A Fast Physical NOx Model Implemented on an Embedded System

Wilhelmsson, Carl; Tunestål, Per; Widd, Anders; Johansson, Rolf

Published in:
Proc. IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling (ECOSM 2009), Nov 30-Dec 2, 2009, Malmaison, Rueil, France

2009

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
A Fast Control Oriented Physical $NO_x$ Model for Embedded Use

Carl Wilhelmsson * Per Tunestål * Anders Widd **
Rolf Johansson **

* Division of Combustion Engines, Department of Energy Sciences, Lund University
** Department of Automatic Control, Lund University

Abstract: This paper offers a two-zone, physical, $NO_x$ model with low computational cost, suitable for vehicle on-board implementation. The paper introduces a model which is able to compute $NO_x$ emission formation with high time resolution during an engine cycle. The physical background is described as well as the equations upon which the model is based. The model was developed with the structure suitable for implementation in embedded systems. Large parts of the effort has been devoted to develop an algorithm implementing the described physical model and techniques used and issues encountered are described in the paper. Ease in computation has been a top priority, making the algorithm implementation feasible in some sort of embedded system, e.g. embedded processor or embedded electronic hardware (FPGA). For the sake of implementation, parts of the algorithm had to be pre-computed and stored in tables, allowing significant acceleration of the computations. Since the model is non-linear, exponentially spaced tables had to be developed in order to successfully tabulate the parts needed without consuming too much memory. The outcome regarding number of operations, memory requirement and feasible computation speed are discussed. The final result is a low-cost $NO_x$ algorithm (implementing a physical $NO_x$ model) which is able to compute several orders of magnitude faster than the $NO_x$ models known so far.

Keywords: Embedded Systems, Micro-controllers, Field Programmable Gate Arrays, FPGA, Mathematical model, Physical model, Algorithms, Air pollution, Emissions, Nitrogen Oxides, NO, NOx, Engine, Diesel Engine, Internal Combustion Engine.

1. INTRODUCTION

To this day there are according to [1] no physical, time resolved (non-mean value), $NO_x$ model simple enough for on-board application. The reason is that the physical $NO_x$ models generally require a lot of complex computations which frequently are iterative by nature. Current engine controllers often do not have that much ‘spare’ computation time (obviously) and $NO_x$ computations within an engine cycle is out of the question using conventional implementation platforms and algorithms. Several authors have made attempts in finding $NO_x$ models which are physical (or at least semi-physical) and still usable in engine control units, for example [2] and [3], the second being based on the first. Andersson et. al. shows a multi-zone model accelerated using partial pre-calculation techniques. The claim of [2] is “real-time” performance of the model, the platform was however a (for the time) high-spec desktop computer, avoiding many of the obstacles present in on-board computation systems. Another contribution in the area is given by [4] on which the model part of this paper (as well as the work of [2] and [3]) is based. The major difference between this model and [4], who utilizes conservation of energy iteratively to find a burned zone temperature, is the observation that it is possible to indirectly compute the temperature of the burned and unburned zones (provided that a two-zone approach is used). The computation of the burned zone temperature is hence exchanged for a physically correct computation which does not require iterations. Computational speed can hence be increased without having to introduce the ad-hoc assumptions of for example [2].

The model used here is described in deep detail in [10], proving the mathematical concept and evaluating its performance. In order to explain the process of developing an algorithm implementing the model a rudimentary description of the model is never the less included in the paper.
2. PROBLEM FORMULATION

This work aims to fill the gap in the area of high resolution, on-line NO\textsubscript{x} modeling by developing a computationally efficient and fast algorithm implementing the model described in [10]. The performance goal of the algorithm is 'in-cycle' performance, meaning to be able to compute the NO\textsubscript{x} formation during the same cycle it occurs with a high time resolution. To do this several obstacles had to be overcome, the model had to be made causal, parts of the model had to be tabulated with sufficient resolution; efficient code or computation format had to be developed. The goal was an algorithm small and efficient enough for embedded implementation. The output from the algorithm is intended for closed-loop Diesel combustion control.

