

## Undersökning av Scilab

För modellbaserad design

*Examensarbete inom högskoleingenjörsprogrammet Mekanik*

*Dan Carlsson*

*Madilyn Ottosson*

Institutionen för signaler och system

*Avdelningen för reglerteknik, automation och mekatronik.*

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige, 2013

# Undersökning av Scilab

För modellbaserad design

*Examensarbete inom högskoleingenjörsprogrammet Mekanik*

*Dan Carlsson*

*Madilyn Ottosson*

Institutionen för signaler och system

*Avdelningen för reglerteknik, automation och mekatronik.*

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige, 2013

## Förord

Detta projekt var ett avslutande moment inom mekatronikprogrammet på Chalmers Tekniska Högskola, ett examensarbete på 15 hp och som löper över 10 veckor. Arbetet utfördes på företaget Cpac Systems i Mölndal.

Cpac Systems kunder är Volvo Penta, Volvo Group och Yamaha, och arbetar bland annat med kontrollsystem för fordon. En del av arbetet utfördes modellbaserat och till detta finns flertalet olika program. Detta examensarbets mål var att undersöka ett av dem.

Vi vill ge ett stort tack till vår handledare Göran Hult för hans feedback på simuleringsmodellen och för tips på litteratur. Tack även till vår handledare Peter Forsberg på Cpac Systems för att ha hjälpt oss och lett in oss på rätt riktning i vårt arbete.

Slutligen även ett stort tack till vår vän Patrik Larsson, på DAI, för datorsupport.

Dan Carlsson  
Madilyn Ottosson

# Sammanfattning

CPAC Systems använder sig idag av bland annat beräkning- och simuleringsprogrammet Matlab/Simulink (MS). MS ägs av Mathworks. Programvaran MS och dess extra bibliotek kostar pengar för varje licens. Dock finns numera open source alternativ, det vill säga öppna programvaror, för beräkningar och simuleringar. Ett av de några är gratisprogramvaran Scilab/Xcos (SX). SX utvecklas av Scilab Enterprises. Scilab Enterprises är ett konsortium med företag som medlemmar.

Då CPAC Systems vill undersöka potentialen hos programvaran SX syftar detta projekt till att undersöka SX genom olika testmoment. SX är ett helt öppet system, vilket betyder att både kod och lagringsformat kan studeras vid behov. Denna öppenhet skapar därför större självständighet och det är anledningen till att CPAC Systems vill undersöka SX.

Undersökningen av Scilab har skett i både Linux och Windows och i detta projekt utfördes det med hjälp av ett testobjekt vilket var en servomotor. Testet utfördes i delmoment som beskrivs här nedan.

1. Det första momentet var att göra en plantmodell av ratt, servomotor och styrning. I SX finns möjlighet att konstruera icke kausala modeller med hjälp av Modelicakod.
2. Det andra momentet en reglermodell med plantmodellen inbyggd vilket kallas att utföra en Software in the loop. Det som reglerades var vridmomentet ut ur servomotorn. Där sker förstärkningen med en så kallad map-styrning vilket är en 2D-kurva, xy-axel, där vridmomentutsignalen ( $y$ ) beror på vridmomentinsignalen ( $x$ ).
3. Det tredje momentet var att beräkna värden till modellen och simulera den.
4. Det fjärde momentet innebar ett försök att upprätta en kommunikation med den fysiska servomotorn via CAN.

Slutligen ger detta som resultat att Scilabs Xcos går att använda för modellbaserad design i framtiden, men än så länge brister programmet Scilab i dokumentationen. Undersökningen har dock visat att SX har potential.

## Abstract

Today CPAC System uses the program Matlab/Simulink (MS) for calculations and simulation. MS is owned by Mathworks. A license for MS is not for free. That is why it is interesting to look at other programs that could replace MS. One of these is Scilab/Xcos (SX). It is open source and for free. SX is developed by Scilab Enterprises which is a consortium.

What CPAC Systems wants is to look at the potential of this program SX, when it comes to model based design. SX is a completely open system where the format for both code and storage can be looked at when needed. The reason for CPAC Systems to investigate SX is this advantage that creates a bigger independency.

The benchmarking has been done both in Linux and Windows and with a specific test object, a servo motor. The benchmarking was done in four different parts:

- 1 Part one was to create a plant model of the steering wheel, servo motor and steering. In SX it is possible to make acausal models by using the language Modelica.
- 2 Part two, a simulation model with a built in plant model and a map controller, a curve in 2D, with the steering wheel torque as a variable in (x) and the wanted machine torque out (y).
- 3 Part three was to calculate the values to the model and simulate it.
- 4 In part four an attempt to communicate with the physical machine with CAN was done.

The final conclusion is that Xcos could be used for model based design in the future. There are still some difficulties finding documentation. This investigation has shown that SX has potential.

The report is written in Swedish.

Keywords: Scilab, Xcos, Modelica, Model Based Design

# Innehållsförteckning

Beteckningar.....	1
1. Inledning.....	2
1.1 Bakgrund.....	2
1.2 Syfte.....	2
1.3 Avgränsningar.....	2
1.4 Precisering av frågeställningen.....	2
2. Teknisk bakgrund.....	3
2.1 Språket Modelica.....	3
2.2 Programvaran Scilab.....	4
2.2.1 SciNotes.....	4
2.2.2 ATOMS.....	5
2.2.3 Xcos.....	5
2.3 Open source.....	6
2.4 CAN.....	6
2.5 Testobjektet.....	7
2.6 Software in the loop/Hardware in the loop (SIL/HIL).....	7
3. Metod.....	8
3.1 Bygga av plantmodell.....	8
3.2 Bygga av reglermodell.....	9
3.3 Värden och signaler.....	10
3.3.1 Värden.....	10
3.3.2 Signal in i Lookup_f.....	11
3.4 Simuleringen.....	11
3.5 CAN.....	12
3.6 Servomotoruppkoppling.....	12
3.7 Initiering av PCAN-dongel.....	13
4. Resultat och slutsatser.....	15
5. Diskussion och kommentarer.....	16
5.1 Om programvaran.....	16
5.2 Tutorials.....	16
5.3 Om plantmodellen.....	16
5.4 Om reglermodellen.....	16
5.5 Om simuleringen.....	16

5.6 Om servomotorn .....	17
5.7 Om antagna värden.....	17
5.8 Om icke utförda moment.....	17
5.8.1 Nedladdning och test .....	17
5.8.2 Arduino.....	17
5.8.3 Generera C-kod i Xcos .....	17
5.8.4 ScicosPowerLab.....	17
5.9 Rekommendationer till fortsatt arbete.....	18
6. Referenser.....	19

## Beteckningar

ATOMS	AutomaTic mODule Management for Scilab
CAN	Controller Area Network
GCC	Gnu Compiler Collection
HIL	Hardware in the loop
<i>J</i>	Tröghetsmoment
<i>Jratt</i>	Tröghetsmoment ratt
<i>Jrotor</i>	Tröghetsmoment rotor
<i>Jstyrväxel</i>	Tröghetsmoment styrväxel
MS	Matlab/Simulink
<i>N</i>	Newton
<i>Nm</i>	Newtonmeter
<i>N/grad</i>	Newton per grad
PC	Personal computer
SIL	Software in the loop
SX	Scilab/Xcos
<i>T<sub>stav</sub></i>	Styvheten i staven mellan ratt och rotor
USB	Universal Serial Bus



# 1. Inledning

## 1.1 Bakgrund

Det finns ett flertal olika matematik- och simuleringsprogram. Ett av de mest kända är Matlab/Simulink (MS), vilket CPAC Systems använder sig av i dagsläget. Det fungerar bra men nackdelarna är att det kostar licenspengar samt att CPAC Systems inte styr över hela utvecklingskedjan till hundra procent. Därför vill CPAC Systems undersöka potentialen i Scilab för modellbaserad design. Scilab är ett gratis open source-program.

