

SSYX02-12-39

Modelica-modell av en dieselmotors luftsystem

Kandidatarbete för Maskinteknik och Teknisk fysik

KARIN UHLÉN
AXEL JONSON
ANTON LUNDIN

Institutionen för Signaler och system
Avdelningen för Signalbehandling
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige. 15 april 2012

Förord

Inledningsvis vill vi tacka tre personer vars hjälp varit oumbärlig i detta projekt. Tack Markus Grahn, doktorand på institutionen för Tillämpad mekanik på Chalmers, för all hjälp med datahantering, teori i luftsystemmodellering och vårt oändliga tjat. Tack Jonas Fredriksson, docent på institutionen för Signaler och system på Chalmers, för din hjälp med Modelica och Dymola. Slutligen vill vi tacka vår handledare professor Tomas McKelvey, även han på Signaler och system, för vägledningen genom projektets gång. Tack också Volvo AB och Volvo personvagnar, för upplåtandet av motordata. Utan er hjälp hade vi inte rott projektet i hamn!

Sammanfattning

Projektet gick ut på att utvärdera det objektorienterade modelleringsspråket Modelica, tillsammans med simuleringsverktyget Dymola, som ett verktyg vid motormodellering. En modell byggdes upp av en dieselmotors luftsystem, innehållande bland annat EGR (avgasåterföring) och VGT (variabel turbinarea). Utifrån denna modell undersöktes användarvänligheten hos programvaran.

Modellen som skapades jämfördes med mätningar från både en Volvo personbilmotor och en Volvo lastbilmotor. Verifikationen visade att modellen följer dynamiken från mätningarna väl och har måttliga avvikelser.

Modelicaspråket har en enkel och logisk uppbyggnad som underlättar kodningen för användaren. Det går relativt snabbt att skapa en modell av en komponent och dessutom behöver inte ekvationer skrivas i en specifik ordning, något som kan ses som en av de största fördelarna med Modelica. Det finns dock en del begränsningar. Till exempel är felkoder som kan dyka upp vid en simulering ofta väldigt svårtydda, vilket leder till tidskrävande felsökning.

Enkelheten och de många fördelarna med Modelica/Dymola skapar goda förutsättningar för framtida användning av programvaran. Efter de erfarenheter som projektgruppen förvärvat under kandidatarbetets gång kan Modelica/Dymola starkt rekommenderas för fortsatt motormodellering.

Abstract

The task of this project was to evaluate the object-oriented modeling language Modelica, together with the simulation software Dymola, as a tool for engine modeling. A model over the air system of a diesel engine was built up, containing EGR (Exhaust Gas Recirculation) and VGT (Variable Geometry Turbine). From this model the usability of the software was then examined.

The simulation of the model was compared to measurements from both a light and a heavy Volvo diesel engine. The verification showed that the model follows the dynamics of the measurements well and have moderate deviations.

The Modelica language has a simple and logical structure that facilitates the coding for the user. Creating a model of a component can be done relatively quick. Also, there is no need to write equations in a specific order, which can be seen as one of the biggest advantages of Modelica. However, there are some limitations. For example, error codes that can appear in the simulation are often very difficult to interpret, resulting in time-consuming troubleshooting.

The simplicity and the many benefits of Modelica/Dymola provides good conditions for future use of the software. From the experiences achieved during the bachelor project, further use of Modelica/Dymola as a tool for engine modeling can be strongly recommended.

Innehåll

1	Inledning	1
1.1	Förutsättningar	1
1.1.1	Komponentbeskrivning	1
1.1.2	Icke-minfasbeteenden	3
1.2	Frågeställningar	3
1.3	Avgränsningar	5
2	Modelica och Dymola	6
3	Metod	9
3.1	Grundläggande uppbyggnad	9
3.2	Verifiering	10
3.3	Källor	11
4	Modell	13
4.1	Strypventil och EGR-ventil	14
4.2	Volym, insugs- och avgasmanifold	15
4.2.1	Utökad modell	15
4.3	Cylindrar	16
4.4	Turbo	17
4.4.1	Turbinaxel	17
4.4.2	Turbin med variabel geometri (VGT)	17
4.4.3	Kompressor	18
4.5	Kylare	19
5	Verifiering och utvärdering	20
5.1	Icke-minfasbeteenden	20
5.2	Volvo personbilmotor	21
5.3	Volvo lastbilmotor	25
5.4	Stabilitet i simuleringen	29
6	Analys	30
6.1	Motormodell	30
6.1.1	Icke-minfasbeteenden	30
6.1.2	Volvo personbilmotor	30
6.1.3	Volvo lastbilmotor	31
6.1.4	Potentiella felkällor	31
6.2	Stabilitet i simuleringen	33
6.3	Modelica och Dymola	33
7	Slutsats	36
	Referenser	38
A	Stökiometrisk förbränning	39
B	Motor- och körningsdata	42
C	Modelica-kod för dieselmotormodellen	50

1 Inledning

Förbränningsmotorers emissionskrav ökar stadigt och kunder ställer idag höga krav på såväl bilars miljöegenskaper som deras pris och körbarhet. Fordonsindustrins traditionella metoder för att analysera och optimera styrstrategier är dyra och tidskrävande. Att förstå luftsystemets dynamik är av största relevans för att kunna prediktera och styra motorns funktion och egenskaper i avseende på emissioner, bränsleförbrukning och körbarhet.

Syftet med projektet är att med datorns hjälp modellera en dieselmotors luftsystem. Datormodeller är ett ypperligt verktyg för att tillfredsställa de uppställda krav, då de ger snabba svar på vad konsekvensen av olika styrstrategier blir. Modellering av motorer och luftsystem tillåter den flexibilitet vid mätningar av dynamiska förlopp som man med ökade krav tvingas eftersträva. Därför förflyttas allt mer av utvecklingsarbetet till datormodeller.

Projektet kan relatera till många olika områden i samhället och industrin idag. För miljödebatten är det i högsta grad relevant att med hjälp av simuleringar kunna utveckla förfinade styrstrategier som minskar miljöpåverkan och medför förbättrade egenskaper. Även inom många andra områden i industrin finns ett växande behov av de tjänster som simuleringar kan erbjuda idag, då de möjliggör ett snabbare, billigare och ofta bättre utvecklingsarbete.

1.1 Förutsättningar

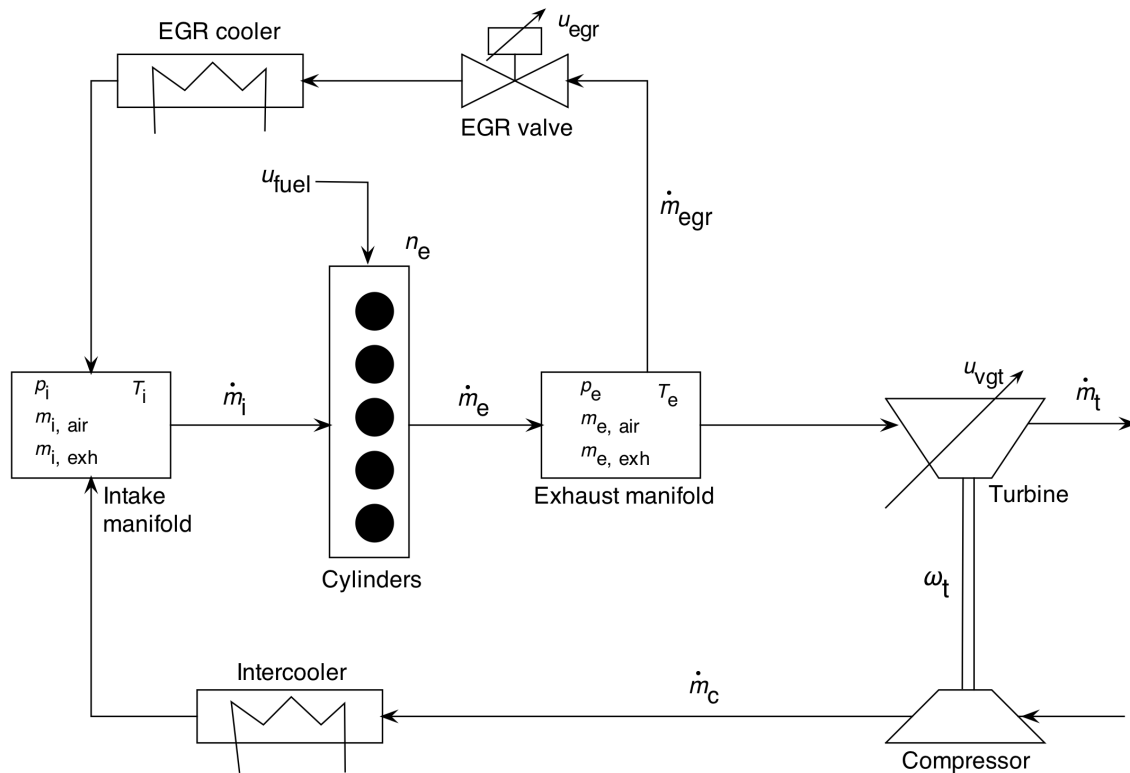
För att modellera dieselmotorns luftsystem ska det objektorienterade modelleringsspråket Modelica och simuleringsverktyget Dymola användas. Nackdelen med många för tillfället vanligt använda modellerings- och simuleringsverktyg är att det kan vara svårt att särskilja delsystem av olika fysikaliska domäner, såsom mekaniska, elektriska och så vidare. Dessutom gör avsaknaden av deklarativ funktionshantering¹ att det kan bli problematiskt att gå in och ändra på någon del i en modell. Förhoppningarna är att Modelica och Dymola ska lösa dessa problem.

1.1.1 Komponentbeskrivning

För att få en grundläggande inblick över hur en dieselmotor fungerar beskrivs de viktigaste beståndsdelarna och deras uppgifter i följande delkapitel.

Funktionen för en allmän överladdad dieselmotor kan betraktas i figur 1. Med överladdning av en förbränningsmotor menas att man skapar ett övertryck i insugsmanifolden, oftast med hjälp av en turbo (turbין och kompressor). I figuren symboliserar \dot{m} massflöden, p och T betecknar tryck respektive temperatur och u är styrsignaler. Turbinens vinkelhastighet betecknas ω_t och motorns varvtal n_e .

¹I praktiken innebär detta att man inte behöver ange ekvationer i en specifik ordning.



Figur 1: Dieselmotorns allmänna funktion.

Cylindrar

En dieselmotor har ett givet antal cylindrar i vilka förbränningen av bränslet äger rum. Tillförsel av bränsle sker genom direktinsprutning med hjälp av munstycken i anslutning till varje cylinder. En hög kompression av luften gör att bränslet självantänder utan tändstift.

Insugsmanifold (intake manifold)

Insugsmanifolden är en volym vars uppgift är att fördela luft till cylindrarna. Den har tre huvudsakliga anslutningar; två ingångar, från kompressorn och EGR-ventilen, samt en utgång mot cylindrarna.

Avgasmanifold (exhaust manifold)

På samma sätt som insugsmanifolden består avgasmanifolden av en volym med tre anslutningar. In kommer avgaser från cylindrarna, ut går avgaser i första hand till turbinen men även till EGR-ventilen.

EGR-ventil (EGR valve)

EGR-ventilen är en strypning med justerbar area, vars uppgift är att återleda en liten del av avgaserna från avgasmanifolden till insugsmanifolden. Avgaserna fungerar sedan som en inert gas i förbränningen, vilket sänker topptemperaturen. Detta förhindrar uppkomsten av kväveoxider, något som annars bildas vid höga temperaturer då kväve spontanreagerar med syre.

Turbin med variabel geometri (VGT)

Efter avgasmanifolden sitter turbinen. När avgaserna flödar igenom turbinen drivs turbinaxeln som i sin tur driver kompressorn. Variabel turbinarea (VGT) gör det möjligt att justera verkningsgraden på turbinen, vilket gör att laddtrycket i insugsmanifolden kan varieras beroende på arbetspunkt.

Kompressor

Kompressorn fungerar som en pump, driven av turbinaxeln, vars uppgift är att genom en tryckhöjning öka densiteten på insugsluften för att kunna få ut en större motoreffekt. En konsekvens av tryckökningen är att temperaturen ökar efter kompressorn.

Kylare (Intercooler, EGR cooler)

En kylare används för att sänka temperaturen på dels insugsluften efter kompressorn, och dels avgaserna efter EGR-ventilen. Omgivande luft fungerar som kylande medium. Gaserna kyls för att öka deras densitet och därmed möjliggöra högre effektuttag respektive högre inblandning av avgaser i insugsluften.

1.1.2 Icke-minfasbeteenden

En motor med EGR och VGT uppvisar ett så kallat icke-minfasbeteende [1]. Det innebär reglertekniskt att systemets överföringsfunktion har ett eller flera nollställen i det högra halvplanet [2]. En konsekvens av detta är att systemet vid ett steg i insignal för EGR- och/eller VGT-area får en "undersläng" i de utsignaler där beteendet återfinns. Den fysikaliska förklaringen till dessa beteenden är att systemet består av två dynamiska effekter: en snabb tryckdynamik i manifoldarna och en långsam turbodynamik. Dessa två effekter arbetar ofta emot varandra och förklarar icke-minfasbeteendet. Samtidigt innebär icke-minfasbeteendet en reglerteknisk utmaning vid utvecklingen av motorstyrssystem.

1.2 Frågeställningar

Huvudmålen med projektet kan sammanfattas i två punkter:

- Skapa en modell i Modelica/Dymola över en dieselmotors luftsystem, som följer uppmätta värden från en riktig motor så bra som möjligt
- Utvärdera användarvänligheten i Modelica/Dymola med avseende på möjligheten att skapa, simulera och bygga vidare på en motormodell

En första uppgift som är relevant att undersöka är om modellen uppvisar icke-minfasbeteende, då den verkliga motorn har sådana egenskaper och det är reglertekniskt intressant.

Modellen ska också jämföras mot en körning från en Volvo personbilsdiesel på 2.4 liter samt en lastbilmotor på 12.8 liter. Båda dessa motorer har EGR och VGT. Denna jämförelse görs för att kunna utvärdera såväl modellens tillförlitlighet som smidigheten i Modelica/Dymola att modifiera modellen för olika motortyper.

Simuleringarna för de båda motorerna ska också jämföras mot samma körning i en utökad modell. I den utökade modellen ska temperaturerna modelleras i samtliga volymer med mass- och energikonservering, till skillnad från grundmodellen som kommer att innefatta masskonservering med en enkel temperaturmodell.

I fallet med personbilsmotorn finns en redan befintlig modell implementerad i Matlab Simulink². Denna kommer inte att vara identisk med projektets modell, men båda ska ha de viktigaste ekvationerna gemensamt. Meningen är att modellen i Simulink ska användas som en jämförelse mot projektets modell för att verifiera att de får väsentligt lika beteenden.

För att verifiera tillförlitligheten i simuleringsresultaten ska stabiliteten i lösaren utvärderas. Detta för att vara säker på att mindre ändringar i lösningsmetoden inte gör stora utslag på resultatet.

Efter att dieselmotorns luftsystem är färdigmodellerat kan slutligen en subjektiv bedömning av Modelica/Dymola göras utifrån de erfarenheter som projektgruppen erhållit.

Huvudmålen ovan kan delas upp i nedanstående frågeställningar.

Motormodell:

- Uppvisar modellen icke-minfasbeteende vid ett steg i EGR- och/eller VGT-area?
- Hur väl följer modellen dynamiken hos de uppmätta värdena från den verkliga personbils-/lastbilsmotorn? Hur stora blir avvikelserna?
- Ger den utökade modellen mindre avvikelser än grundmodellen för personbils-/lastbilsmotorn?
- Hur väl följer Simulink-modellen de uppmätta värdena från personbilsmotorn, jämfört med modellen i Dymola?

Modelicaspråket och Dymola:

- Hur logisk är koduppbyggnaden? Hur enkelt är det att bygga en komponent?
- Hur är arbetsmiljön för modelleringen respektive simuleringen?
- Vilka fördelar ses med objektorienteringen?
- Vilka fördelar ses med den deklarativa funktionshanteringen?
- Vad blir simuleringstiden?
- Är ekvationslösaren pålitlig?
- Hur enkelt är felsökning?
- Hur hanterar man data från simuleringen?
- Kompatibilitet ihop med andra programvaror?
- Hur är befintlig litteratur om modellering i Modelica/Dymola?
- Lämpar sig Modelica och Dymola till motormodellering?

²Kommersiellt verktyg utvecklat av MathWorks för modellering, simulering och analys av dynamiska system. [3]

1.3 Avgränsningar

I en förbränningsmotor är det i princip två saker som måste beskrivas av en modell som är relevant för styrsystemet. Det är dels luftsystemets dynamik och dels förbränningen i cylindrarna (till exempel andel av olika emissioner). Projektet har inriktningen att endast studera luftsystemets dynamik, då omfånget skulle bli för omfattande om förbränningen i cylindrarna också togs hänsyn till. Dock används en medelflödesmodell över cylindrarna, men denna inkluderar inte emissioner, såsom andel sot, kväveoxider, kolväten med mera. Denna medelflödesmodell tar inte heller hänsyn till om det blir bränsleöverskott under förbränningen. Detta eftersom modellen blir mycket mer komplex i det fallet och för en personbilsmotor är körfall med bränsleöverskott sällan förekommande. För hela luftsystemet antags ideala gaslagen alltid vara giltig. Detta bör vara en god approximation eftersom luftsystemet har en relativt långsam dynamik som verkar inom ett temperaturintervall där ideala gaslagen fungerar bra.

För avgastemperatur och volymetrisk verkningsgrad i cylindrarna, samt verkningsgrader för turbin och kompressor, används mappningstabeller istället för specifika ekvationer. Detta är en förenkling för att undvika för komplex modellbeskrivning. Mätningar av förbränningstemperaturen i cylindrarna är för övrigt utförd i avgasmanifolden, varför temperaturen i dessa komponenter antas vara densamma.

Modellen kommer att vara endimensionell, vilket medför vissa förenklingar. Värmeffluktuationer, snabba tryckstötter och andra tredimensionella effekter tas inte någon hänsyn till. Till exempel approximeras temperaturen för varje komponent att vara konstant inom hela dess volym. Dessutom förbjuds flöden i "fel" riktning genom att använda backventiler, något som visar sig vara nödvändigt med den beskrivning av strypventiler som används. För övrigt är backflöden genom luftsystemet i de flesta fall ovanliga för en verklig motor.

I ett verkligt körfall begär föraren ett vridmoment av motorns styrsystem. Därefter ger styrsystemet insignaler till motorn beroende av motorns aktuella tillstånd, för att leverera rätt vridmoment. Samtidigt ska andra krav på motorn, såsom emissioner och dylikt, hållas inom gränserna. Då modellen inte innefattar förare, styrsystem och last, styrs motorn istället med hjälp av förutbestämda insignaler tagna från en verklig körning.

För att begränsa projektets tidsomfång modelleras inte aktuator- och givardynamik. Eftersom dessa i regel har en mycket snabbare dynamik än det övriga luftsystemet kan deras inverkan ofta försummas. Ett undantag är dock temperaturgivare som har en jämförelsevis långsam dynamik, men inte heller dessa påverkar det övriga systemet. Dessa dynamiker är av intresse först vid utveckling av styrsystemet, vilket inte tas hänsyn till i detta projekt.

2 Modelica och Dymola

I detta avsnitt beskrivs den grundläggande teorin om de verktyg som har använts i projektet för att skapa och simulera luftsystemet hos en dieselmotor.

Modelica

Modelica är ett objektorienterat modelleringsspråk som används för att modellera komplexa system, det vill säga system som till exempel innehåller mekaniska, elektriska, hydrauliska eller termiska delkomponenter [4]. Språket är fritt att använda och har utvecklats sedan 1996 av den ideella organisationen Modelica Association. Modelica används idag av flertalet biltillverkare – bland annat VW, Daimler och BMW – men även inom kraftverksindustrin hos till exempel ABB och Siemens.

Objektorienteringen medför att en modell byggs upp komponentvis, och att varje komponent i sin tur beskrivs på *deklarativ* form utifrån sina specifika ekvationer. Detta innebär att de ingående ekvationerna kan vara beroende av varandra, men behöver inte anges i en speciell ordningsföljd. Komponenterna kopplas sedan ihop med hjälp av specifika gränssnitt till en komplett modell. Det finns för Modelica en stor samling bibliotek med färdigkonstruerade komponenter och gränssnitt att använda för diverse olika tillämpningar, men det går även att konstruera helt egna.

Gränssnitten beskriver sambandet mellan anslutna komponenter genom ett bestämt antal variabler, som antingen blir så kallade ”flöden” eller ”potentialer”. De variabler man väljer som flödesvariabler kommer att summeras till noll i anslutningen, med konventionen att flöden ut från en komponent är negativ och flöden in i en komponent är positiv. De variabler man väljer som potentialer kommer istället att ha samma värde i anslutningen. Ett exempel på ett flöde och en potential är ström respektive spänning i en elektrisk krets. I kopplingen mellan två komponenter i en elektrisk krets kommer strömmarna att summeras till noll, om man tar hänsyn till riktningen på strömmen, och spänningen kommer att ha samma värde i kopplingen.

I figur 2 ses ett exempel på koduppbyggnaden av en komponent. Man deklarerar i början av koden vad det är för sorts klass som ska beskrivas, till exempel ett gränssnitt (connector) eller en modell över en komponent. I detta exempel skapas en modell av en kylare. Texten inom citationstecken är kommentarer och ignoreras vid simulering av modellen. De gränssnitt som är sammankopplade med komponenten anges lämpligen först. Därefter deklarerar vilka variabler, parametrar och konstanter som kommer att användas. Många av dessa kan hämtas ur ett färdigt bibliotek så att de får rätt enhet specificerad. Alla nödvändiga ekvationer beskrivs under ”equation” på ett deklarativt sätt, alltså i valfri ordning. En viktig del är också att specificera vilka variabler som ska föras vidare till nästa komponent via gränssnitten. Klassen avslutas till sist med ”end”.

```
model heat_exchanger
  "Heat exchanger that can be used for IC and other applications,
  cools to ambient temperature with an efficiency eta"

  Connectors.standard_connector c1
  a;
  Connectors.standard_connector c2
  a;
  Connectors.real_connector_in engine_in //Connector to look-up table
  a;

  parameter Real eta=0.7 "Heat exchanger efficiency";
  Modelica.SIunits.Temperature T_atm "Atmospheric temperature";

equation
  engine_in = T_atm;

  c2.T = T_atm*eta + (1 - eta)*c1.T;
  c1.p_air = c2.p_air;
  c1.p_exh = c2.p_exh;
  c1.dm_air = -c2.dm_air;
  c1.dm_exh = -c2.dm_exh;

end heat_exchanger;
```

Figur 2: Exempel på koduppyggnad för en komponent (en kylare i detta fall).

När man har byggt ihop en modell, med delkomponenter som har angivna gränssnitt mellan varje komponent, kompileras koden till en körbar fil som sedan kan simuleras i ett lämpligt simuleringsverktyg. När koden kompileras ställer programmet själv upp ett ekvationssystem som skapas av de ihopkopplade delmodellerna, sorterar variablerna och löser ekvationssystemet. Det finns flera olika simuleringsplattformar som kan användas ihop med Modelica, till exempel Dymola, AMESim och OpenModelica [4].

Dymola

Dymola är en programvara, baserad på Modelicaspråket, för modellering och simulering av komplexa, dynamiska system [5]. Dymola utvecklas och distribueras av Dassault Systèmes AB.

En modell skapad i Dymola kan förses med ett pålagt grafiskt gränssnitt, som till exempel ikoner eller blockdiagram för ingående komponenter och grafiska kopplingslinjer mellan dem [6]. Koden för denna grafiska struktur finns sedan inbakad som en annotering i Modelica-koden (därav beteckningarna "a;" i figur 2). Det finns också ett antal färdiga bibliotek specifikt för Dymola som framförallt innehåller komponenter lämpliga för modellering av mekaniska, hydrauliska och elektriska system inom bilindustrin [7].

Arbetsstrukturen i Dymola kan delas upp i två huvuddelar; en för modellering och en för simulering. Under modelleringsdelen finns fem grundvyer som man arbetar i:

- Kodvyn – där man skriver Modelicakod för respektive komponent
- Ikonvyn – där kan man rita en förklarande bild över komponenten
- Diagramvyn – om man har flera delkomponenter i en modell kopplas dessa samman i denna vy
- Dokumentationsvy – förklarande text om vad komponenten har för uppgift, vilka gränssnitt och parametrar den innehåller, vilken sökväg den ligger på och annan relevant information

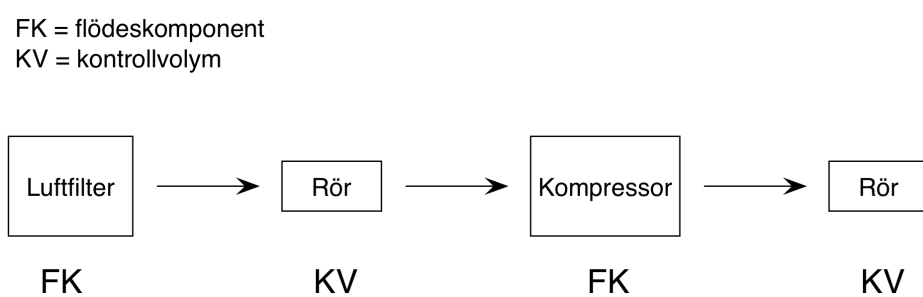
- Under simuleringsdelen sker kompileringen av koden och naturligtvis själva simuleringen, där samtliga ingående variabler kan studeras och jämföras i grafiska diagram (se exempel i figur 3). Här görs också diverse val av bland annat ekvationslösare, simuleringsintervall och tolerans. Resultatet kan sedan sparas undan för vidare analys i till exempel Matlab.