3. THE MODEL

3.1 Overview

The model in this work is a two-zone model, one burned zone in which the NO\textsubscript{x} formation takes place and one unburned zone composed of only air, as explained by Fig. 1. In this work the burned zone temperature was computed using knowledge about the number of moles in the different zones and the global temperature as well as the temperature of the unburned zone (under the assumption of isentropic compression), rather than using the iterative energy balance approach. Using this method the physical interpretation can be maintained while the algorithm is significantly simplified.

If the pressure in the cylinder is known (which is a basic condition for much combustion engine related modeling) it is possible to compute the global temperature as well as the temperature of the unburned zone using a number of assumptions. Furthermore it is possible to compute the number of moles in the burned zone using 'conventional' heat release analysis. It is also possible to compute the total number of moles in the combustion chamber and hence the number of moles in the unburned zone. When the number of moles in the unburned zone, burned zone and globally is known as well as the temperature globally and in the unburned zone it is possible to compute the temperature of the burned zone. A more extensive model of the combustion is hence not needed to compute the temperature of the burned zone. It is possible to avoid a direct computation of burned zone temperature and the related complex and iterative numerical solution of an energy balance 'normally' used to determine burned zone temperature and mole content.

3.2 Zone temperatures and number of moles

As stated earlier one of the novel features with this model compared to earlier work is the computation of burned zone temperature. A two-zone approach together with uniform pressure and the ideal gas law for each zone individually as well as for the complete combustion chamber (the sum of both zones) allows computation of the burned zone temperature. Eq. (1) describes an equation for the temperature of the burned zone that depends on the number of moles in the complete combustion chamber, in the burned zone and in the unburned zone and the global temperature in the combustion chamber as well as the temperature of the unburned zone.

\[ T_{bz} = \frac{n_g T_g - (n_g - n_{bz}) T_{uz}}{n_{bz}} \]

Given the cylinder pressure and assuming that the gas temperature in the cylinder just after Inlet Valve Close (IVC) is roughly equal to the intake temperature it is possible to compute the initial state in the cylinder. The initial state here refers to initial global temperature \( T_0 \) and the total number of moles of air in the cylinder at IVC, \( n_0 \). The number of moles in the entire combustion chamber (the global number of moles, \( n_g \)) can be computed using the ideal gas law based on a small number of data-points just after IVC giving the initial number of moles, \( n_0 \). Assuming that the number of moles is not significantly changed by combustion the global number of moles \( (n_g) \) is known from \( n_0 \) throughout the complete cycle. Furthermore the global temperature \( (T_g) \) can be computed from the cylinder pressure and the global number of moles, \( n_g \) using the ideal gas law, all according to Eq. (2).

\[ n_0 = \frac{p_0 V_0}{R T_0} \]
\[ n_g = n_0 \]
\[ \Rightarrow T_g = \frac{p V}{n_0 R} \]

The unburned zone temperature \( (T_{uz}) \) is computed using a datum point during the cycle and the assumption of an isentropic relationship, Eq. (3), similarly to [4] and [2]. The temperature and pressure datum \( (T_{g0} \text{ and } P_{g0}) \) used to compute the temperature of the unburned zone is acquired just before the combustion has started to reduce the effect of losses, e.g. heat losses and mass loss. The ratio of specific heats, \( \gamma \) is here assumed to be constant.

\[ T_{uz} = T_{g0} \left( \frac{P}{P_{g0}} \right)^{-\frac{1}{\gamma}} \]
The number of moles in the burned zone \( (n_{bz}) \) is obtained from heat release analysis which is significantly more complex than computing the other variables mentioned.

