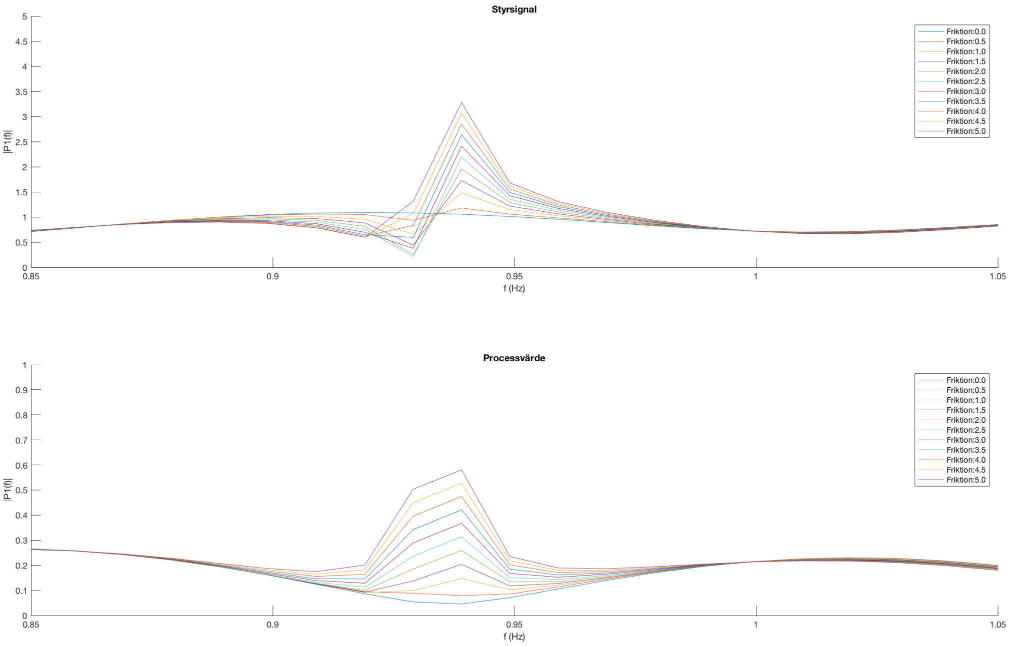
CHALMERS



Systemidentifiering och analys av olinjäriteter i styrdon

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

Pontus Svensson & Gustav Rosin

Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2018

Examensarbete 2018

Systemidentifiering och analys av olinjäriteter i styrdon

Pontus Svensson & Gustav Rosin

Institutionen för Elektroteknik
Chalmers Tekniska Högskola
Göteborg, Sverige 2018

Systemidentifiering och analys av olinjäriteter i styrdon
Pontus Svensson & Gustav Rosin

Handledare: Anders Karlström, Institutionen för Elektroteknik
Examinator: Veronica Olesen, Institutionen för Elektroteknik

Examensarbete 2018
Institutionen för Elektroteknik
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Typeset in LaTeX
Göteborg, Sverige 2018

Sammanfattning

I industrianläggningar världen över finns det extremt många reglerkretsar. Dessa används exempelvis för att kontrollera temperatur, flöde, tryck och många andra parametrar inom produktionen. Dessa reglerkretsar blir med tiden slitna med dålig reglering som följd. Det är inte alltid självklart för en processtekniker varför en reglerkrets fungerar dåligt eftersom detta kan bero på flera olika saker exempelvis dåligt inställda regulatorparametrar, defekta styrdon eller givare. Arbetets fokus har varit att utvärdera två fenomen kopplade till dåligt fungerande styrdon: glapp samt friktion. Modeller för dessa två fenomen togs fram i Simulink, modellerna kördes med hjälp av Matlab för att generera data för vidare analys.

Data analyserades med hjälp av frekvensanalys samt systemidentifiering av system med inverkan av glapp samt friktion. För att göra en lyckad systemidentifiering gjordes även analys av erforderlig samplingstid. Dimensioneringsalgoritmer utvecklades som ett verktyg för att ta fram parametrar för de system som användes i simuleringen.

Analys och simulering har fokuserat på friktion därför att detta i jämförelse med glapp skapade tydligare oscillationer hos system. Glapp visade sig även kunna motverkas effektivt med hjälp av ventillägesställare. Viss karaktäristik hos datan samlad från simuleringar av system med friktion kunde fastställas. Vidare verifiering kräver processdata från ett verkligt system och har därför inte genomförts.

Abstract

There are extremely many control loops in industrial plants around the world. These are for example used to control temperature, flow, pressure and many other parameters in the production. These control circuits will eventually be worn with bad regulation as a consequence. It is not always obvious for a process engineer why a control circuit works poorly, as this may be due to several different things, for example, badly tuned control parameters, defective actuators or sensors. This project has focused on evaluating two phenomena linked to poorly functioning control loops: friction and gaps. Models of these two phenomena were developed in Simulink, the models were then run using Matlab to generate data for further analysis.

Data was analyzed by frequency analysis and system identification of systems with impact of friction and gaps. To make a successful system identification, an analysis of recomended sampel time was made. Dimensioning algorithms were developed as a tool to obtain parameters for the systems used in the simulation.

Analysis and simulation have focused on friction as this in comparison to the gap created clearer oscillations in systems. A valve positioner also proved to be able to counteract impact from gaps in control circuits. Certain characteristics of the data collected from simulations of friction in systems could be determined. Further verification requires process data from a real system and has therefore not been made.

Förord

Vi vill tacka Veronica Olesen och Anders Karlström för handledning och hjälp under arbetets gång.

Pontus Svensson & Gustav Rosin, Göteborg, Juni 2018

Innehåll

Figurer

Tabeller

1
Introduktion

1.1 Bakgrund

Regulatorer finns i nästan alla industrier på ett eller annat sätt. Inom processindustrin används regulatorer för allt från att styra flödet i ett rör till att placera en ventil i rätt läge. Korrekt inställda regulatorparametrar är därför essentiellt för att erhålla en effektiv och optimerad produktion. Detta kan bidra till minskad energiåtgång, bättre kvalité på produkten med mera.

Att ställa in och trimma flera tusen olika regulatorer är emellertid i dagsläget oerhört tidskrävande. Detta resulterar i att företag idag väljer att använda sig av förinställda regulatorer som ej är trimmade efter processen de skall reglera. Ett annat vanligt fel som kan orsaka problem hos reglerkretsar är slitna styrdon. Detta kan vara svårt att upptäcka därför att slitaget på donet kan orsaka självsvägningar som ofta associeras med felaktiga regulatorparametrar. En undersökning visar att upp till 32% av dåligt presterande reglatorer orskas av problem med reglerventilen [1]. En ytterligare undersökning visade att det vanligaste problemet är friktion i ventiler samt feldimensionerade regulatorer [2]. I boken *Praktisk processreglering* nämns även glapp som ett vanligt förekommande fenomen i styrdon [3]. Av dessa anledningar så har arbetet fokuserat på det här tre problemen. Detta arbete är en fortsättning på Ernfjäll & Larsssons examensarbete vid Chalmers Tekniska Högskola [4].

1.2 Syfte

Huvudsyftet med detta examensarbete är att ta fram ett program som kan använda insamlad data från olika regulatorer och göra en systemidentifiering utifrån denna data. När en matematisk modell för processen finns tillgänglig, ska programvaran kunna räkna ut lämpliga reglerparametrar utefter processens olika krav. Det slutgiltiga progammet ska gå att användas tillsammans med programmet från Ernfjäll & Larssons examensarbete. Därför används i i delar av analysen samma industriella data-set som användes i Ernfjäll & Larssons arbete. Dessutom skall simuleringar samt analys av glapp och friktion utföras i syfte att kunna avgöra ifall ett systems felaktiga beteende beror på dåliga regulatorparametrar eller ifall något av styrdonen lider av dessa symptom. För att utföra dessa simuleringar skall Simulinkmodeller utvecklas och verifieras.

1.3 Avgränsningar

Data-setet som användes i Ernfjäll & Larsssons arbete är på grund av dess långa sampeltid inte lämpligt för systemidentifiering, därför kommer en stor del av arbetet att utföras med hjälp av simulerad data från Simulink. Inget arbete med hårdvara kommer att utföras och resultat kommer att verifieras teoretiskt.

1.4 Precisering av frågeställningen

Följande punkter är de som kommer undersökas:

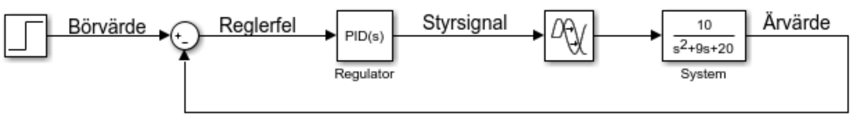
- Identifiering av process
- Framtagning av reglerparametrar efter önskade specifikationer
- Förslag till möjliga fel i systemet
- Rekommenderad samplingstid
- Simulering och analys av glapp i Simulink
- Simulering och analys av friktion i Simulink

2

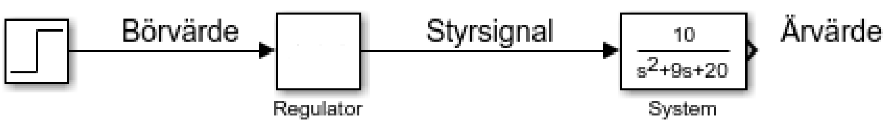
Teori

2.1 Reglerkretsar

En reglerkrets består i huvudsak av en regulator, styrdon, givare och ett system som skall regleras. En förenklad reglerkrets utan överföringsfunktion för givare eller styrdon visas i figur 2.1. I ett reglersystem brukar man prata om följande signaler: ärvärde, börvärde, styrsignal samt reglerfel. Ärvärde benämns ibland som processvärde. Regulatorns uppgift är att givet reglerfelets värde reglera kretsen. Detta görs i huvudsak med tre olika typer av regulatorer se tabell 2.1. Utöver återkopplade reglerkretsar finns det även öppna. Dessa kretsar fungerar på samma sätt som återkopplade förutom att ingen återkoppling finns. Detta leder till att ingen självreglering existerar utan processen utför exakt det som den blir tillsagd att göra oavsett resultat. Ett exempel på ett öppet system är en gammal torktumlare. I detta fall är processen de blöta kläderna som skall torkas, styrdonet är värmeelementen och regulatorn är enbart en timer. Vid slutet av timern kommer värmeelementen stängas av oberoende av ifall kläderna är torra eller inte (se figur 2.2).



