ISSN 0280-5316
ISRN LUTFD2/TFRT--5887--SE

Modeling of NO_x formation in heavy duty engines

Kenan Murić

Department of Automatic Control
Lund University
September 2011

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
September 2011

Document Number
ISRN LUTFD2/TFRT--5887--SE

Author(s)
Kenan Murić

Supervisor
Ola Stenlåås, Scania CV AB Södertälje, Sweden
Rolf Johansson Automatic Control Lund, Sweden (Examiner)

Sponsoring organization

Title and subtitle
Modeling of NO_x formation in heavy duty engines. (Modellering av NO_x –bildning i HD-motorer)

Abstract
In order to apply combustion feedback control in diesel engines to control pollution formation, good models are needed to estimate the states. One of those would be NO_x levels which are regulated by European law. There are conventional sensors on the market that can measure NO_x but these are too slow for in–cycle feedback control.
The purpose of this thesis was to derive a model to estimate NO_x where the engine very well could have a built in EGR function. The model seems promising, however additional tuning and modification is needed for it to be fully useful in practice.
The model presented in this thesis is a zero–dimensional two–zone model. The NO_x model assumes that combustion locally occurs at stoichiometric conditions. Model validation is done against single–cylinder engine NO_x output data received from the Royal Institute of Technology (KTH). Least–squares fitting methods are also used to simplify calculations considerably. The results indicate that the model is applicable in practice.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
80

Recipient's notes

Security classification

Acknowledgements

First off, I would like to thank my supervisor Dr. Ola Stenlåås for all the encouragement and support during this master thesis project. I would also like to thank my examiner Professor Rolf Johansson for valuable inputs on the report.

I am also thankful to Associate Professor Per Tunestål at the combustion engine department in Lund for inputs on the NO_x model. Mikael Lindström has been essential in this thesis by providing me with data from the Royal Institute of Technology (KTH). Christer Lundberg at NMED also provided me with valuable data on heat release.

Last but not least, I would like to thank all the good people at NMEB, the engine control group, for many interesting discussions around the "fika" table.

The thesis work was done at Scania CV AB in Södertälje.

Contents

1 Introduction

1.1 Motivation

In today's modern society, where dependence on fossil fuel is an environmental problem, more and more engine producers are forced to face tightening emission legislations. Beside the traditional carbon dioxide debate there are also other pollutants that need to be kept at low levels because of their health hazardous impact. There are a total of four exhaust pollutants that are regulated by law: carbon monoxide, hydrocarbons, particulate matter and nitric oxides. The first two are easily brought down to low emission levels by simple oxidation in a catalytic converter. On the other hand, much more work needs to be done in the areas of NO_x (nitrogen oxides) and PM (particulate matter).[1]

In December 2007 the European Commission proposed a new set of regulations in the heavy–duty vehicle industry to lower the amount of pollutions. The new sets of regulations are known by name as Euro VI. The Euro VI proposal tightens the emission standards for heavy– duty vehicles in the areas of NO_x, PM, HC (hydrocarbons) and ammonia emissions. The motivation is that the road transport is a major source of pollutants in Europe. The road transports are in fact the largest contributors to NO_x emissions in urban societies. Therefore there is of course a need for heavy– duty vehicle companies to adapt to the new regulations and to contribute in the overall mission to lower pollution of the environment.[2]

	Date	CO (g/kWh)	NO_x (g/kWh)	HC (g/kWh)	PM (g/kWh)
Euro IV	October 05'	1.5	3.5	0.46	0.02
Euro V	October 08'	1.5	2.0	0.46	0.02
Euro VI	October 13'	1.5	0.4	0.13	0.01

Table 1.1 – Regulations set by the European Commission (for the stationary case). One should pay special notice to the tightening regulations on NO_x emissions. [2]

When talking about NO_x one should keep in mind that it consists of two separate molecular compounds, the nitric oxide and the nitric dioxide. Both compounds are hazardous to people and animals alike. The nitric dioxide can produce ozone according to reaction R1.1, which on ground level is a carcinogenic molecule:

R1.1 $$NO_2(g) + O_2(g) \Leftrightarrow NO(g) + O_3(g)$$

Inhalation of either nitric oxide or ozone is dangerous and in a worst case scenario, lethal.

At Scania CV AB, advanced engine technology is developed to fulfill the Euro VI standards set by the Commission of the European Union. For future engine development, there is of course an interest in having good models that can describe the NO_x formation in heavy duty engines. The purpose of this thesis was to derive a theoretical model for NO_x estimations and also to implement the model in C for future real time executions on an FPGA. The idea is that the model later on can be used in the area of control theory, for effective and fast calculations of NO_x concentrations in diesel feedback control applications.[3]

1.2 Problem formulation

In this master thesis, the investigated problem is twofold. First, a suitable zero dimensional model must be derived that will satisfy two necessities. First and foremost, the model must track the true NO_x formation sufficiently accurate. Secondly, the model has to be simple enough to be implemented on a real time running FPGA. In the case of combustion in a diesel engine, the time constraints are hard to satisfy. It will be essential to fully utilize the parallel structure in the FPGA, as much as possible. This is described in detail in section 2.5. Besides the parallelism, some considerations are needed when it comes to the implementation of algorithms. Although the scope of this thesis does not include a finished FPGA implementation, these aspects will be taken into account nevertheless. The code has to be kept simple and avoidance of division operations will be crucial for future practical utilization of the NO_x model (explained in section 2.5).

1.3 Methodology

There are three clear steps in this master thesis, which can be stated as follows:

1. *Theoretical derivation of a NO_x model.*
2. *Running and testing the model on a PC in MATLAB.*
3. *Implementing the model in C.*

In the first step, a literature study is done to build up a framework for the theoretical work which is needed to derive the NO_x model. The model is then derived using an approach that will be possible to implement later on. This does render some approximations in the theoretical modeling process; however they should generally be sufficient and theoretically valid. In the second step, the paper model is implemented, run and tested in MATLAB. When the test results are satisfying enough, the model will be implemented in C. The resulting C code should then work as a template for a future implementation that will be done on an FPGA.

2 Background

There are fundamentally two key approaches in maintaining NO_x at sufficiently low levels, namely EGR and SCR.

2.1 The Exhaust Gas Recirculation

The basic concept of Exhaust Gas Recirculation is that exhaust gases can be used to control formation rates of NO_x. To see why this is the case; one has to keep John Dec's conceptual model[4] of combustion in mind (see below, section 2.4). The NO_x formation takes place in oxygen rich areas with sufficiently high temperatures. This can also be seen by a NO_x formation rate approximation[5] of initial NO_x formation, where also k_1^+ is a temperature dependent variable:

Eq. (2.1) $$\frac{d[NO]}{dt} = 2k_1^+ K(T)\sqrt{[O_2]_e[N_2]_e}$$

The EGR, consisting of exhaust gases (that is nitrogen, carbon dioxide, oxygen and water), will reduce the oxygen concentration somewhat. This is because of the fact that only a specific quantity of gas can pass during the intake stroke, where EGR is mixed with air. Thus, the NO_x formation rate will be reduced according to Eq. (2.1).

There is also a major thermal effect because of the CO_2 and H_2O in the EGR, that will have an impact on the mean specific heat value of the charge. The reason is the higher heat capacity of the two products, compared to the specific heat of clean air. The combustion temperature will then be lower than it would be if only clean air was used. The variables in Eq. (2.1) are described more extensively in the theory section. This is not the only method to decrease NO_x emissions to meet European regulations.[3] However, the EGR will be an essential part in the NO_x modeling procedure while the same cannot be said about the Selective Catalytic Reduction.

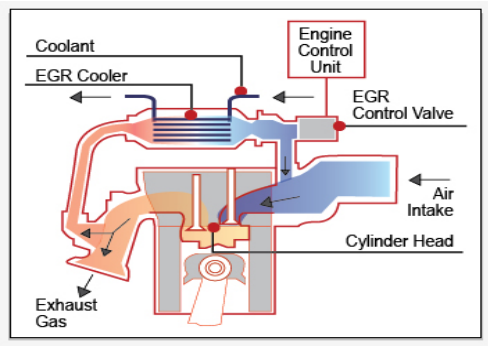

Fig. 2.1 – A schematic picture of the EGR function. Some of the exhaust gases are recirculated in the engine. The EGR is first cooled in the EGR cooler, then released by an EGR control unit (at a desired reference value) and finally mixed with fresh air at the intake [6].

2.2 The Selective Catalytic Reduction

The idea of EGR was to reduce the formation of NO_x itself; however this does not reduce NO_x that is already present. To get rid of already formed NO_x, one needs to find suitable reaction mechanisms to break down the NO_x into molecular compounds that are less harmful to the environment. One common way is to add a Selective Catalytic Reduction (SCR) in the after treatment system of exhaust gases. A reductant, for example urea or ammonia, is used to break down the harmful nitrogen oxide molecules. In the case of urea, this will lead to the following reactions:

R2.1 $4NH_3 + 4NO + O_2 \rightarrow 4N_2 + 6H_2O$ (Standard SCR)

R2.2 $2NH_3 + NO + NO_2 \rightarrow 2N_2 + 3H_2O$ (Fast SCR)

R2.3 $8NH_3 + 6NO_2 \rightarrow 7N_2 + 12H_2O$ (Slow SCR)

The nitrogen oxide is broken down into nitrogen and water. For the reaction to take place there is also a need of a suitable catalyst. There are different catalysts that could be used, where one of them is a titanium based ceramic honey comb catalyst.[7] One problem however, is that the Selective Catalytic Reduction contributes to global warming if it's not implemented wisely in the after treatment system. When there is an overdosing of the reductant in the SCR, a substantial quantity of laughing gas (N_2O) is formed. Laughing gas has 300 times greater impact, per unit weight, on the global warming than carbon dioxide.

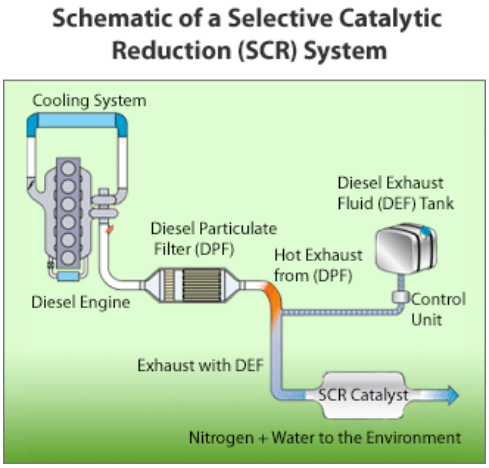

Fig. 2.2 – A simplified schematic picture of the SCR where harmful NO_x is converted to nitrogen and water. [8]

2.3 Models for NO_x estimation purposes

The possibility to produce good models for NO_x level estimations has been researched extensively during the years. The range of complexity of the models derived is wide. The models span from simple zero dimensional models to a variety of advance three dimensional models. This does not necessarily imply that three dimensional models are better to use. In fact, the advance models with high level of complexity have not been successfully implemented for the purposes of real time use. The author's opinion is that these complex models should rather be seen as tools to a better conceptual understanding of NO_x formation during combustion in an engine. The conceptual knowledge gained by this could very well be used to improve zero dimensional models that are applied in real time systems.

The main difference between a zero dimensional model (ZDM) and a three dimensional model (TDM) is that the ZDM does not take spatial properties into account in describing combustion phenomena, such as NO_x formation. The TDM is a more ambitious approach to the spatial properties and its contribution to NO_x formation. For example, one could take into account the turbulence, formation of droplets and the flow field in the combustion engine to produce a high level TDM.[9]

In this thesis however, a ZDM will be implemented and run. The main reason is that the ZDM can be sufficiently good at estimations while being a lot faster than its TDM counterpart. It is possible to evade gruesome and time consuming partial differential equation solvers by using a ZDM instead. The thesis ZDM assumes two zones, an unburned and a burned zone. This concept of modeling will be explained in depth in the theory section (see theory section, section 3).

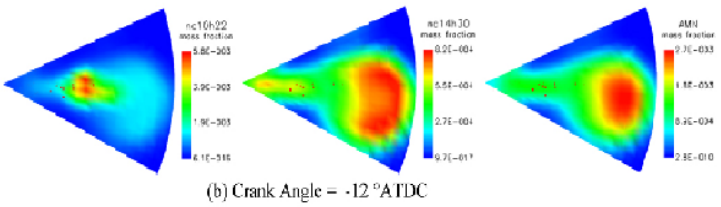

Fig. 2.3 – An example of a 3–dimensional simulation model of fuel vapor, produced with computational fluid dynamics (CFD). The CFD approach is also used in calculations of NO_x formation in a TDM. [10]

2.3.1 NO_x measurement

It is in fact possible to measure NO_x levels using a NO_x–sensor. Nevertheless, a NO_x model is still useful because of the fact that it updates the emission levels considerably faster than the sensor. This property will be essential in crank angle resolved diesel feedback control. The sensor does also not work as good at cold starts and it takes some amount of time for it to work properly. It is also a good idea to have some kind of backup when the sensor cease to work, as it could do, for example, due to a harsh environment (high pressure etc.). The NO_x model can also fulfill the function of surveying the NO_x–sensor and detecting when it starts to fail or drift away.

In the future, with sufficiently accurate and fast NO_x models, it would be possible to remove a NO_x–sensor in the engine system (in a normal case, there are several sensors). More specifically, it would be the NO_x–sensor right before the SCR. This would of course decrease cost for both the manufacturer and the purchaser of heavy duty vehicles.

Fig. 2.4 – A NO_x–sensor used to measure NO_x levels. [11]

2.4 A conceptual model of diesel combustion

Before embarking on a mission to model the NO_x formation one should of course seek a deeper understanding of how combustion takes place in a diesel engine and where the NO_x is formed. John Dec[4] and his research team presented a fairly thorough conceptual model of the combustion process. The work was done by using optically accessible diesel engines. In combination with laser based techniques, they used Laser–Induced Incandescence (LII) and elastic scattering to determine distribution of liquid fuel, soot and particle sizes. To quantify local lambdas, Rayleigh Scatter was used and a Laser Induced Fluorescence (LIF) enabled the possibility to measure local concentrations of NO, hydrocarbons and OH–radicals.[9]

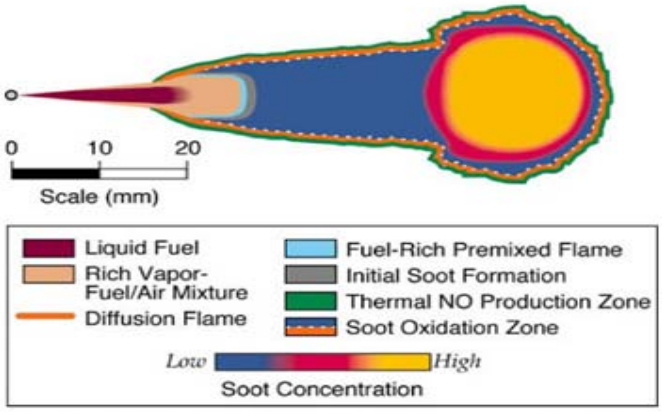

Fig. 2.5 – Quasi–steady Diesel combustion plume as presented by Dec (1997). Courtesy Dr. John E. Dec (Sandia National Laboratories).

As can be seen in Fig. 2.5, the NO_x production takes place on the lean side of the diffusive layer. This is because of the fact that the area is rich on oxygen while at the same time the temperatures are sufficiently high.

Another interesting fact is that the combustion can be divided into two separate steps. In the first step, the combustion takes place in a fuel rich and premixed flame where, although temperatures are high, there is not enough oxygen for NO_x formation to occur. Instead of NO_x, initial soot particles are formed in this section. The second step is the combustion at the diffusive layer (orange). The content after the combustion is directly added to the post–combustion zone, where NO_x is formed on the lean side of the diffusion flame. This fact simplifies the calculations somewhat in the NO_x modeling process.

The conclusion from Dec's research is that the NO_x formation takes place on the lean side of the diffusive flame and in the post combustion zones after the end of combustion.[9]

2.5 The FPGA

Models and control algorithms in engine development are implemented in embedded systems. This is of course due to practical reasons; a PC would not be a suitable option because of its size. The embedded system is essentially an information processing device suitable for products where the processor isn't the most important part of the product. The FPGA (**F**ield **P**rogrammable **G**ate **A**rray) is a silicon chip with a whole net of logic blocks that, depending on the purpose, can be configured to solve a specific task, while at the same time being reconfigurable. This means that a programmed task can be erased and replaced with a completely new and different task. The FPGA also consists of an I/O pad that is used for communication with the outside process.

There is a fundamental benefit gained by using the FPGA instead of an ordinary processor. If an input signal is sent to the device the operations, programmed by the developer, propagate through the FPGA in a parallel manner. The different logical blocks in the FPGA are interconnected through small wires which enables the transport of intermediate results and calculations between units.

The FPGA is best suited for tasks that use integer arithmetic and exhibits a high degree of parallelism. In reality this means that algorithms have to be transformed from ordinary floating point calculations to a fixed point implementation. However, this is rarely a trivial task and in the case of models used in this thesis, a lot of considerations need to be made to maintain a sufficient degree of model stability and precision.[12] As stated, the FPGA is especially suited for tasks with parallel structures. Thus, to exploit the full potential of the FPGA circuit some amount of time should be spent on dividing the task in parallel sub tasks, where it is possible to do so.[13]

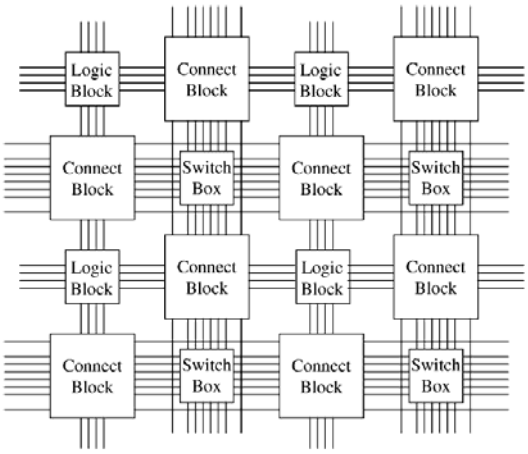

Fig. 2.3 – The principal layout of the FPGA. The connect block works as a switching unit. [13]

One major setback with the Field Programmable Gate Array, in terms of calculations for practical use, is its insufficient handling of division operations. When using the FPGA one should avoid unnecessary divisions if possible. In some cases this is not possible due to the fact that the divisor is an unknown variable that changes, where an example of such a case is the division dp/p. There are a couple of rather neat ways of solving this problem. One of them is to use a Taylor approximation of a suitable function (in this case: $ln\ p$) and applying basic differential calculus, where it will be possible to evade gruesome divisions.[3]

Fig. 2.4 – Example of an Altera Stratix IV GX FPGA [14]

The FPGA can either be programmed by using Low Level Design (LLD) or a High Level Design (HLD) approach. In the LLD, implementation is done using a hardware description language. This is not a preferred way for software developers because of the fact that a lot of the implementation time goes to telling the FPGA which logic block is used for what task. The focus should instead be on programming the functionality. For this purpose it is possible to use the HLD approach instead. This basically means that a high level programming language is used for behavioral descriptions which then are translated to LLD language. Luckily, the FPGA is user friendly enough that it is possible to use a large variety of high level languages, such as C, C++, Java etcetera. It is also possible to use graphical programming languages such as LabVIEW.

In this thesis, the C programming language will suffice for the implementation of a crank angle resolved NO_x model. C is often used in implementation and development of embedded systems, and most engine developers at Scania use C in engine software implementation.