### 3.3 Number of moles in the burned zone

One way to compute the number of moles in the burned zone is to use heat release to compute the amount of released energy. Assuming that combustion does not change the number of moles significantly, it just 'moves' moles from the unburned zone to the burned zone it is possible to compute the number of moles in the burned zone from the heat release. It does however require some further assumptions, the heating value of the fuel must be known and the combustion efficiency is assumed to be 100%. The local air/fuel ratio, \( \lambda \), must also be known. Knowing the exact local \( \lambda \) is difficult, \( \lambda_{cal} \) is hence introduced as a tuning parameter in the model in a fashion similar to [4], hence combustion is assumed to take place at a constant \( \lambda \).

The heat release equation, Eq. (4), used in this work represents an integrated version of the apparent heat release equation in [5] assuming constant \( \gamma \). It was first presented in [9] and has been applied in e.g. [8].

\[
Q = \frac{1}{\gamma - 1} \frac{p(\alpha)V(\alpha)}{M_f} \left( \int_{\alpha_{start}}^{\alpha} p(\alpha) dV d\alpha \right) - \frac{1}{\gamma - 1} \frac{p(\alpha_{start})V(\alpha_{start})}{M_f}
\]

Both heat losses to the combustion chamber walls and the effect of varying \( \gamma \) is neglected in this step in order to decrease the computational load. The entire heat release was then normalized against the energy content of the injected fuel (assuming 100% combustion efficiency), in this way the 'shape' of the heat release is maintained as well as a full physical interpretation, while still avoiding extensive models for varying \( \gamma \) and heat losses.

Having an accurate heat release analysis, it is possible to compute the number of moles in the burned zone from Eq. (5) which uses average molar masses of fuel and air together with local, burned zone, \( \lambda \) (\( \lambda_{cal} \)) and the stoichiometric air fuel ratio \( (\alpha_{safr}) \).

\[
n_{bz} = \frac{Q}{Q_{HV,M_f}} \left( 1 + \frac{M_f \lambda_{cal} \alpha_{safr}}{M_a} \right)
\]

**Computation trigger** The computation of the number of moles in the burned zone can not be started directly after IVC and this is the case for some of the other computations as well. The reason is that the burned zone contains zero moles until combustion has started and for example Eq. (1) will hence include a divide by zero. A computation trigger hence has to be introduced in order to avoid numerical problems before combustion has started. For that purpose the time instance when 10% of the total heat has been released (CA10%) was used.

### 3.4 Burned Zone Composition

To be able to compute \( NO \) formation rate (using Eq. (12)) the concentrations of a number of species in the burned zone have to be known. It is possible to compute those based on the assumption that corresponding species have reached chemical equilibrium. To compute \( NO \) emissions using the Zeldovich mechanism the equilibrium concentration of free oxygen (O), nitrogen (\( N_2 \)) and nitrogen oxide (\( NO \)) must be known (see Eq. (12)). These concentrations can be computed as functions of temperature, pressure-ratio and local \( \lambda \) by taking dissociation reactions into account. The dissociation reactions (meaning the 'backwards' reactions) are considered important since they provide the most important source of free oxygen available for \( NO \) formation after the combustion event. Dissociation reactions are taken into account through modeling the two chemical reactions shown below:

\[
CO + H_2O \leftrightarrow H_2 + CO_2 \\
CO + \frac{1}{2}O_2 \leftrightarrow CO_2
\]

The approach of [4] and [7] was used, two reactions (Reaction 6 and Reaction 7), relative Gibbs energy and carbon, hydrogen, oxygen and nitrogen balances are put together to form a nonlinear equation system with six unknown variables the equations is explained in detail in [7]. Solving these equations gives dimensionless equilibrium concentrations of \( CO_2 \) and \( H_2O \) from which the concentrations \( CO \), \( H_2 \), \( O_2 \) and \( N_2 \) can be computed.

