Reglerdesign för energi- och klimatsystem

Hans Högström

Department of Automatic Control
Lund University
December 2008
Reglerdesign för energi- och klimatsystem (Embedded Controller Design for Energy and Climate Systems)

Abstract
This Master’s Thesis describes the development and implementation of a controller aimed at the climate area business in buildings. Practical design issues for a robust, cheap and yet flexible product is presented. The system should be used in real market applications and the regulator used is equipped with PID-function and an autotuner for easy adjustment. The result can be used for tying heat and cold engines together as well as alarm signals for fault detection. The purpose is also to serve as a platform for further development of control algorithms and surveillance in buildings.
Förord

Under projektets gång har flera personer och företag bidragit med värdefulla synpunkter och idéer och för detta är jag djupt tacksam. Jag vill också tacka Tore Hägglund (examinator) och Anders Robertsson (handledare) som båda genom sitt djupa kunnande satt sin prägel på arbetet.
Innehållsförteckning

1. Inledning ....................................................................................................................... 1
   1.1 Bakgrund ............................................................................................................. 1
   1.2 Målsättning ......................................................................................................... 1
2. Hårdvara ................................................................................................................... 2
   2.1 AVR Mega16 från ATMEL ............................................................ 3
   2.2 AVR-Dragon ................................................................. 3
   2.3 Transformator ................................................................................................. 4
   2.3 Diobrygga ......................................................................................................... 4
   2.4 Stabiliserad +5V spänningskälla ......................................................................... 4
   2.5 Reläer ............................................................................................................... 5
   2.6 Frihjulsdiod ....................................................................................................... 5
   2.7 Givare ............................................................................................................. 5
   2.8 Realdidsklocka .................................................................................................. 5
   2.9 Extern EEPROM ............................................................................................... 6
   2.10 Serieport .......................................................................................................... 6
3. Mjukvara .................................................................................................................. 7
   3.1 ICCAVR från Imagecraft ................................................................................... 7
   3.2 AVR Studio från ATMEL .................................................................................. 7
   3.3 Programvara för datorkommunikation ............................................................... 7
   3.4 Kommunikationsmodul för matlab ................................................................... 7
   3.5 Programvara för extern styrning ......................................................................... 8
4. Utveckling och design ............................................................................................. 9
   4.1 Prototyptest ....................................................................................................... 9
   4.2 Förändring av givargränssnitt ......................................................................... 10
   4.3 Serie till USB-port adapter ............................................................................... 11
5. PID-Algoritmen ....................................................................................................... 12
   5.1 Digital implementering .................................................................................... 12
6. Autotuner för PID-reglering .................................................................................. 14
   6.1 Relämetoden .................................................................................................... 14
   6.2 Beskrivande funktion ....................................................................................... 15
   6.3 Val av PID-parametrar .................................................................................... 16
   6.4 Tuningsteg ....................................................................................................... 17
   6.5 Simulering ....................................................................................................... 18
   6.6 Tuning av brusstörd signal ............................................................................. 20
   6.7 Exempel på tuning ........................................................................................... 22
   6.8 Efterjustering ................................................................................................... 23
7. Resultat ................................................................................................................... 25
   7.1 Slutsatser och kommentarer ........................................................................... 25
   7.2 Framtida utveckling ........................................................................................... 26
8. Referenser .............................................................................................................. 27
9. Appendix ............................................................................................................... 28
   9.1 Kommunikationsmodul för matlab ............................................................... 28
   9.2 SSL .................................................................................................................. 29
   9.3 I2C-bussen ...................................................................................................... 29
Figurförteckning

Figur 1 Översiktsbild huvudkort ................................................................. 2
Figur 2 Översiktsbild strömförsörjning ............................................................ 2
Figur 3 AVR-Dragon .................................................................................. 3
Figur 4 AVR Connector .............................................................................. 8
Figur 5 Prototyp 1 ...................................................................................... 9
Figur 6 3-signalmätning med UTI ................................................................. 10
Figur 7 Reläfunktionens placering i reglerslingan ....................................... 14
Figur 8 Nyqvistdiagram ............................................................................ 15
Figur 9 Simulatorbox KI-101 .................................................................... 18
Figur 10 AVR med interface för KI-101 ....................................................... 19
Figur 11 Jämförelse av effekterna utan resp. med hysteres vid reläexperiment 20
Figur 12 Förberedelse för tuning ................................................................. 21
Figur 13 Auto-tuning av ett 4:e ordningens system .................................... 22
Figur 14 Jämförelse mellan Ziegler-Nichols och AMIGO-metoden för PID-inställning 23
Figur 15 Inställningarna Fast, Medium och Slow ........................................ 24
1. Inledning

1.1 Bakgrund


Här finns utrymme för mindre skräddarsydda system som inte dras med den höga initialkostnaden som ofta krävs för att lösa ovanstående uppgifter.

