ISSN 0280–5316
ISRN LUTFD2/TFRT--5466--SE

Modell av strömriktare i SIMULINK

Magnus Johansson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
December 1992

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
December 1992

Document Number
ISRN LUTFD2/TFRT--5466--SE

Author(s)
Magnus Johansson

Supervisor
Rolf Johansson LTH, Johan Galic ABB

Sponsoring organisation

Title and subtitle
Modell av strömriktare i SIMULINK. (Model of a DC-converter in SIMULINK.)

Abstract

ABB Drives AB has developed a computer controlled converter named TYRAK MIDI for speed- and current control of a DC-motor. There is a need to be able to simulate different operation conditions for the converter without having to do it on an actual converter. Development in the PC-industry has led to a number of programs for simulating dynamic systems.

This master thesis project is about developing a model of the current controller in TYRAK MIDI in the simulation program SIMULINK. It also includes a model of a 6-pulse thyristorbridge.

The model of the thyristorbridge takes into account overlop due to inductances on the net. There is also a simple bridge that feed the motor main voltage without taking overlap into account.

SIMULINK is a good program for simulating small dynamic systems. It is very illustrative to graphical be able to construct and simulate a model. When the model becomes large the time to simulate tends to be very long. To simulate 50 ms on a 386/20-processor with the simple thyristor bridge (BRIDGE1) takes about 15 minutes which is acceptable. To simulate 50 ms with the complex bridge (BRIDGE2) takes about 90 minutes, which is too long. To be able to continue the work with the models developed in this project a 486-processor or similar is needed.

The results from the simulations show that the model of the current controller in TYRAK MIDI is performing as wanted. The model of the 6-pulse thyristor bridge with inductive net gives the same overlap as calculated theoretically.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
Swedish

Number of pages
65

Recipient's notes

Security classification

Förord

Med detta examensarbete avslutar jag min civilingenjörsutbildning på elektroteknisk linje vid Lunds Tekniska Högskola.

Arbetet med att konstruera en modell av TYRAK MIDI's strömregulator och en 6-puls tyristorbrygga har utförts under sommaren och hösten 1992 på ABB Drives, avdelning DKU i Västerås.

Denna rapport är ett utdrag ur teknisk rapport ABB Drives AB nr DK 92-046. Anledningen till detta är att rapport nr DK 92-046 innehåller detaljerad information om TYRAK MIDI's strömregulatorn som ej kan vara offentligt enligt regler om sekretess på ABB Drives AB.

Jag vill passa på att tacka min handledare på ABB Drives Johan Galic, som tålmodigt hjälpt mig under projektets gång. Ett tack också till Rolf Johansson på Reglertekniska instutionen vid Lunds Tekniska Höskola.

Västerås 921130

Magnus Johansson

INNEHÅLLSFÖRTECKNING

1. PROJEKTBESKRIVNING

1.1. Bakgrund

Likströmsmotordrifter är mycket vanliga i dagens industri där stora krav på reglerprestanda och överlastförmåga finns. En av anledningarna är förmågan att ge stora moment vid låga varvtal och att momentstyrning av en likströmsmotor är mycket enklare än för en växelströmsmotor. ABB Drives AB har sedan 1960-talet utvecklat och konstruerat strömriktare för likströmsdrifter. Dagens strömriktare, TYRAK MIDI, innehåller u-processorer. Fördelen med att använda u-processor är att det är möjligt att implementera avancerade regulator-algoritmer.

Det finns ett behov av att kunna testa och simulera olika driftsfall utan att behöva göra uppkopplingar i den fysiska verkligheten. Utvecklingen inom PC-industrin har lett fram till att det har utvecklats ett antal simuleringsprogram för simulering av dynamiska system. Dessa simuleringsprogram är till stor hjälp vid utprovning av olika modeller.

1.2. Målsättning

Projektetet innebär att i simuleringsprogrammet SIMULINK ta fram en simuleringsmodell för TYRAK's strömregulator. Modellen skall även innehålla en 6-puls tyristorbrygga, som skall ta hänsyn till överlappning vid kommuteringsförloppet av tyristorerna.

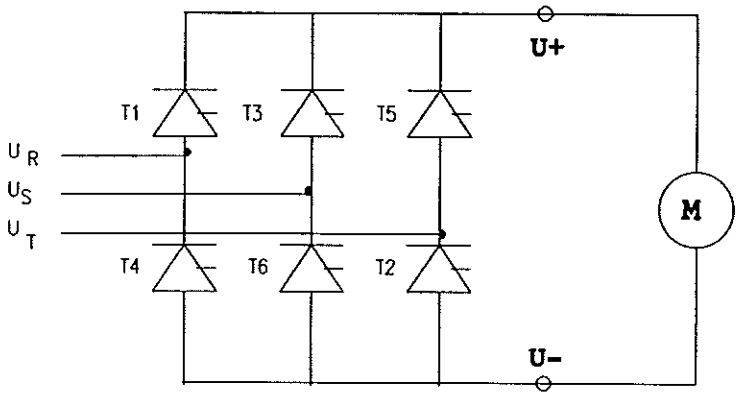
2. STRÖMREGLERING

I kapitel 2.1 beskrivs allmänt hur man kan reglera strömmen till en likströmsmotor och i kapitel 2.2 visas hur TYRAK MIDI's strömregulator är konstruerad.

2.1. Allmänt om strömreglering

Under förutsättning att fältströmmen är konstant är det drivande momentet från en likströmsmotor proportionellt mot strömmen genom den. Genom att styra spänningen till motorn kan strömmen regleras och därmed det drivande momentet hos motorn.

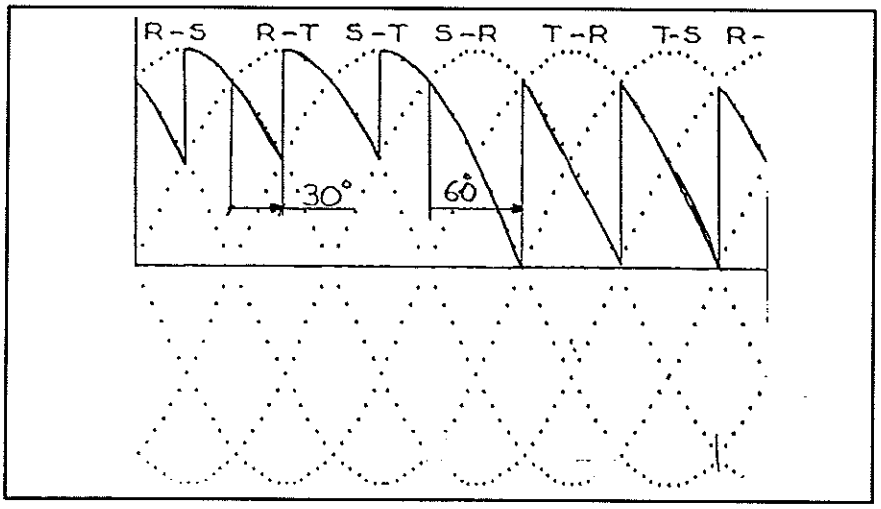
Spänningen kan styras genom att använda en 6-puls brygga med tyristorer. Se figur 1. Tyristorn arbetar så att den förutom framspänning behöver en tändpuls på styret för att leda ström. För en detaljerad beskrivning av tyristorn se kapitel 3.2.2.2 .



Figur 1. 6-puls tyristorbrygga kopplad till motor.

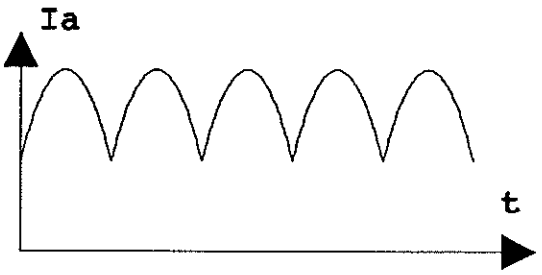
Genom att styra vid vilken tidpunkt tändpulserna skall komma kan man variera när tyristorn skall börja leda och därmed variera spänningen ut från tyristorbryggan. När ett nytt tyristorpar börjar leda sker ett byte av matande huvudspänning, som kallas kommuteringsförlopp. En kommutering kan tidigast ske när det tyristorpar som skall börja leda är framspänt. Med tyristorns egenskap att inte börja leda förrän tändpuls kommer kan man fördröja tiden för kommuteringen. Fasfördröjningen av tändpulsen uttryckt i vinkelmått kallas styrvinkeln

alfa. Alfa definieras lika med noll vid den tidpunkten då tyristorparet blir framspänt.



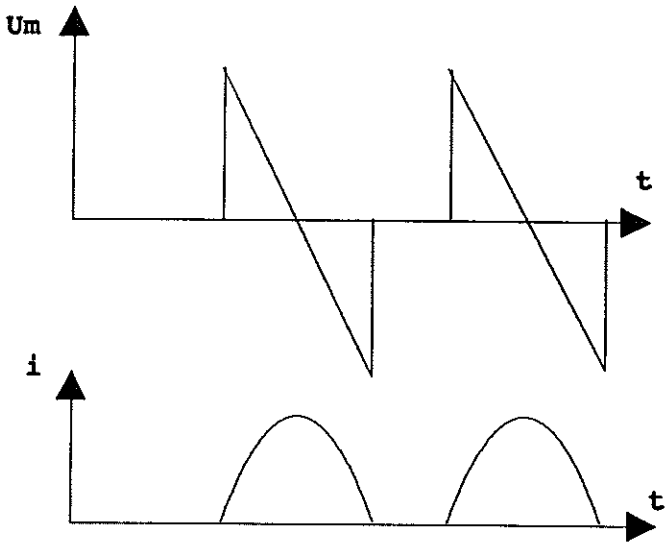
Figur 2. Spänning från tyristorbrygga med alfa=30 grader och alfa=60 grader.

Den spänning som kommer från tyristorbryggan ger upphov till en bågformad ström i motorn, som kallas strömpuls. När strömpulserna hänger ihop utan att gå ner till noll, kallas strömmen kontinuerlig.



Figur 3. Kontinuerlig ström (Ia) i motorn.

Tyristorn kan inte mata negativ ström. När strömmen närmar sig noll spärrar tyristorn för den matande spänningen. Strömmen kommer då att vara noll tills nästa tändpuls kommer. Vid sådan form på strömpulserna kallas strömmen diskontinuerlig.



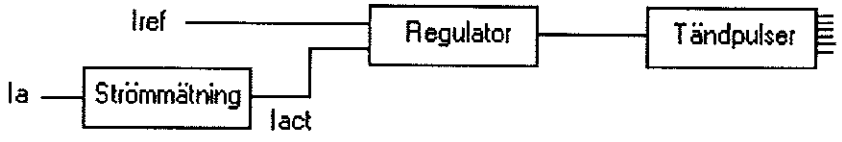
Figur 4. Exempel på diskontinuerlig spänning (Um) och ström (i) vid induktiv last.

2.2. Strömregulatorn i TYRAK MIDI

Detta avsnitt är endast en sammanfattning av TYRAK's strömregulator. För en fullständig redogörelse hänvisas till M Johanssons rapport "The current controller of TYRAK L/MIDI Teknisk Rapport ABB Drives AB nr. DK 89-021".

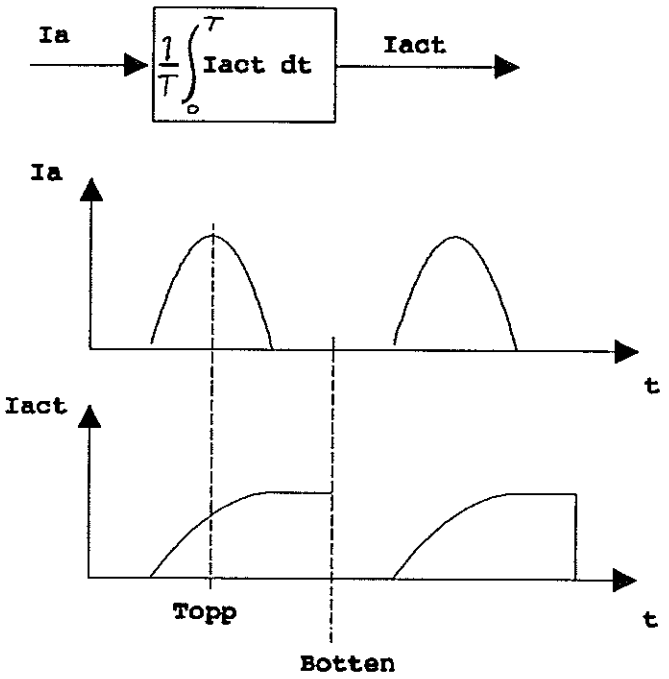
För att uppnå maximal reglersnabbhet är TYRAK's strömregulator avbrottsstyrd, vilket innebär att olika typer av avbrott signaleras till datorn vid vissa lägen på strömpulserna. Datorn utför då beräkningar knutna till dessa lägen.

Strömregulatorn är uppbyggd i grunden av tre block; strömmätning, regulator och tändpulsenhet. De är sammanbundna enligt figur 6.



Figur 6. Strömregulatorns struktur. Iref är strömreferensen, Ia är aktuell motorström och Iact medelvärdet av aktuell motorström.

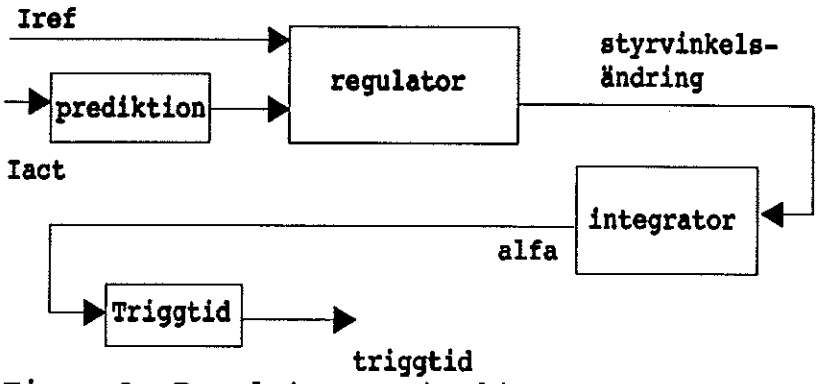
2.2.1. Strömmätning



Figur 7. Medelvärdesbildning av motorström.

Strömmens medelvärde mäts för varje strömpuls för sig genom integrering av aktuell ankarström. I botten, mellan två toppar i strömpulsen, nollställs integratorn för att på nytt integrera nästa strömpuls. Blocket mäter även strömpulsens längd och detekterar när strömmen är noll. Den informationen behövs i regulatorn för beräkning av styrvinkeländringen.

2.2.2. Regulatorn



Figur 8. Regulatorns struktur.

Regulatorn är en polplacerande regulator med prediktion av strömmätningen och arbetar synkroniserat med strömpulserna.

Regulatorn beräknar nödvändig ändring av styrvinkeln för att strömmens medelvärde skall följa referensen. För att undvika dödtid i beräkningarna predikteras strömmens medelvärde. Prediktionen sker utifrån integration av halva strömpulsen, dvs mellan botten- och topp av strömpulsen. Det kan man göra på grund av att strömpulsen är ganska symmetrisk. Hela strömpulsens medelvärde kan beräknas genom att multiplicera integrationen av halva strömpulsen med två. På grund av den asymmetri som finns, orsakad av olinjäritet i huvudspänningen, adderas en korrektionsfaktor till prediktionen så att medelvärdet av strömmen blir riktigt.

Baserat på prediktionen av strömpulsens medelvärde beräknas nödvändig styrvinkeländring så att felet (Iref-Imedel) går mot noll. Tändvinkeln alfa får man genom att addera den beräknade styrvinkels-ändringen med tändvinkeln för föregående puls. Detta ger integralverkan på tändvinkeln alfa. För att förhindra integratoruppvridning är alfa begränsad, så kallad anti windup.

När tändvinkeln alfa beräknats ska tiden för nästa triggtillfälle bestämmas. För att beräkna det måste man veta när i tiden alfa=0 för det tyristorpar som står i tur att börja leda. Det görs genom att synkronisera triggtillfällena med nätets S-fas. Tiden för S-fasen positiva nollgenomgångar registreras. Utifrån den tiden beräknas när i tiden

alfa=0 för varje tyristorpar och därmed vid vilken tid triggning av tyristorparet skall ske för att ge beräknat alfa.

Inställning av regulatorn sker genom att variera två motorspecifika parametrar: CONSTCON och CONSTRL. CONSTCON är ett mått på resistansen och induktansen som finns i motorn. CONSTRL är förhållandet mellan motorns induktans och resistans. Härledning av CONSTCON och CONSTRL återfinns i kapitel 3.2.1. Dessutom finns regulatorparametern IAGAIN som bestämmer var systemets reella dubbelpoler ska placeras.

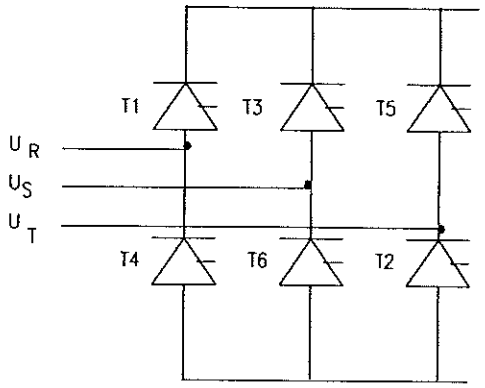
IAGAIN=1.0 ger poler i 0.0
IAGAIN=0.9 ger poler i 0.1
etc

Ju högre värde på IAGAIN desto snabbare reglering.

2.2.3. Tändpulsenheten

Tändpulsenheten har till uppgift att ge de nödvändiga tändpulserna till rätt tyristorpar i tyristorbryggan. Regulatorn ger information om när nästa triggtillfälle och tändpulsenheten bestämmer vilket tyristorpar, som då skall tändas.

