ISSN 0280-5316
ISRN LUTFD2/TFRT--5891--SE

Multi-Cylinder Valve Control FPGA-controlled Pneumatic Variable Valve Actuation

Niklas Everitt

Department of Automatic Control
Lund University
January 2012

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

Document name
MASTER THESIS

Date of issue
January 2012

Document Number
ISRN LUTFD2/TFRT--5891--SE

Author(s)
Niklas Everitt

Supervisor
Maria Henningsson Automatic Control, Lund Sweden. Kent Ekholm Combustion Engines, Lund Sweden
Rolf Johansson Automatic Control LTH, Lund Sweden (Examiner)

Sponsoring organization

Title and subtitle
Multi-Cylinder Valve Control – FPGA-controlled Pneumatic Variable Valve Actuation. (Flercylindring pneumatisk ventilaktuation)

Abstract
The society of today relies heavily on transportation. Goods and people are transported across the globe at an unpreceded scale and volume and the internal combustion engine is an integral part of the world's transportation systems. Combustion engines emit greenhouse gases that contribute to global warming, one of the most serious global threats today. One of several ways to increase efficiency and reduce exhaust emissions of ICEs is the use of variable valve actuation. The purpose of this thesis is to implement a control system for a variable valve actuation system from Cargine in a Volvo D12 cylinder head, a 12-liter heavyduty engine with six cylinders and 24 valves. The cylinder head is to be used in a laboratory engine at Lund University. The task consisted of two parts. Investigation of relationship between control pulse sent to the actuators, actuator pressure and valve displacement. The second part is to create a *Field Programmable Gate Array* (FPGA) based feedback control system in Labview to obtain desired valve open time, valve close time and lift. A second order model of the exhaust valve was identified. An FPGA based feedback control was implemented for all 24 valves in Labview. Satisfactory control of valve open timing and valve close timing and lift for a fixed engine condition was obtained. A Kalman filter was implemented on one valve, leading to a small increase in performance. Implementing a Kalman filtering in the controller on all valves would either require, optimization of the Kalman filter implementation, some kind of signal condition, or additional computational resources. The system defined and implemented in this thesis can serve as the basis of further experiments for modeling the effect of cylinder pressure and actuator pressure on valve open timing, valve close timing and lift.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
36

Recipient's notes

Security classification

http://www.control.lth.se/publications/

Acknowledgements

I would like to express my gratitude to everyone who has helped me along the course of this project.

I would like to thank my supervisors Maria Henningsson and Kent Ekholm for their continuous guidance and support throughout the project.

Urban Carlson and Anders Höglund, CEO and combustion engine expert at Cargine Engineering AB, for their time, effort and making this thesis possible.

Tommy Petersen for his help in designing and building the measurement system. Kjell Jonholm for the lift sensor calibration and help with the pnuematics and the other technicians for always being there to help when needed.

Hjalmar Arvidsson, Olivian Marku, Patrik Svensson for all the fun times in our office.

Everybody at the Division of Combustion Engines for their support and making my time at the departement a pleasant experience.

Contents

List of Figures

List of Tables

Nomenclature

AIC	Akaike's Information Criterion
ARMAX	Autoregressive moving average with exogenous input
CAD	Crank Angle Degree
DIO	Digital Input/Output
DMA	Direct Memory Access
EGR	Exhaust Gas Recirculation
EHV	Electro Hydraulic Valve actuation
EMV	ElectroMagnetic Valve actuation
EPV	Electro Pneumatic Valve actuation
FIFO	First In First Out
FPE	Akaike Final Prediction Error
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
HCCI	Homogeneous Charge Compression Ignition
ICE	Internal Combustion Engine
LUT	Lookup table
MIVEC	Mitshubishi Innovative Valve timing and lift Electronic Control
PC	Personal Computer
PI	Proportional Integral
RAM	Random Access Memory
RPM	Revolutions Per Minute
SISO	Single Input Single Output
VAF	Variance Accounted For
VC	Valve Closing
VO	Valve Opening
VVA	Variable Valve Actuation

1 Background

The society of today relies heavily on transportation. Goods and people are transported across the globe at an unpreceded scale and volume. The *Internal Combustion Engine* (ICE) is today an integral part of the world's transportation systems. Combustion engines emit greenhouse gases that contribute to global warming, one of the most serious global threats today. Although the extent and the urgency of taking action are debated, there is no doubt that we have to drastically reduce our carbon emissions. Rising fuel prices is another concern for engine manufacturers and their consumers. Growing environmental concerns coupled with increasing fuel prices are driving the development of cleaner and more efficient engines. One of several ways to increase efficiency and reduce exhaust emissions is the use of variable valve actuation.

1.1 Introduction

This thesis is part of a *Homogeneous Charge Compression Ignition* (HCCI) project that is a cooperation of Volvo Powertrain, Cargine, the department of Combustion Engines and the Department of Automatic Control at Lund University (Swedish Energy Agency, Diesel Combustion with Low Environmental Impact, Ref FFI-LV, project nbr 32067-1). The cylinder head of the target Volvo D12 engine, a 12-liter six cylinder heavy-duty engine with six cylinders and 24 valves, has been outfitted with pneumatic valve actuators from Cargine [1] that enables *Variable Valve Actuation* (VVA). VVA essentially gives full control of valve lift, lift duration and timing at all engine speeds and loads. Free valve control gives control of swirl and internal *Exhaust Gas Recirculation* (EGR). It can also be used for pneumatic hybridization, capturing energy usually lost from braking and storing it as pressurized air. This thesis aims at implementing a control strategy that minimizes cylinder-to-cylinder variations and cycle to cycle variations. Black-box modeling was used to develop a suitable valve model and a feedback controller was implemented.

1.2 Task

The aim of this thesis is to implement a control system of Cargine valves in a Volvo D12 cylinder head, a 12-liter heavy-duty engine with six cylinders and 24 valves. The cylinder head is to be used in a laboratory engine at Lund University. The task consisted of two parts. Investigation of relationship between control sequence u_d and valve displacement y_d. The second part is to create a feedback controller to obtain desired *Valve Opening* (VO) timing y_{VO}, *Valve Closing* (VC) timing y_{VC} and lift y_L through the control parameters: start of control pulse u_{VO}, end of control pulse u_{VC} and actuator pressure u_P.

1.3 Variable Valve Actuation

Valves are the standard option for facilitating gas-exchange in an ICE. Gas-exchange is the process in which burnt gases are removed from the combustion chamber and

exchanged with fresh air (direct injected ICE) or a mix of air and fuel (port injected ICE). Conventional valve actuation is done through camshafts with a fixed valve profile. However, the optimal valve profile is dependent on engine speed and load, whereby the valve profile is bound to be a compromise. VVA gives the opportunity to have specific lift profiles for every operating condition. Optimal VO and VC timings are dependent on load and speed conditions. For example, a spark ignition race engine operating at high engine speeds keeps the inlet valve and exhaust valve open longer to facilitate a quick gas exchange. However this setup will produce rough idling and poor performance at low engine speeds since unburned fuel will exit the open valves. Variable valve time thus gives greater efficiency and power across a wider range of engine speeds [2, p. 215].

Variable valve actuation is fundamental for concepts yet to be commercialized; pneumatic hybrids, cylinder deactivation and it's beneficial for HCCI. Pneumatic Hybrids are a hybridization that has the potential of regenerating a substantial amount of braking energy and reuse it for acceleration. A simulation study has shown the potential benefits of the pneumatic hybrid bus. A fuel consumption reduction of between 8 % and 58% was achieved, highly dependent on driving patterns [3]. Deactivating cylinders enables the engine to run more efficiently at low loads because the still active cylinders operates more efficiently at higher load. HCCI is an ICE in which well-mixed fuel and oxidizer are compressed to the point of auto-ignition and promises greatly increased fuel efficiency and lower emissions.

1.4 Camshaft-based VVA Strategies

Camshaft-based strategies have some extra components that enables the change of timing, lift or a combination of the two. Camshaft phasing system rotates the camshaft in relation to the crankshaft. The valve timing are changed but the duration and lift remain the same. A simple system offer the possibility to switch between a set of profiles while a more complex system can change the timings continuously.

Cam profile switching is a way to change the duration and lift by switching the cam profile. There are several different implementations of this, one being the *Mitshubishi Innovative Valve timing and lift Electronic Control* (MIVEC) [4]. The MIVEC system has one low speed cam, one high speed cam and two rocker arms. The system uses one rocker arm at a time or none, in other words the system offers valve deactivation. There are also systems combining cam phasing with cam profile switching. Porsche's VarioCam Plus is such a system which has three cam profiles that offers 4 different lift curves for the intake valves, all can be seen in Figure [5].

Combining both variable lift and timing can be done in cam-based systems in several ways. See for example [6] for a description of the BMW Double Vanos system that is the foundation for the BMW Valvetronic system that offer fully variable valve actuation [7].