3 Metod

3.1 Grundläggande uppbyggnad

De två viktigaste grundkomponenterna i modellen består av en flödeskomponent, där ett flöde skapas utifrån en tryckdifferens mellan in- och utlopp i komponenten, och en kontrollvolym som ger upphov till ett tryck baserat på gassammansättningen inuti komponenten. Kontrollvolymen innehåller mass- och/eller energikonserveringsekvationer och flödeskomponenter bestämmer mass- och energitransporten. Exempel på kontrollvolymen är sådana där massa ansamlas, såsom insugsmanifold, avgasmanifold och rördelar mellan komponenter. Exempel på flödeskomponenter i en överladdad motor är: luftfilter, kompressor, ventiler, cylindrar (förbränningsrum) och turbin. För att skapa ett flöde genom systemet behövs minst en flödeskomponent mellan kontrollvolymerna. I figur 4 kan denna metod exemplifieras genom en liten del av förbränningsmotorns luftsystem. Att använda denna metod för att bygga upp modellen gör det lätt att utveckla och upprätthålla ett bibliotek av generella komponenter. Det underlättar samtidigt för förståelsen av modelluppbyggnaden, och metoden är dessutom väl beprövad sedan tidigare av till exempel [1].



Figur 4: Exempel på en kedja av flödeskomponenter och kontrollvolymen.

Modellen av dieselmotorn skapas genom att konstruera egna delkomponenter och sammankoppla dessa med hjälp av lämpliga egenkonstruerade gränssnitt. Anledningen till att bygga upp alla komponenter från grunden är dels för att få kontroll över modellstrukturen och dels för att få en frihet att anpassa komponenterna fritt efter egen vilja. För själva luftsystemet skapas ett "standardgränssnitt", som överför partialtryck och massflöden av luft respektive avgaser samt absolut temperatur, mellan två komponenter. Av dessa är massflöden flödesvariabler som vid varje kontaktnod summeras till noll. Partialtryck och temperatur är potentialer som får samma värde i kontaktpunkten mellan två komponenter. Valet av variabler i gränssnittet kan göras på flera olika sätt och kan variera beroende på applikationsområde. För dieselmotormodellen väljs variablerna beroende på enkelheten och det som är mest intuitivt. Att använda partialtryck och massflöden för luft och avgaser var för sig, gör det möjligt att studera andelen av respektive gas i eller genom varje komponent. Totaltrycket och det totala massflödet är helt enkelt summan av partialtrycken respektive massflödena.

Sammankopplingen av turbin och kompressor sker med ett mekaniskt gränssnitt som har vridmoment som flödesvariabel och vinkelhastighet som potential. Detta är lämpligt eftersom produkten mellan dessa storheter ger en effekt, vilket är väl motiverat då det ger ändringen i vinkelhastighet enligt Newtons andra lag. Slutligen är cylinderkomponenten och bränslesprutaren sammankopplade med ett enkelt gränssnitt som endast överför en mängd bränsle per cykel och cylinder.

En mer utförlig modellbeskrivning behandlas i kapitel 4.

3.2 Verifiering

En första studie är att studera eventuella icke-minfasbeteenden för modellen. Som nämnts tidigare (avsnitt 1.2) är ett system som uppvisar icke-minfasbeteende reglertekniskt intressant då det är svårare att reglera än om systemet inte hade haft detta beteende. Studien utförs genom att placera modellen i en stationär arbetspunkt och utföra en stegändring i två olika fall; dels i styrsignalen för EGR-arean och dels i motsvarande styrsignal för VGT-arean. Stegsvaret vid respektive stegändring studeras därefter hos de variabler där beteendet förväntas uppkomma.

Modellen ska jämföras mot uppmätt data från en personbilmotor från Volvo på 2.4 liter, under en cirka 37 minuter lång körning i Alperna. Under körningen var motorn utrustad med mätdon för att mäta upp olika variabler av intresse. Vid denna jämförelse genomförs en simulering av modellen där vissa uppmätta data låts vara insignaler till modellen. Insignalerna utgörs av:

- Atmosfärstryck
- Atmosfärstemperatur
- EGR-area
- Motorvarvtal
- Tillförd bränslemängd
- VGT-area

Dessa är givna signaler från körningen av motorn, och i detta fall när ingen hänsyn tas till förare, last, styrsystem och så vidare, så används insignalerna för att bestämma modellens arbetspunkt. Övriga variabler som är uppmätta på motorn simuleras av modellen och kan då jämföras mot körningens värden. De utsignaler som studeras är:

- Massflöde av luft genom kompressorn
- Temperatur i avgasmanifolden
- Temperatur efter turbinen
- Tryck i insugsmanifolden
- Tryck i avgasmanifolden
- Turbinvarvtal

För att utvärdera hur väl modellen stämmer med verkligheten studeras dels hur väl dynamiken följs, men också relativa fel mellan modellen och värden från körningen.

En motsvarande jämförelse av modellen görs mot en Volvo lastbilmotor på 12.8 liter, under en körning på cirka 30 minuter. Syftet med denna jämförelse är dels att se om dieselmotormodellen stämmer bra även för en större motor, men också för att utvärdera hur enkelt det är att kunna anpassa modellen till andra motoridentiteter. Simuleringen av lastbilmotorn utförs på liknande sätt som för personbilmotorn. De uppmätta värdena som låts vara insignaler till modellen är:

- Area för strypventil vid insuget
- Atmosfärstryck
- Atmosfärstemperatur
- EGR-area
- Motorvarvtal
- Tillförd bränslemängd

– VGT-area

Den justerbara strypventilen har som syfte att garantera flöde genom EGR-ventilen, genom att se till att sänka trycket i insugsmanifolden när det överstiger trycket i avgasmanifolden³. Utvärderingen av lastbilsmodellen utförs på samma sätt som för personbilsmotorn, genom att studera dynamiken och uppskatta felen. Utsignalerna som jämförs vid verifieringen är i detta fall:

- Massflöde av luft genom kompressorn
- Massflöde genom EGR-ventilen
- Tryck i insugsmanifolden
- Tryck i avgasmanifolden
- Turbinvarvtal

En utökning av modellen som känns relevant, är att ta hänsyn till temperaturdifferenser in i volymer, som inte görs i grundmodellen. Detta är en mer fysikaliskt riktig förklaring och är ett bra sätt att utvärdera hur enkelt det är att bygga ut modellen. När denna ombyggnad gjorts görs simuleringar av både personbils- och lastbilsmodellen med jämförelse mot de verkliga körningarna. Skillnader från grundmodellen kan studeras och en frågeställning om vilken modell som passar bäst för de båda modellerna kan besvaras.

Som en ytterligare verifikation jämförs en motsvarande dieselmotormodell, utvecklad i Simulink, med de uppmätta värdena från personbilsmotorn. Simulink-modellen simuleras under samma tidsintervall som för projektets modell och jämförs sedan mot den verkliga körningen för personbilsmotorn. På samma sätt som modellen i Modelica/Dymola studeras dynamiken och de relativa felen. Med hjälp av felen undersöks sedan hur lika resultat dessa två modellimplementeringar ger.

Simuleringsresultatens tillförlitlighet utvärderas också, genom att kontrollera ekvationslösarens stabilitet. Detta för att vara säker på att mindre ändringar i lösningsmetoden inte gör stora utslag på resultatet. För att utvärdera stabiliteten i denna körs två simuleringar under samma tidsintervall, identiska på allt sånär som lösarens tolerans. Den relativa skillnaden mellan de båda körningarna jämförs sedan.

När alla ovanstående undersökningar av dieselmotormodellen har genomförts finns förutsättningar för att utvärdera vad projektgruppen anser om Modelica och Dymola.

3.3 Källor

Ett uttryck som bygger på Bernoullis ekvation är hämtad ur Heywood [8], vilket är en klassisk bok om förbränningsmotorer. Denna källa är flitigt använd inom fordonsindustrin och har gott förtroende. Volymer samt insugs- och avgasmanifold är uppbyggda av enkla fysikaliska samband baserade på ideala gaslagen. Cylinderkomponenten är till stor del uppbyggd från fysikaliska ekvationer hämtade från Wahlström [9]. Dessa är mer eller mindre elementära och är verifierade av källan. Utökningen av modellen, där hänsyn tas till temperaturdifferenser in i volymer, är gjort utifrån Eriksson [1]. Då denna källa är författad av samma personer som [9] och de aktuella ekvationerna även har påträffats i andra publikationer anses den hålla god kvalitet.

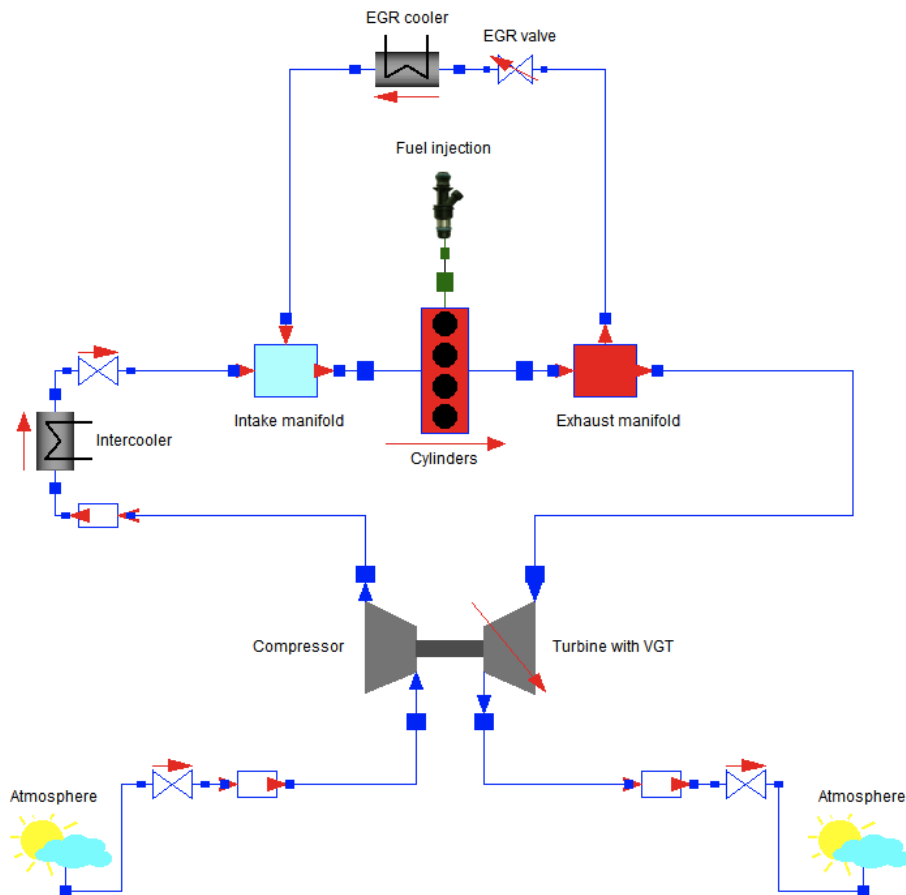
³Denna strypventil finns även för personbilsmotorn, men var helt öppen under körningen och fanns därför inte som uppmätt data.

Den befintliga Simulink-modellen är utvecklad av Markus Grahm [10] och används internt på Volvo personvagnar. För att kunna jämföra projektets modell med denna är det lämpligt att båda bygger på samma grundläggande ekvationer, i synnerhet det komplexa turboaggregatet. De ekvationer som beskriver flödet genom och verkningsgraden för turbinen och kompressorn baseras på en företagshemlig modell som inte kan redovisas i denna rapport. Dessa används trots allt på grund av att jämförelsen mot modellen gjord i Simulink ska kunna göras på ett bra sätt. Samtidigt sänker de komplexiteten på modellen med bibehållen noggrannhet. De övriga ekvationerna som nämns i turbodelen bygger på fysikaliska samband hämtade från en rapport från SAE [11], i vilken ekvationerna härleds utifrån termodynamiska begrepp.

Markus Grahm har även bistått med ett antal parametrar (se appendix B) för den aktuella personbilsmotorn, som till exempel specifika areor, volymer och verkningsgrader [12]. Några av dessa beräknas fram genom att kalibrera modellen efter uppmätt data från körningen med den verkliga motorn [13]. Motsvarande parametrar för lastbilsmotorn [14] har erhållits genom en liknande kalibrering av modellen efter körningsdata [15], som har erhållits från Volvo AB. Dessutom härleds (se appendix A) ett antal viktiga begrepp gällande förbränningskemin.

4 Modell

Modellstrukturen kan betraktas i sin helhet i figur 5. Centralt i modellen är cylinderkomponenten där förbränningen sker. Utifrån partialtrycket av främst luft – men i viss mån även avgaser – i insugsmanifolden skapas ett flöde in i cylinderrummen. En bränslespridare ser till att ge en önskad mängd bränsle. Masskonservering ger att den totala mängden gas in i cylindrarna är densamma som trycks ut till avgasmanifolden.



Figur 5: Visuell implementering av en dieselmotor i Dymola.

I anslutning till avgasmanifolden sitter EGR-ventilen som leder tillbaka en del av avgaserna till insugsmanifolden. En värmeväxlare med given verkningsgrad ser till att kyla ned avgas-temperaturen till en lägre nivå, utan att påverka tryck eller massflöde.

Efter avgasmanifolden sitter turbinen som med hjälp av avgasernas kinetiska energi får en axel att rotera. Avgaserna passerar sedan igenom turbinen och hamnar i en behållare följt av en strypventil, som tillsammans ser till att ett flöde skapas ut i atmosfären. Dessa komponenter kan betraktas som ett enkelt avgassystem.

På motsvarande sätt kan strypventilen och kontrollvolymen mellan atmosfären och kompressorn betraktas som ett enkelt luftfilter. Kompressorn, som drivs av turbinaxelns rotation, komprimerar luften, varvid temperaturen stiger. För att skapa ett givet tryck efter kompressorn krävs en kontrollvolym i direkt anslutning till denna. Luften kyls sedan ned med samma typ av värmeväxlare som EGR-kylaren. En strypventil ser sedan till att skapa ett flöde in

till insugsmanifolden, och förbränningsprocessen är därmed sluten.

I följande avsnitt beskrivs varje komponent mer ingående utifrån de ekvationer som används.

4.1 Strypventil och EGR-ventil

Strypventilen är en flödeskomponent som bygger på en tryckdifferens mellan in- och utloppet. Enligt [8] ges, efter en omskrivning av Bernoullis ekvation för inkompressibelt flöde, det totala massflödet genom en ideal strypventil som

$$\dot{m}_{\text{tot}} = \frac{A_{\text{out}}}{\sqrt{1 - A_{\text{out}}/A_{\text{in}}}} \cdot \sqrt{2 \cdot \rho_{\text{in}}} \cdot \sqrt{p_{\text{tot},\text{in}} - p_{\text{tot},\text{out}}} \quad (1)$$

där A_{in} och A_{out} är de fysiska areorna för det inkommande flödets tvärsnitt respektive själva strypningen. Detta uttryck ska fungera bra så länge som tryckdifferensen är relativt liten, men samtidigt medför rotuttrycket att negativa tryckdifferenser inte är tillåtna. Genom att införa beteckningen $C_D \equiv \frac{1}{\sqrt{1 - A_{\text{out}}/A_{\text{in}}}}$ som en dimensionslös flödeskoefficient, och sedan låta $A \equiv C_D \cdot A_{\text{out}}$ vara strypventilens *effektiva* area, kan det totala massflödet skrivas som

$$\dot{m}_{\text{tot}} = A \cdot \sqrt{2 \cdot \rho_{\text{in}}} \cdot \sqrt{\max(p_{\text{tot},\text{in}} - p_{\text{tot},\text{out}}, 0)} \quad (2)$$

där densiteten ρ_{in} fås ur ideala gaslagen i inloppet. Villkoret under det andra rottecknet förbjuder flöde i backriktningen och gör ekvationen fysikaliskt giltig. Backflöde skulle kunna tillåtas genom exempelvis en villkorssats, vilket dock hade ökat komplexiteten avsevärt för en komponent som i övrigt är relativt enkel.

Vidare ges partialflödena av luft respektive avgaser från gassammansättningen i strypventilens inlopp. Då gasen här antas vara homogen kan partialmolflödet och partialtrycket av till exempel luft direkt relateras till det totala molflödet \dot{n}_{tot} och trycket enligt

$$\frac{p_{\text{air}}}{p_{\text{tot}}} = \frac{\dot{n}_{\text{air}}}{\dot{n}_{\text{tot}}} \quad (3)$$

Massflödet av luft genom ventilen kan sedan beräknas utifrån molflödet och luftens molmassa (se appendix A). Massflödet av avgaser ges av skillnaden mellan det totala massflödet (2) och massflödet av luft.

Temperaturdifferensen över strypventilen antas vara noll. Detta är inte det reella fallet, men anses vara en fullgod approximation då temperaturdifferensen i verkligheten är mycket liten.

EGR-ventilen fungerar på samma sätt som strypventilen, med undantaget från en styrvariabel u_{egr} , som kan ges värden mellan noll och ett och används för att justera öppningen av ventilen. Det totala massflödet genom EGR-ventilen ges därför av

$$\dot{m}_{\text{tot}} = u_{\text{egr}} \cdot A \cdot \sqrt{2 \cdot \rho_{\text{in}}} \sqrt{\max(p_{\text{tot},\text{in}} - p_{\text{tot},\text{out}}, 0)} \quad (4)$$

Det ska dock påpekas att då EGR-ventilen är i kontakt med insugs- och avgasmanifolden kan det uppkomma betydligt större tryckdifferenser mellan dess in- och utlopp än för en vanlig strypventil i övriga delar av luftsystemet. Därför skulle en mer sofistikerad modell behöva användas för att säkerställa ett mer realistiskt massflöde, särskilt när flödet blir chokat. Det

finns diverse olika metoder för detta ändamål, varav en variant finns beskriven i [9], men de tillför samtidigt en betydligt större komplexitet. Eftersom EGR-ventilen förväntas släppa igenom en förhållandevis liten mängd gas per tidsenhet, och endast påverka luftsystemet i mindre grad, används därför det enkla uttrycket för massflödet givet av ekvation (4).

4.2 Volymer, insugs- och avgasmanifold

Insugs- och avgasmanifolderna går under benämningen kontrollvolym och är utrustade med tre anslutningar (in-, ut- och EGR-anslutningar). De bestämmer med hjälp av ideala gaslagen trycket utifrån massan de innehåller. Flödet genom deras anslutningar bestäms sedan av omgivande flödeskomponenter. Partialmassorna i manifolderna beräknas genom masskonservering enligt

$$\begin{aligned}\frac{d}{dt}m_{\text{air}} &= \dot{m}_{\text{air,in}} + \dot{m}_{\text{air,out}} + \dot{m}_{\text{air,egr}} \\ \frac{d}{dt}m_{\text{exh}} &= \dot{m}_{\text{exh,in}} + \dot{m}_{\text{exh,out}} + \dot{m}_{\text{exh,egr}}\end{aligned}\tag{5}$$

med teckenkonvention att massflöden ut från en komponent är negativa. Partialtrycken går, då ideala gaslagen är linjär, att bestämma med hjälp av

$$\begin{aligned}p_{\text{air}} \cdot V &= m_{\text{air}} \cdot R_{\text{air}} \cdot T \\ p_{\text{exh}} \cdot V &= m_{\text{exh}} \cdot R_{\text{exh}} \cdot T\end{aligned}\tag{6}$$

där V är volymen, T är den absoluta temperaturen samt R_{air} och R_{exh} är de specifika gaskonstanterna för luft respektive avgaser (se appendix A). Temperaturen T anses vara jämn över hela volymen och approximeras till att vara samma som temperaturen in i volymen.

Förutom insugs- och avgasmanifold behövs kontrollvolymen även på andra ställen i modellen för att generera tryck. Dessa beskrivs på precis samma sätt men saknar anslutningar för EGR-ventil.

Det är denna enklare modell av en kontrollvolym som används vid simulering och verifiering, utom i de fall då annat anges.

4.2.1 Utökad modell

Sambanden ovan är en förenkling i den mening att temperaturen inuti en volym är densamma som temperaturen hos gasen in i volymen. En bättre modellbeskrivning vore enligt [1] att använda såväl mass- som energikonservering, vilket kan härledas genom tidsderivering av ideala gaslagen. Denna beskrivning kan sammanfattas i följande tre ekvationer:

$$m = \frac{p \cdot V}{R \cdot T}\tag{7}$$

$$\frac{dT}{dt} = \frac{1}{m \cdot c_v} \left[\dot{m}_{\text{in}} \cdot c_v (T_{\text{in}} - T) + R(T_{\text{in}} \cdot \dot{m}_{\text{in}} + T \cdot \dot{m}_{\text{out}}) + \dot{Q} \right]\tag{8}$$

$$\frac{dp}{dt} = \frac{R \cdot T}{V} (\dot{m}_{\text{in}} + \dot{m}_{\text{out}}) + \frac{m \cdot R}{V} \cdot \frac{dT}{dt}\tag{9}$$

där m symboliserar den *totala* massan samt \dot{m}_{in} och \dot{m}_{out} det *totala* massflödet in respektive ut. Återigen gäller teckenkonvention att massflöden ut är negativa. På samma sätt är p totaltrycket inuti volymen och T är temperaturen, som antas vara densamma som temperaturen

ut. Den specifika värmekapaciteten för gasen är c_v , R är den specifika gaskonstanten och \dot{Q} är eventuell värmeöverföring genom kontrollvolymens ytor. Värmeöverföringen antogs här alltid vara noll. På samma sätt som för den enklare modellen beskrivs partialmassorna och partialtrycken enligt ekvation (5) och (6).

De allmänna volymerna (utan EGR-anslutning) har endast en in- och en utgång, vilket underlättar när de totala massflödena i ekvation (8) - (9) betecknas som summan av massflödena för luft och avgaser. Då finns nämligen bara två temperaturer att hantera: den gemensamma temperaturen in och temperaturen inuti, som är densamma som temperaturen ut till nästa komponent. Beräkningen av c_v utifrån andelen av respektive gas utförs genom en linjär anpassning enligt

$$c_v = c_{v,\text{exh}} + (c_{v,\text{air}} - c_{v,\text{exh}}) \cdot \frac{\dot{m}_{\text{air}}}{\dot{m}_{\text{tot}}} \quad (10)$$

Som appendix A visar är $R_{\text{exh}} \approx R_{\text{air}}$, vilket gör att R kan anses vara konstant oberoende av gassammansättningen.

De två manifolderna blir något mer komplicerade att beskriva i och med den extra anslutningen för EGR-ventilen, ty för dessa finns en tredje temperatur att ta med i beräkningen. Det gör att ekvation (8) - (9) ger några extra termer. I övrigt beskrivs dessa på samma sätt som den utökade modellen av en kontrollvolym.

4.3 Cylindrar

Cylinderkomponenten är en flödesgenererare som skapar flöden utifrån bland annat partialtrycken av luft och avgaser i insugsmanifolden. Enligt [9] ges massflödet (efter uppdelning i partialflöden) in i cylindrarna av

$$\begin{aligned} \dot{m}_{\text{air},\text{in}} &= \frac{\eta_{\text{vol}} \cdot p_{\text{air},\text{in}} \cdot n_e \cdot V_d}{120 \cdot R_{\text{air}} \cdot T_{\text{in}}} \\ \dot{m}_{\text{exh},\text{in}} &= \frac{\eta_{\text{vol}} \cdot p_{\text{exh},\text{in}} \cdot n_e \cdot V_d}{120 \cdot R_{\text{exh}} \cdot T_{\text{in}}} \end{aligned} \quad (11)$$

där η_{vol} är cylindrarnas volymetriska verkningsgrad, $p_{\text{air},\text{in}}$ och $p_{\text{exh},\text{in}}$ är partialtrycket av luft respektive avgaser i insugsmanifolden, n_e är motorns varvtal i varv per minut, V_d är den totala cylindervolymen samt T_{in} är temperaturen för gasen i insuget. Den volymetriska verkningsgraden, som i sin tur beror på motorvarvtalet och mängden bränsle som sprutas in i cylindrarna, modelleras efter tabellerade värden [12].

Massflödet av bränsle kan relateras till styrsignalen u_{fuel} , uttryckt i [mg/(cykel-cylinder)], som

$$\dot{m}_{\text{fuel}} = \frac{10^{-6}}{120} \cdot u_{\text{fuel}} \cdot n_e \cdot n_{\text{cyl}} \quad (12)$$

där n_{cyl} är antalet cylindrar för den aktuella motorn. Beroende på typ av simulering styrs sedan u_{fuel} manuellt eller som indata från uppmätta värden [13].

Massflödet av avgaser ut från cylindrarna ges enligt masskonservering av

$$\dot{m}_{\text{exh,out}} = \dot{m}_{\text{exh,in}} + \dot{m}_{\text{fuel}} + AFR \cdot \dot{m}_{\text{fuel}} \quad (13)$$

där den sista termen beskriver mängden luft som vid förbränningen blir avgaser. AFR -värdet (air-to-fuel ratio) är sambandet mellan massa för luft och bränsle vid stökiometrisk förbränning, vilket behandlas i appendix A. Massflödet av luft ut från cylindrarna är sedan differensen mellan den luft som kommer in och den som omvandlas till avgaser.