1.2 Målsättning

- Ta fram och presentera en regulator som också kan övervakas, styras och ställas in via extern dator.
- Implementera en robust autotuner för PID-reglering.
- Undersök de problem och möjligheter som uppstår med den framtagna lösningen.
- Ge förslag på fortsatt utveckling av projektet.
2. Hårdvara

Avsnittet beskriver de viktigaste grundläggande byggenomarna samt motivering av komponentval. Strömförsörjningsenheten har ett eget kretskort vilket underlättar vid utveckling och felsökning. Ett exempel på hur dessa kort kan se ut visas i Figur 1 och Figur 2.

Figur 1 Översiktsbild huvudkort
Bilden visar hur ett kretskort uppbryggd kring en AVR mikroprocessor kan se ut. Kortet har fyra stycken reläutgångar och lika många ingångar avsett för PT-100 givare.

Figur 2 Översiktsbild strömförsörjning
Transformatorn syns här med glassäkringar vilket är ett bland flera alternativ vid avsäkring. Notera också att den tar halva kortet i anspråk vilket gör den till den mest utrymmeskrävande komponenten.
2.1 AVR Mega16 från ATMEL
Det finns många tillverkare mikrokontrollers som skulle kunna utföra samma uppgifter men valet föll på en AVR från ATMEL [6].

Det finns tre typer av AVR:
1. TinyAVR
2. AVR Classic
3. MegaAVR

Kärnan är samma men de olika varianterna skiljer sig åt med antal register, typ av minne och huruvida den försetts med AD-omvandlare, UART m.m.

Ett lämpligt val i detta fall är en Mega16 som har tillräckligt med In/utgångar för att hantera display minne, kommunikation, aktuator och sensor. Den har också en 10 bits AD-omvandlare inbyggd vilket är bra då dessa kostar relativt mycket som lös krets.

2.2 AVR-Dragon

Figur 3 AVR-Dragon
Notera den färgade bandkabeln som är ansluten till kortets JTAG-gränssnitt samt USB-kabeln till höger som används för kommunikationen mot AVR-studio, en utvecklingsmiljö på värd-PC.

2.3 Transformator

Vid dimensionering av transformator får maximalt strömbehov samt en säkerhetsmarginal avgöra vilken som är lämplig. Detta bör utformas så att det totala behovet för kretsarna är mindre än säkringarnas strömmärkning som i sin tur är mindre än maximal ström som transformatorn förmår leverera.

Någon form av överbelastningsskydd måste alltid finnas i form av PTC-motstånd eller säkringar. Avseddes en transformator med en sekundärlindning så kan skyddet sättas på primärsidan och i annat fall på sekundärsidan. Det går också bra att för säkerhets skull avsäkra båda sidorna och används gläsäkringar så bör dessa vara tröga. En överhettning i transformator med eventuell brand som följd kan bli kostbart så det är bättre med rejäla marginaler.

2.3 Diodbrygga


2.4 Stabiliserad +5V spänningskälla

Transformatorn ger +12V oreglerad spänning ut och mikroprocessorn samt de flesta andra kretsar behöver +5V.

En standard 3-terminal 7805 regulator av den lite kraftigare sorten som klarar strömmar upp till 1.5A och med TO-220 kapsel är lämplig för vår konstruktion. Så länge strömmen håller sig under 1 ampere så klarar kapseln själv att kyla kretsen annars får en extra kylfläns hjälpa till. Detta beror givetvis också på miljön där kretsen skall verka och även om de flesta kretsar både är kortslutningsskyddade och försetts med termiskt överbelastningsskydd så skadar det inte att redan i början montera den ihop med en kylfläns.

För att få en stabil spänning och förbättra LF och HF egenskaperna så används en stor elektrolytkondensator ihop med en mindre polypropyleen på både in och utgång hos spänningsstabilisator 7805. Att ha en stabil spänningskälla som kan leverera tillräcklig ström med lägt brus och lite "ripple" är nödvändigt för att inte mikroprocessorn skall hamna i ett obestämt läge med ev. reset som följd. Om +5V används som spänningsreferens vid t.ex. AD-omvandling så ger en ostabil spänning onoggrannhet i mätningen.

In/utgångar ändrar status vid klockflankerna och orsakar strömvariationer och ihop med variation i övriga kretsar så behövs dessa kondensatorer för att motverka spikar samt att fungera som en reservoar för energi t.ex. till och från reläerna, ESD avkoppling mm.

Vid dimensionering så säger tumregeln att kondensatorerna minst ska klara dubbla spänningen som används. För att ha ordentlig marginal och likartade val så används kapacitanser gjorda för 63V vilket ger ordentlig marginal eftersom transformatorn ger +12V ut (belastad).
2.5 Reläer
Här används 1-poliga växlande reläer avsett för kretskortsmontering med ordentlig marginal till tänkt maxström.

Arbetsspänning på +12V kan tas direkt ifrån den oreglerade spänningen från transformatorn. Här kunde också +5V reläer ha använts men då hade strömmen blivit högre. Strömkonsumtionen ligger på ca 50mA per relä och det gör komponenten till den mest strömkrävande i kretsen.

Detta påverkar i hög grad vilken transformator som är möjlig att använda och ett val med låg effekt gör det lättare i ett senare skede då storleken på kretskortet skall optimeras. Komponenten får gärna ha genomskinligt hölje och monteras med fördel ihop med en lysdiod för att underlätta ev. felsökning.