All of the variables needed to compute \( dNO/dt \) are however not yet known. \( N_2 \) is known but \( O \) and \( NO \) have to be found. This requires models for the reactions forming \( O \) from \( O_2 \) (Reaction 8) and \( NO \) from \( O_2 \) and \( N_2 \) (Reaction 9).

\[
O_2 \leftrightarrow O + O
\]

\[
O_2 + N_2 \leftrightarrow 2NO
\]

All equilibrium concentrations needs to be computed with an angular resolution of 0.2 CAD within each cycle. Knowing equilibrium concentrations of \( O \), \( NO \) and \( N_2 \) allows computation of \( dNO/dt \).

### 3.5 \( NO \) formation

Once the equilibrium concentrations of the different species are known it is possible to compute the \( NO \) formation rate using the Zeldovich mechanism shown in Eq. (12). The Zeldovich mechanism which models Reaction 10 and Reaction 11 is commonly used for \( NO \) modeling.

\[
O + N_2 \leftrightarrow NO + N
\]

\[
N + O_2 \leftrightarrow NO + O
\]

\[
\frac{d\xi_{NO}}{dt} = \frac{15.2 \times 10^{13} e^{-38000 T_a} c_0^7 c_{NO}^5 (1 - (\frac{c_{NO}}{c_{NO,r}})^2)}{1 + 7.6 \times 10^{14} e^{-38000 T_a} c_0^7 c_{NO}^5 (\frac{c_{NO}}{c_{NO,r}})}
\]
During this work it was however unveiled that the original Zeldovich mechanism (which equals Eq. 12) as presented by [4] and [6] actually is not valid if the volume of the zone it is applied to varies! For a two-zone model it is obviously the case that the volume of the burned zone increases as combustion progresses and more and more moles are included in the burned zone. Eq. (12) hence had to be rewritten to form Eq. (13) (which from now on is denoted the modified Zeldovich mechanism) in order to be valid even when the volume of the burned zone varies.

$$\frac{dc_{NO}}{dt} = \frac{\partial c_{NO}}{\partial t} - \frac{c_{NO}}{V} \frac{dV}{dt}$$ (13)

$\frac{dc_{NO}}{dt}$ is given by the original Zeldovich mechanism in Eq. (12), $\frac{dV}{dt}$ and $V$ represent volume derivative and volume, respectively, of the burned zone computed using the ideal gas law and the previously computed temperatures and mole numbers of the burned zone. Solving the modified Zeldovich in Eq. (13), which can be done for example using a simple Euler method, gives the NO concentration (the unit is $[\text{mole/cm}^3]$) of the burned zone! It is however the concentration of NO in the exhaust gases that is to be determined by the model. To compute the overall NO fraction the 'size' of the burned zone must be divided by the 'size' of the total combustion chamber. Since NO tail-pipe emissions commonly are measured as a mole-based fraction the best way to compute overall NO is to compute the number of moles of NO in the burned zone and divide it by the total, global, number of moles. The procedure is shown in Eq. (14).

$$X_{NO} = \frac{V_{bz}c_{NO}}{n_g}$$ (14)

This actually is the final step of the NO model. The fraction of NO in the exhaust gases is now known throughout the engine cycle, a typical NO formation case for 100 cycles is shown in bottom part of Fig. 4.

4. THE ALGORITHM

4.1 Overview

Implementing the model described above took quite some consideration. The aim throughout the work was to be able to model NO during the cycle and having to wait for the cycle to complete in order to normalize the heat release and compute the trigger point (CA10%) was hence never an option. To make it possible to compute the model during the cycle the algorithm had to to be made causal, meaning that it does not use values 'from the future'. Implementing algorithms in embedded systems often takes special considerations, in such environments there is a more limited infrastructure and computations hence must be adapted to that. For the sake of limiting hardware use and maximizing speed simplifications has to be made to the model when implementing it as an algorithm. Such simplifications were made e.g. selecting filter coefficients so that they are power-of-two and computing CA12.5% instead of CA10%, avoiding one division. Last but not least, as large a part of the model as possible should be pre-computed and stored in memory to reduce the computational load at run-time.