Figur 2.1: Återkopplad reglerkrets



Figur 2.2: Öppen reglerkrets

Tabell 2.1: Tre olika regulatorer [5].

Regulator	Egenskaper
P	Ger en reglering där styrsignalen är proportionell mot felsignalen. Med endast P-reglering är det är svårt att uppnå god snabbhet och stablitet samtidigt. P-reglering medför ofta kvarstående fel i systemet.
PI	PI-reglering kombinerar P-Regulatorns proportionerliga verkan med en integrerande del. Detta gör att styrsignalen alltid kommer vara strängt växande om felsignalen ≥ 0 samt strängt avtagande om felsignalen ≤ 0 detta gör att man får ett system utan kvarstående fel. PI-regulatorn är däremot långsam och förlitar sig delvis på historiska mätvärden för att beräkna styrsignalen. Den är därmed dålig på att dämpa störningar och oscillationer hos snabba system
PID	PID-Regulatorn kombinerar PI-regulatorns egenskaper tillsammans med en deriverande del. Den deriverande delen ändrar kontinuerligt styrsignalen med avseende på dess lutning. Detta gör det möjligt att åstadkomma en snabb reglering som även är bra på att dämpa störningar och oscillationer. Den deriverande verkan kommer ändra styrsignalen väldigt frekvent, detta leder till slitage på styrdon

2.2 Systemidentifiering

Det arbete som utförs för att ta fram en matematisk modell för ett system kallas för systemidentifiering. Genom tillgängliga in/ut-signaler används identifieringsalgoritmen för att beräkna en matematisk modell av systemet. Identifieringsalgoritmer finns tillgängliga i Matlab genom tillägget *System identification toolbox.*

2.2.1 System identification toolbox

System identification toolbox är ett tillägg i Matlab som tar fram en matematisk modell med hjälp av in/ut-signaler från ett system. Modellen beräknas med hjälp av parametrisk identifiering, eftersom att ingen fysikalisk tolkning av dessa parametrar finns tillgängliga används en black-box-modell. Fördelen med en sådan modell är att systemets fysikaliska egenskaper inte behöver vara kända, utan endast insignalen och utsignalen behövs. Utifrån denna modell beräknas systemets parametrar utefter hur utsignalen förändras av insignalen [6]. Det finns ett flertal olika black-box-modeller och i detta arbete har ARX använts.

2.2.1.1 ARX modellstruktur

ARX-struktur kan betecknas med differensekvationen:

$$y(t) + a_1 y(t-1) + \ldots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \ldots + b_{n_b} u(t-n_b) + e(t)$$

Där $y(t)$ är utsignalen och $u(t)$ är insignalen. Termen $e(t)$ är en representation av störningar in till systemet. För att få den struktur som ARX använder introduceras funktionerna $A(q)$ och $B(q)$ där:

$$A(q) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a}$$

$$B(q) = b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b}$$

Här är q en förskjutning vilken definieras enligt:

$$q^{-1} x(t) = x(t-1)$$

Med hjälp av denna förskjuting kan differensekvationen skrivas om som:

$$A(q) y(t) = B(q) u(t) + e(t)$$

Ibland innehåller systemet en dödtid som pågår i n_k sampel. Detta leder till att formeln för $B(q)$ måste skrivas enligt:

$$B(q) = b_{n_k} q^{-n_k} + b_{n_k+1} q^{-n_k-1} \ldots + b_{n_k+n_b-1} q^{-n_k-n_b+1}$$

Där n_k defineras enligt $n_k = \frac{L}{h}$. Här är L betecknign för dödtiden och h är beteckning för samplingsintervallet. Vanligen sätts $n_a = n_b$ eftersom dessa beskriver ordningstalet [6].

Funktionen diskretiseras och följande uttryck erhålls:

$$A(z) y(t) = B(z) u(t) + e(t)$$

Överföringsfunktionen fås ur.[6]

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_1 z^{-n_k} + b_2 z^{-n_k-1} \ldots + b_{n_k} z^{-n_k-n_b+1}}{1 + a_1 z^{-1} + \ldots + a_{n_a} z^{-n_a}}$$

2.3 Egenskaper hos det återkopplade systemet

I det återkopplade systemet finns ett flertal egenskaper vilka bestämmer systemets karaktäristik. Egenskaperna som avhandlas i detta kapitel är främst kopplade till stabilitet.

2.3.1 Fasmarginal

Fasmarginalen defineras som avståndet från faskurvan ned till $-180°$ i kretsöverföringens bodediagram vid den frekvens där amplitudförstärkningen är 1, det vill säga vid överkorsningsfrekvensen ω_c. En högre fasmarginal leder till ett stabilare, men långsammare system. Tvärt om leder en lägre fasmarginal till ett snabbare och mer instabilt system. Det matematiska uttrycket för denna beskrivs av ekvation 2.1. Vanligtvis rekommenderas en fasmarginal mellan 30° och 60° [5].

$$\varnothing_m = 180° + \underline{/G_k(\omega_c)} \tag{2.1}$$

2.3.1.1 Överkorsningsfrekvens

Överkorsningsfrekvensen vilken betecknas ω_c avläses i bodediagrammet för kretsöverföringen vid den frekvensen som amplitudkurvan skär 1. Denna frekvens har ett direkt samband med hur snabbt ett system är eftersom en högre sådan betyder att systemet kan förstärka även de högre frekvenserna vilket i sin tur ger en mer aggressiv reglering [5].

2.3.2 Amplitudmarginal

Vid inversen av kretsöverföringens amplitudförstärkning vid självsvängningsfrekvensen ω_π kan man avläsa kretsöverföringens amplitudmarginal. Denna marginal kan avläsas i ett bodediagram som avståndet mellan amplitudkurvan och frekvensaxeln vid frekvensen ω_π. Värdet på amplitudmarginalen anger hur många gånger förstärkningen i regulatorn kan öka innan systemet blir instabilt. Ett vanligt värde på denna ligger mellan 2-5 gånger. Det matematiska uttrycket för amplitudmarginalen beskrivs av ekvation 2.2 [5].

$$A_m = \frac{1}{|G_k(\omega_\pi)|} \tag{2.2}$$

2.3.2.1 Självsvängningsfrekvens

Självsvängningsfrekvensen betecknas som ω_π och anger vid vilken frekvens som systemet hamnar i självsvängning beroende på hur stor förstärkningen $|G_k(\omega_\pi)|$ är. Ifall $|G_k(\omega_\pi)|$ är större än 1 i bodediagramet kommer störningar in till systemet att försätta det i självsvängning eftersom att den givna frekvensen kommer förstärkas och resonera i det återkopplade systemet. Ifall $|G_k(\omega_\pi)|$ är mindre än 1 kommer störningarna att dämpas när de resonerar i det återkopplade systemet, systemet beskrivs därför som stabilt. Matematiskt kan detta fenomen beskrivas som: Om $|G_k(\omega_\pi)| < 1$ är systemet stabilt, men om $|G_k(\omega_\pi)| \geq 1$ är systemet instabilt [5].

2.4 Dimensionering av regulator

Dimensionering av regulatorer innebär att man genom en given metod tar fram lämpliga reglerparametrar för en given process. Man brukar prata om två huvudmetoder för att utföra detta, teoretisk dimensionering samt experimentell dimensionering. Den senare är den metod som är vanligast och det är även denna som har använts i detta arbete. Experimentell dimensionering syftar på dimensionering med tumregelmetoder och teoretisk dimensionering syftar på matematiska metoder. För experimentell identifiering finns en mängd olika metoder att använda, i detta projekt har Ziegler-Nichols metod använts. Den teoretiska metod som använts är dimensionering med bodediagram [5].

2.4.1 PIDTuner

För att dimensionera en regulator till system har till en början Matlabs funktion PIDTuner använts. I projektets slutskede användes dock dimensionering med hjälp av bodediagram eftersom att det var önsvärt att ha full kontroll och insikt hur dimensioneringen skedde.

PIDTuner är ett tillägg till Matlab som tar fram regulatorparametrar till ett givet system med hänsyn till önskad snabbhet/robusthet. Algoritmen som PIDTuner använder sig av tar främst hänsyn till tre olika kriterier [7].

- **Stabilitet hos det slutna systemet.** - Det slutna systemets utsignal förblir begränsad för en begränsad insignal
- **Snabbhet.** - Det slutna systemet skall vara tillräckligt snabbt för att dämpa störningar. Högre bandbredd hos det slutna systemet kräver snabbare reglering.
- **Robusthet.** - Systemet skall ha tillräckligt med amplitudmarginal samt fasmarginal för att tillgodose fel i modellering eller variationer i systemet

Algoritmen tillgodoser dessa kraven genom att välja parametrar vilket leder till att systemet får en bra balans mellan snabbhet samt robusthet. Fasmarginalen väljs till 60° som standard. I PIDTuners interface finns möjlighet att ändra snabbhet/robusthet efter eget behov.