2.6 Frihjulsdiod
När spänningen över reläet bryts så kollapsar det magnetfält som byggts upp och fått reläet att dra. Detta fält ger då upphov till en hög spänning i motsatt riktning som kan skada drivkretsen, i det här fallet transistorn.

Som skydd används en backförspänd s.k. frihjulsdiod som kortsluter den uppkomna spänningen till positiv spänningskälla.

2.7 Givare
Eftersom utrustningen främst är tänkt att användas för värmereglering av fastigheter så blir kravet på givaren att den skall kunna visa temperatur på en tiondels grad när. Den bör också omfatta området -40 till +100°C för att vara flexibel nog att kunna användas i de flesta tillämpningarna.

En standard PT-100 givare användes vid utvecklingen och uppfyller dessa krav.

2.8 Realtidsklocka
För tidsstämpling av driftdata samt tidsstyrda reglering och kontroll behövs en realtidsklocka. Vid användning i fastigheter vill man ibland t.ex. använda nattsänkning eller någon form av helg och årstidsbunden styrning.

En extern 32,768 kHz kristall av samma typ som sitter i vanliga armbandsur går att använda och kopplades in på en av räknaringångarna hos mikroprocessorn. Genom att specificera ett lämpligt intervall så att räknaren genererar ett avbrott varje sekund och uppdaterar en struktur innehållande datum och tid så har processorn ständigt tillgång till uppdaterade värden.

2.9 Externt EEPROM


EEPROM:en är billiga och finns med ett flertal kommunikationsgränssnitt. I första prototypen sitter det ett sådant för lagring av driftdata. Minnet använder I2C-bussen och upptar bara två portar från mikroprocessorn samt stöder kommunikation på upp till 400kHz. Storlekar finns lättillgängligt upp till ca 1 Mbit och räcker mer än väl för loggningsändamål.

Se Appendix 9.3 I2C-bussen för mer information om I2C-bussen.

2.10 Serieport


Det är givetvis också möjligt att göra ett eget interface om man inte som nu har tillgång till en inbyggd UART. Det kräver ändå någon form av hårdvara för signalomvandling samt en mjukvarulösning för synkronisering av kommunikationen.


Lösningen med serieport är robust och fullt tillräcklig för kommunikation mellan AVR/dator inom det tänkta användningsområdet.

Anm. MAX202CPE är en uppgradering av den populära MAX232 kretsen.
3. Mjukvara

I detta kapitel följer en beskrivning av de program som behövs för programmering och utveckling. Här ingår både färdiga och egenutvecklade programvaror som under arbetes gång har visat sig nödvändiga.

3.1 ICCAVR från Imagecraft


Open Source alternativen WINAVR som använder den välrenommerade GNU GCC kompilatorn hade också kunnat användas om man inte vill köpa kompilator.

3.2 AVR Studio från ATMEL

AVR Studio är ett gratis verktyg från ATMEL som används för att skriva och felsöka program och innehåller en simulator och assembler. Den är nödvändig för programmering av AVR-chippet och hantering av AVR-Dragon.

3.3 Programvara för datorkommunikation

För kommunikationen mellan mikroprocessorn och dator så gjordes ett program i Delphi 2005. Tanken är att detta skall underlätta för tekniker som behöver göra inställningar, felsöka, göra funktionskontroll samt tömma och spara EEPROM minne om sådant används.

Ett annat användningsområde är att öppna upp möjlighet att via programmet göra detta via webben. Kommunikationsmodulerna är då kompilerade som serviceapplikation eller DLL utan grafiskt gränssnitt.

3.4 Kommunikationsmodul för matlab


For utförligare beskrivning see Appendix 9.1 Kommunikationsmodul för matlab.
3.5 Programvara för extern styrning

Vid inställning och övervakning av drift underlättar det om användaren har möjlighet till någon form av grafisk överblick istället för enbart siffervärden. Programmet i Figur 4 visar grafer från mikroprocessorn samt ger möjlighet att ställa in regulatorn, utföra auto-tuning mm.

Kommunikationsmodulen i avsnitt 3.4 utnyttjas för att ansluta direkt mot AVR:en. Möjlighet att ansluta över nätet finns också om man angett att programmet som är direktanslutet ska fungera som en server. Därefter går det att ansluta utifrån och fjärrstyra mikroprocessorn. Kommunikationen sker då via SSL (Secure Socket Layer) och klienten har samma funktionalitet som serverprogrammet. För mer om SSL se Appendix 9.2 SSL.

Figur 4 AVR Connector
4. Utveckling och design

Produktutveckling är en iterativ process av möten, informationssökning, tester och förändringar. Avsnittet beskriver de förändringar och tillägg som oundvikligen uppstår då projektet forskrider.

4.1 Prototyptest

Alla delarna är implementerade och testade både hård och mjukvarumässigt. Några viktiga detaljer av speciell vikt från första prototyptest är funktion, minnesbeläggning, fysisk storlek på kretskortet och nytan av de olika delarna. Figur 5 visar första testkortet.

