Non-linear Identification, Estimation and Control of Automotive Powertrains

by

Arkadiusz Stanisław Dutka

THESIS

Submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

Industrial Control Centre

Department of Electronic and Electrical Engineering

University of Strathclyde, Glasgow, UK

March 2005

Declaration of Author's Rights

Abstract

Increasingly tight emission regulations put a pressure on control engineers to come up with improved engine control systems. The task is difficult, as it is desired to minimize complexity, cost and maximize reliability and performance, all at the same time. Fortunately, modern control techniques offer assistance in achieving these goals. This motivation resulted in a range of topics developed in this thesis. A modelling, estimation and fault detection theory is presented. The estimation theory is often used for the system identification, but its main application is the model-based filtering, so important in real systems. The real systems are subject to failures. A theoretical development of the fault detection algorithm for non-linear systems is presented. The emphasis moves then to the control algorithms design. The non-linear algorithms based on the state-dependent model structure are introduced. An extension of the state-dependent Riccati equation method with a future trajectory prediction is developed. Also, the non-linear version of generalized predictive control algorithm is presented. Optimality of solutions is analyzed and corrections to algorithms are introduced to preserve the optimality. The theory needs practical verification. The identification of the spark ignition engine is presented next. A data-driven system identification method is developed. It provides an accurate model for control design purposes. The predictive control algorithm design is presented next. A simple air-fuel ratio control as well as a full multivariable control system design, with a torque as an output, is introduced. Improved tracking and tighter air-fuel ratio regulation is achieved. The control system efficiency may be impaired by the system noise and the model uncertainty. For that reason the model-based estimation techniques are very important. It is demonstrated that not only the noise immunity, but also robustness is significantly improved when Kalman filtering methods are employed. Last important topic of fault diagnosis is then presented. Faults must be detected, isolated and identified to enable successful control system re-configuration.

Acknowledgements

I want to thank my supervisors, Prof. Mike Grimble and Dr. Andrzej Ordys for their support, guidance and creating the environment that helped to pursue my research. Great links with the industry and leading scientists are the greatest asset in today's world where access to the information is so important. I would also like to acknowledge a great atmosphere created by all colleagues working in the Industrial Control Centre and useful, inspiring discussions we had.

I would also like to give special thanks to Dr. Hossein Javaherian for valuable introduction, assistance and help with the engine application. I would also like to thank Dr. Gerrit van der Molen and Mr. Andy Buchanan for project discussions and comments.

Finally I would like to thank my parents and Anna for their encouragement and support during the work.

List of contents

Chapter 1

Introduction

In this introductory chapter a motivation for work presented in this thesis is given. The discussion starts in section 1.1 with the justification for control, or more precisely, non-linear control. The control engineering is a multidisciplinary area and system modelling is a foundation for design methods presented in this thesis. For that reason a brief introduction to the modelling is given in this opening chapter in section 1.2. A control theory requires some justification and demonstration of potential benefits. That point will be supported by the application. The identification, control, estimation and fault detection for the spark ignition engine is a subject of consideration in this thesis. A short introduction of the problem is given in section 1.3. Finally the overview of the thesis is presented in section 1.4 followed by the main contributions in section 1.5 and the list of publications in section 1.6.

The control is an extremely important area of engineering. However, the importance of control is very different across the industries. The automotive powertrain control is one of the areas where control stands out as the main area of interest. Automotive application requires a comprehensive approach to the control design. A wide scope of issues ranging from the system identification, estimation through the control to the fault detection and condition monitoring should be considered. To be able to tackle this wide array of engineering challenges the non-

linear control and estimation techniques must be well understood. This will give the momentum that will be utilized during the automotive application. The algorithms proposed in this thesis are not assumed to be restricted by the computational capabilities of the target hardware. The developments are intended to find the upper bound for achievable performance. However, a range of algorithms is introduced and some of them require only modest computing power.

1.1 Real-world control engineering

The control engineer must possess a vast range of both theoretical and practical skills. The multidisciplinary nature of control engineering necessitates achieving a high level of understanding of physics of the considered problem. In addition to that, the mathematical analysis tools, programming skills and implementation knowledge involving modern microprocessor technology and data processing must be involved.

The physical phenomena must be well understood during modelling. Sufficient system knowledge allows necessary simplifications in the model to be introduced. It is always advantageous to use the system-specific engineering knowledge for modelling. However, some models are developed purely from measurements and for these models both: the model structure and parameters must be identified. The model must capture key system features and, at the same time, be simple enough for control system design. For the data driven system identification simplicity of the model is a key to success. Parameter estimation for complex models of a non-linear nature may be extremely difficult. This brings the problem of the model parameter identification to attention. The model parameter identification may be considered as an estimation problem. It can be advantageous if the model structure is based on the physical insight. Some simplifications are often considered but it is important to make sure that these do not introduce too significant model mismatch. Most well established estimation techniques may be employed for parameter identification. Although modelling is extremely important, it attracts far

less attention in the literature in favour of control algorithms design. This is mainly due to the application dependent nature of the identification problem. Each system is different and methods that are successful on one application may fail when used somewhere else. In most control system design applications modelling and identification is the most involved task. Sometimes it takes years of effort and significant expertise to establish a reliable model.

Control algorithm design is by far more rigorous than the modelling. The choice of a design technique depends purely on the requirements. The concept of feedback is a central idea in control engineering. A parameter of interest, if measured, may be compared with its desired value and action is taken according to the error signal. This however requires a parameter measurement to be available to the control system. This seemingly easy philosophy gets complicated when the parameter of interest is not measured. In that case the model may be used to extract unmeasured information from the available input and output signals. This of course requires the model to be as accurate as possible. Some uncertainty is inevitable and must be accounted for. A convenient way to represent the uncertainty in the model is to use stochastic disturbance signals. The state estimator uses this information and unmeasured parameters reconstruction is achieved in the optimal fashion. The model however represents the nominal system, even if stochastic disturbances are included. Faults occurring in the system may cause a significant change in the system properties. If not accounted for, faults may cause estimation errors and potentially control system instability. This brings out the problem of fault detection, isolation and identification. For most systems this task is extremely important. The efficient fault detection is necessary to guarantee safety of people, environment and the system itself. The failure if not detected may cause risks to life, environmentally harmful operation and lead to far more serious failures with their financial consequences.

1.2 System modelling

The modelling is one of the most important tasks that have to be carried out in order to use modern control techniques. There are a number of reasons why models are so important for the control system design [1]. These include:

- The requirement for better process or product performance,
- The increasing complexity of advanced technological systems,
- The growing need for competitive advantage, e.g., efficiency, economy,
- The phenomenal increase in available computer power and decrease in cost,

The automotive industry is a good example of technological advances. The performance which is determined by the power output, emissions, safety and comfort has greatly increased in the last 25 years. At the same time the complexity of the entire package of engine, powertrain, chassis and other features has increased at least 10-fold [1]. Control algorithm design for a complex system, without the global knowledge of its behaviour, would be extremely difficult, if not impossible. With the advent of advanced computing facilities and with a plant model, the analysis, design and testing may be carried out without the necessity for using the hardware. Theoretically complex issues like stability, robustness and immunity to faults may be assessed in simulation. This however relies strongly on the model of the system. For complex objects modelling is a difficult issue, especially if the system is non-linear.

Modelling is often regarded as a combination of art and science. Models may be classified based on their application [2]. Intuitive models which are located at the most heuristic level exist only in the engineer's mind as his/her personal conception of the system. These models have no physical or mathematical representation. Next, models intended for investigation of fundamental properties of the system may be distinguished. Simulation models are aimed at investigating basic phenomenological features of the system or process. This class of models includes two types:

- Computer simulation models based on mathematical formulation of plant's behaviour. Such simulations may be extremely complex and can include all possible system details,
- Scale models that are small-scale replicas of the process under study. Such models allow detailed study of the variety of design and operational conditions in controlled environment.

Finally, the class of dynamic models that are intended for control law design is outlined. These models will be analyzed in this thesis and in principle can be less complex than computer simulation models [1], [2]:

- Dynamic analysis ("white box") models obtained from the analysis of physical systems at the fundamental level. These use scientific principles like Newton's law, Kirchhov's laws, laws of thermodynamics, reaction kinetics, to derive an analytical model. The feasibility of building such a model varies a great deal from one discipline to another. The task may be easier for electro-mechanical systems, but very often difficult in some biological application areas,
- Dynamic identification models ("black box") inferred from the observed behaviour of the physical system (data-driven). For these models the structure must be defined first and the model parameters identified. A large number of methods and software packages exist for such model identification. These packages identify both: the model structure (e.g. order) and parameters based on time-series data. These include frequency response (nonparametric) modelling, regression, least squares techniques, maximum likelihood, instrumental variables [1], [3], [4].

A "grey-box" modelling is an attempt to bridge the "black box" (purely data based) modelling with the "white box" (purely theory based) modelling. It is intended to combine the best of two worlds: knowledge-based modelling and the data-driven model estimation [5]. The underlying dynamics of a practical engineering system and some of its physical parameters are usually known a-priori

[6]. The "white-box" mathematical model equations are derived in order to describe a process, based on a physical (or chemical, biological, etc.) analysis. However, some of the, mostly minor, dynamics cannot be modelled, due to the system nonlinearity, complexity and constraints on physical ability to measure. The "black-box" modelling is used to introduce parameters of the model that are estimated solely from measurements made on the process. The "grey-box" modelling technique is very valuable whenever a knowledge-based model exists, but is not fully satisfactory and cannot be improved by further analysis (or can only be improved at a very large computational cost). The "grey-box" model identification is a systematic approach to the problem of making models for industrial processes by combining physical modelling with experimental data. The "grey-box" model should explicitly use the knowledge such as the dynamic structure derived from physical laws. In the "grey-box" model, the system structure is not replaced by an artificial structure approximation as it takes place in the "black box" modelling [6]. As a consequence the physical meaning of model parameters may be retained, which is beneficial during identification. The "white-box" models are physically based and may be constructed based on physical constants and other measurable parameters (e.g. volume, length, mass). For the "grey-box" and "black-box" models values of parameters need to be identified. The estimation methods outlined later in this thesis may be used for that purpose.

In recent years the significant popularity of neural and fuzzy-neural black-box models has emerged. These methods are claimed to provide a convenient solution for vast array of non-linear modelling problems. Neural networks are able to approximate any continuous mapping to a sufficient accuracy if they have resources to do so [7], [8]. These resources are simply number of neurons, layers and structure of the network. An important part of the modelling with neural networks is the so called training of the network which assigns values of parameters. Learning procedure must be carefully designed. The neural network as a data driven model relies heavily upon a training set. This is especially important for non-linear systems. The training data should cover all non-linearities and excite all dynamics. Failure to do so may lead to a significant model mismatch if a neural network model is used in

the region where a sufficient amount of data was not supplied. Since the neural network model parameters have no physical significance it is almost impossible to predict if the model is likely to fail in operating regions where insufficient data was supplied during training.

The models based on fuzzy logic allow incorporating rules based on linguistics [9] in the model. Consequently expert knowledge about the object may be used during modelling. This makes the technique useful when intuitive engineering knowledge about the plant behaviour is available from an experienced operator. This knowledge may be combined with measurements collected from the sensors. The main advantage claimed of models based on fuzzy logic is the direct interpretation that can be obtained from their rule structure. Each rule in the model represents an operation zone of the process [9]. So far, methods for systematic structure identification are still lacking, and this is a subject of current research.

1.3 Automotive powertrain application

The identification, estimation, control and fault detection methods presented in this thesis are applied to the automotive powertrain control problem. The automotive powertrain control problem is characterized by two main objectives: torque tracking and emissions control. The spark advance is not considered here and is assumed to be controlled separately as a function of current engine state by its dedicated controller. Torque tracking is achieved by varying the mass of combustible gas which enters engine cylinders. Composition of this mixture determines both: the amount of torque that is generated by the engine and the emissions. The minimization of engine emissions is achieved by maintaining the air-fuel ratio at stoichiometry, which allows a complete combustion. The air-fuel ratio is often scaled to unity which is then denoted by lambda. The lambda is then air-fuel ratio scaled by the stoichiometric ratio specific for a particular fuel: $\lambda = \frac{\text{Air mass}}{\text{Fuel mass}} \frac{1}{AF_{Stoich.}}$. For a

stoichiometric mixture the efficiency of catalytic converter is optimal and the best conversion is achieved for the combined pollutant fraction (HCs, NOx and CO).

The engine block diagram is shown in Figure 1-1. The following measurements are often available in the system: P_a, T_a – ambient pressure and temperature respectively; $\dot{m}_{at}$ - measured throttle flow rate (often referred to as MAF); TPS - indicated throttle position; P_{im}, T_{im} - intake manifold pressure (often referred to as MAP) and temperature respectively; P_{em}, T_{em} - exhaust manifold pressure and temperature respectively; λ_0, λ - lambda measurements; N - engine speed [rpm].

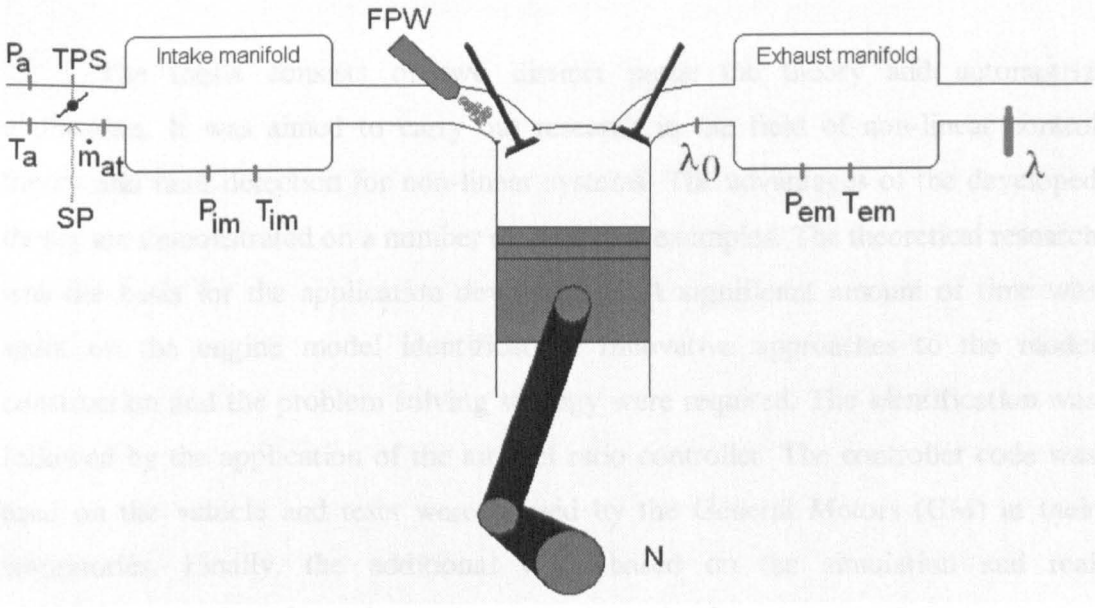

Figure 1-1: Engine block diagram

The manipulated variables used for engine control are the injector command: fuel pulse width (FPW) and the throttle position command (SP) used in the multivariable control case. The fuel pulse width is the injector opening time applied as an electrical pulse signal to the injector solenoid. The solenoid opens the fuel flow valve and the amount of injected fuel is proportional to the pulse width. The throttle position setpoint changes alter the air flow into the manifold. In that way the amount of air that enters the cylinder is manipulated. With the advent of electronically

actuated throttle it is possible to decouple the driver (or the accelerator pedal) from the engine (or the position of the throttle plate - throttle angle). This allows compensation of the intake manifold filling dynamics and faster torque response is achieved. In a conventional control strategy, with the driver controlling the throttle position directly, faster change of the throttle position meant significant excursions of the air-fuel ratio from stoichiometry. The only way to prevent this effect was to inject fuel on open valves resulting in increased hydrocarbons (HCs) emissions. It will be demonstrated that a full multivariable control strategy is capable of improving both: emissions and torque response.

1.4 Overview of the Thesis

The thesis consists of two distinct parts: the theory and automotive application. It was aimed to carry out research in the field of non-linear control theory and fault detection for non-linear systems. The advantages of the developed theory are demonstrated on a number of academic examples. The theoretical research was the basis for the application development. A significant amount of time was spent on the engine model identification. Innovative approaches to the model construction and the problem solving strategy were required. The identification was followed by the application of the air-fuel ratio controller. The controller code was used on the vehicle and tests were carried by the General Motors (GM) at their laboratories. Finally, the additional work based on the simulation and real measurements taken from datasets was performed. This was aimed at developing multivariable control strategies for the combustion engine. Next, the effort was concentrated on the simulation analysis of the application of non-linear estimation and fault detection methods to the engine control/diagnosis.

In Chapter 2 the basic estimation and parameter identification theory is briefly presented. This is followed by the formulation of the non-linear estimation theory, namely the extended and state-dependent Kalman filters. The fault detection theory is presented next. The methods for sensor and components fault detection are elaborated. The fault detection filter presented there follows developments presented

in [10], which are extended in this thesis with the improved state estimation when a fault is present in the system. The filter provides detection, isolation and identification capabilities for process faults. For the sensor fault detection and isolation the discrete time state-dependent Kalman filter the open-loop residual generation techniques are employed. The fault isolation based on the Generalized Observer Scheme. This enables simultaneous isolation of the process and sensor faults.

In Chapter 3 developments in the non-linear optimal and predictive control algorithms are presented. The development of algorithms is based upon state-dependent state-space models. Two groups of algorithms are presented. The extensions of the well-known linear-quadratic control algorithm are presented. The discrete version of the state-dependent Riccati equation (SDRE) method is first elaborated. This is followed by the suboptimal predictive extension of the SDRE technique, which offers a simple solution of the non-linear control problem [11]. The analysis of the optimality of the solution followed next and the method that recovered the optimality was presented [12]. In the next section the development of the non-linear predictive control algorithms is presented. The application of the non-linear predictive control algorithm presented earlier in [13] [14] is then introduced [15]. The method providing the optimal solution for non-linear predictive control algorithm follows next [16]. Finally, the non-linear Generalized Predictive Control algorithm is presented that provides the trajectory tracking capabilities.

The second part of the thesis includes the implementation of the theoretical developments presented in Chapter 2 and Chapter 3 to Powertrain problems. The powertrain control developments presented in this thesis will start with the model identification. The spark ignition combustion engine model identification is elaborated in Chapter 4. The accurate modelling of the engine is of great importance for the control system design. The identification data used in this thesis is collected during a driving cycle test. The usual identification procedure requires the data to be collected at a series of steady state operating points by performing small perturbations. That involves creating a dense grid within the full operating range and the identification of a large number of local linear models scheduled with the operating point. The labour intensity of that task is enormous and the development

time very long. The driving cycle data used in this thesis for the identification was sampled at a frequency determined by the current engine speed (event based sampling). The sampling event was triggered every 90 degrees of the crankshaft position. There are clear advantages and disadvantages of such a sampling technique. Advantages are associated with the fact that some components of system's time delays become constant if event based sampling is used. Intake manifold dynamics depend on the engine speed and consequently the sampling rate variable with the speed provides the best discretization of the model for control purpose. Also, since the control action is required every event (e.g. 90 degrees for 8 cylinders engine), the sampling based on that frequency gives a good performance. Controllers with fixed sampling rate generate a control signal update at the same rate regardless the engine speed. It has a very important advantage for the hardware implementation. For an eight cylinder engine, in steady state at 600RPM, injection takes place every 25ms. However, for 6000RPM this time is reduced to only 2.5ms. Event based controllers have to cope with the fastest sampling rate and consequently the processing power demand is increased. Now, the important question may be asked: Why would the identification use the same type of data? The same remarks on the sampling properties apply here. There is also a disadvantage associated with the identification based on event sampled data. Some sensor dynamics are too fast to be identified at the slowest sampling rates. While event based sampling is suitable for the intake manifold dynamics it may not be possible to identify fast sensor dynamics at the minimum speed required for reliable identification. The mass flow sensor is a good example and with the typical time constant of 5ms the measurements collected every 25ms are by far not sufficient. Probably the most important advantage of the presented identification strategy based on driving cycle data is the on-line model identification/adaptation. If it is possible to identify the model from that type of data off-line it will also be possible to do it on-line. The increased processing power demand to match the shortest possible sampling period (i.e. 2.5ms) means that when sampling is much slower (i.e. 25ms) there will be at least 22.5ms of processor time available to run the model adaptation task. The task could run in the background and use only any excess of processing power and would aim to accommodate for the slow engine parameter changes due to wear and tear. The analysis of the sampling

issues is presented in [17]. The air-fuel ratio model identification task is split in this thesis in two distinct parts:

- Intake manifold model identification which consists of the throttle, the intake manifold and the cylinder air charge model identification. This is performed without the necessity of considering the fuelling path, from only upstream measurements (e.g. intake manifold pressure, temperature).
- Fuel delivery dynamic model identification consisting of the wall fuel dynamics, injector parameters identification and the varying exhaust gas transport time delay modelling (based on the lambda measurements).

Additionally an approximate torque modelling will be carried out with the assumption that air-fuel ratio is controlled close to its stoichiometric value. With this assumption the net torque produced by the engine is proportional to the cylinder air charge.

To justify the split of the model identification into two distinct parts the following should be analyzed. The air-fuel ratio output of the system is simply given as $AF = \frac{\text{Air charge}}{\text{Fuel charge}}$. If only the ratio is given as a measurement some reference measurement is necessary since the ratio itself does not contain sufficient information. There are two options for a choice of the reference measurement. These are either air or fuel charge. The air charge may be obtained using the upstream measurements (e.g. pressure, throttle flow rate, throttle position) without the need for using air-fuel ratio measurements (obtained from the lambda sensor). The fuel charge, because of the complex character of fuel delivery and because no other measurements are present in the system, can not be determined without using the air-fuel ratio measurement. Therefore, the intake manifold model identification (air charge model) is performed first and is followed by the fuel path identification based on the air-fuel ratio measurement.

With the accurate model identified the controller design may begin. The accurate air fuel ratio control helps to keep pollution caused by the operation of the engine at minimum level. Due to the engine operating cycle and a significant transport delay in the exhaust, the measurement of air-fuel ratio is available only long after the control decision has to be made. This constrains the maximum performance (or bandwidth) which may be achieved by the feedback control. The performance is maintained by the feedforward controller. The feedforward controller is a model-based device which uses the disturbance measurement to counteract its effects. The disturbance for the air-fuel ratio control is a change in the throttle position, which causes changes in the cylinder air charge. The fuel charge should match the air charge to achieve stoichiometric gas mixture in cylinders. The feedforward controller should control the mass of injected fuel to ensure that the gas which gets in the cylinder has the required composition. The possible inaccuracies of the feedforward control are subject to the feedback correction. With an accurate model, the feedforward control action becomes better and consequently the control system relies less on the feedback controller. This is very important, since the performance of any feedback loop is limited by the significant time delay in the air-fuel ratio measurement.

The study and implementation of the model based engine predictive control is presented in Chapter 5. The engine tests with the model-based feedforward controller were carried out with the main intention of the data driven model assessment. The performance of the air-fuel ratio feedforward controller is limited by the time delay in the fuel delivery path and the uncertainty associated with unpredictable driver's behaviour. The time delay in the fuel delivery path implies that the feedforward controller must have a built-in predictor. The accuracy of the predictor depends on the model precision. The predictive feedforward fuel control algorithm was developed and tested in the vehicle. Further improvements in the air-fuel ratio control are achieved by multi-input multi-output (MIMO) nonlinear control strategies presented in this thesis. Models of exhaust gas air-fuel ratio and generated torque are used for the design of the control system. Improvements in the air fuel ratio regulation may be achieved by decoupling the accelerator pedal and the throttle

opening through the introduction of a simple delay in the application of throttle command. Additional improvements in transient torque control are achieved through the magnitude decoupling of the throttle movement and the accelerator pedal position as well. The compensation of cylinder air charge dynamics using simple conventional control methods is used to speed-up the torque response. Finally, the focus is moved on to the multivariable predictive air-fuel ratio and torque control methods. This provides a suitable solution for the torque tracking problem and the air-fuel ratio control. The control algorithm design procedures are described in the Chapter 5.

In Chapter 6 a very important aspect of the control system that requires careful consideration is finally analyzed. The noise filtering and fault detection for the intake manifold is the last major subject of this thesis. The results are based on the simulation of the engine model identified earlier. The faults and process/measurement noise are fed into the simulated system. It is demonstrated in simulation how the model-based non-linear estimation methods substantially improve the performance of the system in the presence of noise as well as model uncertainties. The fault detection of the intake manifold system and sensor faults analysis follows next. The proposed fault detection algorithm is based upon theoretical developments of Chapter 2. The efficient operation of the fault detection algorithm is demonstrated in simulation. The control system reconfiguration provides superior performance in a presence of faults.

1.5 Contributions

The main contributions in this thesis fall in two groups: the theory and application developments. The following work is presented in the thesis:

- Developments of the non-linear fault detection filter, including the state estimation method for the system being subject to process fault

- Analysis of the sensor fault residual generation based on open loop and state-dependent estimation for non-linear systems
- Development of the sub-optimal predictive state-dependent Riccati equation control method
- Optimality recovery methods developed for the predictive control algorithms based on the state-dependent models
- Engine identification executed in two stages: the intake manifold identification based on the upstream measurements followed by the time delays removal and the fuel delivery path identification
- The lookup-table construction method based on clusters of data and the extrapolation methodology
- Intake manifold constant parameter identification based on the engine cycle re-sampled data offering improved convergence
- Modified intake manifold heat transfer model formulation
- Exhaust manifold time delay model structure formulation and the identification method
- Development of the improved air-fuel ratio control based on the pedal and throttle position decoupling ranging from a simple variable time delay introduction to the full dynamic compensation of the intake manifold filling dynamics
- Application of the non-linear predictive control to the multivariable torque and air-fuel ratio control with the innovative structure
- The application and the robustness analysis of the intake manifold estimation with the state-dependent and extended Kalman filter
- The fault detection and isolation for the intake manifold with the control system reconfiguration

1.6 List of publications

- *Giovanini, L and Dutka, A., (2003), Fault Isolation Filter for Nonlinear Systems, IFAC Safeprocess 2003, Washington, USA*
- *Dutka, A. S., Grimble, M. J., (2004). State-Dependent Riccati Equation Control with Predicted Trajectory, Proceedings of the American Control Conference, Boston, USA*
- *Dutka, A. S., A. W. Ordys, Grimble, M. J., (2005). Optimized discrete-time State Dependent Riccati Equation regulator, Proceedings of the American Control Conference, Portland, USA.*
- *Dutka, A. S., Ordys, A. W., Grimble, M. J. (2003). Non-linear Predictive Control of 2 dof helicopter model, IEEE CDC conference proceedings, Maui, Hawaii, USA*
- *Dutka, A. S., Ordys, A. W. (2004). The Optimal Nonlinear Generalized Predictive Control By The Time-Varying Approximation, Proceedings of IEEE International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland*
- *Youssef A., Grimble, M.J., Ordys, A.W., Dutka A.S., Anderson, D., (2003). Robust Nonlinear Predictive Flight Control, European Control Conference, Cambridge, UK.*

Chapter 2

Non-linear Estimation and Fault detection

The estimation and the fault detection are closely related tasks. The model identification employs estimation techniques for parameters estimation. Once the model is identified estimation techniques may be used for filtering and reconstruction of unmeasured states of the dynamic system. The model based estimation techniques use data fusion to remove the noise that is ever present in real systems. This allows using cheaper sensors and less accurate models while still achieving acceptable performance. Real systems are subject to faults occurring in sensors or system elements. The fault detection techniques are based on the same as the estimation data fusion principle. In the context of fault detection, the data fusion is referred to as analytical redundancy. This allows detecting faults subject to the certain minimum number of measurements being available. The analytical redundancy removes the necessity of having multiple sensors installed in the system and is known as a hardware redundancy. In this chapter the theoretical foundations of state and parameter estimation are presented. The theoretical development of fault detection algorithm for non-linear system is carried out next. The developments are based on the application of state dependent models to the estimation and the fault detection.

2.1 Linear Estimation methods

Linear estimation methods are well researched and only a short overview of available techniques will be presented as a reference. For the parameter identification the most commonly used is the Least Squares method. The least squares method is a deterministic approach to estimation. It means that the measurements and quantities being estimated are not modelled as random variables or random signals [18]. It minimizes the squared sum of model output estimation errors. In that sense the minimized function is an indication of the parameter estimation error and the Least Squares parameters estimate is obtained.

Consider the discrete time signal y_n given by the following equation [4]:

$$y_n = \varphi_n^T \Theta + \varepsilon_n \tag{2.1}$$

where

$\Theta = \left[\Theta_1, \Theta_2, ..., \Theta_q\right]^T$ is a vector of q unknown parameters

$\varphi(n) = \left[\varphi_{1,n}, \varphi_{2,n}, ..., \varphi_{q,n}\right]^T$ is the vector of known functions of n.

It is aimed to find an estimate $\hat{\Theta}$ of Θ at time n using the measurements $y_1, y_2, ..., y_N$ and $\varphi_1, \varphi_2, ..., \varphi_N$. Given these measurements a system of linear equations is obtained, which in a matrix form is given by the following equation:

$$Y = \Phi\Theta + \mathrm{E} \tag{2.2}$$

where

$$Y = \left[y_1, y_2, ..., y_N\right]^T, \quad \Phi = \left[\varphi_1, \varphi_2, ..., \varphi_N\right]^T, \quad E = \left[\varepsilon_1, \varepsilon_2, ..., \varepsilon_N\right]^T$$

Assuming that $E = 0$ it is possible to find Θ if $N=q$ and Φ is non-singular. In practice the noise and disturbances are the reason to use more measurements: $N>q$.

An exact solution will then, in general, not exist [4]. The least squares estimate of Θ is defined as the vector $\hat{\Theta}$ that minimizes the following function:

$$V(\Theta)=\frac{1}{2}E^TE \tag{2.3}$$

Lemma 2.1

The function $V(\Theta)$ (2.3) has a unique minimum point if the matrix $\Phi^T\Phi$ is positive definite and is given by:

$$\hat{\Theta}=\left(\Phi^T\Phi\right)^{-1}\Phi Y \tag{2.4}$$

Proof: See [4], [18].

The estimate (2.4) is unbiased if ε_n is a zero mean white noise. Least squares method may be used for the state estimation for state-space systems. The following system may be considered for that purpose:

$$\begin{aligned} x_{n+1} &= Ax_n \\ y_n &= Cx_n + v_n \end{aligned} \tag{2.5}$$

where

$x_n \in \Re^p$ is a state vector, $u_n \in \Re^q$ is a control vector, $v_n \in \Re^r$ is the zero mean white noise vector

Note that only the noise in the measurement equation is included in the system model (2.5). The derivation of the least squares estimate may be found in [18]. The full derivation of the recursive least squares method may be found in [4], [18]. There are similarities between the recursive least squares method and the Kalman filter for which the equations will be presented in sequel.

Consider the system given by the following state-space model [19]:

$$x_{n+1} = A_n x_n + B_n u_n + G_n w_n$$
$$y_n = C_n x_n + v_n \tag{2.6}$$

where

$x_n \in \Re^p$ is a state vector, $u_n \in \Re^q$ is a control vector, $y_n \in \Re^r$ is an output vector, w_n is a zero mean white process noise, $\mathrm{cov}(w_n) = Q_n$, v_n is a zero mean white measurement noise, $\mathrm{cov}(v_n) = R_n$.

It is assumed that Q_n is semi positive and R_n is positive definite and symmetric. The initial condition vector is assumed to be uncorrelated with the noise processes, the expected value $m_0 = E\{x_0\}$ and the initial covariance $P_0 = E\left\{(x_0 - m_0)(x_0 - m_0)^T\right\}$. The optimal discrete linear filter minimizes the following estimation error criterion:

$$J_e = trace\ E\left\{x_n - \hat{x}_{n|n}\right\} \tag{2.7}$$

The solution of this optimal estimation problem was presented by Kalman [20]. The full derivation of Kalman filter equations is given in [18], [21]. The final result given by the recursive equations is presented below.

$$\begin{aligned}
P_{n|n-1} &= A_{n-1} P_{n-1|n-1} A_{n-1}^T + G_{n-1} Q_{n-1} G_{n-1}^T \\
K_n &= P_{n|n-1} C_n^T \left(C_n P_{n|n-1} C_n^T + R_n\right)^{-1} \\
P_{n|n} &= P_{n|n-1} - K_n C_n P_{n|n-1} \\
\hat{x}_{n|n-1} &= A_{n-1} \hat{x}_{n-1|n-1} + B_{n-1} u_{n-1} \\
\hat{x}_{n|n} &= \hat{x}_{n|n-1} + K_n \left(y_n - C_n \hat{x}_{n|n-1}\right)
\end{aligned} \tag{2.8}$$

For a system, whose model is time-invariant and the noise sources are stationary the Kalman filter, in the steady state reduces to a time invariant filter [19]. The solution for the steady state Kalman filter is obtained by solving the algebraic Riccati equation. The algebraic Riccati equation is derived by substituting for $P_{n|n-1} = P_\infty$ in

(2.8). The following equation has a positive definite and symmetric matrix P_∞ solution that results in the steady state gain K.

$$\begin{aligned} P_\infty &= A\left(P_\infty - P_\infty C^T\left(CP_\infty C^T + R\right)^{-1} CP_\infty\right)A^T + GQG^T \\ K &= P_\infty C^T\left(CP_\infty C^T + R\right)^{-1} \end{aligned} \tag{2.9}$$

The constant gain K is used for the correction of the state estimate in the same way as the time-varying gain K_n was employed in equation (2.8).

2.2 Estimation extensions for Non-Linear systems

The Kalman filter derived originally for linear systems was extended to systems described by the non-linear state-space representation:

$$\begin{aligned} x_{n+1} &= f_n(x_n, u_n) + G_n(x_n)w_n \\ y_n &= h_n(x_n) + v_n \end{aligned} \tag{2.10}$$

where

$f_n(x_n, u_n)$ and $h_n(x_n, u_n)$ are vector-valued functions, $G_n(x_n)$ is a matrix-valued function.

It is required that partial derivatives of $f_n(x_n, u_n)$ and $h_n(x_n, u_n)$ with respect to all components of x_n are continuous. Statistical properties of signals w_n and v_n are identical to assumptions made for the linear filter. Using the linear Taylor series approximation of the system (2.10) the linear approximation at current estimated state is derived. A detailed analysis of the method is presented in [18], [21] and [22].

$$
\begin{aligned}
P_{n|n-1} &= \left[\frac{\partial f_{n-1}}{\partial x_{n-1}}\left(\hat{x}_{n-1|n-1}, u_{n-1}\right)\right] P_{n-1|n-1} \left[\frac{\partial f_{n-1}}{\partial x_{n-1}}\left(\hat{x}_{n-1|n-1}, u_{n-1}\right)\right]^T \\
&\quad + G_{n-1}\left(\hat{x}_{n-1|n-1}\right) Q_{n-1} G_{n-1}\left(\hat{x}_{n-1|n-1}\right)^T \\
\hat{x}_{n|n-1} &= f_n\left(\hat{x}_{n-1|n-1}, u_{n-1|n-1}\right) \\
K_n &= P_{n|n-1}\left[\frac{\partial h_n}{\partial x_n}\left(\hat{x}_{n|n-1}\right)\right]^T \left(\left[\frac{\partial h_n}{\partial x_n}\left(\hat{x}_{n|n-1}\right)\right] P_{n|n-1} \left[\frac{\partial h_n}{\partial x_n}\left(\hat{x}_{n|n-1}\right)\right]^T + R_n\right)^{-1} \\
P_{n|n} &= P_{n|n-1} - K_n\left[\frac{\partial h_n}{\partial x_n}\left(\hat{x}_{n|n-1}\right)\right] P_{n|n-1} \\
\hat{x}_{n|n} &= \hat{x}_{n|n-1} + K_n\left(y_n - h_n\left(\hat{x}_{n|n-1}\right)\right)
\end{aligned}
\tag{2.11}
$$

The main difference between the linear Kalman filter and its non-linear extension is the non-linear model (2.10) used in the state $\hat{x}_{n|n-1}$ update in (2.11). Also, the covariance P and gain K are propagated using the Jacobian linearization of the non-linear model at the estimated state $\hat{x}_{n-1|n-1}$ for $\frac{\partial f_{n-1}}{\partial x_{n-1}}$ and $\hat{x}_{n|n-1}$ for $\frac{\partial h_n}{\partial x_n}$. For non-linear systems due to approximations made during linearization the state estimates convergence is not guaranteed. Convergence problem may occur especially, if the initial state value is far from the actual. The linearized model matrices are functions of the state about which the linearization was carried out. The model mismatch is a consequence of the state estimation error. This may lead to improper state update and consequently to the divergence.

An alternative for the extended Kalman filter is presented in sequel. In this thesis the emphasis is put on state-dependent models. The estimation problem using state-dependent models is now analyzed. A class of non-linear models given by the following non-linear discrete time state-space equations is considered [23]:

$$
\begin{aligned}
x_{n+1} &= f(x_n) + B(x_n) u_n + w_n \\
y_n &= h(x_n) + v_n
\end{aligned}
\tag{2.12}
$$

where

$x_n \in \Re^p$ is a state vector, $u_n \in \Re^q$ is a control vector, $y_n \in \Re^r$ is an output vector.

The model (2.12) is re-arranged into the state-dependent coefficient form [23]:

$$\begin{aligned} x_{n+1} &= A_n x_n + B_n u_n + w_n \\ y_n &= C_n x_n + v_n \end{aligned} \tag{2.13}$$

where

$A_n = A(x_n)$; $B_n = B(x_n)$; $C_n = C(x_n)$; $F_n = F(x_n)$ are state-dependent matrices.

Note that the model (2.1) is identical to (2.12) if $A(x_n)x_n = f(x_n)$ and $C(x_n)x_n = h(x_n)$. Additionally it is assumed that $\underset{x_n \in \Omega_x}{\forall} \{C_n, A_n\}$ is point-wise observable in the operating region Ω_x [23].

The way how the system is parameterized provides an additional degree of freedom for the design. As a guideline the following strategy might be adopted. The vector-valued function $f(x_n)$ is often given by the following non-linear structure:

$$f(x_n) = \begin{bmatrix} f_{1,1}(x_{1,n},...,x_{p,n}) + ... + f_{1,p}(x_{1,n},...,x_{p,n}) \\ \vdots \\ f_{p,1}(x_{1,n},...,x_{p,n}) + ... + f_{p,p}(x_{1,n},...,x_{p,n}) \end{bmatrix} \tag{2.14}$$

It is important to assess which state contributes the most in each of elements in each row of the vector in equation (2.14). It is assumed that the row components in equation (2.14) (e.g. $f_{1,1}(x_{1,n},...,x_{p,n}) + ... + f_{1,p}(x_{1,n},...,x_{p,n})$) are ordered according to the most significant dependence upon the corresponding state. The natural parameterization will be carried out as follows:

$$A(x_n)=\begin{bmatrix} \dfrac{f_{1,1}(x_{1,n},\ldots,x_{p,n})}{x_{1,n}} & \cdots & \dfrac{f_{1,p}(x_{1,n},\ldots,x_{p,n})}{x_{p,n}} \\ \vdots & \cdots & \vdots \\ \dfrac{f_{p,1}(x_{1,n},\ldots,x_{p,n})}{x_{1,n}} & \cdots & \dfrac{f_{p,p}(x_{1,n},\ldots,x_{p,n})}{x_{p,n}} \end{bmatrix} \tag{2.15}$$

Additional analysis of the state-dependent parameterization methods is given in [24]. The process noise w_n and measurements noise v_n are independent white Gaussian signals with $\mathrm{cov}\{w_n\}=Q_n$ and $\mathrm{cov}\{v_n\}=R_n$. Q_n and R_n are diagonal semi-positive and positive definite matrices respectively.

The system (2.12) is non-linear and the non-linear state dependent Kalman filter is employed. The state-dependent Kalman filter was originally presented by Mracek et al. [25]. This was an extension of the state-dependent Riccati equation control method. Using the duality of control and estimation problems the filter equations follow. The discrete version of the state-dependent Kalman filter is given by the following equations:

$$\begin{aligned} \hat{x}_{n+1} &= \hat{A}_n \cdot \left(\hat{x}_n + K_n\left(y_n - \hat{C}_n\hat{x}_n\right)\right) + \hat{B}_n u_n \\ \hat{y}_n &= \hat{C}_n\hat{x}_n \end{aligned} \tag{2.16}$$

The state dependent model matrices are denotes as $\hat{A}_n = A(\hat{x}_n)$, $\hat{B}_n = B(\hat{x}_n)$, $\hat{C}_n = C(\hat{x}_n)$. The filter gain K_n is given by the following equation:

$$K_n = P_n\hat{C}_n^T\left(\hat{C}_nP_n\hat{C}_n^T + R_n\right)^{-1} \tag{2.17}$$

The P_n is a solution of the discrete algebraic Riccati equation:

$$P_n = \hat{A}_n\left[P_n - P_n\hat{C}_n^T\left(R+\hat{C}_nP_n\hat{C}_n^T\right)^{-1}\hat{C}_nP_n\right]\hat{A}_n^T + Q \tag{2.18}$$

The solution of the Riccati equation (2.18) minimizes a frozen system's (2.1) expected squared state estimation error in the same way as for the linear steady state Kalman filter given by equations (2.9) [18], [19], [20], [21], [22].

It should be noticed that the system matrices $\hat{A}_n = A(\hat{x}_n)$, $\hat{B}_n = B(\hat{x}_n)$, $\hat{C}_n = C(\hat{x}_n)$ are based on state estimates. An estimation bias may result in the model mismatch. This may cause state estimates divergence. The estimates convergence analysis for the general non-linear system representation is very difficult. This property of the filter should be analyzed for particular application. The type of non-linearity is an important feature that should be analyzed. The solution of the Riccati equation results in the local convergence of estimates. This however is not sufficient to guarantee global properties. In practice the convergence analysis may have to be limited to simulation tests.

2.3 Parameter Identification with the Extended Kalman Filter

The extended Kalman filter may be applied for the parameter identification [18], [21]. Suppose that the system is given by the following state-space model with the following structure:

$$\begin{aligned} x_{n+1} &= f_n(\Theta, x_n, u_n) + G_n(\Theta, x_n)\, w_n \\ y_n &= h_n(\Theta, x_n) + v_n \end{aligned} \tag{2.19}$$

The usual assumptions about the noise statistics must be made as for the extended Kalman filter. To identify constant parameters vector Θ it is augmented with the system state x_n. Consequently the following non-linear system is subject of estimation:

$$\begin{bmatrix} x_{n+1} \\ \Theta_{n+1} \end{bmatrix} = \begin{bmatrix} f_n(\Theta_n, x_n, u_n) \\ \Theta_n \end{bmatrix} + \begin{bmatrix} G_n(\Theta, x_n) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_n \\ \zeta_n \end{bmatrix}$$
$$y_n = h_n(\Theta, x_n) + v_n \tag{2.20}$$

For an identification the parameter vector Θ_n is often treated as a random constant vector ($\Theta_{n+1} = \Theta_n + \zeta_n$) and ζ_n is a zero mean Gaussian white noise sequence uncorrelated with w_n [21]. The covariance of the signal ζ_n is used to achieve the adaptive capabilities of the filter. For the identification of the constant parameters it is sufficient to initialize the covariance P_0. The noise ζ_n may be assumed to be zero.

Defining the augmented state vector as $\chi_n = \begin{bmatrix} x_n \\ \Theta_n \end{bmatrix}$ and process noise vector as $\xi_n = \begin{bmatrix} w_n \\ \zeta_n \end{bmatrix}$ the model (2.23) may be written in the following form:

$$\chi_{n+1} = \tilde{f}_n(\chi_n, u_n) + \tilde{G}_n(\chi_n)\xi_n$$
$$y_n = \tilde{h}_n(\chi_n) + v_n \tag{2.21}$$

The model (2.21) has the identical structure to (2.10) and the extended Kalman filter algorithm equations (2.11) may be applied directly. In order to apply the extended Kalman filtering to the system (2.21) the initial state χ_0 that consists of the system states as well as the initial guess of the parameter vector $\hat{\Theta}_0$ must be supplied to the algorithm. For the system for which some physical insight exists – namely for grey-box models – initial guesses for states and for parameters are usually available. This makes the identification procedure relatively easy as opposed to black-box identification.

The system process and measurement noise covariances must be known for the extended Kalman filter identification. The measurement noise covariance will usually be supplied by the sensor manufacturer. The process nose w_n covariance reflects the confidence in the model structure accuracy. The initial state error

covariance P_0 is often assumed as high as possible with the requirement that the state estimates convergence is maintained. As more accurate estimates of parameters are obtained the covariance P_0 is reduced. The covariance adjustment is often a trial and error process and requires running a number of estimation experiments.

2.4 Fault detection, isolation and identification

The fault detection improves the safety and may help to avoid major breakdowns. The fault detection consists of three tasks: fault detection, fault isolation and fault identification [26]. Aforementioned three tasks are subsequent actions and in some applications not all are required. The fault detection is the most crucial task and in most practical applications must be carried out. The isolation comes as the second, but is not less important. The isolation aims to pinpoint the fault and may be used for the system reconfiguration. In that sense the system operation may continue even in a presence of the fault. The fault identification attracts far less attention. The task requires numerical estimates of the extent of a fault. For some applications the fault identification may be needed if controller re-configuration requires a numerical estimate of the fault.

The fault detection methods may fall in two major categories: model-free methods and model based methods. The model-free methods include [26]:

- Sensor outputs limit checking for sensor fault detection (e.g. using the normal range for sensor output)
- Installation of special sensors that monitor plant parameters (e.g. pressure, vibrations)
- Installation of multiple sensors and comparing sensor outputs (voting system)
- Frequency analysis that gives indication of fault occurrence if the signal spectrum differs from its usual signature
- Expert system approach uses logical rules based on symptoms obtained by the detection hardware and software

Several survey papers on fault detection theory based on analytical redundancy were written by Frank [27], [28], Gertler [26], [29], [30] and Patton and Chen [31]. Books by Chen and Patton [32], Gertler [33] also provide a wide scope of the fault detection methods. The model based methods are built around an analytical redundancy. The analytical redundancy combines measurements collected at different points of the system with the model. The parameter of interest is reconstructed and its measurement is compared with the analytically obtained value based on other measurements. To facilitate fault detection diagnosis signals called residuals are generated. The residuals generation employs the model and inputs/outputs of the system. The residual signals indicate that the fault has occurred. In practice residuals, even if faults are not present, will not be zero. The measurements in real systems are always noisy and models have a finite accuracy. In this case important information is the signal to noise ratio. If the noise level is relatively low the fault detection algorithm will be highly sensitive to the system faults, which results in highly reliable fault detection method. For systems with higher signal to noise ratios the fault detection algorithm must take this fact into account and consequently only faults of higher magnitude may be detected. The noise and uncertainty, if not considered carefully, could cause false alarms. To avoid that, residuals properties should be analyzed to create the algorithm that is not sensitive to the noise/uncertainty. The residual analysis employs logical analysis of residual patterns which are called signatures.

To enhance the isolability of faults, the directional properties of the residuals in response to a particular fault may be used. The fault detection filter, a special dynamic observer that generates directional residuals, was first developed at the beginning of the seventies [34], [35]. After that the problem was studied by several authors employing different approaches [36], [37], [38], [39]. Keller [40] developed a fault isolation filter for the linear stochastic systems with multiple faults and unknown inputs. This filter is a particular form of the Kalman filter that can isolate q faults given at least q output measurements. In Giovanini and Dutka [10] the fault detection filter for non-linear system with unmeasured inputs and multiple faults is

presented. It uses the state-dependent coefficient parameterization. This methodology transfers the non-linear system into a quasi linear structure. Then, the fault detectability matrix is used by the filter to update the state estimate with specially designed gain matrix. The remaining freedom of design is used to shape the dynamics of the filter. For stochastic systems, additional degrees of freedom may be used to minimize the effect of the noise on generated residuals.

2.4.1 Fault modelling

The faults in the system may be modelled in an additive or a multiplicative way. The additive measurement faults are the discrepancies between measured and true values and these naturally model biases. The multiplicative faults may represent loss of sensitivity in sensors or actuators. The additive process faults are the disturbances that may act upon the system as additional unknown inputs like loads or leaks. The multiplicative process faults describe changes like a gradual or abrupt deterioration of the plant equipment. The actual nature of the fault should be considered based on the application and the fault model chosen accordingly. In this work the additive faults are considered. The multiplicative faults like a loss of the sensor gain may also be modelled as the additive time-varying signal.

The following state-space system model with additive faults will be considered:

$$\begin{aligned} x_{n+1} &= A(x_n)x_n + B_n(x_n)u_n + F_n(x_n)f_n + w_n \\ y_n &= C(x_n)x_n + f_{S,n} + v_n \end{aligned} \tag{2.22}$$

where

$x_n \in \Re^p$ is a state vector, $u_n \in \Re^q$ is a control vector, $y_n \in \Re^r$ is an output, $f_{S,n} \in \Re^l$ is sensor faults vector, $f_n \in \Re^m$ is an actuator and a component faults vector, w_n is the process noise, v_n is the measurement noise, $A(x_n)$, $B_n(x_n)$, $F_n(x_n)$, $C(x_n)$ are the model non-linearities.

2.4.2 Sensors fault detection

The fault detection is linked to the residuals analysis. The residuals are generated based on system inputs/outputs and the model. With the hypothesis that input faults f_n are not present in the system (2.22) the residuals that are sensitive to only one output fault are generated using the dedicated observer structure [41]. Dedicated output estimators based on the state-dependent Kalman filter may be used for that purpose. The system block diagram is shown in Figure 2-1.

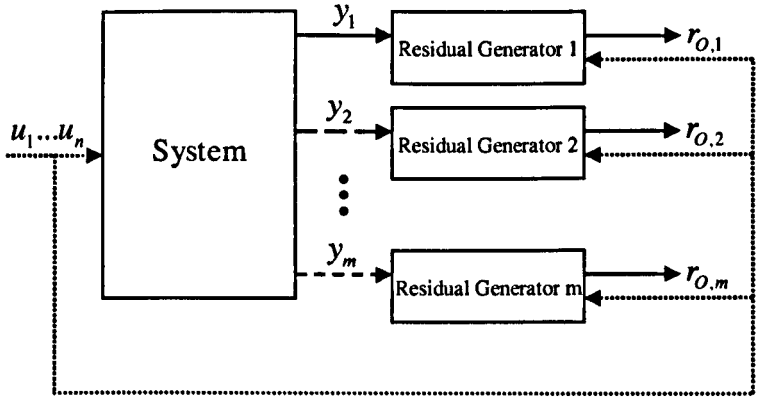

Figure 2-1: Dedicated observer scheme

The state-dependent Kalman filter requires state estimates for the model update. In the presence of the additive sensor fault, state estimates will diverge from the actual system state values. It may be demonstrated using the following equation:

$$\hat{x}_{n+1} = \left(\hat{A}_n - \hat{A}_n K_n \hat{C}_n\right)\hat{x}_n + \hat{A}_n K_n C_n x_n + \hat{A}_n K_n f_{S,n} + \hat{A}_n K_n v_n + \hat{B}_n u_n \tag{2.23}$$

The senor fault $f_{S,n}$ which is assumed to have a non-zero mean value acts as an additional input to the estimator resulting in the state estimation offset. This offset may result in the discrepancy between model matrices $\hat{A}_n = \hat{A}(\hat{x}_n)$, $\hat{B}_n = \hat{B}(\hat{x}_n)$, $\hat{C}_n = \hat{C}(\hat{x}_n)$ and the actual system matrices $A_n = A(x_n)$, $B_n = B(x_n)$, $C_n = C(x_n)$.

The discrepancy depends on particular system non-linearity and the general analysis for an arbitrary non-linear system model is not possible. For the linear systems, the system matrices are constant and model mismatch does not occur. The state-dependent Kalman filter error signal generated for *i-th* output and used as a residual is given by the following expression:

$$r_{i,n} = C_n x_n - \hat{C}_n \hat{x}_n + f_{s,n} + v_n \tag{2.24}$$

The residual $r_{i,n}$ will directly reflect the sensor fault $f_{s,n}$. Unfortunately, the past values of the fault signal $f_{s,n}$ are also present in the state estimate $\hat{x}_n$ (see equation (2.23)). This results in rather unpredictable response of the residual $r_{i,n}$ to the fault $f_{s,n}$. This is due to the non-linear nature of the system (2.22). Aforementioned state-dependent model mismatch may have the negative effect on the residual signal sensitivity to the fault. This negative influence is caused by the state estimate $\hat{x}_n$ bias and the model matrix $\hat{C}_n$ mismatch.

For systems that are open loop stable a direct use of the non-linear system model (2.22) may provide better results. The state of the system may be obtained using the following equation:

$$\hat{x}_{OL,n+1} = \hat{A}_{OL,n} \hat{x}_{OL,n} + \hat{B}_{OL,n} u_n \tag{2.25}$$

where

$$\hat{A}_{OL,n} = \hat{A}\left(\hat{x}_{OL,n}\right),\ \hat{B}_{OL,n} = \hat{B}\left(\hat{x}_{OL,n}\right) \text{ and } \hat{C}_{OL,n} = \hat{C}\left(\hat{x}_{OL,n}\right).$$

The state $\hat{x}_{OL,n}$ estimation mismatch results from the process noise w_n that is not attenuated in the open loop estimation (as opposed to the closed-loop estimation is the equation (2.23)). However, the fault $f_{s,n}$ does not influence the state estimate in any way. The residual is proportional to the state estimation error ($C_n x_n - \hat{C}_{OL,n} \hat{x}_{OL,n}$),

the measurement noise v_n and the fault magnitude $f_{S,n}$. Note, that the negative effect of the state estimation error $C_n x_n - \hat{C}_n \hat{x}_n$ that was present in (2.24) is eliminated in (2.26).

$$r_{OL,i,n} = C_n x_n - \hat{C}_{OL,n} \hat{x}_{OL,n} + f_{S,n} + v_n \tag{2.26}$$

The above dedicated observer based residual generator given either be the equation (2.24) or by the equation (2.26) must be built for each output that is equipped with the sensor being subject to the fault. Note that the choice of the residual generation method ((2.24) or (2.26)) depends on the particular application.

2.4.3 Process faults detection

For the process (or input, actuator) fault estimation a fault detection filter with directional residuals must be designed. The filter generates directional residuals that reflect unknown inputs magnitudes and the remaining residuals are orthogonal to the system fault(s). The orthogonal residuals are used for unbiased estimation and the process and measurement noise attenuation. The system model of interest is given by the equation (2.22) with the assumption that that output faults $f_{S,n}$ are not present in the system. In this section the following notation for the model (2.22) matrices is used: $A_n = A(x_n)$, $B_n = B(x_n)$, $C_n = C(x_n)$ $F_n = F(x_n)$.

It is assumed that $\underset{x_n \in \Omega_x}{\forall} \{C_n, A_n\}$ is point-wise observable [23] in the operating region Ω_x. The process noise w_n and the measurements noise v_n are independent white Gaussian signals with $\mathrm{cov}\{w_n\} = Q$ and $\mathrm{cov}\{v_n\} = R$. Q and R are diagonal semi-positive and positive definite matrices respectively.

The actuator and component faults may be modelled by unknown inputs to the system. With the model, unknown inputs may be de-coupled from residuals. The unknown input observer that de-couples the state-dependent Kalman filter innovation

sequence from the fault/unknown input is constructed as in [10]. The following assumption is made:

$$rank(C_n) = r,\ rank(F_n) = m,\ r \geq m \tag{2.27}$$

The above assumption must be fulfilled for the system's operating state space. It defines that the number of faults/unknown inputs cannot be greater than the number of outputs.

The Fault Isolation Filter design

The following notation is introduced: $\hat{F}_n = \left[\hat{F}_{1,n} \ldots \hat{F}_{m,n}\right]$, $\hat{F}_n = F(\hat{x}_n)$, $f_n = \left[f_{1,n} \ldots f_{m,n}\right]$. Definitions of fault detectability index and matrix for state dependent model are established in a similar way as it was done for the time invariant continuous time system by Liu and Si [39].

