



LUND UNIVERSITY

An Ultra High Bandwidth Automotive Rapid Prototype System

Wilhelmsson, Carl; Tunestål, Per; Johansson, Bengt

2007

[Link to publication](#)

Citation for published version (APA):

Wilhelmsson, C., Tunestål, P., & Johansson, B. (2007). *An Ultra High Bandwidth Automotive Rapid Prototype System*. Paper presented at Fifth IFAC Symposium on Advances in Automotive Control.

Total number of authors:

3

General rights

Unless other specific re-use rights are stated the following general rights apply:

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

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

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.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

AN ULTRA HIGH BANDWIDTH AUTOMOTIVE RAPID PROTOTYPE SYSTEM

Carl Wilhelmsson * Per Tunestål *
Bengt Johansson *

* *Division of Combustion Engines, Department of Energy
Sciences, Faculty of Engineering, Lund University*

Abstract: For developers of automotive control, prototyping and initial tests are a hassle. Commercial solutions are available but the price and especially the price/performance ratio opens the field for more cost effective solutions. Automotive rapid prototype systems seen so far are mainly processor based systems with standard interrupt driven measurement and actuation. Control systems based on high time resolution measurements of for example cylinder pressure are difficult to implement using these systems, neither is it possible to implement controller loops with an extremely high bandwidth in combination with expensive algorithms. Measurement and actuation within the same engine cycle, In Cycle Control (ICC) are not possible. The proposed system is based on a mixed system consisting of one standard x86 processor which is configured through Simulink and a reconfigurable application specific integrated circuit (an FPGA) configured either by relevant FPGA design tools or by Simulink. This layout of the rapid prototype system enables the designer to implement either ICC with very high bandwidth (only limited by the capacity of the injection system) or between-cycle control with medium bandwidth. The aim of this paper is to describe one possible configuration of such a system and to discuss the possible performance outcome of the final system.

Keywords: Engine Control, FPGA, Simulink, Rapid Prototype, xPC target.

1. INTRODUCTION

Demands on internal combustion engines increase year by year, both regarding emissions and efficiency. As demands on the internal combustion engines increase, feedback control of the actual combustion is emerging as an important tool in order to implement new combustion concepts as shown by for example (Olsson *et al.* 2001). According to among others (M. Oki *et al.* 2006) the bandwidth of the actuators, meaning fuel injectors (such as piezoelectric fuel injectors) is currently increasing. In order to utilize the performance gains of the fuel injectors, new non-conventional engine control systems have to be developed. Con-

ventional engine control systems typically have a performance grade which enables them to control the engine from one cycle to another, until now this has not been an issue since normal actuators so far have lacked capability beyond cycle to cycle control. The new high bandwidth fuel injectors in combination with an high bandwidth control system would give interesting feedback combustion control possibilities. In-cycle combustion control, ICC, might be possible!

The Field Programmable Gate Array (FPGA) technology is another area of research which has matured rapidly in recent years and the area of ap-

plication is very large. Possible application areas are added day by day and there is a large number of different design architectures in the FPGA design field, a number of these are accounted for by (Todman *et al.* 2005). FPGAs could give very large benefits and enable the high bandwidth needed for a control system attempting in-cycle feedback control of Internal Combustion (IC) with the help of for example high bandwidth piezoelectric injectors. The possibility of implementing pipelined and parallel calculations in combination with very high throughput frequency, low jitter and low latency are features which are typical for FPGA devices. Very interesting work has been undertaken utilizing FPGA devices to implement as “heavy” control algorithms as Model Predictive Control (MPC) by, among others (Ling *et al.* 2006).

Besides the possibility of in-cycle feedback combustion control loops residing purely within an FPGA the possibility exists to let an FPGA assist the processor normally present in the engine controller with calculations for the cycle-to-cycle control loop. Such an FPGA based “co-processor” or reconfigurable computer may prove useful from two different perspectives;

- Algorithms normally too complex for the, normally cheap and low spec, engine controller can be successfully implemented.
- The spec, and hence price of the engine control unit can be reduced by replacing software implementation with more efficient FPGA implementation of algorithms.

These two points would mainly be interesting from a production perspective and no further attention is given to them in this paper.