## 1.2 Syfte

Syftet är att undersöka om det går att använda Scilabs Xcos vid modellbaserad design. För företaget skulle det innebära större självständighet att arbeta i ett program med öppet format då kod och lagringsformat kan studeras vid behov. En annan fördel är att Scilab är gratis.

## 1.3 Avgränsningar

Det är Scilabs modul Xcos som ligger i focus. I Xcos arbetas det enbart med ett speciellt testobjekt, en servomotor. Vid kommunikationen mellan dator och testobjekt används CAN-kommunikation.

## 1.4 Precisering av frågeställningen

- Hur ska ratt, servomotor och styrning modelleras samt simuleras i Xcos?
- Finns det färdiga motormodeller som kan användas?
- Hur ska kommunikation ske mellan program och fysisk servomotor?
- Kan en fungerande C-kod genereras av Xcos?

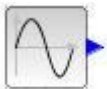
## 2. Teknisk bakgrund

För att utföra uppgiften behövdes åtkomst till programvarorna, kunskap om varje dels användningsområde och hur de olika delarna samarbetar med varandra.

### 2.1 Språket Modelica

Modelica används bland annat i kontrollsystem. Modelicaspråket är utvecklat av den ideella organisationen Modelica Association för att simulera fysiska system. Det bygger på att i en uppbyggd modell är det inga pilar mellan blocken, och visar på att signalerna i modellen inte har en specifik riktning. Den visar till exempel att en uppskissad linje mellan blocken istället utgör en axel eller ett rör. En fördel med Modelica är att det i en och samma modell går det att kombinera olika system, så som mekaniska, hydrauliska, termiska och elektriska [3].

De Modelicablock som används i detta arbete är följande:



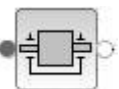
Figur 2.1. Genererar en sinussignal. Inställbara parametrar är amplitud och frekvens.



Figur 2.2. Skapar ett vridmoment på en axel. Vridmomentets storlek är direkt beroende av insignalen, som till exempel kan vara en sinussignal.



Figur 2.3. Simulerar en fjäder. Inställbara parametrar är styvhet [Nm/rad] och fjädervinkel i opåverkat läge [rad].



Figur 2.4. Simulerar ett tröghetsmoment [ $\text{kgm}^2$ ].



Figur 2.5. Används för att mäta vinkelskillnaden. Till exempel mellan vardera ände av en fjäder.



Figur 2.6. Anpassar Modelicasignal till Xcosignal.



Figur 2.7. Anpassar Xcossignal till Modelicasignal.

## 2.2 Programvaran Scilab

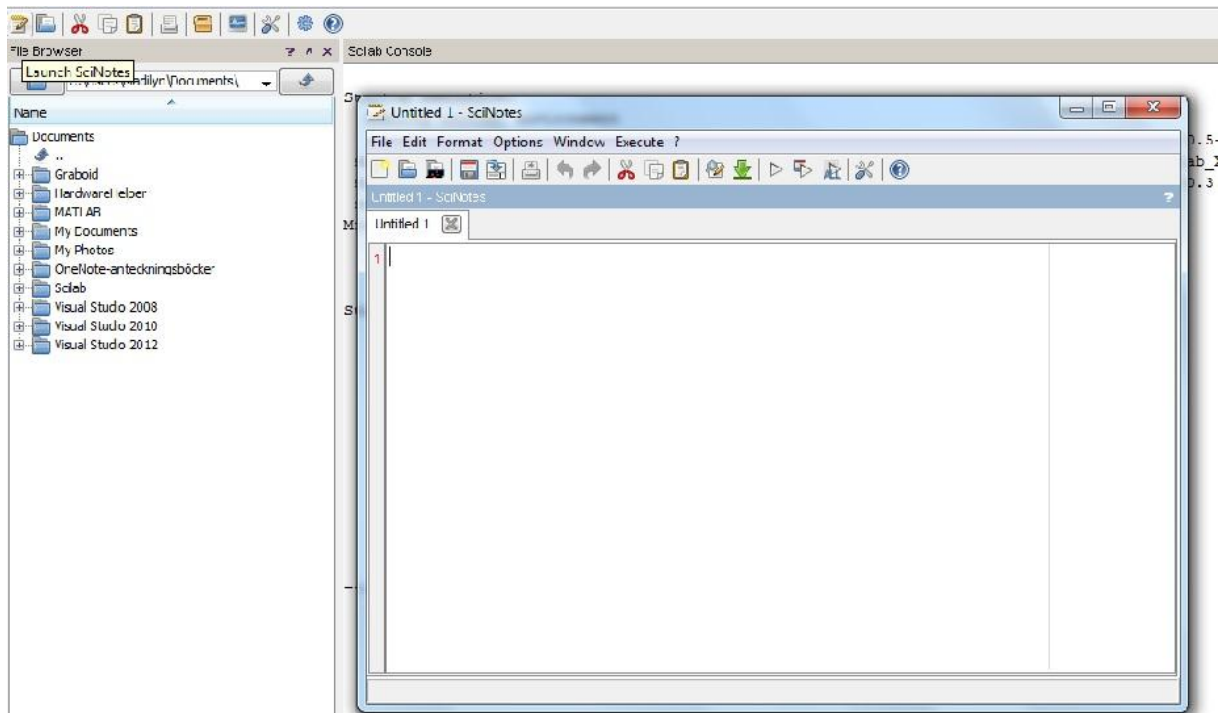
Avsnittet tar upp Scilab och de olika inbäddade paketen och verktyg det går att använda sig av.

Scilab är ett gratis open source program för numeriska beräkningar och kan jämföras med programvaran Matlab. Scilab kan enklast laddas ner via hemsidan, [1], till antingen Windows, Linux eller MAC. I detta projekt körs Scilab i Linux och Windows.

### 2.2.1 SciNotes

I Scilab finns en möjlighet att skriva kod i den inbyggda texteditorn, kallad SciNotes. Den fungerar som Matlabs editor där man bland annat kan skriva funktioner [1].

