FPGA Based Engine Feedback Control Algorithms

Wilhelmsson, Carl; Tunestål, Per; Johansson, Bengt

Published in:
FISITA 2006 World Automotive Congress

2006

F2006P039

FPGA Based Engine Feedback Control Algorithms

[1]Wilhelmsson, Carl*, [1]Tunestål, Per, [1]Johansson, Bengt
[1]Division of Combustion Engines, Faculty of Engineering, Lund University, Sweden

KEYWORDS ASIC, closed-loop engine control, FPGA, heat release analysis, virtual sensor.

ABSTRACT High resolution real time heat release analysis will become increasingly important in the future development of engine control systems. The increased demands on efficiency and emissions will put high demands on future engine control. Future engine concepts, for example the HCCI engine concept might crave cylinder pressure based Closed-Loop Combustion Control (CLCC). The analysis of cylinder pressure is a relatively computationally expensive task that is difficult to implement in existing engine controllers due to the real time demands.
This paper describes an approach to obtain such a high speed heat release analysis. The described system could act as a platform for further feedback control experiments. An experimental setup is put together. The heat release algorithm is then developed using MATLAB and SIMULINK. The emerging environment will serve as a prototyping system that can be used for further development of advanced cylinder pressure based feedback control strategies. The performance of the developed algorithm/system is examined in a simulated engine environment.
The heart of the system is a Field Programmable Gate Array (FPGA), an FPGA is best described as an reconfigurable Application Specific Integrated Circuit (ASIC). The usage of an FPGA gives the possibility of very high throughput and very low delay time and jitter of the final system.
This system could of course also be developed using a normal Commersial Of The Shelf (COTS) processor and a Real Time Operating System (RTOS). The high performance that would be needed to calculate the heat release in the desired time in a multi cylinder engine would however put high demands on the used processor; hence the price of the processor might make the system too expensive, the FPGA describes an alternative approach.

INTRODUCTION As commonly known the price of fuel in combination with the urge to create internal combustion engines that are less harmful to the environment pushes the development of new internal combustion engine concepts. In this development new combustion concepts like Homogeneous Charge Compression Ignition (HCCI), Partially Premixed (diesel) Combustion (PPC), Spark Assisted Compression Ignition (SACI), Controlled Auto Ignition (CAI), etc. emerge. These new combustion concepts could all be thought of as variants of the traditional types of combustion (Otto or Diesel) with one thing in common separating the new concepts from conventional Otto and Diesel engines, a lowered maximum combustion temperature. The desire with these new combustion concepts is to lower the emissions both of the green house gas carbon dioxide CO_2 and the other harmful compounds like nitric oxides (NO_x), hydrocarbons (HC), carbon monoxide (CO) and particulate matter (PM). Using the conventional Otto and Diesel engine techniques it is difficult to reduce fuel consumption (and hence CO_2) and at the same time avoid emissions of the other harmful compounds. The conventional Otto

engine fitted with three way catalyst emits very low concentrations of NO_x, CO, HC and PM, it does on the other hand suffer from a bad break efficiency resulting in higher fuel consumption and hence more CO_2 emissions than a corresponding Diesel engine. The Diesel engine on the other hand emits much NO_x and PM but it has a higher efficiency. The new engine concepts are trying to maintain the high efficiency of the Diesel engine and combine this with the low emissions of the other harmful compounds.

The new compression ignited combustion concepts of course also share some drawbacks. Low combustion temperature causes the combustion efficiency to decrease which in turn causes higher emissions of HC compared to the Otto and Diesel engines. Higher HC emissions is however possible to deal with using a cheap and well-known oxidising catalyst. Another, more serious drawback with the low temperature combustion concepts is the lack of a direct means for combustion phasing control. In the Otto engine the sparkplug sets off the combustion, and you hence have a very powerful actuator usable for combustion phasing control. In the Diesel case combustion is simply initiated when fuel is injected into the cylinder through the direct injectors. Most of the low temperature combustion concepts do not have any of these direct actuators, combustion is instead initiated when the charge temperature is high enough for the current fuel-air mixture to autoignite. The strong connection between temperature, fuel properties and start of combustion makes it more difficult to control combustion phasing using the low temperature combustion concepts compared to the Otto and Diesel ones. There are numerous experiments carried out in order to find powerful control means and a number of solutions are suggested in many different publications. Methods using Variable Valve Actuation (VVA), Fast Thermal Management (FTM), Variable Compression Ratio (VCR), dual fuel solution (altering fuel properties), unconventional Diesel injection patterns and using a conventional Otto flame propagation as the trigger for the homogeneous combustion have all been suggested [1]. The main principle behind all of the different control methods is however to control the combustion phasing indirectly by setting the in-cylinder properties for a suitable autoignition timing.