Simulink based development environments for rapid prototyping of automotive control have been previously attempted and commercial products exist on the market from among others the company dSpace. Besides dSpace, IFP in France has a solution enabling rapid prototyping and regression tests both on the software and system level. Other rapid prototype systems are for example presented by (Viele *et al.* 2005) showing a solution based upon National Instruments products, the CompactRIO system, in combination with an FPGA module. In this case LabView is chosen as programming/FPGA configuration environment. The approach taken by (Viele *et al.* 2005) is a more conventional one and even though the theoretical possibility of in cycle control would exist, it is not utilized. The system is used for implementation of a “standard” control unit on a motorcycle, it is also reasonable to believe that LabView would not be the first choice for complex and high performance FPGA implementations. (Beaumont *et al.*

2006) shows one approach where an “off the shelf” processor based rapid prototype system from the company Ricardo is fitted with an FPGA based IO system. The intention of (Beaumont *et al.* 2006) seems to be to use the FPGA as a co-processor to assist the processor based rapid prototype systems with data acquisition, analog to digital conversion and data pre-processing. The architecture is less suited for feedback control loops implemented purely in the FPGA and ICC since the FPGA does not reach any of the actuator outputs directly, they are connected to slower subsystems.

2. SYSTEM

The over-all idea with the system described here is to combine very flexible parts, featuring high flexibility in implementation, but with a more limited capacity with less flexible but very high performance parts. The parts regarded as flexible would be an x86 PC running the Matlab/Simulink xPC-Target environment, being flexible in the sense that it is very easy to re-implement and change controllers and supporting software/algorithms residing in the xPC-Target environment. The part of the system regarded as less flexible would be reconfigurable hardware (FPGA) which has to be configured using special tools featuring more limited or no rapid prototyping capabilities. The resulting system will be referred to as Engine Dyno Controller (EDC), the reason being that it is a control system suitable for controlling the complete dyno setup, not only the engine.

2.1 The Setup

Figure 3 gives an overview of the engine dynamometer setup. The figure describes a “typical” engine control design setup and the devices present in the figure would be present in most setups in one form or another. The center of the setup is of course the actual engine, which also is the actual plant for the control engineer. The engine generates energy during operation, this energy needs to be taken care of, a device capable of maintaining the rotation speed is hence necessary, an engine dynamometer. In this case the dynamometer consists of an electric motor in combination with control electronics. The EDC system would need the capability to demand an engine speed setpoint from the dynamometer in order to carry out transient identification and validation experiments. Preferably control data for the dynamometer would be transferred through MODBUS to the control electronics of the electric motor, RS-485 will be used as the physical layer for the MODBUS communication.

In each cylinder of the engine a piezoelectric cylinder pressure transducer will be present. The pressure transducers would be connected to a charge amplifier which converts the electronic charge originating from the transducer to a voltage. Besides analog outputs for pressure, the charge amplifier has the possibility of control related communication. In reality however the control parameters of the charge amplifier is very seldom changed, and this communication path will hence not be implemented. Each analogue output of the charge amplifier are connected to two different analogue I/O inputs of the EDC.

A vast variety of other devices might also be connected to the engine setup to perform different measurements, for example thermocouples, air/fuel ratio sensor(s), emission measurement devices, non-cylinder pressure transducers and the crank angle counter (angular positions sensor). These devices are typically sampled by the EDC (apart from the crank angle counter) either by an analogue I/O device (preferable) or in some cases by digital communication (GPIB, RS-XXX, or ethernet). The crank angle counter communicates through parallel digital channels physically carried on fiber optics. Based on the inputs of the sensors the control engineers wants to perform two task:

- Implement and/or validate automatic control logic
- Collect sensing data for post analysis

In order to implement control logic actuators are, of course, needed. In this case there are mainly two types of actuators with significantly different bandwidth. The high bandwidth kind are typically piezoelectric fuel injectors, spark coils and inductive fuel injectors. Low bandwidth actuators would typically be stepper motors, controlling for example throttle positions and turbo settings (for a non fixed turbo).

