför nu resultatet från experimentet med din ritning i A.2.1:  a) Vad stämmer bra överens?  Övre tanken stämmer bra med ritningen eftersom vattennivån snabbt ökar för att sedan långsammare öka. Givetvis är u konstant hela tiden eftersom vi har satt den så det verkligen blir stegsvar.  b) Vad är mest annorlunda och varför? Hur förklarar du skillnaden om det finns några?  Nedre tanken verkar också att stämma, dock tycker vi att vattennivån kanske borde öka någorlunda långsammare i början. Lite död tid syns ändå, vilket vi hade förväntat oss. |

Resultatet från stegsvaret för h1 och h2 kommer ni att använda för att identifiera systemet i form av en differensekvation samt för att få fram parameterinställningar för en rad regulatorer med hjälp av tumreglerna. Men först behöver dem filtreras något.

# C.2 Filtrering av mätvärden

Mätvärden för nivån i övre och i nedre vattentanken från stegsvarsexperimentet 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:

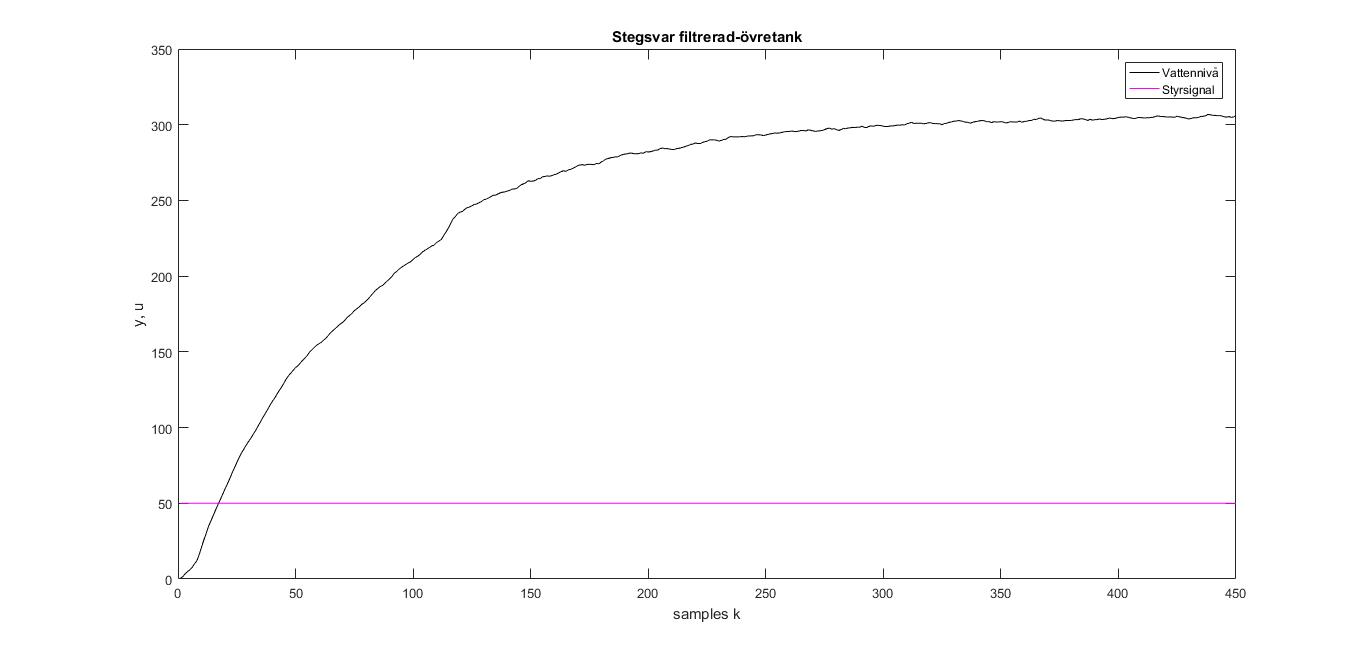
Alternativet är att du bygger en enkel ”moving average filter”, se också utdraget ur Matlabs dokumentation nedan

|  |
| --- |
| **Examples**  **[Moving-Average Filter of Vector Data](javascript:void(0);" \o "Expand/Collapse)**  Find the moving-average of a vector without using a for loop.  A moving-average filter is represented by the following difference equation,  $$ y(n)=\frac{1}{windowSize}\left(x(n)+x(n-1)+...+x(n-(windowSize-1))\right). $$  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); |

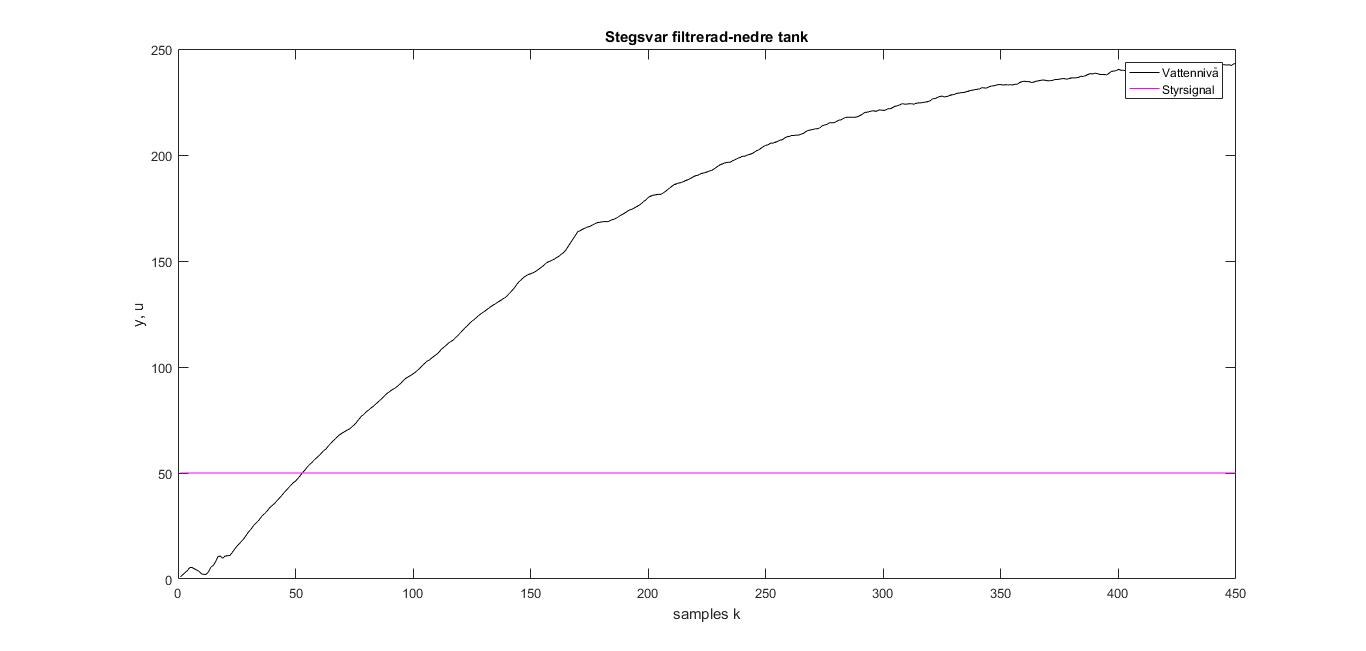
Testa gärna med olika värden för windowSize. Istället för ”x” ska ni ange er variabel med mätvärden från nivåmätningen. Resultatet y blir då det filtrerade signal.

