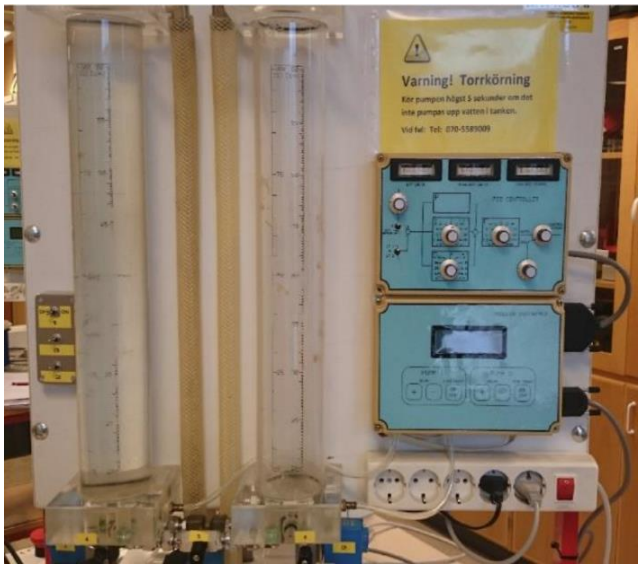
CHALMERS



Utveckling av en PID-regulator

Examensarbete inom Data- och informationsteknik

Anders Nordqvist
Omar Kelfaoui

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET

Göteborg 2017

EXAMENSARBETE

Ombyggnad av PID-Regulator

ANDERS NORDQVIST
OMAR KELFAOUI

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET

Göteborg 2017

Ombyggnad av PID-Regulator

ANDERS NORDQVIST
OMAR KELFAOUI

Examinator: Peter Lundin

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola
Göteborgs Universitet
412 96 Göteborg
Telefon: 031-772 1000

Institutionen för Data- och Informationsteknik
Göteborg 2017

Sammanfattning

Ett hjälpmedel på Chalmers Tekniska Högskola som har till syfte att underlätta studier av reglersystem har nu efter 20 år i drift blivit utslitet, är baserat på gammal teknik och behöver bytas ut. Utvecklingen av små, enkla, kraftfulla och framförallt billiga mikrokontrollerkort ger möjlighet till att med enkla medel ersätta äldre analog hårdvara med digital hårdvara med avsevärt bättre prestanda.
En PID-regulator implementerades på ett Arduino mikrokontrollerkort och för att visa information till användaren användes både en LCD display samt en pekskärm. För att programmera kontrollkortet användes två olika utvecklingsmiljöer, Arduino IDE samt Microsoft Visual med tillägget Visual Micro. Ett alternativt sätt att skapa en PID regulator utvecklades med hjälp av programvaran LabView.

Abstract

A device at Chalmers University of Technology, that is used in the study of control systems has become worn out after 20 years and must be replaced. The current device is based on analog technology should be upgraded to a digital platform. This report deals with the technical systems used to develop such a system. The development of small, simple, powerful, and inexpensive microcontroller cards allows for easy replacement of older analog hardware with digital hardware with significantly better performance. A PID-controller was developed and deployed to a Arduino microcontroller card. A LCD display and a touch screen was used to display information to the user and to develop the software two different developing environment, Arduino IDE and Microsoft Visual Studio, were used. An alternative PID-controller was developed in conjunction with LabView software.

FÖRORD

Examensarbetet är utfört på institutionen för Data- och Informationsteknik på Chalmers Tekniska Högskola. Examensarbetet omfattar 15hp och har utförts 2017 som en del av elektroteknikprogrammet 180hp. Vi vill tacka Göran Hult för att vi fick möjligheten att utföra detta examensarbete och för hjälpen vi fått under projektets gång. Ett speciellt tack vill vi ge till vår handledare Sakib Sistek som varit ett oumbärligt stöd och fantastisk mentor under dessa veckor.

Innehåll

1. Inledning

Avsnittet innehåller bakgrund, mål och syfte med detta projekt. Även valda avgränsningar och frågeställningar behandlas

1.1 Bakgrund

På Chalmers Tekniska Högskola i Göteborg finns ett laborationssystem vars syfte är att ge förståelse för och på ett bra sätt visualisera hur en PID-regulator fungerar. Detta system är nu över 20 år gammalt, de komponenter som bygger upp systemet börjat bli slitna och behöver bytas ut. Ansvariga för laborationssystemen har föreslagit att man vill ersätta själva PID-regulatorn i systemet med modernare hård- och mjukvara.

1.2 Syfte

Att utveckla en ny PID-regulator baserat på ett mikrokontrollerkort för att ersätta den äldre analog regulatorn som behöver bytas ut.

1.3 Mål

Målet är att utveckla en PID-regulator som enligt beställaren i första hand skall realiseras baserat på ett Arduino mikrodatorkort. Regulatorn skall kunna styras av omkopplare och kontrollers på en panel och data från regulatorn skall presenteras på en digital skärm. Data skall även visas på en ansluten PC i form av grafer. Inom ramen för projektet skall även alternativa lösningar av PID-regulatorn undersökas.

1.4 Avgränsningar

Projektet kommer enbart bestå av konstruktion av själva regulatorn. Komponenter som inte kommer att beaktas eller modifieras är sensorer, kontrollbox för inställning av fördröjning, tankar och pumpar.

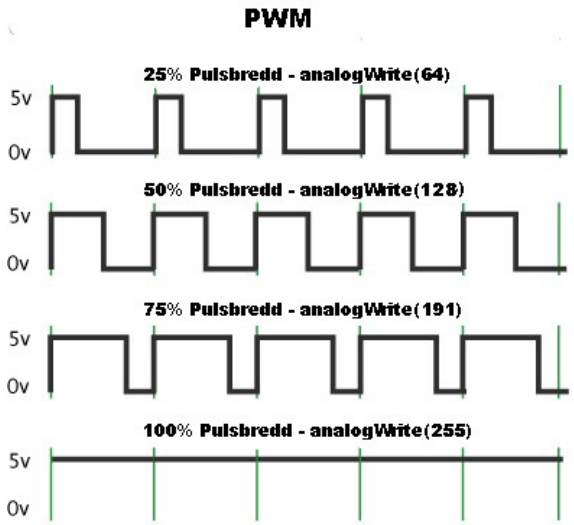
2. Teori

2.1 PWM / Arduino

Pulsbreddsmodulering (Eng. Pulse Width Modulation, PWM) är en teknik för att med digitala pulser kontrollera och styra analoga kretsar. PWM används i många olika sammanhang. PWM signalen består av ett spänningspulståg där pulsbredden avgör hur mycket effekt som överförs till aktuell last. Både DC och AC kan simuleras med hjälp av PWM. PWM signalen tolkas som en likspänning när den elektromekaniska tidskonstanten för lasten ifråga är längre än periodtid för PWM signalen [1]. Följande ekvation visar hur spänningen *v* är beroende av pulsernas längd och hur ofta dessa upprepar sig. Amplituden för PWM signalen är Vs eller 0. Om signalen matar en last med en tidskonstant som är mycket större än t_{pwm}, kommer lasten att uppfatta signalen som en likspänning med medelvärdet *v*. [1]

$$v = v_s \frac{T_{puls}}{T_{pwm}}$$

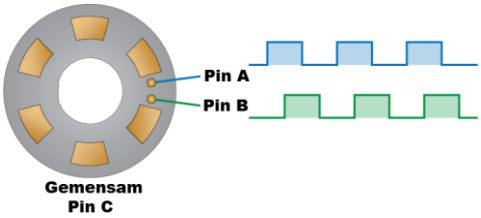
Kvoten $t_{puls} \,/\, t_{pwm}$ kallas för fyrkantspulsens pulskvot (eng. duty cycle) och för att reglera spänningen v ökar eller minskar man denna kvot. Arduino Mega som används i detta projekt har 54 digitala in/utgångar varav 14 av dessa kan användas som PWM utgångar. Frekvensen på den PWM signal som genereras är 490Hz, förutom på pin 4 och 13 där frekvensen är 980Hz. De spänningsnivåer som en PWM signal från ett Arduino Mega kan anta är 0V respektive 5V. I vårt fall ger detta att v_S= 5 V och vid en frekvens f=490Hz ges en periodtid t_{pwm}=2.04ms. För att skapa en PWM signal med ett Arduinokort kan man använda en färdigt biblioteksfunktion analogWrite(), där man i funktionens argument anger ett 8-bitars tal som motsvarar spänningsområdet 0-5V [2].