Definition 1: The state-dependent system (2.22) has fault detectability indexes $\rho = \{\rho_1, \ldots, \rho_n\}$ defined as

$$\rho_i = \min\left\{o : C_n \tilde{A}_{n,o} \hat{F}_{i,n-o} \neq 0, o = 1,2,\ldots\right\} \tag{2.28}$$

where

$$\tilde{A}_{n,o} = \begin{cases} I & o = 1 \\ \hat{A}_{n-1}\hat{A}_{n-2}\cdots\hat{A}_{n-(o-1)} & o > 1 \end{cases} \tag{2.29}$$

It is assumed, that for the operating space Ω_X state detectability indexes for faults/unmeasured inputs are invariant.

Definition 2: Assuming that the system (2.22) has finite detectability indexes, the fault detectability matrix $\hat{\Psi}_n$ is defined as

$$\hat{\Psi}_n = \hat{C}_n \hat{\Xi}_n \tag{2.30}$$

where

$$\hat{\tilde{\Xi}}_n = \left[\tilde{A}_{n,\rho_1} \hat{F}_{1,n-\rho_1} \ \cdots \ \tilde{A}_{n,\rho_m} \hat{F}_{m,n-\rho_m} \right]. \tag{2.31}$$

Faults associated with detectability matrix (2.31) are given in the following form

$$\Phi_n = \left[f_{1,n-\rho_1} \ \cdots \ f_{m,n-\rho_m} \right]^T \tag{2.32}$$

The output of the system is defined as follows:

$$y_n = \hat{C}_n \tilde{x}_n + \hat{C}_n \hat{\tilde{\Xi}}_n \Phi_n + v_n \tag{2.33}$$

where

$\tilde{x}_n$ is the state of the system without the last fault that may be seen on the output.

The dynamic observer is given by the following equation:

$$\begin{aligned} \hat{\tilde{x}}_{n+1} &= \hat{A}_n \hat{\tilde{x}}_n + \hat{B}_n u_n + K_n q_n \\ \hat{\tilde{y}}_n &= \hat{C}_n \hat{\tilde{x}}_n \end{aligned} \tag{2.34}$$

where

$\hat{\tilde{x}}_n$ and $\hat{\tilde{y}}_n$ are state and output estimate vectors.

In the equation (2.34) the output residual q_n is given by:

$$q_n = y_n - \hat{\tilde{y}}_n = \hat{C}_n e_n + \hat{C}_n \hat{\tilde{\Xi}}_n \Phi_n + v_n \tag{2.35}$$

where

e_n is the estimation error and

$$e_n = \tilde{x}_n - \hat{\tilde{x}}_n \tag{2.36}$$

Observe that the residual q_n has three components: the estimation error due to state errors e_n, the effect of the last faults/unmeasured inputs over the system outputs and the measurement noise. The second component may lead to the biased estimation and to the divergence of estimated states. To solve this problem two matrix coefficients T_n and Σ_n are introduced in a similar way as in Keller [40]. The residual q_n is now represented by two auxiliary residuals, which are given by the following expression:

$$\begin{bmatrix} \hat{q}_n \\ \hat{\Phi}_n \end{bmatrix} = \begin{bmatrix} \Sigma_n \\ \mathrm{T}_n \end{bmatrix} q_n \tag{2.37}$$

Replacing for the residual q_n the equation (2.37) is rewritten as

$$\begin{aligned} \hat{q}_n &= \Sigma_n \hat{C}_n e_n + \Sigma_n \hat{C}_n \hat{\Xi}_n \Phi_n \\ \hat{\Phi}_n &= \mathrm{T}_n \hat{C}_n e_n + \mathrm{T}_n \hat{C}_n \hat{\Xi}_n \Phi_n \end{aligned} \tag{2.38}$$

where

$\hat{\Phi}_n \in R^m$ are residuals associated with the faults/unmeasured inputs and

$\hat{q}_n \in R^{r-m}$ are residuals decoupled from faults/unmeasured inputs.

To obtain this decoupling an effect of faults/unmeasured inputs has to be removed from the estimation error. Thus, the matrix Σ_n must satisfy:

$$\Sigma_n \hat{C}_n \hat{\Xi}_n = 0 \in R^{r-m \times m} \tag{2.39}$$

where

Σ_n must be a matrix of full rows rank (*r-m*) and orthogonal to $\hat{C}_n \hat{\Xi}_n$.

The directional residuals ($\hat{\Phi}_n$) should contain unchanged information about faults/unmeasured input magnitudes, therefore T_n is required to satisfy:

$$T_n \hat{C}_n \hat{\Xi}_n = I \in R^{m \times m} \tag{2.40}$$

where

T_n is generalized inverse or pseudo-inverse matrix.

$$T_n = \left(\hat{C}_n \quad \hat{\Xi}_n\right)^+ \tag{2.41}$$

Finally auxiliary residuals are given by:

$$\begin{aligned} \hat{q}_n &= \Sigma_n \hat{C}_n e_n, \\ \hat{\Phi}_n &= T_n \hat{C}_n e_n + \Phi_n \end{aligned} \tag{2.42}$$

With the equation (2.42) the dynamic observer may be written in the following form:

$$\begin{aligned} \hat{\hat{x}}_{n+1} &= A_n \hat{\hat{x}}_n + B_n u_n + \begin{bmatrix} K_{UIKF,n} & W_n \end{bmatrix} \begin{bmatrix} \Sigma_n \\ T_n \end{bmatrix} q_n \\ \hat{\hat{y}}_n &= C_n \hat{\hat{x}}_n \end{aligned} \tag{2.43}$$

where

$K_{UIKF,n}$ is the filter gain and W_n is the matrix that propagates the effect of faults/unmeasured inputs into the next time instant and

$$W_n = \hat{A}_n \hat{\Xi}_n \tag{2.44}$$

By performing the feedforward update of the state estimate it is maintained that the state of the filter contains information about all but the last fault/unmeasured input. The dynamic properties of the filter are determined by the gain $\underline{K}_{UIKF,n}$. To design this gain, the fault isolation filter (2.43) is re-written as follows

$$\begin{aligned} \hat{\hat{x}}_{n+1} &= \underline{A}_n \hat{\hat{x}}_n + B_n u_n + \underline{K}_{UIKF,n} y_n, \\ \hat{\hat{y}}_n &= \hat{C}_n \hat{\hat{x}}_n \end{aligned} \tag{2.45}$$

where

$$\underline{A}_n = \hat{A}_n - W_n \mathrm{T}_n \hat{C}_n - K_{UIKF,n}\Sigma_n \hat{C}_n,$$
$$\underline{K}_{UIKF,n} = K_{UIKF,n}\Sigma_n + W_n \mathrm{T}_n \tag{2.46}$$

The gain $K_{UIKF,n}$ for deterministic systems is designed to place poles of the closed loop system matrix $\underline{A}_n$ such that the desired dynamic properties of the observer are obtained. For stochastic systems the gain $K_{UIKF,n}$ is obtained from the solution of the Riccati equation [40]:

$$K_{UIKF,n} = \underline{P}_n \underline{C}_n^T \left(\underline{C}_n \underline{P}_n \underline{C}_n^T + \underline{R}_n\right)^{-1}$$
$$\underline{P}_n = \underline{A}_n \left[\underline{P}_n - \underline{P}_n \underline{C}_n^T \left(\underline{R} + \underline{C}_n \underline{P}_n \underline{C}_n^T\right)^{-1} \underline{C}_n \underline{P}_n\right] \underline{A}_n^T + \underline{Q} \tag{2.47}$$

where

$$\underline{C}_n = \Sigma_n \hat{C}_n,$$
$$\underline{R}_n = \Sigma_n R \Sigma_n^T$$

System state estimation

It should be noticed that the state estimates $\hat{x}_n$ and $\hat{\tilde{x}}_n$ will differ in the presence of faults. For the state-dependent model $(\hat{A}_n, \hat{B}_n, \hat{C}_n, \hat{F}_n)$ the accuracy of state estimates is very important. Any discrepancy results in the model mismatch as a consequence of the state-dependent model nature. The state $\hat{\tilde{x}}_n$ is the estimate of artificially introduced state. This state does not reflect the state of the real system since it has no information about the last fault and faults which occurred in less than the detectability index discrete time steps in the past. The detectability index determines the time delay after which the fault may by observed on the output. It is

possible to use directional residuals to estimate state of the system, but s steps behind the current time ($s = \max\{\rho_i,\ i = 1 \ldots m\}$) as given by the equation (2.48).

$$\begin{aligned} \hat{x}_{n-s+1} &= \hat{A}_{n-s}\left(\hat{x}_{n-s} + K_{n-s}\left(y_{n-s} - \hat{C}_{n-s}\hat{x}_{n-s}\right)\right) + \hat{B}_{n-s}u_{n-s} + \hat{F}_{n-s}f_{n-s} \\ \hat{y}_{n-s} &= \hat{C}_{n-s}\hat{x}_{n-s} \\ f_{n-s} &= \left[\hat{\Phi}_{1,n+\rho_1-s} \quad \cdots \quad \hat{\Phi}_{m,n+\rho_m-s}\right]^T \end{aligned} \tag{2.48}$$

The filter gain K_{n-s} is computed using equations (2.17) and (2.18) with the system model computed at the state estimate $\hat{x}_{n-s}$. The second stage of the estimation involves projection of the past state estimate to the current time instant. The Kalman filter equations are iterated until the current time ($i = s-1, s-2, \ldots, 0$). The dynamic observer is used alongside with the past output measurements as in the equation (2.49).

$$\begin{aligned} \hat{x}_{n-i+1} &= \hat{A}_{n-i}\left(\hat{x}_{n-i} + K_{n-i}\left(y_{n-i} - \hat{C}_{n-i}\hat{x}_{n-i}\right)\right) + \hat{B}_{n-i}u_{n-i} + \hat{F}_{n-i}f_{n-i} \\ \hat{y}_{n-i} &= \hat{C}_{n-i}\hat{x}_{n-i} \\ f_{n-i} &= \left[\hat{\Phi}_{1,\min(n,n+\rho_1-i)} \quad \cdots \quad \hat{\Phi}_{m,\min(n,n+\rho_m-i)}\right]^T \\ i &= s-1, s-2, \ldots, 0 \end{aligned} \tag{2.49}$$

The equation (2.49) is very similar to (2.48). The fault estimates with the fault detectability index greater than i are not available therefore the most recent fault estimate $\hat{\Phi}_{k,\min(n,n+\rho_k-i)}$ is used instead. It is assumed that the fault/unmeasured input did not change until *n-th* (current) time instant. This implies that the method should be used for systems where fault magnitudes are stationary or slowly varying. As the result of the iteration $i = s-1, s-2, \ldots, 0$ the state estimate $\hat{x}_n$ is obtained. The current state estimate is used for the state-dependent model matrices calculation.

2.4.4 Generalized Observer Scheme, fault detection and isolation

To detect the fault a threshold tests for residuals are carried out. The result of these tests is the fault detection statement. The fault isolation is the next task. The dedicated observer scheme requires a separate estimator for each fault [42], [43]. For the sensor fault detection the hypothesis that process faults are not present must be made. The dedicated observer scheme was elaborated in section 2.4.2. For the process faults the non-linear fault detection filter that generates directional residuals is employed (see section 2.4.3). The hypothesis that sensor faults are not present must be made to facilitate the fault detection. Based on the fault detection filter formulated in section 2.4.3 the signal $\hat{\Phi}_n$ being the estimate of the fault may be used as the residual. Alternatively, the signal $\hat{q}_n$ that is orthogonal to the fault could also be used.

For stochastic or uncertain systems the residuals are not zero for the fault-free system. The thresholds for residuals must be established. If residuals are within pre-defined boundaries, the system is assumed to be fault-free. The residual(s) exceeding threshold(s) indicate the presence of the fault in the system. It is important to define thresholds in a way that the noise or the system uncertainty does not trigger false alarms. In practice, thresholds should be based on the information about extreme values of the fault-free system residuals. The formal analytical derivation of thresholds for complex systems is not possible in practice. Thresholds may be determined based on the simulation results for the fault-free system. The same procedure for the threshold formulation may be used with the real experiment. The safety margin should be included in the design to achieve the robustness. The fault in the system is detected if the value of any residual violates its threshold. This indicates that the fault is present in the system but does not locate the fault.

The fault isolation method based on the generalized observer scheme uses the table with 'fault signatures' [43]. In this way the hypotheses made for the residual generation are managed. The sensor faults may be detected since the sensor fault results in its dedicated residual change. The other sensors residuals are insensitive to

such fault. The sensor faults also interact with process fault residuals. The process faults result in the change of its directional residual and all process faults are mutually decoupled. The sensor observer directional residuals also change their values as the effect of the process fault. Based on such analysis the relationship between faults and residuals is established, as in the Table 2.1, for the fault isolation. The logical value 1 denotes that the relationship exists, 0 that it does not. The value X denotes that the relationship is not specified (could be 1 or 0 depending on system structure specific to the particular application).

	f_{S1}	f_{S2}	⋯	$f_{S,n}$	f_1	f_2	⋯	f_m
$r_{S,1}$	1	0	⋯	0	X	X	⋯	X
$r_{S,2}$	0	1	⋯	0	X	X	⋯	X
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋱	⋮
$r_{S,n}$	0	0	⋯	1	X	X	⋯	X
r_1	X	X	X	X	1	0	⋯	0
r_2	X	X	X	X	0	1	⋯	0
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋱	⋮
r_m	X	X	X	X	0	0	⋯	1

Table 2.1: Fault signatures table

The sensor fault f_{Si} associated with *i-th* output affects only the residual $r_{S,i}$. For process faults, the fault f_i results in the process fault residual r_i being affected. Based on the Table 2.1 the logic rules for fault isolation may be devised. Since the non-linear systems are subject of this analysis, the ability to efficiently isolate multiple faults depends on the particular application. This is due to the unknown behaviour of the system shown in Table 2.1 and associated with unspecified states X.

If senor faults are not considered (i.e. there is a hardware redundancy in sensors) the ability of the process fault detection filter to detect multiple faults is determined by the fault detectability matrix as explained in section 2.4.3. In general it is not possible to detect number of faults that is greater than number of outputs. Sometimes, the fault isolation is carried out with the assumption that only one fault occurs at a time. In that case number of faults may exceed number of measured

system outputs. The fault isolation technique may be improved by introduction of the statistical analysis of residual signals. The variance and mean value of residuals may be tested for that purpose. Additionally the specific system knowledge may be incorporated to improve the fault detection speed and robustness. This however depends on the particular application and is not considered in this chapter. The application of the fault detection theory presented here will be given in section 6.4.

2.5 Summary

In Chapter 2 the basic estimation theory and the non-linear fault detection methods were introduced. The chapter started with the introduction of the parameter and state estimation methods for linear systems. These methods were outlined to give the basis for the non-linear techniques analysis. The estimation methods for the non-linear system include the extended Kalman filter that may be used for the parameter and state estimation. An alternative to the extended Kalman filter was given and the state-dependent Kalman filter for the discrete time systems was presented. Next, the analysis of the application of the dedicated observer scheme for the sensor fault detection was given. The fault detection filter based on the state-dependent models was elaborated next. The filter provided the ability to detect multiple process faults. The fault directional and orthogonal residuals were generated for the process fault detection and isolation. Finally, the methodology of the simultaneous detection of process (or input) and sensor (or output) faults was given. The methodology used both: sensor and process fault residuals generators within the logic-based framework.

Chapter 3

Non-linear Optimal and Predictive Control

The non-linear control techniques attract now most of the attention of researchers working in the field of control engineering. This stems from the fact that the majority of real objects is non-linear. For that reason efforts presented in this chapter are directed on the development of non-linear control algorithms. The model based techniques are employed here. It is important to remember that a good model of the system is required if these techniques are to be considered.

The state space control techniques analyzed in this chapter require state to be available to the controller. In some cases the state is measured directly. Sometimes when the noise distorts measurements filtering techniques presented in previous chapter have to be used. If the state in not measured directly the state estimator must be constructed. The extended and state-dependent Kalman filters presented in Chapter 2 may be used for the noise filtering and the state estimation. With the state assumed available two main types of control techniques will be presented. In section 3.1 the Riccati equation based techniques are introduced. These methods extend the linear-quadratic (LQ) techniques to non-linear systems through the state-dependent model parameterization. In a similar way in section 3.2, with the same state-dependent model parameterization, the state-space predictive control algorithm is used in a non-linear context.

In section 3.1.1 the discrete state-dependent Riccati equation (DSDRE) method is presented. In section 3.1.2 a prediction of the trajectory is used to approximate the non-linear system with the time-varying linear model. The difference Riccati equation is employed in the solution. This method significantly improves the closed loop system response. However, it does not lead to the optimal solution is a sense of the minimal value of the cost function. The remedy is presented in section 3.1.3 where the correction tensors are introduced in the solution. The numerical example in section 3.1.4 shows results obtained with the discrete SDRE, the discrete SDRE with the predicted trajectory and its optimized version. Next, in section 3.2 the predictive control algorithms are presented. Two first algorithms are similar to the discrete SDRE with the prediction and its optimized version. It is aimed to drive the state of the system to the origin. The main difference here is a predictive context. The future control action vector from previous iteration of the predictive control algorithm is used. This methodology is different from Riccati equation based solution where previous state-feedback gains were used. The direct use of control vector is less demanding from the computational point of view but the robustness may suffer. This regulatory predictive algorithm is presented in section 3.2.1. Its optimized version follows in section 3.2.2. The numerical example is presented in section 3.2.3. Lastly, the non-linear GPC algorithm with the explicit reference trajectory is presented in section 3.2.4. The algorithm provides the ability to use the future reference signal if such is available in the system. This will be used within the engine control algorithm in section 5.4.

3.1 State-Dependent Riccati Equation with Predicted Trajectory

There is a need for control laws that are simple to compute, suitable for nonlinear systems [44] that may be optimized in some sense [45]. The family of LQ and LQG design methods [46], [19] have been very successful for linear systems and it is desired to provide an equally simple method that can be used for nonlinear systems. Over the past three decades several non-linear versions of LQG algorithm have emerged. In 1962 Pearson [47] used a linear state-dependent representation of

the non-linear system for the first time. The assumption was that the non-linear system described by the non-linear state space model may be re-arranged into a so-called 'state dependent' linear form. The non-linear behaviour of the system in such a representation is determined by the state dependent matrices. Later, in 1969 Boughart [48] continued this work but there was little interest in this method for a period. In 1996 Cloutier [49], [50] resurrected this idea using a stricter mathematical justification for the method. It was reported [51] that the state-dependent Riccati Equation (SDRE) method has many advantages over other non-linear design methods. The main drawback is the lack of a guarantee of global asymptotic stability which in general is a difficult issue for non-linear systems. The local stability at the origin of the closed loop system results from the stabilizing properties of the solution of the algebraic Riccati equation. Unfortunately, so far, one of the most efficient methods of assessing the stability of the SDRE controller is by simulation. Recent work in the stability analysis of the SDRE method either gave rather difficult conditions to check or imposed difficult requirements. In [52] the region of attraction for the SDRE controller, around the origin of the closed loop, is determined and for this region the stability of the controller is guaranteed. This may be difficult since closed-loop system equations are usually not known explicitly. In [53], [54] the stability of the system controlled by the SDRE method is ensured via "satisficing" provided that a Control Lyapunov Function for considered system is known. The main difficulty with this technique is to find the global control Lyapunov function for the non-linear system. For some systems such a function may easily be determined and in this case the method may be employed. In [55] the estimation of the region of stability is substituted by the functional search problem. The state-dependent model matrices were assumed to be polynomial functions of the state and the stability region estimate was obtained though optimization.

The evidently questionable assumption made by the state-dependent Riccati equation (SDRE) method will be tackled in this section. In the SDRE method the calculations are performed, assuming the system remains fixed (time-invariant) at the state value measured or estimated at current operating conditions. The frozen system matrices calculated at this point are used for the solution of the algebraic Riccati

equation. To remove that inaccurate assumption the linear quadratic optimal control [56] results for time-varying linear systems will be used in section 3.1.2. The main idea is to estimate the future variations in the nonlinear system characteristics [57] and apply the linear time-varying optimal control results. A restricted class of nonlinear systems is used, which is the same as that employed in papers on the state-dependent Riccati equation approach [49], [50]. The state-dependent Riccati equation technique with predicted trajectory assumes that the system may be approximated using the linear time-varying system model.

The optimality of the solution will be analyzed next in section 3.1.3. For some applications a sub-optimal solution of the minimization problem may not be sufficient. The optimal solution based on the infinite horizon cost function will be derived. The improvements in terms on the value of the performance index will be demonstrated in section 3.1.4.

3.1.1 Discrete Time SDRE Method

The SDRE method was originally developed for continuous time systems [50], [23]. The solution is a direct result of adopting the linear continuous time optimal control method that was based on the algebraic Riccati equation [16]. Thus, the theory that is well established for linear systems may be used in the context of non-linear systems.

In this thesis, the attention is focused on non-linear discrete time systems. For the linear discrete time systems the control minimizing an infinite horizon quadratic performance index is given by the solution of the discrete algebraic Riccati equation (DARE). In a manner similar to the original SDRE the solution of the algebraic Riccati equation, or rather its discrete version [11] is used in the context of non-linear discrete time systems. The non-linear discrete time system considered here is given by the following control affine non-linear difference equation. The control-affine structure may pose restrictions in applicability of presented algorithm. The assumption may however be dropped in the algorithm proposed in section 3.1.2.

$$x_{n+1} = f(x_n) + B(x_n)u_n \tag{3.1}$$

The model (3.1) is re-arranged and the state-dependent form (3.2) of the system is obtained:

$$x_{n+1} = A(x_n)x_n + B(x_n)u_n \tag{3.2}$$

where

x_n - state vector, u_n - control vector.

Detailed discussion of possible methods of getting the state-dependent form is given in [58]. In general there are an infinite number of such re-arrangements. This may be regarded as an additional degree of freedom of design. An assumption on point-wise controllability must be made here, i.e. $\underset{x}{\forall}\left(A(x_n)B(x_n)\right)$ is controllable. The infinite horizon cost function being minimized is given by the following expression:

$$J_n = \frac{1}{2}\sum_{i=n}^{\infty}\left\{x_i^T Q x_i + u_i^T R u_i\right\} \tag{3.3}$$

where

Q and R are symmetric and semi-positive and positive definite matrices respectively.

Note that the assumption on positive definite properties of weighting matrix R may pose limitations if one of control signals is not desired to the penalized. The solution of the minimization problem (3.3) is obtained by solving the discrete state-dependent Riccati equation (DSDRE). This yields, in general, a sub-optimal solution. The DSDRE is obtained by freezing the system (3.2) at current state x_n and assuming that it will remain time-invariant in the future. The solution of such problem is well known from the linear optimal control theory [59], [60] and will not be presented here. The DSDRE is given by the following equation:

$$P(x_n) = A(x_n)^T \Big[P(x_n) - P(x_n)B(x_n)^T \times \Big(R + B(x_n)^T P(x_n) B(x_n) \Big)^{-1} B(x_n) P(x_n) \Big] A(x_n) + Q \tag{3.4}$$

The non-linear control action is computed from the expression (3.5). It is similar to the linear-quadratic optimal control law for linear systems [59], [60]:

$$\begin{aligned} u_n &= -K_n x_n \\ K_n &= \Big(B(x_n)^T P_n B(x_n) + R \Big)^{-1} B(x_n)^T P_n A_n \end{aligned} \tag{3.5}$$

If the analytical solution of DSDRE (3.4) exists, the state feedback gain (3.5) based on the solution for $P(x_n)$ from the equation (3.4) is used as a non-linear feedback control law. Otherwise, the equation is solved at each sampling instant numerically. It is also possible to pre-compute solutions of the Riccati equation and state feedback gains. These may be used for the gain scheduling control with the state employed as a scheduling parameter. Note that if number of states (order of the system) is high, the gain scheduling may require a significant amount of memory to store pre-computed gains.

The solution of the DSDRE for the system (3.1) results in a locally stabilizing control. As mentioned earlier, the most efficient method of the stability analysis for the DSDRE controller is by simulation. Recent work in the stability assessment for the SDRE method either gave rather difficult to check conditions or imposed difficult to fulfil requirements. The optimality of the solution depends on the form of the state-dependent parameterization (3.2) and in general the solution is sub-optimal. The non-linear control method extending the DSDRE method by removing the assumption about the frozen state will now be analyzed. The problem of the optimality of the solution will follow that development.

3.1.2 Discrete Time SDRE with Predicted Trajectory

The discrete time state-dependent Riccati equation (DSDRE) method employs a solution of the discrete algebraic Riccati equation (DARE). The state-space matrices in the equation (3.2) must be frozen at current state to obtain the linear time invariant model representation. This is equivalent to the assumption that the system will remain fixed at the current operating point in the future. This assumption represents a severe approximation since it is only true for the steady state at the origin.

In this section it is assumed that prediction of the future state trajectory may be determined. With this knowledge, the DARE may be solved not just for the current state (as it was done in the DSDRE) but also for the prediction of the future state. For a discrete time system controlled at time n it would mean that the DARE is solved at $n+N$, where N is the last state prediction available. If the state at $n+N$ time instant represents the steady state of the system then the solution of the DARE may be used as a boundary condition for the solution of the difference Riccati equation which is iterated backwards using predictions of the system matrices. Finally the state feedback gain and the control signal may be obtained. The assumption on the knowledge of the state trajectory may be satisfied at a given time instant n by using the model of the system and predicting the future control and state values for n, $n+1$, $n+2$,..., $n+N-1$. These values might for example be approximated using the last calculated value of the gain matrix $K_C(n-1)$ (or the sequence of time-varying gains from previous iteration of the control algorithm) and the state-dependent model of the system. The future trajectory provides an indication of the likely time variation of the system matrices. Given the time-varying system matrices the linear time-varying quadratic optimal controller results may then be applied. Thus, the solution of the DARE is first determined using the system model at time $n + N$, which is assumed time invariant from that point on. The solution of the algebraic Riccati equation (say P_∞) can then be used to initialize the time-varying Riccati difference equation to solve backwards in time. The values of the Riccati solution $\{P(.)\}$ at times $n+N-1$, $n+N-2$,..., $n+1$ may then be computed. The gain at time n, which is to be used to

compute the control signal at time instant n then follows. The whole process must be repeated at the next time instant in a receding horizon control fashion.

If the system is controllable the state may be driven sufficiently close to the origin in a finite number of steps. It is important to make sure that the DSDRE method is capable of stabilizing the system. The stability issues are analyzed in [52], [53], [55]. If those methods cannot be applied it is quite common for non-linear systems that the stability is evaluated though simulation. As was already stated, the matrices in the state-dependent linear parameterization (3.2) are implicit functions of time through the dependence on state. With the knowledge of the future trajectory the non-linear system may be approximated by a linear time varying system [11]. The future trajectory is obtained with the state feedback gain computed in the previous iteration. The minimization of the cost function may be split in two parts:

$$J_n = \underbrace{\frac{1}{2}\sum_{i=n}^{n+N-1}\left\{x_i^T Q x_i + u_i^T R u_i\right\}}_{J_1} + \underbrace{\frac{1}{2}\sum_{i=n+N}^{\infty}\left\{x_i^T Q x_i + u_i^T R u_i\right\}}_{J_2} \tag{3.6}$$

The state feedback drives the system (3.1) to the steady state at the origin after a finite number of steps N. The solution of the discrete algebraic Riccati equation (DARE) (3.7) computed for the system (3.2) at state x_{n+N} minimizes J_2 part of the cost function (3.6).

$$\begin{aligned} P_{n+N} = A(x_{n+N})^T \Big[& P_{n+N} - P_{n+N} B(x_{n+N})^T \\ & \left(R + B(x_{n+N})^T P_{n+N} B(x_{n+N})\right)^{-1} B(x_{n+N}) P_{n+N} \Big] A(x_{n+N}) + Q \end{aligned} \tag{3.7}$$

The state of the system from the current value to the steady state at the origin evolves in time and so the state dependent model matrices do. The DARE solution P_{n+N} is used as a boundary condition for the time-varying optimal control problem. Derivation of the optimal control solution for the linear time varying systems is presented in [60]. The solution for the finite horizon part J_1 minimization in (3.6) is based upon the time-varying approximation of the non-linear system. This requires the following difference Riccati equation:

$$P_i = A(x_i)^T \left[P_{i+1} - P_{i+1} B(x_i)^T \left(R + B(x_i)^T P_{i+1} B(x_i) \right)^{-1} B(x_i) P_{i+1} \right] A(x_i) + Q \qquad (3.8)$$

The equation is iterated from $i = n+N-1$ using the solution of the DARE $P_{n+N} = P(x_{n+N})$ given by equation (3.4). The iterations of (3.8) are terminated at $i = n+1$. The state feedback gain resulting from the linear time-varying solution [60] is given by the following expression:

$$\begin{aligned} u_n &= -K_n x_n \\ K_n &= \left(B(x_n)^T P_{n+1} B(x_n) + R \right)^{-1} B(x_n)^T P_{n+1} A_n \end{aligned} \qquad (3.9)$$

The idea behind this control strategy is similar to the dual mode control solution for model predictive control algorithms [61]. The following algorithm summarizes the described control technique. This provides the refined DSDRE method. The state feedback gain K_n is obtained and the receding horizon technique is used in the algorithm.

Algorithm 3.1

- Use the state feedback gains computed in previous iteration for the finite horizon N and simulate the closed loop system with the model (3.1) starting from the current state x_n. This provides prediction of the state trajectory.
- The solution of the Riccati equation (3.4) is calculated at x_{n+N}. The state-dependent model matrices are assumed time invariant after the time $n+N$.
- Within the finite horizon the state dependent matrices are calculated along the prediction of the state trajectory. This results in the linear time varying model that is an approximation of the non-linear system.
- Within the finite horizon N the equation (3.8) is iterated and $P_{n+N} \ldots P_{n+1}$ are computed. Based on that, the state feedback gains $K_n \ldots K_{n+N-1}$ are obtained and the first gain K_n is used for the control.

- In the next discrete time event the algorithm is repeated and the remaining from the current iteration gains $K_{n+1}...K_{n+N-1}$ are used. Note, that for the discrete time algorithm, the state x_{n+N} prediction is obtained using gain K_{n+N-1}.

The use of the prediction of future trajectory results in a better performance of the controller which will be demonstrated on the numerical example that will follow. This is due to more realistic assumptions about the future state. In the next section the optimality of the solution, which is determined by the value of the cost function, will attract attention.

3.1.3 Optimized Discrete SDRE Method

The continuous time state-dependent Riccati equation (SDRE) method gives, in general, a suboptimal locally stabilizing solution of the infinite horizon minimization problem of a quadratic (in control) cost-function, subject to non-linear differential constraints [50]. For scalar systems the solution of the SDRE yields an optimal solution [23]. For systems of higher order the optimality of the solution is determined by the state-dependent parameterization of the system matrix [62]. The proper choice of that parameterization may be difficult, if not impossible, since that may require the solution of the Hamilton-Jacobi-Bellman equation.

The attention is now focused on the optimality of the solution for discrete time systems. The discrete state-dependent Riccati equation (DSDRE) is not guaranteed to give an optimal solution of the minimization of the performance index. For some systems the state-dependent parameterization giving an optimal solution may not exist at all. This may be demonstrated by solving the DSDRE problem for a simple discrete time scalar non-linear system given by the following equation:

$$x_{n+1} = (x_n)^2 + u_n \tag{3.10}$$

The DSDRE controller with the only possible state-dependent parameterization $(x_n)^2 = x_n \cdot x_n$ does not provide an optimal solution, which may easily be verified by the simulation. The numerical optimization provides a lower value of the cost function when compared with the DSDRE solution. This clearly indicates sub-optimality of the DSDRE solution.

The refinement of the DSDRE with the predicted trajectory, given by Algorithm 3.1, brings the improvement. However, the optimality still depends on the state dependent parameterization. The method of recovering optimality will now be presented. The method may also be seen as an alternative to the numerical optimization.

In this section the optimal control for the system (3.1), with the infinite horizon cost function, is analyzed. In the preparation to the controller derivation substitute for $J_2 = \frac{1}{2} x_{n+N}{}^T P_{n+N} x_{n+N}$ in the cost function (3.6). The resulting performance index is given by the following expression:

$$J_n = \frac{1}{2} \sum_{i=n}^{n+N-1} \left\{ x_i{}^T Q x_i + u_i{}^T R u_i \right\} + \frac{1}{2} x_{n+N}{}^T P_{n+N} x_{n+N} \tag{3.11}$$

The matrices Q, P_{n+N} and R are assumed symmetric and semi-positive and positive definite respectively. The P_{n+N} is a final state penalty matrix for the finite horizon optimization. Assume that the system is driven to the origin (or sufficiently close) within the horizon N. The value of the terminal penalty matrix may be obtained from the solution of the discrete algebraic Riccati equation for the system matrices frozen at x_{n+N}. If the state x_{n+N} is in the neighbourhood of the origin, the fixed gain control is capable of stabilizing the system within this closed region. The stability region for the system controlled by the linear state feedback controller may be determined using Lyapunov theory [63], [61].

The Hamiltonian for the minimization of the cost function (3.11) subject to equality constraints of the original non-linear system representation (3.1), is given by:

$$H_i = \frac{1}{2}\left(x_i^T Q x_i + u_i^T R u_i\right) + \lambda_{i+1}^T \left(f(x_i) + B(x_i)u_i\right) \tag{3.12}$$

The optimality conditions for the minimization problem solution are given as follows [16]:

$$\frac{\partial H_i}{\partial u_i} = R u_i + B(x_i)^T \lambda_{i+1} = 0 \tag{3.13}$$

$$\frac{\partial H_i}{\partial x_i} = Q x_i + \left[\frac{\partial f(x_i)}{\partial x_i} + \frac{\partial B(x_i)}{\partial x_i} u_i\right]^T \lambda_{i+1} = \lambda_i \tag{3.14}$$

$$\frac{\partial H_i}{\partial \lambda_{i+1}} = f(x_i) + B(x_i)u_i = x_{i+1} \tag{3.15}$$

The boundary condition for the co-state in the equation (3.14) is $\lambda_{n+N} = P_{n+N} x_{n+N}$. The initial condition for the state in the equation (3.15) (the system state) is x_n. The optimization with the initial value given for the state equation and the final value available for the co-state is known as a two point boundary problem.

To find a solution to the problem introduce the matrix coefficient $P_i = P(x_i, x_{i+1}, ..., x_{n+N})$. Without loss of generality it may be assumed that the following expression for the co-state λ_i holds [60]:

$$\lambda_i = P_i x_i \tag{3.16}$$

From the system equation (3.15), the stationary condition (3.13) and the assumption (3.16) the following expression may be computed:

$$x_{i+1} = \left(I - B(x_i)R^{-1}B(x_i)^T P_{i+1}\right) f(x_i) \tag{3.17}$$

From the co-state equation (3.14), the assumption (3.16) and the equation (3.17) the following expression is obtained:

$$P_i x_i = \left(\frac{\partial f(x_i)}{\partial x_i} + \frac{\partial B(x_i)}{\partial x_i} u_i\right)^T P_{i+1}\left(I - B(x_i)R^{-1}B(x_i)^T P_{i+1}\right) f(x_i) + Q x_i \tag{3.18}$$

The above equation should hold for all x_i in the state-space. The equation (3.18) is re-arranged using the matrix inversion Lemma [60]. The state-dependent parameterization of the system (3.1) given by (3.2) is employed. The following equation is obtained:

$$\begin{aligned} P_i = &\left(\frac{\partial f(x_i)}{\partial x_i} + \frac{\partial B(x_i)}{\partial x_i} u_i\right)^T \\ &\left[P_{i+1} - P_{i+1}B(x_i)\left(B(x_i)^T P_{i+1}B(x_i) + R\right)^{-1} B(x_i)^T P_{i+1}\right] A(x_i) + Q \end{aligned} \tag{3.19}$$

The equation (3.19) has a similar structure to the difference Riccati equation (3.8). Only the following term which can be re-written using the state-dependent parameterization makes two equations different:

$$\frac{\partial f(x_i)}{\partial x_i} + \frac{\partial B(x_i)}{\partial x_i} u_i = \frac{\partial A(x_i)}{\partial x_i} x_i + A(x_i) + \frac{\partial B(x_i)}{\partial x_i} u_i \tag{3.20}$$

The derivatives $\frac{\partial A(x_i)}{\partial x_i}$ and $\frac{\partial B(x_i)}{\partial x_i}$ are tensors (third dimension has to be introduced to accommodate derivatives of each element of A matrix). Note that for the linear system where A and B are constant or time varying but state-independent the equation (3.18) becomes the ordinary difference Riccati equation (3.8). The same result is obtained if matrices A and B are frozen.

The optimal control minimizing the cost function (3.11) is computed from equations (3.13), (3.15), (3.16), which results in the equation (3.9).

The value of P_{n+1} is obtained by iterating the equation (3.19) from $i = n+N-1$ back in time to $i = n+1$. The equations (3.19), (3.20) must be used with the predicted future trajectory. For this trajectory, tensors $\frac{\partial A(x_i)}{\partial x_i}$ and $\frac{\partial B(x_i)}{\partial x_i}$ as well as $A(x_i), B(x_i)$ are computed.

Algorithm 3.2

- Use the state feedback gains for the finite horizon N computed in previous iteration and simulate the closed loop system with the model (3.1) starting from the current state x_n. This provides prediction of the state *and control* trajectory.
- The solution of the Riccati equation (3.4) is calculated at x_{n+N}. The state-dependent model matrices are assumed time invariant after $n+N$.
- Within the finite horizon the state dependent matrices *and tensors* are calculated along the prediction of the state trajectory.
- Within the finite horizon N the equation (3.19) is iterated and $P_{n+N}...P_{n+1}$ are computed. Based on that, state feedback gains $K_n...K_{n+N-1}$ are obtained, the first gain K_n is used for the control.
- In the next discrete time step the algorithm is repeated and gains $K_{n+1}...K_{n+N-1}$ are used.

The optimality of the solution depends strongly on the accuracy of the trajectory that is itself based on the gain sequence obtained in previous iteration.

3.1.4 Numerical Example and Conclusions

As an example a discrete-time model of the driven inverted pendulum is employed. The pendulum diagram is shown in Figure 3-1.

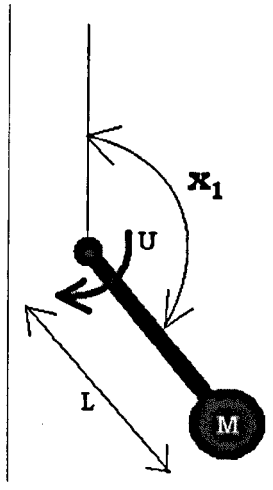

Figure 3-1: Inverted Pendulum

The control task is to find the optimal control sequence for the pendulum from the assumed initial level to the unstable equilibrium point. The model derivation is based on the moment balance at the axis of rotation. The physical model is discretized using the Euler method. Assuming that the state-space origin corresponds to the unstable equilibrium the model is given as follows:

$$\begin{aligned} x_{1,n+1} &= x_{1,n} + T_s x_{2,n} \\ x_{2,n+1} &= \left(1 - \frac{T_s \gamma}{ML}\right) x_{2,n} + \frac{T_s g}{L} \sin\left(x_{1,n}\right) + u_n \end{aligned} \tag{3.21}$$

where

$$T_s = 0.05, M = 0.1,\ L = 0.1,\ g = 10,\ \gamma = 0.05$$

The state-dependent parameterization of the system (3.21) is given as follows:

$$\underline{x_{n+1}} = \begin{bmatrix} 1 & T_s \\ \dfrac{T_s g \sin(x_{1,n})}{L x_{1,n}} & \left(1 - \dfrac{T_s \gamma}{ML}\right) \end{bmatrix} \underline{x_n} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_n \tag{3.22}$$

To avoid a division by zero the $\dfrac{T_s g \sin(x_{1,n})}{L x_{1,n}}$ term in (3.22) is substituted for at $x_{1,n} = 0$ by the limit $\lim_{x_{1,n} \to 0} \dfrac{T_s g \sin(x_{1,n})}{L x_{1,n}} = \dfrac{T_s g}{L}$. The cost function employed in the example is given by the equation (3.11). The following weights and the control horizon were chosen:

$$Q = I,\ R = 1,\ N = 40$$

The boundary condition P_N is obtained from the solution of the discrete state-dependent Riccati equation at the origin. The length of the control horizon is chosen such that the state is driven to zero within that time frame. The following results are obtained. The state trajectories for the DSDRE, predictive DSDRE (Algorithm 3.1) and predictive optimized DSDRE (Algorithm 3.2) are shown in Figure 3-2.

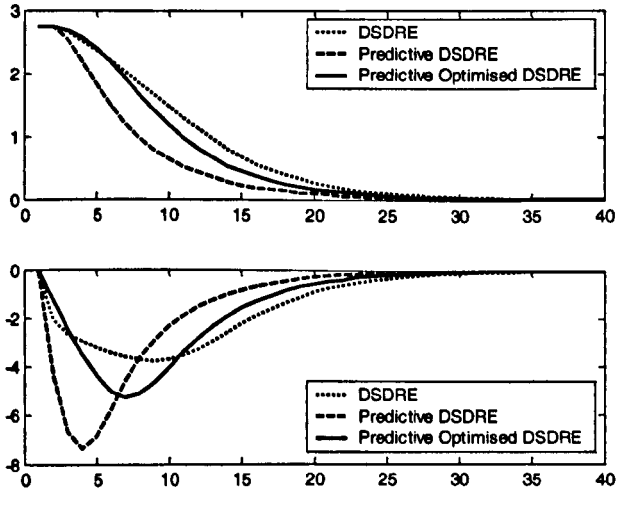

Figure 3-2: State Trajectory for the DSDRE, P-DSDRE, PO-DSDRE

It may be noticed that speed of response is fastest for the predictive DSDRE algorithm. The optimized predictive DSDRE provides slower response and the DSDRE is the slowest. This would suggest that the predictive DSDRE provides the best performance. However, one may measure the performance of the control system with the value of cost function (3.11). This performance index is used for the derivation of the control algorithm. Thus, it is a good indicator of the controller performance. The slower response of the optimized predictive DSDRE algorithm may be explained by the lower control effort. The control effort trajectories are shown in Figure 3-3.

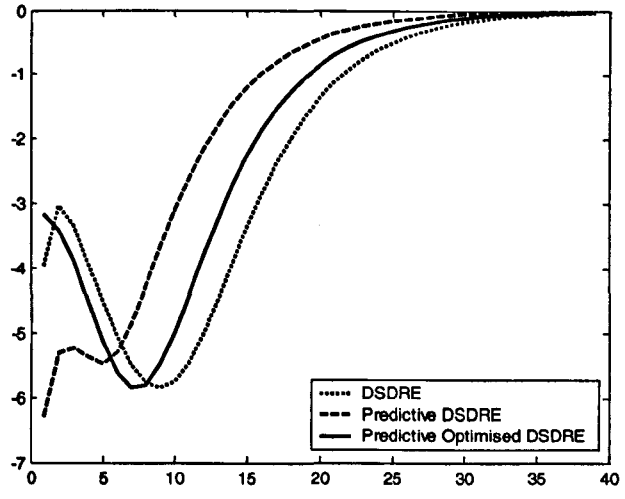

Figure 3-3: Control effort for the DSDRE, P-DSDRE, PO-DSDRE

The value of the performance indexes (3.11) calculated along state and control trajectories for three algorithms are given below.

$J_{\text{DSDRE}} = 557.72$: DSDRE controller,

$J_{\text{P-DSDRE}} = 554.78$: The predictive DSDRE (Algorithm 3.1),

$J_{\text{PO-DSDRE}} = 541.53$: The predictive optimized DSDRE (Algorithm 3.2).

The optimized predictive DSDRE algorithm provides the best performance. The predictive DSDRE and DSDRE algorithms result in higher costs. The system trajectories for three controllers plotted in state-space in Figure 3-4.

The main advantage of the presented non-linear control techniques is the simplicity of the approach. In the steady-state the control law reduces to the optimal control for a time-invariant system, which for small perturbations is sufficient. When there are large reference or disturbance signal changes the control law is evaluated taking into account the future changes in the system parameters brought about by the presence of non-linearities. This is an improvement over the state-dependent Riccati equation method, which assumes the system remains fixed at the nonlinear function values at the time *n*.

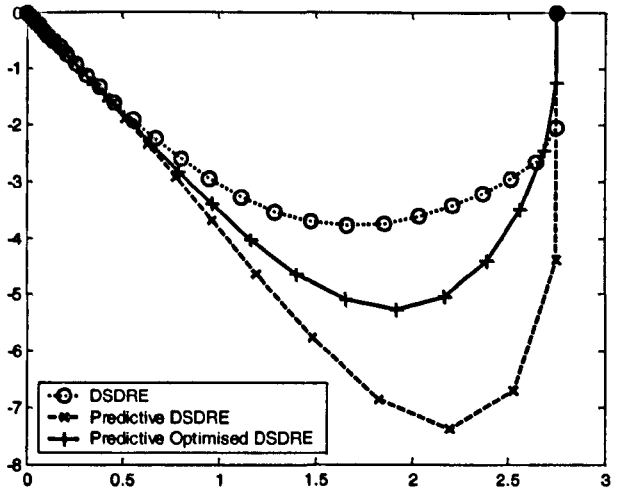

Figure 3-4: State Trajectory for the DSDRE, P-DSDRE, PO-SDRE method in State Space

For most nonlinear control design approaches stability issues are central to the theory and this requires either elegant mathematical results or empirical procedures [64]. The presented approach is optimization based and the focus is more on the performance, under different operating conditions. The analysis of performance is rather easier to achieve, either from operating records, or from theoretical results. Thus, the confidence necessary to encourage the use of the approach is more likely to be achieved by this optimization method. This does not imply that a measure of stability is not important, but it changes the focus of the design onto property, which is easier to measure and benchmark.

The method which uses the predicted trajectory and tensors provides the control law that results in lower values of the cost function. The method is based on the state dependent Riccati equation (SDRE) for the discrete-time non-linear systems. This method does not give an optimal solution to the cost function minimization problem. Originally, the SDRE was developed for the continuous time systems. For these systems, optimality was achievable theoretically, e.g. for the scalar case, or for higher order systems - providing the state dependent representation was selected properly. In the discrete-time case the DSDRE is not guaranteed to provide optimality even in the simplest first-order case.

The reference signal tracking may be achieved by augmenting the reference signal model with the plant model. With the appropriate reference signal modelling the desired time variation of trajectory may be approximated. It was noticed that if the prediction of the future trajectory was refined iteratively at a given time instant, the method not only decreased the value of the cost function. Additionally, the control trajectory converged to the globally optimal control sequence, minimizing the given cost function. This however has been tested only for a limited number of examples and only in simulation.

3.2 Non-Linear Predictive Control

The non-linear predictive control methods presented in this section are based on the state-dependent class of models that were previously used in section 3.1. The underlying idea is similar to the discrete state-dependent Riccati equation algorithm with the predicted trajectory. The main difference is the predictive control context of the method that is utilized in this section.

The model based predictive control (MBPC) algorithms became very popular in recent years and are used extensively in the petrochemical industry for large scale supervisory systems (Richalet et. al. 1978 [65], 1993 [66]). The best known predictive control approach is probably dynamic matrix control (DMC), which was

introduced for complex multivariable plants with strong interactions and competing constraints (Cutler and Remarker 1980 [67]). In Garcia [68] proposed an extension of the DMC to non-linear processes (NLQDMC). In this approach although a non-linear model is used, only a single Quadratic Program is solved on-line. The predictive control algorithms based upon multi-step cost functions and the receding horizon control law, were generalized by Clarke and co-workers in the Generalized Predictive Control (GPC) algorithm (Clarke et. al. 1987 [69], 1989 [70]). The future set-point information has been used in a number of Linear Quadratic (LQ) optimal control problems (Tomizuka and Rosenthal 1979 [71]) and summarized in the seminal work of Bitmead et. al. (1989 [72]). The use of state-space models for Generalized Predictive Control (GPC) was proposed by Ordys and Clarke (1993 [73]).

It is well known that the early predictive control algorithms did not have guaranteed stability property. Several extensions were proposed to overcome this difficulty. Mainly, the idea was to extend the optimization horizon (possibly to infinity). This can be shown to be equivalent to assuming equality (or set) constraints on the final value of the state in the standard, finite horizon, e.g. multistep cost function. In Mosca, Zhang (1992) [74] stability of predictive control was assured by the constraint that the terminal state goes to zero. In 1992 a GPC algorithm with guaranteed closed loop stability was presented by Kouvaritakis et. al. [75]. This approach deploys most of the ideas of GPC but yields control configurations with guaranteed stability. The algorithm first stabilizes the system by a feedback and then the GPC controller is used for optimization in the outer loop.

In Mayne and Michalska (1993) [61] stability of the predictive algorithm was assured by introducing inequality constraint for the state at the end of the prediction horizon. This approach is also called a dual-mode control. After a finite number of steps (equal to the prediction horizon) the state of the system is assumed to reach the neighbourhood of the origin where it is stabilized by the linear feedback law. A similar methodology was used in the previous section in the optimized predictive state-dependent Riccati equation method. The part of the cost function after the

horizon N was replaced by the final state penalty. In Kothare and Morari (2000) [76] the inequality end constraint is introduced and is called the contractive constraint. Comparing to the inequality end constraint presented in [61] no region of attraction W needs to be computed. The contractive constraint is a Lyapunov function for the closed loop system itself and stability can be easily proved.

The approach to the non-linear predictive control employing the optimal control trajectory calculated in the previous time instant of the predictive control algorithm was used by Kouvaritakis et. al. (1999) [77]. The extension of the previous optimal trajectory to the current time instant is referred to as the "tail". The system is linearized around this trajectory and this linearized time varying system is employed to obtain the optimal control, which is calculated as a perturbation from the "tail" trajectory. A similar non-linear control approach with the model linearized about predicted trajectory and the optimization based on generalized predictive control algorithm was presented in [78]. In Lee et. al. [79] the similar methodology employing linearization at points of the seed trajectory is introduced using a discrete time model representation of the system. Asymptotic stability of the algorithm can be guaranteed by the proper selection of the terminal penalty term of the predicted cost.

In the sequel two non-linear predictive regulators will be presented. Both use the remainder of the control trajectory from the previous iteration to compute the time-varying approximation of the non-linear system. The first algorithm, elaborated in section 3.2.1, uses this approximation directly while the second, presented in section 3.2.2, will concentrate on the optimality of the solution. The discussion of the performance will be given in section 3.2.3 based on the academic example. Finally, the non-linear tracking GPC algorithm will be presented in section 3.2.4.

3.2.1 Non-linear predictive regulator

The approach to the non-linear predictive control method presented in this section employs the control trajectory calculated in the previous time instant of the

control algorithm. The technique presented here [13] [14] [15] uses a similar idea to [77], [79], but with the different model representation and optimization technique. The non-linear system described by the discrete time non-linear state space model is re-arranged into the state and control dependent linear form of the state-space model [23], [62]. The non-linear behaviour of the system in such a representation is included in the state and control dependent matrices. Such a system along with the predicted trajectory can be treated as linear and time-varying. The linear GPC control technique can easily be extended to this class of non-linear systems. The cost function is minimized through a static optimization. Note that the state dependent representation is equivalent to the original non-linear system model. It results from an algebraic re-arranging of the original state-space model. For such a model the main source of errors is the difference between the predicted trajectory calculated in previous time instant and the actual one of the object. The difference between the trajectory prediction at current time and the trajectory from the previous time instant results in the model-object mismatch.

The non-linear predictive control algorithm will now be presented. The system is represented by the following difference equation:

$$x_{n+1} = f(x_n, u_n) \tag{3.23}$$

The finite horizon cost function is given by the following expression:

$$J_n = \sum_{i=1}^{N} \left\{ x_{n+i}{}^T \Lambda_E^i x_{n+i} + u_{n+i-1}{}^T \Lambda_U^i u_{n+i-1} \right\} \tag{3.24}$$

The vector form of the cost function is given as follows:

$$J_n = X_{n+1}{}^T \Lambda_E X_{n+1} + U_n^T \Lambda_U U_n \tag{3.25}$$

where

$$X_{n+1}{}^T = \left[x_{n+1}{}^T \quad x_{n+2}{}^T \quad \cdots \quad x_{n+N}{}^T \right], \; U_n^T = \left[u_n{}^T \quad u_{n+1}{}^T \quad \cdots \quad u_{n+N-1}{}^T \right],$$

Λ_U, Λ_E - positive definite symmetric matrices.

The equation (3.23) is re-arranged in the state and control dependent linear state-space representation.

$$x_{n+1} = A(x_n, u_n)x_n + B(x_n, u_n)u_n \tag{3.26}$$

In general, the number of possible linear representations (3.26) of (3.23) is infinite. As a general rule $A(x_n, u_n)$ and $B(x_n, u_n)$ should exhibit as small variations with state and control signals as possible. Also, the pair $A(x_n, u_n)$, $B(x_n, u_n)$ should be controllable in the operating region. Similar rules as for the state-dependent Riccati equation method apply here. Note that the matrices in (3.26) are permitted to depend upon control action. At each iteration of the predictive control algorithm not only the current control action, but the full vector of future control trajectory is computed. In the receding horizon predictive control algorithm only the first element of the control vector is used. The algorithm presented here uses all elements of the vector of future control predictions. The first element is used for the controlled object input manipulation and the remaining elements for the future trajectory prediction.

At the time instant n, the vector $U_{n-1} = \left[u_{n-1}^T, u_{n-1+1}^T, \ldots, u_{n-1+N-1}^T\right]^T$ calculated at previous iteration of the control algorithm is considered. The first element of this vector has already been used, but the remaining part $\left[u_n^T, u_{n+1}^T, \ldots, u_{n+N-2}^T\right]^T$ can be employed to predict the future trajectory. At the time n it is assumed, that current state x_n is available either by direct measurement or from the state estimator. Using the control vector, the state predictions $x_{n+1}, x_{n+2}, \ldots x_{n+N-1}$ are obtained from the model. Using the past control trajectory and the resulting state trajectory, matrices $A(x_n, u_n)$, $B(x_n, u_n)$ are calculated at subsequent points on the trajectory. Note that the control action u_{n+N-1} is not available and its value is assumed to be $u_{n+N-1} = u_{n+N-2}$. Due to what was stated, the non-linear system (3.23) is approximated

by the linear time-varying state-space model with the following matrices: $A_n = A(x_n, u_n), ..., A_{n+N-1} = A(x_{n+N-1}, u_{n+N-1}), B_n = B(x_n, u_n), ..., B_{n+N-1} = B(x_{n+N-1}, u_{n+N-1})$.

The prediction of the future state as a function of current state and future control actions is given by the following equation based on the time-varying approximation:

$$\begin{aligned} x_{n+j} = &[A_{n+j-1}A_{n+j-2}...A_n]x_n + [A_{n+j-1}A_{n+j-2}...A_{n+1}]B_n u_n + \\ &+ [A_{n+j-1}A_{n+j-2}...A_{n+2}]B_{n+1}u_{n+1} + [A_{n+j-1}A_{n+j-2}...A_{n+N_u}]B_{n-1+j}u_{n-1+j} \end{aligned} \tag{3.27}$$

For the time varying linear model the linear relationship between X_{n+1} and U_n may be established:

$$X_{n+1} = F_n x_n + S_n U_n \tag{3.28}$$

where

$$F_n = \begin{bmatrix} A_t \\ A_{t+1}A_t \\ \vdots \\ A_{t+N-1} \cdot ... \cdot A_t \end{bmatrix}, \quad S_n = \begin{bmatrix} B_t & 0 & \cdots & 0 \\ A_{t+1}B_t & B_{t+1} & \ddots & \ddots \\ \vdots & \vdots & \ddots & 0 \\ A_{t+N-1} \cdot ... \cdot A_{t+1}B_t & A_{t+N-1} \cdot ... \cdot A_{t+2}B_{t+1} & \cdots & B_{t+N-1} \end{bmatrix}$$

The future state prediction equation (3.28) is substituted in the cost function (3.25) and the static minimization carried out. The control U_n minimizing (3.25) is finally derived as:

$$U_n = -\left(S_n^T \Lambda_E S_n + \Lambda_U\right)^{-1} S_n^T \Lambda_E F_n x_n \tag{3.29}$$

The control vector is computed based on the current state x_n, matrices S_n, F_n based on the time-varying model approximation and the weighting matrices Λ_U and Λ_E.