Besides the work of finding appropriate actuators to be used to control the low temperature combustion phasing, important work has been carried out regarding the best method to use the available actuators, i.e. how to implement Closed-Loop Combustion Control (CLCC). Olsson et al. [2] published the first results of CLCC for low temperature combustion, feedback control of a dual fuel full time HCCI engine was implemented using PID controllers. Bengtsson et al. [3] performs a very interesting investigation regarding different feedback control candidates to be used in the feedback control loop, especially dealing with different means of calculating combustion phasing. This is some of the early work, the current direction of the CLCC investigations appear to be more and more complex controller structures and different model based control strategies. Shaver et al. [4] has for example shown the possibility to utilize models in the CLCC loop. The models utilized by Shaver are physics-based and maintained in real time. Besides the methods of introducing physics-based models of different complexity in the CLCC loop, there is also a possibility to use models directly identified from experiments. Bengtsson et al. has in [5] for example shown interesting results using for example Model Predictive Control. The models were here obtained from special identification experiments.

Application Specific Integrated Circuit (ASIC) technology is, as internal combustion engines, an area where development is explosive. ASIC technology is getting more and more available since the introduction of the Field Programmable Gate Array (FPGA). An FPGA can be de-

scribed as a reconfigurable ASIC, and it makes it possible to utilize the high performance of the ASIC without having to pay the cost/time penalty of traditional ASIC design. Using an FPGA it is hence possible to realize small series of fully customized hardware systems allowing very high performance due to the fact that the implementation resides in hardware rather than in software/processor. FPGA:s have already been utilized in a large number of very high performance applications such as batch computing, scientific simulations, real time image treatment/analysis and real time signal processing. Even though FPGA:s can be customized to perform much better than a more general purpose processor on a specific task, it is however not as versatile as a processor. The FPGA internal layout has to be redesigned to fit it's specific task, this redesign is a more complex task than to reprogram a general purpose processor. Since an FPGA features high performance and a processor high flexibility the next given question is if it would be possible to combine these two systems. The answer is yes, it is possible to obtain a system with both high flexibility and high performance combining hardware parts (residing in an FPGA) with software parts (run on a processor) to a complete high performance flexible hardware/software system. A nice survey of the different FPGA related design methodologies is provided by Todman et al. in [8]. A more specialized survey on signal treating applications is provided by Tessie and Burleson in [9].

For successful control of low temperature combustion throughout the operational area it is from the referenced work safe to conclude that a CLCC system with fairly large complexity is needed. One or more models needs to be maintained in real-time by the CLCC system, this and the ever increasing complexity of the control systems constitute the background to this work. From the previous surveys carried out regarding ASIC:s and FPGA:s it is understood that an FPGA could well serve as a very useful tool for implementing CLCC systems. The intention with this paper is to show this possibility and at the same time implement a part of a future CLCC system, namely the Heat Release analysis (HR).

RELATED WORK Even though FPGA technology for some time has been utilized in for example signal processing applications the development of closed-loop control systems residing in FPGA systems seems just to be starting. Not much is published in the area. Two very interesting publications deserve attention. Wei et al. in [10] implemented and evaluated different PID controller architectures residing in an FPGA environment. They also give further references of this topic. The evaluation of different structures of known closed-loop controllers is an important topic in order to find forms of the closed-loop controllers suitable for hardware implementation on an FPGA.
Another publication that deserves attention is He and Ling [11] who implemented and evaluated an MPC controller on an FPGA and performed some desktop experiments Hardware In the Loop Simulation (HILS) applying this controller to a MATLAB model of an aircraft. An FPGA or a mixed processor/FPGA system is a very suitable environment for implementation of MPC controllers on high frequency closed-loop problems.

EXPERIMENTAL SETUP The experimental setup necessary for these experiments consists both of software and hardware. The hardware parts are made up of an FPGA prototype board, a board simulating the engine pulses and cylinder pressure, an expansion module for the FPGA board featuring AD/DA converters and some surrounding circuitry adjusting signal levels etc. The FPGA system is connected to a Personal Computer through a JTAG cable, enabling display

of debugging data, FPGA/PC co-simulation and reconfiguration of the FPGA. The design of the FPGA configuration and the hardware co-simulation are carried out in MATLAB/Simulink with the aid of a Simulink toolbox supplied by Xilinx, "Xilinx System Generator DSP". In order to generate an FPGA design from the Simulink diagrams the Xilinx development suite "Xilinx ISE" is necessary.