Mikroprocessorns minne fylls snabbt och det gäller speciellt om tunga flyttalsberäkningar används. Det syntes tydligt när tabellen för linjärisering av PT-100 givaren implementerades som formel istället för minnestabell. Rådet här är att i möjligaste mån undvika flyttal eller om möjligt göra decimaltransformering för att därefter räkna vidare i heltalsområdet.

Figur 5 Prototyp 1
Överst i bild är en inkopplad PT-100 givare, på bilden saknas knappssatsen för navigering i LCD menyn. Nätdelen till vänster i bilden är separerad från resten av komponenterna och matas med 230V.
4.2 Förändring av givargränssnitt


Strömkonsumtionen är låg, högst 2.5mA och upplösningen ligger på 13-14 bitar vilket är högre än mikroprocessorns egen AD-omvandlare. Det är också möjligt att använda olika typer av givare och flera per UTI krets t.ex. kapacitiva, PT100, PT1000, termistorer, resistiva bryggor, potentiometer eller kombinationer av dessa. Det går också i de flesta fallet att använda 2, 3 och fyrrådsmätning om det visar sig behövas.


Mätningen är en s.k. 3-signal mätning som bygger på referensmätning av ett element av samma typ som givaren. Kompenserings av okänd offset och förstärkning med 3-variabel teknik går till på följande vis:

Lineär överföringsfunktion:

\[ M_i = kE_i + M_{off} \]

3 signaler skickas in i systemet:

![Referenselement](image)

Genom att jämföra mätvärden från givaren med ett känt referenselement och en intern noll punkts mätning går det att kompensera för offset och förstärkningsfel.
\[ E_1 = 0 \]
\[ E_2 = E_{\text{ref}} \]
\[ E_3 = E_x \]

Signaler ut blir då:
\[ M_1 = M_{\text{off}} \]
\[ M_2 = M_{\text{ref}} = kE_{\text{ref}} + M_{\text{off}} \]
\[ M_3 = M_x = kE_x + M_{\text{off}} \]

\[ M = \frac{(M_x - M_1)}{(M_2 - M_1)} = \frac{(M_x - M_{\text{off}})}{(M_{\text{ref}} - M_{\text{off}})} = \frac{E_x}{E_{\text{ref}}} \]

Där \( M_1 \) är en nollpunktsmätning, \( M_2 \) en mätning av referenselement och \( M_3 \) är en mätning av givaren.

Eftersom \( E_{\text{ref}} \) är känd så är också \( E_x \) känd samt oberoende av förstärkning och offset.

### 4.3 Serie till USB-port adapter


Efter test av många olika fabrikat visade sig att Belkin hade en modell som fungerade. Alternativet vore att byta ut serieporten mot en USB-port eller konstruera en egen adapter. Båda alternativen hade emellertid fördyrat konstruktionen och tagit för mycket tid i anspråk.
5. PID-Algoritmen

Temperaturreglering av fastigheter brukar vanligen vara en relativt långsamt process eftersom det tar tid för värmen att sprida sig jämte i lokalerna. Dessutom innehåller klimatskalet ofta en avsevärda mängd energi som hjälper till att jämma ut effekter från väder och andra tillfälliga "störningar". Om störningarna är mätbara i god tid innan de slår igenom på temperaturen i fastigheten kan man med framkoppling kompensera bort dessa. Oftast görs det med en yttertemperaturgivare då man vill att innomhustemperaturen inte skall påverkas av hur temperaturen utomhus varierar.

En mycket vanlig inställning för PID-regulatorn är att bara använda PI-delarna som också kan ha förjusterade parametrar från leverantören. Detta går bra om man vet vilken utrustning och system som ska regleras. D-delen skulle i många fall kunna förbättra responsen men då tar det längre tid att trimma in regulatorn och regleringen blir mer bruskänslig.

För att underlätta driftsättningen av regulatorn kan en autotuning-funktion användas vid inställning av PID-parametrarna, se Kapitel 6. Autotuner för PID-reglering

5.1 Digital implementering

Signalen från PID-regulatorn kan skrivas som:

\[
 u = K \left( e + \frac{1}{T_i} \int e(t) dt + T_d \frac{de}{dt} \right)
\]

\[
e = r(t) - y(t) = \text{bovärd-ärvärde}
\]

\[
 K = \text{Förstärkning}
\]

\[
 T_i = \text{Integraltid}
\]

\[
 T_d = \text{Derivat蒂d}
\]

Den diskreta versionen av PID-algoritmen ser ut som följer:

P-del: \[ P(kh) = K(r(kh) - y(kh)) \]

I-del: \[ I(kh + h) = I(kh) + K \frac{h}{T_i} e(kh) \]

D-del: \[ D(kh) = \frac{T_d}{T_d + Nh} D(kh - h) + K \frac{T_d N}{T_d + Nh} (y(kh - h) - y(kh)) \]

\[\rightarrow u(kh) = P(kh) + I(kh) + D(kh)\]

där h är samplingsintervallt och N är derivatans maximala förstärkning.
D-delen använder en bakåtdifferens på mätsignalen istället för felet vilket gör att kraftiga "spikar" undviks om börvärdet skulle ändras hastigt.