3.2.2 Optimal Non-Linear Predictive Regulator

The problem of the minimization of the cost function subject to the non-linear system dynamics was substituted in previous section by the linear time-varying approximation. The optimality of the solution based on the static optimization of the time-varying approximation of the non-linear system may depend on the parameterization employed. In the receding horizon control scheme employed by the predictive control algorithm presented in section 3.2.1, the predicted trajectory derived in the previous iteration may differ from the actual trajectory. This may also affect optimality of the solution. To analyze this, the optimal regulation problem over a finite horizon is considered with respect to its convergence properties. The prediction of the future trajectory is calculated, the time-varying approximation of the model is re-calculated and the updated future trajectory is computed. The time-varying model refinement is repeated until the trajectory based on the time-varying model approximation does not change. In this case the model mismatch between the time-varying approximation and the original non-linear model is completely removed. In fact, if the final time-varying model is compared with the original non-linear model simulated with the derived control sequence, the two are identical. Unfortunately the value of the cost function after that procedure, still, may not be optimal. The optimality depends on the state-dependent form chosen. To recover the optimality, consider an optimization for the non-linear system model (3.23) subject to (3.25) without using a time-varying approximation of the system.

The future state prediction derived from the non-linear system model (3.23) is given by the following equation:

$$X_{n+1}(U_n, x_n) = \begin{bmatrix} x_{n+1} \\ x_{n+2} \\ \vdots \\ x_{n+N} \end{bmatrix} = \begin{bmatrix} f(x_n, u_n) \\ f(f(x_n, u_n), u_{n+1}) \\ \vdots \\ f(f \ldots (f(x_n, u_n), u_{n+1}) \ldots u_{n+N-1}) \end{bmatrix} \tag{3.30}$$

To find the minimum of the cost function (3.25) the derivative with respect to the control vector is calculated. For the extreme point this derivative must be equal zero.

$$\frac{\partial J_n}{\partial U_n} = \frac{\partial X_{n+1}}{\partial U_n}^T 2\Lambda_E X_{n+1} + 2\Lambda_U U_n = 0 \tag{3.31}$$

The derivatives matrix $\frac{\partial X_{n+1}}{\partial U_n}$ in equation (3.31) is defined as follows:

$$\frac{\partial X_{n+1}}{\partial U_n} = \begin{bmatrix} \frac{\partial x_{n+1}}{\partial u_n} & \frac{\partial x_{n+1}}{\partial u_{n+1}} & \cdots & \frac{\partial x_{n+1}}{\partial u_{n+N-1}} \\ \frac{\partial x_{n+2}}{\partial u_n} & \frac{\partial x_{n+2}}{\partial u_{n+1}} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_{n+N}}{\partial u_n} & \frac{\partial x_{n+N}}{\partial u_{n+1}} & \cdots & \frac{\partial x_{n+N}}{\partial u_{n+N-1}} \end{bmatrix}. \tag{3.32}$$

The state at any time instant does not depend on future (and also current) control action. The matrix $\frac{\partial X_{n+1}}{\partial U_n}$ is therefore lower triangular. This results from Bellman's principle of optimality [80]:

$$\frac{\partial X_{n+1}}{\partial U_n} = \begin{bmatrix} \frac{\partial x_{n+1}}{\partial u_n} & 0 & \cdots & 0 \\ \frac{\partial x_{n+2}}{\partial u_n} & \frac{\partial x_{n+2}}{\partial u_{n+1}} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \frac{\partial x_{n+N}}{\partial u_n} & \frac{\partial x_{n+N}}{\partial u_{n+1}} & \cdots & \frac{\partial x_{n+N}}{\partial u_{n+N-1}} \end{bmatrix}. \tag{3.33}$$

For the arbitrary non-linear model and for the arbitrary control horizon the derivative $\frac{\partial X_{n+1}}{\partial U_n}$ is a complicated function of the current and future controls and states. Also,

to solve the equation (3.31) with the definition (3.30) and (3.33) a numerical method would have to be employed since the control vector appears in the problem in an implicit form. For the solution of the problem presented here, it is assumed that the initial guess of the optimal control trajectory is known. This may be a control trajectory calculated at previous iteration of the algorithm if the receding horizon control strategy is employed. The control trajectory is given by the following vector:

$$U_{n,0}^{T} = \left[u_{n,0}^{T} \quad u_{n+1,0}^{T} \quad \cdots \quad u_{n+N-1,0}^{T} \right] \tag{3.34}$$

For this control trajectory, the state trajectory may be obtained easily using the current state and the model.

$$X_{n+1,0} = \begin{bmatrix} f(x_n, u_{n,0}) \\ f\left(f(x_n, u_{n,0}), u_{n+1,0}\right) \\ \cdots \\ f\left(f \ldots \left(f(x_{n,0}, u_{n,0}), u_{n+1,0}\right) \ldots u_{n+N-1,0}\right) \end{bmatrix} \tag{3.35}$$

Using the Taylor series the state X_{n+1} is represented by the following expression:

$$X_{n+1}(U_n) = X_{n+1,0} + \left.\frac{\partial X_{n+1}}{\partial U_n}\right|_{X_{n+1,0}, U_{n,0}} (U_n - U_{n,0}) + \rho_n \tag{3.36}$$

If the initial guess of the optimal trajectory is sufficiently close to the actual optimal trajectory then the higher order terms ρ_n are negligible. In the receding horizon approach this is fulfilled if the prediction horizon is sufficiently long and the control trajectory from the previous iteration differs from the current control trajectory only by a small perturbation. With this assumption the following relationship is established from (3.36):

$$\frac{\partial X_{n+1}(U_n)}{\partial U_n} \approx \frac{\Delta X_{n+1}(U_n)}{\Delta U_n} \approx \left.\frac{\partial X_{n+1}}{\partial U_n}\right|_{X_{n+1,0}, U_{n,0}} \tag{3.37}$$

Using the chain rule for (3.33) the following matrix is defined with the assumption that subsequent control actions are independent (the open loop control within the horizon):

$$S_{L,n}=\left.\frac{\partial X_{n+1}}{\partial U_n}\right|_{X_{n+1,0},U_{n,0}}=\begin{bmatrix} \bar{B}_n & 0 & \cdots & 0 \\ \bar{A}_{n+1}\bar{B}_n & \bar{B}_{n+1} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \bar{A}_{n+N-1}\cdot\ldots\cdot\bar{A}_{n+1}\bar{B}_n & \cdots & \cdots & \bar{B}_{n+N-1} \end{bmatrix} \tag{3.38}$$

where $\bar{A}_k \triangleq \left.\frac{\partial f(x_k,u_k)}{\partial x_k}\right|_{x_{k,0},u_{k,0}}$ $\bar{B}_k \triangleq \left.\frac{\partial f(x_k,u_k)}{\partial u_k}\right|_{x_{k,0},u_{k,0}}$

The $S_{L,n}$ matrix is defined using the predicted trajectory $U_{n,0}$, $X_{n+1,0}$. To find a stationary point the derivative of the cost function (3.31) must be zero. The quadratic cost function (3.25) has a semi-positive and positive definite state and control weighting matrices respectively. If the system dynamics are linear, the minimization of this cost results in a global minimum. For general non-linear system this property does not hold.

The necessary condition for the cost function extremum is $\frac{\partial J_n}{\partial U_n}=0$. From equations (3.30), (3.31) it is clear that the derivative of the cost is a nonlinear function of the control vector. To solve this problem a state and control dependent form of the non-linear system model (3.26) is used. Also, an approximate derivative $\frac{\partial X_{n+1}(U_n)}{\partial U_n}$ resulting from equations (3.37) and (3.38) are employed. Now substitute X_{n+1} from (3.28) in (3.31) and find the extremum of the cost $\frac{\partial J_n}{\partial U_n}=0$:

$$\frac{\partial X_{n+1}}{\partial U_n}^T \Lambda_E F_n x_n + \frac{\partial X_{n+1}}{\partial U_n}^T \Lambda_E S_n U_n + \Lambda_U U_n = 0 \tag{3.39}$$

The control U_n is finally derived as:

$$U_n = -\left(S_{L,n}{}^T \Lambda_E S_n + \Lambda_U\right)^{-1} S_{L,n}{}^T \Lambda_E F_n x_n \tag{3.40}$$

where

$$S_{L,n} \approx \frac{\partial X_{n+1}}{\partial U_n}$$

The optimal control sequence minimizing the finite horizon cost function is obtained in an iterative way. The initial guess of the trajectory is required to be known. It might be a part of the control trajectory from the previous iteration of the predictive control algorithm. The control algorithm follows:

Algorithm 3.3

- Use the initial control trajectory and compute the state trajectory from (3.35).
- Update the time-varying approximation of the model using the trajectory and the state-dependent model description (3.26). Update the $S_{L,n}$, S_n, F_n matrices and calculate the control vector from (3.40).
- Check the difference between the control vector from previous and current iterations. If it is larger then a stop condition the algorithm continues from the step 2 otherwise it returns the final optimal control vector.

Note the difference between the control vector given by (3.29) and (3.40). The equation (3.29) obtained for the predictive algorithm that uses only the time-varying approximation of the non-linear system uses matrices S_n F_n Λ_U Λ_E: $U_n = -\left(S_n{}^T \Lambda_E S_n + \Lambda_U\right)^{-1} S_n{}^T \Lambda_E F_n x_n$. In the equation (3.40) the gradient $S_{L,n}$ is introduced. This provides the corrected search direction and the minimization of the performance index gives the lower cost value. This will be demonstrated on the numerical example in the next section.

3.2.3 Predictive Regulation Numerical Example and Conclusions

As an example a discrete model of the driven pendulum shown in Figure 3-5 is employed.

The control task is to find the optimal control sequence to drive the pendulum from the certain initial level to the steady state (origin) using the minimum control energy which is consistent with minimizing the performance index.

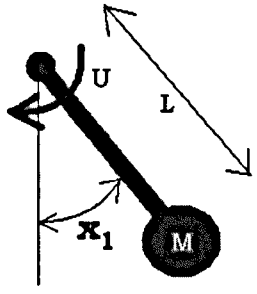

Figure 3-5: Pendulum

The system model is given as follows:

$$\begin{aligned} x_{1,n+1} &= x_{1,n} + T_s x_{2,n} \\ x_{2,n+1} &= \left(1 - \frac{T_s \gamma}{ML}\right) x_{2,n} - \frac{T_s g}{L} \sin\left(x_{1,n}\right) + u_n \end{aligned} \tag{3.41}$$

where

$$T_s = 0.05, M = 0.1, L = 0.1, g = 10, \gamma = 0.05$$

The state-dependent parameterization of the system (3.41) is given as follows:

$$\underline{x_{n+1}} = \begin{bmatrix} 1 & T_s \\ -\dfrac{T_s g \sin\left(x_{1,n}\right)}{L x_{1,n}} & \left(1 - \dfrac{T_s \gamma}{ML}\right) \end{bmatrix} \underline{x_n} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_n \tag{3.42}$$

To avoid the division by zero the $\frac{T_s g \sin(x_{1,n})}{L x_{1,n}}$ term in (3.22) is substituted for $x_{1,n}=0$ by the limit $\lim_{x_{1,n}\to 0} \frac{T_s g \sin(x_{1,n})}{L x_{1,n}} = \frac{T_s g}{L}$. The cost function employed in the example is given by the equation (3.24). The following weights and control horizon were chosen:

$$\Lambda_E^i = I,\ \Lambda_U^i = 1,\ N = 30$$

The length of the control horizon is chosen so that the state reaches zero at the end of it. The following results are obtained. The state trajectory for the non-linear GPC (NLGPC) controller and the optimal NLGPC (ONLGPC) controller are shown in Figure 3-6. The control trajectories are shown in Figure 3-7. The trajectories for both controllers in state-space are shown in Figure 3-8. The trajectory of the ONLGPC controller presented in section 3.2.2 is optimal. The optimal trajectory obtained using numerical optimization (MATLAB *fminunc* function) is identical to the ONLGPC.

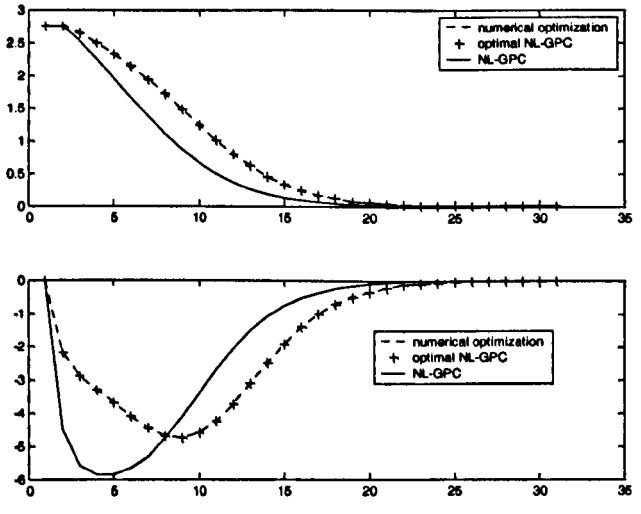

Figure 3-6: State trajectory for the MATLAB fminunc, Optimal NL-GPC and NL-GPC

The values of the cost function for algorithms are:

$J_{\text{fminunc}} = 356.373$: the MATLAB *fminunc*

$J_{\text{ONLGPC}} = 356.373$: the Optimal Non-linear Generalized Predictive Control

$J_{\text{NLGPC}} = 385.217$: the Non-linear Generalized Predictive Control

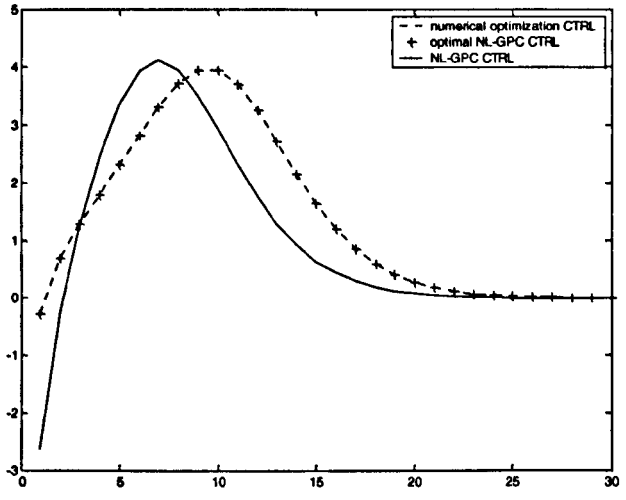

Figure 3-7: Control trajectory for the MATLAB fminunc, Optimal NL-GPC and NL-GPC

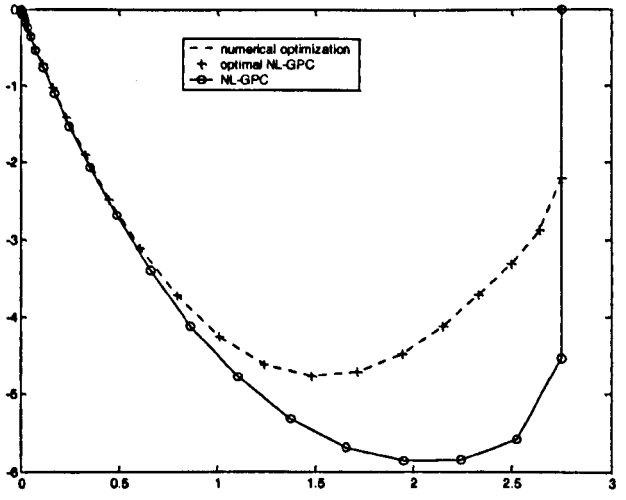

Figure 3-8: State trajectory for the MATLAB fminunc, Optimal NL-GPC and NL-GPC in state space

The predictive control algorithm that provides the optimal control solution for the non-linear systems was presented in section 3.2.2.The algorithm preserves the optimality when the non-linear system is approximated by the linear time varying model. The optimized method may be used for benchmarking of the state-dependent form that would normally be used for the time-varying model approximation (as in section 3.2.1). The cost was compared with the result of the numerical optimization that provided the identical to the optimal non-linear predictive control algorithm solution.

3.2.4 Non-Linear GPC Algorithm with Explicit Reference Trajectory

In sections 3.2.1 and 3.2.2 the predictive control regulator was presented. In this section the explicit setpoint information will be incorporated in the predictive control algorithm [13], [14]. The regulatory algorithm presented there aimed to drive the state to the origin. The non-zero setpoint could be introduced there by changing the coordinates of the system. The predictive algorithm presented in this section aims to track any deterministic reference signal.

The system model is given by the following state-dependent model:

$$x_{n+1} = \tilde{A}(x_n)x_n + \tilde{B}(x_n)u_n \tag{3.43}$$

$$y_n = \tilde{C}(x_n)x_n \tag{3.44}$$

Further, for the state dependent discrete time model (3.43), (3.44) an integral action is incorporated. Instead of control signal in the model (3.43) the control increment is used and the following state space model is obtained:

$$\chi_{n+1} = A(\chi_n)\chi_n + B(\chi_n)\Delta u_n \tag{3.45}$$

$$y_n = C(\chi_n)\chi_n \tag{3.46}$$

where

$$A(\chi_n)=\begin{bmatrix} A(x_n) & B(x_n) \\ 0 & I \end{bmatrix},\quad B(\chi_n)=\begin{bmatrix} B(x_n) \\ I \end{bmatrix},\quad C(\chi_n)=[C(x_n)\quad 0]\quad \chi_n=\begin{bmatrix} x_n \\ u_{n-1} \end{bmatrix},$$

$$\Delta u_n = u_n - u_{n-1}$$

State-space model (3.45), (3.46) matrices are calculated along the predicted trajectory. The time-varying linear model approximation is used for the controller design. The following notation is employed $A_n = A(\chi_n)$ $B_n = B(\chi_n)$ $C_n = C(\chi_n)$. For the state-space model (3.45), (3.46) the following GPC cost function is considered:

$$J_n = \sum_{i=1}^{N_e}\left\{(r_{n+i} - y_{n+i})^T \Lambda_E^i (r_{n+i} - y_{n+i})\right\} + \sum_{i=1}^{N_u}\left\{\Delta u_{n+i-1}{}^T \Lambda_U^i \Delta u_{n+i-1}\right\} \tag{3.47}$$

Where

r_n is a vector of size n_y of the setpoint at time n, $\Lambda_E^i, i=1...N_e$ and $\Lambda_U^j, j=1...N_u$ are weighting matrices (symmetric) and N_e N_u are positive integer numbers greater or equal one.

The following vectors containing current and future values of the control u_n and future values of state x_n, and output y_n are introduced:

$$\begin{aligned} X_{n+1,N_e} &= \left[\chi_{n+1}^T,...,\chi_{n+N_e}^T\right]^T, \Delta U_{n,N_u} = \left[\Delta u_n^T,...,\Delta u_{n+N_u-1}^T\right]^T \\ Y_{n+1,N_e} &= \left[y_{n+1}^T,...,y_{n+N_e}^T\right]^T, R_{n+1,N_e} = \left[r_{n+1}^T,...,r_{n+N_e}^T\right]^T \end{aligned} \tag{3.48}$$

The cost function (3.47) with the notation (3.48) may be written in the vector form:

$$J_n = (R_{n+1,N_e} - Y_{n+1,N_e})^T \Lambda_E (R_{n+1,N_e} - Y_{n+1,N_e}) + \Delta U_{n,N_u}^T \Lambda_U \Delta U_{n,N_u} \tag{3.49}$$

where

$$\Lambda_E = diag(\Lambda_E^1 \Lambda_E^2,...,\Lambda_E^{N_e}),\ \Lambda_U = diag(\Lambda_U^1, \Lambda_U^2,...,\Lambda_U^{N_u})$$

It is possible now to determine the future state prediction. For $j=1...N_e$ the future state predictions may be obtained from:

$$\begin{aligned}\chi_{n+j} &= [A_{n+j-1}A_{n+j-2}...A_n]\chi_n + [A_{n+j-1}A_{n+j-2}...A_{n+1}]B_n\Delta u_n + \\ &+[A_{n+j-1}A_{n+j-2}...A_{n+2}]B_{n+1}\Delta u_{n+1} + [A_{n+j-1}A_{n+j-2}...A_{n+N_u}]B_{n-1+\min(j,N_u)}\Delta u_{n-1+\min(j,N_u)}\end{aligned} \tag{3.50}$$

Note that to obtain the state prediction at time instance $n+j$ the knowledge of matrix predictions $A_n...A_{n+j-1}$ and $B_n...B_{n-1+\min(j,N_u)}$ is required. The control increments after the control horizon are assumed to be zero. From (3.48) and (3.51) the following equation for the future state prediction vector $X_{n+1,N}$ is obtained:

$$X_{n+1,N_e} = \Omega_n A_n \chi_n + \Psi_n \Delta U_{n,N_u} \tag{3.51}$$

where

$$\Omega_n = \begin{bmatrix} I \\ A_{t+1} \\ \vdots \\ A_{t+N_e-1}\cdot...\cdot A_{t+1} \end{bmatrix}, \quad \Psi_n = \begin{bmatrix} B_t & 0 & \cdots & 0 \\ A_{t+1}B_t & B_{t+1} & \ddots & \ddots \\ \vdots & \vdots & \ddots & 0 \\ A_{t+N_e-1}\cdot...\cdot A_{t+1}B_t & A_{t+N_e-1}\cdot...\cdot A_{t+2}B_{t+1} & \cdots & B_{t+N_u-1} \end{bmatrix}$$

From the output equation (3.46) it is clear that

$$y_{n+j} = C_{n+j}\chi_{n+j} \tag{3.52}$$

Combining (3.48) and (3.52) the following relationship between vectors $X_{n+1,N}$ and $Y_{n+1,N}$ is obtained:

$$Y_{n+1,N_e} = \Theta_n X_{n+1,N_e} \tag{3.53}$$

where

$$\Theta_n = diag\left(C_{n+1}, C_{n+2}, ..., C_{n+N_e}\right)$$

Finally substituting in (3.53) for X_{n+1,N_e} with (3.51) the following equation for output prediction vector is obtained:

$$Y_{n+1,N_e} = \Phi_n A_n \chi_n + S_n \Delta U_{n,N_u} \tag{3.54}$$

where

$$\Phi_n = \Theta_n \Omega_n, \quad S_n = \Theta_n \Psi_n$$

The output predictions vector Y_{n+1,N_e} given by the equation (3.54) is combined with the cost function (3.49). The static optimization of the quadratic cost function yields the following control vector:

$$\Delta U_{n,N_u} = \left(S_n^T \Lambda_E S_n + \Lambda_U\right)^{-1} S_n \Lambda_E \left(R_{n+1,N_e} - \Phi_n A_n \chi_n\right) \tag{3.55}$$

All elements of the vector of future control predictions are going to be used by the algorithm. The first element is utilized for the controlled input manipulation, and the remaining elements to predict the future trajectory. At the time instant n, the vector $\Delta U_{n-1,N_u} = \left[\Delta u_{n-1}{}^T, \Delta u_{n-1+1}{}^T, \ldots, \Delta u_{n-1+N_u-1}{}^T\right]^T$ calculated at previous iteration of the control algorithm is considered. The first element of this vector has already been used, but the remaining part $\left[\Delta u_n{}^T, \Delta u_{n+1}{}^T, \ldots, \Delta u_{n+N_u-2}{}^T\right]^T$ can be used to predict the future trajectory. At the time n it is assumed, that current state χ_n is available either by direct measurement or from the state estimator. Using the control vector with the assumption that after the control horizon N_u control increments are zero and that the state-space model is given by (3.45), (3.46) the state predictions $\chi_{n+1}, \chi_{n+2}, \ldots \chi_{n+N_e}$ are obtained. Next the future matrices for the model (3.45), (3.46) are calculated and the resulting Φ_{n,N_e}, S_{n,N_e} matrices are obtained. The control vector $\Delta U_{n,N_u}$ is

computed from (3.55). The above control algorithm consists of the following operations (at time instance n):

Algorithm 3.4

- Measure the current state vector χ_n (or estimate its value)
- Take the vector $\Delta U_{n-1,N_u} = \left[\Delta u_{n-1}{}^T, \Delta u_n{}^T, ..., \Delta u_{n+N_u-2}{}^T\right]^T$ calculated in previous iteration and remove the first element $\Delta u_{n-1}{}^T$, which has already been used in the previous iteration for control. Using this vector get the future state predictions $\bar{X}_{n+1,N_e} = \left[\bar{\chi}_{n+1}{}^T, \bar{\chi}_{n+2}{}^T, ..., \bar{\chi}_{n+N_e}{}^T\right]^T$
- Using the predictions $\bar{X}_{n+1,N_e}$ and the known χ_n calculate the future matrix predictions A_{n+i}, C_{n+i+1} for $i = 0...N_e - 1$ and B_{n+i} for $i = 0...N_u - 1$ and finally obtain $\Phi_{n,N}, S_{n,N}$ matrices
- From (3.55) calculate $\Delta U_{n,N_u}$ and control $u_n = u_{n-1} + \Delta u_n$ for the plant input manipulation

The Algorithm 3.4 summarizes Non-Linear GPC control technique presented in this section. The main difference between the algorithm presented here and ones in preceding sections is the explicit reference trajectory used as an input to the controller. Previous algorithms could be used for tracking, after augmenting the system with the reference signal model. This however does not provide in general a sufficient degree of freedom for the deterministic tracking and a direct use of the reference signal is more useful from the practical standpoint.

3.3 Summary

In this chapter a development of non-linear control algorithms was presented. The algorithms were based on the state-dependent models. Two groups of algorithms were analyzed. The first group uses the linear-quadratic optimal control theory as a

basis and extends its applicability to non-linear systems. The optimality of the solution was also subject of consideration. The second group of algorithms uses the generalized predictive control algorithm as a basis and extends these control techniques to the non-linear systems using state-dependent models. Depending on control system requirements the basic non-linear or their optimal versions may be used. For control systems where the cost function and weighting matrices are introduced as tuning parameters, the optimality of the solution may not be of interest. However, for some applications where the cost function determines some meaningful energy used in the process, achieving the minimum value may be of interest. In such cases, the optimal versions of the presented algorithms should be used. The following chapters will present how the theory developed in this and previous chapters may be used in practice in the automotive application.

Chapter 4

Identification of Combustion Engine model

In this chapter the combustion engine modelling and identification is considered. The modelling and identification may follow several paths as described in the introduction in Chapter 1. The white-box modelling for a system of high complexity is extremely difficult. Such models have a limited applicability for control system design and will not be considered here. The accurate physical modelling often results in distributed parameters models that are of limited use for control design. At the beginning of this chapter, in section 4.1, the comparison of two main modelling philosophies for control purpose will be analyzed.

The methodology introduced in this thesis is aimed at using as few parameters as possible. Also, only the data collected during the driving cycle is used for modelling. Consequently, the identification procedure can naturally be adopted for on-line engine operation. This in turn will allow model adaptation to gradual parameter variations. Associated engine modelling was presented in the literature many times over the past two decades. The mean value engine models are regarded as sufficient for control purpose [82] [83] [84] and this type of model will be used in this thesis.

The engine model identification based on the driving cycle data was a subject of research in [85]. This however presented the model for the idle speed control only. This implies that only a restricted range of engine speeds and loads was considered. Event based sampling will be employed throughout this thesis which has a numerous advantages over time-based sampling [86], [87], [89]. There are also disadvantages associated with the main system noise resulting from the engine pumping

fluctuations [88]. The analysis of advantages and disadvantages of the event based sampling was also carried out in [89].

The identification task presented in this chapter is split in two distinct parts: i) intake manifold model identification (air charge model) using upstream engine sensor information from the driving cycle data (sections 4.2 and 4.3) and, ii) fuel path identification using the measured air-fuel ratio from the driving cycle data (section 4.4). Additionally an approximate engine torque model is identified, where it is assumed that the air-fuel ratio is maintained at around the stoichiometric value (section 4.5). The air-fuel models developed have been validated using three different sets of criteria: an integrated absolute, integrated squared error and a correlation between the measured and estimated variables. Depending on the complexity of the model structure selected, various measures of accuracy are developed and presented. Good model accuracy was achieved as more measured variables and model parameters were incorporated in the model structure. These measures include the transient as well as steady state errors in the air-fuel ratio model during the FTP (Federal Test Procedure) driving cycle. These models are intended for predictive feedforward fuel control and subsequent vehicle testing presented in Chapter 5.

As already mentioned, it is assumed that the identification procedure presented in this chapter uses only the driving cycle data. The driving data is collected during the test driving sequence. In this chapter the FTP driving cycle data collected from Chevrolet Corvette with V8 5.7L engine was used. Due to the engine operating cycle and a significant transport delay in the exhaust manifold, the actual air-fuel ratio is measured long after the fuel injection is completed. Therefore, accurate modelling of the engine forward path is of great importance for accurate air-fuel ratio control. The delayed measurement of the actual air-fuel ratio imposes an inherent limitation in the maximum achievable performance using any classical feedback control method. The performance may, however, be improved through extensive use of accurate models in a feedforward control loop.

4.1 Model analysis for air-fuel ratio and torque control: grey-box vs. black-box approaches

The identification of the engine using 'black-box' and 'grey-box' models is now considered. The term 'grey-box' represents parameter estimation methods with the model derived from physical principles. Some simplifying assumptions are usually made during the development. The 'grey-box' is therefore a combination of the 'black-' and 'white-box' models.

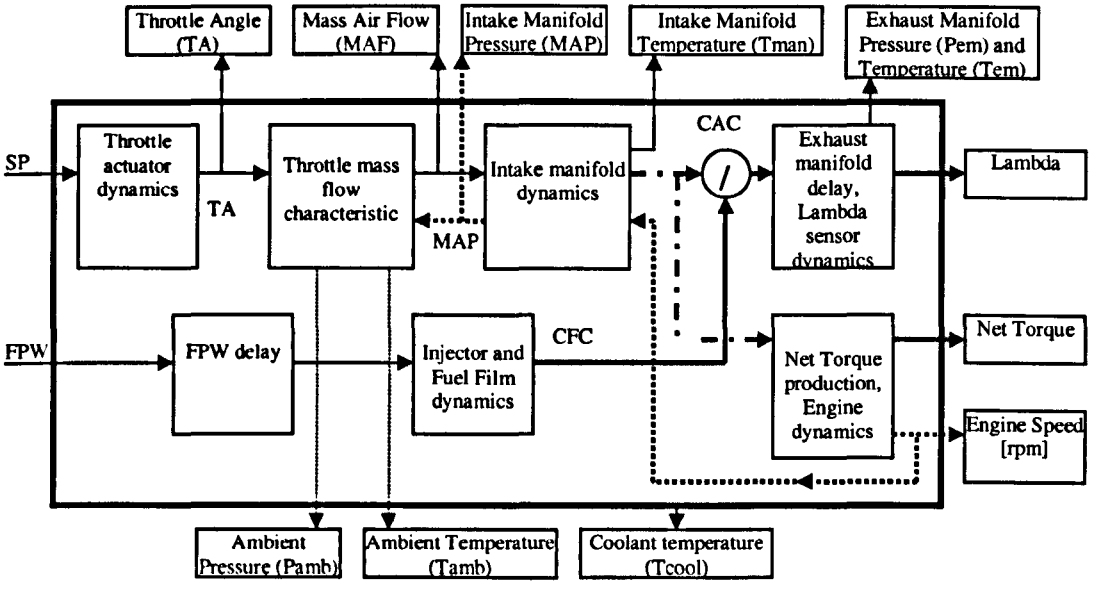

Figure 4-1: System diagram

The structural block diagram of the spark ignition engine model with the relevant measured signals is shown in Figure 4-1. The inputs to the model are the throttle angle setpoint (SP) signal and the fuel pulse width (FPW) command. For the throttle sub-system, the indicated throttle angle (TA), mass airflow rate (MAF), ambient pressure (Pamb) and temperature (Tamb) are measured. For the intake manifold, intake manifold pressure (MAP) and intake manifold gas temperature (Tman) are available. In the exhaust manifold, gas pressure (Pem) and gas temperature (Tem) and exhaust air-fuel ratio (commonly referred to as AFR or lambda) are measured. Additionally, engine torque on the crankshaft is directly

measured. The engine speed and the coolant temperature measurements are also available.

For engine identification purposes, it is important to determine the type of model structure (i.e. black- or grey-box). First, we analyze the conventional control strategy with feedback and feedforward control. This strategy considers the air-fuel ratio as the only controlled parameter. The 'Throttle mass flow characteristic', 'Intake manifold dynamics' and the 'Injector and Fuel Film dynamics' blocks must be identified as separate elements. In the conventional control strategy, torque is commanded by the driver through the pedal position and is controlled indirectly by the throttle position. For the air-fuel ratio this control action is regarded as a disturbance. Consequently the feedforward command is a function of the throttle position as well as other upstream (e.g. MAF, MAP) engine parameters. The blocks mentioned earlier are important for the feedforward control. The cylinder air charge estimate that is the output of the 'Intake manifold dynamics' block is important for the control strategy. To identify the model of cylinder air charge (CAC) its measurement is required. This however is not possible and the only way of obtaining CAC is to use a physical model of the intake manifold. This however implies that some physical insight is required and the grey-box modelling method must be used. Lack of accurate measurement of important engine parameters is an important issue for black-box identification methods. For grey-box methods internal parameters represent physical quantities. The cylinder air charge may be determined from the mass and energy balance established for the intake manifold and the identification is based entirely upon other measured parameters (i.e. MAP, Tman, MAF). The black-box identification considers only input-output relationships and its internal states or model coefficients have no physical meaning. The cylinder air charge is not directly measurable and consequently cannot be modelled in a pure black-box structure.

The other problem for black-box modelling is associated with the fact that the lambda measurement represents both the air and fuel path. The non-linearity associated with the division (ratio) may cause problems during identification. This is due to changes in the operating point following changes in engine states over the

driving cycle. For the grey-box identification method the intake manifold is identified separately with the CAC being one of the physically modelled outputs. The lambda measurement is then used only for the identification of the fuel delivery model. This will be demonstrated in the later parts of this chapter. Since CAC is determined from only upstream measurements, the additional non-linearity associated with the fact that lambda represents a ratio of two unknown variables does not pose any problem. The CAC may be used as a parameter in the identification of the fuel delivery and lambda measurement paths. In the black-box modelling approach, separate models for the intake manifold and fuel dynamics will not be identified effectively. This results from the fact that black-box methods aim to model input-output relationships only and the use of lambda is not sufficient for proper identification of air and fuel parameters.

This problem with black-box methods is not so apparent for multivariable control where both throttle position and the fuel pulse width are used for the air-fuel ratio and torque control. For the multivariable controller the full model with 'Throttle actuator' and 'Net torque production' is required. The throttle must be available as a manipulated input (drive-by-wire) and the accelerator pedal position may be used as the setpoint for the torque. In the conventional engine control structure, a feedforward controller is used to compensate for throttle position changes introduced by the driver. In the multivariable control framework, the throttle position is manipulated by the controller and the feedforward action may not be required. The necessity for the feedforward controller to be present in the system is determined by the type of multivariable controller being used. For that reason fuel and air paths do not have to be separated during modelling.

For multivariable control purposes (i.e. torque and air-fuel ratio control) the input-output non-linear model, in general, should be sufficient. The non-linear black-box modelling techniques such as neural networks or neuro-fuzzy techniques may also be considered. The use of such models in control design is the subject of separate studies outside the scope of this thesis. The estimation of the model parameters from the driving data may be a challenging task. This is due to fast

changes of the operating point and significant nonlinearities encountered in the system. The excitation of non-linear dynamics is also relatively deterministic and may not be sufficient to guarantee a convergence. The linear model structures will not be sufficient, especially if the driving cycle data has to be used. For fully non-linear model structures lack of physical meaning makes the analysis of its global validity difficult and only the validation through simulation may be possible. Also, model parameter identification and convergence is expected to be a real problem. Again, lack of physical insight will not give any indication on what initial parameter values should be used during the identification phase. The black-box non-linear modelling approach will not be attempted in this chapter. The difficulties with changing operating point and the need for a physical model to extract the internal (not measured) variables indicate that the grey-box techniques are advantageous.

The black-box model identification methods could, however, be used with the data collected in a controlled experiment. Many problematic issues, like time delays, would be resolved, since these in steady state are almost constant and may easily be removed from the data. Also the cylinder airflow in steady state is equal to the throttle flow rate and cylinder flow characteristics may easily be obtained. The required perturbations for identification may be added to system inputs to obtain all parameters of interest. The system identification based on the driving cycle data is by far more complex. The engine operation includes fast transients and the identification algorithm must cope with these fast changes. The identification based on the driving cycle data imposes some limitations on the complexity of the system model employed for the identification. However, the important benefit must not be overlooked. If it is possible to identify the system off-line from the driving cycle data, the same may be performed on-line. The identification method introduced here combined with the model-based control techniques will provide better control accuracy for the benefit of reduced emissions and improved performance. In section 4.2 the identification of engine models starts with the throttle actuator dynamic model.

4.2 Throttle model

The identification of the throttle model is presented in this section. The model for the electronic throttle actuator dynamics and for the flow rate as a function of the throttle position will be given.

4.2.1 Throttle actuator dynamic model

The throttle is controlled by its local controller at higher than the main (engine control system) sampling rate. The input to the throttle actuator (or Electronic Throttle) is the setpoint command supplied either by the driver or the engine controller (PCM). The actuator model is identified from the driving cycle dataset. The FTP driving cycle data was used for the analysis presented in this section. For this study, the structure of the model is assumed to be linear. The order of the linear model is chosen after evaluation of the validation result. The linear model is only an approximation of the real non-linear characteristic of the throttle, but its accuracy is sufficient for control design purposes. For this purpose, the model of the throttle dynamics must be discretized according to the current sampling rate that is inversely proportional to the engine speed. If on the other hand the identification process first identified the continuous time model which then was discretized with the variable sampling rate, the identification of the continuous model parameters would certainly be difficult. This is due to the fact that the data used for the identification is provided at discrete events and the temporal sampling rate varies.

An alternative method is to re-sample the date with a fixed sampling rate using interpolation between the original sampling events. The re-sampled signal is used for identification of the discrete model. The identification is carried out using the ordinary Least Squares method. The discrete model parameters for the sampling rate at which the event-based signal was re-sampled is identified. The following second order discrete time model structure was chosen for the identification:

$$TPS_{n+1} = -p_1(N_n)\cdot TPS_n - p_2(N_n)\cdot TPS_{n-1} + p_3(N_n)\cdot SP_n + p_4(N_n)\cdot SP_{n-1} \qquad (4.1)$$

where

TPS_n - the indicated throttle position [V]

SP_n - the throttle setpoint (input to the ET) [V]

$p_1(N_n), p_2(N_n), p_3(N_n), p_4(N_n)$ - discrete model parameters at given engine speed

N_n - engine speed [rpm]

The least squares parameters regression is repeated for a number of different engine speeds. As a result a number of discrete time models is obtained. For the control algorithm, the nearest discrete time model parameters (via a lookup table) will be used. This will be done according to the current engine speed and the sampling rate that is given by $T_{s,i} = 15/N_i$ for 90 degrees sampling event.

The alternative approach of using the continuous time model would require on-line discretization of the continuous dynamics at the current engine speed.The location of the poles and zeros for the models identified for sampling rates resulting from engine speed between 200 and 3100 rpm (i.e. 15/RPM) are shown in Figure 4-2 and in Figure 4-3.

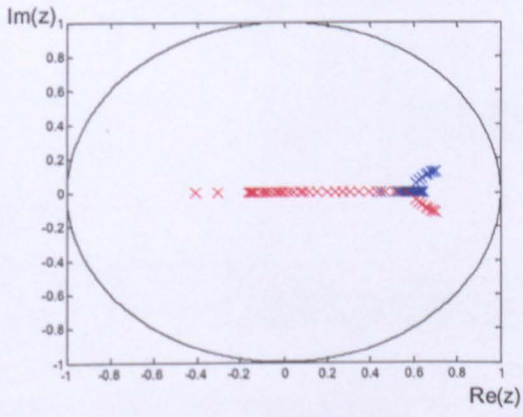

Figure 4-2: Discrete-time model poles locations

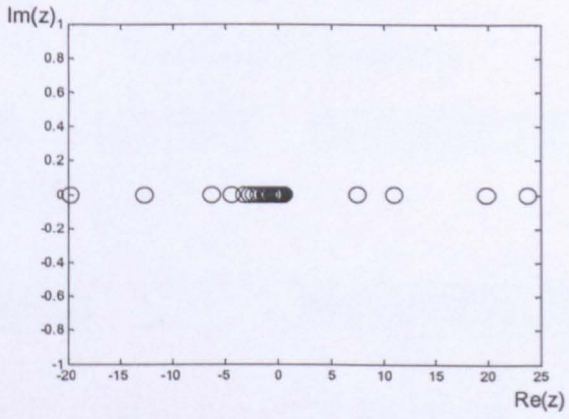

Figure 4-3 Discrete-time model zeros locations

The discrete time dynamics change with the engine speed. This is a result of the continuous time dynamics combined with the variable event-based sampling.

More accurate modelling of the throttle actuator requires a detailed knowledge of the internal controller structure. Also, additional internal signals and measurements sampled at a rate higher than the event-based rate may be required for that purpose. The model validation carried out against the measured throttle position response during FTP driving cycle is shown in Figure 4-4. The model parameters $p_1(N_i), p_2(N_i), p_3(N_i), p_4(N_i)$ are given only at discrete values of the vector $N_{thr} = [200, 250, ..., 1200, 1300, ..., 3100]$. The nearest value is used for engine speeds different to the discrete values of N_{thr}.

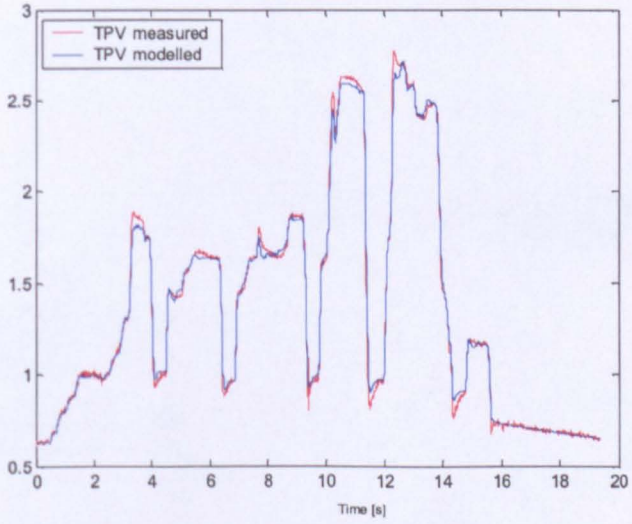

Figure 4-4: Throttle actuator model validation

4.2.2 Throttle flow rate model

The throttle flow rate model is aimed to capture the relationship between the indicated throttle position (obtained from the throttle position measurement) and the mass air flow through the throttle. The angle between the closed throttle plate

position and the actual position will be denoted as α. The angle α is derived from equation (4.2) as a linear function of TPS_n. In this way the output of the position sensor given in volts is scaled into radians.

$$\alpha = k_{Thr} TPS_n + O_{Thr} \tag{4.2}$$

where

k_{Thr}, O_{Thr} - constant parameters

The throttle body model structure is assumed to be given by the one-dimensional isentropic compressible flow equation for flow across the orifice [90]. For non-choked flow the equation is given by:

$$\dot{m}_{at} = C_d \cdot A_{th}(\alpha) \cdot \frac{P_a}{\sqrt{R_{air} \cdot T_a}} \cdot \left(\frac{P_{im}}{P_a}\right)^{\frac{1}{\kappa}} \sqrt{\frac{2\kappa}{\kappa - 1}\left[1 - \left(\frac{P_{im}}{P_a}\right)^{\frac{\kappa-1}{\kappa}}\right]} \tag{4.3}$$

where:

$\dot{m}_{at}$ - throttle mass flow rate

$C_d = C_d\left(P_{im}/P_a, \alpha\right)$ - discharge coefficient

$A_{th}(\alpha)$ - throttle cross-sectional area

P_a - upstream pressure (ambient)

T_a - upstream temperature (ambient)

P_{im} - downstream pressure (intake manifold)

κ - ratio of specific heats for dry air

R_{air} - ideal gas constant for dry air

For choked flow or when:

$$\frac{P_{im}}{P_a} \leq \left(\frac{2}{\kappa + 1}\right)^{\frac{\kappa}{\kappa-1}} \tag{4.4}$$

the throttle mass flow is given by the following equation:

$$\dot{m}_{at} = C_d \cdot A_{th}(\alpha) \cdot \frac{P_a}{\sqrt{R \cdot T_a}} \cdot \sqrt{\kappa} \left(\frac{2}{\kappa - 1} \right)^{\frac{\kappa+1}{2 \cdot (\kappa-1)}} \tag{4.5}$$

The cross-sectional area $A_{th}(\alpha)$ is a function of the throttle body dimensions and the angle between closed and current throttle position. In a very simplified form it may be given by the following equation:

$$A_{th}(\alpha) = \pi \cdot R_{th}^2 \left(1 - \frac{\cos(\alpha + \alpha_0)}{\cos(\alpha_0)} \right) \tag{4.6}$$

where

R_{th} - radius of the throttle

α_0 - throttle offset angle (minimum throttle angle)

α - throttle angle

4.2.2.1 Air flow sensor

Air flow sensor (MAF) may be modelled by the first order dynamic system:

$$MAF = \frac{1}{sT_{MAF} + 1} \dot{m}_{at} \tag{4.7}$$

where:

MAF - throttle mass flow measurement

T_{MAF} - MAF sensor time constant

$\dot{m}_{at}$ - throttle mass flow rate

The MAF sensor is sampled internally at a higher rate than the event based sampling rate. The measurements are averaged over the sampling time and that value is read at this lower rate (every event). The overall dynamics seen on that output may be modelled by (4.7). The typical time constant of the MAF sensor is $T_{MAF} = 5[ms]$. The output of that sensor is sampled every engine event (i.e. for 8 cylinder engine every 25ms for 600RPM). Under these circumstances dynamics of that sensor cannot be identified from the data (the response of the sensor is too fast relative to sampling rate). To be able to identify the sensor dynamic model, data would have to be sampled with the shorter sampling period (by the rule-of-thumb at least five times faster than the time constant i.e. typically 1[ms]). With the time constant of the MAF sensor available from a separate experiment, it is possible to reconstruct the actual flow rate using the inversion of the model in (4.7). Note however the dynamics of this sensor may be neglected for the typical driving pattern without loss of accuracy. This was confirmed during the identification experiment. With that assumption parameters of the model (4.3) will be identified with $\dot{m}_{at} \cong MAF$.

4.2.2.2 Air flow parameter estimation

The physical dimensions (radius of the throttle and throttle offset angle) may easily be measured and therefore are known for the identification. Note, that any inaccuracies of these measurements will be accounted for in the identified discharge coefficient map $C_d = C_d(P_{im}/P_a, \alpha)$. This coefficient is a function of P_{im}/P_a and α. The structure of this dependence is very complex and in practice is modelled by the lookup table. The inaccuracies of the equation (4.6), which is a simplified version of the exact relationship for the cross section area of the throttle, are also compensated by the C_d coefficient. The procedure of the lookup table construction will be explained in the sequel. The dynamics of the MAF sensor are neglected and only the non-linear static model (4.3) or (4.5) which describes the mass air flow through the throttle as a function of the upstream pressure (ambient), upstream temperature (ambient), downstream pressure (intake manifold) and the throttle angle

is considered during parameter estimation. Measurements of all these parameters are available in the vehicle dataset.

The value of the discharge coefficient may easily be calculated from the flow equations (4.3), (4.5) based on the measurements. The calculated discharge coefficient value for the given throttle position and the pressure ratio is used for the lookup table construction. The result of direct calculation is contaminated by the measurement noise contributed by all parameters used for the calculation. However, the method employed for the lookup table construction uses least squares parameter fitting that efficiently removes the noise. It is assumed that the noise signal is white. As was mentioned earlier the discharge coefficient C_d is not constant and is a function of the throttle position and the pressure ratio. The discharge coefficient resulting from the calculation is a function of other measured states. The relationship between the C_d and α, P_{im}/P_a is of unknown structure, therefore the lookup table is employed for our modelling purposes. This is an example of a grey-box modelling technique. The physical principles model structure derived from (4.3), (4.5) is complemented by the black-box type of model implemented as a lookup table. Lookup table gives the relationship of a variable as a function of k underlying parameters only at its discrete levels.

For the discharge coefficient C_d a two-dimensional lookup table is used. Values of time-varying parameter $C_{d,n}$, in general, are not available at desired grid values of α_n and $P_{im,n}/P_{a,n}$ which are used as coordinates of the lookup table. The example time-varying series for $C_{d,n}$, α_n and $P_{im,n}/P_{a,n}$ is shown in Figure 4-5. The method of constructing the lookup table is now explained. The $C_{d,n}$ data is given as a function of the two dimensional pressure ratio - throttle angle space (Figure 4-6). This space is partitioned, but the important issue is how the grid is chosen. The following has to be ensured: for regions where change of the value of parameter is significant the grid density must be higher. Also, the density of the data available has to be sufficient for the purpose. Increased density of the grid requires sufficient data density. The density of the data available in the FTP driving pattern is shown in

Figure 4-6. The rate of change of the discharge coefficient may be observed there. The data around one lookup table grid point is shown in Figure 4-7. A visible amount of noise is evident there. This however will be filtered out by the regression procedure described in the sequel.

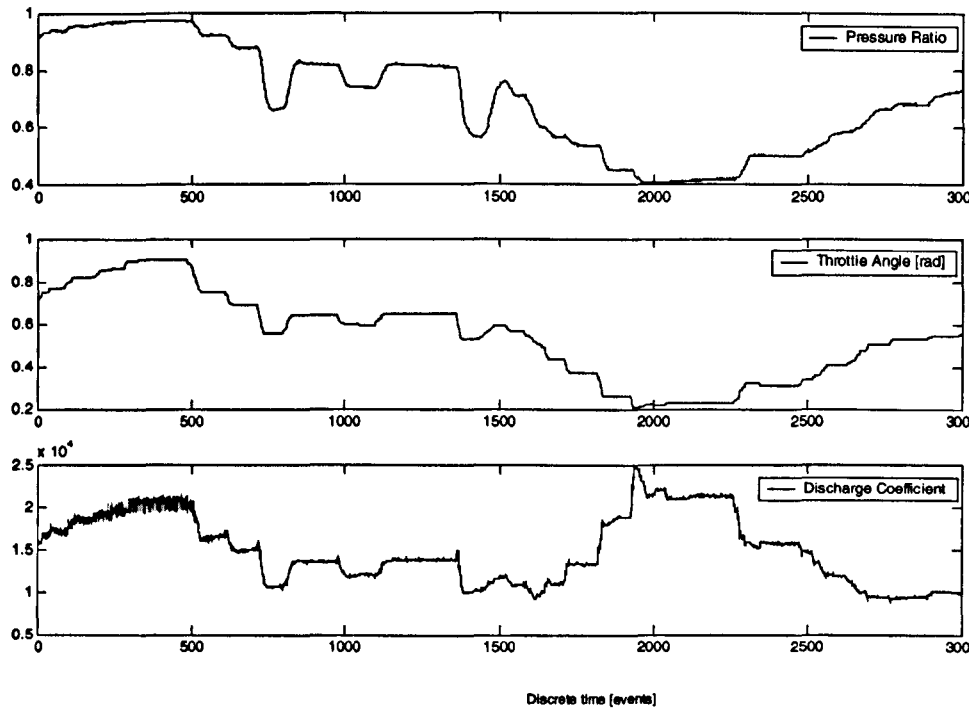

Figure 4-5: Time-varying trace of discharge coefficient, throttle angle and pressure ratio

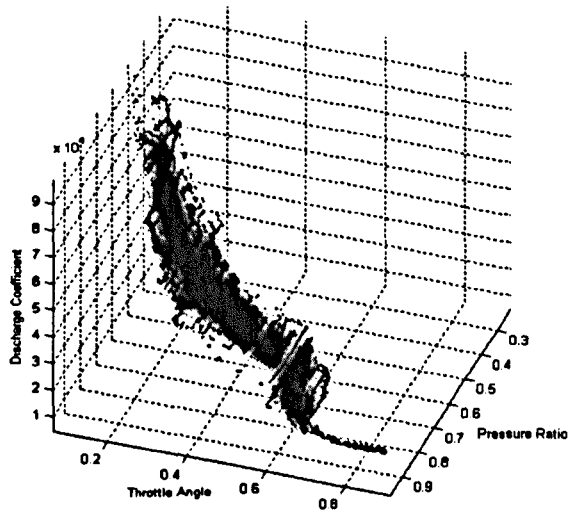

Figure 4-6: Data points in Lookup-Table coordinates for the FTP driving cycle

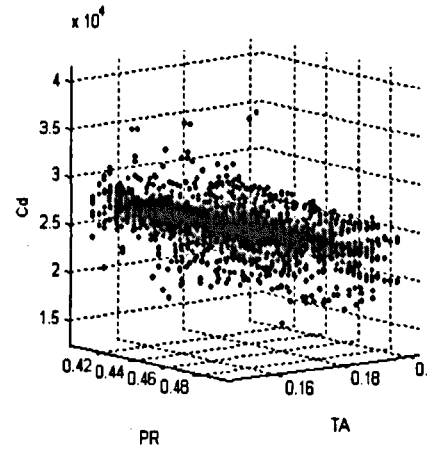

Figure 4-7: Example limited data range used for the modelling

The way the data points (α_n, $P_{im,n}/P_{a,n}$, $C_{d,n}$) shown in Figure 4-6 in (α_n, $P_{im,n}/P_{a,n}$) are used for the lookup table construction is now considered. For the calculation of the encircled parameter of the lookup table shown in Figure 4-8 the data points that fall into clusters 1-4 are considered.

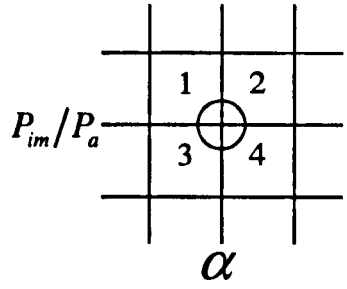

Figure 4-8: Lookup table construction

The discharge coefficient $C_{d,n}$ data point for α_n and $P_{im,n}/P_{a,n}$ is associated with one of clusters. If, for the point of interest (encircled in Figure 4-8), data is available only for one cluster (1 or 2 or 3 or 4) then the lookup table entry for these coordinates cannot be computed. If data is available in all clusters (1,2,3,4) then quadratic surface equation is computed using the regression by the Least Squares. The lookup table entry is calculated from the surface equation. Additionally, the minimum number of points required for reliable regression is defined for each cluster. If number of data points within a cluster is below this threshold the available data is discarded. If data-points are available at least in either 1 and 4 or 3 and 2 (also the combination e.g. 1, 2, 3 falls into this category) the plane equation parameters are fitted in the data. Similarly to the quadratic surface case, the lookup table entry is calculated from the plane equation.

By following this methodology subsequent discharge coefficient values at grid points in the lookup table are computed. For some lookup-table coordinates there is an insufficient amount of data available (observe the FTP cycle data in Figure 4-6). For these points the available lookup-table entries are extrapolated. The

extrapolation procedure is as follows. First borders of the data points that are available in the lookup-table are found. Grid points which are adjacent to the border which were not computed due to insufficient amount of measurements are now computed. The mask is applied (grey area in Figure 4-9) that defines that only the data in a limited distance from the point of interest is used. Black points in Figure 4-9 indicate data points that are available, white denote points at which there is no data in the lookup table.