4.4 Turbo

4.4.1 Turbinaxel

Turbinaxeln förbinder turbinen och kompressorn med ett mekaniskt gränssnitt. Den har en inre egenskap i form av ett givet masströghetsmoment J . Masströghetsmomentet beskriver trögheten i axeln samt turbinens och kompressorns roterande delar. Turbinaxelns rotation beskrivs sedan med hjälp av Newtons andra lag enligt

$$\frac{d}{dt}\omega = \frac{M_{\text{turb}} + M_{\text{cmpr}}}{J} \quad (14)$$

där ω är vinkelhastigheten, M_{turb} är turbinens drivande vridmoment och M_{cmpr} är vridmomentet som driver kompressorn. Här gäller teckenkonvention att ett vridmoment som drivs av axeln är negativ (M_{cmpr} i detta fall).

4.4.2 Turbin med variabel geometri (VGT)

Turbinen går under benämningen flödeskomponent och ger ett massflöde baserat på tryckdifferensen över komponenten. Utöver standardanslutningar för massflöden in och ut har den i modellen även en mekanisk anslutning mot turbinaxeln. Effekten som turbinen genererar beror i sin tur på energin hos avgaserna som passerar genom den. Enligt termodynamikens första huvudsats kan effekten enligt [11] skrivas

$$P = \dot{m} \cdot \Delta h = \dot{m} \cdot c_p \cdot \Delta T \quad (15)$$

där \dot{m} är massflödet, c_p är den specifika värmekapaciteten samt Δh och ΔT är entalpi- respektive temperaturförändringen mellan turbinens in- och utlopp. Det sista uttrycket i ekvation (15) gäller när avgaserna betraktas som en ideal gas. För en ideal isentropisk process kan temperaturkvoten över komponenten relateras till en tryckkvot enligt

$$\frac{T_{\text{in}}}{T_{\text{out}}} = \left(\frac{p_{\text{in}}}{p_{\text{out}}} \right)^{\frac{\gamma-1}{\gamma}} \quad (16)$$

där γ är värmekapacitetskvoten för avgaserna (se appendix B). Den verkliga entalpiförändringen över turbinen är enligt [11] inte ideal, vilket ger upphov till en isentropisk verkningsgrad definierad som

$$\eta_{\text{trbn,is}} = \frac{\Delta T}{\Delta T_{\text{is}}} \quad (17)$$

Genom att kombinera ekvation (15), (16) och (17) ges effekten som turbinen genererar ut mot turbinaxeln av uttrycket

$$P_{\text{trbn}} = \dot{m} \cdot c_p \cdot \eta_{\text{trbn,is}} \cdot T_{\text{in}} \left(1 - \left(\frac{p_{\text{out}}}{p_{\text{in}}} \right)^{\frac{\gamma-1}{\gamma}} \right) \quad (18)$$

Därefter återstår att modellera massflödet \dot{m} och verkningsgraden $\eta_{\text{trbn, is}}$ med någon lämplig metod. Det finns ett flertal olika tillvägagångssätt⁴ som ofta inkluderar optimeringsfunktioner med parametrar som anpassas efter experimentellt uppmätta värden. Som tidigare nämnt används i detta projekt en intern företagshemlig modell för massflödet, medan turbinens verkningsgrad beräknas direkt utifrån mappningstabeller [12].

Det mekaniska gränssnittet kräver per definition ett vridmoment M_{trbn} som flödesvariabel, vilket kan relateras till effekten enligt

$$P_{\text{trbn}} \cdot \eta_{\text{shaft}} = M_{\text{trbn}} \cdot \omega \quad (19)$$

där η_{shaft} och ω är turbinaxelns verkningsgrad respektive vinkelhastighet.

Temperaturen efter turbinen ges enligt [11] av uttrycket

$$T_{\text{out}} = T_{\text{in}} \left(1 - \eta_{\text{trbn, is}} \left(1 - \left(\frac{p_{\text{out}}}{p_{\text{in}}} \right)^{\frac{\gamma-1}{\gamma}} \right) \right) \quad (20)$$

4.4.3 Kompressor

Kompressorn är liksom turbinen en flödeskomponent som skapar ett luftflöde utifrån givna tryck före och efter komponenten. Syftet är att med hjälp av turbinaxelns rotation komprimera insugsluften för att skapa ett laddtryck i insugsmanifolden. Den har liksom turbinen två gränssnitt; dels ett standardgränssnitt för luftflödet och dels ett mekaniskt gränssnitt mot axeln.

Effekten som krävs för att driva kompressorn beskrivs enligt [11] på samma sätt som turbinen, med entalpiförändringen över komponenten som utgångspunkt. Med verkningsgraden definierad av ekvation (17) kan den drivande effekten skrivas

$$P_{\text{cmpr}} = \dot{m} \cdot c_p \cdot \eta_{\text{cmpr, is}} \cdot T_{\text{in}} \left(\left(\frac{p_{\text{out}}}{p_{\text{in}}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) \quad (21)$$

Precis som för turbinen behöver massflödet och verkningsgraden modelleras med någon metod som ofta involverar optimeringsparametrar som anpassas efter uppmätta värden. I modellen använts även i detta fall en intern modell för massflödet, medan verkningsgraden modelleras utifrån en mappningstabell [12].

Effekten förhåller sig till vridmomentet M_{cmpr} och turbinaxelns vinkelhastighet enligt uttrycket

$$P_{\text{cmpr}} = M_{\text{cmpr}} \cdot \omega \quad (22)$$

Slutligen kan temperaturen efter kompressorn beräknas på samma sätt som i ekvation (20).

⁴Några olika exempel finns presenterade i [9] och [11].

4.5 Kylare

Kylaren är en typ av värmeväxlare som använder omgivande luft som kylmedium för att sänka temperaturen på den genomströmmande gasen. Kylaren är i denna modell utrustad med standardanslutningar in och ut och är den enda komponenten som vare sig är en direkt flödeskomponent eller en kontrollvolym. Den är definierad genom att ha samma massflöden och tryck ut som den har in. Med en given verkningsgrad η kan temperaturen ut från kylaren skrivas som en linjär anpassning mellan omgivande lufttemperatur T_{amb} och inkommande gastemperatur T_{in} . De fullständiga ekvationerna som beskriver kylaren kan sammanfattas som

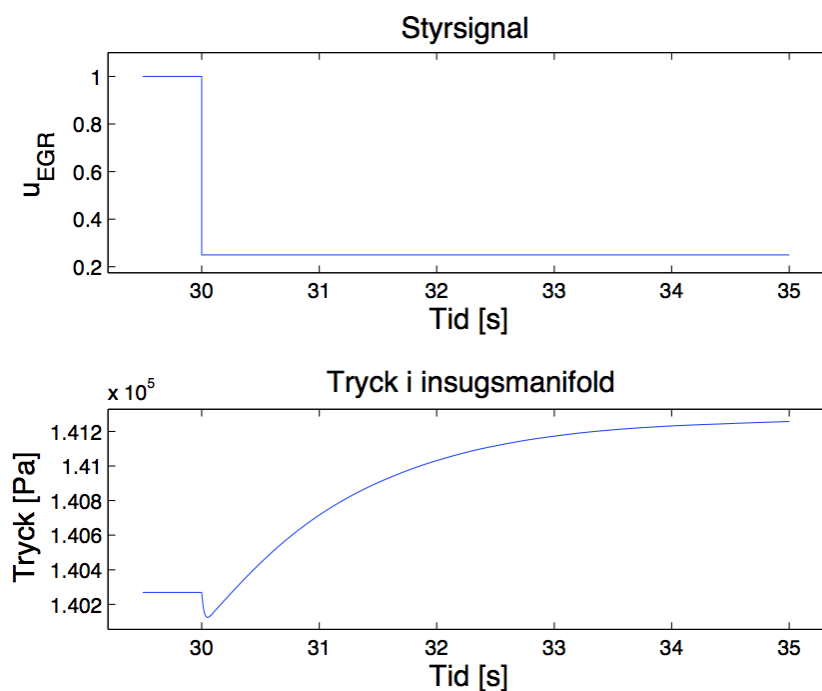
$$\begin{aligned} \dot{m}_{\text{out}} &= \dot{m}_{\text{in}} \\ p_{\text{out}} &= p_{\text{in}} \\ T_{\text{out}} &= T_{\text{amb}} \cdot \eta + (1 - \eta) \cdot T_{\text{in}} \end{aligned} \tag{23}$$

5 Verifiering och utvärdering

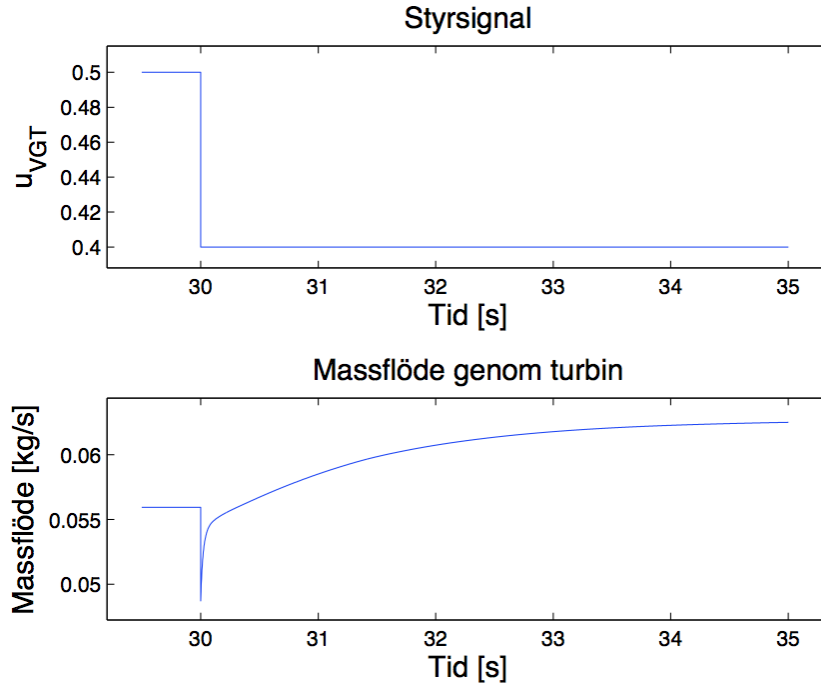
I följande avsnitt presenteras resultaten av simuleringarna med dieselmotormodellen i de fall som beskrevs i kapitel 3.2. Efter att simuleringarna utfördes i Dymola sparades datan undan och behandlades i Matlab.

5.1 Icke-minfasbeteenden

För att kontrollera om modellen var av icke-minfastyp ändrades styrsignalen till EGR:en med ett steg, varefter trycket i insugsmanifolden studerades. Man kan i figur 6 tydligt se icke-minfasbeteendet på insugstrycket. På samma sätt visas i figur 7 hur styrsignalen för VGT-arean ändrades med ett steg och hur detta påverkade massflödet genom turbinen. Även här ser man tydligt det eftersökta beteendet. Simuleringen kördes i en stationär arbetspunkt med ett motorvarvtal på 2000 rpm.



Figur 6: Undersökning av icke-minfasbeteende för trycket i insugsmanifolden vid steg i EGR-arean.



Figur 7: Undersökning för icke-minfasbeteende för massflödet av luft genom turbinen vid steg i VGT-arean.

5.2 Volvo personbilsmotor

Utvärderingen av hur väl modellen följer de uppmätta värdena för en personbilsdiesel på 2.4 liter utfördes genom att låta några av dessa värden vara insignaler (grafer för dessa visas i appendix B). EGR-ventilen var stängd under hela körningen. Modellen kompilerades och simulerades sedan med 100 000 tidssteg under de totalt 2 200 sekunder (ca 37 minuter) som det fanns uppmätta värden för. Hela simuleringsprocessen krävde ca 15 sekunder i anspråk.

De simulerade resultaten studerades med de uppmätta motsvarigheterna som referens. Figur 8 - 13 (a) visar resultaten under de första 500 sekunderna av simuleringen (för att inte få en allt för rörig plot). Figur 8 - 13 (b) visar de relativa felen η_{rel} , beräknade vid varje tidpunkt i för respektive variabel y enligt

$$\eta_{\text{rel}}(i) = \frac{y_{\text{mod}}(i) - y_{\text{meas}}(i)}{y_{\text{meas}}(i)} \quad (24)$$

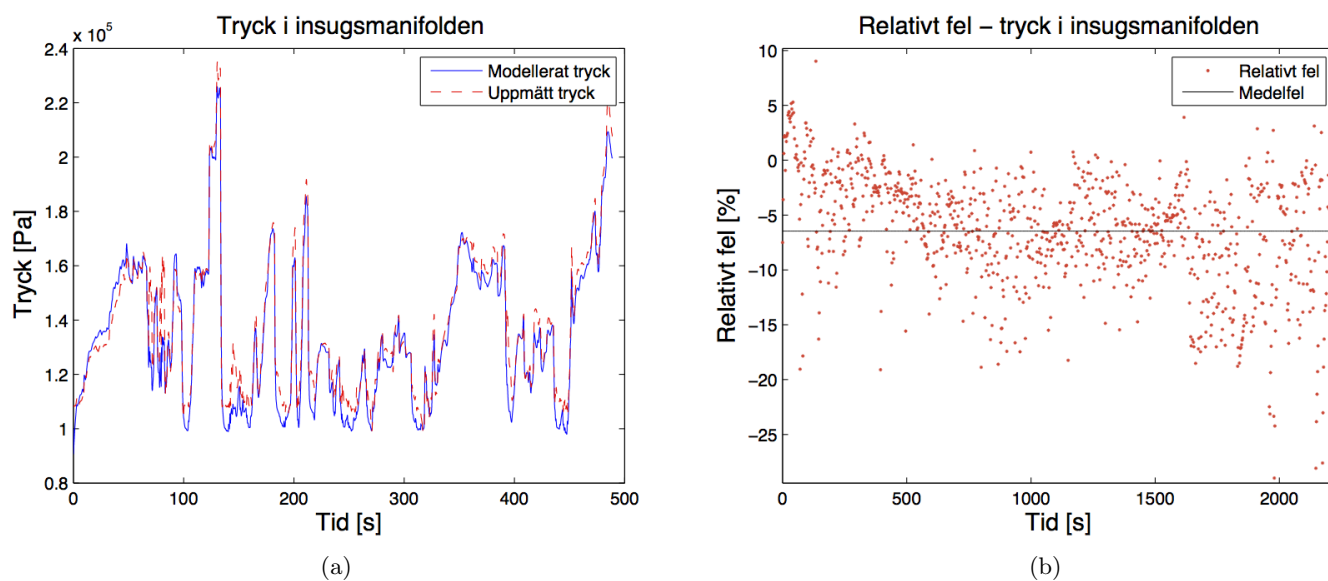
Dessa fel beräknades för hela simuleringen, men i figurerna visas de relativa felen för var 100:e tidpunkt för att få en bättre överskådlighet. Därtill uppskattades medelfelet genom att summera de relativa felen över alla tidpunkter och dividera med det totala antalet tidssteg. Även standardavvikelsen beräknades för att få en uppfattning om hur mycket felet avviker från medelvärdet. Feluppskattningarna sammanfattas i tabell 1.

Den utökade modellen, där temperaturen i alla volymer beräknades utifrån mass- och energikonservering, simulerades på samma sätt som den enklare varianten. Resultaten visade sig vara så lika de föregående, som visas i figur 8 - 13, att det inte ger något mervärde att utföra liknande grafer. Feluppskattningarna visas i tabell 1.

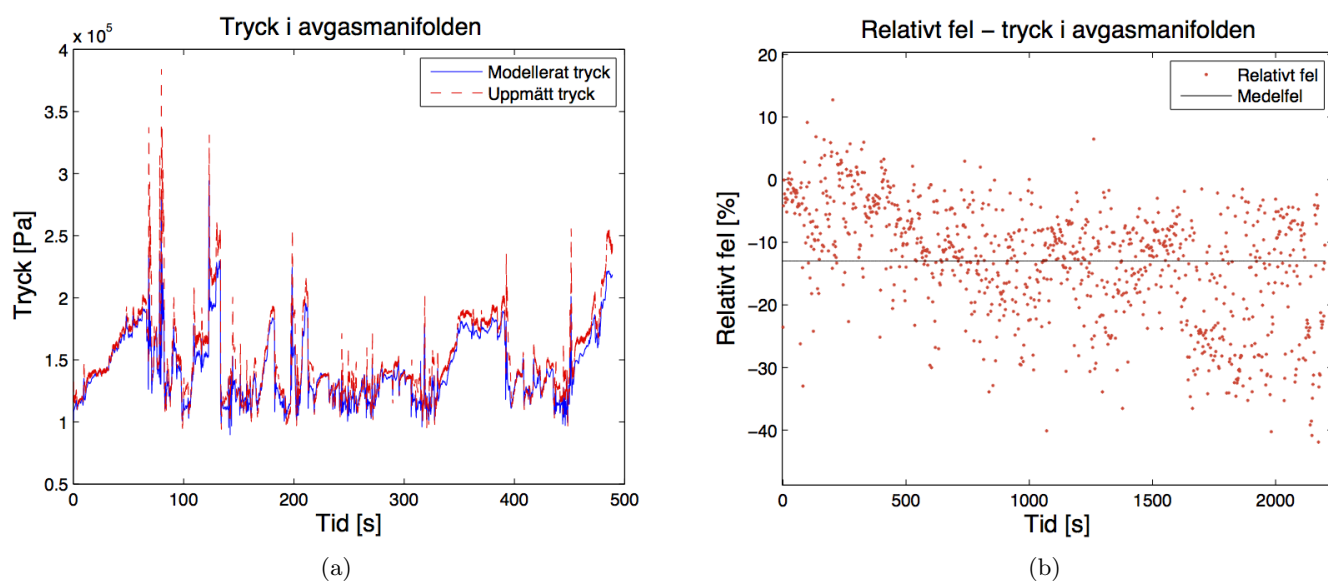
Dieselmotormodellen jämfördes också mot en modell utvecklad i Simulink, som simulerades under samma tidsintervall. Feluppskattningar utfördes på samma vis som tidigare enligt

ekvation (24), med de uppmätta värdena som referens, och finns sammanfattade i nedre delen av tabell 1.

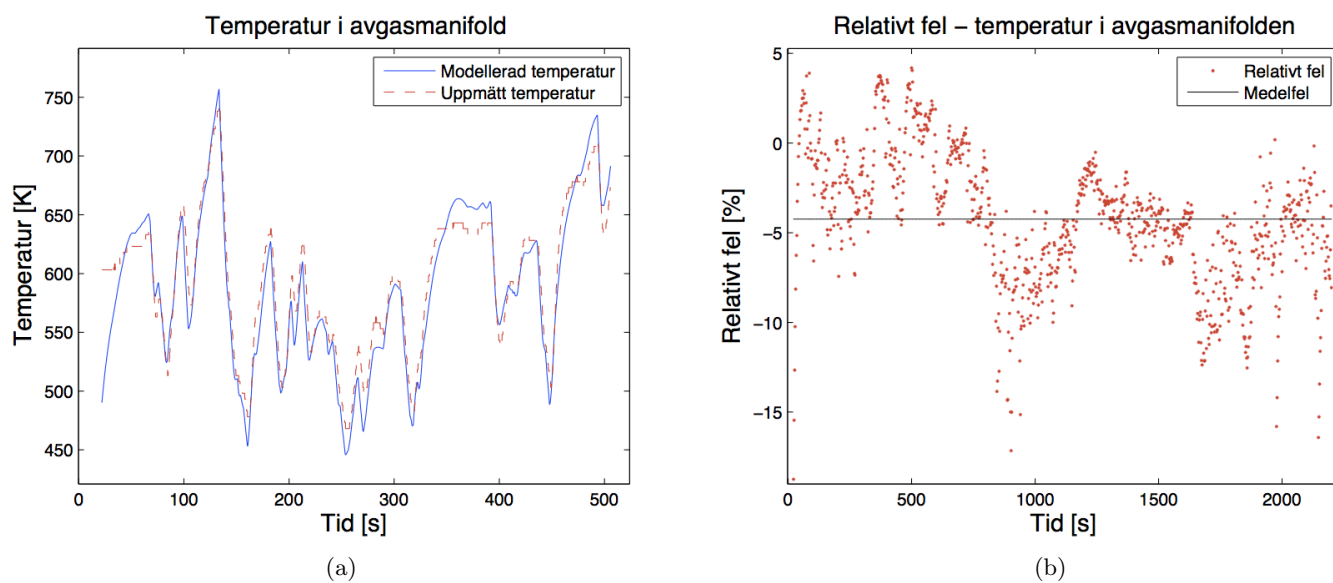
För temperaturen i avgasmanifolden och efter turbinen för samtliga simuleringar utfördes en lågpasfiltrering på de modellerade värdena för att bättre följa den relativt långsamma dynamiken hos den verkliga temperaturgivaren. Detta realiserades genom ett första ordningens lågpasfilter med en tidskonstant $\tau \approx 1$ sekund, vilket är i samma storleksordning som dynamiken hos temperaturgivaren. En konsekvens av detta är att de första sekunderna av varje simulering måste bortses från, eftersom de filtrerade resultaten börjar från noll vid startögonblicket (vilket kan urskiljas i figur 10 - 11 (a)).



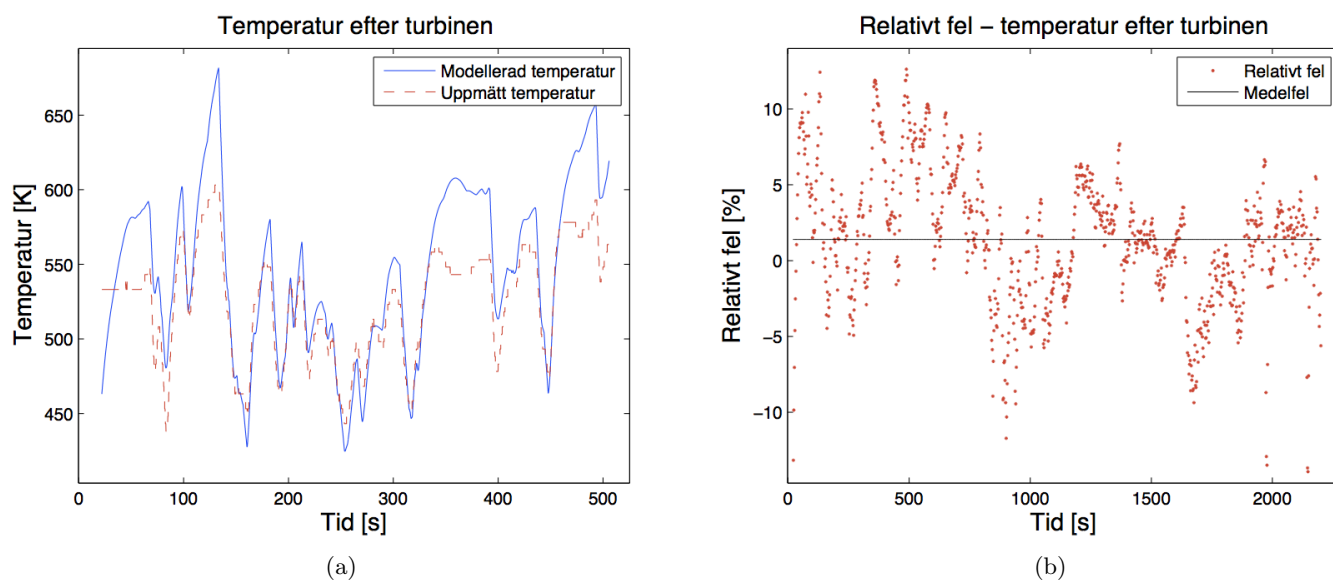
Figur 8: Tryck i insugsmanifold. Medelfel: -6.4521 %, maximalt fel: -28.9915 %, standardavvikelse: 5.0895 %.



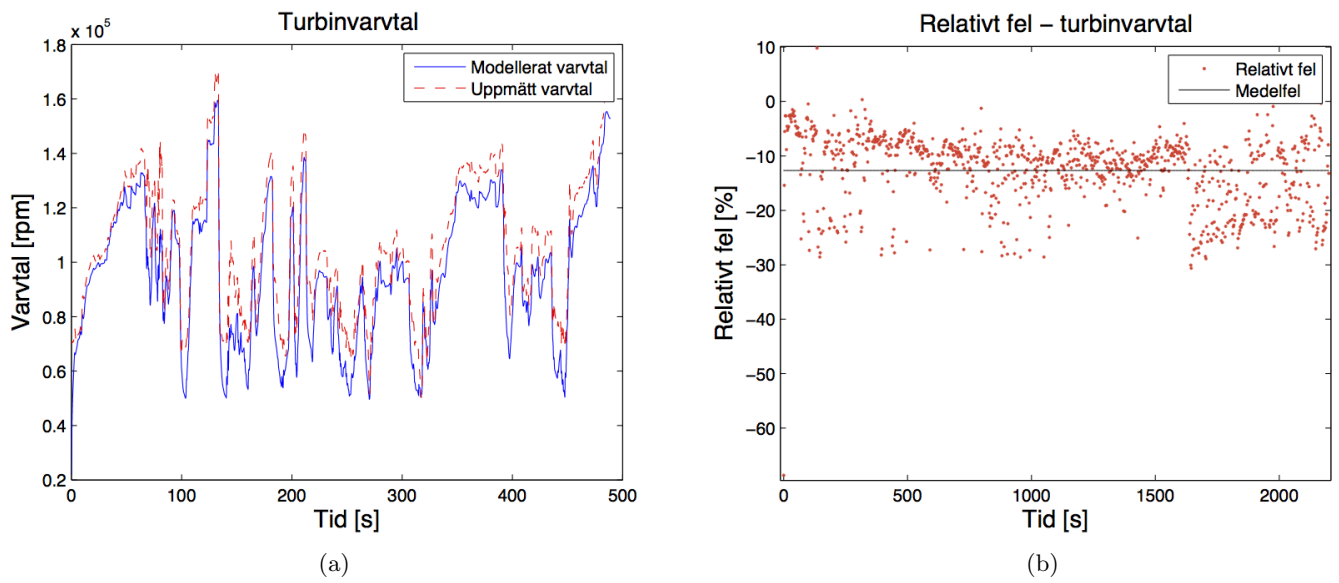
Figur 9: Tryck i avgasmanifold. Medelfel: -12.9910 %, maximalt fel: -52.7734 %, standardavvikelse: 9.2322 %.



