

**Automatväxelsystem för lastbil**  
-modellering, simulering och integrering i fordonsmodell

Examensarbete utfört i Fordonssystem  
vid Tekniska Högskolan i Linköping  
av

**Johan Tengroth**

Reg nr: LiTH-ISY-EX-2069

Handledare: Urban Sjädhahl  
Tony Sandberg  
Examinator: Lars Nielsen  
Linköping, 12 november 1999.

## **Sammanfattning**

Scania håller på att bygga upp nästa generation fordonssimuleringssystem för lastbilar. En viktig komponent i detta system är valet av växel. Scania har ett automatväxelsystem, OptiCruise, som automatiskt väljer och genomför växlingar i en manuell växellåda.

Syftet med examensarbetet var att implementera en växellådsmodul som gör att fordonmodellen väljer växlar på samma sätt som OptiCruise. Fordonsmodellen kan då användas dels för att studera bränsleförbrukning och emissioner, dels för att pröva nya regleralgoritmer och styrsystem.

Fordonmodellen byggs upp med komponenter i simuleringsprogrammet Dymola, som har valts efter utvärdering. Som stöd till modulerna implementeras vissa delar i C-kod. MATLAB används som användargränssnitt och för att lagra data.

En växellådsmodul har modellerats i Dymola. Automatvalslogiken för växelval har implementerats i C. Gränssnitt mot övriga moduler har definierats och alla signaler har samlats i en koordinator.

Verifieringar har utförts, dels genom helbilssimuleringar, dels genom verkliga körningar. Växlingarna sker som förväntat. Resultaten vid längre landsvägskörningar är mycket goda. De begränsningar som satts på systemet visar sig; resultaten har sämre överensstämmelse vid korta simuleringar i låga hastigheter.

Nyckelord: växellåda, OptiCruise, Dymola, modellering, simulering

## **Abstract**

Scania is now building the next generation of vehicle simulation system for heavy trucks. An important component in such a system is the selection of gears. Scania produces an automatic gearchange system , called OptiCruise, which automatically selects the gear in a manual gearbox.

The purpose of this thesis was to implement a gearbox module so that the vehicle model selects gears like OptiCruise does. The vehicle model can then be used both for studying fuel consumption and emissions, as well as for testing new control algorithms and systems.

The vehicle model is built with components in the simulation software Dymola, which has been chosen after evaluation. As support to the modules some parts are implemented in C-code. MATLAB is used as user interface and for data storage.

A gearbox module has been modelled in Dymola. The automatic gearchange logic for gear selection has been implemented in C. Interface towards other modules has been defined and all signals have been collected in a coordinator.

Model verifications have been done, partly by complete vehicle simulations, partly by real driving. The gearchanges behave as expected. The results in long distance highway cruising are very good. The limitations of the model are visible; short simulations in low speed are less significant.

Keywords: gearbox, OptiCruise, Dymola, modelling, simulation

## **Förord**

Examensarbetet avslutar mina studier på programmet Teknisk Fysik och Elektroteknik vid Linköpings Tekniska Högskola (LiTH). Arbetet har utförts inom gruppen transmissionsstyrsystem (DTE) vid Scania Tekniskt Centrum i Södertälje.Handledare på Scania har varit Urban Sjädhall och Tony Sandberg. Examinator på LiTH var Professor Lars Nielsen, avdelningen för Fordonssystem vid institutionen för systemteknik (ISY).

Ett stort tack till dessa personer för all hjälp med arbetet.

Södertälje i november 1999

Johan Tengroth

# INNEHÅLL

<b>1</b>	<b>INLEDNING.....</b>	<b>1</b>
1.1	BAKGRUND OCH SYFTE .....	1
1.2	BEGRÄNSNINGAR .....	1
1.3	MÅLBESKRIVNING.....	1
<b>2</b>	<b>DYMOLA.....</b>	<b>3</b>
2.1	ALLMÄNT.....	3
2.2	ETT ENKELT MODELLERINGSEXEMPEL .....	4
2.2.1	<i>Dymodraw</i> .....	4
2.2.2	<i>Dymola</i> .....	6
2.2.3	<i>Dymosim</i> .....	7
2.2.4	<i>Dymoview</i> .....	7
2.2.5	<i>Begränsningar i Dymola</i> .....	8
<b>3</b>	<b>OPTICRUISE AUTOMATVÄXELSYSTEM .....</b>	<b>9</b>
3.1	ALLMÄNT.....	9
3.2	SCANIA DRIVLINA .....	10
3.2.1	<i>Växellåda GRS900R</i> .....	11
3.3	VÄXELVALET – AUTOMATVALSLOGIKEN .....	11
3.3.1	<i>Baslogik för växelpunksberäkning</i> .....	12
3.3.2	<i>Normalväxlingspunkter</i> .....	12
3.3.3	<i>Basväxlingspunkten</i> .....	13
3.3.4	<i>Accelerationsmodifieraren</i> .....	13
3.3.5	<i>Cruisecontrolmodifieraren</i> .....	13
3.3.6	<i>Motorbromsprogram</i> .....	13
3.3.7	<i>Stegvalslogiken</i> .....	14
3.3.8	<i>Dämplogiken</i> .....	14
3.4	GENOMFÖRANDET AV VÄXLINGEN.....	14
3.5	SLUTSATSER FÖR MODELLENS DEL.....	15
<b>4</b>	<b>STARS – SCANIA TRUCK AND ROAD SIMULATION .....</b>	<b>17</b>
<b>5</b>	<b>DESIGN AV VÄXELLÅDSMODUL.....</b>	<b>19</b>
5.1	STRUKTUR.....	19
5.1.1	<i>Verkningsgrad och utväxlingstal</i> .....	19
5.1.2	<i>C-funktioner</i> .....	20
5.1.3	<i>Modellklassen</i> .....	22
5.2	ALTERNATIVET MED VÄXLINGSFÖRLOPP.....	23
5.2.1	<i>Petrinät</i> .....	23
5.2.2	<i>Växlingsförloppet med Petrinät</i> .....	24
5.3	GRÄNSSNITT STARS .....	25
5.4	BERÄKNING AV BRÄNSLEFÖRBRUKNINGEN.....	26
<b>6</b>	<b>RESULTAT OCH UTVIDGNINGAR.....</b>	<b>27</b>
6.1	VERIFIKATION AV MODELL.....	27
6.1.1	<i>Steg i referenshastigheten</i> .....	29
6.1.2	<i>Motorbromsprogrammets inverkan</i> .....	30
6.1.3	<i>Hill- och ekonomiprogrammen</i> .....	31

6.1.4	Bränsleförbrukning.....	34
6.1.5	Byte av växellåda.....	35
6.2	REFLEKTION OCH UTVIDGNINGAR.....	37
<b>7</b>	<b>REFERENSER.....</b>	<b>38</b>
<b>8</b>	<b>BILAGOR.....</b>	<b>39</b>
8.1	OPCGEARBOX.....	39
8.2	BERÄKNING AV BRÄNSLEFÖRBRUKNING.....	41

## FIGURER

Figur 2.1.	Arbetsgången i Dymola.....	4
Figur 2.2.	Dymodraw, med biblioteksfönster till vänster. Överst för transmission och under signalbiblioteket.....	5
Figur 2.3.	Använda klasser. De ärvda ekvationerna i nedre halvan och de egna, som dock saknas i exemplet, i de övre.....	6
Figur 2.4.	Dymolafönstret.....	7
Figur 2.5.	Dymoview plotfönster.....	8
Figur 3.1.	OptiCruise växelspak i lastbil. Notera hillknappen nertill.....	10
Figur 3.2.	Exempel på drivlina med 12-liters motor och växellåda GRS900R med retarder..	10
Figur 3.3.	Växellådan GRS900R. Range-splitter låda med retarder.....	11
Figur 3.4	Blockschema över delarna i växelvalskoden.....	12
Figur 4.1.	Schematisk bild av STARS olika beståndsdelar.....	17
Figur 6.1	Hel fordonsmodell. Växellådsmodulen heter OPCGearbox.....	27
Figur 6.2.	Ett lastat fordon (20 ton) svar på steg i referenshastigheten.....	29
Figur 6.3.	Ett olastat fordon hoppar över växlar och tar maximalt antal steg (3 upp).....	30
Figur 6.4.	Acceleration i uppförbacke med konstant lutning 5% och ekonomiprogram.....	32
Figur 6.5	Samma körning som i figur 6.4, men med hillprogram. Motorns högre varvtal syns tydligt.....	33
Figur 6.6.	Steg i referenshastigheten med 8 växlad låda. Varvtalsomfånget ökar på varje växel.....	36

## 1 Inledning

*Detta kapitel innehåller en introduktion till examensarbetet med syfte, mål och begränsningar i arbetet.*

### 1.1 Bakgrund och syfte

Med hjälp av datormodeller av fordon, vägtopografi och förare kan man simulera drivkraftbehovet hos ett fordon. Det kan användas för att beräkna till exempel restid och bränsleförbrukning. Scania bygger nu upp nästa generation simuleringspaket, STARS – Scania Truck And Road Simulation. Systemet kan också användas för att prova nya konstruktions- och reglerprincipers påverkan på fordonet, eller delar av det. En viktig komponent är strategin för val av växel. När i tiden växelvalet ska ske och till vilken växel är en komplex fråga. Den är viktig eftersom växelvalsstrategin kommer att påverka såväl bränsleförbrukning som komfort, backtagningsförmåga och körbarhet.

Syftet med arbetet är att kunna studera inverkan av växlingsstrategi på fordonet, genom att utifrån befintliga och tilltänkta växellådor utveckla en växellådsmodul som kan sättas in i detta simuleringspaket och implementera en verklig växelstrategi. Vidare är syftet att studera alternativa växelstrategiers inverkan på bland annat bränsleförbrukning. Det är önskvärt med en så enkel modell av fordonet som möjligt, så att den blir snabb att simulera. Gränsen för enkelhet sätts dels av hur detaljrik nivå man vill studera, dels av hur väl modellen efterliknar verkligheten.

### 1.2 Begränsningar

Modelleringen av fordonet ska ske i programmet Dymola (se vidare i kapitel 2). Programmet har valts efter utvärdering, då det visat sig fylla Scantias syften. Därmed är det naturligt att även bygga växellådsmodulen i Dymola.

Växellådans mekanik förenklas, eftersom det inte bedöms som nödvändigt med en mer detaljrik modell för att den ska fylla sitt syfte. Vill man studera växellådan djupare eller det visar sig att modellen inte fungerar tillräckligt bra, kan man i ett senare skede modellera växellådan mer ingående. Tyngdpunkten ligger istället på växelvalslogiken.

### 1.3 Målbeskrivning

Mer specifikt är examensarbetets mål följande:

- En växellådsmodul ska implementeras i Dymola.
- Växelval ska ske på samma sätt som OptiCruise (OPC) gör i verkligheten. Det innebär att det ska implementeras växelpunktsberäkning och stegvalsberäkning. Följande önskemål finns:
  1. Tabellupplägget i verklig OPC kod ska om möjligt behållas.
  2. Undersöka om motorbromsprogram (se 3.3.6) kan/behöver användas.
  3. Möjlighet att i Dymola välja hill (se 3.1) eller ekonomi program.
  4. Data för olika motor-växellådscombinationer används. Koden ska vara generell för dessa olika combinationer.
- Gränssnitt mot övriga moduler ska definieras och implementeras. Det innebär till exempel att bestämma signalvägar och fördela ansvarsområden mellan modulerna.

- Verifiering av växellådsmodellen görs genom modellkörningar av komplett fordon i typiska körfall. Eventuellt också jämförande körningar i verkligt fordon.



## 2 Dymola

*Simuleringsystemet STARS byggs bland annat upp i programmet Dymola. Växellådsmodulen ska byggas i Dymola och verifieringskörningar kommer att göras i samma miljö. Därför beskrivs i detta kapitel grunderna i programmet.*

### 2.1 Allmänt

Dymola – Dynamic Modeling Laboratory är ett objektorienterat språk och program för modellering och simulering av stora system. Modellerna byggs hierarkiskt upp i bibliotek, modellklasser och modeller. Med hjälp av modellklasser kan modeller återanvändas genom att ärva egenskaper från den ursprungliga modellklassen och sedan bygga på med nytt. Ändras modellklassen kommer också de modeller eller modellklasser där den ändrade klassen används att uppdateras. Det bygger på objektorienteringens grundegenskap arv.

Till Dymola finns färdiga bibliotek som innehåller grundkomponenter inom olika områden, till exempel bibliotek för elektriska komponenter, regulatorer, mekaniska komponenter och tillståndsmaskiner. Med hjälp av dessa och egna komponenter bygger man sina modeller, med fördel bestående av egna modellklasser ingående i egna bibliotek. Dymola är non-casual, vilket betyder att ekvationer skrivs in på valfri form, oavsett vilken variabel man söker lösningen på. Enkelt uttryckt innebär det att man slipper arbetsamma räkningar med att överföra ekvationer, som beskriver system, till tillståndsform (som man gör i till exempel Simulink), avgöra vilka variabler som är insignaler och utsignaler och så vidare. I Dymola sätts ekvationer och differentialekvationer automatiskt samman till en ekvation av typen:

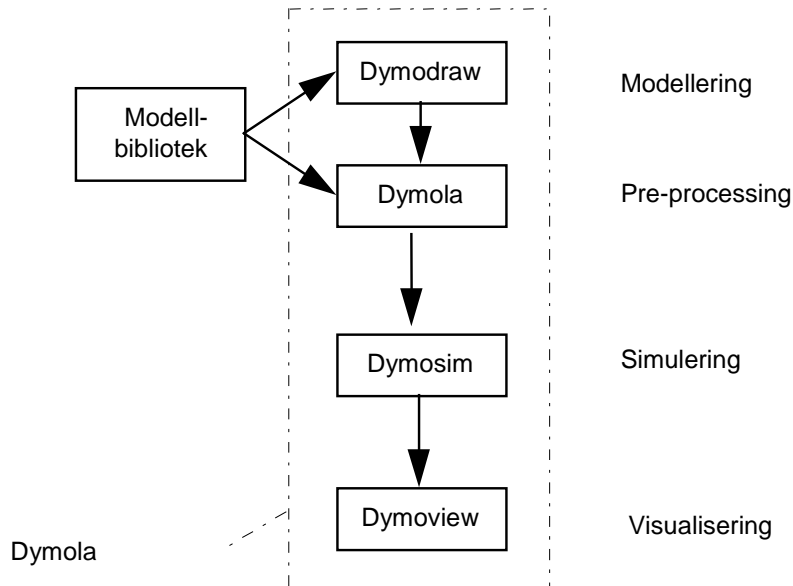
$$0 = f(\dot{x}, x, y, u) \quad (2.1)$$

där  $x$  är vektorn av okända som förekommer tidsdifferentierade,  $y$  är vektorn av övriga okända. Ekvation 2.1 kallas differentialalgebraisk form (DAE).