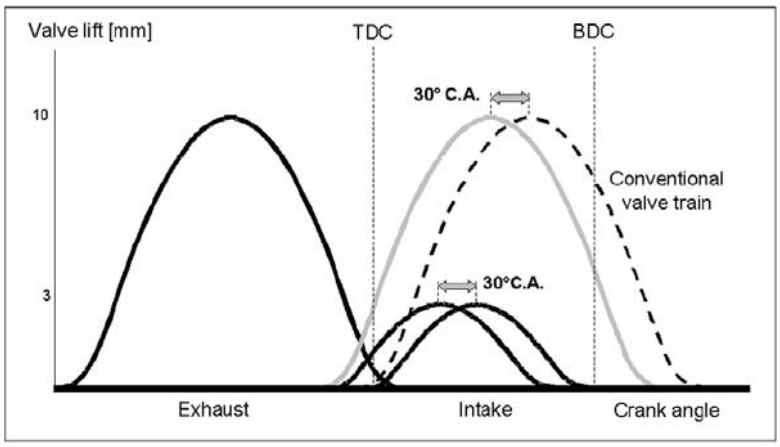


Figure 1: VarioCam Plus valve lift curves [5]

1.5 Camless-based VVA Strategies

Camless-based VVA strategies offer great adjustability of valve timing and lift. The drawback is that these systems are often complex and expensive and mainly used in laboratory settings. Several different approaches has been proposed including electromagnetic, electro hydraulic and electro pneumatic systems.

ElectroMagnetic Valve actuation (EMV) systems uses a series of solenoids, springs, a permanent magnet and an electromagnet to actuate the valve. Besides the flexibility, the benefit with the EMV systems compared to a conventional camshaft-based system is the steep VO and VC. One critical problem with an EMV system is the control of valve seating velocity, a high seating velocity leads to high noise levels. It has been shown that to mitigate this problem closed-loop feedback is needed [8]. A successful solution is to have feedback from a measurement of the location of the armature inside the actuator, see Figure 2.

Electro Hydraulic Valve actuation (EHV) systems use the elastic properties of a compressed hydraulic fluid. The fluid acts as a liquid spring, accelerating each valve during its opening motion and decelerating it during the closing motion. Ford has developed an EHV system [9], which has been illustrated in Figure 3a. The seating velocity is also a challenge with this system but can be solved by a practical solution, a hydraulic snubbing action illustrated in Figure 3b.

Electro Pneumatic Valve actuation (EPV) system is a promising alternative to the aforementioned EHV and EMV system. Both EHV and EMV systems perform well in research settings, but have issues making them less suitable for production engines. The EMV system suffers from high noise and packaging issues, while the EHV system have problems with temperature variations and is expensive. EPV system are comparatively cheap, have the characteristics of a fully variable valve actuation system and low seating velocity. Additionally the system is more robust since to air leaks are less hazardous than oil leaks in an hydraulic system [10].

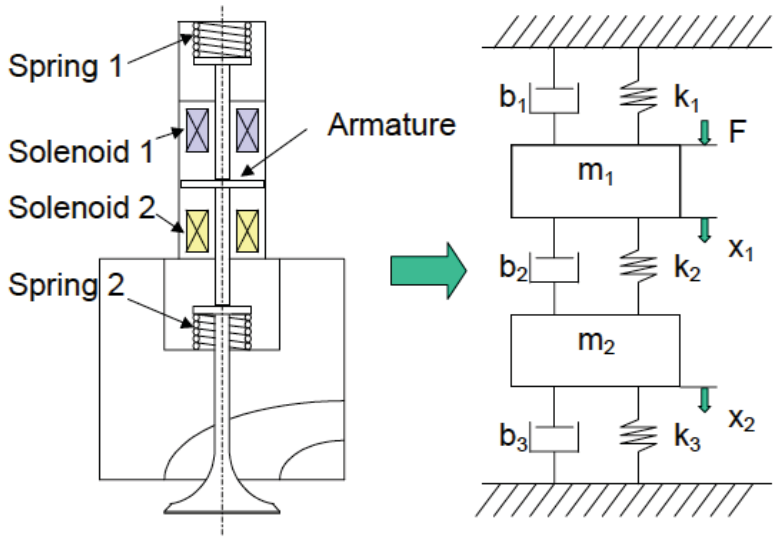
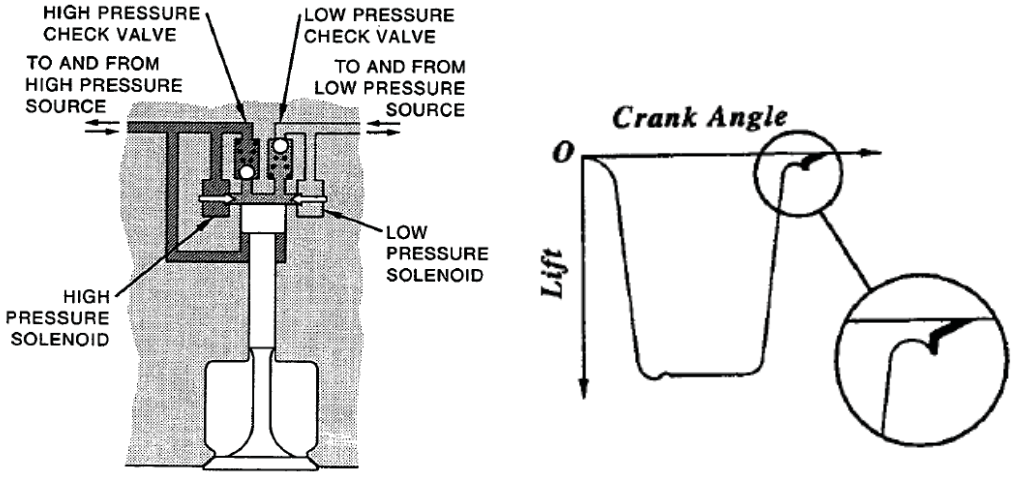


Figure 2: Electromagnetic valve actuation system [8]



(a) Illustration of EHV system (b) Lift profile

Figure 3: Ford's electrohydraulic valve actuation system [9]

In this project, the EPV system used has been developed by Cargine Engineering AB. A previous version of the system has been evaluated in [11] and a mathematical model of the same system can be found in [12]. The current version used in this project has been partially redesigned and only use one solenoid in the actuator, but the fundamental dynamics of the actuation remain the same. This project is the first multi-cylinder laboratory setting in which the system is used and the first time the system is used in a diesel engine.

2 Experimental Setup

The test rig is a Volvo D12 cylinder head, from a 12-liter heavy-duty engine with six cylinders and 24 valves controlled by 24 Cargine actuators. A dummy cylinder has been added beneath the second cylinder head as seen in Figure 4 in order to have the exhaust valves open against a variable pressure.

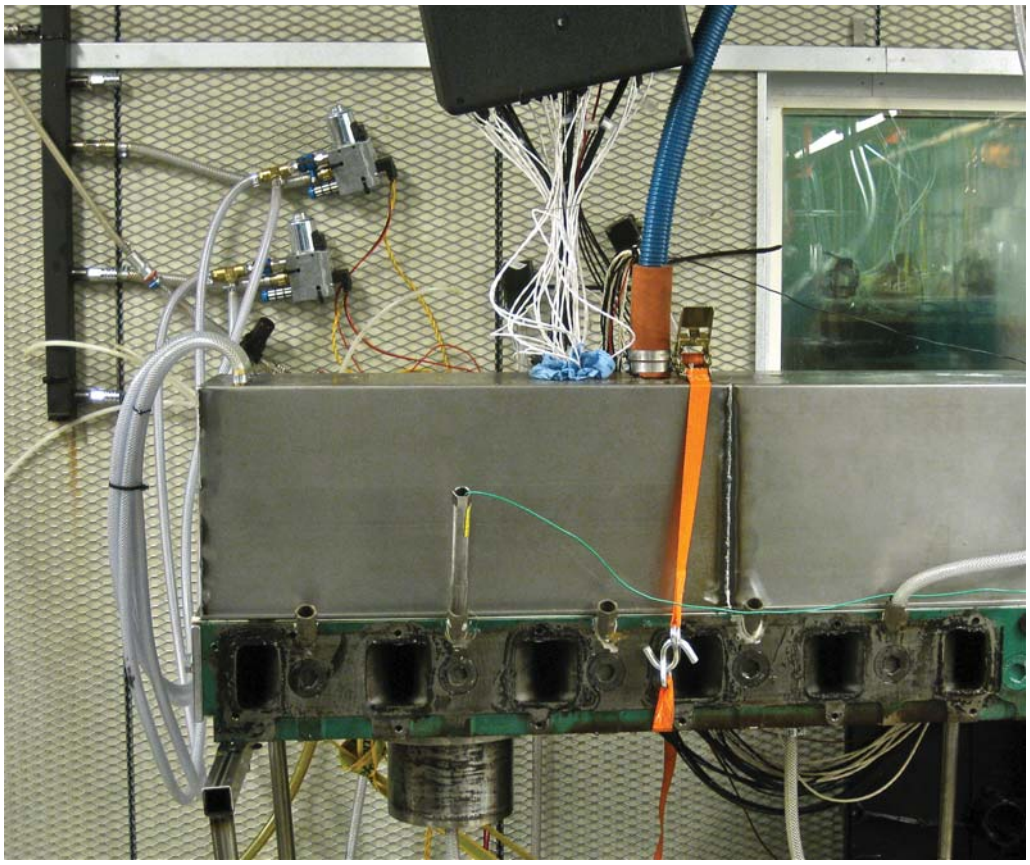


Figure 4: Experimental setup

2.1 Actuators and Sensors

The control system setup consists of a target *Personal Computer* (PC) equipped with a NI PCIe-7842R FPGA card [13]. The FPGA has programmable logic utilized for sensor data acquisition, preprocessing and control signal generation. The sensors and actuators consists of 24 Cargine EPV actuators, 24 analog lift sensors and three air pressure regulators, two of which are electronically controlled Rexroth ED05 [14] and one is a Norgren R07-200-RNKG [15]. The Rexroth pressure regulator controls the intake valve air pressure and the exhaust valve air pressure and can be seen mounted on the back wall in Figure 4. The Norgren pressure regulator is used to change the pressure in the dummy cylinder. A mock engine is attached to the FPGA card, sending a pulse each 1/5 *Crank Angle Degree* (CAD) to simulate a moving piston, and another pulse at the start of each cycle, running at 1200 *Revolutions*

Per Minute (RPM). The valve control system was built in Labview, a graphical programming language developed by National Instruments.