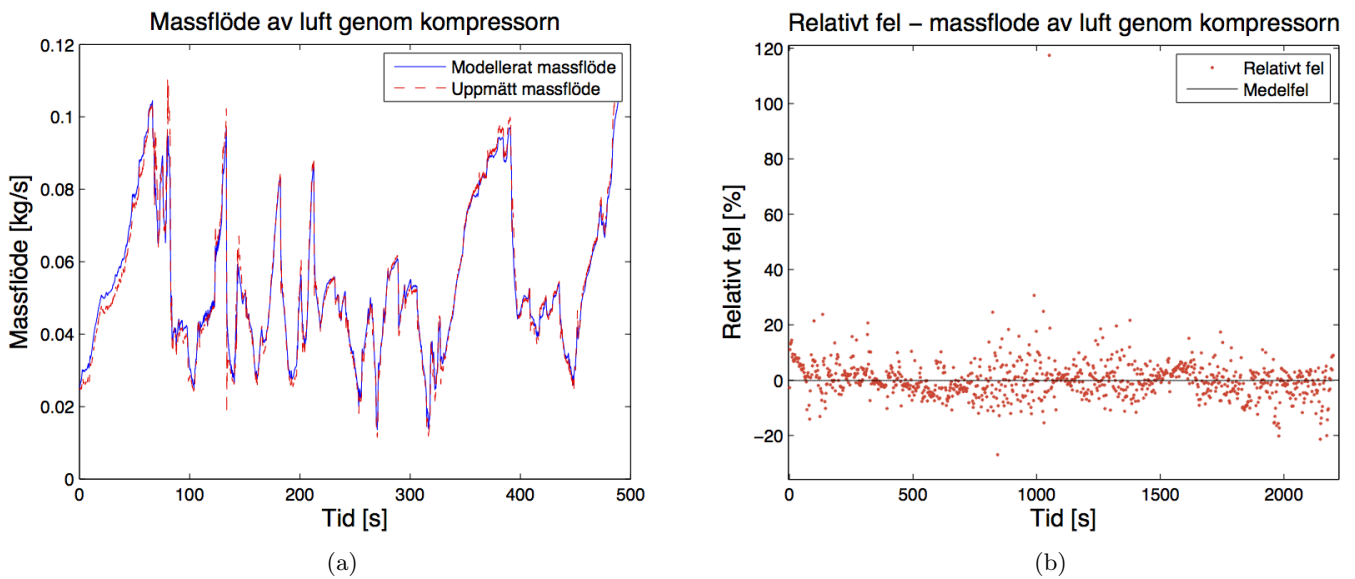
Figur 10: Temperatur i avgasmanifold. Medelfel: -4.2471 %, maximalt fel: -18.7484 %, standardavvikelse: 3.6491 %.



Figur 11: Temperatur efter turbinen. Medelfel: 1.4014 %, maximalt fel: -16.6161 %, standardavvikelse: 4.3782 %.



Figur 12: Turbinvarvtal. Medelfel: -12.6608 %, maximalt fel: -68.6727 %, standardavvikelse: 6.2090 %.



Figur 13: Massflöde av luft genom kompressorn. Medelfel: -0.1794 %, maximalt fel: 174.2544 %, standardavvikelse: 6.0883 %.

Tabell 1: Feluppskattningar vid verifiering mot personbilsmotor.

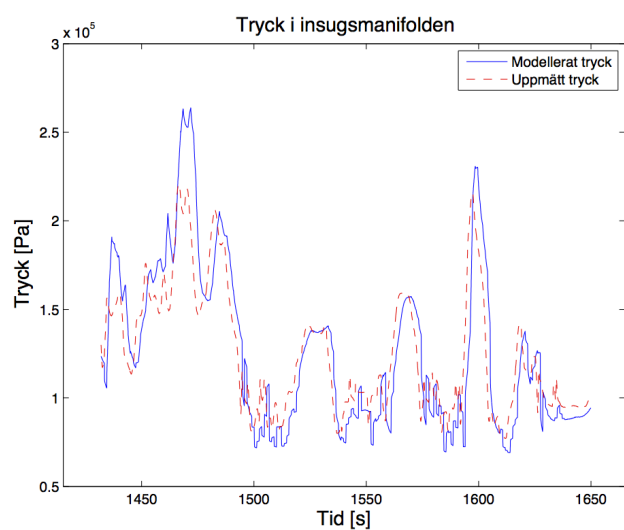
Dymola	Medelfel [%]	Maximalt fel [%]	Standardavvikelse [%]
Tryck i insugsmanifold	-6.4521	-28.9915	5.0895
Tryck i avgasmanifold	-12.9910	-52.7734	9.2322
Temperatur i avgasmanifold	-4.2471	-18.7484	3.6491
Temperatur efter turbin	1.4014	-16.6161	4.3782
Turbinvarvtal	-12.6608	-68.6727	6.2090
Massflöde genom kompressor	-0.1794	174.2544	6.0883
Dymola (utökad modell)	Medelfel [%]	Maximalt fel [%]	Standardavvikelse [%]
Tryck i insugsmanifold	-6.5833	-28.4789	5.1323
Tryck i avgasmanifold	-13.0702	-48.6621	9.3276
Temperatur i avgasmanifold	-4.3247	-18.7295	3.6583
Temperatur efter turbin	1.3396	-16.8022	4.4134
Turbinvarvtal	-12.8200	-68.6727	6.3094
Massflöde genom kompressor	-0.2962	179.7312	6.2657
Simulink	Medelfel [%]	Maximalt fel [%]	Standardavvikelse [%]
Tryck i insugsmanifold	-6.3604	-29.5400	5.4092
Tryck i avgasmanifold	-9.2816	-38.0394	7.8824
Temperatur i avgasmanifold	-5.4190	-19.4333	3.6270
Temperatur efter turbin	0.0082	-18.0505	4.3929
Turbinvarvtal	-11.9542	-85.7366	5.8775
Massflöde genom kompressor	1.1330	177.6680	6.6625

5.3 Volvo lastbilsmotor

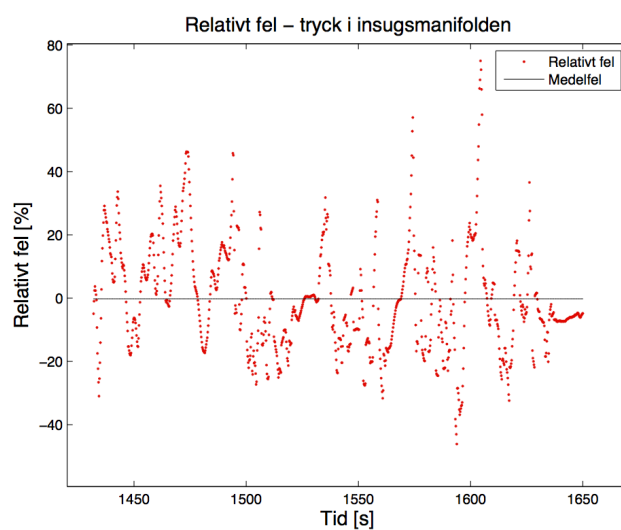
Simuleringen av modellen vid verifikation mot lastbilsmotorn utfördes enligt samma metod som i avsnitt 5.2. Samma insignaler som tidigare användes, med ett tillägg för en strypventil med justerbar area vid insuget. Till skillnad från personbilssimuleringen var EGR-ventilen i detta fall inte stängd under körningen. Modellen simulerades vid totalt 10 000 tidssteg under ett ca 200 sekunder långt intervall i slutet av den 30 minuter långa körningen. På grund av allt för låga varvtal visade sig modellen inte fungera i vissa andra delar av körningen. Simuleringstiden var för övrigt i samma relativa storleksordning som tidigare.

De simulerade resultaten visas i figur 14 - 18 (a), där utsignalerna jämförs med sina uppmätta motsvarigheter. Notera att de modellerade värdena för tryck i insugs- och avgasmanifold är något förskjutna i tiden jämfört med de uppmätta trycken. Figur 14 - 18 (b) visar felen för var 10:e tidpunkt samt en markering för medelfelet. Feluppskattningarna utfördes på samma sätt som för personbilsmotorn och finns sammanfattade i tabell 2. Enda undantaget är massflödet genom EGR-ventilen där de absoluta felen gav en mer sanningsenlig bild för de små numeriska värdena.

En simulering utfördes också efter samma förutsättningar med den utökade modellen, där insugs- och avgasmanifolden byggdes ut med kontrollvolym baserade på såväl mass- som energikonservering för modellering av temperaturen. Feluppskattningarna för denna simulering visas i nedre delen av tabell 2.

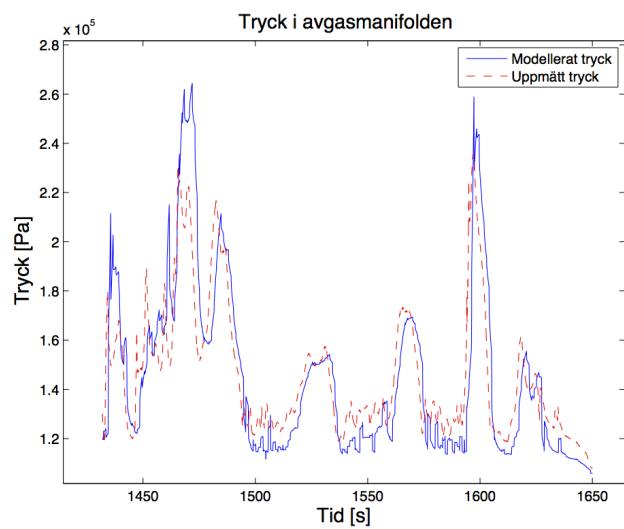


(a)

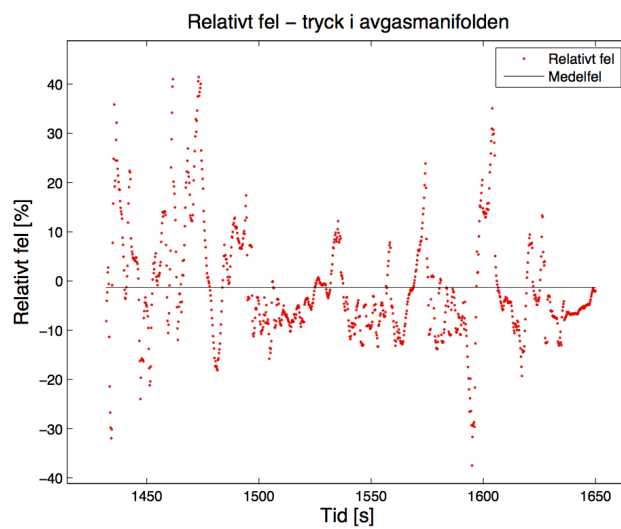


(b)

Figur 14: Tryck i insugsmanifold. Medelfel: -0.2021 %, maximalt fel: 74.9994 %, standardavvikelse: 17.2425 %.

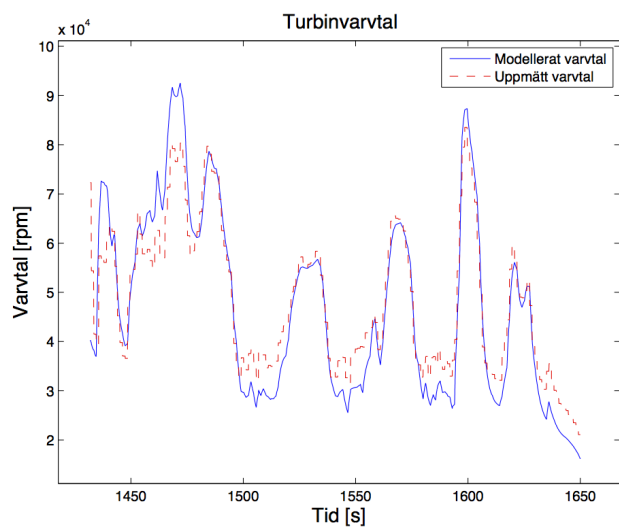


(a)

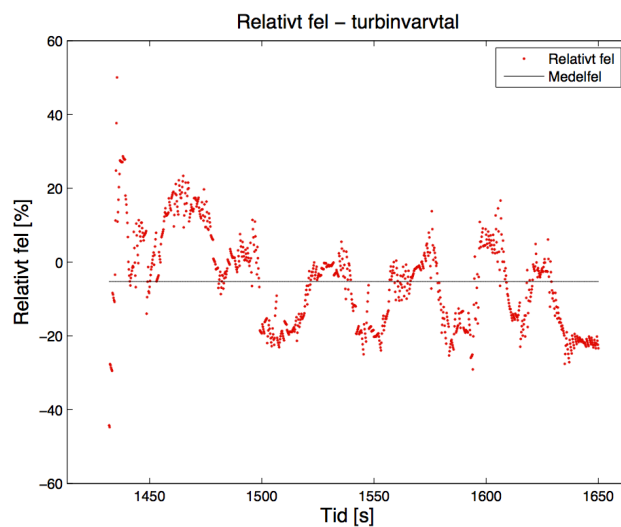


(b)

Figur 15: Tryck i avgasmanifold. Medelfel: -1.2783 %, maximalt fel: 43.9100 %, standardavvikelse: 11.3024 %.

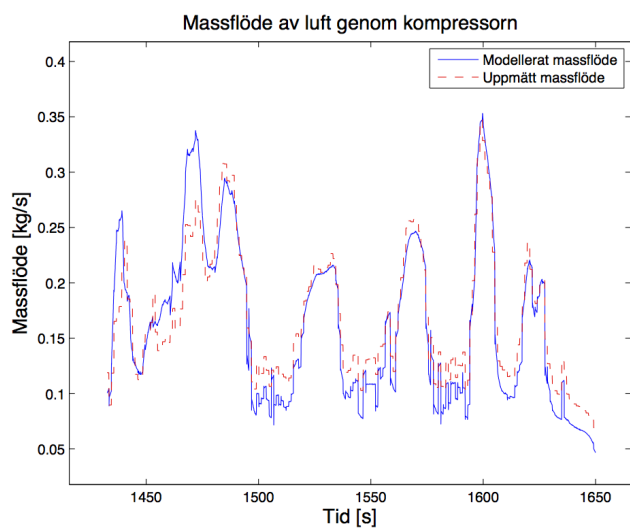


(a)

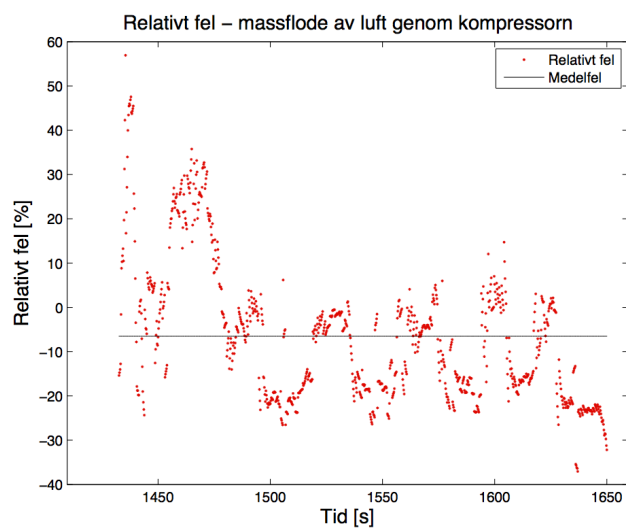


(b)

Figur 16: Turbinvarvtal. Medelfel: -5.2819 %, maximalt fel: 56.1837 %, standardavvikelse: 12.6798 %.

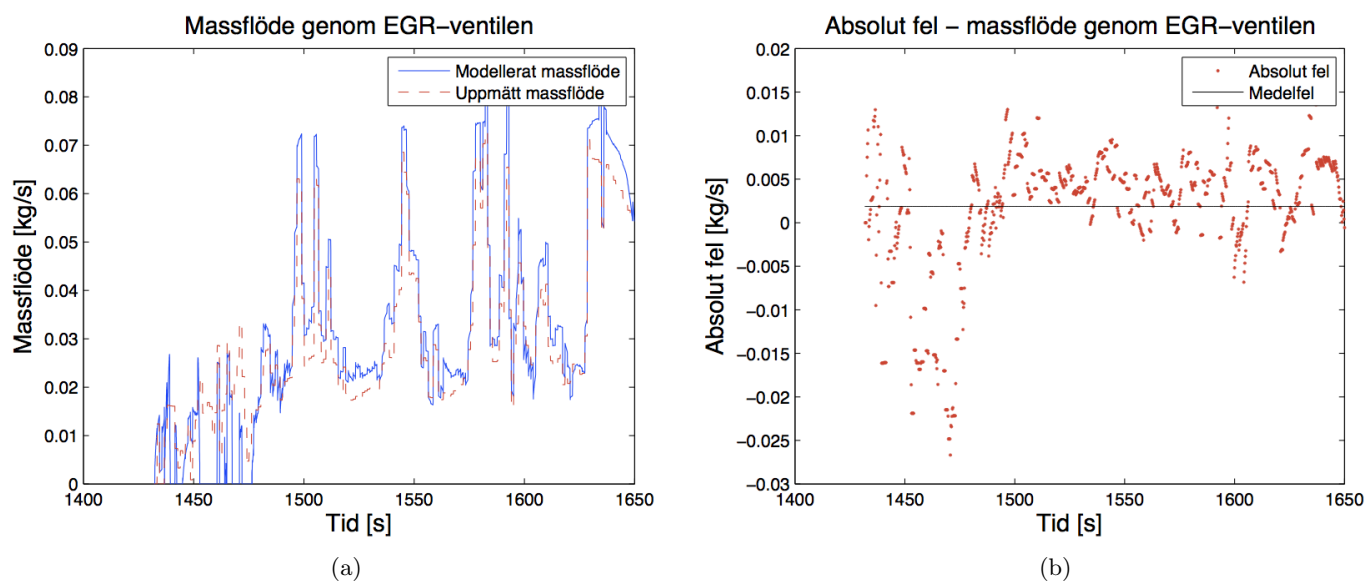


(a)



(b)

Figur 17: Massflöde av luft genom kompressorn. Medelfel: -6.5313 %, maximalt fel: 62.3485 %, standardavvikelse: 15.5528 %.



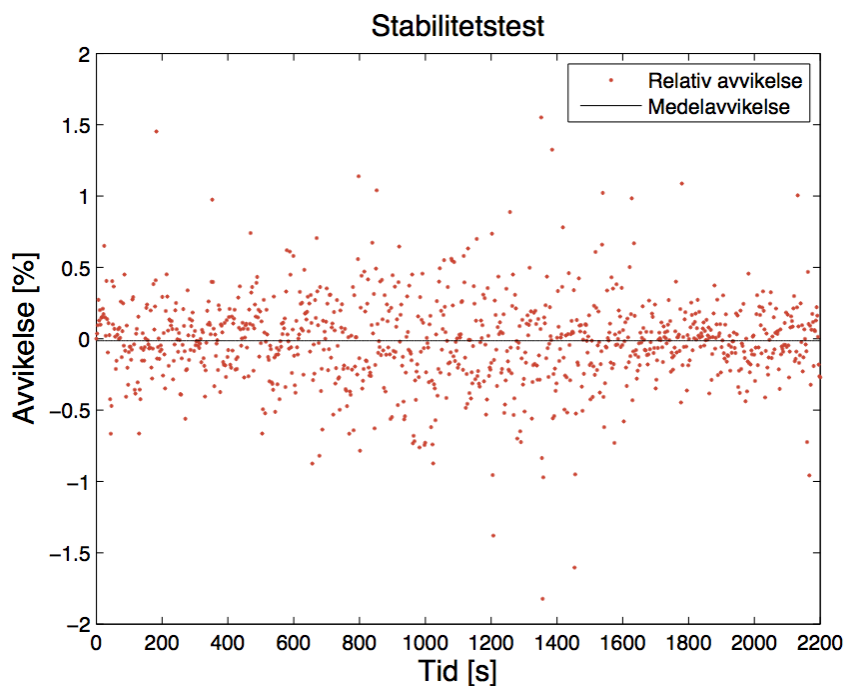
Figur 18: Massflöde av luft genom kompressorn. Medelfel: 0.0019 kg/s, maximalt fel: -0.0286 kg/s, standardavvikelse: 0.0074 kg/s.

Tabell 2: Feluppskattningar av modellen vid verifiering mot lastbilsmotor.

Dymola	Medelfel	Maximalt fel	Standardavvikelse
Tryck i insugsmanifold	-0.2021 %	74.9994 %	17.2425 %
Tryck i avgasmanifold	-1.2783 %	43.9100 %	11.3024 %
Turbinvarvtal	-5.2819 %	56.1837 %	12.6798 %
Massflöde genom kompressor	-6.5313 %	62.3485 %	15.5528 %
Massflöde genom EGR-ventil	0.0019 kg/s	-0.0286 kg/s	0.0074 kg/s
Dymola (utökad modell)	Medelfel	Maximalt fel	Standardavvikelse
Tryck i insugsmanifold	0.5786 %	75.9922 %	16.8787 %
Tryck i avgasmanifold	-1.8676 %	44.0987 %	11.2664 %]
Turbinvarvtal	-5.5743 %	54.2038 %	12.5298 %
Massflöde genom kompressor	-8.8048 %	59.2370 %	15.8153 %
Massflöde genom EGR-ventil	$-3.0854 \cdot 10^{-4}$ kg/s	-0.0279 kg/s	0.0068 kg/s

5.4 Stabilitet i simuleringen

Ekvationslösaren som användes i samtliga simuleringar är Dymolas standardlösare DASSL. För att utvärdera stabiliteten i denna kördes två simuleringar under samma tidsintervall, identiska på allt sånär som lösarens tolerans. Antalet itereringspunkter var i båda fallen 50 000. Varianser i detta ansågs överflödigt att testa då luftsystemet har en relativt långsam dynamik. Lösarens tolerans ändrades med en faktor tio, från 0.0001 till 0.001. Den relativa skillnaden i totaltrycket i insugsmanifolden från de båda simuleringarna redovisas i figur 19. Man kan se att skillnaden är liten.



Figur 19: Relativ skillnad i insugsmanifoldens totaltryck med olika toleranser på lösaren. Medelskillnad -0.015 %, maximal skillnad -4.392 %, standardavvikelse 0.319 %.

6 Analys

I avsnitt 6.1 nedan analyseras resultaten från simuleringarna av dieselmotormodellen, och avslutas med att behandla ett antal felkällor som kan tänkas påverka resultatens utgång. Avsnitt 6.2 behandlar sedan kort stabiliteten i simuleringen, medan i avsnitt 6.3 diskuteras slutligen kompetensen hos Modelica/Dymola som ett verktyg vid motormodellering.

6.1 Motormodell

6.1.1 Icke-minfasbeteenden

Man kan se icke-minfasbeteendet tydligt i de stegsvar som studerades (se figur 6 och 7). Detta tyder på en tillförlitlig modell som uppvisar verkliga beteenden. De variabler där effekterna syntes tydligast valdes att studera, det vill säga trycket i insugsmanifolden respektive massflödet genom turbinen. Effekterna av icke-minfasbeteendet går även att se på många andra variabler, men de blir då dämpade eftersom dynamiken är långsam mellan sammankopplade komponenter.

6.1.2 Volvo personbilmotor

Det är anmärkningsvärt hur väl de simulerade resultaten följer dynamiken hos den verkliga motorn. Däremot verkar modellen i allmänhet ge något för låga värden jämfört med de uppmätta, vilket både kan betraktas i figur 8 - 13 och verifieras av feluppskattningarna i tabell 1. I samtliga fall, utom för temperaturen efter turbinen, är medelfelet negativt och ligger storleksmässigt mellan ca -0.18 % och -13 %, vilket får anses vara ett godkänt resultat med tanke på alla de antaganden och förenklingar som har gjorts i utvecklingen av modellen.

Några av de maximala felen som utmärker sig kan i vissa fall bero på tillfälliga avvikelser. Tydligast är det maximala felet för massflödet genom kompressorn – vilket annars är den variabel som följer de uppmätta resultaten allra bäst – som vid ett tillfälle toppar vid drygt 174 %. Enligt figur 13 (b) uppkommer detta spontana fel strax efter 1 000 sekunder efter start. Genom att plotta de uppmätta värdena för hela körningen upptäcks snabbt att massflödet just vid den aktuella tidpunkten blir mycket litet, vilket enligt ekvation (24) kan ge ett mycket stort relativt fel. Det beräknade maximala felet är därför missvisande i ett sådant läge.

Varför de två modellerade temperaturerna behövde lågpasstreras beror som tidigare nämnt på den långsamma dynamiken hos temperaturgivarna. Det gör att snabba temperatur-toppar inte kan registreras av givarna med den noggrannhet som hade varit önskvärt. Modellen ger (utan lågpasstrering) i jämförelse en näst intill momentan beskrivning av temperaturerna. Även utan filtreringen följde dessa trenden hos de uppmätta värdena mycket väl. Det skulle därför vara möjligt att modellen i detta fall faktiskt ger en mer tillförlitlig skildring av de verkliga temperaturerna än vad de uppmätta värdena ger sken av.