FPGA SYSTEM The main part of the setup is the experimental card fitted with the FPGA. The card is a Commercial Off the Shelf (COTS) product supplied by "MEMEC/Avnet", the (rather long) name of the card is "Memec Xilinx Virtex-4 LX XC4VLX25-SF363 LC Kit". As understood the card holds a "Xilinx Virtex-4 LX XC4VLX25-SF363 LC" FPGA, which holds 24,192 logic cells, $168\ Kb$ distributed RAM memory and 448 user IO ports. The board holds support circuitry to develop a complete system. Maximum clock speed of the FPGA is $100\ MHz$ and the clock signal is supplied by an oscillator present on the board, there is also a possibility to provide a custom oscillator. Besides the above mentioned the card holds a number of peripheral devices; $16\ Mb$ Serial Flash for FPGA configuration, $64\ Mb$ DDR SRAM, On-board oscillators, "P160" expansion header, JTAG programming/configuration port, 10/100 Ethernet, Alpha numeric LCD panel, RS232 port, Led:s, pushbuttons, DIL switches and General Purpose IO pinout.

Desired FPGA configuration is either loaded directly onto the FPGA or stored in a serial flash memory produced by Atmel. If the configuration is loaded on to the serial flash it is automatically reloaded onto the FPGA on power-up with the help of a Complex Programmable Logic Device (CPLD). There exists of course other solutions for the FPGA re-configuration that might be more suitable in automotive applications, this approach is however flexible in a development environment. Configuration both of the serial flash and directly of the FPGA are carried out through the JTAG interface with the help of a Xilinx "Parallel Cable 4".

Not included on the basic FPGA board is the Analog to Digital (AD) and Digital to Analog (DA) converters. To gain access to the analogue world, a "bolt-on" circuit board that fits in the "P160" expansion header was used. This board is, as the FPGA board supplied by "Memec/Avnet" and is named "160-Analogue-Kit". The board features dual AD and DA channels. Both AD and DA converters are made by "Burr-Brown". Maximum clock speed of the AD, $AD_{clk} = 53MHz$, the DA handles $DA_{clk} = 165MHz$ maximum. Clock signal is supplied from the FPGA via the "P160" header, the FPGA runs at $100\ MHz$ limiting maximum AD_{clk} to $50\ MHz$ and DA_{clk} to $100\ MHz$. The surrounding circuitry does, on the AD channels, consist of a input buffer, a low-pass filter, a single ended to differential amplifier and latches. The DA has the same latches, an output buffer and a low-pass output filter. The expansion card was, considering the application, less "bolt on" than expected at the time of purchase and modifications were made to the input circuitry of the AD channels in order to adapt the channels to suit the relatively low frequency that is of interest in this application, the DA channels are however left untouched. Overall documentation of the circuits can be found at the "AVNET" homepage.

The total FPGA system price was at the time of purchase $\approx$ €700 and must hence be considered as a low cost system, it is never the less a high performance system!

DESIGN TOOLS As briefly mentioned earlier the design of the FPGA HR algorithm is carried out in MATLAB/Simulink with the help of "Xilinx system generator DSP" (referred to as SGDSP). SGDSP contains a number of Simulink blocks that are already implemented in VHDL, using these blocks it is possible to generate VHDL from a Simulink diagram. FPGA layout with the help of Simulink in DSP applications is discussed by Todman et al. [8] Please note that it is not possible to implement "standard" Simulink blocks in the FPGA, it is necessary to possess a VHDL implementation of the blocks to implement (a number of such blocks comes with SGDSP). This is somewhat limiting. When running the "hardware co-simulation" (compare Hardware In The Loop "HILS") during the debug process. It is however possible to implement "standard" Simulink systems on the PC communicating data with the FPGA through the JTAG interface, this is a handy feature during the debugging process. Note that it is necessary to posses a copy of the Xilinx ISE design suite in order to use SGDSP since SGDSP uses the underlying ISE tools to generate the design files. After co-simulation of the system, generic VHDL files are created from Simulink, processed by the relevant FPGA design tools and downloaded to the serial flash. When the design finally resides on the serial flash the FPGA system is "stand alone" from the computer.

TEST ENVIRONMENT Desktop tests were carried out during the development. The "interface" to the simulated /real engine consists of three signals, Crank Angle Degree Pulses (CADP), Top Dead Center Pulses (TDCP) and analogue cylinder pressure P_{cyl}. Current engine position is assumed to be measured with 0.2 Crank Angle Degree (CAD) accuracy, the engine hence produces 5 CADP every physical CAD. Besides CADP the angle sensor is assumed to give one TDCP every time the engine has revolved two complete revolutions. Figure 1 provides an overview of the experimental setup, this engine "interface" concurs with the setups in [2], [3] and [5].