2.2 Electro Pneumatic Valve Actuators

Cargine EPV actuators use pressurized air to push the valves into the open position when the control signal is sent. Once in place, a hydraulic latch mechanism holds the valves in place until the control signal is driven low; thereafter the valves dislodge and close. The actuators have three parameters by which they can be controlled: the air pressure u_P, the VO timing u_{VO} and the VC timing u_{VC}. The actual signal sent to the actuators u is given by

$$u(k) = \begin{cases} 1, & \textit{if } u_{VO} \leq k \leq u_{VC} \\ 0, & \textit{otherwise} \end{cases} \tag{2.1}$$

and u_P is sent to the air pressure controller. The actuators are fed oil with 6 bar pressure and air with a pressure between 0-10 bar. The building's common air compressor provides a 50 liter tank with pressurized air at about 7 bar. The tank is coupled with an air pressure amplifier that raises the pressure to 11 bars, which is fed to the air pressure regulators seen on the back wall in Figure 4 that regulates the pressure of the air fed to the Cargine EPV actuators.

2.3 Analog Lift Sensor

The analog lift sensor was used to measure valve lift. The lift sensor is mounted as seen in Figure 5. The lift sensor outputs a square-wave with a frequency that depends on the valve lift. As the frequency was not a linear function of the lift, a second order mapping was obtained.

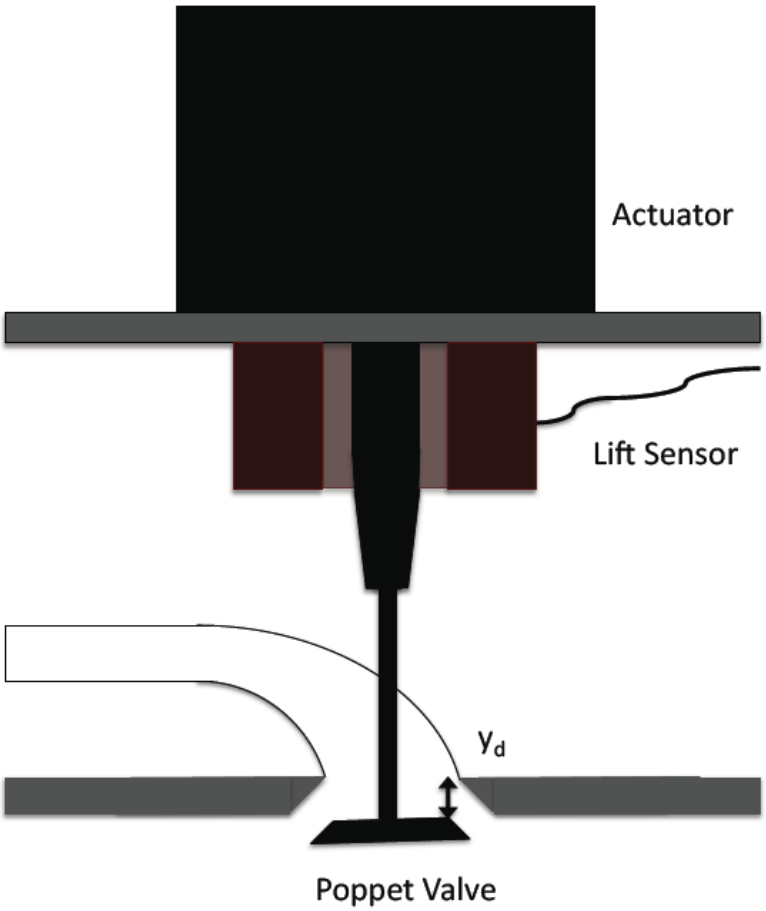


Figure 5: Actuator setup

3 Implementation

The control program is distributed in the sense that the actual controller is implemented in the labview control program, while the FPGA card sends control signals, receives measurement data, performs some signal processing, and communicates with the PC. The structure and flow of the control program has been visualized in Figure 6. The lift sensor readings is received by the FPGA card where outliers are removed and the data filtered with a moving average filter and down-sampled. The lift data is sent to the host, converted to valve lift, and then optionally fed to a Kalman filter. Each cycle, three outputs are extracted from the valve valve lift data. First, the VO time y_{VO}, defined as the CAD when the valve position has reached the trigger level T. Second, the VC time y_{VC}, defined as the the CAD when the valve position has fallen to T. Two trigger levels have been tried, 1mm and 4mm. The third output is the valve lift y_L when the valve is fully opened, calculated as the average lift during a section of the valve lift event. The first two extracted outputs can be expressed conveniently as

$$\begin{cases} T = y_d(y_{VO}) \\ T = y_d(y_{VC}) \\ y_{VO} < y_{VC} \end{cases}$$

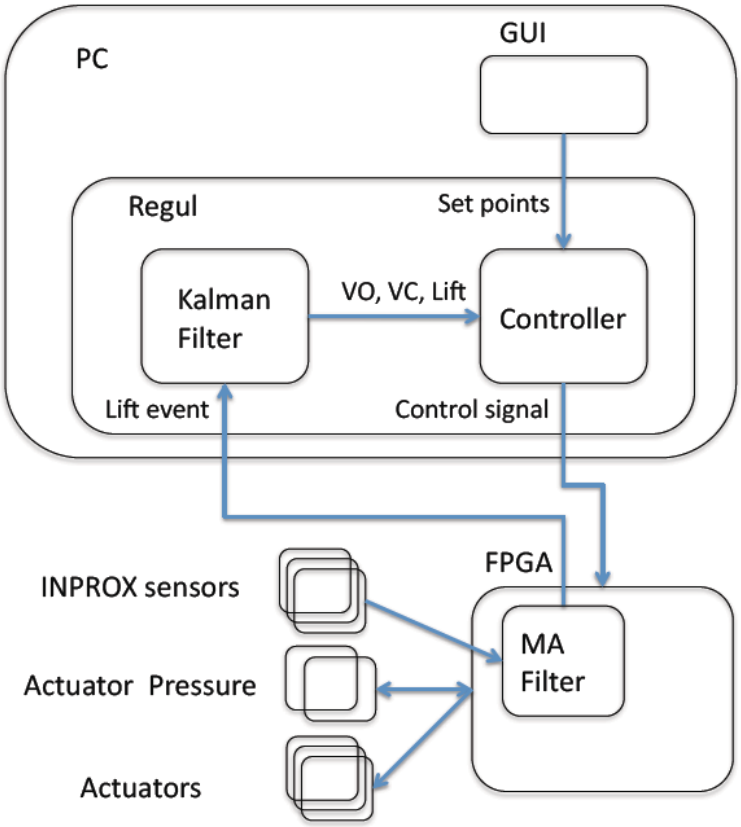


Figure 6: Control program structure

3.1 Labview

Labview is a graphical programming language where an application is constructed by wiring together provided building blocks. The Labview application has two sides, the front panel and the block diagram. The block diagram is where functional blocks are wired together and the controls and indicators of the front panel are connected. For an introduction to Labview programming and a reference manual see the Labview User Manual [16]. The front panel is the *Graphical User Interface* (GUI), designed to simulate the interface of a real hardware. The GUI of the control program can be seen in Figure 7. To the left, there are toggles to activate the control of VO and VC, in the center, the VO and VC times can be set, and to the right the valve lift is shown.

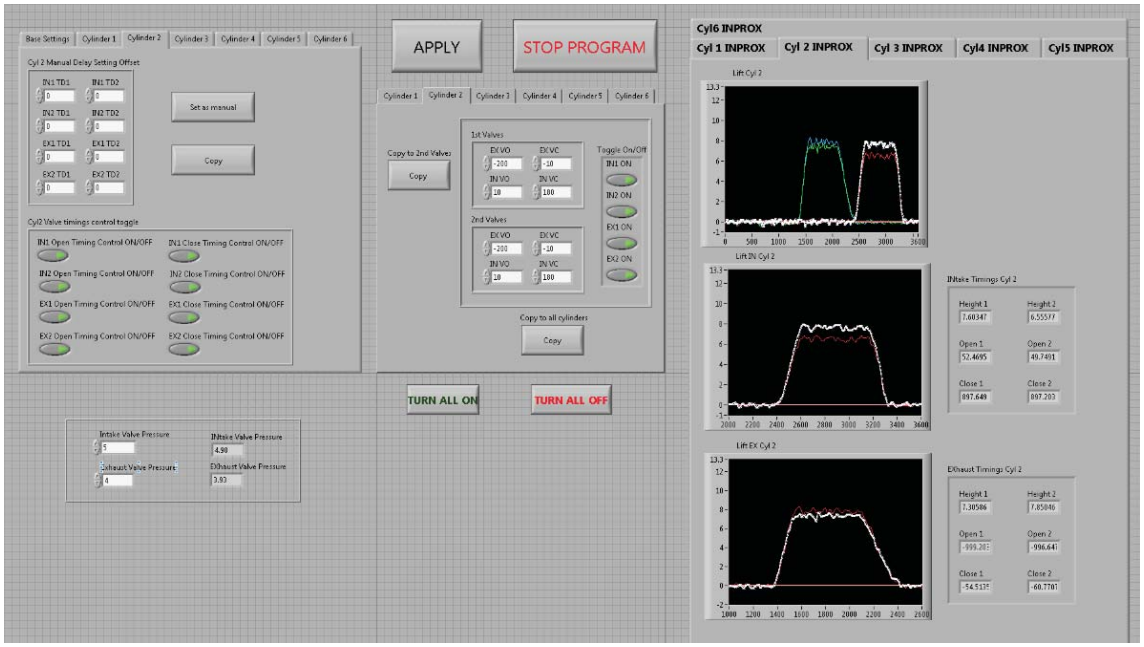


Figure 7: Labview control program

3.2 Field Programmable Gate Array

The FPGA can be reconfigured after manufacture, which gives the programmer opportunity to change the function of the FPGA card as the requirements of the application changes. The drawback is that the design is less optimized, leading to greater use of area, reduced execution speed and increased power consumption [17]. The FPGA card used, NI PCIe-7842R, has 96 digital *Digital Input/Output* (DIO) lines, 8 analog inputs, 8 analog outputs and a built in clock that runs at a frequency of 40 MHz. The FPGA card enables simultaneous execution of parallel task. In Table 1 The programmable logic blocks of the FPGA is the flip-flop, *Lookup table* (LUT), embedded block *Random Access Memory* (RAM) and DSP48 slices. The n-input LUT can implement any n-input boolean function as a truth table. Flip-flops are circuits that have two stable states and can be used as data storage elements. The flip-flop circuit change state when the control input signals change. Block RAM can