In the end it is of course the engine control logic which is of main interest however, in order to have an efficient experimental setup, the peripheral devices needs to be controlled by the same system as the engine control logic reside in. If they are, it enables the control engineer to perform identification and transient experiments on for example the engine speed (if the controller can communicate with the dynamometer). It also enables the control engineer to perform experiments on associated control problems, for example boost pressure control or other air-path control. Besides capability to communicate with “need to use” devices always present in the system it is of utmost importance that it is quick and easy to plug in new devices, sensors or actuators which may be needed in future control experiments.

2.2 Loop Overview

The main idea with the described system is to have the possibility of two control loops with significantly different bandwidths, Figure 1 describes this concept. Color marking distinguishes the two different loops in the figure, the loop marked in black would be the high bandwidth loop carrying out ICC control tasks and assisting the low bandwidth loop (marked in grey) with the cycle-to-cycle control. The high bandwidth loop, indicated in Figure 2, is found to consist of; the engine, cylinder pressure sensor (CPS), charge amplifier (CAMP), high frequency AD converter (HFADC), the FPGA board and last but not least the injection system (communication protocol between the components vary). The low bandwidth loop, noted in grey in Figure 1 and still with Figure 2 in mind, would consist of; the engine, the cylinder pressure sensors (CPS), the charge amplifiers (CAMP), the framed ADC (FADC), the x86-PC and the FPGA. Note that the FPGA board is the master controller of the injection system and the output of the low bandwidth combustion control loop hence has to pass through it. Pressure sampling in the high bandwidth loop is asynchronous to the engine revolution which means that some synchronisation algorithm is needed in order to implement the injection control and other algorithms based on the engine crank angle. The low bandwidth loop on the other hand is clocked synchronously with the engine revolutions and synchronisation is hence built-in.

The high bandwidth loop is complemented by the x86-PC running xPC-Target environment. Implementation of controllers in the high bandwidth loop is time consuming but controller implementation in xPC target is not. The existence of the PC hence enables the designer to, in a very rapid manner, implement controllers with the support of a large amount of plug in modules for I/O. The xPC-Target environment will be used for implementation of user interface, implementation of the low-bandwidth control of the combustion (cycle-to-cycle controllers) and the low bandwidth control of peripheral devices, such as air path control. Not indicated in Figure 1 are the controller loops handling the peripheral control, they would sample data and output controller demand. Both the normal ADC and the FADC could act as input, normally however the FADC would act as an input since peripheral control tasks have lower bandwidth requirement. It would typically be implemented in the Simulink environment. Protocols like RS-XXX, GPIB and ethernet would be used to output controller demand.