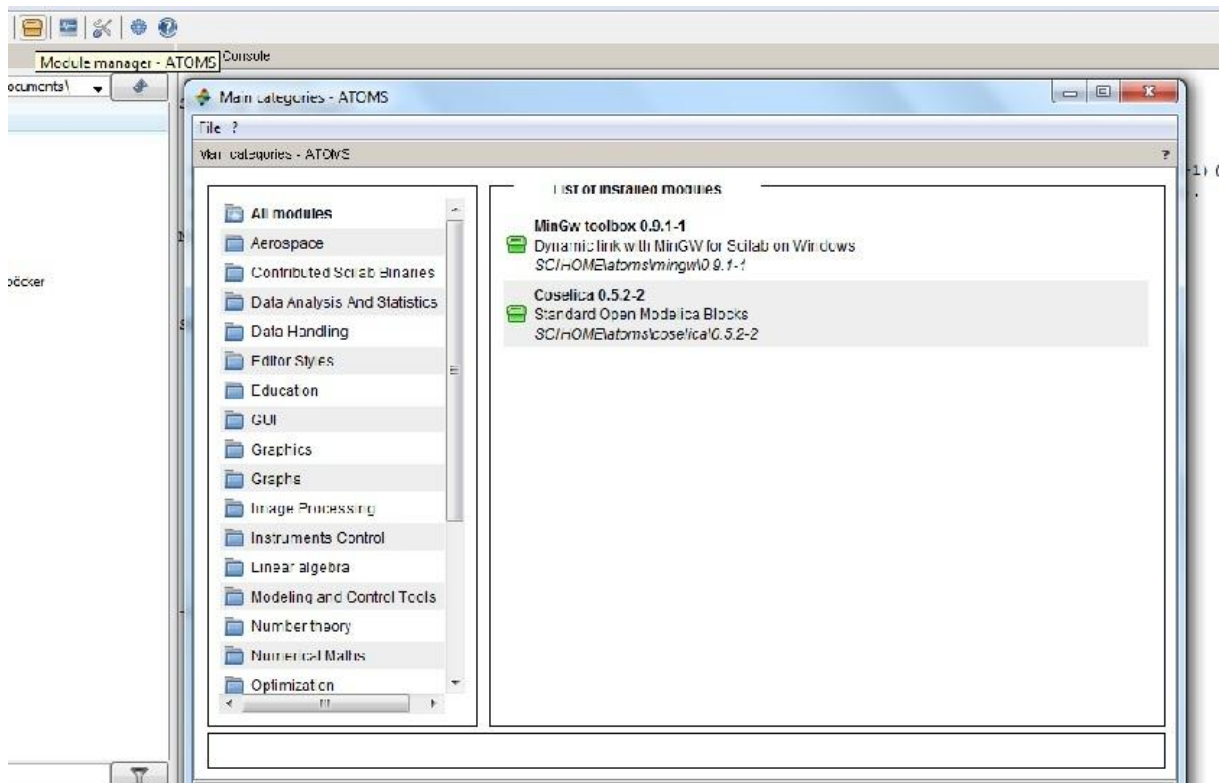
– Användes dock inte i projektet.



Figur 2.8. Scinotes.

## 2.2.2 ATOMS

I ATOMS kan det laddas ner och installeras jättemånga olika verktyg och tillbehör, som inte är med i standardinstallationen av Scilab [9]. Dessa är gratis.



Figur 2.9. ATOMS.

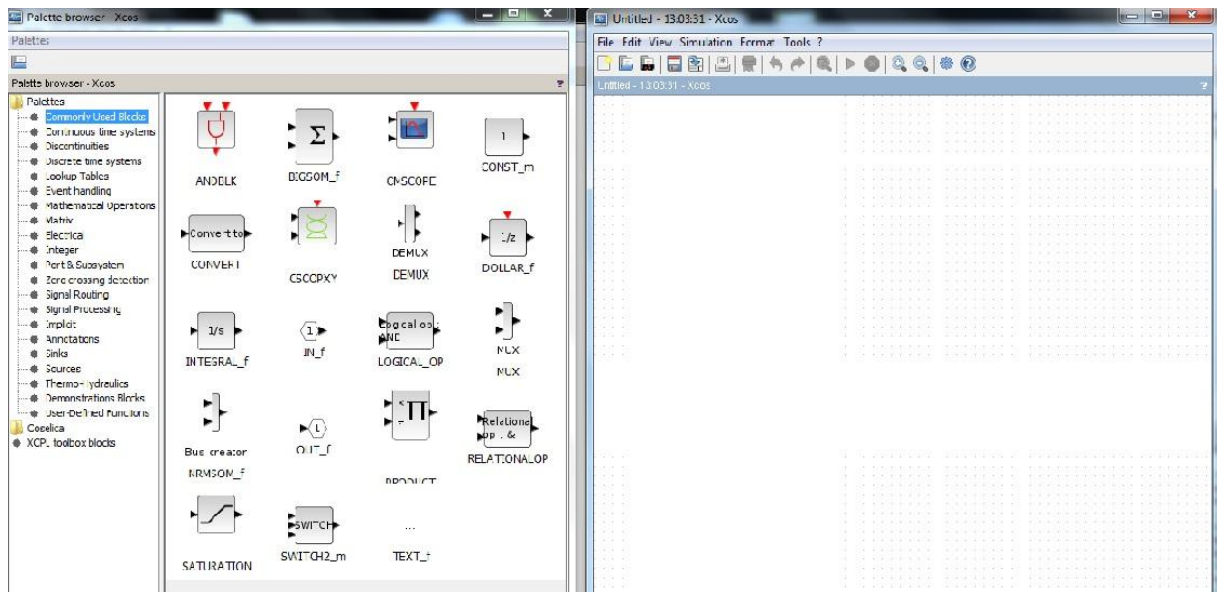
Via ATOMS installerades de verktyg som behövdes till Xcos i projektet.

Installerade verktyg:

- Mingw - Länkar C-biblioteket, som finns i GCC, till Scilab för Windowsversionen [10].
- Coselica - Modelicablock, både för Linux och för Windows [10].

## 2.2.3 Xcos

Xcos är en grafisk editor inbäddad i Scilab och fungerar som Matlabs grafiska editor Simulink. Här ställer man upp plantmodellen och reglermodellen i Xcoseditorn. Blocken finns att hämta ur Xcos blockbibliotek, Palette browser, och precis som i Simulink finns det block för bland annat elektriska-, flödes-, termiska-, och mekaniska system. Programmet utför kompilering och simulerar uppställningen vilket kan visas i grafiska fönster. Man kan även skapa egna block i både Modelica-form och signalform och därmed anpassa modellen efter eget behov och på så sätt bygga ut blockbiblioteket.



Figur 2.10. Palette browser och Xcos editor.