be used to store datasets or pass values between parallel loops and thereby reduce the number of flip flops used. Multiplication is resource intensive in the number of LUTs and flip-flops used. DSP48 slices integrate a 25-bit by 18-bit multiplier with adder circuitry. Please see the documentation for further details [18]. On the FPGA, data acquisition and some rudimentary signal processing take place. Each rising edge of each lift sensor triggers a FPGA DIO channel and the time between rising edges is measured. The time measurements are in clock ticks of the built-in clock, running at $40MHz$.

Table 1: FPGA specifications

NI 7842R	
FPGA type	Virtex-5 LX50
Number of flip-flops	28,800
Number of 6-input LUTs	28,800
Number of DSP48 slices ($25 * 18$ multipliers)	48
Embedded block RAM	1,728 kbits
Analog Input Channels	
Number of Analog Input channels	8
Maximum sampling rate	200 kS/s (per channel)
Analog Voltage Input Range	$\pm$10 V
Analog Output Channels	
Number of Analog Output channels	8
Maximum update rate	1 MS/s
Analog Voltage Output Range	$\pm$10 V
Digital I/O channels	
Number of Digital I/O channels	96
Minimum sampling period	5 ns
Digital Voltage Input Range	-20.0 to 20.0 V, single line

3.2.1 Communication to Host

The FPGA Communicates with the host PC, through a *Direct Memory Access* (DMA) *First In First Out* (FIFO) queue. The DMA FIFO queue allows the FPGA to directly access the RAM of the host for increased throughput compared to transferring the data between two local FIFO queues. The FPGA FIFO queue consist of block RAMs on the FPGA side, RAM memory on the host side, and an DMA engine that automatically transfers data between the two. The DMA FIFO is setup to write 6 unsigned 64 bit integers every 6th sample. The 64 bit unsigned integer consist of four valve lift readings, the CAD value and the cylinder pressure. The valve lift reading is a 8 bit unsigned integer, the CAD value is a 16 bit integer and the cylinder

pressure reading is also a 16 bit integer. Each 64 bit unsigned integer contain the valve lift readings from one cylinder. Each write cycle, one valve lift reading from each valve is sent to the host PC, which receives 600 valve lift readings per valve and cycle.

3.2.2 Analog Lift Signal

The analog lift sensor output a square-wave with a frequency depending on valve lift. The time between two rising flanks is measured on the FPGA card in clock ticks of the built in clock. In Figure 8 100 samples from an analog lift signal is presented, the signal was acquired by an oscilloscope. The signal is subject to noise which results in outliers and noise when handled by the FPGA. See section 6.1 for a discussion of

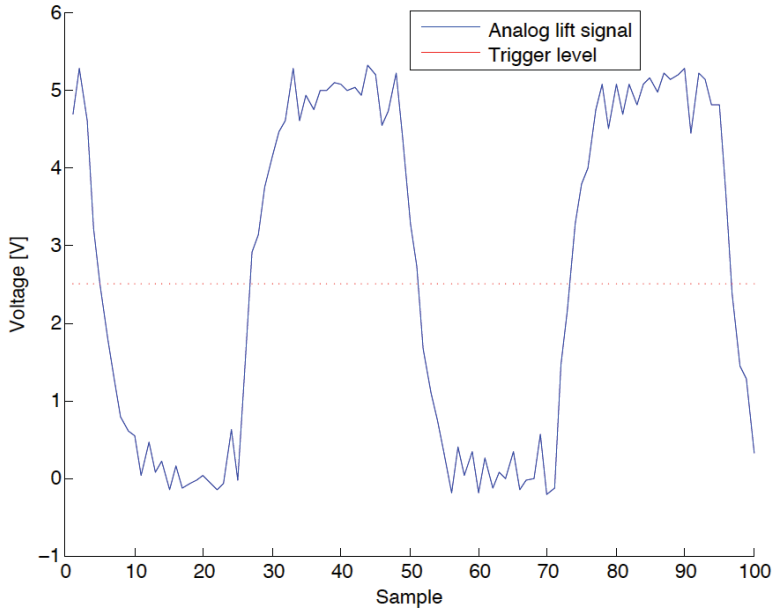


Figure 8: Analog lift signal

outliers and noise. The raw sensor data acquired by the FPGA are plotted during a lift event in Figure 9a. The unit clock ticks refers to the FPGA clock and the sample rate is 1800 samples per revolution or 3600 samples per cycle, or 5 samples per CAD. Using a simple heuristic algorithm, the outliers is easily identified and removed. The implemented solution compares the difference between two consecutive samples, y_i and y_{i+1} , and if

$$y_{i+1} - y_i = \Delta y < \Delta_{min}$$

where $\Delta_{min} = 5$, then the sample is replaced by the previous sample. A solution that was chosen for simplicity, causality and low logical requirements when implemented on the FPGA. The signal without outliers can be seen in Figure 9b.

3.2.3 Moving Average Filter and Downsampling

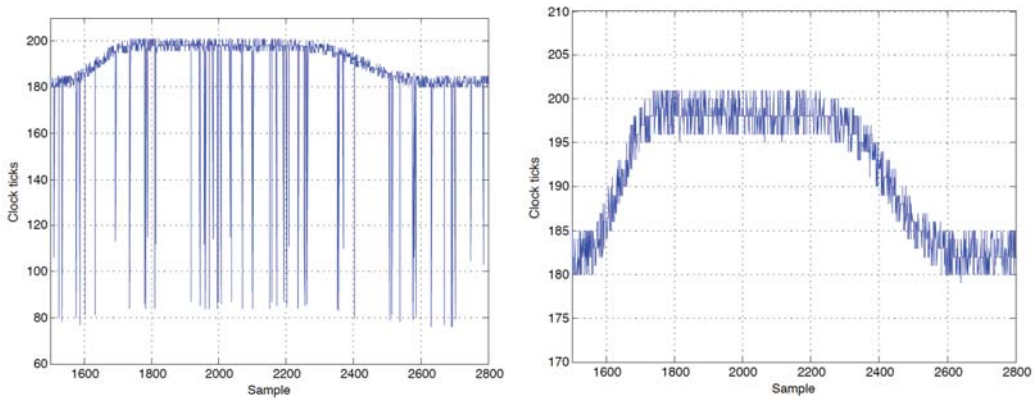
Since only one valve position can be sent every 6th sample, an average of the six most recent samples is calculated and subsequently sent. In general, it is advisable

to use unfiltered data in the identification procedure and then come up with a filter. The necessity to use one here is due to the fact that otherwise only one sixth of the data could be sent to the PC because of limitations in the amount of data that can be sent from the FPGA.

The valve position calculated on the FPGA is in clock ticks, where a lift has a resolution of about 25 clock ticks, varying between approximately 180 clock ticks when the valve is closed to approximately 205 when the valve is fully opened. To minimize the use of FPGA resources and to fit the averaged valve position reading into an 8 bit unsigned integer the built in overflow behavior is used. An 8 bit unsigned integer can represent a value between 0 and 255. A value above 255 rolls over, e.g. 260 rolls over to $4 = 260 - 256$. Thus simply adding the six lift values together will cause rollover to occur four times. The lift resolution is amplified six times, with a range of 150 ticks, but can still fit in an 8 bit unsigned integer. A closed valve will generate a value of approximately 56 to be sent to the host, and a fully open valve will generate a value of 206. Values that are translated into frequencies and then to valve lifts on the host side. In total, the averaging operation only requires five add operations per valve on the FPGA card. A comparison between the lift curve only filtered by the moving averaging filter and also Kalman filtered is seen in Figure 10.