Jämförelsen av Simulink-modellen mot den uppmätta körningen gav ett dynamiskt beteende som var mycket likt modellen i Dymola. Även feluppskattningarna får anses vara mycket likvärdiga. I en del fall gav den ena modellen en något mindre avvikelse från den uppmätta datan än den andra, och i övriga fall vice versa. Det går därför inte utifrån dessa jämförelser att avgöra om någon av modellerna, på det stora hela, följer den verkliga motorn på ett bättre sätt, vilket heller inte är relevant i sammanhanget. Det som däremot är möjligt att dra

en slutsats om är att två modeller, med samma grundläggande uppbyggnad⁵, ger likvärdiga resultat trots helt olika implementeringsverktyg. Det styrker Dymola som ett tillförlitligt verktyg för motormodellering.

Att den utökade modellen, med mass- och energikonservering hos samtliga volymer, inte nämnvärt påverkade resultatet måste sägas vara en överraskning. I vissa fall blev till exempel medelfelet något lägre, men i andra fall blev det istället något större. Det går därför inte att säga att den utökade modellen skulle vara en bättre beskrivning än den betydligt enklare varianten, åtminstone inte för så pass små volymer som används i modellen.

6.1.3 Volvo lastbilsmotor

Även när modellen har anpassats till en lastbilsmotor följs dynamiken förvånansvärt väl. Dessutom stämmer de absoluta värdena ofta någorlunda med de uppmätta. Faktum är att flera av medelfelen, i synnerhet de två trycken och turbinvarvtalet, är mindre än motsvarande fel för simulering av personbilsmotorn. Däremot indikerar de relativt höga standardavvikelserna i tabell 2 att resultatet inte riktigt är så tillförlitligt som de små medelfelen antyder. En mer noggrann kalibrering av modellparametrar hade troligtvis gett något bättre resultat.

Varför de simulerade trycken i insugs- och avgasmanifolden är tidsförskjutna gentemot de uppmätta värdena beror sannolikt på felsynkning i datahanteringen av dessa variabler, eftersom resterande värden inte uppvisar detta beteende.

Massflödet genom EGR-ventilen skapade problem vid feluppskattningen i början av simuleringen när massflödet var mycket litet (se figur 18 (a)). Därför var de absoluta felen en bättre uppskattning i detta fall. Vid den senare delen av simuleringen måste ändå resultatet vara helt acceptabelt, särskilt med tanke på att EGR-ventilen endast bygger på Bernoullis ekvation för inkompressibelt flöde. Genom att studera trycken i insugs- och avgasmanifolden syns dock att skillnaden mellan dem sällan är större än ca 0.2 bar, vilket gör att den enkla ekvationen (4) för massflödet trots allt får anses vara tillräcklig.

Den utökade modellen gav även i detta fall likvärdiga resultat utan några större avvikelser från den enklare beskrivningen, vilket är något oväntat eftersom massflödet genom EGR-ventilen borde påverka temperaturen i insugs- och avgasmanifolden. De enkla kontrollvolymererna med enbart masskonservering får därför anses vara fullt tillräckliga även i fallet med den större dieselmotorn.

Det bör nämnas att modellen inte fungerade när lastbilsmotorn arbetade vid mycket låga motor- och turbinvarvtal (på ca 450 rpm respektive 1 000 rpm). Detta beror sannolikt på att den interna modellen för turbin och kompressor är anpassad för en personbilsmotor, som normalt inte kan arbeta vid så låga varvtal. Vid simuleringen valdes därför ett intervall när motorn aldrig gick ned till dessa problematiska arbetspunkter. För att få en fullt fungerande modell för en lastbilsmotor, som beskriver hela dess arbetsregister, behövs därför en anpassad variant av turbomodellen som tar hänsyn till dessa situationer.

6.1.4 Potentiella felkällor

Modellen är kalibrerad mot Simulink-modellen, som trots många likheter skiljer sig på vissa punkter. Man kan därför argumentera att resultatet möjligtvis blivit något bättre om

⁵De viktigaste ekvationerna (för exempelvis turbin och kompressor) har de båda modellerna gemensamt, och de använder samma mappnings- och kalibreringstabeller.

kalibreringen gjordes mot Modelica-modellen. Det har dock ej varit möjligt under givna tidsramar. Vidare gjordes ingen omkalibrering innan den utökade modellen simulerades. Detta hade kanske förbättrat resultaten något. Speciellt för lastbilsmotorn där avgasåtercirkulering användes, då det hade lett till en något ändrad temperatur inuti kontrollvolymerna i stationärpunkterna som användes för kalibrering.

Ventilerna bygger i samtliga fall på Bernoullis ekvation. Den förutsätter inkompressibel och friktionsfri strömning, något som dock korrigeras för med en korrektionsfaktor C_D . Korrektionsfaktorn är dock en konstant som inte tar hänsyn till stora skillnader i tryckdifferens, varför ekvationen kan antas ge smärre fel vid vissa förhållanden. Bernoullis ekvation tar inte heller hänsyn till när flödet blir chokat, vilket gör den olämplig för justerbara ventiler i vilka höga gasflöden kan förekomma.

Modellen är endimensionell och bygger på medelvärden i kontrollvolymerna. Fluktuationer i temperatur, eller tryckstötar som tillkommer på grund av ventilernas öppning och stängning, tas därför inte i beaktning. Dessa tryckstötar påverkar flödet in i cylindrarna, vilket minskar noggrannheten i modellen. Den volymetriska verkningsgraden som används i simuleringarna tar dock hänsyn till flödesvariationer som härrör från dessa tryckstötar, varför felet som detta genererar tros vara litet.

Temperaturgivarna har en jämförelsevis långsam dynamik eftersom de innehåller en del egen massa som måste värmas upp eller kylas ned. Dessutom är mätningen av temperaturökningen i förbränningen, som sedan används som insignal till modellen, tagen i stationära punkter. Omgivande komponenter, såsom till exempel avgasmanifolden, håller vid dessa punkter en konstant temperatur. I dynamiska förlopp blir det i verkligheten även en viss dämpning av temperaturen på grund av ett värmeutbyte mellan gaserna och dessa komponenter. I stationära punkter visar de ett gott resultat men i snabba dynamiska förlopp blir utsignalen dämpad. För att resultatet från simuleringen ska stämma bättre lågpasfiltreras den modellerade utsignalen innan felet beräknas. Eftersom tidskonstanten för lågpasfiltreringen är approximerad är den inte riktigt tillförlitlig, varför de resulterande felen kan ha en viss avvikelse från det optimala fallet.

Värmeutbytet över ytorna i kontrollvolymerna är antaget att vara konstant noll. Det är dock möjligt att omgivningen kan kyla såväl som värma gaserna i volymerna beroende på omgivningens och motorns tillstånd, något som ej tagits i beaktning. Dessutom är ideala gaslagen en approximation som antas gälla med god noggrannhet. En mer exakt metod hade annars varit att använda sig av aggregationstabeller.

Modellen tillåter inte ett överskott av bränsle, eftersom det hade ökat komplexiteten mycket då oförbränt bränsle hade behövt införas som ett tillstånd i standardgränssnittet. Detta är inget problem för personbilsmotorn som alltid körs med ett stökiometriskt förhållande eller med bränsleunderskott. Lastbilsmotorn kör dock kortare perioder med ett överskott av bränsle, vilket cylindermodellen inte klarar av att hantera riktigt. Det gör att gasflödet inte hela tiden uppfyller villkoret för masskonservering vilket kan ge mindre fel i resultatet. I de simuleringsfall som körts handlar det dock inte om några stora fel.

I det verkliga fallet existerar en viss trycksänkning över laddluftskylaren som inte tas i beaktning. Dessutom är kylarens värmeutbyte med omgivningen i realiteten beroende av kylarens genomflödande av kyl Luft, med andra ord fordonets hastighet, vilket inte heller modelleras.

För att få modellen att fungera krävs kontrollvolymen innan och efter kompressorn samt turbinen. Deras volymen är inte kalibrerade utan väljs små för att inte kunna ackumulera mycket gasmassa, vilket skulle ge en långsammare tryck- och temperaturdynamik. Hur stora dessa volymen egentligen ska vara för att ge det bästa resultaten är inte klargjort.

6.2 Stabilitet i simuleringen

Simuleringarna för motorerna i resultatdelen är gjorda med lösarens tolerans satt till 0.0001. Med toleransen som 0.001 är den maximala skillnaden i insugsmanifoldens tryck -4.392 %, men medelskillnaden och standardavvikelsen är betydligt lägre än så (se figur 19). Satt i relation till de fel som ges med den högre noggrannheten relativt den verkliga motorn (se avsnitt 5.2), anses lösarens tillförlitlighet god.

6.3 Modelica och Dymola

Modelluppbyggnad

För arbetsgången i Dymola underlättar det mycket för förståelsen med de fem olika vyerna som finns tillgängliga (se avsnitt 2). Man får en grafisk överblick över hur modellen ser ut, samtidigt som man enkelt kan gå in i en delkomponent och se koduppbyggnaden av den. Koduppbyggnaden för en komponent känns för övrigt logisk och det är väldigt intuitivt att skapa en komponent från grunden.

Under programmeringen går det relativt snabbt att skapa en modell av en komponent så länge man vet vilka, och hur många, ekvationer som behövs. Man behöver heller inte hålla reda på vilka variabler som är okända så länge som antalet ekvationer och obekanta stämmer överens, vilket är en klar fördel med Modelicas deklarativa funktionshantering. Däremot kan det deklarativa arbetssättet i en del fall medföra en viss svårighet att hålla koll på vilka variabler som kan betraktas som kända respektive okända. Det gäller i synnerhet för komponenter som beskrivs av ett stort antal komplexa ekvationer, men detta får anses vara en petitesse i jämförelse med de många fördelar som det innebär. Det kan nämnas att möjligheten finns att precisera ordningsföljden för ekvationerna, något som i vissa fall kan ge en bättre överskådlighet.

Objektorienteringen får också anses vara en av de största fördelarna med Modelica. Den möjliggör exempelvis att flera komponenter av samma typ enkelt kan kopplas samman efter varandra, vilket enligt [10] annars är en svårighet i andra programvaror såsom Simulink. Det gäller dock att definiera aktuella startvärden för de sammankopplade komponenterna på ett lämpligt sätt för att få detta att fungera optimalt.

En annan styrka med objektorienteringen är när ändringar av en viss komponent behöver utföras; då ändras nämligen alla instanser av denna komponent automatiskt. Det gäller att också ha detta i åtanke ifall endast justeringen ska göras hos en särskild instans av komponenten och inte hos de övriga. I ett sådant fall skapas enkelt en kopia av komponenten, i vilken den aktuella justeringen görs.

Det största problemet som stöttes på under modellbyggnaden var hanteringen av algebraiska loopar. Enligt [6] ska programmet klara av att hantera dessa svårigheter i allmänhet, men tyvärr inte i detta fall när ekvationslösaren skulle behandla en loop i den interna företags-hemliga modellen av typen:

$$f(g(f, a), b) \tag{25}$$

där f och g är funktioner på så sätt att f beror av sig själv (och andra faktorer). Varför detta inte kunde lösas kvarstår som okänt. Problemet löstes dock till slut genom att införa ett lågpasfilter med en mycket kort tidskonstant, vilket är en vanlig metod för att hantera algebraiska loopar i andra simuleringsprogram.

Dymolas grafiska metod för inläsning av värden från tabeller är förhållandevis smidig och grundas på att linjer dras mellan tabeller och komponenter. Detta kan dock bli något rörigt när många tabeller behöver användas samtidigt, vilket uppdagades framförallt när all körningsdata skulle implementeras som insignal till ett flertal komponenter. Figur 53 i appendix C visar hur detta kan gå till rent illustrativt. Eventuellt hade detta kunnat lösas på ett smidigare sätt genom att kanske bortse från den grafiska biten och istället beskriva anslutningen mellan en tabell och en komponent direkt i koden. Det finns även andra, mer avancerade anslutningsalternativ, som dock inte har studerats i projektet.

Att anpassa modellen från en personbils- till en lastbilsmotor krävde endast några mindre justeringar i programmet. Det handlade framförallt om att ändra vissa motorspecifika parametrar, som till exempel antal cylindrar och motorvolym, och att byta ut strypventilen vid insuget mot en variant med justerbar area. Detta får anses vara oerhört smidigt, särskilt i jämförelse med andra simuleringsverktyg som saknar objektorientering, då även en mindre ändring kan medföra stora anpassningar av övriga delar i modellen.

Information om hur Modelica och Dymola fungerar och hur man ska modellera finns mycket litteratur om. De engelska manualerna som finns tillgängliga⁶ är ofta enkla att förstå och är överskådliga. Något som däremot saknas är en inbyggd hjälpmニュアル i programmet, i stil med den som finns i Matlab. En sådan hade underlättat genom att snabbt kunna få grundläggande information, vilket hade gynnat tiden för modelluppbyggnaden. Information om de färdiga biblioteken, och de komponenter som finns där, innehåller däremot tydliga förklaringar som kan vara till stor nytta. Det kan också nämnas att det är möjligt att bygga upp ett eget komponentbibliotek, med tillhörande information, av de komponenter man har skapat.

Simulering

När modellen är färdigutvecklad och ska simuleras, går man via en ikon direkt över till simuleringsdelen. Vyn för simuleringen känns väldigt intuitiv och lätt att arbeta i. Via en lista kan man kryssa i exakt vilka variabler man vill studera och kan i en plot bredvid zooma, markera värden, studera utseendet med mera. Listan med alla variabler kan bli väldigt omfattande vid en större modell och det kan bli svårt att hitta i den, något som hade kunnat göras mer överskådligt. Man kan dock välja att inte visa till exempel konstanter och variabler som inte varierar något under simuleringen, vilket ger en något bättre överblick. Dessutom ska det gå att skriva ett script i koden för att direkt kunna välja hur och vilka variabler man vill plotta, utan att manuellt behöva bocka för dessa i efterhand. Detta har däremot inte undersökts närmare under projektets gång.

En del startvärden som krävdes i modellen var inte alltid helt uppenbara att bestämma. Det löstes ofta genom en simulering av modellen i en rimlig arbetspunkt, antecknade de stationära värdena och använde sedan dessa som startvärden. Något som hade underlättat, är att kunna samla alla startvärden för alla komponenter på ett och samma ställe. I nuläget

⁶Det finns till exempel en omfattande manual i två delar som ges ut av Dassault Systèmes AB [6] och kan erhållas från Dymolas hjälpmeny.

får man gå in i respektive komponent och ändra startvärdena, men detta skulle eventuellt kunna hanteras något smidigare genom ett script.

Den största nackdelen som ses med programvaran är svårtydda felkoder. När en felkod uppkommer kan den peka på ett ställe där den först hittade ett problem, men det egentliga felet kan ligga på något helt annat ställe i modellen. Detta skapar förvirring och arbetet med att hitta fel kan bli tidsödande. Självklart blir det lättare och går snabbare att tyda felkoderna med mer erfarenhet av programvaran, men det får ändå anses vara ett betydande problem som kan skapa stor frustration.

Efter simuleringen kan relevant data sparas undan för vidare analys. Det finns flera olika tillvägagångssätt beroende på vilken form man vill ha utdata på, och överlag är detta inget problem. Dymola har en bra kompatibilitet ihop med framförallt Matlab, till exempel kan datan från simuleringen lätt göras om till en överskådlig .mat-fil.

Framtidsutsikter

Med de erfarenheter som har erhållits under projektets gång finns inga egentliga hinder att också modellera system av helt andra karaktärer. En försmak på detta är den turbo som användes i dieselmotormodellen, där två olika gränssnitt gjorde det möjligt att hantera både ett luftflöde och en mekanisk axel simultant. Tack vare användningen av lämpliga gränssnitt kan exempelvis ett reglertekniskt styrsystem enkelt interagera med andra, helt skilda komponenter. Ett exempel på detta är ett examensarbete [16] vid Uppsala Universitet som nyligen visade på att Modelica/Dymola med gott resultat kan användas för att styra en kvarnkrets för finfördelning av malm inom gruvindustrin.

Det finns många påtagliga fördelar med Modelica/Dymola som nämnts ovan. I synnerhet skapar objektorienteringen och den deklarativa hanteringen av ekvationer ett logiskt arbetsflöde vid modelluppbbyggnaden. Men som med allting finns det också vissa baksidor som skulle behöva åtgärdas eller förbättras. Det handlar framförallt om att felsökningen behöver göras tydligare, och att skapa en mer överskådlig struktur för variablerna vid simuleringen. Överlag bedöms dock Modelica/Dymola vara ett fullgott modelleringsverktyg för motorers luftsystem och kan mycket väl få en ökad ställning inom industrin i framtiden.

7 Slutsats

Syftet med projektet var att utvärdera modelleringsspråket Modelica, i simuleringsmiljön Dymola, som ett verktyg för motormodellering. En modell av en dieselmotors luftsystem skapades, där varje komponent konstruerades utifrån sina specifika ekvationer och koplades samman med lämpliga, väldefinierade gränssnitt. Två viktiga komponenter var en EGR-ventil för avgasåterföring och en turbin med variabel geometri (VGT). Modellens tillförlitlighet studerades sedan genom jämförelser mot uppmätta värden från en körning med såväl en personbils- som en lastbilsmotor. Dessutom undersöktes om modellen uppvisar icke-minfasbeteenden, vilket är vanligt hos en verklig motor med EGR och VGT. Användarvänligheten och möjligheterna med Modelica/Dymola kunde därefter utvärderas utifrån de erfarenheter som uppnåts under projektets gång.

Dieselmotormodellen grundades till stor del på fysikaliska ekvationer, som till exempel ideala gaslagen, men också på empiriska samband med optimeringsparametrar hämtade ur mappningstabeller. Till stor del användes samma eller liknande samband som för en befintlig motormodell implementerad i Simulink, vilket gjorde det möjligt att också jämföra hur väl modellen i Modelica/Dymola står sig mot denna.

Modellen uppvisade icke-minfasbeteenden vid stegändring i arean för såväl EGR som VGT, vilket är ett tecken på att modellen kan anses vara tillförlitlig. Jämförelsen mot personbilsmotorn gav ett resultat som följer dynamiken förvånansvärt väl och som generellt sett har små fel. Detta var även mycket likt resultatet vid motsvarande simulering av Simulink-modellen, något som ytterligare styrker modellens trovärdighet. Även dynamiken vid jämförelsen mot lastbilsmotorn får anses följa de uppmätta värdena på ett tillfredsställande sätt. Däremot uppvisar modellen relativt stora avvikelser, och vid mycket låga varvtal fungerar den inte att simulera överhuvudtaget. Modellen behöver därför anpassas vidare för att vara pålitlig vid motormodellering av en tung lastbilsdiesel. Den utökade, mer förfinade varianten för en kontrollvolym, som användes som en extra verifikation, påverkade i samtliga fall resultatet marginellt.

Det ska dock påpekas att bedömningen huruvida resultaten är tillräckligt goda eller inte är mer eller mindre subjektiv. Det är inte helt trivialt att kvantifiera hur ”bra” modellen beskriver en verklig dieselmotor utifrån de simuleringar som har gjorts, förutom om dess feluppskattningar faller inom ett givet toleransområde eller ej. De begränsningar som har satts, exempelvis att modellen bortser från tredimensionella effekter såsom tryckstötter, ger en övre gräns på den noggrannhet som kan uppnås. Det hela handlar om en balansgång mellan hur stora avvikelser som tillåts i förhållande till komplexiteten hos modellbeskrivningen.

En intressant vidareutveckling av motormodellen skulle kunna vara att gå bortom luftsystemet och modellera andra vitala delar av ett fordon, som till exempel en last, en förare och ett motorstyrsystem, för att på så sätt simulera en mer realistisk körcykel. Här kan objektorienteringen hos Modelica underlätta när system av olika fysikaliska domäner ska kopplas samman. En ytterligare möjlighet är att modellera förbränningen i cylindrarna så att även utsläpp av diverse emissioner tas hänsyn till, något som i synnerhet är av stor vikt för undersökning av motorns miljöpåverkan.

Att utveckla och simulera modellen i Dymola var överlag både intuitivt och logiskt, mycket tack vare Modelicas objektorientering och deklarativa funktionshantering. Gränssnitten, som förbinder komponenterna med varandra, är ett viktigt nyckelkoncept i modelleringsspråkets

väl fungerande integration mellan delsystem av helt olika fysikaliska domäner. Valfriheten i att välja vilka storheter som gränssnitten ska överföra ger en bra förutsättning att även modellera andra tekniskt intressanta system. Därutöver medför Dymolas pålagda grafiska programmeringsmiljö att inlärningströskeln för programvaran är relativt låg. Däremot skulle själva simuleringsmiljön, och framförallt felsökningen, behöva en viss förbättring för att göra verifieringen av en modell mer effektiv. På det hela taget kan dock Modelica/Dymola starkt rekommenderas för fortsatt användning vid motormodellering – men också inom andra tekniska områden där ett behov finns för en mer modern, effektiv och hållbar arbetsprocess.

Referenser

- [1] Wahlström, J., Eriksson, L., Klein, M. (2010) *Chapter 4: Physical Modeling of Turbocharged Engines and Parameter Identification*. Department of Electrical Engineering, Linköping, Sweden.
- [2] Lennartson, B. (2002). *Reglerteknikens grunder*. Lund: Studentlitteratur AB.
- [3] Simulink – Simulation and Model-Based Design. *MathWorks*.
<http://www.mathworks.se/products/simulink/>. (9 apr. 2012).
- [4] Modelica and the Modelica Association.
<https://modelica.org/>. (30 mar. 2012).
- [5] Dymola: Overview. *Dassault Systemes*.
<http://www.3ds.com/products/catia/portfolio/dymola>. (10 maj 2012).
- [6] Dymola Product Line: Key advantages. *Dassault Systemes*.
<http://www.3ds.com/products/catia/portfolio/dymola/dymola-product-line/key-advantages/>. (9 apr. 2012).
- [7] Dymola Model Libraries. *Modelon*.
<http://www.modelon.com/products/dymola/dymola-model-libraries/>. (9 apr. 2012).
- [8] Heywood, J.B. (1988). *Internal Combustion Engine Fundamentals*. McGraw-Hill, Inc.
- [9] Wahlström, J., Eriksson, L. (2009). *Modeling of a Diesel Engine with VGT and EGR capturing Sign Reversal and Non-minimum Phase Behaviors*. Report: LiTH-ISY-R-2882, Department of Electrical Engineering, Linköping University, Linköping, Sweden.
- [10] Markus Grahn, institutionen för Tillämpad mekanik, Chalmers Tekniska Högskola. Telefon: +46 31 772 13 95.
- [11] Moraal, P., Kolmanovsky, I. (1999). *Turbocharger Modeling for Automotive Control Applications*. SAE International. Detroit, Michigan, USA.
- [12] Mappningstabeller och motorspecifika parametrar (erhållna från Markus Grahn) för en Volvo personbilmotor. För visualisering se Appendix B.
- [13] Uppmätta värden (erhållna från Markus Grahn) från en körning med en Volvo personbilmotor. För visualisering se Appendix B.
- [14] Mappningstabeller och motorspecifika parametrar (erhållna från Volvo AB) för en Volvo lastbilmotor.
- [15] Uppmätta värden (erhållna från Volvo AB) från en körning med en Volvo lastbilmotor.
- [16] Persson, S. (2009). *Kvarnkretssimulering för test av styrsystem*. Examensarbete vid Uppsala Universitet.

A Stökiometrisk förbränning

Antag att luft består till antalet partiklar av 79 % N_2 och 21 % O_2 , det vill säga att andelen av andra gaser som till exempel Ar, CO_2 och H_2O anses vara försumbara. Definiera en enhet luft som

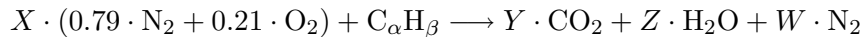
$$\text{air} = 0.79 \cdot N_2 + 0.21 \cdot O_2$$

Definiera också en enhet bränsle som

$$\text{fuel} = C_\alpha H_\beta$$

där α och β anger antalet C respektive H per enhet. Den dieselolja som används som for-donsbränsle innehåller normalt sett en blandning av kolväten mellan $C_{10}H_{20}$ och $C_{15}H_{28}$, men i detta projekt används $C_{12}H_{23}$ som ett medelvärde⁷.

Förbränning av en enhet bränsle i luft ger sambandet



Genom att balansera antalet C i båda led fås att $Y = \alpha$, och balans av antalet H ger på samma sätt att $Z = \frac{\beta}{2}$. Balansering av O ger

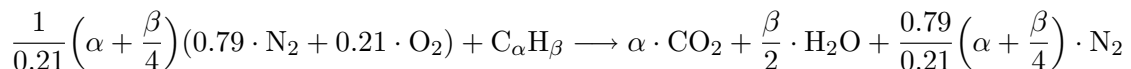
$$0.21 \cdot 2 \cdot X = 2 \cdot Y + Z = 2 \cdot \alpha + \frac{\beta}{2}$$

$$\implies X = \frac{1}{0.21} \left(\alpha + \frac{\beta}{4} \right)$$

Slutligen ger balansering av antalet N att

$$0.79 \cdot 2 \cdot X = 2 \cdot W \implies W = 0.79 \cdot X = \frac{0.79}{0.21} \left(\alpha + \frac{\beta}{4} \right)$$

Den slutliga reaktionsformeln ges därför av



och kan skrivas på en komprimerad form som

$$\frac{1}{0.21} \left(\alpha + \frac{\beta}{4} \right) \cdot \text{air} + \text{fuel} \longrightarrow \text{exh} \quad (26)$$

där "air", "fuel" och "exh" syftar på respektive enhet. Definiera utifrån reaktionsformeln en enhet avgaser som

$$\text{exh} = \alpha \cdot CO_2 + \frac{\beta}{2} \cdot H_2O + \frac{0.79}{0.21} \left(\alpha + \frac{\beta}{4} \right) \cdot N_2$$

Beteckna antalet molekyler i en avgasenhets som

$$n_{\text{exh}} = \alpha + \frac{\beta}{2} + \frac{0.79}{0.21} \left(\alpha + \frac{\beta}{4} \right) = \frac{1}{0.21} \cdot \alpha + \frac{1.21}{0.84} \cdot \beta$$

Med $\alpha = 12$ och $\beta = 23$ blir

$$n_{\text{exh}} \approx 90.27$$

⁷Enligt Heywood [8] stämmer detta bra för lätt dieselolja.