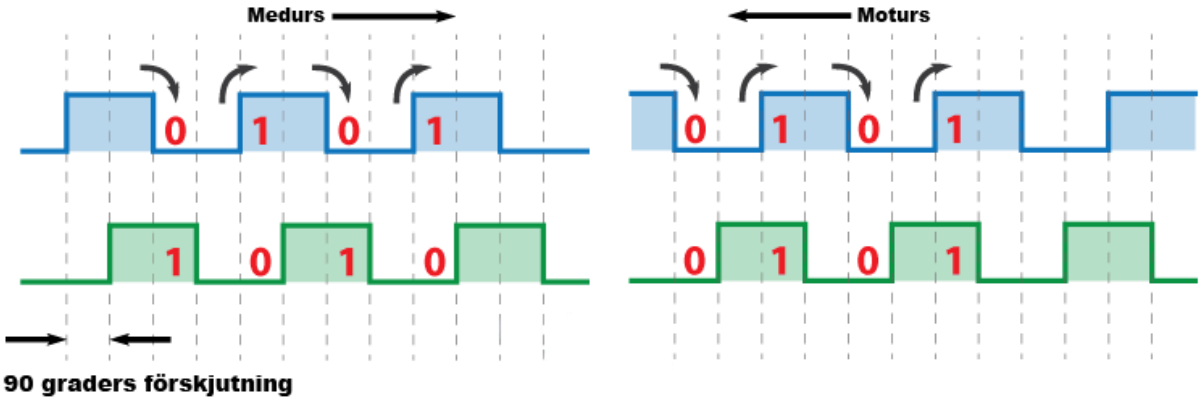
2.2 Pulsgivare

En rotary encoder, som kallas pulsgivare på svenska, är en elektromekanisk komponent som när dess axel roterar genererar en analog eller digital signal. Det finns flera olika typer av pulsgivare och för detta projekt valdes den enklaste varianten, inkrementell pulsgivare. Denna pulsgivare nollställs varje gång strömmen slås på eller av och axelns position anges i förhållande till dess startposition. [3]

Pulsgivaren består av en rund disk med jämt åtskilda kontaktytor som är anslutna till ett gemensamt stift C och två andra stift A och B enligt figur 1



Figur 1. Visar pulsgivares funktion



Figur 2. Visar hur pulsgivare roterar.

När disken roterar kommer i tur och ordning stiften A och B att sluta kretsen med stift C och två fyrkantspulser skapas härmed. Valfri puls kan användas för att detektera om givaren har roteras men för att bestämma i vilken riktning måste båda dessa signaler jämföras med varandra. Som figur 2 visar ser vi att signalerna från A och B är förskjutna med 90 grader. Om pulsgivaren roterar medurs är pulsen från A förre den från B.

Om vi nu detekterar varje gång som pulserna ändrar tillstånd, från hög till låg eller låg till hög, upptäcker vi att i fallet när givaren roterar medurs får det två signalerna motsatta värden. När givaren roterar moturs får signalerna samma värden [4]. Se figur 2

2.3 PID Regulator

PID / Återkopplade system

Moderna reglersystem finns idag i många olika skepnader och i många olika tekniska sammanhang. Olika tillämpningsområden där reglersystem är mycket vanligt är bland annat inom processindustrin, inom kraft och energiproduktion och inom fordonsindustrin. Storheter som regleras kan vara temperatur, tryck, flöden och riktning, bara för att nämna några få.

Maskiner som automatiskt varierar en vald storhet för att reglera en annan har funnits sedan antiken där flera anordningar användes för att reglera vattenflöden med hjälp av en flottör [5]

Reglersystem kan delas in i öppna och slutna system. I ett öppet system återförs ej systemets utsignal och någon jämförelse görs ej med önskat börvärde, därav kan utsignalen inte påverka styrsignalen. Fördelen med ett öppet system är att det är enkelt men används framförallt i

system där insignaler och utsignaler är kända samt att eventuella störningar som kan påverka systemet är få [5].

I ett slutet system återförs utsignalen och jämförs med börvärdet för att kunna påverka den aktuella styrsignalen. De flesta reglersystem utnyttjar idag någon form av återkoppling där informationen om tillståndet för en given process används för att styra denna i önskvärd riktning. Det är framförallt fyra huvudorsaker till varför man använder återkopplade system [5].

- Med **störningsdämpning** vill man att dämpa inverkan av störningar. Att systemet automatiskt motverkar de störningar som kan komma att påverkar systemet
- Att så snabbt och noggrant som möjligt följa ett varierande **börvärde.**
- Att åstadkomma ett system som är **robust**, dvs att effektivt dämpa de **dynamiska variationer** som kan uppkomma hos de ingående delarna i ett system.
- Att **stabilisera** instabila processer.

Idag finns det ett flertal olika regulatorer på marknaden, men där flertalet av dessa härstammar från PID regulatorn [6]. PID är en förkortning av proportionell, integrerande och deriverande funktion. Regulatorns uppgift är att styra ärvärdet mot önskat börvärde genom att ändra styrsignalen. I en sådan regulator kan styrsignalen anta alla värden i ett visst givet intervall. PID regulatorn kan beskrivas i följande matematiska form:

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

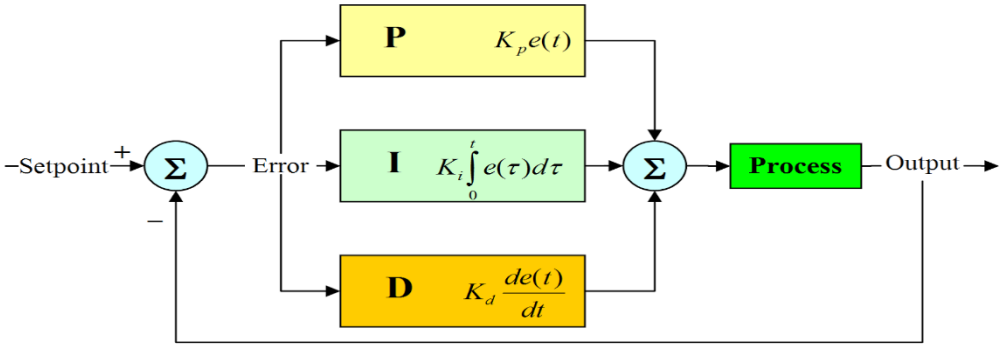
K = Förstärkning

T_I = Integrationstiden

T_D = Deriveringstid

e = felsignal

Figur 3 visar en PID regulator med hjälp av ett blockschema.



Figur 3. Visar PID Blockschema.

- System/Process – Det aktuella system eller process som skall regleras
- Återkoppling – Information om den reglerade storheten
- Börvärde – Det värde som man önskar att systemet/processen skall anta
- Felsignal – Differensen mellan önskat börvärde och aktuell utsignal
- Störning - En störning som påverkar systemet
- Regulatorn – I vårt fall en PID Regulator

Regulatorn består som tidigare nämnts av tre delar.

2.3.1 P-Regulatorn

Den proportionella delen, beskrivs matematisk med formeln:

$$u(t) = K \cdot e(t) + u_0 \text{ [5]}$$

K är här förstärkningen medan u_0 är styrsignalens normalvärde, dvs det värde styrsignalen skall ha då reglerfelet är noll. K talar om hur mycket styrsignalen skall ändras vid en förändring av aktuellt reglerfel. Det finns dock begränsningar på hur hög respektive låg styrsignalen kan bli, dvs styrsignalen är enbart proportionell mot reglerfelet inom ett visst intervall, det så kallade proportionella bandet [5]. Hur man väljer K bestäms av vilka egenskaper man vill att det aktuella reglersystemet skall ha samt om processen som regleras har en inbyggt förstärkningen eller ej. Litet värde ger ett stabilt men långsamt system medan ett större värde kan skapa instabilitet men skapar ett system som snabbare svarar på

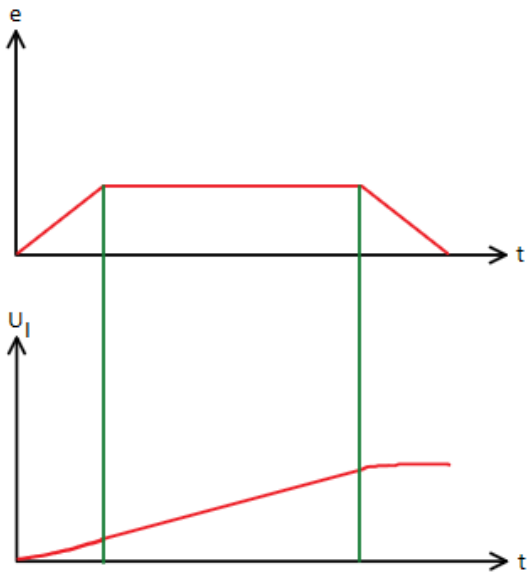
störningar. En avigsida av att enbart använda proportionell reglering är att den i många fall ger ett kvarstående fel vid tex stegformade störningar eller börvärdesändringar, om inte styrsignalens normalvärde ändras vid varje börvärdesändring men detta är ofta inte praktiskt möjligt. För att motverka bestående fel används P-regulatorn ofta i kombination med en enhet med Integrerande verkan [5].