### C.2.1 Filtrera mätningarna från stegsvaren, spara dem i globala variabler och klistra in grafen (plotts) här:

1. Filtrerad stegsvarsmätning från övre vattentank



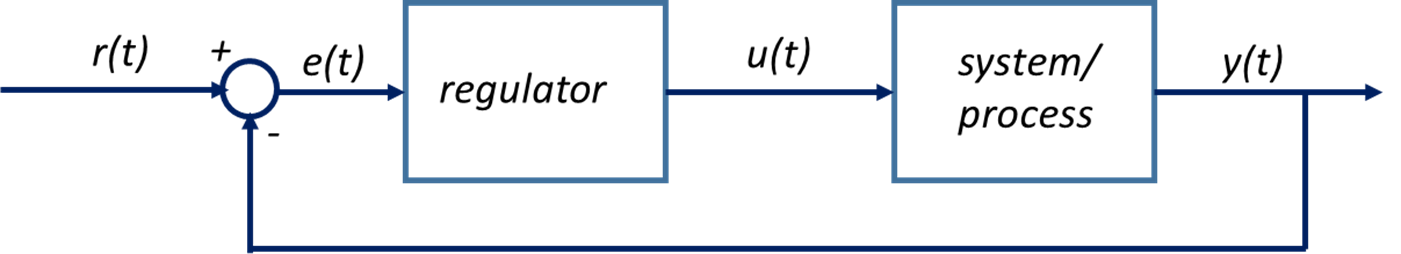
1. Filtrerad stegsvarsmätning från nedre vattentank



1. Val av ”windowSize”: \_\_**5**
2. Global variabel för filtrerad stegsvar, övre vattentank: **yf1**
3. Global variabel för filtrerad stegsvar, nedre vattentank: **yf2**

# A.3 Klassiska reglerprinciper

Nu ska ni 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.



*Fig A.3: blockdiagram av en enkel regelkrets*

Först måste de olika regulatorerna programmeras. Ni kan utgå från samma stomme som ni redan har använt för programmeringen av stegsvaren.

## A.3.1 Programmering av en tidsdiskret tvålägesreglering

Anpassa programstommen till en tidsdiskret tvålägesreglering och spara programmet som .m-fil.

A.3.1.1 Kopiera in Matlabkoden för regulatorblocken som implementerar en ”tvålägesreglering”:

% Regulatorblock

% beräkna styrvärdet, t.ex tvåläge-regulator med förstärkning Kp=1

if e(k) <= 0

u(k) = 0;

else

u(k) = 200;

end

## A.3.2 Programmering av en tidsdiskret flerstegsreglering

Anpassa programstommen till en tidsdiskret flerstegsreglering och spara programmet som .m-fil.

A.3.2.1 Kopiera in Matlabkoden för regulatorblocken som implementerar en ” flerstegsreglering”:

% Regulatorblock

% beräkna styrvärdet, t.ex flersteg-regulator med förstärkning Kp=1

% Flersteg-regulator, Kp=1

if e(k) > 100

u(k) = 255;

else if e(k) <= 0

u(k) = 0;

else if e(k) < 100 && e(k)>50

u(k) = 150;

else if e(k) < 50

u(k) = 80;

end

end

end

end

## A.3.3 Programmering av en tidsdiskret P-reglering

Anpassa programstommen till en tidsdiskret P-reglering och spara programmet som .m-fil.

A.3.3.1 Kopiera in Matlabkoden för regulatorblocken som implementerar en ”P-reglering”:

% beräkna felvärdet som skillnad mellan ärvärdet och börvärdet

e(k)=v-y1(k);

% Regulatorblock

% beräkna styrvärdet, t.ex p-regulator med förstärkning Kp=1

u(k)=e(k); % p-regulator, Kp=1

if u(k)> 255

u(k)=255;

end

if u(k) < 0

u(k) = 0;

end

## A.3.4 Programmering av en tidsdiskret PID-reglering

Anpassa programstommen till en tidsdiskret PID-reglering och spara programmet som .m-fil.

A.3.4.1 Kopiera in Matlabkoden för regulatorblocken som implementerar en ”PID-reglering”:

% Regulatorblock

% beräkna styrvärdet, t.ex pid-regulator med förstärkning Kp=1

if(k==1)

w(k)=e(k);

u(k)= 1\*(e(k)+(1/Ts)\*e(k)+(Ts/1)\*w(k));

else

w(k)=w(k-1)+e(k);

u(k)= 1\*(e(k)+(1/Ts)\*(e(k)-e(k-1))+(Ts/100)\*w(k)); % PID-regulator, Kp=1, Td=1, Ti=100, Ts=0.8s

end

% begränsa styrvärdet till lämpliga värden, vattenmodellen t.ex. u >=0 och u <255, samt

% heltal

if u(k)> 255

u(k)=255;

end

if u(k) < 0

u(k) = 0;

end

% skriva ut styrvärdet

analogWrite(a, u(k), 'DAC0'); %DAC-utgång

## A.3.5 Programmering av en tidsdiskret PD-reglering

Anpassa programstommen till en tidsdiskret PD-reglering och spara programmet som .m-fil.

A.3.5.1 Kopiera in Matlabkoden för regulatorblocken som implementerar en ”PD-reglering”:

% Regulatorblock

% beräkna styrvärdet, t.ex pd-regulator med förstärkning Kp=1

if(k==1)

u(k)= 1\*(e(k)+(1/Ts)\*e(k));

else

u(k)= 1\*(e(k)+(1/Ts)\*(e(k)-e(k-1))); % PID-regulator, Kp=1, Td=1, Ti=100, Ts=0.8s

end

% begränsa styrvärdet till lämpliga värden, vattenmodellen t.ex. u >=0 och u <255, samt

% heltal

if u(k)> 255

u(k)=255;

end

if u(k) < 0

u(k) = 0;

end

% skriva ut styrvärdet

analogWrite(a, u(k), 'DAC0'); %DAC-utgång

# B.3 Reglering med tidsdiskreta klassiska reglerprinciper

Dags att testa era olika regulatorer på riktigt vid vattentankmodellen. Följ informationen från tabellen i dokumentet ”Överblick över reglerteknikdelen” angående vilka regulatorer som ska testas med övre och vilka med undre vattentanken.

Vänligen läs noggrant igenom följande anvisningar innan ni fortsätter med experimenten:

1. Resultaten (dvs mätvärden av vattennivåerna) 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”.

1. 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

1. Spara era resultat på detta viset i samma fil efter varje experiment, så att ingenting tappas bort.
2. 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