Tyristorerna är numrerade från 1 till 6 i den ordning de tänds. Med medurs fasföljd blir ordningen som tyristorerna får tändpulser 1-2, 2-3, 3-4, 4-5, 5-6 och 6-1. Se figur 10.



Figur 10. Numrering av tyristorerna i 6-pulsbrygga.

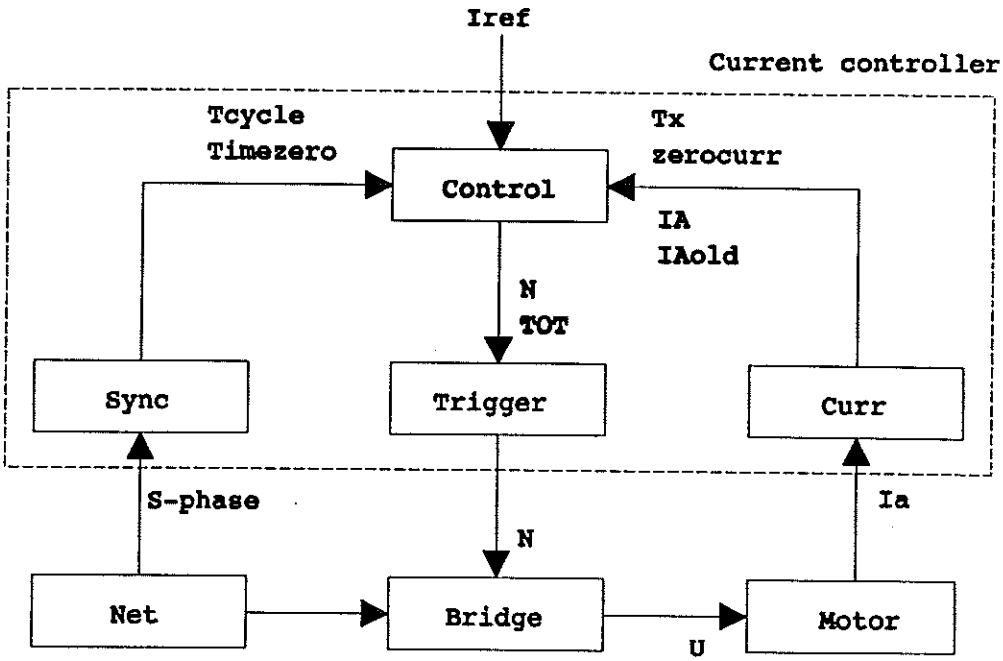
3. SIMULERINGSPROGRAM

TYRAK's strömregulator har implementerats i simuleringsprogrammet SIMULINK. I kapitel 3.1 beskrivs hur modellen av TYRAK's strömregulator är implementerad i SIMULINK. För förståelse av SIMULINK hänvisas till "SIMULINK, A program for simulating dynamic systems".

SIMULINK är ett program för simulering av dynamiska system. Programmet är en utbyggnad av Matlab för Windows. Styrkan med programmet är fram allt att man på ett illustrativt sätt grafiskt kan bygga upp system och simulera dem. I SIMULINK är det enkelt att arbeta med system som inte innehåller så många tillstånd. Men när antalet tillstånd växer tar en simulering avsevärd tid. Att simulera 50 ms på en 386/20-processor med regulatorn och den enkla bryggan (Bridge1) tar ca 15 minuter och med den komplexa bryggan (Bridge2) tar ca 90 minuter, vilket är alldeles för lång tid. Det är även en stor nackdel att SIMULINK inte ger något stöd för implementering av variabel sampeltid. Hur det har lösts beskrivs i kapitel 3.1.

3.1. Implementering av strömregulator

I implementeringen av TYRAK's strömregulator har så långt som möjligt befintlig struktur följts. Den uppdelning i olika funktionsblock som beskrivs i kapitel 2.2 återkommer i implementeringen med i stort sett samma funktioner. Alla signal- och parameternamn som finns i min modell och den riktiga implementeringen överenstämmer med varandra. I figur 11 visas de block som finns i min modell och de viktigaste signalerna. Varje block i modellen är en Matlab-fil.



Figur 11. Datormodell av block och signalflöden.

Viktiga variabler i datormodellen är

Iref	Strömreferens
Timezero	Tid för s-fasens nollgenomgång
Tcycle	Faspänningen periodtid
Ttop	Tidpunkt för topp i strömpuls
Tbot	Tidpunkt mellan två toppar i strömpuls
TOT	Tidpunkt för triggning av tyristorpar
N	Anger vilket tyristorpar som skall triggas
Tx	Strömpulsens längd
Zerocurr	Nollström har inträffat sedan föregående Ttop
IA	Aktuell strömpuls medelvärde
IAold	Föregående strömpuls medelvärde
Ia	Aktuell ström i motorn
U	Spänning från tyristorbryggan

Ttop och Tbot används i min modell som sampeltillfälle för de olika blocken. Dessutom finns sampeltiden TOT som är tidpunkten för nästa triggtillfälle. Alla sampeltider för respektive block lagras som ett tillstånd.
Strömregulatorblockens sampeltillfällen är

Regulatorn	Control	Ttop
Strömmätningen	Curr	Tbot
Tändpulsenheten	Trigger	TOT

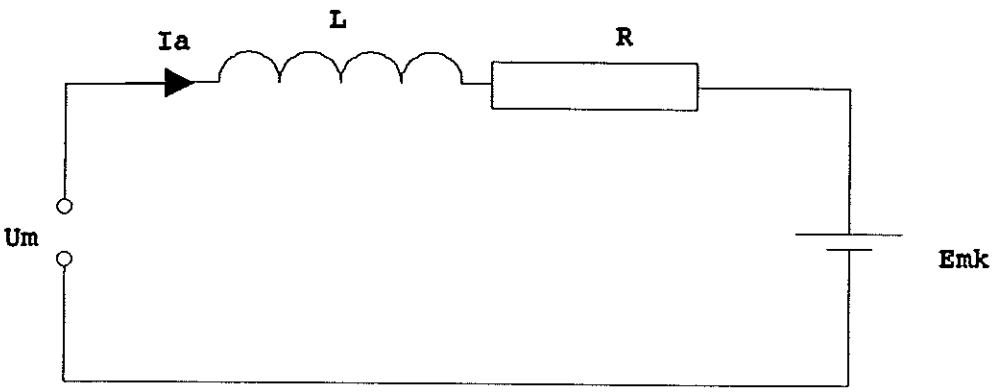
Modulerna Curr och Trigger samplas dessutom vid Ttop för att uppdatera när nästa Tbot respektive TOT inträffar.

Simulink har inget direkt stöd för implementering av variabel sampeltid. Nästa sampeltillfälle uppdateras innan tillstånden uppdateras. På grund av att respektive blocks sampeltid lagras som ett tillstånd, görs efter varje sampeltillfälle ytterligare ett sampel av blocket enbart för att uppdatera tiden för nästa sampel. Det innebär att varje blocks sampeltillfälle ger två samplingar av blocket.

De variabelnamn som nämns i texten är beskrivna i figur 11. För en mer detaljerad beskrivning av varje block hänvisas till bilagan med programlistning där fullständig redovisning finns av hur blocken är implementerade.

3.1.1. Motor

Motorn är inte en del av strömriktaren. Den är den externa enhet som styrs. Modellen för en likströmsmotor består av två differentialekvationer, som beskriver ankarström och varvtal. Ekvationen för ankarström härleds ur följande modell av likströmsmotorn.



Figur 12. Modell av Likströmsmotor.

Då strömmen är kontinuerlig genom motorn gäller

$$\frac{dIa}{dt} = \frac{Um-Emk}{L} - \frac{R}{L} Ia$$

vilket ger sambandet mellan ström och spänning i motorn. Om systemet samplas med sampelintervallet

Ts med styckvis konstant spänning blir z-transformen av motorekvationen

$$I(z)= \frac{Um - Emk}{L} * \frac{1-e^{\frac{-R}{L}} Ts}{\frac{R}{L}(z-e^{\frac{-R}{L}} Ts)}$$

Sätt

$$K1= \frac{1-e^{\frac{-R}{L}} Ts}{R}$$

$$K2= e^{\frac{-R}{L}} Ts$$

så får man

$$I(z)=\frac{K1}{z-K2} (Um-Emk)$$

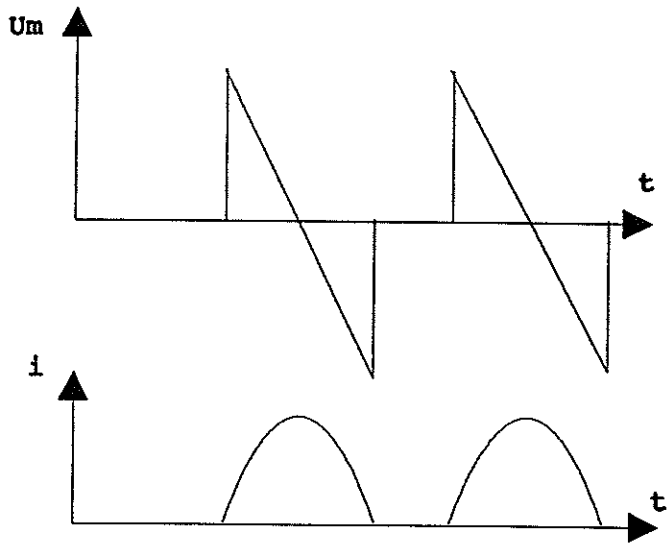
Sätt

$$Kcon = \frac{1}{K1}$$

$$Kl = \frac{K2}{K1}$$

CONSTCON och CONSTL är skalade datorvärden av Kcon respektive Kl och

$$CONSTRL = 1- \frac{CONSTL}{CONSTCON}$$



Figur 13. Diskontinuerlig ström (i) och spänning (Um).

När strömmen (i) till motorn är diskontinuerlig, se figur 13, har man ett helt annat dynamiskt system och därför beter sig motorekvationen annorlunda. Medel värdet av strömmen genom motorn är

$$Imedel = \frac{1}{T}\int_0^{tp} i(t)\,dt = \frac{u}{12\ L\ T^2}\ tp^3$$

Vi kan då lösa ut tp som

$$tp=\sqrt[3]{\frac{12\ L\ T^2}{u}} * \sqrt[3]{Imedel}$$

Ur detta fås kdis som finns härledd i M Johanssons rapport "The current controller of TYRAL L/MIDI Teknisk rapport ABB Drives AB nr. DK 89-021".

$$kdis= \frac{1}{2}\sqrt{\frac{12\ L\ T^2}{u}} * \frac{1}{2\ pi\ f}$$

CONSTDIS är det skalade datorvärdet av kdis.

Ekvationen för motorns varvtal fås ur Newtons andra lag

$$\frac{dw}{dt} = \frac{T}{Jm}$$

där T är det totala momentet.

Eftersom Ia är proportionellt mot det drivande momentet i motorn, och med samtliga signaler normerade blir ekvationen för motorns varvtal

$$\frac{dw}{dt} = \frac{Ia - T}{Jm} * \frac{Tbas}{wbas}$$

där w är varvtalet, Jm tröghetsmomentet och T det belastande vridmomentet.

Motorns emk är proportionell mot varvtalet och vid normerat varvtal fås

Emk=Ebas * w.

För att underlätta simuleringen av olika fall specificerar man en parameter i motormodellen som heter E. Om motorns Emk inte ska vara med i differentialekvationen för motorströmmen sätts E = 0, annars sätts E = 1. En konstant Emk specificeras genom att sätta parametern EMK till önskad Emk.

3.1.2. Tyristorbrygga

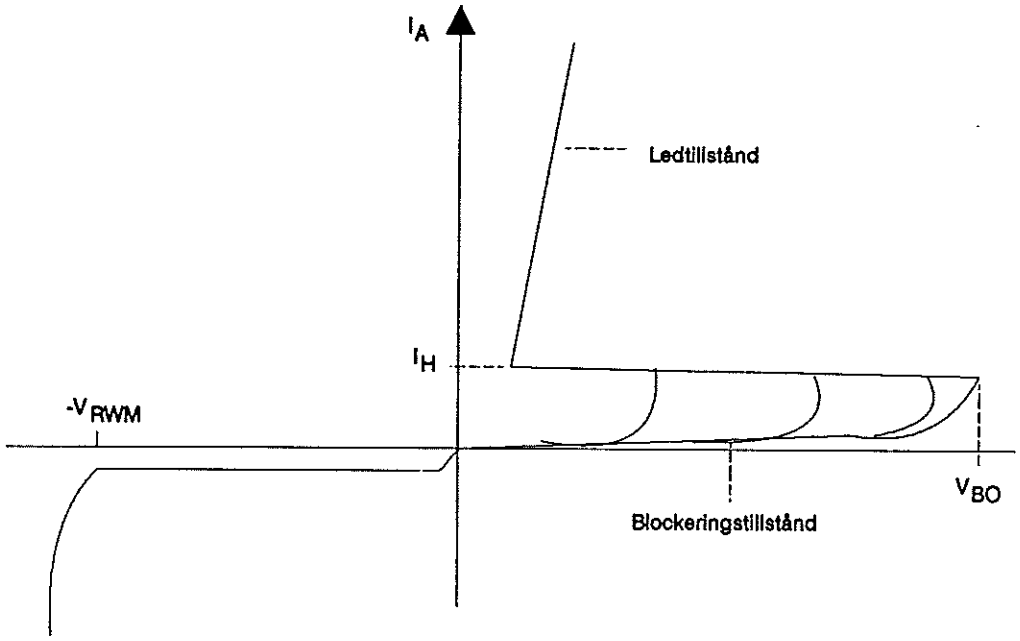
För att ta hänsyn till överlappning vid kommuteringsförloppet har en komplex modell av en 6-puls tyristorbrygga utvecklats i Bridge2 innehållande en matematisk modell av en tyristor. Vid enklare simuleringsfall där ingen hänsyn till överlappning behöver tas räcker det att använda den enklare modell som finns i Bridge1.

3.1.2.1. Bridge1

Bridge1 är en enkel modell av tyristorbryggan som inte tar hänsyn till någon överlappning. Insignalen N anger det tyristorpar som ska leda, dvs vilken huvudspänning som skall mata motorn. När strömmen närmar sig noll ska tyristorerna spärra spänningen. Det simuleras genom att om strömmen närmar sig noll, detekteras av signalen zerocurr=1, blir utspänningen från Bridge1 lika med motorns Emk. På så sätt blir di/dt i motorekvationen (se kap 3.2.1) lika med noll och strömmen stannar på noll. En ny huvudspänning läggs inte ut till motorn förrän ett nytt N kommer från modulen Trigg1.

3.1.2.2. Bridge2

Tyristorn är en av de äldsta krafthalvledarna (första fungerade 1957 i General Electrics forskninglaboratorium) och är fortfarande den halvledartyp som kan hantera störst effekt. Just tyristorns förmåga att hantera stora effekter är orsaken till att den används för att styra ut ström till DC-motorer. Tyristorns karaktäristik framgår av figur 14.

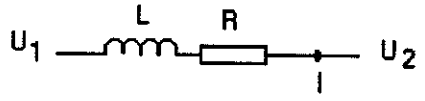


Figur 14. I-V karaktäristik för tyristorn.

Det unika med tyristorn ligger i I-V karaktäristiken. I backriktningen liknar tyristorn dioden, dvs tyristorn leder obetydligt med ström fram till avalanche-genombrottet vid V_{RWM}. I framriktningen har tyristorn två stabila tillstånd, som binds samman av en instabil övergång. Blockeringstillståndet karaktäriseras av låg ström och hög spänning. Ledtillståndet karaktäriseras av hög ström och låg spänning. Övergången mellan blockerat till ledande tillstånd sker genom att ge en tändpuls på styret. Hållströmmen I_H är den minsta ström med tyristorn framspänd, som kan drivas utan att tyristorn övergår till blockerat tillstånd.

Modellen av tyristorn baseras på följande antagande.

- Blockeringspänning = ∞
- Genombrottspänning = ∞
- Inget temperaturberoende



Figur 15. Modell av tyristor

För att få ett system med differentialekvationer representeras tyristorn av en induktans i serie med en resistans enligt figur 15. Tyristorns differentialekvation blir då

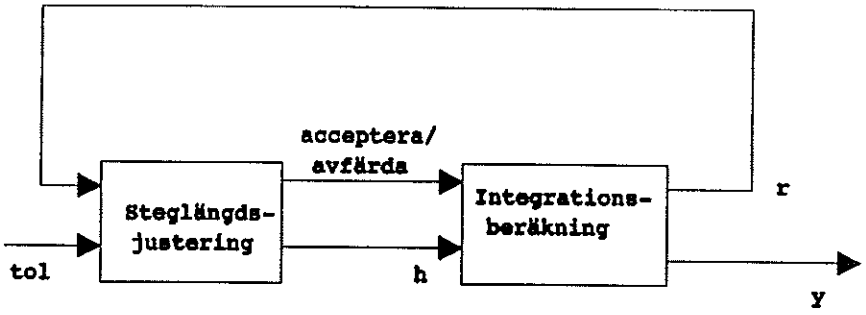
$$\frac{di}{dt} = \frac{U_1 - U_2}{L} - \frac{R}{L} i$$

Tyristormodellen kan befinna sig i två tillstånd antingen blockerat eller ledande och värdet på L och R beror på vilket tillstånd tyristorn befinner sig i.

Ledande	R=Ron	L=Lon
Blockerat	R=Roff	L=Loff

Övergång från blockerat till ledande tillstånd sker genom att tyristorn får en tändpuls från modulen Trigger2. Övergång från ledande till blockerat sker genom att strömmen genom tyristorn underskrider hållströmmen I_H.