2.3.2 PI-Regulatorn

Denna regulator består som tidigare av en proportionell funktion och av en integrerande funktion. Integralfunktionen ersätter P-regulatorns normalvärde u_0 men med den skillnaden att justeringen sker här automatiskt. PI-regulatorns ekvation är som följer:

$$u(t) = k\left(e(t) + \frac{1}{T_i}\int_0^t e(t)\,dt\right)$$

Integraldelen utsignal i relation till felförloppet. Se figur 4



Figur 4.

Styrsignalen är följaktligen beroende på hur stor felintegralen varit fram till aktuell tidpunkt. Med denna integrerande funktion elimineras det bestående reglerfel som tidigare uppkom, eftersom integralen av aktuellt reglerfel ökar eller minskar så länge detta är skilt från noll. Detta gäller vid börvärderändringar som vid störningar. Integraldelen måste dock förhindras

att växa alternativt minska när styrsignalen har antagit sitt maximal eller minsta värde. Denna funktion kallas för anti-windup och ser till att integraldelen upphör om styrsignalen nått sitt maximala eller minimala värde. Om integraldelen skulle tillåtas att växa okontrollerat skulle det ta lång tid att minska alternativt öka denna när reglerfelet ändras [5].

2.3.3 PID-Regulatorn

Vad som saknas hos PI-regulatorn är att den inte gör några försökt till att förutspå i vilken riktning som reglerfelet rör sig mot. Den tredje delen i vår ekvation som beskriver PID regulatorn är $K \cdot T_D \cdot e'(t)$, där styrsignalen är proportionell mot på derivatan av aktuellt reglerfel. Tanken med detta är att skapa en förmåga att uppskatta hur reglerfelet förväntas utvecklas. Styrsignalen från denna del är skild från noll endast då regelfelet ändrar på sig [6]. Vid större ändring av reglerfelet svarar regulatorn med en större motverkande styrsignal.

3. Systemkomponenter

Detta kapitel kommer att beskriva de olika komponenterna som har använts i projektet.

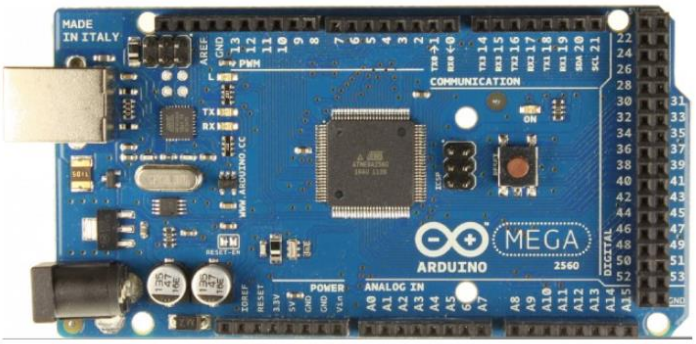
3.1 Val av kontrollerkort och alternativa system

Valet av att implementera en PID-regulator på ett Arduino mikrokontrollerkort är bestämt av beställaren. Alternativ till att använda ett Arduino kort undersöktes, dessa beskrivs i korthet nedan

- PIC Mikrokontroller
  Ett billigare alternativ än Arduino. Dock inte lika användarvänlig och enkel att programmera.
- FPGA Mikrokontroller
  Snabbare och kan utföra flera operationer samtidigt. Ett dyrare alternativ till Arduino.
- Rasberry Pi
  Används när det krävs en fulländad dator. Har ett operativsystem baserat på Linux.

3.2 Mikrokontrollerkort Arduino Mega 2560

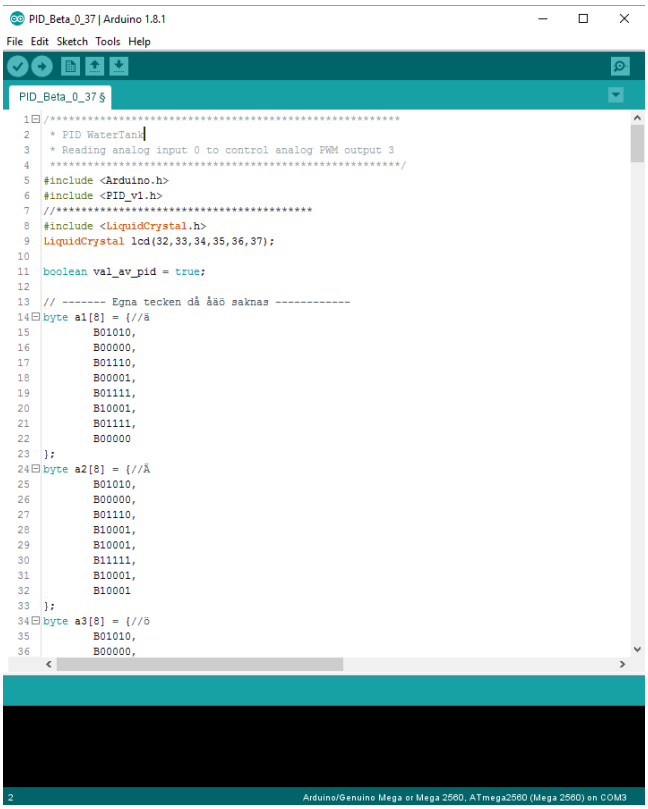
Utvecklingen av det nya systemets PID-regulator kommer att ske med hjälp av ett Arduino Mega 2560 mikrokontrollerkort. Detta kort är baserad på en Atmel 2560 processor på 16Mhz, har 16 stycken analoga ingångar och 54 stycken digitala in/ut gångar varav 14 kan användas som PWM utgångar. Det har 256Kb flashminne för lagring av kod (8Kb används av bootloader), 8Kb av SRAM och 4Kb EEPROM minne. Kortets har en arbetsspänning på 5 volt och en matningsspänning mellan 7–12 volt [7].



Figur 5. Arduino Mega 2560.

Det är ett mikrokontrollerkort med öppen hårdvara och har en användarvänlig utvecklingsmiljö där man utvecklar programmen och överför dem till mikroprocessorn som sitter på Arduinokortet. Arduino IDE levereras med ett C/C++ bibliotek som kallas wiring, som gör in- och utmatningsoperationer mycket enklare. Utvecklingsmiljön är skrivet i Java och är ett derivat från projekt som Processing och Wiring.

Efter att kod har skapats, kompilerar och översätts koden i IDE till maskinvarukod som sedan laddas upp till processorn. Utvecklingsmiljön har en inbyggd kodtolk (eng. code parser) som kontrollerar syntaxen för koden innan den kompileras och laddas upp till processorn.



Figur 6. Arduino IDE.

Koden laddas upp via ett USB gränssnitt som varierar från modell till modell.

3.3 Pulsgivare och omkopplare

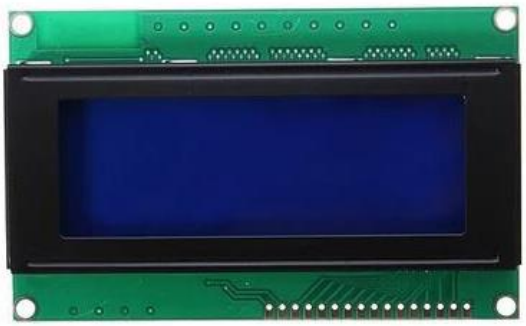
Inmatning av parametrar (K, Ti, Td) till regulatorn skall enligt specifikation av projekt ske via omkopplare och kontroller som liknar de befintliga på kontrollpanelen. I nuvarande form sker detta med hjälp av tvålägesomkopplare och potentiometrar.