During the development and testing P_{cyl} was simulated by recorded cylinder pressure traces, both a motored and a fired trace was recorded. The basis for P_{cyl} are pressure traces recorded from the $12\,l$ Scania engine used by Olsson et al in [2], geometric data of this engine is noted in Table 1. A "in house" developed device simulating CADP, TDCP and P_{cyl} synchronously at an engine speed of $1200\,rpm$ was used for system test during the development. P_{cyl} was simulated with a vertical resolution of $8\,bit$, horizontal resolution was 720 samples, that is one sample each CAD. DA conversion was carried out with the help of a R/2R ladder and the output of the device was buffered. The original pressure curve is somewhat distorted due to the limitations of the engine simulator.

FPGA LAYOUT The design that was implemented in the FPGA can be viewed as a DSP design, the different properties of such a design are covered by Todman et al. [8]. No processor core was present in the system. It would however of course be possible to implement the HR algorithm on the reconfigurable fabric of a mixed processor/hardware system, see Todman et al. [8] on the properties of a mixed system. The DSP design approach was selected due to the relative simplicity of the calculations and the system. If the system is expanded to a complete engine control system a processor core might be added

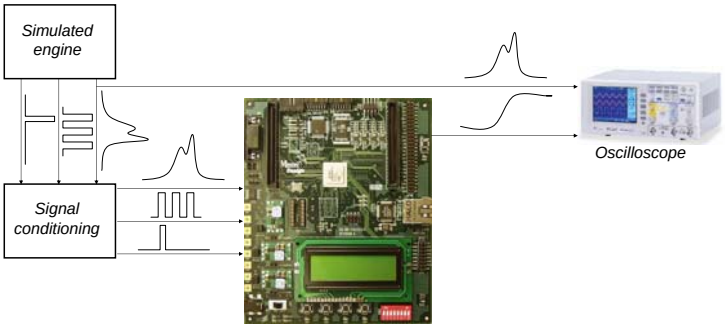

Figure 1: FPGA experimental system overview.

ALGORITHM A net HR (Q_{HR}^{net}) calculation was implemented in the FPGA, that is the HR calculation disregards heat transfer losses and crevice losses. Heat transfer losses are caused by convective energy loss to the combustion chamber walls. Crevice losses are caused by trapping fuel-air mixture in the crevices between the piston and the cylinder wall, thus avoiding combustion. The reason for neglecting crevice and heat-transfer losses in this work was to somewhat simplify the implementation. Since Bengtsson et al. [3] finds that Q_{HR}^{net} is sufficient for feedback purposes this simplification is regarded as legitimate.

$$\frac{dQ}{d\theta} = \frac{\gamma}{\gamma - 1} p \frac{dV}{d\theta} + \frac{1}{\gamma - 1} V \frac{dp}{d\theta} \tag{1}$$

The calculation of Q_{HR}^{net} is carried out in a non-conventional manner. The conventional way to calculate Q_{HR}^{net} is through the integration of Equation 1 as described by Gatowski et al. [6]. For signal processing applications it is however inconvenient to include the pressure derivative in the calculation since the process of differentiating a measured signal infers severe noise issues, especially in combination with a very high "over-sampling" rate. Instead of using Equation 1 it is possible to calculate Q_{HR}^{net} through Equation 6. Equation 6 origins from the conservation of energy and is motivated through Equation 2-5. This is a promising optional method to calculate the HR which Tunestål showed in [7].

$$\left.\begin{array}{l} dU = dQ - dW \\ dU = nC_v dT \\ dW = pdV \end{array}\right\} \Longrightarrow nC_v dT = dQ - dW \Longrightarrow dQ = nC_v dT + pdV \tag{2}$$

$$pV = nRT \Longrightarrow \frac{1}{nR} d(pV) = dT \tag{3}$$

$$C_v = \frac{R}{\gamma - 1} \Longrightarrow nC_v dT = \frac{nR}{\gamma - 1} dT \overset{(3)}{\Longrightarrow} nC_v dT = \frac{1}{\gamma - 1} d(pV) \tag{4}$$

$$dQ = \frac{1}{\gamma - 1} d(pV) + pdV \tag{5}$$

$$Q = \frac{1}{\gamma - 1}\int_{\theta_{start}}^{\theta} d(pV) + \int_{\theta_{start}}^{\theta} pdV =$$

$$= \frac{1}{\gamma - 1}(p(\theta)V(\theta) - p(\theta_{start})V(\theta_{start})) + \int_{\theta_{start}}^{\theta} p(\theta)\frac{dV}{d\theta}d\theta =$$

$$= \underbrace{\frac{1}{\gamma - 1}p(\theta)V(\theta) + \int_{\theta_{start}}^{\theta} p(\theta)\frac{dV}{d\theta}d\theta}_{\text{Calculated on-line}} - \underbrace{\frac{1}{\gamma - 1}(p(\theta_{start})V(\theta_{start}))}_{\text{Estimated constant, added off-line}} \quad (6)$$