2.4.2 Dimensionering med bodeddiagram

Vid dimensionering med bodediagram krävs det att bodediagramet för den processen finns tillgänglig. Fördelen med denna metod är att man enkelt kan bestämma vilka kvalitéer som önskas av regulatorn, det vill säga snabbheten samt stabiliteten. Oftast ger ett högre värde på förstärkningen en snabbare, men mindre stabil reglering (och en ökad fasmarginal motsvarar en stabilare fast långsammare reglering). En avvägning mellan snabbhet och stabilitet är oftast önskvärd, valet av fasmarginal ligger i de flesta fall mellan 30-60 grader.

Först utvärderas vilket värde på förstärkningen som skall användas för att erhålla önskad fasmarginal. Bodediagramet för kretsöverföringen ritas upp och värdet på K sätts till 1. Därefter undersöks vilken frekvens som överkorsningsfrekvensen ω_c skall ligga på för att erhålla den önskade fasmarginalen, vid denna frekvensen kan K avläsas som inversen av amplitudkurvans värde enligt ekvation 2.3. Integrationstiden T_I beräknas med ekvation 2.4 [5].

$$K = \frac{1}{\left|G_k(\omega_{c)}\right|} \tag{2.3}$$

$$T_I = \frac{1}{0.2\omega_c} \tag{2.4}$$

2.4.3 Ziegler-Nichols metod

Ziegler-Nichols metod är en tumregelmetod. Detta innebär att en tabell med förbestämda värden utifrån systems egenskaper används för att räknar ut lämpliga parametrar. Den stora fördelen med denna metod är att den är väldigt enkel att använda. Nackdelen är att det är en tumregelmetod vilket inte alltid ger de mest optimala parametrarna. Arbetsmetodiken för denna metod lyder [5]: Ställ först in regulatorn som en P-regulator och öka värdet på K tills dess att systemet precis börjar självsvänga. Detta värde på K kommer att kallas för K_0. Mät vid denna tid också upp periodtiden på svängningarna och kalla detta värde för T_0. Nu kan parametrarnas värde räknas ut enligt tabell 2.2.

Metoden kan även användas för att ta fram parametrar med hjälp av en känd överföringsfunktion för en given process [5]. Först beräknas självsvägningsfrekvensen ω_π samt amplitudmarginalen A_m med hjälp av överföringsfunktionen. Dessa används för att bestämma K_0 samt T_0 enligt ekvationerna 2.5 samt 2.6

$$K_0 = A_m \tag{2.5}$$

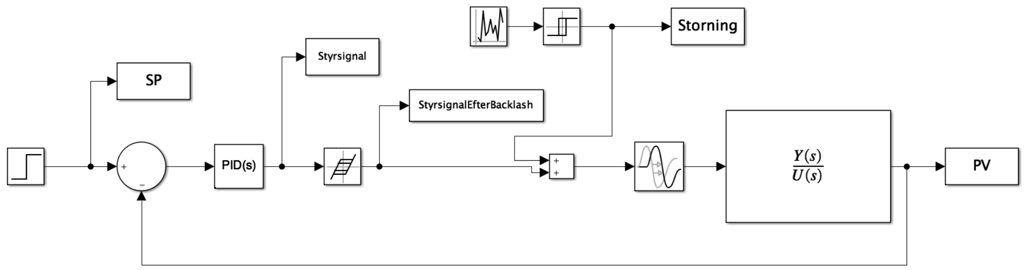
$$T_0 = \frac{2\pi}{\omega_\pi} \tag{2.6}$$

Tabell 2.2: Tabell Ziegler Nichols [5]

Regulatortyp	Parametrar		
	K	T_I	T_D
P	$0.5K_0$	-	-
PI	$0.45K_0$	$0.85T_0$	-
PID	$0.6K_0$	$0.5T_0$	$0.125T_0$

2.5 Simulink

Genom att bygga system med blockdiagram i Simulink kan man på ett smidigt och effektivt sätt simulera systemets beteende. Simulink är integrerat i Matlab vilket gör att man kan använda algoritmer från Matlab i sina simulinkmodeller och köra simuleringar parallellt i simulink via Matlab. Man kan därefter visualisera samt analysera resultatet i Matlab. I detta arbete har Simulink främst använts för att simulera system med olika värden på glapp och friktion. Ett typiskt system i form av ett blockschema ses i figur 2.3.



Figur 2.3: Modell av reglersystem med glapp

2.6 Samplingsintervall

Ett samplingsintervall hänvisar till med vilken frekvens som data inhämtas när man mäter signalerna i ett system. En för låg samplingsfrekvens i ett data-set resulterar i att identifieringen av system blir opålitlig eftersom inte tillräckligt med information finns tillgänglig. Högfrekventa ändringar kommer därför inte kunna mätas upp och registreras eftersom de rör sig med för hög frekvens. För att få en tillräckligt snabb sampeltid bör man i praktiken undersöka stegsvaret hos ett system och välja en sampeltid som ger 4-6 sampel under systemets stigtid [6].

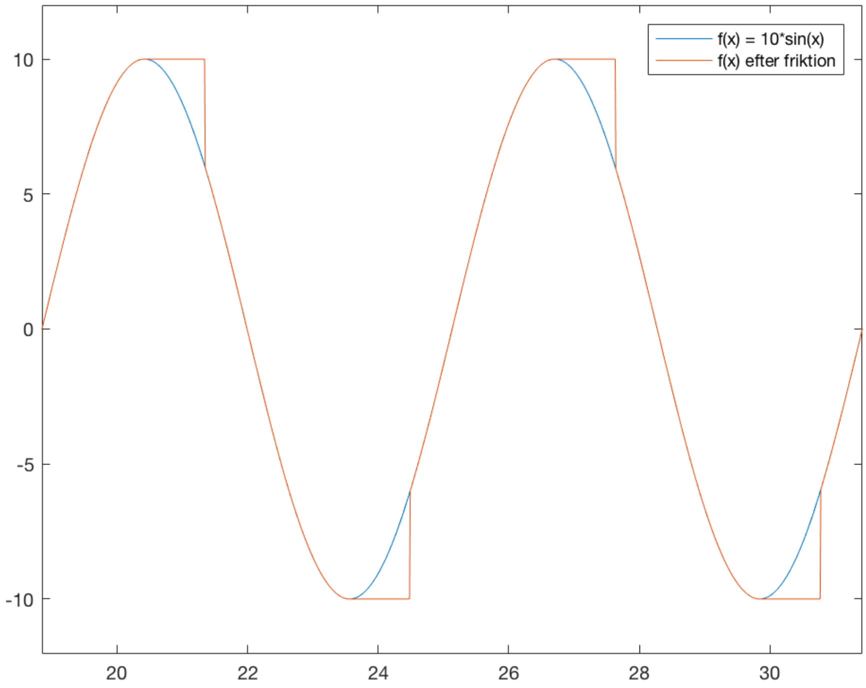
Samplingsfrekvensen ω_s erhålls enligt ekvation 2.7, där T_s är tiden mellan varje sampel. Den högsta frekvensen som kan återskapas ur en samplad signal är $f = \frac{\omega_s}{2}$ enligt Nyquists lag [8].

$$\omega_s = \frac{2\pi}{T_s} \tag{2.7}$$

2.7 Friktion

Ett stort problem i industriella reglersystem är friktion i styrdonen. I mekaniska styrdon är detta fenomen väldigt framträdande eftersom de rörliga delarna slits ut vilket leder till att friktionens inverkan blir större och större. Friktion leder till att styrdonet ej ändrar sig vid små börvärdesändringar eftersom friktionskraften ej övervinns. Detta leder till att styrsignalen ökar tills dess att styrdonet “hoppar” till sitt läge, detta gör att små ändringar av styrdonet är svårt. I takt med att slitage på styrdonet ökar och friktionens inverkan på denna blir större och större kan självsvängningar uppkomma. Självsvägningarna beror på styrdonets svårighet att göra små ändringar, detta gör att ventiländringen blir för stor. Detta återupprepas i omvänd riktning när regulatorn försöker röra tillbaka ventilen för att kompensera för överkorrigeringen [3]. I figur 2.4 illustreras ett exempel av friktion. Låt $f(x) =$

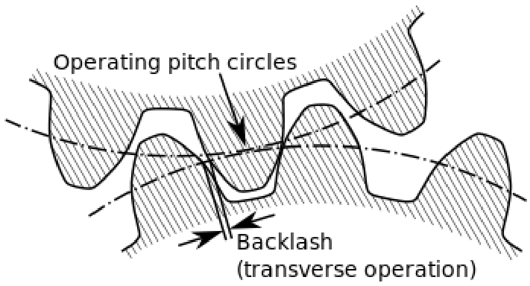
$10 \sin x$ beskriva rörelsen av styrdonet där y är styrdonets läge och x tiden. När stydonet stannar är hastigheten $f'(x) = 0$ och det krävs en signifikant ändring av styrsignalen innan donet börjar röra på sig i andra riktningen.