3 Theory

3.1 The rate of heat transfer

To derive a suitable combustion model for a diesel engine, the first law of thermodynamics can preferably be applied. First and foremost, the heat release can be stated in mathematical terms as following[9]:

Eq. (3.1) $$\delta Q_{\,combustion} = dU + \delta W + \delta Q_{\,hl}$$

In this case the dU corresponds to change in the internal energy, $\delta Q_{\,hl}$ the heat loss to the surrounding environment, for example the cylinder walls or due to radiation heat transfers. The combustion also performs a desirable physical work δW. Rewriting the dU–term as $dU = \mathrm{m} \cdot \mathrm{c_v} \cdot \mathrm{dT}$, where m equals mass and $\mathrm{c_v}$ is the specific heat at a constant volume, gives:

Eq. (3.2) $$\delta Q_{\,combustion} = \mathrm{m} \cdot \mathrm{c_v} \cdot \mathrm{dT} + \delta W + \delta Q_{\,hl}$$

This equation can be rewritten further by differentiating the ideal gas law, where change in amount of substance is neglected:

Eq. (3.3) $$m \cdot dT = \frac{pdV + Vdp}{R}$$

Inserting the right side of Eq. (3.3) in Eq. (3.2) and realizing that the work done is $\delta W = p \cdot dV$, gives:

Eq. (3.4) $$\delta Q_{\,combustion} = \mathrm{c_v} \frac{pdV + Vdp}{R} + \mathrm{pdV} + \delta Q_{\,hl}$$

Eq. (3.5) $$\delta Q_{\,combustion} = \mathrm{c_v} \frac{pdV + Vdp + \frac{R}{\mathrm{c_v}}\mathrm{pdV}}{R} + \delta Q_{\,hl}$$

In an ideal gas the constant R, the specific gas constant, can be calculated as $R = c_p - c_v$. Eq. (3.5) can then be written as:

Eq. (3.6) $$\delta Q_{\,combustion} = \mathrm{c_v} \frac{Vdp + (1 + \frac{\mathrm{cp - cv}}{\mathrm{c_v}})\mathrm{pdV}}{\mathrm{cp - cv}} + \delta Q_{\,hl}$$

Differentiating Eq. (3.6) with regards to the crank angle, with $\gamma = \frac{\mathrm{c_p}}{\mathrm{c_v}}$, results in a differential equation on the form[8]:

Eq. (3.7) $$\frac{dQ_{combustion}}{d\varphi} = \frac{\gamma}{\gamma - 1} p \frac{dV}{d\varphi} + \frac{1}{\gamma - 1} V \frac{dp}{d\varphi} + \frac{dQ_{\,hl}}{d\varphi}$$

Solving the equation will give the heat released by the combustion of fuel. Additionally, if the equation is solved, the mass of the burned fuel in the combustion engine can be calculated by integration: $m_{fuel}(\phi) = \frac{1}{Q_{LHV}} \int dQ_{fuel}\, d\phi$.

The Q_{LHV} is the lower heating value of the fuel. The relation between the combustion heat and the heat produced by burning the fuel is simply $Q_{combustion} = Q_{fuel}$. The total fuel mass in the cylinder, at a certain time step, is not equal to the burned mass of fuel. A term has to be added for the unburned fuel, i.e. introducing a combustion efficiency coefficient.

If Eq. (3.7) ought to be useful pressure must be measured. This is not the case in most commercial vehicles today. Therefore an alternative heat release equation can be derived for NO_x modeling purposes. This is just a simple rewriting of Eq. (3.7) on a form where pressure is assumed to be the unknown variable, instead of the heat release:

$$\frac{dp}{d\varphi} = \left(\frac{dQ_n}{d\varphi} - \frac{\gamma}{\gamma-1} p \frac{dV}{d\varphi}\right)\left(\frac{\gamma-1}{V}\right)$$

What is missing for the equation to be useful is a model of the injection of fuel and a model of the combustion process. There are several different approaches to model fuel injection and combustion. A rather simple model of combustion is an assumption that the heat release is proportional to the difference between the energy content of the injected fuel and the accumulated heat release as $dQ = C(Q_{fuel} - Q)$. Injection can on the other hand be modeled as a dependency of the injector needle lift. A common approach is to use a top hat model of the injector needle lift. The approach is suggested in several papers and due to the fact that today's development of injectors goes toward fast piezoelectric units it seems fairly reasonable.[1] However, in this thesis pressure is measured and used for modeling of NO_x formation. This because of the fact that pressure sensors are getting commercialized and are already so on the market.[15]

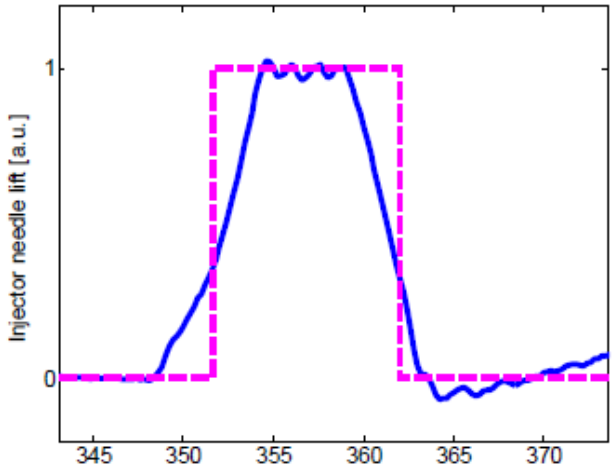

Fig. 3.1 – A top hat model (purple–dashed) of the injector needle lift compared to the real result (blue). [1]

3.2 Calculation of NO_x formation rate

3.2.1 Background

Principally, NO_x can be formed in three different ways: prompt NO_x formation, thermal NO_x formation and fuel NO_x formation. In the prompt formation, the nitrogen in the air reacts with unburned hydrocarbons (HC) where nitrogen radicals are produced. The free radicals are reactants in further chemical reactions where NO_x is finally formed. The fuel NO_x formation can only take place if there are nitrogen atoms chemically bound in the fuel used for combustion. When the bounded nitrogen reacts with the oxygen in the air, NO_x is produced. Although there are three different ways for NO_x to be formed, the most contributing mechanism to NO_x production is the thermal NO_x formation. In this case the combustion provides sufficiently high temperatures for the stable oxygen and nitrogen molecules to react and form NO_x.[16]

A common assumption is that the NO_x formed is equal to the NO formed. This is however not completely true due to the fact that there are several reaction mechanisms that will form NO_2 molecules as well. Nevertheless, the assumption is somewhat valid and practical for further modeling of NO_x formation.[5]

3.2.2 The Zeldovich mechanism

There are many possible reaction paths where NO_x could be formed. Luckily enough Yakov Zeldovich, a Russian physicist, identified two main reaction paths, that could explain most of the NO_x formation, R3.1 and R3.2 as:

R3.1 $$N(g) + NO(g) \Leftrightarrow N_2(g) + O(g)$$

R3.2 $$NO(g) + O(g) \Leftrightarrow N(g) + O_2(g)$$

However, this was proven to be too insufficient and therefore a third reaction, here R3.3, was introduced in the extended Zeldovich mechanism.[5]

R3.3 $$NO(g) + H(g) \Leftrightarrow N(g) + OH(g)$$

For the final step in the theoretical NO_x formation modeling, some basic definitions are needed.

An arbitrary chemical product formation rate can be expressed in the following equations:

Eq. (3.9) $$-\frac{d[M_{Pi}]}{dt} = v_{Ri}(R^+ - R^-)$$

Eq. (3.10) $$R^+ = k^+ \prod_{i=1}^{m} [M_{Ri}]^{v_{Ri}}$$

Eq. (3.11) $$R^- = k^- \prod_{i=1}^{m} [M_{Pi}]^{v_{Pi}}$$

P_i and R_i corresponds to the products and reactants respectively. The $v_{Ri}$and $v_{Pi}$are the stoichiometric coefficients of products and reactants, P_i and R_i. The k^+ and k^- are the coefficients of the rates of the forward and backward reactions respectively. These coefficients can be expressed on an Arrhenius form[9] i.e.:

Eq. (3.12) $k = A(T)\mathrm{e}^{\left(-\frac{E_A}{\tilde{R}T}\right)}$

Where:

$A(T)$ is a temperature dependent constant. However, this temperature dependency is fairly weak.

E_A is the temperature dependent activation energy.

$\tilde{R}$ is the universal gas constant.

Although, there are tabulated values for the different k. The concentrations of reactants and products are noted with the $[M_{Xi}]$ – expression. The R^+ and R^- represent the corresponding forward direction and backward direction reaction rates respectively.[9]

Finally, using the notations given in Eq. (3.9) – Eq. (3.11), the NO_x formation rate can be written as:

Eq. (3.13)

$$\frac{d[NO]}{dt} = k_1^+[\mathrm{O}][\mathrm{N_2}] + k_2^+[\mathrm{N}][\mathrm{O_2}] + k_3^+[\mathrm{N}][\mathrm{OH}] - k_1^-[\mathrm{N}][\mathrm{NO}] - k_2^-[\mathrm{NO}][\mathrm{O}] - k_3^-[\mathrm{NO}][\mathrm{H}]$$

In the combustion engine, the formation of $N$is sufficiently less than the formation of other species that it is possible to neglect it completely, i.e. $\frac{d[N]}{dt} \approx 0$.

The formation rate of N, using the same notation as in Eq. (3.13) is:

Eq. (3.14)

$$\frac{d[N]}{dt} = k_1^+[\mathrm{O}][\mathrm{N_2}] - k_2^+[\mathrm{N}][\mathrm{O_2}] - k_3^+[\mathrm{N}][\mathrm{OH}] - k_1^-[\mathrm{N}][\mathrm{NO}] + k_2^-[\mathrm{NO}][\mathrm{O}] + k_3^-[\mathrm{NO}][\mathrm{H}]$$

With $\frac{d[N]}{dt} \approx 0$ the $[N]$ is given by:

Eq. (3.15) $(k_1^-[\mathrm{NO}] + k_2^+[\mathrm{O_2}] + k_3^+[\mathrm{OH}])[\mathrm{N}] = k_1^+[\mathrm{O}][\mathrm{N_2}] + k_2^-[\mathrm{NO}][\mathrm{O}] + k_3^-[\mathrm{NO}][\mathrm{H}]$

Eq. (3.16) $[\mathrm{N}] = \frac{(k_1^+[\mathrm{O}][\mathrm{N_2}] + k_2^-[\mathrm{NO}][\mathrm{O}] + k_3^-[\mathrm{NO}][\mathrm{H}])}{(k_1^-[\mathrm{NO}] + k_2^+[\mathrm{O_2}] + k_3^+[\mathrm{OH}])}$

Substitution of $[N]$ with the expression above in Eq. (3.13) results in:

Eq. (3.17) $$\frac{d[NO]}{dt} = 2k_1^+[O][N_2]\frac{1-\frac{k_1^-k_2^-[NO]^2}{k_1^+k_2^+[N_2][O_2]}}{1+\frac{k_1^-[NO]}{k_2^+[O_2]+k_3^+[OH]}}$$

The next step is to assume equilibrium concentrations of O, O_2, OH, H and N_2 at the local temperature and pressure in the post combustion zones[9]. This assumption is due to the fact that most of the NO formation takes places in the combustion zones where these conditions are met. This further implies that the backward and forward reaction rates are equal, where $[\,]_e$ denotes the equilibrium concentrations:

Eq. (3.18) $$R_1 = k_1^+[O]_e[N_2]_e = k_1^-[NO]_e[N]_e$$

Eq. (3.19) $$R_2 = k_2^+[N]_e[O_2]_e = k_2^-[NO]_e[O]_e$$

Eq. (3.20) $$R_3 = k_3^+[N]_e[OH]_e = k_3^-[NO]_e[H]_e$$

Thus Eq. (3.17) can be expressed, on an easy form, as:

Eq. (3.21) $$\frac{d[NO]}{dt} = \frac{2R_1\left\{1-\left(\frac{[NO]}{[NO]_e}\right)^2\right\}}{1+\frac{[NO]}{[NO]_e}\frac{R_1}{R_2+R_3}}$$

As can be seen in Eq. (3.18) – Eq. (3.20), the only equilibrium concentrations that have to be calculated are $[O]_e$, $[OH]_e$ and $[NO]_e$. An interesting case is when $[NO] \ll [NO]_e$. Then Eq. (3.21) can be simplified as follows:

Eq. (3.22) $$\frac{d[NO]}{dt} = 2R_1 = 2k_1^+[O]_e[N_2]_e$$

The $[O]_e$ can be written in the form of $[O]_e = K(T)\sqrt{[O_2]_e}$, which results in the expression:

Eq. (3.23) $$\frac{d[NO]}{dt} = 2k_1^+K(T)\sqrt{[O_2]_e[N_2]_e}$$

One lesson from Eq. (3.23) is that the NO_x formation is dependent on temperature, oxygen concentrations and the concentration of nitrogen. This is an expected result from the experimental research conducted by Dec et al.

If only the Original Zeldovich mechanism is used, i.e., the first two reactions, the corresponding equation to (Eq.3.22) will then be[13]:

(Eq.3.24) $$\frac{d[NO]}{dt} = \frac{15.2\cdot10^{13}e^{-\frac{38000}{T}}[O]_e[N_2]_e\left(1-\left(\frac{[NO]}{[NO]_e}\right)^2\right)}{1+50667e^{-\frac{18500}{T}}[N_2]_e\frac{[NO]}{[NO]_e{}^2}}$$

There are reasons that one would prefer the Original Zeldovich. First and foremost, the third reaction is relevant at temperature under 1400 Kelvin. Also, it is a bit simpler to implement Eq. (3.24) in C code. After all, high calculation speed is also a requirement as well as accuracy.

The $[O]_e$ and $[OH]_e$ can be calculated from[5] :

Eq. (3.23) $$[O]_e = 3.6 \cdot 10^3 \sqrt{[O_2]_e} e^{-31090/T} \frac{1}{\sqrt{RT}}$$

Eq. (3.24) $$[OH]_e = 2.129 \cdot 10^2 T^{-0.57} e^{-4595/T} \sqrt{[O]_e [H_2O]_e}$$

The equilibrium concentration of NO, $[NO]_e$, can be derived by using following reaction mechanism:

R3.4 $$N_2(g) + O_2(g) \Leftrightarrow 2NO(g)$$

This results in an equilibrium concentration:

Eq. (3.25) $$[\mathrm{NO}]_\mathrm{e} = \sqrt{\mathrm{K_{p(NO)}[O_2][N_2]}}$$

Finally, the NO_x formation rate is calculable in Eq. (3.23) and the concentration of NO is:

Eq. (3.26) $$[NO] = \int \frac{2R_1\left\{1-\left(\frac{[NO]}{[NO]_e}\right)^2\right\}}{1+\frac{[NO]}{[NO]_e}\frac{R_1}{R_2+R_3}} dt$$

The resolution is most often given in CAD and not time. To get the corresponding time step at each incremental CAD step, following substitution is done:

Eq. (3.27) $$\phi = \phi(\mathrm{t}) \text{ and } \frac{\mathrm{d}\phi}{\mathrm{dt}} = \phi'(\mathrm{t})$$

Eq. (3.28) $$dt = d\phi \frac{\pi}{180} \cdot \frac{1}{\omega}$$

The function $\phi(t)$ is dependent on time through the rate per minute and the ω is the angular velocity. It is now possible to calculate the NO_x formation rate and thus the NO_x concentration for each time step taken or rather, for each new change in crank angle degree.

3.3 The two zone combustion

The theory above is not enough for a good model of the NO_x formation rate. Combustion has to be modeled in a manner that is sufficient for estimations but also for implementation purposes. In the two zone approach to combustion, the zero dimensional model assumes two separate zones during the combustion. In the first zone, Zone 1 (burned zone), combustion takes place and NO_x is formed. The second zone, Zone 2 (unburned zone), is assumed to only consist of unburned gas where no NO_x formation occurs. To maintain a constant λ, mass flows from Zone 2 to Zone 1 and except this mass flow, masses are maintained in both zones. What in fact does transfer is the heat released by the combustion in Zone 1. Optionally, even these transfers can be neglected as well, due to calculation time issues, in a more simplified model of the combustion process.

The burned zone is however split up in two additional subzones, as illustrated in Fig. 3.2. In the first subzone, the combustion zone, fuel and air reacts in a combustion reaction and forms the products H_2O, O_2, N_2, CO_2, OH and CO. These products are directly added to the second subzone, the post combustion zone. Here the newly formed products are added to the products from combustion at previous time steps. It is in fact in this zone that NO_x is assumed to form. The combustion zone is merely there to describe the heat release and temperature increase, but the zone that utilize the Zeldovich mechanism is in fact the post combustion zone. This decoupling of NO_x formation and combustion greatly simplifies the modeling process and also the code implementation procedure.

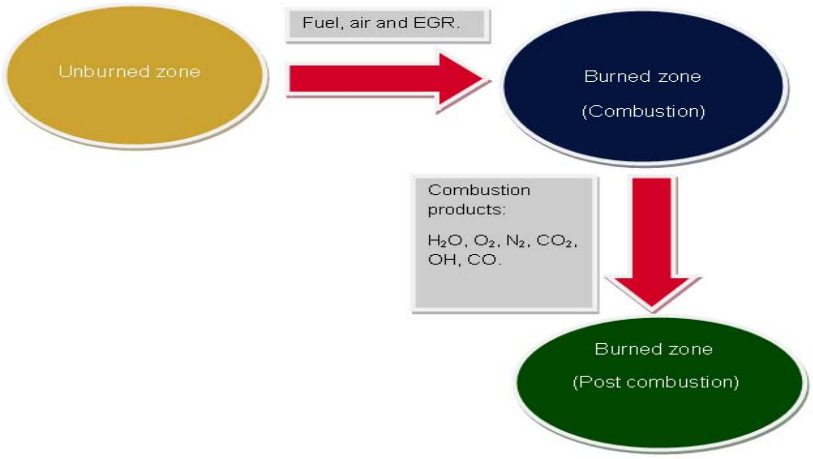

Fig. 3.2 – The concept of the two–zone model with an additional subzone, the post combustion zone.

This is however still not enough for a useful combustion model, and therefore some other assumptions and statements are added accordingly[9]:

(1) Gases in the cylinder can be treated according to the ideal gas law.
(2) The sum of the volumes of Zone 1 and Zone 2 has to equal the volume of the cylinder.
(3) Temperature differences between Zone 1 and Zone 2 depend solely on ROHR and a difference in heat transfers to the cylinder walls, due to different temperature gradients.
(4) The heat losses to cylinder walls can be described with the Woschni heat[17] transfer equation as stated:

Eq. (3.29) $$\frac{dQ_{hl}}{dt} = h \cdot A_{wall}(\phi)(T_{zone\,1}(\phi) - T_{wall}(\phi))$$

Eq. (3.30) $$h = 3.26B^{-0.2}p^{\,0.8}T^{-0.55}w^{0.8}$$

p = cylinder pressure
w = average gas velocity
T = mean gas temperature
h = heat transfer coefficient
B = cylinder bore
A_{wall}= cylinder surface area

(5) The pressure is uniform in the cylinder. This means that Zone 1 and Zone 2 has exactly the same pressure at a given time.
(6) The temperature is the same in both zones, just before the injection of fuel and combustion.
(7) Concentrations of residual gases are the same in both zones.

The burned fuel mass is given in Eq. (3.8). The lambda value in Zone 1 is known and therefore the total mass in the zone can be calculated:

Eq. (3.31) $m_{Zone\,1} = (\lambda_{zone\,1} \cdot \psi + 1)m_{fuel}(\phi)$, ψ = stoichiometric air-to-fuel ratio

The temperatures in Zone 1 and Zone 2 also changes due to compression and expansion. Using the ideal gas law and the fact that the product pV^n is constant, an expression for ΔT_c , changes due to compression, and ΔT_e ,changes due to expansion, is derived.

The temperature change due to compression (or expansion) is given by the relation[13]:

Eq. (3.32) $\frac{T_{n+1}}{T_n} = \left(\frac{V_n}{V_{n+1}}\right)^{\gamma-1}$

This implies a ΔT due to isentropic compression (expansion):

Eq. (3.33) $\Delta T = T_{n+1} - T_n = (\left(\frac{V_n}{V_{n+1}}\right)^{\gamma-1} - 1)T_n$

There is also a temperature change because of the combustion itself. The $Q_{combustion}$ was previously derived. However, this will not suffice, because one has to take the heat losses into account. This will give a $\frac{dQ_{net}}{d\varphi}$:

Eq. (3.34) $\frac{dQ_{net}}{d\varphi} = \frac{dQ_{combustion}}{d\varphi} - \frac{dQ_{hl}}{d\phi}$

And the temperature change given by the ROHR is (where the air can be diluted with EGR):

Eq. (3.35) $\Delta T_{ROHR} = \frac{1}{m} Q_{net} \frac{1}{c_p}$

The heat transfer from the warmer gas in Zone 1 to the colder Zone 2 contributes to a temperature gradient from the first zone to the second zone.

Eq. (3.36) $\Delta T_{zones} = T_{zone\,1}(\phi) - T_{zone\,2}(\phi) = B(\phi)A^*$

Where

Eq. (3.37) $B(\phi) = \frac{K - \int_{\phi SOC}^{\phi}(p(\phi) - p_0(\phi))m_{Zone\,1}(\phi)d\phi}{K}$

$$K = \int_{\phi SOC}^{\phi EVO}(p(\phi) - p_0(\phi))m_{Zone\,1}(\phi)d\phi$$

The variable A^* is the maximum temperature difference between the two zones and $p_0(\phi)$ is the pressure of a motored cycle at CAD ϕ. *SOC* and *EVO* are the "Start of combustion" and "Exhaust valve opens" respectively.[9] Although, this approach is rather gruesome and therefore another path can be taken, if there is a desire to include heat transfers between zones in the model. The alternative calculation path is derived in section 3.5.8 (see below).