Dymola består av fyra delprogram: Dymodraw, Dymola, Dymosim och Dymoview. I Figur 2.1 visas den normala arbetsgången vid modellering och simulering av ett system. Arbetet börjar med modellbygge i Dymodraw. Som hjälp finns modellbibliotek där komponenter kan plockas ifrån. Tillägg kan göras till dessa komponenter och egna kan skapas. När modellen är klar laddar man in den i Dymola. Här genereras alla ekvationer för de komponenter och sammankopplingar som tidigare ritats upp. Därefter sorteras ekvationerna och kod skapas för det valda simuleringspråket. Allt detta kallas kompilering av modellen. Om Dymosim är valt som simuleringspråk startas simuleringen från Dymolas arbetsfönster efter slutförd kompilering. Resultatet av simuleringen är en fil, som Dymoview kan läsa. I Dymoview finns plot- och animeringsverktyg där resultaten kan studeras.

Eftersom Dymosim är ett fristående program, finns också matlabfiler (\*.m) för att göra det möjligt att skapa indata till simuleringen i MATLAB, starta den och även studera resultaten i MATLAB. Arbetsgången blir då att först skapa och kompilera modellen i Dymodraw respektive Dymola, därefter gå in i MATLAB och ropa på Dynasim med en indatafil man själv skapat, innan man slutligen kan studera resultatet. På så vis kan simuleringen styras från MATLAB och MATLABs alla möjligheter att behandla data kan nyttjas. Ett alternativ, om man vill använda den normala arbetsgången men ändå plotta resultat i MATLAB, är att läsa in resultatfilen (en \*.mat fil).

Dymola har även en enkel inbyggd C-kompilator som möjliggör användandet av C-program i sin modell. I Dymodraw kan man helt enkelt ropa på den önskade C-funktionen med de argument som behövs, lägga resultatet i en variabel och använda i modellen.



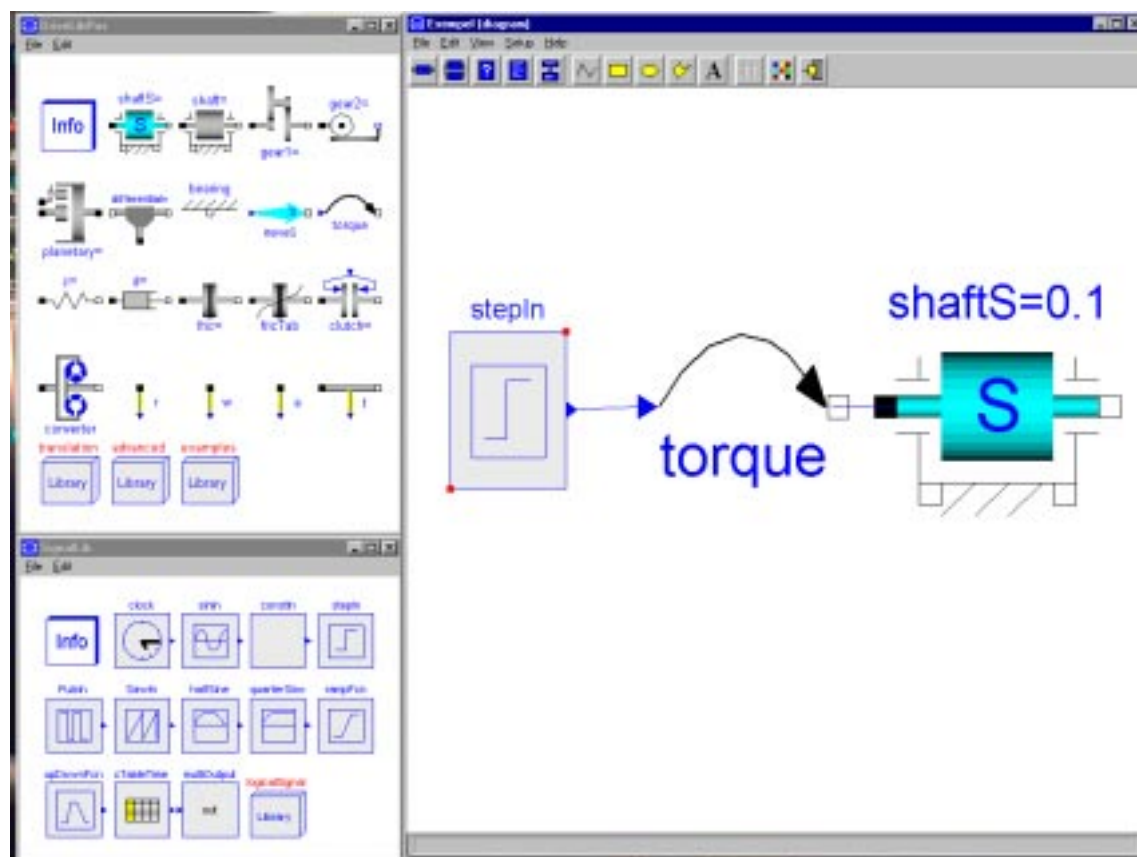
**Figur 2.1.** Arbetsgången i Dymola.

## 2.2 Ett enkelt modelleringsexempel

För att illustrera arbetsgången och beskriva de olika delprogrammen genomförs här ett enkelt exempel, vilket ger en bra introduktion till programmet och en nödvändig grund för att förstå senare kapitel. Exemplet visar dock inte alla möjligheter med Dymola.

### 2.2.1 Dymodraw

Dymodraw är ett grafiskt gränssnitt där modeller byggs upp, komponenter plockas från de olika biblioteken och binds samman genom att dra linjer. Det är intuitivt och likt hur systemet ser ut i verkligheten. Så länge man nöjer sig med grundkomponenternas egenskaper behövs ingen djupare matematisk kunskap om systemet för att bygga modellen.

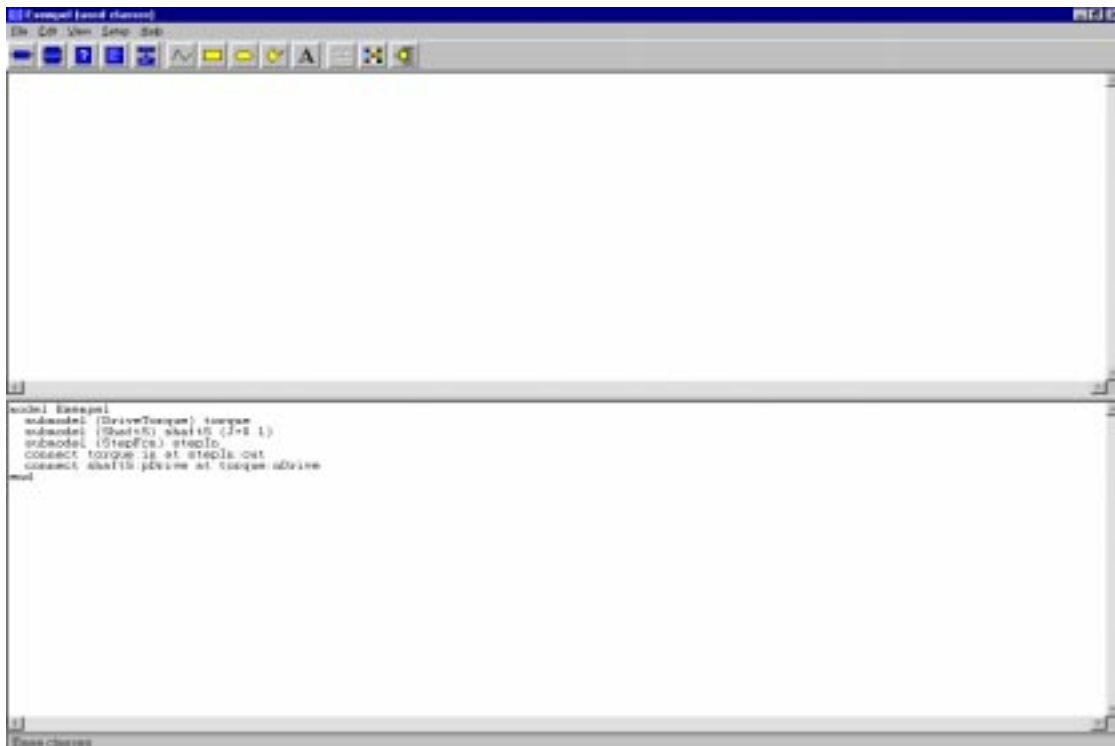


**Figur 2.2.** Dymodraw, med biblioteksfönster till vänster. Överst för transmission och under signalbiblioteket.

Här används ”dra och släpp” teknik för att, som i figur 2.2, bygga en roterande tröghet som drivs av ett moment. Momentet i sin tur styrs i exemplet av en stegsignal.

Varje modell och modellklass består av fem lager (i figur 2.2 visas diagramlagret):

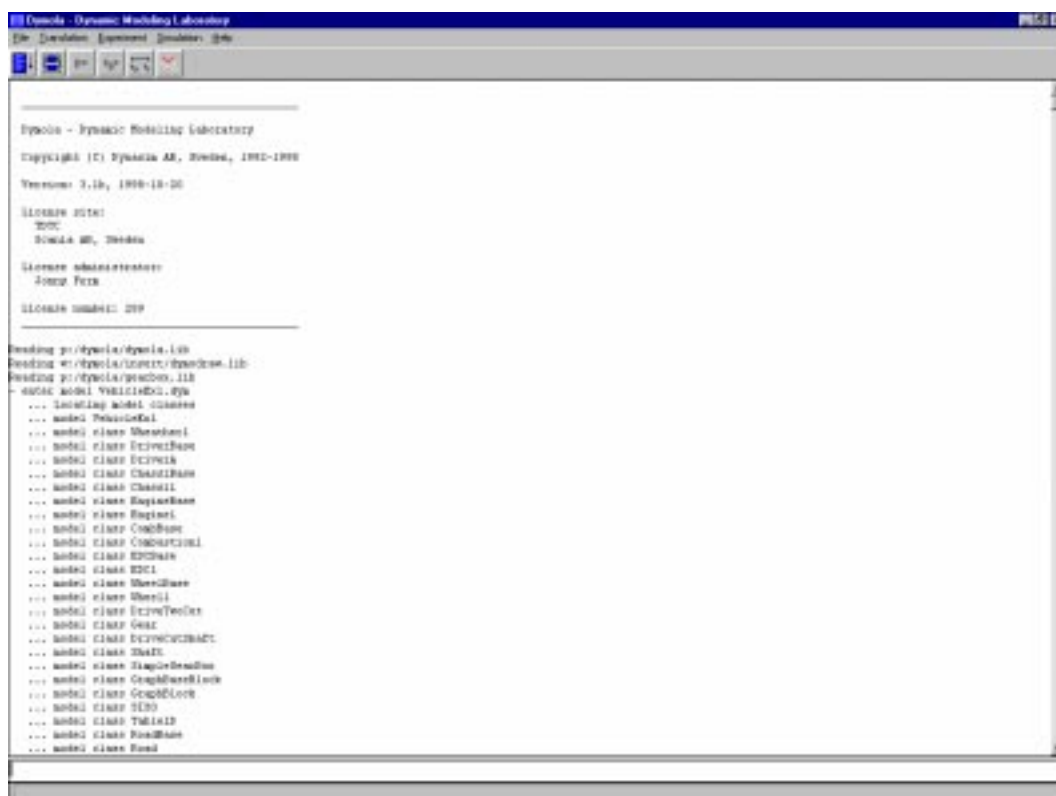
- Ikonlager: Här ritas symbolen för komponenten, på det vis den kommer att se ut då den används av överliggande modells diagramlager. Symbolen ritas fritt och har ingen annan funktion än att tydliggöra för användaren.
- Diagramlager: Specificerar ingående modellklasser samt förbinder dem med linjer. Utifrån dessa linjer genererar programmet de fysiska kopplingarna som intensitets- och flödesvariabler (se referens 12).
- Dokumentationslager: Fri text som beskriver komponenten skrivs ner här och är endast en hjälptext för användaren.
- Ekvationslager: Alla egna ekvationer, parametrar och variabler skrivs här. Lagret syns i övre halvan av figur 2.3.
- Använda klasser: Här visas de ekvationer som de använda klasserna är uppbyggda av och således är de ekvationer som modellen har ärvt se figur 2.3.



**Figur 2.3.** Använda klasser. De ärvda ekvationerna i nedre halvan och de egna, som dock saknas i exemplet, i de övre.

## 2.2.2 Dymola

Alla linjer mellan komponenter betraktas som fysiska av Dymola och ekvationerna bygger på att alla variabler i de olika systemen kan delas in i intensitets- och flödesvariabler. Exempelvis i det elektriska fallet är flödet ström och intensitet spänning. Ekvationerna blir då Kirchoffs första respektive andra lag. Övriga ekvationer hämtas från ekvationslagret. Därefter genereras kod, förinställt för Dymosim.