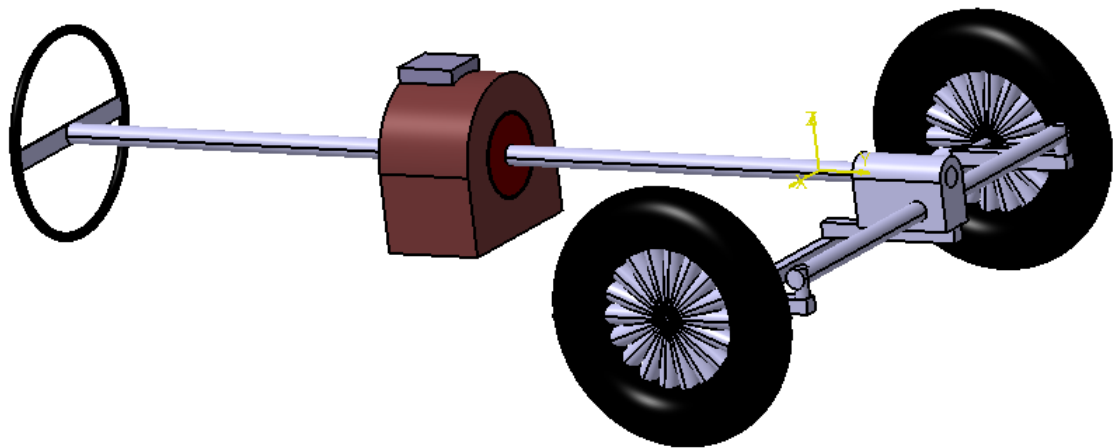
## 2.3 Open source

Open source, det vill säga öppen programvara eller öppen källkod, innebär att användaren får tillgång till information om vad programmet gör, fri tillgång till koden. Att ändra på en produkt som är av open source-typ är tillåtet och ger på så sätt större flexibilitet för användaren. Det innebär att open source inte ägs av någon och ingen innehar upphovsrätten till programvaran. Vid eventuella fel i varan behövs det inte anlitas en specifik leverantör för reparation, till exempel som Mathworks till Matlab, utan kan repareras av den som har kunskapen. Ett program av denna typ utvecklas snabbt då fler har möjlighet att påverka den. Fler funktioner dyker upp och buggar fixas snabbt efterhand [4].

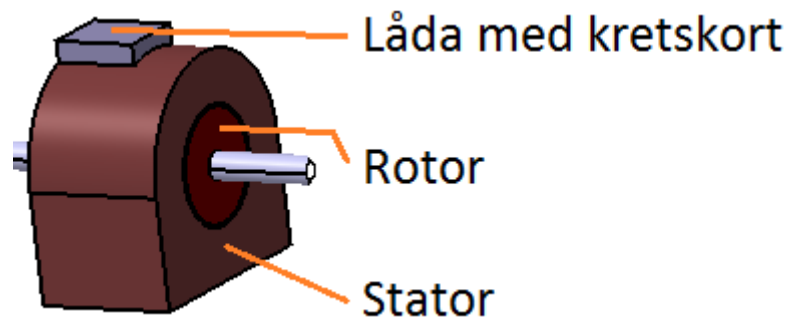
## 2.4 CAN

CAN är ett nätverk som klarar upp till 40 meter nätlängd och används ofta i fordonsindustrin. Man talar om att noder skickar meddelanden på CAN-bussen med en hierarkisk prioritering [14, 15].

## 2.5 Testobjektet



Figur 2.11. Testobjektet: Ratt, servomotor och styrning.



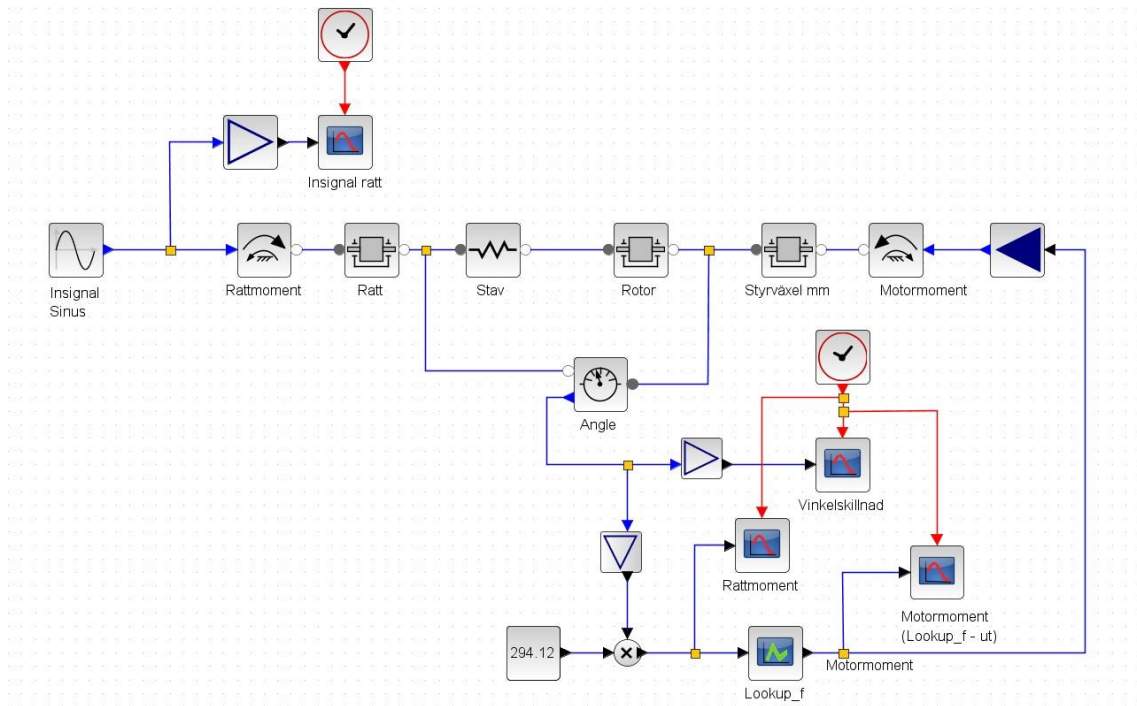
Figur 2.12. Servomotorn.

Testobjektets delar, som modellen i detta projekt beskriver, består av ratt, servomotor och styrning. Se figur 2.11. Den centrala delen är servomotorn. Se figur 2.12. Rotorn i servomotorn är sammankopplad med ratten via en stav. Vinkelskillnaden i denna stav används för att ta reda på vilket vridmoment ratten bidrar med. Detta vridmoment förstärks till styrningen med hjälp av servomotorn.

## 2.6 Software in the loop/Hardware in the loop (SIL/HIL)

SIL innebär att endast modeller i datorn används vid simulering. HIL innebär att reglermodellen kopplas upp mot den verkliga fysiska enheten [5, 6].

### 3. Metod

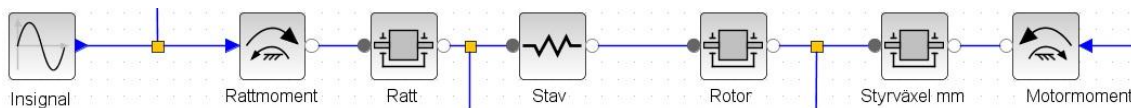


Figur 3.1. Simuleringsmodell.

#### 3.1 Bygge av plantmodell

Plantmodellen byggs i Xcos med de Modelicablock som finns i toolboxen Coselica. Modellen representerar en förenklad version av ratt, servomotor och styrning. En sinussignal simulerar ratt rörelser omväxlande höger och vänster. Sinussignalens amplitud representerar vridmomentet som läggs på ratten. Ett visst vridmoment går då in från rattstyrningen och det representeras av momentblocket *Rattmoment*.