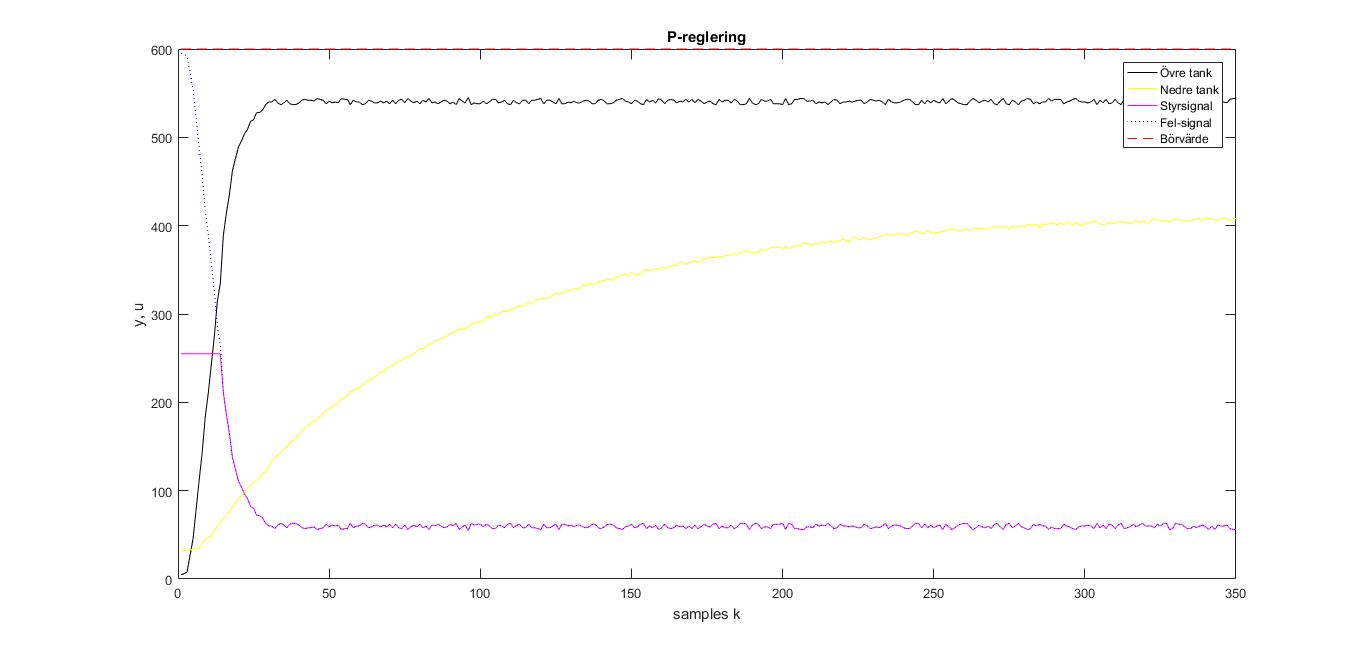
(Och sedan sparar ni enligt pkt 2 och 3)

1. Angående de olika experimenten med olika regulatorer:
   * Testa er fram angående olika börvärden. Om börvärdet är för högt kan det ta längre tid för att uppnå värdet. Om det är för lågt 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!! (Minst 3 minuter kan vara lagom).
2. 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

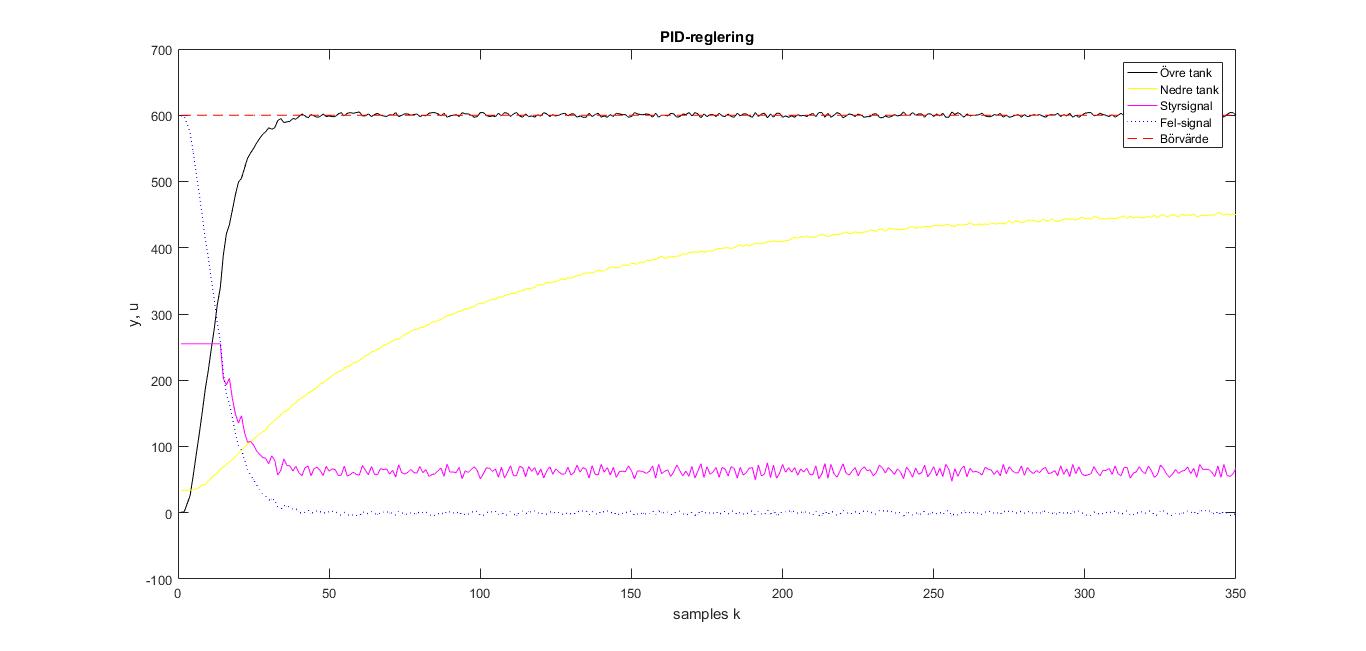
### C:\Users\yurdaer\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2-lages.jpgB.3.1. Kopiera in grafen från ”tvålägesregleringen” här:

### C:\Users\yurdaer\AppData\Local\Microsoft\Windows\INetCache\Content.Word\flerlages.jpgB.3.2 Kopiera in grafen från ” flerstegsregleringen” här:

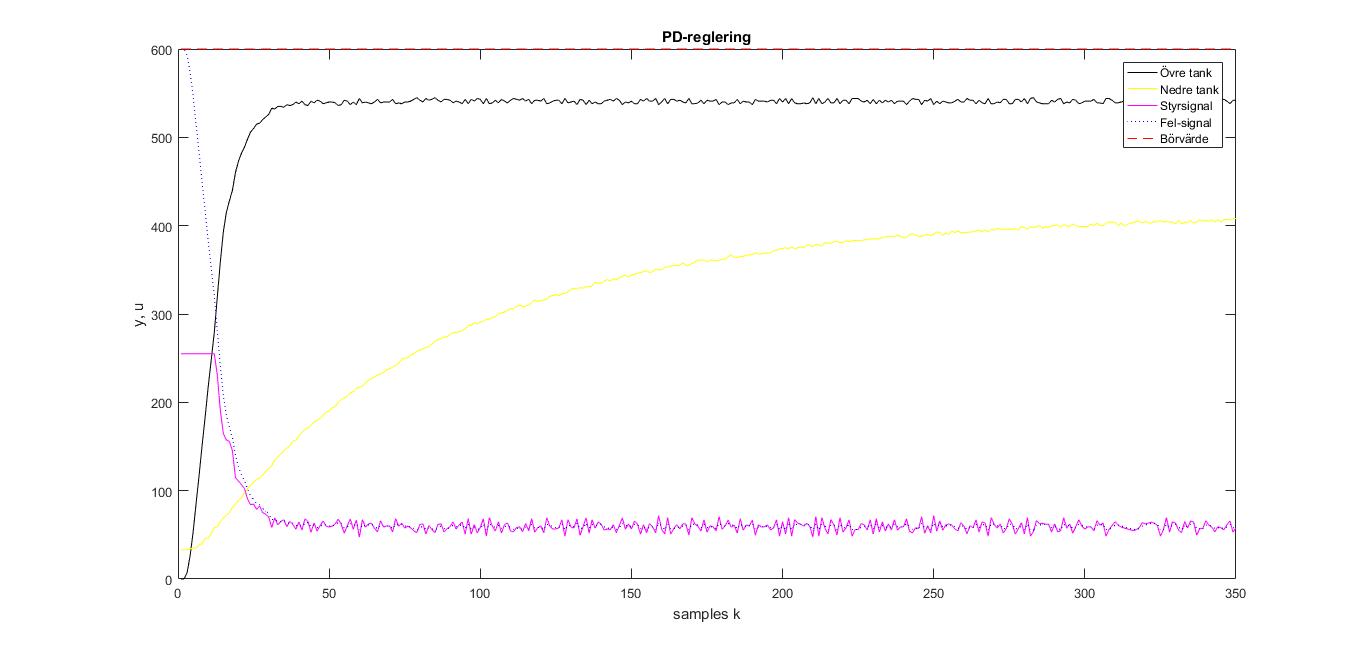
### B.3.3 Kopiera in grafen från ”P-regleringen” här:



### B.3.4 Kopiera in grafen från ”PID-regleringen” här:



### B.3.5. Kopiera in grafen från ”PD-regleringen” här:



# C.3 Dimensionering av reglersystem med tumregelmetoder

Att välja reglerparametrarna, t.ex. för en PID-regulator kan vara tidskrävande och besvärlig. Istället för att använda sig av ”trial and error” finns det tumregelmetoder. Ni ska få testa Ziegler-Nichols svängningsmetoden samt Amigo-metoden (som i sin tur baserar på Lambdametoden).