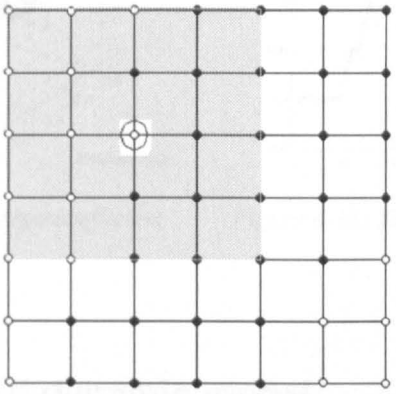

Figure 4-9: Extrapolation mask

Following the procedure all empty grid points in the lookup table are filled up using available entries which are extrapolated. Either a plane or quadratic surface is fitted in the available lookup table data entries. The equation of the fitted surface is used for the calculation of the discharge coefficient being a function of throttle angle and pressure ratio coordinates. This operation is applied for all points that are adjacent to the lookup-table available entries. It is repeated until the lookup table is filled with the data entries.

4.2.2.3 Throttle flow model validation

The discharge coefficient C_d is modelled by the lookup table shown in Figure 4-10. The lookup table is obtained using the method described above. The

lookup table was built using three FTP datasets. The validation of the model against the throttle flow measurement is shown in Figure 4-11.

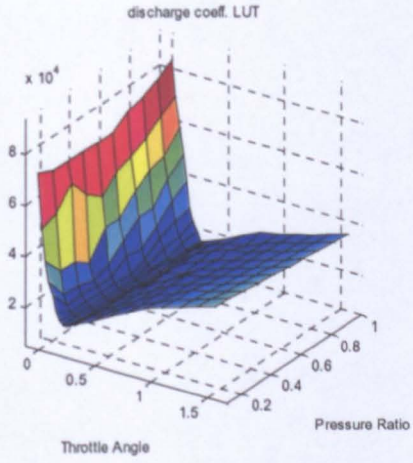

Figure 4-10: Throttle discharge coefficient lookup table

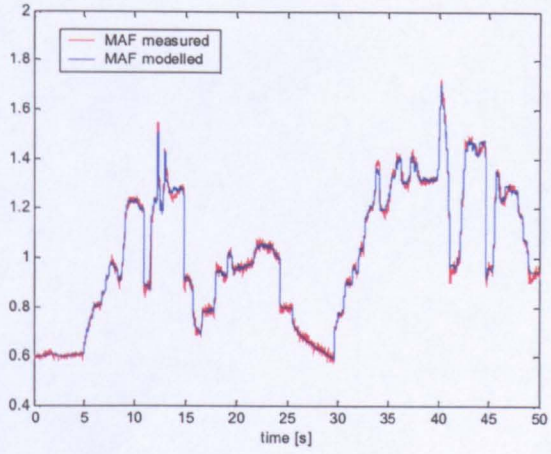

Figure 4-11: Throttle model validation

4.3 Intake manifold dynamic model

The intake manifold is represented by two types of models. The first employs only the mass conservation law. It is widely accepted that it gives a high degree of accuracy for the intake manifold modelling. The second type employs both: mass and energy conservation law and, is by far, more non-linear than the first but offers better accuracy. The intake manifold model is built based on the physical laws. However, the volumetric efficiency – the parameter of the model – is a non-linear function of a number of engine variables with an unknown structure. It is modelled by a lookup table similar to the representation of the throttle discharge coefficient. This again defines the grey-box nature of the modelling procedure employed here.

4.3.1 One-state model analysis

The one state intake manifold parameters are identified in this section. As was mentioned before, the model structure is a hybrid of the known non-linear physical and the unknown black-box types of relationship. The black-box part of the

model – the volumetric efficiency requires its dependence to be defined. The volumetric efficiency is known to be a complicated function of the number of engine variables. It will therefore be modelled with the lookup-table, a method typically used for this parameter.

4.3.1.1 The model

In this section the identification of the one-state intake manifold is carried out. Assuming that the temperature is slowly varying and using the ideal gas law the intake manifold pressure time derivative is given by the following equation:

$$\dot{P}_{im}(t) = \frac{R_{air} T_{im}(t)}{V_{im}} \left(\dot{m}_{at}(t) - \dot{m}_{ac}(t) \right) \tag{4.8}$$

where:

$\dot{P}_{im} = \frac{\partial P_{im}}{\partial t}$, P_{im} - intake manifold pressure [kPa]

T_{im} - intake manifold temperature (assumption: perfect mixing) [K]

$\dot{m}_{at}$ - air flow rate through the throttle [g/s]

$\dot{m}_{ac}$ - air flow rate into the cylinder [g/s]

R_{air} - gas constant $[J/gK]$

V_{im} - intake manifold volume

The cylinder (valve) flow rate may be modelled by the following speed-density equation.

$$\dot{m}_{ac}(t) = \frac{V_d}{120 R_{air} T_{im}(t)} \eta \left(P_{im}(t), N(t) \right) P_{im}(t) N(t) \tag{4.9}$$

where:

V_d - engine displacement [dm^3]

$\eta(P_{im}, N) = \eta$ - volumetric efficiency [-]

N - engine speed [rpm]

The continuous time model of the intake manifold obtained as a combination of (4.8) and (4.9) follows:

$$\dot{P}_{im}(t) = -\frac{V_d N(t)\eta}{120 V_{im}} P_{im}(t) + \frac{R_{air} T_{im}(t)}{V_{im}} \dot{m}_{at}(t) \tag{4.10}$$

Note that the dynamics of that model strongly depend on engine speed. The model (4.10) discretized with the event based sampling rate is given by the following equation.

$$P_{im,n+1} = \left(1 - \frac{V_{cyl}}{V_{im}} \eta_n\right) P_{im,n} + \frac{R_{air} T_{im,n} T_{s,n}}{V_{im}} \dot{m}_{at,n} \tag{4.11}$$

where:

$V_{cyl} = \frac{V_d}{8}$ - cylinder displacement [dm^3]

$T_{s,n} = \frac{0.25[rev] \cdot 60[s/\min]}{N_n[rev/\min]}$ - sampling period

The discrete model (4.11) pole location does not depend upon the engine speed. This is a very important feature of the event sampled discrete description of the intake manifold. The time-based discretization of (4.10) would result in a model with the root of the characteristic equation approaching the unit circle as engine speed increases.

4.3.1.2 Identification of model parameters

Equation (4.11) contains the following parameters:

- The constant parameters are V_{im}, V_{cyl}. Those parameters are typically known, but the estimation of the 'effective' values may be carried out.
- The intake manifold air charge per event $T_s \cdot \dot{m}_{at}$ is the input of the system and is measured.
- The pressure P_{im} is the measured output of the system.
- The temperature of the intake manifold T_{im} is measured and assumed as a given parameter for the one-state model.
- The volumetric efficiency is an unknown function of the intake manifold pressure and engine speed. It is modelled by a table lookup. The volumetric efficiency lookup table is to be identified. Assuming that all constant parameters (i.e. V_{im}, V_{cyl}) and the input ($T_s \cdot \dot{m}_{at}$) and the output (P_{im}) are known, the volumetric efficiency $\eta(P_{im,n}, N_n)$ may easily be calculated as at each discrete event n. The lookup table is built using the method described in section 4.2.2.

Ideally, the constant parameters V_{im}, V_{cyl} should be available. However, these are not necessary for the identification. The identification becomes slightly more complex if these physical quantities are not known. The cylinder displacement V_{cyl} and the volumetric efficiency η in the model (4.11) may be substituted by one variable η_0:

$$P_{im,n+1} = \left(1 - \frac{\eta_{0,n}}{V_{im}}\right) P_{im,n} + \frac{R_{air} T_{im,n} T_{s,n}}{V_{im}} \dot{m}_{at,n} \tag{4.12}$$

where

$$\eta_0 = V_{cyl} \eta$$

The model parameters $\eta_0 = V_{cyl}\eta$ and V_{im} may be identified using an extended Kalman filter (EKF). The unknown parameters are defined as additional states and are modelled as follows:

$$\begin{aligned} \eta_{0,n} &= \eta_{0,n} + \xi_n \\ V_{im,n} &= V_{im,n} \end{aligned} \tag{4.13}$$

where

ξ_n is the Gaussian noise.

Parameter η_0 depends on the intake manifold pressure and the speed of the engine; therefore, the requirement for ξ_n to be Gaussian may not be fulfilled and may also depend on a particular realization of the driving cycle. In this situation, the identification of the volume V_{im} may be biased if the assumptions made during the identification about parameters such as covariance and whiteness of ξ_n are incorrect. In practice, the bias is negligible if the noise distribution function is close to Gaussian.

4.3.1.3 Intake manifold volume identification based on event sampling of data

To investigate the statistical properties of the volumetric efficiency variation noise ξ_n, the volumetric efficiency is calculated assuming that V_{im} is known. The autocorrelation of the signal ξ_n sequence is calculated. The volumetric efficiency η_0 may directly be calculated from the following equation:

$$\eta_{0,n} = \frac{\left(P_{im,n+1} - P_{im,n}\right)V_{im} - R_{air}T_{im,n}T_{s,n}\dot{m}_{at,n}}{P_{im,n}} \tag{4.14}$$

For a Gaussian sequence, the autocorrelation between the original signal and the signal shifted by a number of events should be zero. In Figure 4-12 the autocorrelation based on 10000 subsequent values of calculated volumetric efficiency noise ξ_n computed from (4.13) is plotted for time shifts of 0...2500. The autocorrelation for the shifted signal being non-zero suggests that the noise ξ_n is not white. A part of the trace shown in Figure 4-12 is presented in Figure 4-13.

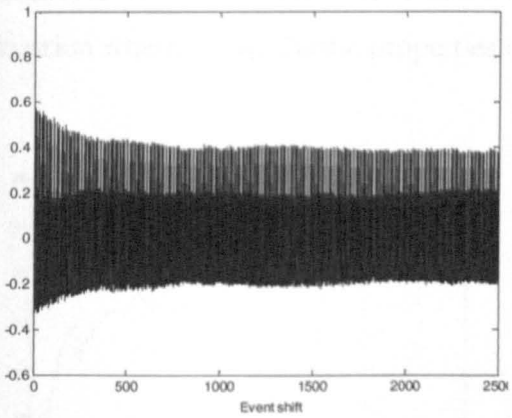

Figure 4-12: Autocorrelation of the volumetric efficiency noise sequence

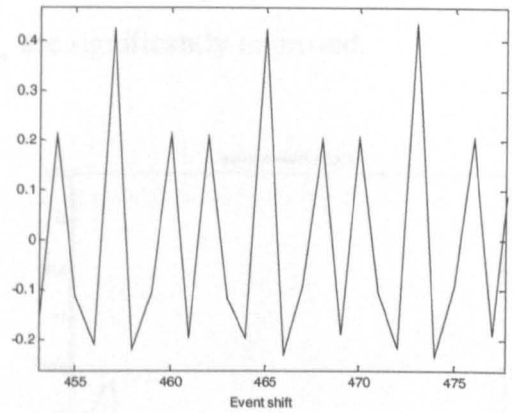

Figure 4-13: Autocorrelation of the volumetric efficiency noise sequence – limited range

It may be noticed that the autocorrelation is periodic with a period of 8 events (8-cylinder engine). This suggests (as expected) that the intake manifold pressure is not uniform in the manifold. In these figures, the imprint of individual cylinder pulses in the intake manifold pressure signal is evident.

The above analysis is in preparation for the intake manifold volume identification. The effect of non-whiteness of the noise on the volumetric efficiency model (4.13) is first demonstrated. The nominal (physical) volume of the intake manifold is $V_{im} = 12[dm^3]$. Using this volume, the volumetric efficiency is calculated from the equation (4.14). The mean value of the volumetric efficiency $mean(\eta_0) = 0.41$ is used as an initial value for this state and the covariance of the noise ξ, $\mathrm{cov}(\xi) = 0.0093$, is used in the covariance matrix. The extended Kalman

filter (EKF) estimation with the model (4.12), (4.13) is carried out. The results of the estimation are shown in Figure 4-14 and Figure 4-15 for the intake manifold volume, and in Figure 4-16 and Figure 4-17 for the volumetric efficiency $\eta = \eta_0 / V_{cyl}$ (where $V_{cyl} = 0.708[\mathrm{dm}^3]$). The initial volume state was assumed to be $V_{im} = 12[dm^3]$. The volume estimate diverges to $24[dm^3]$. For the slightly changed covariance of the signal ξ, $\mathrm{cov}(\xi) = 0.0001$, the result of the estimation of the volume is completely different: $9.45[dm^3]$. The non-Gaussian properties of ξ_n explain the erratic observed behaviour of the estimates. After this test, the data will be processed to achieve the situation where the stochastic properties of ξ_n are significantly improved.

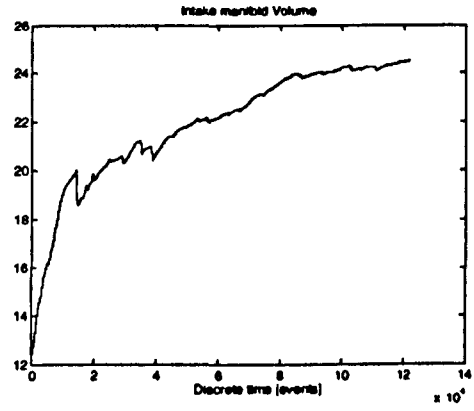

Figure 4-14: Intake manifold volume identification

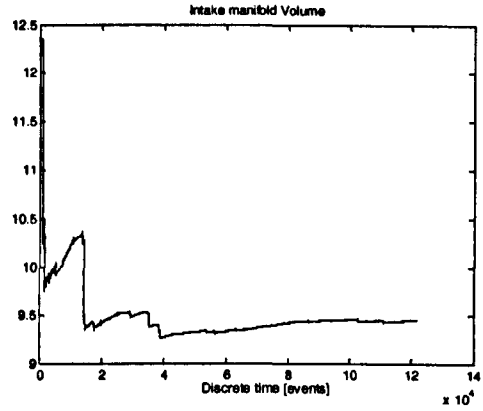

Figure 4-15: Intake manifold volume identification

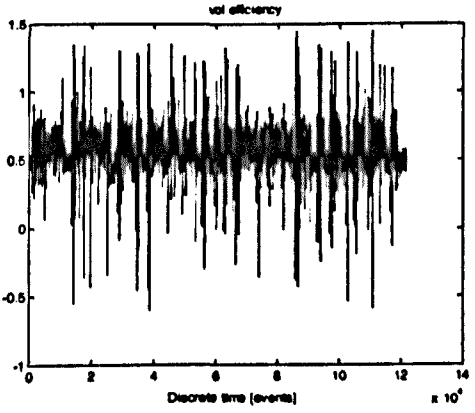

Figure 4-16: Volumetric efficiency η

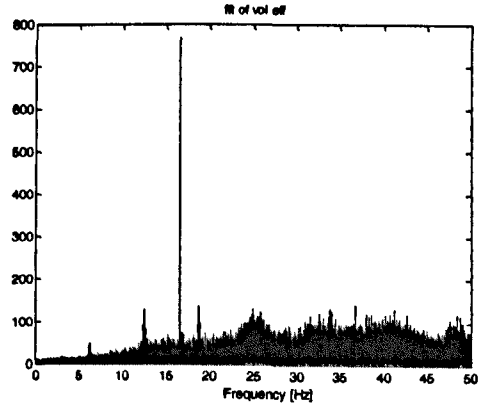

Figure 4-17: Frequency analysis of volumetric efficiency error signal

4.3.1.4 Cycle-based sampling intake manifold volume identification

The periodic behaviour of the autocorrelation function suggests that the sampling period of one engine cycle (8 events for 8-cylinder engine) should be employed. In that case the sampling time is $T_{s,cycle} = 8 \cdot T_s = 120/N$. The event based data is pre-processed and the estimation is repeated with the engine cycle sampled data. The throttle air flow measurement $\dot{m}_{at}$ is averaged over 8 events. The pressure measurement samples taken from the dataset every 8 events are used. This is based upon mass conservation principle for the intake manifold. The cycle-based procedure may not be suitable for the volumetric efficiency η_0 lookup table identification, i.e. the averaging process may remove high-frequency information in the data. However, for the identification of the constant intake manifold volume, the filtering process will not negatively impact the estimate.

The autocorrelation of the volumetric efficiency noise ξ_n obtained based on the model discretized on the engine cycle basis is shown in Figure 4-18. In contrast to earlier results shown in Figure 4-12, this indicates that the signal ξ_n may now be regarded to be much closer to the ideal white noise autocorrelation characteristic. Relatively speaking, the frequency spectrum in Figure 4-19 does not have any significant dominant component as opposed to the frequency spectrum in Figure 4-17.

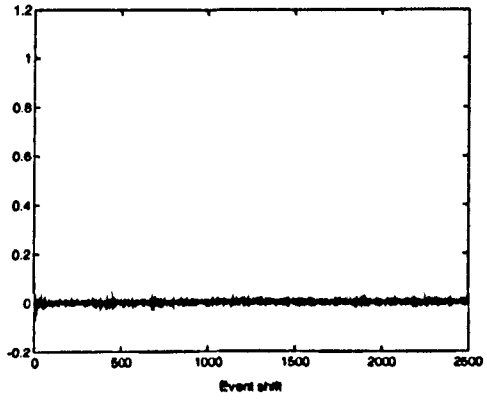

Figure 4-18: Autocorrelation of engine-cycle based volumetric efficiency

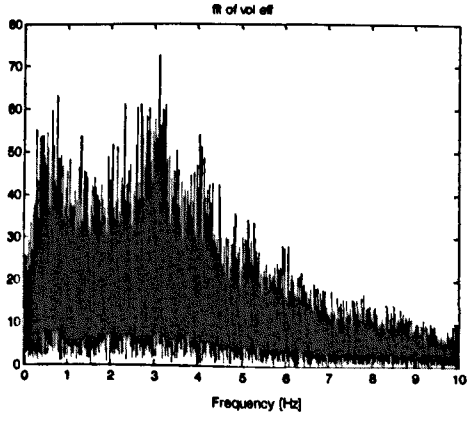

Figure 4-19: Frequency analysis engine-cycle based volumetric efficiency error

So far it has been demonstrated that cycle-based sampling improves the estimation accuracies. The nominal volume of intake manifold is $V_{im} = 12[dm^3]$. For this volume, the volumetric efficiency, again, is calculated in the same way as for the signal statistics analysis. The noise ξ_n covariance $\mathrm{cov}(\xi) = 0.016$ and mean value of the volumetric efficiency $mean(\eta_0) = 3.27$.

The EKF estimation with the model (4.12) and (4.13) is performed. The results of the estimation are shown in Figure 4-20 for the intake manifold volume and Figure 4-21 for the volumetric efficiency $\eta = \eta_0 / V_{disp}$ (where $V_{disp} = 5.67$ [dm^3]). The initial volume estimate was $V_{im} = 12[dm^3]$. The volume estimate converges to $11.87[dm^3]$. Changes in covariance only have a marginal impact on the final volume estimate: repeating the experiment several times and decreasing the values of the initial covariance, the volume always converged to a value close to $V_{im} = 12[dm^3]$. Changes in the initial estimate do not affect the final estimation of the volume either. The intake manifold pressure and engine speed profiles are shown in Figure 4-22 and Figure 4-25.

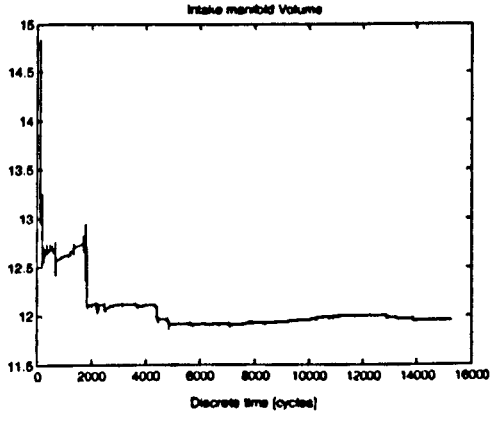

Figure 4-20: Intake manifold volume identification

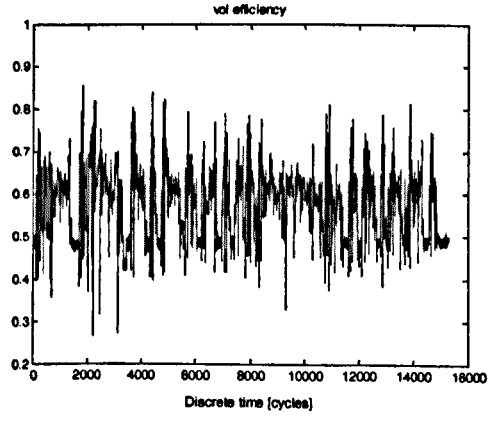

Figure 4-21: Volumetric efficiency η

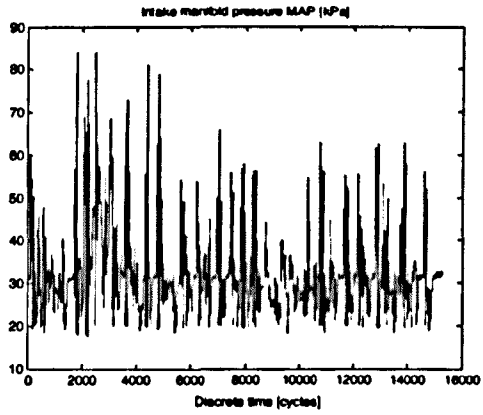

Figure 4-22: Intake manifold pressure profile – cycle based sampling

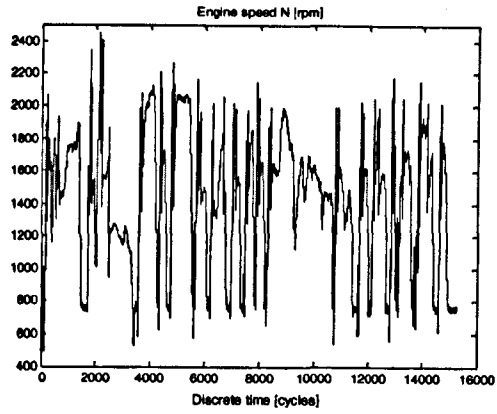

Figure 4-23: Engine speed profile – cycle based sampling

4.3.1.5 Identification of the Volumetric Efficiency model

Given the volume of the intake manifold (either typical value or identified from the data), the volumetric efficiency may be computed using the available measurements and equation (4.15).

$$\eta_n = \frac{R_{air} T_{im,n} T_{s,n} \dot{m}_{at,n} - V_{im}\left(P_{im,n+1} - P_{im,n}\right)}{V_{cyl} P_{im,n}} \tag{4.15}$$

Note that the cylinder displacement volume V_{cyl} is not necessary, since it may be identified together with the volumetric efficiency. However, the displacement volume will give the physical insight into the estimated volumetric efficiency values. Typically the volumetric efficiency must be positive and lower than 1. As an effect of the measurement and process noise the value may temporarily violate these limits. A lookup table for the volumetric efficiency $\eta(P_{im}, N)$ is built using the method explained in section 4.2.2. The engine speed and the intake manifold pressure are used as the lookup table coordinates.

The model with the volumetric efficiency lookup table and the intake manifold volume identified in previous section is validated. The pressure model

output, simulated with the throttle flow rate measurement as an input, is compared with the pressure measurement. The data used during validation is taken from the FTP driving cycle. The validation will indicate the accuracy of the model. The computed error measures will later be used for a comparison with a two-state model. An integrated squared and absolute error performance index are computed during validation:

$$J_{(\,)^2} = \sum \left(MAP_{MODEL} - MAP_{MEASURED} \right)^2 = 40959, \; J_{|\,|} = \sum \left| MAP_{MODEL} - MAP_{MEASURED} \right| = 46278$$

The volumetric efficiency lookup table is presented in Figure 4-24. The intake manifold pressure validation trace is shown in Figure 4-25. For clarity, only for the limited range is presented there.

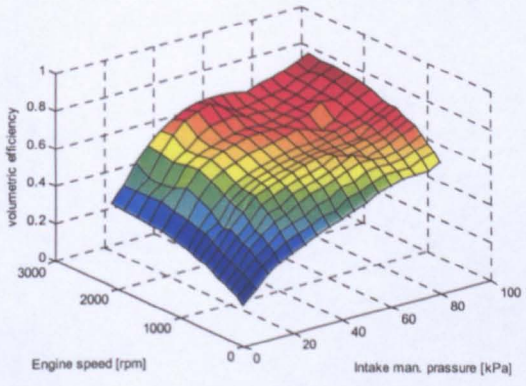

Figure 4-24: Volumetric efficiency lookup table (one-state model)

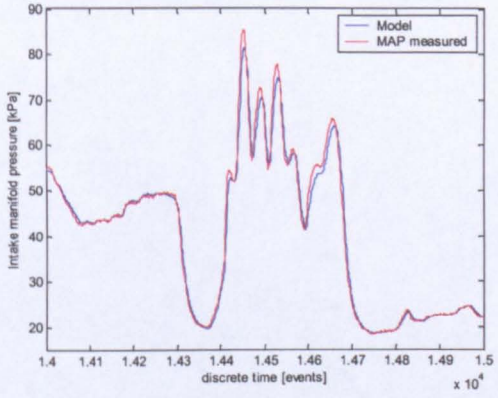

Figure 4-25: Pressure validation (one-state model)

4.3.2 Two-state model analysis

The two state intake manifold parameters are identified in the sequel. As was mentioned before, the model structure is a hybrid of the known non-linear structure based on physical principles with the volumetric efficiency given by the lookup-table. The volumetric efficiency is a function of other engine variables. Other parameters (i.e. heat transfer coefficients) are assumed constant.

4.3.2.1 The model

The model for the intake manifold is given by the equation (4.16) [91]. This model may be easily derived using the mass and energy balance. The derivation will not be presented here for brevity. The heat transfer equation used in [91] was modified to accommodate the intake manifold wall temperature variations (different to the ambient temperature).

$$\dot{P}_{im}(t) = P_{im}(t)\frac{\kappa R_{air} T_a(t)}{V_{im} P_{im}(t)}\left(\dot{m}_t(t) - \frac{T_{im}(t)}{T_a}\dot{m}_{ac}(t) + \frac{\kappa - 1}{\kappa R_{air}}\frac{\dot{Q}_{ext}}{T_a}\right)$$

$$\dot{T}_{im} = T_{im}\frac{\kappa R_{air} T_a}{V_{im} P_{im}}\left(\dot{m}_{at}(t)\left(1 - \frac{T_{im}}{\kappa T_a}\right) - \frac{T_{im}}{T_a}\dot{m}_{ac}(t)\left(1 - \frac{1}{\kappa}\right) + \frac{\kappa - 1}{\kappa R_{air}}\frac{\dot{Q}_{ext}}{T_a}\right) \tag{4.16}$$

where

$\dot{Q}_{ext} = h_1(T_{coolant} - T_{im}) + h_2(T_a - T_{im})$ - heat transfer equation

h_1 - heat transfer coefficient (from engine temperature)

h_2 - heat transfer coefficient (from ambient temperature)

P_{im} - intake manifold pressure [kPa]

T_{im} - intake manifold temperature [K]

T_a - ambient temperature [K]

$T_{coolant}$ - engine coolant temperature [K]

$\dot{m}_{at}$ - air flow rate through the throttle [g/s]

$\dot{m}_{ac}$ - air flow rate through the intake valve (in-cylinder) [g/s]

R_{air} - gas constant $[J/gK]$

V_{im} - intake manifold volume $[[dm^3]]$

In contrast to the original formulation of the two-state model, the heat transfer equation was changed. Two heat transfer components are considered. This implicitly assumes that the intake manifold temperature may be expressed as a weighted

average of ambient and coolant temperatures. The main difficulty for the parameter identification results from the fact that the intake manifold wall temperature is not measured, and more importantly, is not uniform. To maintain simplicity, the manifold wall is modelled as a weighted average of the ambient and coolant temperature. The heat transfer equation given by $\dot{Q}_{ext} = h_1(T_{coolant} - T_{im}) + h_2(T_a - T_{im})$ is equivalent to the following:

$$\dot{Q}_{ext} = h_0[T_{wall} - T_{im}] \tag{4.17}$$

where

$$T_{wall} = \frac{h_1 T_{coolant} + h_2 T_a}{h_0}$$

$$h_0 = h_1 + h_2$$

The assumption, that the wall temperature in (4.17) is the average of coolant and ambient temperatures gives a sufficient degree of freedom for modelling with minimum number of unknown parameters. Note that if the intake wall temperature was modelled separately to the heat transfer equation, the weighted average coefficients would not necessarily be the same as coefficients used in the heat transfer equation. The best alternative to the methodology that assumes the intake manifold wall temperature derived as an average value would require a direct measurement of the intake manifold wall temperature. The thermo-resistive material could be wrapped around the manifold, which would measure an average temperature. This would take into account the fact that the upper part of the manifold is usually cooler.

The port (valve) flow rate is modelled by the speed-density equation (4.9) in the same way as for the one state model in section 4.3.1. The temperature sensor is modelled by a first-order lag. The model parameter $\tau_{InvTemp}$ [1/s] is the inverse of the time constant of the sensor.

$$\dot{T}_{im,measured} = -\tau_{InvTemp} T_{im,measured} + \tau_{InvTemp} T_{im} \tag{4.18}$$

The model (4.16) is discretized on the 90° event basis ($T_s = 15/N$) and the following equations are obtained after some algebra:

$$\begin{aligned}
P_{im,n+1} &= \left(1 - \frac{\kappa}{V_{im}} V_{cyl} \eta_n\right) P_{im,n} + \frac{\kappa R_{air} T_{a,n}}{V_{im}} T_{s,n} \dot{m}_{at,n} + \frac{\kappa - 1}{V_{im}} T_{s,n} \dot{Q}_{ext} \\
T_{im,n} &= \left(1 - \frac{\kappa}{V_{im}} V_{cyl} \eta_n \left(1 - \frac{1}{\kappa}\right)\right) T_{im,n} + \\
&\left(T_{im,n} \frac{\kappa R_{air} T_{a,n}}{V_{im} P_{im,n}} - T_{im,n}^{\,2} \frac{R_{air}}{V_{im} P_{im,n}}\right) T_{s,n} \dot{m}_{at,n} + T_{im,n} \frac{\kappa - 1}{V_{im} P_{im,n}} T_{s,n} \dot{Q}_{ext,n} \\
T_{im\,measured,n+1} &= \left(1 - \tau_{InvTemp} T_{s,n}\right) T_{im\,measured,n} + \tau_{InvTemp} T_{s,n} T_{im,n}
\end{aligned} \tag{4.19}$$

where

$$\dot{Q}_{ext,n} = h_1 \left(T_{coolant,n} - T_{im,n}\right) + h_2 \left(T_{a,n} - T_{im,n}\right)$$

$V_{cyl} = \frac{V_d}{8}$ - cylinder displacement, V_d - engine displacement

4.3.2.2 Identification of two-state model parameters

In the model (4.19) the following constant parameters are subject of the identification: h_1, h_2, V_d, V_{im}, $\tau_{InvTemp}$. The engine displacement V_d and the volumetric efficiency may be combined as $\eta_0 = V_{cyl}\eta$. The engine cycle based sampling period (720 degrees of the crankshaft revolution) is used for the identification of constant parameters.

The model parameters are estimated using an extended Kalman filter (EKF). The unknown parameters are augmented with the model (4.19) as additional states:

$$\begin{aligned} \eta_{0,n+1} &= \eta_{0,n} + \xi_n \\ V_{im,n+1} &= V_{im,n} \\ h_{1,n+1} &= h_{1,n} \\ h_{2,n+1} &= h_{2,n} \\ \tau_{InvTemp,n+1} &= \tau_{InvTemp,n} \end{aligned} \tag{4.20}$$

where

ξ_n is the white noise.

Unlike the one-state model, the noise analysis cannot be performed until the parameters are identified. The values of the heat transfer are unknown and physically justified assumptions are difficult to make. To limit the number of parameters, initially the intake manifold volume was fixed and assumed to be $V_{im} = 11.87\left[dm^3\right]$ (value identified using the one-state model). Performing the identification of the heat transfer coefficients and the temperature measurement sensor dynamics, the following values are obtained: $\tau_{InvTemp} = 0.422$ [1/s] $h_1 = 13.2\left[W \cdot K^{-1}\right]$ and $h_2 = 4\left[W \cdot K^{-1}\right]$. The identified $\tau_{InvTemp} = 0.422$ [1/s] is relatively close to the typical sensor time constant estimate given by the sensor manufacturer: $\tau_{InvTemp} = 0.5$ [1/s]. Autocorrelation analysis of the volumetric efficiency noise ξ_n performed after the identification experiment gives identical results to the one-state model. The noise ξ_n may be regarded as white. The conclusion is that a sampling period equal to the engine cycle should be used for the identification of constant parameters.

An experiment leading to the simultaneous identification of all parameters is carried out next. The investigation of the convergence dependency on the number of unknown parameters was conducted. The model validation was carried out at each

step of the experiment. The results obtained using sets of fixed parameters are collected in Table 4-1 together with costs (integrated squared errors) obtained during the validation phase.

Parameter set	h_1	h_2	V_{im}	$\tau_{InvTemp}$	$\sum(P_{im,Model} - P_{im,Measured})^2$	$\sum(T_{im,Model} - T_{im,Measured})^2$
1	16	4.5	13.5	0.422	3.9691e+4	1.3118e+5
2	10.5	3.6	5.37	0.58	28.981e+4	0.62447e+5
3	13.2	4	**11.87***	0.42	2.4315e+4	1.1466+5
4	14	4.15	**11.87***	**0.5****	2.4496e+4	1.1498e+5

* **fixed** using identified from 1 state model value, ** **fixed** using manufacturer's default value

Table 4-1: Identified parameters and cost associated with each validation

The results for the full set of parameters identified simultaneously are not satisfactory (rows one and two of Table 4-1). The intake manifold volume V_{im} for different choices of initial state error covariance values converges to either higher or lower level. The temperature sensor time constant $\tau_{InvTemp}$ converges to a value in the range of $\tau_{InvTemp} = 0.4...0.6$ [1/s]. Parameters h_1, h_2 also converge to different values. This indicates that by increasing the number of parameters the convergence becomes an issue and the identification results depend on the assumed covariance.

In the last row (4) of Table 4-1, only parameters h_1 and h_2 are identified. The intake manifold volume and the temperature sensor time constant must be known for the identification. In the third row, only the intake manifold volume is fixed to the volume identified with the one-state model. Using integrated squared errors index for the temperature model validation, the errors range from 0.62e+5 to 1.31e+5 for all experiments and different parameter sets. At the same time the pressure validation gives a wider parameter range: 2.43e+4...28.98e+4. The wide span of identified parameters suggests that it is inevitable that some of them should be supplied (or identified in a separate experiment) and fixed during the identification. Comparing costs in Table 4-1, it may be concluded that the procedure where the **intake manifold volume** is identified using the one-state model gives the most reliable results. The intake manifold temperature model identification in this experiment may

not be as reliable as the pressure model identification, due to the relatively small and slow variation of the intake manifold temperature in the dataset used here. The temperature measurements also contain significant measurement noise; therefore, it is sensible to put a stronger emphasis on the accuracy of the model for pressure and the cost for pressure validation should be regarded as the most important one.

As a final conclusion the identification of the **intake manifold volume** should be carried out using the one-state model. There is a slight difference between the heat transfer coefficients identification with or without using the temperature sensor time constant supplied by the manufacturer. The assumption on the sensor time constant may clearly be made as it does not have a strong impact on the model accuracy. However, it will simplify the identification procedure and reduce the computational power requirements for on-line execution of the algorithm, if required. The volumetric efficiency lookup table identification for the two-state model will now be carried out.

4.3.2.3 Volumetric Efficiency model Identification using 2-state model

Using the already identified parameters, the volumetric efficiency may easily be estimated. The EKF state parameters estimation for the model (4.19) and (4.20) is carried out. The estimated time-series for the volumetric efficiency is used along with the pressure and engine speed measurements. The lookup table is built using the method used for the throttle discharge coefficient in section 4.2.2 and later for the one state model in section 4.3.1. The constant parameters are fixed (given in parameter set 3 in Table 4-1): $h_1 = 13.2[W/K]$, $h_2 = 4[W/K]$, $V_{im} = 11.87[dm^3]$. Some model parameters (i.e. the intake manifold volume and temperature sensor time constant) may alternatively be assumed as known. The intake manifold volume is a known physical parameter for the considered engine type. Also, the temperature sensor time constant value may be assumed to be equal to the typical value quoted by the manufacturer of the sensor. In that case it would not have to be identified

(compare the validation results in Table 4-1, row 3 and 4). After the volumetric efficiency lookup table is constructed the model (4.19) is validated against the pressure and temperature measurements logged in the driving cycle. The following integrated squared and absolute errors performance indexes are calculated during validation:

$$J_{(\,)^2,Press} = \sum (MAP_{MODEL} - MAP_{MEASURED})^2 = 23695,\quad J_{|\,|,Press} = \sum |MAP_{MODEL} - MAP_{MEASURED}| = 40740,$$

$$J_{(\,)^2,Temp} = \sum (T_{MODEL} - T_{MEASURED})^2 = 115150,\quad J_{|\,|,Temp} = \sum |T_{MODEL} - T_{MEASURED}| = 90433.$$

The volumetric efficiency lookup table is shown in Figure 4-26 and the intake manifold pressure validation (for clarity only for the limited range) is shown in Figure 4-27. The intake manifold temperature validation is given in Figure 4-28.

The integrated squared and absolute errors of the intake manifold pressure validation are lower for the two-state model when compared with the one-state model ($J_{(\,)^2} = 40959$, $J_{||} = 46278$). This indicates that the two-state model gives a more accurate description of the intake manifold dynamics. At the same time, the temperature model gives relatively accurate estimates (Figure 4-28). Note that for the one state model the temperature measurement obtained from the slow sensor is very slow (identified time constant $\tau_{Temp} = 1/\tau_{InvTemp} = 2.38[s]$). For fast transients this sensor is not capable of providing an accurate temperature measurement. For control purposes, this suggests that the temperature should be estimated.

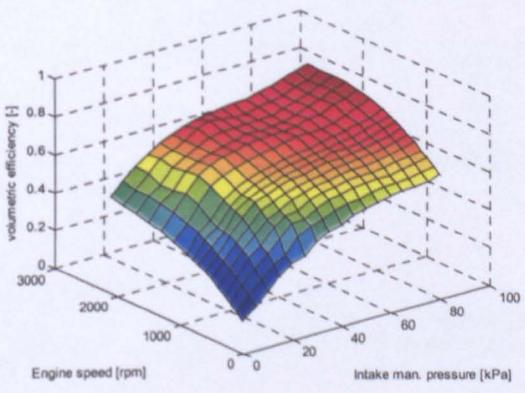

Figure 4-26: Volumetric efficiency lookup table (two-state model)

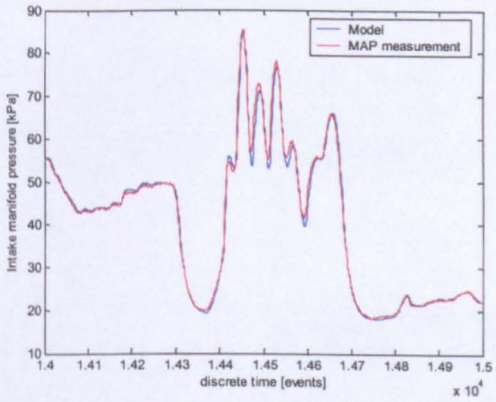

Figure 4-27: Pressure validation (two-state model)

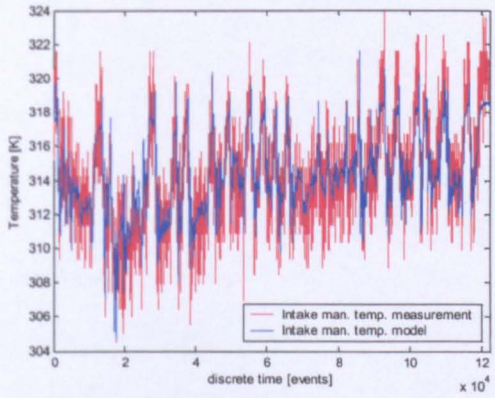

Figure 4-28: Intake manifold temperature validation (two-state model)

4.3.3 Cylinder air charge estimation

After validation of the one-state and two-state models, it is clear that the two-state model gives better results. The following procedure for parameters identification was established:

- In the first stage, the volume of the intake manifold is identified using the one-state model with cycle-sampled data,
- In the second stage, the heat-transfer parameters and temperature-sensor time constant (if not known) are identified – using the two-state model and cycle sampled data,
- In the third stage, employing constant parameters identified at stage one and two, the volumetric efficiency is estimated and the lookup table built that describes the volumetric efficiency as a function of intake manifold pressure and engine speed

The port (or cylinder) airflow rate is computed as:

$$\dot{m}_{ac} = \frac{V_d}{120 R_{air} T_{im}} \eta P_{im} N \qquad (4.21)$$
$$\eta = \eta(P_{im}, N)$$

The volumetric efficiency lookup table, the measurement of the intake-manifold pressure, the engine speed and the intake-manifold temperature estimate are employed. The temperature estimate may be obtained using the second equation of the two-state model (4.19) using the extended Kalman filter. Alternatively, one can employ an EKF-generated estimate of the volumetric efficiency for port flow estimation in equation (4.21).

During experiments it was noticed that the port flow estimates obtained from the lookup table gave more accurate results. The problem with the directly estimated flow rate using EKF estimates of volumetric efficiency is the high level of noise. By changing covariances in the EKF, it is possible to remove the noise, but unfortunately side effects of filtering removes high frequency signal information and deteriorates the transient response. It will have an impact on the identification of the rest of the engine model (fuelling and exhaust), too. In contrast, each value of the volumetric efficiency lookup table is based on many measurements; this results in better noise rejection. Thus, the filtering is performed during model identification rather than on the time-based data estimates and therefore mostly the noise (not 'real' changes in flow-rate) is filtered out. Accuracy of the cylinder air charge is determined by a number of factors. First is the model structure. The structure mismatch between the actual intake manifold and its model introduces an upper limit for the model accuracy. Even for optimal parameters values there will still be a model mismatch. The other factor that determines the model precision is the measurement accuracies of various engine states. Inaccuracies may results in the model parameter mismatch and consequently biased cylinder air charge estimates. To show this, we analyze the steady state conditions where the intake manifold pressure, throttle air flow (MAF) and intake manifold temperature remain constant. From equations (4.8) and (4.9), assuming steady-state conditions, the following expression may be computed:

$$\dot{P}_{im}(t) = \frac{R_{air}T_{im}(t)}{V_{im}}\dot{m}_{at}(t) - \frac{V_d}{V_{im}120}\eta(t)P_{im}(t)N(t) = 0 \tag{4.22}$$

Finally

$$\frac{R_{air}T_{im}(t)120}{V_d N(t)\eta(t)}\dot{m}_{at}(t) = P_{im}(t) \tag{4.23}$$

From the equation (4.23) it is clear that any offset or noise with non-zero mean value in measurements will result in volumetric efficiency modelling errors. Throttle flow measurement bias will result in a corresponding bias in the port flow estimation.

4.4 Fuel delivery and lambda model

The cylinder air charge (CAC) is computed from the equation (4.24) below using the estimate of the port flow rate $\dot{m}_{ac,n}$ at the event n. It provides the information about the amount of the air that enters the cylinders over the time of event (i.e. 90°).

$$CAC_n = \dot{m}_{ac,n}\frac{15}{N_n} \tag{4.24}$$

In an 8-cylinder engine as much as three cylinders may be charged at the same time. Therefore the amount of the air inducted during one engine event cannot simply be associated with only one cylinder. But it may be argued, that only one cylinder gets a major part of the estimated air charge. For a mean-value model, it is assumed that the estimated air charge is associated with only one cylinder. Thus, it is assumed that the amount of the air per cylinder for the 8 cylinder engine equals the amount of air that enters all cylinders over one event. This results from the division of the 720 degrees cycle (two full revolutions of the crankshaft) by the number of cylinders that are being filled with the air over one cycle. With the knowledge of

CAC, the fuel delivery path (with the FPW delay, injector parameters and fuel film dynamics) and the lambda path (with the exhaust manifold transport delay and the lambda sensor dynamics) may be identified.

The in-cylinder air-fuel ratio represents both the air and fuel path. The non-linearity associated with the division (ratio) may cause problems during identification since the operating point changes over a wide range during the driving cycle. To circumvent this problem, the intake manifold is identified separately and the lambda measurement used for identification of the fuel delivery parameters only. In this way the problem of additional non-linearity associated with lambda representing the ratio of two unknown variables is eliminated.

The time delays must be tackled in an unconventional way. The method used here removes the time delays from the data since these are either on input or output of the system. The data pre-processing may simplify the problem to the identification of the delay-free system. However, time delays must first be determined.

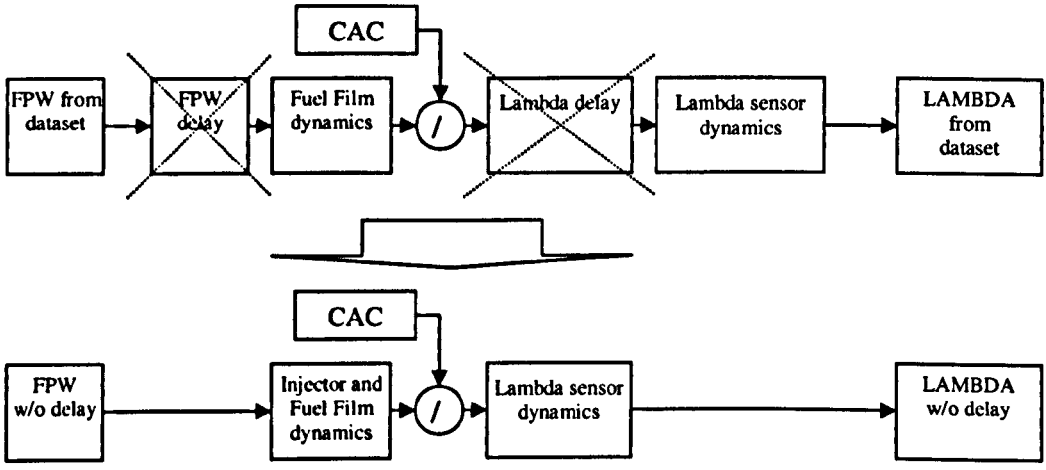

Figure 4-29: Structure conversion diagram

Finally, the identification of the simplified model structure is carried out. The structure conversion is presented in Figure 4-29. The pre-processed FPW signal without the time delay is used as an input of the system which is being identified. The output of the model is the pre-processed lambda measurement with the time delay removed. The cylinder air charge (CAC) obtained from the port flow estimate

from section 4.3.3 is used as a known time-varying parameter. In the subsequent sections, the details of this approach are addressed.

4.4.1 Fuel injection delay

The fuel injection delay results from the pulse width modulation used for the fuel measuring and the injection strategy. There are two possible strategies for the port fuel injection:

- the injection starts, when the intake valve opens
- the end of the injection is fixed at a few degrees of crankshaft rotation before intake valve opening

As part of the fuel injection strategy and for the vehicle under study, the fuel injection at every event is ended before the intake valve opens. To pre-process the data and remove the correct time delay the injection strategy must be well understood. The engine cycle takes 720° of the crankshaft rotation with the 90° event. The beginning of the engine cycle is associated with the position of the 60° of the crankshaft before TDC (Top Dead Centre) on compression. For the *k-th* cycle at this position assume that $C_k^0 = 0°$ (position of crankshaft relative to the beginning of the cycle 60° before TDC). An illustration of this is given in Figure 4-30.

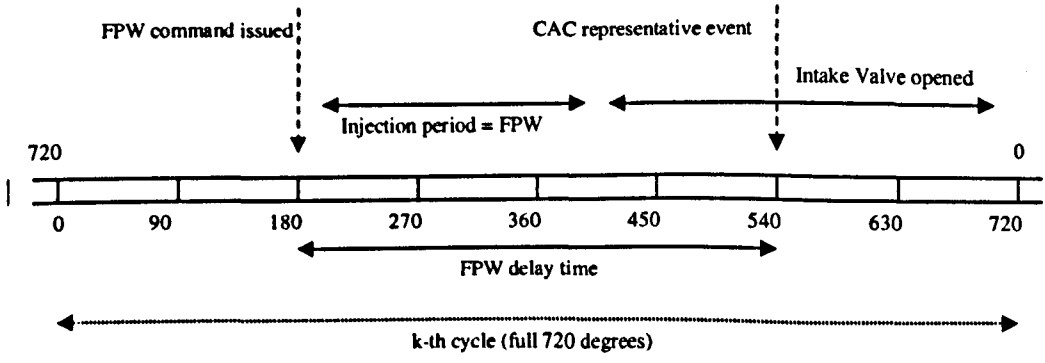

Figure 4-30: The FPW delay timing

Consider the engine with the following valve timing for one cylinder (Figure 4-30). The intake valve opens at $\alpha_{open} = 403^\circ$ and closes at $\alpha_{close} = 697^\circ$. Note, that the angles are calculated relative to the 60° before TDC on compression.
The discrete events between the opening and closing of the intake valve occur at: $C_k^5 = 450^\circ$, $C_k^6 = 540^\circ$, $C_k^7 = 630^\circ$. Assume, that the event with the highest port flow rate is $C_k^6 = 540^\circ$. The injection must end α_c degrees before the intake valve opens. The last discrete event before the valve opens occurs at $C_k^4 = 360^\circ$.

Now, depending on the amount of the fuel which has to be injected the duration of injector opening $FPW_{i,n}$ and the current engine speed, the angle of the injection may be calculated.

$$\alpha_{injection} = 6 \cdot N_n \cdot FPW_{i,n} \tag{4.25}$$

where

N_n - speed of the engine [rpm]

The injection should end at $\alpha_{open} - \alpha_c$ degrees. If $\alpha_{open} - \alpha_c - \alpha_{injection} \geq 360^\circ$ then the command should be available from the controller at time $C_k^4 = 360^\circ$. If $\alpha_{open} - \alpha_c - \alpha_{injection} < 360^\circ$ then $C_k^3 = 270^\circ$ is considered. Using the information about the fuel pulse width $FPW_{i,n}$ and the speed of the engine N_n, the time delay is determined.

As a **numerical example**, assume that speed N_n and $FPW_{i,n}$ are their maximum values in the dataset. From equation (4.25) the following value may be obtained: $\alpha_{injection} = 149.1^\circ$. Assume, that $\alpha_c = 5^\circ$:

$$\alpha_{open} - \alpha_c - \alpha_{injection} = 403^\circ - 5^\circ - 149.1^\circ = 248.9^\circ$$

Therefore the $C_k^2 = 180°$ event determines the amount of the injected fuel which gets into the cylinder at $C_k^6 = 540°$ which determines the time of the air induction. The discrete time delay equals 4.

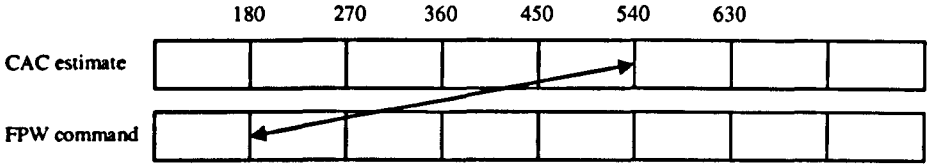

Figure 4-31: Fuel delivery time delay processing

4.4.2 Exhaust manifold delay

The in-cylinder air-fuel ratio is determined by the amount of the air and the fuel that enters the cylinder at each event. The homogeneous charge is compressed, combusted and finally released through the exhaust valves. This takes about 6 engine events. A further delay is introduced in the exhaust manifold and pipes. This time delay is a variable; it is inversely proportional to the exhaust gas flow rate. A full analysis of the time-delay estimation follows.

The exhaust manifold and pipe may be considered as a duct that introduces pure time delay. Since it is difficult to determine the time constant of the exhaust manifold gas mixing it is better to assume that most of the lag is lumped in the lambda sensor. Some of the mixing effect may also be accounted for in the lambda sensor dynamics. Assume that the volume of the exhaust manifold and pipe are known. The time delay may be calculated as a parameter proportional to the integrated exhaust gas flow rate into the exhaust manifold. The identification of the variable time delay is rather difficult to carry out. An iterative identification method that uses an additional lambda measurement is proposed.

Using the air mass flow into the cylinder the volume gas flow out of the cylinder is obtained. For this purpose the ideal gas law is used. It is assumed that any

cylinder blow-by during compression and combustion is negligible. Also, it is assumed that the stoichiometric Air-Fuel ratio is maintained at all times. Using the exhaust manifold pressure and the exhaust manifold gas temperature measurements, the volume of the gas entering the exhaust manifold over one engine event is given by the following expression.

$$V_{cyl,n} = \frac{\left(m_{a,n} + \frac{1}{AF_{stoich}} m_{a,n} \right) R_{EM} T_{EM,n}}{P_{EM,n}} \tag{4.26}$$

where

R_{EM} - ideal gas constant for the exhaust gas

$T_{EM,n}$ - exhaust gas temperature

$P_{EM,n}$ - exhaust gas pressure

$m_{a,n}$ - mass of the air trapped in the cylinder

AF_{stoich} - stoichiometric air-fuel ratio

The discrete time delay $k_{EM,n}$ at time n is implicitly given by the following equation:

$$V_{EM} = \sum_{i=n}^{n+k_{EM,n}} V_{cyl,i} \tag{4.27}$$

where

V_{EM} - exhaust manifold volume (with the pipe)

The meaning of the equation (4.27) is that the exhaust gas has to be pushed out of the exhaust manifold by gases leaving the combustion chamber in the next engine events. This is illustrated in Figure 4-32.

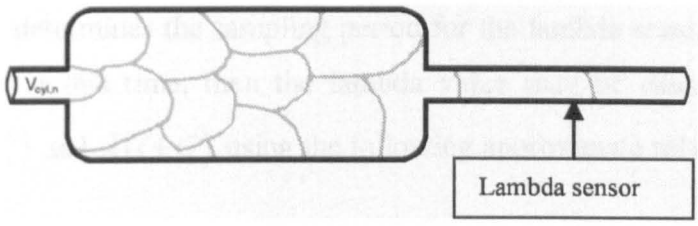

Figure 4-32: Exhaust manifold delay modelling

In practice the volume of the exhaust manifold will not be equal to an integer number of volumes of exhaust gas entering the manifold. The $k_{EM,n}$ that is required to be an integer number in the equation (4.27) will be determined with some approximations. Define the discrete time delays $k^-_{EM,n} \le k_{EM,n} < k^+_{EM,n}$ as:

$$k^-_{EM,n} : V_{EM} \ge \sum_{i=n}^{n+k^-_{EM,n}} V_{cyl,i} \tag{4.28}$$

$$k^+_{EM,n} : V_{EM} < \sum_{i=n}^{n+k^+_{EM,n}} V_{cyl,i} \tag{4.29}$$

and

$$k^+_{EM,n} = k^-_{EM,n} + 1 \tag{4.30}$$

The corresponding continuous time (or real time) delays are determined by the sampling times: $t^-_n = t_{k^-_{EM,n}}$, $t^+_n = t_{k^+_{EM,n}}$ resulting from discrete events $k^-_{EM,n}$, $k^+_{EM,n}$ obtained from equations (4.28), (4.29).