För att få en stabil numerisk lösning är det viktigt att den integrationsalgoritm som används vid simuleringen är lämplig. Enligt manualen för SIMULINK ("SIMULINK A program for simulating dynamic systems") rekommenderas att Runge-Kuttas metod används som integrationsalgoritm för olinjära problem och system som innehåller både diskreta och kontinuerliga block. I detta projekt har Runge-Kutta 23 (RK23) använts vid simulering. RK23 använder som integrationsalgoritm 3'e ordningens Runge-Kutta och till steglängdsjustering används 2'a ordningens Runge-Kutta metod.

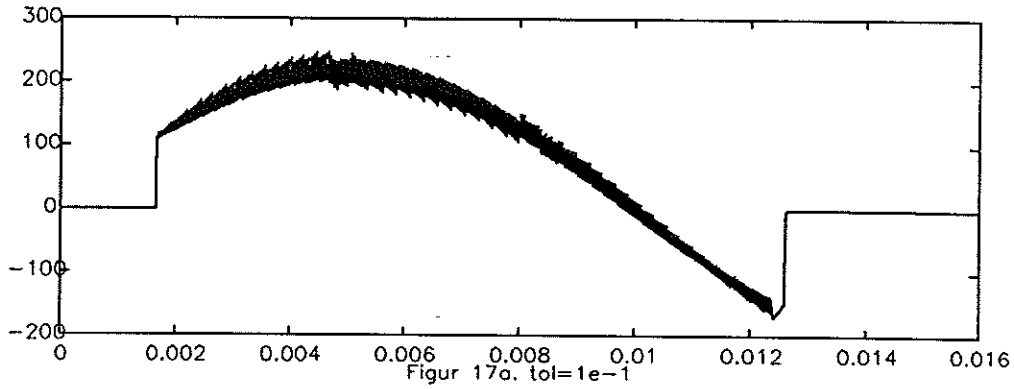


Figur 16. Uppbyggnad av integrationsalgoritmen. y är den beräknade punkten, h är steglängden, r är estimerade felet och tol är angiven tolerans.

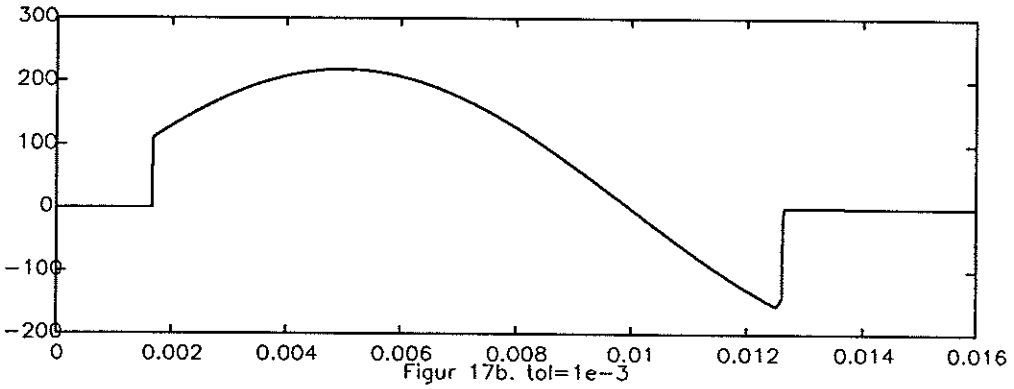
För en utförlig förklaring av hur integrationsalgoritmer med Runge-Kuttas metod är uppbyggda hänvisas till Å Björk G Dahlquist "Numeriska metoder" kapitel 8.3.3 och F.B Hildebrand "Introduction to numerical analysis" kapitel 6.16.

Steglängdsjusteringen bestämmer utifrån estimerade felet (r) och det angivna relativa felet (tol) om den beräknade punkten i integrationsalgoritmen skall accepteras eller avfärdas och om den använda steglängden (h) skall behållas, ökas eller minskas. I integrationsberäkningen sker beräkning av nästa punkt (y) och det estimerade felet (r). I K Gustafsson "Control of Error and Convergence in ODE Solvers" kapitel 2.5 beskrivs mer detaljerat hur steglängdsjustering i Runge-Kutta fungerar och i kapitel 3 beskrivs hur det relativa felet och steglängden beror på varandra och vilka begränsningar som finns.

Ett för litet angivet relativt fel leder till att steglängden blir liten och onödigt många punkter beräknas. Ett för stort angivet relativt fel gör att lösningen från integrationsalgoritmen avviker för mycket från önskat resultat. I figur 17 visas resultatet av simulering av tyristormodellen med induktiv last för två olika val av det relativa felet.



Figur 17a. tol=1e-1



Figur 17b. tol=1e-3

Figur 17. Spänning vid kommutering av en tyristor vid alfa = 30 grader med induktiv last vid a. tol = 10^{-1} respektive b. tol = 10^{-3}.

I figur 17a visas en simulering av spänningen från tyristormodellen med det relativa felet tol = 10^{-1}. Det syns klart att det relativa felet inte räcker till för att få en stabil kurva. I figur 17b visas det nödvändiga relativa felet för att simuleringen skall bli stabil. För vidare simuleringar med tyristormodellen används det relativa felet tol = 10^{-3}.

Det globala felet, vilket är summan av alla lokala fel, är inte möjligt att beräkna (kräver att den sanna lösningen är känd) och är svår att estimera. Integrationsmetoden begränsar sig till att estimera och kontrollera det lokala relativa felet. Estimering av felet behandlas i K. Gustafsson "Control of Error and Convergence in ODE Solvers" kapitel 2.2.

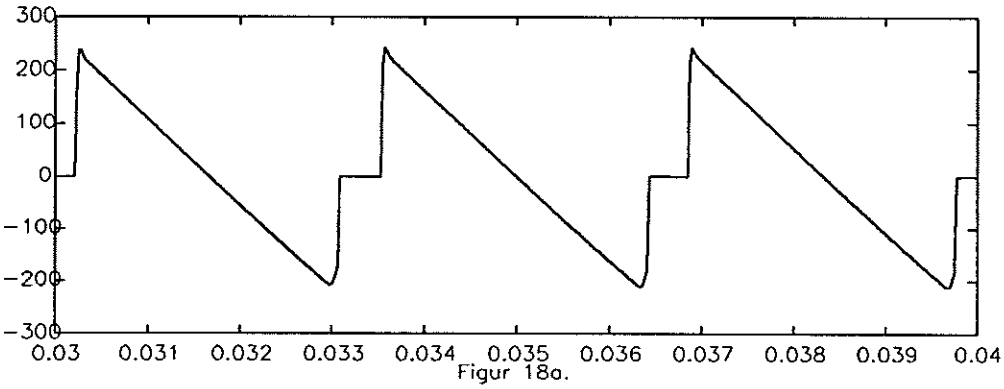
Tyristormodellen kopplas samman till en 6-puls tyristorbrygga enligt figur 20. Ekvationen för

motorn, som är last till bryggan skall som insignal ha en spänning. Tyristorekvationerna har ström som tillstånd. För att få spänningsnoderna Up respektive Un efter tyristorn ansluts en RC-krets till tyristorbryggans strömekvationer. Från RC-kretsen får man då den spänning som matar motorn genom att

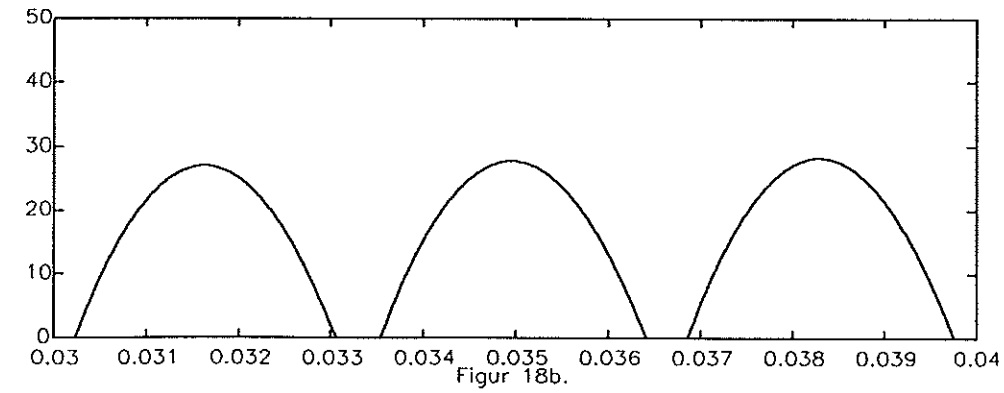
$$Up = Rc * (Ir + Is + It - Ilast) + Uc$$

$$\frac{dUc}{dt} = \frac{Ir + Is + It - Ilast}{C}$$

där Ir, Is och It är strömmar från respektive tyristorgren. Iast är strömmen till lasten och Uc är spänningen över kondensatorn. För att inte nämnvärt påverka kommuteringen av tyristorerna skall RC-kretsen ha mindre tidskonstant än tyristorn. För att inte få någon efterledning när samtliga tyristorer är i blockerat tillstånd, sätts Up=E/2 och Un=-E/2 då strömmen till lasten närmar sig noll. E är motorns emk. I figur 18 visas spänning och ström från 6-puls tyristorbryggan.



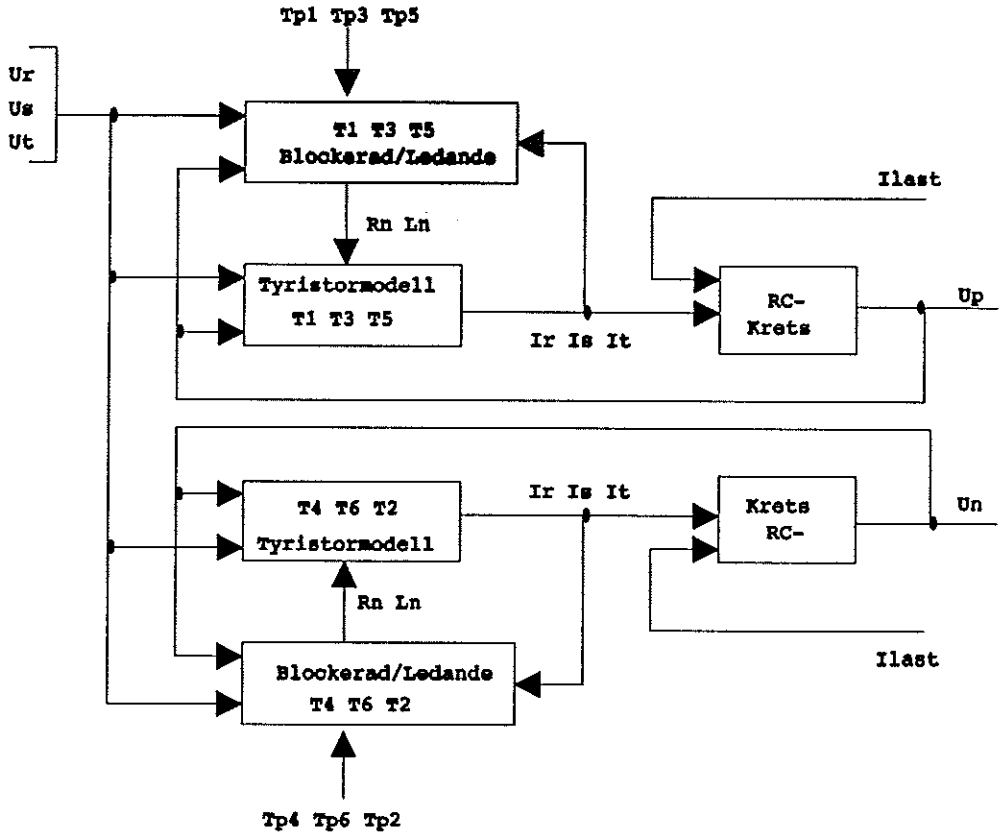
Figur 18a.



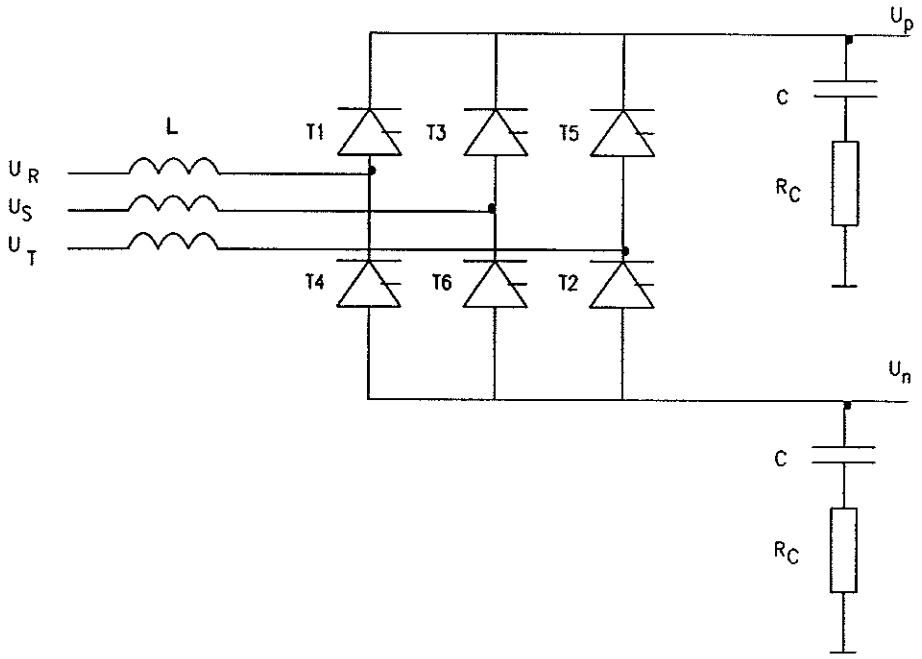
Figur 18b.

Figur 18a: Spänningen Up-Un från tyristorbryggan.
Figur 18b: Strömmen från tyristorbryggan.

I figur 19 visas ett blockschema hur modellen av tyristorbryggan är implementerad i SIMULINK.



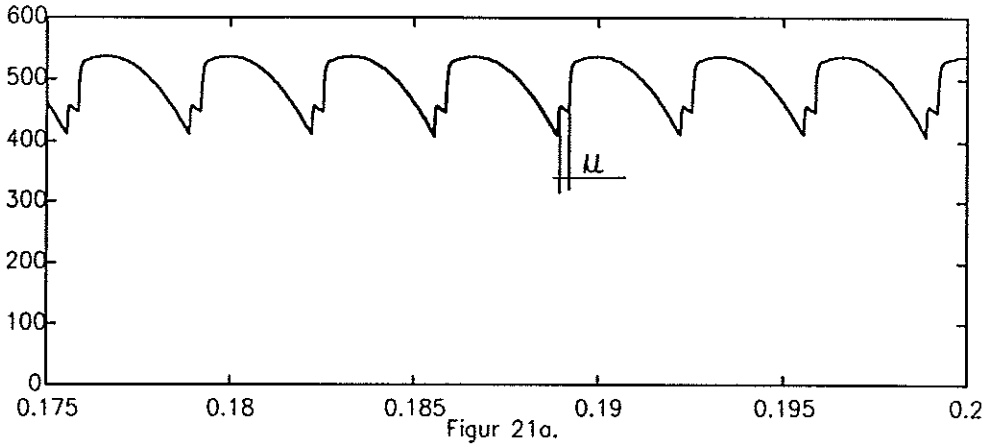
Figur 19. Modell av tyriristorbrygga. Rn och Ln är n'te tyristorns resistans respektive induktans. Tpn är n'te tyristorns tändpuls. För övriga beteckningar se figur 20.



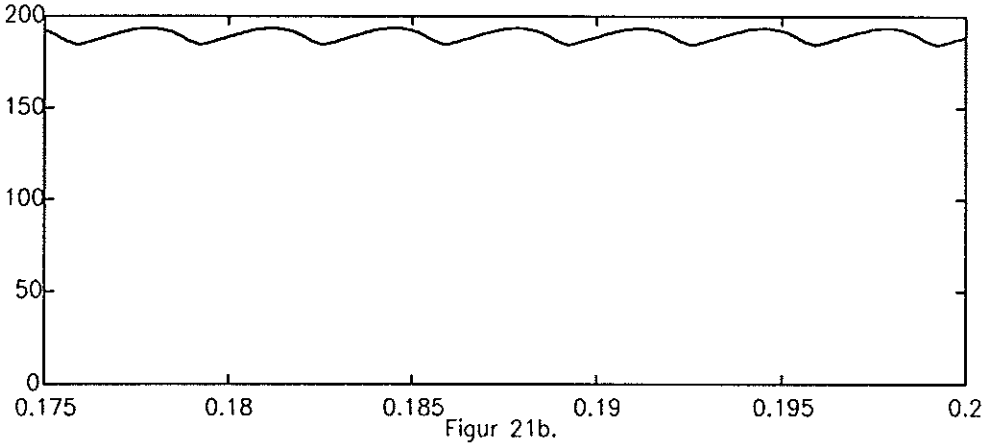
Figur 20. 6-puls tyristorbrygga med nätinduktans.

Den induktans som finns i modellen för tyristorn används för att simulera ett verkligt nät med induktanser. När nätet innehåller induktanser får man överlappning vid kommuteringsögonblicket. Se figur 21.

Överlappning beror på att när nätet innehåller induktanser kan strömmen inte stiga respektive falla momentant. Det krävs en viss tid. Detta leder till att kommuteringen mellan tyristorerna kan ta lång tid. Under kommuteringen leder samtidigt både den nytändande och den slocknande tyristorn. Resultatet är, att spänningen ut till motorn under tiden för överlappning kommer att vara medelvärdet av de båda fasspänningar man växlar emellan. Den tid man har överlappning ökar med ökad nätinduktans och ju mindre alfa man har och ju större kvoten mellan ström och spänning är, desto mer ökar överlappningen.