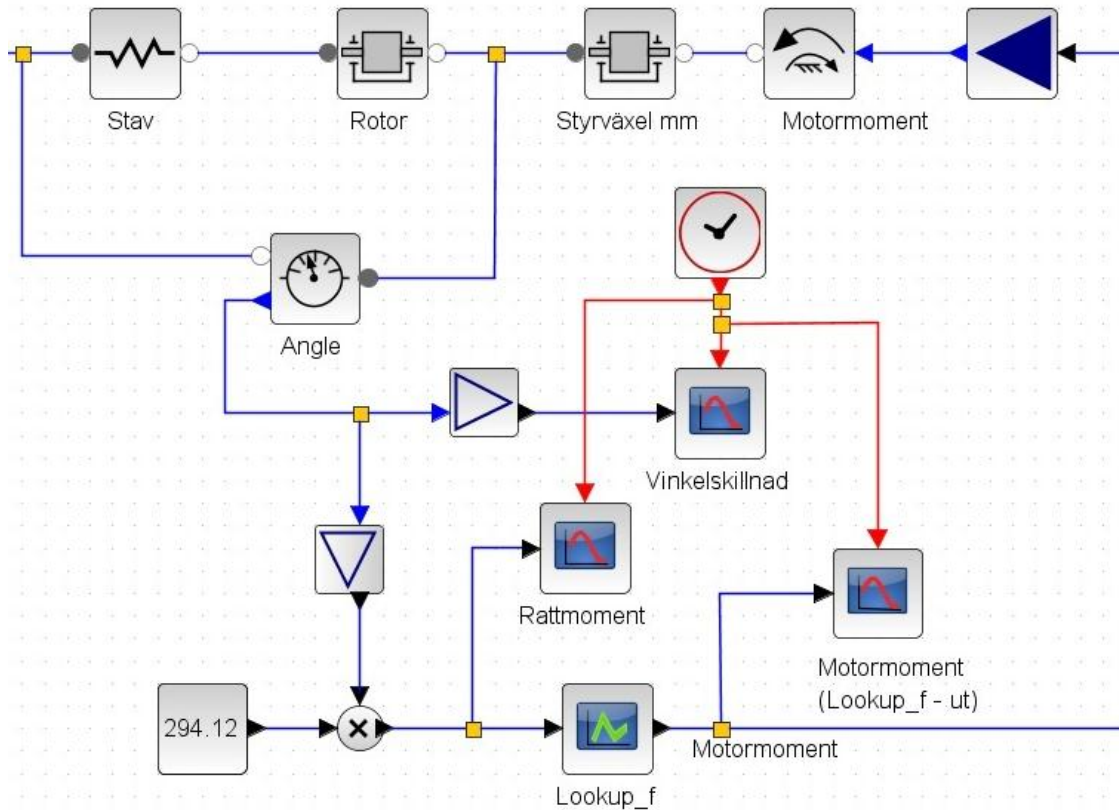
Vridmomentet från hjul och styrväxel representeras av blocket *Motormoment*, se figur 3.2. Den komponent som ger moment från ratt till servomotor får tolkas som en fjäder och visas med ett fjäderblock med viss styvhet, se *Stav* i figur 3.2. Trögheter i komponenter illustreras med inertialblock, se *Ratt*, *Rotor* och *Styrväxel mm* i figur 3.2.



Figur 3.2. Plantmodellen. En del av simuleringsmodellen.

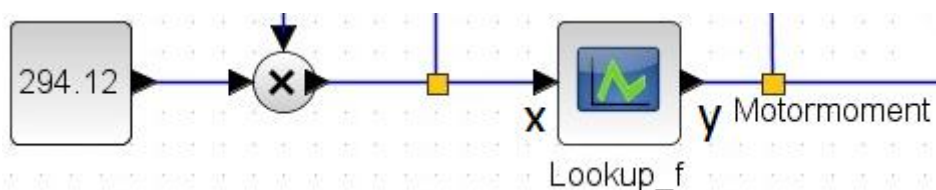
## 3.2 Bygge av reglermodell

För att styra plantmodellen byggdes en reglering. Se figur 3.3.



Figur 3.3. Reglering, en del av simuleringssmodellen.

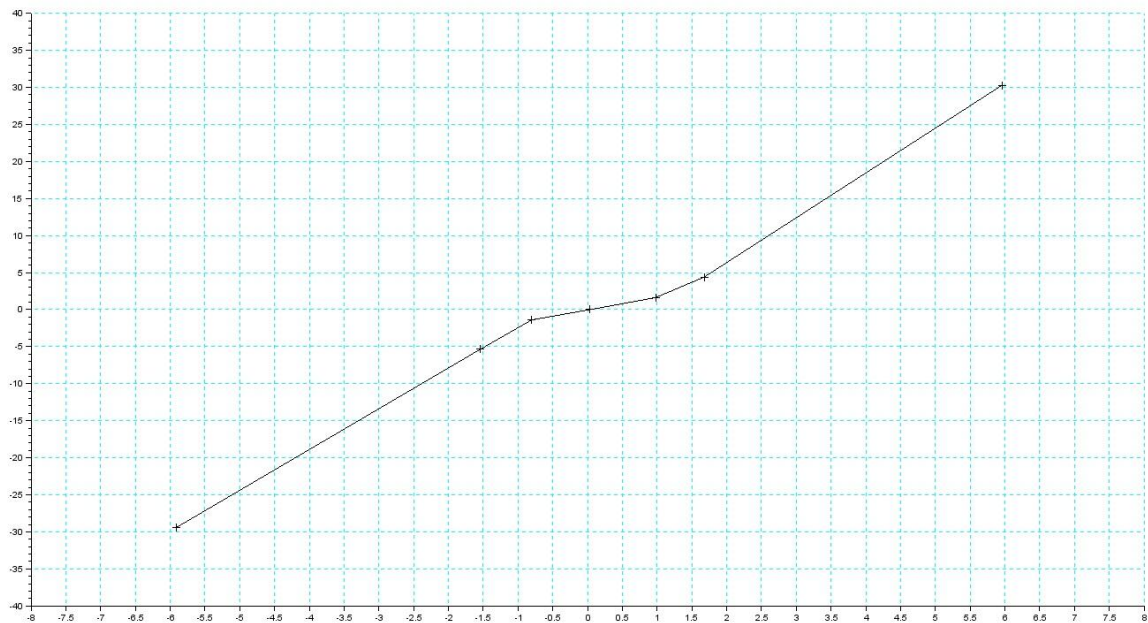
Istället för P-, PI-, eller PID-reglering så styrdes förstärkningsenheten från ratt till servomotor med ett Xcos-block, *Lookup\_f*, där värden plottats ut på en 2D-kurva vilket är en xy-axel. Ett visst värde x skickades in i blocket, där förstärktes den, och y sändes ut. Detta kallas att reglera med map-modell. X-värdet representerar vridmomentet över staven och y-värdet representerar motormomentet. Se figur 3.4.



Figur 3.4. *Lookup\_f*, förstärkningen i regleringen.

Ett x-värde nära noll ska ge ett litet y-värde ut, då det vid mindre rörelser på ratten önskas små utslag till hjulen, men vid större x-värden önskas större y-värden ut.





Figur 3.5. *Lookup\_f*, förstärkningskurvan.

Då motormodellen byggdes med hjälp av Modelica och reglermodellen med Xcos-block så behövde signalerna mellan servomotorn och regleringen anpassas till varandra. Detta utfördes med signalkonverterande Modelica-block.

### 3.3 Värden och signaler

#### 3.3.1 Värden

Antaganden och beräkningar har gett följande värden av tröghetsmoment och styvhet till de olika delarna [Bilaga Beräkningar].