The exact time delay is denoted as t^0_n and may be calculated from the following equation.

$$t^0_n \approx t^-_n + \left(t^+_n - t^-_n\right) \frac{V_{EM} - \sum_{i=n}^{n+k^-_{EM,n}} V_{cyl,i}}{V_{cyl,n+k^+_{EM,n}}} \tag{4.31}$$

This time delay determines the sampling period for the lambda sensor. If the sample is not available at this time, then the lambda value may be calculated from the samples $\lambda(t+t_n^+)$ and $\lambda(t+t_n^-)$ using the following approximate relationship:

$$\lambda(t+t_n^0) \approx \lambda(t+t_n^-) + \left(\lambda(t+t_n^+) - \lambda(t+t_n^-)\right)\frac{t_n^0 - t_n^-}{t_n^+ - t_n^-} \tag{4.32}$$

The above method of calculation of the time delay may be implemented on-line using shift registers and storing the required data there. The lambda sensor ideally should be sampled at the time given by the equation (4.31). In the FTP dataset the lambda sensor is sampled at each event. The lambda value is therefore obtained using the linear interpolation given by the equation (4.32). Since the lambda sensor response is lagged, the interpolation will result in a sufficiently good accuracy. Alternatively, for simplicity, the transport time delay may be assumed to be equal to $k_{EM,n}^-$ (or $k_{EM,n}^+$). The six- event time delay due to gas entrapment in the cylinder has to be added to the exhaust manifold transport delay. The resulting delay is removed from the data as it is shown in Figure 4-33. The lambda measurement associated with the considered intake event is therefore logged $k_{EM,n}+6$ events later.

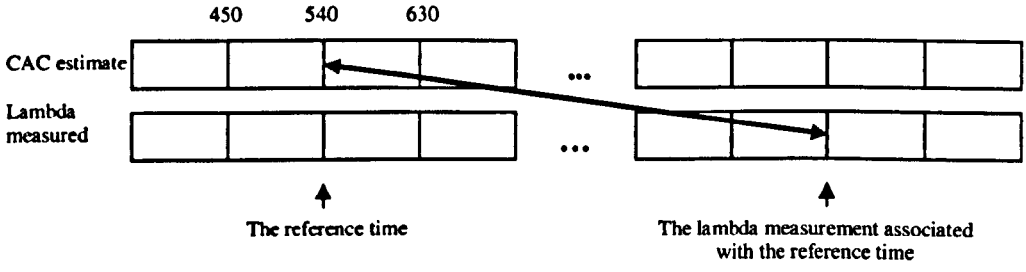

Figure 4-33: Exhaust time delay processing

The mathematical representation of the variable time delay is quite complicated. In the state-space model additional states have to be introduced, with the associated changes in the model structure. During off-line model identification, the time delay may be removed from the data. The same data processing procedure may be implemented on-line employing shift registers.

4.4.3 Exhaust manifold delay identification

To continue with the identification procedure, the exhaust manifold time delay has to be determined and removed from the data. The exhaust manifold volume V_{EM} may be measured and used directly with the measurements of the pressure and the temperature. However, the effective volume of the exhaust manifold may be different from the value measured. To determine the effective exhaust manifold volume an additional lambda measurement (known as wraf5) close to the one of exhaust valves is employed. An iterative solution to the problem is presented. The physical volume of the exhaust manifold (sum of volumes of two manifolds for 'V' engine) is a known engine parameter. The series of hypotheses on the effective exhaust manifold volume is made (e.g. 2.5L, 2.6L,…,3.5L). The correlation between the lambda measurement at the exhaust valve location and the shifted main lambda measurement (known as wraf3) is computed. The lambda time shift results from the assumed exhaust manifold volume (e.g. 2.5L, 2.6L,…,3.5L) and the estimated exhaust gas flow rate based on the cylinder air charge estimates. It is calculated from equations (4.31) and (4.32). The correlation analysis [93], [94] for different hypotheses is shown in Figure 4-34.

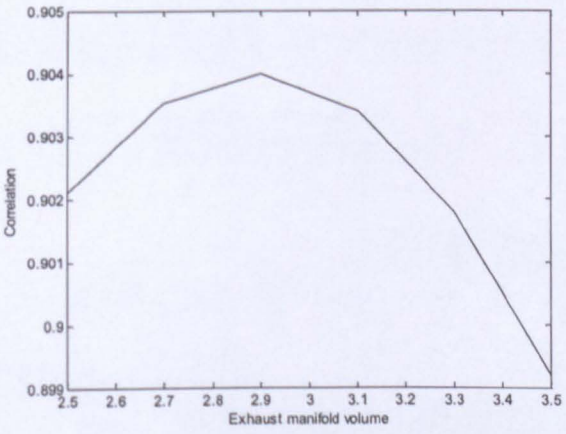

Figure 4-34: Correlation coefficient between wraf5 and shifted wraf3

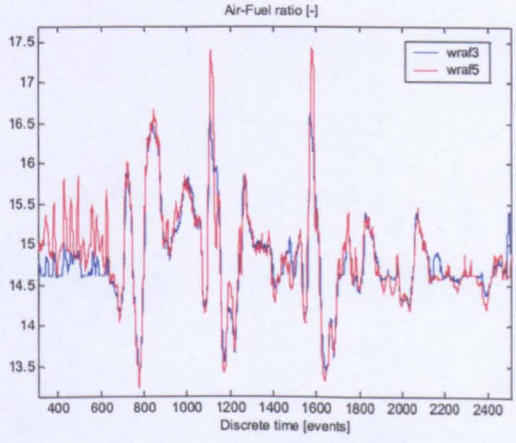

Figure 4-35: wraf3 shifted using 2.9-litre exhaust manifold volume and wraf5

The maximum correlation occurs at 2.9L. The volume of 2.9L is concluded to be the effective exhaust manifold volume is used for the final exhaust manifold

transport delay calculation. The lambda signal at the exhaust valve position and the main lambda measurement shifted using the Cylinder Air Charge estimate and the volume of 2.9L is shown in Figure 4-35. Two signals are plotted to show the efficiency of the time delay removal. The estimated effective volume of 2.9L is very close to the actual physical measured exhaust manifold volume (two banks of cylinders) which is 3.0L.

4.4.4 Lambda sensor dynamics

The air-fuel ratio is measured in the exhaust manifold by the lambda sensor. The sensor is modelled by a first-order lag:

$$\lambda = \frac{1}{s\tau_\lambda + 1}\lambda_{EM} \tag{4.33}$$

where

τ_λ - time constant of the lambda sensor

λ - lambda measured

λ_{EM} - 'real' lambda of the exhaust gas

The model is discretized as follows

$$\lambda_{n+1} = \left(1 - \frac{T_{s\lambda,n}}{\tau_\lambda}\right)\lambda_n + \frac{T_{s\lambda,n}}{\tau_\lambda}\lambda_{EM} \tag{4.34}$$

where

$T_{s\lambda,n}$ - sampling period

Exhaust manifold mixing is modelled together with the lambda sensor. Exhaust manifold mixing in general is a complicated process and may not be accurately modelled by the first-order lag. By increasing the time constant of the

lambda sensor, the exhaust manifold mixing may be modelled in an approximate way. It is expected that for control purposes, the accuracy of that method is sufficient. No additional dynamics is introduced in the exhaust manifold mixing modelling.

$1/\tau_\lambda$	correlation
5	0.6639
6	0.6753
7	0.6801
7.5	0.6809
8	***0.6810***
8.5	0.6804
9	0.6721
11	0.6721
13	0.6627

Table 4-2: Correlation for a different lambda time constants

The exhaust manifold time delay is computed based on the effective exhaust manifold volume. It was identified using an additional lambda measurement in section 4.4.3. With the lambda measurements pre-processed by removing the delay, the lambda sensor time constant is now determined. With the time delay already removed from the data, the correlation coefficient for the measured and the modelled lambda is computed. This is a good parameter that may be exploited for the identification of the sensor time constant τ_λ. The series of hypotheses on the lambda sensor time constant τ_λ is tested. For each time constant average parameters for the injector (gain and offset) are identified using the EKF methodology. The X and τ parameters of the fuel film dynamics (detailed model will be presented later) are neglected for the lambda sensor time constant identification. As was mentioned earlier, the model accuracy is tested through correlation analysis done for the lambda measurement and the lambda modelled with assumed lambda sensor time constant. The test was performed over 10000 samples from the dataset. The results are presented in Table 4-2. The maximum correlation is at $1/\tau_\lambda = 8[1/s]$ which is equivalent to $\tau_\lambda = 125[ms]$.

The above method was introduced for the τ_λ parameter because the extended Kalman filter (EKF) estimation did not give satisfactory results. The estimate of the time constant converged to an infeasible value (1[s]). The possible reason for that might be a significant uncertainty in the system if several parameters are identified simultaneously. It could also be due to non-persistency of the data (the peril of a data-driven approach). This may become particularly important for large number of parameters and most importantly due to noise. The estimation algorithm tends to filter the modelled lambda signal heavily by increasing the lambda sensor time constant. It may occur if the modelled exhaust time delay is slightly different from the actual, encountered in the real system. The method presented here does not result in aforementioned lambda sensor time constant divergence. The results obtained with hypothesis-based method are physically justified in a way that are within the typical for the lambda sensor range of 50...150[ms]. Other lambda sensor identification techniques may use the step response to the fuel injection command under steady state conditions. The step response test simultaneously provides an estimate of the exhaust time delay. The method however is tedious and may not be accurate enough. Visual inspection of the response may, however, provide a value for the sensor lag at the operating condition. In this chapter, only the driving cycle data from a vehicle is used for the identification and the alternative method referred to was not exploited.

4.4.5 The fuel injector model

The fuel injector may be modelled by the following equation:

$$m_{fi,n} = k_{fi,n}\left(FPW_{fi,n} - O_{fi,n}\right) \tag{4.35}$$

where

$m_{fi,n}$ - mass of injected fuel [g]

$k_{fi,n}$ - injector gain [g/sec]

$FPW_{fi,n}$ - pulse-width command [sec]

$O_{fi,n}$ - offset [sec]

Additional corrections of the injector gain (state-dependent gain) will be introduced in the sequel. The equation presented here reflects the most important part of the injector characteristics.

4.4.6 Fuel Film dynamics model

The fuel film dynamics may be modelled by the first-order X-τ model [95], [91]:

$$\dot{m}_w = -\frac{1}{\tau_n} m_w + X_n \dot{m}_{fi} \tag{4.36}$$

$$\dot{m}_{fc} = \frac{1}{\tau_n} m_w + (1 - X_n)\dot{m}_{fi} \tag{4.37}$$

where

m_w - mass of the wall fuel

$\dot{m}_{fi}$ - fuel mass flow rate through the injector

$\dot{m}_{fc}$ - fuel mass flow rate into the cylinder

τ, X - model parameters

Assuming linearity, the model may be discretized using the Euler method:

$$m_{w,n+1} = \left(1 - \frac{T_{s,n}}{\tau_n}\right) m_{w,n} + X_n m_{fi,n} \tag{4.38}$$

$$m_{fc,n} = \frac{T_{s,n}}{\tau_n} m_{w,n} + (1 - X_n) m_{fi,n} \tag{4.39}$$

where

$T_{s,n}$ - sampling period at current engine event

$m_{fi,n} = T_{s,n}\dot{m}_{fi,n}$ - mass of the fuel injected during one event

$m_{fc,n} = T_{s,n}\dot{m}_{fc,n}$ - cylinder fuel charge for one event

The input for the injector is the time of its opening. Pulse-width modulation is used to control the amount of fuel injected.

4.4.7 Fuel film and fuel injector parameter identification

Three models for the injector and the fuel film dynamics are identified in this section. Three models are of increasing degree of complexity. The objective is to identify the least complicated but satisfactory models for use in the next stage (i.e. control).

4.4.7.1 Model 1

The following model is considered for the parameter identification. This model includes the injector and the fuel film dynamics.

$$\begin{aligned} m_{w,n+1} &= \left(1 - T_{s,n}\frac{1}{\tau_n}\right)m_{w,n} + X_n m_{i,n} \\ \lambda_{n+1} &= \left(1 - \frac{T_{s\lambda,n}}{\tau_\lambda}\right)\lambda_n + \frac{T_{s\lambda,n}}{\tau_\lambda}\frac{m_{ac,n}}{T_{s,n}\frac{1}{\tau_n}m_{w,n} + (1 - X_n)m_{i,n}} \end{aligned} \tag{4.40}$$

$$m_{i,n} = k_{fi,n}\left(FPW_{fi,n} - O_{fi,n}\right) \tag{4.41}$$

where

$m_{ac,n} = \dot{m}_{ac,n} T_{s,n}$ - the Cylinder Air Charge used as a parameter

Unknown parameters are defined as additional states for parameter estimation with the extended Kalman filter.

$$\Theta_{M1,n+1} = \Theta_{M1,n} \tag{4.42}$$

where

$$\Theta_{M1,n} = \left[X_n, \frac{1}{\tau_n}, k_{fi,n}, O_{fi,n} \right]^T$$

The model (4.40) identification will only provide the average values of the actual wall wetting parameters. This is the case since it is known that the X_n and τ_n are not constant and depend on other states. This feature of the system will be accounted for later in this section. Also, the injector parameters may depend on other engine parameters (e.g. intake manifold pressure, engine speed...). However, in this simplest structure parameters (4.42) are modelled to be time-invariant. The main difficulty associated with extended Kalman filter parameter estimation is the choice of initial values and the initial covariance P_0. The following initial parameters in (4.42) are used: $m_{w,0} = 0.005$, $\lambda_0 = 14.57$, $X_0 = 0.05$, $1/\tau_0 = 5$, $k_{fi,0} = 3.6$, $O_{fi,0} = 5e-4$.

The above values are initial guesses of parameters being subject of the identification. These were established based on an approximate knowledge about the system.

The extended Kalman filter (EKF) identification is repeated several times with the state estimate obtained at the end of the FTP driving cycle. These final values were used as initial estimates for the next iteration. The initial state estimate covariance P_0 is always scaled as 20% the value of the state estimate value. This was an experimentally derived tuning parameter. Alternatively, the covariance may

simply be decreased from iteration to iteration. It was noticed that when the EKF algorithm started with different initial states and covariances, the parameters converged to slightly different values at the end of the driving cycle data. For this reason the final estimates for several identification experiments are collected (see illustration in Figure 4-36). The mean values of these final parameter estimates are calculated as the identified parameters sought.

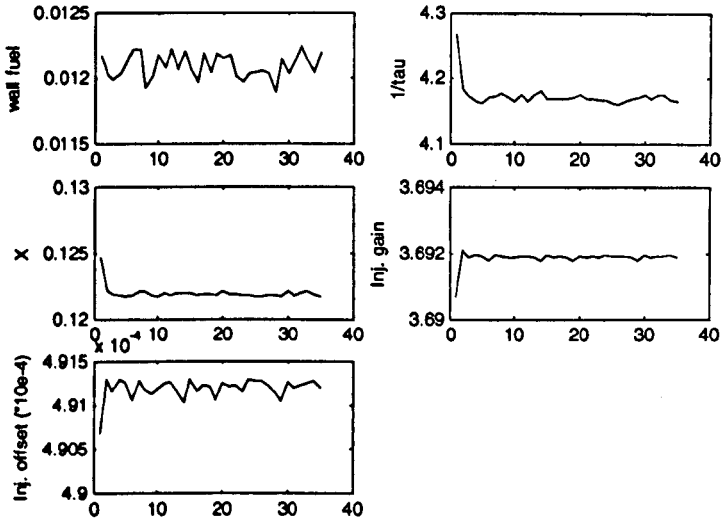

Figure 4-36: Subsequent values for identified parameters for identification experiments

The mean values of parameters are computed from a set of subsequent results of the identification experiments (starting from 5-th) are:

$$X = 0.122,\ \frac{1}{\tau} = 4.169,\ k_{fi} = 3.692,\ O_{fi} = 4.912\ \text{e-4}.$$

4.4.7.2 Model 1 validation

The validation of the model (4.40) was performed and the result is shown in Figure 4-37 and in Figure 4-38. The performance index as the sum of the absolute or

squared values of the difference between the model and measurement lambda values was employed. The following modelling error indexes were computed: $J_{\lambda,abs} = \sum|\lambda_{model} - \lambda_{measurement}| = 29531.6636$, $J_{\lambda,squared} = \sum(\lambda_{model} - \lambda_{measurement})^2 = 13038.3385$. Additionally the correlation coefficient for the modelled and measured lambda signals was computed. Its value for *Model 1* is $J_{\lambda,corr} = 0.7049$.

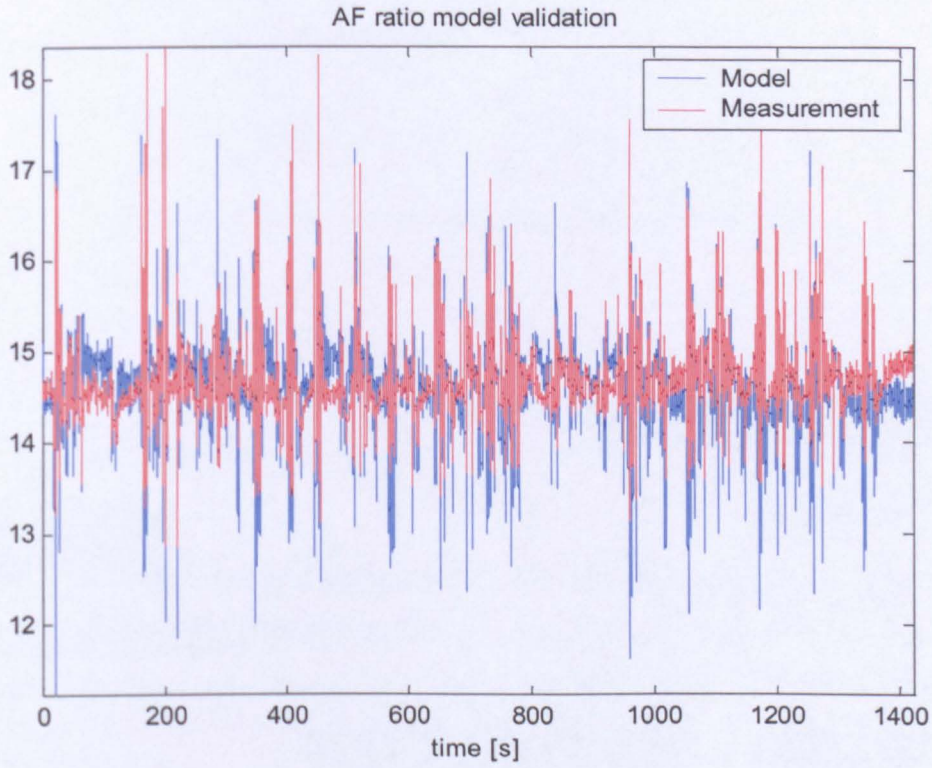

Figure 4-37: AF ratio model validation for model 1

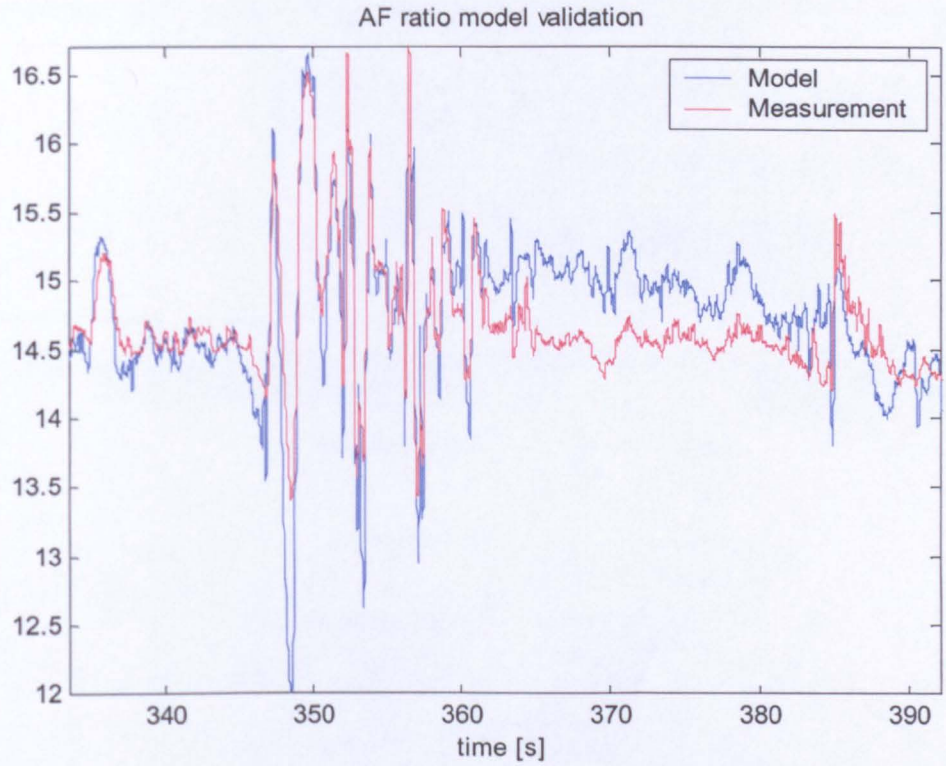

Figure 4-38: AF ratio model validation for model 1 (expanded view)

4.4.7.3 Model 2

Model 2 contains the state-dependent injector gain as a linear function of battery voltage and intake manifold pressure. The same procedure with repeated EKF identification is employed. The model (4.40) is employed with the following injector model:

$$m_{i,n} = \left(k_{1,fi,n} + k_{2,fi,n} U_{batt,n} + k_{3,fi,n} P_{im,n}\right)\left(FPW_{fi,n} - O_{fi,n}\right) \tag{4.43}$$

For the extended Kalman filter parameter estimation, as before, unknown parameters are defined as states. The additional states are:

$$\Theta_{M2,n+1} = \Theta_{M2,n} \tag{4.44}$$

where

$$\Theta_{M2,n} = \left[X_n, \frac{1}{\tau_n}, k_{1,fl,n}, k_{2,fl,n}, k_{3,fl,n}, O_{fl,n} \right]^T$$

The Extended Kalman Filter identification is repeated several times as for *Model 1*.

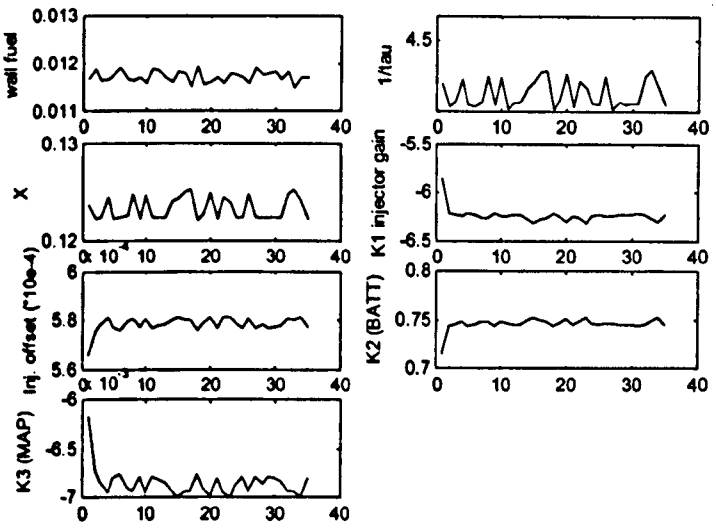

Figure 4-39: Subsequent values for identified parameters for identification experiments

The mean values of the identified parameters are computed to be: $X = 0.123$, $\frac{1}{\tau} = 4.273$, $k_{1,fl} = -6.251$, $k_{2,fl} = 0.747$, $k_{3,fl} = -0.00687$, $O_{fl} = 5.787$ e-4.

4.4.7.4 Model 2 validation

The results of the model validation are shown in Figure 4-40 and Figure 4-41. The following modelling error indexes were computed as the sum of the absolute and squared values of the difference between the model and measurement lambda values: $J_{\lambda,abs} = \sum|\lambda_{model} - \lambda_{measurement}| = 23327.1708$, $J_{\lambda,squared} = \sum(\lambda_{model} - \lambda_{measurement})^2 = 8902.3906$. The correlation of the modelled and measured signals was computed to be $J_{\lambda,corr} = 0.8033$.

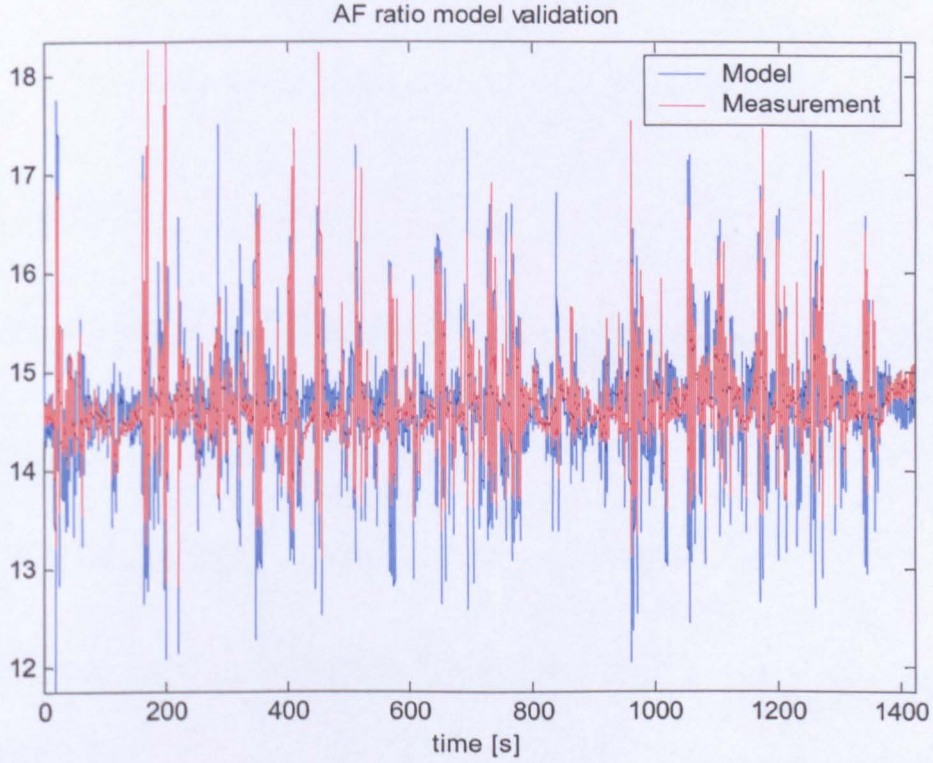

Figure 4-40: AF ratio model validation for model 2

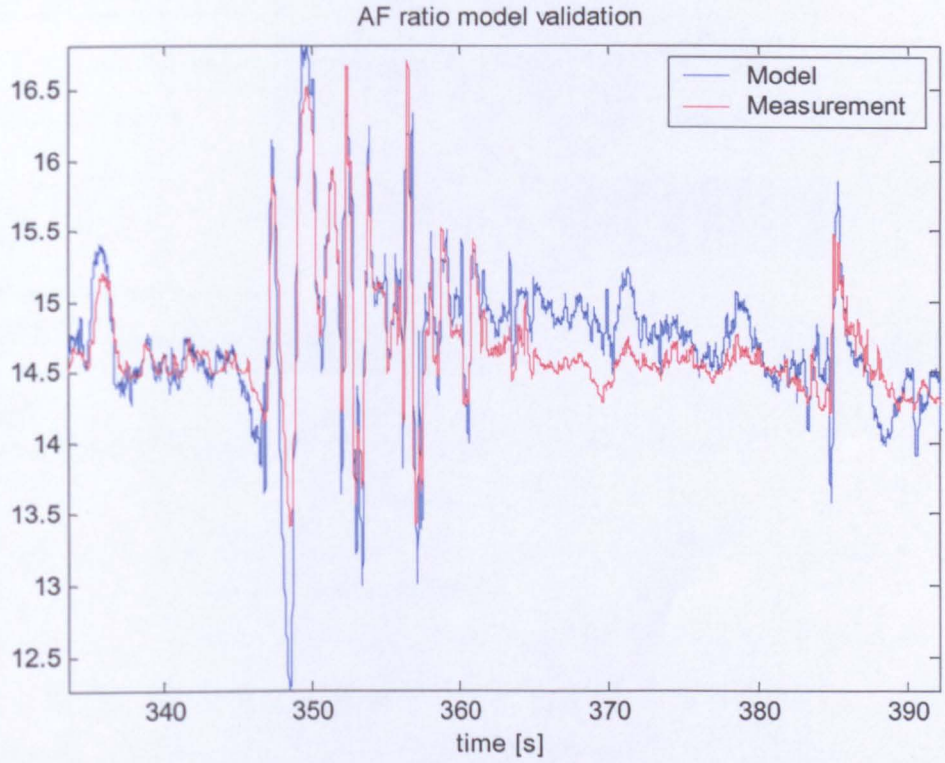

Figure 4-41: AF ratio model validation for model 2 (expanded view)

4.4.7.5 Model 3

Model 3 is a further extension of *Model 2* where state-dependent X and τ parameters are employed. These are assumed to be linear functions of the intake manifold pressure. The injector gain is defined as a state-dependent linear function of the battery voltage and the intake manifold pressure. The model (4.40) is employed with the following injector model and fuel film dynamics coefficients.

$$m_{i,n} = \left(k_{1,fi,n} + k_{2,fi,n} U_{batt,n} + k_{3,fi,n} P_{im,n}\right)\left(FPW_{fi,n} - O_{fi,n}\right)$$

$$\frac{1}{\tau_n} = \frac{1}{\tau_{1,n}} + \frac{1}{\tau_{2,n}} P_{im,n} \qquad (4.45)$$

$$X_n = X_{1,n} + X_{2,n} P_{im,n}$$

where

$m_{ac,n} = \dot{m}_{ac,n} T_{s,n}$ - cylinder air charge (over one event)

For the extended Kalman filter parameter estimation, the unknown parameters are defined as states are:

$$\Theta_{M3,n+1} = \Theta_{M3,n} \tag{4.46}$$

where

$$\Theta_{M3,n} = \left[X_{1,n}, X_{2,n}, \frac{1}{\tau_{1,n}}, \frac{1}{\tau_{2,n}}, k_{1,fl,n}, k_{2,fl,n}, k_{3,fl,n}, O_{fl,n} \right]^T$$

Finally, the mean values of the identified parameters are obtained: $X_1 = 0.0334$, $X_2 = 0.00168$, $1/\tau_1 = 3.511$, $1/\tau_2 = 0.0166$, $k_{1,fl} = -6.174$, $k_{2,fl} = 0.740$, $k_{3,fl} = -0.00647$, $O_{fl} = 5.687$ e-4. Note that some of the estimates were removed from the calculation due to large excursions from the mean values (Figure 4-42).

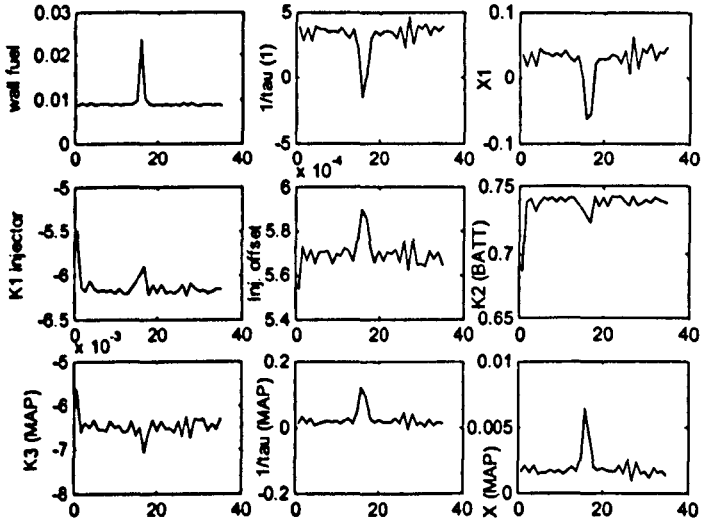

Figure 4-42: Subsequent values for identified parameters in identification experiments

4.4.7.6 Model 3 validation

The results of the model validation are shown in Figure 4-43 and Figure 4-44. The following modelling error indexes were computed as an indication of the model accuracy: $J_{\lambda,abs} = \sum|\lambda_{model} - \lambda_{measurement}| = 23233.7806$, $J_{\lambda} = \sum(\lambda_{model} - \lambda_{measurement})^2 = 8758.5762$. The correlation of modelled and measured signals was computed as: $J_{\lambda,corr} = 0.8015$. It may be noticed that the state-dependency of X-τ parameters do not contribute to a substantial improvement in modelling accuracy.

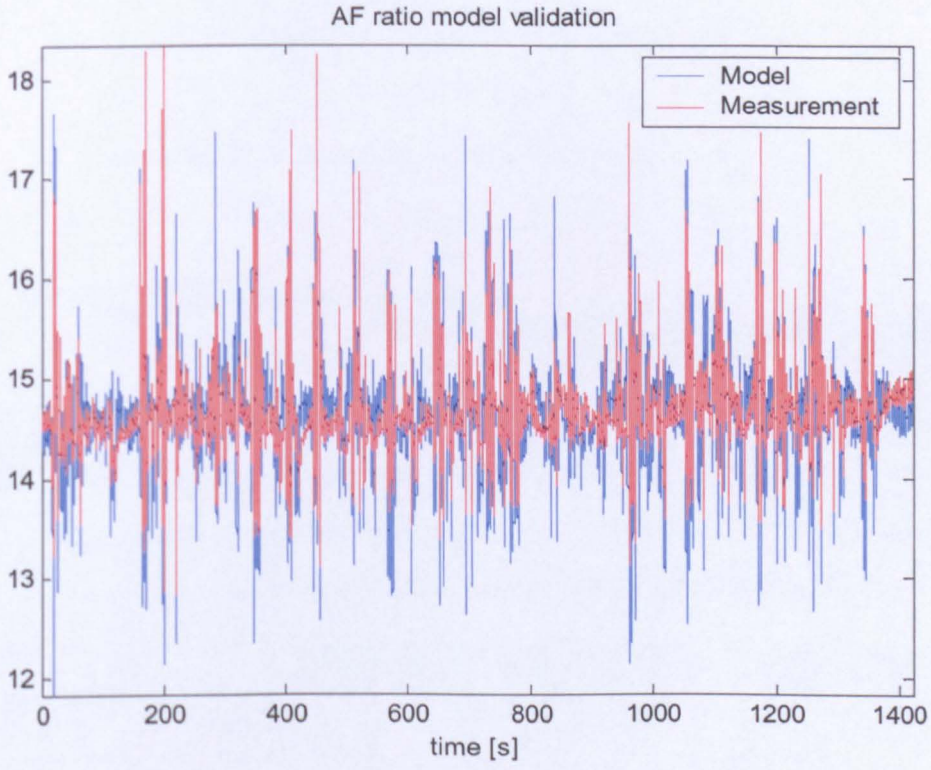

Figure 4-43: AF ratio model validation for model 3

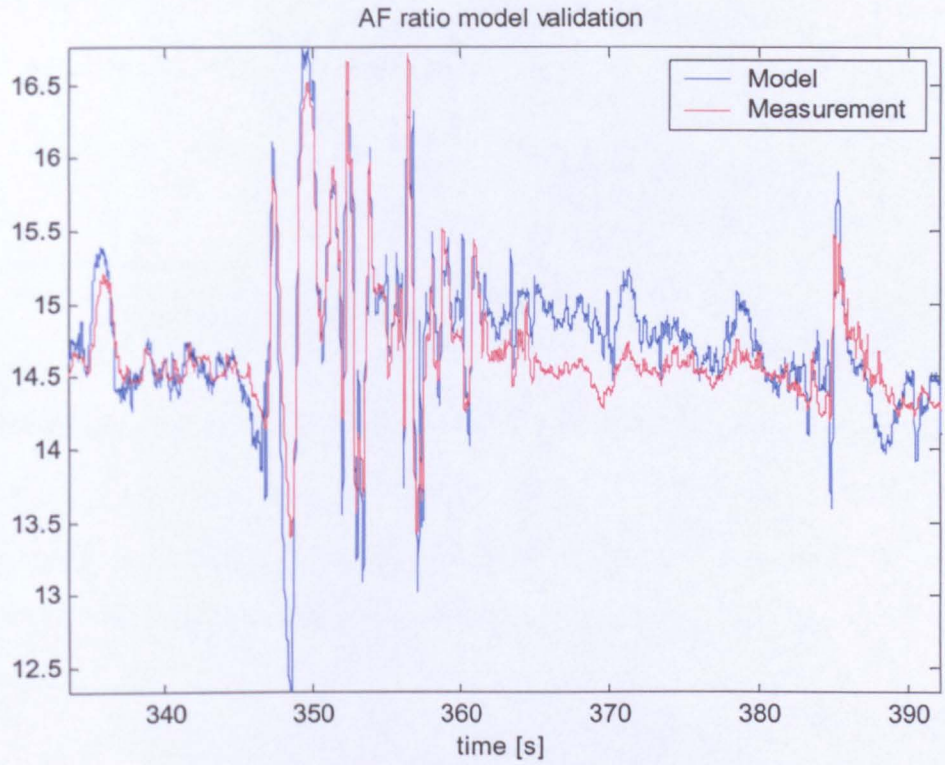

Figure 4-44: AF ratio model validation for model 3 (expanded view)

4.4.7.7 Model 3 with the correction

During the identification of the fuelling and lambda models, the cylinder air charge (CAC) was assumed to be given by the intake manifold model. In the case of a model structure mismatch (either for CAC or fuel models) or measurement errors, the lambda model becomes inaccurate. For this purpose, Model 3 (equations (4.40), (4.45) and (4.46)) is used with the already identified parameters and a static correction is developed.

$$\begin{aligned} m_{w,n+1} &= \left(1 - T_{s,n}\frac{1}{\tau_n}\right) m_{w,n} + X_n m_{i,n} \\ \lambda_{n+1} &= \left(1 - \frac{T_{s\lambda,n}}{\tau_\lambda}\right)\lambda_n + \frac{T_{s\lambda,n}}{\tau_\lambda}\frac{k_{corr,n} m_{ac,n}}{T_{s,n}\frac{1}{\tau_n} m_{w,n} + (1 - X_n) m_{i,n}} \end{aligned} \tag{4.47}$$

where

$m_{ac,n} = \dot{m}_{ac,n} T_{s,n}$ - cylinder air charge

$\frac{1}{\tau_n}, X_n, m_{i,n}$ - are given by equation (4.45)

For the extended Kalman filter parameter estimation, the $k_{corr,n}$ parameter is defined as additional state. It is assumed that $k_{corr,n}$ is modelled by the stochastic model with the white noise ξ_n. The estimation with such model generates the time series for $k_{corr,n}$ which is used for the construction of a lookup table. The method of lookup table construction is identical to that used for the throttle flow discharge coefficient and the intake-manifold volumetric efficiency modelling (see section 4.2.2).

$$k_{corr,n+1} = k_{corr,n} + \xi_n \tag{4.48}$$

4.4.7.8 Corrected Model 3 validation

The results of the model validation are shown in Figure 4-45 and Figure 4-46. The following modelling error indexes were computed as the sum of the absolute and squared values of the difference between the model and measurement lambda values: $J_{\lambda,abs} = \sum |\lambda_{model} - \lambda_{measurement}| = 19767.8396$, $J_{\lambda} = \sum (\lambda_{model} - \lambda_{measurement})^2 = 6537.3500$. The correlation of model and measurement signals was computed as: $J_{\lambda,corr} = 0.8548$. It may be noticed that the correction significantly improved both the correlation and the integrated errors. This leads to the conclusion that either the intake manifold model has a limited accuracy, or that measurements are not accurate. The correction lookup table uses the same grid as the volumetric efficiency table. It may be noticed that the resulting lookup table is noisy. However, it may be claimed that the noise will be removed if more data is used in the identification process.

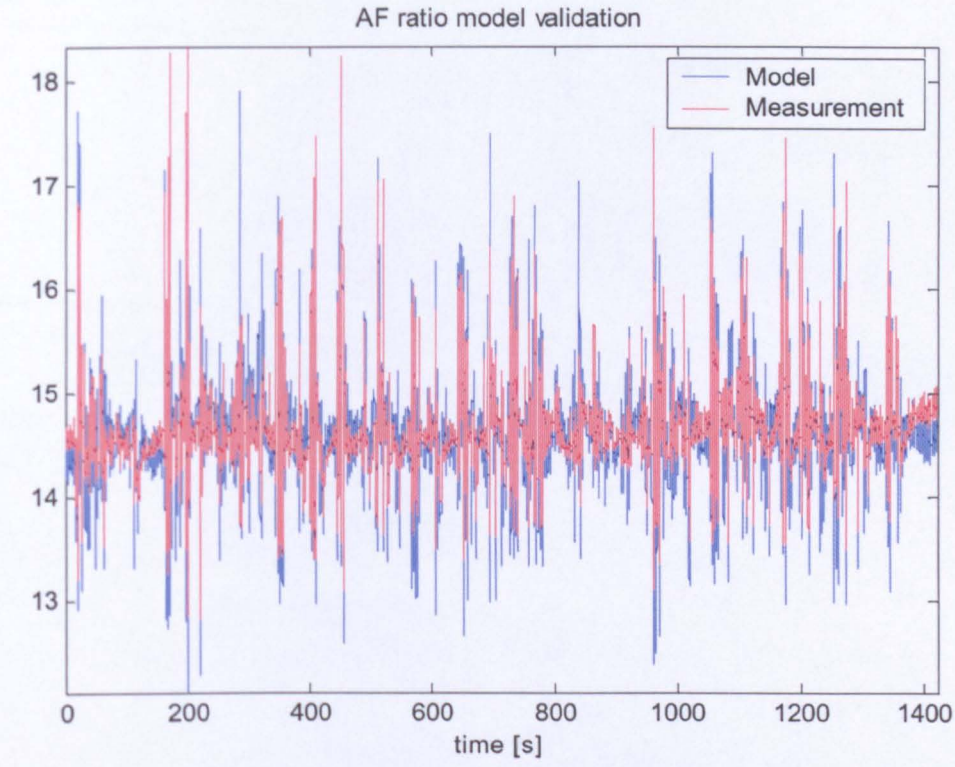

Figure 4-45: AF ratio model validation for model 3 with correction

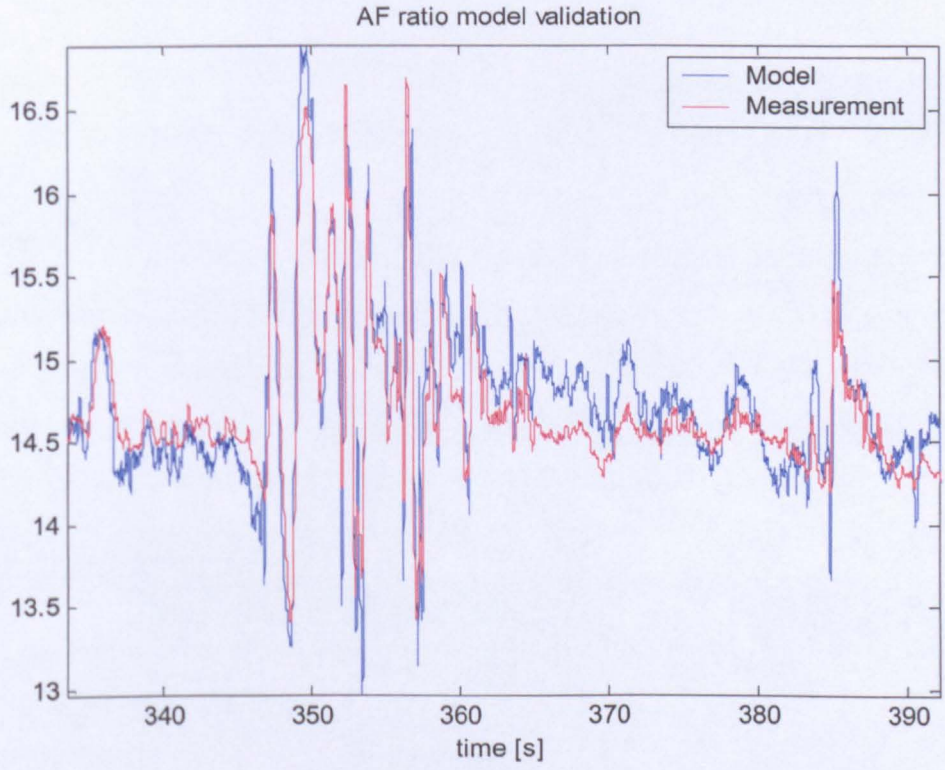

Figure 4-46: AF ratio model validation for model 3 with correction (expanded view)

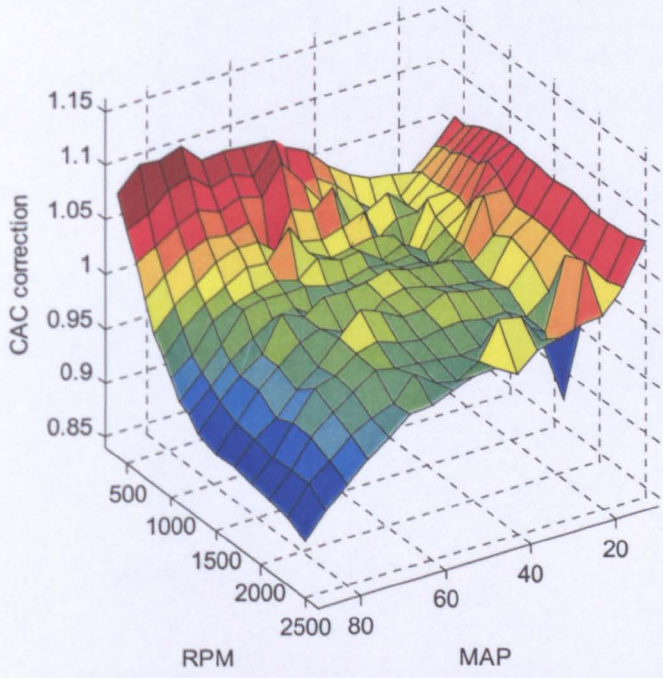

Figure 4-47: CAC correction lookup table

4.4.8 Air-fuel ratio model: analysis and conclusions

The analysis of the air-fuel ratio measurements from two sensors (one on each bank of the V8 engine) are related to the model output. Results are shown together with the modelled air-fuel ratio. Note, that the averages of the two air-fuel ratio measurements were used for the identification. It may be noticed that the differences between the air-fuel ratio measurements for each bank may be quite significant. In some parts of the trace shown in Figure 4-48 the model output follows closely one sensor and then the other. This suggests that with the uncertainty present in the measurements, further significant improvement of the air-fuel ratio model accuracy may not be possible.

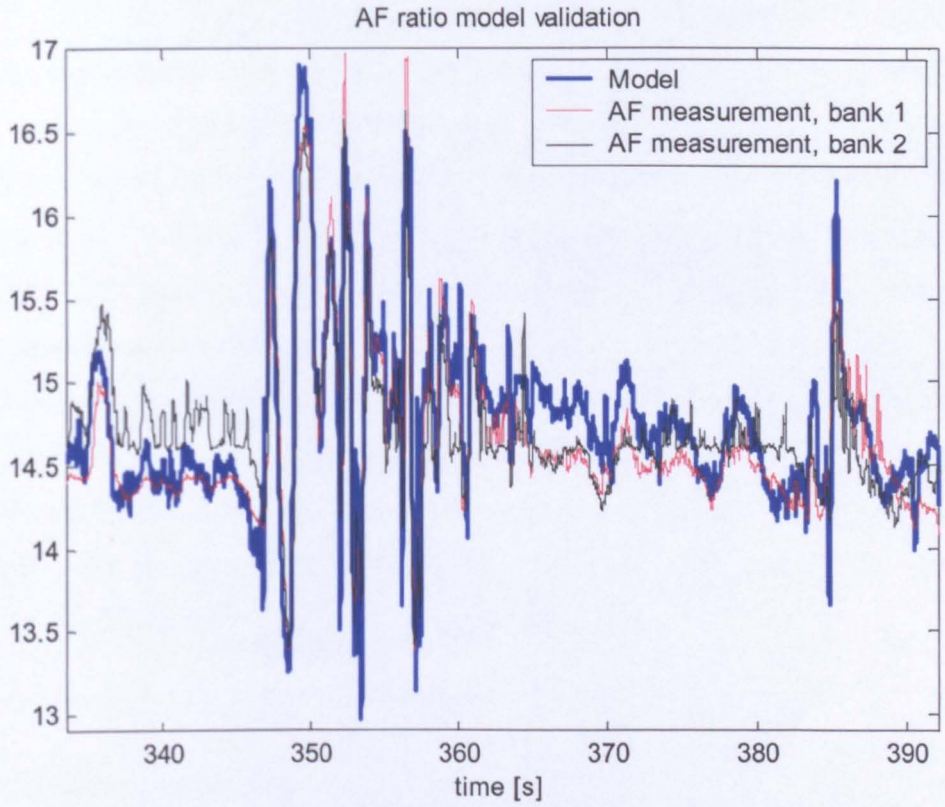

Figure 4-48: AF ratio model validation for model 3 with correction – 2 AF measurements

To draw the final conclusion about the accuracy of each model the following summary Table 4-1 is generated. It may be noticed that among models without correction lookup-table, *Model 3* gives the lowest integrated and absolute error measures. At the same time, the output of *Model 2* has the highest correlation with the measured air-fuel ratio.

Performance measure	*Model 1*	*Model 2*	*Model 3*	*Model 2 corrected*	*Model 3 corrected*
$J_{\lambda,abs} = \sum\lvert\lambda_{model} - \lambda_{measurement}\rvert$	29531.6636	23327.1708	**23233.7806**	19815.3417	**19767.8396**
$J_{\lambda} = \sum(\lambda_{model} - \lambda_{measurement})^2$	13038.3385	8902.3906	**8758.5762**	6617.7776	**6537.3500**
$J_{\lambda,corr}$	0.7049	**0.8033**	0.8015	**0.8562**	0.8548

Table 4-1: Model accuracy comparison

The most significant improvements are achieved for models with the use of correction coefficients. Again, *Model 2* and *Model 3* give very similar performance with either integrated squared or absolute error measures. These are better for *Model 3*, but correlation coefficient is lower than for *Model 2*. The reason for similar performance of *Models 2* and *3* is the nature of the validation data. For warmed-up engine the major part of the injected fuel enters the cylinder in the first event. This may suggest that the wall-wetting dynamics are less important than the fuel injector (static) characteristics. Please note that the differences between the accuracy of *Model 1* and *Models 2* and *3* are significant. This is the direct result of improved injector model. The same level of improvement is achieved by adding a static correction table to the model. This corrects for system behaviour that was not included in the model.

4.5 Net torque model

For identification of the model of net torque produced, a brake torque measurement between the engine and the transmission is utilized. The net torque produced is the gross torque generated by the combustion reduced by energy dissipation (which includes friction, pumping and the load torque of accessories attached directly to the crankshaft). The data collected during the FTP driving cycle was used for the analysis presented in this section. The block diagram describing the engine dynamics is shown in Figure 4-49.

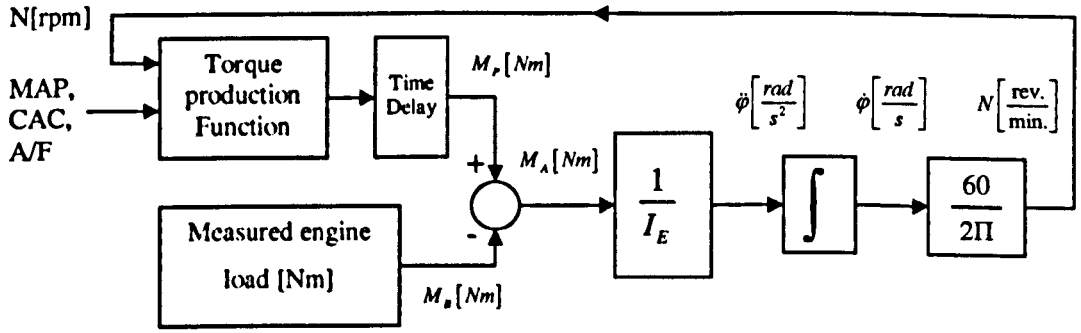

Figure 4-49: Engine dynamics block diagram

The data used in the study is sampled using the event-based sampling period (every 90° crank angle). The engine speed N_n at the event n is differentiated. First, it is represented as $\dot{\varphi}_n = \frac{2\pi}{60} N_n$ and next the acceleration $\ddot{\varphi}_n$ is computed. The acceleration is given by the following expression:

$$\ddot{\varphi}_n = \frac{\dot{\varphi}_n - \dot{\varphi}_{n-1}}{T_{s,n}} = \frac{\pi (N_n - N_{n-1}) N_n}{450} \tag{4.49}$$

The net torque is computed by the following equation:

$$M_{P,n} = M_{B,n} + I_E \ddot{\varphi}_n \tag{4.50}$$

where

$M_{P,n}$ - net torque (after subtracting friction and the accessories load) produced by the engine

$M_{B,n}$ - measured brake torque

I_E - engine inertia, (e.g. for Corvette $I_E = 0.35\left[kg \cdot m^2\right]$)

From equations (4.49), (4.50) and the FTP driving cycle datasets the net torque $M_{P,n}$ is computed. This derivation assumes that the engine inertia (with flywheel and accessories) was identified in a separate experiment and for the work presented in this chapter is known. Non-linear torque modelling employs data clustering and least-squares fitting introduced in section 4.2.2. The same method was used for the throttle flow modelling and the intake manifold volumetric efficiency lookup table construction.

The results of the non-linear torque model identification are presented in Figure 4-50 and Figure 4-51. The torque model using MAP and RPM

$(M_{P,n} = function(MAP_{n-4}, N_{n-4}))$ as scheduling inputs is shown in Figure 4-50. The intake manifold temperature influence is not considered in this model. Such a temperature-compensated model may only be used with the additional intake manifold temperature correction function. In the model that uses the Cylinder Air Charge (CAC) as an input, the temperature is already taken into account through CAC. The cylinder air charge estimated value is obtained from the equation (4.24). This model is shown in Figure 4-51. The following relationship between the net torque and the cylinder air charge / engine speed is established.

$$M_{P,n} = function(CAC_{n-4}, N_{n-4}) \tag{4.51}$$

In this study, the spark advance is assumed to be obtained from the existing controller (based on engine load and speed) and is implicitly included in the model (4.51). The torque modelling method assumes that the lambda is controlled close to unity (i.e. the air-fuel ratio at stoichiometry). Lambda deviations from that level may be regarded as a disturbance with zero mean value. In a torque/lambda control strategy where lambda is regulated at one, such a model is sufficient.

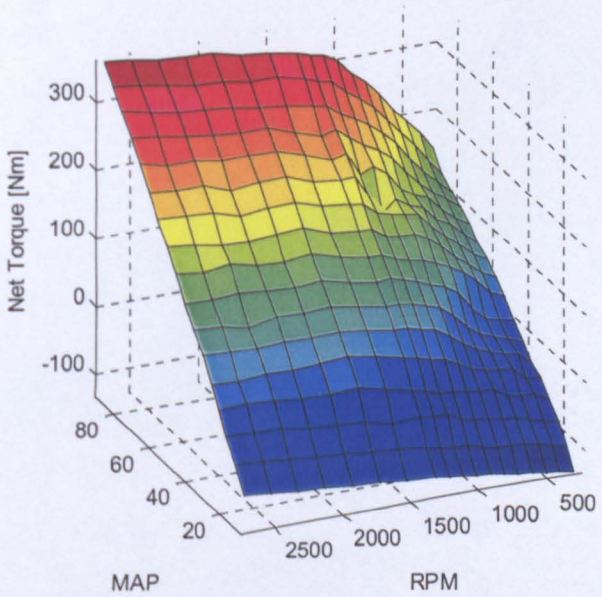

Figure 4-50: Net Torque model: f(MAP,RPM)

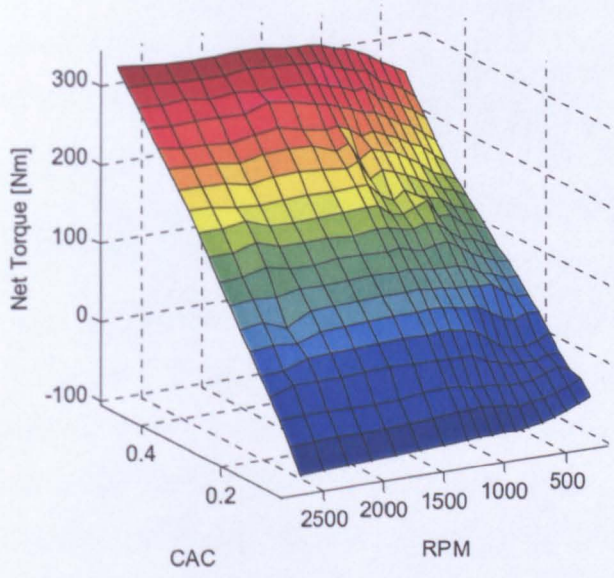

Figure 4-51: Net Torque model: f(MAP,RPM)

4.6 Summary

In this chapter the identification of the combustion engine model was presented. The methodology introduced here aimed to use as few parameters supplied by the engine manufacturer as possible. Also, the identification procedure may be adopted for the on-line operation. This in turn would allow the model adaptation that is desired since some engine parameters may be subject to gradual change as a result of wear and tear. The model presented in this chapter may be further developed. The additional model parameters may be introduced. For example, the fuel film dynamics depend upon the engine temperature. This is a very important dependence, especially for the cold start conditions. In this work only the warmed-up engine identification was considered. The data collected during the driving cycle was used for the modelling. Consequently, the identification procedure can naturally be adopted for on-line engine operation. This in turn will allow model adaptation to gradual parameter variations in engine variables in online operation of the vehicle.