Finally, the temperature in Zone 1 can be calculated at each ϕ, the mass of burned fuel and air per ϕ is now known, and the pressure can be measured. The volume of Zone 1 is then given by the ideal gas law as:

$$V_{zone\,1} = \frac{m(\phi)\tilde{R}T(\phi)}{p(\phi)}$$

This volume is used for calculations of gas concentrations in Zone 1, where the NO_x formation is occurring.

3.4 Stoichiometric calculations

The chemical reaction of combustion in a diesel engine is given by[18]:

R3.5 $$C_aH_bO_c + \lambda\left(a + \frac{b}{4} - \frac{c}{2}\right)(O_2 + 3.773N_2) \rightarrow aCO_2 + \frac{b}{2}H_2O + (\lambda - 1)\left(a + \frac{b}{4} - \frac{c}{2}\right)O_2 + \left(a + \frac{b}{4} - \frac{c}{2}\right)3.773\lambda N_2$$

In some cases there is also an EGR function in the diesel engine that will have an impact on the reaction mechanism. However, this is a fairly simple addition to R3.5:

R3.6 $$C_aH_bO_c + \lambda(EGR_{H_2O}H_2O + EGR_{O_2}O_2 + EGR_{N_2}N_2 + EGR_{CO_2}CO_2 + \left(a + \frac{b}{4} - \frac{c}{2}\right)(O_2 + 3.773N_2)) \rightarrow (a + \lambda EGR_{CO_2})CO_2 + \left(\frac{b}{2} + \lambda EGR_{H_2O}\right)H_2O + (\lambda - 1)\left(a + \frac{b}{4} - \frac{c}{2}\right)O_2 + \left(a + \frac{b}{4} - \frac{c}{2}\right) \cdot 3.773\lambda N_2 + \lambda EGR_{O_2}O_2 + \lambda EGR_{N_2}N_2$$

The amount of substance of the fuel, $C_aH_bO_c$, is:

Eq. (3.38) $$n_{C_aH_bO_c} = \frac{m_{fuel}}{M_{C_aH_bO_c}}$$

The $M_{C_aH_bO_c}$ is the specific molar mass of the fuel and the mass of fuel burnt is previously known. Every concentration needed in Eq. (3.21) can now be calculated and used to determine the NO_x formation at each CAD−step, following the assumption that λ is constant in Zone 1. The content of the combustion zone is directly added to the post combustion zone (see section 2.4). The post combustion content produced at the current time step is added to the sum of contents produced in the previous time steps and that will result in the total post combustion zone at current time. Despite of this, some considerations need to be made regarding the temperature change when fresh combustion products are added, and this is due to a difference in temperatures between the contents. When the temperatures are determined it is possible to do simple calculations of the different combustion product concentrations needed in the NO_x formation rate formula, assuming homogeneous mixtures in the zones.

3.5 Calculations and estimations of parameters

3.5.1 Estimating γ and c_p

To successfully implement the above theory in algorithms, parameters have to be known in advance. For example the γ in the Rate of Heat Release–equation is not trivial to determine at all. One way is to omit heat transfers and mass flows during the adiabatic compression phase.[3] The heat release equation can then be simplified as follows:

Eq. (3.39) $\frac{dQ_{combustion}}{d\phi} = 0 \quad \text{and} \quad \frac{dQ_{hl}}{d\phi} = 0$

Thus

Eq. (3.40) $\frac{\gamma}{\gamma - 1} p \frac{dV}{d\phi} + \frac{1}{\gamma - 1} V \frac{dp}{d\phi} = 0$

This gives an estimate of γ:

Eq. (3.41) $\overline{\gamma} = -\frac{V}{dV}\frac{dp}{p}$

The combustion and expansion stroke will of course change the γ –estimate. Assuming small deviations, this impact can be neglected.

The parameter c_p, needed in temperature calculations, can be calculated using NASA Glenn tabulation of specific heats[19]. Another approach to calculate the specific heat is to utilize the estimate of γ. First and foremost the specific ideal gas law constant can be stated as:

Eq. (3.42) $\overline{R} = \frac{R}{M} = c_p - c_v$

Where R is the ideal gas constant and M is the molar mass for the mixture of gases. Further, the ratio between c_p and c_v is known through the γ – estimate.

Eq. (3.43) $\overline{\gamma} = \frac{c_p}{c_v}$

Substitution of c_v in Eq. (3.42) gives:

Eq. (3.44) $\overline{c}_p = \frac{R}{M_{Zone1}(1 - \frac{1}{\overline{\gamma}})}$

The downside is that the approach assumes a fairly good estimate of γ and the mass contained in Zone 1, which generally cannot be guaranteed. Therefore, this method is neglected and not presented in the result section. It is much more convenient to use NASA tabulations and interpolations of thermo–chemical data.

3.5.2 Calculation of molar masses and SAFR

The $M_{zone\,1}$ is the molar mass of Zone 1. This of course is also a parameter that has to be calculated on beforehand. One fitting way is as follows:

Eq. (3.45) $$M_{zone\,1} = \frac{m_{fuel}}{m_{zone\,1}} M_{fuel} + \left(1 - \frac{m_{fuel}}{m_{zone\,1}}\right) M_{air}$$

The mass of Zone 1 is previously known, given by Eq. (3.31) and thus this result can be used in Eq. (3.44). Additionally, a calibration parameter could be added to account for EGR. The M_{fuel}, M_{air} and $SAFR$ can be calculated with some help from R3.5 and R3.6 as:

Eq. (3.46) $$M_{fuel} = A_C a + A_H b + A_O c$$

Eq. (3.47) $$M_{air} = \frac{(3.773 \cdot 2A_N + 2A_O)}{3.773+1}$$

A_x is the atomic mass of atom X.

This will give an air molar mass and a $SAFR$:

Eq. (3.48) $$M_{air} = \frac{3.773 \cdot 2 \cdot 14.1 + 2 \cdot 16}{3.773+1} = 28.96\ g/mol$$

Eq. (3.49) $$SAFR = \frac{\left(a + \frac{b}{4} - \frac{c}{2}\right)}{M_{fuel}} M_{air}$$

3.5.3 The average gas velocity estimate

The average gas velocity w, needed in Eq. (3.30) can be estimated by calculating[12]:

Eq. (3.50) $$\overline{w} = \left[C_1 \overline{S}_p + C_2 \frac{V_d T_{ivc}}{p_{ivc} V_{ivc}} (p - p_m)\right]$$

The variable $\overline{S}_p$ is the mean piston speed while V_d is the volume displacement. Also, the T_{ivc}, p_{ivc} and V_{ivc} is the temperature, pressure and volume at valve inlet closing respectively. Pressure p is the current pressure in the cylinder and p_m is the motored cylinder pressure, at the same CAD as pressure p. C_1 and C_2 are empirical constants. To simplify the estimate of w, the second term is omitted in this thesis. Thus the estimate of w is:

Eq. (3.51) $$\overline{w} = C_1 \overline{S}_p$$

Where the constant C_1 can be determined emperically[9]:

$C_1 = 6.18$ at the gas exchange period

$C_1 = 2.28$ at compression, expansion and combustion period

3.5.4 Approximation of equilibrium concentrations

When calculating the radical concentrations in Eq. (3.23) and Eq. (3.24), one assumption has to be made. The equations are essentially derived from the equilibrium equations:

Eq. (3.52) $\frac{[O]_e{}^2}{[O_2]_e} = K_{p(O)}$

Eq. (3.53) $\frac{[OH]_e^2}{[O]_e[H_2O]_e} = K_{p(OH)}$

The amount of substances of oxygen and water is known through equilibrium calculations (see section 3.6) and the volume is known from the ideal gas law. The concentrations at equilibrium for Eq. (3.52) will be, by applying basic chemistry:

Eq. (3.54) $\frac{\left(\frac{x}{V_{burnt}}\right)^2}{\left(\frac{n_{oxygen}-x}{V_{burnt}}\right)} = K_{p(O)}$

The following approximation is made, based on the fact that the amount of substance of the oxygen radical is far less than the amount of substance of the oxygen:

Eq. (3.55) $n_{oxygen} - x \approx n_{oxygen}$

The same assumption is made for Eq. (3.53), i.e. the amount of substance of the OH radical is less than the same, for oxygen radicals and water molecules.

3.5.5 A description of the EGR dynamics

If the reaction mechanism in R3.6 ought to be useful, the EGR inflow has to be known. The approach to solve this problem is to establish a model for mass flows. One way to do so is described in Fig. 3.3.

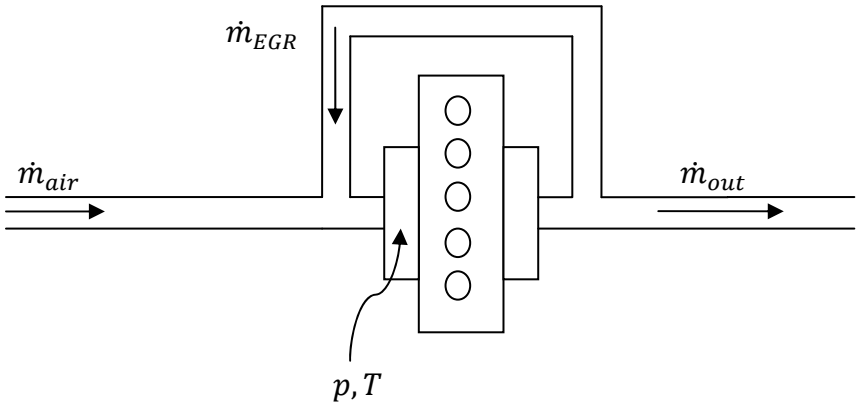

Fig. 3.3 – Schematic picture of gaseous mass flows to the engine. The variables p and T are directly measurable.

The intake pressure and temperature is measured and the ideal gas law is applicable as long as most of the H_2O in the EGR is not liquefied in too large quantities. The equation of interest, in this case, is[3]:

Eq. (3.56) $\dot{m}_a = \rho \cdot V_d \cdot \frac{N}{2} \cdot \mu_v$

Where ρ is the density, V_d is the engine displacement, μ_v is the volumetric efficiency degree, the $N$is the rate per minute and the $\dot{m}_a$ is the intake mass flow.

Although, the problem is that the density is unknown due to the fact that air is mixed with EGR. A suitable trick is simply to rewrite Eq. (3.56) as:

Eq. (3.57) $\frac{\dot{m}_a}{\rho} = \dot{V}_{intake} = V_d \cdot \frac{N}{2} \cdot \mu_v$

Now, using the time derivative of the ideal gas law gives (at constant inlet pressure and temperature):

Eq. (3.57) $p\dot{V} = \dot{n}_{intake} RT$

And finally,

Eq. (3.58) $\dot{n}_{intake} = p \cdot V_d \cdot \frac{N}{2} \cdot \frac{\mu_v}{RT}$

Knowing that the air mass flow is measured, the $\dot{n}_{EGR}$ can be calculated:

Eq. (3.59) $\dot{n}_{EGR} = \dot{n}_{intake} - \frac{\dot{m}_{air}}{M_{air}}$

The amount of substance of the EGR in the combustion zone can easily be determined using assumption (7) in the two–zone model (see section 3.3).

Eq. (3.60) $n_{EGR\ comb\ zone} = \frac{n_{EGR}}{n_{air} + n_{EGR}} n_{air\ comb\ zone}$

The parameter $n_{air\ comb\ zone}$ is given by the reaction mechanism in R3.6. It should be noted that the fractions of carbon dioxide and water in the recirculated exhaust gas need to be determined. However, this is a trivial task that is possible to solve by calculations of particular fractions in the exhaust gas in R3.5. This possibility is based on the fact that the water and carbon dioxide in the EGR originates from the combustion itself. The combustion reactions are fully described by the reaction mechanism in R3.5. The only thing that has to be known is how much air is added during the intake stroke.

It is possible to calculate the amount of air taken in by integration:

Eq. (3.61) $n_{intake\ of\ air} = \int \frac{\dot{m}_{air}}{M_{air}} dt = (1 - EGR\%)\eta_v V_d$

$\eta_v V_d$ is the total intake volume at a volumetric efficiency, η_v.

The proportion between oxygen and nitrogen in the air is 1:3.773, as can be seen in reaction R3.6.

One last step in the EGR is to understand the coefficients EGR_{H_2O}, EGR_{CO_2}, EGR_{O_2} and EGR_{N_2}. A percentage of the combustion products in R3.5 will be in the EGR, this percentage will be labeled EGR%. Hence, if 20% of the exhaust gas is re–circulated then $EGR\% = 20\%$. Given this information, it is possible to quantify the coefficients during different running conditions, where the injection of fuel may very well vary.

The EGR dynamics are modeled assuming a homogeneous EGR–tank, where a percentage of the exhaust gases enter. The composition of the EGR is fully explained by the reaction mechanism in R 3.5. The local λ value will still be maintained around 1, which is a necessity in this NO_x model (see section 3.3). Hence, the resulting principal equation in the combustion zone (see Fig. 3.1), during stationary conditions, will be:

Eq. (3.62)

$$\left((1 - EGR\%)\lambda_{local} \cdot AIR + FUEL\right) + EGR\% \cdot \lambda_{local} \cdot EXHAUST\ GAS = EXHAUST\ GAS$$

It is although rarely the case that the fuel consumption remains unchanged. The variation in fuel consumption will render different carbon dioxide, water and oxygen concentrations. However, treating the EGR tank as a well stirred tank of gas, this does not introduce any calculation problems. The amount of substances and, assuming a well stirred tank, concentrations can still be calculated, utilizing a global lambda λ_g. Oxygen in the EGR is assumed not to react with fuel. This is purely a mathematical trick due to a desire for algebraic simplicity.

Eq. (3.63)

$$\begin{aligned}
&(C_aH_bO_c + (1 - EGR\%)\lambda_{local}(a + b/4 - c/2)(O_2 + 3.773N_2)) + EGR\% \cdot \lambda_{local} \\
&\quad \cdot \left((\lambda_g - 1)\left(a + \frac{b}{4} - \frac{c}{2}\right)O_2 + 3.773\lambda_g N_2 + aCO_2 + \frac{b}{2}H_2O\right) \\
&\quad = (1 - EGR\%) \cdot \left(aCO_2 + \frac{b}{2}H_2O + (\lambda_{local} - 1)\left(a + \frac{b}{4} - \frac{c}{2}\right)O_2\right. \\
&\quad \left. + \left(a + \frac{b}{4} - \frac{c}{2}\right)3.773\lambda_{local}N_2\right) + EGR\% \\
&\quad \cdot \lambda_{local}\left((\lambda_{global} - 1)\left(a + \frac{b}{4} - \frac{c}{2}\right)O_2 + \lambda_g 3.773N_2 + aCO_2 + \frac{b}{2}H_2O\right)
\end{aligned}$$

A comment is needed on the local lambda. In a case where no EGR is added, the lambda is easy to understand. If EGR is added, the assumption in the theoretical model is altered. As previously stated, the EGR has both a thermal effect and an effect on the oxygen concentration. In Eq. (3.63) this is clearly seen as a de facto decrease in the amount of substance ratio between oxygen and the fuel.

This could also be seen as a change in local combustion lambda by a factor $(1 - EGR\%) + EGR\%(\lambda_{global} - 1)$, which will result in a oxygen concentration decrease.

The EGR will have an influence on the oxygen concentration and specific heat value during the combustion phase. The easiest way to account for EGR influence on the specific heat value is to calculate the specific heat capacities for each species in the gas mixture. The average specific heat for the gas mixture can then be calculated as:

Eq. (3.64) $$c_P = \frac{1}{n_{Zone\,1}}(c_{pH2O}n_{H2O} + c_{pN2}n_{N2} + c_{pO2}n_{O2} + c_{PCO2}n_{CO2})$$

The specific heat of each species is calculated using NASA Glenn interpolations of thermo dynamical data.

The exhaust gas content, produced each cycle, is added to the EGR tank in following way, if seen as a function of time:

Eq. (3.65) $$n_{total}(T) = n(T) + \sum_{k=1}^{T-1} n_{old}\,(k)$$

The equilibrium concentrations, relevant for NO_x formation, can be determined using the fact that the split up of the EGR, in the two different zones, will be:

Eq. (3.66) $$n_{EGR\ zone\ 1} = \frac{1}{\lambda_g} n_{EGR}$$

Eq. (3.67) $$n_{EGR\ zone\ 2} = \frac{\lambda_g - 1}{\lambda_g} n_{EGR}$$

Now the equilibrium concentrations in the burned zone can be calculated in diesel engines that utilize an inner EGR function.

What is not seen in Eq. (3.63) is a delay due to the fact that it takes some time for the recirculated gases to really recirculate. This fact is hard to formulate using chemical nomenclature, but mathematically and algorithmically this is fairly easy, assuming a well stirred tank of EGR, at a mean global lambda $\bar{\lambda}_g$.

3.5.6 The cylinder geometry

The cylinder geometry will be of major importance in the model. The volume, volume angle derivative and the surface area are, respectively[18]:

Eq. (3.68) $$V_{cyl} = V_C + \frac{V_d}{2}\sin\theta\left(R + 1 - \cos\theta - \sqrt{R^2 - \sin^2\theta}\right)$$

Eq. (3.69) $$\frac{dV_{cyl}}{d\theta} = \frac{V_d}{2}\sin\theta\,(1 + \frac{\cos\theta}{\sqrt{R^2 - \sin^2\theta}})$$

Eq. (3.70) $$S_{cyl} = 2\frac{\pi B^2}{4} + \frac{\pi BL}{2}\left(R + 1 - \cos\theta - \sqrt{R^2 - \sin^2\theta}\right)$$

B is the bore diameter, R is the ratio between the conrod length and half the stroke. V_C and V_d are compression and displacement volume respectively.

The piston position and velocity will be functions of the CAD. These relations are given, with some approximations[18], by following equations:

Eq. (3.71) $$h(\varphi) = r_0(1 - \cos(\varphi) + 2\frac{r_0}{l}\sin^2(\varphi))$$

Eq. (3.72) $$\frac{dh(\varphi)}{dt} = r_0\omega(\sin(\varphi) + 2\frac{r_0}{l}\sin(2\varphi))$$

The r_0 and l are the crank radius and rod length respectively. The angular velocity ω is a function of the rate per minute, where f is the frequency and δ is the angular velocity in rpm:

Eq. (3.73) $$\omega = 2\pi f = 2\pi \cdot \frac{\delta}{60}$$

With the above equations, the geometry of the cylinder and the cylinder motion are completely described.

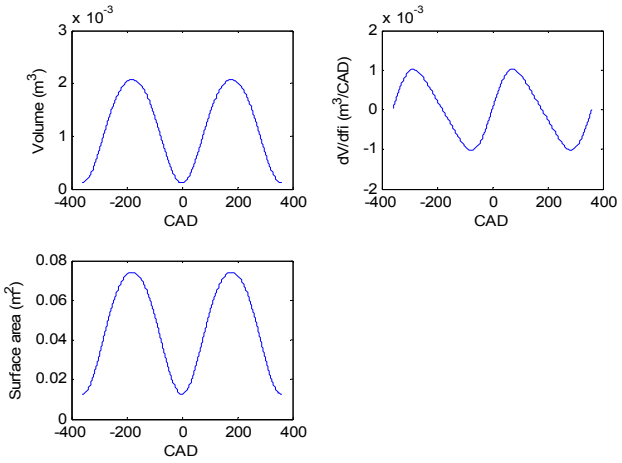

Fig. 3.4 – Geometric calculations with the parameter values:

$$B = 127\ mm,\ L = 154\ mm,\ r = \frac{L}{2}\ and\ l = 255\ mm.$$

3.5.7 Mass relation between the two zones

One interesting issue is the contribution of the unburned zone to the burned zone in terms of mass flow. First of all, the initial amount of gas in the cylinder before compression is:

Eq. (3.74) $n_{init} = \frac{p_{init} V_{init}}{R T_{init}}$

Mass flows from the unburned zone to the burned zone in a way that keeps the local λ constant at ≈ 1. The relation between the amount of substances in Zone 1 (burned zone) is:

Eq. (3.75) $n_{zone\ 1,EGR+air} = \psi_n \cdot \lambda_1 \cdot n_{fuel}$

The equivalent relation in Zone 2, the unburned zone, is:

Eq. (3.76) $n_{air\ zone\ 2} + n_{EGR\ zone\ 2} = n_{init} - n_{zone\ 1,EGR+air}$

Rewriting Eq. (3.76) by using Eq. (3.75) gives the amount of air in the unburned zone, zone 2:

Eq. (3.77) $n_{air\ zone\ 2} = n_{init} - n_{EGR\ zone\ 2} - \psi_n \cdot \lambda_1 \cdot n_{fuel}$

The EGR in Zone 2 is easy to calculate when statement (7) is applied (see section 3.3 above). The total substance of EGR is known from integration of Eq. (3.59), thus:

Eq. (3.78) $n_{air\ zone\ 2} = n_{init} - \frac{V_{uz}}{V_{total}} n_{EGR} - \psi_n \cdot \lambda_1 \cdot n_{fuel}$

Where V_{uz} is the volume of the unburned zone and the V_{total} is the total volume at a particular CAD. The volume of the burned zone, V_{bz}, was previously calculated from the ideal gas law. The fact that volume is conserved gives:

Eq. (3.79) $n_{air\ zone\ 2} = n_{init} - \frac{V_{total} - V_{bz}}{V_{total}} n_{EGR} - \psi_n \cdot \lambda_1 \cdot n_{fuel}$

Eq. (3.80) $n_{air\ zone\ 2} = n_{init} - \left(1 - \frac{V_{bz}}{V_{total}}\right) n_{EGR} - \psi_n \cdot \lambda_1 \cdot n_{fuel}$

All the variables on the right side of Eq. (3.80) are known through previous calculations. The mass flow from the unburned zone to the burned zone is now completely known. The amount of air, relevant for the NO_x formation, can thus be calculated, assuming 21% oxygen in the air content.