3.3 Sensor Mapping

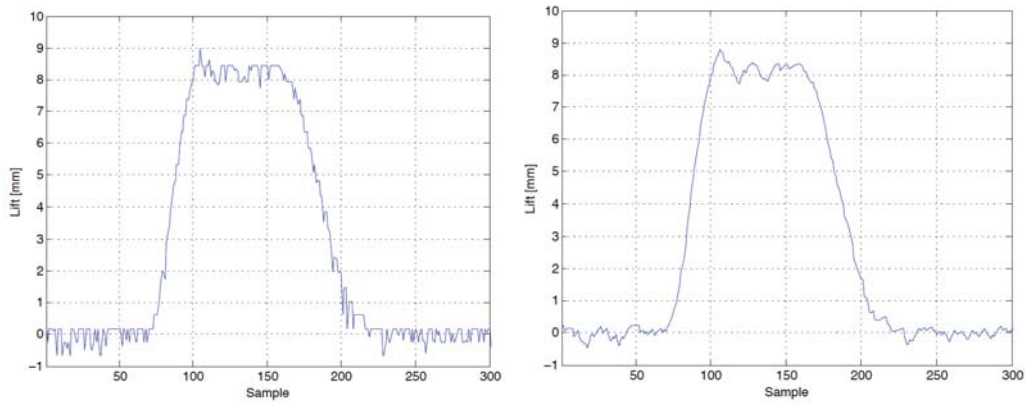
Due to the analog lift sensor being nonlinear, a second order mapping was done, using sensor readings from two different sensors. In Figure 11a the frequency of the square-wave from the lift sensor is shown at different positions. It can be seen that there is a significant frequency discrepancy between the two sensors, however the discrepancy predominantly being a constant offset in frequency, which can be seen more clearly in Figure 11b, where the mean frequency difference has been removed from the second sensor. It is assumed that all sensors can be normalized by removing the mean frequency difference when the valves are closed. With the relative position



(a) Raw lift sensor lift curve

(b) Raw lift sensor lift curve with outliers removed

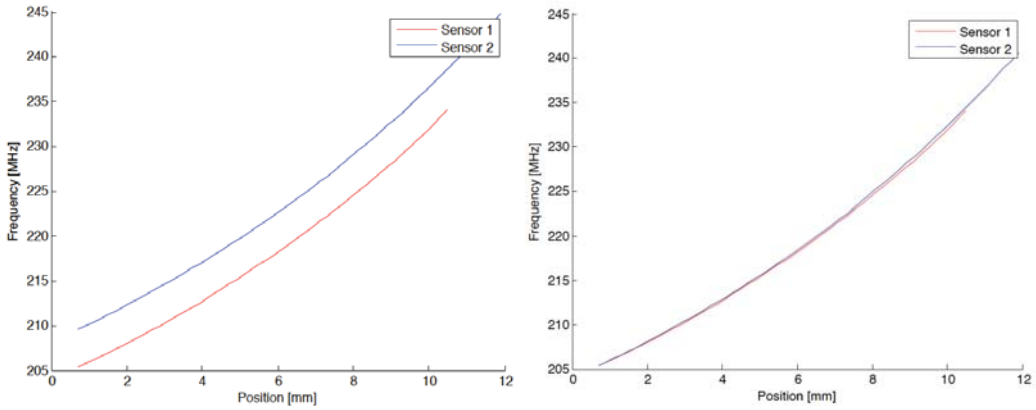
Figure 9: Raw lift sensor readings



(a) MA filtered lift curve (b) MA and Kalman filtered lift curve

Figure 10: Filtered lift sensor readings

available with sufficient accuracy, the absolute position profile can be obtained by observing that the position of the valve is known when the valve is closed, thus the frequency offset and the absolute valve position can be calculated.



(a) Calibration measurement (b) Calibration measurement with frequency offset removed

Figure 11: Lift calibration using two sensors

3.4 Labview Implementation

To implement feedback from the previous cycle, start and stop timings of the control pulse need to be set at several different times because there is basically always valves open. Therefor one cycle was split into 6 parts, and in each part, outputs are calculated for two exhaust valves and two intake valves and control pulse timings sent to the FPGA. The duration of a valve lift event has an absolute constraint in that it can not open for roughly more than 360 CAD, since otherwise the piston would hit the open valves. In practice, the valves are open far shorter periods. This gives ample time to calculate the control signal with the implemented solution. For

details on Labview programming see the reference manual or [19], one of the many books available on the subject.

Due to time constraints due to the scope of the thesis, a set of simple controllers where implemented. The controller setup consists of three *Single Input Single Output* (SISO) systems [20], where the lift height is controlled by the actuator pressure, the start of the control pulse controls the VO time, and VC time is controlled by the end timing of the control pulse. All controllers are *Proportional Integral* (PI) controllers [20] with very conservative controller gains.

4 Modeling

The purpose of the identified model is to facilitate filtering of the lift sensor signal, in an attempt to get more consistent lift data. For a general treatment of system identification and modeling see [21]. Two types of discrete time models where evaluated, *Autoregressive moving average with exogenous input* (ARMAX) models and state-space models. The model was the used to filter the output sequence to get a better VO and VC signals. The state-space models were developed using the *n4sid* Matlab command, see the Matlab documentation [22], or Ljung [23] for algorithm details. The input sequence was defined as the pulse sent to the actuators multiplied by the actuator pressure, and the output sequence is just the filtered and down-sampled lift measurement. The control sequence u_d is defined as

$$u_d = u_p \cdot u \tag{4.1}$$

where u_p is the actuator pressure and u is the generated physical control pulse sent to the actuators, defined in Equation (2.1). For successful identification it is necessary to have enough variability in the input [21], which is why the control signals were generated by the following procedure. The control parameters, u_p, u_{VO} and u_{VC} were only changed if the realization of a binary random variable X_u was one. The binary random variable had a probability of $P(X_u = 1) = 0.15$. Each input that was to be changed was then drawn from a range of valid control signals with constant probability density function. The Input-output data were subdivided into an estimation part of 80% of the data and a evaluation part of 20% of the data. The models were estimated using the estimation data and evaluated on the validation data based on the criteria introduced in section 4.3.

4.1 ARMAX Modeling

ARMAX models have the following structure:

$$A(z^{-1})y_k = z^{-d}B(z^{-1})u_k + C(z^{-1})w_k \tag{4.2}$$

where the A polynomial is the autoregressive part, the B polynomial is a moving average of the input signal and the C polynomial is a moving average part of noise terms. The input $u_k \in \mathbb{R}^m$ is modeled affecting the output $y_k \in \mathbb{R}^p$ with a delay d time steps and the noise process $\{w_k\}$ is assumed to be a white-noise stochastic process. The A,B and C polynomials have the structure

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} \cdots + a_{na} z^{-na} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} \cdots + b_{nb} z^{-nb} \\ C(z^{-1}) &= 1 + c_1 z^{-1} \cdots + c_{nc} z^{-nc} \end{aligned} \tag{4.3}$$

4.2 State Space Modeling

The state-space equations for system with m inputs and p outputs is given by the equations

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned} \tag{4.4}$$

with state vector $x_k \in \mathbb{R}^n$ and noise sequences $w_k \in \mathbb{R}^n$, $e_k \in \mathbb{R}^p$ affecting the dynamics of the system and the output respectively.

4.3 Model Evaluation

The developed models were evaluated on the evaluation data set by three evaluation statistics based on the prediction errors and residual tests.

The *Variance Accounted For* (VAF) statistic defined as

$$\tau_{VAF} = \left(1 - \frac{(y_N - \hat{y}_N)^T (y_N - \hat{y}_N)}{\hat{y}_N^T y_N}\right) \times 100\%$$

where y_N is the N point vector containing the output data and $\hat{y}_N$ is the estimated output. The *Akaike Final Prediction Error* (FPE) defined as

$$FPE(p) = \frac{N+p}{N-p} V$$

where p is the identified number of parameters and V is the loss function defined as

$$V = det\left(\frac{1}{N}\sum_{i=1}^{N} \epsilon(i, \theta_N)\epsilon(i, \hat{\theta}_N)^T\right)$$

Akaike's Information Criterion (AIC) is defined as

$$AIC(p) = log(V) + \frac{2p}{N}$$

where V is the loss function defined earlier.

The transfer function of both the ARMAX models and the state space models can be expressed as $Y(z^{-1}) = H_u(z^{-1}) + H_w(z^{-1})W(z^{-1})$. The residuals $\epsilon(z^{-1})$, of the evaluated models are obtained by

$$\epsilon(z^{-1}) = \hat{H}_w^{-1}(z^{-1})(Y(z^{-1}) - \hat{H}_u(z^{-1})U(z^{-1})) \tag{4.5}$$

The residuals can be considered a disturbance input that would explain the disparity between the observed data and data from the estimated model. If the model is correct the residuals should not contain any structure, especially they should be uncorrelated to inputs and outputs [21].

5 Results

In this section results from the sensor calibration, model identification and the implemented controller are presented.

5.1 Lift Sensor Noise

The lift sensor readings are subject to noise. The noise is concluded to be white noise and the estimated power spectrum is shown in Figure 12, the power spectrum is calculated for 64 frequencies, logarithmically spaced. The power spectrum has been calculated for a valve turned off, all 3600 data points are used without the average filter, using the Matlab *psd* command and a Kaiser window, see Matlab documentation for details [22].