IMPLEMENTATION System implementation in an FPGA environment differs from implementation in a "normal" processor system. The major difference between FPGA and processor implementation is the possibility to implement parallel branches of the current computation in the FPGA. It is possible to make two things happen in the exact same time, this does of course have the benefit that very high throughput in combination with low latency can be obtained, the drawback is that it is important for the designer to make sure that any intermediate results are synchronous with each other. A typical FPGA environment does also hold some other peculiarities. If floating point numbers for example are to be used within the FPGA environment, logics supporting floating point arithmetics has to be added manually. This is however not unique, it is the case in many processor systems as well. Fixed point arithmetics was hence used throughout the implementation of the HR calculation.

When Equation 6 was implemented the parts of the equation that were known in advance were not calculated on-line. Instead they were mapped as a function of current engine CAD in the distributed RAM of the FPGA, this goes for V and dV. As basis for the volume maps the geometry listed in Table 1 was used (since this engine is the basis for the pressure traces used in the test environment).

The time resolution is also of high algorithmic interest and it has to be noted. The described setup had the properties of an asynchronous system since the engine delivers CADP:s at one "clock speed" (which varies with the engine speed) and the FPGA board ran at a totally different one, meaning that the FPGA clock was non-synchronous with the CADP. This is an unconventional approach in an engine control system. The described calculation of Q_{HR}^{net} does, as understood from Equation 6, demand some synchronization between the calculation of Q_{HR}^{net} and the engine position (in order to retrieve the correct V and dV). This synchronization was provided through a CADP counter, a simple incremental counter which was reset on the rising edge of TDCP. Overrun detection was also provided on this CADP counter in order to detect issues with the CADP and TDCP. The output of the CADP counter indicated the current position of the engine and it was used as index for the tabulated values of V and dV which were kept in the RAM memory. In this manner the FPGA system had all the information needed for the calculation of Q_{HR}^{net} without synchronizing FPGA clock pulses with the CADP.

The clock-speed of the AD converter were, as previously noted $50\ MHz$, this was hence the sampling rate of P_{cyl}. This is by far a higher sampling rate than the update speed of V and dV, meaning that a number of Q_{HR}^{net} samples were calculated "in vain". The outcome was a system that calculates Q_{HR}^{net} based on the same P, V and dV values several times, a system with

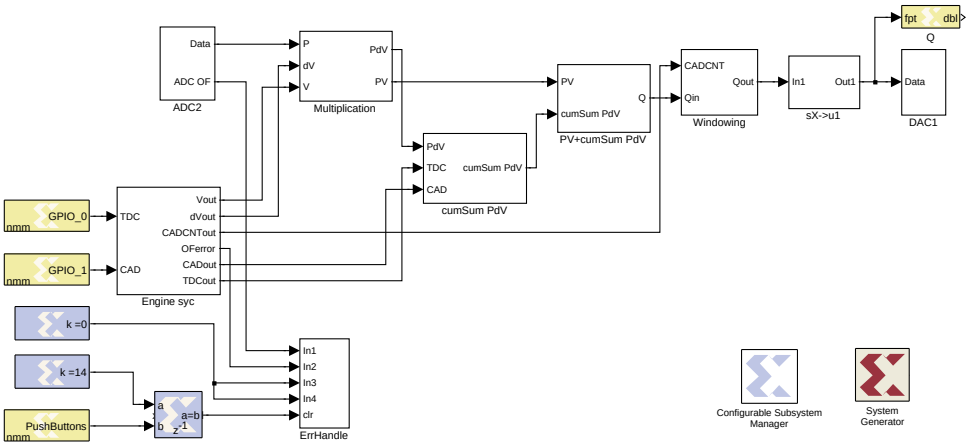

Figure 2: FPGA experimental system overview.

Number of Cylinders	6
Swept Volume	11705 cm^3
Compression Ration	18 : 1
Bore	127 mm
Stroke	154 mm
Connection Rod	255 mm

Table 1: Geometric properties of the Scania engine.