Figur 21a.



Figur 21b.

Figur 21. Exempel på överlappning vid alfa=10. Överlappningsvinkeln=u, nätinduktansen är L = 0.1 mH och I = 180 A.
21a: Spänning (Up-Un) från tyristorbryggan.
21b: Strömmen till lasten.

3.1.3. Trigger

Triggern har till uppgift att ge tändpuls till rätt tyristorpar, dvs styra bryggan att ge rätt huvudspänning till motorn. Det finns två uppsättningar av triggmoduler. Den enkla, Trigger1 används när man har den enkla varianten på tyristorbrygga. Den andra triggmodulen, Trigger2 används när man har den mer komplexa tyristorbryggan. Anledningen till att det finns två triggmoduler är att för den enkla bryggan behöver man endast ange vilket tyristorpar som skall leda, men för den komplexa bryggan måste tändpuls till respektive tyristor ges. Implementeringen av Trigger överenstämmer med funktionen av tändpulsenheten beskriven i kapitel 2.2.3.

3.1.3.1. Trigger1

Trigger1 används tillsammans med Bridge1. Modulens sampeltillfälle sker vid två tillfällen för varje strömbubbla. Vid tiden Ttop läses nya värden på Ttop, TOT och N in, vilka har beräknats i modulen Control. Vid tiden TOT läggs det nya värdet på N ut som utsignal. Utsignalen N ligger kvar ända tills nästa tid TOT.

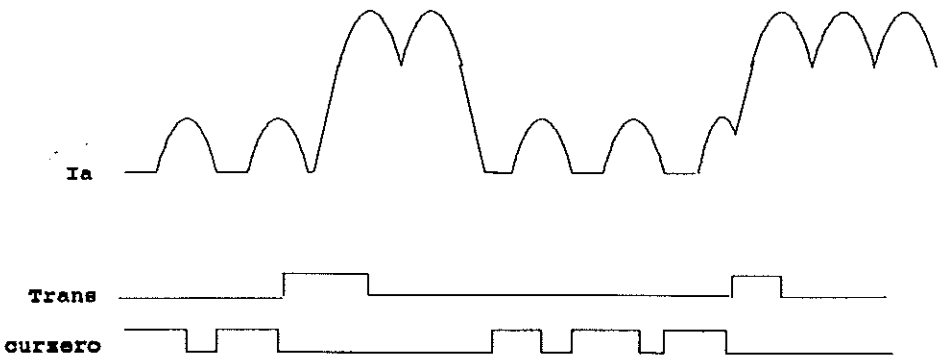
3.1.3.2. Trigger2

Trigger2 används tillsammans med Bridge2. Modulen består av två delmoduler. Den ena modulen ftrigger fungerar på samma sätt som den föregående modulen Trigger1 med undantag av att N endast ligger kvar som utsignal tills nästa tid Ttop. Från modulen ftrigger exporteras N till modulen Adapt. Adapt ger som utsignal tändsignal till det aktuella tyristorparet.

3.1.4. Control

I Control simuleras strömriktarens regulator och är den centrala delen i hela systemet. Alla beräkningar för de övriga blockens sampeltillfällen Ttop, Tbot och TOT sker här. Control fungerar på samma sätt som regulatorn beskriven i kapitel 2.2.2.

Blocket samplas vid tiden Ttop. Först predikteras medelvärdet av nuvarande strömpuls och korrektionsfaktorn Icorr adderas. Om det predikterade värdet är större än det verkliga ökas Icorr med ett litet steg, annars minskas Icorr med ett litet steg. För att kunna bestämma nästa triggtillfälle TOT beräknas nödvändig styrvinkeländring Deltsum.



Figur 22. Viktiga signaler i modulen Control.

Styrlagarna för beräkning av styrvinkelsändringen Deltsum finns härledda och beskrivnai M Johansson rapport "The current controller of TYRAK L/MIDI Teknisk rapport ABB Drives AB nr 89-021".

När Deltsum är beräknat kan man bestämma tändvinkeln i tid Talpha och tiden för nästa triggtillfälle TOT. Det görs på samma sätt som beskrivs i kapitel 2.2.2. Från modulen Sync fås tiden Timezero (se kapitel 3.2.6), vilken anger tiden för nästa gång alfa=0 för huvudspänningen Urs. Avståndet mellan två på varandra följande kommuteringar är i stationärt tillstånd 1/6 av nätets periodtid. Genom att addera en multipel av 1/6 av nätets periodtid till Timezero fås tiden för alfa=0 för ett godtyckligt tyristorpar. Variabeln N bestämmer den multipel som skall adderas till Timezero, för att tillsammans med Talpha ge rätt triggtidpunkt TOT. Vid varje Ttop adderas N med 1 för att flytta fram alfa=0 till nästa huvudspänning. När ett nytt värde på Timezero kommer sätts N till sitt utgångsvärde.

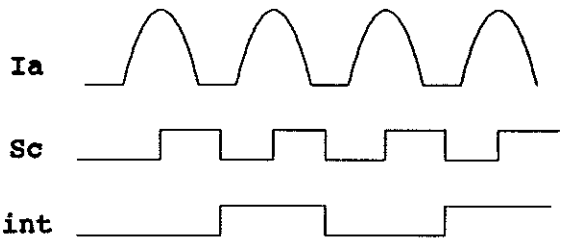
Utifrån tiden TOT beräknas först tiden för nästa Tbot. Därefter beräknas tiden för nästa Ttop som Tbot + 30 grader.

För att regulatorn skall fungera smidigt vid start av simulering är referensen Iref låst till 10 under de första 0.02 sekunderna. En ändring av Iref innan den tiden får inte någon verkan.

3.1.5. Curr

I detta block sker medelvärdesbildning av ankarströmmen. Blocket fungerar i princip på samma sätt som den beskrivna strömmätningen i kapitel 2.2.1. Medelvärdesbildning sker genom att integrera motorströmmen. För att hinna nollställa integratorn innan nästa strömpuls kommer används två integratorer som arbetar växelvis. Vid tiden Tbot börjar den ena integratorn integrera, medan den andra nollställs. Vid nästa Tbot växlar man över till den andra integratorn. Vid varje tid Ttop läses den i regulatorn beräknade Tbot. Det finns två variabler som synkroniserar modulen. Variabeln int håller reda på vilken integrator som skall användas och variabeln Sc bestämmer vilken tid som

är nästa sampeltillfälle, Ttop eller Tbot. Se figur 23.

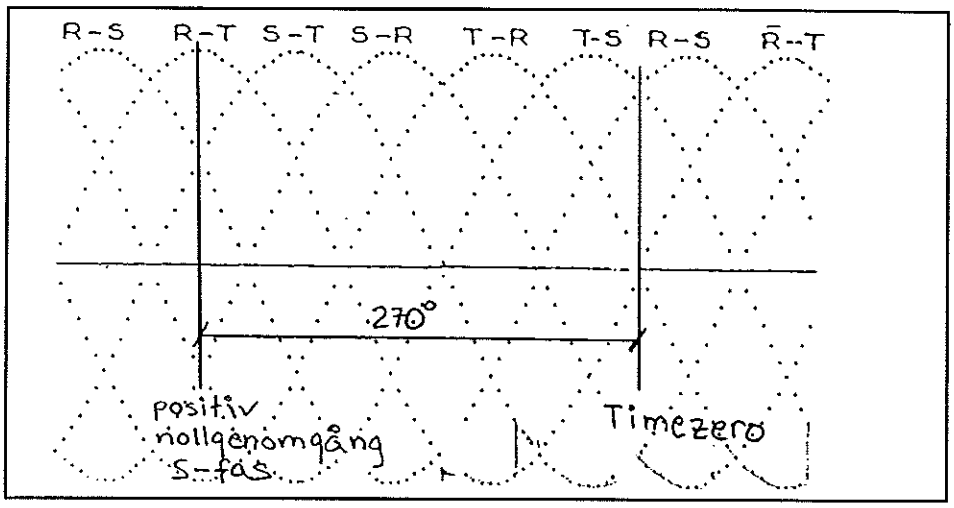


Figur 23. Synronisering av strömmätning med hjälp av variablerna int och Sc.

Även detektering av nollström och mätning av strömpulsens ledtid sker i detta block. Om strömmen blir noll indikeras det med att variabeln zerocur sätts till ett.
Strömpulsens ledtid mäts genom att integrera en etta då strömmen inte är noll. Vid varje Tbot tar man differensen mellan aktuell ackumulerad tid och ackumulerad tid vid förra Tbot. Strömpulsens ledtid ges i variabeln Tx.

3.1.6. Sync

För att triggerpulserna som beräknas i regulatorn ska komma vid rätt tillfälle relativt nätet, måste man synkronisera nät och triggerpulserna. Det görs genom att detektera tiden för S-fasens positiva nollgenomgång. Ur den tiden beräknas tiden för tändvinkeln alfa=0 vid huvudspänningen Urs genom att addera 270 grader till tiden för nollgenomgång. Resultatet finns i signalen Timezero. Se figur 23. Blocket ger också huvudspänningens periodtid i signalen Tcycle genom att ta differensen mellan två på varandra följande nollgenomgångar.



Figur 24. Beräkning av Timezero.

För att minska modulens exekveringsfrekvens sker nästa sampel, efter det att nollgenomgång detekterats, efter 95% av spänningens periodtid.

3.1.7. Net

Modulen definerar nätets fasspänningar som ren sinus genom att ange toppvärde. Modulen är implementerad med standardblock i SIMULINK.

4. RESULTAT AV SIMULERING

Simuleringarna har utförts i två olika modeller. En modell innehåller strömregulatorn och den enkla tyristorbryggan Bridge1. Denna modell används för att studera hur regulatorn fungerar för olika driftsfall. Den andra modellen som har använts innehåller strömregulatorn och den komplexa tyristorbryggan Bridge2. I denna modell har studerats inverkan på nätinduktanser.

4.1. Strömregulatormodellen

För att regulatorn skall uppföra sig på bästa sätt användes igångkörningsinstruktioner för TYRAK MIDI för att trimma regulatorparametrarna CONSTCON, CONSTRL och IAGAIN. Först trimmas regulatorn till en dead beat regulator (IAGAIN=1.0) vid kontinuerlig ström. Se figur 1 i bilagan. Resultatet verifieras i diskontinuerlig ström. Se figur 2 i bilaga. För att få en stabil regulator sätts därefter IAGAIN till 0.8. Därefter studeras regulatorns uppförande vid olika utseende på strömmar. Resultaten visas i figur 3 - 6 i bilaga där a är spänningen till lasten och b är strömmen till lasten.

Figur 1: Stegsvar vid kontinuerlig ström.
IAGAIN = 1.

Figur 2: Stegsvar vid diskontinuerlig ström.
IAGAIN = 1.

Figur 3: Stegsvar vid diskontinuerlig ström.
IAGAIN = 0.8.

Figur 4: Övergång diskontinuerlig till kontinuerlig ström utan överlappning. IAGAIN = 0.8.

Figur 5: Övergång diskontinuerlig till kontinuerlig ström med överlappning. IAGAIN = 0.8.

Figur 6: Stegsvar vid kontinuerlig ström.
IAGAIN = 0.8.

4.2.Komplex tyristorbrygga (Bridge2)

Med Bridge2 studerades inverkan av induktanser på nätet, ett så kallat verkligt nät. Induktanser på nätet ger en överlappning vid kommutering, så både den släckande och tändande leder samtidigt för en kort tid. Resultatet visar en god överenstämmelse mellan simuleringar och teoretiska beräkningar. Överlappningsvinkeln u beräknas enligt

$$u = \arccos\left(\cos(alfa) - \frac{4\ pi\ f\ L\ I}{\sqrt{2}\ U}\right) - alfa$$

där f = nätfrekvensen, L = Nätinduktansen, I = strömmen och U = huvudspänning.

Med U = 380 V och f = 50 Hz.

alfa (grader)	L (mH)	u teoretiskt (grader)	u simulerat (grader)
0	0.01	3.3	3.5
0	0.1	10	10
10	0.1	4.4	4.5

Resultaten visas i figur 7 - 9i bilaga där a är spänningen till lasten och b är strömmen till lasten.

Figur 7: alfa = 0 grader och L = 0.01 mH.

Figur 8: alfa = 0 grader och L = 0.1 mH.

Figur 9: alfa = 10 grader och L = 0.1 mH.

4.3.Jämförelse mellan Bridge1 och Bridge2

En simulering gjordes för att jämföra regulatorns uppförande när man har Bridge1 respektive Bridge2. Det visar sig att regulatorn beter sig på samma sätt oavsett om man använder Bridge1 eller Bridge2. Resultatet visas i figur 10 - 11 i bilaga där a är spänningen till lasten och b strömmen till lasten.

Figur 11: Övergång från diskontinuerlig till kontinuerlig ström. IAGAIN = 0.8.

Figur 12: Stegsvar vid kontinuerlig ström. IAGAIN = 0.8.

BILAGOR:

Litteraturförteckning

Simuleringsresultat

Bruksanvisning

Programlistning av Matlabfiler

LITTERATURFÖRTECKNING:

Å. Björk G. Dahlquist: "Numeriska metoder"
AB CWK Gleerup Bokförlag, LUND, 1969.

Franz P. Meyer A: "DIGITAL SIMULATION OF A COMPLETE SUBSYNCHRONOUS CONVERTOR CASCADE WITH 6/12-PULSE FEEDBACK SYSTEM" IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No.12 December 1981.

Gustafsson Kjell:"Control of Error and Convergence in ODE Solvers". Department ofAutomatic Control, LUND Institute of Technology. LUND 1992.

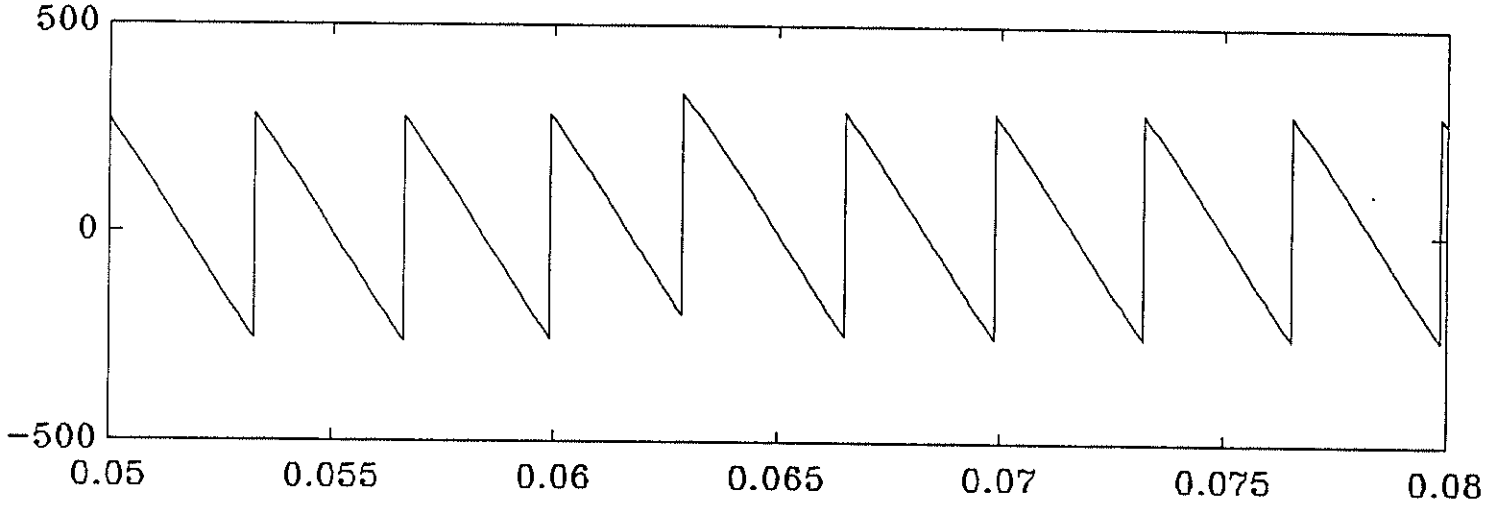
F.B Hildebrand:"Introduction to numerical analysis".
Mc-Graw Hill BOOK COMPANY INC. 1956.

Johansson Mikael:"The Current Controller of TYRAK L/MIDI"
Teknisk rapport ABB Drives AB Nr DK 89-021.

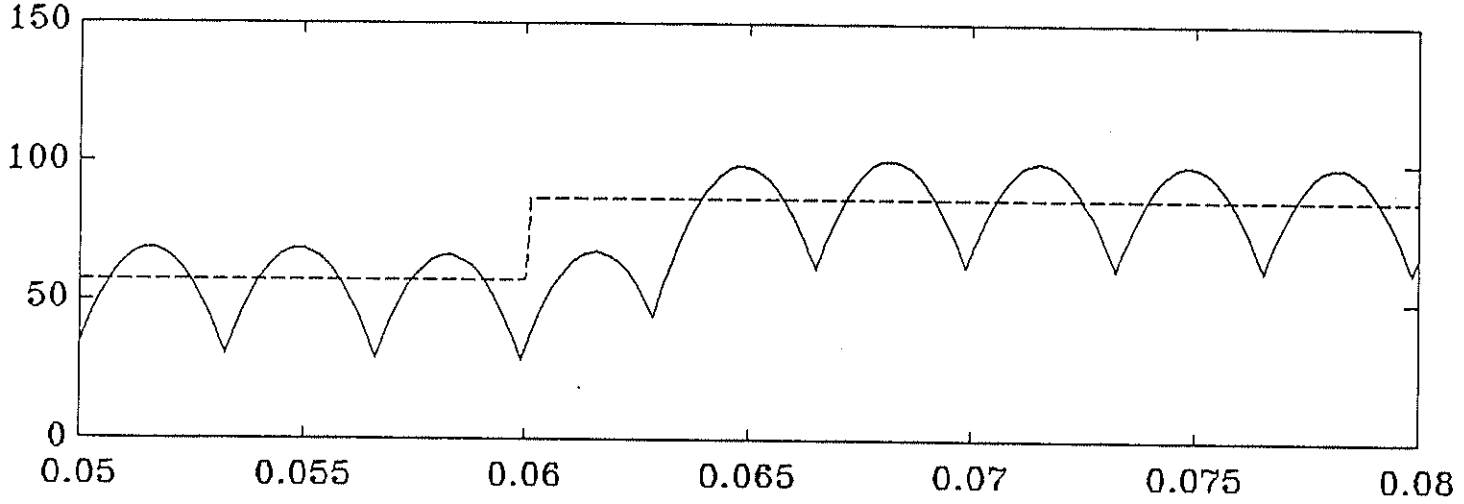
Magnusson Tommy: "Kompendium Kraftelektronik"
Industriell Elektronik och Automation, Lunds Tekniska Högskola , 1991

"ELMASKINSYSTEM"
Industriell Elektronik och Automation, Lunds Tekniska Högskola
KF-Sigma, LUND 1991.