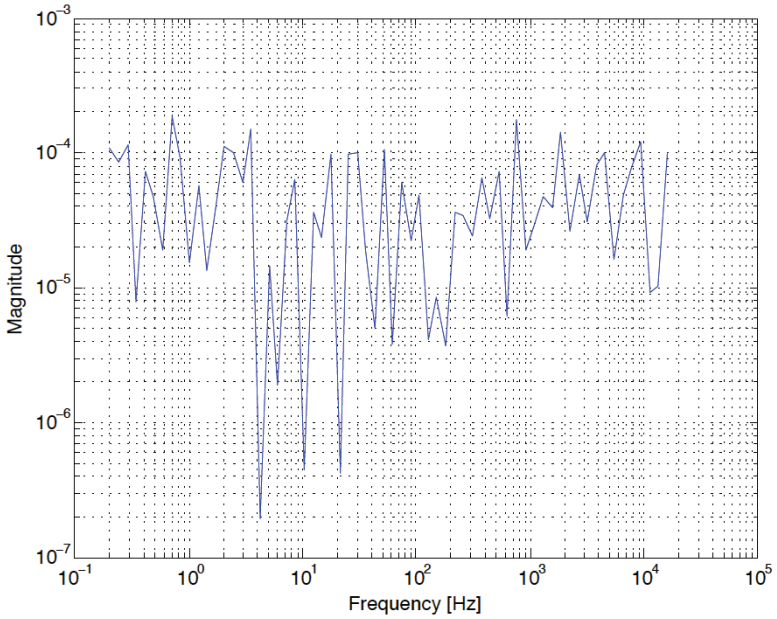


Figure 12: Power spectrum estimate of the noise affecting the lift sensor using a periodogram with Kaiser window

5.2 Model Evaluation

The statistics for the identified models are presented in Figure 13. For easier comparison, an offset of 2.5 has been added to the AIC statistic and the VAF statistic has ben replace by 1-VAF. The raw statistics can be found in Table 3 and the model parameters in Table 2. A filter based was implemented in Labview based on the best model, the third order n4s3 model, however the Labview program could not finish execution in time consistently. The n4s2 model was then chosen as a compromise between performance and implementation requirements. Figure 14 show the cross correlation of the residuals and correlation between the input and the residuals for the three state space models. The residuals of the first order model clearly contain

structure, but to a lesser extent this is also true for the models of order 2 and three as well. All three models exhibit cross correlation between input and output, that manifests itself in that the models predict too low lift with lower than average values of u_p and too high lift with higher than average values of u_p. The n4s2 state space model is given by equation (4.4) and the matrices

$$A = \begin{bmatrix} 0.9682 & -0.04126 \\ 0.08114 & 0.8981 \end{bmatrix}, \qquad B = 10^{-5} \begin{bmatrix} 2.814 \\ 39.75 \end{bmatrix}, \tag{5.1}$$
$$C = \begin{bmatrix} 0571.1 & 12.47 \end{bmatrix}, \qquad D = 0$$

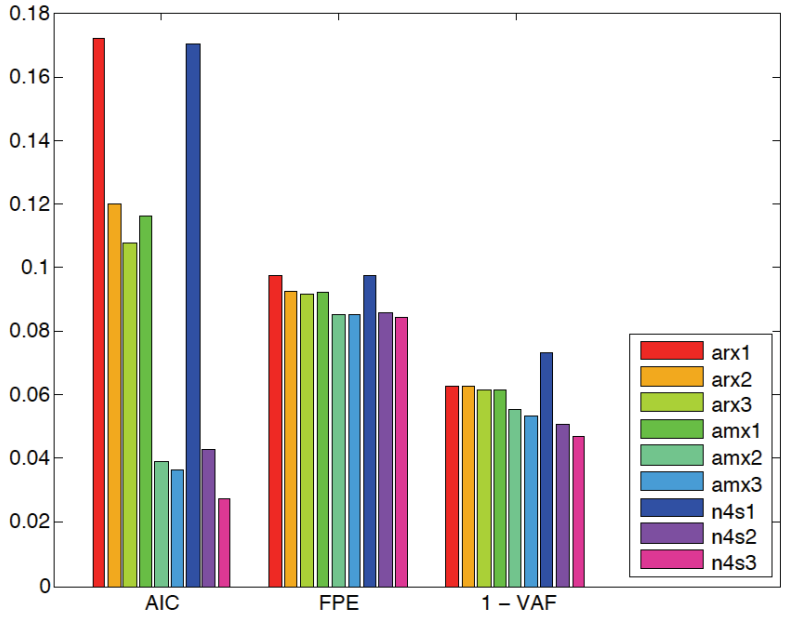


Figure 13: Exhaust valve model evaluation. Smaller values imply a better fit. An offset has been added to the AIC statistic

Table 2: Exhaust valve models

Name	Type	n_a	n_b	n_c	n (n4sid only)	delay
arx1	ARX	2	1	-	-	23
arx2	ARX	3	2	-	-	23
arx3	ARX	4	3	-	-	23
amx1	ARMAX	2	1	1	-	23
amx2	ARMAX	3	2	2	-	23
amx3	ARMAX	4	3	3	-	23
n4s1	n4sid	-	-	-	1	18
n4s2	n4sid	-	-	-	2	18
n4s3	n4sid	-	-	-	3	18

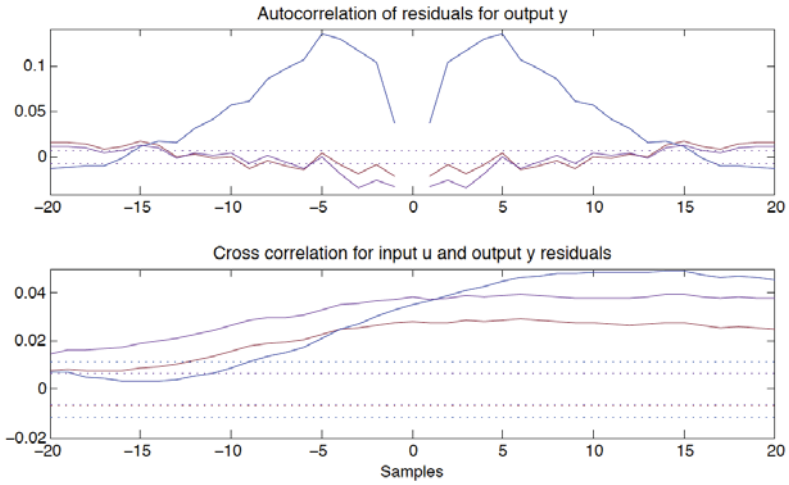


Figure 14: Correlation of Residuals for state space models, n4s1 (blue), n4s2 (purple) and n4s3 (pink).

Table 3: Exhaust valve model evaluation data

Model	AIC	FPE	VAF [%]
arx1	-2.3284	0.0975	93.75
arx2	-2.3799	0.0926	93.71
arx3	-2.3920	0.0914	93.84
amx1	-2.3836	0.0922	93.82
amx2	-2.4610	0.0854	94.44
amx3	-2.4636	0.0851	94.67
n4s1	-2.3295	0.0973	92.69
n4s2	-2.4572	0.0857	94.94
n4s3	-2.4722	0.0844	95.30

5.3 Controller

A Kalman filter based on the n4s2 system model was implemented in the controller for one valve. The Kalman gain K is

$$K = 10^{-4} \begin{bmatrix} 5.243 \\ 18.02 \end{bmatrix} \tag{5.2}$$

To evaluate the controller with and without the use of the Kalman filter two kinds of input signal were used. The first signal used was a constant signal of 5 bar actuator pressure and the signal sent to the actuator chosen according to Equation (2.1) with $u_{VO} = -200$ CAD BTDC and $u_{VC} = 0$ CAD BTDC. The second signal was the same type of signal as the one used in the modeling part, but with constant air pressure at 5 bar. For both input signals a trigger level of 1 mm and 4 mm was tested and the length of the input sequences was 200 cycles.

5.3.1 Constant Input

The variance of the first input sequence, the constant input sequence, is presented in Table 4.

Table 4: Variance of steady state VO and VC with and without Kalman filter

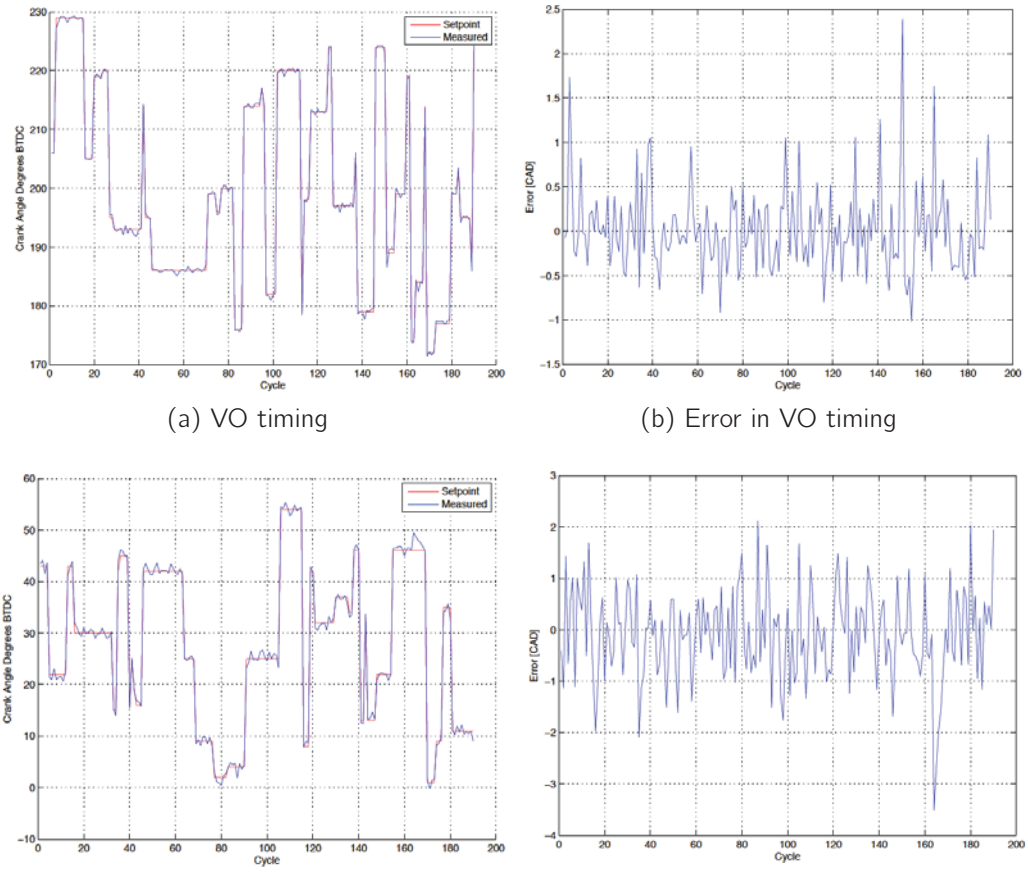
	With Kalman Filter	w/o Kalman filter
VO 1 mm trigger level	0.1447	0.5364
VC 1 mm trigger level	0.7494	0.4790
VO 4 mm trigger level	0.0921	0.1573
VC 4 mm trigger level	0.1913	0.3297

5.3.2 Stochastic Input

The controller with Kalman filter response to a stochastic VO and VC input signal similar to the signal used in identification can be seen in Figure 15 for 1 mm trigger level and in Figure 16 for 4 mm trigger level. The response for the controller without the Kalman filter can be seen in Figure 17 for 1 mm trigger level and in Figure 18 for 4 mm trigger level. The variance and mean of the error in the just mentioned experiments are presented in Table 5.