$P_{lsb} = 1464.84\ Pa$	$V_{lsb} = 5.05 * 10^{-7}\ m^3$
$dV_{lsb} = 1.74 * 10^{-9}\ \frac{m^3}{CADP}$	$Q_{lsb} = 3.85\ J$

Table 2: the resolution of the input and output variables.

very high rate of over-sampling. The parallel nature of the FPGA however still made this the preferred way to perform the design. The FPGA simply outputs Q_{HR}^{net} samples at $50\ MHz$ no matter what, meaning that the inferred latency, counted in number of FPGA cycles was constant regardless of the engine speed. The commonly used method to synchronize the engine and the control electronics by clocking the control electronics from engine driven pulses (ie CADP) were not used here since it would totally destroy the largest benefit of using the FPGA, i.e. low computational latency. It was not at all necessary to synchronize the clock of the FPGA with the CADP since it was possible to maintain sync with the engine with help of the CADP counter. Disregarding the synchronous thinking between the engine and FPGA board enables very low latencies without any major drawback.

EXPERIMENTAL RESULTS The major result of this investigation was of course the successful implementation of the described HR algorithm in the FPGA environment as described. The performance of the implementation is a point that can not be stressed enough, a P_{cyl} sample that arrives to the FPGA from the AD converter is, considering the timescale of an engine, calculated immediately. Immediately in this case means 12 FPGA clock cycles, $clk_{fpga} = 100\ MHz \rightarrow 120\ ns$! This does in other words mean that when the AD converter has delivered a sample on the FPGA pins it takes $120\ ns$ before the FPGA has delivered the corresponding

Q_{HR}^{net} sample on the pins of the DA converter! In $120\ ns$ an engine revolving at $1200\ rpm$ moves $0.000864\ CAD$, if the engine were revolving at $24000\ rpm$ it would move $0.01728\ CAD$! The latency between an arriving P_{cyl} sample and the output of the corresponding Q_{HR}^{net} sample is in other words negligible. Since it is possible to measure the Q_{HR}^{net} with as high frequency as P_{cyl} and concurrently with P_{cyl}, it is motivated to call the system a "virtual Q sensor" meaning that Q_{HR}^{net} is calculated the moment P_{cyl} is measurable. It is difficult to measure the latency through the FPGA with standard measuring equipment, the calculation of the latency is based on the fact that it is possible to extract precise latency information from the Simulink layout. To give an indication to the numbers mentioned Figure 3 is included. Figure 3 shows P_{cyl} output from the engine simulator together with the output of the FPGA (or "virtual Q sensor"), the fist cycle is a motored cycle and the following cycle are a fired one. P_{cyl} and Q_{HR}^{net} are synchronously sampled, the sampling is not halted and no data is cut away between the two corresponding cycles. It is easily understood from Figure 3 that Q_{HR}^{net} is calculated with very low latency with respect to P_{cyl}.

Besides very low latency the system does feature a very high throughput. If the engine would be able to produce pressure in a rate high enough it would be possible to perform 12 "complete" (meaning 120*5 point) heat release analyses each CAD at $1200\ rpm$! The limit of the throughput is the AD conversion speed.

The last point of the results are the correctness of the output. To verify the correctness 100 cycles are sampled to the PC. Due to poor quality of the pulses originating from the simulated engine some cycles however had to be removed due to sudden steps in the middle of the calculation. The number of disregarded cycles are in the range of 20-30 cycles depending on the "mood" of the simulated engine. Figure 4 shows all the sampled cycles that are non damaged, as understood from the figure most of the calculated cycles holds a high enough signal quality to use in a feedback control loop. Figure 5 shows the average of the cycles shown in Figure 4, it also shows the "corrected" values calculated off-line by the PC and Matlab. As understood from the figure the FPGA implemented algorithm is able to calculate Q_{HR}^{net} accurately enough. If the FPGA output and off-line Matlab calculated Q_{HR}^{net} are compared to a Q_{HR}^{net} calculated from Equation 1 consistency is again found. Q_{HR}^{net} according to Equation 1, is however not shown in order to not to blur the figure.

DISCUSSION The results in Figure 3 - Figure 5 show that implementation of Equation 6 was successful. As previously noted some cycles are removed from the results before presentation. The intention of the final system is of course that every cycle will be calculated perfectly, this is also achievable, better edge detection logics in combination with better quality of the positioning pulses will do the trick. The reason for the erroneous cycles are mainly thought to be issues with the simulated engine. As it was difficult to find a suitable signal simulator, the simulated engine had to be developed in house, and even though it performs well on average some cycles are not true to the real engine and it is these cycles that were removed. The cycles were removed based on a derivative threshold of the measured Q_{HR}^{net}. As visible in Figure 4 some "bad" cycles do however still persist (bad cycles meaning cycles that have a sudden "jump" in the middle of the signal).