I strävan att efterlikna känslan och utseendet hos dessa komponenter beslutades att fem så kallade pulsgivare (eng. Rotary Encoders) skulle användas. Pulsgivaren fungerar så att den genererar pulser när den vrids ett steg. (se Kap 2.2). Dessa pulser detekteras och räknas upp i en variabel som sedan används för att avgöra hur många steg pulsgivaren har vridits från sitt ursprungliga läge.

Figur 7. Pulsgivare / Encoder.

3.4 Skärmar

En LCD-skärm valdes med storleken 20x4, 4 rader med 20 tecken på varje rad, med bakgrundsbelysning. Skärmen har en inbyggd drivkrets som är kompatibel med Hitatchi HD44780 LCD kontroller. Detta innebär att det finns ett färdigt standardbibliotek att använda, LiquidCrystal.h, när man programmerar Arduino. Upplösningen på skärmen är 0.54mm x 0.54mm per punkt.



Figur 8. LCD-Skärm.

För att visa på ett annat sätt att presentera och mata in värden till regulatorn byttes pulsgivarna och LCD-skärmen ut mot en pekskärm. Pekskärmen är 3.5” med en upplösning på 320x480 pixlar.

Figur 9. Pekskärm.

3.5 OP Förstärkare (LM358)

Operationsförstärkaren LM358 är en förstärkare som är byggd för flera olika syften och kan användas inom flera områden. Den består av två stycken av varandra oberoende förstärkare med hög förstärkning och frekvenskompensering. Både förstärkarna drivs med samma drivspänning i ett intervall om 3-32V likspänning. Spänningsförstärkningen är 100 dB med en bandbredd på 1Mhz. Kretsen användes både som en spänningsföljare samt som en förstärkare.

3.6 E-tape Nivågivare / Sensor

Syftet med en sensor är att översätta en fysisk storhet som till exempel temperatur eller tryck till en elektrisk signal som kan hanteras av elektroniska kretsar. Sensorer är komponenter som mäter en fysikalisk storhet och omvandlar den till en analog eller digital spänning. Ett exempel på en sensor är en optoresistor som omvandlar ljusstyrka till en resistans som kan påverka en spänning. Sensorvärdet kan kopplas till en analog ingång på Arduinokortet som med hjälp av en A/D omvandlare omvandlar spänningssignalen till ett digitalt tal som sedan kan bearbetas i processorn.

En typ av nivågivare består av en 25 cm lång och 2,5 cm bred plastfilm som komprimeras av det hydrostatiska tryck som utövas av den vätska som den är nedsänkt i. Detta resulterar i en förändring av resistansen vilket motsvarar avståndet från sensorns övre del ner till vätskeytan. Sensorns resistivitet är omvänt proportionell mot vätskans nivå, ju lägre vätskenivå desto

högre resistans, högre vätskenivå ger lägre resistans. En sådan sensor användes vid bygget av ett demonstrations system för PID regulatorn.

Figur 10. Nivåsensor / ETape.

4. Metod

I början av arbetet ägnades största delen av tiden till att söka information. För att lösa de föreskrivna uppgifterna på ett tillfredställande sätt krävdes ingående kunskaper om både reglersystem i allmänhet och om PID regulatorn i synnerhet. Mikrokontrollerkortet Arduino är en grundläggande komponent i vår konstruktion och då ingen tidigare erfarenhet fanns angående denna plattform var det nödvändigt att lägga stor fokus på denna del av informationssökningen. Aktuell litteratur, både i bokform, manualer och på internet användes för att inhämta denna kunskap.

Mjukvaran som användes för programmering var Arduino's egenutvecklade utvecklingsmiljö, IDE, samt Microsoft Visual Studio med tillägget Visual Micro. Anledningen till användandet av Visual Studio var att felsökning var avsevärt förbättrad i denna mjukvara. Vidare undersöktes det om möjligheten att skapa en regulator i alternativ programvara, Labview.

Flera olika delprojekt skapades för att testa de olika ingående delarnas funktion var för sig såsom PID algoritm, reglage och display. Även andra exempel programmerades för att få en övergripande kunskap om hur mikrokontrollerkortet fungerar i praktiken.

5. Genomförande

5.1 Informationssökning

Uppstarten på projektet startade med att definiera och skriva en planeringsrapport samt samla information om mikrokontrollerkortet Arduino. Då ingen tidigare erfarenhet fanns i projektgruppens medlemmar angående valt mikrokontrollerkort ägnades de två första veckorna åt att lära sig denna nya hårdvara. Vi hade två utvecklingsmiljöer till vårt förfogande. Till en början användes Arduino IDE då denna är en lämplig utvecklingsmiljö att börja med för nybörjare och när projekt inte är för komplicerade. När sedan projektet växte byttes Arduino IDE ut mot Microsoft Visual Studio med tillägget Visual Micro, speciellt framtagen för att användas med Arduino. Den främsta anledningen till att använda denna IDE är att det här finns en debugger som används vid felsökning av kod. Aktuell utvecklingsmiljön installerades och bibliotek intressanta för projektet studerades. Dessa bibliotek laddades ner från Arduinos hemsida och installerades i respektive utvecklingsmiljö. De bibliotek som var mest intressanta handlade om inmatning (Encoders), regleralgoritmer (PID) samt styrning av LCD-skärmar.

Specifikationerna ställda på projektet angav fysiska kontroller på instrumentpanelen, att en LCD-skärm skall användas för att visa aktuella värden samt att det skall vara möjligt att via en PC få information om börvärde, styrsignal och ärvärde i form av kurvor.

5.2 Utveckling

5.2.1 Befintligt reglersystem

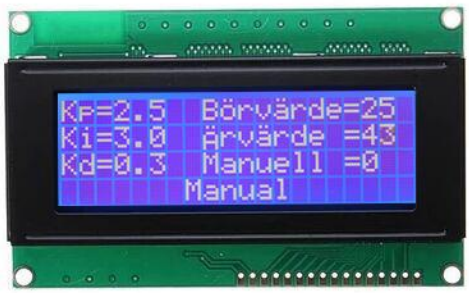
Det befintliga reglersystem som nu är i bruk är konstruerat med hjälp av analog teknik. PID-regulator är här realiserad med hjälp av operationsförstärkare i olika konstellationer och för inmatning av parametrarna K, Ti och Td används potentiometrar. För att visa regulatorns ärvärde, börvärde samt styrsignal används analoga givare.

5.2.2 Inmatning via pulsgivare

Arbetet börjar med att undersöka vilka olika möjligheter det finns för inmatning av värden till regulatorn. De olika alternativen som undersöktes var olika konfigurationer av knappar, potentiometrar samt pulsgivare (eng. Rotary Encoders). Det fastställdes att konfigurationen med pulsgivare var det som bäst passade vårat syfte och var mest användarvänlig. Flertalet olika programvaruexempel konstruerades för att få en pulsgivare att uppträda som önskat med inkrementering/dekrementering i valda steg och fungera i specificerat intervall. Dessa pulsgivare kopplades in på mikrokontrollerkortets avbrottsportar med avsikten att göra programmet så snabbt som möjligt.

5.2.3 Skärmar

Värden inställda med hjälp pulsgivare skall visas på en skärm. Dessa värden visas på en LCD-skärm. Tillvägagångssättet är som tidigare att små exempel konstruerades för att förstå hur denna skärm programmerades och fungerade. Då skärmen inte var kompatibel med svenska bokstäver (å,ä,ö) behövde standardbiblioteket LiquidCrystal.h kompletteras med egen kod. Pulsgivarna länkades sedan till denna LCD-skärm. Se figur 11



Figur 11

För att ytterligare utveckla regulatorn ersattes pulsgivarna och LCD-Skärm med en pekskärm. Denna programmerades med virtuella knappar och reglage för att motsvara den specifikation som anges i projektet. Se figur 12.

Figur 12

5.2.4 PID-algoritm

Det finns olika sätt att programmera och angripa problemet med att konstruera en PID-algoritm mjukvarumässigt men i stora drag liknar de varandra. Det konstruerades en enklare algoritm som visas i figur 13.