A comment on ψ_n is needed in this case. The regular ψ takes air/fuel mass ratio into account. However, ψ_n concerns the equivalent amount of substances instead.

Nevertheless, ψ_n is simple to calculate from the reaction formula stated in R3.5:

Eq. (3.81) $\psi_n = \frac{n_{air}}{n_{fuel}} = \frac{\frac{m_{air}}{M_{air}}}{\frac{m_{fuel}}{M_{fuel}}} = \frac{a + \frac{b}{4} - \frac{c}{2}}{1} = a + \frac{b}{4} - \frac{c}{2}$

3.5.8 Heat transfer between zones

One of the last remaining puzzle pieces in the modeling part is the heat transfer from the burned zone to the unburned zone. The Eq. (3.37) and Eq. (3.38) are not on a desirable form for an FGPA implementation. The heat transfer can instead be calculated from the fact that the unburned zone, in a normal case, would get a temperature rise from compression only. The isentropic compression would result in following, with a resulting temperature T_c and compression pressure p_c:

Eq. (3.82) $T_c = \frac{p_c V_{uz}}{n_{uz} R}$

However, the result will be different because of the heat release in the burned zone and a pressure rise will occur:

Eq. (3.83) $T_{real} = \frac{p_{real} V_{uz}}{n_{uz} R}$

The temperature difference will then be:

Eq. (3.84) $\Delta T = T_{real} - T_{compression} = (p_{real} - p_c) \frac{V_{uz}}{n_{uz} R}$

In fact, this equation can be simplified even further by realizing that n_{ubz} is $n_{uz} = \frac{V_{uz}}{V_{total}} n_{init}$ (see Eq. 3.74).

Eq. (3.85) $\Delta T = \frac{(p_{real} - p_c) V_{total}}{n_{inital} R} = constant \cdot V(\varphi) \cdot \Delta p$

The amount of heat that is needed to warm up the content in the unburned zone is thus:

Eq. (3.86) $Q = m_{uz} c_{P,uz} \Delta T$

If the EGR contribution is not neglected (i.e. the air is diluted with combustion products from previous cycles), the mass in the unburned zone is approximated and the c_p, specific heat at constant pressure, of air is calculated using tabulated specific heats. If one would want to calibrate the model due to the approximations, especially of global lambda, an empirical constant could be added in Eq. (3.86):

Eq. (3.87) $Q = K_{cal} m_{uz} c_P \Delta T$

The parameter K_{cal} would have to be empirically tabulated for a successful implementation, or determined by NASA tabulation of specific heats. Because of this tuning problem and the fact that the calculation would be delayed one step due to an unknown unburned mass, a common assumption in NO_x modeling is to omit heat transfers between zones. Instead the unburned gases are assumed to change temperature solely due to isentropic compression and expansion. This assumption implies $p_{real} = p_c$.

3.6 Calculating equilibrium concentrations

The combustion products in R3.5 will, unfortunately, not be the only ones. It is fairly well known that combustion also produces carbon monoxide in the gas water shift reaction. The two following reactions have a huge impact on the concentrations in the post combustion zone[9]:

R3.7 $CO + H_2O \Leftrightarrow H_2 + CO_2$

R3.8 $CO + \frac{1}{2}O_2 \Leftrightarrow CO_2$

The equilibrium, for R3.7 and R3.8, will be at:

Eq. (3.88) $K_{P1} = \frac{P_{CO_2} P_{H_2}}{P_{CO} P_{H_2O}}$

Eq. (3.89) $K_{P2} = \frac{P_{CO_2}}{P_{CO}\sqrt{P_{O_2}}}$

If the amount of substance, in relation to one mole fuel, of CO_2, CO, H_2O, H_2, O_2 and N_2 are called a_1, a_2, a_3, a_4, a_5 and a_6, respectively, a system of nonlinear equations can be derived, using normalized pressure. If Eq. (3.88), Eq. (3.89) and atom balances are utilized, the following nonlinear equations will hold to be true, where P is the normalized pressure[9]:

Eq. (3.90) $K_{P1} = \frac{a_1 a_4}{a_2 a_3}$

Eq. (3.91) $K_{P2}{}^2 = \left(\frac{a_1}{a_2}\right)^2 \left(\frac{1}{a_5}\right) \frac{\sum_{j=1}^{6} a_j}{P}$

Eq. (3.92) $a = a_1 + a_2$

Eq. (3.93) $b = 2(a_3 + a_4)$

Eq. (3.94) $2\left(n + \frac{m}{4}\right)\lambda = 2a_1 + a_2 + a_3 + 2a_5$

Eq. (3.95) $\left(n + \frac{m}{4}\right)\lambda = \frac{a_6}{3.773}$

The equations are simple to solve with for example the *fsolve* function in Matlab. However, if the objective is to implement an algorithm in an embedded system, then the fsolve function is not a suitable approach. In this thesis a modified Newton–Raphson method was successfully derived. First and foremost Eq. (3.90) and Eq. (3.91) were slightly rewritten:

Eq. (3.96) $K_{P1} a_2 a_3 = a_1 a_4$

Eq. (3.97) $K_{P2}{}^2 P a_2^2 a_5 = a_1^2 \sum_{j=1}^{6} a_j$

Five functions were then defined and differentiated symbolically to form the Jacobi matrix ($\bar{a} = (a_1, a_2, a_3, a_4, a_5)$, fuel $C_a H_b$ and $a_6 = N$) :

Eq. (3.98) $f_1(\bar{a}) = K_{P1} a_2 a_3 - a_1 a_4$

Eq. (3.99) $f_2(\bar{a}) = K_{P2}{}^2 P a_2^2 a_5 - a_1^2 \sum_{j=1}^{5} a_j - a_1^2 N$

Eq. (3.100) $f_3(\bar{a}) = a - (a_1 + a_2)$

Eq. (3.101) $f_4(\bar{a}) = b - 2(a_3 + a_4)$

Eq. (3.102) $f_5(\bar{a}) = 2\left(n + \frac{m}{4}\right)\lambda - 2a_1 + a_2 + a_3 + 2a_5$

For practical reasons, the resulting Jacobi is omitted in this report (see Matlab code instead). The resulting algorithm is given as:

Eq. (3.103) $\bar{a}_{n+1} = \bar{a}_n - \beta J^{-1} F(\bar{a}_n)$

The β term, a relaxing square matrix, is needed because of the often large magnitudes involved, especially in Eq. (3.98). A unit matrix would cause instability and the values would diverge to infinity, no matter how close the initial guess is set to the true value. β is added to guarantee stability, therefore two values in β were set to be substantially less than 1. This does inflict some convergence rate issues. However, there are reasons to believe that the above algorithm can be improved substantially by fitting β at certain points of interest. Another possible approach would be to make β dependent on the magnitude of $J^{-1} F(\bar{a}_n)$.

The equilibrium constants, K_{p1} and K_{p2}, are calculated with:

Eq. (3.104) $\ln K_p = \frac{\sum_i v_i G_i{}^0}{RT}$

The G_i^0 is the Gibbs free energy for species *i*, at atmospheric pressure (1 bar). These values are calculated at each temperature using tabulated interpolations of entropy and enthalpy, that can be found in NASA Glenn thermo chemical

tables.[19] The temperature range of interest, in the burned zone, is around 1500−3000 K.

The Gibbs free energy can then be calculated as:

Eq. (3.105) $G(T) = H - ST$

S is the entropy and H is the enthalpy of the species.

In the case of K_{p1}, another interpolation[4] can be used:

Eq. (3.106) $\ln K_{p1}(T) = 2.743 - 1.761 \cdot \frac{10^3}{T} - 1.611 \cdot \frac{10^6}{T^2} + 0.2803 \cdot \frac{10^9}{T^3}$

Now the system described by Eq. (3.90)–Eq. (3.95) can be solved and the equilibrium substances determined, when n_{fuel} is known.

3.7 Adding products to the post combustion zone

Each time step the combustion products produced at the current time are added to the combustion products that were produced at previous time steps. This can be stated as:

Eq. (3.107) $M_{sum} = \sum_{i=0}^{t-1} M_i + M_t$

M_{sum} is the mass of the total amount of combustion products, M_t is the added contribution at the current time step, t. This will of course have an impact on the temperature of the content. The new and warmer product mass, M_t, will heat up the rest of the post combustion zone.

The calculations can be rather complicated if an assumption is not made about the c_p values in question. For simplicity the c_p is assumed to be the same for the content given by $\sum_{i=0}^{t-1} M_i$ and the content in the mass M_t. When the heat exchange in the post combustion zone has been considered, the ideal gas law will be applicable. The resulting temperature, when the new content is added to the combustion zone, is:

Eq. (3.108) $T_{total\ content} = T_{old\ content}\left(\frac{n_{old\ content}}{n_{total}}\right) + T_{new\ content}\left(\frac{n_{new\ content}}{n_{total}}\right)$

The old content will be cooled down due to isentropic expansion during the combustion and expansion phase. This is of course an approximation of the physical reality which assumes that there are no heat losses to the unburned charge.

A good question to ask oneself is if it is reasonable to assume only one burned zone in the NO_x model. What if one would assume a new separate zone at each additional combustion step taken (CAD–step), instead of only one where new components are simply added to the old ones? Under the assumption that the pressure is instantaneously uniform in the cylinder and N zones (N is equal to the amount of time steps between start and end of the heat release) are used, a calculation can be done. Application of the gas law, at a pressure p and at time t_c, gives:

Eq. (3.109) $p\sum_{i=1}^{N} V_i = pV_{bz}$

V_{bz} is the total volume of the burned zones created at time t_c.

Rewriting Eq. (3.109) gives the equation:

Eq. (3.110) $\sum_{i=1}^{N} n_i R T_i = n_{bz} R T_{bz}$

The mean burned zone temperature at step N, $T_{bz,N}$, is then:

Eq. (3.111) $T_{bz,N} = \frac{1}{n_{bz}} \sum_{i=1}^{N} n_i T_i$

Remembering that there is a temperature drop (increase) due to isentropic expansion (compression), the temperature of an element i combusted at time t_i , T_{i,t_i}, can be expressed:

Eq. (3.112) $T_{i,t_i} = \prod_{t=t_i+\Delta t}^{t_c} \left(\frac{p_t}{p_{t-1}}\right)^{\frac{\gamma_t - 1}{\gamma_t}} T_{i,initial}$

Inserted in Eq. (3.111) this renders a $T_{bz,N}$:

Eq. (3.113) $T_{bz,N} = \frac{1}{n_{bz}} \sum_{i=1}^{N} n_i \prod_{t=t_i+\Delta t}^{t_c} \left(\frac{p_t}{p_{t-1}}\right)^{\frac{\gamma_t - 1}{\gamma_t}} T_{i,initial}$

At time $t_c + \Delta t$ a new zone is created, and when the gas law is applied the result will be a recognizable T_{bz}:

Eq. (3.114) $T_{bz,N+1} = \frac{\Delta n}{n_{bz}+\Delta n} T_{N+1} + \frac{1}{n_{bz}+\Delta n} \sum_{i=1}^{N} n_i \prod_{t=t_i+\Delta t}^{t_c+\Delta t} \left(\frac{p_t}{p_{t-1}}\right)^{\frac{\gamma_t - 1}{\gamma_t}} T_{i,initial}$

This expression is in fact very similar to the one used in the single burned zone model, Eq. (3.108), considering that the initial condition, for the temperature of the first element combusted, is:

Eq. (3.115) $T_{bz,N=1} = T_{1,initial}$

Hence, if the pressure is uniform throughout the cylinder and combustion occurs at a constant or near–constant local lambda, nothing essential can be gained by keeping the zones apart, except for a higher resolution of the temperature distribution. Thus, the single burned zone used in this thesis is fairly relevant and also easy to implement and keep track of.

This does however not imply that a multi–zone model is worthless, even if the pressure is uniform and lambda constant. Because, by studying Eq. (3.26), one can see that the NO_x formation rates have a non–linear relation to temperature. A mean temperature will give a different value of NO_x formation rates than a multi–zone model with distributed temperatures. However, a single burned zone seems feasible even when temperature nonlinearities are considered. Especially considering the fact that zones most probably do mix during combustion. The burned zones do in fact also mix with the air and EGR content in the unburned zone. This is the so called air dilution. However, this effect is often neglected in zero dimensional models due to the inherent complexity of describing such a phenomena in a physically accurate and time efficient way.[9]

4 Implementation in MATLAB

Theoretical work aside, the model must also be implemented in an efficient and easy way for future use in practice. As stated in previous sections, the model should also be useful when ran online. Therefore, even if it works in MATLAB, the model has to be kept simple with regards to time consumption. The model was scripted for diagnostic purposes, i.e. a complete data set was run from beginning to end and results of NO_x concentrations were acquired. In this section, the final MATLAB implementation of the NO_x model is briefly described.

4.1 Heat release calculations

The MATLAB code utilize a fairly trivial solution algorithm to the heat release equation. A function *ROHRSolver* was implemented for this purpose.
The rate of heat release was calculated with Euler discretization of the derivatives in Eq. (3.7).

Eq. (4.1) $$\frac{dQ}{d\varphi} = \left(\frac{\Delta p}{\Delta \varphi} V + \gamma p \frac{dV}{d\varphi}\right)\frac{1}{\gamma - 1} + \frac{dQ_{conv}}{d\varphi}$$

Volumes and volume derivatives are calculated analytically with Eq. (3.69). However, the pressure has to be differentiated with an approximation. The convective heat loss utilizes the Woschni heat transfer equation[22], as previously given. γ can either be set to a fixed value or approximated with the algorithm presented in section 3.5.1. To keep things simple, gamma was fixed at 1.3, which proved to give a fairly good fit in the heat release solution.

4.2 Temperature calculations

A function *calcT* calculates and returns the temperature in the burned zone. The calculations assume that the temperatures of the reactants are equal to the unburned zone temperature. A simplified assumption of the unburned zone temperature is that it only changes due to isentropic compression and expansion:

Eq. (4.2) $$T_{unburned} = T_0 \left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{\gamma}}$$

NASA interpolations of specific heats were implemented in the code and used to calculate the temperature in the burned zone:

Eq. (4.3) $$T_{burned} = \frac{dQ_{net}}{c_p n_{burned}} + T_{unburned}$$

Finally, the new combustion products were then added to the total burned zone. This will result in a temperature change in the burned zone. The easiest way to account for this is with the usage of temperature averaging between old content

and the newly added component (see Eq. (3.107)). However, this approach assumes temperature constant specific heats, as previously stated.

Isentropic expansion will also result in a temperature decrease in the burned zone that cannot be neglected. Luckily, this is implemented in the same manner as the temperature change in the unburned zone, as stated in Eq. (4.2). It is in the temperature calculations that the EGR contribution is taken into account.

4.3 Equilibrium calculations

When temperatures and masses are known, it is fairly straightforward to calculate the equilibrium concentrations. The equations Eq. (3.98) – Eq. (3.102) are iteratively solved for a given set of $[K_{p1}, K_{p2}, \lambda, p]$. λ is set to 1, p is measured pressure, while K_{p1} and K_{p2} are calculated according to section 3.6. A MATLAB function *burnedZone* executes the procedure.

4.4 Zeldovich mechanism

Lastly, a function *ZeldovichMechanism* calculates the radical concentrations and the NO_x concentration formation rate. Integration according to Eq. (3.26) returns the NO_x concentration at a given CAD. No additional post processing of data is done, except for a trivial recalculation of NO_x concentration, from mole/cm^3 to ppm.

4.5 C code implementation

The C code implementation follows the same principal layout as the MATLAB implementation. However, some other considerations are taken into account, as described in detail below (see section 5).

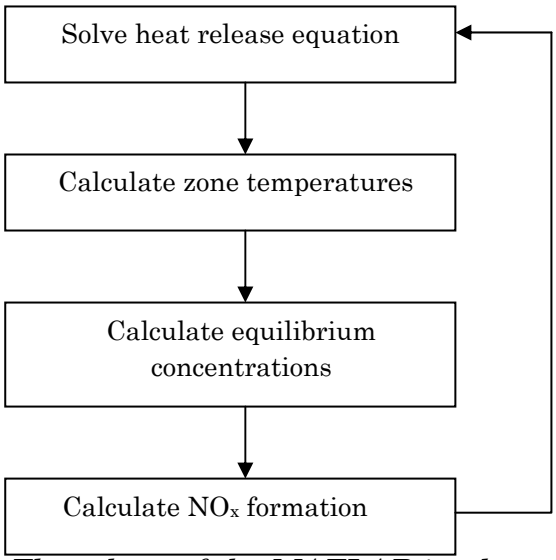

Fig. 4.1 – Flow chart of the MATLAB implemented model.

5 Implementation considerations for FPGA code

5.1 The division operation

It is a well known fact that an implementation in FPGA should avoid division operations. A study of the equations in the theory section (see section 3) implies an appalling fact. There are a lot of divisions in some way or another that cannot be avoided by simply neglecting them. One way to overcome this problem is to implement algorithms that approximate the divisions.

5.1.1 Thesis algorithm

In this thesis such an algorithm was produced. The algorithm can be split up in two separate steps, where in some cases the first step is sufficient enough. For demonstrative purposes, following division will be considered in the first step:

Eq. (5.1) $$p = \frac{nRT}{V}$$

The dividend and divisor, nRT and V respectively, are known. The pressure is determined via the ideal gas law which includes a rather simple division. The first step is to rewrite Eq. (5.1):

Eq. (5.2) $$p = -\frac{nRT}{V_M} \cdot \frac{1}{-\frac{V}{V_M}}$$

The V_M is the engine cylinder volume, where $V_M > V$. Now, by using the fact that a division on the form $1/(1-x)$ is:

Eq. (5.3) $$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots + x^n + \vartheta(x^{n+1})$$

This is however only true when $x > -1$ and $x < 1$.

The equation Eq. (5.2) can easily be rewritten on the form that is given by Eq. (5.3):

Eq. (5.4) $$p = \frac{nRT}{V_M} \cdot \frac{1}{1-\left(1-\frac{V}{V_M}\right)}$$

This approximately equals:

Eq. (5.5) $$p = \frac{nRT}{V_M} \cdot \left(1 + \left(1 - \frac{V}{V_M}\right) + \left(1 - \frac{V}{V_M}\right)^2 + \left(1 - \frac{V}{V_M}\right)^3 + \cdots + \vartheta\left(\left(1 - \frac{V}{V_M}\right)^n\right)\right)$$

The approximation given by Eq. (5.5) is true due to the fact that $1 - V/V_M > 0$ and $1 - V/V_c < 1$. The parameter V_c is known beforehand and thus the division $1/V_M$ can be set a priori. The introduction of V_M, on a first sight, might be thought of as a pure mathematical trick to ensure that Eq. (5.3) holds. However, this can

also be seen as an introduction of a damper which will have an effect on the convergence rate of the algorithm.