Integratorn måste skyddas från ”windup”. Det kan göras på många olika sätt, se t.ex. [5]. Metoden som används begränsar storleken på I-delen både uppåt och neråt och integrationen stannar upp när dessa gränser nås.
6. Autotuner för PID-reglering

De flesta regulatorer justeras idag manuellt. Ett iterationsförfarande som både tar tid och kan orsaka ett visst huvudbry når det gäller att hitta lämpliga värden för PID-parametrarna. Dessutom är det inte ovanligt att inställningarna är långt ifrån optimala. Vid reglering av långsamma system får man dessutom vända länge innan effekterna av justeringen syns i mätsignalen.


6.1 Relämetoden

Genom att koppla in en reläfunktion istället för PID-regulatorn enligt Figur 7 kan man få processen att börja svänga. Denna svängning ger ett mått på processens kritiska förstärkning och periodtid som i sin tur kan användas för att välja lämpliga PID-parametrar, se avsnitt 6.3 Val av PID-parametrar samt [4].

Självsvängningen inträffar vid skärningspunkten mellan processens Nyqvistkurva och inversen av reläets beskrivande funktion, se avsnitt 6.2 Beskrivande funktion.

En stor fördel med relämetoden är att amplituden på självsvängningen kan kontrolleras. Att inducera kraftiga oscillationer i känsliga system som kanske också är i drift är i många fall uteslutet. Målet är givetvis att inte störa systemet mer än nödvändigt och den undre gränsen utgörs av signal/brusförhållandet i kretsen.

Figur 7 Reläfunktionens placering i reglerslingan. Vid tuning så kopplas reläfunktionen in via switchen för att erhålla en kontrollerad självsvängning som kan användas för att beräkna PID-parametrar.

Metoden har visat sig fungera tillfredsställande på många ”normala” processer men metoden är approximativ och det finns ingen garanti att den fungerar på alla system. Tillfälliga brustoppar och system där Nyqvistkurvan skär reläets beskrivande funktion på flera ställen kan orsaka problem. Det sistnämnda indikerar att ett flertal möjliga självsvängningar kan uppstå.
Om reläet inte har någon hysteres inlagd kommer skärningspunkten att ligga på realaxeln i det komplexa talplanet och med hysteres så kommer skärningen att förskjutas in i tredje kvadranten och bli komplexvärd, se Figur 8 nedan.

![Nyquistdiagram](image)

**Figur 8 Nyquistdiagram**  
Visar hur skärningspunkten mellan systemet $G(j\omega)$ och $-1/N(a)$ påverkas av hysteresen, där $N(a)$ är reläets beskrivande funktion. Om hysteresen ökar så kommer skärningspunkten att flyttas neråt i figuren.

### 6.2 Beskrivande funktion

För att kunna räkna fram kritisk förstärkning och periodtid $T_c$ behövs en matematisk beskrivning av reläfunktionen som till sin natur är olinjär.

Om insignalerna till reläet matas med en sinussignal så kommer utsignalen också vara en periodisk funktion som varierar i samma takt. Denna signal Fourierutvecklas med förbehandlet att bara grundtonen tas med eftersom efterföljande system antas ha utpräglade lågpassegenskaper och dämpa ut högre frekvenser. Detta ger tillsammans med systemet en funktion som beskriver hur reläet påverkar utsignalen och kan ses som en amplitudberoende förstärkning, se [2], [3].
Räkningarna som med hjälp av den beskrivande funktionen ger kritisk förstärkning $K_c$ ser ut på följande sätt:

Om produkten av överförings-funktionen $G(j\omega)$ och reläets beskrivande funktion $N(a)$ är lika med -1 så finns risk för självsvängning.

\[ G(j\omega)N(a) = -1 \]
\[ K_c = \frac{1}{|G(j\omega)|} = |N(a)| \]
\[ -\frac{1}{N(a)} = -\frac{\pi}{4d} \sqrt{a^2 - \epsilon^2} - i \frac{\pi \epsilon}{4d} \]

där $\epsilon$ är hysteresbredd, $d$ reläamplitud och $a$ är amplitude på självsvängningen.

\[ K_c = \left| \frac{-4d}{\pi \sqrt{a^2 - \epsilon^2} + i \pi \epsilon} \right| \rightarrow \]
\[ K_c = \frac{4d}{\pi a} \]

6.3 Val av PID-parametrar

När mätningen av självsvängningens amplitude och periodtid $T_c$ är klar kan man använda någon av inställningsmetoderna som bygger på självsvängningsexperiment t.ex. Ziegler-Nichols, se Tabell 1 och [1]. Detta ger i många fall tillfredsställande inställningar för regulatorn men givetvis kan man finjustera värdena manuellt om man inte är nöjd.