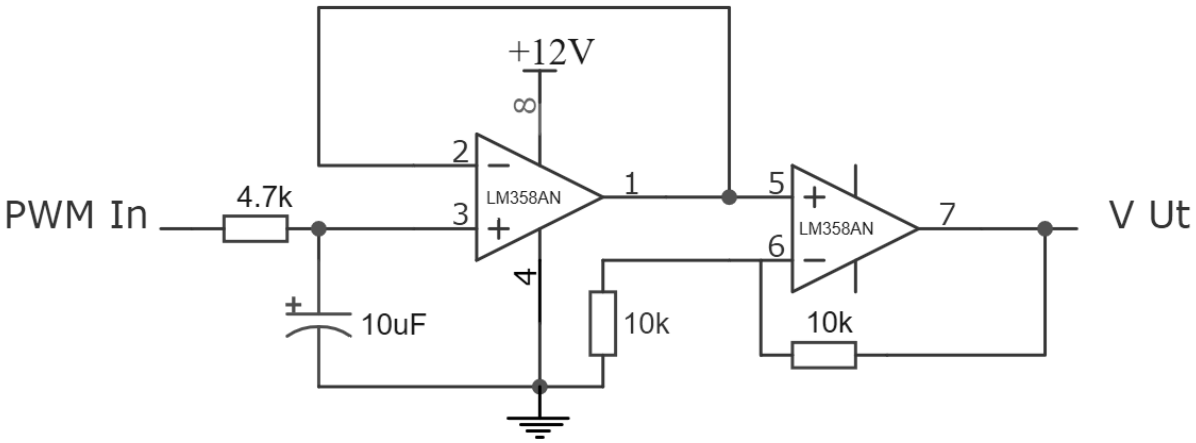
```
7
8   unsigned long L_Timer;
9   double Insignal, Utsignal, Borvarde;
10  double Fel_Integral, L_Fel;
11  double Kp, Ki, Kd;
12  int Sampling_tid = 1000; // En sekund
13  Borvarde = Encoder_Borvarde();
14  void PID_Kontroller()
15  {
16     unsigned long aktuell_tid = millis();
17     int Tid_skillnad = (aktuell_tid - L_Timer);
18     if(Tid_skillnad>=Samplings_tid)
19     {
20        double fel = Borvarde - Insignal;
21        Fel_Integral += fel;
22        double derivata_fel = (fel - L_Fel);
23
24        Output = Kp * fel + Ki * Fel_Integral + Kd * derivata_fel;
25
26        L_Fel = Fel;
27        L_Timer = aktuell_tid;
28     }
29  }
30
31  void Set_parametrar(double Kp, double Ki, double Kd)
32  {
33    double Samp_tid = ((double)Sampling_tid)/1000;
34     kp = Kp;
35     ki = Ki * Samp_tid;
36     kd = Kd / Samp_tid;
37  }
38
39
```

Figur 13

Denna algoritm svarar mot de specifikationer som givits för projektet och som motsvarar den tidigare regulatorns funktion. Denna algoritm behöver dock kompletteras med fler funktioner för att fungera optimalt. Dessa funktioner är inställning av samplingstid, anti integrationsuppvridning och att förhindra att derivata delen orsakar plötsliga toppar i utsignalen, på engelska så kallade derivate spikes. För att kunna kontrollera hur ofta som regulatorn beräknar styrsignalen behöver man styra hur ofta rutinen körs, ange samplingstiden. Ett annat problem som bör åtgärdas har koppling till att styrsignalen bara kan arbeta inom ett visst begränsat intervall, 0-10V. Det betyder att signalen ej kan sänkas under 0V och ej över 10V. Då måste man förhindra att integraldelen växer eller sjunker när styrsignalen sina ändlägen. Denna funktion kallas anti windup. I uppstart skedet av projektet studerades också ett flertal andra olika lösningar där färdiga bibliotek finns att använda. Dessa bibliotek har dessa ovanstående funktioner och även andra avancerade funktioner i syfte att skapa en effektiv kontrollalgoritm. Istället för att lägga tid på att skapa en algoritm från början beslutades det att använda ett sådant färdigt bibliotek. Det bibliotek som användes heter pid.h. Parametrarna K, Ti och Td som är inställda med hjälp av pulsgivarna användes sedan i som parametrar i ett funktionsanrop till detta bibliotek.

5.2.5 Omvandlig av spänningsnivåer

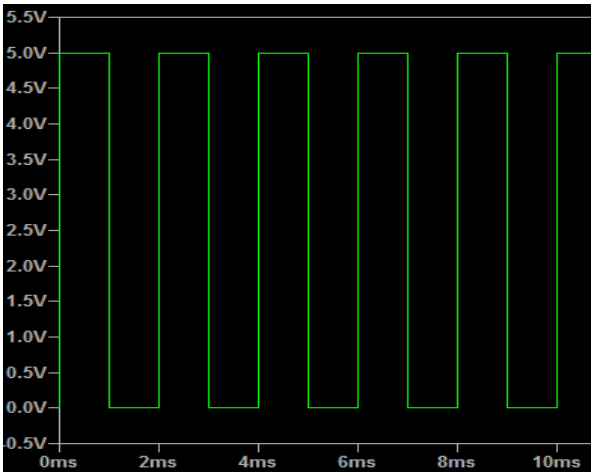
De nivågivare och pumpar som nu används på aktuell regulator arbetar i intevallet 0-10 volt. Mikrokontrollerkortet är begränsad till att använda 0-5 volt. För att styra och ta emot data från pumpar och nivågivare är det nödvändigt att göra en konvertering från intervallet 0-10V till 0-5V i fallet med nivågivaren och motsatt när det gäller pumparna. Då kontrollerkortet inte har någon D/A omvandlare styrs pumpar med hjälp av en av dess PWM utgångar. Konstruktionen för att omvandla en 0-5V PWM signal till en jämn likspänningssignal i intervallet 0-10V bygger på operationsförstärkaren LM358. LM358 är en operationsförstärkare med två förstärkare i samma chip. Kretsen som här presenteras använder sig av en spänningsföljare och en förstärkare.



Figur 14. Visar kopplingsschema för spänningsomvandlare.

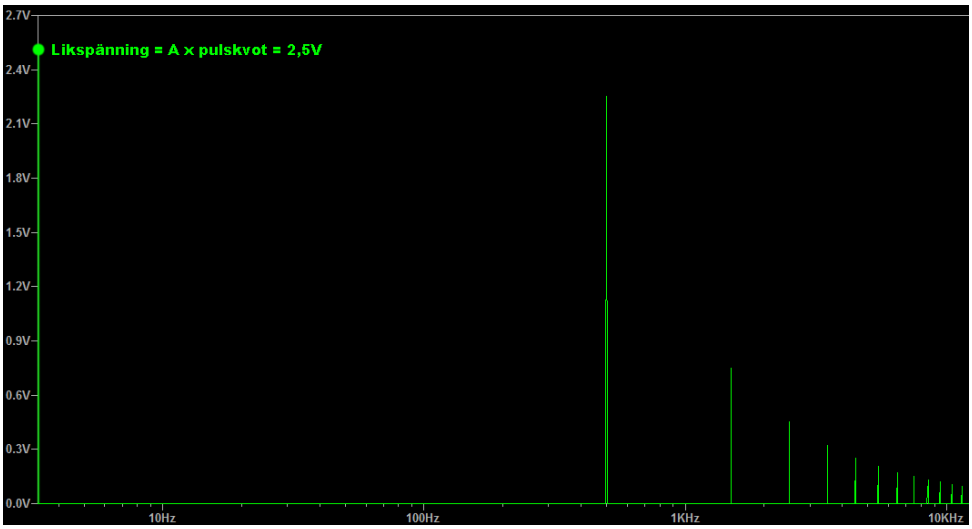
Anledningen till att använda en spänningsföljare är att operationsförstärkaren har en hög in-impedans vilket betyder att minimalt med ström dras från porten på kontrollerkortet. Den maximala strömstyrkan från en port på Arduino är 40mA, men man bör enligt datablad ej ligger över 30mA under någon längre period [7]. Spänningsföljaren agerar således som en slags buffert, säkerhet.

PWM signalen från Arduino ser ut enligt figur 3. Frekvensen för signalen är ca 500hz (Se Kapitel 2)



Figur 15. Visar PWM signal med en pulskvot på 50%

I detta exempel är pulskvoten (eng. duty cycle) 50%. Med hjälp av en fouriertransform gjord på aktuell PWM signal visas signalens amplitudspektrum i figur 16.