4.2 Making the Model Causal

As mentioned previously the NO computation was triggered at CA12.5% and the heat release was normalized against the total energy of the fuel injected. To perform these procedures two points were needed; the minimum and maximum point of the heat release curve. When computing the NO model off-line, using recorded data, this would cause a problem. However since the computations were to be carried out during the cycle issues arose with the important maximum and minimum points of the heat release. A naive way to solve this was to just use the maximum and minimum points from the previous cycle. Doing so did however give rise to some practical issues; oscillations occurred in the two variables. The oscillations could be removed using third-order low-pass FIR filters on the corresponding variables. Having a very short step response, no amplification and with filter coefficients selected so that division could be avoided the two filters had virtually no negative impact on the algorithm as such. The filters successfully removed the issues with oscillations and made it possible to develop a causal version of the algorithm.

4.3 Pre-calculation and tabulating

In order to achieve high computation speed it is essential to avoid solving the nonlinear equation system, describing the dissociation and Zeldovich reactions, at run-time. Even though it might be possible to implement a fast solver in some sort of electronic hardware it would be much easier and significantly faster to tabulate the result and store it in memory. This action was performed using the observation that $dc_{NO}/dt$ actually is a function of only three variables; burned zone temperature, the pressure ratio in the cylinder and current NO concentration as shown in Eq. (15). Since the number of input variables was not too large it was possible to use pre-computed values to avoid solving the nonlinear equations and the 'Zeldovich expression' at run-time.

$$\frac{dc_{NO}}{dt} = f(T_{bz}, c_{NO}, c_{NO}, c_{NO}, d_{NO}) = f(T_{bz}, P, c_{NO})$$ (15)

The equations involved were however non-linear and performing this pre-computation yields results tables like the surface shown in Fig. 2. An attempt was made to tabulate these functions using evenly (linearly) spaced breakpoints as a 'grid', the table look-up was performed using rounding towards nearest value. Using such simple logic proved not to work very well and a second attempt was made using a linearly spaced table together with trilinear interpolation. Even tough it significantly improved the result it was not good enough, at the same time the trilinear interpolation required 24 multiplications and 19 additions to be carried out on-line. Such a computational cost is un-acceptable considering that the interpolation would demand more multiplication operations and about the same amount of
addition operations as the rest of the model, alternatives had to be found.

One possibility was to use unevenly spaced table breakpoints and exponential spacing between the breakpoints would have been a logical choice considering Fig. 2. Exponential spacing would however have the drawback that it would require significant computation effort to compute the indices when performing look-up in the table at run-time. As an intermediate approach, avoiding the difficulties computing exponential index, piecewise linear polynomials were used. The polynomials were selected in an 'ad-hoc' manner studying different surface plots similar to Fig. 2. It is important to try to keep the order of the polynomials as low as possible to reduce the number of tests needed to implement the algorithm (thus reducing program size or hardware requirement). Another factor to take into account when producing these tables is the fact that most devices have limited amount of memory available to store the tables. Naturally we end up in a precision/consumption trade-off situation when increasing the degree of the polynomials and/or the length of the tables. The three different polynomials which were used as look-up indices are shown in Fig. 3, they together represent the 'grid' of the table.

It was not only the NO formation mechanism that had to be tabulated, the isentropic relationship (Eq. 3) was also too complex to be computed at run-time, hence Eq. 3 was tabulated as well. As was the case with combustion chamber volume and derivative volume, which both is a function of crank angle. In the two later cases normal linear tables with round-to-nearest look-up was found sufficient.

5. EXPERIMENTAL PLATFORM

All computation and simulation has been carried out using Matlab running on a normal desktop PC. As a 'benchmark' comparison a 'golden model' was used. This golden model was exactly the same as presented in [10].