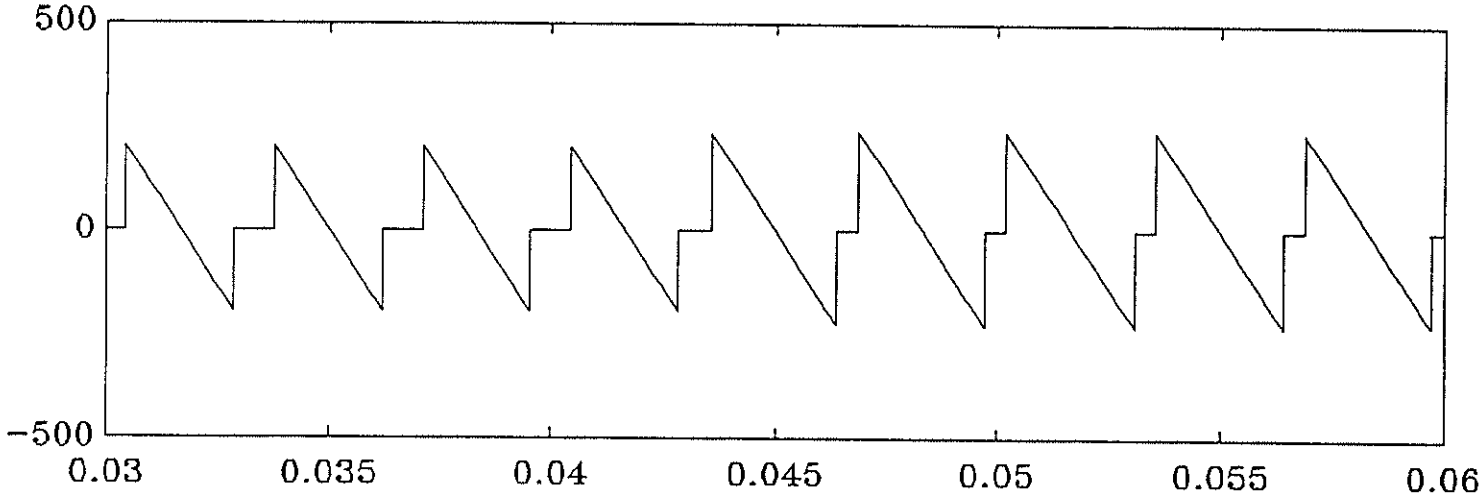
" SIMULINK A program for Simulating Dynamic Systems" The MathWorks Inc. March 1992.



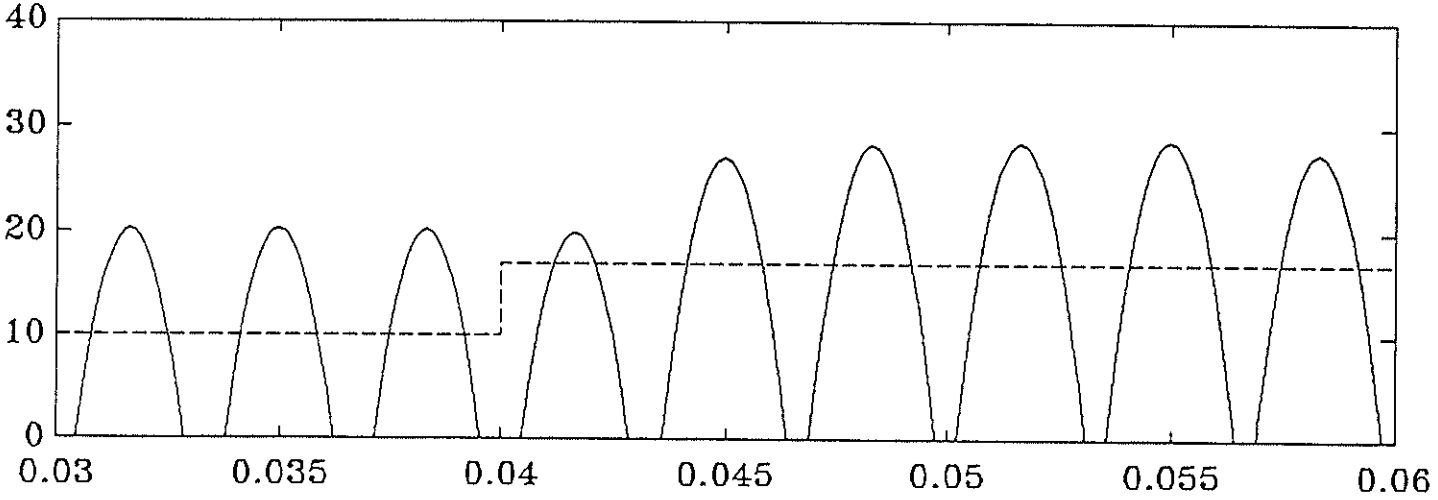
Figur 1a.



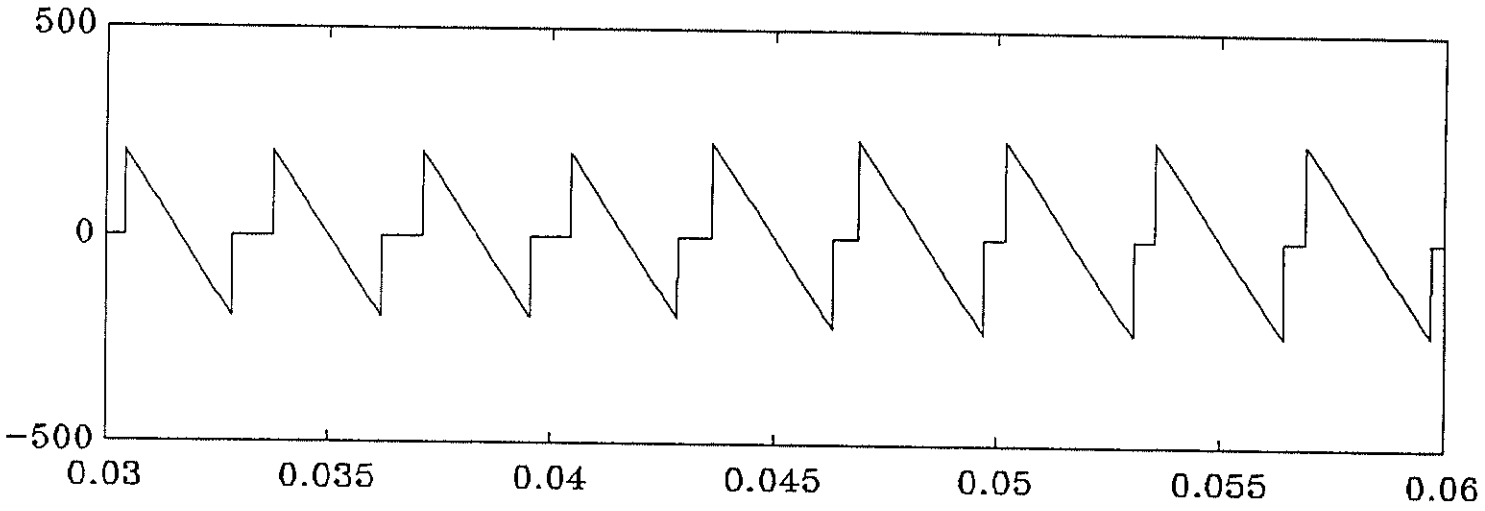
Figur 1b.



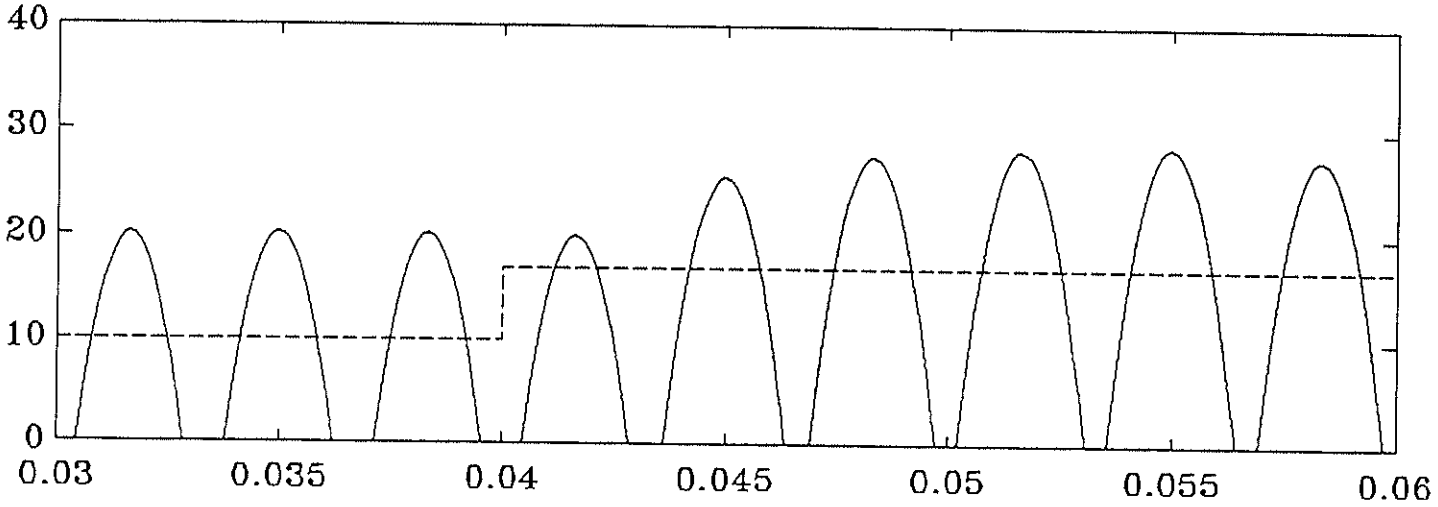
Figur 2a.



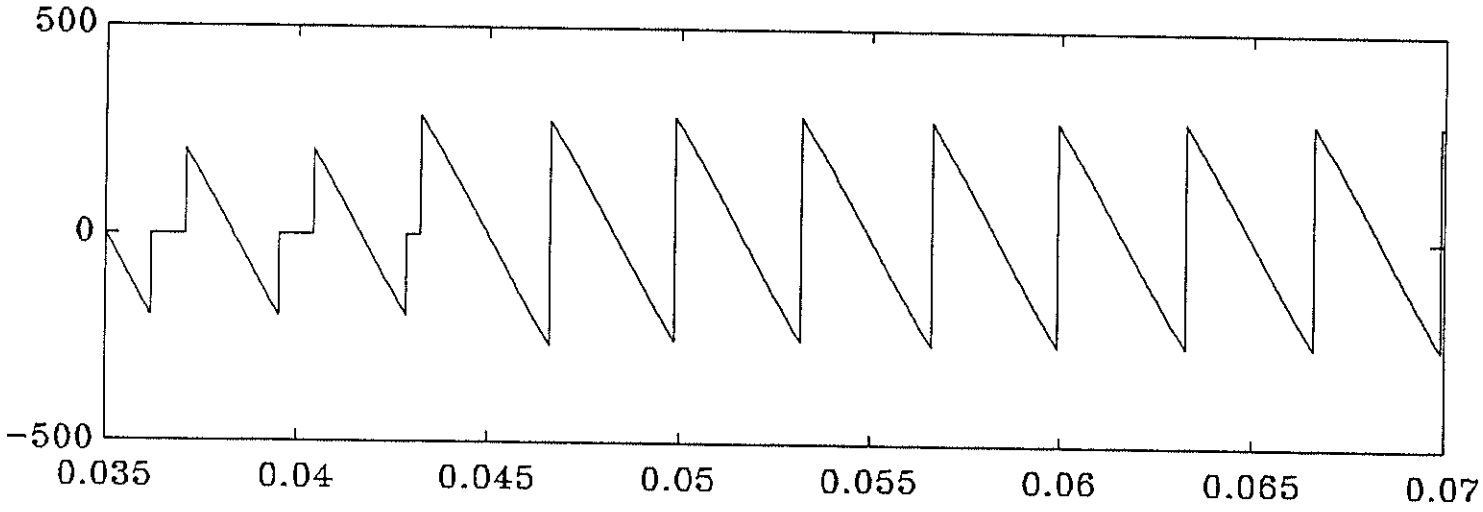
Figur 2b.



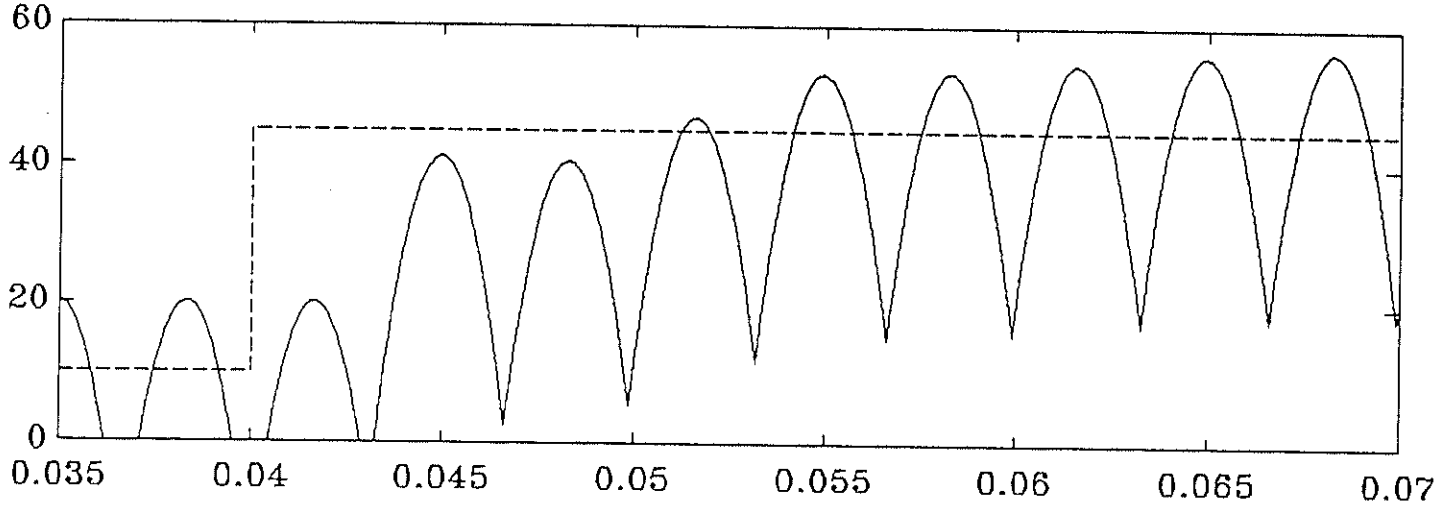
Figur 3a.



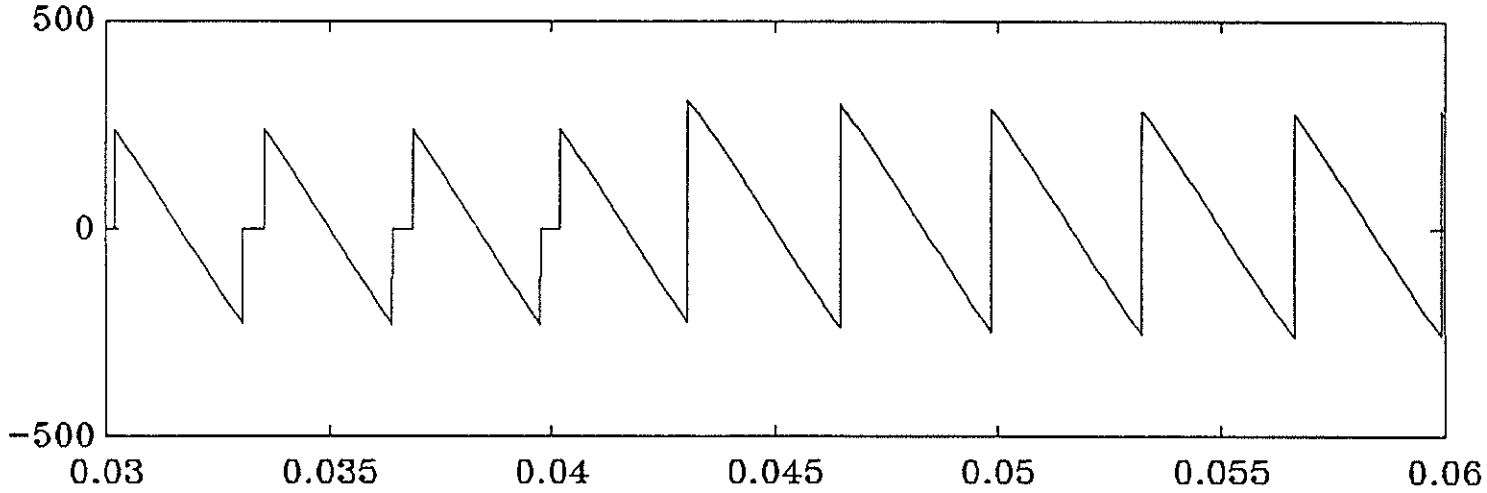
Figur 3b.