Besides the issue with "bad cycles" there appears to be an offset problem, in Figure 4 there

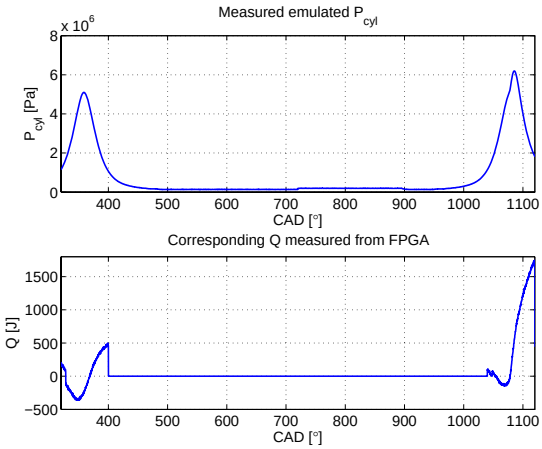

Figure 3: the output from the FPGA system when the “combustion” suddenly is switched on, showing true “in-cycle” performance.

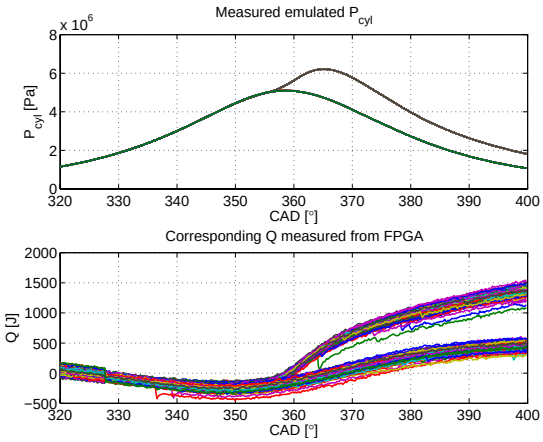

Figure 4: non average results corresponding to figure 5.

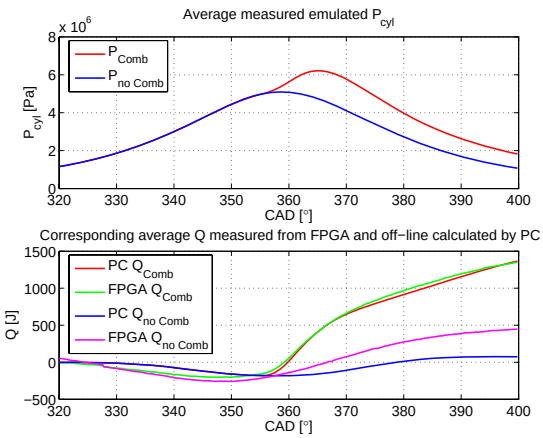

Figure 5: average output from the FPGA system compared to the corresponding values corrected in the PC.

appears to be different bias in the different cycles of Q_{HR}^{net}. This offset problem is explained by the nature of the AD converter connected to the FPGA. The AD converter was purchased as an expansion card suitable for the expansion header on the FPGA board, the case was however that the input circuitry on the AD converter card was intended for usage in very high frequency systems and the signal paths to the AD converter hence blocked the very low frequency cylinder pressure signal. The input circuitry hence had to be replaced by an "in house" alternative which unfortunately suffered form a slight problem with the input bias which explains the "cycle to cycle" variation of Q_{HR}^{net} in Figure 4.

The two noted issues are unfortunate and an effort will be made to correct them for future versions of the system. When the errouros data was removed as described and the remaining cycles were averaged Figure 5 emerged. Figure 5 does show that the FPGA system despite the noted issues in average performs well. The difference between Q_{HR}^{net} calculated by the PC off-line and Q_{HR}^{net} calculated by the FPGA is not large, at least not in the case with "combustion". Q_{HR}^{net} in the "motored" case is less consistent between the PC calculations and the FPGA, the explanation for this is thought to be the bias issues on the input of the AD converter. This bias issue will strike differently depending on the signal level. Since the signal level is lower in the "motored" case the error will also be larger.

The difference between the corrected Q_{HR}^{net} and the FPGA Q_{HR}^{net} is mainly thought to be explained by the above noted problem. The fact that fixed point number had to be used in the implementation is not thought to account for any large error in the signal as motivated by Table 2. In the table it is clearly visible that despite the limited input word-length of only 12 bit the resolution (known by the value of LSB) is, by far, enough (P_{lsb} is calculated on a $60\ bar$ P_{max} assumption if $P_{max} = 200\ bar$ are used $P_{lsb} \approx 0.05\ bar$).