Five data points consisting of some 150 cycles each was used for the validation of the algorithm. The data was obtained on a single-cylinder version of a passenger-car sized Volvo 'D5' diesel engine (0.5l displacement per cylinder). The data points were taken at increasing loads with varying intake-pressure, no external or internal Exhaust Gas Recirculation was present. Global λ hence varied between about 4 and 1.5, and one could probably say that the data-points represent a common Diesel load sweep.

6. RESULTS AND PERFORMANCE

The results in this work was obtained by comparing the outcome of the algorithm with the 'golden model', such comparisons can be seen in Fig. 5 - Fig. 6. The first figure show 100 cycles average when testing the algorithm on an average data point, Fig. 5 correspond to Fig. 4. A key part of the model was the actual look-up operation using exponentially spaced tables shown in the upper part of Fig. 5. The look-up gives the NO formation rate, \( \partial c_{NO}/\partial t \), during the cycle provided that combustion zone temperature etc. is computed correctly. Using the NO formation rate it is possible to compute the overall NO concentration, as described earlier, which is shown in the lower part of the figure. Fig. 4 and Fig. 5 indicate the signal quality and precision of the algorithm, alone and compared to the golden model.

The only parameter available for 'tuning' in the model namely \( \lambda_{cal} \) was swept with high resolution between 0.95 and 1.2 both for the 'golden model' and for the tabulated algorithm, the result is shown in Fig. 6. The vertical lines indicate which average \( \lambda_{cal} \) that gave best agreement between measured NO emission and computed NO for all of the five data-points. Solid lines shows the output of the golden model and dashed lines correspond to the tabulated algorithm. The circles and squares which are also included in the figure indicate the measured NO emission value for each data-point put on the curves of the golden model and the algorithm respectively.

Besides the 'accuracy part' it should be stated that the algorithm needs 15 shift operations, 24 additions, 21 multiplications and 10 divisions, in total 70 operations excluding a few memory access operations, to process one sample. From which only the heat release part of the
algorithm requires 29 operations (10 shift, 11 additions, 7 multiplications and 1 division).

The tables generated needs to be stored in some sort of memory and the size of the tables are hence of great importance. The results presented are obtained using a NO formation-rate table with 6-bit (64 levels) resolution with respect to burned-zone temperature and pressure ratio, current NO concentration was tabulated with a resolution of 5-bits (32 levels). The total table size was hence \( 64 \cdot 64 \cdot 32 = 131072 \) points. Added to the large NO formation-rate table other tables were needed. To compute the isentropic relationship a 8-bit (256 values) table was used. Further on two tables containing the combustion chamber volume and derivative combustion chamber volume as a function of crank-angle were needed, they spanned over 3600 values respectively.