Tröghetsmoment:

$$J_{ratt} = 0,38 \text{ kgm}^2$$

$$J_{rotor} = 0,0017 \text{ kgm}^2$$

$$J_{styrväxel} = 38 \text{ kgm}^2$$

Styvhet:

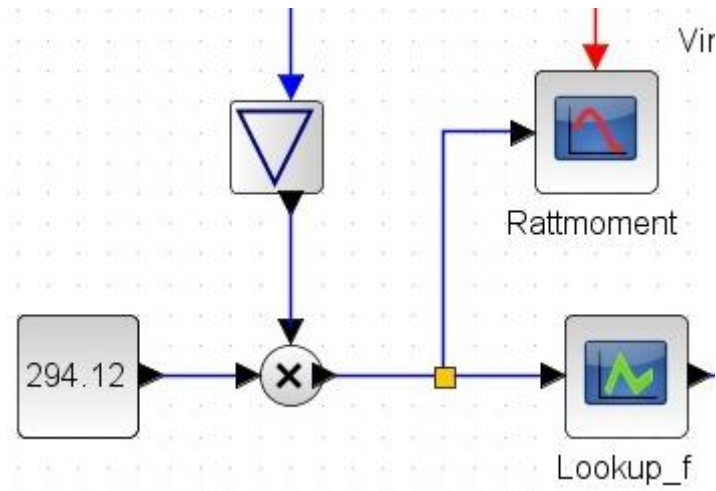
$$T_{stav} = 5 \text{ Nm/grad} = 294,12 \text{ Nm/rad}$$

Max/min rattmoment, insignal in i *Lookup\_f* = +/- 5 Nm.

Max/min motormoment, utsignal från *Lookup\_f* = +/- 25 Nm.

### 3.3.2 Signal in i Lookup\_f

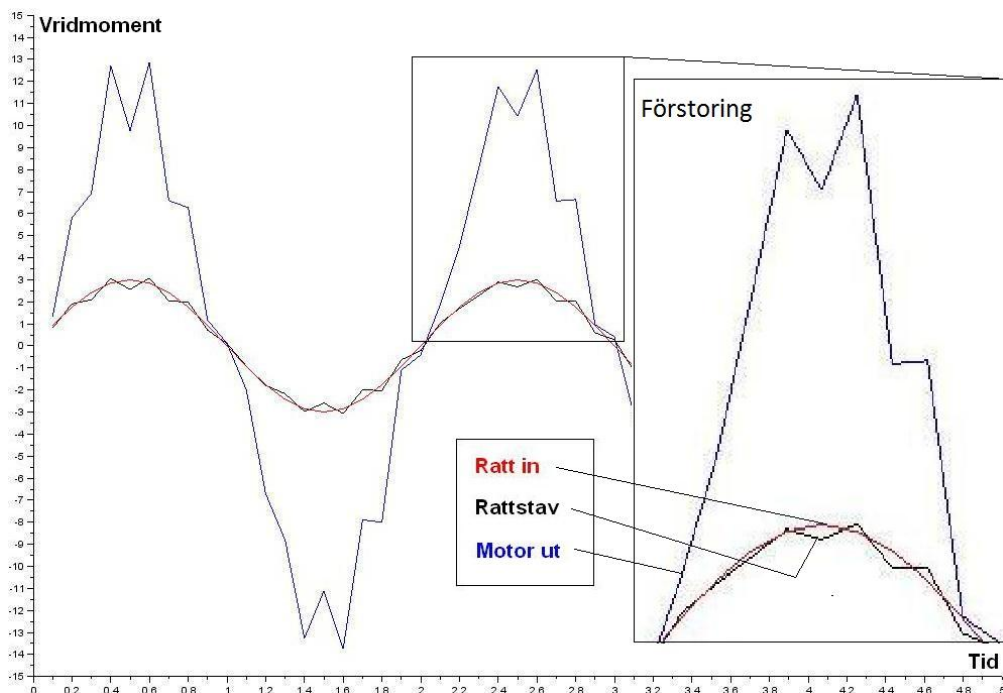
Signalen som går in i *Lookup\_f* är vridmomentet över styrstaven. Detta vridmoment fås fram genom att multiplicera vinkelskillnaden över styrstaven med stavens styvhet.



Figur 3.6. Insignal i *Lookup\_f*

### 3.4 Simuleringen

Vid simulering av modellen med ovanstående värden genererades nedanstående kurvor, se figur 3.6. *Ratt in* är en ren sinuskurva. Den kurvan visar de simulerade rörelserna omväxlande höger och vänster som läggs på ratten. Sinuskurvans amplitud representerar vridmomentet som läggs på ratten. Vridmomentet, som påverkar rattstaven, visas av *Rattstav* och är den kantiga kurva som följer den släta sinuskurvan. *Motor ut* är den stora kantiga kurvan. Den visar det vridmoment som servomotorn ger.



Figur 3.7. Simuleringskurvor.

### 3.5 CAN

Meningen var att kommunikationen mellan programvara och servomotor skulle ske via ett CAN-nätverk och en USB-dongel för att fullgöra HIL-uppkopplingen. Denna kopplas upp till en mikroprocessor i servomotorn.



*Figur 3.8. PCAN-dongel.*

### 3.6 Servomotoruppkoppling

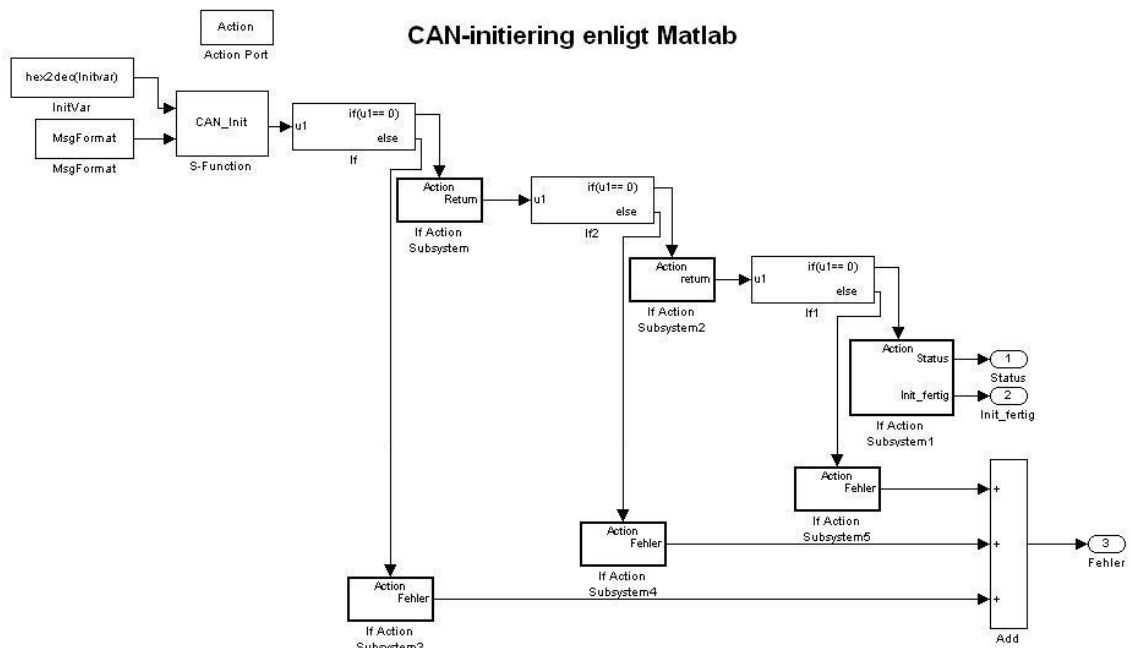
För kommunikation mellan PC och servomotor behövdes en anslutning till servomotorns mikroprocessor upprättas. PCAN-dongeln kopplas från PC:ns USB-port till en dsub-kontakt som fick lödas på servomotorns kretskort. Se figur 3.9. Vid HIL-uppkoppling ska servomotorn kunna styras från en dator.