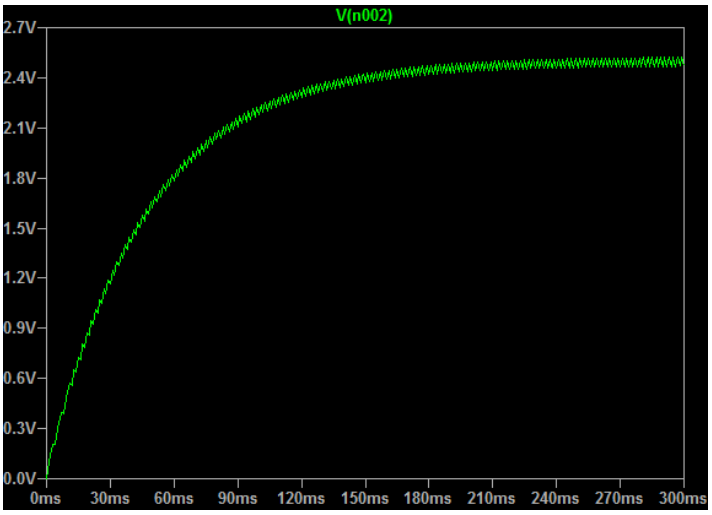


Figur 16. Visar PWM signalens amplitudspektrum.

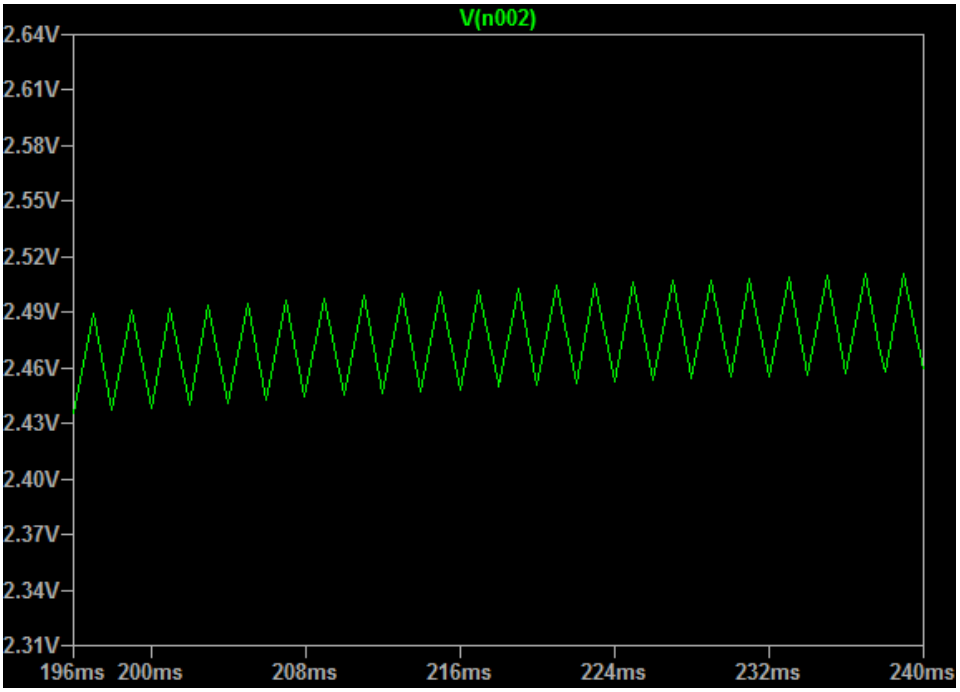
Den visar, som väntat, att när frekvensen är noll är signalens överlagrade likspänning 2.5V. Det är denna spänning vi är intresserade av. Vi ser även att vi har en signalkomponent vid PWM signalens frekvens 500Hz och sedan avtagande signalamplituder som är ligger på udda multiplar av denna frekvens. Med hjälp av ett lågpassfilter behåller vi den sökta likspänningen och dämpar de högre frekvenserna. För att räkna ut lämpliga värden på R och C användes följande formel.

$$f_g = \frac{1}{2\pi RC}$$

Signalen efter lågpassfiltret visas i figur 17 och 18. Här kan vi se att vi får en likspänning med en acceptabel variation (eng. ripple) och att detta värde nås efter våra förhållande godtagbar tid. Signalen förstärks sedan med en förstärkning på 2.



Figur 17.

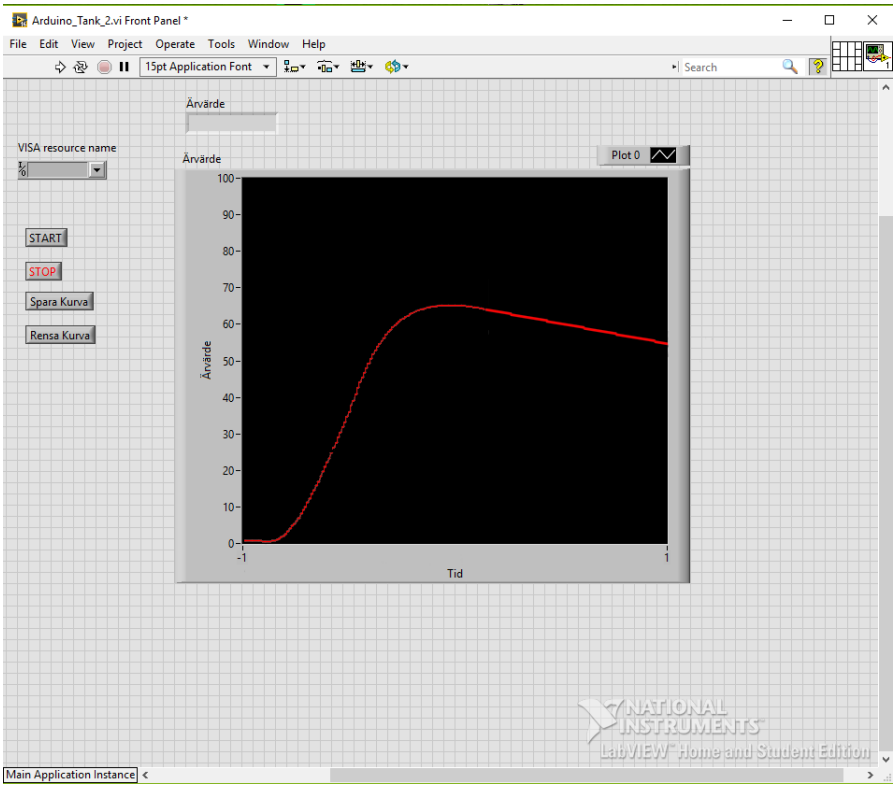


Figur 18. Visar variation (eng. ripple) i utsignalen

5.2.6 Uppkoppling mot PC

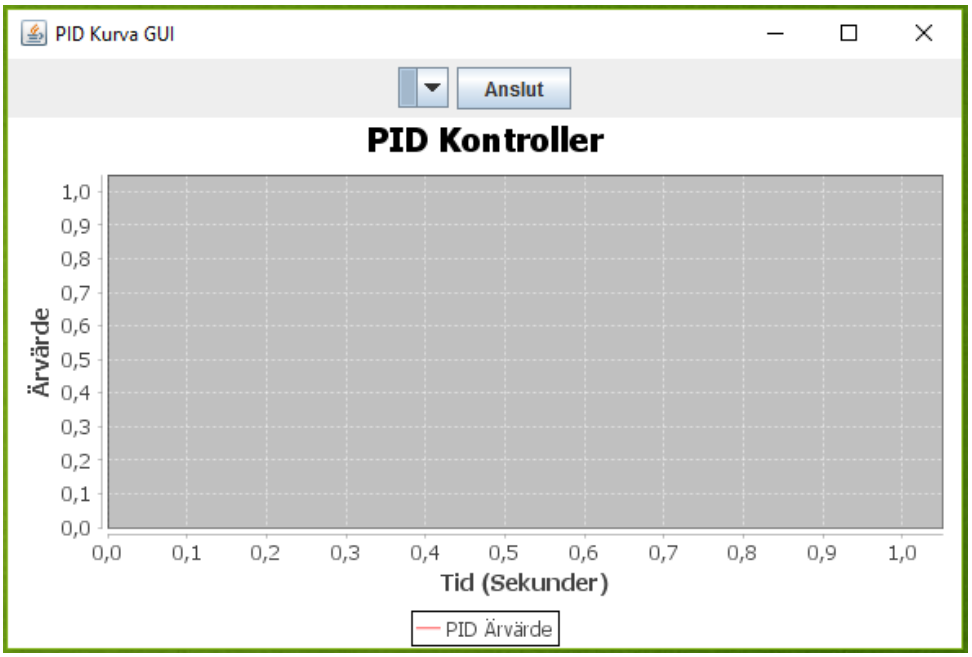
Ett antal olika alternativ undersöktes för att presentera kurvor från regulatorn på en ansluten PC. Alternativen som undersöktes var Matlab, Labview, Processing [], Serial Plott [] och utveckling av egen programvara i Java. Anslutning till PC saker via USB-kabel till Arduinokort.