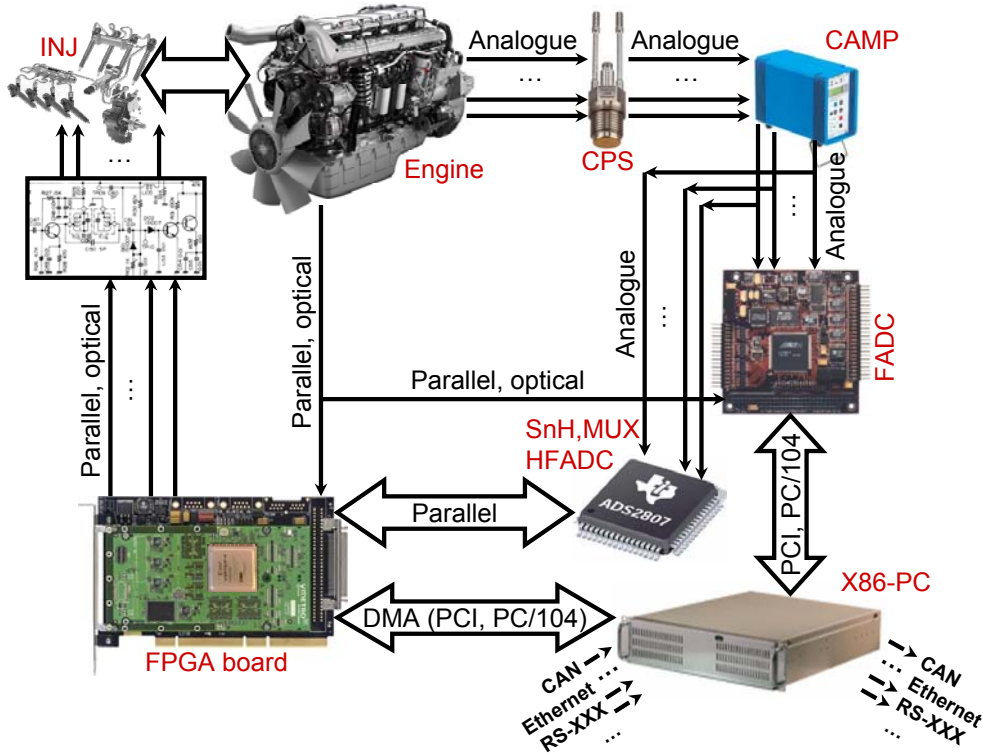


Fig. 2. Overview of the components within the suggested control system.

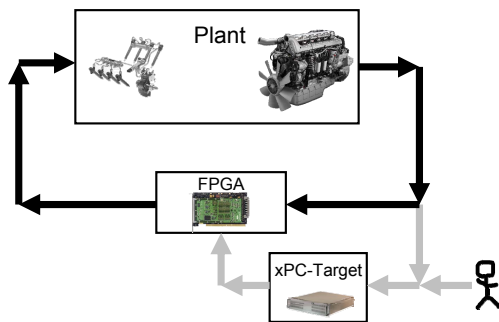


Fig. 1. overview of the two different loops.

2.3 Loop Bandwidth

The bandwidth of the two loops in absolute terms would of course be determined by the slowest component included in each loop. To determine the bandwidth of the two different loops the bandwidth of each component hence must be found, starting with the high bandwidth loop;

The clock frequency of the FPGA would be or exceed 100MHz depending on device selection, FPGA clock would hence not be an issue. Clock frequency of the HFADC would, again depending on device, range between a few kHz up to 100MHz . As input the HFADC takes the output

signal from the CAMP which in turn takes the output from the CPS. CAMP cut-off frequency would be in the range of 200kHz and CPS natural frequency in the range of $40 - 200\text{kHz}$ all according to manufacturer specifications. Last link in the HF loop would be the fuel injectors, work published by (M. Oki *et al.* 2006) indicates the available bandwidth of diesel fuel injectors. It is found that the time it takes the injector to lift the nozzle needle to half, from the instance the command signal is given, is $200\mu\text{s}$ the -3dB frequency is hence $\approx 5\text{kHz}$.

Appropriate loop bandwidth can be decided in many different ways, if the Nyquist criterion is used the sampling frequency of the HFADC would be decided to $2 * 200\text{kHz} = 400\text{kHz}$ with the cut-off frequency of the CPS in mind, or $2 * 5\text{kHz} = 10\text{kHz}$ based on injector bandwidth. However, since this system is intended for control purposes another sampling criterion should be used ensuring control system performance. (*Computer-Controlled Systems* 1997) suggests a sampling frequency between 10 to 30 times the loop bandwidth. Loop bandwidth is limited to the injector bandwidth of 5kHz and an appropriate minimum sampling/control frequency would hence be $50 - 150\text{kHz}$, if all six cylinders together are sampled by the same ADC it has to be able to sample at $6 * 150\text{kHz} = 900\text{kHz}$. For the high speed loop the Nyquist frequency will consequently be

$150/2 = 75kHz$. A fairly high loop bandwidth in combination with the fact that the FPGA clock frequency is, compared to the sampling frequency, significantly higher enables large amounts of calculations to be carried out between each CPS sample. For example in order to implement complex controllers (MPC etc) with such high bandwidth requirement, in multiple cylinders simultaneously the very high throughput, parallelism and high clock speed of the FPGA is regarded as essential and the use of the FPGA system is hence motivated. Preferably the HFADC is operated synchronously with the FPGA clock. The FPGA should drive the HFADCs and in this way traditional real-time and synchronisation issues are avoided. The performance of the FPGA board is the enabling factor, letting the systems perform very complex calculations between each sample, without violating the bandwidth of the loop, this loop is the one enabling in cycle combustion control.