Figur 2.4: Rörelse med inverkan av friktion

2.8 Glapp

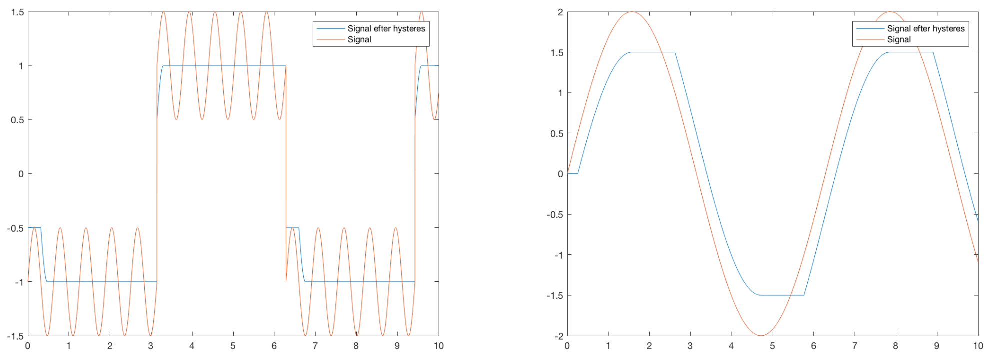
Glapp är ett fenomen som uppkommer till någon grad i nästan alla mekaniska styrdon när dess rörelseriktning kastas om. Exempelvis uppkommer fenomenet i kugghjul där det alltid finns ett visst glapp mellan kuggarnas ingrepp se figur 2.5. Detta gör att systemet upplever en viss dödtid ifall rörelseriktningen kastas om eftersom att kugghjulet måste färdas över glappet innan ingreppet i motsatt riktning. Kugghjul konstrueras alltid med ett visst glapp för att tillåta smörjning, termisk expansion, tillverkningsfel med flera.



Figur 2.5: Glapp hos mekaniska kugghjul. Bild återgiven med tillstånd. [9]

Fenomenet används ofta medvetet i tvålägesreglering för att undvika att styrdonet slår av/på för ofta. Exempel på detta är temperaturreglering av ett rum. I detta fall appliceras en dödzon runt önskad temperatur vilket gör att styrdonet aktiveras under önskad temperatur och avaktiveras över önskad temperatur.

I reglersystem kommer glappet ge upphov till en ”dödtid” i signalen, detta för att signalen släpar efter. Dödzonen som orsakas av glappet kan också medföra att signaler av låg amplitud filtreras bort helt ur signalen. Dessa två fenomen illustreras i figur 2.6



Figur 2.6: Hysteres(Glapp) med dödzon = 1, med insignalerna $\text{rect}(t) + sin(10t)$ samt $2\sin(t)$. Röd kurva är signalen innan hysteres och blå kurva är signalen efter.

3
Metod

Processdata ifrån en processindustriell anläggning lästes in från en excelfil. Ett program utvecklades i Matlab för att genomföra en systemidentifiering. Ett program utvecklades för att dimensionera regulatorer. En metod för att beräkna en lämplig samplingstid undersöktes.

En undersökning av glapp i ställdon utfördes. En Simulinkmodell för detta utvecklades. Ett system av andra ordningen användes för simuleringarna. Analysen gick ut på att undersöka hur glapp i ställdon påverkade regleringen.

En Simulinkmodell för friktion togs fram och användes i vidare analys av reglersystem. Undersökningarna bestod främst av analys med bodediagram, frekvensanalys samt jämförelse med oscillationer orsakade av en instabil regulator.

För att verifiera att Simulinkmodellerna för både glapp samt friktion var korrekta, jämfördes de karaktäristiska beteendena för dessa fenomen med de resultat som erhölls vid användning av modelllerna.

4
Systemidentifiering

Ett program för att identifiera system från data utvecklades i Matlab. Programmet använde sig av ARX-modellering för att beräkna den totala överföringsfunktionen för det slutna systemet mellan börvärde och ärvärde. Med hjälp av den kända nuvarande regulatorn kunde processen lösas ut enligt ekvation 4.1.

$$G_P = \frac{G_{TOT}}{G_R - G_{TOT} G_R} \tag{4.1}$$

För att verifiera att programmet var korrekt valdes ett system av andra ordningen, se ekvation 4.2. En regulator togs fram med hjälp av PID-tuner och en Simulinkmodell av kretsen skapades. Data samlades in med hjälp av Simulinkmodellen.

$$G_P = e^{-0.1s} \frac{10}{s^2 + 9s + 20} \tag{4.2}$$

Tillfredsställande identifiering av systemet genom att identifiera över det slutna systemet var ej möjligt, eftersom ARX-modellering inte hanterar dödtid. Dödtiden måste specificeras som en konstant av n sampels. ARX-modellen använder sig av denna specificerade dödtid för att förskjuta utsignalen n sampels och räkna ut överföringsfunktionen utan dödtid. Att förskjuta utsignalen för att kompensera för en dödtid i en sluten krets fungerar inte eftersom att den fördröjda utsignalen återkopplas och fördröjs ytterligare. Detta kan beskådas genom insättning av ekvation 4.2 i 4.3. Bryt ut $e^{-0.1s}$ enligt ekvation 4.4, ekvationens uppbyggnad ger en förskjutning av den totala signalen med 0.1 sekund. Den återkopplade delen i nämnaren förskjuts ytterligare 0.1 sekund. Detta gör att en fördröjning för hela systemet från in till utsignal ej går att bestämma.

$$G_{TOT} = \frac{G_R G_P}{1 + G_R G_P} \tag{4.3}$$

$$G_{TOT} = e^{-0.1s} \frac{G_R G_{P2}}{1 + e^{-0.1s} G_R G_{P2}} \quad där \quad G_{P2} = \frac{10}{s^2 + 9s + 20} \tag{4.4}$$

För att komma runt detta problem gjordes istället identifieringen av det öppna systemet G_P, där styrsignal användes som insignal samt processvärde som utsignal. Metoden att identifiera mellan styrsignal och processvärde rekommenderas även i artikeln *Closed-loop identification revisited* [10]. Dödtiden kan nu approximeras mellan styrsignal och processvärde med Matlab-funktionen Delayest vilken returnerar en approximation av antal sampels dödtid mellan signalerna. För att ARX-modelleringen skall funka krävs att systemet exciteras, detta åstadkoms genom att utsätta systemet för en signifikant börvärdesändring vilket i sin tur gav en signifikant ändring av styrsignalen.

5
Dimensionering

Dimensioneringsalgoritmer för regulatorer utformades i Matlab. Dimensioneringsalgoritmernas syfte var att ge förslag på nya regulatorparametrar till dåligt fungerande processer ur det industriella data-setet. Ett program utvecklades för att ge förslag på ett antal olika parametrar för de två metoderna som användes: Ziegler-Nichols och dimensionering med bodediagram. Programmet gav förslag på parametrar enligt Ziegler-Nichols metod och dimensionering med bodediagram med valfri fasmarginal.

Verifiering av metoderna gjordes med hjälp av data från industrin med kända reglerparametrar. Det öppna systemet G_p identifierades från det slutna systemet genom att använda styrsignalen från regulatorn samt processvärdet. Processen användes i dimensioneringsprogrammet för att föreslå nya parametrar, dessa parametrar visade viss likhet gentemot de parametrarna som framgick att processen använde sig utav i data-setet. Identifieringen av detta systemet kom dock att ifrågasättas på grund av datasettets långa sampeltid $T_S = 5s$. En undersökning av rekommenderad samplingstid gjordes. I boken *System Identification* rekommenderades 4-6 sampel under processens stigtid [6]. Denna rekommendation användes genom det resterande arbetet.

Ytterliggare verifiering av koden för dimensionering med bodediagram gjordes med hjälp av givna parametrar för ett känt system vilka finns i boken *Modern Reglerteknik* [5]. Parametrar för systemet beräknades med vårt program och jämfördes sedan med de givna parametrarna ur boken se tabell 5.1.

Tabell 5.1: Regulatorparametrar ur Modern Reglerteknik samt beräknade regulatorparametrar

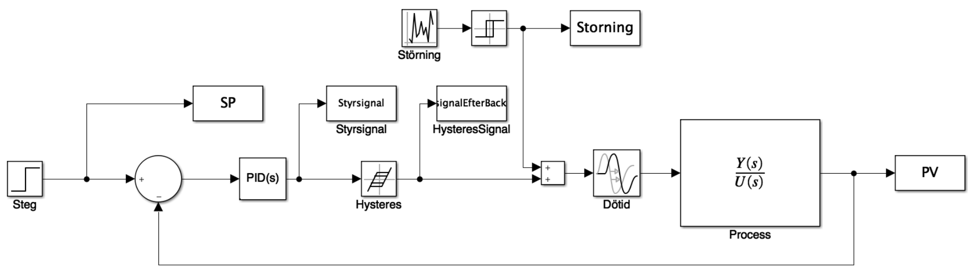
Parametrar	P	T_I
Modern Reglerteknik	2.5	2.3
Beräknad regulator	2.59	2.32

Verifiering av Ziegler-Nichols metod gjordes inte eftersom dimensionering med bodediagram ansågs vara tillräckligt.

6
Glapp

6.1 Simulering av system med glapp

Simulering genomfördes på ett andra ordningens system under påverkan av olika graders glapp. Simuleringar genomfördes för att generera data för vidare analys i syfte av att identifiera metoder för att avgöra om en process påverkas av glapp. Ett andra ordningens system med en sekund dödtid valdes $G_P = e^{-s}\frac{10}{s^2+9s+20}$. En dödtid lades på för att göra systemet känsligare. En simulinkmodell skapades för simuleringen se figur 6.1. Glapp lades till i modellen efter regulatorn i form av ett Backlash-block. En störning genererades genom att använda en slumpgenerator samt ett reläblock.