## C.3.1 Lambda- och Amigo-metoden

Utgå från stegsvaret för undre vattentanken, dvs övre- och undre vattentank tillsammans. Analysera stegsvaret enligt beskrivningen till Lambda-metoden. Bestäm Ks, L och T. Använd sedan Amigo-metoden för att räkna ut K, TI och TD för PID-regulatorn.

### C.3.1.1 Vilka värden får ni från Lambda-metoden för:

dT = 0,8

L + T = 164 sampels

a) Ks = 4,864

b) L = 35 sampels

c) T = 164 – 35 = 129 sampels

### C.3.1.2 Vilka värden får ni från Amigo-metoden för:

a) K= 0.3821

b) TI = 85.6367

c) TD = 16.1828

# B.4. Ziegler-Nichols svängningsmetod

Tillämpa nu också svängningsmetoden för att få fram alternativa PID-regulatorparametrar.

### C:\Users\yurdaer\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sengning.jpgB.4.1 Klistra in en graf som visar hur ni fick nivån i undre vattentanken att självsvänga enligt svängningsmetoden:

### B.4.2 Vilka kritiska förstärkning K0 och periodtid T0 får ni?

K0 =20

T0 = 44 samples

### B.4.3 Vilka blir PID-regulatorparametrarna enligt Ziegler-Nichols tabell?

K= 20 \* 0,6 = 12

TI= 44 \* 0,5 = 22 samples

TD= 44 \* 0,125 = 5,5 samples

### B.4.4 Vilka blir PID-regulatorparametrarna enligt Åström & Hägglunds föreslag?

K= 20 \* 0,35 = 7

TI= 44 \* 0,77 = 33,9 samples

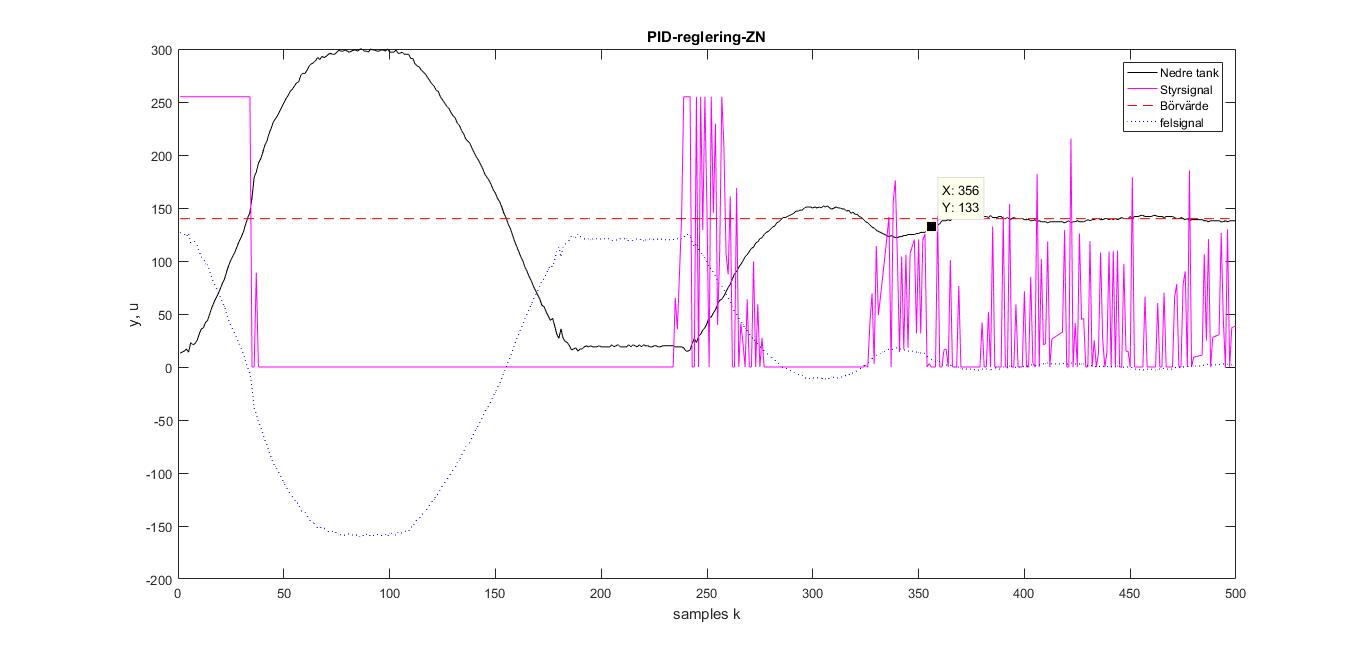
TD= 44 \* 0,19 = 8,4 samples

# B.5 PID-regulator med olika parametrar enligt tumregler

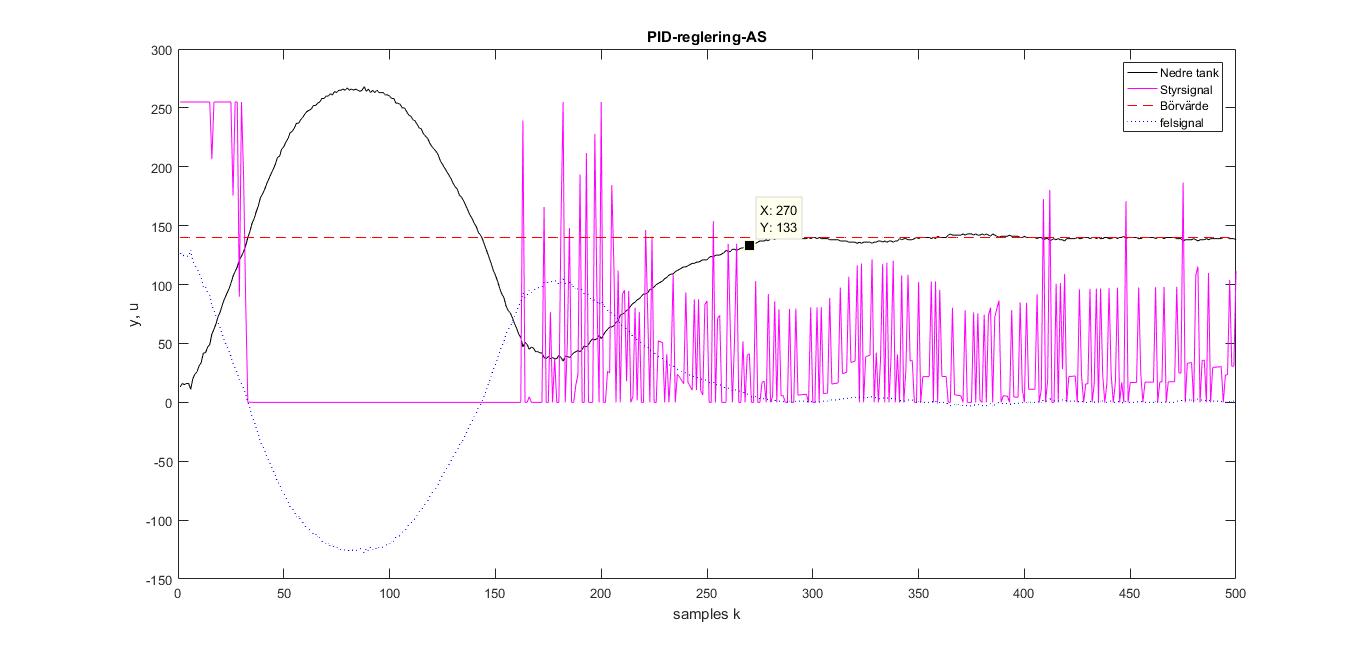
Testa nu de olika regulatorparametrarna för er PID-regulator av den nedre vattentanken.