The low bandwidth xPC target based loop shares some components with the high bandwidth one, the CPS, CAMP, the FPGA and the injection system. In the case of the low bandwidth loop the FADC would be clocked by the crank shaft pulses, a method which can be regarded as the “standard” method. The sampling frequency will hence depend on current engine speed, making anti-aliasing filtering more difficult. For simplicity an anti-aliasing filter with cut-off frequency decided by maximum engine speed would be used. Maximum clock speed of the FADC are decided by the crank angle sensor resolution and the maximum physical engine speed. Results are published using different angular resolution. One pulse each Crank Angle Degree (CAD) is frequently used, in this case a resolution of five pulses each CAD as used by among others (Olsson *et al.* 2001) and (Wilhelmsson *et al.* 2006) is preferred. Maximum engine speed is selected to be $6000rpm$ and with five crank pulses each physical CAD the crank pulses will arrive at a frequency of $180kHz$ according to Equation 2, at maximum engine speed and frequencies up to $180/2 = 90kHz$ are resolvable according to the Nyquist criterion. Consequently the crank pulses will arrive at a frequency of $24kHz$ at the minimum engine speed of $800rpm$ (according to Equation 1) still using five crank pulses each CAD, Nyquist frequency for minimum engine speed is hence $12kHz$. The desire is to be able to handle at least a 6 cylinder engine at maximum engine speed of $6000rpm$ preferably with a time resolution of five samples each CAD (there are of course 360 CADs on one engine revolution).

$$\frac{800 [n/min]}{60 [s/min]} * 360 [CAD/n] * 5 [cad^{-1}] = (1)$$

$$= 24\ 000 [Hz]$$

$$\frac{6000 [n/min]}{60 [s/min]} * 360 [CAD/n] * 5 [cad^{-1}] = (2)$$

$$= 180\ 000 [Hz]$$

2.4 Suitable EDC Hardware

The suggested electronic hardware setup are visible in Figure 2, the system are intended to be built around an PC (lower right in the figure). There are a number of different communication interfaces which could be used between the x86 computer and peripheral components, for example PCI, PC/104 or ethernet. Which interface that is best suited for the application is mainly a question which IO devices that holds support for Simulink/xPC-target. The support for framed sampling in xPC-Target is somewhat limited and one has to choose the I/O device with this in mind. As for the FADC there are mainly two suitable devices supported, either an analog I/O module from “Diamond Systems” named “Diamond-MM-32-AT” or one from “United Electronic Industries (UEI)” named “PD2-MFS-2M/14”. The Diamond Systems device is a 16-bit one, using the PC/104 bus, the device holds framed sampling support in xPC-target and is capable of a sample frequency of $200\ 000\ sps$. However with this sampling frequency it is possible to handle an engine speed of $5556rpm$ in a 6 cylinder engine (as desired) only if a resolution of 1 CAD is used (see Equation 3). The Nyquist frequency would then be $\approx 16700Hz$ at maximum engine speed (Equation 4).

$$\frac{\frac{200\ 000[s^{-1}]}{6}}{360[n^{-1}]} * 60[s/min] \approx (3)$$

$$\approx 5556 [n/min = rpm]$$

$$\frac{\frac{200\ 000[s^{-1}]}{6}}{2} \approx 16\ 667 [Hz] (4)$$

The other option, the UEI device, is a 14-bit device using the PCI bus and capable of $2\ Msps$. Using the UEI device it is possible to sample five times each CAD for a six cylinder engine up to $\approx 11\ 000rpm$ according to Equation 5 or handling the engine speed criterion ($6000rpm$) for up to a ten cylinder engine according to Equation 6. Which, in combination with the fact that the UEI device uses the more modern and faster PCI bus would make the UEI device the preferred one even though its resolution is 2 bit less than the Diamond Systems device. Framed sampling is an essential feature in order to implement

the system with cylinder pressure based cycle-to-cycle combustion control with the cycle-to-cycle controllers implemented in xPC target, running on the x86 processor on an interrupt driven basis. Due to the desired time resolution of cylinder pressure measurements the processor will not be able to maintain the controllers if the sampling would be taken care of in an interrupt driven sample to sample manner.