**Figur 2.4.** *Dymolafönstret.*

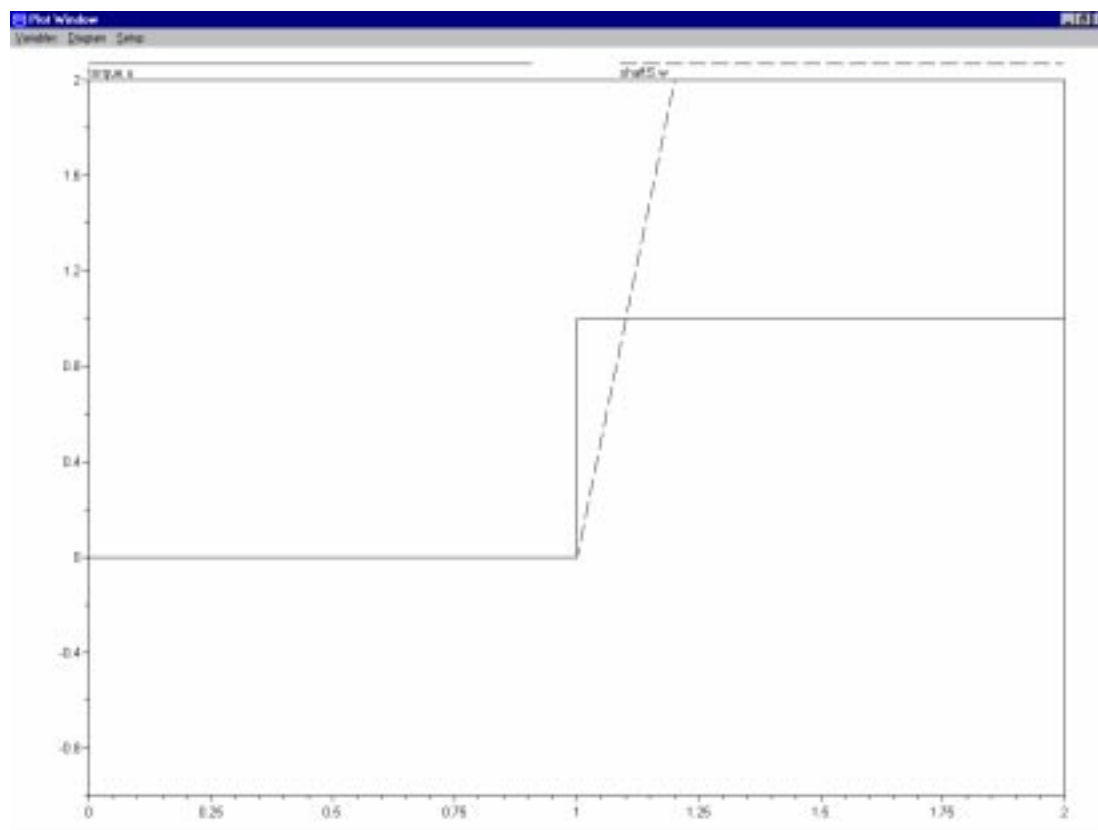
I Dymolafönstret laddas modell in, eventuellt justeras initialvärden och parametrar, för att därefter kompilera och simulera. I menyerna kan diverse inställningar göras och under resans gång fås information om förloppet, felinformation med mera.

### 2.2.3 *Dymosim*

Dymosim är simuleringsmotorn som används från Dymolafönstret. Den är självständig och kan till exempel användas från MATLAB som tidigare beskrivits. Normalt används Dymosim med Dymola. Den har olika lösningsalgoritmer som kan väljas, alla med sina för- och nackdelar, som löser system beskrivna av ordinära differentialekvationer, algebraiska differentialekvationer och överbestämda algebraiska differentialekvationer. Resultatet från Dymosim kan plottas i Dymoview eller MATLAB.

### 2.2.4 *Dymoview*

Programmet läser resultatfilen och man väljer vilka variabler som ska plottas, med valfri variabel på horisontella axeln.



**Figur 2.5.** *Dymoview plotfönster.*

I figur 2.5 är exemplet simulerat och den heldragna linjen är  $u$  (insignalen, dvs steget) i modellklassen torque som går från noll till ett vid tiden en sekund. Den streckade kurvan "shaftS.w" är rotationshastigheten  $w$  i radianer per sekund hos modellklassen shaftS (det vill säga rotationströgheten). Trögheten börjar en konstant acceleration då momentet läggs på.

### 2.2.5 Begränsningar i Dymola

En begränsning i Dymola som har väsentlig påverkan för användandet av C-funktioner är att gränssnittet mellan C och Dymola endast tillåter funktioner i C som returnerar en variabel. Det går alltså inte att exempelvis låta en C-funktion som Dymola anropar returnera en vektor eller en post. Det visar sig att detta ställer till en del problem, när det gäller gränssnittet mellan C och Dymola. Det är dock åtgärdat i version 4.0 av Dymola.

### 3 OptiCruise automatväxelsystem

*Jag beskriver här principerna i det växelsystem som finns i verkligheten och det leder fram till slutsatser för implementeringen i modellen.*

#### 3.1 Allmänt

Växlingar i tunga fordon består idag av fyra olika principer:

- Manuell växellåda. Föraren växlar själv med hjälp av koppling.
- Manuell växellåda med automatisk koppling.
- Automatlåda. Växling sker helt automatiskt med en helt annan växellådskonstruktion.
- Automatväxelsystem med motorstyrning. Mjukvara styr växlingen hos en manuell låda.

Scantias automatväxelsystem OptiCruise (OPC) använder sig av den sista principen. En vanlig manuell växellåda används där växelspaken tas bort och i stället sätts en ”joystick” in. Med den väljs automat eller manuell. I manuell mode sker växling genom att föra spaken åt vänster eller höger för ned- eller uppväxling. I automatmode räknar styrlogiken själv ut växlingarna och föraren behöver inte göra någonting. Kopplingen används endast vid start och stopp. Då man startar väljer man startväxel, trycker ner kopplingen och startar på vanligt sätt. Fördelarna med detta system jämfört med de andra principerna ovan är:

- Samma växellåda och koppling som i det manuella fallet kan användas i lastbilen.
- Förlusterna och de relativt höga kostnader som finns i en automatlåda undviks, vilket ger bra ekonomi.
- Föraren kan välja att växla manuellt om han så önskar.
- Det enda tillägg som behövs till en manuell låda är en aktuator som flyttar växelföraren, samt hård- och mjukvara för automatvalslogiken.



**Figur 3.1.** *OptiCruise växelspak i lastbil. Notera hillknappen nertill.*

Utöver spaken finns en hillknapp. Genom att trycka på den aktiveras ett annat växelprogram. I själva verket är det samma logik, med skillnaden att tabellerade data byts ut mot en annan uppsättning poster. Hillknappen ska användas som namnet antyder, vid körning i branta uppförsbackar. Då görs växlingarna snabbare och vid lite högre varv. Även stegvalet justeras då föraren använder hill.

### **3.2 Scania drivlina**

Scania har många olika drivlinekonfigurationer på marknaden. Med drivlina menas de delar som driver fordonet, från motor via växellåda, kardanaxel, centralväxel, drivaxlar till hjulen.



**Figur 3.2.** *Exempel på drivlina med 12-liters motor och växellåda GRS900R med retarder.*



Beroende på vilken motor-växellådskombination man har finns olika data för OptiCruise att använda för att beräkna växelbyten.

### 3.2.1 Växellåda GRS900R

En växellåda som kommer att användas i helbilssimulering är GRS900R (figur 3.3). Den har 12 växlar framåt plus två krypväxlar, vilket ger en utväxling mellan 16.38:1 (kryp låg) till direktväxeln 12 (utväxling 1:1). Växelföraren har fem lägen: kryp, back, 1, 2, och 3. Dessutom en split som delar varje läge i två och även rangeväxel som ger dubbelt utväxlingsområde för 1, 2, och 3:an. Det ger totalt  $2 \times 3 \times 2 = 12$  växlar framåt plus kryp.

Växellådan är förutom ovanstående utrustad med en hydraulisk broms, retarder, som används för längre kontinuerliga inbromsningar. Exempelvis för konstantfarthållning i utförsbacke. På så sätt kan hjulbromsarna sparas till kortvariga inbromsningar. Både bromslitaget och risken för värmemattning (fading) minskas avsevärt. Bromseffekten kyls bort via motorns ordinarie kylsystem.



**Figur 3.3.** Växellådan GRS900R. Range-splitter låda med retarder.

### 3.3 Växelvalet – automatvalslogiken

Automatvalslogikens uppgift är att svara på frågorna: När ska jag växla? och till vilken växel ska jag växla? För att svara på dessa frågor behövs flera signaler som beskriver tillståndet hos fordonet. Exempelvis nuvarande växel och motorvarvtal. Det behövs också ett strukturerat sätt att koda detta på. Denna beräkning görs med frekvensen 10 Hz, det vill säga var 0.1:e sekund kontrolleras om en växling ska göras.

Resultatet är en struktur i följande delar:

- Baslogik för växelpunktsberäkning
- Baslogik för stegvalsberäkning
- Dämplogik för att undvika onödiga växlingar

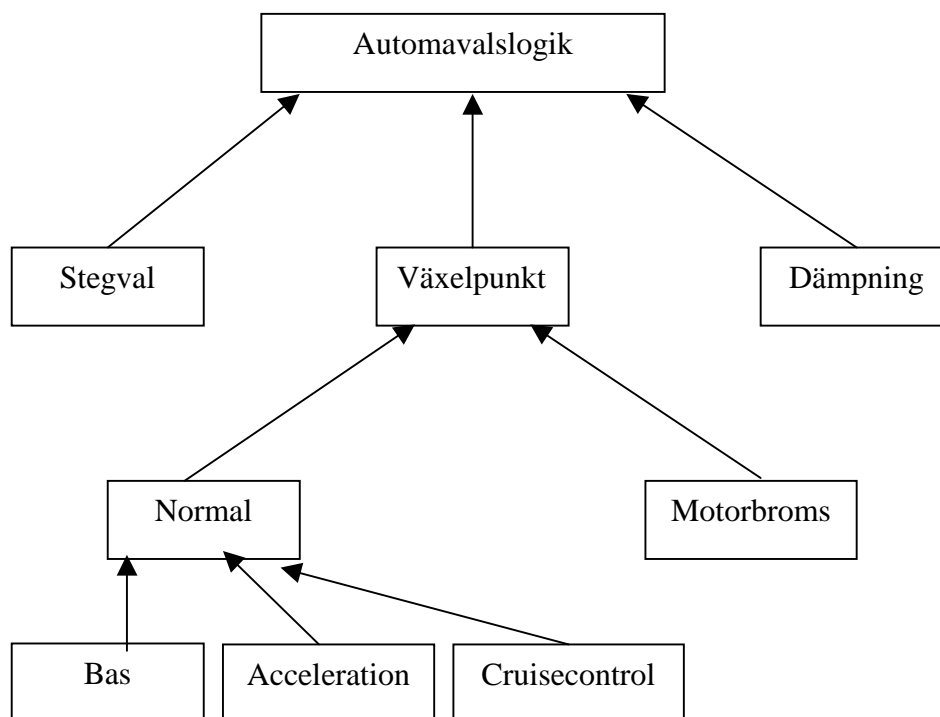
### 3.3.1 Baslogik för växelpunktsberäkning

Med växelpunkt menas det motorvarvtal vid vilket växling ska ske. Vid varje anrop till automatvalslogiken beräknas växelpunkter för 3, 2 och 1 steg upp samt 3, 2 och 1 steg ner. Sedan avgör baslogiken för stegval vilken växel som blir aktuell. I de flesta fall blir resultatet att inte växla alls, eftersom anropen görs med frekvensen 10 Hz.

Växelpunktsberäkningen är indelad i två typer:

- Normalväxlingspunkter
- Motorbromsväxlingspunkter

I Figur 3.4 visas hur delarna hänger samman, men alla delar används inte i alla driftsfall. Beskrivningen här kommer att fokusera på de delar som används i simuleringen.



**Figur 3.4** Blockschema över delarna i växelvalslogiken.

### 3.3.2 Normalväxlingspunkter

Normalväxlingspunkten räknas ut som en summa:

$$\text{Normalväxlingspunkten} = \text{Basväxlingspunkten} - \text{Accelerationsmodifieraren} + \text{Cruisecontrolmodifieraren} \quad (3.1)$$

### 3.3.3 Basväxlingspunkten

Basväxlingspunkter finns lagrade i poster, men bara för en utväxling. Med det menas utväxling före växling. Basväxlingspunkten beror alltså på vilken utväxling man ligger på. Sedan finns korrigeringsfaktorer lagrade i promille. En faktor för ändringen i riktning mot högre växlar (lägre utväxlingstal) och en för riktning mot lägre växlar. Dessa faktorer används för att korrigera basväxlingspunkten då man ligger på en utväxling som avviker från det tabellerade.

Beräkningen blir:

$$\text{Korrigerad basväxlingspunkt} = \text{Basväxlingspunkten} * (1 + \text{korrigeringsfaktor} * (\text{skillnad i utväxling})) \quad (3.2)$$

Funktionen som löser detta i C heter Ratio\_Corr.

Ett exempel:

Tabellutväxlingen: 3.75 dvs lagrade punkter gäller för växel 6 som har utväxling 3.75

Parametrar: {2000, -20, 30} = {varvtal, faktor högre växel, faktor lägre växel}

Detta betyder att för utväxling 3.75 sker växling vid 2000 varv/min. Antag nu att man ligger på en högre växel med utväxling 2.75, då gäller att växling sker vid  $2000 * (1 + (-20) * (3.75 - 2.75) / 1000) = 1960$  varv/min.

### 3.3.4 Accelerationsmodifieraren

Accelerationsmodifieraren ger en linjär påverkan på växlingspunkten, proportionell mot kardanaxelaccelerationen, upp till en viss nivå och däröver är påverkan konstant. Även dessa parametrar korrigeras med Ratio\_Corr på samma sätt som beskrivits ovan. Accelerationsmodifieraren kan ge både positivt och negativt bidrag beroende på om man har acceleration eller retardation. Observera tecknet i (3.1): ju högre acceleration, desto tidigare uppväxling (växlingspunkten trycks ner).

### 3.3.5 Cruisecontrolmodifieraren

Påverkan på växlingspunkten är proportionell mot motormomentet efter att en viss knäpunkt överskridits. Under knäpunkten är påverkan noll. Parametrarna korrigeras inte, utan är oberoende av aktuell växel. Syftet är att påverka växelpunkten så att vid ett högt motormoment växla upp senare (vid ett högre varvtal) eftersom det då går tyngre att driva fordonet.

### 3.3.6 Motorbromsprogram

I vissa fall behövs ett visst varvtal för att få en effektiv bromsning, varvid systemet får en begäran om att övergå i motorbromsprogram. Det fungerar som så att växelpunkten finns tabellerad direkt och påverkas inte av antalet växelsteg. Korrigering görs med Ratio\_Corr, men varken accelerationsmodifieraren eller cruisecontrolmodifieraren påverkar.

Behovet av motorbromsprogram beror dels på avgasbromsen, dels på retardern. Avgasbromsen bromsar bättre ju högre varv motorn har, eftersom bromseffekten är proportionell mot produkten av moment och varvtal. I retarderns fall sätter kylvattentemperaturen en gräns för uthålligheten hos bromsningen. Det beror på att vattnet i sin tur kyler hydrauloljan som trycks in i retardern och blir varm när den tar upp rörelseenergi. Med högre motorvarv uppnås bättre kyleffekt. Motorbromsprogrammet har således inget med

bättre växelval att göra när det gäller till exempel komfort eller ekonomi, utan är ett krav som andra system ställer på växlingen.