### B.5.1 Kopiera in grafen från PID-regulatorn med Amigo-metoden-parametrarnaC:\Users\yurdaer\AppData\Local\Microsoft\Windows\INetCache\Content.Word\PID-amigo.jpg

### B.5.2 Kopiera in grafen från PID-regulatorn med Ziegler-Nichols-metoden-parametrarna



B.5.3 Kopiera in grafen från PID-regulatorn med Åström & Hägglunds parametrar



# C.4 Jämförelse mellan olika reglersystem

Ni har reglerat samma process (=vattnetankmodell) med tre olika inställningar av en PID-regulator. Jämför de tre resulterande systembeteende med varandra genom att fylla i tabellen nedan.

### C.4.1 Jämförelse av stystemens egenskaper. Fyll i tabellen:

Stigtid är tiden mellan 10% och 90 % av slutvärdet

Vi använde 140, så 10% av 140 är 14 och 90% av 140 är 126

|  |  |  |  |
| --- | --- | --- | --- |
|  | PID-Amigo | PID-ZN | PID-ÅH |
| Stigtid | 148.8 s eller  186 samples | 22.4 s eller  28 samples | 22 s eller  28 samples |
| Max. översvängning | 0 | 160 | 128 |
| Insvängningstid | 171.2 s eller  214 samples | 284 s eller  356 samples | 216 s eller  270 samples |
| Kvarstående fel | 0.9375 | 2.2500 | 1 |

# A.4 Bestämning av tidsdiskreta överföringsfunktionen

Utifrån stegsvaren av de övre och undre vattentankarna kan man med hjälp av minsta kvadratmetoden räkna ut överföringsfunktionerna. Det kallas också för en ”Black-box”-systemidentifikation.

Använd de filtrerade stegsvaren för bästa resultat.

A.4.1 Beskriv och visa steg för steg hur ni bestämmer överföringsfunktionen för den övre vattentanken utifrån det filtrerade stegsvaret:

Vi behöver en matris Y som har en kolumn och N-1 rader, dvs 450-1 = 449

Vi har alla filtrerade värden i en array yf1 med minsta värdet först och största sist.

De värden ska vi kopiera till matrisen Y så att matrisen börjar med arrayens högsta värde och slutar med arrayens minsta+1-värde.

Därefter ska vi skapa en annan matris A som har en två kolumner och igen 449 rader. Den första kolumnen ska ha näst-största värdet från arrayen yf1, eller andra elementet från matrisen Y, dvs A(1,1) = Y(2) osv.

Andra kolumnen i matrisen A är konstant eftersom det är stegsvar och vi använde oss av 50 som konstant värde på styrsignalen.

Sedan ska vi skapa en matris R som har en kolumn och två rader som ska föreställa de konstanter vi behöver för vår överföringsfunktion.

Nu har vi en ekvation, matrisekvation: Y = AR

Om vi multiplicerar A med invA från vänster får vi enhetsmatris och R blir ensamt på högersida. Då ska vi multiplicera Y med invA från vänster också, och nu får vi

invAY = R

Med matlab kan vi lösa denna ekvation enkelt:

y = zeros(1,449);

s = 449;

A = zeros(449, 2);

R = zeros(1, 2);

for k = 2:450

y(s) = yf1(k);

A(k-1, 1) = yf1(s);

A(k-1, 2) = 50;

s = s - 1;

end

Y=y';

R = A\Y;

Då får vi att R =

0,988382029269941

0,0719456487030274

Från den blir a = 0,99838 och b = 0,07194

Vilket motsvarar överföringsfunktionen

A.4.2 Beskriv och visa steg för steg hur ni bestämmer överföringsfunktionen för den övre och undre vattentanken utifrån det filtrerade stegsvaret:

Vi kan bestämma överföringsfunktionen för den undre vattentanken på samma sätt som vi gjorde i föregående uppgiften men vi ska använda mätvärden från filtrerade stegsvar för undre vattentank yf2.

y = zeros(1,449);

s = 449;

A = zeros(449, 2);

R = zeros(1, 2);

for k = 2:450

y(s) = yf2(k);

A(k-1, 1) = yf2(s);

A(k-1, 2) = 50;

s = s - 1;

end

Y=y';

R = A\Y;

Då får vi att R =

0,995776078085127

0,0247480196743680

Från den blir a = 0,99578 och b = 0,02475

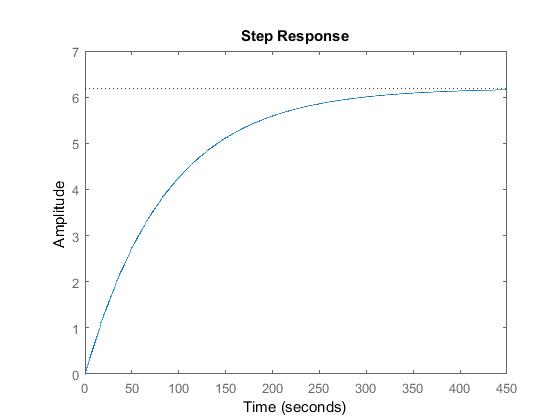
Vilket motsvarar överföringsfunktionen

Eftersom vi har andra ordningens system blir slutgiltiga överföringsfunktionen för nedre tank ;

# A.5 Simulation av stegsvaren med Matlab Control-Toolbox

Med hjälp av överföringsfunktionerna från Black-box systemidentifikationen kan man i Matlab simulera stegsvaren.