7. DISCUSSION

Taking a closer look at Fig. 5 it is obvious that the tabulated algorithm is a sufficient implementation of the NO model. The output from the algorithm is in-fact within \( \pm 10\% \) compared to the golden model, cycle-by-cycle and sample-by-sample. On average the algorithm performs much better than 10\%. The table look-up worked very well on average as indicated by the upper part of Fig. 5, agreement between the NO formation rate from the model and the one computed using the exponential table is excellent. Another interesting result is that the signal quality of NO formation rate computed using the table is good enough to use directly, even though the signal is somewhat affected by the switching-noise originating from the table look-up procedure. Attempts made using ‘normal’ linearly spaced tables could not obtain such good result by far. Neither regarding agreement with the golden model or signal quality of the formation rate. The exponentially spaced table surely was a key component in developing the algorithm. Another interesting thing to notice is that even though the NO formation rate signal (upper part Fig. 4) has a significant switching-noise level none of the noise is propagated through, neither to the \( \epsilon_{CN2O} \) signal, nor to the total NO (lower part Fig. 4). The reason for this is that the Euler method used to solve Eq. 13 has strong low-pass characteristics and the switching-noise present on the NO formation-rate signal is hence less of a problem, it is the end NO that is of interest. Considering the high precision obtained using the exponential look-up table (as can be seen in the upper part of Fig. 5) it might seem surprising that the agreement in the lower part of the figure is not better. In order to have a good agreement on the computation of total NO fraction there are however more things needed than ‘only’
the NO formation-rate, the 'size' of the combustion zone must be computed as well (according Eq. 14). To compute the 'size' the output from the heat release comes into play. Since the heat release had to be computed causally it is impossible to obtain the same precision as when computing the heat release off-line. This results in a computation of combustion-zone volume which is not as precise as the one in the golden model, in turn resulting in slight errors in the combustion-zone volume which is not as precise as the one computed in the golden model, in turn resulting in slight errors in the computation of overall NO fraction. Such an explanation gains support investigating Fig. 7 which shows the average error of NO fraction, heat release and table look-up. The figure shows that the table look-up is able to perform within a few percent in the interesting area 360CAD – 385CAD which is where NO formation takes place. After 385CAD the relative error is large due to the fact that there is no NO formation (a division-by-zero effect). The heat release on the other hand has an average difference of roughly 7J compared to the golden model. In the data-point at hand 7J is about 10% which explains a large part of the average error found in the lower part of Fig. 5. Finding a more accurate but equally fast version of the heat release may hence improve the accuracy of the model.

Another examination which was performed was to vary the calibration variable, λcal, between 0.9 and 1.2. The result from both the golden model (in solid lines) and the algorithm (in dashed lines) is available in Fig. 6. Ideally λcal would be expected to be very close to one but the golden model performs best at a λcal slightly higher (leaner) than one as previously reported, so is the case with the algorithm. The difference between the golden model and the algorithm in the figure is not large, the shapes of the curves are the same and none of the curves have changed their internal order. Basically the only thing differing is a slight bias on the curves belonging to the algorithm. It is encouraging to see that the golden model and the tabulated algorithm respond similarly to the calibration variable, it actually makes it possible to just 'calibrate away' the difference between the golden model and the algorithm. The algorithm needs a slightly higher value of λcal than the model does. It is important to note that the golden model is not the actual, physical, truth in this context.

One important goal with this work was to develop an algorithm of such a format and with so low demands on computational power so that it can be computed in an embedded system. Computational load of the developed algorithm is only a fraction of the computational load of the original model and the algorithm can hence compute several orders of magnitude faster then the original model, even when implemented in an embedded system. The number of operations required to compute one sample of the model is well within what can be considered as reasonable and computing the algorithm within the engine cycle should be possible even using off-the-shelf embedded processors. With a crank angle resolution of 0.2CAD and a maximum engine speed of 6000rpm the algorithm needs to compute one sample faster than 1/6000/60*3600 = 2.8µs in order to finish before the next sample arrives. Assuming that the processor is able to complete one operation in every clock cycle on average (which is optimistic) the algorithm would demand a processor clock frequency of at least 1/(2.8*10^-6/70) = 25.2MHz, which is fulfilled in most processors intended for embedded systems today. On the other hand it is seldom possible to entirely devote a processor to one computation, there will be other processes competing for processor and memory resources. The processor would neither be able to compute one operation per clock cycle on average even using pipelining. Needless to say the 25.2MHz value represents some sort of absolute minimum needed, in reality the processor would have to be more powerful.

Memory is another limiting factor in this context, the NO formation-rate table requires some memory for its storage. Assuming that the table values are realized using 8bit numbers the table would need 2^6 · 2^6 · 2^5 · 8 = 1048576bit which would occupy about 30% of the on-chip memory of an medium-sized Xilinx Virtex-4 FPGA. Including the other tables into the computation makes no difference, in this context they are small assuming that a word-length of 8bit is sufficient. Processor systems commonly have access to far more memory than FPGAs and the size of the tables needed are not considered a problem at all in modern processor systems.