<table>
<thead>
<tr>
<th>Regulator</th>
<th>K</th>
<th>Ti</th>
<th>Td</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.5 $K_c$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>$0.4 K_c$</td>
<td>0.8$T_c$</td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>$0.6 K_c$</td>
<td>0.5$T_c$</td>
<td>0.12$T_c$</td>
</tr>
</tbody>
</table>

Tabell 1 Parametervärden enligt Ziegler-Nichols med kritisk förstärkning $K_c$ och kritisk periodtid $T_c$

Ziegler-Nichols metod kan i vissa fall ge olämpliga regulatorinställningar som ger stabilitetsproblem. En mer robust metod kallad AMIGO (Approximate M-constrained Integral Gain Optimization) har tagits fram av Karl Johan Åström och Tore Hägglund, se [5]. Metoden kan användas ihop med självsvängningsexperimentet samt ett stegsvärstest avsett för beräkning av processens statiska förstärkning.
AMIGO-design för PI och PID kan därefter tas fram enligt nedan.

PI-regulator:
\[
K = 0.16K_c \\
Ti = \frac{1}{(1 + 4.5\kappa)T_c}
\]

PID-regulator:
\[
K = (0.3 - 0.1\kappa^4)K_c \\
Ti = \frac{0.6}{(1 + 2\kappa)T_c} \\
Td = \frac{0.15(1 - \kappa)}{1 - 0.95\kappa}T_c
\]

Där förstärkningskvoten \(\kappa = \frac{(1/K_c)}{K_p} = \frac{(1/Kritisk förstärkning)}{Statisk förstärkning}\)

6.4 Tuningsteg

För att använda självinställaren krävs det bara att användaren anger eller accepterar det börvärde som för tillfället är inställt. Detta för att den inte ska bli krångligare att använda jämfört med att ställa in PID-regulatorn manuellt.

Följande steg utförs i samband med att tuning väljs från menyn. Skulle tunern misslyckas av någon anledning så får användaren information om detta och regulatorn fortsätter att jobba med sina gamla värden.

1. Ställ manuellt in börvärde
2. Tuning väljs från menyn
3. Tunern inväntar stabil signal
4. Brusnivån bestäms
5. Hysteres väljs på grundval av brusnivå
6. Amplitudjustering av självsvängning
7. Reläexperiment utförs
8. PID-parametrar räknas ut
9. Användaren får val att sätta de framräknade värdena
10. Åter till PID-mode

Börvärdet i steg ett specificerar också arbetspunkten där reläexperimentet kommer att utföras. Naturligtvis utför man tuningen i det område där regulatorn är tänkt att jobba.

I steg tre så inväntar mikroprocessorn en stabil mätsignal som indikerar att satt börvärde är uppnått och att det bara är brusnivån som ger variation i mätsignalen.
För att inte mätbruset ska orsaka växlingar i reläet används en reläfunktion med hysteres. Storleken på denna sätts efter brusets amplitud i mätsignalen och gör således tuningen mer robust.

När hysteresen väl är satt i steg fem, bestäms utsignalssnivåerna för reläet så att självsvingningen minimeras och samtidigt att mätsignalen blir så stor att den når över hysteresnivån. När detta är gjort sätts själva tuningprocessen igång och amplitud och periodtid för den resulterande självsvingningen mätts.

Om testet gått bra och en stabil självsvingning uppstått så räknar mikroprocessorn fram lämpliga parametrar för PID-kontroll och ger användaren möjlighet att spara och sätta igång regulatorn med de nya parametrarna.

**6.5 Simulering**

Vid utvecklingen av tuning-algoritmen så behövdes referenssystem att testa mot. I detta fall användes en simulatorbox, Figur 9, där systemordning, tröghet, brus, fördröjning m.m. snabbt kan ändras.

![Figur 9 Simulatorbox KI-101](image)

Boxen kan simulera ett flertal olika system och indikerar in/utsignaler via lysdioder i staplar. Det går också att testa kombinationer av två olika system via korskopplingen som går mellan system A i rött och B i blått.

Boxen arbetar med signaler i området 0-5 V och klarar att simulera en mängd olika processer. Eftersom systemen som ställs in är kända så går det också snabbt att verifiera resultaten teoretiskt och även jämföra med simuleringsresultat gjorda i t.ex. Matlab/Simulink.
För att underlätta arbetet med tuning-algoritmen samt för att få ett gränssnitt mot simuleringsboxen så användes kortet som visas i Figur 10. Via serieporten (högst upp till vänster i bild) går det via extern dator att övervaka, skicka styrkoder, erhålla driftdata m.m.

Figur 10 AVR med interface för KI-101
Kretskortet på bilden är avsett att kopplas in till simulatorboxen KI-101 vid utveckling av regler-algoritmer och test på olika system. Kortet är bestyckat med mikroprocessor, strömförsörjning, serikommunikation och diverse in/utgångar samt JTAG programmeringsgränssnitt.
6.6 Tuning av brusstörd signal

Vid mätningar på brusstörda system är hysteres praktiskt för att erhålla stabila mätserier. Detta syns tydligt överst i Figur 11 där reläomslaget är starkt beroende av bruskomponenten i signalen.