Due to the engine operating cycle and a significant transport delay in the exhaust manifold, the actual air-fuel ratio is measured long after the fuel injection is completed. Therefore, accurate modelling of the engine forward path is of great importance for the precise air-fuel ratio control. The delayed measurement of the actual air-fuel ratio imposes an inherent limitation in the maximum achievable performance using any classical feedback control methods. The performance may, however, be improved through extensive use of accurate models in a feedforward control loop. The real vehicle data was used in conjunction with physical models of engine processes and system identification techniques, to determine accurate engine models. This process was executed offline, however as more powerful microcontrollers are adopted for the engine control, some of the system identification and parameter estimation techniques may be carried out in a real time. This would bring the advantage of real-time compensation of modelling inaccuracies. In addition, the use of nonlinear models so identified will reduce the required memory, development time and effort in conventional open-loop fuel control systems where dense grids are used to approximate the engine nonlinearities.

The chapter contains the following separate identification tasks: i) intake manifold model identification (air charge model) using upstream engine sensor information from the driving cycle data and, ii) fuel path identification using the measured air-fuel ratio from the driving cycle data and, iii) an approximate engine torque model where it is assumed that the air-fuel ratio is maintained at around the stoichiometric value. The air-fuel models developed have been validated using three different sets of criteria: an integrated absolute, squared error and a correlation between the measured and estimated variables. Depending on the complexity of the model structure selected, various measures of accuracy are developed and presented. Good model accuracy was achieved as more measured variables and model parameters were incorporated in the model structure. These measures include the transient as well as steady state errors in the air-fuel ratio model during the FTP driving cycle. These models are intended for control system design presented in the next chapter.

Chapter 5

Predictive Control of the Combustion Engine

The model identified in Chapter 4 will be used her for the control system design. The model identification presented in Chapter 4 used the driving cycle data for the model estimation. This implies that the methodology presented there could be embedded in the controller and used for the on-line adaptation. The aim of this chapter is to present the model-based control system design strategy. Depending on allowed complexity of the target control technique two control approaches will be considered.

The first approach called 'conventional' presented in section 5.1 will assume that there is a direct mechanical link between the accelerator pedal and the throttle plate movement. Only current throttle position is available and the control system cannot interact with that manipulated input. The predictive control strategy uses available at the time engine state information with the model. The Fuel Pulse Width command is used to change the amount of injected fuel and maintain the desired air-fuel mass ratio.

The drive-by-wire throttle control is becoming more and more common on current production engines. The drive-by-wire strategy uses the pedal position as a setpoint for the electronic throttle actuator. Between the pedal position and the throttle position setpoint signals the non-linear mapping is applied. This mapping is aimed to scale the position of the accelerator pedal with the torque achieved by the

engine in steady state. Additionally the pedal-throttle position characteristic may be used to shape the response according to driver's requirements (e.g. sporty response, more aggressive to the pedal movement). The static torque control through the throttle position mapping is presented in section 5.2. The feedforward controller is used to maintain the desired air-fuel ratio.

The drive-by-wire brings also some benefits to the conventional air-fuel ratio control. The pedal position passed through the non-linear torque mapping function drives the throttle actuator. The actuator dynamic model (given in section 4.2.1) may be used to for the future throttle position prediction. The actuator filters driver's command through its limited speed of response. This property is used in section 5.3 to improve the air-fuel ratio control precision. Finally, the multivariable control strategy is discussed in section 5.4. The analyzed approach uses a cost function since the torque-lambda control problem is a trade-off. The conclusion given at the end of this section leads to the simplified MIMO control strategy presented in section 5.4.1. In this section the throttle setpoint and the fuel pulse width signals are used to achieve two objectives: tight lambda regulation and torque tracking. The non-linear predictive control algorithm introduced in section 3.2.4 is employed there. The comparison of the simulation results obtained using the conventional and multivariable control methods is presented. The US06 – an aggressive driving cycle profile is used during simulations.

The results presented in section 5.1 are based on engine tests. The rapid prototyping dSpace equipment was used during the implementation at the GM. The controller was built in Simulink and compiled into a dSpace code. The driving tests in the GM testing facilities were carried out. The remaining results presented in this chapter are based on the simulations using the model identified in Chapter 4 and the data collected during driving cycles for unmodeled engine variables (e.g. engine speed, ambient conditions).

5.1 Conventional predictive air-fuel ratio control

The accuracy of the air charge estimates is essential for the tight air-fuel (A/F) ratio control. Due to the time delay present in the fuelling path, the future, not the current, value of the cylinder air charge (CAC) estimate is needed [91], [92], [84]. The accuracy of the estimate is determined by two important factors: model accuracy and the model input information. The model accuracy is the most important factor responsible for the feedforward controller performance. The engine modelling and identification was presented in Chapter 4. The overall model accuracy is determined by the identification procedure, sensors precision and the complexity of the model structure. The second important factor is the information about the future engine parameters that are required by the controller. Assuming that the model obtained during the identification is 100% accurate, A/F excursions are still unavoidable due to the lack of the future throttle position information. The injection must stop before the valve opens and the fuel vapour enter the cylinders. The future throttle position information is required for an accurate future air charge prediction. However, the throttle position under the command of the driver and/or the computer, is a priori unknown (i.e. depends on driving condition and the driver). An improvement may be made by introducing intentional delays [91] (or filter) in the throttle position command. Depending on the length of delay introduced, this may negatively impact the driveability. In Figure 5-1 the air-fuel ratio is shown assuming a perfect model and 6-event prediction horizon for the CAC (6 events delay from the beginning of the FPW computation to the cylinder charging event). The simulation results presented here are based on the throttle position and other engine measurements taken from the FTP driving cycle and using the model identified in Chapter 4. The throttle position is assumed to be fixed at the current indicated value over the prediction horizon for the CAC prediction calculation. The engine speed and the intake manifold pressure profiles are shown in Figure 5-3 and Figure 5-4 respectively.

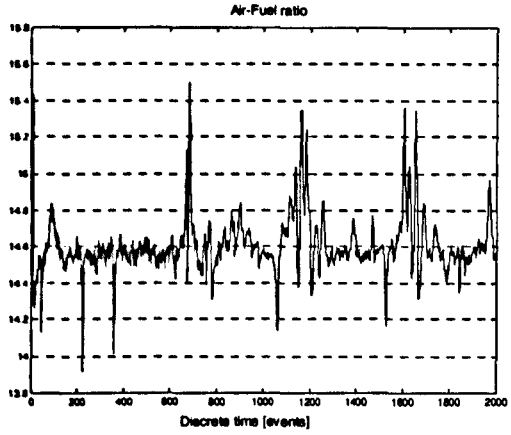

Figure 5-1: 6 events prediction (delay) - with 14.57 as target A/F

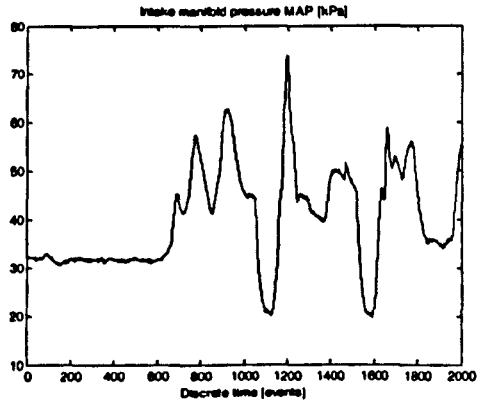

Figure 5-2: Intake manifold pressure

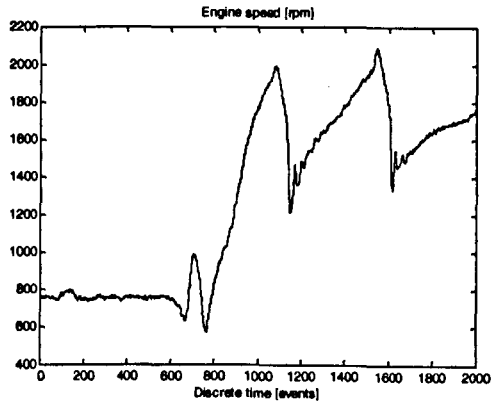

Figure 5-3: Engine speed

For more aggressive driving patterns, the lack of future throttle information will result in even higher A/F excursions. As was mentioned, other important parameters of interest within the prediction horizon are the engine speed, battery voltage and ambient conditions. The future values of these variables are available and they may change with the driving conditions (i.e. load, traffic and environment).

The most efficient way to improve the performance of A/F ratio regulation, which is affected by both the model accuracy and quality of the input information, is to reduce the prediction horizon. Results of the prediction horizon reduction to 3 and 2 events are shown in Figure 5-4 and Figure 5-5, respectively.

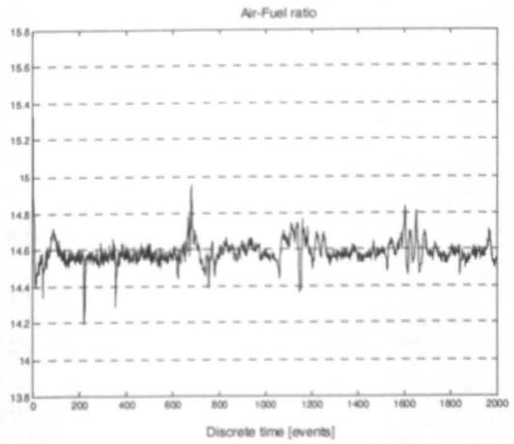

Figure 5-4: 3 events prediction (delay) (14.57 target)

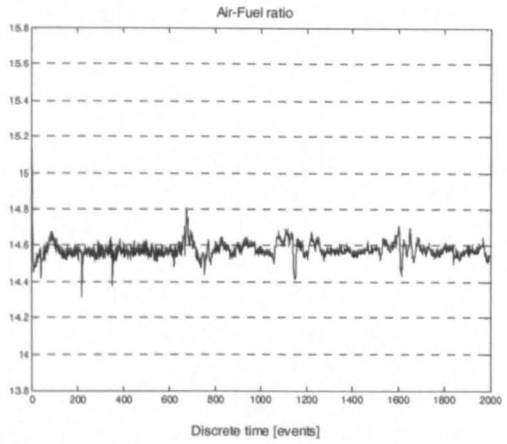

Figure 5-5: 2 events prediction (delay) (14.57 target)

The reduction of the fuel delivery delay results in lower A/F excursions due to lower uncertainty associated with the future variations of engine parameters. However, when the delay is reduced there may not be sufficient time for fuel vapour formation. This is particularly important for engine operation under cold conditions. This implies that in practice it is not possible to minimize the time delay and the aforementioned performance limitation will always be present in the system.

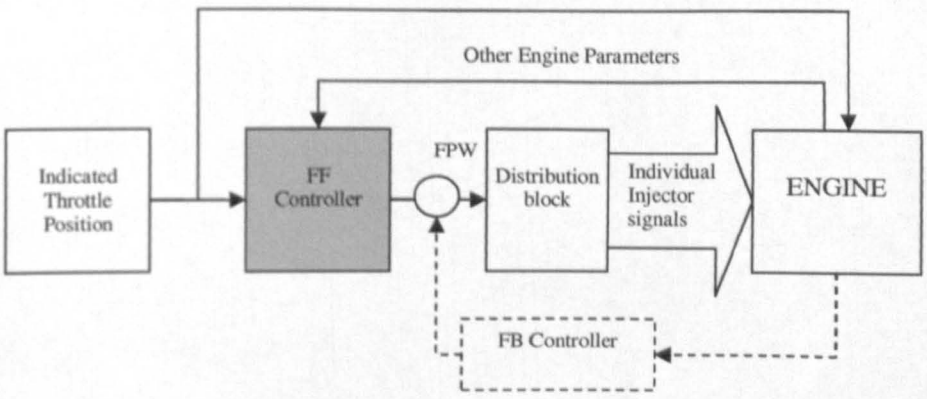

Figure 5-6: The basic control structure diagram

The feedforward controller employed in this section computes the fuel pulse width (FPW) command based on upstream (e.g. intake manifold, throttle)

measurements. The variable prediction horizon is proportional to engine speed (N) and FPW signal itself. The FPW signal is then utilized by the 'Distribution block' to generate the individual signals for injectors.

The way the 'Distribution block' operates determines the time delay length and its variability. While the additional correction from the feedback controller is incorporated for improved response, the transient performance is mostly determined by the quality of the feedforward controller. The block diagram of the control system structure is shown in Figure 5-6. In this section only the feedforward controller design is presented. The mathematical model of the controller is presented. The throttle flow and the intake manifold models are used for the future cylinder air charge (CAC) prediction. The prediction horizon is determined by the duration of the injection and the engine speed. The fuel film dynamic model together with the wall fuel mass estimate is used and the *FPW* command is computed.

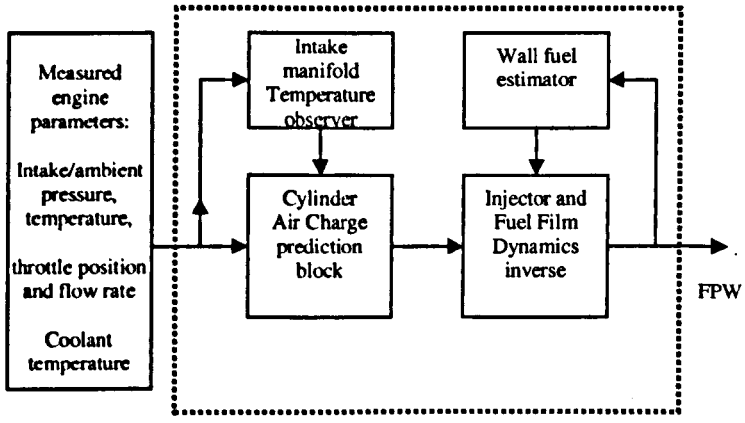

Figure 5-7: The Feed-Forward controller diagram

5.1.1 Determination of the prediction horizon

The delay between the time of fuel (or FPW) command and the cylinder charging (see Figure 4-30) are now explained.

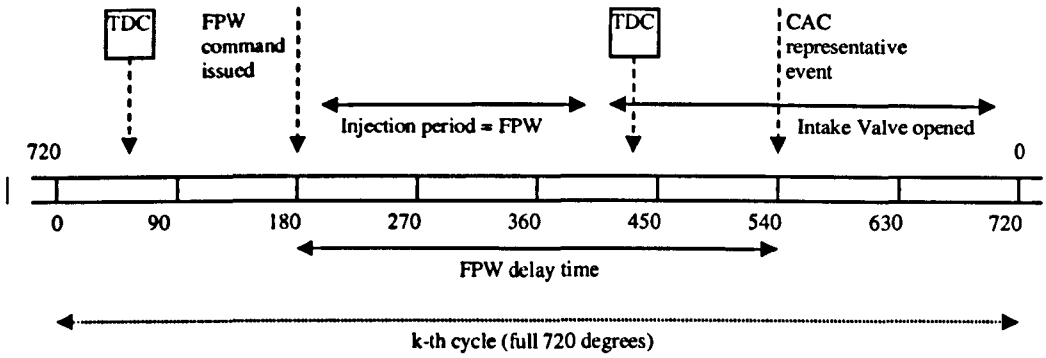

Figure 5-8: The FPW delay timing

It is assumed that the delay, expressed in crankshaft angle, consists of four components:

- Computational and data transfer delay. This delay covers the time elapsed between the start of the calculation of the FPW command, and the actual start of fuel injection. The assumptions made here are: The discrete-time fuel controller is 'called' once during each engine event, i.e. once every 90° of crankshaft angle rotation (for an eight-cylinder, four-stroke engine). The transfer delay refers to the signal transfer delay resulting from the memory read-write cycles. This is due to control algorithm implementation using rapid prototyping controllers (e.g. dSpace). The overall computational delay is assumed to be 180° of crankshaft angle rotation: $\alpha_{cd} = 180^{o}$
- FPW duration. The output of the feedforward controller is the FPW duration, given in seconds. Conversion to an angle is through multiplication by engine speed: $\alpha_{FPW} = 360 \cdot \frac{N}{60} \cdot FPW = 6 \cdot N \cdot FPW$.
- The time between the end of injection and intake valve opening. The injection strategy seeks to maintain an angle α_c (in crankshaft angle) between the end of fuel injection and the opening of the cylinder intake valve. In this case, this angle is assumed to be zero.
- Duration of cylinder intake valve opening. The actual flow of air (and fuel vapour) into the cylinder during intake valve opening is not instantaneous,

but takes place over the full duration of valve opening. It is reasonable to assume that the mean 'delay' in air and fuel charging of the cylinder is half the angle over which the intake valve is open. $\alpha_{cyl,mean} = \frac{\alpha_{ivc} - \alpha_{ivo}}{2} = \frac{637-343}{2} = 147$. The nearest discrete event is at 480°. This leads to a value for the in-cylinder delay of: $\alpha_{cyl} = 480 - 343 = 137$.

The total fuel injection delay between the fuel command and cylinder charging is therefore:

$$\alpha_{FPW\,delay} = \alpha_{cd} + \alpha_{FPW} + \alpha_{c} + \alpha_{cyl} = 317 + 6 \cdot N \cdot FPW \tag{5.1}$$

Converted to a number of engine events or controller time steps, this becomes:

$$n_{FPW\,delay} = \frac{\alpha_{FPW\,delay}}{90} = \frac{317}{90} + \frac{N \cdot FPW}{15} \tag{5.2}$$

The time delay in the delivery path may be determined from the equation (5.2). For the FTP driving cycles, the engine speed varies between 600 rpm and 2500 rpm. The fuel demand results in the FPW command varying between 2 ms to 12 ms. In the worst case, the maximum number of delays is $n_{FPW\,delay} = \frac{317}{90} + \frac{2500 \cdot 0.012}{15} = 5.52$ events. The maximum integer time delay is therefore 6 events. However, the typical delay may be shorter by one or two events. Since, in general, $n_{FPW\,delay}$ is not an integer, the nearest higher integer number is used. The integer prediction horizon length will be denoted $N_{FPW\,delay} = ceil(n_{FPW\,delay})$ (Note, *ceil(.)* – MATLAB® function).

As shown above, the estimated time delay $N_{FPW\,delay}$ depends on the applied FPW and the engine speed. The FPW in turn depends on the predicted time delay, through the prediction of the future CAC coinciding with the currently applied FPW.

In principle, this forms an algebraic loop, which would have to be solved in the controller:

$$(RPM_n, FPW_n) \rightarrow n_{FPW\,delay,n} \rightarrow CAC_n \rightarrow FPW_n$$

The algebraic loop introduces additional computational burden. Assuming that the FPW changes little from one step to the next, it is possible to use the previous FPW value in the time delay and CAC prediction:

$$(RPM_n, FPW_{n-1}) \rightarrow n_{FPW\,delay,n} \rightarrow CAC_n \rightarrow FPW_n$$

This simplified method has been employed in the FPW controller built in the Simulink code. Employing the past FPW command, the prediction horizon is computed from the $n_{FPW\,delay}$.

5.1.2 Cylinder Air Charge prediction

The Cylinder Air Charge is computed based on the measurements of the current engine states. The following states are used as inputs: engine speed, throttle position, ambient pressure and temperature, coolant temperature, intake manifold pressure, and the throttle mass flow rate.

The cylinder air charge predictor is based on the intake manifold model identified in section 4.3. The intake manifold pressure measurement is used directly as a starting point for the prediction. This is due to the relatively fast sensor that was used for the intake manifold pressure measurement. The intake manifold temperature open loop estimate is generated internally within the feedforward controller. This is due to very slow response of this element. This approach has also a practical justification. Quite often only the air temperature that enters the intake manifold is measured and in that

case the open loop temperature estimation is the only option. The length of the prediction horizon is based on the previous FPW command and the current engine speed measurement. The maximum prediction horizon is estimated to be 6 events. The output of the prediction block in Figure 5-9 is selected according to the computed delay in the fuel delivery path.

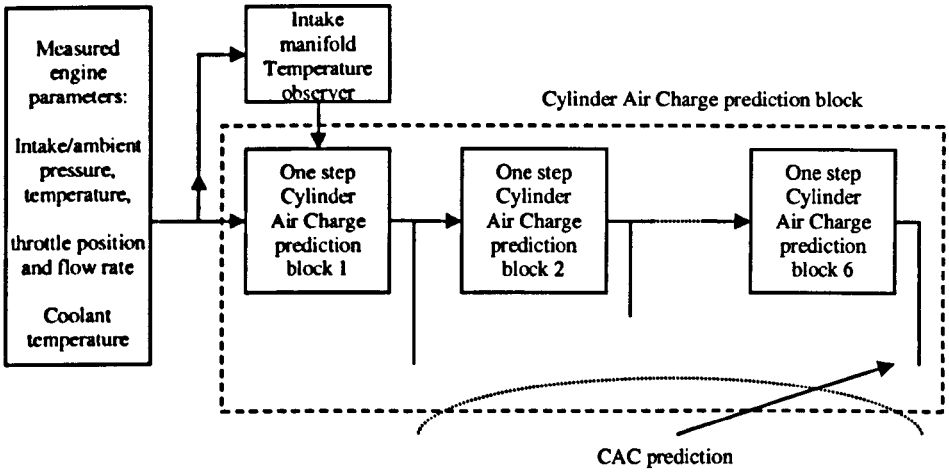

Figure 5-9: The Cylinder Air Charge Prediction block with the variable horizon

5.1.3 Wall wetting dynamics and injector compensation

The wall fuel dynamics is modelled by the X-tau model [95]. The Model 2 identified in section 4.4.7 with the cylinder air charge compensation function constructed for this model structure is employed by the feedforward (FF) controller. The fuel injection must take into account the amount of the fuel on the wall. This fuel mass is estimated in an open-loop fashion within the FF controller. The mass of the wall fuel is consequently an internal state of the controller.

The wall fuel model is given by the equation (4.40). The mass of the fuel $m_{w,n+N}$ on the wall is available from the built-in observer based on the equation (4.40). The mass of the fuel required to enter the cylinder is derived from the

equation (5.3) as the function of the cylinder air charge prediction and the air-fuel ratio stoichiometric target.

$$m_{fc,n+N} = \frac{CAC_{n+N}}{A/F_{target}} \tag{5.3}$$

From the equation (4.40) the mass of the fuel $m_{fi,n+N}$ that needs to be injected is derived using the result of equation (5.3) and the wall fuel mass estimate $m_{w,n+N}$.

$$m_{fi,n+N} = \frac{m_{fc,n+N} - \frac{T_{s,n}}{\tau} m_{w,n+N}}{(1-X)} \tag{5.4}$$

The sampling period in the equation (5.4) is inversely proportional to the engine speed ($T_{s,n} = 15/RPM_n$). The assumption that the engine speed change is negligible within the prediction horizon must be made. The injector is modelled by the equation (4.43). The FPW_n command is computed from the equation (4.43) and is given by the following equation:

$$FPW_n = \frac{m_{fi,n+N}}{k_{1,fi} + k_{2,fi}U_{batt,n} + k_{3,fi}P_{im,n}} + O_{fi} \tag{5.5}$$

The equation (5.5) provides the FPW signal that is used for the control. This formulates the Feed-Forward control strategy. Note that due to the comprehensive set of engine measurements used for the FPW signal calculation the described strategy is not purely feedforward. Some feedback elements are also present. For instance, the intake manifold pressure is one of the output measurements. The use of output measurements is a feature of feedback controller.

The control algorithm presented in this section provides the best possible control action with the assumption that driver's actions are purely stochastic. In that case the best guess about the future throttle position is to assume that it remains fixed

at current point. If however it was possible to characterize (model) driver's behaviour the extra information could be used to improve the future the cylinder air charge prediction. In the next section the driving test results will be presented to demonstrate performance of the controller. Also, the accuracy of the model used for the controller design will be confirmed in this test.

5.1.4 Engine Test Results

The driving cycle tests were carried out using the designed predictive feedforward controller. The engine model employed by the controller uses the model identified from the driving cycle data. The controller was built using Simulink™ and implemented using dSpace® rapid prototyping controllers. The test vehicle is a Chevrolet Corvette with 5.7 L V8 engine. The FTP and US06 driving cycles were employed during tests. The US06 provide a more aggressive driving pattern of driving than the FTP cycle.

Results in Figure 5-10 present lambda measurements collected in the exhaust for the gas entering the catalytic converter. For the FTP tests, engine load and speed profiles are shown in Figure 5-11. The V8 engine used in the test has two banks of cylinders and separate measurements for each were collected. Simulation results are presented in Figure 5-12. The data show the best possible measured lambda trace that may be achieved assuming perfect model information and perfect measurements are available. The lambda excursions visible there reflect only the uncertainty introduced by the inaccuracies in the prediction of future cylinder air charge estimate.

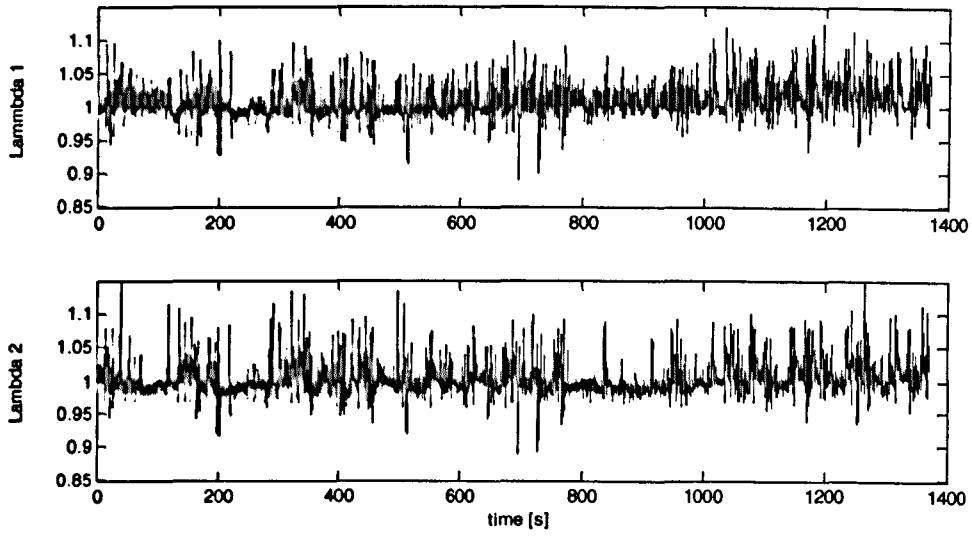

Figure 5-10: FTP driving cycle test: pre-catalyst lambda for 2 banks of cylinders

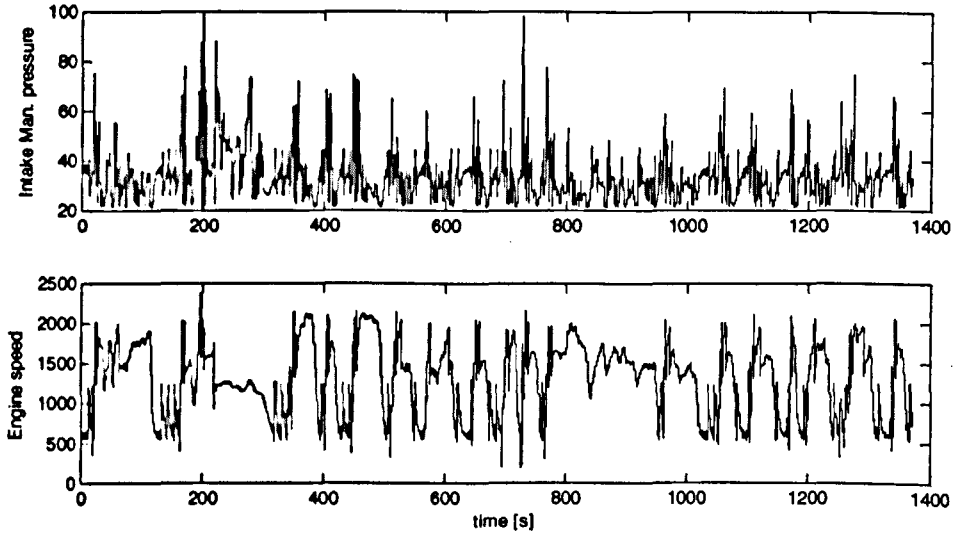

Figure 5-11: FTP driving cycle intake manifold pressure and speed profiles

It should be noticed that the lambda variations in real-time test results do not differ significantly from the simulated results shown in Figure 5-12. This indicates good accuracy of the model.

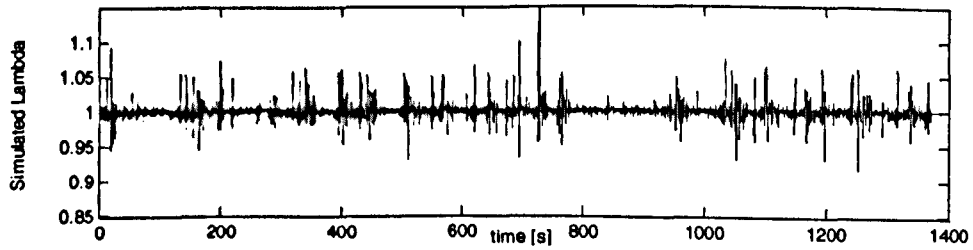

Figure 5-12: FTP driving cycle test: simulated pre-catalyst lambda

The driving test results with the US06 cycles are presented in Figure 5-13. Simulation results for US06 cycle are presented in Figure 5-14. The data show the best possible measured lambda trace that may be achieved assuming perfect model information and perfect measurements are available (e.g. the upper bound for the performance). It should be noticed that the lambda variations for real-time test results do not differ significantly from the simulation results shown in Figure 5-14. The intake manifold pressure and engine speed profile of the US06 driving cycle is presented in Figure 5-15.

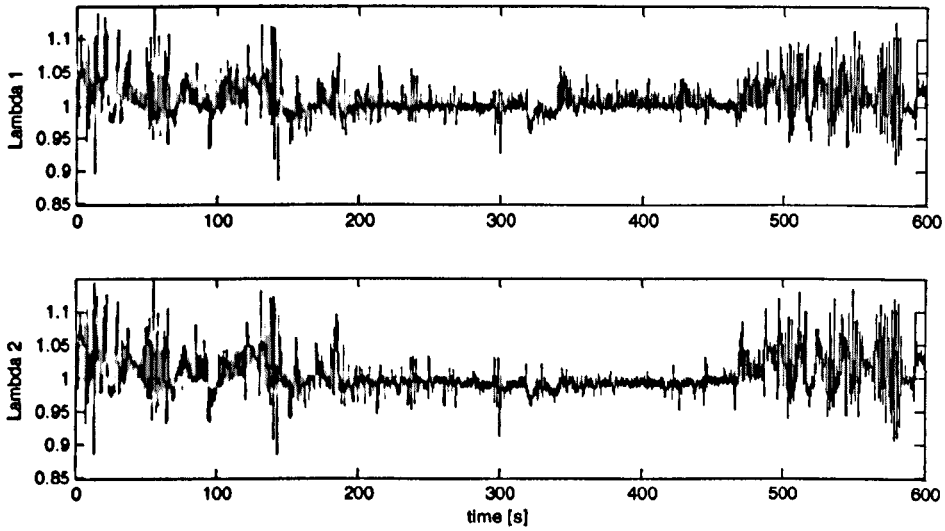

Figure 5-13 US06 driving cycle test: pre-catalyst lambda for 2 banks of cylinders

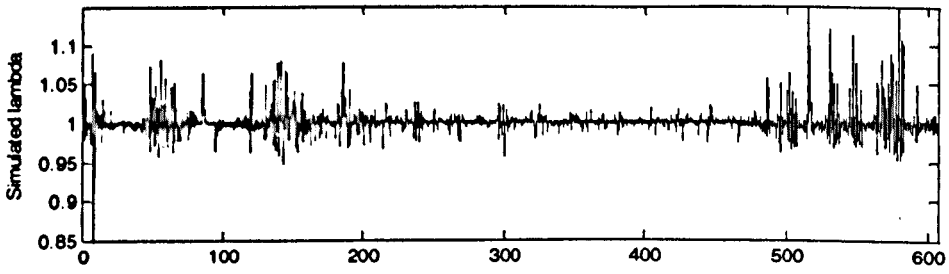

Figure 5-14: US06 driving cycle test: simulated pre-catalyst lambda

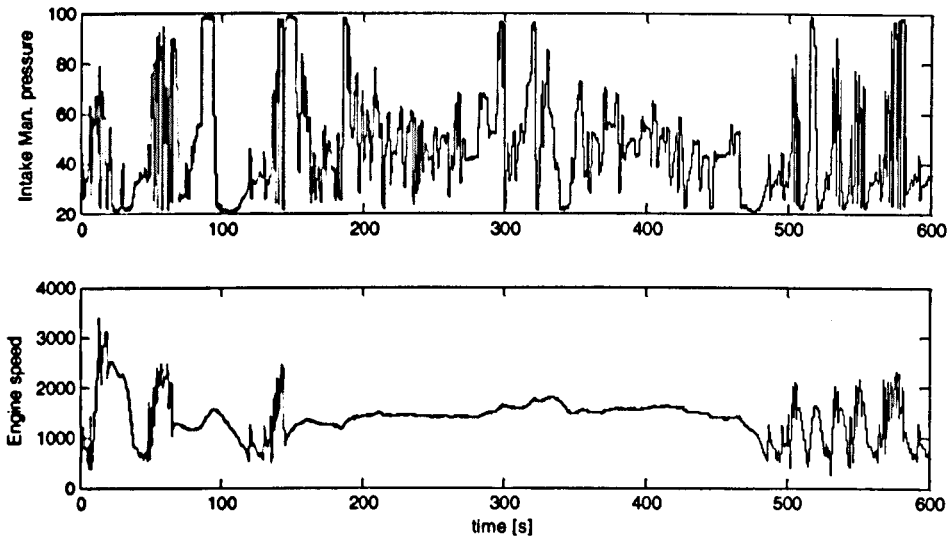

Figure 5-15: US06 driving cycle test pressure and speed profiles

5.2 The Basic Torque Controller

In the previous section the air-fuel ratio control was the only task of interest for the control system. The fuel pulse width command was computed as a function of measured engine parameters and current indicated throttle position that was assumed to be time-invariant in the future. The approximate torque model was identified in section 4.5 with the assumption that the lambda is controlled with sufficient accuracy and the air-fuel ratio remains at stoichiometry. During modelling it was assumed that the spark advance signal was derived from the engine load and implicitly included in the estimated torque model. Such a model is sufficient for a control strategy that aims to maintain stoichiometry at all times. The static (feedforward) torque control system with conventional air-fuel ratio control is shown in the block diagram in Figure 5-16. The air-fuel ratio is controlled through the fuel pulse width command by the feedforward controller. The feedback controller may also be included in the structure. The throttle angle command is derived from the accelerator pedal position and it controls the engine torque. The conventional torque control strategy is based on a static mapping. The throttle angle that provides desired torque in steady state is a function of the pedal position and other engine parameters. The most important

parameter that influences the conversion of the pedal position into desired throttle angle is the engine speed. The basic torque control strategy is purely static. The air-fuel ratio is independently regulated at stoichiometry by the feedforward (and feedback) controller.

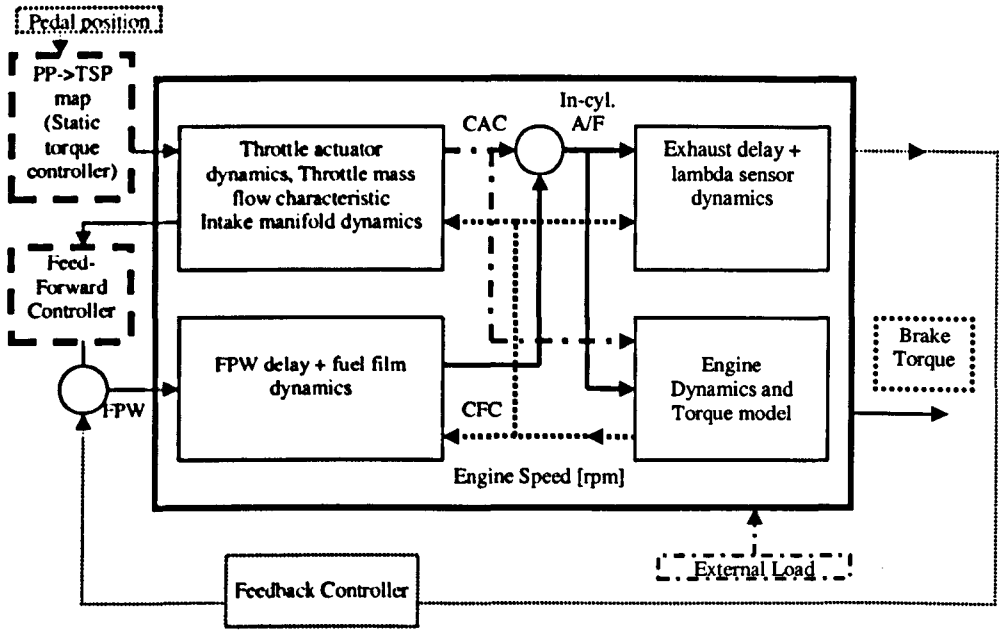

Figure 5-16: Air-Fuel ratio control diagram

The static torque control strategy is based on the engine model identified in Chapter 4. The pedal position is used as an indication of the torque demand. According to the model, the maximum and minimum achievable torque for current operating conditions is determined mainly by engine speed, but also by ambient air conditions and heat transfer in the intake manifold. The maximum and minimum static torque is computed from the engine model and sensor measurements.

The net torque setpoint $M_{SP,n}$ is computed from the pedal position using the equation (5.6).

$$M_{SP,n} = M_{P,\min,n} + \overline{PPS}_n \cdot \left(M_{P,\max,n} - M_{P,\min,n}\right) \tag{5.6}$$

where

$M_{P,min,n}$ - minimum achievable torque at current operating conditions

$M_{P,max,n}$ - maximum achievable torque at current operating conditions

$\overline{PPS_n}$ - scaled pedal position ($PPS = 0...1$).

Next, the cylinder air charge (CAC) required to achieve the demanded torque is computed from the model identified in section 4.5: $M_{P,n} = function(CAC_{n-4}, N_{n-4})$. It is assumed that the fuel is regulated by the feedforward (and feedback) controller such that the stoichiometric air-fuel ratio is maintained. The pedal position that maintains the required steady state airflow rate must be computed. For this purpose, the steady state intake manifold air temperature is computed from the following equation:

$$T_{im,n} = \frac{\kappa R_{air} T_a \frac{CAC_n \cdot N_n}{15} + (\kappa - 1)(h_1 T_{coolant} + h_2 T_a)}{\kappa R_{air} \frac{CAC_n \cdot N_n}{15} + (\kappa - 1)(h_1 + h_2)} \tag{5.7}$$

where

CAC_n - required cylinder air charge obtained from the torque map

N_n - engine speed [rpm]

κ - ratio of specific heats for dry air

R_{air} - ideal gas constant for dry air

h_1 - heat transfer coefficient (from engine)

h_2 - heat transfer coefficient (from ambient temperature)

P_{im} - intake manifold pressure [kPa]

T_{im} - intake manifold temperature [K]

T_a - ambient temperature [K]

$T_{coolant}$ - engine coolant temperature [K]

$\dot{m}_{ac}$ - air flow rate through the intake valve (in-cylinder) [g/s]

R_{air} - gas constant $[J/gK]$

V_{im} - intake manifold volume $[[dm^3]]$

After the steady state temperature T_{im} is calculated, the intake manifold pressure is computed from the following non-linear equation:

$$\eta(P_{im,n}, N_n) P_{im,n} = \frac{8 R_{air} T_{im,n}}{V_d} \frac{15 \dot{m}_{ac,n}}{N_n} \tag{5.8}$$

where

V_d - engine displacement $[dm^3]$

η - volumetric efficiency [-]

Finally the throttle angle α that results in the required flow rate may be computed. This requires a solution of the following non-linear equation:

$$C_d\left(\alpha, \frac{P_{im}}{P_a}\right) \cdot A_{th}(\alpha) = \frac{CAC_n}{\frac{15}{N_n} \Psi(P_a, P_{im,n}, T_a)} \tag{5.9}$$

where:

$\dot{m}_{at}$ - air flow rate through the throttle [g/s]

C_d - discharge coefficient

$A_{th}(\alpha) = \pi \cdot R_{th}^2 \left(1 - \frac{\cos(\alpha + \alpha_0)}{\cos(\alpha_0)}\right)$ - throttle cross-sectional area

P_a - upstream pressure (ambient)

α - throttle angle [rad]

α_0 - throttle offset angle (minimum throttle angle)

R_{th} - radius of the throttle

$$\Psi(P_a, P_{im,n}, T_a) = \begin{cases} \dfrac{P_a}{\sqrt{R_{air} \cdot T_a}} \cdot \left(\dfrac{P_{im}}{P_a}\right)^{\frac{1}{\kappa}} \sqrt{\dfrac{2\kappa}{\kappa-1}\left[1-\left(\dfrac{P_{im}}{P_a}\right)^{\frac{\kappa-1}{\kappa}}\right]} & \text{if } \dfrac{P_{im}}{P_a} > \left(\dfrac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \\ \dfrac{P_a}{\sqrt{R \cdot T_a}} \cdot \sqrt{\kappa}\left(\dfrac{2}{\kappa-1}\right)^{\frac{\kappa+1}{2\cdot(\kappa-1)}} & \text{if } \dfrac{P_{im}}{P_a} \le \left(\dfrac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \end{cases}$$

The above procedure may be implemented using simplifications and/or lookup tables. Finally, the multidimensional lookup table that returns the required throttle angle as a function of pedal position, engine speed, ambient temperature, pressure and coolant temperature can be constructed. This lookup table provides the simplest static torque controller. A variant of the torque control method described above is used in current production systems and will not be elaborated in more detail. However, the proposed model based procedure linked to the identification method presented in Chapter 4 will minimize the development time for dynamometer tests and simplify the engine calibration procedure.

5.3 Conventional predictive air-fuel ratio control including throttle actuator dynamics

The conventional predictive air-fuel ratio feedforward controller presented in section 5.1 was designed with the assumption that the throttle position is fixed over the prediction horizon. This assumption does not always hold true as the throttle position is manipulated by the driver. For an engine equipped with the electronic throttle control, the pedal position is passed through a non-linear function that maps the pedal position into the throttle position. The non-linear function reflects the engine torque characteristic that was the subject of analysis in section 5.2.

The throttle actuator model identified in section 4.2.1 may be used for the prediction of the future throttle position. In this approach, the current pedal position is assumed to be fixed (at the current level) over the horizon and the current measured throttle position is used as the starting point. The only modification in the

conventional air-fuel ratio controller is with respect to the computation of the cylinder air charge (CAC) prediction. The modified CAC prediction block is presented in Figure 5-17. Six one-step throttle prediction blocks are added to the controller. Output of each block is linked with the corresponding CAC prediction block. In that way the future throttle position information is improved by using the information about the throttle dynamics.

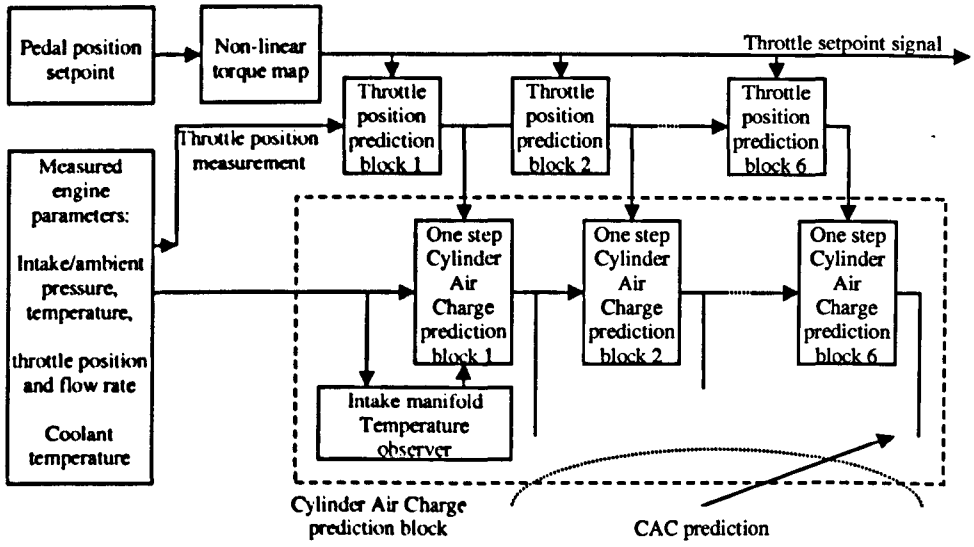

Figure 5-17: The Cylinder Air Charge Prediction block with the variable horizon

5.3.1 Impact of throttle model accuracy on CAC precision

The dynamic throttle actuator model was identified in section 4.2.1 using the linear structure. The air-fuel ratio control precision was significantly improved when the feedforward controller utilized knowledge of the throttle setpoint command combined with the model and the throttle position measurement. The approach taken, however, assumed that the (throttle) model was perfect. No model mismatch or process noise was considered. Clearly, such omissions could lead to inaccuracies and hence have a negative impact on the accuracy of the cylinder air charge prediction calculation. The cylinder air charge prediction accuracy is assessed using the

structure presented in Figure 5-18 and compared with the controller that uses only the current indicated throttle measurement (Figure 5-19).

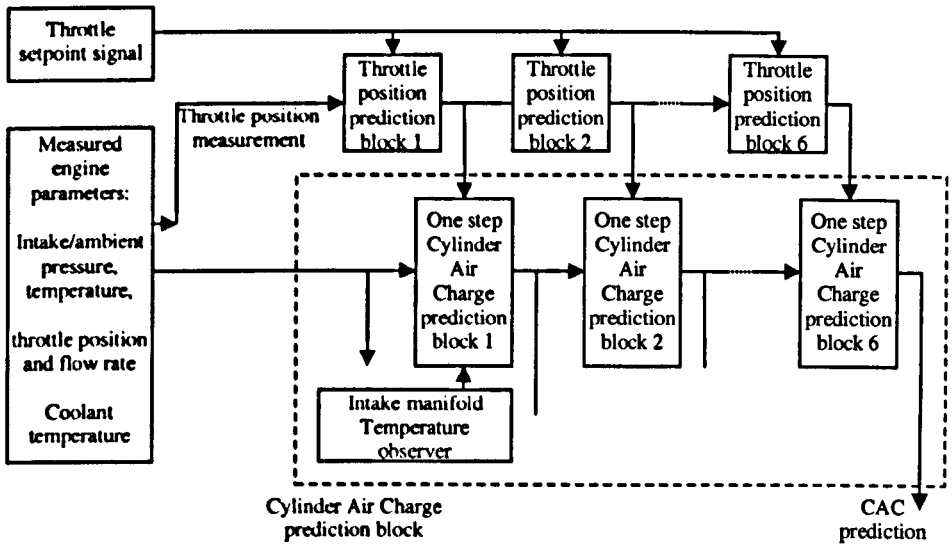

Figure 5-18: System diagram: controller prediction block – with throttle trajectory

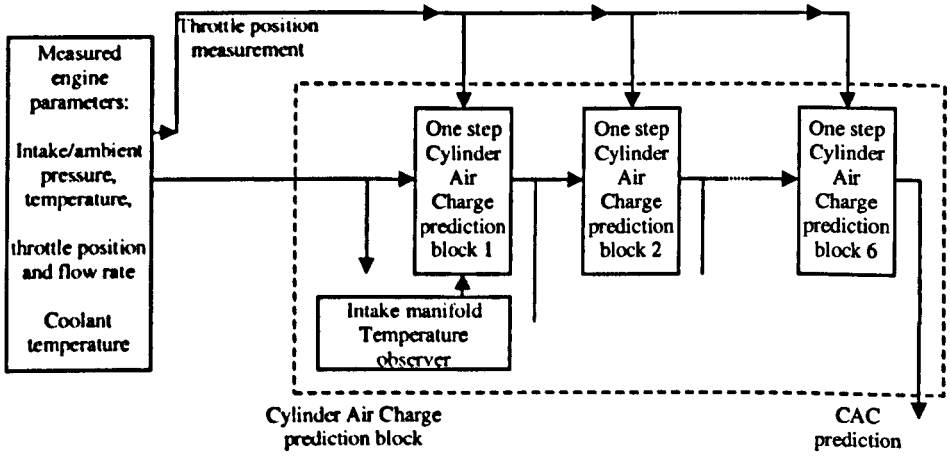

Figure 5-19: System diagram: controller prediction block – with fixed at current value throttle trajectory

The intake manifold block model structure used for the simulation is presented in Chapter 4. Two alternative setups are used. The first employs only the throttle measurement logged in the dataset. The second uses the throttle setpoint signal and computes the throttle angle based on the throttle model. The first setup, in

conjunction with the throttle actuator model used for the CAC prediction, provides a realistic robustness test. The simulation test results provide information on controller performance in the presence of modelling errors.

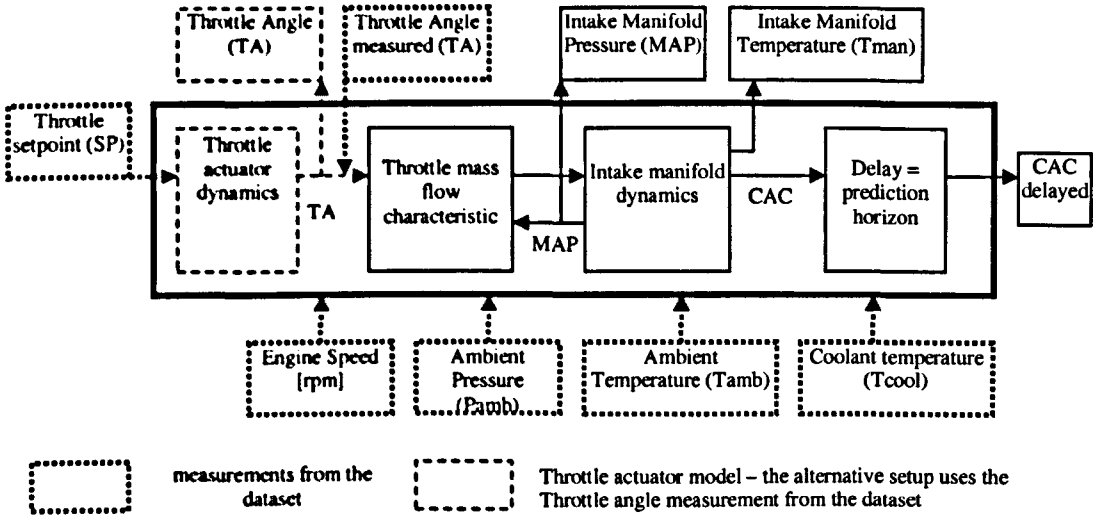

Figure 5-20: Engine simulation block

The simulation results are summarized in Table 5-1. Cylinder air charge prediction accuracy results obtained in four simulation setups are presented. Two main groups are:

- engine simulation where the throttle flow characteristic and the intake manifold model are used with indicated throttle position as an input
- engine simulation where the *throttle position dynamic model*, throttle flow characteristic and the intake manifold model are used with *throttle position setpoint* as an input

Both simulation setups are used with the cylinder air charge predictor, which assumes that either the throttle position remains fixed over the prediction horizon or the throttle position trajectory prediction is computed. The throttle trajectory prediction is obtained from the setpoint, current throttle position measurement and the throttle model.

Throttle position prediction has greatly improved the CAC prediction accuracy. When model mismatch is present in the system (row 2 and 3 in Table 5-1 or in Figure 5-21), substantial improvement can still be achieved. In Figure 5-21 the simulation results with the model mismatch being created in the controller are presented. The improvement resulting from the introduction of better future throttle position prediction is substantial. This improvement is, however, less significant than the result achieved from the theoretical analysis (rows 4 and 5 in Table 5-1 or in Figure 5-22). In Figure 5-22 the simulation results with the ideal model assumption are presented. Again, the improvement resulting from the introduction of better future throttle position is substantial. In Figure 5-21 and in Figure 5-22 the trace of the CAC error is shown. The error is required to stay as low as possible.

Throttle Model	Future throttle prediction in controller	$\sum\lvert CAC_{actual} - CAC_{predicted}\rvert$	$\sum(CAC_{actual} - CAC_{predicted})^2$
Throttle direct measurement	Throttle direct measurement fixed over prediction horizon	7501.9	39700
Throttle direct measurement	Throttle prediction obtained from current throttle position, throttle setpoint and the model	5965.5	29734
Throttle modelled	Throttle direct measurement (modelled) fixed over prediction horizon	6069	35317
Throttle modelled	Throttle prediction obtained from current throttle position (modelled), throttle setpoint and the model	3938.7	26767

Table 5-1: Simulation results with throttle measurements/throttle model and with/without throttle position prediction

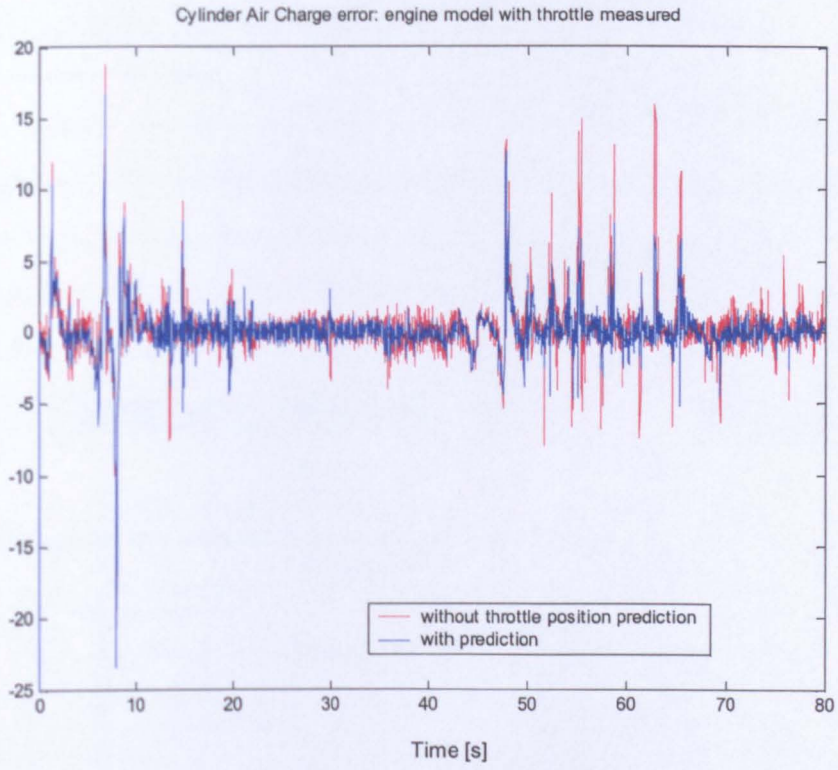

Figure 5-21: Simulation results with the throttle measurements used as an input to the engine simulation model

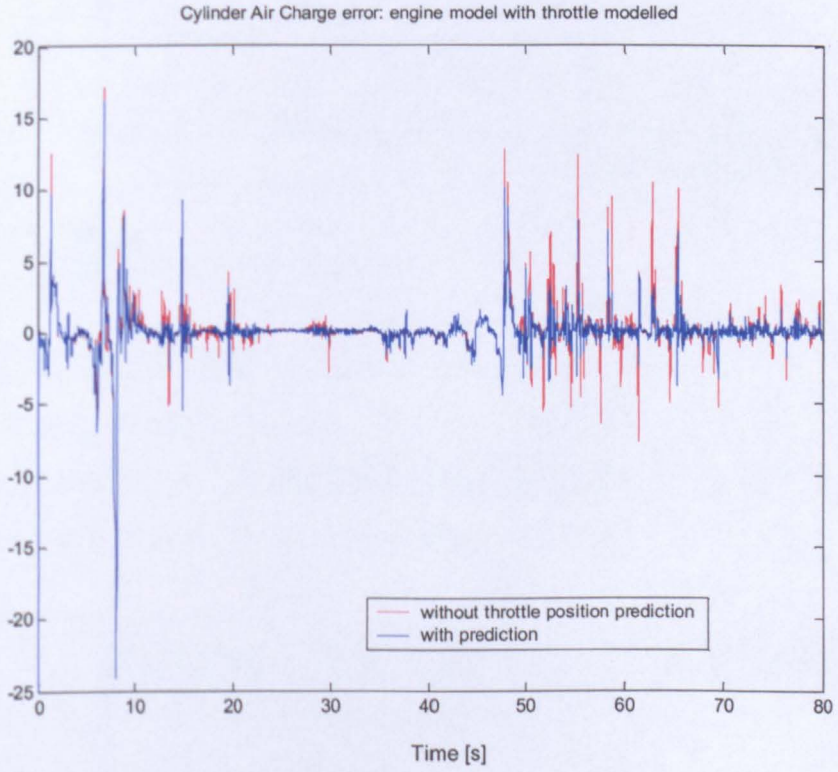

Figure 5-22 Simulation results with the throttle setpoint used as an input to the engine simulation with the throttle model

5.4 Multivariable Torque and Air-Fuel Ratio control

The MIMO control task may be formulated as a mixed tracking and regulation problem. The net torque produced by the engine must follow the driver's requirements expressed by the accelerator pedal position. At the same time, the air-fuel ratio must be regulated at the stoichiometric value. Manipulated inputs are the throttle position setpoint signal and the fuel pulse width command. The block diagram of the MIMO control system is shown in Figure 5-23.