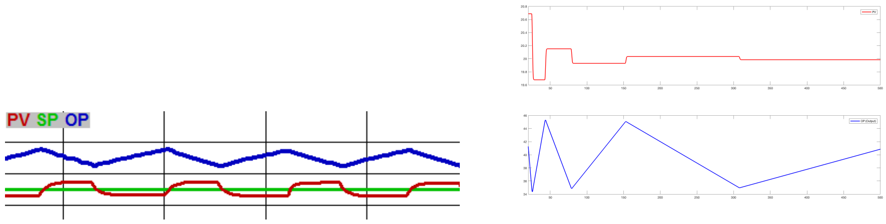


Figur 6.1: Simulinkmodell, system med inverkan av glapp

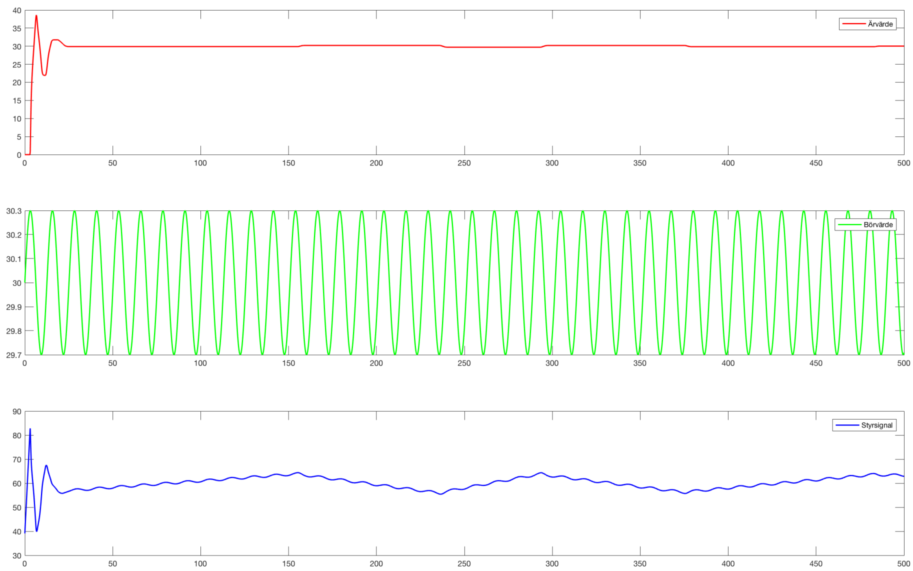
Samtliga inställningar för blocken deklareras som variabler i Matlabprogrammet som delar av en struct se bilaga A. Genom iteration genomfördes flera simuleringar av modellen med olika grader av glapp. För att verifiera att den framtagna modellen var korrekt studerades systemets styrsignal samt ärvärde. Det karaktäristiska beteendet för ett system påverkat av glapp stämde väl överens med den framtagna modellen [3]. För en illustration av fenomen orsakade av glapp samt mätvärden från den framtagna modellen se figur 6.2. Slutsatsen att modellen var korrekt kunde dras.

6.2 Analys av system med glapp

För att undersöka ifall små ändringar i styrsignalen påverkade ställdonet infördes en liten sinusformad signal som börvärde till systemet tillsammans med ett steg. Detta resulterade i att den sinusformade signalen med låg amplitud inte påverkade börvärdet eftersom dess låga amplitud samt höga frekvens gjorde att styrdonet ej hann sluta glappet se figur 6.3.



Figur 6.2: Karaktäristiska utseenden av styrsignal samt ärvärde vid glapp och uppmätt data från simulering. Styrsignal i blått, ärvärde i rött och börvärde i grönt [11].



Figur 6.3: Karaktäristiska utseenden av styrsignal samt ärvärde vid glapp och uppmätt data från simulering. Styrsignal i blått, ärvärde i rött och börvärde i grönt med sinusformade ändringar.

Resultaten visade på att glapp inte leder till självsvängningar hos systemet utan snarare att systemet blir lite segare eftersom att ventilen ”släpar” efter dess tänkta position. Vidare testning visade också att en ventillägesställare eliminerar effekterna av glapp effektivt. Ventillägesställaren med hjälp av en givare utvärderar ifall ventilen/styrdonet ger det utslag som förväntas vid en given styrsignal. Ifall detta inte är fallet reglerar ventillägesställarens regulator signalen till styrdonet tills dess att utslaget blir korrekt.

7

Simulering av system med inverkan av friktion

En modell av friktion som gick att applicera i ett reglersystem skapades. Simuleringar gjordes med hjälp av denna för att samla data till vidare analys.

7.1 Simulinkmodell för friktion

Modellen för friktion utgick ifrån en matematisk modell som beskriver rörelsen av ett föremål med massan m mot ett underlag [12].

Ett föremål med massan m förflyttas över ett underlag, och dess lägeskoordinat är x ges dess rörelseekvation av ekvation 7.1.

$$m\ddot{x} = F_d + F \tag{7.1}$$

där F_d är kraften som inverkar på föremålet och F är friktionen mot underlaget. Ekvation 7.2 beskriver friktionskraften om man antar att friktionskraften vid vila är F_0 samt friktionskraften vid rörelse är F_1

$$f(n) = \begin{cases} -F_1 \operatorname{sgn} \dot{x} & \dot{x} \neq 0 \\ -F_d & \dot{x} = 0, |F_d| \leq F_0 \\ -F_0 \operatorname{sgn} F_d & \dot{x} = 0, |F_d| > F_0 \end{cases} \tag{7.2}$$

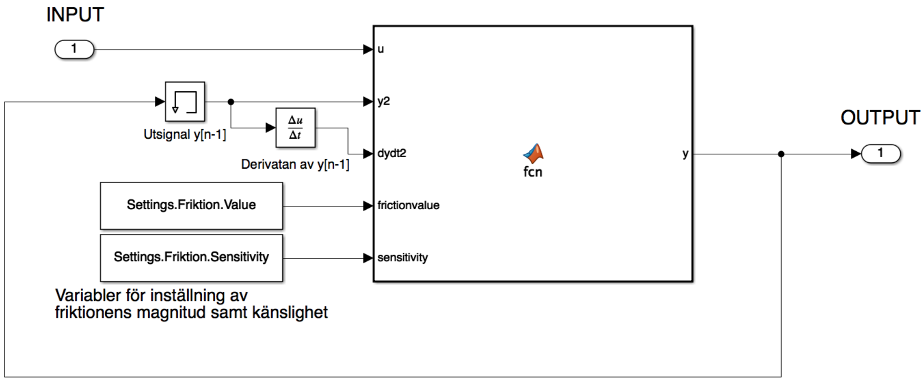
Där $\operatorname{sgn}(x)$ är signumfunktionen som beskrivs av ekvation 7.3

$$\operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases} \tag{7.3}$$

En simulinkmodell utvecklades med underlag av ekvationerna 7.1, 7.2 samt 7.3 för att simulera friktion hos ett styrdon. Simulinkblocket *MATLAB Function* användes för att programmera huvudfunktionsblocket i simulinkmodellen. Styrdonets rörelse representeras av utsignalen från modellen. Insignalen till blocket representerar styrdonets tänkta rörelse utan friktion närvarande. För att funktionen skall uppfyllas krävs återkoppling av systemet. Detta görs med ett memory-block i Matlab. Blocket sparar signalen till nästa sampel, för att inte skapa en algebraisk loop i koden.

Programvaran avgör om styrdonet rör sig eller ej. Om styrdonet är i rörelse sätts utsignal till insignalen. Ifall donet ej är i rörelse utreds ifall $|insignal - utsignal| > friktionsvärde$. Om detta stämmer sätts utsignal till insignal. Ifall villkoret inte stämmer betyder detta att styrdonet står stilla och att insignalens ändring inte är tillräckligt stor för att sätta styrdonet i rörelse. I detta fall sparas den gamla utsignalen till nästa sampel. Se Appendix B för fullständig kod.

Blocket använder sig av två valbara inställningar som reglerar friktionens känslighet samt dess storhet. Dessa variabler har ändrats vid simulering för att samla data med olika friktionsvärden.



Figur 7.1: Simulinkmodell för friktion

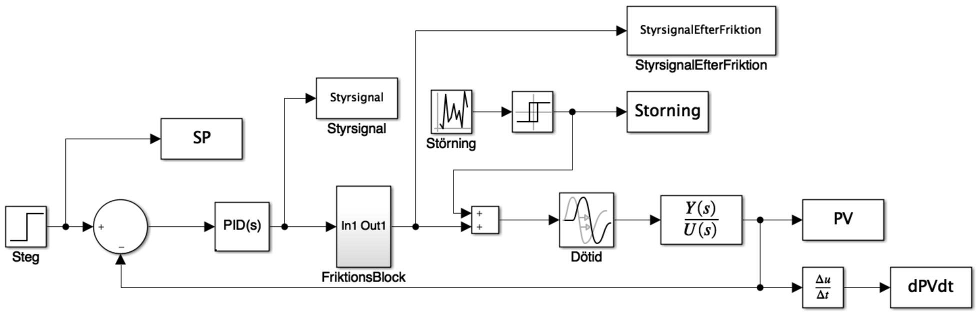
Simulink användes för att utvärdera hur ett system påverkas av olika grader friktion. Ett andra ordningens system med dödtid simulerades med hjälp av simulinkmodellen se ekvation 7.4 samt figur 7.2.

$$G_P = e^{-ts}\frac{10}{s^2 + 9s + 20} \qquad \text{där } t = 0.1 \tag{7.4}$$

Modellen bygger på ett återkopplat reglersystem där friktionsmodellen kopplas in efter regulatorn som en modell av styrdonet. Dödtiden applicerades på systemet eftersom att simuleringar i ett tidigt stadie visade att en viss dödtid krävdes för att processen skulle ge det friktionsliknande beteendet som det beskrivs i boken *Praktisk processreglering* [3].

Arbetsgången i denna analys är att från en processingenjörs synvinkel undersöka en reglerkrets som inte fungerar som den ska. Analysen går ut på att data samlas in från en process som självsvänger på grund av friktion. Därefter sker en identifiering av processen med hjälp av den insamlade datan. Med den identifierade överföringsfunktionen kan nya regulatorparametrar föreslås med hjälp av dimensioneringsprogrammet.