LabView visade sig vara ett bra alternativ då man har tillgång till avancerad signalbehandling och möjlighet till att spara kurvor och värden för senare bruk. Figur 19 visar ett interface uppbyggt i LabView där ärvärdet från processen ritas upp som en kurva.



Figur 19. Visar LabView inteface.

Egen programvara utvecklades i form av ett javascript som ritar upp en graf utifrån processens utsignal. Se figur 20

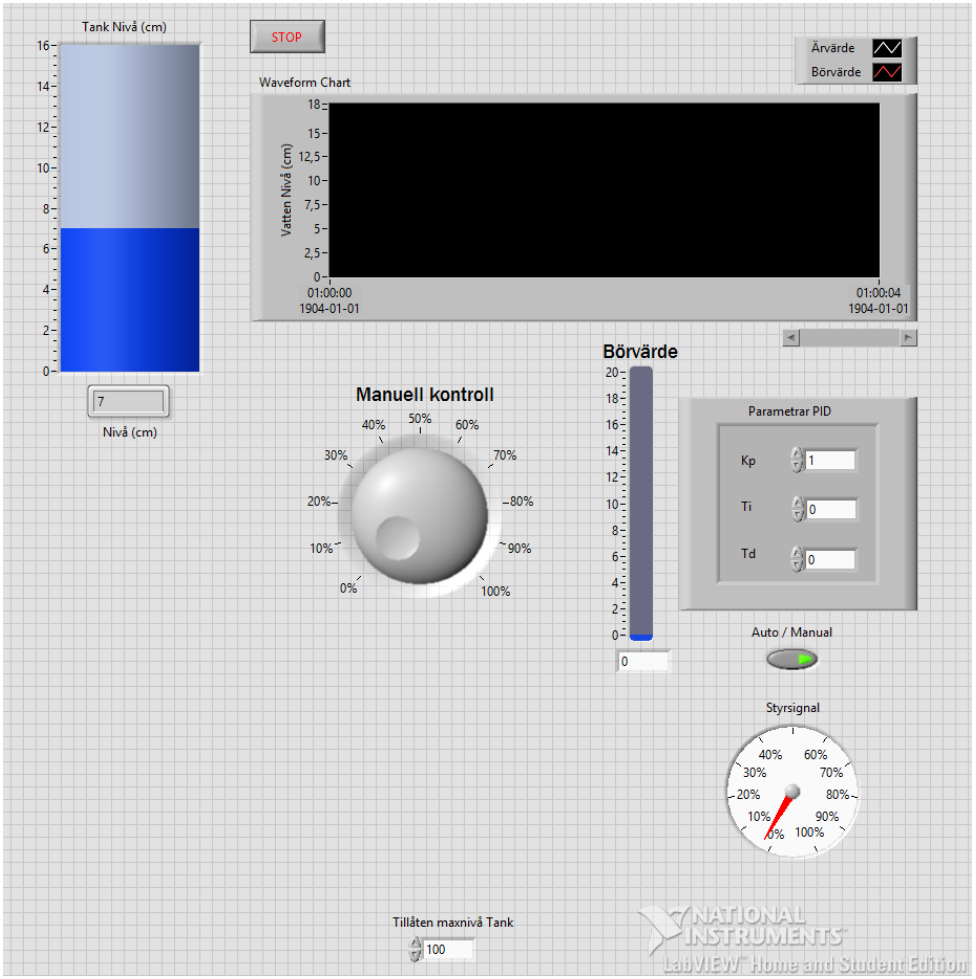


Figur 20. Visar javascript som plottar utsignal från tankprocess

Serial Plott är en gratis programvara för att läsa av serialporten, den port som mikrokontrollerkortet är ansluten till och plotta dess värde som en kurva. Det är ett enkelt program men fungerar bra för våra syften.

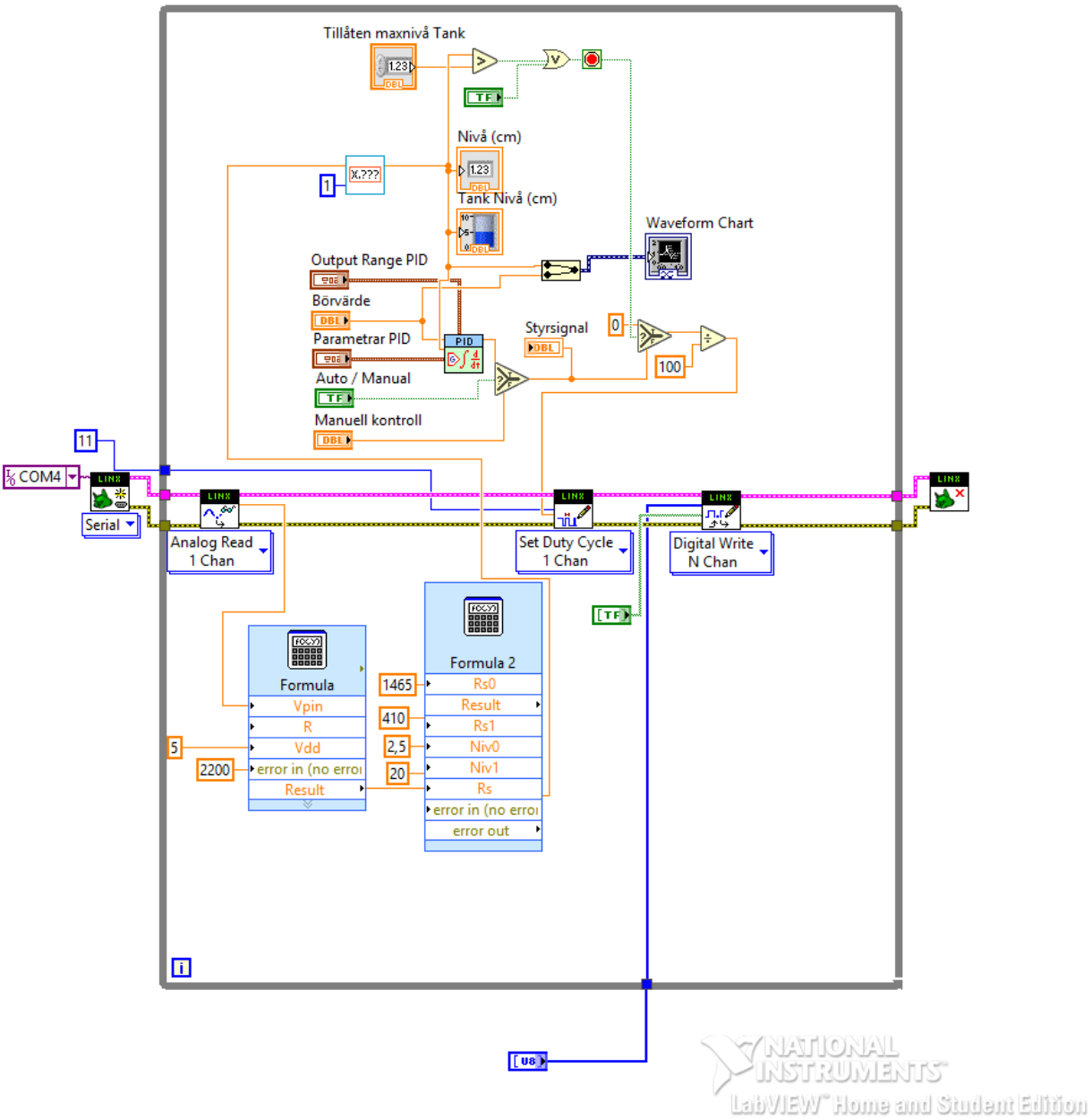
5.2.7 LabView PID Regulator

Labview är en utvecklingsmiljö för ett grafiskt programmeringsspråk som är utvecklat av National Instruments. Det kan användas för lösa olika programmeringsuppgifter inom flera olika områden. Områden som LabView är speciellt bra lämpat för är datainsamling, dataanalys, datalagring och styrning av GPIB instrument. LabView är utformat för att enkelt kunna kopplas ihop med en mängd olika hårdvaruplattformar som till exempel mikrokontrollerkortet Arduino. I LabView bygger man virtuella instrument, instrument som består av en frontpanel och en bakpanel. Frontpanelen är det som användaren ser och det är här man placerar knappar, visare och grafer etc. Bakpanelen innehåller den kod som styr instrumentets uppförande. För att kunna koppla ihop Arduino med LabView användes LINX VI, där VI står för Virtuellt Instrument. Via LINX VI kan man styra mikrokontrollerkortets alla in och utgångar. En frontpanel skapades som svarar mot de specifikationer som anges för projektet med avseende på inställningsmöjligheter för regulatorn och för display av valda värden. Se Figur 21 och 22 på följande sidor.