First and foremost, the algorithm can be rewritten on an arbitrary form as:

Eq. (5.6) $$div = \frac{A}{B}$$

The divisor B is known to be constrained by a B_{max} . This could for example be the maximal cylinder volume, in the case of the ideal gas law, as previously stated. B_{max} ensures the fulfillment of Eq. (5.3). Then the division, as stated in Eq. (5.6), is approximately:

Eq. (5.7) $$div = \frac{A}{B} = AC(1 + (1 - BC) + \ (1 - BC)^2 + \cdots + (1 - BC)^n + \vartheta((1 - BC)^{n+1})$$

The term ϑ is a rest term and C is set a priori as:

Eq. (5.8) $$C = \frac{1}{B_{max}}$$

However, as previously stated, the C is a damper. This will have a huge impact on the convergence rate. If B_{max} is sufficiently close to the true B, the algorithm will be close enough by only using terms up to second degree in Eq. (5.7).

The good thing is that C may very well be updated to be close to the real solution, due to the intrinsic mechanical work of a combustion process (compression and expansion). The simplest way to see this is to study the compression phase of combustion and what this implies for Eq. (5.1). A good initial damper would be the volume V_{total}, but this will only be close to V at initial states, when compression starts. However, if the division $1/V_n$ is calculated where V_{total} is sufficiently close to V_n, the damper at next time step can be set equal to $1/V_n$. This is due to the fact that the volume will decrease at compression and thus $V_n > V_{n+1}$. The total volume, V_{total}, is always known a priori as:

$$(1 + \frac{1}{r_c - 1})V_d = V_{total}$$

Unfortunately, the divisor will not decrease in all cases during the combustion and some divisors will even lack basic physical interpretation. Therefore a second step can be added that utilize a Newton−Raphson method. The idea is to use the previous algorithm, as given in Eq. (5.7), to get an initial guess. Assuming that the guess is $\tilde{p}$, the true value is p and the divisor is q, the equation to be solved is:

Eq. (5.9) $$qp = 1$$

The function that will be used in the Newton method is written as:

Eq. (5.10) $f(p) = qp - 1$

Eq. (5.11) $f'(p) = q$

The corresponding Newton method will be:

Eq. (5.12) $p_{n+1} = p_n - \frac{f(p_n)}{f'(p)} = p_n - \frac{f(p_n)}{q}$

One problem in Eq. (5.12) is the division *1/q*, which is in fact the division to be solved. Approximating *1/q* with p_n gives:

Eq. (5.13) $p_{n+1} = p_n(2 - qp_n)$

At the first step, the guess $\tilde{p}$ is used, $p_0 = \tilde{p}$. The Newton–Raphson method will unfortunately be sensitive to poor initial guesses, therefore the first step is essential in the implementation to get a good guess. It is not a good idea to use an arbitrary guess. However, this method implies a large time consumption and is thus not very time efficient. It is a far better idea to sometimes sacrifice accuracy in favor of better calculation speed.

5.1.2 Modified thesis algorithm

There is also a possible alternative approach to a division calculation algorithm which could be suitable for a C code implementation. Assuming that all float point numbers are converted to fixed point numbers, using the representation 2^n, the most significant bit becomes crucial in this modified algorithm. The most significant bit of a number (MSB), is the bit position having the greatest value. If MSB is calculated fast enough, the algorithm could be stated as:

Eq. (5.14) $\frac{A}{B} \ll n = \big(A \gg (MSB_B + 1 - n)\big)(2^n + (2^n - X) + (2^n - X)^2 \gg n) \gg n$

Eq. (5.15) $X = B \gg (MSB_B + 1 - n)$

The MSB_B is the MSB of number B. This algorithm can handle all possible divisions if the MSB of B is known. Eq. (5.14) can be further simplified by algebraic calculations:

Eq. (5.16) $\frac{A}{B} \ll n = \Big(\big(A \gg (MSB_B + 1 - n)\big)(3 \cdot (2^n - X) + X^2 \gg n)\Big) \gg n$

The bit shift will limit $X \in [2^{n-1}, 2^n]$ which corresponds to [0.5, 1] in a float point representation. Hence, the stability criteria is satisfied and the convergence rate will be sufficient. A possible approach to find the MSB is to use the parallel structure of the FPGA. The MSB can then be found fairly quickly by implementing the algorithm suggested by Warren in Hacker's Delight.[20]

In a case when the divisor is $B \ll 1$ in a division operation *A/B*, a slight modification can be done to the thesis algorithm to make it far more precise in terms of convergence. Remembering that the fundamental part of the algorithm is to rewrite it on a form *1/(1-x),* following trick can be done in this case which proves to be especially accurate for small divisors:

Eq. (5.17) $\frac{1}{1-x} = \frac{1+x}{1-x^2} = \frac{(1+x)(1+x^2)}{1-x^4}$

This can in fact be simplified as follows:

Eq. (5.18) $(1+x)(1+x^2)(1+x^4)\dots(1+x^n)\frac{1}{1-x^{2n}} \approx (1+x)\prod_{k=1}^{\frac{n}{2}}(1+x^{2k})$

As can be seen in Eq. (5.18), the algorithm will be accurate when:

$$\frac{1}{1-x^{2n}} \approx 1$$

This of course implies that $x^{2n} \approx 0$, which will be especially true for divisors of small magnitudes.

5.2 Tabulation of data

When running the model in MATLAB offline there are no time constraints that have to be met. However, the case is quite different online on an FPGA. For example, the non−linear equations cannot be solved sufficiently fast. Also the NO_x formation equations, as given by the Zeldovich mechanism, are nonlinear and gruesome to calculate. The easiest way to approach this particular problem is to tabulate the solutions for different pressures and temperatures and to utilize interpolations. In this sub−section this is explained in short.

A multi−dimensional least−squares method is an appropriate approach in the case of interpolating tabulated data.[21]

5.2.1 The least−squares method

If there is a set of output data $\bar{z} = \{z_1, z_2, z_3, \dots, z_n\}$ with a corresponding input set $\bar{x} = \{x_1, x_2, x_3, \dots, x_n\}$ and $\bar{y} = \{y_1, y_2, y_3, \dots, y_n\}$, a least−squares interpolation is possible. For example, the output data may be the specific heat capacity at constant pressure, and the input data could be EGR percentage and temperature. To find a least square fit, the error function has to be minimized:

Eq. (5.19) $$E(\bar{P}) = \sum_{i=1}^{n} (z_i - \bar{P}\bar{Q}_i)^2$$

Where $\bar{P} = (p_1, p_2, p_3, p_4, p_5, p_6)$ and $\bar{Q}_i = ({x_i}^2, x_i y_i, {y_i}^2, x_i, y_i, 1)$. Minimizing the error is equivalent to solving the equation:

Eq. (5.20) $$\nabla E(\bar{P}) = 2 \sum_{k=1}^{n} (z_k - \bar{P}\bar{Q}_i)\ \bar{Q}_i = 0$$

The $\bar{Q}_i$indicates that the polynomial fit to Eq. (5.20) is parabolic. This will be a sufficient degree of complexity for the purposes of this thesis. In this case the equation can then be rewritten in a matrix form as:

Eq. (5.21) $$\begin{bmatrix} \sum x_i^4 & \sum x_i^3 y_i & \sum x_i^2 y_i^2 & \sum x_i^3 & \sum x_i^2 y_i & \sum x_i^2 \\ \sum x_i^3 y_i & \sum x_i^2 y_i^2 & \sum x_i y_i^3 & \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i y_i \\ \sum x_i^2 y_i^2 & \sum x_i y_i^3 & \sum y_i^4 & \sum x_i y_i^2 & \sum y_i^3 & \sum y_i^2 \\ \sum x_i^3 & \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i^2 y_i & \sum x_i y_i^2 & \sum y_i^3 & \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i^2 & \sum x_i y_i & \sum y_i^2 & \sum x_i & \sum y_i & n \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} \sum x_i^2 z_i \\ \sum x_i y_i z_i \\ \sum y_i^2 z \\ \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}$$

For example, it is now easy to find a fit for the specific heat as a function of EGR and temperature. The resulting polynomial will only require multiplications when implemented in C. Multiplication is a lot faster operation than actually calculating the specific heat from the NASA interpolations. These calculations would require both divisions and tabulations of NASA thermodynamical data, which are both time consuming and memory inefficient.

5.2.2 Functions of a single variable

In other cases, where there is only one dependent variable, it is possible to use a simple linear fit that can be retrieved from suitable MATLAB functions. This is the case for the isentropic compression term given by:

Eq. (5.22) $$\left(\frac{p}{p_o}\right)^{\frac{\gamma-1}{\gamma}}$$

If the sampling rate is known, discretizations of geometrical properties such as surface area, volume and volume derivative, are fairly easy to tabulate and store a priori.

5.2.3 Data storage in memory maps

One of the most common ways to solve the issues involving nonlinearities is to use the possibility to retrieve and store data. This data can then be accessed by an FPGA very fast.[13]

6 Results

6.1 Influence of the EGR

As previously stated, EGR is used to lower temperature conditions in the burned zone during combustion. This will have an effect on the NO_x formation in such a manner that the conditions would be poorer for NO_x to form. The temperature calculations reveal why this is the case. The increase in temperature can be calculated with basic physics as dT/mc_p. The effect of EGR on oxygen concentration in the combustion zone is however omitted in the validation.

The increase of EGR during the intake stroke will have an effect on the specific heat value. If only CO_2, H_2O, N_2 and O_2 are assumed to be present in the exhaust gas, the NASA specific heat interpolation for each species will be:

Eq. (6.1) $$\frac{C_p}{R} = a_1T^{-2} + a_2T^{-1} + a_3 + a_4T + a_5T^2 + a_6T^3 + a_7T^4$$

The specific heat for carbon dioxide and water will be somewhat higher. This implies a smaller increase in temperature if EGR is present during the combustion stroke. Eq. (6.1), the NASA polynomial used in the tabulation, of course holds for other molecular compounds as well.

	C_p at 500 K	C_p at 800 K	C_p at 1500 K	C_p at 2500 K
N_2	29.6	31.4	34.8	36.6
O_2	31.1	33.8	36.6	39.2
CO_2	44.6	51.5	58.4	61.4
H_2O	35.3	38.7	47.7	57.9
Pure air	29.8	31.9	35.2	37.1
Air+10% EGR	30.1	32.2	35.6	37.6
Air+20% EGR	30.4	32.5	36.0	38.1

Table 6.1 – Values of specific heats at different temperatures for N_2, O_2, CO_2 and H_2O. The values are given in J/K·mole. The Air + EGR values are calculated for a global mean λ = 1.3.

The interesting thing however is not the specific heat values of the species individually, but rather the averaged C_p. The averaged C_p will be:

Eq. (6.2)

$$\bar{C}_p = (1 - EGR\%)\frac{1}{4.773}\left(3.773C_{pN_2} + C_{pO_2}\right) + EGR\%C_{pEGR}$$

Eq. (6.3)

$$C_{pEGR} = \frac{1}{a+\frac{b}{2}+\left(4.773\lambda_{gl}-1\right)\left(a+\frac{b}{4}\right)}\left(aC_{pCO_2} + \frac{b}{2}C_{pH_2O} + \left(a+\frac{b}{4}\right)\left(\left(\bar{\lambda}_{gl}-1\right)C_{pO_2} + 3.773\lambda_{gl}C_{pN_2}\right)\right.$$

The dependency of the mean specific heat on temperature and EGR percentage is shown in Fig. 6.1 and Fig. 6.2. The main thing that can be concluded is that the EGR will have a non linear influence on the mean specific heat for higher temperatures. On the other hand, the level curves shown in Fig. 6.2 imply an almost linear relation between EGR and temperature for lower temperatures.

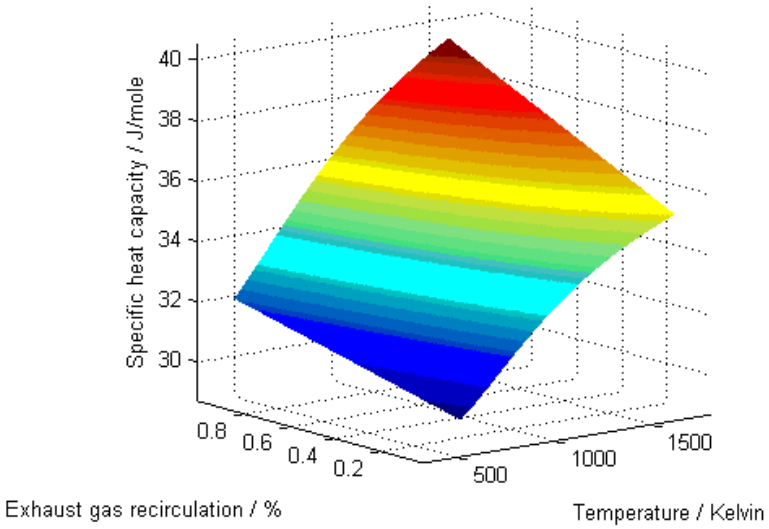

Fig. 6.1 – Mean specific heat capacity as function of temperature and EGR.

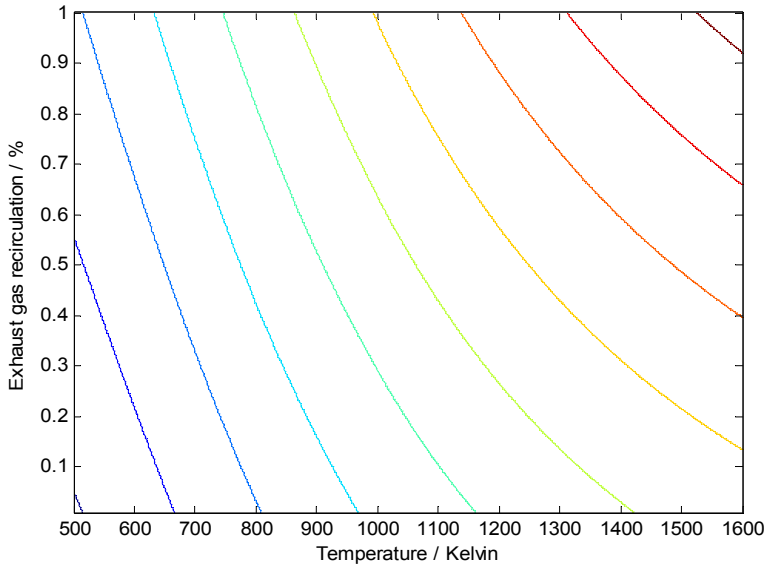

Fig. 6.2 – The corresponding level curves. Notice the nonlinear curves at higher temperatures.

The EGR will also have an impact on the specific heat ratio that will be relevant in heat release calculations. However, if γ is estimated during the compression phase the EGR will be taken into account. Because of calculation time issues, γ was set to a constant value. This value will approximately be 1.3 at all times. The deviation from this value will purely be due to temperature and EGR inflow changes. The calculations described by Eq. (3.41) assume only one zone. This is due o the fact that the heat release calculations also assume only one zone.

6.2 Calculation of heat release

In the calculation of heat release, pressure data and volume calculations must be used. The volume calculations are purely theoretical (see section 3.5.6) and quite accurate, although approximative. However, the same cannot be said about the pressure trace used in the calculations. The combustion process itself will initiate acoustic phenomena in the cylinder, with a certain frequency, due to abrupt pressure changes. This frequency will be relatively high (≈3.7 kHz) and will cause problems when the differential dp is calculated (see Eq. (3.7)). Differentiation of noisy measurements is not a good idea, as can be seen in the calculation of net heat release in Fig. 6.3.

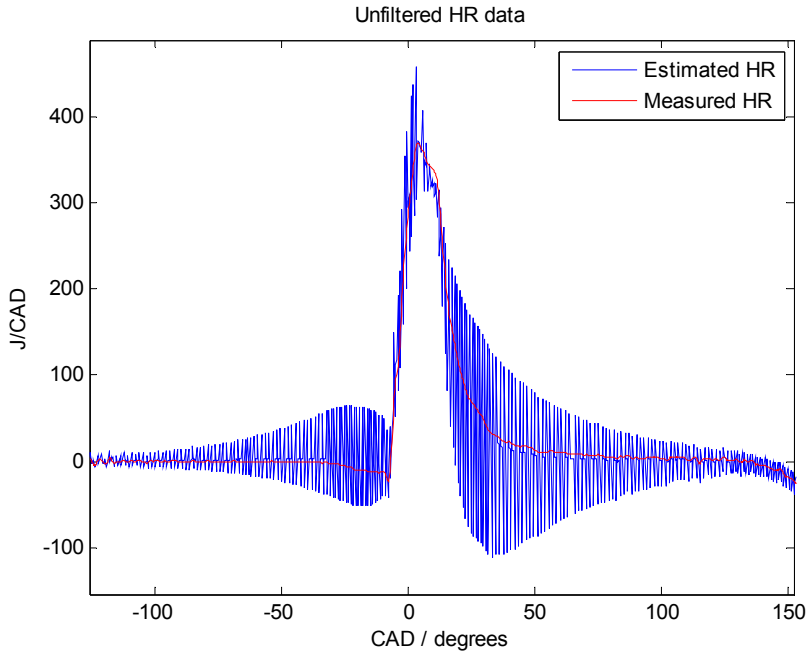

Fig. 6.3 – Calculations of heat release when data is unfiltered.

Several low pass filters were tested in MATLAB. The filter that proved to be the most satisfying was a Savitzky–Golay smoothing filter[22]. This particular filter performs a polynomial regression on a data collection in local sections. Another interesting fact is that it does not matter when data is filtered. The filtering can

be done directly on pressure trace data or on the calculated heat release. The filtered result is shown in Fig. 6.4.

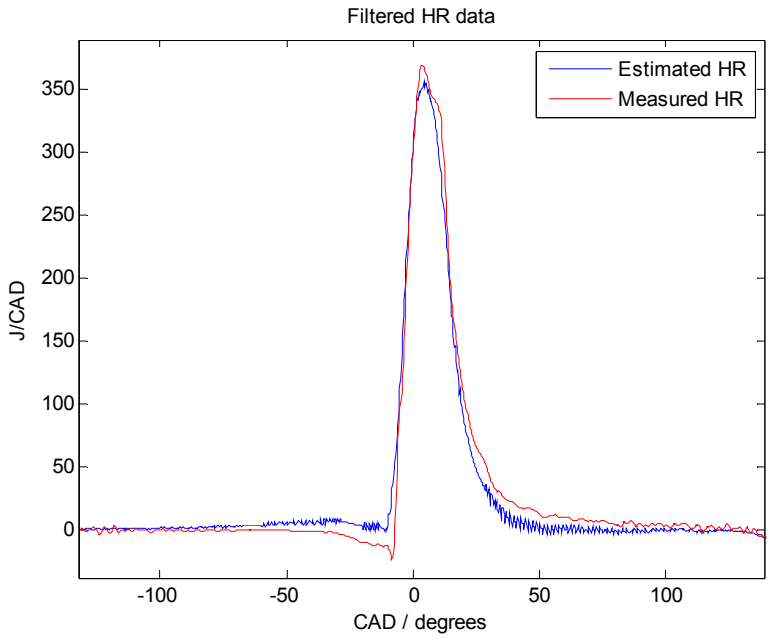

Fig. 6.4 – Savitzky–Golay filtered heat release data.

As can be seen, the filter is very effective in eliminating high frequency noise without deteriorating the true heat release. The filter could also be applied to the pressure trace directly, before any calculations of the heat release.

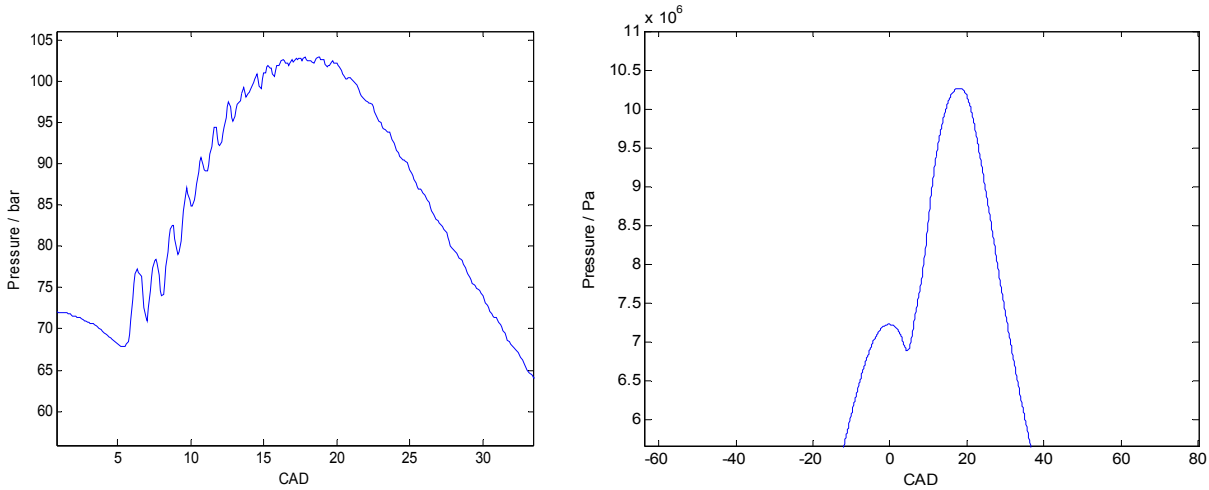

Fig. 6.5 – Unfiltered pressure data at CAD interval [0,35]. The measurements contain very peaky resonances that will cause problems in the differentiation of pressure. The graph to the right is the same pressure data when a Savitzky–Golay filter is applied.