Simuleringar genomfördes med friktionsvärde $f = 0.5 \cdot n \qquad \text{där } n = 0, 1, 2...10$. Datan lagrades i en struct för att användas i identifieringsfunktionen. Identifieringen på det slutna systemet skedde mellan styrsignal och processvärde. Dödtiden upp-



Figur 7.2: Simulinkmodell, system med inverkan av friktion

skattades med hjälp av Matlabs inbygda funktion Delayest vilken uppskattar hur många samples dödtid systemet innehåller. Styrsignal, processvärde samt dödtiden användes för att identifiera systemet utan dödtid med hjälp av ARX-modellering. Därefter applicerades dödtiden på det identifierade systemet.

Nya parametrar togs fram för alla processidentifieringar med varierande friktion, ytterligare simuleringar med nya reglerparametrar utfördes för alla olika friktionsvärden. Detta för att avgöra hur system med inverkan av friktion beter sig vid nyidentifierade reglerparametrar. Försöket visade på att trimning av regulatorn inte har någon nämnvärd effekt när systemet påverkas av friktion.

Datan analyserades främst genom att:
- Studera nämnar- samt täljarpolynomet av den identifierade processen
- Analysera de olika identifierade processernas bodediagram
- Plotta datan i tidsgrafer
- Studera frekvensanalysen hos systemet

8

Analys av system med friktion

8.1 Analys av polynom hos identifierad process

Ett andra ordningens system med 0.1 sekund dödtid användes vid simuleringarna se ekvation 8.1.

$$G_P = e^{-0.1s} \frac{10}{s^2 + 9s + 20} \tag{8.1}$$

Identifiering av systemet från styrsignal till processvärde gav polynom enligt tabell 8.1. Där konstanterna förklaras av ekvation 8.2.

$$G_P = e^{-ts} \frac{a_n s^n + ...a_2 s^2 + a_1 s^1 + a_0 s^0}{b_n s^n + ...b_2 s^2 + b_1 s^1 + b_0 s^0} \tag{8.2}$$

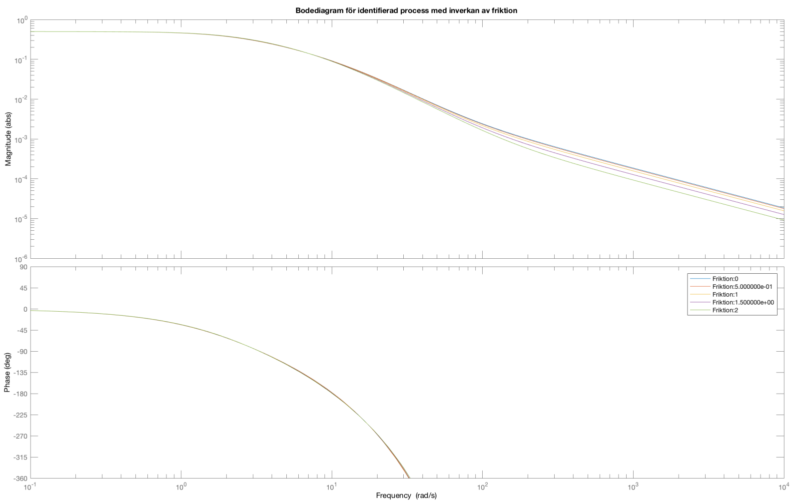
Friktion vid identifiering	t	b_2	b_1	b_0	a_1	a_0
0	0.1	1	15.73	31.32	-0.1854	15.66
0.5	0.1	1	15.56	31	-0.1778	15.5
1	0.1	1	15.07	30.11	-0.1567	15.06
1.5	0.1	1	14.35	28.82	-0.1264	14.41
2	0.1	1	13.51	27.32	-0.09179	13.66
2.5	0.1	1	12.64	25.78	-0.05699	12.89
3	0.1	1	11.8	24.32	-0.0245	12.16
3.5	0.1	1	11.03	23	0.004441	11.5
4	0.1	1	10.33	21.84	0.02946	10.92
4.5	0.1	1	9.713	20.84	0.05057	10.42
5	0.1	1	9.174	19.99	0.06824	9.992

Tabell 8.1: Koefficienter för identifierade processen

Tabellen visar att identifiering av ett system som påverkas av olika magnituder av friktion liknar varandra till stor del, relationen mellan konstanterna för varje identifiering liknar originalsystemet utan friktion. Det framgår ur tabellen att $a_0/b_0 \approx 0.5$ för alla friktionsvärden vilket betyder att den statiska förstärkningen $|G(\omega)|_{\omega \to 0} \approx 0.5$.

8.2 Analys av bodediagram

Likt analys av polynomet framgick vid analys med bodediagram att den statiska förstärkningen $|G(\omega)|_{\omega \to 0} \approx 0.5$ se figur 8.1. Ur bodediagrammet framkom också att förstärkningen för låga frekvenser ökade med ökande friktion samt att förstärkningen för höga frekvenser minskade med ökande friktion, brytpunkten för dessa två fenomen inträffade vid 8 rad/s. Något visuellt samband vid ökande friktion kunde ej urskiljas i bodediagrammets faskurva.



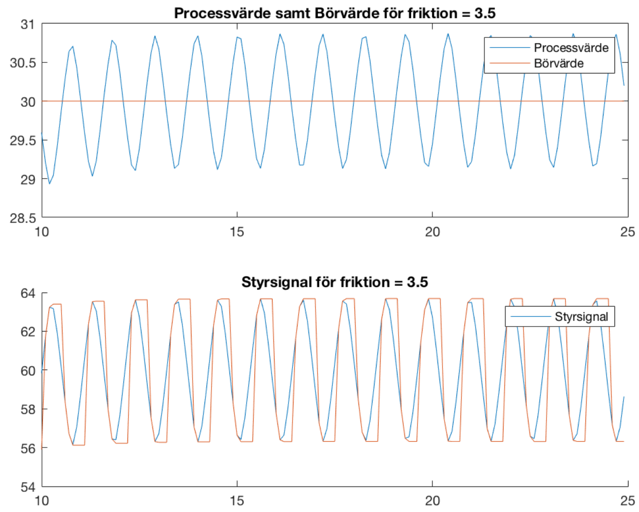
Figur 8.1: Bodediagram för identifiering av processer med olika grader av friktion

8.3 Analys av tidsgraf

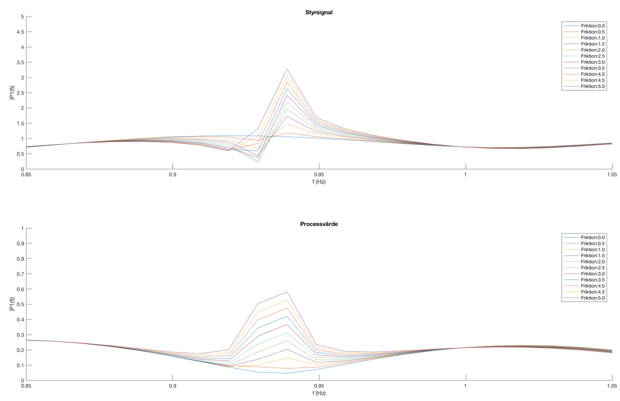
Tidsgrafer för olika graders funktion plottades se figur 8.2. De jämfördes med det karaktäristiska beteendet för system med inverkan av friktion beskrivet i boken *Praktisk processreglering* [3]. Denna jämförelse ansågs verifiera friktionsmodellen.

8.4 Frekvensanalys av data

Frekvensanalys genomfördes på de identifierade processerna samt styrsignalerna där friktionsvärdet varierade från 0-5. Vid frekvensanalysen framkom att system med friktion endast avvek vid ett ställe i jämförelse med system utan friktion se figur 8.3. Avvikelsen framkom i form utav en spik vid frekvensen 0.93Hz. Denna avvikelse framkom på grund utav självsvägningen orsakad av friktion. Självsvägningen visade sig vara vid samma frekvens för samtliga grader av friktion, för både styrsignal samt processvärde. Vid övriga frekvenser i frekvensgrafen avvek inte system påverkade av friktion något i jämförelse med system utan friktion.



Figur 8.2: Tidsgraf av friktionsvärde 5



Figur 8.3: Frekvensgraf av olika graders friktion

8.5 Jämförelse av självsvängande regulator och friktion

Efter att ha gjort upptäckten att olika grader av friktion orsakar självsvägningar vid samma frekvens valde vi att dimensionera en regulator vars självsvägningsfrekvens $\omega_\pi = 0.93Hz = 5.84rad/s$. Regulatorn dimensionerades genom att ta fram G_k och utvärdera dess parametrar så att $\omega_\pi = 5.84rad/s$ samt att $|Gk|_{\omega \to 5.84rad/s} = 1$. Kretsöverföringen togs fram enligt ekvation 8.3

$$Gk(j\omega) = Gr(j\omega)Gp(j\omega) = K(1 + \frac{1}{j\omega T_i})e^{-0.1j\omega}(\frac{10}{-\omega^2 + 9j\omega + 20}) \tag{8.3}$$