I den nedersta plotten har reläfunktionen en hysteres som är brusberoende och reläomslagen blir betydligt jämnare. Storleken på bruset är ca 5 % och har korrekt identifierats vilket leder till att omslagsnivåerna kan justeras så att ”falska” reläomslag undviks.

Figur 11 Jämförelse av effekterna utan resp. med hysteres vid reläexperiment.
Notera fyrkantsvågens ojämna utseende i den övre figuren där brus i mätsignalen orsakat falska reläomslag. Den undre figuren har samma brusnivå (5 %) men med en reläfunktion med hysteres. Reläomslagen sker här mer kontrollerat och lämpar sig bättre för beräkning av kritisk period och förstärkning.

Justering för brus och lämpliga utsignalnivåer som föregår reläexperimentet visas i Figur 12 och utförs på följande sätt.

Metoden med successiva höjningar garanterar inte att nivåerna hamnar rätt på alla system men verkar fungera bra på stabila enkla processer med lågpasskarakter.

Figur 12 Förberedelse för tuning
6.7 Exempel på tuning


I nedersta plottbilden körs regulatorn med de uppdaterade parametrarna samt med en D-del inkopplad (PID). Felet avtar nu betydligt snabbare och regulatorn svänger in sig på ca 20s.

Ziegler-Nichols metod användes vid tuning och parametrarna för regulatorinställningarna ges av tabellen nedan.

<table>
<thead>
<tr>
<th>Tidpunkt</th>
<th>P</th>
<th>I</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Före tuning</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Efter tuning</td>
<td>1.14</td>
<td>4.59</td>
<td>1.10</td>
</tr>
</tbody>
</table>

Tabell 2 Beskriver hur regulatorparametrarna ändras efter en tuning med Ziegler-Nichols.

Figur 13 Auto-tuning av ett 4:e ordningens system

Genom ett enkelt självsvängningsexperiment (den mittersta plotten) kan en bättre uppsättning regulatorparametrar räknas fram. Effekten syns tydligt i insvängningsförloppet mellan den översta plotten där integraltiden är för högt satt och de nya framräknade regulatorinställningarna i den nedersta plotten.
En jämförelse mellan Ziegler-Nichols och AMIGO-metoden för inställning av regulatorparametrarna, Figur 14 visar hur den senare som är mer konservativ ger en mjukare insvängning och snabbare när börvärdet. Att också styrsignalen är mindre oscillativ med AMIGO är givetvis bra ur ett livslängdsperspektiv då detta sliter mindre på utrustningen.

Figur 14 Jämförelse mellan Ziegler-Nichols och AMIGO-metoden för PID-inställning
Notera de betydligt mjukare signalkurvorna i nedersta plotten som genererats med AMIGO. Systemet är av fjärde ordningen och har en fördröjning på 1s. Skillnaden i uppförande med de olika inställningsmetoderna för denna process är markant och visar att Ziegler-Nichols ger en kraftigare förstärkning.

6.8 Efterjustering
Reglerkraven kan variera mellan olika system t.ex. om det är viktigare att undvika överslängar än en snabb insvängning eller att man vill att regulatorn ska agera mer aggressivt på signaländringar. För att underlätta för användaren så finns det tre olika inställningar att välja mellan:

- Fast: ökar förstärkningen med 20 %
- Medium: oförändrad förstärkning
- Slow: minskar förstärkningen med 20 %

Valet kan göras efter auto-tuning och modifierar förstärkningen medan eventuella I eller D-delar behåller sitt värde. Detta ger användaren möjlighet att på ett enkelt sätt styra karaktären på regulatorns uppförande.
Om inte något val görs kommer medium att användas i regulatorn vilket är likvärdigt med o-modifierade inställningar. En jämförelse mellan effekten av de olika valen visas i Figur 15.

Figur 15 Inställningarna Fast, Medium och Slow
Den mellersta plotten visar insvängningsförloppet med efterjustering av förstärkningen med valet ”Medium” vilket innebär att regulatorparametrarna är oförändrade. I den övre figuren med valet ”Fast” har förstärkningen ökats något i jämförelse med normalfallet ”Medium”. Detta visar sig genom en snabbare uppgång mot börvärdet och en kraftigare oversläng. Det sista valet ”Slow” visar sig som en mjukare insvängning och resulterar i en bättre amplitudmarginal. Systemet är av 4:e ordningen och Ziegler-Nichols frekvensmetod har använts för grundinställning av regulator-parametrarna.
7. Resultat

Arbetet har gett värdefull insikt i den hård- och mjukvara som krävs för att få ett fungerande system som är enkelt att handha både för användaren och servicetekniker.

7.1 Slutsatser och kommentarer

Autotunern sparar mycket tid och fungerar utmärkt som hjälpmedel vid inställning av PID-regulatorn. En stor fördel med att använda relämetoden är att man kan begränsa störningen på systemet vilket är bra då inställningen kan göras under riktiga driftsförhållanden.

Vid reläexperimentet så kommer kritisk förstärkning och periodtid skilja sig något från de teoretiska värdena. Införandet av hysterer kommer också att påverka frekvens och amplitud. I praktiken så har det inte så stor betydelse utan det är viktigare att ha en fungerande robust tuner.