Figur 21. Visar frontpanel av PID regulator

Figur 21 visar den frontpanel som är konstruerad i Labview. Den består av en virtuell tank där vattennivån är synlig. Manuell kontroll av pump finns samt inställningar för börvärde och aktuella parametrar så som Kp, Ki, Kd. Möjlighet att välja om regulatorn skall köras per automatik eller i manuellt samt en indikation på hur styrsignalen varierar. Högst upp på panel finns en graf som skriver ut aktuellt börvärde samt ärvärde.



Figur 22. Visar kod för aktuell PID Regulator

Figur 22 visar kod för PID-regulatorn. Består av fem block (LINX) som hanterar kommunikationen med mikrokontrollerkortet Arduino. Blocken från vänster till höger är Serial Read, Analog Read, Set Duty Cycle, Digital Write samt Close Serial. En PID VI bygger upp PID algoritmen. Till denna kopplas användarens parametrar och inställningar.

6. Resultat

En PID-Regulator skapades med hjälp av ett Arduino mikrokontroller kort. Pulsgivare, LCD-skärm kombinerades med detta kort och olika sätt för att uppnå en tillfredställande funktion undersöktes. En egenutvecklad mjukvarubaserad PID algoritm utvecklades som efter tester uppträdde på ett tillfredställande sätt, likvärdigt med den tidigare analoga regulatorn. Algoritmen behövdes dock kompletteras med fler funktioner för att göra den optimal. Det viktigaste funktionerna för att få en stabil reglering är möjlighet att kunna bestämma samplingstid, att förhindra integrationsuppvridning och att motverka att så kallade ”Derivation spikes”. Ett färdigt bibliotek, pid_v2.h, som innehöll dessa funktioner användes och befanns vara mycket lämpat för aktuell uppgift. Möjligheten att ansluta denna regulator till en PC undersöktes. Flera olika alternativ togs fram för att visa aktuella värden från regulatorn som kurvor på skärmen, där programvaran Labview visade sig vara det bästa alternativet.

Ytterligare förbättringar gjordes med införandet av en pekskärm, som ersättare för både LCD-skärm och pulsgivare. Till sist utvecklades en PID regulator i LabView, där man kan kombinera kontroll av regulatorn, visning av inställda värden och plotta kurvor av intressanta signaler och variabler i ett och samma virtuella instrument.

7. Slutsats / Analys

Att utveckla en mjukvarubaserad PID algoritm på ett Arduino mikrokontroller kort är förhållande vis enkelt och man uppnår ett bra resultat. Rapporten anger också på flera olika sätt hur detta kan ske, antigen via en egen konstruerad PID-algoritm som följer det allmänna sätet att konstruera en kontrollalgoritm, på färdiga bibliotek samt genom programvaran LabView. Införandet av en pekskärm visar att den mycket väl kan ersätta de fysiska pulsgivarna. Systemet blir mer flexibelt, det är lättare att i framtiden lägga till mer funktioner på pekskärmen än att fysiskt behöva addera fler pulsgivare eller omkopplare till regulatorn. Man undviker även det slitage som förr eller senare uppkommer på fysiska knappar och omkopplare.

Det bästa alternativet är att välja en konfiguration som består av en kombination av ett mikrokontrollerkort och LabView som i vårt fall körs i operativsystem Windows. LabView är mycket väl lämpat för denna uppgift. Här finns möjlighet att i framtiden uppgradera regulatorn med nya kontrollers, att spara värden och plotta kurvor och att analysera signaler. Då kombinationen med LabView inte kräver mer än ett fåtal portar på kontrollerkortet kan man välja ett kort som är avsevärt mycket mindre och till en lägre kostnad. Med ett mindre kort kan man även göra regulatorn fysiskt mindre.

8. Kritisk diskussion och framtida förslag

Arbetet tog ca 12 veckor, där merparten av tiden användes till utvecklandet av programvara. I början av utvecklingen skulle mer tid har lagts på att utvärdera hur befintlig regulator mer i detalj fungerade för att lättare få ett mått på, något att jämföra med, hur den nya digitala regulatorn fungerar. Vi har inga siffror, hårda fakta, att luta oss tillbaka på. Är den nya regulatorn snabbare, mer robust eller noggrannare än sin föregångare? Vi har en subjektiv uppfattning när vi observerat regulatorn att den är likvärdig föregångaren. Att inte vara för fokuserad på de specifikationerna som kom med projektet utan att snabbare ta fram alternativ som i vårt tycke skulle fungera bättre.

I framtiden skulle det vara möjligt att utveckla en applikation till mobiltelefoner för styrning av PID-regulator. En enklare sådan gjordes dock bara med funktionen att styra regulatorn manuellt. Andra funktioner som skulle kunna hanteras av en sådan applikation är inställningar av parametrar, ange manuell eller automatiskt läge samt lagring av data från regulatorn för framtida bruk. Även någon form av simulering av regulatorn i mobilen skulle vara möjligt.

Referenser

[1] Evans, Brian. Beginning Arduino Programming, Apress, 2011

[2] Secret of Arduino PWM [Internet]. Nedladdad 24/2 2017. Hämtad från: https://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM

[3] Penlink Ab, Pulsgivare [Internet]. Nedladdad 25/2 2017. Hämtad från: http://www.penlink.se/sv/produkter/pulsgivare

[4] Reading Rotary Encoders [Internet]. Nedladdad 28/2 2017. Hämtad från: http://playground.arduino.cc/Main/RotaryEncoders

[5] Thomas, Bertil. Modern Reglerteknik, 4 rev. Uppl. Stockholm, Liber AB, 2011

[6] Åström, Karl Johan. Murray, Richard. Feedback Systems: An Introduction for Scientists and Engineers [Internet]. Nedladdad 16/3 2017. Hämtad från: http://www.cds.caltech.edu/~murray/amwiki/index.php/Main_Page

[7] Arduino Mega [Internet]. Nedladdad 10/3’ 2017. Hämtad från: https://www.arduino.cc/en/Main/arduinoBoardMega

[8] Söderström, Fredrik. Nikka, Karl Emil. Hur funkar Arduino, Uppl. 7.01, Malmö, 2016.

Bilaga A: Kod PID Algoritm

```
void loop() {

  unsigned long L_Timer;
  double Insignal, Utsignal, Borvarde;
  double Fel_Integral, L_Fel;
  double Kp, Ki, Kd;
  int Samplings_tid = 1000;
  void PID_Kontroller()
  {
       /*Senaste beräkningen*/
       unsigned long aktuell_tid = millis();
       int Tid_skillnad = (aktuell_tid - L_Timer);
        if(Tid_skillnad>=Samplingstid)
        {
       /*Beräkna fel källor */
       double Fel = Borvarde - Insignal;
       Fel_Integral += Fel;
       double derivata_fel = (Fel - L_Fel);


       /*Beräkna PID utsignal*/
       Utsignal = Kp * Fel + Ki * Fel_Integral + Kd * derivata_fel;

       /*Spara variabler*/
       L_Fel = Fel;
       L_Timer = aktuell_tid;
        }
  }

  void Set_Parametrar(double Kp, double Ki, double Kd)
  {
       kp = Kp;
       ki = Ki;
       kd = Kd;
  }
}

```

Bilaga B: Specifikationer Regulator

Följande krav ställs på den nya digitala regulatorn:

Kontroller på panelen:	K (Förstärkning)	0,5 – 50	i steg om 0,5
	T_I (Integrationstid)	0,5 – 50	i steg om 0,5
	T_D (Deriveringstid)	0,1 – 5	i steg om 0,1
	Börvärde	0 – 100 %	
	Man styrvärde	0 - 100 %	
Omkopplare på panel:	Auto/Manuell		
	Nivågivare1/Nivågivare 2 som ärvärdesgivare		

LCD-Display på panel ska visa:	K	
	T_I	
	T_D	
	Börvärde	0 – 100 %
	Ärvärde	0 – 100 %
	Styrsignal	0 – 100 %

Matningsspänning till PID regulator: +15V

Styrsignal ut: Analog spänning 0 - 10 V
Ärvärde in: Analog spänning 0 - 10 V

Börvärde, ärvärde och styrsignal ska kunna presenteras som kurvor på en ansluten PC.