The first peak in pressure, as can be seen in Fig. 6.5, at 0 CAD is caused by the isentropic compression performed during the compression stroke. It is in fact this isentropic pressure trace data that is used when γ is estimated in Eq. (3.41).

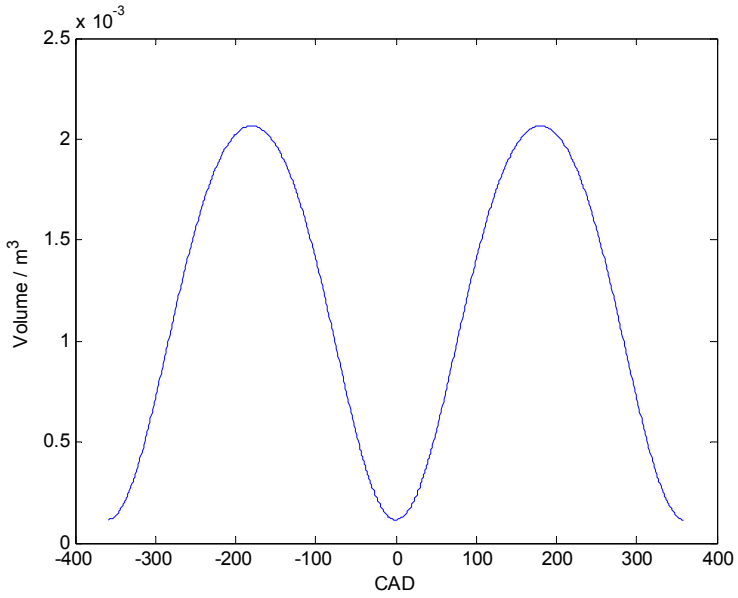

Fig. 6.6 – Cylinder volume as function of CAD. Notice that the low volume at CAD = 0 has a corresponding pressure peak due to isentropic compression.

6.2.1 Estimation results of γ and the impact on heat release

The estimation of gamma, as proposed in Eq. (3.41), resulted in a mean value of $\gamma = 1.32$. This γ is somewhat higher than the constant value used in most of the simulations ($\gamma = 1.3$). The corresponding heat release is presented in Fig. 6.9.

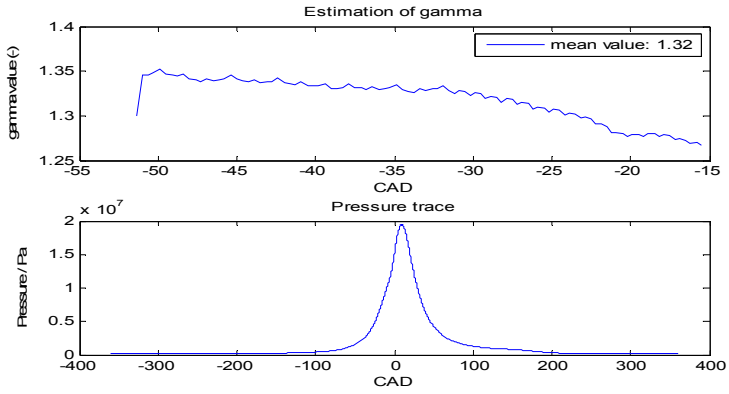

Fig. 6.8 – Pressure trace and corresponding gamma estimate. The mean value was determined to be 1.32.

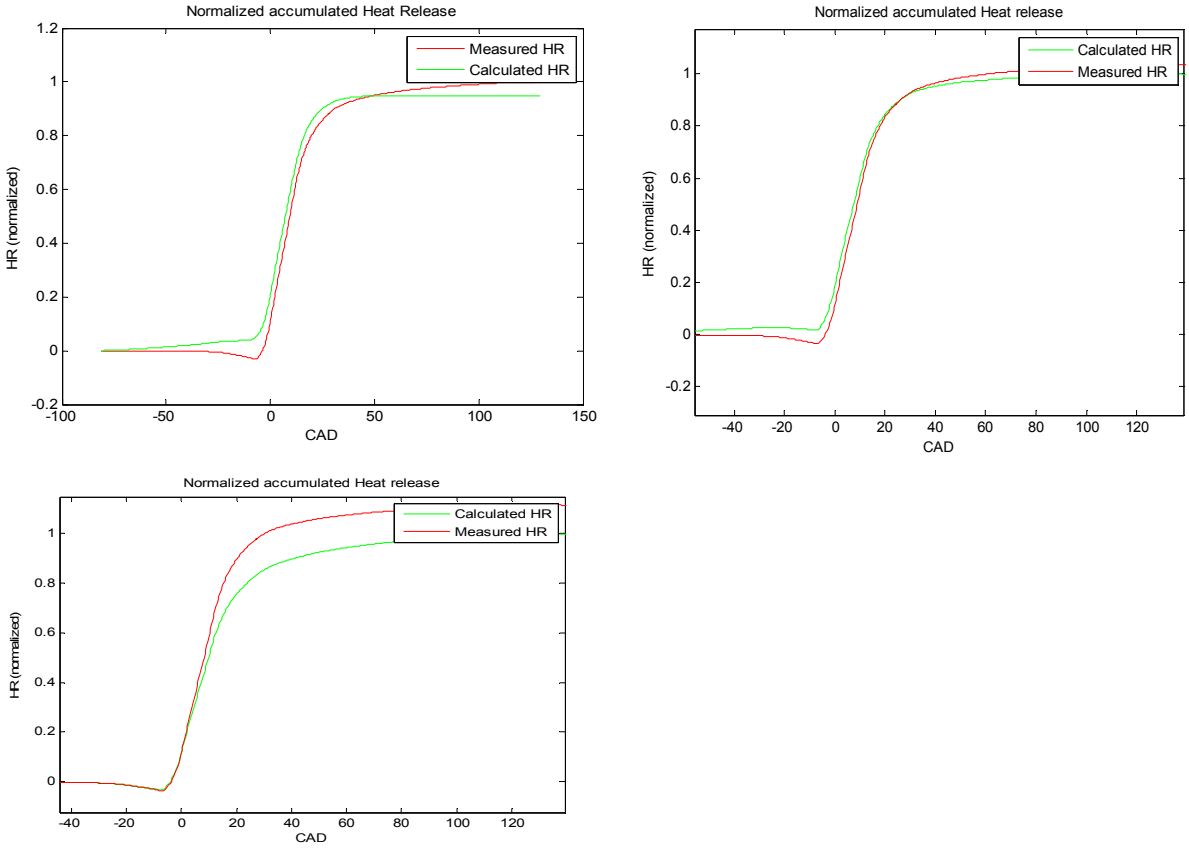

Fig. 6.9 – Normalized accumulated heat release at the estimated gamma, $\gamma_{estimate} = 1.32$ (top left), γ=1.3 (top right) and γ=1.35 (bottom left). CA50 = 9° in top cases and CA50 = 10° when γ=1.35.

Greater accuracy could be achieved by utilizing the fact that γ can be estimated before and after combustion fairly accurately, assuming adiabatic compression. Tunestål proposes a linear approximation in–between to account for changing γ during the combustion process itself.[23] However, there is an upside by the simple algorithm as given by Eq. (3.41). It does in fact account for any EGR that may contribute to change in the polytropic exponent, remembering that $\gamma = \frac{c_p}{c_v}$.

In this thesis, c_p, the specific heat value at constant pressure, is determined using NASA Glenn interpolation tables. Remembering that the polytropic exponent is in fact a ratio between the specific heat value at constant pressure and at constant volume it is possible to have a temperature dependent estimate of γ. The specific heat at constant volume can be determined from $R = c_p - c_v$, and thus c_v is also known. Results from heat release calculations when this approach is used, is shown in Fig. 6.10 – 6.12.

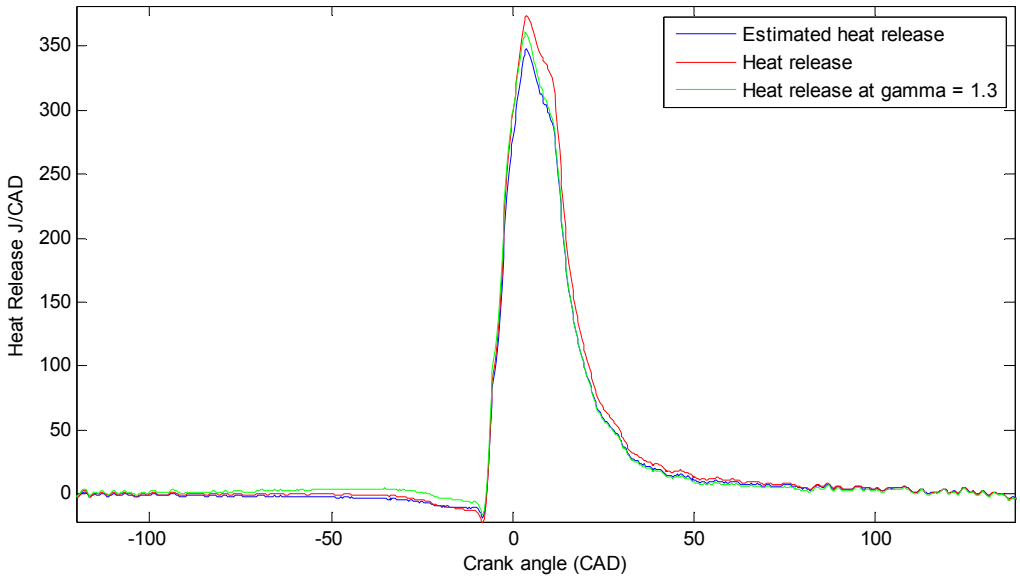

Fig. 6.10 – Results of heat release calculations when the polytropic exponent is made adaptive by using NASA Glenn tabulations of specific heats. Calculated with EGR% = 10%. Interpolated specific heats are used i.e. $\lambda_g = 1.3$. (*see Fig.* 6.25).

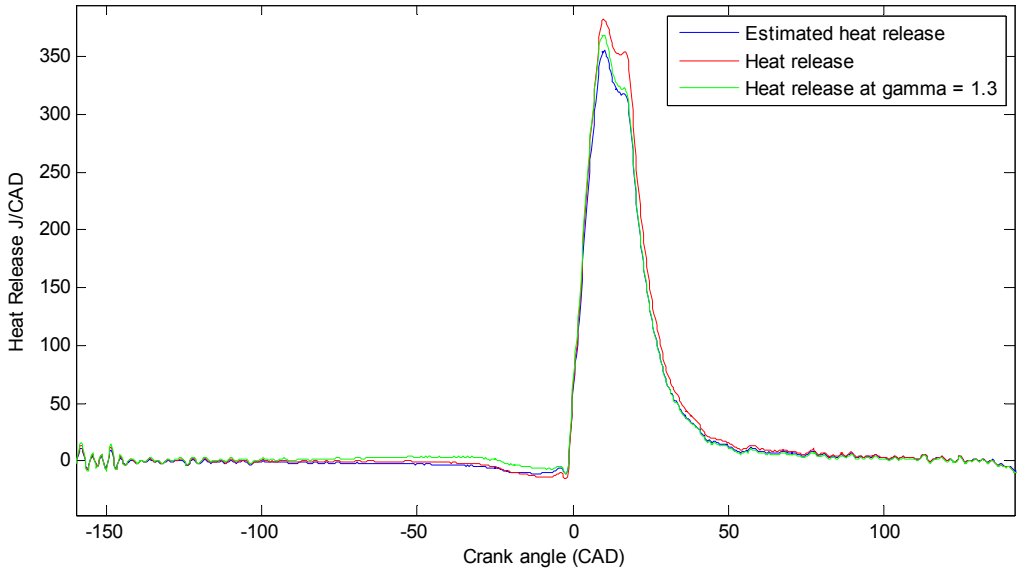

Fig. 6.11 – Results as described in Fig. 6.10 but for another set of data. Calculated with EGR% $= 15\%$. $\lambda_g = 1.3$. (*see Fig.* 6.25).

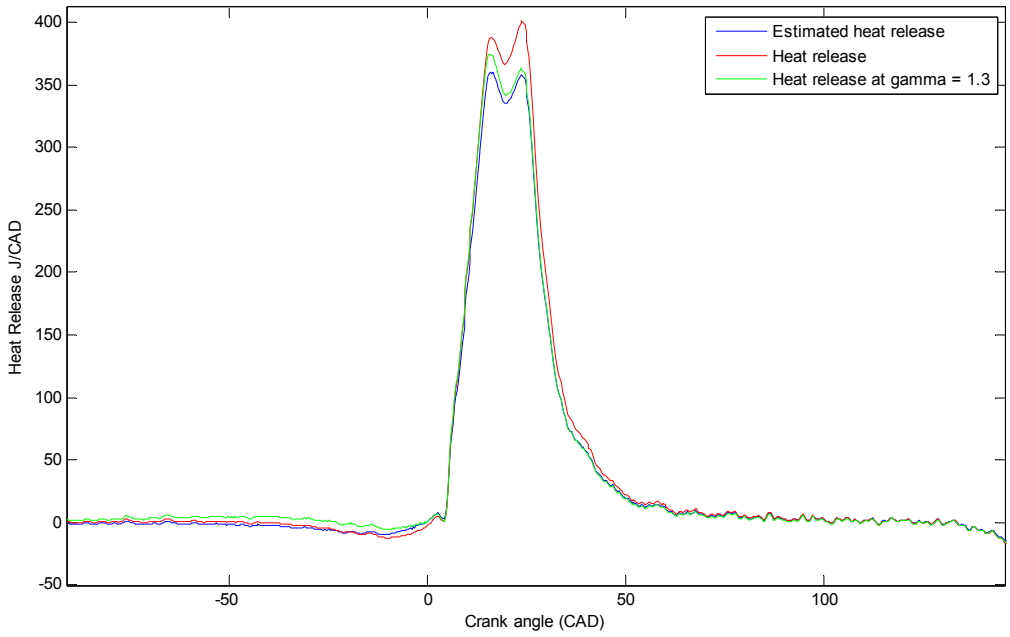

Fig. 6.12 – Results as described in Fig. 6.10 but for another set of data. Calculated with EGR% $= 0\%$ $.\lambda_g = 1.3$. (*see Fig.* 6.25).

As can be seen in Figs. 6.10–6.12, the heat release estimated by using an updating γ will have a somewhat smaller peak in all three cases. The case when a constant $\gamma = 1.3$ gives almost the exact heat release results as the adaptive γ.

6.3 Temperature estimations

The model assumes two different zones during the combustion phase. One zone will contain all burned products due to combustion of fuel, EGR included. The other zone contains unburned fuel, air and EGR. The concentration of EGR is assumed to be uniform in the whole cylinder. That means that the EGR concentrations in both zones are in fact equal at all times. The temperature will be essential in the Zeldovich mechanism and thus also essential to NO_x formation.

The unburned zone was assumed to change temperature purely due to isentropic compression. This means that the heat transfers between the zones where neglected. Although somewhat physically unjustified, this assumption is common in zero dimensional modeling.

As can be seen in Fig. 6.13, the temperature in the burned zone peaks at around 2500 K. The temperature will in fact, as stated previously, vary with the EGR in the gas content. The peak temperature in the unburned zone is much lower, at around 1000 Kelvin. This emphasizes the unrealistic aspects of the assumption that the temperature changes in the unburned zone is solely due to isentropic compression. This would imply a temperature gradient, between the zones, that would be too large in magnitude for it to hold in reality.

Additionally a temperature decrease was added due to radiative heat losses of the combustion products. This was stated as:

$$dT = \frac{C_{calib} T_{flame}^4}{c_p}$$

The variable C_{calib} is used to calibrate the temperature loss.

Aside from the radiative temperature loss, there is also a decrease in temperature due to cylinder expansion which is taken into account with a factor for the isentropic compression. However, the temperature decrease due to dissociation of carbon dioxide into carbon monoxide and water to hydrogen was neglected. The relationship between dissociation and temperature decrease is complex and hard to calibrate, especially with EGR present in the gas. A previous study by Andersson[1] indicates that the temperature decrease due to dissociation is at average circa 5 % of the total flame temperature at the pressure range of interest and at a λ close to 1.

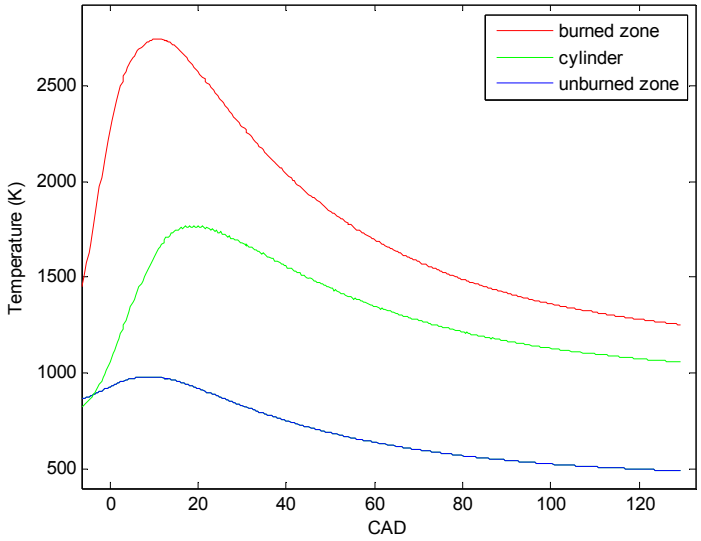

Fig. 6.13– Temperature in burned and unburned zone with the mean gas temperature included.

A closer observation of Fig. 6.13 implies that the generated temperature results seems feasible. It would be expected that the burned zone is the hottest one, followed by the mean gas temperature and finally the unburned zone temperature which is the colder zone in the cylinder.

6.4 Equilibrium concentrations

When the temperatures are known it is possible to calculate the equilibrium concentration in the burned zone using the equations described in the theory section. The variables that are dependent on temperature are in fact only the equilibrium constants that are needed in two of the equations. The resulting equilibrium constants at a specific temperature are shown in the figures below.

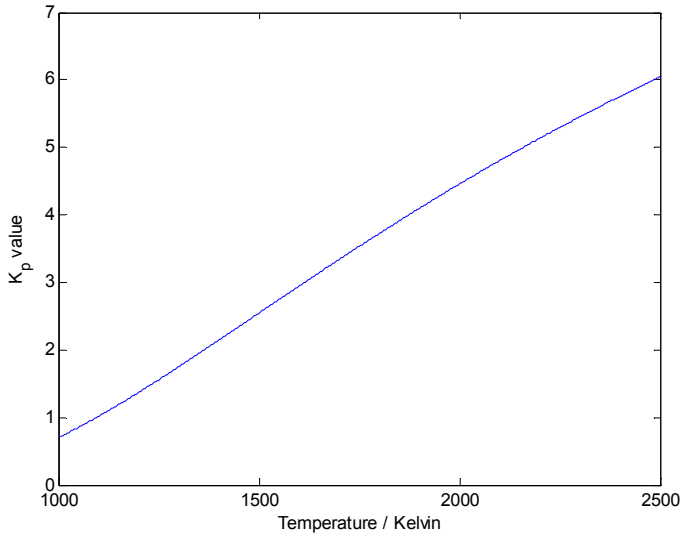

Fig. 6.14 – Values of K_{p1} at different temperature (see R3.7). This constant shows a near linear dependency on temperature in the given temperature range.

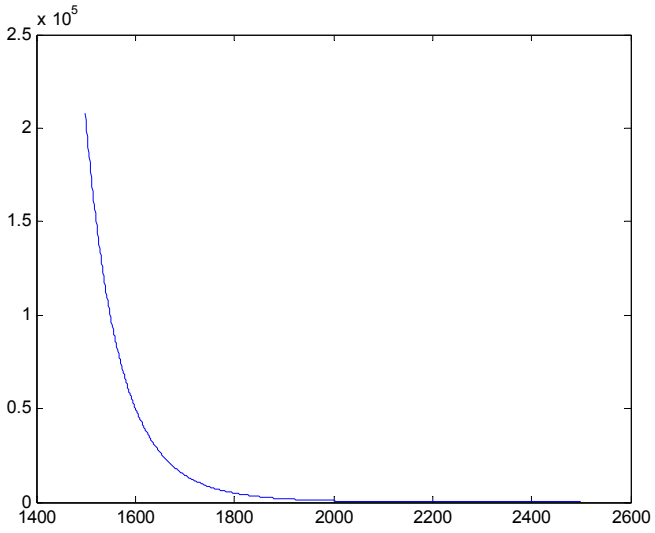

Fig. 6.15 – Values of K_{p2} at different temperatures (see R3.8). This equilibrium constant is far more dependent on temperature than the values of K_{p1}.