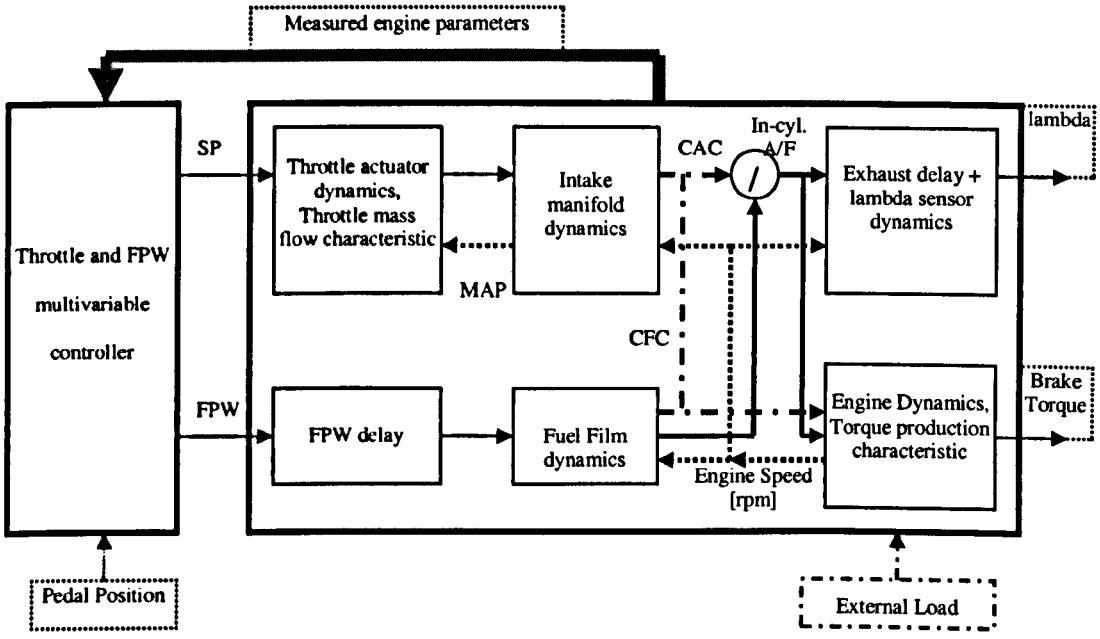

Figure 5-23: The MIMO control problem diagram

It is also known that the control solution is a trade-off between torque tracking and air-fuel ratio regulation. The trade-off may be formulated as a cost function. Minimization of the performance index leads to the optimal control solution. The cost function structure given by the equation (5.10) will be considered.

$$J_i = \sum_{k=i}^{i+T_{PRED}} \left\{ \begin{array}{l} \|AF_{k+1} - AF_{Stoich}\| + \|M_{P,n+4} - M_{SP,n+4}\| + \\ \|FPW_n - FPW_{n-1}\| + \|SP_k - SP_{k-1}\| + \|FPW_n\| \end{array} \right\} \tag{5.10}$$

Where

$\|AF_{k+1} - AF_{Stoich}\|$ - measure of the air-fuel regulation error

$\|M_{P,n+4} - M_{SP,n+4}\|$ - measure of the torque tracking error – 4 events delay is incorporated in this measure

$\|FPW_n - FPW_{n-1}\|$ - measure of the fuel pulse width increment

$\|SP_k - SP_{k-1}\|$ - measure of the throttle setpoint increment

$\|FPW_n\|$ - measure of the fuel pulse width duration (may be used for minimization of the fuel consumption)

The cost function (5.10) is minimized subject to the system dynamics given in a general form by the equation (5.11).

$$\begin{aligned} x_{n+1} &= f(x_n, N_n, SP_n, FPW_n) \\ \begin{bmatrix} M_{P,n} \\ AF_n \end{bmatrix} &= h(x_n, N_n, SP_n, FPW_n) \end{aligned} \tag{5.11}$$

Additionally, the system constraints on inputs given by the equation (5.12) must be considered during the optimization.

$$\begin{aligned} TPS_{\min} \le SP_n \le TPS_{\max} \\ FPW_n \ge 0 \end{aligned} \tag{5.12}$$

There are two questions that have to be answered before the control strategy is designed:

The first one is 'how is the cost function (5.10) formulated?' The norms $\|\ \|$ used in the cost function must be defined. In reality a simple quadratic norm may not reflect the actual requirements. Also, special care must be taken in choosing the weights in the cost function. While quadratic norm is preferred due to simplicity of the optimization solution it may be desired to use other norms like L^1 or L^2. It may also be beneficial to introduce hard constraints on the error. That is especially important

for the air-fuel ratio control since large excursions may lead to misfires and even may stall the engine.

The second important question is 'how is the minimization of the cost function (5.10) carried out?' The minimization, subject to non-linear system dynamics (5.11) and constraints (5.12), is not a trivial task. It may not be possible to guarantee convergence to the true minimum as the solution may be trapped in one of local minima leading to a non-optimal performance. Also, the computational burden associated with the optimization is substantial. Simulation tests using a simple quadratic cost function (performed with MATLAB® Optimization toolbox) indicated aforementioned convergence problems.

The above problems pose serious difficulties that can be faced if the usual MIMO control strategies based on the minimization of a cost function (5.10) were to be used. Moreover, to take into account the dependency of the torque on the air-fuel ratio, the torque model with an A/F correction coefficient shown in Figure 5-24 is introduced.

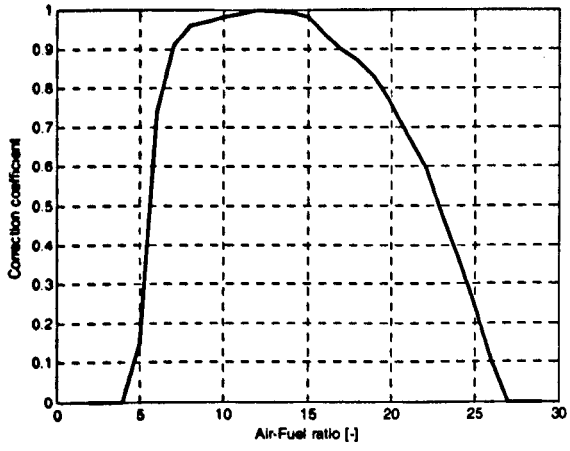

Figure 5-24: Air-Fuel ratio torque correction coefficient

The torque model $M_{P,n} = function(CAC_{n-4}, N_{n-4})$ presented in section 4.5 is combined with the air-fuel ratio correction shown in Figure 5-24 [90]. The resulting torque model block diagram is shown in Figure 5-25.

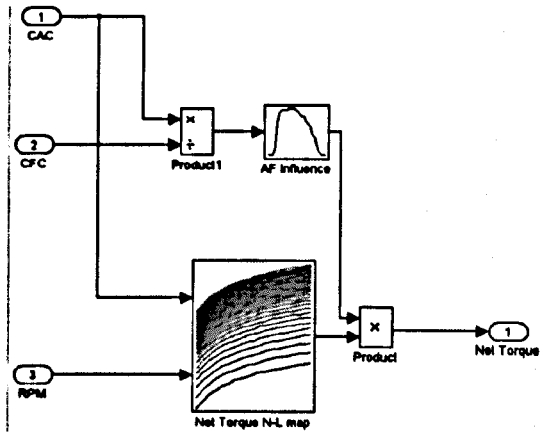

Figure 5-25: Net Torque production model

In the optimization criterion given by (5.10), the goals of torque tracking problem and air-fuel ratio regulation may, however, be in conflict. Since torque generation is a function of the air-fuel ratio, the optimal control algorithm may try to maximize the torque by reducing the air-fuel ratio to a value where the coefficient in Figure 5-24 has its maximum value. Also, to reduce the torque, minimization of the cost function may indicate that the optimal solution is to reduce/increase the air fuel ratio to obtain required torque tracking performance. To prevent this from happening, the penalty function defined by the norm $\|AF_{k+1} - AF_{Stoich}\|$ on the air-fuel ratio excursion from stoichiometry needs to be increased significantly. This however results in the air-fuel ratio kept at the stoichiometric value at all times. The engine block diagram that will help to analyze the system properties is shown in Figure 5-26.

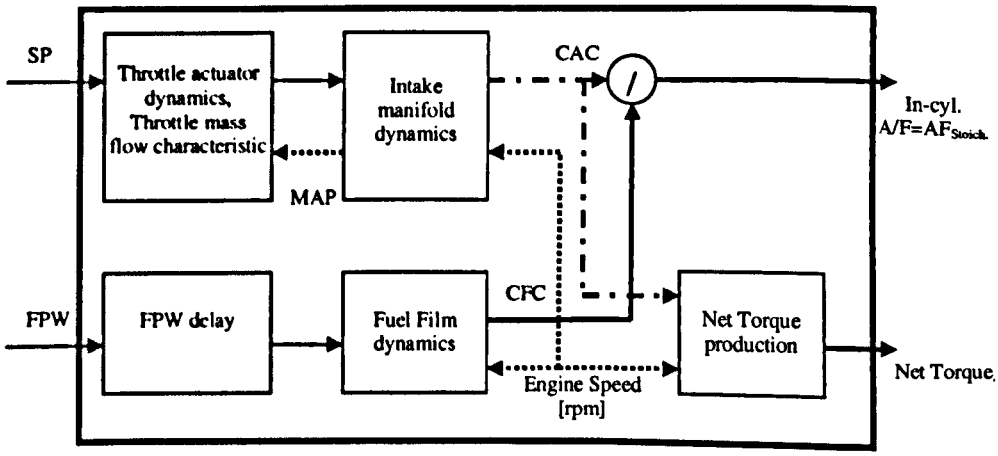

Figure 5-26: Engine block diagram

The complex problem of optimizing the cost function (5.10) may be simplified further using the following arguments. Since the excursion penalty in $\|AF_{k+1} - AF_{Stoich}\|$ must be high, it gives the effect that $CAC/CFC = AF_{Stoich}$. The AF_{Stoich} is constant thus the CAC and CFC should be controlled such that $CAC = CFC \cdot AF_{Stoich}$. The torque setpoint M_{SP} may be expressed by CAC_{SP} based on the torque model $M_{P,n} = function(CAC_{n-4}, N_{n-4})$ and current engine speed N_n. This recasts the torque tracking and air-fuel regulation problem into a CAC and CFC tracking problem with the CAC_{SP} and the $CFC_{SP} = CAC_{SP}/AF_{Stoich}$ respectively. The detailed analysis of this control strategy will be in the next section.

5.4.1 Simplified MIMO control strategy

In the conventional air-fuel ratio control, the throttle position is directly controlled by the driver. The result of this approach is an indirect torque control presented in section 5.2. The fuel is directly controlled by the feedforward (and feedback) controller. In the multivariable control approach, the throttle position and the fuel pulse width are both manipulated with the objectives of meeting driver's torque demand (determined by the pedal position) and regulation of air-fuel ratio at the stoichiometric value.

The block diagram of the fuel and air delivery paths is shown in Figure 5-27.

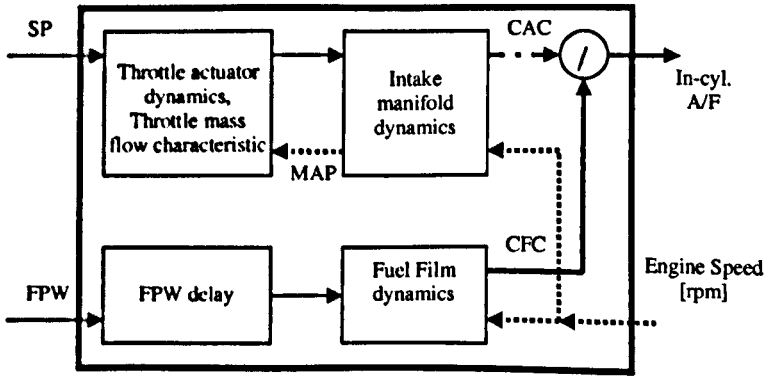

Figure 5-27: Engine block diagram

The important task for the control system is the maintenance of the air-fuel ratio at the stoichiometric level. The system structure suggests that if air-fuel ratio is to be maintained at the desired level at all times, the time delay in the fuel delivery path must be exactly predicted. In some situations even this stringent requirement does not guarantee achieving the stoichiometric target. For example, if fuel evaporation from the wall is already higher than the required cylinder fuel charge the desired air-fuel ratio will not be achieved. This however takes place only if the engine is cold and there is a significant change (decrease) in the power demand. In most of the operating conditions the knowledge of the future cylinder air charge and the fuel delivery model is sufficient to guarantee the required strength of the combustion mixture.

The future cylinder air charge is computed using the throttle position and intake manifold model. The difficulty in accurate prediction stems from the uncertainty in the future driver command. For a system where the accelerator pedal position is not directly linked to the position of the throttle plate, the delay introduced between the pedal and the throttle setpoint brings about a clear improvement in the performance of air-fuel ratio controller.

To demonstrate the benefits of introducing delays between the driver's command and the throttle position setpoint, a simulation study is undertaken.
The torque model used in the simulation is shown in Figure 5-25. It employs the map $M_{P,n} = function(CAC_{n-4}, N_{n-4})$ with the air-fuel ratio correction coefficient shown in Figure 5-24. The block diagram of the engine with the controller is presented in Figure 5-28. Two control scenarios are analyzed. In the first one the standard controller setup with the model-based predictive feedforward controller presented in section 5.1 is employed. The controller uses the current throttle position measurement. Within the prediction horizon, it is assumed that the throttle position remains fixed at the currently measured level.

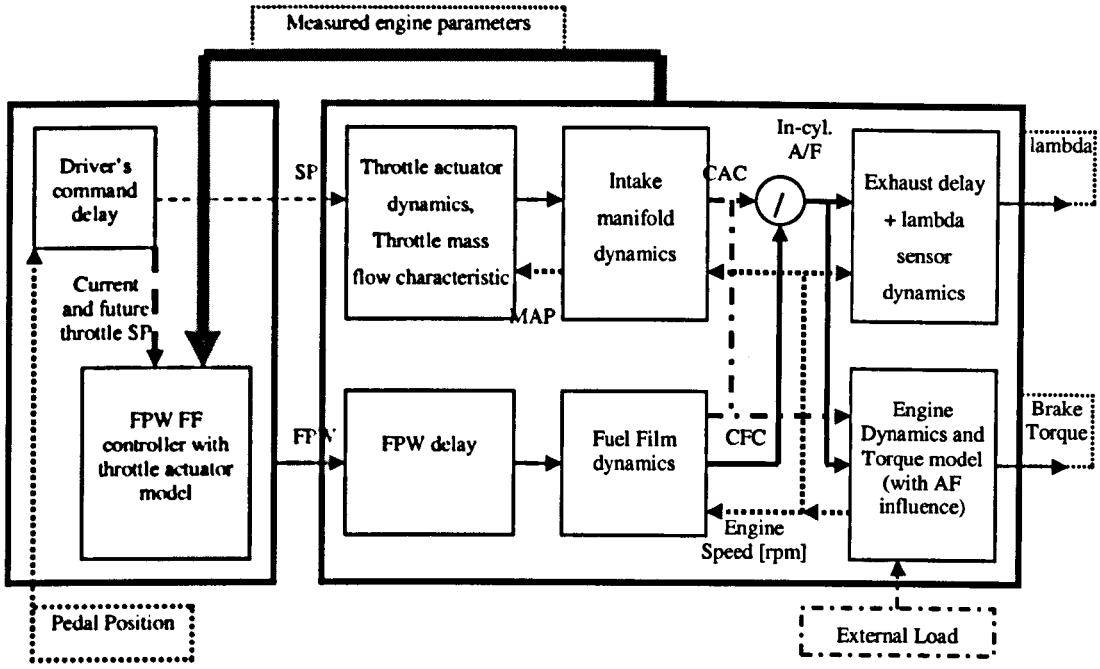

Figure 5-28: Engine model block diagram with the controller

The second scenario assumes that the driver's command is delayed. The FPW time delay used during simulation is 5 events (i.e. 5x90°) with an additional 1-event delay for control signal computation. This assumption will be used in the control system design in this section. This is a slight simplification of the injection strategy described in section 5.1.2. It may however be argued that the injectors driver circuit may work in a way that the injection duration is updated up to the certain fixed time delay. This results in a fixed FPW time delay measured by the engine events. This delay is assumed here to be equal to 5 events. As a consequence the driver's command will also be delayed by 5 events. In that way the throttle position prediction within a 6-event horizon is computed (from 5 future throttle input signals the discrete model for a sample-and-hold control provides 6 events ahead prediction). The delay introduced in the driver's command path would suggest that the torque response may deteriorate. This however is not necessarily the case. The simulation results in Figure 5-29 demonstrate the performance. The simulated driver's command (pedal position after applying the static torque mapping) is shown in Figure 5-30.

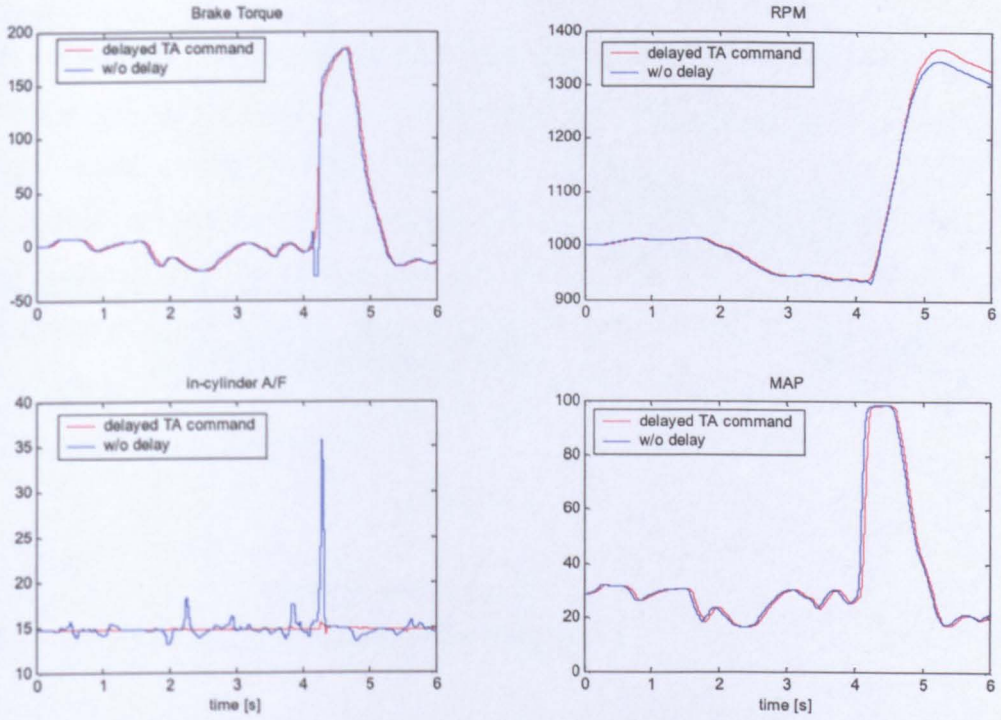

Figure 5-29: Simulation test results

During tip-ins, the in-cylinder air-fuel ratio excursion is so significant that proper combustion may not take place and the expected torque not produced for a short period of time (visible in upper left trace in Figure 5-29). This results in a visible hesitation in the brake torque. It is observed that the overall engine speed response during heavy transients shown in Figure 5-29 is improved if the commanded signal is delayed.

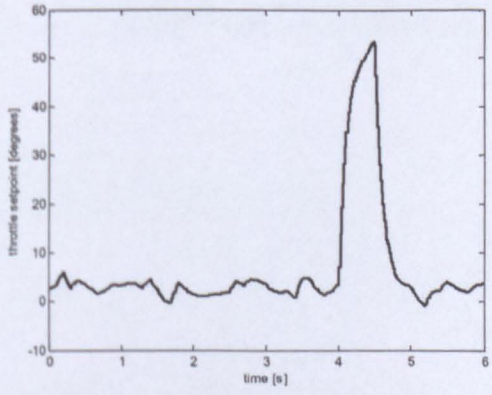

Figure 5-30:Throttle command used during simulation

Improvements in both emissions and drivability, due to the throttle command delay, are expected mainly during heavy transients. For moderate changes in demand, the torque response may slightly deteriorate. The proper trade-off must be made to satisfy both the emission standards and driver's torque demand. This implies that the time delay added in the throttle path may depend on current engine conditions, load and driver's torque demand. In the sequel, two simple MIMO torque and air-fuel ratio control methods will be analyzed. The torque model identified at stoichiometry will be used during the design.

5.4.2 Intake manifold response analysis

To analyze system properties knowledge of the system dynamic response is essential. The step responses of the engine sub-systems shown in Figure 5-27 for the intake manifold and fuel delivery models identified in Chapter 4 are plotted in Figure 5-31.

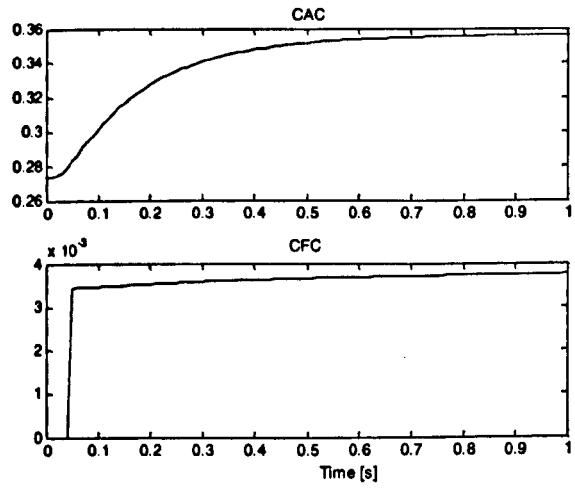

Figure 5-31: Cylinder air and fuel charge step responses (1500 RPM, initial MAP=50kPa)

The cylinder air charge response to the throttle position change is of fourth order-with non-linear dynamic behaviour (see the model development presented in

section 4.2.1). The cylinder fuel charge response is modelled by the first order system with the direct through term and the time delay. The direct through term present in the wall wetting dynamics suggests that the fuel delivery may be controlled with good precision. However, the time delay present in the fuelling path poses a difficult limitation to the control system and trade-offs must be considered.

The following control situation may be analyzed to give some insight into the control problem. Assume that the engine operates at constant speed with a constant load. The throttle position setpoint is derived from the accelerator pedal position in a conventional way. The dynamics of the throttle actuator and the intake manifold act as a filter for the abrupt change in pedal position and the cylinder air charge increases gradually as presented in Figure 5-31. The time delay present in the fuelling path makes it impossible to supply the fuel immediately and, as a result, the initial increase in the cylinder air charge will lead to an increase in the air-fuel ratio. It may also be noticed that the cylinder air charge reaches its steady state long after the throttle position changes.

Closer analysis of the problem may lead to the following conclusions. Due to the time delay in the fuel delivery path the torque response to the driver's command cannot be faster than the combined time delay in the fuel delivery path and the intake-to-power torque production delay. The faster response may only be achieved by moving away from stoichiometry. This may not necessarily mean better torque response as it was presented in the simulation in Figure 5-29. To achieve optimal torque and stoichiometric combustion mixture the response time of the intake manifold and fuel delivery path should be similar. In the case of the intake manifold and throttle actuator dynamics it will not be possible to achieve the ideal response of the cylinder air charge. The ideal response would have a shape of a pure time delay followed by the dynamic response with the direct through term. This type of response is observed in the fuel delivery path. However the dynamic compensation is possible and the transient cylinder air charge response may be improved.

In next sections the dynamic control of the throttle position will be analyzed. Two possible methods will be introduced. The first one presented in section 5.4.3 will only manipulate the length of the additional time delay and the second (in section 5.4.4) will not only introduce the time delay but also compensate the intake manifold dynamics.

5.4.3 Algorithm with additional time delay in throttle setpoint path

As a motivation for the work presented in this section, analyze the situation where the engine runs at steady speed and load conditions and the throttle setpoint is suddenly increased. Due to the time delay in the fuel delivery path the change in the fuel command will have an effect on cylinder fuel charge only after the time that equals the delay time. The simulation of this situation is presented in Figure 5-32. The throttle position setpoint was increased from 15.8° to 36.8°.

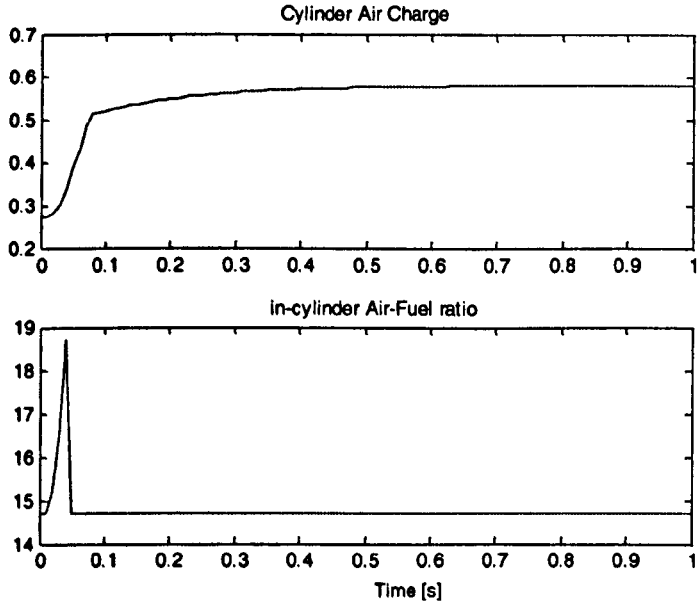

Figure 5-32: The cylinder air charge response to change of the throttle setpoint and the in-cylinder air-fuel ratio response (assuming the best possible control N=1500 rpm, initial MAP=50kPa)

There are two ways that the air-fuel ratio excursion may be avoided. The first one is to reduce the time delay in the fuel delivery path. This may entail injection on open valves and consequently increased emissions. The second solution of the problem requires dynamic shaping of the cylinder air charge response.

Analysis of the cylinder air charge response presented in Figure 5-33 will help in understanding the effects of the additional time delay in the throttle setpoint path. As before, the throttle position setpoint was increased from 15.8° to 36.8°. This change takes place at time t=0[s]. The time t=0 denotes the time when the control output is already computed and transferred to the injector timing subsystem (slave) and the throttle actuator. The fuel delivery time delay equals 5 engine events (90°/event). The additional 1-event delay for microcontroller computation is already removed from analysis. The effect of the additional delay introduced in the throttle command signal path may be observed in Figure 5-33, Figure 5-34 and Figure 5-35. The results presented in these figures differ since the simulation was carried out at different engine speeds.

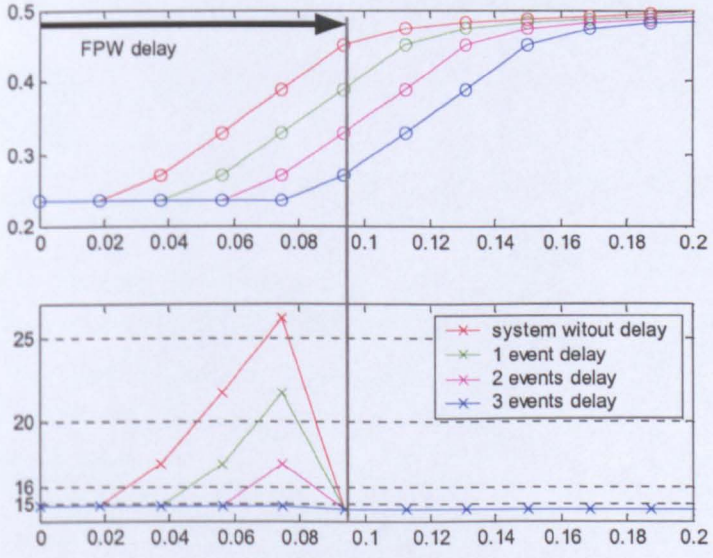

Figure 5-33: The cylinder air charge response to change of the throttle setpoint and the in-cylinder air-fuel ratio response (assuming the best possible control and additional delays in throttle setpoint path, N=800 rpm, initial MAP=50kPa, Throttle setpoint 15.8°-> 36.8°)

The conclusion is that for lower engine speeds longer time delay must be introduced to improve the air-fuel ratio response. To compute the optimal number of delay steps that are required in the throttle command path, the maximum acceptable air-fuel ratio excursion must be established. For the analysis that will follow it is assumed that the maximum permitted lean excursion is 16. From results presented in Figure 5-33 the delay of 3 engine events is needed to meet the requirements at *N=800 rpm*. From results presented in Figure 5-34, *2*-event delay is sufficient to meet demands for *N=1500 rpm*. At the higher speed *N=2200 rpm,* 1-event delay is already sufficient.

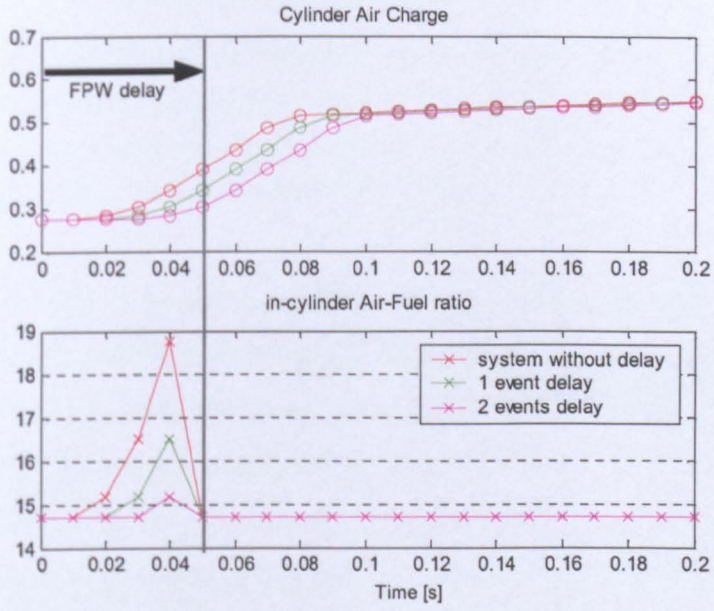

Figure 5-34: The cylinder air charge response to change of the throttle setpoint and the in-cylinder air-fuel ratio response (assuming the best possible control and additional delays in throttle setpoint path, N=1500rpm, initial MAP=50kPa, Throttle setpoint 15.8°->36.8°)

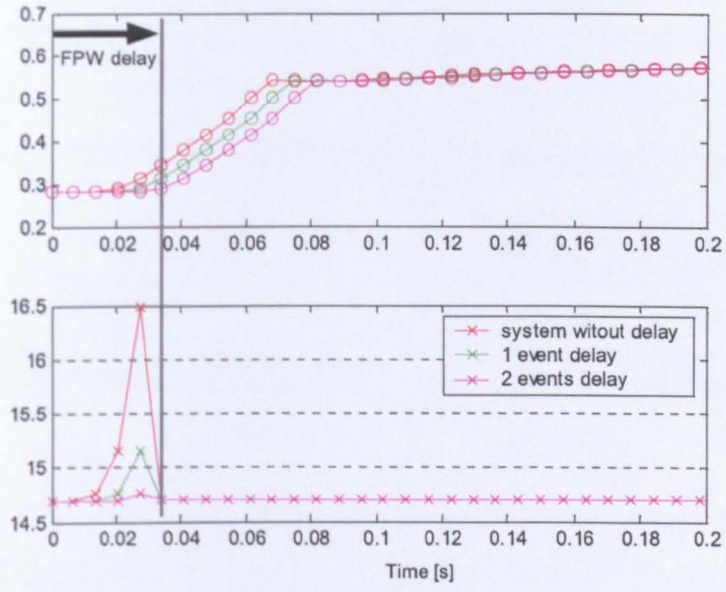

Figure 5-35: The cylinder air charge response to change of the throttle setpoint and the in-cylinder air-fuel ratio response (assuming the best possible control and additional delays in throttle setpoint path, N=2200 rpm, initial MAP=50kPa, Throttle setpoint 15.8°-> 36.8°)

The simulation results show the dependency of the minimal additional delay length on engine speed. However, engine speed is not the only factor that will alter the optimal duration of the delay. If the magnitude of the throttle setpoint change is less significant – e.g. throttle angle changes from 15.8° to 26.3° – the minimum delay that fulfils the maximum excursion constraint will also be different. In Figure 5-36 it is demonstrated that for N=1500 rpm the delay of 1 event is now sufficient. Recall that, at the same engine speed and for the throttle angle change from 15.8° to 36.8°, 2-event delay was required to keep the air-fuel ratio excursion below 16.

The presented analysis of the required time delay that needs to be introduced to keep AF excursions below 16 may be extended to the full operating range of the engine. The experiments conducted off-line for the full engine space resulted in series of lookup tables containing the optimal number of additional delays. The lookup tables for different levels of the throttle setpoint change are shown in Figure 5-37 and in Figure 5-38. The lookup tables were constructed assuming that a step change of the throttle setpoint occurred at steady state at a given engine speed and

pressure. In practice, only the air-fuel ratio control poses a problem for the control system analysis during transients and a step change covers the most challenging situation. In discrete time control strategy (with sample and hold), the throttle setpoint change is regarded as a step change from the previous sampling event.

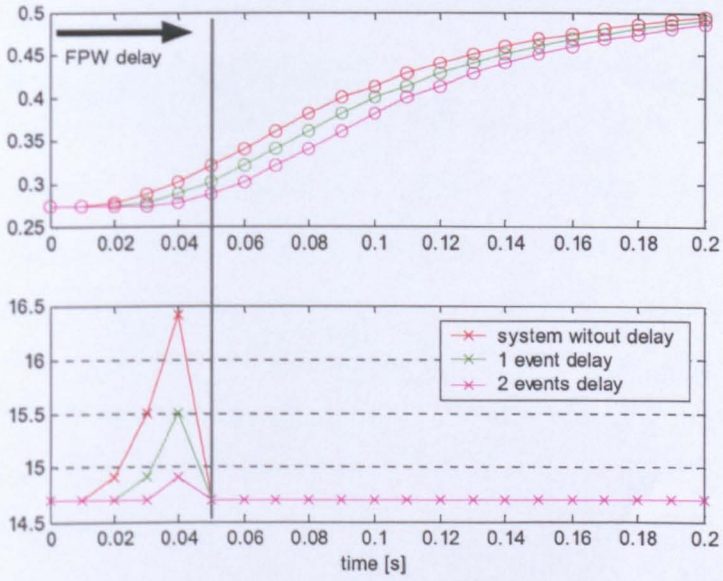

Figure 5-36: The cylinder air charge response to change of the throttle setpoint and the in-cylinder air-fuel ratio response (assuming the best possible control and additional delays in throttle setpoint path, N=1500 rpm, initial MAP=50kPa, Throttle setpoint 15.8°-> 26.3°)

Depending on the required accuracy, additional scheduling parameters may be considered. The throttle actuator is modelled by a 2nd-order discrete time model. The intake manifold has two states – air pressure and temperature. In steady state the intake manifold air temperature is determined by manifold pressure and engine speed. During heavy transients this should also be considered as a scheduling parameter.

For accurate scheduling of the additional time delay, a 6-dimensional lookup table is required. The following parameters have to be used: the throttle actuator states TPS_i - current throttle position, TPS_{i-1} - previous throttle position, SP_{i-1} -

previous throttle setpoint, SP_i - current throttle setpoint and the intake manifold parameters: P_{im} - intake manifold pressure and T_{im} - intake manifold temperature. Additionally, parameters like ambient temperature, pressure and coolant temperature (that contributes to the intake manifold wall temperature) may also be considered depending on the computational power of the target processor. For the study presented in this chapter the simplest method consists of a limited number of scheduling parameters which include changes in the throttle setpoint value (that may be regarded as SP_i - TPS_i), current engine speed N and intake manifold air pressure P_{im}. Any extension of the number of scheduling parameters is straightforward and requires only minor modification of the off-line algorithm that computes the required time delay based on the intake manifold model. The lookup tables for a negative change to the throttle position are shown in Figure 5-39 and in Figure 5-40. For this change to the throttle position the maximum rich excursion level is set at 13.4. It may be noticed that the rich excursions are less significant and in general less additional time delays are required.

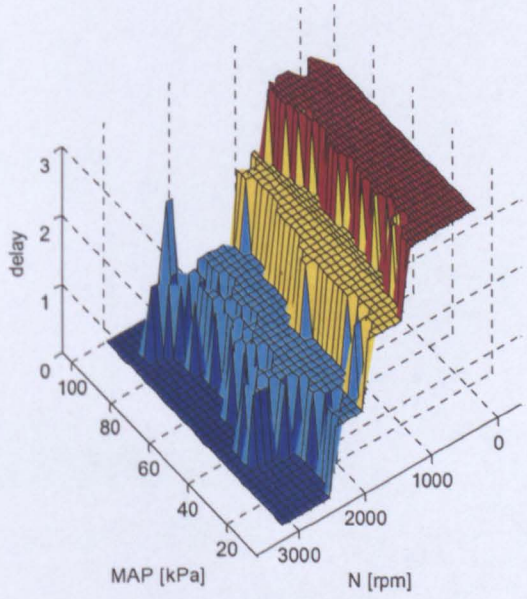

Figure 5-37: Time delay lookup table for throttle setpoint change of +1V (i.e. ~21°)

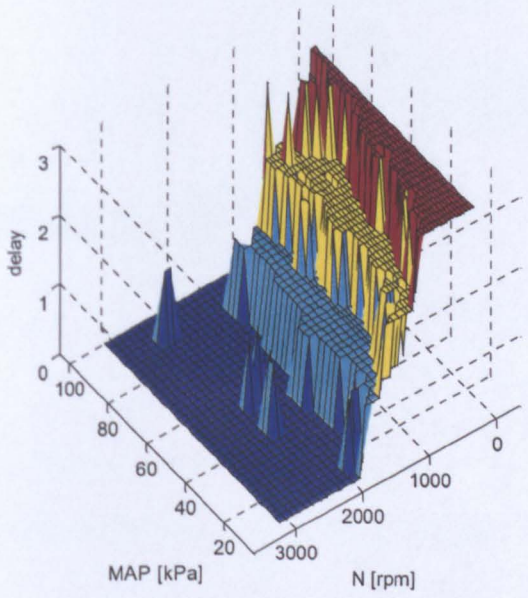

Figure 5-38: Time delay lookup table for throttle setpoint change of +0.5V (i.e. ~10.5°)

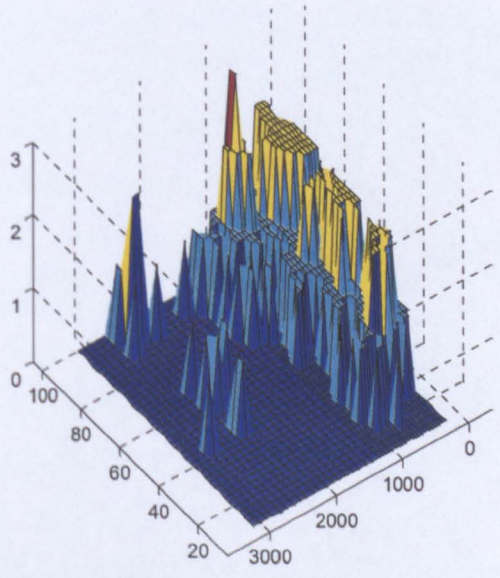

Figure 5-39: Time delay lookup table for throttle setpoint change of -1V (i.e. - 21°)

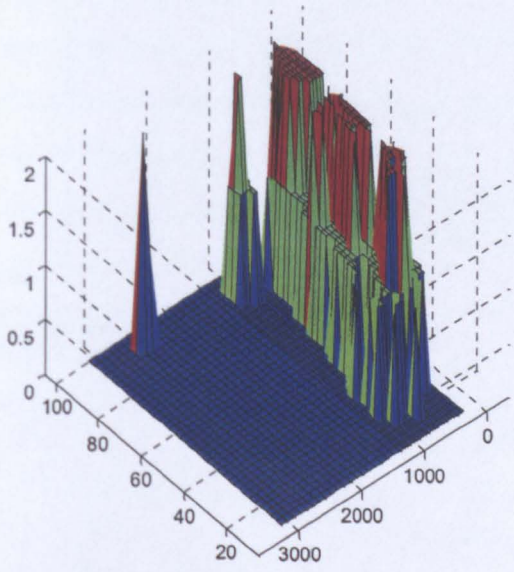

Figure 5-40: Time delay lookup table for throttle setpoint change of -0.5V (i.e. -10.5°)

5.4.4 Algorithm with the cylinder air charge response shaping

The introduction of the time delay improves the air-fuel ratio response. Given the maximum limit of the air-fuel ratio excursion, the desired air-fuel ratio control performance has to be achieved. From the performance point of view, the desired cylinder air charge (and the torque) should be achieved as fast as possible. The analysis of this problem is presented in Figure 5-41. Initially the cylinder air charge should remain unchanged, since the amount of the fuel that enters cylinders cannot be changed (i.e. not to be taken back as already injected). This however will not be the case after the time of the fuel delivery path delay. After the time of the delay, commanded an amount of fuel will enter the combustion chamber. At that moment (after the delay), it is desired that the cylinder air charge reaches the desired level as soon as possible. The ideal response of the system is shown as the blue line. This, of course, is not achievable due to the constraints in the throttle actuator and system non-linear dynamics.

Two types of constraints are responsible for this: the maximum throttle flow rate (for the wide open throttle) and the finite rise time of the throttle actuator. The

throttle is manipulated by its own controller that operates at a rate higher than the main (event-based) sampling rate. As a consequence, due to constraints in the electronic throttle control system, the maximum rate of change of the throttle plate position is limited. The throttle actuator linear model gives only an approximate description of the non-linear dynamics of the electronic throttle module. This is however sufficient for the control strategy synthesis.

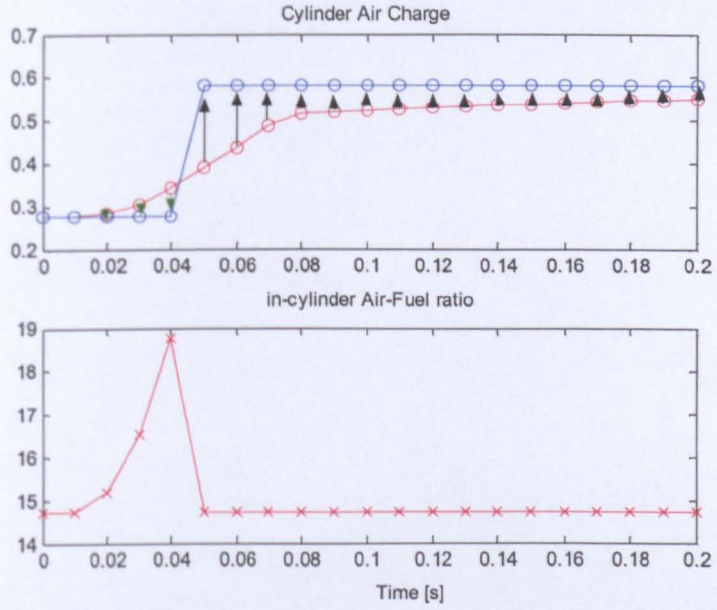

Figure 5-41: The cylinder air charge response to change of the throttle setpoint and the in-cylinder air-fuel ratio response (assuming the best possible control N=1500 rpm, initial MAP=50 kPa)

For the current engine speed the pedal position provides information about the torque demand (e.g. using the procedure described in section 5.2). For the desired torque the cylinder air charge may be computed from the torque model $M_{P,n} = function(CAC_{n-4}, N_{n-4})$. Additionally, the amount of fuel entering the cylinder over the time of the fuel delivery delay (i.e. the amount of fuel already injected) is known. It is desired that the air-fuel ratio is kept at stoichiometry at all times. Based on that requirement the desired cylinder air charge trajectory can be computed. This is indicated by the green dotted arrows in Figure 5-42. The cylinder air charge that results in the desired torque should ideally be achieved at event k=6. The actual CAC trajectory differs from the ideal due to the system dynamics and

constraints. The optimal trajectory must satisfy the maximum AF excursion constraint. This means that the trajectory for k=1…5 should not diverge from the desired trajectory by more than the maximum allowed level. At the event k=6, the actual CAC trajectory should be as close as possible to the desired value determined by the torque demand. Based on the actual reachable cylinder air charge value at k=6 the cylinder fuel charge is calculated (the red dotted arrow). The fuel pulse width command is a function of the desired fuel charge that is given by the fuel-film dynamics model. This FPW command computed here is the output of the controller. Also, the current throttle setpoint is obtained through the optimization of the CAC trajectory and is used for control. The throttle position setpoint is another output of the controller.

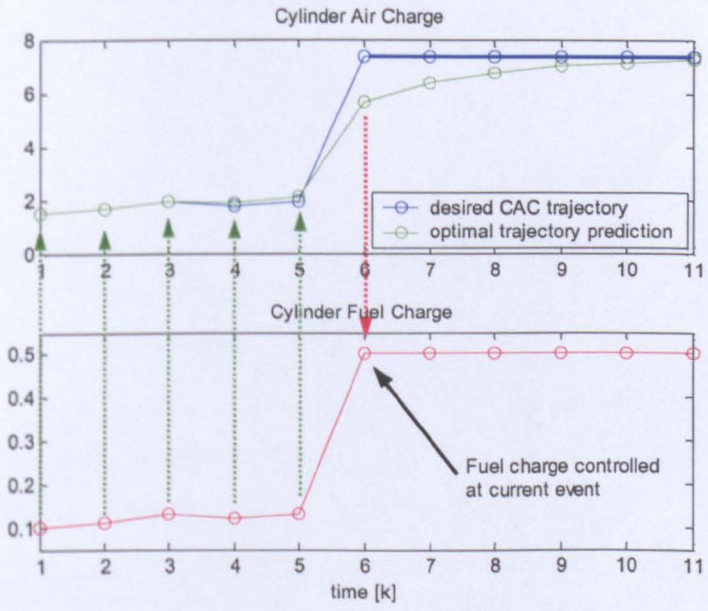

Figure 5-42: Optimal throttle and fuel control strategy

5.4.4.1 Feedback compensation solution

The tracking of the given reference trajectory shown in Figure 5-42 may be achieved using dynamic compensation or PID control theory.

The required response of the system (e.g. like in Figure 5-42) is achieved in two steps. The faster dynamic response of the system is achieved by the compensation of the cylinder air charge response of the intake manifold and throttle systems. The results of such compensation are shown in Figure 5-43.

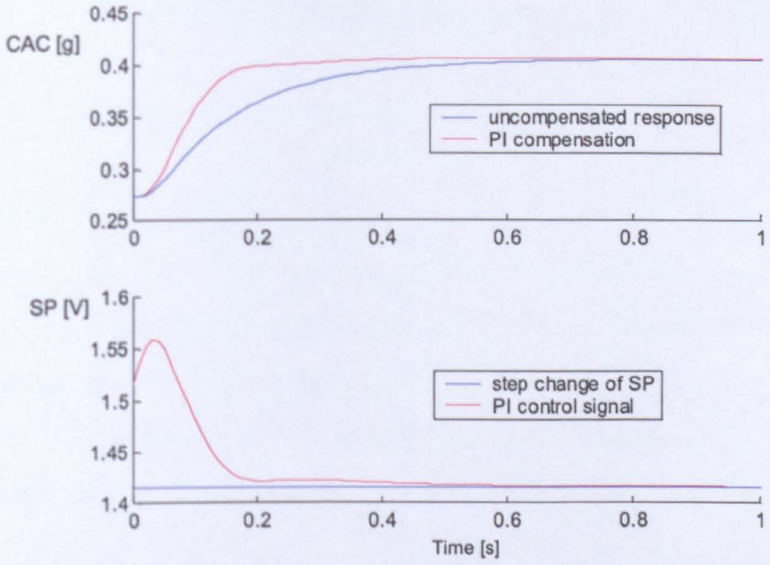

Figure 5-43: PI compensation of the cylinder air charge (CAC) response (SP_{0}=1.22[V])

The result demonstrates a well-known property of feedback control which makes it possible to achieve a faster dynamic response. The feedback action is capable of increasing the bandwidth and speed of transient response. This however comes at the expense of increased actuation effort. Note, that the feedback from the CAC signal is used. The CAC is not measured, but may be determined using the model as a virtual sensor for the CAC estimation. For the non-linear system a PID controller with gain scheduling could be used. Also other dynamic control methods may be used here. The compensated (faster) cylinder air charge (CAC) response will definitely result in higher excursions in the air-fuel ratio if CAC_{SP} (proportional to driver's torque demand) was used directly as the setpoint SP_0 (see Figure 5-44).

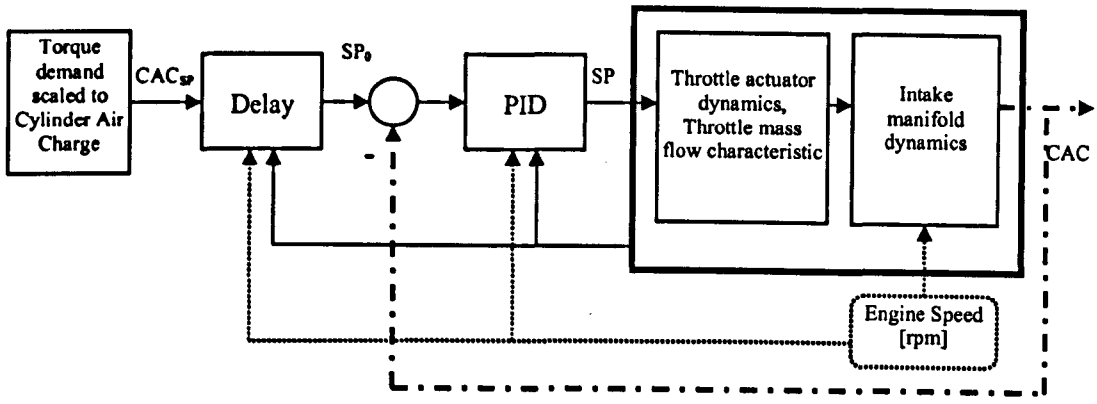

Figure 5-44: PID intake manifold response compensation

This problem may be tackled in the same way as it was done for the uncompensated system in section 5.4.3. The appropriate time delay must be added in the throttle command path. Note that in section 5.4.3, the torque demand was expressed by the corresponding throttle angle which provides the required CAC under steady state conditions. The setpoint SP_0 is already the CAC setpoint and the throttle flow characteristic and the intake manifold dynamics are only needed during PID tuning.

As a result of the additional time delay, the setpoint CAC_{SP} command is delayed by n, where n is the number of event delays introduced in the throttle command path (see Figure 5-44). The additional time delay depends on the operating point (state) which also determines the PID controller gains. The desired CAC setpoint values over the horizon n are indicated in Figure 5-45 by the green dotted arrows. In this example additional delay in the CAC_{SP} command path $n=3$. The change of the setpoint occurs at event $k=n+1$ and is indicated by the magenta arrows. The closed-loop system response with a PID controller is shown by the green line in Figure 5-45. The green line is intended to be as close as possible to the ideal CAC trajectory indicated by the blue line in Figure 5-45. For this reason it is desirable to change the setpoint SP_0 earlier (red line, change at k=4).

The MIMO control strategy manipulates both the throttle position and the injection time. In this case, the required throttle setpoint command is delayed and effectively proportional to the fuel already injected, e.g. at the event $k=1$ in Figure 5-45. To compute the fuel command, the model with the PID controller must be

simulated over the horizon that equals the time delay in the fuel delivery path. The setpoint signal SP_0 is determined by the additional delay introduced in the throttle command path (Figure 5-44) and the CAC_{SP} level which is proportional to the desired torque. The fuel pulse width command results from the achievable cylinder air charge at time *k=6*. Due to hard constraints on the throttle actuator, the PID controller must be equipped with anti-windup protection.

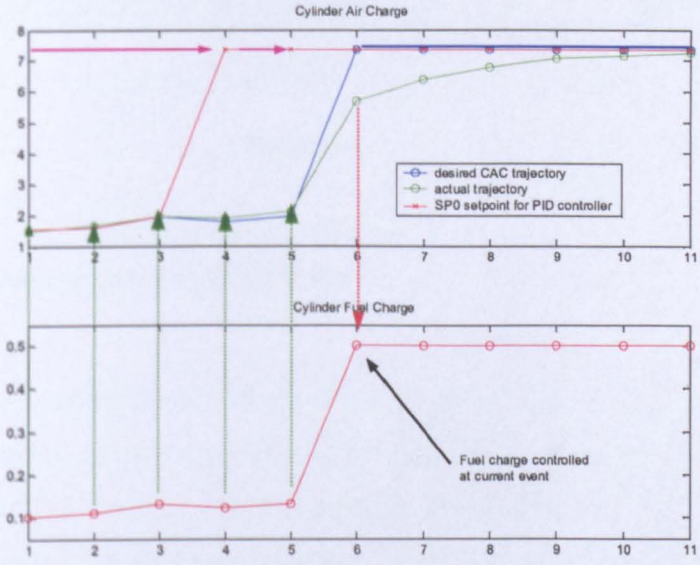

Figure 5-45: Optimal throttle and Fuel Pulse Width control strategy for PID controller

The control algorithm consists of the following steps:

- The current throttle setpoint SP command is proportional to the delayed CAC setpoint (and effectively is based on the amount of fuel that is predicted to enter the cylinder in the event *k=1*) and the current state of the throttle actuator (position). It is the output of the PID controller
- The CAC is assumed to follow the desired trajectory with the PID controller manipulating the setpoint SP value supplied with the SP_0 signal. The CAC trajectory may be computed using the model of the system and the controller over the prediction horizon (k=6 in Figure 5-45)

- The CAC after the fuel delivery delay (at 6-th event in Figure 5-45) is used for the fuel control. The wall-wetting dynamics and the injector model are taken into consideration and the fuel pulse width command is issued.