För att dimensionera regulatorn bestämdes T_i så att $\underline{/Gk(\omega = 5.84)} = -180°$. Se ekvationerna 8.4, 8.5, 8.6 samt 8.7. Detta gav ett $T_i = 0.194$

$$\angle Gr(\omega) = \arctan -\frac{1}{\omega T_i} \tag{8.4}$$

$$\angle Gp(\omega) = -0.1\omega\frac{180}{\pi} - \arctan\frac{9\omega}{20-\omega^2} \tag{8.5}$$

$$\angle Gk(\omega) = \angle Gr(\omega) + \angle Gp(\omega) \tag{8.6}$$

$$\angle Gk(\omega = 5.84) = 180^\circ \implies T_i = 0.194 \tag{8.7}$$

K bestämdes så att $|Gk(\omega = 5.84)| = 1$ Se ekvationerna 8.8, 8.9, 8.10 samt 8.11

$$|Gr(\omega)| = K\sqrt{1 + \frac{1}{\left(\omega T_i\right)^2}} \tag{8.8}$$

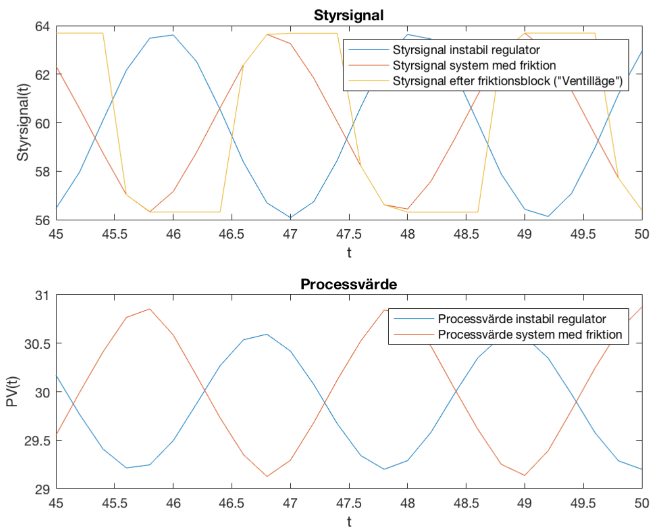
$$|Gp(\omega)| = \frac{10}{\sqrt{\left(20-\omega^2\right)^2 + \left(9\omega\right)^2}} \tag{8.9}$$

$$|Gk(\omega)| = |Gr(\omega)|\,|Gp(\omega)| \tag{8.10}$$

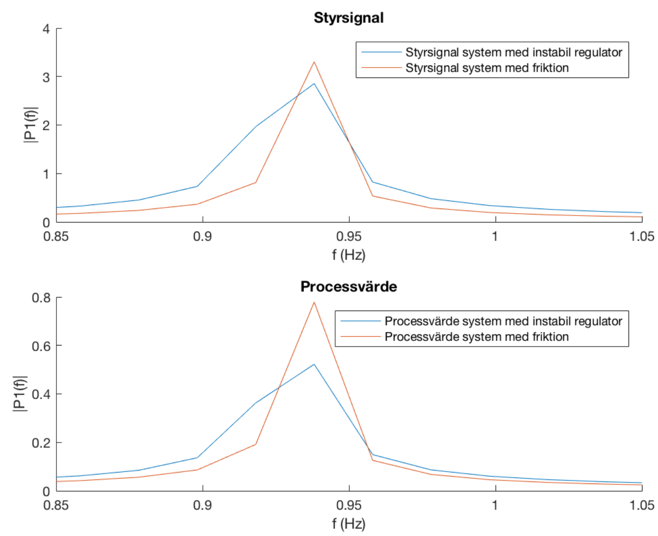
$$|Gk(\omega = 5.84)| = 1 \quad \text{där} \quad T_i = 0.194 \implies K = 4.077 \tag{8.11}$$

Den nyframtagna regulatorn vars regulator dimensionerades efter att oscillera i samma frekvens som systemen med friktion jämfördes med detta. Systemet med friktion genomgick ett stegsvar på 30 enheter likt alla andra simuleringar. Systemet med instabil regulator genomgick ett tillräckligt högt stegsvar för att oscillationernas amplitud i styrsignalen skulle likna dess hos friktionen. Detta stegsvaret visade sig vara ett steg på 0.7 enheter. Oscillationerna jämfördes genom att observera tidsgrafen för båda systemen samt frekvensanalys för svägningarna se figur 8.4 samt 8.5. I frekvensanalysen ingick inte den första delen av signalen. Detta för att ta bort enhetssteget och bara analysera oscillationerna.

Eftersom den instabila regulatorn genomgick ett enhetssteg på 0.7 enheter för att dess styrsignal skulle självsvänga i samma amplitud som systemet med friktion har dess signaler justerats i tidsgrafen genom att addera en konstant för att förskjuta signalen i Y-led. Det framgår tydligt i grafen att systemet med friktion har en större förstärkning jämfört med det självsvängande systemet. Detta för att styrsignalerna har liknande amplitud men att processvärdet för systemet med friktion har en tydligt högre amplitud. Detta återspeglas också i frekvensanalysen där man tydligt ser en större skillnad i spik vid den givna frekvensen. I tidsgrafen är det svårt att urskilja någon skillnad utöver den förändrade amplituden. Däremot ser man en viss skillnad i frekvensanalysen där frekvensbidragen verkar vara lite mer spridda hos systemet med instabil regulator.



Figur 8.4: Styrsignal samt processvärde för instabilt system orsakat av dåligt inställd regulator samt för friktion



Figur 8.5: Frekvensanalys av system med dåligt inställd regulator samt system med friktion

9
Resultat och tolkning

9.1 Systemidentifiering

Identifieringsprogramet baserades till en början på en identifiering i ett slutet system mellan ärvärde och börvärde. Detta var en problematisk metod eftersom dödtiden i processen ej kunde fastställas. En metod som föreslås i artikeln *Closed-Loop identification revisited* var identifiering mellan styrsignal och processvärde [10]. Denna metod användes för att kunna bestämma korrekt dödtid. Vid verifieringen användes ett system av andra ordningen se ekvation 4.2. Det resulterande programmet använder sig av identifiering från styrsignal till processvärde i ett slutet system. Detta system kunde identifieras korrekt och en slutsats drogs att programmet fungerade.

9.2 Dimensionering av regulator

De metoder som användes för att dimensionera regulatorn i programmet var Ziegler-Nichols samt dimensionering med bodediagram. För att verifiera programmet dimensionerades regulatorparametrar till systemet från data-setet. Dessa parametrar visade en likhet mot de som använts i processindustrin, men den långa samplingstiden gjorde att ytterligare verifiering krävdes. Med hjälp av ett system av andra ordningen med givna reglerparametarar vilka fanns i boken *Modern Reglerteknik* [5], kunde programmet verifieras genom att beräkna reglerparametrar för det givna systemet och jämföra dessa med de givna. De beräknade parametrarna stämde väl överens med de givna och slutsatsen drogs att programmet fungerade korrekt.

9.3 Glapp

Den utvecklade Simulinkmodellen verifierades mot det karaktärisktiska beteendet hos system med inverkan av glapp. Simuleringar visade att glapp i styrdon inte orsakar självsvängningar i hög grad, systemet kommer däremot bli segare att reglera eftersom styrdonet behöver sluta glappet innan ingrepp i motsatt riktning sker. Detta gör att styrdonet har svårare att göra ”små” ändringar eftersom att den alltid kommer behöva sluta ett relativt till ändringen stort glapp innan donet kan göra en ändring i motsatt riktning.

Ytterligare analys visade att en ventillägesställare kan eliminera problem med glapp effektivt eftersom att ventillägesställaren aktivt reglerar styrdonet tills dess att den ger förväntat utslag.

9.4 Friktion

9.4.1 Simulering

Vid analys av tidsgrafen framgick att de simulerade signalernas dynamik stämde väl överens med det karaktäristiska beteendet för signaler med inverkan av friktion [3]. Det framgick även att trimning av regulatorn inte hade någon nämnvärd effekt, oscillationer i systemet framkom fortfarande. Detta var väntat med tanke på hur friktionsmodellen är konstruerad. Allt eftersom att börvärdet blir mer stabilt och närmar sig ärvärdet kommer mindre justeringar av styrsignalen behövas. Friktionens effekt på styrdonet förhindrar detta genom att endast en stor ändring av styrsignalen kan få donet att flytta på sig. När det väl sätts i rörelse är överkorringeringen ett faktum och fenomenet återupprepas i omkastad riktning. Detta betyder att även en långsam regulator som ger systemet en god stabilitet inte eliminerar oscillationerna.

9.4.2 Analys av friktion

Systemidentifiering gjordes för samtliga system med inverkan av olika graders friktion. Identifieringarna jämfördes mot originalsystemet, detta visade att koefficienterna i nämnar samt täljarpolynomet skiljer sig med ökande friktion. Däremot skiljde sig inte relationen mellan koefficienterna mycket, detta visade sig i bodediagrammet eftersom amplitud/faskurvan för de olika identifieringarna liknade originalsystemet. Vissa skillnader och likheter kunde dock bestämmas med hjälp av polynomen samt bodediagrammet.

- Den statiska förstärkningen $|G(\omega)|_{\omega\to 0} \approx 0.5$ för samtliga friktionsidentifieringar. Detta stämmer överens med den statiska förstärkningen utn friktion närvarande.
- Förstärkningen för låga frekvenser höjdes med ökande friktion
- Förstärkningen för höga frekvenser sänktes med ökande friktion

Vid frekvensanalys av styrsignal samt processvärde upptäcktes att systemen med friktion skiljde sig från systemet utan friktion vid endast en frekvens. Denna avvikelsen sågs vid 0.93Hz vilket var frekvensen hos oscillationerna vid de system som påverkades av friktion. Vad som var intressant här är att oscillationerna hade samma frekvens oavsett hur hög inverkan av friktion systemet utsattes för.

Självsvängningarna orsakade av friktion jämfördes med självsvängningar orsakade hos ett system med dåligt inställd regulator. Regulatorn dimensionerades så att frekvensen av dess självsväningar matchade frekvensen hos systemet med friktion (0.93Hz). Frekvensanalysen visade att frekvensbidragen var mer spridda runt 0.93Hz hos systemet med dåligt inställd regulator. Det framkom även att förstärkningen för frekvensen var högre för systemet med friktion. Detta stämmer även överens med iakttagelserna gjorda från identifieringarnas bodediagram.