The equilibrium constants were interpolated with polynomials for easier and faster calculations. K_{p1} was approximated with only one fourth degree polynomial, but K_{p2} had to be split up in three separate sections due to its heavy dependence on temperature. The resulting polynomials are given as:

Eq. (6.4)

$$K_{p1} = 2.6223 \cdot 10^{-13} T^4 + 2.4793 \cdot 10^{-9} T^3 + 7.8685 \cdot 10^{-6} T^2 - 0.0064678 T + 1.5537$$

Eq. (6.5)

$$K_{p2} = -2.9827 \cdot 10^{-5} T^3 + 0.19986 T^2 - 447.42 T + 3.3477 \cdot 10^5, \quad T = (1500, 2250) K$$

$$K_{p2} = -2.9375 \cdot 10^{-6} T^3 + 0.022093 T^2 - 55.554 T + 46730, \quad T = (2250, 2500)\ K$$

$$K_{p2} = 8.8 \cdot 10^{-11} T^4 + 1.13 \cdot 10^{-6} T^3 + 0.0054 T^2 - 11.4 T + 9156\,, \quad T = (2500, 3500) K$$

The large magnitudes involved in the Jacobi matrix rendered the Newton–Raphson method(see section 3.6) unstable if a relaxation matrix was not introduced. However, the method still needed a good initial guess. Fortunately, the initial guess is simple to set if the fuel is known. At combustion without the gas water shift reaction, it is possible to calculate the equilibrium of CO_2, H_2O, N_2 and O_2 analytically with some help from the stoichiometric relation given in R3.5. A good initial guess for iso–octane is:

Eq. (6.6) $\quad \bar{a}_0 = \{8 \quad 0.1 \quad 13 \quad 1 \quad 0.1\}$

There are a total of 6 concentrations that need to be determined. However, the initial guess contains only 5 concentrations. This is due to the fact that $a_{N_2}$is trivial to determine. This can be seen in Eq. (3.95) by the fact that the concentration of nitrogen is independent of all the other equilibrium concentrations. The nitrogen is in the first step non–reactive, because the model assumes two subzones in the burned zone. The nitrogen will only participate in chemical reactions in the post combustion zone where NO_x is formed. This simplifies the system of equations to a total of five equations.

The convergence of the Newton–Raphson method will not depend on deviations of lambda or normalized pressure. The largest contribution to stability issues are large magnitudes of the equilibrium constant K_{p2}. In some cases the Newton–Raphson method can solve the equations in only 20 iterations with fairly good accuracy. However, there is much more room for improvement when it comes to the relaxation matrix as described in section 3.6. One suitable solution would be to make it adaptive.

The most suitable approach to the issue of solving these linear equations is simply to solve them offline for variations in normalized pressure and equilibrium constants. The algorithm used above and the MATLAB function fsolve are too slow to motivate an implementation in C for real time use in an embedded system. The solutions retrieved offline can then be tabulated and used online in a table format.

The Newton−Raphson method derived in this thesis was also validated for some specific values of parameters. The Newton method results matched the fsolve results fairly well in most cases, with a far lesser computational time. However, even the Newton method, as it is today, is probably not time efficient enough for real time executions and for large magnitudes of the equilibrium coefficients, the method was unstable at times. The average time to solve the equations with ~1500 iterations took 35 milliseconds in MATLAB. This is of course highly unsuitable results if a model ought to be implemented and run in real time applications. In cases where the convergence was sufficient after only 20 iterations, it took 6 ms in MATLAB. However, the method is considerably faster than fsolve, which utilize a Gauss–Newton method on the form $\min\|r(\beta^s) + J_r(\beta^s)\Delta\|_2^2$. This criterion is minimized over $\Delta = \beta - \beta^s$, which will give the solution β^s.

6.5 Model NO_x output

The output of the model is NO_x concentration, given in parts per million. As stated in Egnell[4], NO_x is not formed during the premixed combustion phase. Therefore this part was ignored during the calculations. Calculations of NO_x concentrations are initiated a number of samples before the temperature peak in the burned zone. This approach gave the most accurate result at given calibration values.

Measurement	*Measured NOx (ppm)*	*Calculated NOx (ppm)*
Point 1 (EGR% = 7.4%)	*86*	*86*
Point 2(EGR% = 0.0%)	*2261*	*2879*
Point 3(EGR% = 5.9%)	*2201*	*1771*
Point 4(EGR% = 9.4%)	*1419*	*1747*
Point 5(EGR% = 11.3%)	*1342*	*1538*
Point 6(EGR% = 13.3%)	*1126*	*1394*

Table 6.2 – Measured NOx vs. calculated NOx from model at 1000 rates per minute and $\lambda_{local} = 1$ at a set calibration.

As can be seen by table 6.2 the model estimate of NO_x is much larger than the measured NO_x in Point 2. This could be motivated by the fact that the calculated temperature in the burned zone is somewhat higher than the real temperature. This will have a large impact on the calculated NO_x concentrations that cannot be neglected. The assumption that the temperature drop due to dissociation can be neglected will hold in cases where EGR is present. However, when EGR is not present in the gas content, the assumption will introduce an error. The same holds for the temperature drop due to radiative heat losses in the post combustion zone.

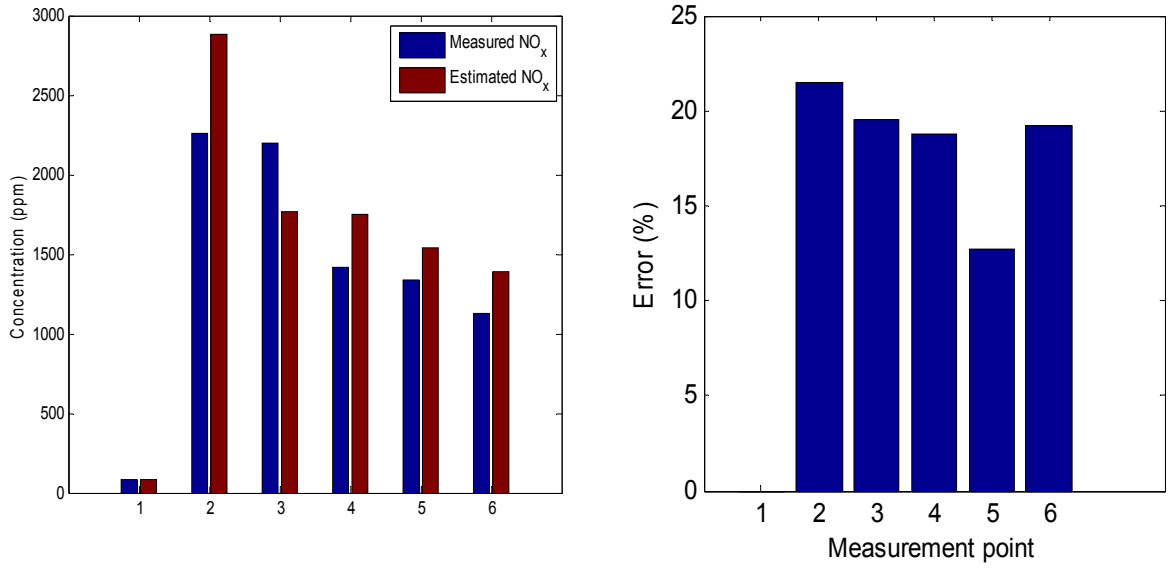

Fig. 6.16 – Graphical comparison of estimated and measured NO_x from Table 6.2. EGR% = 7.4%, 0.0%, 5.9%, 9.4%, 11.3% and 13.3%, respectively.

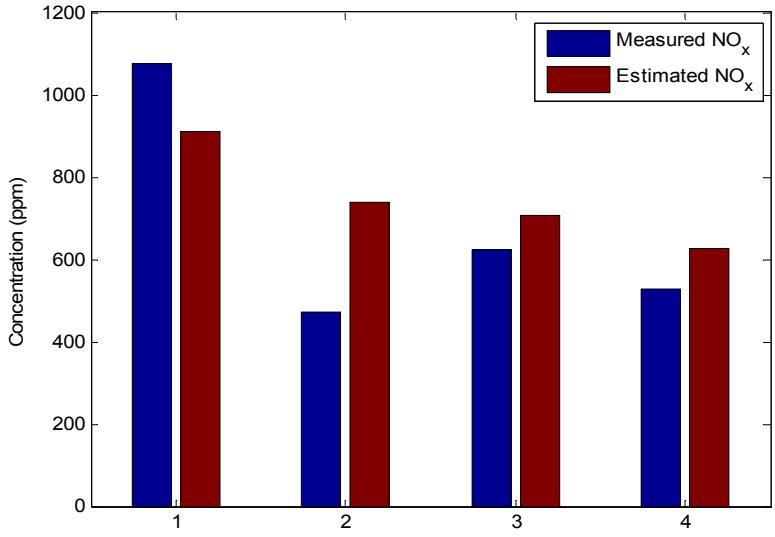

Fig. 6.17 – Model comparison to measurement values of NO_x for a data set at rpm = 1200, $\lambda \approx 1.2$ and EGR 14.9%, 21.3%, 20.7% and 22.6%, respectively.

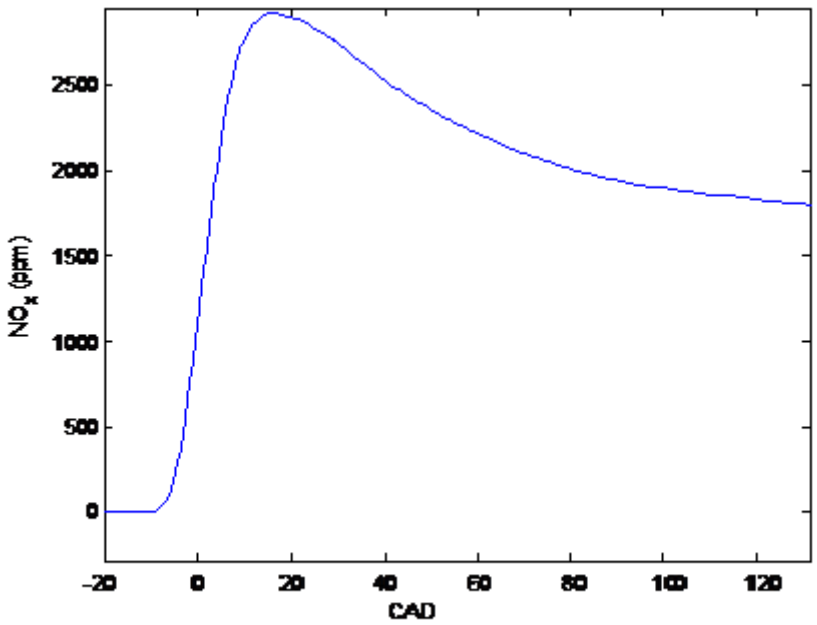

Fig. 6.18 – An example of a CAD–resolved simulation results of NO_x formation.

As can be seen by Fig. 6.18, the NO_x concentration will, at some point in time, have an overshoot and then reach a final stationary value. This simulation result is consistent with other results that utilize the Zeldovich mechanism to describe NO_x formation.[8]

Additionally, a revised version of the Zeldovich mechanism was used. The NO_x concentration as given in Eq. (3.26) assumes a constant volume. However, this is hardly the case during combustion and therefore the equation was slightly modified to account for volume changes[12]:

Eq. (6.7) $$\frac{dc_{NO}}{dt} = \left(\frac{dc_{NO}}{dt}\right)_{Heywood} - \frac{c_{NO}}{V}\frac{dV}{dt}$$

The calibration parameter of the NO_x model was the temperature drop due to radiative heat losses generated. This specific approach of model calibration is supported by previous research conducted by Ericson.[24]

6.6 The division algorithm

The division algorithm derived in this thesis was evaluated in MATLAB with fixed point representation implemented with the MATLAB built–in function *bit shift*. The method seems promising as long as it is possible to retrieve the most significant bit in a reasonable amount of time. The error is displayed in Fig. 6.19 where the function *1/x* was plotted with utilization of the thesis division algorithm and compared to the real division. One thing should be underlined: the error in Fig. 6.19 also takes quantization into account. This means that the graph shows what the error would be if the algorithm was run real time in C.

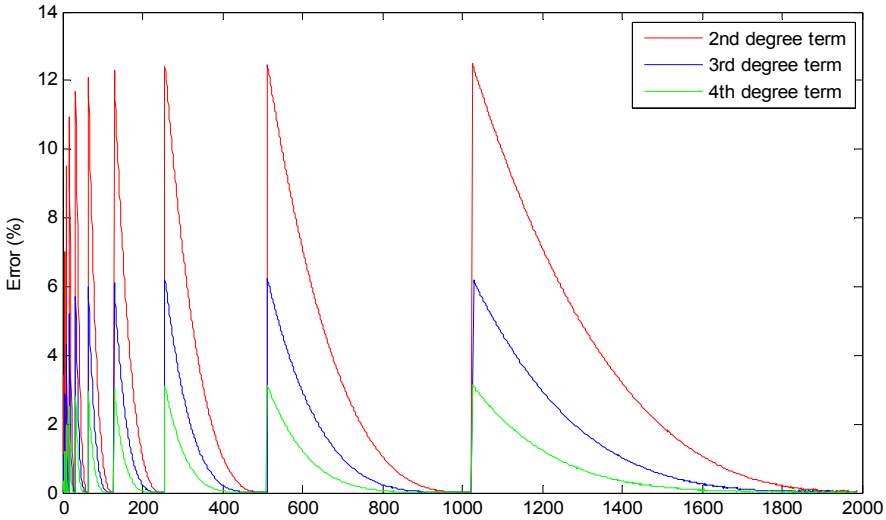

Fig. 6.19 – The error relative to the true value of division $\frac{1}{x}$*. The maximum error is around 12% and most often below 8% for the 2nd degree case.. Errors caused by quantization are also included in the plot. The x–axis shows the divisor. As can be seen, the error is cut in half for each step higher in degree.*

As can be seen, the error is reasonably low considering the degree of the approximation (see Eq. 5.16). Accuracy, unfortunately, must be sacrificed in order to get a good run time online. Another positive aspect of the algorithm is the fact that it's stable at all times. This is in contrast to the Newton–Raphson approach to the problem, where the initial guess has to be very close to the true value.

Initial guess	0.1	0.5	0.75	0.9
Result	0.2857	0.2828	−675	0.67 million

Table 6.3 – Results with Newton–Raphson applied to the division 1/3.5 after 4 iterations. The true value is in this case 0.2857. The Newton can use a relaxation value λ << 1, if more stability is desired. Higher λ yield greater stability issues. A conclusion from this is that the Newton method is too poor with respect to convergence and speed for a reasonable online utilization.

For the case when the divisor is close to 0, the alternative algorithm renders an error plot as shown in Fig. 6.20. The approximation used was:

Eq. (6.8) $\frac{1}{1-x} \sim (1+x)(1+x^2)$

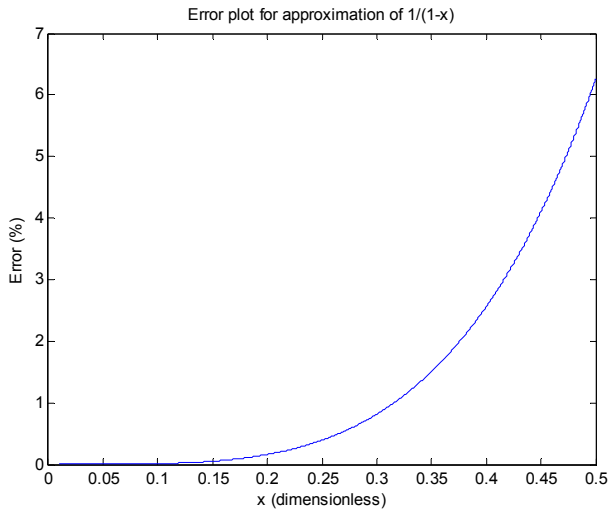

Fig. 6.20 – The error plot when the alternative algorithm is used in a range $x \in [0.0,0.5]$.

Higher accuracy can of course be reached by using terms of higher degree. The next, more accurate approximation would be (see Eq. 5.18):

Eq. (6.9) $\frac{1}{1-x} \sim (1+x)(1+x^2)(1+x^4)$

However, this is rather unnecessary in, terms of accuracy, if following mathematical trick is done:

Eq. (6.10) $\frac{1}{1-x} \sim (1+x)(1+x^2)(1+x^4) = (1+x)(1+x^4+x^2+x^6)$

Neglecting the term x^6 will give an approximation of the division as follows:

Eq. (6.11) $\frac{1}{1-x} \sim (1+x)(1+x^2+x^4) = (1+x)(1+x^2(1+x^2))$

The resulting error plot can be seen in Fig. 6.21. As can be seen in Fig. 6.21 the maximum error is far smaller than the maximum error shown in Fig. 6.20. This is of course not for free as the approximation in Eq. (6.11) needs 2 additional operations (1 multiplication and 1 addition). However, the same accuracy can be achieved by using the approximation in Eq. (6.8) and simply bit−shifting into the range where $x \in [0.0,0.25]$. This seems like a promising approach to approximate divisional operations.

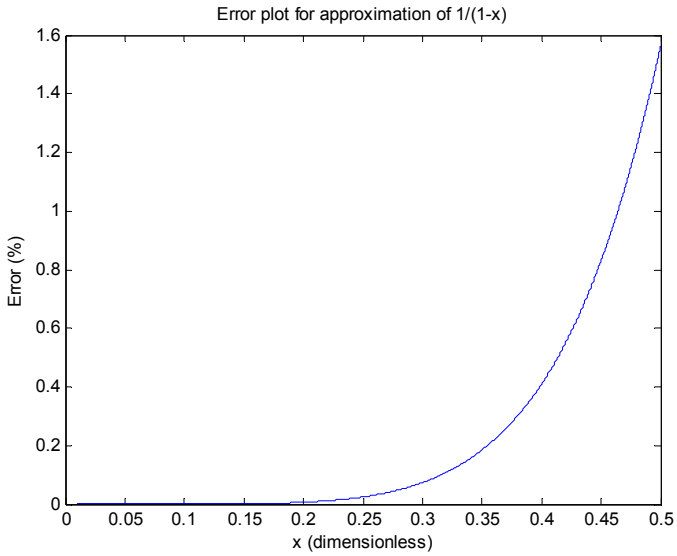

Fig. 6.21 – The corresponding error plot when $(1 + x)(1 + x^2(1 + x^2))$ *is used as an approximation instead.*

6.7 Interpolation results

This section is supposed to illustrate how interpolation can simplify rather gruesome and nonlinear calculations. To avoid unnecessary nonlinear calculations some interpolations were made to enable fast calculations. For example, the isentropic compression gives a relation on the form:

Eq. (6.12) $\left(\frac{P_n}{P_0}\right)^{\frac{\gamma-1}{\gamma}}$

This is however neither desirable nor efficient to calculate on an FPGA. The approximation of the relation was done by linear interpolation in specific segments. As can be seen in Fig. 6.22, three segments give a decent fit even when a simple linear interpolation is used.

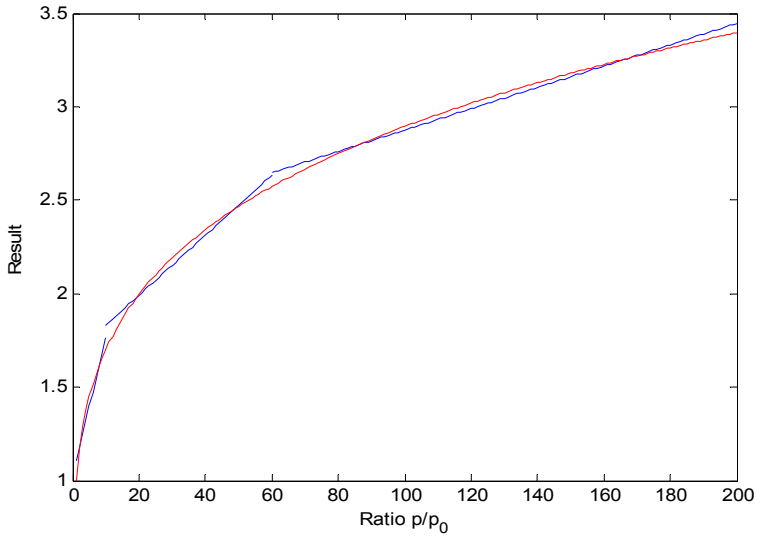

Fig. 6.22 – Isentropic compression calculations with a fixed gamma, $\gamma = 1.3$. The three segments match the analytical calculation quite decently, in the interval of interest.