Att minimera effekten av brusstörda signaler är nödvändigt för en stabil drift samtidigt som signalerna heller inte får filtreras för hårt. Testerna visar att det vore en god ide att använda adaptiv filtrering baserat på bruskomponenten. För mycket brus kan leda till kraftigt svängande styrsignaler och är i många fall olämpliga att skicka ut då livslängden för styrdonen minskar.

I mjukvaran måste olika skydd användas t.ex. för att förhindra att variabler går utanför sitt talområde (overflow). Logisk felkontroll används och är nödvändig för att göra rimlighetsbedömningar och förhindra felfortplantning i beräkningar.

Strukturen på koden är mycket viktig för att få väldefinierade tillståndsväxlingar under exekvering. Samtidigt som tillståndsväxlingar görs kommer avbrottsrutiner att exekveras och dessa måste samverka.
7.2 Framtida utveckling

För kommersiellt bruk bör systemet implementeras med ytmonterade komponenter. Detta är önskvärt av både EMC skäl, kretskortsstorlek och monteringskvalite som maskinell produktion ger. Utrustningen skenmonteras vanligen i apparatskap men med begränsat utrymme så ett litet standardformat som passar i något färdigt instrumenthölje är att föredra.

Framtida arbete omfattar även att:

1. Fortsätta utveckla och förfina programvaran för styrning via dator och nätet med krypterad förbindelse.
2. Ta fram en databaslösning för att samla och åskådliggöra driftstatistik.
3. Införandet av en funktion för mjuk övergång vid växling av regulator-parametrar och kraftiga börvärdesändringar.
8. Referenser


9. Appendix

9.1 Kommunikationsmodul för matlab

Modulen har ett grafiskt gränssnitt som kan visa inkommande trafik från serieporten. Det går att välja om detta ska vara synligt eller ej men för utveckling och felsökning så är det praktiskt att ha det synligt. Genom att skicka ett logkommando till DLL:en så kan den också lagra inkommande data i olika format och den resulterande filen hamnar i samma mapp. För att använda den får man från matlabpromten ladda in modulen och ange kommunikationshastighet och port enligt följande:

```matlab
loadlibrary('ComPort.dll','ComPort.h')
calllib('ComPort','init',9600,1)
```

Ett styrkommando till DLL:en ser ut på följande vis:

```matlab
calllib('ComPort','LogRecivedBin')
```

För att sända och ta emot data:

```matlab
calllib('ComPort','Send','k1:1')
calllib('ComPort','get')
```

När man vill avsluta måste man ladda ur DLL:en ur minnet med följande commando:

```matlab
calllib('ComPort','free')
```

Varje ”send” tömmer mottagarbuffern så om innehållet inte ska gå förlorat måste man först hämta det med ”get”.

Listan nedan visar hjälpfilen i matlab över samtliga kommandon för DLL:en.

```matlab
% ComPort library is a Communication port dll utility
% send/receive, monitor, write to file
% Presetted values for raw communication:
% DataBits=8 StopBits=1 Parity=none FlowControl=none
% *****************************************************
%
% void init(int Baud,int ComPort);// initiate comport
% void free();                    // shut down comport
% void show();                    // Show traffic window
% void hide();                    // Hide traffic window
% void send(char *s);   // send string to comport
% char* get();                   // read data from comport
% void LogReceivedBin();          // Log incoming bytes as bin, to file
% void LogReceivedHex();          // Log incoming bytes as hex, to file
% void LogTraffic();             // Log bytes received/transmitted, to file
% void LogStop();                // stop all logging

% Hans Högström
% Revision: 0.2
```
9.2 SSL
Secure Sockets Layer (SSL) är en standard för kryptering av transportlagret i OSI-modellen och protokollet är ursprungligen utvecklat av Netscape. SSL befinner sig under applikationslagret som består av t.ex. HTTP, LDAP och IMAP och läggs över TCP/IP i nätverkslagret i OSI-modellen.


Vanliga symmetriska kryptoalgoritmer:
- DES
- Triple-DES
- SKIPJACK

Vanliga asymmetriska kryptoalgoritmer:
- RSA
- RSA key exchange

Vid symmetrisk kryptering används samma nyckel för kryptering/avkryptering medan den asymmetriska använder två olika nycklar.

9.3 I2C-bussen

Olika hastigheter är specificerade och finns i:
- low-speed mode 10 kb/s
- standard mode 100 kb/s
- fast mode 400 kb/s
- high speed mode 3.4 Mb/s

Kapacitansen får dock inte övertaga 400 pF vilket gör att endast korta ledningar bör användas och i praktiken begränsas också antalet kretsar på bussen.


Två kommunikationslinjer behövs, en datalineje Serial Data (SDA) och en klocklinje SerialClock (SCL). Eftersom I2C elektriskt är specificerad som open drain så måste pull-upp motstånd läggas
på linjerna. D.v.s. de är antingen låg eller högimpediva och klarar av egen kraft inte av att höja spänningsnivån.