Växelpunkter som till slut blir är antingen normalväxelpunkterna eller motorbromsprogrammet växelpunkter. Man får alltid 6 växelpunkter uträknade: 3, 2, 1 upp samt 1, 2, 3 ner.

### 3.3.7 Stegvalslogiken

Växelpunkter finns uträknade då beräkningen kommer till stegvalslogiken, vars uppgift är att välja till vilken växel växlingen ska ske. För varje stegval finns två värden lagrade, accelerationsfaktor och nivå. Dessa tillsammans med uppgifter om kardanacceleration och kardanmoment gör att man kan bilda ett villkor. Först prövas alltid de långa stegen, 3 upp respektive 3 ner. Om dessa inte blir uppfyllda prövas istället 2 steg. Om inte heller 2 steg blir uppfyllt blir 1 steg det giltiga stegvalet. Principen är den att ju högre kardanaccelerationen är och ju lägre kardanmoment man har, desto lättare är det att få 3 steg upp godkänt. Det innebär till exempel att en lätt bil (olastad) vid hög acceleration hoppar över växlar. Det är precis så man kör då man växlar manuellt. Omvänt i en tung bil som inte orkar accelerera fort utnyttjas varje växel att dra på. Resultatet av stegvalslogiken är en godkänd växel uppåt och en godkänd växel nedåt. Sedan beror det på vilken motsvarande växelpunkt som eventuellt är uppfyllt om en växling blir av eller inte.

### 3.3.8 Dämplogiken

Enligt den logik som beskrivits ovan kan en växling i princip ske vid varje anrop av automatvalslogiken, dvs var tiondels sekund. Detta är naturligtvis inte möjligt i verkligheten, själva genomförandet av växlingen tar tid (se kapitel 3.4). Men även om man inte kan växla hur snabbt som helst, upplevs för täta växelbyten som mycket störande för föraren. Det är dessutom så att man har en viss tid under växlingsförloppet då motorn inte levererar något moment till kardanaxeln. Det innebär att ett alltför frekvent växlande sätter stopp för framdrivningen, tiden utan drivning blir helt enkelt för stor.

Det är alltså väl motiverat att införa en dämplogik. Den fungerar på det viset att när en växling har skett sätts en dämptid. En växling kan endast ske vid speciella dämpvarvtal så länge dämptiden inte har överskridits. Först därefter sker växlingar enligt den normala logiken. Olika dämptider sätts beroende på vilket stegval systemet gjort, det vill säga värden finns tabellerade för 3, 2, 1 upp samt 3, 2, 1 ner. Sedan finns de speciella dämpväxelpunkterna i tabellen.

Dämplogikens parametrar justeras inte med Ratio\_Corr. Accelerations- och cruisecontrolmodifierarna påverkar inte dämpväxelpunkterna.

## 3.4 Genomförandet av växlingen

Automatvalslogiken som beskrivits i 3.3 talar alltså om när man ska växla och till vilken växel. Men hur ska detta verkställas? Ett sätt att lösa det är ju att titta på hur växlingen går till i verkligheten och sedan försöka efterlikna det i modellen. Ett annat sätt kan vara att göra det så att det passar modellen optimalt. Här beskrivs hur växlingen i stora drag fungerar i verkligheten.

Växling med OPC görs utan hjälp av kopplingen utom vid start och stopp. Vid start väljer föraren startväxel och kopplar därefter ur för att starta på samma sätt som om man har en manuell låda. Därefter sker växling automatiskt utan koppling. Växlingen sker i en sekvens enligt följande:

1. OPC talar om att det är dags att växla.
2. OPC tar befälet över motorn och styr motorn så att nollmoment uppnås i kardanaxeln.
3. Växelföraren förs till neutralläge. Det krävs att momentet i axeln är litet för att detta ska vara möjligt.
4. Motorvarvtalet anpassas för den nya växeln.
5. Den nya växeln förs i.
6. OPC rampar på momentet motsvarande det ursprungliga gaspådraget.
7. Till sist lämnas befälet över till föraren igen.

Denna sekvens tar i storleksordningen någon sekund att genomföra.

### **3.5 Slutsatser för modellens del**

Målet med modelleringen är att förenkla växellådan så mycket som möjligt utan att den försämrar prestanda i fordonmodellen som helhet. För att uppnå detta måste man hitta en avvägning mellan enkelhet och funktionalitet. Mot en alltför komplex modell står också kraven på en snabb simulering, vilken naturligtvis påverkas negativt av en beräkningstung modell.

Då det gäller mekaniken i växellådan kan man i det enklaste fallet tänka sig enbart en variabel utväxling. Detta skulle dock ge en förlustfri utväxling, vilket inte kan anses acceptabelt. Skälen är uppenbara, eftersom ett av huvudsyftena med modellen är att studera bränsleförbrukning som påverkas av förluster i drivlinan. Ett sätt att lösa detta på är att införa en variabel verkningsgrad, som beror på vilken växel man ligger på. Sådana värden finns uppmätta på verkliga växellådor. Fördelen med den lösningen är att man får bra överensstämmelse i förhållande till den komplexitet beräkningen har.

För att uppnå modellens syften och samtidigt få ett snabbsimulerat system över långa tider är det nödvändigt att modellen väljer samma växel som i verkligheten i så stor utsträckning som möjligt. Därför implementeras all den logik som beskrivits ovan när det gäller automatvalslogiken. Detta "kostar" inte heller så mycket i simuleringsprestanda eftersom logiken kodas direkt i C. Väljer man C fås två andra fördelar också. Dels kan man lätt skriva logiken i sekvens, det vill säga som en algoritm. I Dymola sätts alla ekvationer samman samtidigt, vilket man inte vill då automatvalslogiken ska köras. Dels blir önskemålet att behålla tabellupplägg från riktig OPC kod lättare att uppfylla.

Vidare måste man bestämma sig för hur genomförandet av växlingen går till. Det finns då flera faktorer att ta hänsyn till:

- Växlingen i bil är en sekvens av operationer på drivlinan. Under en viss tid finns ingen drivning, då växeln är i neutralläge. Detta kan ha avgörande betydelse i vissa körfall, till exempel i en kraftig uppförsbacke i låg fart. Det kan då inträffa att växlingen misslyckas och föraren tvingas ingripa.

- Körfall som modellen ska användas i. Tidshorizonten har betydelse för hur snabba förlopp som är intressanta. Risken finns att systemet blir styvt.
- Möjligheter som Dymola erbjuder att lösa detta.

Genomförandet av växlingen i Dymola kommer att ske på enklaste sätt, det vill säga som en variabel utväxling med verkningsgrad. Skälen till det är:

- Simuleringens komplexitet hålls nere, vilket medför att det blir snabbare att simulera.
- Risken för att styvhet uppträder i systemet minskar om man väljer att inte simulera själva växlingsförloppet.
- Modellen ska användas främst för landsvägskörning över längre sträckor. Den påverkan som ett noggrannare växlingsförlopp ger på till exempel bränsleförbrukning är irrelevant. Dels därför att påverkan är så liten över en lång simuleringstid, dels för att andra modulers approximationer riskerar att dominera över växellådans.
- Modellen har en begränsning till hastigheter över 30 km/h. De låga hastigheterna, där ett noggrannare förlopp är viktigast, är därmed uteslutna.

En förutsättning för att detta ska fungera är att motormodellen är gjord så att den kan anpassa varvtalet momentant till den nya växeln. Det innebär att motorn inte får vara trög, i Dymola betyder det att motorns rotationshastighet inte får utgöra ett tillstånd i den kompletta fordonmodellen.

För att i framtiden ha möjlighet att studera växlingsförloppet kommer ett förslag på implementering av en sådan växellojik att presenteras som en alternativlösning i kapitel 5.2.

## 4 STARS – Scania Truck And Road Simulation

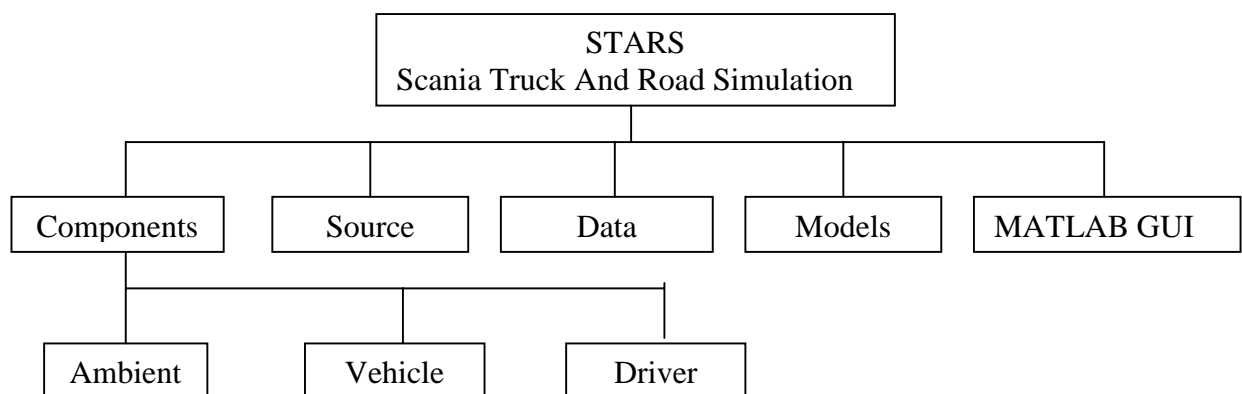
*Examensarbetet är en del i simuleringsystemet och projektet STARS - Scania Truck And Road Simulation. De grundidéer som finns beskrivs och arbetet sätts in i sitt sammanhang.*

STARS är ett simuleringspaket som håller på att byggas upp på Scania för att kunna simulera hela fordonet i olika driftfall med avseende på till exempel bränsleförbrukning, emissioner och så vidare. Det är framförallt i två avseenden som det föreligger ett simuleringsbehov:

- Landsvägssimulering. För att studera bränsleförbrukning, emissioner samt kunna testa körning med nya motortyper.
- Som hjälp vid produktutveckling, till exempel för att pröva styrsystem och regleralgoritmer.

STARS är ett levande projekt och en del i examensarbetet är att utveckla systemet. I kapitel 5 finns beskrivet hur till exempel gränssnitt definieras. Här beskrivs grundstrukturen i systemet.

STARS byggs upp med hjälp av Dymola, MATLAB och C-kod. Figur 4.1 beskriver hur de olika delarna i systemet hänger samman. Components byggs upp i Dymola och är komponenter i fordonet och dess omgivning, till exempel chassi, motor och väg. Components delas in i tre grupper: omgivning, fordon och förare. Dessa komponenter används då man bygger en simuleringsmodell, som återfinns under Models. Models består av flera olika modeller, alla uppbyggda av grunddelar från Components. Source är källkod skriven i C, som olika komponenter använder för att beräkna olika saker, exempelvis motor- och retarderstyrning. Som tidigare beskrivits, kan man från Dymola anropa C-funktioner för beräkning av vissa variabler. I Data återfinns information om olika system i MATLAB format, till exempel mappningar av motorer. Rutan MATLAB GUI är ett användargränssnitt för hela systemet, där användaren kan specificera fordonstyp, last, väg med mera och sedan starta simuleringen samt få resultatet presenterat i olika typer av grafer.



**Figur 4.1.** Schematisk bild av STARS olika beståndsdelar.

Sammanfattat innebär STARS konceptet:

- Ett modulbaserat system, vilket medför flexibilitet.
- Scania komponenter skapas i en biblioteksstruktur.

- Användandet av verklig C-kod från till exempel retarderstyrenhet (RCU = Retarder Control Unit), motorstyrenhet (EDC = Engine Diesel Control) och OptiCruise (OPC).
- Mätdata och användargränssnitt i MATLAB miljö.

Några begränsningar finns på systemet:

- Endast dieselmotorer simuleras. Motorer som går på biobränslen, etanol eller liknande bygger på Ottoprincipen och behandlas inte i STARS.
- Start/stopp förlopp simuleras ej. Lägsta hastighet är 30 km/h. Det innebär inte någon väsentlig begränsning vid landsvägskörning. Körning sker med cruisecontrol.
- Endast longitudinell rörelse simuleras. Det innebär att till exempel kurvor och fjädring inte påverkar förloppet.

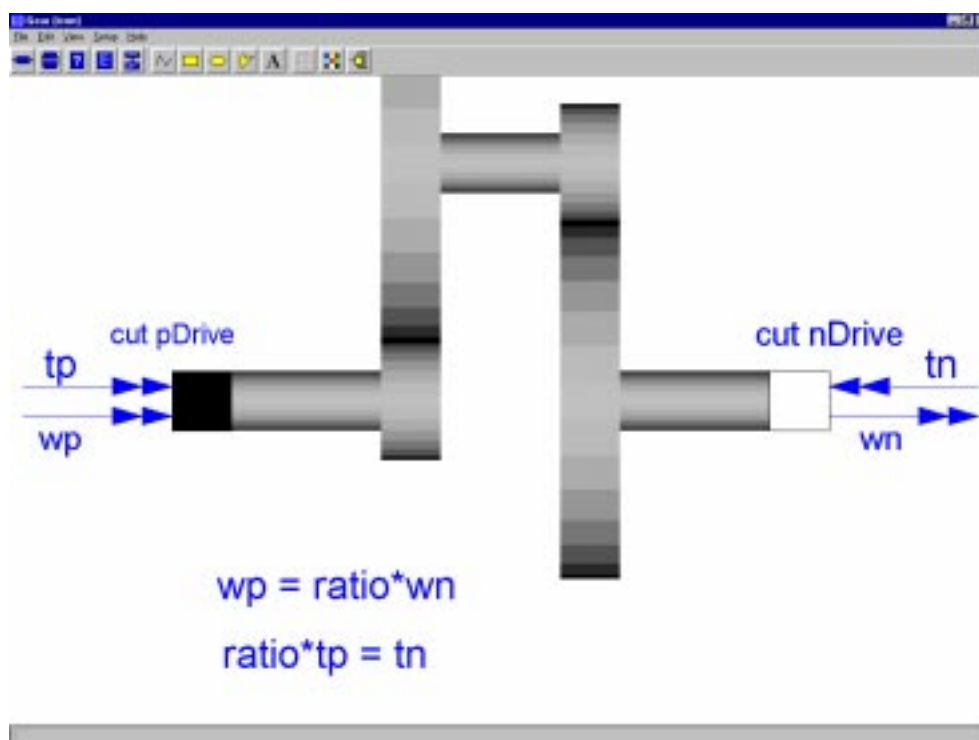


## 5 Design av växellådsmodul

Designen av den del i STARS som examensarbetet svarar för beskrivs i detta kapitel. Modulen designas enligt de slutsatser som dragits i kapitel 3.5, vilket innebär en variabel utväxling med förluster utan växelförlopp. Som alternativ presenteras i kapitlet en lösning på en modell med fullständigt växlingsförlopp.