Table 5: Mean and variance of error sequences

		With Kalman filter		w/o Kalman filter	
		variance	mean	variance	mean
1 mm Trigger	VO error	0.23	0.02	0.53	-2.72×10^{-4}
	VC error	0.79	-0.10	0.72	-0.08
4 mm Trigger	VO error	0.85	0.08	0.28	-0.0039
	VC error	0.3126	-0.2575	0.7984	-0.0792



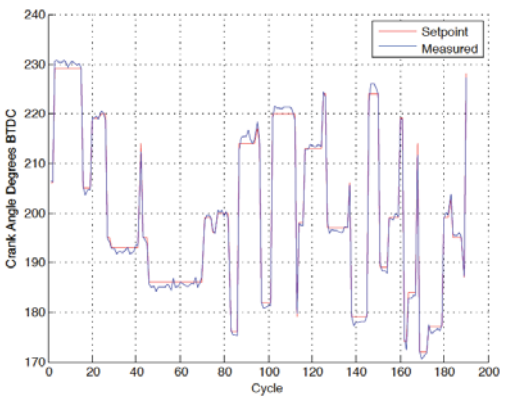
(a) VO timing

(b) Error in VO timing

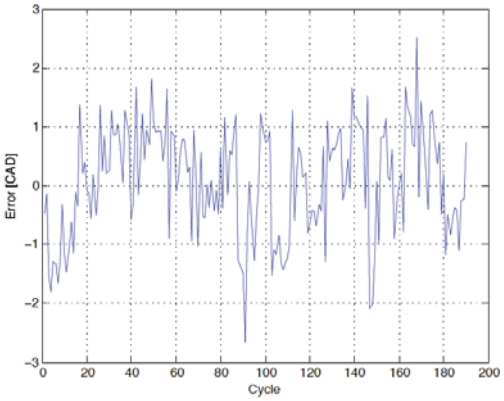
(c) VC timing

(d) Error in VC timing

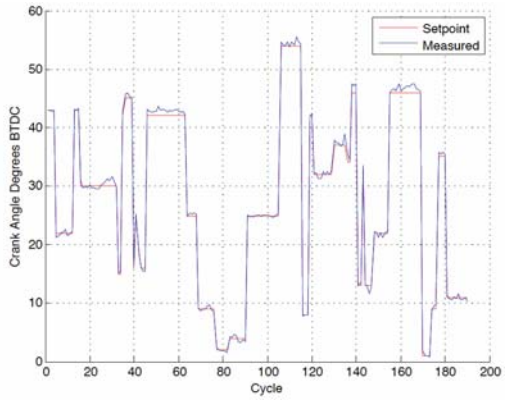
Figure 15: Control of exhaust VO and VC with Kalman filter (1mm trigger level)



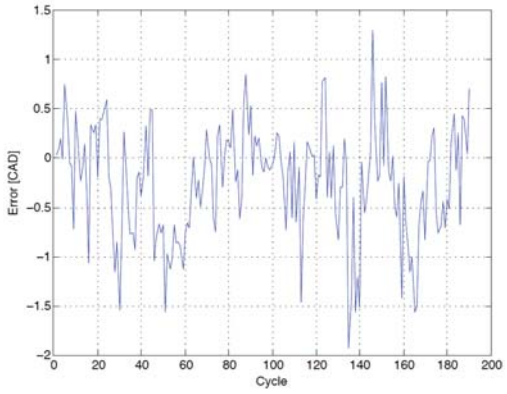
(a) VO timing



(b) Error in VO timing

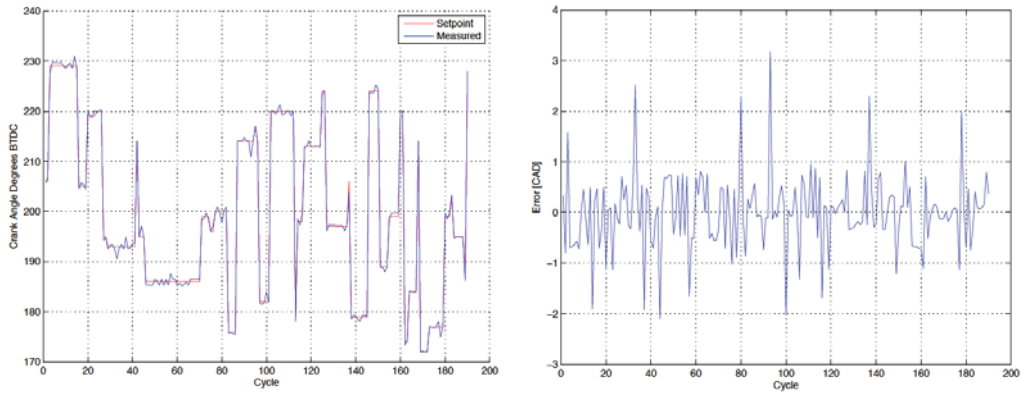


(c) VC timing

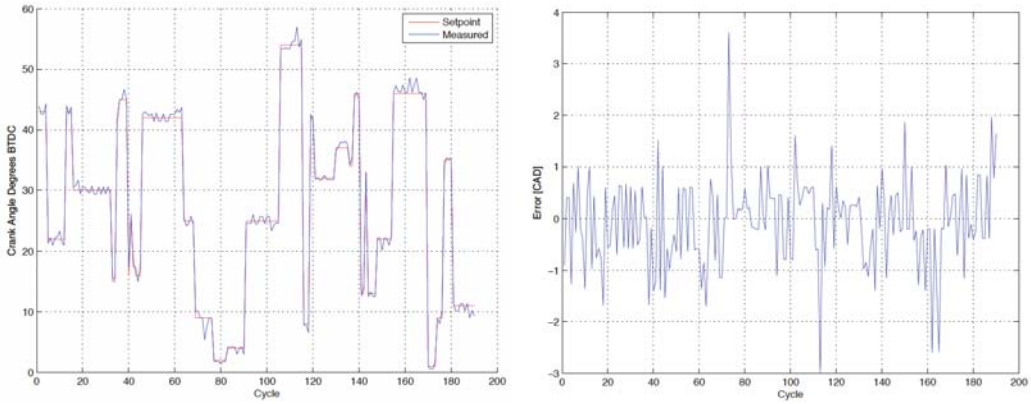


(d) Error in VC timing

Figure 16: Control of exhaust VO and VC with Kalman filter (4mm trigger level)

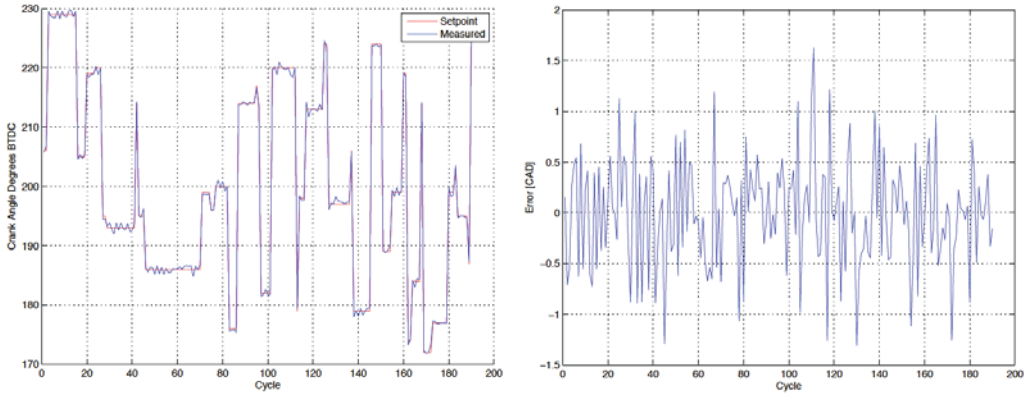


(a) VO timing (b) Error in VO timing



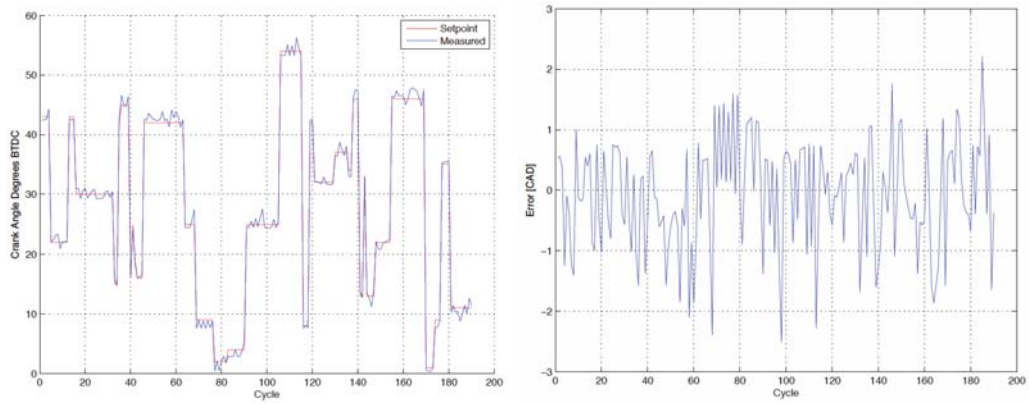
(c) VC timing (d) Error in VC timing

Figure 17: Control of exhaust VO and VC without Kalman filter (1mm trigger level)