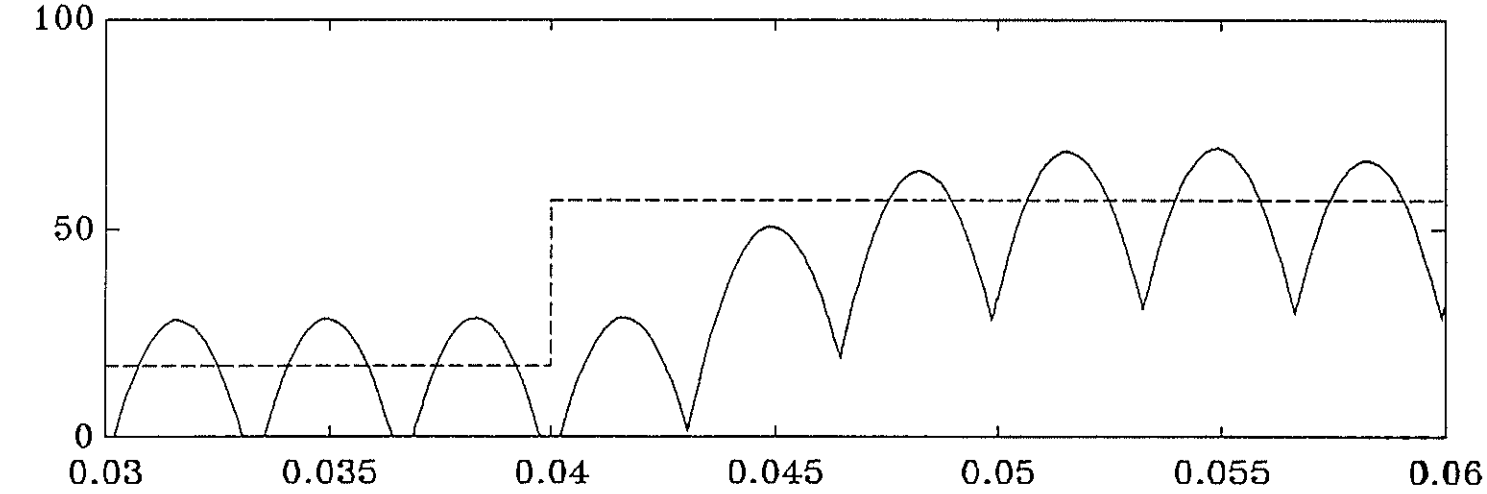
Figur 4a.



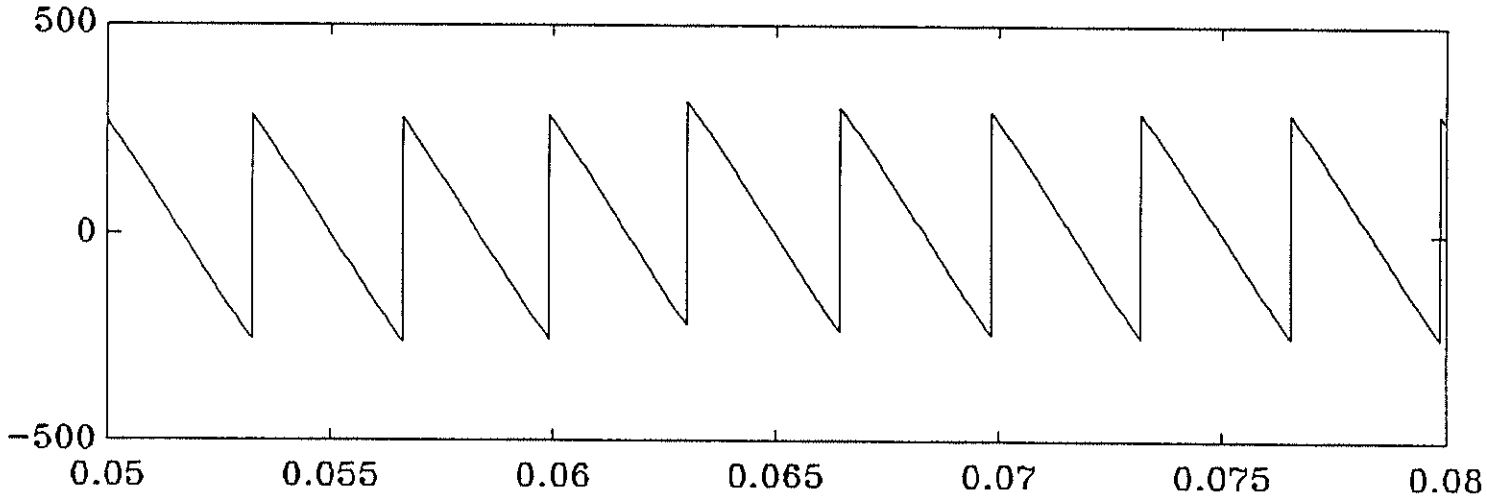
Figur 4b.



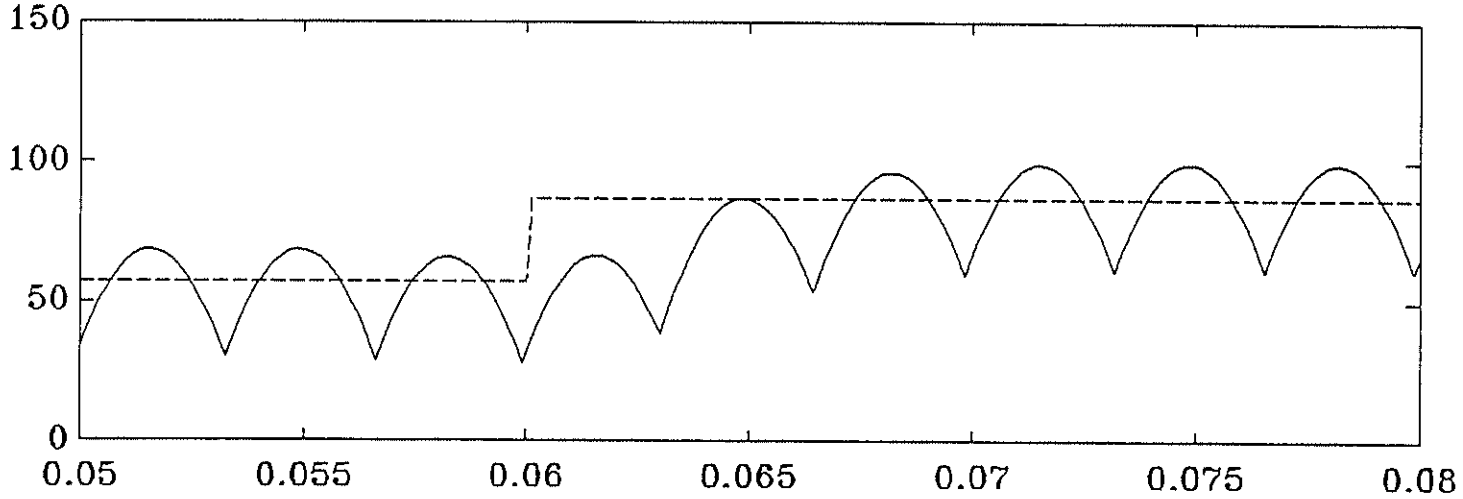
Figur 5a.



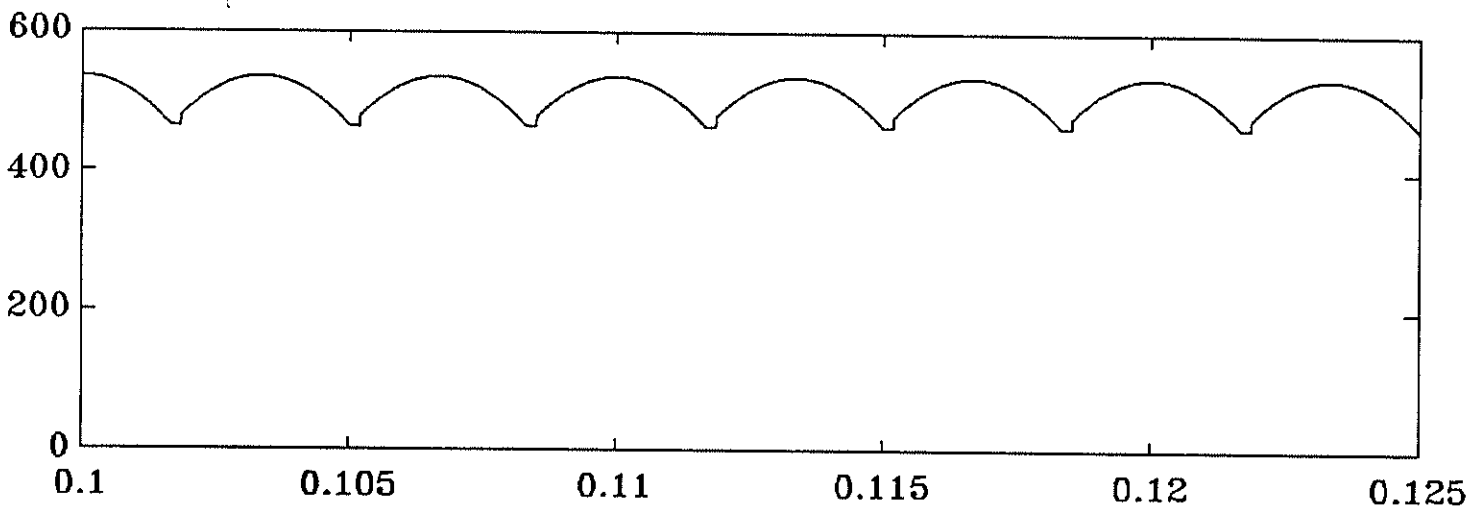
Figur 5b.



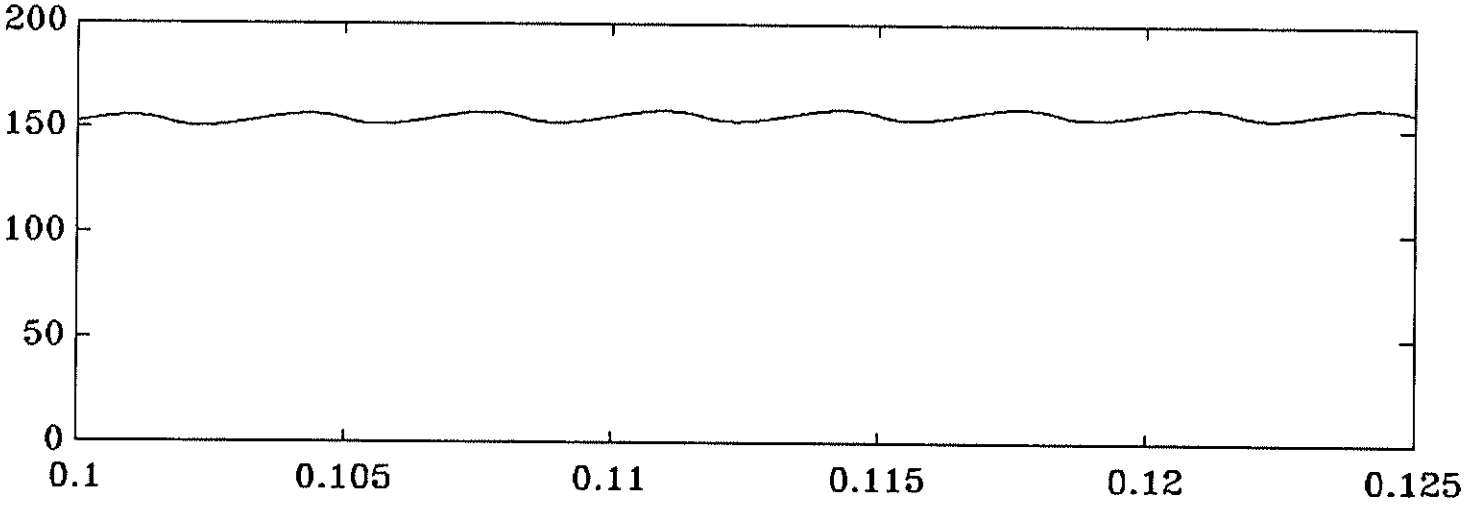
Figur 6a.



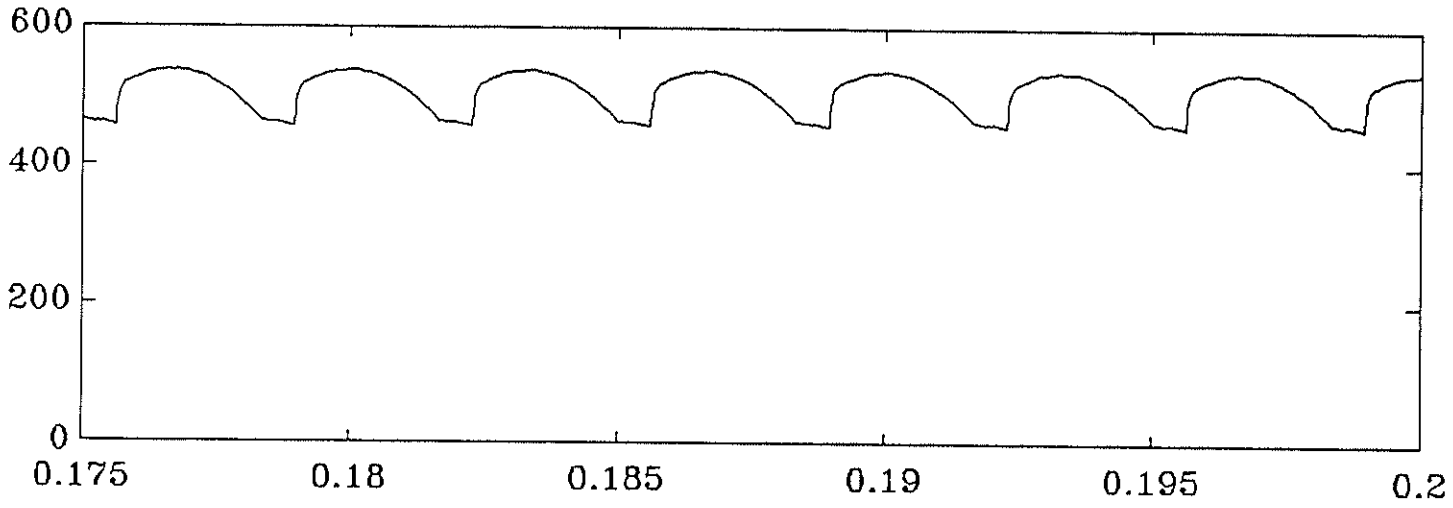
Figur 6b.



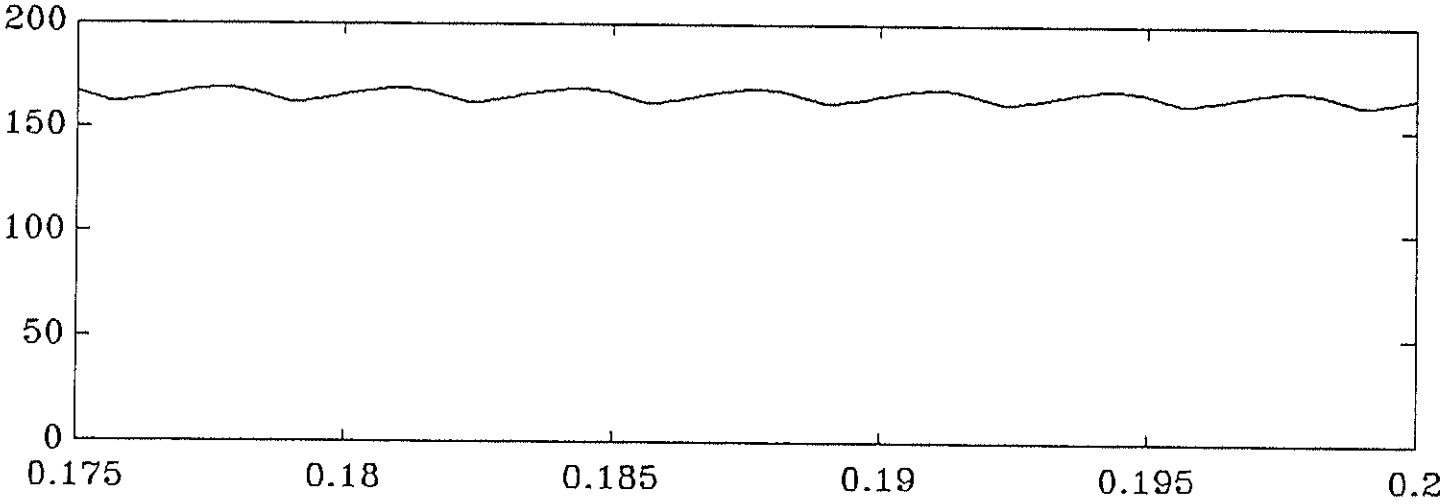
Figur 7a.