*Figur 3.9. D-sub.*

### 3.7 Initiering av PCAN-dongel

Tanken var att härma Matlabs initieringslösning. Se figur 3.10.

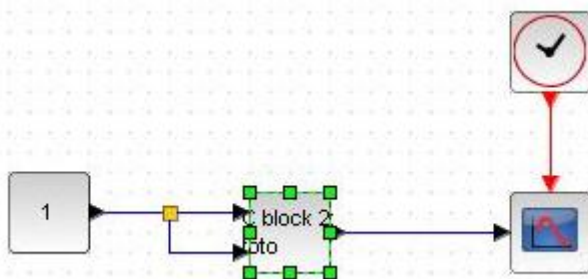


Figur 3.10. Skärmdump från Matlab.

En sekvens av C-kod skrevs i ett lämpligt Xcos-block för PCAN-kommunikation, men blev aldrig avslutat. Målet var att använda CBLOCK i Xcos för att med hjälp av C-kod koppla samman Xcos med servomotorn. Den fysiska kopplingen skulle vara en PCAN-dongel [Bilaga C-kod].

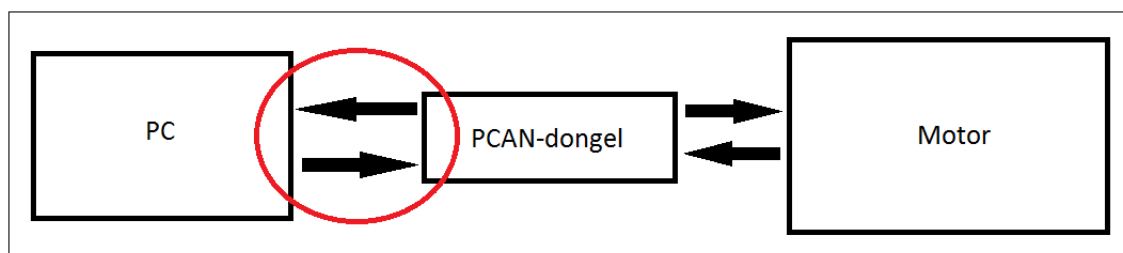
Det som behövdes göras med C-kod:

1. Anrop av en initieringsfunktion.
2. Sända/Ta emot CAN-meddelanden med hjälp av färdiga funktioner.



Figur 3.11. C\_block2.

Denna kommunikation var problemet. Se figur 3.12.



*Figur 3.12. Kommunikation mellan PC och servomotor.*

## 4. Resultat och slutsatser

Under projektets gång upptäcktes det att en stor nackdel med Scilab/Xcos är bristen på dokumentation. Informationen hämtades ofta på diverse forum- och wiki-sidor. I några fall fanns dokumentationen som behövdes till SX på Mathworks produkter.

Resultatet visar att det i Scilabs Xcos går att skapa en plant- och reglermodell, samt simulera och rita ut kurvor.

Vid arbetet med CAN-kommunikationen via PCAN-dongeln var det enklare att arbeta i Scilabs Windowsversion, jämfört med Linux. Därför att drivrutiner och testprogram var förberett för att enkelt kunna installeras i Windows.

Att en kommunikation mellan program och hårdvara inte lyckades upprättas gör att denna del ej går att dra en slutsats om.

Undersökningen har visat att SX har potential.

## 5. Diskussion och kommentarer

### 5.1 Om programvaran

SX är en programvara utvecklad av Scilab Enterprises och som byggs på efterhand som behoven uppstår. Vid några tillfällen uppstod problem. Till exempel att ett verktyg inte startar eller att modeller slutar fungera efter en programuppdatering. En del av dessa problem kunde lösas efter en buggrapportering eller i vissa fall via forum.

Det finns inte dokumentation och beskrivningar i tillräcklig utsträckning över de olika simuleringsblocken, utan man hänvisas till dokumentation hos Simulink.

Eftersom Xcos tidigare namn var Scicos, så har detta namn också använts vid sökning efter information.

### 5.2 Tutorials

I projektets inledande fas arbetades det med olika tutorials vid inläringen av programmet. Dessa finns på Openeering [11].

### 5.3 Om plantmodellen

Först gjordes försök att konstruera plantmodellen genom att utgå ifrån ett redan färdigt motorblock, men på grund av bristande dokumentation om det färdiga blocket byggdes plantmodellen från grunden i Xcos. Plantmodellen blev då förenklad men representerade ändå maskinens huvudbeståndsdelar.

### 5.4 Om reglermodellen

Reglermodellen är enkelt konstruerad och styrs av förstärkningen i blocket *Lookup\_f*, detta för att åstadkomma map-styrningen. Tanken var att göra en större reglermodell med minst fem stycken map:ar som skulle styra servomotorn, men tidsbrist ledde till att fokus lades på att utföra de alla delmomenten och få en signal skickad till servomotorn istället.

### 5.5 Om simuleringen

Simuleringskurvorna hade blivit bättre om en dämpare hade lagts till i modellen.

## 5.6 Om servomotorn

De motorvärden som användes innefattar endast det maximala vridmomentet servomotorn är inställd på att generera samt uppskattad vikt och rotordiameter.

## 5.7 Om antagna värden

Enligt vår handledare på CPAC Systems så sägs att normalt styrmoment är 5 Nm och maximum styrmoment som förare orkar hålla emot om personen är beredd är 25 Nm. Det saknas källor för att styrka detta, men det är därför värdena på map:ens maximum- och minimumvärden är +/- 30 Nm.

## 5.8 Om icke utförda moment

### 5.8.1 Nedladdning och test

Meningen var att reglermodellen skulle kodgenereras, kompileras och därefter laddas ner i befintlig controller i servomotorn.

### 5.8.2 Arduino

Arduino är en fysisk datorplattform som också är i form av open source. Detta är en annan möjlighet till CAN-kommunikation med servomotorn. Det finns speciella kretskort att koppla till datorns USB-port. I detta fall skulle Arduino UNO tillsammans med ett CAN-kort kunna användas. I Xcos går det att få tillgång till redan färdiga kommunikationsblock för Arduino. Toolboxen Arduino finns att installera i ATOMS.

### 5.8.3 Generera C-kod i Xcos

I Xcos går det att generera C-kod av en modell med endast en knapptryckning. Då detta sker genereras sex filer. En av dessa filer slutar med *.standalone.c*. Troligtvis är det den filtypen som används till inbäddade system. Detta undersöktes aldrig, på grund av tidsbrist.