$$\frac{\frac{2M[s^{-1}]}{6[]}}{1800[n^{-1}]} * 60[s/min] \approx \quad (5)$$

$$\approx 11\ 111 [n/min = rpm]$$

$$\frac{\frac{2M[s^{-1}]}{10[]}}{1800[n^{-1}]} * 60[s/min] \approx \quad (6)$$

$$\approx 6667 [n/min = rpm]$$

Besides the I/O module connected to the PCI or PC/104 bus to work with xPC-Target one more ADC will be present in the system, the HFADC sampling values to the FPGA. That ADC will share the input of the FADC in an analogue manner and will have a high specification, a data resolution of $16bit$ and a maximum time resolution of up to $25MHz$ is possible. Minimum time resolution for this ADC is $150 * 6kHz = 900kHz$. Using a high spec ADC does however enable the system to expand, add or remove high speed signals. The high speed ADC will not run on a higher clock frequency than necessary. The selected ADC could preferably be LTC2203 from Linear Technology (other options from Linear in the sampling range of $10 - 105MSPS$ are LTC2207, LTC2206, LTC2205, LTC2204, LTC2203, or LTC2202). A MUX and sample-and-hold circuit might be needed as well. The outputs of the HFADC will be communicated in a parallel manner to the digital I/O of an FPGA development board.

The FPGA development card present in the lower left of Figure 2 will be the center of the system. Communications between the FPGA card and the x86 PC will be carried out using Direct Memory Access (DMA), which enables very high speed communication in a manner which is fairly simple to implement. Thus the FPGA board will have to be connected to one of the data buses of the PC. Devices suitable for this are for example the "Virtex-5 LX110 PCI Development Board" from Vmeto which can be connected to the PCI bus or the "TS-104-3001" from GE Fanuc Embedded Systems, which handles the PC/104 bus.

3. SOFTWARE/HARDWARE CONFIGURATION

The configuration of the rapid prototype system is of course a key issue, if the configuration process is too difficult and time consuming the prototyping may prove to be too slow. The configuration of the cycle-to-cycle controllers which are run on xPC-Target are of course carried out in Simulink with the relevant block sets. It should hence be possible for the control engineer to implement the cycle to cycle controllers in a rapid manner. Configuration of the target computer (downloading of firmware) as well as operator interface data will be carried out over ethernet.

Design of FPGA configurations however tend to be less rapid than the graphical programming of the cycle to cycle controllers in Simulink. There exists rapid prototype tools for FPGA environments as well, for example the Xilinx toolbox System generator DSP used in (Wilhelmsson *et al.* 2006). System generator, being a plug-in tool to Simulink, features the same graphical programming environment as Simulink. Altera and other manufacturers of EDA tools also have options similar to System generator DSP either to use in combination with Simulink or stand alone. Previous experiences made by the authors (Wilhelmsson *et al.* 2006) however point in the direction of using tools more specialised on FPGA configuration. The reason being that these tools, for example System generator DSP are not regarded to supply a high enough grade of automation and they appear not to be completely mature. In combination with a lack of transparency it is in some cases, according to previous experiences very difficult to successfully carry out larger FPGA designs using for example system generator DSP. The approach selected is instead to implement the main part of the intended design in a better suited and more complete design tool like the Xilinx ISE. Xilinx ISE is a toolbox containing the complete tool-chain for implementation in Xilinx FPGAs. Other FPGA manufacturers provides similar tools for their FPGAs. With this it is not said that every part of the design has to be carried out in pure VHDL, the possibility of carrying out graphically based designs/design parts still exist in ISE and the other similar tools. It is also possible to design modules in Simulink with the help of system generator DSP and "plug them in" to a skeleton system designed in Xilinx ISE or similar. Design of logics designated for the FPGA is never the less more time consuming than implementing logics designated for an xPC-target system in Simulink.