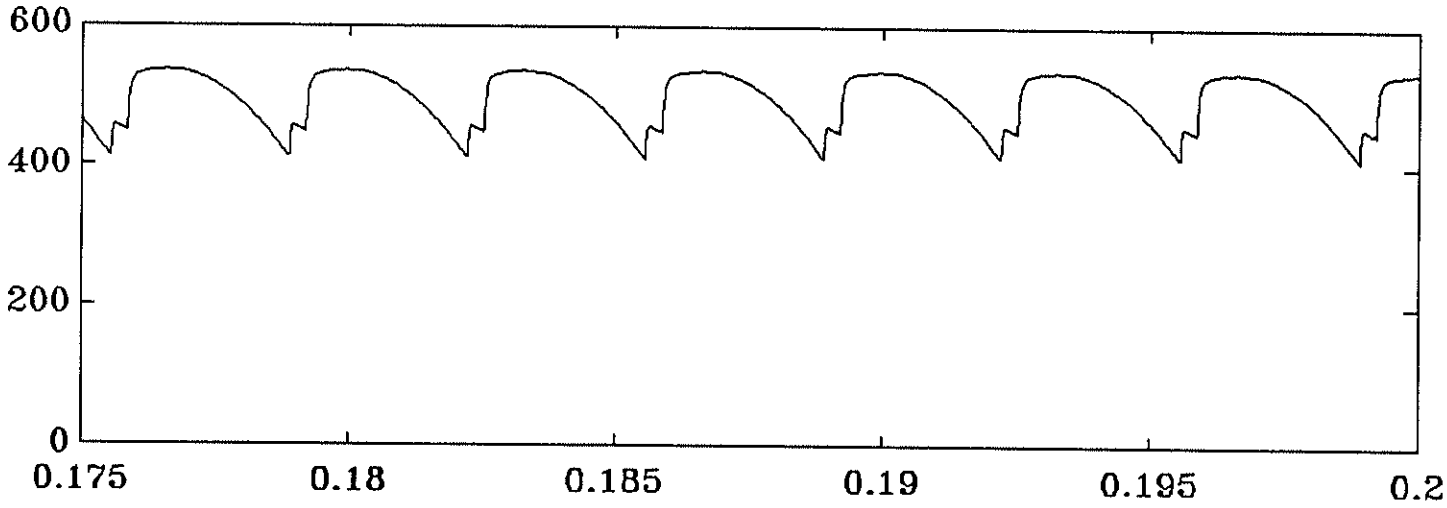
Figur 7b.



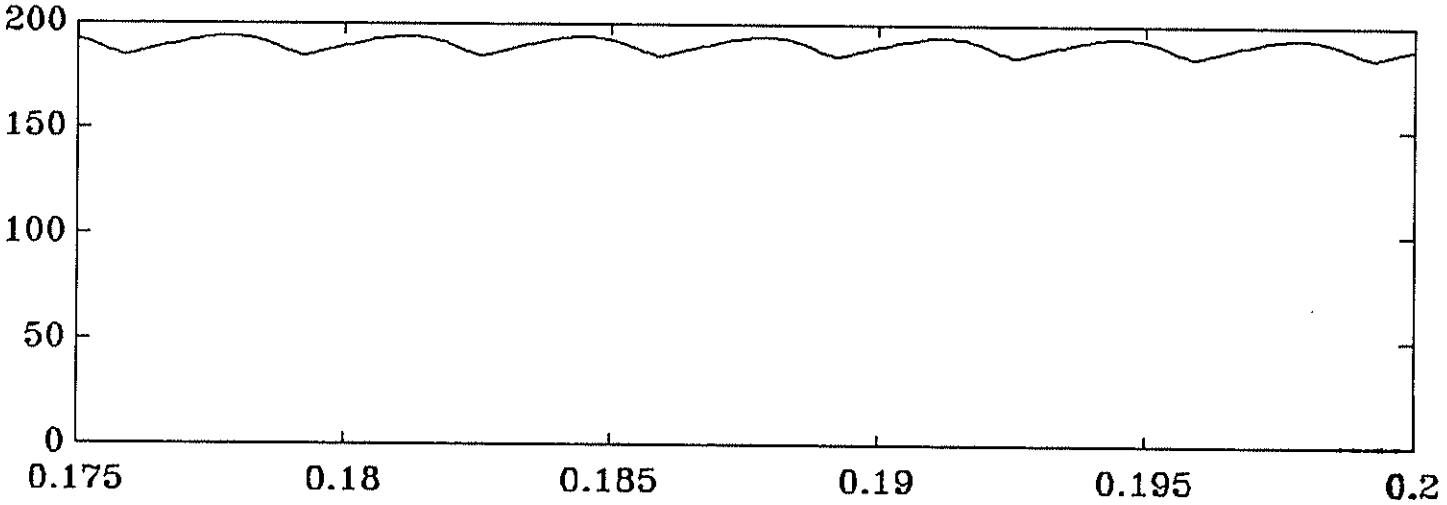
Figur 8a.



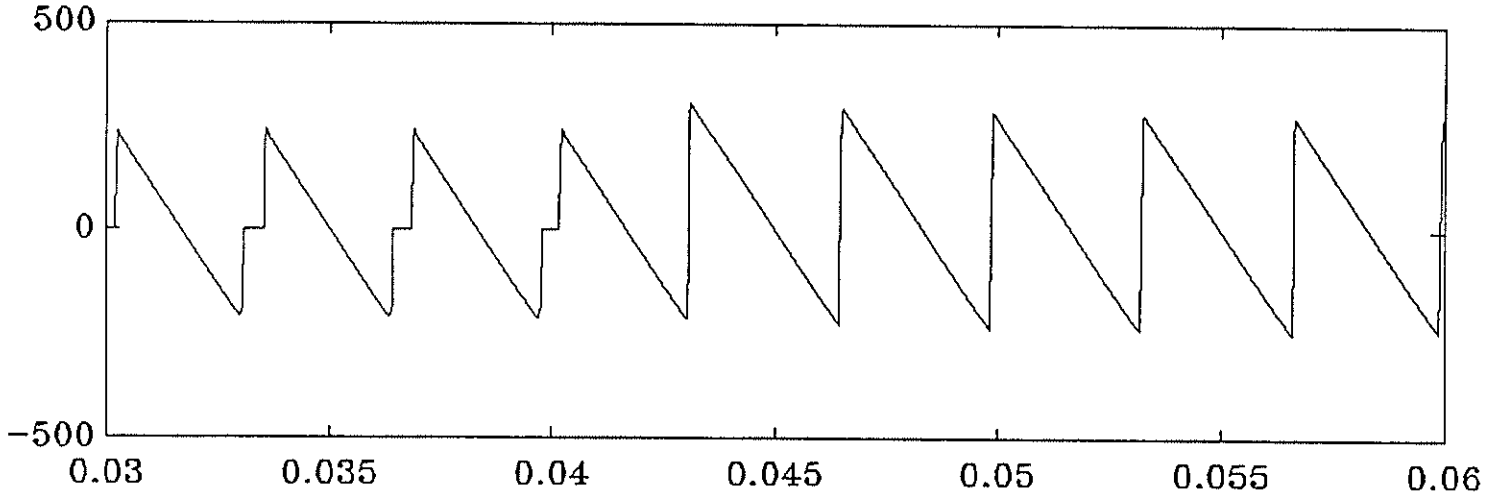
Figur 8b.



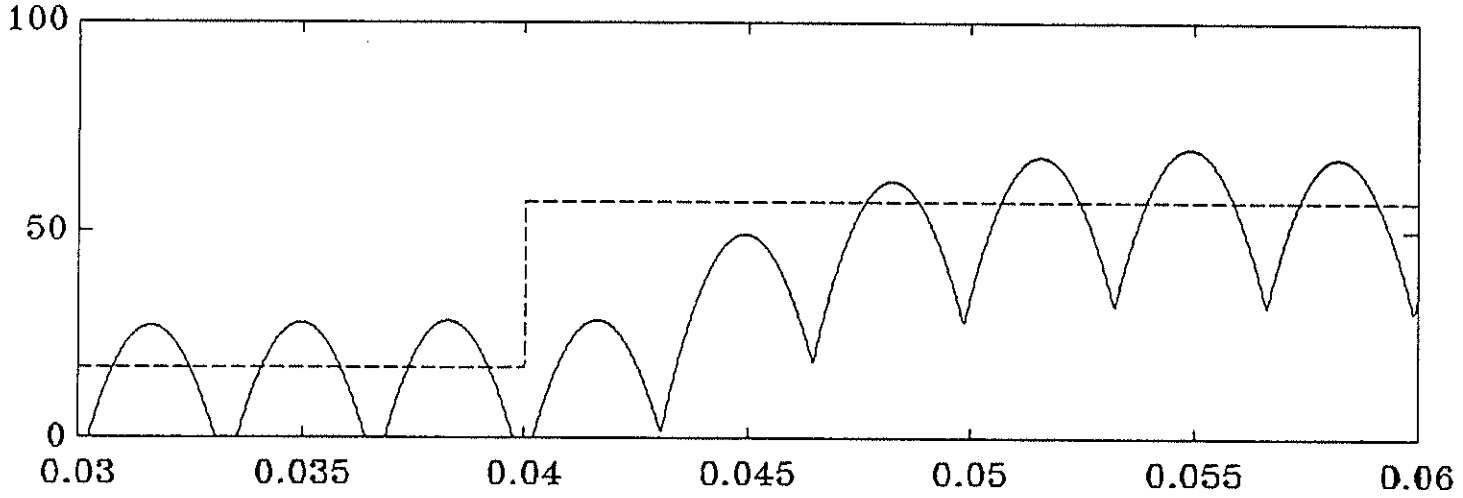
Figur 9a.



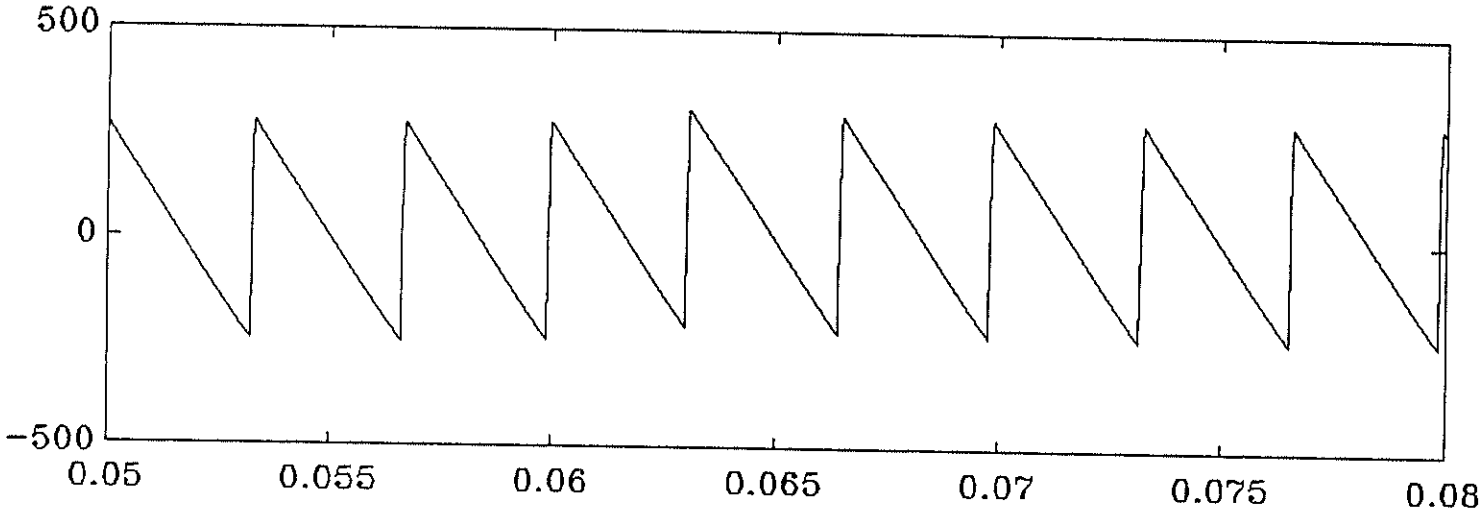
Figur 9b.



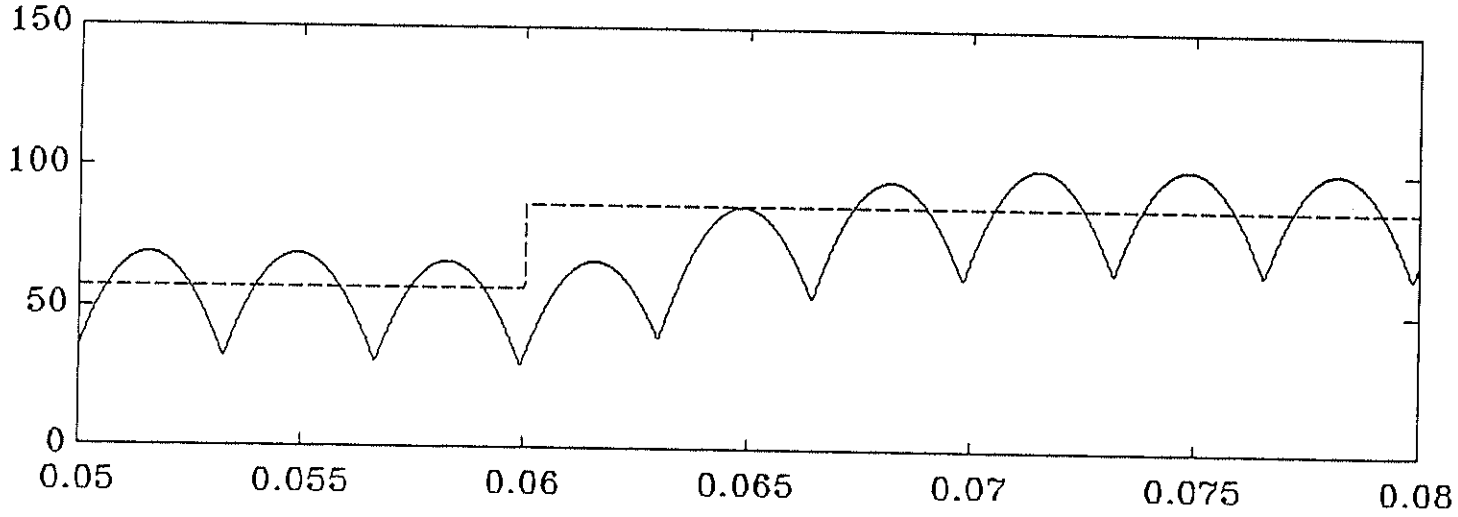
Figur 10a.



Figur 10b.



Figur 11a.



Figur 11b.

BRUKSANVISNING

Här ges en kort beskrivning av hur man skall göra för att komma igång med simuleringarna. För en mer detaljerad beskrivning av kommandon hänvisas till Simulink's manual "SIMULINK, a program for simulating dynamic systems".

Det finns två färdiga modeller att använda.

Control1 innehåller regulatorn hopkopplad med den enkla 6-pulsbryggan (Bridge1). I den modellen finns det inga nätinduktanser och bryggan ger ut ideala spänningar till motorn.

Control2 innehåller regulatorn med den komplexa 6-pulsbryggan (Bridge2), som innehåller en modell av tyristorn och nätinduktanser.

Dessutom finns samtliga block samlade i biblioteket Blocks. Där finns ett antal block som inte används i Control1 respektive Control2, men som kan vara till användning om man vill bygga en egen modell.

Simulering av modellen kan man göra genom att välja start/stop under rubriken Simulation. Genom att dubbelklicka på oscilloskopblocket visas aktuell spänning från tyristorbryggan och strömmen genom motorn.
Man kan även titta på andra signaler med det lösa oscilloskop som finns med genom att dubbelklicka på oscilloskopet och därefter på den önskade signalen.
För att kunna plotta signaler i Matlab skickas det som skall plottas i blocken "To workspace". Efter simuleringen finns de tillgängliga i Matlabs workspace.

Man kan också simulera från kommandofönstret i Matlab. Det finns ett antal färdiga funktioner som då kan användas.

Init1: Simulera Control1 från t=0.

Init2: Simulera Control2 från t=0.

Get: Sparar tiden och alla tillstånd där simuleringen stoppades i tstart och xstart.

Go1: Startar simulering av Control1 från tiden tstart och med tillstånden xstart.

Go2: Startar simulering av Control2 från tiden tstart och med tillstånden xstart.

De ingående blockens parametrar visas och kan ändras genom att dubbelklicka på respektive block. Beskrivning av blockets ingångar respektive utgångar finns under blockets hjälpfunktion.