På samma sätt ges antalet molekyler i en enhet luft av

$$n_{\text{air}} = 0.79 + 0.21 = 1$$

Även en bränsleenhet innehåller endast en molekyl, det vill säga

$$n_{\text{fuel}} = 1$$

Med hjälp av molmassorna⁸ för de ingående atomerna och molekylerna kan de ekvivalenta molmassorna för luft, bränsle och avgaser beräknas var för sig utifrån sambandet $M = \frac{m}{n}$, där m är massan för en enhet och n är antalet molekyler i den aktuella enheten.

Molmassan för definitionen av luft ges därmed av

$$M_{\text{air}} = \frac{m_{\text{air}}}{n_{\text{air}}} = \frac{0.79 \cdot M_{\text{N}_2} + 0.21 \cdot M_{\text{O}_2}}{1} = 0.79 \cdot 28.01 + 0.21 \cdot 32.00 \approx \mathbf{28.85 \text{ g/mol}}$$

Med $\alpha = 12$ och $\beta = 23$ fås molmassan för bränslet på samma sätt enligt

$$M_{\text{fuel}} = \frac{m_{\text{fuel}}}{n_{\text{fuel}}} = \frac{12 \cdot M_{\text{C}} + 23 \cdot M_{\text{H}}}{1} = 12 \cdot 12.01 + 23 \cdot 1.01 = \mathbf{167.35 \text{ g/mol}}$$

Molmassan för definitionen av avgaser ges av

$$M_{\text{exh}} = \frac{m_{\text{exh}}}{n_{\text{exh}}} = \frac{\alpha \cdot M_{\text{CO}_2} + \frac{\beta}{2} \cdot M_{\text{H}_2\text{O}} + \frac{0.79}{0.21} \left(\alpha + \frac{\beta}{4} \right) \cdot M_{\text{N}_2}}{\frac{1}{0.21} \cdot \alpha + \frac{1.21}{0.84} \cdot \beta}$$

Med de numeriska värdena för α och β blir slutligen

$$M_{\text{exh}} = \frac{12 \cdot 44.01 + \frac{23}{2} \cdot 18.02 + \frac{0.79}{0.21} \left(12 + \frac{23}{4} \right) \cdot 28.01}{90.27} \approx \mathbf{28.86 \text{ g/mol}}$$

I modellen är ofta den specifika gaskonstanten för luft och avgaser av intresse. Med den allmänna gaskonstanten $R = 8,314462 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ ges den specifika gaskonstanten för luft av uttrycket

$$R_{\text{air}} = \frac{R}{M_{\text{air}}} = \frac{8,314462 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}}{28.85 \text{ g} \cdot \text{mol}^{-1}} \approx 0.2882 \text{ J} \cdot \text{g}^{-1} \cdot \text{K}^{-1} = \mathbf{288.2 \text{ J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}}$$

och för avgaser av

$$R_{\text{exh}} = \frac{R}{M_{\text{exh}}} = \frac{8,314462 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}}{28.86 \text{ g} \cdot \text{mol}^{-1}} \approx 0.2881 \text{ J} \cdot \text{g}^{-1} \cdot \text{K}^{-1} = \mathbf{288.1 \text{ J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}}$$

Det faktum att R_{air} och R_{exh} är i princip identiska underlättar situationer då en gasblandning innehåller såväl luft som avgaser, då den specifika gaskonstanten för dessa kan anses vara densamma.

En sista storhet som är viktig för förbränningen är *AFR*-värdet (air-fuel ratio) som anger

⁸ $M_{\text{N}_2} = 28.01 \text{ g/mol}$, $M_{\text{O}_2} = 32.00 \text{ g/mol}$, $M_{\text{C}} = 12.01 \text{ g/mol}$, $M_{\text{H}} = 1.01 \text{ g/mol}$, $M_{\text{CO}_2} = 44.01 \text{ g/mol}$, $M_{\text{H}_2\text{O}} = 18.02 \text{ g/mol}$.

massförhållandet mellan luft och bränsle vid stökiometrisk förbränning. Denna storhet kan beräknas ur ekvation (26). För varje bränsleenhet krävs $\frac{1}{0.21} \left(\alpha + \frac{\beta}{4} \right)$ enheter luft. Låt $\frac{m_{\text{fuel}}}{M_{\text{fuel}}}$ och $\frac{m_{\text{air}}}{M_{\text{air}}}$ beteckna antalet enheter av bränsle respektive luft. Det sökta AFR -värdet ges därför av

$$AFR = \frac{m_{\text{air}}}{m_{\text{fuel}}} = \frac{1}{0.21} \left(\alpha + \frac{\beta}{4} \right) \cdot \frac{M_{\text{air}}}{M_{\text{fuel}}}$$

som med de numeriska värdena ger

$$AFR = \frac{1}{0.21} \left(12 + \frac{23}{4} \right) \cdot \frac{28.85}{167.35} \approx \mathbf{14.57}$$

B Motor- och körningsdata

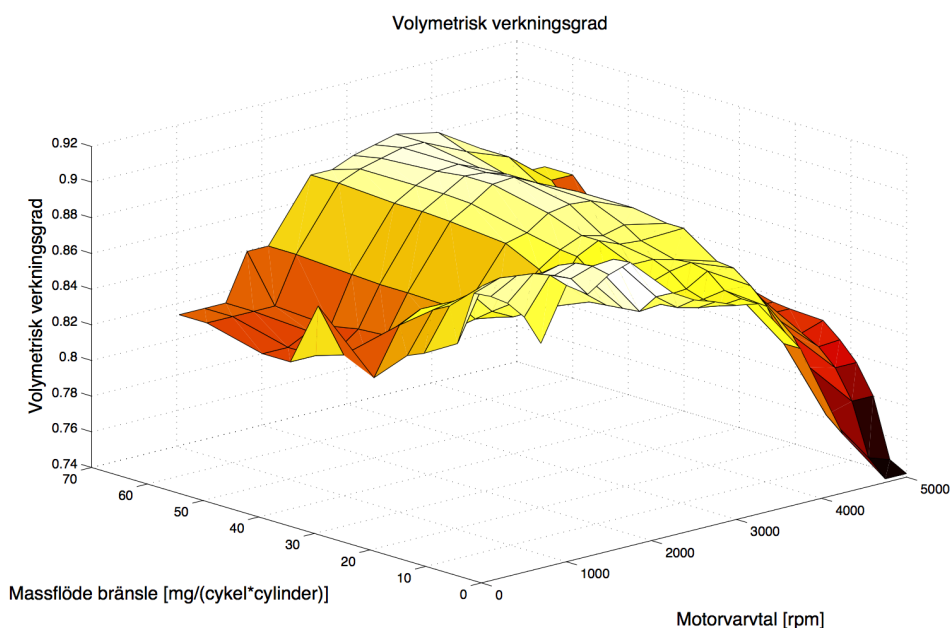
Volvo personbilmotor

Motorspecifik data

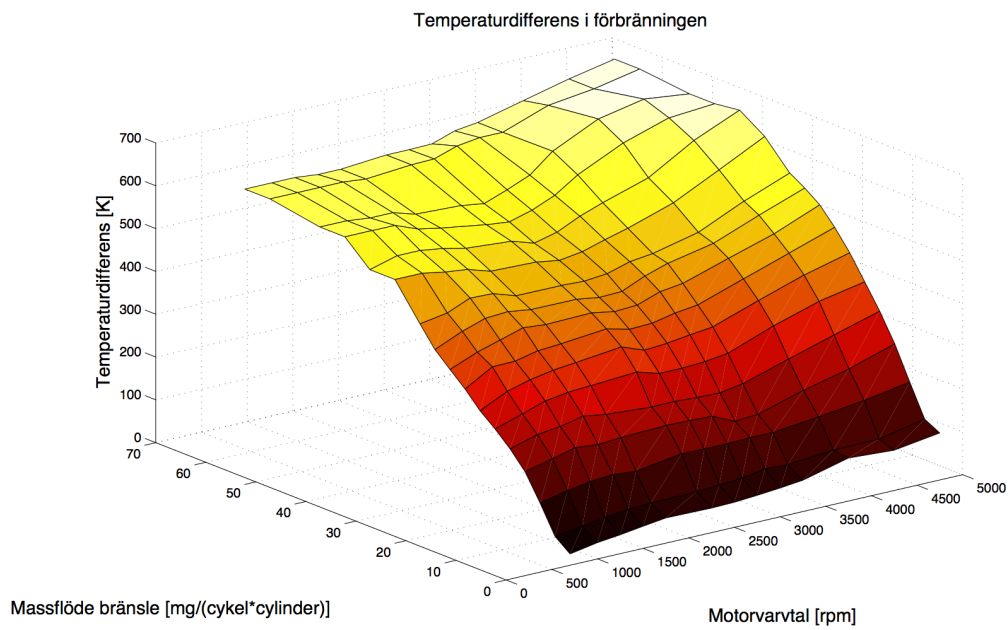
Nedan följer motorspecifika parametrar och mappningstabeller [12] som är unika för motorn som modellen jämförs mot. Några av dessa har beräknats fram genom kalibrering av modellen utifrån uppmätt data från en verklig körning [13].

Tabell 3: Motorspecifika parametrar för personbilmotorn.

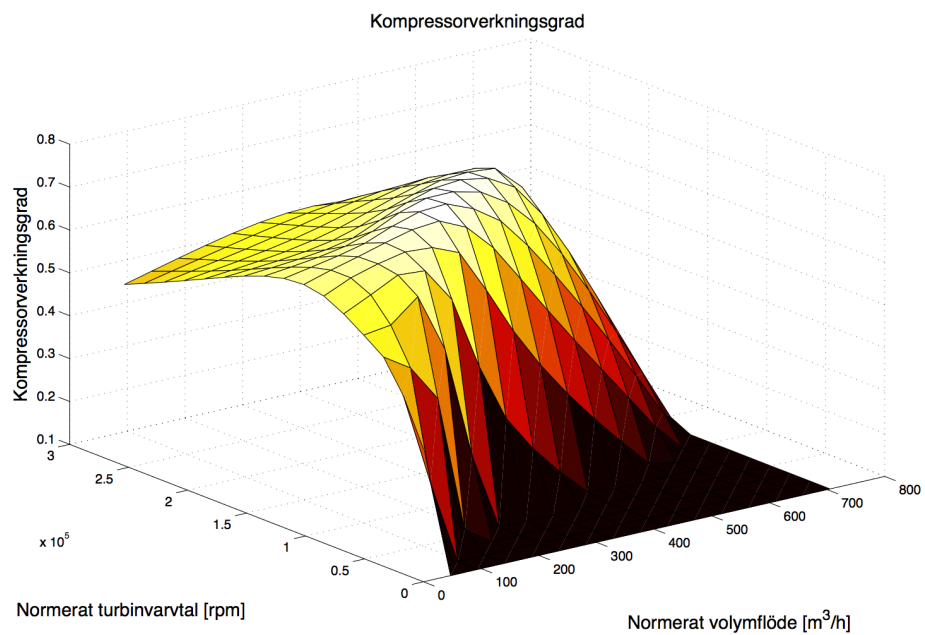
Parameter	Kommentar	Värde	Enhet
V_d	Motorvolym	2.4	liter
V_{im}	Effektiv volym, insugsmanifold	5	liter
V_{em}	Effektiv volym, avgasmanifold	5	liter
$A_{air\ filter}$	Effektiv area för luftfilter	18	cm ²
$A_{exhaust\ pipe}$	Effektiv area för avgassystem	8.3	cm ²
$A_{intercooler}$	Effektiv area för intercooler	16.4124	cm ²
$\eta_{intercooler}$	Kylverkningsgrad intercooler	0.7	–
$\eta_{egr\ cooler}$	Kylverkningsgrad EGR-kylare	0.7	–
J	Turbinaxelns tröghetsmoment	$3.000 \cdot 10^{-5}$	kg·m ²
η_{shaft}	Turbinaxelns verkningsgrad	0.95	–



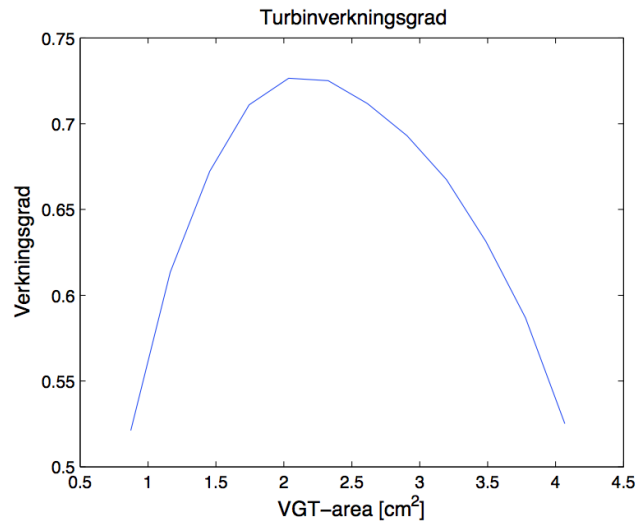
Figur 20: Cylindrarnas volymetriska verkningsgrad. Verkningsgraden är beroende av motorns varvtal samt tillförd bränslemängd.



Figur 21: Temperaturskillnad orsakad av förbränningen. Temperaturökningen är beroende av motorns varvtal samt tillförd bränslemängd.



Figur 22: Kompressorns isentropiska verkningsgrad som funktion av turbinaxelns normerade varvtal samt normerat volymflöde genom kompressorn. Normeringen sker med avseende på temperaturen, vilket gör att mappningstabellen inte behöver ta hänsyn till de yttre omständigheterna.



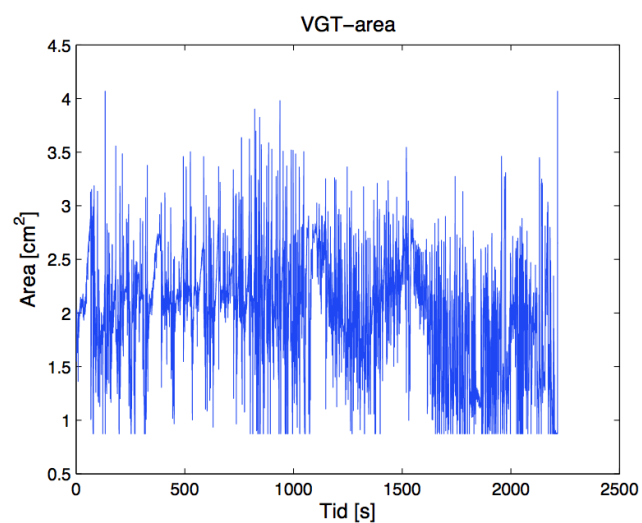
Figur 23: Turbinens isentropiska verkningsgrad som funktion av turbinens effektiva inloppsarea.

Körningsspecifik data

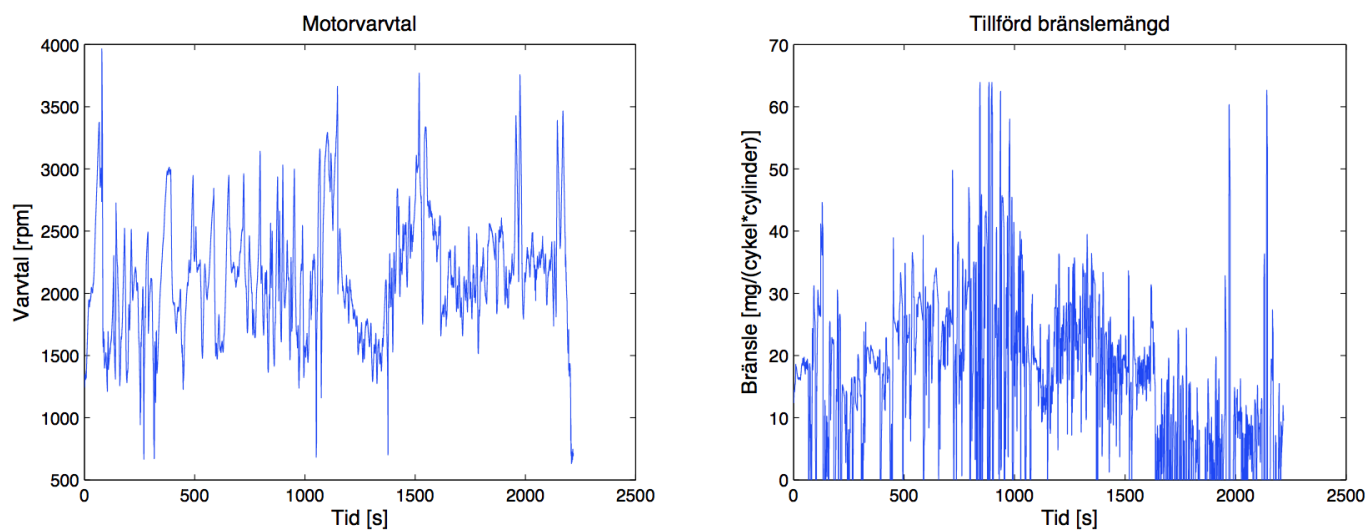
Nedan följer uppmätt data från personbilsmotorn och dess omgivning som används som insignaler under en simulering. Dessa värden mättes upp under en 37 minuter lång körning i Alperna [13].

Tabell 4: Givna parametrar för körning med personbilsmotorn.

Parameter	Kommentar	Värde	Enhet
$c_{p,\text{air}}$	Specifik värmekapacitet för luft	1004.5	J/(kg·K)
$c_{p,\text{exh}}$	Specifik värmekapacitet för avgaser	1183.875	J/(kg·K)
γ_{air}	Värmekapacitetskvot för luft	1.4	—
γ_{exh}	Värmekapacitetskvot för avgaser	1.32	—



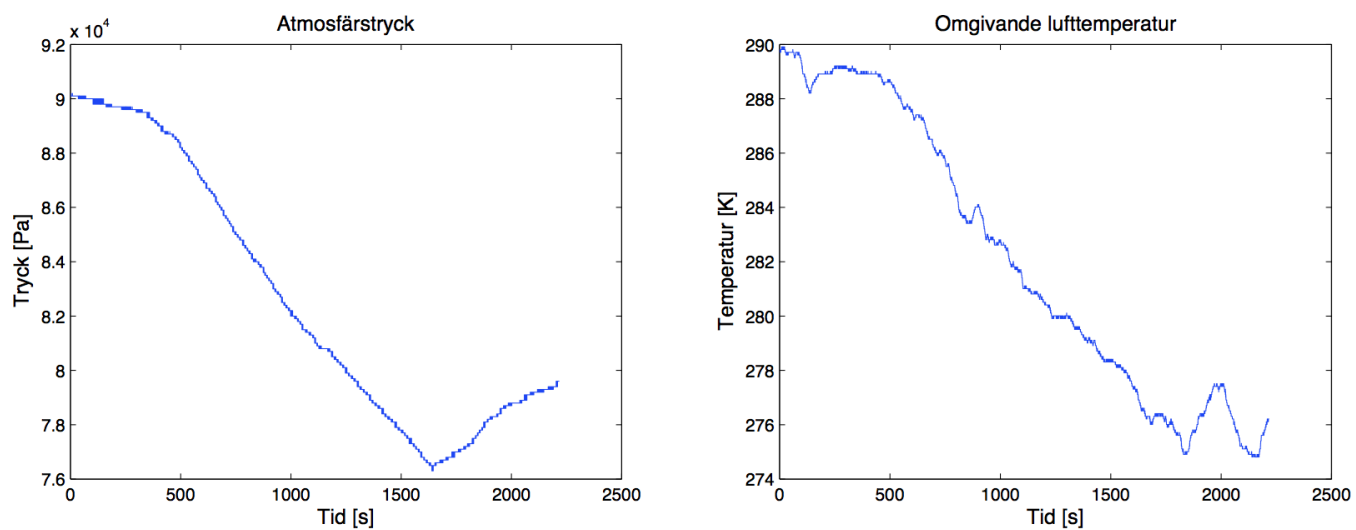
Figur 24: VGT-arean under hela körningens tidsintervall.



(a) Motorns varvtal under hela körningens tidsintervall.

(b) Tillförd bränslemängd under hela körningens tidsintervall.

Figur 25



(a) Uppmätt atmosfärstryck under hela körningens tidsintervall.

(b) Uppmätt atmosfärstemperatur under hela körningens tidsintervall.

Figur 26

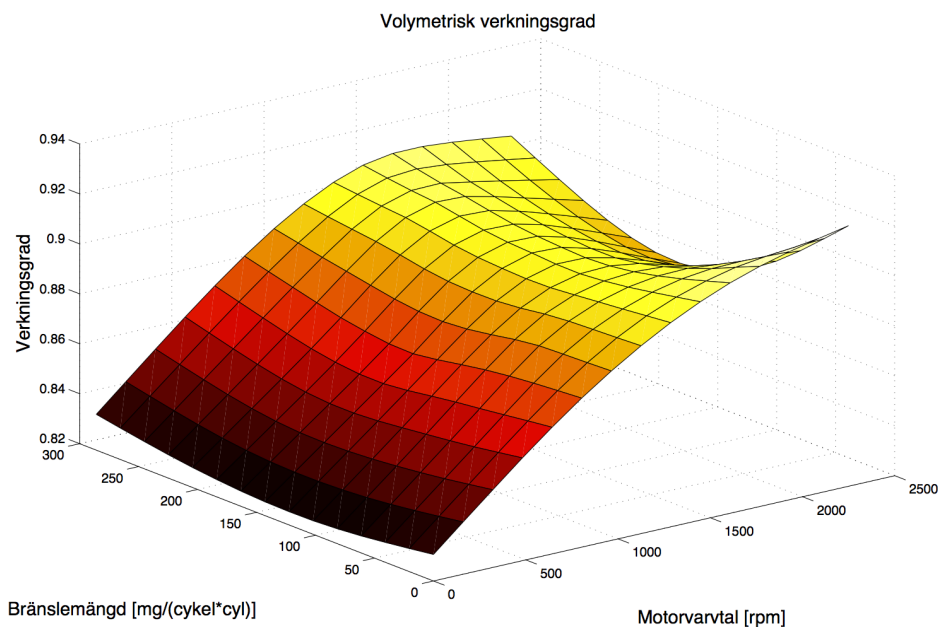
Volvo lastbilsmotor

Motorspecifik data

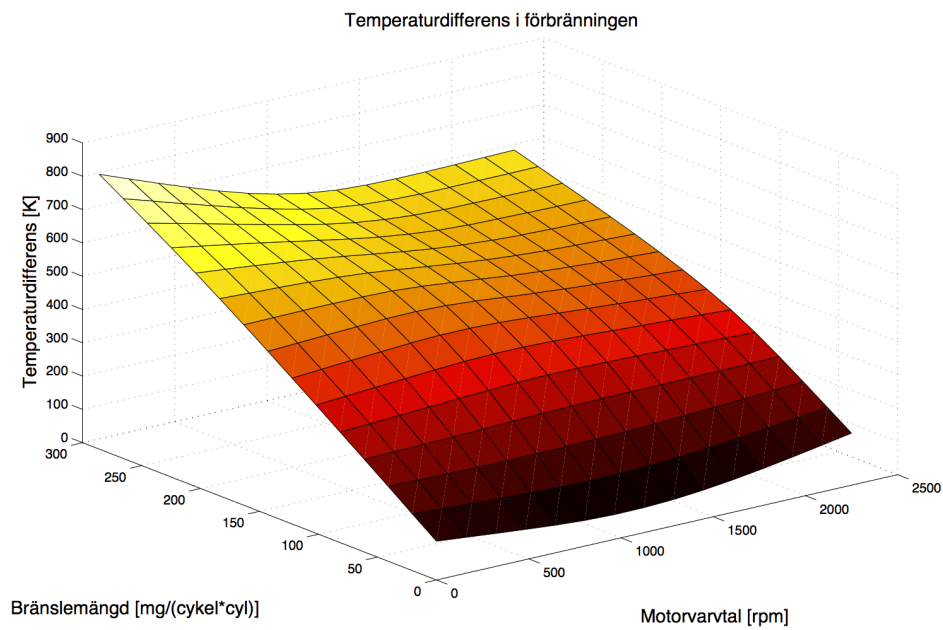
Nedan följer motorspecifika parametrar och mappningstabeller [14] som är unika för motorn som modellen jämförs mot. Några av dessa har beräknats fram genom kalibrering av modellen utifrån uppmätt data från en verklig körning [15].

Tabell 5: Givna motorspecifika parametrar för lastbilsmotorn.