10
Slutsats och diskussion

10.1 Identifiering av process

Identifiering av process skedde med hjälp av en identifiering mellan styrsignal och processvärde. Identifieringen verifierades med ett känt system av andra ordningen. Resultatet av verifieringen var tillfredställande och målet ansågs vara uppfyllt.

10.2 Dimensionering av regulatorer

Två metoder användes för att dimensionera regulatorer, Ziegler-Nichols och dimensionering med Bodediagram. Programmet verifierades och slutsatsen drogs att målet var uppfyllt. Förbättringsmöjligheter är möjliga eftersom verifiering mot ett verkligt system ej är utfört. Även program för ytterligare dimensioneringsmetoder kan utvecklas.

10.3 Samplingstid

En metod för att hitta lämplig samplingstid undersöktes. Målet var att hitta en metod för rekommenderad samplingstid vilket denna undersökning gjorde.

10.4 Glapp

Målet var att utveckla en simuleringsmodell för glapp samt analysera innebörden av detta i reglerkretsar. Modellen utvecklades och verifierades. Slutsatsen drogs att Simulinkmodellen var korrekt. Analysen visade att en ventillägesställare kan användas för att eliminera inverkan av glapp. Ingen djupare analys har genomförts eftersom friktion ger upphov till större problem. Förbättringsmöjligheter är möjliga med en mer djupgående analys av glapps inverkan samt hur detta kan detekteras i en reglerkrets.

10.5 Friktion

Simuleringarnas syfte var att hitta likheter och karaktäristiska drag hos datan i de systemen som påverkades av olika grader av friktion. Detta gjordes till viss del eftersom förstärkningen för de identifierade systemen med friktion skiljde sig något mot det verkliga systemet utan friktion. Små skillnader återfanns också i frekvensanalysen där jämförelse mellan oscillationerna skapade av ett instabilt system samt de skapade av inverkan av friktion gjordes. Målet var att analys av friktion i system skulle generera karaktäristiska kännetäcken för att upptäcka friktion i verkliga system.

I verkligheten är originalsystemet utan påverkan av friktion ofta inte givet. Förslagsvis kan iakttagelserna som hittats i detta arbetet användas för att hitta tendenser av friktion i system genom att göra flera identifieringar med tid i mellan. Identifieringar kan jämföras för att avgöra ifall friktion har uppkommitt eller förvärrats.

De kännetecknen som genereras har inte kunnat verifieras mot verklig data. Dessutom är skillnaderna som upptäckts väldigt små och de har endast hittats med jämförelse av originalsystemet utan påverkan av friktion. Upptäckterna som gjorts möjliggör inte upptäckten av friktion i ett system genom att endast kolla på dess signaler. Signalerna måste jämföras med en tidigare referens av systemet alternativt systemet utan friktion närvarande. De resultat som framkommit vid analys av friktion är också osäkra gentemot beteenden i verkligheten. Detta för att den konstruerade friktionsmodellen är ”ideal” i bemärkelsen att den påverkas av samma friktionsmotstånd i båda riktningar samt att motståndet alltid är konstant genom simuleringen. I verkligheten skiljer sig säkerligen motståndet beroende på ett flertal parametrar: vilken riktning donet rör sig åt, vilken position donet har med mera. Dessa faktorer har inte ingått i analysen och det är osäkert hur de kan påverka resultatet. För ett mer säkert resultat krävs vidare analys samt jämförelse med verklig data.

10.6 Övriga slutsatser kring identifiering av olinjärheter i styrdon

Några metoder för att direkt identifiera friktion eller glapp har ej funnits. Metoderna som utvecklats för att upptäcka friktion kräver att det ursprungliga systemet utan friktion är känt, alternativt en tidigare referens med mindre inverkan av friktion.

Ett förslag för att utvärdera ifall styrdonen behöver ses över i ett oscillerande system är att utesluta fel hos regulatorn. Detta kan göras genom att studera fasmarginalen hos kretsöverföringen. Ifall denna ligger inom ett acceptabelt intervall (30°- 60°), kan slutsatsen att felet troligen återfinns i processen eller styrdonet dras. Därefter kan vidare analys av felets natur göras.

Litteraturförteckning

[1] Desborough, L., & Miller, R. (2002). "Increasing customer value of industrial control performance monitoring—Honeywell's experience." In AIChE symposium series, No. 326 (Vol. 98, ss. 153–186).

[2] M. Bauer, A. Horch, L. Xie, N. Thornhill "The current state of control loop performance monitoring – A survey of application in industry," Journal of Process Control, vol. 38, ss. 1-10, Feb. 2017. [Online]. Tillgänglig: http://www.sciencedirect.com/science/article/pii/S0959152415002127. Hämtad: 18 juni. 2018.

[3] T.Hägglund, *Praktisk processreglering.* uppl. 3, Malmö, Sverige: Studentlitteratur. 2008.

[4] R.Ernfjäll och J.Larsson, "Diagnostisering av regulatorer", examensarbete för kandidatexamen, Institution för signaler och system, Chalmers tekniska högskola, Göteborg, Sverige, 2017.

[5] B.Thomas, *Modern Reglerteknik.* uppl. 5, B.Magnusson, Red. Stockholm, Sverige: Liber AB.

[6] L.Ljung, *System identification: Theory for the user.* uppl. 2, T.Kailath, Red. New Jersey, USA: Prentice Hall, 1999.

[7] Mathworks, "PID Tuning Algorithm", 2018. [Online]. Tillgänglig: `https://se.mathworks.com/help/control/getstart/pid-tuning-algorithm.html`, hämtad: 2018-06-04.

[8] https://it-ord.idg.se/sprakwebben/ord.asp/?ord=Nyquists

[9] D.Richfield, "Backlash", 2010, [Elektronisk bild]. Tillgänglig: `https://en.wikipedia.org/wiki/Backlash_(engineering)#/media/File:Backlash.svg`. Hämtad: 2018-06-04.

[10] U.Forsell och L.Ljung, "Closed-loop identification revisited", ss.3, jan 1999. doi:10.1016/S0005-1098(99)00022-9, [Online]. Tillgänglig: `https://ac.els-cdn.com/S0005109899000229/1-s2.0-S0005109899000229-main.pdf?_tid=131027bc-b07d-41f1-adc5-3eb34d53b7f3&acdnat=1528387117_b5e7e4f515e66aca323dbe251e6f06b1`, Hämtad: 2018-06-07.

[11] C.Hardy, "Limited Duty Deadband Oscillation", 2014, [Elektronisk bild]. Tillgänglig: `https://www.crossco.com/blog/basics-tuning-pid-loops`. Hämtad: 2018-06-04.

[12] T.Glad och L.Ljung, *Reglerteori: Flervariabla och olinjära metoder.* uppl. 2, Lund, Sverige: Studentlitteratur.

A

Appendix 1

```
%Transferfcn
Settings.transferFcn.Delay = 1.0;
Settings.transferFcn.A = [10];
Settings.transferFcn.B = [1 9 20];
Settings.transferFcn.Gp = tf(Settings.transferFcn.A...
     ,Settings.transferFcn.B,'InputDelay'...
     ,Settings.transferFcn.Delay);

%Relay
Settings.Relay.SwitchOnPoint = 0.9;
Settings.Relay.SwitchOffPoint = -0.95;
Settings.Relay.OutPutWhenOn = 10;
Settings.Relay.OutPutWhenOff = 0;

%RandomNumberGenerator
Settings.RndGenerator.Seed = randi(1000);
Settings.RndGenerator.Minimum = -1;
Settings.RndGenerator.Maximum = 1;
Settings.RndGenerator.SampleTime = 5;

%Backlash
Settings.Backlash.DeadbandWidth = 0;
Settings.Backlash.InitialOutput = 0;

%PID
Settings.PID.K = 1.0156;
Settings.PID.Ti = 0.67119;
Settings.PID.D = 0;
Settings.PID.N = 0;

%Step
Settings.Step.StepTime = 0;
Settings.Step.InitialValue = 0;
Settings.Step.FinalValue = 30;

%DataStorage
Settings.Data.Sampletime = 0.05; % -1 for inherited.
```

```
Backlash = (0:10)*2;


A = 0;
for i=1:11
    tic
    Settings.Backlash.DeadbandWidth = Backlash(i);

    sim('SimuleringDatainsamlingr2016b.slx');
    Data(i, i2).PV = PV;
    Data(i, i2).SP = SP;
    Data(i, i2).Styrsignal = Styrsignal;
    Data(i, i2).Storning = Storning;
    Data(i, i2).StyrsignalEfterBacklash...
        = StyrsignalEfterBacklash;
    Data(i, i2).BacklashSetting = Backlash(i);
    Data(i, i2).Kfaktor = Kfaktorer;


    A = A + toc;
    avgtime = A/i;
    timeleft = avgtime*(11-i);


    X = sprintf('Calculated Time Left %d Seconds',timeleft);
    disp(X)

end
```

B

Appendix 2

```

function y   = fcn(u ,y2 ,dydt2 ,frictionvalue ,sensitivity)

if abs(dydt2) > sensitivity
% Rör sig donet? Isåfall utsignal = insignal

    y = u;

else  %Donet rör sig inte
    if abs(y2 - u) > frictionvalue
    %Är  |utsignal-insignal| > FrictionValue?
        y = u;    %Isåfall utsignal = insignal
    else
        y = y2;
        %Är inte skillnaden >FrictionValue?
        %Sätt y[n] = y[n-1]
    end
end
```