### 5.1 Struktur

Växellådan ska kopplas in i drivlinan mellan motor och kardanaxel och utgörs i grunden av en enkel utväxling. Den beskrivs i sitt enklaste utförande av två ekvationer, se figur 5.1.



**Figur 5.1** Enklaste sättet att implementera en utväxling.

I figuren är "cut pDrive" och "cut nDrive" anslutningarna till drivlinan. Variabeln ratio är utväxlingen som systemet måste förses med utifrån. Variablerna  $tp$  och  $tn$  är moment medan  $wp$  och  $wn$  är vinkelhastigheter. Växling görs genom att ändra ratio.

Utifrån den modellen (figur 5.1) kan man bygga på med annat. Eftersom C har valts för OPC beräkningarna (kapitel 3.5), behöver ett gränssnitt mellan C-funktionerna och modellen definieras.

#### 5.1.1 Verkningsgrad och utväxlingstal

Verkningsgraden och utväxlingstalen läggs i vektorer i MATLAB. Tillsammans med andra vektorer som andra moduler använder sparas de i en \*.mat fil som heter "dsdata.mat". I Dymola finns några olika funktioner som ger åtkomst till dessa data. I figur 5.2 visas hur det fungerar i fallet utväxlingstal, verkningsgraden är helt analog.

```

>>
>>
>> Ratio
Ratio =
Columns 1 through 7
    11.2700    9.1400    7.1700    5.8100    4.6200    3.7500    3.0100
Columns 8 through 12
    2.4400    1.9100    1.5500    1.2300    1.0000
>> Gear
Gear =
     1     2     3     4     5     6     7     8     9    10    11    12
>> |
    
```

**Figur 5.2** Vektorerna Ratio (utväxlingstalen) och motsvarande växel (Gear) i MATLAB.

I figur 5.2 syns att växel ett har utväxling 11,27 och högsta växeln (12:an) har utväxling 1, den så kallade direktväxeln.

I Dymola görs linjär interpolation mellan dessa vektorer med hjälp av funktionen ExternalTable1. Vektorerna och aktuell växel ges som parametrar och ut fås den interpolerande utväxlingen. I Dymolas ekvationslager skriver man:

```
ratio = ExternalTable1("Gear","Ratio",gear)
```

Så länge ”gear” är ett heltal, vilket det alltid kommer att vara, blir resultatet av interpoleringen exakt den rätta utväxlingen för respektive växel. Det kommer med andra ord att fungera som en ”look-up-table”. Frågan är varför detta behöver ligga i MATLAB? Svaret är att systemet blir mer flexibelt om dessa data kan ändras från MATLAB gränssnittet utan att dymolamodellen behöver kompileras om. På så vis blir det lätt att ändra dessa värden även för den normale användaren som endast kommer i kontakt med MATLAB gränssnittet och inte modellen i Dymola.

### 5.1.2 C-funktioner

C-funktioner används för att beräkna ny växel. C-filen placeras i en speciell katalog som Dymola inkluderar vid kompileringen. Funktionen används i Dymola ekvationslager så här:

```
counter = Counter(tn)
```

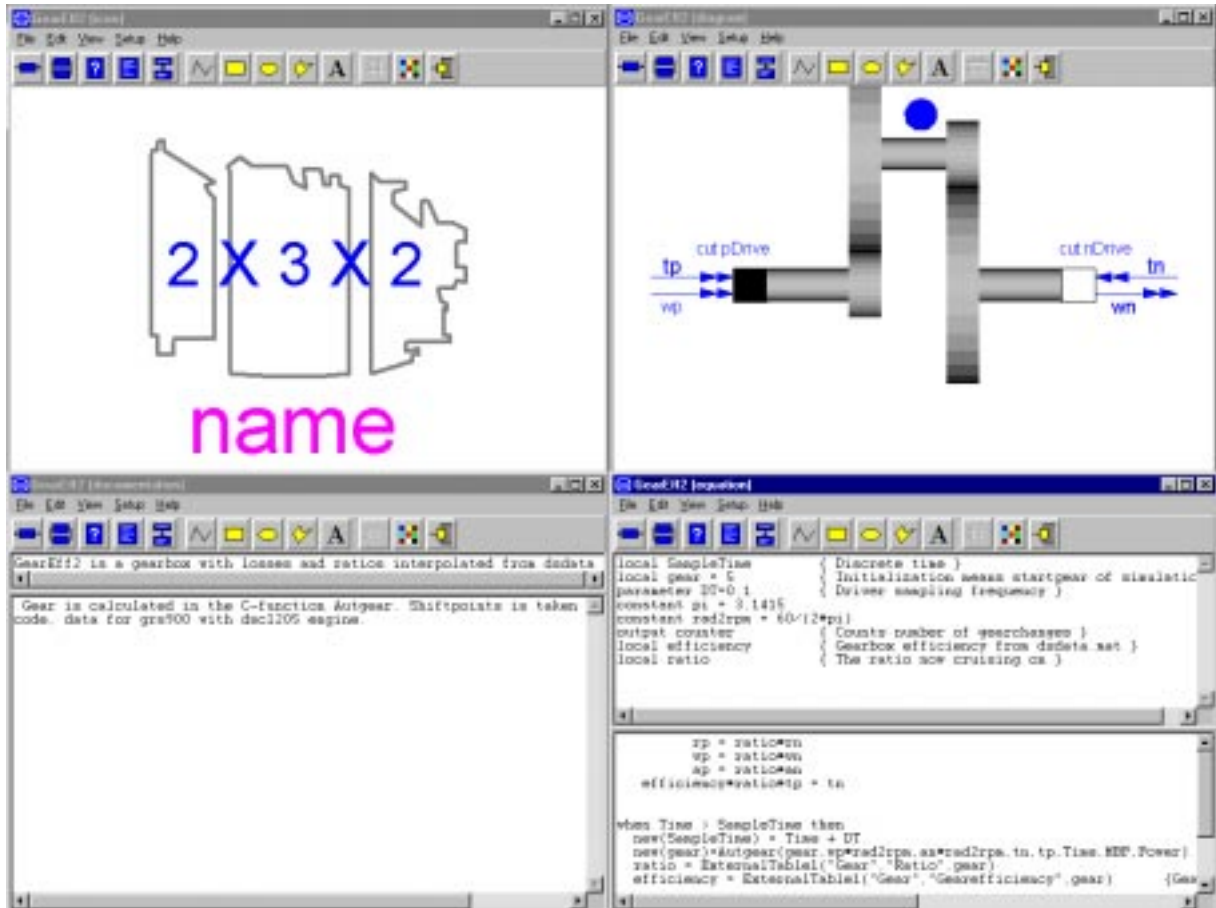
I det här fallet är Counter en funktion som räknar antalet växlingar under en körning. Variabeln counter finns deklarerad i Dymola, liksom invariabeln tn.

Det mesta beräkningsarbetet som görs i C gäller dock automatvalslogiken. Huvudfunktionen för det heter Autgear. Till den finns några mindre stödfunktioner och poster med data. Funktionaliteten i automatvalslogiken finns beskriven i kapitel 3 och det är enligt den beskrivningen implementeringen är gjord. Utöver vad som beskrivs där, finns det en uppsättning postpekare som sätts att peka ut rätt dataposter beroende på vilken motor-växellåds kombination som är aktuell. De variabler som behövs för att åstadkomma detta beskrivs i 5.3.

Lösningen på problemet att växelvalskoden ska vara generell för olika typer av kombinationer av motor och växellåda har lösts med en pekarstruktur. I princip finns tabeller lagrade för olika växellådor och motorer. Sedan läses en parameter, kallad Type, som sätter några pekare att peka på rätt datafält. Finessen är att parametern Type finns i Dymolamodellen och därigenom kan den ändras från MATLAB utan att modellen behöver kompileras om. På så vis kan en användare från GUI (se Figur 4.1) välja växellåda och motor och samtidigt få rätt OPC kod.

### 5.1.3 Modellklassen

Växellådan blir en modellklass i Dymola som sedan kan användas i de modeller man bygger.



**Figur 5.3** Växellådan med ikonlager (övre vänstra), diagramlager (övre högra), dokumentationslager (nedre vänstra) och ekvationslager (nedre högra).

Ikonlagret är så som klassen kommer att synas i de modeller där den används. I diagramlagret syns att det finns tre gränssnitt, ”cuts”, en på var sida om utväxlingen som kopplas till drivlinan samt en signalkoppling högst upp. Den kopplas till koordinatorn.

## 5.2 *Alternativet med växlingsförlopp*

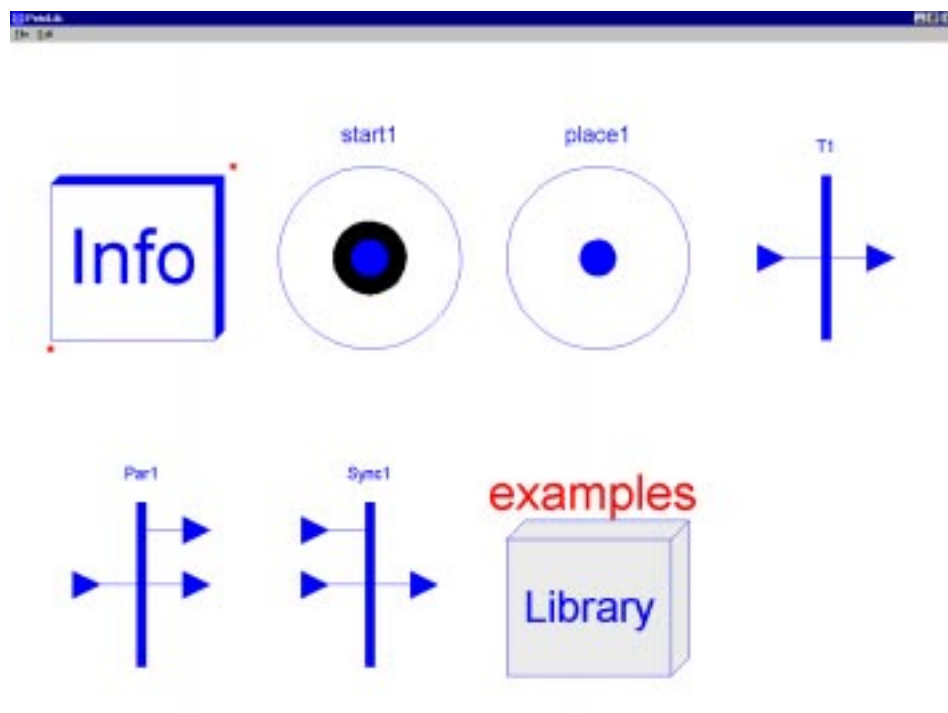
Som tidigare diskuterats kan det finnas skäl att få med de egenskaper som själva växlingsförloppet bidrar med. För de landsvägskörningar som modellen ska användas till har det emellertid liten betydelse. Tidsförlopp på någon sekund ska ställas mot en modell som körs i storlekordningen någon timme. Om man däremot vill titta på snabbare förlopp och studera vad som händer under växlingarna, passar Dymolas modellbibliotek "Finite state machines and Petri nets" bra. Anledningen till det är att växlingen är en sekvens av händelser.

### 5.2.1 *Petrinät*

Systemet som modelleras är i grunden kontinuerligt. Men vid växling sker snabba "hopp" mellan olika tillstånd hos växellåda och drivlina. Det medför att det finns diskreta inslag i växlingsförloppet. Överlag är systemet kontinuerligt, men dess kontinuerliga moder skiljs åt av ett ändligt antal diskontinuerliga modövergångar. När en modövergång inträffar kallas det en händelse. En händelse har ingen utbredning i tiden utan sker momentant. Ett styrengrepp som driver systemet från en mod till en annan kallas handling. En handling har en utsträckning i tiden. För att driva systemet en viss väg genom alla möjliga moder som systemet har krävs en sekvens av handlingar. Därför kallas den här typen av reglering för sekvensstyrning.

I början av 1960-talet definierade den tyske matematikern Carl Adam Petri ett grafiskt modellspråk för att beskriva kopplingen mellan villkor och handlingar. Det kom att kallas Petrinät. Mer om detta finns att läsa i referens 13.

Petrinätet kan sägas vara en generalisering av tillståndsmaskinen då mer än ett tillstånd kan vara aktivt samtidigt. Tillstånden symboliseras i Dymola av cirklarna "start1" och "place1", se figur 5.4. "start1" symbolen är densamma som "place1" symbolen med den skillnaden att vid simuleringsstart är "start1" aktivt initialt. Övergångsvillkoren anges normalt med symbolen "T1". Önskas flera tillstånd aktiva samtidigt kan en parallellförgrening, "par1", användas. Vid återgång till ett aktivt tillstånd måste de invänta varandra och då används "Sync1".