Even though the resolution is regarded as accurate enough the FPGA environment still did cause some problems during the implementation phase. System generator DSP is a tool that should be used with care; both the FPGA internal number representation and the internal timing of the FPGA are very difficult to manage. The transparency of the tool is simply not good enough to enable the designer to design the FPGA layout in a controlled manner. Many issues had to be solved *"ad-hoc"* causing delays in the design work, FPGA design is very sensitive to the available tools regardless of its application.

CONCLUSIONS

- The paper has shown the possibility to implement an internal combustion analysis related algorithm in an ASIC/FPGA environment.
- Disregarding the synchronous thinking between the engine and FPGA board enables very low latencies without any major drawback.
- Extraordinary performance can be achieved using the described system:
  - A sample of cylinder pressure P_{cyl} is processed and corresponding Q_{HR}^{net} is calculated before any engine can move 0.02 CAD.
  - A throughput of $50\ MHz$ enables the FPGA system to perform 12 120*5 sample HR analyses within a single CAD @ $1200\ rpm$.

  - Further performance improvement can easily be obtained through the use of more powerful hardware.

- It is through the use of Equation 6 possible to calculate the cumulative heat release avoiding the use of any (often noisy) pressure derivative.
- The developed system can be thought of as a “virtual Q sensor”.

REFERENCES

[1] R. Johansson, A. Rantzer (Eds.), “Nonlinear and Hybrid Systems in Automotive Control”, Springer-Verlag London Berlin Heidelberg, ISBN 1-85233-652-8, 2003.

[2] J-O. Olsson, P. Tunestål, B. Johansson, “Closed-Loop Control of an HCCI Engine”, SAE 2001-01-1896, 2001.

[3] J. Bengtsson, R. Johansson, P. Strandh, B. Johansson and P. Tunestål, “Closed-Loop Combustion Control of Homogeneous Charge Compression Ignition (HCCI) Engine Dynamics” International journal of adaptive control and signal processing, 2004, Vol. 18, No. 2, pp 167-179, Wiley 2002; 00:1-6.

[4] G.M. Shaver, J.C. Gerdes, M. Roelle, “Physics-Based Closed-Loop Control of Phasing, Peak Pressure and Work Output in HCCI Engines Utilizing Variable Valve Actuation”, Proceeding of the 2004 American Control Conference, June-July, 2004, Vol. 1, pp 150-155, IEEE 0-7803-8335-4/0.

[5] J. Bengtsson, P. Strandh, R. Johansson, P. Tunestål, B. Johansson, “Hybrid Control of Homogeneous Charge Compression Ignition (HCCI) Engine Dynamics”, International Journal of Control, Vol. 79, No. 5, 2006, pp 422-448, Taylor&Francis.

[6] J.A. Gatowski, E.N. Balles, K.M. Chun, F.E. Nelson, J.A. Ekchian, J.B. Heywood, “Heat Release Analysis of Engine Pressure Data”, SAE841359, 1984.

[7] P. Tunestål, “Estimation of the In-Cylinder Air/fuel Ration of an Internal Combustion Engine by the Use of Pressure Sensors”, Doctoral Thesis , Department of Energy Sciences, Lund University, ISSN 0282-1990, ISRN LUTMDN/TMVK–1025–SE

[8] T.J. Todman, G.A. Constantinides, S.J.E. Wilton, O. Mencer, W. Luk, P.Y.K. Cheung, “Reconfigurable computing: architectures and design methods”, IEEE Proceedings, Computers and Digital Techniques, Vol. 152, No. 2, pp 193-207, IEEE, ISSN: 13502387

[9] R. Tessie, W. Burleson, “Reconfigurable Computing for Digital Signal Processing: A Survey”, Journal of VLSI Signal Processing, 2001, Vol. 28, No. 727, Kluwer Academic Publishers.

[10] Z. Wei, H.K. Byung, A.C. Larson,R.M. Voyles, “FPGA Implementation of Closed-Loop Control System for Small-Scale Robot”, Advanced Robotics Proceedings, 12th International Conference on Advanced Robotics, 2005, pp 70-77, IEEE 0-7803-9177-2/05.

[11] M. He, K-V. Ling, “Model Predictive Control On A Chip”, International Conference on Control and Automation, 2005, vol 1, pp 528-532, IEEE 0-7803-9137-3/05.