(a) VO timing

(b) Error in VO timing



(c) VC timing

(d) Error in VC timing

Figure 18: Control of exhaust VO and VC without Kalman filter (4mm trigger level)

6 Discussion

A second order model of the exhaust valve was identified. A Kalman filter and a simple feedback was implemented on one valve. Both with and without the Kalman filter the feedback control achieved variances beneath 1 CAD. From the results, there is some support to conclude that the Kalman filter did improve the performance, however not by much and not consistently. The valve used for both model identification and the control experiments was the one least susceptible to noise. It is reasonable to believe that the Kalman filter would give stronger benefit on valves affected more by noise.

There is some remaining structure in the auto correlation of output residuals and cross correlation between input and output residuals as seen in Figure 14. The cross correlation between input and output residuals suggest some dynamics that the model fails to capture and that model complexity is insufficient. Increased model complexity, maybe in the form of a nonlinear model, might be needed since increased model order was not sufficient remove the cross correlation. There are several sources to the remaining variance. The resolution of the FPGA is one fifth CAD and possibly give a quantization error of one tenth CAD. Variations in supplied air pressure probably have an effect on the variance. The model that the Kalman filter is was based on, does not describe the system satisfactory as can be seen by the correlation analysis and Figure 14.

Having access to whole lift curve each cycle gives endless possibilities in what data to extract and use for the cycle-to-cycle feedback. It is interesting to note that there is not that much difference in variance between the two trigger levels 1 mm and 4 mm. It would be interesting to investigate the cross-correlation between the two, because if the added noise is uncorrelated, better information can be obtained by combining several points on each flank of the lift curve.

6.1 Lift Signal Noise

The outliers seen in Figure 9a have a certain regularity to them, the measured time seem to be approximately half the clock ticks of the valid data points. The probable cause of this is that the analog lift signal seen in Figure 8 triggers a second time on the zero-crossing on the falling flank when the signal is only supposed to trigger on the rising flank. This would explain the characteristics of the outliers. However, an occasional faulty trigger event would result in two consecutive samples of half the number of clock ticks, but in Figure 8 there is often several correct samples in between two faulty samples. This implies that the FPGA triggers only on the falling flank for several samples, a behavior that lacks explanation.

6.2 Implementation Issues

Given the timing requirements of feedback from the previous cycle, the PC was unable to run more than one Kalman filter in the control program, and even with one Kalman filter the computations was not finished in time or not deterministic enough, resulting in missed deadlines. There are at least thee possible ways to mitigate this problem. Implementing the Kalman filter on the FPGA would lead to a reduced load on the computer's CPU. It would require a fixed-point implementation of the Kalman filter which is not feasible to implement for all 24 valves on the current FPGA card. When attempting a fixed-point low-pass filter of first order, the FPGA requirements scaled up nicely when used on 12 lift signals, utilizing less than 50% of the FPGA resources in terms of both LUTs and flip-flops. But when attempting to use it on all 24 valves, the utilization reached above 200%. Further investigation is needed to determine the reason for this behavior and if a second FPGA card or an improved one would suffice to solve the issue. The second possible solution would be to optimize the Kalman filter on the PC, removing features of the built-in Labview version and neglecting small terms. The third would be to remove the need for Kalman filtering by reducing the noise level which seem possible since with only one sensor attached, the signal did not contain noise. The noise origin is unknown, however it is likely to be common ground related, since all lift input share the same power source and have the same ground signal on the FPGA card. Furthermore, filtering one of the lift signals through an analog low-pass filter did not lead to any visually detectable reduction in noise level on the raw analog lift sensor signal observed in an oscilloscope, the signal still had the noise as in Figure 8.

6.3 Future Work

The next step would be to model and implement model based control of lift, VO and VC timings with the additional input parameter of cylinder pressure. Such modeling was attempted during the course of this project, but it turned out hard to fit a model. One of the reasons was that the length of the datasets were limited. If the whole lift curve was to be saved, the data acquisition and processing became to cumbersome. During the acquisition of data, the disk could not write fast enough to have more than a couple of hundred cycles of data at a time and the processing of datasets with a large number of samples became time-consuming. Now, when the preprocessing is done during execution, only a small number of values need to be saved each cycle, which should make it possible to have larger data sets and a better chance of developing models. There also need to be some way to change the speed of the mock engine to cover that aspect. One way would be to implement the mock engine on the FPGA instead. The overarching final goal would be to integrate the valve control with the existing engine control platform, implemented in C++, Matlab and Real-time Workshop.

7 Conclusion

The main result of this project is that a FPGA based feedback controller was implemented for all 24 valves in Labview. A second order model of the exhaust valve was identified. A Kalman filter and a simple feedback was implemented on one valve. Satisfactory control of VO and VC and Lift for a fixed engine condition was obtained. The Kalman filtering in the controller had only small effect on the controller performance. In order to have a Kalman filter on all valves, optimization of the filter implementation, some kind of signal condition, or additional computational resources is needed.

References

[1] Cargine Engineering AB, 2011, http://www.cargine.com [25-11-2011].

[2] B. Johansson, *Förbränningsmotorer*, Faculty of Engineering, Lund University, 2006.

[3] S. Trajkovic, P. Tunestål, and B. Johansson, "A simulation study quantifying the effects of drive cycle characteristics on the performance of a pneumatic hybrid bus," in *Proceedings of the ASME 2010 Internal Combustion Engine Division Fall Technical Conference*, San Antonio, TX, 2010.

[4] K. Hatano, K. Iida, H. Higashi, and S. Murata, "Development of a new multi-mode variable valve timing engine," in *SAE Technical Paper 930878*, 1993.

[5] C. Brüstle and D. Schwarzenthal, "Variocam plus - a highlight of the porsche 911 turbo engine," in *SAE Technical Paper 2001-01-0245*, 2001.

[6] R. Flierl and M. Klüting, "The third generation of valvetrains – new fully variable valvetrains for throttle-free load control," in *SAE Technical Paper 2000-01-1227*, 2000.

[7] R. Hofmann, J. Liebl, M. Klueting, and R. Flierl, "The new BMW 4-cylinder gasoline engine-reduction of fuel consumption without compromise," in *Proceedings. JSAE Annual Congress*, 2001.

[8] C. Tai and T. Tsao, "Control of an electromechanical camless valve actuatuor," in *Proceedings of the American Control Conference, Anchorage, AK May 8-10*, 2002.

[9] M. M. Schechter and M. B. Levin, "Camless engine," in *SAE Technical Paper 960581*, 1996.

[10] S. Trajkovic, "The pneumatic hybrid vehicle," Ph.D. dissertation, Lund University, 2010.

[11] A. Milosavljevic and S. Trajkovic, "Fpga controlled pneumatic variable valve actuation," in *SAE Technical Paper 2006-01-0041*, 2006.

[12] J. Ma, H. Schock, U. Carlson, A. Höglund, and M. Hedman, "Analysis and modeling of an electronically controlled pneumatic hydraulic valve for an automotive engine," in *SAE Technical Paper 2006-01-0042*, 2006.

[13] National Instruments Corporation, *NI R Series Multifunction RIO Specifications*, 2009, http://www.ni.com/pdf/manuals/372492c.pdf [23-11-2011].

[14] *Data Sheet, BoshRexroth E/P pressure regulator, Series ED05*, http://www.boschrexroth.com/pneumatics-catalog/Pdf.cfm?Language=EN&file=en/pdf/PDF_p58395_en.pdf.

[15] Norgren Ltd, 2011, http://www.norgren.com/document_resources/usa/R07.pdf [28-11-2011].

[16] National Instruments Corporation, *LabVIEW User Manual*, 2003, http://www.ni.com/pdf/manuals/320999e.pdf.

[17] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," in *Proceedings of the FPGA '06 Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, ACM New York, NY, USA, 2006.

[18] National Instruments Corporation, *NI PCIe-7842R multifunction RIO Documentation*, 2008, http://sine.ni.com/ds/app/doc/p/id/ds-98/lang/en [23-11-2011].

[19] J. Travis and J. Kring, *LabVIEW for Everyone: Graphical Programming Made Easy and Fun*, 3rd ed. Prentice Hall, 2006.

[20] K. J. Åstrom and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, NJ, 2008.

[21] R. Johansson, *System Modeling and Identification*, 2nd ed. Dept. Automatic Control, Lund University: Prentice Hall, Englewood Cliffs, 2011.

[22] The MathWorks Incorporation, *Matlab Documentation*, 2011, http://www.mathworks.se/help/techdoc/ [23-11-2011].

[23] L. Ljung, *System Identification: Theory for the User*. Prentice-Hall, Upper Saddle River, NJ, 1999.