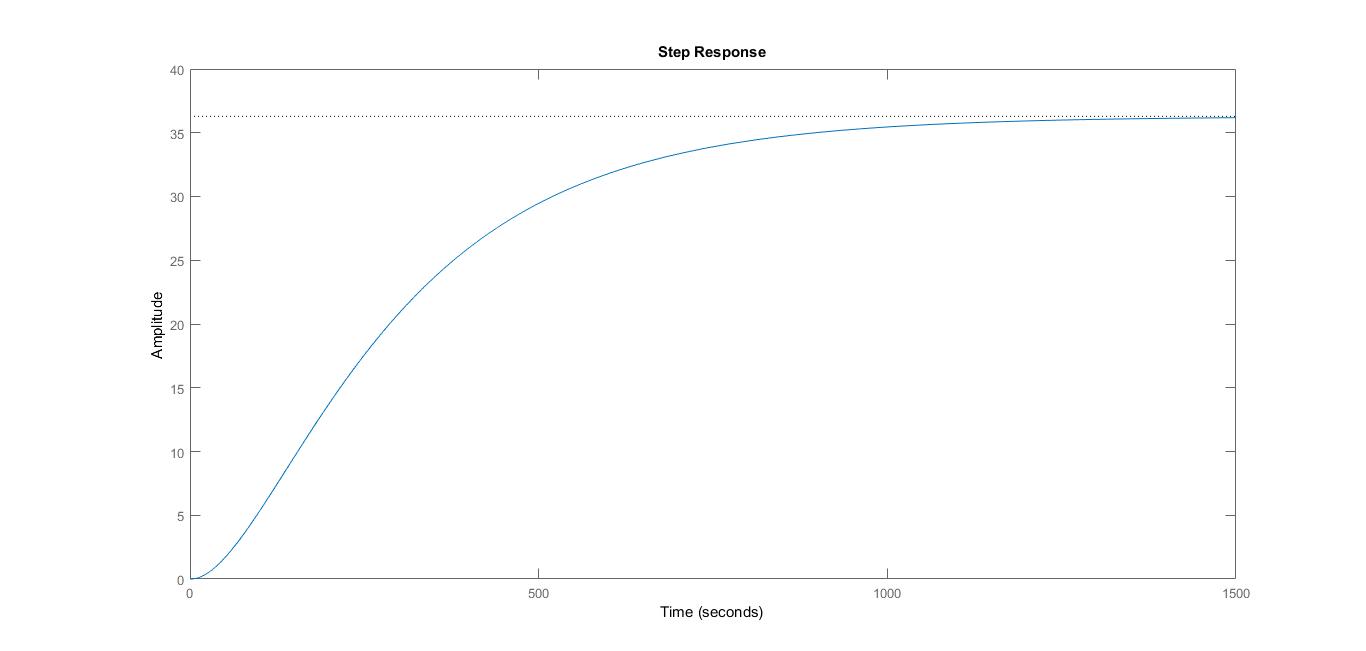
### A.5.1 Simulera stegsvaren av övre vattentank med Matlab och spara resultatet i en variabel.

Klistra in grafen här:

Variabelnamn med sparade stegsvar: **sim1**

A.5.2 Simulera stegsvaren av övre och nedre vattentank tillsammans med Matlab och spara resultatet i en variabel.

Klistra in grafen här:

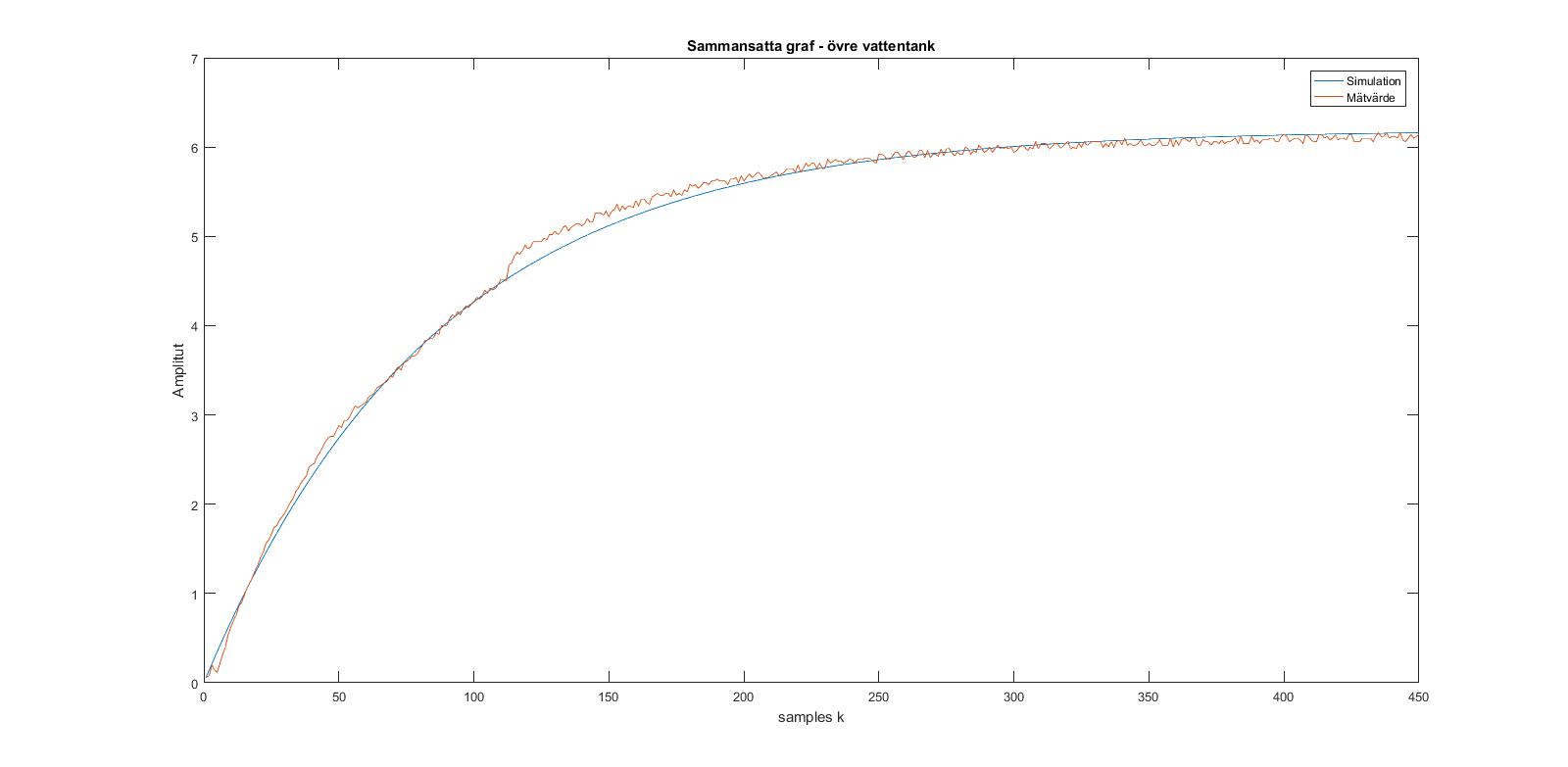


Variabelnamn med sparade stegsvar: **sim2**

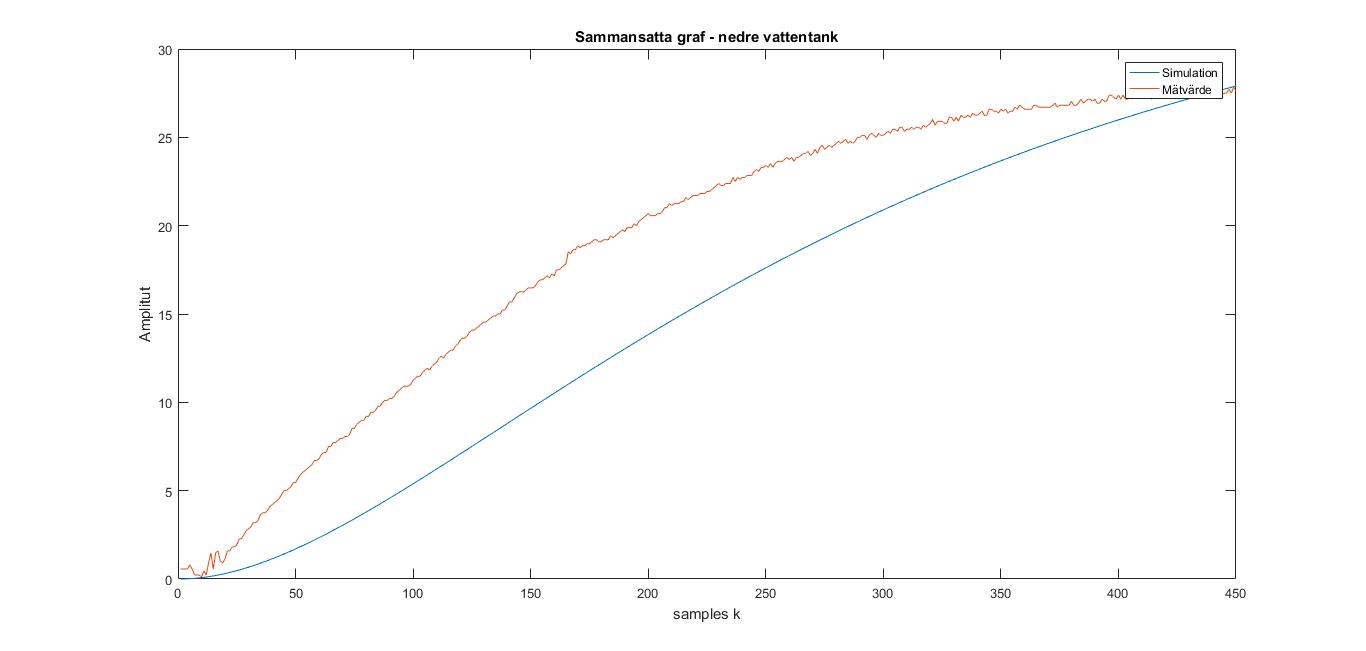
# C.5 Jämförelse mellan verkliga och simulerade stegsvaren

Till slut ska ni jämföra de verkliga stegsvaren (innan filtrering) från själva stegsvarsexperimenten med de simulerade. Genom att använda sig av Matlabkommandon ”hold” kan man plotta flera kurvor över varandra i samma plot, se också Matlabguiden. Ni ska ha sparat de olika stegsvaren i variabler. Dessa kan ni nu plotta i samma graf om ni ser till att de är lika långa.

C.5.1 Jämförelse mellan verkliga och simulerade stegsvaren för övre vattentank. Klistra in sammansatta grafen här:



C.5.2 Jämförelse mellan verkliga och simulerade stegsvaren för övre och nedre vattentank tillsammans, dvs för nivån i nedre vattentanken. Klistra in sammansatta grafen här:



# Reflektion och utvärdering över det egna lärandet

Tidigare studenter har framfört olika åsikter om denna tredje del. För att förklara vad den går ut på hänvisar jag till kapitel om ”Reflektionsdokumentet” i Andersen och Schwenckes bok ”Projektarbete – en vägledning för studenter”, Studentlitteratur. Författarna beskriver syftet med reflektion över sitt lärande sammanfattningsvis ungefär på följande sätt:

Det handlar om att öka medvetenheten om den egna inlärningsprocessen. Genom denna reflektion lär du känna dig själv, och hur du förhåller dig till dina medstudenter. Du blir också tryggare i dig själv och det du har gjort och står för. Du blir också mer medveten om hur du lär, så att du kan styra inlärningen och studierna i din egen riktning och ta ansvar för din egen inlärning. Du utvecklar medvetenhet om inlärning och stärker din egen förmåga att arbeta effektivt på längre sikt.

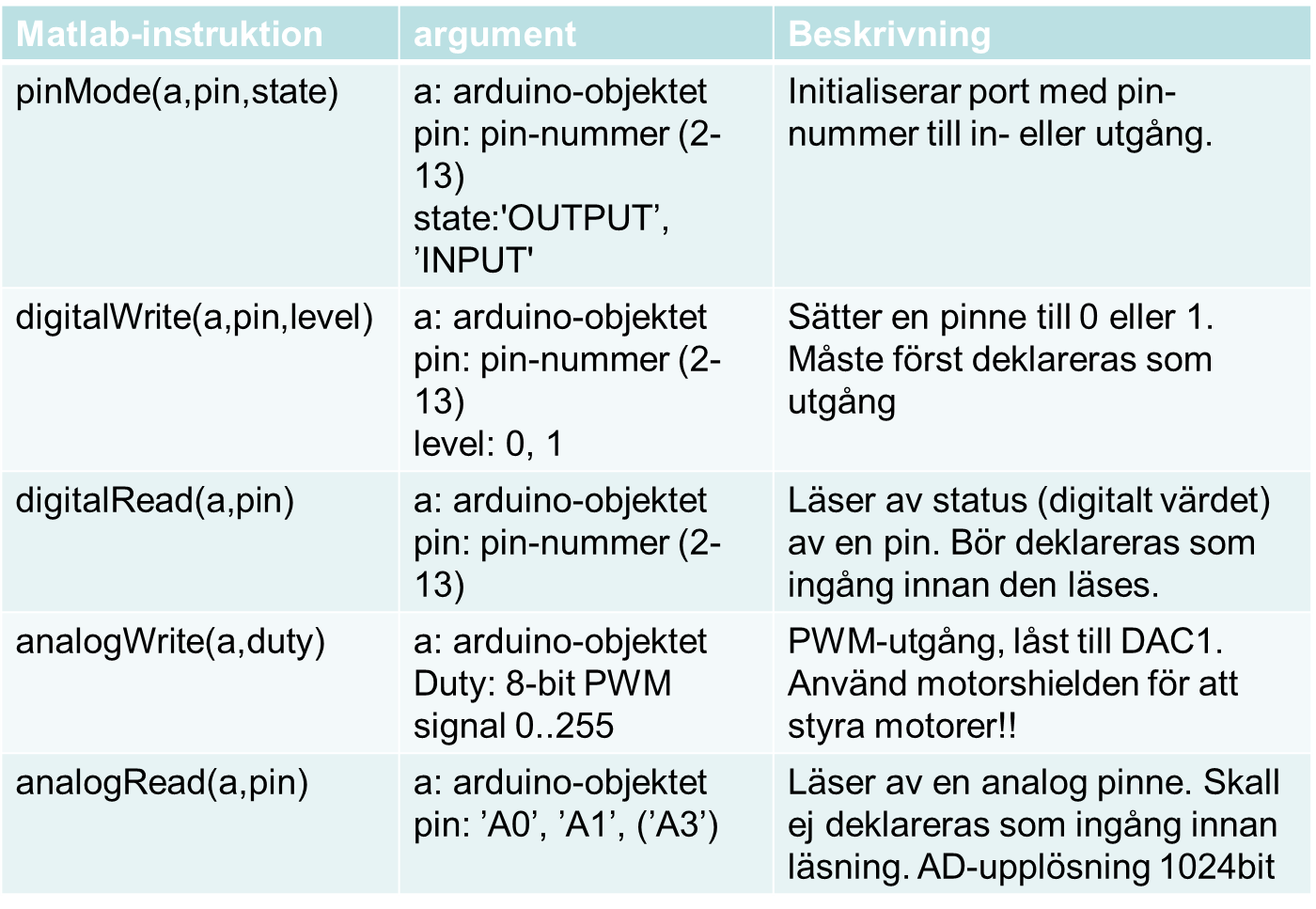
Vidare är det ett viktigt syfte att ge feedback, både till läraren, kursansvarig och till skolan. Därmed blir det möjligt att göra förbättringar för senare studentkullar.