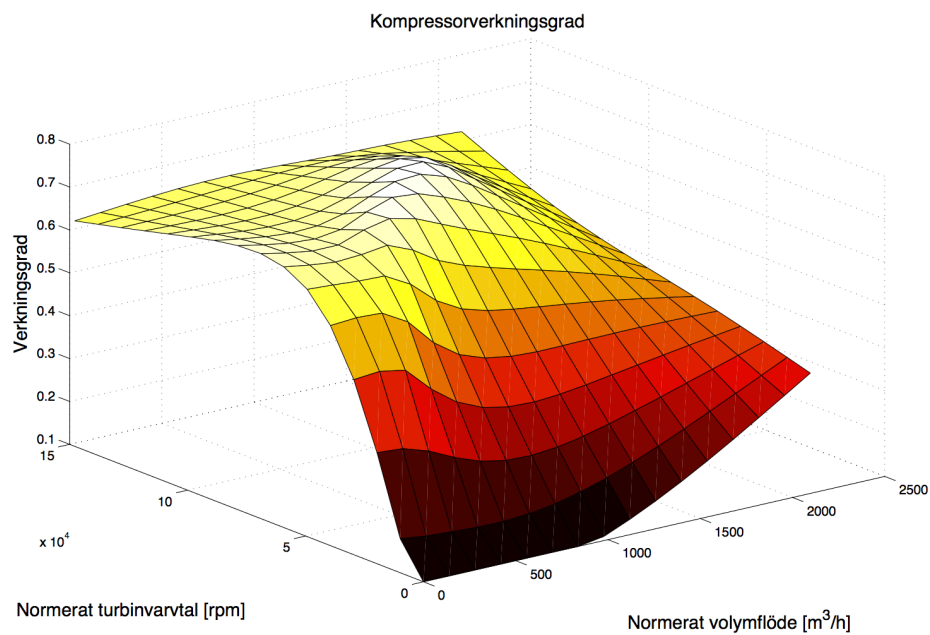
Parameter	Kommentar	Värde	Enhet
V_d	Motorvolym	12.8	liter
V_{im}	Effektiv volym, insugsmanifold	7.2	liter
V_{em}	Effektiv volym, avgasmanifold	3.1	liter
$A_{air\ filter}$	Effektiv area för luftfilter	54	cm ²
$A_{exhaust\ pipe}$	Effektiv area för avgassystem	20	cm ²
$A_{intercooler}$	Effektiv area för intercooler	20	cm ²
$\eta_{intercooler}$	Kylverkningsgrad intercooler	0.7765	–
$\eta_{egr\ cooler}$	Kylverkningsgrad EGR-kylare	0.7	–
J	Turbinaxelns tröghetsmoment	$5.000 \cdot 10^{-4}$	kg·m ²
η_{shaft}	Turbinaxelns verkningsgrad	0.95	–



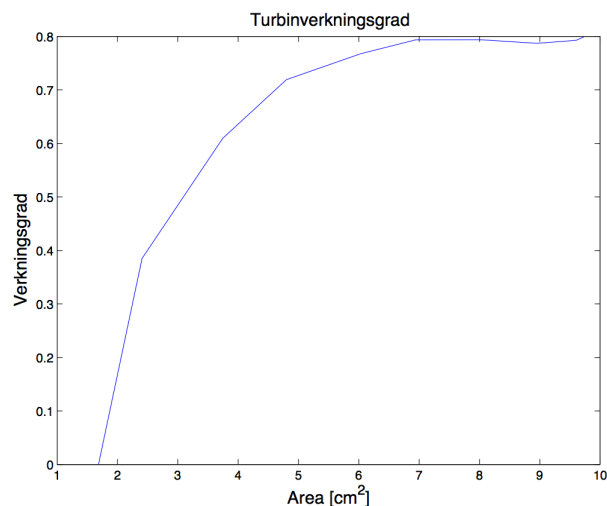
Figur 27: Cylindrarnas volymetriska verkningsgrad. Verkningsgraden är beroende av motorns varvtal samt tillförd bränslemängd.



Figur 28: Temperaturskillnad orsakad av förbränningen. Temperaturökningen är beroende av motorns varvtal samt tillförd bränslemängd.



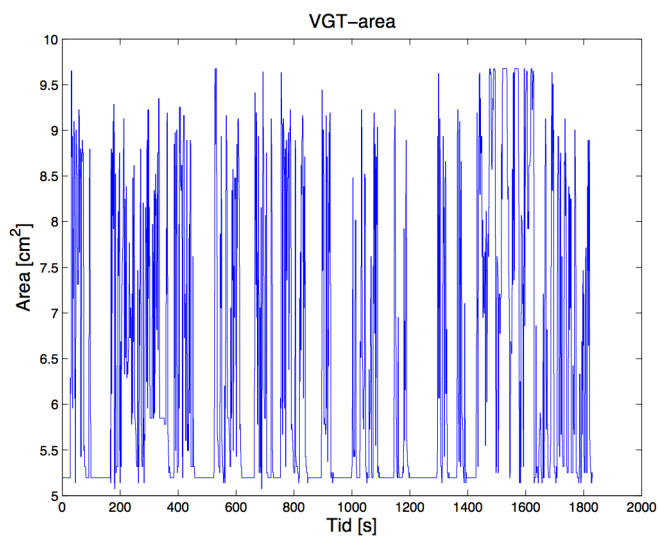
Figur 29: Kompressorns isentropiska verkningsgrad som funktion av turbinaxelns normerade varvtal samt normerat volymflöde genom kompressorn. Normeringen sker med avseende på temperaturen, vilket gör att mappningstabellen inte behöver ta hänsyn till de yttre omständigheterna.



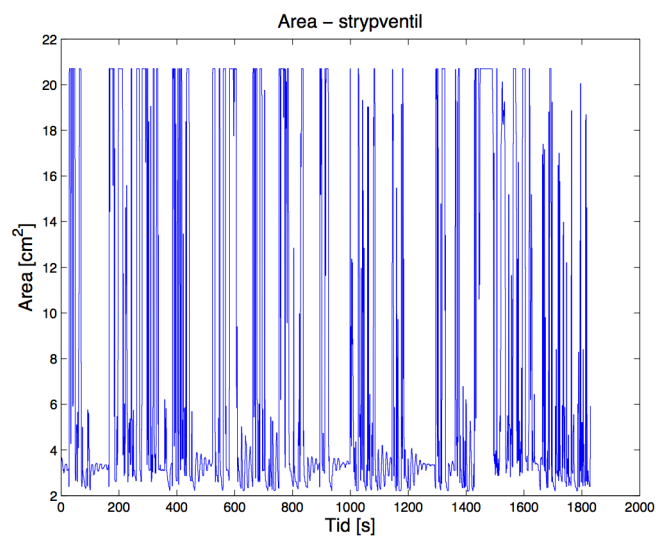
Figur 30: Turbinens isentropiska verkningsgrad som funktion av turbinens effektiva inloppsarea.

Körningsspecifik data

Nedan följer uppmätt data för lastbilsmotorn och dess omgivning som används som insignaler under en simulering. Dessa värden har erhållits från Volvo AB [15].

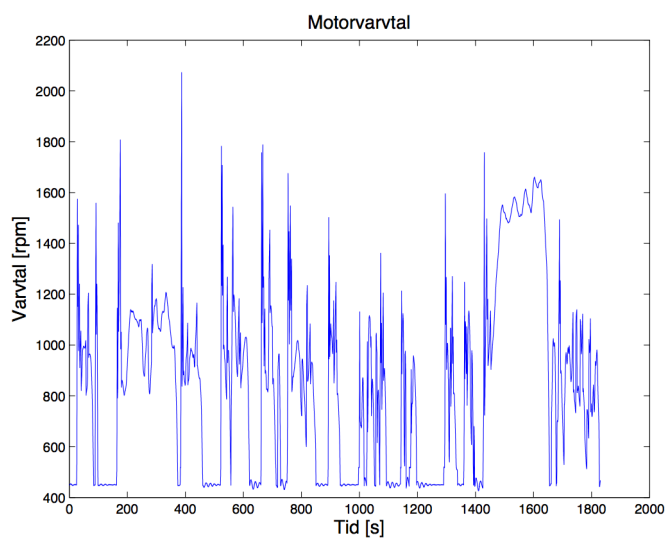


(a) VGT-arean under hela körningens tidsintervall.

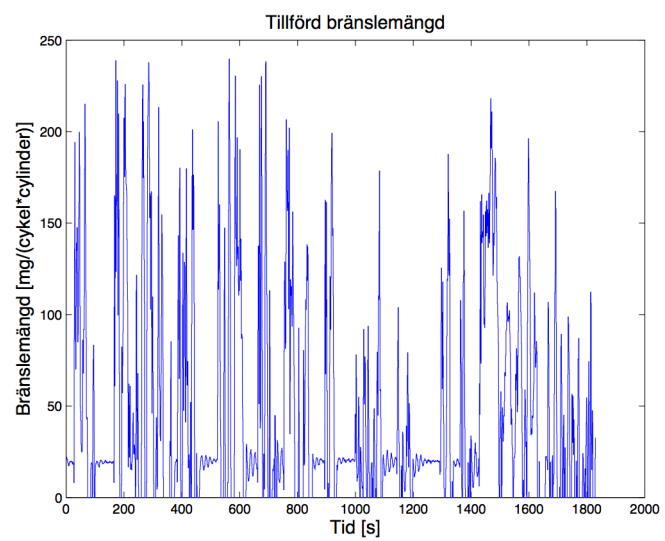


(b) Den justerbara strypventilens area under hela körningens tidsintervall.

Figur 31

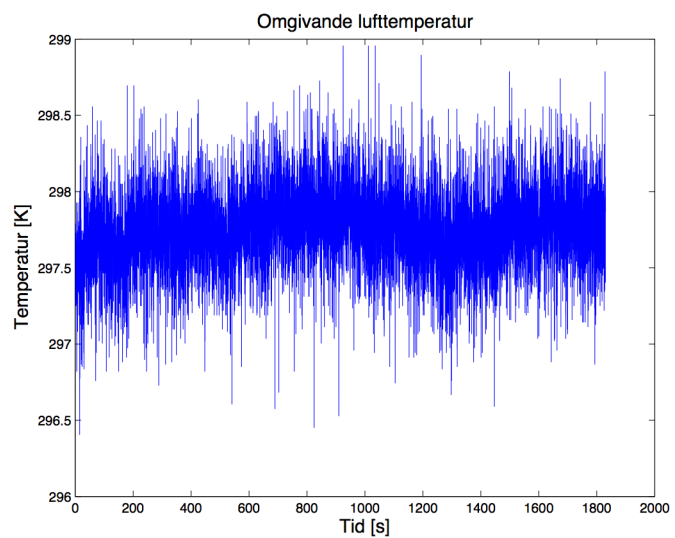
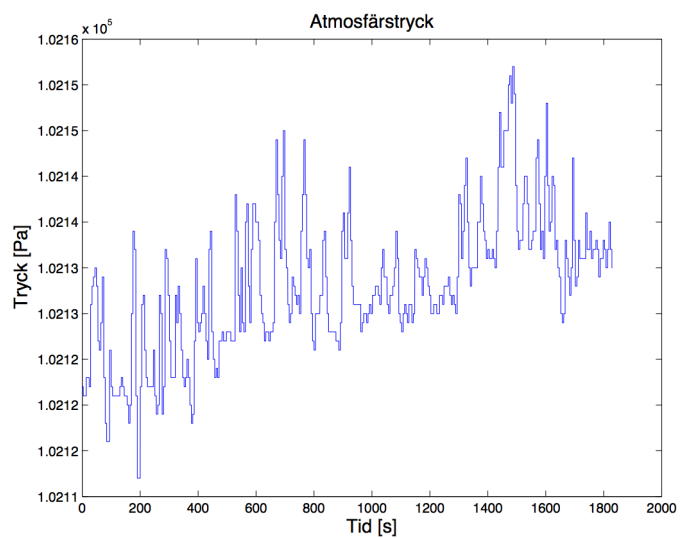


(a) Motors varvtal under hela körningens tidsintervall.



(b) Tillförd bränslemängd under hela körningens tidsintervall.

Figur 32



(a) Uppmätt atmosfärstryck under hela körningens tidsintervall. (b) Uppmätt atmosfärstemperatur under hela körningens tidsintervall.

Figur 33

C Modelica-kod för dieselmotormodellen

Här nedan följer koduppbyggnaden för de komponenter som användes i projektet för att modellera personbilsmotorn och jämföra denna mot en verklig motor. Samma komponenter, men med andra parametrar, användes även för lastbilsmotorn. Flera av komponenterna läser in en del av den uppmätta körningsdatan (i tabellform) som en insignal, som exempelvis motorvarvtal för cylinderkomponenten. Av sekretesskäl kan inte koden för turbinen och kompressorn redovisas.

Först presenteras de olika gränssnitten som användes för att koppla ihop komponenterna med varandra, och att hantera inläsningen av körningsdata och värden från tabeller. Sedan visas en så kallad partialmodell som ärvdades av alla ingående komponenter. Denna innehåller diverse konstanter som flertalet komponenter har gemensamt. Därefter visas koden för varje komponent var för sig, och avslutningsvis hur alla komponenter kopplas samman till den kompletta modellen.

Gränssnitt

```
connector standard_connector

  flow Modelica.SIunits.MassFlowRate dm_air "Mass flow of air";
  flow Modelica.SIunits.MassFlowRate dm_exh "Mass flow of exhaust";
  Modelica.SIunits.Pressure p_air "Air pressure";
  Modelica.SIunits.Pressure p_exh "Exhaust pressure";
  Modelica.SIunits.Temperature T "Temperature";
  a;
end standard_connector;
```

Figur 34: Standardgränssnitt för luftsystemet.

```
connector axis_connector "Ideal axis with angle theta and torque T"

  flow Modelica.SIunits.Torque Tq "Torque";
  Modelica.SIunits.AngularVelocity w "Angular velocity";

  a;
end axis_connector;
```

Figur 35: Mekaniskt gränssnitt för turbons komponenter.

```
connector fuel_connector "Basic fuel injector"

  Modelica.SIunits.Mass fuel_delta "Mass flow of fuel per cycle and cylinder";

  a;
end fuel_connector;
```

Figur 36: Gränssnitt för tillförd bränslemängd mellan spridare och cylinder.

```
connector real_connector_in =
  input Real "'input Real' as connector"
  a;
```

Figur 37: Gränssnitt för inläsning av värden från tabeller.

```
connector real_connector_out =
    output Real "'output Real' as connector"
a;
```

Figur 38: Gränssnitt för inläsning av värden *till* tabeller.

Partialmodell

```
partial model Start_values "Contains start variables and parameters "

constant Real n_C_fuel=12 "Number of carbon atoms in a fuel molecule";
constant Real n_H_fuel=23 "Number of hydrogen atoms in a fuel molecule";
constant Modelica.SIunits.MolarMass M_air=(0.79*28.01 + 0.21*32)*1e-3
    "Molar mass of air";
constant Modelica.SIunits.MolarMass M_exh=(n_C_fuel*(44.01 + (0.79/0.21)*
    28.01) + n_H_fuel*(0.5*18.02 + (0.79/0.21)*0.25*28.01))/((1/0.21)*
    n_C_fuel + (1.21/0.84)*n_H_fuel)*1e-3 "Molar mass of exhaust";
constant Modelica.SIunits.MolarMass M_fuel=(n_H_fuel*1.01 + n_C_fuel*12.01)
    *1e-3 "Molar mass of fuel";
constant Modelica.SIunits.SpecificHeatCapacity Cp_air=1004.5
    "Specific heat capacity for air";
constant Modelica.SIunits.SpecificHeatCapacity Cp_exh=1183.9
    "Specific heat capacity for exhaust";
constant Real gamma_air=1.4 "Heat capacity ratio for air";
constant Real gamma_exh=1.32 "Heat capacity ratio for exhaust";
constant Real Cv_air = Cp_air/gamma_air;
constant Real Cv_exh = Cp_exh/gamma_exh;
constant Real R(unit="J/(mol.K)") = 8.314462 "Gas constant";
constant Real R_air=R/M_air "Specific gas constant for air";
constant Real R_exh=R/M_exh "Specific gas constant for exhaust";
constant Real n_cyl=5 "Number of cylinders";

equation
//
end Start_values;
```

Figur 39: Diverse konstanter som används av flertalet komponenter. Här används "partial model" för att indikera att klassen inte är komplett. Ärvs sedan av alla ingående komponenter.

Komponenter

```
model atm_conditions
    "Source containing ambient atmospheric data based on \"Start_values\""
    extends Start_values;
    Connectors.standard_connector c_atm
    a;

    Modelica.SIunits.Pressure p_atm "The atmospheric pressure";
    Modelica.SIunits.Temperature T_atm "Atmospheric temperature";
    parameter Modelica.SIunits.Pressure p_exh=0 "Partial exhaust pressure";

    //Ambient temperature from measured data:
    Connectors.real_connector_in engine_in_T
    a;
    //Atmosphere pressure from measured data:
    Connectors.real_connector_in engine_in_p
    a;
equation

    engine_in_p = p_atm;
    engine_in_T = T_atm;

    c_atm.p_air = p_atm;
    c_atm.p_exh = p_exh;
    c_atm.T = T_atm;

a;
end atm_conditions;
```

Figur 40: Modell av omgivande atmosfär. Inläsning av tryck och temperatur från körningsdata.

```

model orifice "Ideal orifice (isentalpic)"
  extends Start_values;
  Connectors.standard_connector c_in
  a;
  Connectors.standard_connector c_out a;

  parameter Modelica.SIunits.Area A=1e-4 "Effective area of orifice";

  Modelica.SIunits.Temperature T "Temperature through component";
  Modelica.SIunits.Density rho_in "Density of air/exhaust-mix in orifice inlet";
  Modelica.SIunits.Pressure p_air_in "Partial air pressure in inlet";
  Modelica.SIunits.Pressure p_air_out "Partial air pressure in outlet";
  Modelica.SIunits.Pressure p_exh_in "Partial exhaust pressure in inlet";
  Modelica.SIunits.Pressure p_exh_out "Partial exhaust pressure in outlet";
  Modelica.SIunits.Pressure p_tot_in "Total pressure in inlet";
  Modelica.SIunits.Pressure p_tot_out "Total pressure in outlet";
  Modelica.SIunits.MassFlowRate dm_tot "Total mass flow";
  Modelica.SIunits.MassFlowRate dm_exh "Exhaust mass flow";
  Modelica.SIunits.MassFlowRate dm_air "Air mass flow";

  Modelica.SIunits.MolarFlowRate dn_air "Molar flow of air";
  Modelica.SIunits.MolarFlowRate dn_exh "Molar flow of exhaust";
  Modelica.SIunits.MolarFlowRate dn_tot "Total molar flow";

equation
// Temperature relations:
  c_in.T = T;
  c_out.T = c_in.T;

// Pressure relations:
  p_air_in = c_in.p_air;
  p_exh_in = c_in.p_exh;
  p_air_out = c_out.p_air;
  p_exh_out = c_out.p_exh;
  p_tot_in = p_air_in + p_exh_in;
  p_tot_out = p_air_out + p_exh_out;

// Mass flow relations:
  dm_exh = c_in.dm_exh;
  c_out.dm_exh = -dm_exh;
  dm_air = c_in.dm_air;
  c_out.dm_air = -dm_air;
  dm_tot = dm_exh + dm_air;

// Molar flow relations:
  dn_air = dm_air/M_air;
  dn_exh = dm_exh/M_exh;
  dn_tot = dn_air + dn_exh;

  p_air_in*dn_tot/p_tot_in = dn_air; //Relation between partial and total molar flows.

// Other relations:
  rho_in = p_air_in/(R_air*T) + p_exh_in/(R_exh*T) "Sum of partial densities";
  dm_tot = A*sqrt(2*rho_in)*sqrt(max(p_tot_in - p_tot_out, 0)) "Total mass flow";

a;
end orifice;

```

Figur 41: Strypventil.

```
model reservoir "Ideal gas reservoir with one inlet and one outlet"
  extends Start_values;
  Connectors.standard_connector c_in
    a;
  Connectors.standard_connector c_out
    a;

  parameter Modelica.SIunits.Pressure p_res_airstart=(101325+100)/2
    "Start mass of air in reservoir";
  parameter Modelica.SIunits.Pressure p_res_exhstart=(101325+100)/2
    "Start mass of exhaust in reservoir";
  parameter Modelica.SIunits.Volume V_res=10e-3 "Reservoir volume";

  Modelica.SIunits.Mass m_air "Mass of air in reservoir";
  Modelica.SIunits.Mass m_exh "Mass of exhaust in reservoir";

  Modelica.SIunits.Pressure p_res_air(start=p_res_airstart, fixed=true)
    "Air pressure in reservoir";
  Modelica.SIunits.Pressure p_res_exh(start=p_res_exhstart, fixed=true)
    "Exhaust pressure in reservoir";
  Modelica.SIunits.Pressure p_res "Total pressure in reservoir";

  Modelica.SIunits.Temperature T_res(start=293.15, fixed=false)
    "Temperature inside reservoir";

equation
  //Mass conservation:
  der(m_air) = c_in.dm_air + c_out.dm_air;
  der(m_exh) = c_in.dm_exh + c_out.dm_exh;

  T_res = c_in.T;
  c_out.T = T_res;

  //Ideal gas law:
  p_res_air = m_air*R_air*T_res/(V_res);
  p_res_exh = m_exh*R_exh*T_res/(V_res);

  p_res = p_res_air + p_res_exh;

  c_in.p_exh = p_res_exh;
  c_in.p_air = p_res_air;
  c_out.p_exh = p_res_exh;
  c_out.p_air = p_res_air;

  a;
end reservoir;
```

Figur 42: Enkel kontrollvolym med masskonservering.

```

model reservoir_varT
  "Ideal gas reservoir with one inlet and one outlet. Mass and energy conservation"
  extends Start_values;
  Connectors.standard_connector c_in
  a;
  Connectors.standard_connector c_out
  a;

  parameter Modelica.SIunits.Pressure p_res_airstart=0
    "Start pressure of air in inlet reservoir";
  parameter Modelica.SIunits.Pressure p_res_exhstart=130000
    "Start pressure of exhaust in inlet reservoir";
  parameter Modelica.SIunits.Volume V_res=4e-3 "Reservoir volume";

  Modelica.SIunits.Mass m_air "Mass of air in reservoir";
  Modelica.SIunits.Mass m_exh "Mass of exhaust in reservoir";
  Modelica.SIunits.Mass m_tot "Total mass in reservoir";

  Modelica.SIunits.Pressure p_res_air(start=p_res_airstart, fixed=true)
    "Air pressure in reservoir";
  Modelica.SIunits.Pressure p_res_exh(start=p_res_exhstart, fixed=true)
    "Exhaust pressure in reservoir";
  Modelica.SIunits.Pressure p_res "Total pressure in reservoir";

  Modelica.SIunits.MassFlowRate dm_air_in "Air mass flow from compressor";
  Modelica.SIunits.MassFlowRate dm_exh_in "Exhaust mass flow from compressor";
  Modelica.SIunits.MassFlowRate dm_air_out "Air mass flow out";
  Modelica.SIunits.MassFlowRate dm_exh_out "Exhaust mass flow out";
  Modelica.SIunits.MassFlowRate dm_tot_in "Total mass flow from compressor";
  Modelica.SIunits.MassFlowRate dm_tot_out "Total mass flow out";
  Modelica.SIunits.Temperature T_res(start=593, fixed=true)
    "Temperature inside reservoir";
  Modelica.SIunits.Temperature T_in "Temperature inside reservoir";
  Modelica.SIunits.Temperature T_out "Temperature inside reservoir";

  Modelica.SIunits.SpecificHeatCapacity Cv;
  Modelica.SIunits.SpecificHeatCapacity Cp;
  Modelica.SIunits.EnergyFlowRate dQ(start=0, fixed=true)
    "Energy flow through reservoir surface";
equation
  // Temperature relations:
  T_in = c_in.T;
  T_out = c_out.T;

  T_out = T_res;

  // Massflow relations:
  dm_air_in = c_in.dm_air;
  dm_exh_in = c_in.dm_exh;
  dm_air_out = c_out.dm_air;
  dm_exh_out = c_out.dm_exh;

  dm_tot_in = dm_air_in + dm_exh_in;
  dm_tot_out = dm_air_out + dm_exh_out;

  // Conservation of mass and energy:
  m_tot = m_air + m_exh;
  der(m_tot) = dm_air_in + dm_exh_out;

  der(p_res) = (R_air*T_res/V_res)*(dm_air_in + dm_exh_in + dm_air_out + dm_exh_out) + m_tot*R_air*der(T_res)/V_res;
  der(T_res) = (1/(m_tot*Cv))*(dm_air_in*Cv_air*(T_in-T_res) + dm_exh_in*Cv_exh*(T_in-T_res) + R_air*(T_in*(dm_air_in + dm_exh_in)
  + T_res*(dm_air_out + dm_exh_out)) + dQ);
  dQ = 0 "Energy flow is always zero over surface";

  // Total heat capacity relations:
  Cv = Cv_air + (Cv_exh-Cv_air)*m_exh/m_tot "Total heat capacity";
  Cp = Cp_air + (Cp_exh-Cp_air)*m_exh/m_tot "Total heat capacity";

  // Pressure relations:
  p_res_air = m_air*R_air*T_res/(V_res);
  p_res_exh = m_exh*R_exh*T_res/(V_res);

  p_res = p_res_air + p_res_exh;

  c_in.p_exh = p_res_exh;
  c_in.p_air = p_res_air;
  c_out.p_exh = p_res_exh;
  c_out.p_air = p_res_air;

  a;
end reservoir_varT;