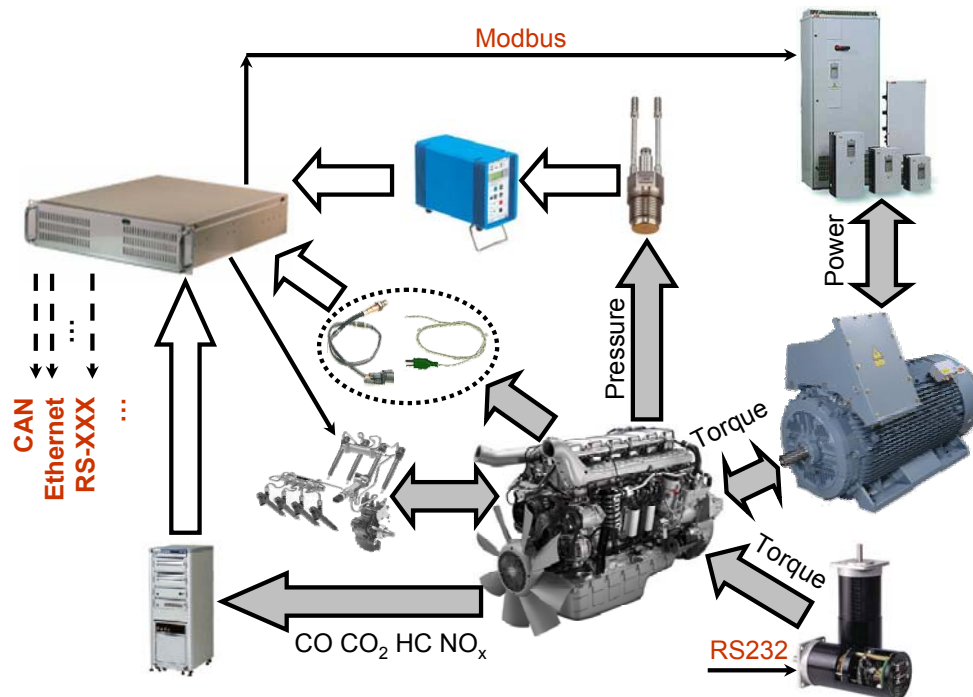


Fig. 3. Overview of a typical engine dynamometer setup, grey arrows decode physical relationship, white is sensor information flow and black is control signal flow.

4. EPILOGUE

A novel suggestion on the structure of an Engine Dyno Controller has been presented. The aim for the EDC is to have the normal capabilities regarding control of axial equipment in combination with the possibility of implementing advanced pressure based combustion control logic. The complete system will have the capability of performing rapid prototyping of cycle to cycle controllers with the help of Simulink and xPC-Target. The system will also have the capability of implementing control logics with a high enough bandwidth for feedback control of combustion using piezoelectric fuel injectors in their full bandwidth range. Thus enabling the control designer to carry out experiments with cylinder pressure based in-cycle control and Heat Release rate shaping. The main drawback with the high bandwidth logics is that the time of the implementation is longer compared to the cycle-to-cycle controllers implemented in Simulink. In-cycle control logics has to be implemented in tools suitable for FPGA design, such as the "Xilinx ISE". Implementation of the high bandwidth in cycle control loop is possible through the usage of reconfigurable hardware (FPGAs).

5. FUTURE WORK

The idea of future works is of course to implement the described system and to validate the performance through experiments. The described system will, after successful validation, provide a platform for future high and low loop bandwidth experiments.

REFERENCES

- Beaumont, A., J. Lemieux, P. Battiston and A. Brown (2006). Design of a rapid prototyping engine management system for development of combustion feedback control technology. number SAE 2006-01-0611.
- Computer-Controlled Systems* (1997). Prentice-Hall.
- Ling, K.V., S.P. Yue and J.M. Maciejowski (2006). A fpga implementation of model predictive control. ACC.
- M. Oki, S. Matsumoto, Y. Toyoshima, K. Ishisaka and N. Tsuzuki (2006). 180mpa piezo common rail system. number SAE 2006-01-0274.
- Olsson, J-O., P. Tunestål and B. Johansson (2001). Closed-loop control of an hcci engine. number SAE 2001-01-1031.
- Todman, T.J., G.A. Constantinides, S.J.E. Wilton, O. Mencer, W. Luk and P.Y.K. Che-

- ung (2005). Reconfigurable computing: architectures and design methods. *IEEE Proc., Comp.and Dig. Tech.* **152**(2), 193–207.
- Viele, M., L. Stein, M. Gillespie and G. Hoekstra (2005). Rapid prototyping an fpga-based engine controller for a 600cc super-sport motorcycle. number SAE 2005-01-0067.
- Wilhelmsson, C., P. Tunestål and B. Johansson (2006). Fpga based engine feedback control algorithms. number F2006P039. FISITA.