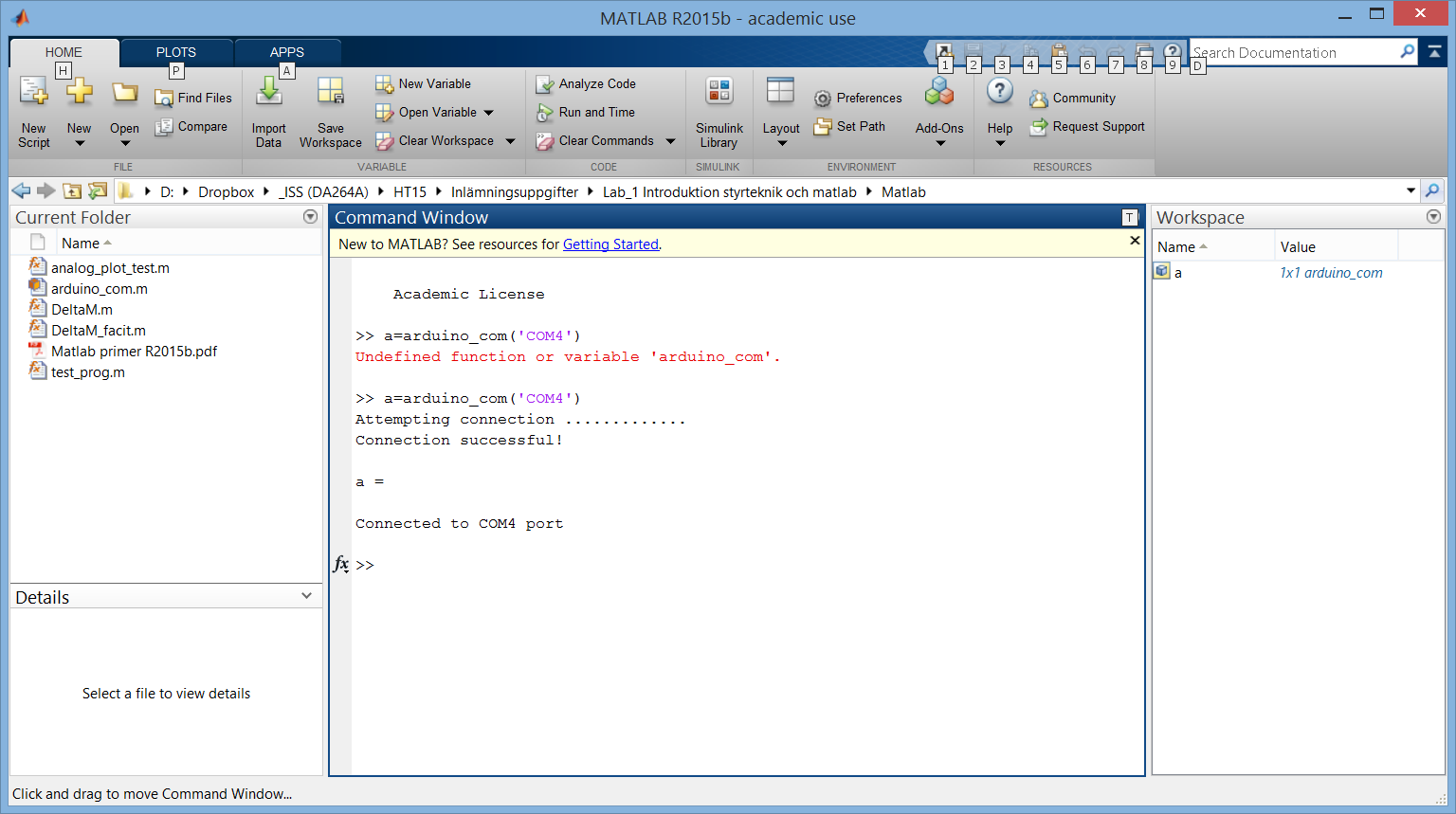
I fall att ni känner att ni inte kan komma på något vettigt att skriva om, så finns en lista med nyckelfrågor i bilagan som ni kan titta på och få inspiration ifrån.

|  |
| --- |
| D.1 Vad tycker du/ni var lärorik med uppgiften? (Minst 5 meningar och minst 75 ord!)  Som andra uppgifter är det svårt att använda tiden så effektivt som möjligt när vi är två olika individer i gruppen med olika åsikter och olika synpunkter. Det har varit tufft i början under tiden lärde vi oss att jobba effektivt och att samarbeta effektivt också. Det har varit lärorikt att fördjupa sig någorlunda i reglering, med tanke på att den bilden vi hade innan kursens start var att reglering var mer eller mindre on-och-off-metod för att få ett önskvärt resultat. Visserligen så ingår on-och-off i reglering, dock visade det sig rätt så snart att det handlar om massa andra metoder som överraskade oss när det gäller resultatet. |
| 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!)  Som sagt så var vi ej så insatta i det området sedan tidigare och de experiment vi har gjort på fritiden tidigare har handlat om att värma ett rum genom att mäta rumstemperaturen och med hjälp av den avgöra när man ska sätta på värmefläkten och när man ska stänga av den. Det var enkel on-och-off-metod. Det som var rena överraskningen kom när vi började med PD och PID-reglering och sedan med amigo, Zieglers och Åhström för där föll äpplet från trädet och WOW! Vi blev imponerade, mest av Åhström-metoden. |
| D.3 Vad var det svåraste med uppgiften? (Minst 5 meningar och minst 75 ord!)  Tack vare ordentligt genomförda genomgångar med Gion var det inte så jättesvårt. Det var dock mycket att göra på en gång. Vi lade mycket tid på bland annat på amigo-metoden och fick göra om mätningar flera gånger på grund av att vattennivån i den undre tanken aldrig började från noll. Några matlab-funktioner var svåra eftersom vi aldrig hade jobbat med dem tidigare som till exempel hur man får ut globala variabler. Rent generellt var det svåraste att arbeta som en ingenjör och se till att ha sparade värden redan från början. Återigen med lite tips och mindre hårda påpekningar från Gion kom vi långt med denna laboration.  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 i genomsnitt 4 timmar per dag i 7 dagar, dvs runt 28 timmar. Av det har vi lagt cirka 12 timmar på de svåra bitarna. |
| D.4 Synpunkter, förslag, kommentarer? (Minst 5 meningar och minst 75 ord!)  Det enda på rak arm vi kan komma på är kanske att tiden eller rättare sagt tillgängligheten av vattenkannorna var inte den bästa under laborationstillfälligt förra onsdagen. Det var dock inte så farligt eftersom vi hade tillgång till dem alla andra dagar och eftersom vi alla i klassen rent generellt har lärt oss att samarbeta och dela på utrustning mellan varandra. Laborationen var kanske lite för stor men å andra sidan är det värt i jämförelse med det vi har lärt oss hittills av den. |

# Bilaga

## Översikt över Matlab instruktioner





## Exempel av nyckelfrågor i samband med reflektioner och utvärdering av det eget lärande

* Är du nöjd med samarbetet i gruppen?

Yurdaer är nöjd med

* Vad fungerade bra vad mindre bra?
* Hur fungerade beslutsprocessen i gruppen? Kunde det ha varit bättre?
* Vilken rolluppdelning valde ni? Hur kändes det?
* Hur effektivt och systematiskt var ni? På vilket sätt finns utrymme till förbättring?
* Hur uppfattade ni sambandet mellan teori och tillämpning?
* Hur uppfattade ni sambandet mellan föreläsning och laboration?
* Genomförde ni uppgifterna i ordningen som de står i rapporten eller i vilken ordning besvarade ni frågorna? Hur kändes det så som ni gjorde?
* Vad gjorde ni när det blev problem och/eller ni inte kom vidare? Använde ni både samma strategi? Hur kändes det?
* Hur organiserade ni er inför slutinlämningen? Gick ni igenom alla frågor en gång till gemensam eller lämnade ni bara in allt när ni kände att ni var färdiga?
* Planerade ni fasta tider när ni gemensam i gruppen genomförde laborationsuppgifterna eller hur organiserade ni er?
* ??