In the case when multivariable functions are interpolated, the approach was to produce a least−squares fit in several variables, as described in Section 5.2. For example, the results for the specific heat is given in Fig. 6.25. As can be seen the least square fit gives results that can be used in practice. Unfortunately, there are times when the least−squares fit gives inappropriate results. However, these cases are extreme and do not occur during engine combustion. As an example, the approximation of the Woschni heat–transfer coefficient would give a negative result when the pressure is at 1 bar and a temperature at [1500, 2000] K. This is not a realistic condition during combustion in a diesel engine (see Fig. 6.26).

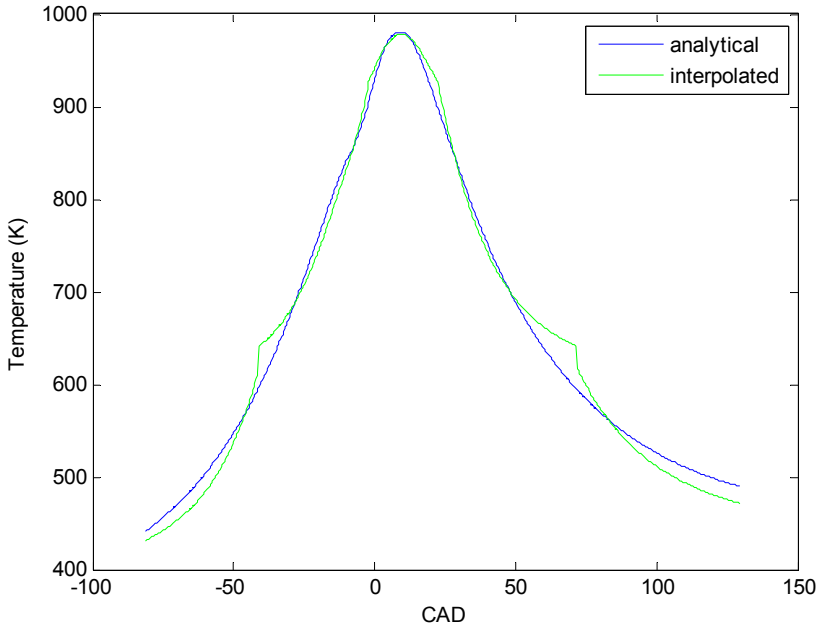

Fig. 6.23 – Comparison between using exact isentropic equation and discretized. The spikes in the interpolated version are due to discontinuities in the interpolation, as can be seen in Fig. 6.17, but this can be smoothed out.

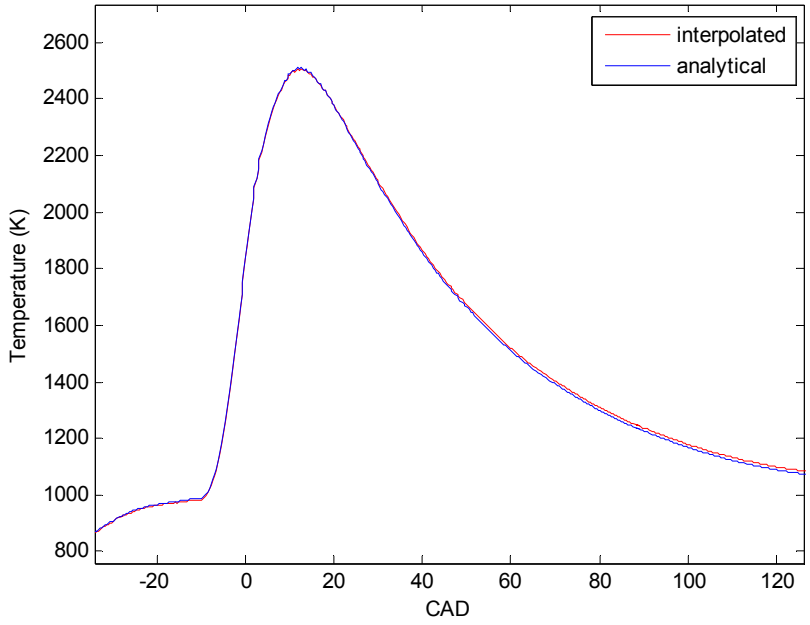

Fig. 6.24 – Interpolated temperature (red) of burned zone versus the analytical calculation of the same. As can be seen the interpolation gives a result that follows the analytical one very accurately.

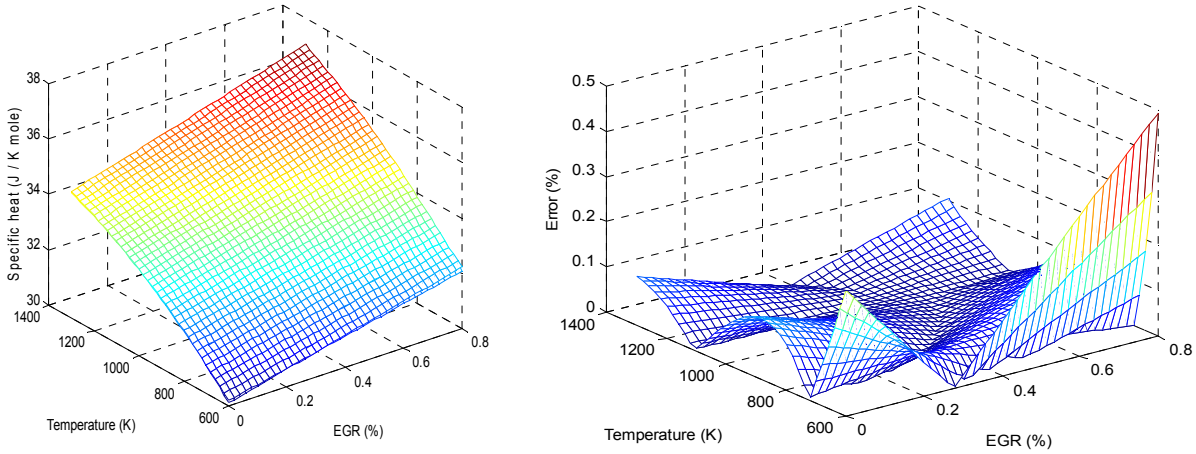

Fig. 6.25 – *Specific heat interpolation results together with an error plot. The interpolation was done at a fixed global lambda, λ_{global} = 1.3. A value which is fairly usual in practice.*

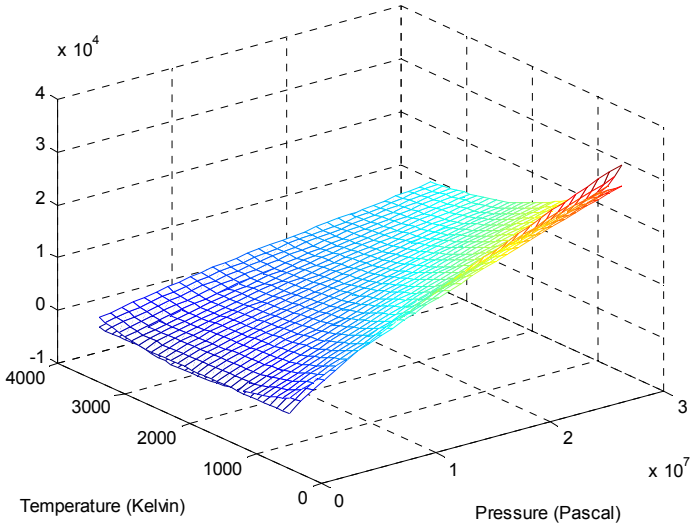

Fig. 6.26 – Approximation of the factor $p^{0.8}T^{-0.55}$ which is used in Woschni (see Eq. (3.30)). The interpolation, approximated with least–squares calculations, is the one that is below the other. Alternatively, $p^{0.8}$ and $T^{-0.55}$ could be interpolated separately due to the trivial multiplicative nature of the factor.

7 Calculations of time efficiency

In this section a study of the time efficiency of the different algorithms and calculations is conducted. Because of the fact that the final model will be implemented on an FPGA, only calculation operations such as multiplication, addition, divisions and subtractions are accounted in the analysis, and not actual time per se. Another aspect to keep in mind is that the FPGA runs in a truly parallel manner. For example, this implies that the calculation $x^2 + y^2$ can be done in only two clock cycles. How so? The parallel behavior of the FPGA gives the possibility to calculate x^2 and y^2 at the same time before being added together.

7.1 Division algorithm

The main purpose of the division algorithm was to avoid "conventional" division on an FPGA, which is proven to be rather time consuming. A conventional division consumes at least 20 clock cycles and also occupies a substantial amount of memory[25]. The time efficiency (number of operations) for the different proposed ways to approximate division is illustrated in Table 7.1. Calculations that can be done in parallel are only counted once in the table, i.e considerations are taken to the FPGA–implementation and its parallel execution.

Algorithm		Additions	Subtractions	Multiplications	Bitshifts	Σ
Eq. (5.16)	First	2	1	2	2	7
Eq. (6.8)	Improved	2	0	2	2	6
Eq. (6.11)	Best fit	3	0	3	2	8

Table 7.1 – Number of operations for each division approximation calculation. As can be seen, the approximation described by Eq. (6.8) only needs a total of 6 mathematical operations.

The most promising one seems to be Eq. (6.8) in terms of speed and accuracy. After all it is possible to bit–shift into a range $x \in [0.0,0.25]$, as has been stated previously.

7.2 Specific heat interpolation

An approximative approach to calculate the specific heat at constant pressure is presented in this thesis. The method was to utilize a multivariable least square approach to fit a function $f(EGR,T)$ to a set of output data (i.e. specific heats). The other approach would of course be to use NASA–interpolations and calculations as suggested by Eq. (6.2) and Eq. (6.3). The results in section 6.7 indicate that the least square interpolation gives highly accurate fits. However, another interesting aspect is of course the time saving of doing so. This is in fact the main idea of the interpolation in the first place: saving calculation time.

The interpolation function of the specific heats of interest, in the NASA Glenn table, is on a form:

Eq. (7.1) $$\frac{C_p}{R} = a_1 T^{-2} + a_2 T^{-1} + a_3 + a_4 T + a_5 T^2 + a_6 T^3 + a_7 T^4$$

This interpolation function holds for all the molecular compounds used in the calculations, which are CO_2, H_2O, O_2 and N_2. There is also the calculations in Eq. (6.2) and Eq. (6.3) that have to be counted. The time comparison between the interpolation and the analytical calculation of the specific heat is given in Table 7.2. Calculations that can be done a priori is of course neglected.

Method	Subtractions	Additions	Multiplications	Divisions	Σ
Analytical	2	30	55	9	96
Interpolation	0	5	8	0	13

Table 7.2– The number of operations needed in the two different approaches to calculate the specific heat at constant pressure needed in temperature and γ calculations.

Yet again, considerations to the fact that FPGA runs in a true parallel manner should be taken into account. Thus the results in Table 7.2 do not correctly mirror the possibility to, for example, calculate Eq. (7.2) in parallel for each molecule: N_2, O_2, H_2O and CO_2. In an FPGA implementation, the number of operations would instead a lot less, which can be seen in Table 7.3. However, it seems reasonable to prefer the interpolation function for specific heat calculations instead.

Method	Subtractions	Additions	Multiplications	Divisions	Σ
Analytical	2	12	22	3	39
Interpolation	0	5	2	0	7

Table 7.3 – Number of operations needed in an FPGA case when true parallelism can be utilized to save calculation time.

The interpolation still saves some considerable computation time, even in an FPGA implementation. Additionally, less memory is needed to store constants and coefficients.

8 Discussion

The heat release calculations are coherent with the heat release calculation (in this thesis the HR is incorrectly called "measured HR") done at a Scania predevelopment group, considering the fact that the automatic heat release calculation at Scania is not 100% accurate with respect to the actual heat–release, as well. The Savitzky–Golay filter[23] successfully removes the high frequency calculation noise. In a real time implementation it would be wise to add some kind of electrical filter to filter the pressure measurements. Another alternative would be to 'smooth' out the derivative part of the pressure in the heat release equation. Also, introducing an adaptive γ did not contribute much change to the heat release. At least not on the particular set of data used in the validation process. However, there are reasons to believe that combustion at small global λ with high EGR rates will make the adaptive γ more important in heat release calculations. It is also quite easy to calculate the γ using the least square fit polynomial of the specific heat at constant pressure, knowing that $c_v = c_p - R$ and $\gamma = c_p/c_v$.

The approach to calculate the temperature in this thesis seems promising and easy enough for a future implementation on an FPGA. It also accounts the impact of EGR on temperature in a trivial manner, through changing mean specific heat. Of course, the approach assumes a homogeneous content in the engine cylinder. The least–squares interpolation of the mean specific heat heavily reduces calculation time at a very low accuracy cost of below 0.5%. Such a small percentage is in fact negligible compared to model errors, and quantization effects in a fixed point implemented code. The temperature calculation results show that the burned zone will be the warmest zone, followed by the cylinder mean temperature and lastly, the unburned zone. In fact, this result is expected. However, large temperature gradients should arise between the two zones which would render heat transfers between the burned and unburned zone. One can easily question the plausibility of assuming zero heat transfers between the two zones in question. The assumption is however, as stated previously, quite standard in combustion modeling.

The NO_x results indicate that the model of NO_x formation is at least acceptable, even if it lacks a realistic physical interpretation. The errors are quite large at over 20% and it would indicate that more work can be done on parameter tuning and model modification.

When it comes to calculating equilibrium concentrations (see theory section 3.6), no time efficient method was found for solving the equations.
The Newton−Raphson method developed in this thesis needs a relaxation matrix

which will slow down the algorithm as a whole, because more iterations will be needed.

The suggested algorithms for approximation of the division operation seem to be promising. Especially the Eq. (6.8) indicates two essential properties: it can be highly accurate for small values of x and it does not require a lot of operations. In a NO_x model, a large number of the quantities relevant are small magnitude. For example, the total amount of substance in the engine cylinder is in the region 0.2 mole, also using SI−units will give small values of volume, equilibrium concentrations etc. This implies that the requirement $x \ll 1$ in Eq. (6.8) will be fulfilled in many cases.

9 Conclusions

In this thesis a NO_x model is derived for the purposes of calculation of NO_x output from heavy duty diesel engines. The model is zero−dimensional and it also utilizes two zones during the combustion phase.

The reactions occurring in the engine cylinder are hard to describe in full extension, especially if the purpose is to simplify things for future real time use. However, considering the results, the three reactions in the Zeldovich mechanism together with the assumption that the system reaches chemical equilibrium seems to be a good approach to a simple NO_x model.

The heat release is calculated by using the diagnostic ROHR−equation. The results are coherent with data received from Scania CV AB. Although, the gross heat release is harder to calculate considering the fact that calibration of the Woschni heat−transfer equation is needed. One way to do this is of course to compare the accumulated heat release with the energy content of the fuel injected.

The division algorithm derived in this thesis can reduce the number of operations needed for a division operation. However, an implementation and validation procedure on an FPGA is needed for a final confirmation of this conclusion.

The least–squares fitting of specific heats give a very accurate fit with errors below 0.5%. This is negligible compared to other errors, such as tuning of the parameters used in the Woschni heat–transfer equation. The least–squares fitting of specific heats reduce the amount of operations and also less memory usage is needed.

10 Future work

The natural continuation in the process of creating a fast and sufficiently accurate NO_x model would be to:

(1) Modify the model to include radiative heat losses in the heat release equation.
(2) Implement and utilize a predictive heat release model instead of the current diagnostic one. This would of course need an additional model of the combustion process and fuel injection. A suitable approach is described extensively in Andersson's Licentiate dissertation.
(3) Produce memory maps of NO_x concentrations as functions of temperature, pressure and λ, instead of using the equations derived from the Zeldovich mechanism.
(4) Calibrate the model using larger data sets to produce a more accurate model of NO_x formation.
(5) Implement the model on an FPGA.

Another interesting investigation would be to study the possibility to derive a black box model of NO_x formation. The input data that could be used in such a model are: pressure trace, EGR rate and cylinder temperature. Alternatively, one could implement a gray box model. After all, the heat release and temperature can easily be calculated using pressure data and the diagnostic heat release equation. Knowing these two, it would be possible to use system identification to produce a NO_x model with burned zone temperature, reaction rates and heat release as input data.

Nomenclature

m_{XX}	Mass of element XX in kilograms (kg)
M_{XX}	Molar mass of element XX in kilograms $(kg/mole)$
p	Pressure in Pascal (Pa)
R	Universal gas constant $(J/K \cdot mole)$
V	Volume in cubic meters (m^3)
V_d	Displacement volume (m^3)
V_c	Compression volume (m^3)
T	Temperature in Kelvin (K)
φ	Crank angle in CAD $(^o)$
ψ	Stoichiometric air−to−fuel ratio $(-)$
γ	Polytropic exponent $(-)$
c_p	Specific heat at constant pressure $(J/K \cdot mole)$
c_v	Specific heat at constant volume $(J/K \cdot mole)$
n	Amount of substance $(mole)$
B	Cylinder bore (m^2)
l	Conrod length (m)
$EGR\%$	Percentage of EGR in gas content (–)

11 References

1: M. Andersson, "*Fast NOx Prediction in Diesel Engines*", Licentiate Thesis, Division of Combustion Engines, Department of Energy Sciences, Lund

2: Euro VI Regulations for emissions from Lorries and Buses, EEB position paper, European Environmental Bureau, Contact person: Dragomira Raeva, dragomira.raeva@eeb.org, 2008.

3: C.G Zander, Single–Cylinder Diesel Engine Experiments, Modeling and In–Cycle Control with Heat–realease Emphasis, Division of Combustion Engines, Department of Energy Sciences, Lund*, 2010.*

4: J.E Dec, A conceptual model of DI diesel combustion based on laser–sheet imaging, SAE technical paper, 1997.

5: J. B. Heywood, Internal Combustion Engine Fundamentals, 1988, ISBN 0–07–100499–8.

6: *http://www.tc.gc.ca/, accessed 21 June, 2011.*

7: http://www.cormetech.com/catalystoverview.htm, accessed 23 march 2011.

8: www.afcd.energy.gov, accessed 20 June, 2011.

9: R. Egnell, On Zero–dimensional Modelling of Combustion and NOx formation in Diesel Engines, 2001, ISSN: 0282-1990.

10: Long Liang, Efficient Simulation of Diesel Engine Combustion Using Realistic Chemical Kinetics in CFD, SAE International, 2010-01-0178, 2010.

11: http://www.ngk.co.jp/english/news/2008/image/0611_06.jpg, accessed March 24, 2011.

12: R. Wain., An overview of FPGAs and FPGA programming; Initial experience at Daresbury, CCLRC Daresbury Laboratory, November 2006.

13: C. Wilhelmsson, Embedded Systems and FPGAs for implementation of control oriented Models, November 2009, LUTMDN/TMHP—09/1068--SE.

14: http://en.wikipedia.org/wiki/Field-programmable_gate_array, accessed 24th March, 2011.

15: http://www.intertechnology.com/Kistler/pdfs/Pressure_Model_6013C.pdf, accessed 6th September, 2011.

16:http://energihandbok.se/x/a/i/10198/NOx-bildningen-vid-forbranning.html, accessed, 30 march, 2011.

17: G. Woschni, "A universally applicable equation for instantaneous heat transfer coefficient in the internal combustion engine", SAE technical Papers 670931, 1967.

18: B. Johansson, Förbränningsmotorer, Division of Combustion Engines, Department of Energy Sciences, Lund*, 2006.*

19: B.J. McBridge et al., Nasa Glenn Coefficients for calculating Thermodynamic properties of Individual Species, TP-2002-211 556, 2002.

20: H. S. Warren, Hacker's delight, 2002, ISBN: 978-0-201-91465-8.

21: D. Eberly, Least–squares Fitting of Data, 2008, http://www.geometrictools.com/, accessed, June, 2011.

22: A. Savitzky & M.J Golay, Smoothing and differentiation of data by simplified least squares procedures, Analytical Chemistry, 36, p. 1627–1639.

23: P. Tunestål et al., Validation of a Self Tuning Gross Heat Release Algorithm, SAE paper, 2008-01-1672, 2008.

24: C. Ericson, Model Based Optimization of a Complete Diesel Engine/SCR System, 2009. ISBN: 978–91–628–7724–8.

25: C.G Zander, Internal document (document name/label code is missing), 2011.