```

Figur 43: Kontrollvolym med både mass- och energikonservering. Användes i den utökade modellen.

```

model intake_manifold
  "Intake manifold with constant volume"
  extends Start_values;
  Connectors.standard_connector c_in
    a;
  Connectors.standard_connector c_egr
    a;
  Connectors.standard_connector c_out a;
  parameter Modelica.SIunits.Pressure p_im_airstart=101325-100
    "Start pressure of air in inlet manifold";
  parameter Modelica.SIunits.Pressure p_im_exhstart=0
    "Start pressure of exhaust in inlet manifold";
  parameter Modelica.SIunits.Volume V_im=4e-3 "Inlet manifold volume";

  Modelica.SIunits.Mass m_air "Mass of air in manifold";
  Modelica.SIunits.Mass m_exh "Mass of exhaust in manifold";

  Modelica.SIunits.Pressure p_im_air(start=p_im_airstart, fixed=true)
    "Air pressure in manifold";
  Modelica.SIunits.Pressure p_im_exh(start=p_im_exhstart, fixed=true)
    "Exhaust pressure in manifold";
  Modelica.SIunits.Pressure p_im "Total pressure in manifold";

  Modelica.SIunits.Temperature T_im "Temperature inside manifold";
equation
  //Mass conservation:
  der(m_air) = c_egr.dm_air + c_in.dm_air + c_out.dm_air;
  der(m_exh) = c_egr.dm_exh + c_in.dm_exh + c_out.dm_exh;

  T_im = c_in.T;
  c_out.T = T_im;

  //Ideal gas law:
  p_im_air = m_air*R_air*T_im/(V_im);
  p_im_exh = m_exh*R_exh*T_im/(V_im);

  p_im = p_im_air + p_im_exh;

  c_egr.p_exh = p_im_exh;
  c_egr.p_air = p_im_air;
  c_in.p_exh = p_im_exh;
  c_in.p_air = p_im_air;
  c_out.p_exh = p_im_exh;
  c_out.p_air = p_im_air;

  a;
end intake_manifold;

```

Figur 44: Enkel insugsmanifold med masskonservering. Motsvarande avgasmanifold beskrivs analogt.

```

model intake_manifold_varT
  "Intake manifold based on mass and energy conservation"
  extends Start_values;
  Connectors.standard_connector c_in
    a;
  Connectors.standard_connector c_egr
    a;
  Connectors.standard_connector c_out a;
  parameter Modelica.SIunits.Pressure p_im_airstart=101325-100
    "Start pressure of air in inlet manifold";
  parameter Modelica.SIunits.Pressure p_im_exhstart=0
    "Start pressure of exhaust in inlet manifold";
  parameter Modelica.SIunits.Volume V_im=4e-3 "Inlet manifold volume";

  Modelica.SIunits.Mass m_air "Mass of air in manifold";
  Modelica.SIunits.Mass m_exh "Mass of exhaust in manifold";
  Modelica.SIunits.Mass m_tot "Total mass in manifold";

  Modelica.SIunits.Pressure p_im_air(start=p_im_airstart, fixed=true)
    "Air pressure in manifold";
  Modelica.SIunits.Pressure p_im_exh(start=p_im_exhstart, fixed=true)
    "Exhaust pressure in manifold";
  Modelica.SIunits.Pressure p_im "Total pressure in manifold";

```

```

Modelica.SIunits.MassFlowRate dm_air_in "Air mass flow from compressor";
Modelica.SIunits.MassFlowRate dm_exh_in "Exhaust mass flow from compressor";
Modelica.SIunits.MassFlowRate dm_air_egr "Air mass flow from egr";
Modelica.SIunits.MassFlowRate dm_exh_egr "Exhaust mass flow from egr";
Modelica.SIunits.MassFlowRate dm_air_out "Air mass flow out";
Modelica.SIunits.MassFlowRate dm_exh_out "Exhaust mass flow out";
Modelica.SIunits.MassFlowRate dm_tot_in "Total mass flow from compressor";
Modelica.SIunits.MassFlowRate dm_tot_egr "Total mass flow from egr";
Modelica.SIunits.MassFlowRate dm_tot_out "Total mass flow out";
Modelica.SIunits.Temperature T_im(start=293, fixed=true)
  "Temperature inside manifold";
Modelica.SIunits.Temperature T_in "Temperature inside manifold";
Modelica.SIunits.Temperature T_egr "Temperature inside manifold";
Modelica.SIunits.Temperature T_out "Temperature inside manifold";

Modelica.SIunits.SpecificHeatCapacity Cv;
Modelica.SIunits.SpecificHeatCapacity Cp;
Modelica.SIunits.EnergyFlowRate dQ(start=0, fixed=true)
  "Energy flow through reservoir surface";

equation
// Temperature relations:
T_in = c_in.T;
T_egr = c_egr.T;
T_out = c_out.T;

T_out = T_im;

// Massflow relations:
dm_air_in = c_in.dm_air;
dm_exh_in = c_in.dm_exh;
dm_air_egr = c_egr.dm_air;
dm_exh_egr = c_egr.dm_exh;
dm_air_out = c_out.dm_air;
dm_exh_out = c_out.dm_exh;

dm_tot_in = dm_air_in + dm_exh_in;
dm_tot_egr = dm_air_egr + dm_exh_egr;
dm_tot_out = dm_air_out + dm_exh_out;

// Conservation of mass and energy:
m_tot = m_air + m_exh;
der(m_tot) = dm_tot;
der(m_tot) = dm_tot;

der(p_im) = (R_air*T_im/V_im)*(dm_tot_in + dm_tot_egr + dm_tot_out) + m_tot*R_air*der(T_im)/V_im;
der(T_im) = (1/(m_tot*Cv))*(dm_tot_in*Cv_air*(T_in-T_im) + dm_tot_egr*Cv_egr*(T_egr-T_im)
+ dm_tot_out*Cv_out*(T_out-T_im) + R_air*(T_im*(dm_tot_in + dm_tot_egr) + T_egr*(dm_tot_egr + dm_tot_out) + dQ));

dQ = 0 "The energy flow is always zero over the surface";

// Total heat capacities relations:
Cv = Cv_air + (Cv_exh-Cv_air)*m_exh/m_tot "Total heat capacity";
Cp = Cp_air + (Cp_exh-Cp_air)*m_exh/m_tot "Total heat capacity";

// Pressure relations
p_im = m_tot*R_air*T_im/(V_im);
p_im_exh = m_exh*R_exh*T_im/(V_im);

p_im = p_im + p_im_exh;

c_egr.p_exh = p_im_exh;
c_egr.p_air = p_im;
c_in.p_exh = p_im_exh;
c_in.p_air = p_im;
c_out.p_exh = p_im_exh;
c_out.p_air = p_im;

a;
end intake_manifold_varT;

```

Figur 45: Insugsmanifold med både mass- och energikonservering. Användes i den utökade modellen. Motsvarande avgasmanifold beskrivs analogt.


```

model egr_valve_in "Simple orifice with adjustable area"

  extends Start_values;

  Connectors.standard_connector c_in
    a;
  Connectors.standard_connector c_out
    a;

  Modelica.SIunits.Area A_egr=2e-4 "Effective area of EGR valve";

  Real u_egr(start=1, fixed=true)
    "EGR-valve opening, 0=closed, 1= fully open";
  Modelica.SIunits.Temperature T "Temperature through component";
  Modelica.SIunits.Density rho_in "Density of air/exhaust-mix in orifice inlet";
  Modelica.SIunits.Pressure p_air_in "Partial air pressure in inlet";
  Modelica.SIunits.Pressure p_air_out "Partial air pressure in outlet";
  Modelica.SIunits.Pressure p_exh_in "Partial exhaust pressure in inlet";
  Modelica.SIunits.Pressure p_exh_out "Partial exhaust pressure in outlet";
  Modelica.SIunits.Pressure p_tot_in "Total pressure in inlet";
  Modelica.SIunits.Pressure p_tot_out "Total pressure in outlet";
  Modelica.SIunits.MassFlowRate dm_tot "Total mass flow";
  Modelica.SIunits.MassFlowRate dm_exh "Exhaust mass flow";
  Modelica.SIunits.MassFlowRate dm_air "Air mass flow";

  Modelica.SIunits.MolarFlowRate dn_air "Molar flow of air";
  Modelica.SIunits.MolarFlowRate dn_exh "Molar flow of exhaust";
  Modelica.SIunits.MolarFlowRate dn_tot "Total molar flow";

  //Effective area from look-up table:
  Connectors.real_connector_in engine_in
    a;
equation

  u_egr = 1;
  A_egr = engine_in;

  // Temperature relations:
  c_in.T = T;
  c_in.T = c_out.T;

  // Pressure relations:
  p_air_in = c_in.p_air;
  p_exh_in = c_in.p_exh;
  p_air_out = c_out.p_air;
  p_exh_out = c_out.p_exh;
  p_tot_in = p_air_in + p_exh_in;
  p_tot_out = p_air_out + p_exh_out;

  // Mass flow relations:
  dm_exh = c_in.dm_exh;
  c_out.dm_exh = -dm_exh;
  dm_air = c_in.dm_air;
  c_out.dm_air = -dm_air;
  dm_tot = dm_exh + dm_air;

  // Molar flow relations:
  dn_air = dm_air/M_air;
  dn_exh = dm_exh/M_exh;
  dn_tot = dn_air + dn_exh;

  p_air_in*dn_tot/p_tot_in = dn_air;

  // Other relations:
  rho_in = p_air_in/(R_air*T) + p_exh_in/(R_exh*T) "Sum of partial densities";
  dm_tot = u_egr*A_egr*sqrt(2*rho_in)*sqrt(max(p_tot_in - p_tot_out, 0));

  a;
end egr_valve_in;

```

Figur 46: EGR-ventil. Inläsning av effektiv area från körningsdata. Den justerbara strypventilen för lastbilmotorn beskrivs på samma sätt.

```

model cylinders_extra_connectors in
  "Cylinder component with extra in- and out ports for volumetric efficiency and combustion temperature"
  extends Start_values;

  parameter Modelica.SIunits.Volume V_d=2.4e-3 "Engine displaced volume";
  constant Real mass_ratio = (1/0.21)*(n_C_fuel + n_H_fuel/4)*(M_air/M_fuel);

  Modelica.SIunits.Efficiency eta_vol(start=1, fixed=false)
    "Volumetric efficiency";
  Modelica.SIunits.Temperature delta_T(start=400, fixed=false)
    "Temperature difference through component";
  Real n_e "[rpm] Engine speed";
  Modelica.SIunits.MassFlowRate dm_tot_in "Total mass flow into cylinder";
  Modelica.SIunits.MassFlowRate dm_air_in "Air mass flow into cylinder";
  Modelica.SIunits.MassFlowRate dm_exh_in "Exhaust mass flow into cylinder";
  Modelica.SIunits.MassFlowRate dm_tot_out "Total mass flow out of cylinder";
  Modelica.SIunits.MassFlowRate dm_exh_out "Exhaust mass flow";
  Modelica.SIunits.MassFlowRate dm_fuel(start=0, fixed=false)
    "Total fuel mass flow";
  Real fuel_delta(start=10, fixed=false)
    "Fuel injected per cycle and cylinder [mg]";

  Connectors.standard_connector c_in
    a;
  Connectors.standard_connector c_out
    a;
  Connectors.fuel_connector c_fuel
    a;
  Diesel_engine.Connectors.real_connector_out real_out_n_e
    a;
  Diesel_engine.Connectors.real_connector_out real_out_fuel
    a;
  Connectors.real_connector_in real_in_delta_T
    a;
  Connectors.real_connector_in real_in_eta_vol
    a;
  Connectors.real_connector_in engine_in
    a;
equation

  n_e = engine_in "Engine speed input";

  fuel_delta = c_fuel.fuel_delta "Fuel injected input";
  dm_fuel = (1/120)*fuel_delta*n_e*n_cyl*1e-6;

  real_out_n_e = n_e "Engine speed as output for VE look up-table";
  real_out_fuel = fuel_delta "Injected fuel as output for VE look up-table";
  eta_vol = real_in_eta_vol "VE as input from look up-table";

  // Temperature relations:
  delta_T = real_in_delta_T;
  c_out.T = c_in.T + delta_T;

  // Mass flow relations:
  dm_air_in = eta_vol*(c_in.p_air)*n_e*V_d/(120*R_air*c_in.T); // Air mass flow from intake manifold
  dm_exh_in = eta_vol*(c_in.p_exh)*n_e*V_d/(120*R_exh*c_in.T); // Exhaust mass flow from intake manifold
  dm_tot_in = dm_air_in + dm_exh_in; // Total mass flow from intake manifold

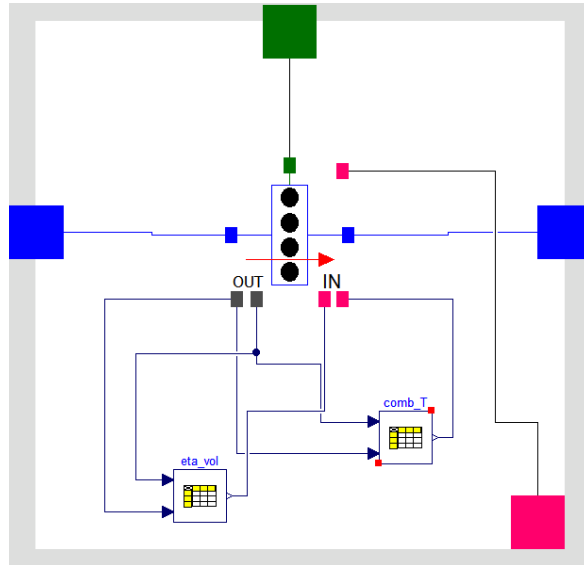
  dm_tot_out = dm_tot_in + dm_fuel; // Total mass flow to exhaust manifold
  dm_exh_out = dm_exh_in + dm_fuel*(1 + mass_ratio); // Mass flow of exhaust to exhaust manifold

  c_in.dm_air = dm_air_in;
  c_in.dm_exh = dm_exh_in;
  c_out.dm_exh = -dm_exh_out;
  c_out.dm_air = -(dm_tot_out - dm_exh_out);

  a;
end cylinders_extra_connectors in;

```

Figur 47: Cylindrar. Inläsning av volymetrisk verkningsgrad och förbränningstemperatur från mappningstabeller, samt motorvarvtal från körningsdata.



Figur 48: Grafisk beskrivning av cylinderkomponenten. Blå koppling = standardgränssnitt, grön = gränssnitt för tillförd bränslemängd, grå/rosa = inläsning till/från mappningstabell och körningsdata.

```

model cylinders_w_enginedata_in
  "Cylinders with engine data over volumetric efficiency and combustion temperatures implemented in look up tables"
  import Diesel_engine_1 = Diesel_engine;

  Connectors.standard_connector standard_connector
  a;
  Connectors.standard_connector standard_connector1
  a;
  Modelica.Blocks.Tables.CombiTable2D eta_vol(
    tableOnFile=true,
    smoothness=Modelica.Blocks.Types.Smoothness.ContinuousDerivative,
    tableName="vol_eff",
    fileName="engine_data_complete.mat")
    "Table of volumetric efficiency, x-axis = engine speed [rpm], y-axis mass fuel injection [mg/(cycle*cyl)]"
  a;
  Modelica.Blocks.Tables.CombiTable2D comb_T(
    tableOnFile=true,
    smoothness=Modelica.Blocks.Types.Smoothness.ContinuousDerivative,
    tableName="T_diff",
    fileName="engine_data_complete.mat")
    "Increase in temperature due to combustion; x-axis = engine speed [rpm], y-axis = mass fuel injection [mg/(cycle*cyl)]"
  a;
  Connectors.fuel_connector fuel_connector
  a;
  Diesel_engine_1.Connectors.real_connector_in engine_in
  a;
  Diesel_engine_1.Components.In_data.cylinders_extra_connectors_in
  cylinders_extra_connectors_in
  a;
equation
  connect(cylinders_extra_connectors_in.c_in, standard_connector) a;
  connect(cylinders_extra_connectors_in.c_out, standard_connector1) a;
  connect(cylinders_extra_connectors_in.c_fuel, fuel_connector) a;
  connect(eta_vol.u2, cylinders_extra_connectors_in.real_out_n_e) a;
  connect(comb_T.u2, cylinders_extra_connectors_in.real_out_n_e) a;
  connect(eta_vol.u1, cylinders_extra_connectors_in.real_out_fuel) a;
  connect(comb_T.u1, cylinders_extra_connectors_in.real_out_fuel) a;
  connect(eta_vol.y, cylinders_extra_connectors_in.real_in_eta_vol) a;
  connect(comb_T.y, cylinders_extra_connectors_in.real_in_delta_T) a;
  connect(cylinders_extra_connectors_in.engine_in, engine_in) a;
  a;
end cylinders_w_enginedata_in;

```

Figur 49: Koduppbyggnad för kopplingarna i den grafiska beskrivningen ovan.

```
model fuel_injector_in "Simple fuel injector"
  extends Start_values;

  //Fuel injected to cylinder component:
  Connectors.fuel_connector fuel_out
  a;
  //Fuel injected from look-up table:
  Connectors.real_connector_in engine_in
  a;
equation
  engine_in = fuel_out.fuel_delta;

  a;
end fuel_injector_in;
```

Figur 50: Bränslespridare i anslutning till cylinderkomponenten. Inläsning av tillförd bränslemängd från körningsdata.

```
model heat_exchanger
  "Heat exchanger that can be used for IC and other applications,
  cools to ambient temperature with an efficiency eta"

  Connectors.standard_connector c1
  a;
  Connectors.standard_connector c2
  a;
  Connectors.real_connector_in engine_in //Connector to look-up table
  a;

  parameter Real eta=0.7 "Heat exchanger efficiency";
  Modelica.SIunits.Temperature T_atm "Atmospheric temperature";

equation
  engine_in = T_atm;

  c2.T = T_atm*eta + (1 - eta)*c1.T;
  c1.p_air = c2.p_air;
  c1.p_exh = c2.p_exh;
  c1.dm_air = -c2.dm_air;
  c1.dm_exh = -c2.dm_exh;

end heat_exchanger;
```

Figur 51: Kylare. Inläsning av omgivande temperatur från körningsdata.

```
model axis_stiff
  "Axis with moment of inertia J, the angle velocity is equal in both ends"

  Connectors.axis_connector a1 a;
  Connectors.axis_connector a2 a;

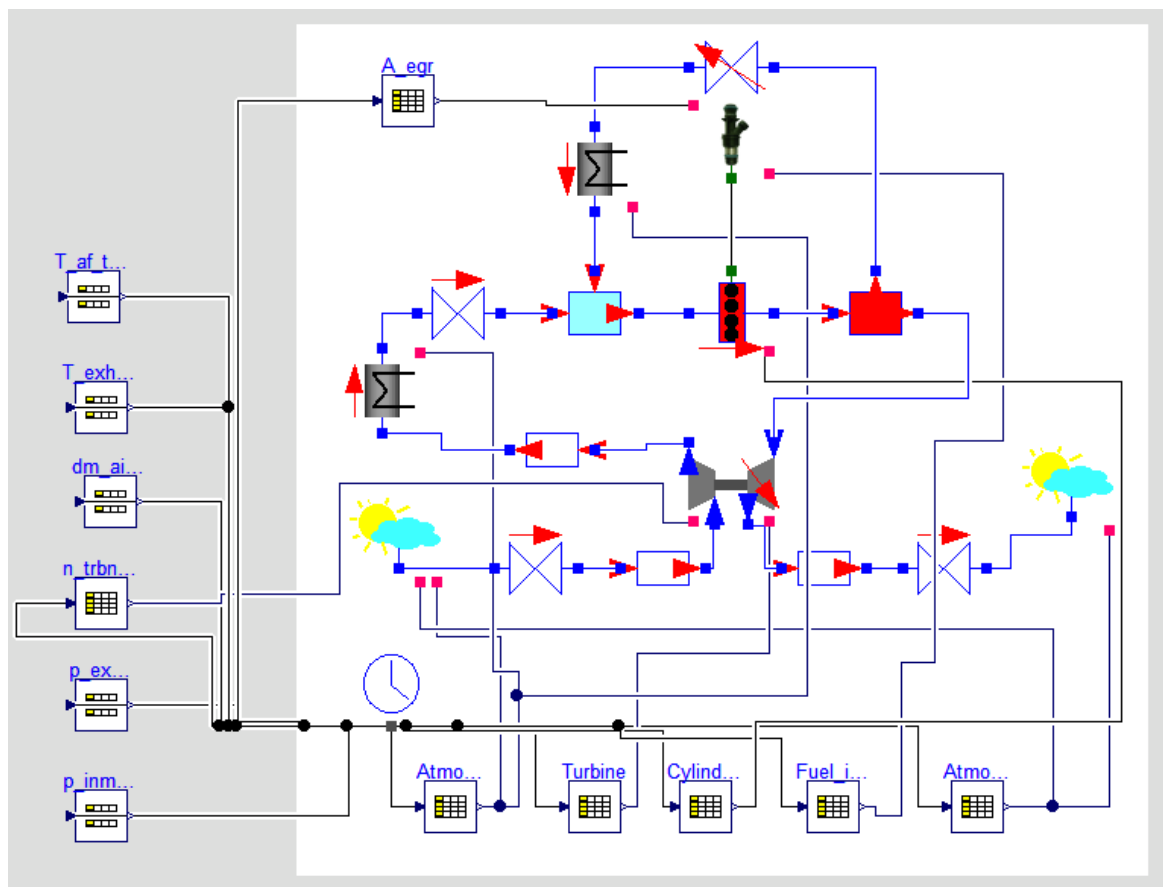
  parameter Modelica.SIunits.AngularVelocity w_start=2300;
  parameter Modelica.SIunits.Inertia J=3e-5
  "Inertia for axis combined with compressor and turbine inertia";
  Modelica.SIunits.AngularVelocity w(start=w_start, fixed=true)
  "Angular velocity of axis";

equation
  der(w) = (a1.Tq + a2.Tq)/J "Angular acceleration of axis";
  a1.w = a2.w;
  a1.w = w;

  a;
end axis_stiff;
```

Figur 52: Turbinaxel. Fungerar som mekanisk länk mellan turbin och kompressor.

Komplett modell



Figur 53: Grafisk beskrivning av den kompletta modellen. Tabellerna i över- och underkanten är körningsdata som används som insignal, medan tabellerna till vänster är uppmätta värden som används som referens vid jämförelse mot den verkliga motorn.

```

model Version_3
equation
  connect(clock.engine_time, Atmosphere_p.u) a;
  connect(Atmosphere_p.y[1], atm_conditions_wo_T.engine_in_p) a;
  connect(atm_conditions.c_atm, Air_filter.c_in)
    a;
  connect(clock.engine_time, Atmosphere_T.u) a;
  connect(Atmosphere_T.y[1], atm_conditions.engine_in_T) a;
  connect(Atmosphere_p.y[1], atm_conditions.engine_in_p) a;
  connect(clock.engine_time, Turbine.u) a;

  connect(Turbine.y[1], turbo_1D_in.engine_in) a;
  connect(intercooler.c2, Intercooler_area.c_in)
    a;
  connect(Atmosphere_T.y[1], intercooler.engine_in) a;

  connect(Atmosphere_T.y[1], EGR_cooler.engine_in) a;
  connect(clock.engine_time, Cylinders.u) a;

  connect(clock.engine_time, Fuel_injector.u) a;
  connect(Fuel_injector.y[1], fuel_injector_simple_in.engine_in)
    a;
  connect(Exhaust_pipe.c_out, atm_conditions_wo_T.c_atm)
    a;
  connect(cylinders_w_enginedata_in.fuel_connector, fuel_injector_simple_in.fuel_out)
    a;
  connect(cylinders_w_enginedata_in.engine_in, Cylinders.y[1]) a;
  connect(clock.engine_time, p_inman_utdata.u)
    a;
  connect(clock.engine_time, p_exman_utdata.u) a;
  connect(clock.engine_time, n_trbn_utdata.u) a;
  connect(clock.engine_time, dm_air_cmpr_utdata.u) a;
  connect(clock.engine_time, T_exhaust_utdata1.u) a;
  connect(n_trbn_utdata.y[1], turbo_1D_in.engine_in_w) a;
  connect(clock.engine_time, T_af_trbn_utdata.u) a;
  connect(Intercooler_area.c_out, intake_manifold.c_in) a;
  connect(intake_manifold.c_out, cylinders_w_enginedata_in.standard_connector)
    a;
  connect(intake_manifold.c_egr, EGR_cooler.c2) a;
  connect(cylinders_w_enginedata_in.standard_connector1, exhaust_manifold.c_in)
    a;
  connect(exhaust_manifold.c_out, turbo_1D_in.c_trbn_in) a;
  connect(turbo_1D_in.c_cmpr_out, reservoir_after_cmpr.c_in) a;
  connect(reservoir_after_cmpr.c_out, intercooler.c1) a;
  connect(Air_filter.c_out, reservoir_before_cmpr.c_in) a;
  connect(reservoir_before_cmpr.c_out, turbo_1D_in.c_cmpr_in) a;
  connect(reservoir_after_trbn.c_out, Exhaust_pipe.c_in) a;
  connect(reservoir_after_trbn.c_in, turbo_1D_in.c_trbn_out) a;
  connect(exhaust_manifold.c_egr, egr_valve_in.c_in) a;
  connect(egr_valve_in.c_out, EGR_cooler.c1) a;
  connect(step.y, egr_valve_in.engine_in) a;
a;
end Version_3;

```

Figur 54: Sammankoppling av komponenter till en komplett modell som läser in körningsdata från tabeller.