Börvärdet Iref kommer från modulen Referens, varifrån den exporteras till modulen Control. För att uppstart skall fungera smidigt är börvärdet låst till till 10 under simuleringens första 0.02 sekunder. En ändring av börvärdet Iref innan den tidpunkten kommer inte att få någon verkan.

```
function [sys,x0] = motor(t,x,u,flag,Emk,Estat)
% DC-motor Equations are normalized, so EMF=n

% Emk- =1 if EMF from motor equation is to be used as output
%      else =0;
% Estat- Define a constant EMF to be used
%        instead of EMF from motor equation.

% Parameters ------------------------------------

% inom- Rated motor current
% ubase- Supply voltage;
% kbase- motor parameter;
% nbase- Rated rpm;
% Rbase- Resistance in motor;
% Lbase- Inductance in motor;
% Tload- Load torque;
% Jm- Moment of inhertia in motor;

inom=137;
ubase=311*sqrt(3);
kbase=2.77;
nbase=1437;
tbase=kbase*inom;
wbase=nbase*pi/30;
Rbase=0.25;
Lbase=6e-3;
Tload= 0;
Jm=0.6;

% Definitions ------------------------------------

if abs(flag)>0
% U- Supply voltage
% I- Current from motor
% E- EMF from motor

   U=u(1);
   I=x(1);
   E=x(2);
 end


% Updating states ------------------------------------

if abs(flag)==1
```

```
            R=Rbase;
            L=Lbase;

      if I< -0.001
         R=1000*Rbase;
          L=Lbase;
      else
          R=Rbase;
          L=Lbase;
      end
       sys(1)=(U-(E*Emk*ubase+Estat)-R*I*inom)/(L*inom);
       sys(2)=(I-Tload)/Jm*tbase/wbase;

% Outputs ----------------------------------------------

elseif abs(flag)==3

% sys(1)- current from motor
% sys(2)- EMF from motor

        sys(1)=I*inom;
        sys(2)=Estat+Emk*E*ubase;

% Initial conditions ---------------------------------------

elseif flag==0
        sys=[2;0;2;1;0;0];
        x0= [0;0];
else
        sys=[];
end
```

```
function [sys,x0] = bridge1(t,x,u,flag)
% 6-pulse thyristor bridge simple version

% Parameters ------------------------------

% delay- delay for thyrsitor to conduct;

delay=1e-6;


% Definitions -----------------------------

% Ur- R-phase;
% Us- S-phase;
% Ut- T-phase;
% zerocur- zero current detected;
% Ntrigg- next thyristor pair to be trigged;
% Nexttrigg- next time to trigg thyrsitor pair;
% Emotor- EMF of motor;

if abs(flag)>0
   Ur=u(1);
   Us=u(2);
   Ut=u(3);
   zerocur=u(4);
   Ntrigg=u(5);
   Nexttrigg=u(6);
   Emotor=u(7);
end

% Output signals----------------------------


if abs(flag)==3

% If zero current is detected output from bridge is emf from mo
tor
% else N decide which main voltage is output

        if (zerocur ==1) & (t < Nexttrigg+delay)
            N=10;
         else
          N=Ntrigg;
         end
         if N==1
             Up=Ur;
```

```
            Un=Us;

        elseif N==2
            Up=Ur;
            Un=Ut;

        elseif N==-3
            Up=Us;
            Un=Ut;

        elseif N==-2
            Up=Us;
            Un=Ur;

        elseif N==-1
            Up=Ut;
            Un=Ur;

        elseif N==0
            Up=Ut;
            Un=Us;
        else
            Up=Emotor/2;
            Un=-Emotor/2;
        end
            sys(1)=Up;
            sys(2)=Un;

% Initial conditions ---------------------------------

elseif flag==0
       sys=[0;0;2;7;0;1];
       x0= [];
else
       sys=[];
end
```

```
function [sys,x0] = bridge2(t,x,u,flag,Ron,Lon,R,C)
% 6-pulse thyristor bridge

% parameters ------------------------------------------------

% thyristor parameters-------------
%  Ih- hold current for thyristor;
%  Roff, Loff, Ron, Lon;
%--------------------------------
% R2- resistance to decharge capacitor;
% R- resitance on RC-circuit on output
% C- Capacitor on RC-ciruit on output

Ih=0.01;
Roff=1e5;
Loff=Lon/Ron*Roff*0.3;
R2=0.1;
Rc=1e4;

% Definitions ----------------------------------------

% Ur, Us, Ut - phase voltage;
% Tp1 - Tp6 - Trigger pulses to thyristor 1-6;
% Iload- actual current from motor;
% E- Emf from motor;
% Irp- current from thyristor1;
% Isp- current from thyristor3;
% Itp- current from thyristor5;
% Ucp- voltage on positive capacitor;
% Irn- current from thyristor4 ;
% Isn -current from thyristor6;
% Itn- current from thyristor2;
% Ucn- voltage on negative capacitor;


if abs(flag)>0
   Ur=u(1);
   Us=u(2);
   Ut=u(3);
   Tp1=u(4);
   Tp2=u(5);
   Tp3=u(6);
   Tp4=u(7);
   Tp5=u(8);
   Tp6=u(9);
   Iload=u(10);
```

```
    E=u(11);
    Irp=x(1);
    Isp=x(2);
    Itp=x(3);
    Ucp=x(4);
    Irn=x(5);
    Isn=x(6);
    Itn=x(7);
    Ucn=x(8);
end
if Iload< Ih
 off=1;
else
 off=0;
end
 Up=Ucp+R*(Irp+Isp+Itp-Iload);
 Un=Ucn+R*(Irn+Isn+Itn+Iload);

% ----------------------------------------

if flag==1
% Turn on/off thyristor 1,3 or 5 --------------

 if (Ur-Up)>1 & (Tp1>1) | (Irp>Ih)
   Rrp=Ron;
   Lrp=Lon;
 else
   Rrp=Roff;
   Lrp=Loff;
 end
 if (Us-Up)>1 & (Tp3>1) | (Isp>Ih)
   Rsp=Ron;
   Lsp=Lon;
 else
   Rsp=Roff;
   Lsp=Loff;
 end
if Iload<= Ih
   Iload=0;
end
 if (Ut-Up)>1 & (Tp5>1) | (Itp>Ih)
   Rtp=Ron;
   Ltp=Lon;
 else
   Rtp=Roff;
   Ltp=Loff;
```

```
 end

% ------------------------------------------

% Turn on/off thyristor 6,4 or 2 -----------------

 if (Ur-Un)<-1 & (Tp4>1) | (Irn< -Ih)
   Rrn=Ron;
   Lrn=Lon;
 else
   Rrn=Roff;
   Lrn=Loff;
 end
 if (Us-Un)<-1 & (Tp6>1) | (Isn< -Ih)
   Rsn=Ron;
   Lsn=Lon;
 else
   Rsn=Roff;
   Lsn=Loff;
 end
 if (Ut-Un)<-1 & (Tp2>1) | (Itn< -Ih)
   Rtn=Ron;
   Ltn=Lon;
 else
   Rtn=Roff;
   Ltn=Loff;
 end

% States for thyristors 1, 3 and 5 ------------

     sys(1)=((-Rrp)*Irp+Ur-Up)/Lrp;
     sys(2)=((-Rsp)*Isp+Us-Up)/Lsp;
     sys(3)=((-Rtp)*Itp+Ut-Up)/Ltp;
     sys(4)=(Irp+Isp+Itp-Iload-(Ucp-E/2)/R2*off)/C;

% States for thyristors 6,4 and 2 ----------------

     sys(5)=((-Rrn)*Irn+Ur-Un)/Lrn;
     sys(6)=((-Rsn)*Isn+Us-Un)/Lsn;
     sys(7)=((-Rtn)*Itn+Ut-Un)/Ltn;
     sys(8)=(Irn+Isn+Itn+Iload-(Ucn+E/2)/R2*off)/C;

 %   ------------------------------------------

elseif abs(flag)==3
```

```
% Outputs ----------------------------------------
        sys(1)=Up;
        sys(2)=Un;



elseif flag==0

% Initial conditions --------------------------------

       sys=[8;0;2;11;0;0];
       x0= [0;0;0;0;0;0;0;0];
else
       sys=[];
end
```

```
function [sys,x0] = trigger1(t,x,u,flag)

% trigger pulse unit for bridge1


% parameters ---------------------------------

% offset- offset from specified sample time
delta=1e-6;
offset=1e-7;

% Definitions --------------------------------

% TOT- time when trigg shall occur;
% N- what thyristor pair to be trigged;
% Ttop- time for top interupt;
% mode- variable to decide which sample time to use for next sa
mple;

if abs(flag)>0
     TOT=u(1);
     N=u(2);
     Ttop=u(3);
     mode=x(4);
   end

% updating discrete states --------------------------------

 if abs(flag)==2

      % if next sampel is at Ttop
      if mode==0
          sys=x;
          if abs(x(3)-t) < 1e-8
             sys(2)=N;
             sys(1)=TOT;
             sys(4)=1;
             sys(3)=Ttop+offset;
          end

      else

      % next sampel is at TOT
           sys=x;
         if abs(x(1)-t) <1e-8
         sys(5)=x(2);
```

```
          sys(4)=0;
        end

      end

% Output signals ------------------------------------
elseif abs(flag)==3

          sys(1)=x(5);
          sys(2)=x(1);

% Updating next sample time ---------------------------
elseif abs(flag)==4

% if mode==0 next sample is at Ttop else next sample is at TOT

          if mode==0
             if t<x(3)
              sys=x(3);
             else
              sys=x(3)+delta;
             end
          else
           if t< x(1)
             sys=x(1);
           else
             sys=x(1)+delta;
           end
          end
% Initial conditions ----------------------------------

elseif flag==0
       sys=[0;5;2;3;0;1];
       x0= [0.008;10;offset;0;10];
else
       sys=[];
end
```

```
function [sys,x0] = ftrigger(t,x,u,flag)

% trigger pulse unit for bridge2


% Parameters -------------------------------
% offset- offset from specified sample time;
delta=1e-6;
offset=2e-8;

% Definitions -------------------------------
% TOT- time when trigg occurs;
% N- which thyristor pair is trigged;
% Ttop- time for top interupt;
% Ts- next time to make a sample;
% mode- variable to decide which time to use for next sample ;
% Nnew- next thyristor pair to be trigged;

if abs(flag)>0
     TOT=u(1);
     N=u(2);
     Ttop=u(3);
     Ts=x(1);
     mode=x(2);
     Nnew=x(3);
   end

% Updating discrete states ------------------
 if flag==2

% if mode==0 sample is at Ttop else sample is at TOT
      if mode==0
          sys=x;
          if abs(Ts-t) < 1e-8
             sys(1)=TOT;
             sys(2)=1;
             sys(3)=10;
          end
      else
          sys=x;
          if abs(Ts-t) <1e-8
             sys(1)=Ttop+offset;
             sys(2)=0;
             sys(3)=N;
          else
             sys=x;
```

```
          end
      end

% Output signals -------------------------------
elseif abs(flag)==3

         sys(1)=Nnew;
         sys(2)=TOT;

% Updating next sample time --------------------
elseif abs(flag)==4
         if t<Ts
          sys=Ts;
         else
          sys=Ts+delta;
         end
% Initial conditions ---------------------------
elseif flag==0
       sys=[0;3;2;3;0;0];
       x0= [offset;0;10];
else
       sys=[];
end
```

```
function [sys,x0] = adapter(t,x,u,flag)
% Convert N from ftrigger to triggerpulses to bridge2

if abs(flag)>0

% N- What thyristor pair to be trigged;

   N=u(1);
end

% Outputs ---------------------------------------------

% Tp1- sys(1)
% Tp2- sys(2)
% Tp3- sys(3)
% Tp4- sys(4)
% Tp5- sys(5)
% Tp6- sys(6)

if abs(flag)==3
       if t==0
           N=10;
        end
         if N==1
             sys(1)=2;
             sys(2)=0;
             sys(3)=0;
             sys(4)=0;
             sys(5)=0;
             sys(6)=2;

         elseif N==2
             sys(1)=2;
             sys(2)=2;
             sys(3)=0;
             sys(4)=0;
             sys(5)=0;
             sys(6)=0;

         elseif N==-3
             sys(1)=0;
             sys(2)=2;
             sys(3)=2;
             sys(4)=0;
             sys(5)=0;
             sys(6)=0;
```

```
        elseif N==-2
            sys(1)=0;
            sys(2)=0;
            sys(3)=2;
            sys(4)=2;
            sys(5)=0;
            sys(6)=0;

        elseif N==-1
            sys(1)=0;
            sys(2)=0;
            sys(3)=0;
            sys(4)=2;
            sys(5)=2;
            sys(6)=0;

        elseif N==0
            sys(1)=0;
            sys(2)=0;
            sys(3)=0;
            sys(4)=0;
            sys(5)=2;
            sys(6)=2;
        else
            sys(1)=0;
            sys(2)=0;
            sys(3)=0;
            sys(4)=0;
            sys(5)=0;
            sys(6)=0;
        end

% Initial conditions ---------------------------------
elseif flag==0
       sys=[0;0;6;2;0;0];
       x0= [];
else
       sys=[];
end
```

```
function [sys,x0] = curr(t,x,u,flag)
% measure mean value of current with two integrals alternating,
% signal for zero current and measure pulse lenght.

% Parameters ----------------------------------

% inom- nominal current on converter;
% offset- offset from specified sample time;
% k- constant for set integral to zero;
delta=1e-6;
inom=137;
offset=2e-8;
T30=30*0.01/180;
k=1e5;
T=0.02/6;

% Definitions ---------------------------------

% Iload- actual current from motor;
% Tbot- Time for next bottom interupt;
% Ia1- value of integral 1;
% Ia2- valuse of integral 2;
% Tlead- accumulated current pulse lenght;
% int- variable to decide which integral to use;
% Sc- variable to decide which sample time to use;
% Ts- next sample time;
% Iaold- previous current pulse;
% zerocur- zero current is detected;
% Tx- current pulse length;
% Tleadold- previous current pulse length;
% Set- variable to make hold on detected zero current;

if abs(flag)>0
                Iload=u(1);
                Tbot=u(2);
                Ia1=x(1);
                Ia2=x(2);
                Tlead=x(3);
                int=x(4);
                Sc=x(5);
                Ts=x(6);
                Iaold=x(7);
                zerocur=x(8);
                Tx=x(9);
                Tleadold=x(10);
                Set=x(11);
```

```
end

% Updating continuos states ----------------------
if flag==1

% if int==1 use integral1 for mean value else use integral2.
% The integral not in use is put to zero.
                if int==0
                  if Iload>0
                   sys(1)=Iload;
                  else
                   sys(1)=0;
                  end
                   if abs(Ia2)>0.001
                      sys(2)=-k*x(2);
                   else
                      sys(2)=0;
                   end
                 else
                 if Iload>0
                   sys(2)=Iload;
                 else
                   sys(2)=0;
                end
                   if abs(Ia1)>0.001
                      sys(1)=-k*x(1);
                   else
                      sys(1)=0;
                   end
                 end

% Accumulated time with current >0 ----------------------------

       if Iload/inom<= 0
         konst=0;
       else
         konst=1;
       end;
       sys(3)=konst;

% Updating discrete states ----------------------------------


elseif flag==2
```

```
            if abs(t-Ts)<1e-8
% if Sc==0 sample time is Ttop. Update next sample time to Tbot
.
% if Sc==1 sample time is at Tbot. Switch from one integrator
% to the other one with variable int. Update sample time to Tto
p.
                if Sc==0
                   sys(1)=int;
                   sys(2)=1;
                   sys(3)=u(2)+offset;
                   sys(4)=Iaold;
                   sys(5)=0;
                   sys(6)=Tx;
                   sys(7)=Tleadold;
                   sys(8)=0;
                else
                   if int==0
                     sys(4)=Ia1;
                     sys(1)=1;
                   else
                     sys(4)=Ia2;
                     sys(1)=0;
                   end
                   sys(2)=0;
                   sys(3)=Ts+T30;
                  sys(5)=zerocur;
                  sys(6)=Tlead-Tleadold;
                  sys(7)=Tlead;
                  sys(8)=Set;
                 end


               else
% Zero curret detection ----------------------------------
% if Set==1 zero current already has occured since last Ttop.
% Set is reseted at timt Ttop.
                if (Iload/inom<-0.0001) & (Set==0)
                    sys(5)=1;
                    sys(8)=1;
                else
                    sys(5)=zerocur;
                    sys(8)=Set;
                end;
                sys(1)=int;
                sys(2)=Sc;
```

```
                sys(3)=Ts;
                sys(4)=Iaold;
                sys(6)=Tx;
                sys(7)=Tleadold;

            end

% Output signals ----------------------------------

elseif abs(flag)==3

            if int==0
              sys(1)=Ia1;
             else
               sys(1)=Ia2;
             end

             sys(2)=Iaold;
             sys(3)=zerocur;
             sys(4)=Tx;

% Updating next sample time -----------------------------

elseif abs(flag)==4
             if t<Ts
               sys(1)=Ts;
              else
                sys(1)=Ts+delta;
              end

% Initial conditions ------------------------------------

elseif flag==0
             sys=[3;8;4;2;0;0];
             x0= [0;0;0;0;0;offset;0.005+offset;1;0;0;0];
else
             sys=[];
end
```

```
function [sys,x0] = sync(t,x,u,flag)

% detect positive zero crossing for S-phase;

% Parameters -------------------------------------------
 delta=1e-6;
 T270=270*0.01/180;

% Definitions ------------------------------------------

% S-  actual value for S-phase;
% Sold- value for S-phase at prvious sample;
% Timezero- Time at Talpha=0 for Urs;
% Tcycle- Period for S-phase;
% Told-Previous time with zero crossing for S-phase;
% Ts- Time for next sample;
if abs(flag)>0

 S=u(1);
 Sold=x(1);
 Timezero=x(2);
 Tcycle=x(3);
 Told=x(4);
 Ts=x(5);
end
% Updating discrete states  ----------------------------
 if flag==2

      if abs(t-Ts) < 1e-8

      % if positive zero detection next sample at 95% of period

            if ((S>=0) & (Sold<=0))
               sys(1)=S;
               sys(2)=t+T270;
               sys(3)=t-Told;
               sys(4)=t;
               sys(5)=Ts+0.95*Tcycle;
            else
               sys(1)=S;
               sys(2)=Timezero;
               sys(3)=Tcycle;
               sys(4)=Told;
               sys(5)=Ts+1e-4;
            end
      else
```

```
            sys=x;
      end

% Outputsignals ---------------------------------------
elseif abs(flag)==3

            sys(1)=Timezero;
            sys(2)=Tcycle;

% Updating next sample time ----------------------------

elseif abs(flag)==4
           if t<Ts
             sys(1)=Ts;
           else
             sys(1)=Ts+delta;
           end

% Initial conditions ----------------------------------

elseif flag==0

       Tzi=30*0.01/180;
       Tci=0.02;
       Toi=-240*0.01/180;
       Tsi=0.0064;
       sys=[0;5;2;1;1;0];
       x0= [0;Tzi;Tci;Toi;Tsi];
else
       sys=[];

end
```