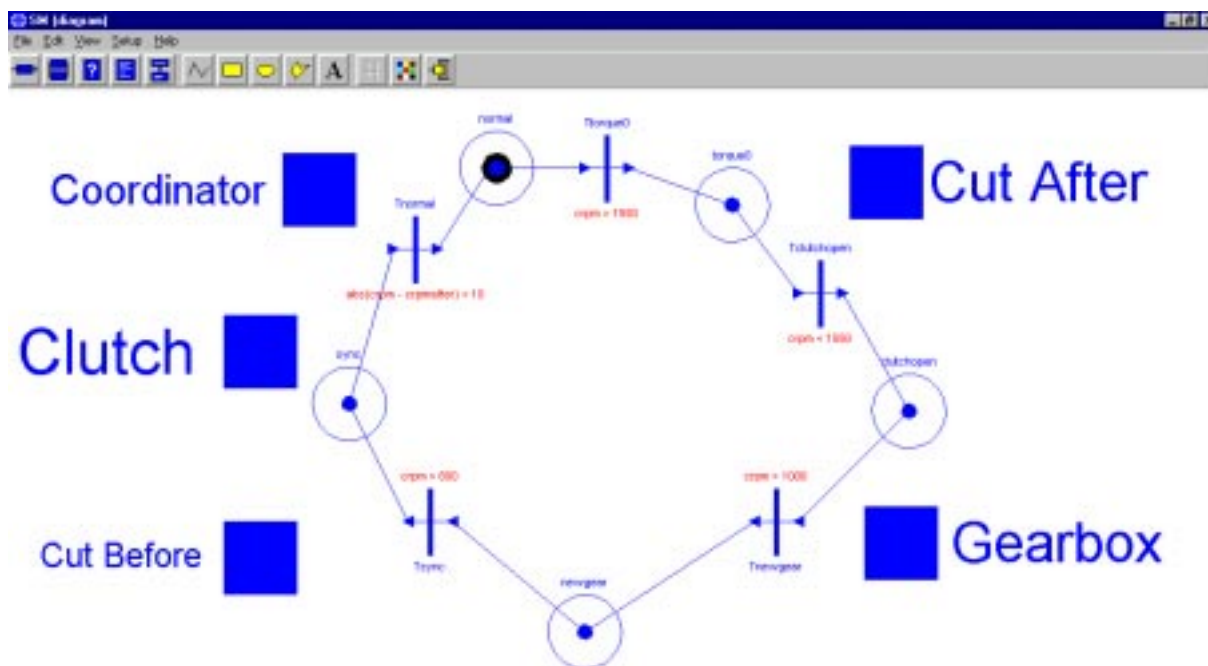
Figur 5.4. Petrinät biblioteket i Dymola.

### 5.2.2 Växlingsförloppet med Petrinät

Växlingen sker, som beskrivits tidigare, genom att ändra utväxlingstalet. Det innebär att neutralläge saknas i den valda modellen av växellåda. Under en viss tid ska drivlinan vara isärkopplad och detta kan man modelleras med en koppling. Följande tillstånd (handlingar) behövs:

- Normal                      Starttillstånd som också är körning mellan växlingarna
- Torque0                    Nedstyrning till nollmoment i drivlinan
- Clutchopen                Öppna kopplingen (dvs lägg i friläge)
- Newgear                    Ändra utväxlingen till den nya
- Sync                         Synkronisera drivlinan så att motor och kardans snurrar lika fort

Slingan är sluten, vilket innebär att när Sync är färdigt går man tillbaka till Normal. Detta kan se ut som i figur 5.5.



**Figur 5.5** Diagramlagret för tillståndsmaskinen som styr växlingsförloppet. Modellklassen heter SM (State Machine). Kvadraterna är cuts (gränssnitt) till omgivande modellklasser.

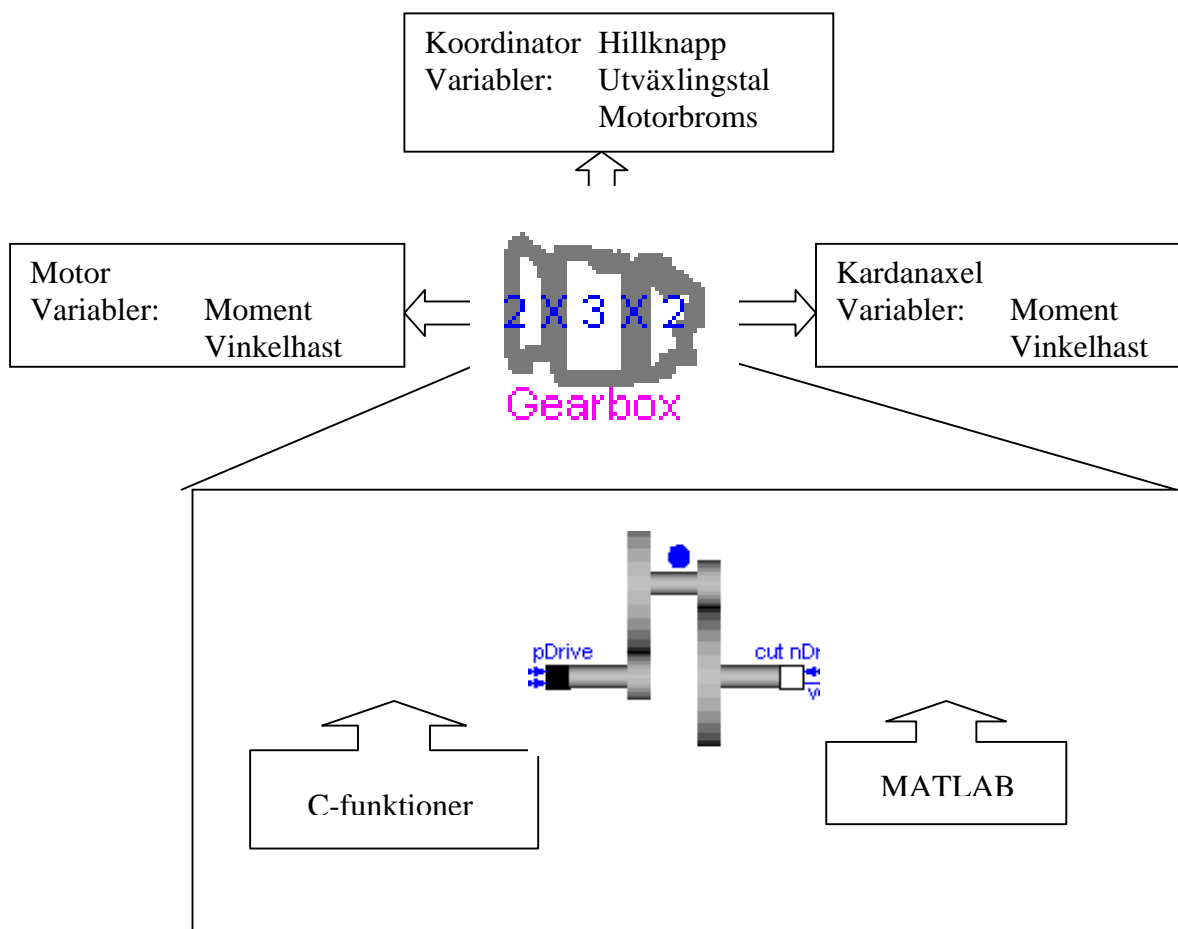
### 5.3 Gränssnitt STARS

För att få kontroll på alla signalvägar mellan de olika modellklasserna, som används i den kompletta fordonsmodellen, definieras en koordinator. För att kunna konstruera koordinatorn måste man veta vilka signaler de olika modulerna behöver.

I växellådans fall behövs följande information:

- Till automatvalslogiken (C-funktionen Autgear)
  - Aktuell växel
  - Motorvarvtal
  - Kardanaxelmoment
  - Kardanaxelacceleration
  - Motormoment
  - Simuleringstid
  - Motorbromssignal
  - Hillknappsignal
  - Motor-växellådskombination
  - Typ av kombination
- Från automatvalslogiken
- Utväxlingstal
- Ny växel

- Verkningsgrad på de olika växlarna



**Figur 5.6** Växellådans signalflöden externt samt uppdelningen internt.

I figur 5.6 visas att växellådsmodulen (Gearbox) i huvudsak består av en utväxling, C-funktioner samt data i MATLAB format. Från koordinatorm kommer information om motorbromsprogram behövs och hillknappens läge. C-funktionerna kodas enligt beskrivningen i avsnitt 3.

#### 5.4 Beräkning av bränsleförbrukningen

Ett av syftena med fordonmodellen är att kunna studera bränsleförbrukningen vid simulering. I motormodulen finns Dymolvariabeln Fueling, som är den bränslemängd (mg/slag) som sprutas in i förbränningsrummet. Genom att integrera över hela simuleringstiden och justera enheterna samt använda densiteten för diesel (840 g/l) kan den totala förbrukningen i liter per mil beräknas. Som exempel visas i kapitel 6.1.4 förbrukningen vid landsvägskörning på ett par olika vägavsnitt. För detaljer i beräkningen av bränsleförbrukning, se bilaga 8.2.



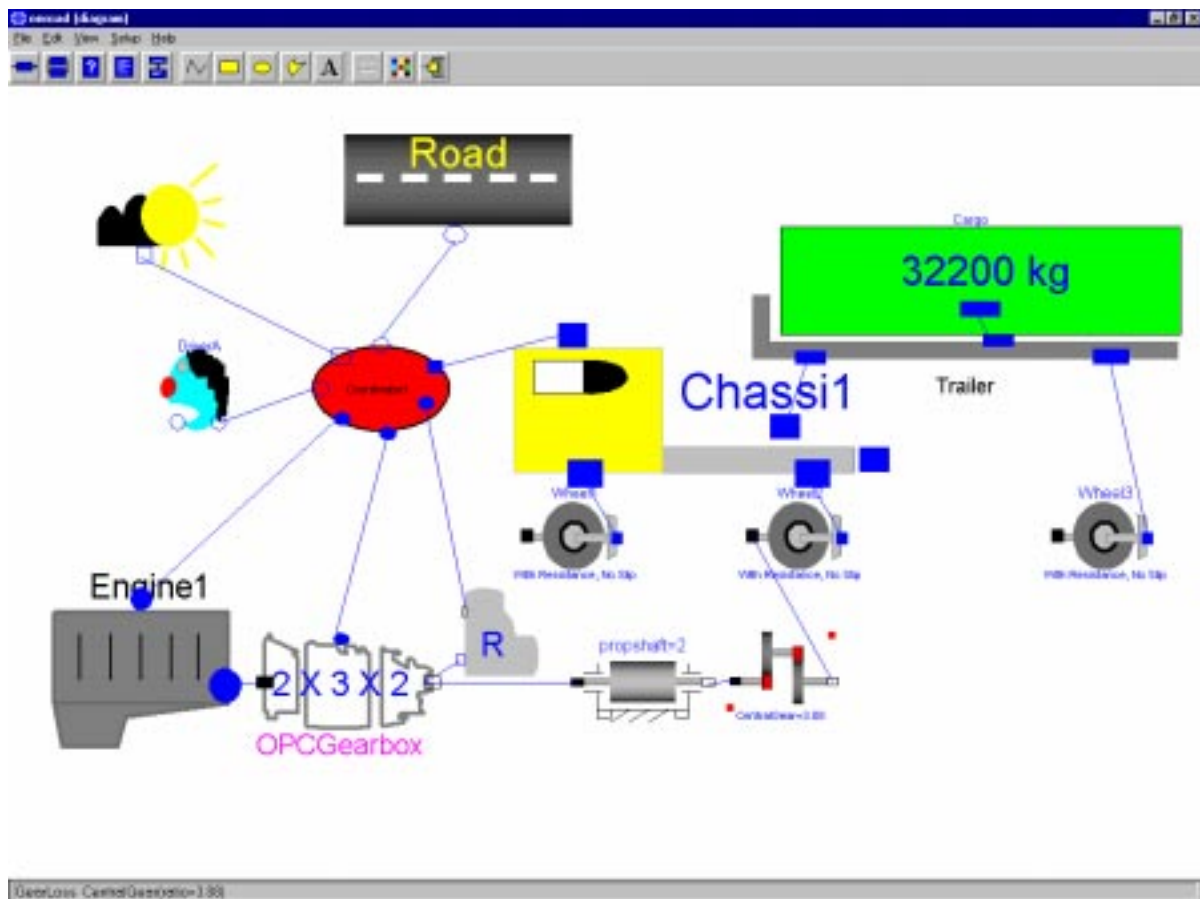
## 6 Resultat och utvidgningar

*Simuleringar av hel fordonsmodell visar att designen fungerar tillsammans med sin omgivning. Detta tillsammans med test av funktionaliteten leder fram till att målen med arbetet är uppfyllda.*

### 6.1 Verifikation av modell

Att verifiera modellen, växellådsmodulen, är inte helt lätt. Eftersom växelvalet i hög grad är ett resultat av hur övriga delar i modellen uppträder, är det mest intressant att pröva växlingarna i den miljön. Samtidigt blir det svårare att lokalisera fel. Verifieringen görs genom att olika simuleringar i typiska driftsfall samt i några speciella fall, till exempel steg i referenshastigheten.

Den fordonsmodell som används i simuleringarna visas i figur 6.1.



**Figur 6.1** Hel fordonsmodell. Växellådsmodulen heter OPCGearbox.

Kordinatorn i mitten agerar signalknutpunkt och runt den finns delsystemen:

- Road                      Vägprofil vars data finns i MATLAB.
- Driver                     Förarmodell, innehåller bland annat hillknapp.
- Engine                    Motor, mappningar finns i MATLAB.
- OPCGearbox              Växellådsmodulen

- Retarder (R)                    Hydromekanisk broms
- Propshaft                        Kardanaxelns tröghetsmoment
- Centralgear                      Bakaxelväxel
- Chassi                             Ekvationer för till exempel luftmotstånd.
- Trailer                            Trailern är valfri, men ger en bra viktfordelning.
- Cargo                             En variabel last. Massan är en parameter.
- Wheel                             Hjulmodell utan slip, men med rullmotstånd.
- Weather                         Väder som anger lufttryck och temperatur