### 5.8.4 ScicosPowerLab

I ATOMS finns ScicosPowerLab. Denna toolbox innehåller färdiga motormodeller. En av dessa undersöktes i hopp om att det var något som gick att använda. Det visade sig att det saknas relevant information och att den är kopierad från Matlab. Så på grund av detta bestämdes det att en enklare modell av servomotorn skulle byggas.



## 5.9 Rekommendationer till fortsatt arbete

Tanken med projektet var att reglera ett styrservo med en reglermodell skapad i Xcos. En motor- och reglermodell skulle göras i Xcos. Därefter skulle reglermodellen justeras mot motormodellen tills den var mogen att jobba mot den verkliga servomotorn. För att vid testkörning koppla servomotorn till datorn skulle en PCAN-dongel användas. Dongeln skulle gå mellan datorns USB-port och en dsub-koppling fastlödad i servomotorns kretskort.

På grund av tidsbrist gick det inte att upprätta kommunikationen mellan Xcos och PCAN-dongeln, men arbetet har påbörjats. C\_BLOCK i Xcos användes för att med hjälp av C-kod försöka initiera och kommunicera med dongeln.

Det gick att inkludera de filer som antogs behövas för funktionerna till initiering, CAN-sändning och CAN-mottagning.

I detta skede avbröts projektet, så här finns det en möjlighet att ta arbetet vidare. Tanken var att härma Simulink. Det finns C-kod och blockbilder som visar hur deras lösningar ser ut.

I Scilab går det inte att ställa in någon path som pekar på var h-filerna ligger. Dessa får istället läggas i en mapp där liknande filer finns.

## 6. Referenser

### Installerade programvaror

1. Scilab (2013): [www.scilab.org](http://www.scilab.org) (2013-01-07)
2. PCAN (2013): [www.peak-system.com/](http://www.peak-system.com/). <http://www.peaksystem.com/fileadmin/media/linux/index.htm> (2013-01-07)

### Information från internet

3. Modelica and the Modelica Association (2013) : <https://modelica.org/> (2013-01-07)
4. Om Open Source: <http://se.assignme.com/> . <http://se.assignme.com/opensource> (2013-01-07)
5. SIL: [www.autonomie.net](http://www.autonomie.net)  
. [http://www.autonomie.net/references/model\\_based\\_design\\_defs\\_24c.html](http://www.autonomie.net/references/model_based_design_defs_24c.html) (2013-01-07)
6. HIL: [www.autonomie.net](http://www.autonomie.net) .  
[http://www.autonomie.net/references/model\\_based\\_design\\_defs\\_24e.html](http://www.autonomie.net/references/model_based_design_defs_24e.html) (2013-01-07)
7. Arduino: <http://www.arduino.cc/>. <http://www.arduino.cc/en/Guide/Introduction> (2013-01-07)

### Dokumentation över programvaror, buggar och verktyg

8. Scilab 's Bug Tracker: <http://bugzilla.scilab.org/> (2013-01-07)
9. ATOMS - Modulpaket: <http://atoms.scilab.org/> (2013-01-07)
10. ATOMS - How to use: <http://wiki.scilab.org/ATOMS> (2013-01-07)
11. Openeering - a Scilab professional partner: [www.openeering.com](http://www.openeering.com) (2013-01-07)
12. Maplesoft - Technical computing software (Modelica-block):  
<http://www.maplesoft.com/>. [http://www.maplesoft.com/documentation\\_center/online\\_manuals/modelica/Modelica.html#Modelica](http://www.maplesoft.com/documentation_center/online_manuals/modelica/Modelica.html#Modelica) (2013-01-07)
13. Scicos: [http://www.scicos.org/HELP/eng/scicos/whatis\\_scicos.htm](http://www.scicos.org/HELP/eng/scicos/whatis_scicos.htm) (2013-01-07)

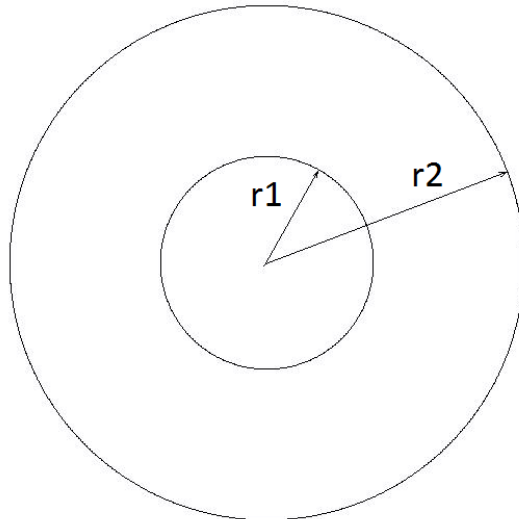
### Böcker

14. CAN1: Bengtsson, L. (2009) *Microcontroller - från assembler till RTOS*. Malmö: Holmbergs
15. CAN2: Johansson, R. och Snedsbøl, R. (2004) *Realtidssystem för högskolans ingenjörutbildningar* Upplaga 8. Göteborg: Göteborgs Mikrovaru

$$\text{Tröghetsmoment} = J = 0,5 \cdot m \cdot (r_1^2 + r_2^2) \text{ [kgm}^2\text{]}$$

m = massa [kg]

r = radie [m]



*Figur Bilaga. Allmän bild som förklarar inner- och ytterdiameter.*

$$J_{ratt} = 0,5 \cdot 5 \cdot (0,25^2 + 0,30^2) = 0,38 \text{ kgm}^2$$

$$J_{rotor} = 0,5 \cdot 2 \cdot (0,01^2 + 0,04^2) = 0,0017 \text{ kgm}^2$$

$$J_{styrväxel \text{ mm}} = k_1 \cdot J_{ratt} = 100 \cdot 0,38 = 38 \text{ kgm}^2.$$

$k_1$  = En konstant, där tröghetsmomentet på Styrväxel mm antogs vara hundra gånger större än rattens tröghetsmoment.

C-kod. Så långt vi kom

```

#include <math.h>
#include <stdlib.h>
#include <scicos_block.h>
#include <windows.h>
#include <PCAN_USB.h>
#include <mex.h>

#define r_IN(n, i)    ((GetRealInPortPtrs(blk, n+1))[(i)])
#define r_OUT(n, i)  ((GetRealOutPortPtrs(blk, n+1))[(i)])

// inputs
#define btr          (r_IN(0,0))
#define msg_type     (r_IN(1,0))

// outputs
#define out          (r_OUT(0, 0))

HANDLE can_handle = NULL;

void toto(scicos_block *block,int flag)
{
  if (flag == 4) { /* initialization */
    toto_bloc_init(block,flag);

  } else if(flag == 1) { /* output computation*/
    set_block_error(toto_bloc_outputs(block,flag));
  } else if (flag == 5) { /* ending */
    set_block_error(toto_bloc_ending(block,flag));
  }
}

int toto_bloc_init(scicos_block *block,int flag)
{
  return 0;}

int toto_bloc_outputs(scicos_block *block,int flag)
{
  int ret=0;

  DWORD __stdcall CAN_Init(WORD wBTROBTR1, int CANMsgType);

  block->outptr[0][0]=2;

  return 0;}

int toto_bloc_ending(scicos_block *block,int flag)
{
  return 0;}

```