8. CONCLUSIONS

The presented results show that it is possible to develop an efficient and low-cost algorithm implementing the NO model. The aim of developing a model which can compute with high time resolution during an engine cycle was hence full filled. To achieve the goal a number of obstacles had to be overcome. First of all a causal heat release analysis had to be developed. It was developed using the minimum and maximum heat release points from the previous cycle in combination with two FIR-filters. Complex and non-linear parts of the model had to be implemented using few computations for combustion and NO formation chemistry. These parts could be implemented as a look-up table using the observation that current NO formation rate is a function of; combustion-zone temperature, pressure ratio and current NO concentration. Using linearly spaced look-up tables proved to be insufficient
for tabulating the nonlinear NO formation rate function. Since it would take too much computation to implement an exponentially spaced table a piecewise linear table was deployed. The piecewise linear table resembles an exponential function but is significantly easier to index. Using a look-up table containing 131072 values it was possible to tabulate NO formation rate with sufficient precision. Some switching noise was however impossible to avoid when using look-up, this noise did however disappear 'automatically' due to low-pass characteristics of the model.

The resulting algorithm had very low demands on computational power. In order to complete one sample it takes 15 bit shifts, 24 additions, 21 multiplications and 10 divisions, in total 70 operations. Given the low demands of operations to be performed and considering the speed of ordinary engines it may be possible to compute the model using an off-the-shelf processor running at 25 MHz.

Finally the precision of the algorithm was examined, the output from the algorithm was compared to the output of a 'golden model'. It was found that the NO formation look-up worked very well; in the important regions it performed within a few percent of the NO formation computed by the golden model. The algorithm was also able to successfully compute the total NO fraction with a very good agreement with the golden model. Comparing total NO fraction cycle-by-cycle and sample-by-sample it was well within 10% of the output from the golden model. When varying the calibration variable (λ_cal) the behaviour of the algorithm was also very good. The algorithm behaved in the same way as the golden model but with a slight bias. The algorithm hence had its best agreement with measured values at a slightly higher value of the calibration variable than the golden model. This behavior makes it possible to 'calibrate away' the difference between the golden model and the algorithm. The algorithm was developed successfully and is, at this point, well suited for implementation in an embedded system such as an embedded processor, or FPGA.

REFERENCES


9. NOMENCLATURE

α Engine crank position (crank angle degree)
α_stoic Stoichiometric air fuel ratio
γ Isentropic constant
λ Relative air-fuel ratio
λ_cal Burned zone lambda, calibration variable
CAXX% Time instance for XX% of total heat release
cNO Concentration of NO
c_{XX} Equilibrium concentration of species XX
IVC Inlet Valve Close
M_a Mole mass of air
M_f Mole mass of fuel
NO Nitrogen Oxide
NO_x Nitrogen Oxides including both NO and NO_2
n_0 Number of moles at a zero datum point
n_{b_0} Number of moles of air
n_{b_z} Number of moles of the burned zone
n_{f_s} Number of moles of fuel
n_g Global number of moles
n_{u_z} Mole number of the unburned zone
P Pressure ratio
p Cylinder pressure
p_0 Pressure at a zero datum point
p_{g_0} Global pressure at a zero datum point
Q Released energy
Q_{hv} Lower heating value (of fuel)
R Universal gas constant on mole basis
T_0 Temperature a zero datum point
T_b Temperature of the burned zone
T_p Temperature of the total combustion chamber
T_{g_0} Global temperature at a zero datum point
T_{u_z} Temperature of the unburned zone
V_0 Volume at a zero datum point
V_{b_z} Volume of the burned zone
V_f Volume of the total combustion chamber
V_g Volume of the unburned zone
X_{NO} Fraction of NO (on mole base)

10. CONTACT

Carl Wilhelmsson
www.carlwilhelmsson.com