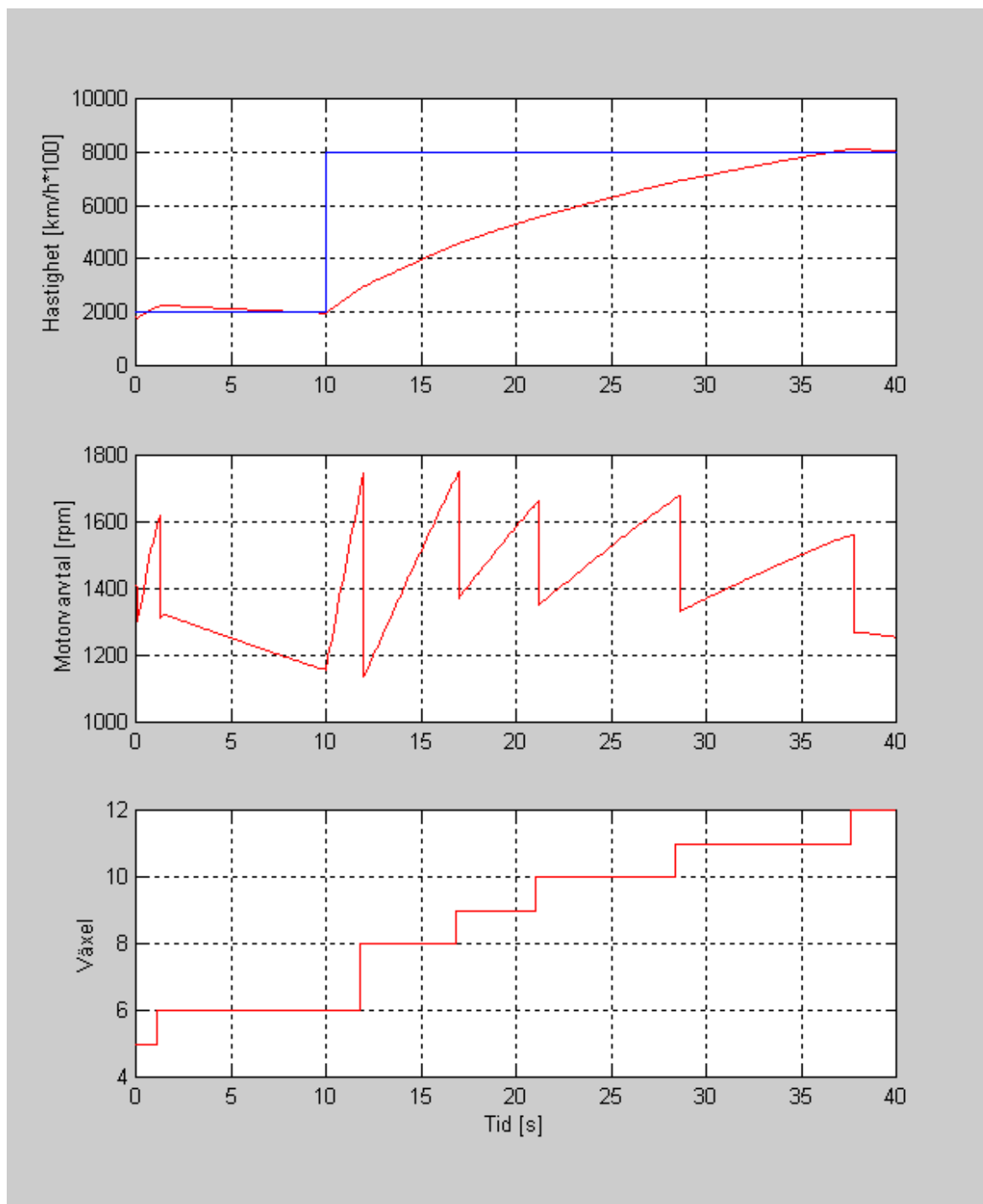
I tabell 1 finns variablerna mellan koordinatorn och de olika modulerna specificerade.

Modul	Variabler in	Variabler ut
Flexible road	Distanceahead	Position
		Referencespeed
		Slope
Driver	Referencespeed	Brake
	Velocity	Schwung
		Retarder
		Distanceahead
		GbxHill
Engine	Velocity	Rpm
	Referencespeed	
	Exhbrake	
	OPCratio	
Gearbox	EBP	OPCratio
	GbxHill	
Retarder	Retarder	Exhbrake
	Rpm	EBP
	Break	
	Referencespeed	
	Schwung	
	Speed	
Chassi	Position	Speed
	Slope	Acceleration

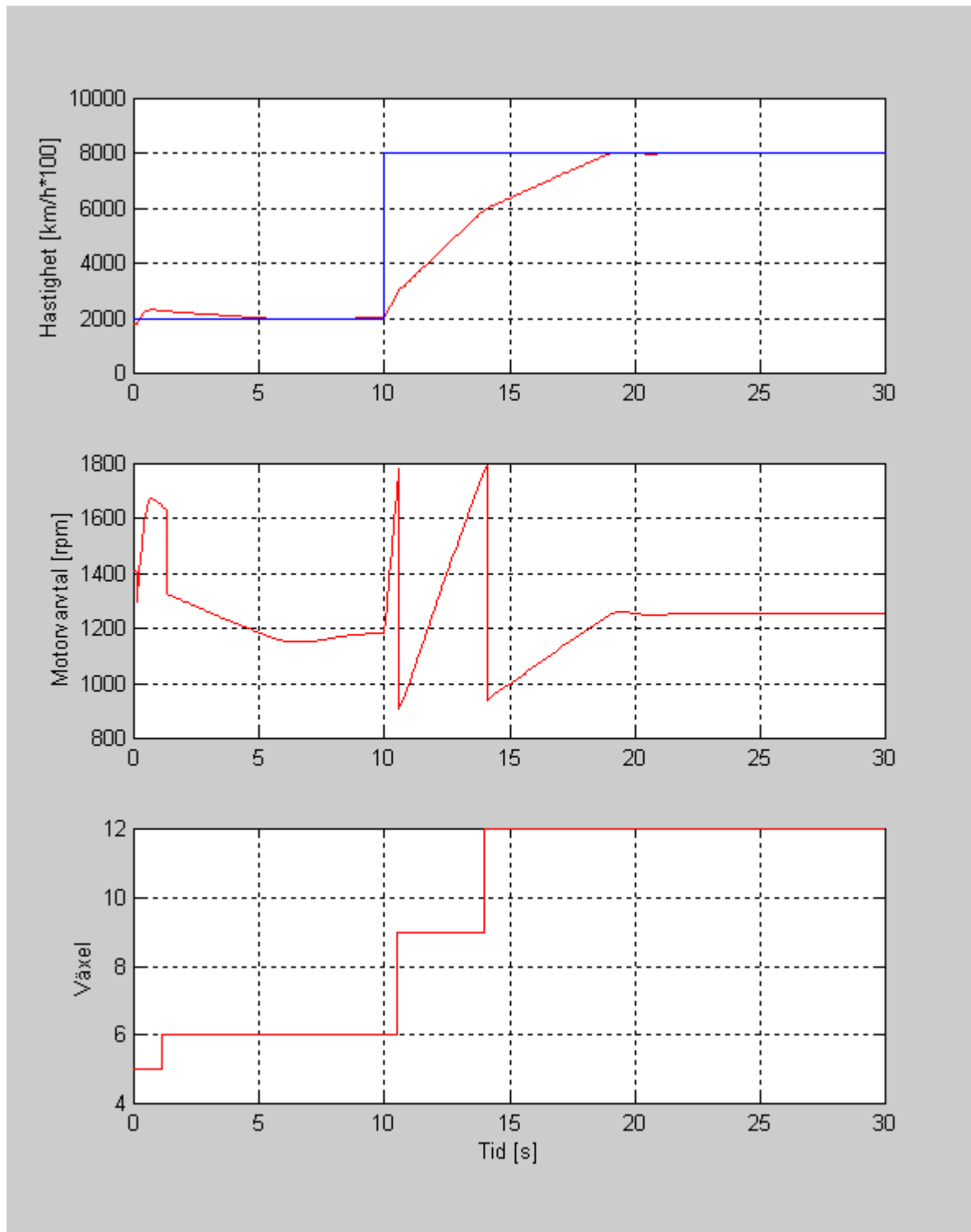
**Tabell 1.** Variablerna till och från modulerna.

### 6.1.1 Steg i referenshastigheten

I den första simuleringen är referenshastigheten ett steg som går från 20 till 80 km/h vid tiden 10 sekunder. Skillnaden visas mellan ett lastat fordon, figur 6.2, och olastat i figur 6.3. Det viktigaste att notera är att stegvalet fungerar. Vid olastat fordon tas så långt möjligt stora växelsteg, i det här fallet 6-9, 9-12. Som jämförelse kommer det lastade fordonet i huvudsak att ta ett steg upp.



**Figur 6.2.** Ett lastat fordon (20 ton) svar på steg i referenshastigheten.



**Figur 6.3.** Ett olastat fordon hoppar över växlare och tar maximalt antal steg (3 upp).

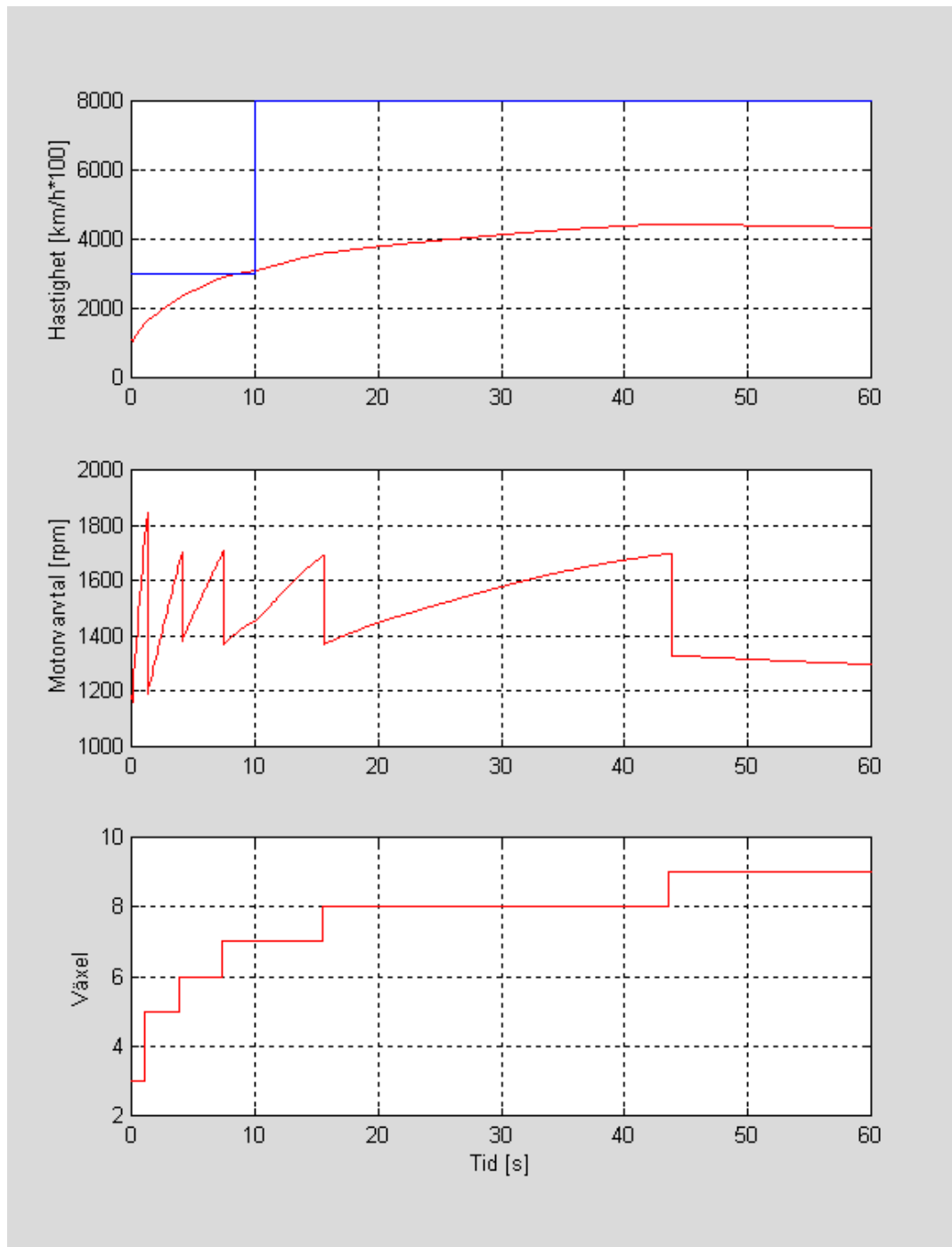
### 6.1.2 Motorbromsprogrammens inverkan

Motorbromsprogrammens funktion är, som tidigare beskrivits, att på begäran hålla uppe motorvarvet. I simulering med ett tungt fordon i utförlöpa aktiveras retardern och begär motorbromsprogram samt använder avgasbromsen. Retarderstyrenheten styr även

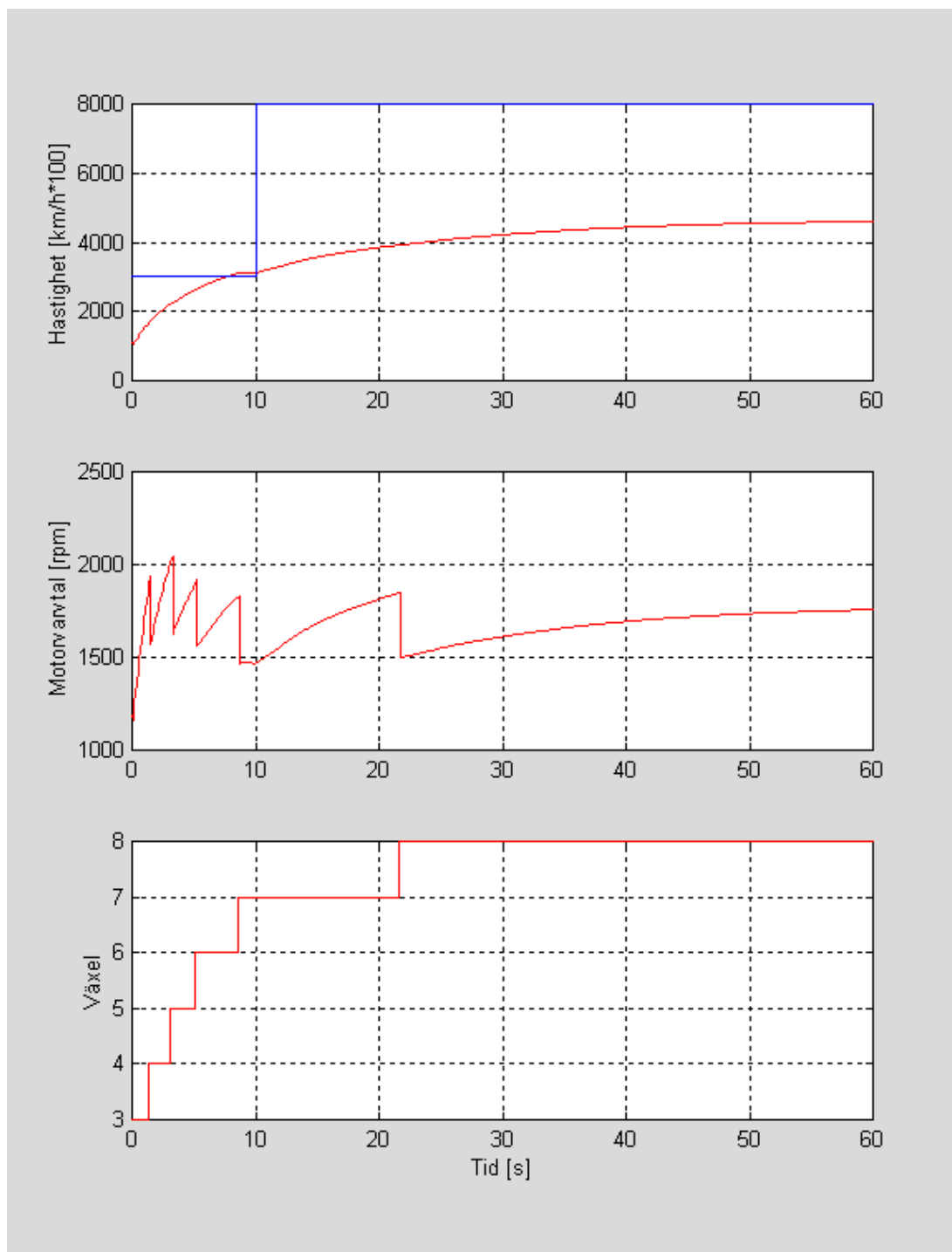
avgasbromsen. Den aktiveras som sista åtgärd när retardern inte har tillräcklig bromskapacitet. Referenshastigheten till retardern följer referenshastigheten till cruisecontrolern plus en konstant offset. Det innebär att fordonet tillåts att rulla fritt i ett hastighetsintervall. Blir hastigheten för låg ger motorn moment och blir den för hög börjar retardern bromsa.

### *6.1.3 Hill- och ekonomiprogrammen*

Hillprogrammet ska användas i branta uppförsbackar. Andra tabeller används och därigenom uppnås bättre backtagningsförmåga. I simuleringarna (figur 6.4 och 6.5) syns att motorn håller ett högre varv med hillprogrammet aktivt, det vill säga växelpunkterna ligger högre upp. Det finns även skillnader i stegvalet, men dessa är svårare att se i figurerna. Notera också att med hillprogrammet blir accelerationen bättre, till exempel syns det i figur 6.5 att motorn regleras ner vid tiden 10 sekunder.



**Figur 6.4.** Acceleration i uppförsläge med konstant lutning 5% och ekonomiprogram.



**Figur 6.5** Samma körning som i figur 6.4, men med hillprogram. Motorns högre varvtal syns tydligt.

#### 6.1.4 Bränsleförbrukning

Som exempel på beräkning av bränsleförbrukningen (se även kapitel 5.4), visas i Tabell 2 och 3 skillnaden i förbrukning mellan hill- och ekonomiprogram samt med olika last på fordonet. Vägen är E4 söderut från Södertälje i tabell 2. I tabell 3 är vägen Oslo – Haugsund, vilket är en betydligt tuffare väg. Den har större variation i både lutning och hastighet. En simuleringstid på 1000 sekunder ger att sträckan är ungefär 22 km i första fallet och 20 km i det andra. Som väntat har lasten och vägens beskaffenhet betydelse, men skillnaden mellan växelprogrammen är liten.

<b>Bruttovikt (ton)</b>	<b>Förbrukning (L/mil) hill</b>	<b>Förbrukning (L/mil) ekonomi</b>	<b>Sträcka (km)</b>
14	1.97	1.97	23
46	3.55	3.52	22

**Tabell 2.** Jämförelse av bränsleförbrukning på E4 från Södertälje.

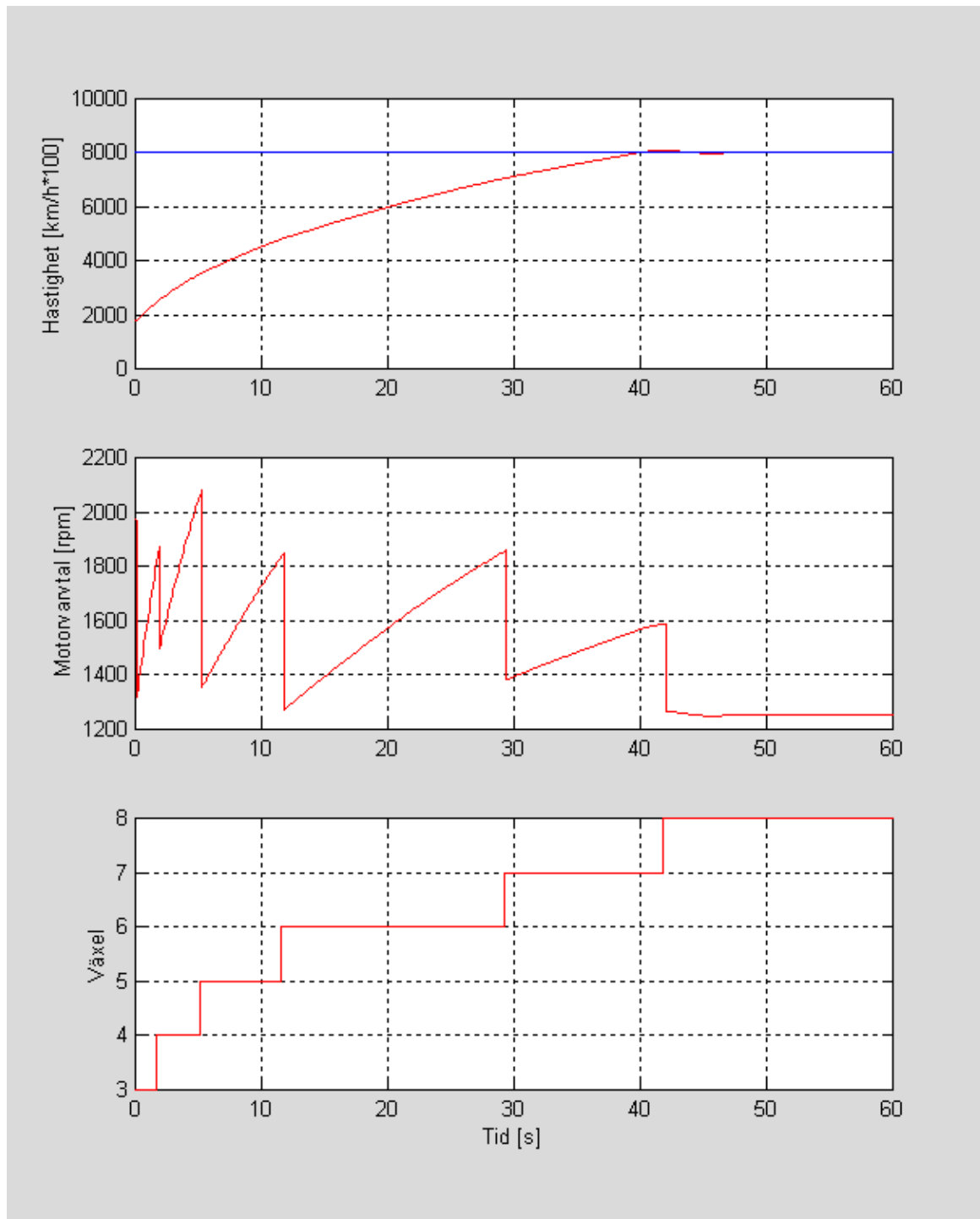
<b>Bruttovikt (ton)</b>	<b>Förbrukning (L/mil) hill</b>	<b>Förbrukning (L/mil) ekonomi</b>	<b>Sträcka (km)</b>
14	2.76	2.76	23
46	6.89	6.90	20

**Tabell 3.** Bränsleförbrukning på vägen mellan Oslo och Haugsund.



### 6.1.5 *Byte av växellåda*

För att verifiera att valet av växellåda fungerar, görs körningar med gr801. Det är en 8 växlad låda utan split. Följden av att ha färre växlar blir att varje växel måste duga i fler situationer. Detta visas i figur 6.6 genom att varvtalsomfånget ökar, det vill säga växling sker vid ett högre varvtal jämfört med en 14 växlad låda. Likaså kommer varvtalet efter växling att vara lägre. Det innebär att motorn måste vara stark nog i ett större varvtalsregister, alternativt att fordonet är lättare.



**Figur 6.6.** Steg i referenshastigheten med 8 växlad låda. Varvtalsomfånget ökar på varje växel.

### **6.2 Reflektion och utvidgningar**

Fordonsmodellen har nu utrustats med en fungerande växellådsmodul. Automatvalslogiken är implementerad enligt de önskemål som finns i avsnitt 1.3. Vidare har ett gränssnitt definierats och implementerats genom en koordinator. Slutligen har verifieringssimuleringar gjorts. Detta gör att jag kan dra slutsatsen att målen med examensarbetet är uppfyllda.

För att den alternativa växlingen med förlopp ska fungera bra behövs ytterligare arbete, så att simuleringar kan göras med helbilmodellen. Den kan då användas för att pröva regleralgoritmer vid motorstyrning under själva växlingen. Det mest realistiska torde vara att ha två olika modeller, en för kortare tidsförlopp med växlingsförlopp och en enligt den princip som beskrivs i detta arbete. Giltighetsområdet för modellerna får då avgränsas till de tidsdomäner som växlingen fungerar bra vid.

Att utläsa statistik över växelvalet, vad gäller tid och sträcka på de olika växlar, hur stort moment de har belastas med och annat är intressant att fortsätta arbeta med.

Bromsstrategi för hela fordonmodellen är ett annat spår som kan utredas mer. Samverkan mellan retarder, avgasbroms och färdbröms är komplicerat. Hur ska det lösas på bästa sätt i modellen? Det är en fråga som inte faller direkt in i växellådsmodulen, men det påverkar i hög grad växelvalet. Jag föreslår att version 4.0 av Dymola används till ovanstående arbete, dels för att konverteringsproblemen minskar, dels för att version 4.0 innehåller förbättringar som underlättar modellbygget.

## 7 Referenser

1. H. Elmqvist, D. Bruck, M. Otter: *DYMOLA – Dynamic Modeling Laboratory user's manual*, ver 3.0, Dynasim AB
2. *Additional Documentation for Dymola 3.1*, Dynasim AB 1998
3. P. Beater: *onedoflib - Library of Mechanical Components for use with Dymola*, ver1.0, Dynasim AB
4. K. Mårdsjö: *Informera om teknik*, Liber förlag, 1986
5. L. Nielsen, L. Eriksson: *Course material vehicular systems*, ISY-LiTH, 1998
6. K. Sigmon: *MATLAB Primer*, Third edition, Dept. of mathematics, University of Florida, 1993.
7. A. Kjell, P. Asplund: *Modellering och Simulering av Hydraulisk Retarder*, examensarbete KTH.
8. K. Gustafsson: *Traps and Pitfalls in Simulation*, Ericsson Mobile Communications AB
9. B. W. Kernighan, D. M. Ritchie: *The C programming language*, Computer press 1989.
10. A. Eriksson: *Simulation based methods and tools for comparison of powertrain concepts*, report no 1997-06-06, CTH.
11. S. Berglund m fl: *027 A powertrain control module for a heavy duty truck with VGT diesel engine and fixed ratio gearbox*, Volvo truck corporation.
12. T. Glad, L. Ljung: *Modellbygge och Simulering*, Studentlitteratur, 1991
13. P. Lindskog, J.E. Strömberg: *Styrning av system med diskreta inslag*, ver 1.1 april 1997, ISY-LiTH

### Programvara:

1. Dymola – Dynamic Modeling Laboratory, version 3.1b, 1998-10-20. Dynasim AB.
2. Microsoft Developer Studio 97, Visual C++ 5.0, Microsoft Corporation.
3. MATLAB, version 5.3.0.10183, The Mathworks Inc.

Alla program är körda på en PC med operativsystem Microsoft Windows NT 4.0

## 8 Bilagor

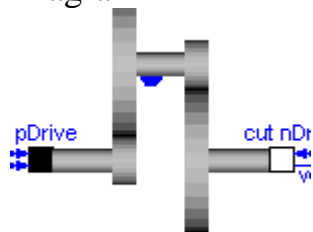
### 8.1 OPCGearbox

Description

OPCGearbox is a gearbox with losses and ratios interpolated from dsdata



Diagram



Information

Gear is calculated in the C-function Autgear. Shiftpoints is taken from OptiCruise code.

Parameters

Name	Default	Description
DT	0.1	Driver sampling frequency
Type	1	The type of gearbox

Dymola definition

```

model class (GearboxBase) OPCGearbox
local SampleTime      { Discrete time }
local gear = 5        { Initialization means startgear of simulation }
parameter DT=0.1      { Driver sampling frequency }
constant pi = 3.1416
constant rad2rpm = 60/(2*pi)
output counter        { Counts number of gearchanges }
local efficiency      { Gearbox efficiency from dsdata.mat }
local ratio           { The ratio now cruising on }
parameter Type = 1   { The type of gearbox }
    rp = ratio*rn
    wp = ratio*wn
    ap = ratio*an
    efficiency*ratio*tp = tn

when Time > SampleTime then
    new(SampleTime) = Time + DT
    new(gear)=Autgear(gear,wp*rad2rpm,an*rad2rpm,tn,tp,Time,EBP,GbxHill,Type)
    
```

```
{ Calls C-function, returns gear no according to OptiCruise }
ratio = ExternalTable1("GearNo","Ratio",gear)
efficiency = ExternalTable1("GearNo","Gearefficiency",gear)
{ Gear no to interpolation }
OPCratio=1000/ratio

{ Gearratio from interpolation to coordinator }
endwhen

counter = Counter(tn)

{ Counts the no of gearchanges }
end
```

---

GearboxBase  
Diagram



Dymola definition

```
model class (DriveTwoCut) GearboxBase
  cut C2 (OPCratio,EBP,GbxHill)
end
```

---

## 8.2 *Beräkning av bränsleförbrukning*

```
% Calculation of fuelconsumption
% File: fuelconsump.m

%1. Define fueldensity [kg/m3=g/dm3=g/L]
Fueldensity = 0.84e3;

%2. Read data calculated by Dymola
[s,n] = tload;

%3. Get trajectory index for distance [m]
Posindex = tnindex(n,'Chassil.Position2');

%4. Get trajectory index for fuelflow [g/h]
Fuelindex=tnindex(n,'Engine1::Combustion1.fuelflow');

%5. Integrate over simulation time
Fuel=1/3600*tinteg(s,Fuelindex);

%6. Final fuel value [g]
Finalfuel=max(Fuel(:,2));

%7. How far has the vehicle driven? [km]
Distance=(max(s(:,Posindex))-s(1,Posindex))/1000

%8. Calculate the consumption [L/10km]
Fuelconsumption=Finalfuel/Fueldensity/(Distance/10)
```