The fuel pulse width (FPW) command is computed based on the required cylinder fuel charge which in turn is computed from the cylinder air charge (assuming stoichiometric air-fuel ratio). The wall fuel mass estimate, wall fuel dynamic model and fuel injector model are also required. The fuel feedforward control procedure was presented in section 5.1.

5.4.4.2 Predictive Control solution

The desired cylinder air charge response presented in Figure 5-42 may be achieved in a number of ways. A very natural method is to use the Model Predictive Control strategy. The desired CAC trajectory in Figure 5-42 may be used as a reference signal for the predictive controller. Within the predictive control algorithm the future control trajectory prediction is the solution to an optimization problem. For the assumed 6-event delay between the control signal and the cylinder charging, the CAC prediction at 6-th event is used for the computation of the fuel command.

Given the system model (developed in Chapter 4):

$$TPS_{n+1} = -p_1(N_n)\cdot TPS_n - p_2(N_n)\cdot TPS_{n-1} + p_3(N_n)\cdot SP_n + p_4(N_n)\cdot SP_{n-1}$$

$$\begin{bmatrix} P_{im,n+1} \\ T_{im,n+1} \end{bmatrix} = f\left(P_a, T_a, P_{im,n}, T_{im,n}, TPS_n\right) \tag{5.13}$$

$$CAC_n = h\left(P_{im,n}, T_{im,n}, N_n\right)$$

The following performance index is minimized:

$$J = \sum_{k=0}^{i+T_{PRED}} \left\| CAC_{k+1} - CAC_{REF,k+1} \right\| + \left\| SP_k - SP_{k-1} \right\| \tag{5.14}$$

where

$\| \ \|$ denotes the norm – e.g. absolute value, quadratic.

T_{PRED} - prediction horizon – must be longer than the fuel path time delay

The model predictive control algorithm performs the minimization of the cost function (5.14) subject the system dynamics (5.13). Additionally, hard constraints on input and output may also be considered.

$$\begin{aligned} &TPS_{\min} \le SP_k \le TPS_{\max} \\ &TPS_{\min} \le TPS_k \le TPS_{\max} \\ &CAC_{REF,k}(1-\varphi) \le CAC_k \le CAC_{\max}(1+\varphi) \quad for \quad k = i+1...i+T_{FPW} \end{aligned} \tag{5.15}$$

where

φ - the coefficient that determines the maximum excursion of the air-fuel ratio. For assumed maximum AF=16, minimum A/F=13.4: $\varphi = 0.088$.

T_{FPW} - determines the number of events when the cylinder air charge is desired to track the fuel already injected – in Figure 5-42 $T_{FPW} = 5$.

The predictive control algorithm (5.13), (5.14), (5.15) requires an on-line numerical solution of the optimization problem. The convergence properties of the optimization algorithm have to be addressed. In general, for non-linear problems, there is no guarantee that the global optimum is achieved. The numerical optimization algorithm may find a local minimum which may not provide satisfactory performance.

Due to the above implementation issues, some simplifications to the predictive control algorithm must be made. The first simplification considered

attempts to remove hard constraints from the optimization algorithm and substitute them with the weights in the cost function. Also, a quadratic norm is used in the cost function.

$$J_i = \sum_{k=i}^{i+T_{PRED}} \left\{ \left(CAC_{k+1} - CAC_{REF,k+1} \right)^2 q_k + \left(SP_k - SP_{k-1} \right)^2 r_{1,k} + SP_k^2 r_2 \left(TPS_k \right) \right\} \qquad (5.16)$$

where

q_k - weight on CAC tracking error (time-varying)

$r_{1,k}$ - weight on control change

$r_2(TPS_k)$ - state-dependent weight on control action that is effective only when the constraint is violated

The simplified predictive control solution presented next uses the Non-Linear Generalized Predictive Control method [15]. This predictive control method requires model (5.15) to be re-arranged into the state-dependent coefficient form.

The discrete electronic throttle state-space model is given by the equation (5.17). A non-minimal representation of the 2-nd order system with 3 states was used to improve the numerical properties for systems discretized with a variable sampling rate [89].

$$\begin{aligned} x_{ET,n+1} &= A_{ET,n} x_{ET,n} + B_{ET,n} SP_n \\ TPS_n &= C_{ET} x_{ET,n} \end{aligned} \qquad (5.17)$$

Where

$$x_{ET,n} = \begin{bmatrix} SP_{n-1} \\ TPS_n \\ TPS_{n-1} \end{bmatrix}; A_{ET,n} = \begin{bmatrix} 0 & 0 & 0 \\ p_4(N_n) & -p_1(N_n) & -p_2(N_n) \\ 0 & 1 & 0 \end{bmatrix}; B_{ET,n} = \begin{bmatrix} 1 \\ p_3(N_n) \\ 0 \end{bmatrix};$$

$$C_{ET} = [0 \quad 1 \quad 0];$$

It should be noted that in the model (5.17) matrices $A_{ET,n}$, $B_{ET,n}$ depend on engine speed N_n and consequently are state-dependent. The throttle angle is a function of the TPS_n signal. The following relationship is established.

$$\alpha_n = (TPS_n - TPS_{\min})\beta \tag{5.18}$$

where

$$\beta = \frac{\alpha_{\max}}{(TPS_n - TPS_{\min})}$$

$TPS_{\min}$ - minimum throttle position sensor voltage associated with the 0 angle

$TPS_{\max}$ - maximum throttle position sensor voltage associated with the angle $\alpha_{\max}$

For the Corvette the following parameters were used: $TPS_{\min} = 0.46$, $TPS_{\max} = 4.76$, $\alpha_{\max} = \pi/2$. The throttle flow is modelled by one-dimensional isentropic compressible flow equation for the flow across the orifice [90]. The intake manifold model is based on the energy and mass balance and is derived from the ideal gas law. The intake manifold pressure sensor is very fast and its dynamic response may be neglected. The temperature sensor dynamic response is relatively slow and is modelled by the first order lag. The model is given in section 4.3.

The port flow rate is modelled as a function of the intake gas density, engine displacement and the volumetric efficiency and is given by the equation (5.19). The cylinder air charge CAC_n is derived from the port flow and the current sampling rate. The sampling rate for 90 degrees event timing equals $15/N_n$. The cylinder air charge is given by the following expression:

$$CAC_n = \frac{V_{cyl}}{R_{air} T_{im,n}} \eta_n P_{im,n} \tag{5.19}$$

The non-linear intake manifold model is parameterized in the state dependent coefficient form. The parameterized discrete form of the intake manifold model is given by the following equation:

$$\begin{aligned} x_{IM,n+1} &= A_{IM,n}x_{IM,n} + B_{IM,n}\cdot\alpha_n \\ CAC_n &= C_{IM}x_{IM,n} \end{aligned} \tag{5.20}$$

The state dependent matrices of the model (5.20) are of the following form:

$$x_{IM,n} = \begin{bmatrix} P_{im,n} \\ T_{im,n} \end{bmatrix}, \quad A_{IM,n} = \begin{bmatrix} \left(1-\frac{\kappa V_{cyl}}{V_{im}}\eta_n\right) & \frac{\kappa-1}{T_{im,n}V_{im}}T_{s,n}\dot{Q}_{ext,n} \\ 0 & a_{IM,22} \end{bmatrix},$$

$$a_{IM,22} = \left(1-\frac{\kappa V_{cyl}}{V_{im}}\eta_n\left(1-\frac{1}{\kappa}\right)\right)+\frac{(\kappa-1)T_{s,n}\dot{Q}_{ext,n}}{V_{im}P_{im,n}}$$

$$B_{IM,n} = \begin{bmatrix} \frac{\kappa R_{air}T_{a,n}T_{s,n}\dot{m}_{at,n}}{V_{im}TPS_n} \\ \left(T_{im,n}\frac{\kappa R_{air}T_{a,n}}{V_{im}P_{im,n}} - T_{im,n}{}^2\frac{R_{air}}{V_{im}P_{im,n}}\right)\frac{T_{s,n}\dot{m}_{at,n}}{TPS_n} \end{bmatrix}; C_{IM} = \begin{bmatrix} \frac{V_{cyl}}{R_{air}T_{im,n}}\eta_n & 0 \end{bmatrix}$$

$$\dot{Q}_{ext} = h_1\left(T_{coolant} - T_{im}\right) + h_2\left(T_a - T_{im}\right)$$

The final augmented model is given by the following set of equations:

$$\begin{aligned} \chi_{n+1} &= A_n\chi_n + B_n u_n \\ y_n &= C_n\chi_n \end{aligned} \tag{5.21}$$

where

$$\chi_n = \begin{bmatrix} x_{ET,n} \\ x_{IM,n} \end{bmatrix}; y_n = CAC_n; A_n = \begin{bmatrix} A_{ET,n} & 0 \\ B_{IM,n}\cdot C_{ET} & A_{IM,n} \end{bmatrix};$$

$$B_n = \begin{bmatrix} B_{ET,n} \\ 0 \end{bmatrix}; C_n = \begin{bmatrix} 0 & C_{IM} \end{bmatrix}; u_n = SP_n.$$

An additional state associated with the integral action is introduced next. The modification of the state-space model (5.21) is carried out as follows:

$$\begin{aligned} x_{n+1} &= \tilde{A}_n x_n + \tilde{B}_n u_n \\ y_n &= \tilde{C}_n x_n \end{aligned} \tag{5.22}$$

where

$$x_n = \begin{bmatrix} SP_{n-1} \\ x_{ET,n} \\ x_{IM,n} \end{bmatrix}; \Delta u_n = \Delta SP_n; \Delta SP_n = SP_n - SP_{n-1},$$

$$\tilde{A}_n = \begin{bmatrix} 1 & 0 \\ B_n & A_n \end{bmatrix}; \tilde{B}_n = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \tilde{C}_n = [0 \quad C_n].$$

Note that the above model introduces a one-step delay in the input. This is due to the discrete-time implementation of the control strategy. A one-step delay reflects time required for computations preformed by the microcontroller.

5.4.4.3 NLGPC solution -results

The first set of results show the comparison of engine response to an abrupt change in the torque demand at time *t=0.1[s]*. For a given torque demand the corresponding cylinder air charge (CAC) is computed. This CAC setpoint is used by the predictive control algorithm. The CAC response for the conventional control algorithm (see section 5.1) and the conventional algorithm that uses a throttle actuator model are presented. For the NLGPC algorithm the following parameters in the cost function (5.16) are used:

Predictive control 1:

$T_{PRED} = 10$

$q_k = 100$ *for* $k = 0...4$, $q_k = 1$ *for* $k = 5...10$, $r_{1,k} = 0.1$ *for* $k = 0...10$

$r_2(TPS_k) = 0$ *for* $k = 0...10$

Predictive control 2:

$T_{PRED} = 8$

$q_k = 100$ *for* $k = 0...4$, $q_k = 1$ *for* $k = 5...8$

$r_{1,k} = 0.1$ *for* $k = 0...8$, $r_2(TPS_k) = 0$ *for* $k = 0...8$

Note that for simplicity the constraint violation penalty term was set to zero. The hard constraints in the throttle actuator were dealt with by using hard constraints on the integral action state in model (5.22) outside of the control algorithm. The penalty term q_k for the initial part of the horizon is increased to minimize the air-fuel ratio excursions – the initial desired cylinder air charge is already determined by the amount of fuel that was injected in past events.

The cylinder air charge response is presented in Figure 5-46.

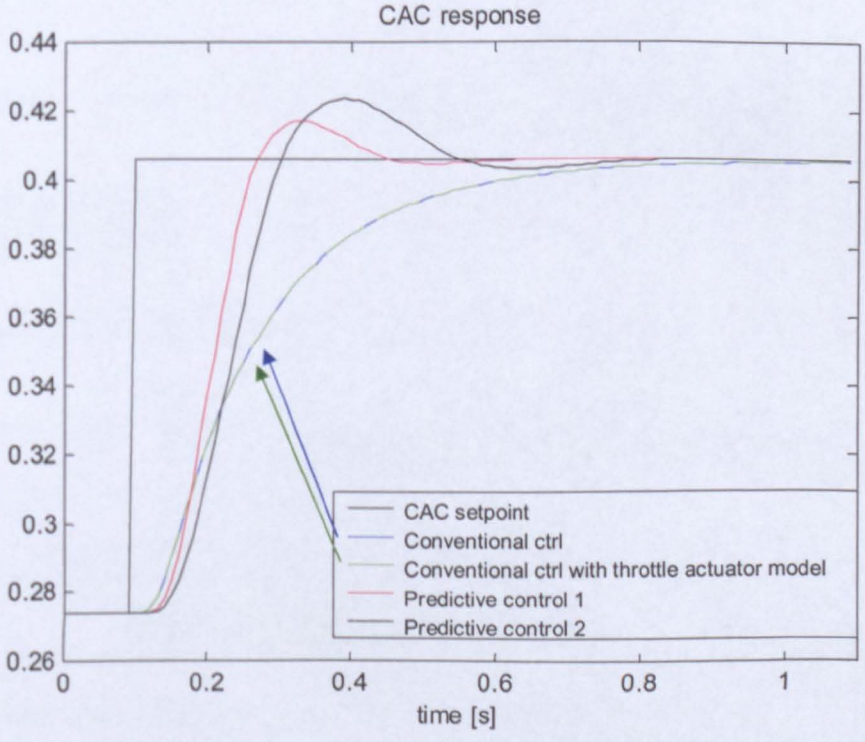

Figure 5-46: CAC response to step change in torque demand

Proportional to the cylinder air charge, torque production response is presented in Figure 5-48. It may be noticed that the tracking is improved with the predictive control method. This results in better drivability. At the same time good regulation of the air-fuel ratio is achieved. The lambda is shown in

Figure *5-47*. It should be noticed that the performance of the conventional control strategy is largely improved when the throttle actuator model is used. This improves the accuracy of the future throttle position information. Results may be related to the conventional control strategy (see section 5.1) which did not use throttle position setpoint derived from the pedal position and was only using the measurement of the current throttle position. The indicated throttle position was assumed to remain fixed over the algorithm's internal prediction horizon.

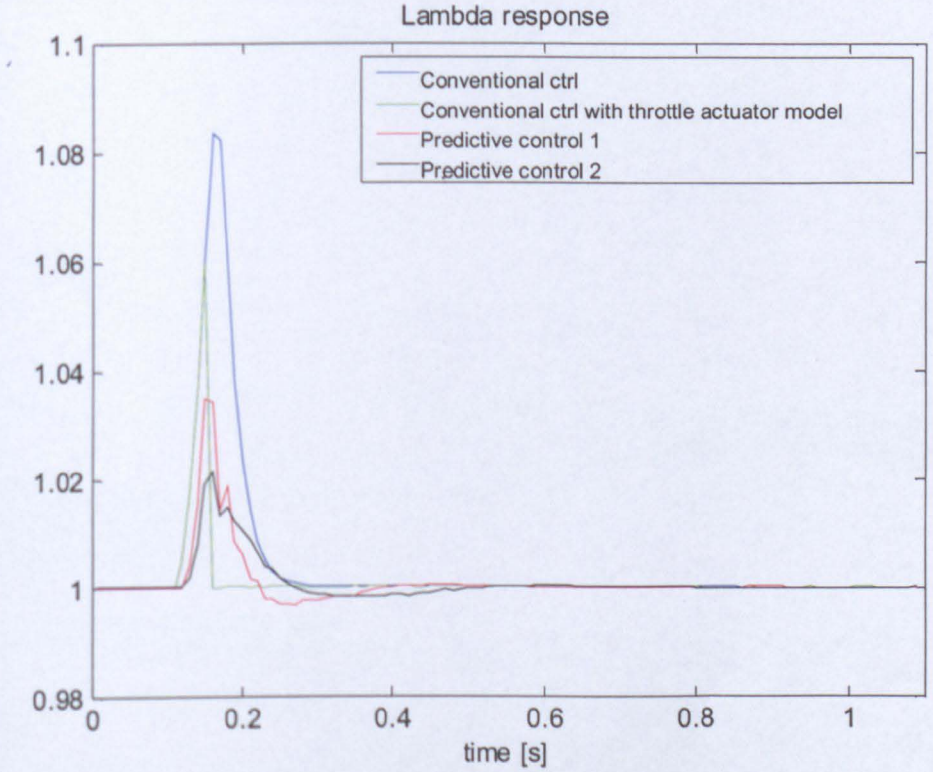

Figure 5-47: In-cylinder Lambda response to step change in torque demand

The choice of the prediction horizon has an impact on the torque response and the precision of lambda regulation. The weighting coefficients serve as a 'tuning knob' which is used to achieve a desired trade-off between the air fuel ratio regulation accuracy and the torque tracking performance. This trade-off may be observed in

Figure 5-47 and in Figure 5-48. The "predictive control 1" algorithm gives higher lambda excursion but at the same time torque response is better when compared to "predictive control 2". Faster torque response is achieved through a more aggressive throttle actuation.

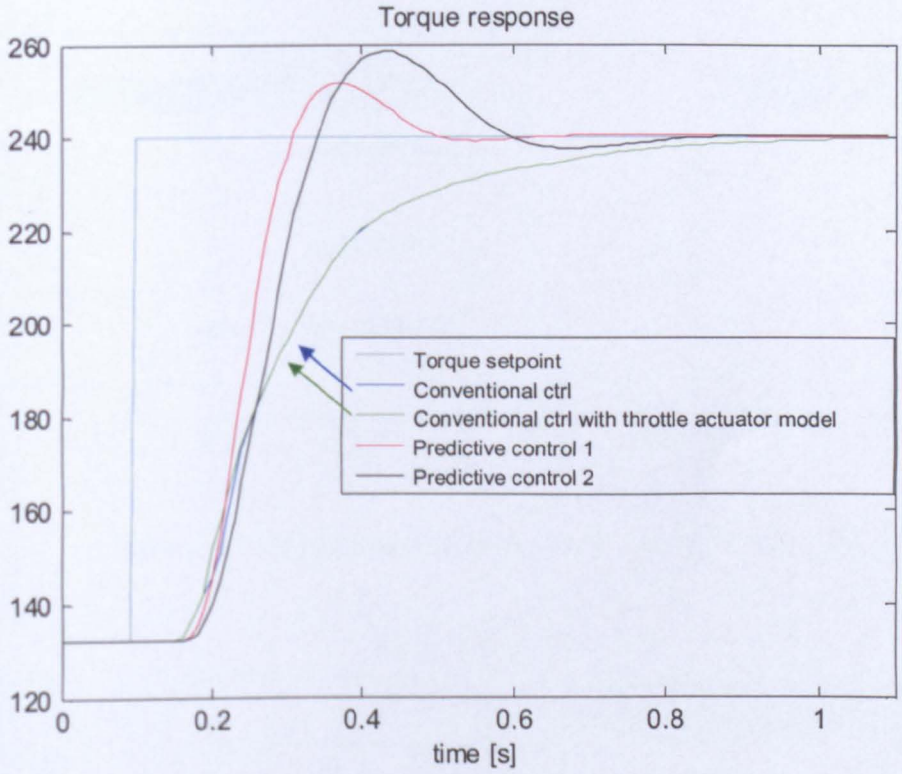

Figure 5-48: Torque response to step change in torque demand

The throttle setpoint signal is presented in Figure 5-50 and the throttle response is shown in Figure 5-49. The faster throttle actuator response guarantees better accuracy of the cylinder air charge control and improved torque response. The throttle dynamic response is the limiting factor in the control system performance and for faster torque tracking the slew rate of the throttle actuator has to be increased.

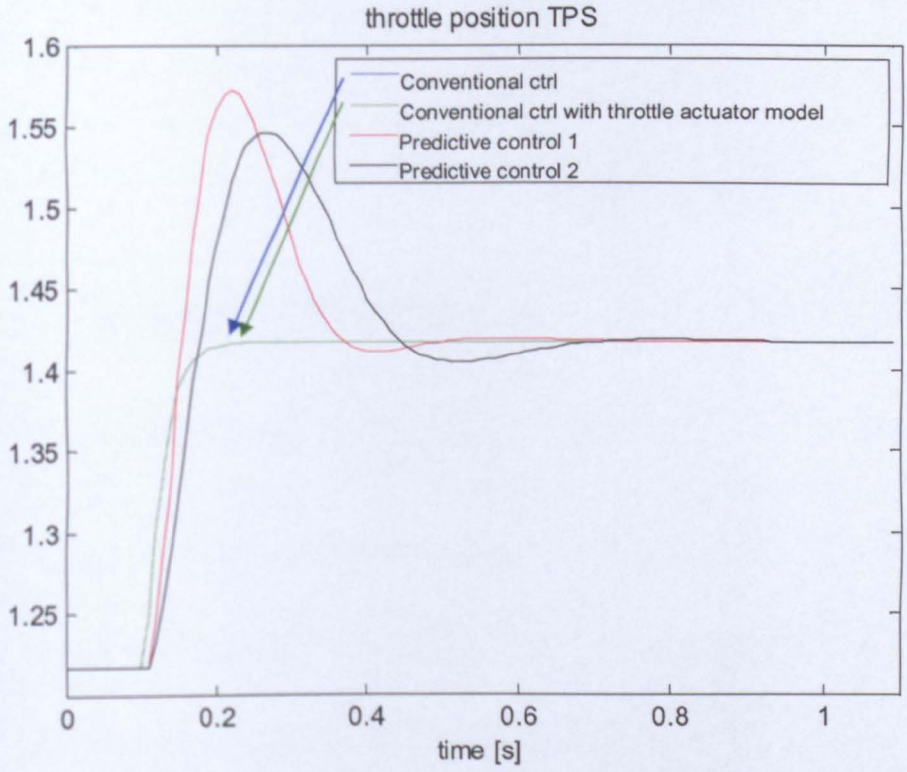

Figure 5-49: Throttle position response to step change in torque demand

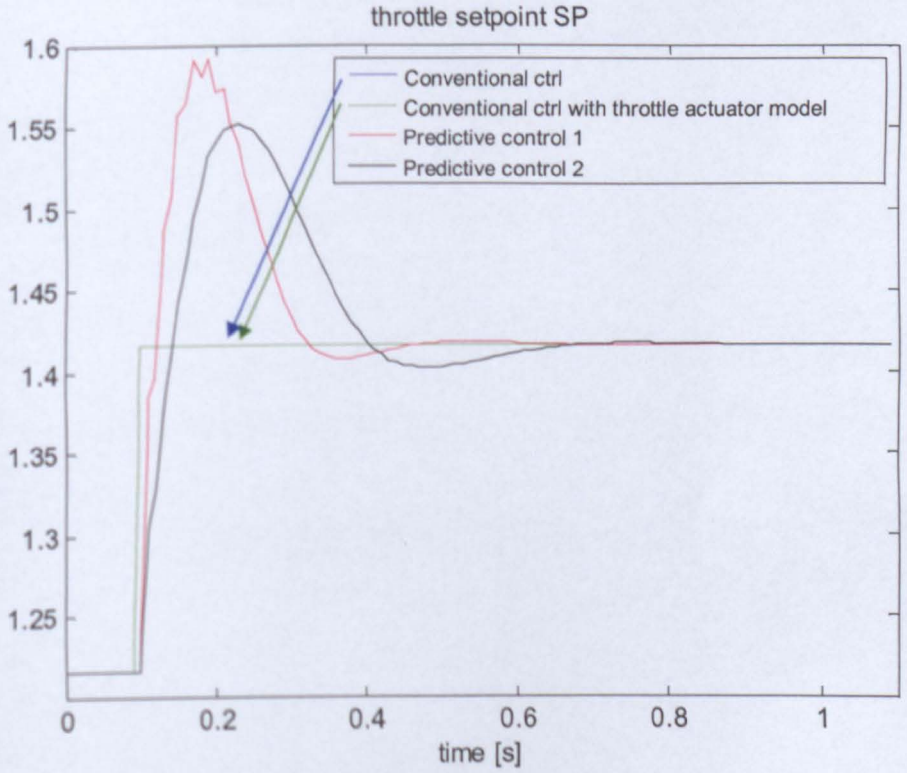

Figure 5-50: Torque setpoint command as a result of step change in torque demand

The conventional control strategies and the predictive controller with two control horizons were tested. The throttle setpoint was extracted from the US06 driving cycles. The engine speed, a parameter in the model, was taken from the same dataset. The benchmark of four control algorithms is presented in Table 5-2.

	Conventional control	Conventional control with throttle actuator model	Predictive control 1 (with q=100, T=10)	Predictive control 2 (with q=100, T=8)
Integrated squared AF error	0.2938	0.0203	0.0290	0.0207
Integrated absolute AF error	28.393	9.059	10.703	8.616
Integrated squared torque tracking error	3.823e+6	3.772e+6	2.114e+6	2.874e+6
Integrated absolute torque tracking error	10.516e+4	10.462e+4	7.672e+4	8.865e+4

Table 5-2: Comparison of control algorithms for US06 driving cycle

In rows of the table the four benchmark indexes are gathered. The columns are associated with four algorithms being subject of comparison. The air-fuel ratio regulation performance of the conventional algorithm with the throttle actuator (presented in section 5.3), offers a significant improvement over the basic feedforward control algorithm (used for control in section 5.1). The air-fuel ratio regulation performance offered by "predictive control 2" algorithm is better based on the integrated absolute value index (by 5%) and worse by 2 percent when compared to the conventional algorithm with the throttle actuator model. This, of course, is subject to tuning. Retaining the lambda performance, the torque response was improved. The tracking performance improvement for the "predictive control 1" is higher (when compared with "predictive control 2") but the lambda regulation performance is worse. This relates the predictive control algorithms to the conventional algorithm with the throttle prediction. However, if the results of the predictive controls are compared with those of the conventional control (without the throttle actuator model), a significant improvement in both lambda and torque response is observed.

To demonstrate the performance of predictive control algorithms, a part of US06 driving cycle simulation is plotted. The responses of cylinder air charge and torque are shown in Figure 5-51 and Figure 5-53, respectively. The speed of response is visibly improved by the predictive control method. It should be emphasized that the minimum time delay in the torque response is on average 4 events and the effect of other delays is added on top of that. The lambda (Figure 5-52) response is similar for the predictive algorithm and conventional algorithm with the throttle model. The low performance in Figure 5-52 is attributed to the conventional control algorithms. A close examination of Figure 5-54 and Figure 5-55 reveal that the improvement in the torque response may be attributed to the aggressive throttle control action using the predictive control techniques.

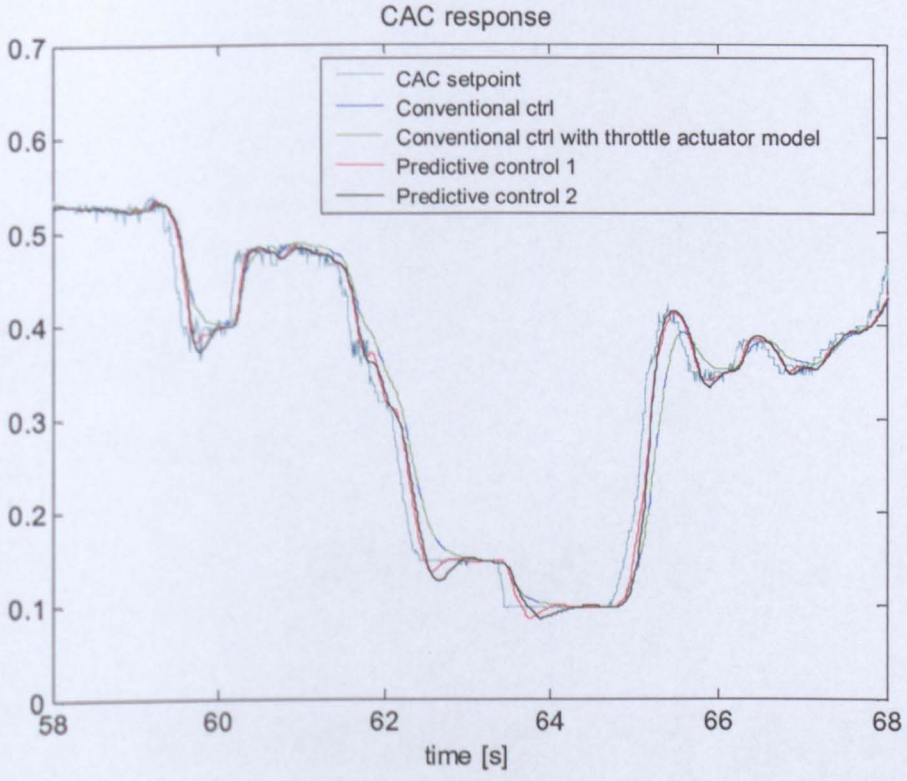

Figure 5-51: CAC US06 response

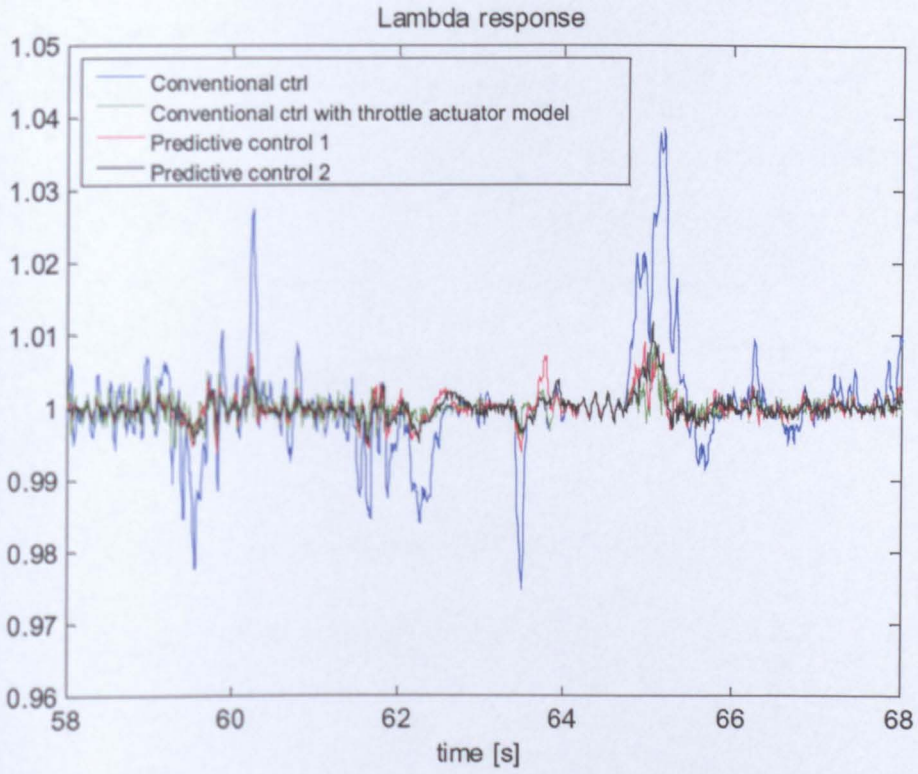

Figure 5-52: Lambda US06 response

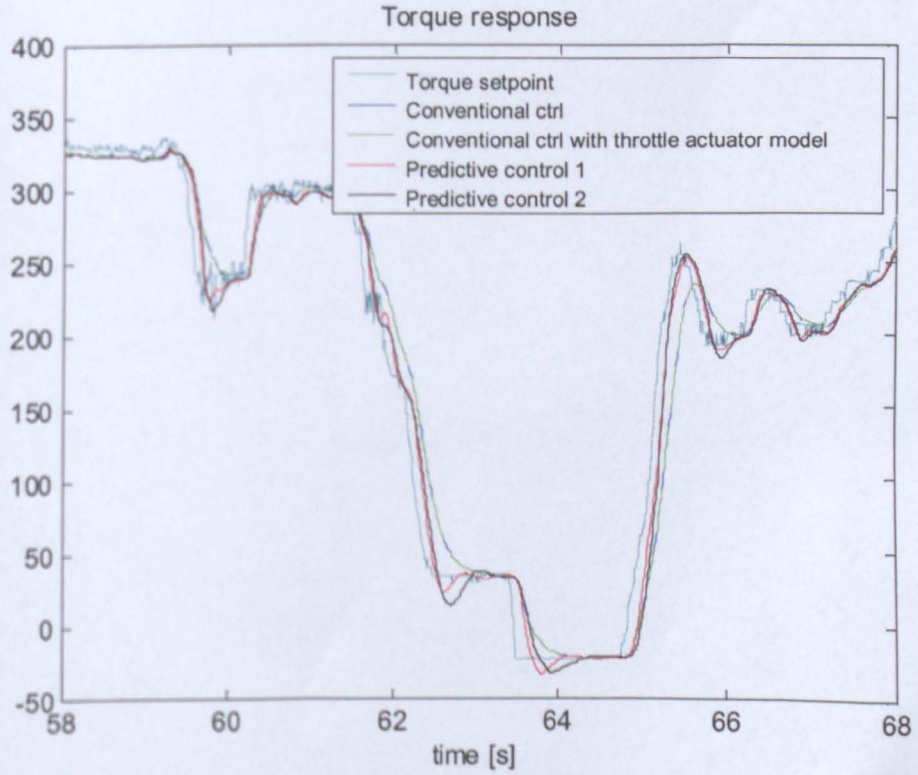

Figure 5-53: Torque US06 response

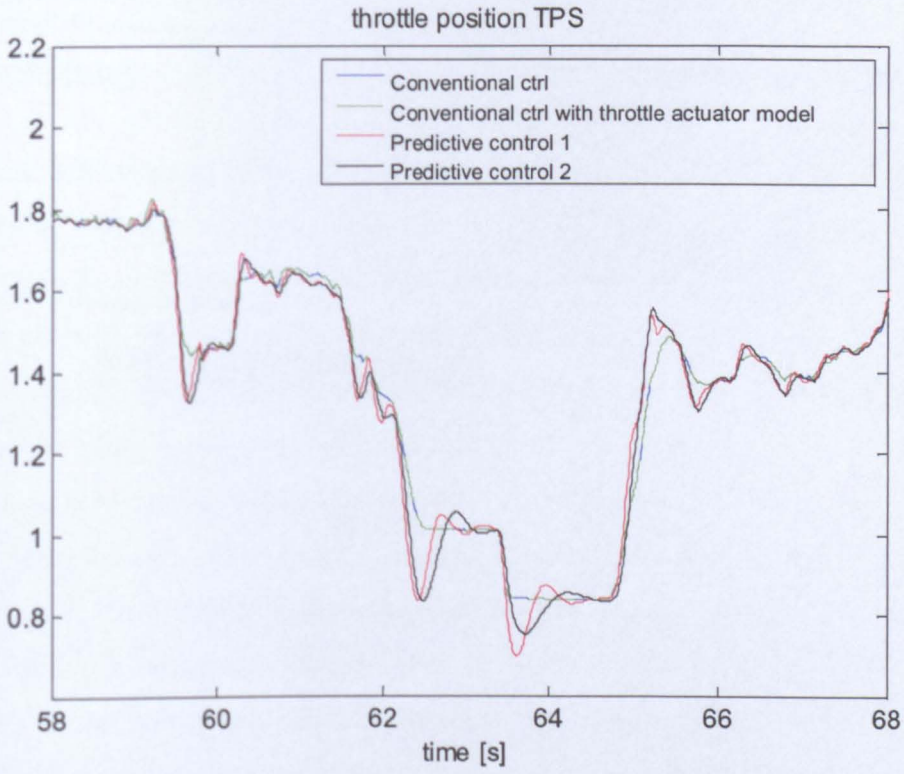

Figure 5-54: Throttle position US06 response

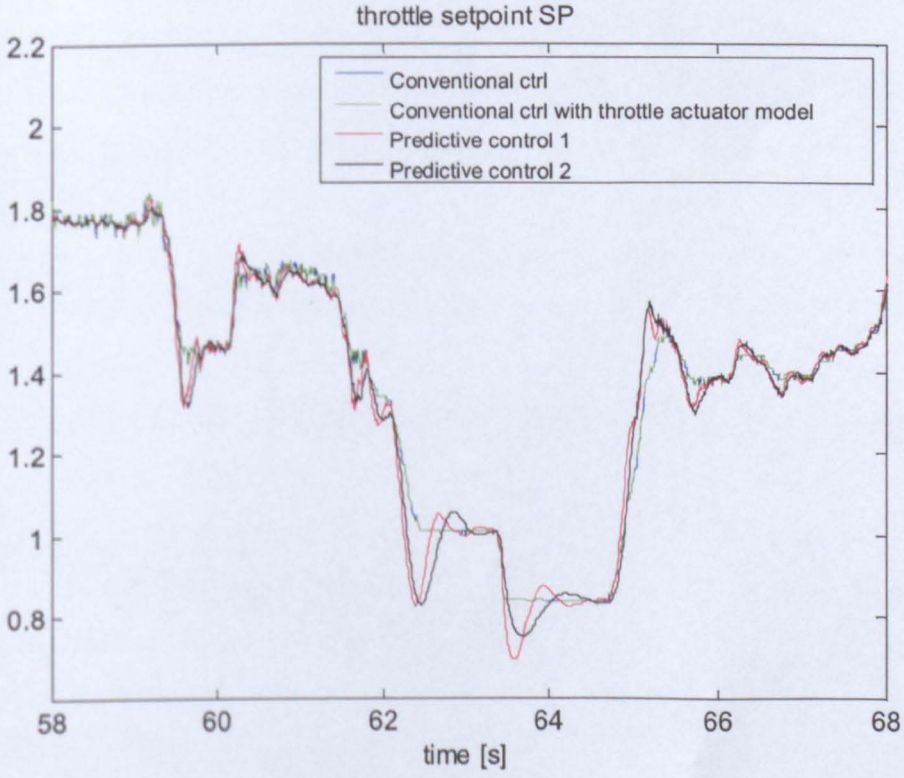

Figure 5-55: Throttle setpoint for US06

5.5 Summary

The performance of the air-fuel ratio feedforward controller is limited by the delay in the fuel delivery path. This implies that the feedforward controller must have a built-in predictor. The accuracy of that element depends on the model precision. A predictive fuel control algorithm was developed and tested in the vehicle and good results were achieved. Further improvements in the air-fuel ratio control are obtained with multi-input multi-output (MIMO) nonlinear control strategies presented in this chapter. Models of exhaust gas air-fuel ratio and generated torque are used for the design of the control system. The MIMO control strategy is based on the predictive control approach as the most natural way of obtaining an optimal solution. The multivariable solution is a trade-off between the accuracy in the regulation of air-fuel ratio (i.e. tailpipe emissions) and tracking of the requested torque profile (i.e. torque responsiveness). Considering the complexity of the non-linear engine model and constraints on actuators ranges of operation, the straightforward optimization of the cost function to obtain a solution is computationally intractable. Therefore, some simplifications based on the specific nature of the problem (e.g. relationship between torque and cylinder air charge) are made to reduce the amount of computational effort required.

Improvements in the air fuel ratio regulation may be achieved by decoupling of the accelerator pedal and throttle opening through the introduction of a simple delay in the application of throttle command. The length of the time delay may be optimized to allow for the maximum permissible air-fuel ratio excursion during the transient operations. While the introduction of the time delayed throttle actuation may slightly slow down the torque response, the benefits in terms of significantly reduced transient air-fuel ratio deviations from the stoichiometric value will outweigh the slower torque response. This is due to the fact that the development of partial burns and misfires in leaner combustion cycles during fast pedal movements can potentially reduce the generated torque with implications for driveability and performance. Therefore, the decoupling of the throttle and pedal movements and the optimization of actuator signals through the introduction of additional delays in the

air path is a reasonable approach. In addition, using a throttle actuator model and the throttle setpoint command, a more accurate throttle position is predicted and, hence, a more accurate cylinder air charge estimated. The estimated cylinder air charge is then used to provide a more accurate fuel injection command.

Additional improvements in transient torque control are achieved through the magnitude decoupling of the throttle movement and the accelerator pedal position as well. The compensation of cylinder air charge dynamics using simple classical control methods (e.g. PID control) is used to speed-up the torque response. Such dynamic corrections cannot be performed without due consideration of interactions between the air-fuel ratio and torque loops. The additional time delay – this time in the cylinder air charge setpoint rather than the throttle position – is introduced to improve the lambda regulation performance. The focus of this chapter has been on the predictive air-fuel ratio and torque control methods. This provides a suitable solution for the torque tracking problem. The predictive method is based on the minimization of a cost function defined in terms of cylinder air charge only. The amount of the fuel to be injected is then derived from the cylinder air charge prediction that is available through the predictive control algorithm. By changing the weights (i.e. tuning) and the prediction horizon, the required performance of lambda regulation and torque tracking can be influenced simultaneously.

Chapter 6

Signal Filtering, Estimation and Fault detection for the Intake manifold

In this chapter the simulation analysis and the development of filtering and fault detection techniques is presented. The simulation is based on the model developed in Chapter 4. The intake manifold subsystem of the combustion engine that consists of the electronic throttle actuator and the intake manifold will be the subject of investigation. The electronic throttle actuator is modelled here by the linear second order continuous time model. The intake manifold is modelled by the two-state model (see section 0) and is equipped with the intake manifold pressure and temperature sensors. The analyzed intake manifold and throttle subsystem is part of the engine model presented in Figure 4-1. The results presented in this chapter are based on simulation experiments.

In real systems measurements are expected to be subject to noise and/or quantization errors. Limited sensors precision is the usual engineering problem that has to be dealt with. The model based estimation greatly improves the accuracy and provides optimal solution to the noise rejection problems. The model based cylinder air charge (CAC) estimation employs extended and state-dependent Kalman filtering methods introduced in section 2.2. In contrast to the speed-density technique that uses the model given by equations (4.21) and (4.24) with parameter measurements, the estimation techniques will provide improved accuracy. This is a direct result of the data fusion. All available measurements, not only the pressure, the temperature and the engine speed, are used for the air charge estimation. This however requires

an accurate model which is of great importance for the estimation. Unfortunately, even with the perfect model and well designed estimation technique the CAC prediction may still be mismatched. This may be caused by faults occurring in the system components or in sensors. The standard sensor output signal limits checking is capable of detecting only large malfunctions. For detection of faults of lower magnitudes (e.g. moderate biases) more sophisticated algorithms are necessary. In the CAC estimation not only senor faults introduce the bias. The leak present in the intake manifold results in unmeasured increase in the air flow. This should not only be detected but also the extent of the leak should be identified. The solution to the problem is offered by fault detection and isolation theory. Traditionally fault diagnosis was based on physical redundancy that required multiple sensors. The electronic throttle may be equipped with a set of position sensors. The sensors output voltages are evaluated and unreliable sensors are excluded from the system. The method is highly reliable, but may increase the cost if used with expensive sensors. For that reason there is a significant interest in the model-based approach to the fault detection and isolation.

In this chapter the extended and state-dependent non-linear Kalman filters presented in section 2.2 are used for the noise attenuation. The generalized observer scheme is employed for the fault detection. Dedicated output observers are designed for pressure and temperature sensors (see section 2.4.2). The fault detection filter presented in section 2.4.3 is used for process faults detection. The filter generates the directional residual for the intake manifold leak. The threshold test for residuals generated by all dedicated observers is performed, which leads to the fault detection. The fault isolation based on residuals analysis is carried out next, as was described in section 2.4.4. To achieve increased reliability and improve the robustness of the fault isolation, the fault detection signal triggers the fault signature diagnosis algorithm. The set of measurements over the period of time is gathered and the properties of residuals are analyzed. The detection of temperature or pressure sensor faults triggers the observer structure reconfiguration. As a result an unbiased CAC estimate is obtained. For the intake manifold leak not only the detection and the isolation, but

also the identification of its extent is required. The non-linear fault detection filter presented in section 2.4.4 provides the solution to that problem.

6.1 The model for filtering, estimation and fault detection

The block diagram of the intake manifold and the throttle subsystem is shown in Figure 6-1.

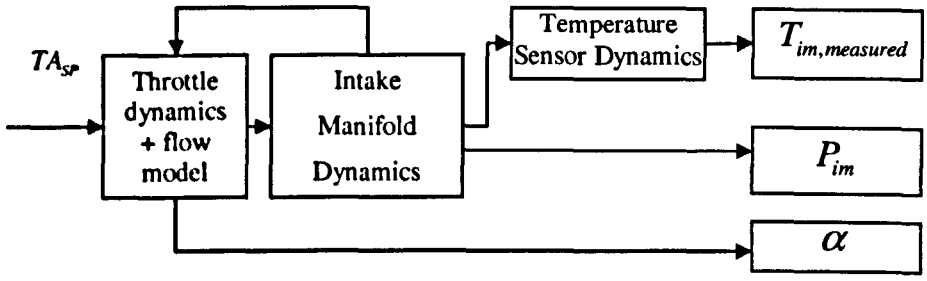

Figure 6-1: System diagram

The electronic throttle is powered by an electric motor and is controlled locally by its dedicated controller. The drive-by-wire actuator with its controller may be modelled by the second order continuous time linear system [91]:

$$\frac{\alpha}{\alpha_{SP}} = \frac{\omega_T^2}{s^2 + 2\zeta_T \omega_T s + \omega_T^2} \tag{6.1}$$

where:

α - throttle angle [rad]

α_{SP} - throttle angle setpoint command [rad]

The throttle flow is modelled by one-dimensional isentropic compressible flow equation for the flow across the orifice (see the equations (4.3), (4.4), (4.5) and (4.6) in section 4.2.2). In a similar way to the throttle flow rate an additional air flow

into the intake manifold is modelled. This additional flow will be referred to as the intake manifold leak.

$$\dot{m}_{leak} = \begin{cases} f_{leak} \cdot \dfrac{P_a}{\sqrt{R_{air} \cdot T_a}} \cdot \left(\dfrac{P_{im}}{P_a}\right)^{\frac{1}{\kappa}} \sqrt{\dfrac{2\kappa}{\kappa-1}\left[1-\left(\dfrac{P_{im}}{P_a}\right)^{\frac{\kappa-1}{\kappa}}\right]} & if \ \dfrac{P_{im}}{P_a} > \left(\dfrac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \\ f_{leak} \cdot \dfrac{P_a}{\sqrt{R \cdot T_a}} \cdot \sqrt{\kappa}\left(\dfrac{2}{\kappa-1}\right)^{\frac{\kappa+1}{2(\kappa-1)}} & if \ \dfrac{P_{im}}{P_a} \le \left(\dfrac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \end{cases} \tag{6.2}$$

where:

$\dot{m}_{leak}$ - leak flow rate,

$f_{leak} = C_{d\,leak} A_{leak}$: A_{leak} - leak area,

$C_{d\,leak}$ - discharge coefficient for the flow through the leak area.

Note that f_{leak} is regarded as a constant parameter if the leak is present in the system. The total air flow into the intake manifold is a combination of the main throttle flow and the leak. The intake manifold model is based on the energy and mass balance and on the ideal gas law. The intake manifold pressure sensor is very fast and its dynamic response may be neglected. The temperature sensor dynamic response is relatively slow and is modelled by the first order lag. The model is given by equations (4.16), (4.17), (4.18) and (4.19) that was introduced in section 0.

6.2 The model discretization and disturbance modelling

The model is discretized using the crank-based sampling rate of 90 degrees. This implies that the sampling rate varies with the crankshaft's angular speed. The throttle and the temperature sensors are given by the linear models and the discrete model is easily obtained. However, it is necessary to consider variable sampling rate during discretization. Consequently the discrete model parameters will vary with the engine speed. As was mentioned earlier, the sampling period is a function of the

engine speed and is given as $T_{s,n} = {}^{120}/_{8N_n}[s]$, where N_n is the engine speed in revolutions per minute at the discrete event n.

The electronic throttle model is discretized as follows. A non-minimal representation with three states was used due to better numerical properties for systems discretized with variable sampling rate [89]. The process noise is also included in the model.

$$\begin{aligned} x_{ET,n+1} &= A_{ET,n}x_{ET,n} + B_{ET,n}TA_{SP,n} + w_{ET,n} \\ TA_n &= C_{ET}x_{ET,n} + v_{ET,n} \end{aligned} \tag{6.3}$$

where

$w_{ET,n}$ is the throttle actuator process noise,

$v_{ET,n}$ is the throttle position measurement noise,

$$A_{ET,n} = \begin{bmatrix} 0 & 0 & 0 \\ b_{2,n} & a_{1,n} & a_{2,n} \\ 0 & 1 & 0 \end{bmatrix}; B_{ET,n} = \begin{bmatrix} 1 \\ b_{1,n} \\ 0 \end{bmatrix}; C_{ET} = [0 \quad 1 \quad 0]$$

$$b_{1,n} = 1 - \alpha_{ET,n}\left(\beta_{ET,n} + \frac{\zeta\omega_0\gamma_{ET,n}}{\omega_{ET}}\right); b_{2,n} = \alpha_{ET,n}{}^2 + \alpha_{ET,n}\left(\frac{\zeta\omega_0\gamma_{ET,n}}{\omega_{ET} - \beta_{ET,n}}\right);$$

$$a_{1,n} = -2\alpha_{ET,n}{}^2\beta_{ET,n}; a_{2,n} = \alpha_{ET,n}{}^2$$

$$\alpha_{ET,n} = \exp\left(-\zeta\omega_0 T_{s,n}\right); \omega_{ET} = \omega_0\sqrt{1-\zeta^2}; \beta_{ET,n} = \cos\left(\omega_{ET}T_{s,n}\right); \gamma_{ET,n} = \sin\left(\omega_{ET}T_{s,n}\right)$$

It should be noticed that due to the variable sampling rate $T_{s,n}$ the model must be re-discretized at each discrete event. For the non-linear intake manifold dynamics an approximate Euler discretization method was used. After the discretization, the model is parameterized into a state-dependent coefficient form (see section 2.2). The parameterized discrete form of the intake manifold model is given by the following equation:

$$x_{IM,n+1} = A_{IM,n}x_{IM,n} + B_{IM,n}\cdot C_{ET}x_{ET,n} + F_{IM,n}\cdot f_{leak,n} + w_{EM,n}$$
$$P_{im.meas,n} = C_{IM}x_{IM,n} + f_{P_{im},n} + v_{EM,n} \tag{6.4}$$

where

$f_{P_{im},n}$ is the intake manifold pressure sensor fault,

f_{leak} is the product of the leak area and the discharge coefficient,

$w_{EM,n}$ is the intake manifold process noise,

$v_{EM,n}$ is the intake manifold pressure measurement noise,

$$x_{IM,n} = \begin{bmatrix} P_{im,n} \\ T_{im,n} \end{bmatrix}$$

$$A_{IM,n} = \begin{bmatrix} \left(1 - \frac{\kappa V_{cyl}}{V_{im}}\eta_n\right) & \frac{\kappa - 1}{T_{im,n}V_{im}}T_{s,n}\dot{Q}_{ext} \\ 0 & \left(1 - \frac{\kappa V_{cyl}}{V_{im}}\eta_n\left(1 - \frac{1}{\kappa}\right)\right) + \frac{\kappa - 1}{V_{im}P_{im}(k)}T_{s,n}\dot{Q}_{ext} \end{bmatrix}$$

$$B_{IM,n} = \begin{bmatrix} \left(\frac{\kappa R_{air}T_{a,n}}{V_{im}}T_{s.n}\dot{m}_{at,n}\right)\frac{1}{C_{ET}x_{ET,n}} \\ \left(T_{im,n}\frac{\kappa R_{air}T_{a,n}}{V_{im}P_{im,n}} - T_{im,n}{}^2\frac{R_{air}}{V_{im}P_{im,n}}\right)T_{s,n}\dot{m}_{at,n}\frac{1}{C_{ET}x_{ET,n}} \end{bmatrix}; C_{IM} = [1 \quad 0]$$

$$F_{IM,n} = \begin{bmatrix} \left(\frac{\kappa R_{air}T_{a,n}}{V_{im}}T_{s.n}\dot{m}_{leak,n}\right)\frac{1}{f_{leak}} \\ \left(T_{im,n}\frac{\kappa R_{air}T_{a,n}}{V_{im}P_{im,n}} - T_{im,n}{}^2\frac{R_{air}}{V_{im}P_{im,n}}\right)T_{s,n}\dot{m}_{leak,n}\frac{1}{f_{leak}} \end{bmatrix}$$

$$V_{cyl} = \frac{V_d}{8}$$

The temperature sensor discrete time model with noise components is given by the equation (6.5).

$$T_{im.meas,n+1} = a_T T_{im.meas,n} + b_T x_{IM,n} + w_{im.meas,n}$$
$$T_{im.out,n} = T_{im.meas,n} + f_{T_{im.meas,n}} + v_{im.meas,n} \tag{6.5}$$

where:

$f_{T_{im,meas,n}}$ is the intake manifold air temperature sensor fault,

$w_{im,meas,n}$ is the temperature sensor process noise,

$v_{im,meas,n}$ is the temperature measurement noise,

$a_T = \exp(-T_s/\tau_{Temp})$; $b_T = \left[0 \quad 1-\exp(-T_s/\tau_{Temp})\right]$.

For the intake manifold estimation/fault detection only the throttle/intake manifold model and its sensors are considered. The engine speed is used as an external parameter extracted from the driving cycle test data.

The final augmented model is given by the following set of equations:

$$\begin{aligned} x_{n+1} &= A_n x_n + B_n u_n + F_n f_{leak,n} + w_n \\ y_n &= C_n x_n + f_{S,n} + v_n \end{aligned} \tag{6.6}$$

where

$$x_n = \begin{bmatrix} x_{ET,n} \\ x_{IM,n} \\ T_{im.meas,n} \end{bmatrix}; y_n \begin{bmatrix} = TA_n \\ P_{im,n} \\ T_{im.out,n} \end{bmatrix}; w_n = \begin{bmatrix} w_{ET,n} \\ w_{IM,n} \\ w_{im.meas,n} \end{bmatrix}; v_n = \begin{bmatrix} v_{ET,n} \\ v_{IM,n} \\ v_{im.meas,n} \end{bmatrix};$$

$$A_n = \begin{bmatrix} A_{ET,n} & 0 & 0 \\ B_{IM,n}C_{ET} & A_{IM} & 0 \\ 0 & b_T & a_T \end{bmatrix}; B_n = \begin{bmatrix} B_{ET,n} \\ 0 \\ 0 \end{bmatrix};$$

$$C_n = \begin{bmatrix} C_{ET} & 0 & 0 \\ 0 & C_{IM} & 0 \\ 0 & 0 & 1 \end{bmatrix}; F_n = \begin{bmatrix} 0 \\ F_{IM,n} \\ 0 \end{bmatrix}.$$

$f_{S,n} = \left[0 \quad f_{P_{im},n} \quad f_{T_{im\_meas},n}\right]^T$ is the sensor faults vector. Note that the throttle position sensor fault is not considered here due to physical redundancy present in the actuator [104]. w_n and v_n are independent white Gaussian noise signals with $\mathrm{cov}\{w_n\} = Q$ and $\mathrm{cov}\{v_n\} = R$. Q and R are diagonal semi-positive and positive definite matrices respectively.

6.3 Intake manifold filtering and estimation

In this section the filtering and estimation for the intake manifold subsystem is presented.

6.3.1 Stochastic process and measurement noise simulation setup

The filtering and estimation simulation experiment uses the model presented in section 6.2 with faults $f_{S,n}$ and $f_{leak,n}$ assumed to be zero. Unmodeled engine parameters logged in the driving cycle dataset (i.e. the engine speed, ambient conditions, the throttle position setpoint) are used in the simulation as external parameters. The air-fuel ratio control system performance strongly relies on the precision of the cylinder air charge (CAC) prediction. The CAC prediction precision relies on the accuracy of the engine parameter measurements. Noise and deterministic biases deteriorate the model-based CAC prediction. The assessment is effected by comparing the simulated delayed engine CAC with the controller internal prediction (see Figure 5-18, Figure 5-19 and Figure 4-1). Stochastic process and measurement noise is added. Since the air-fuel ratio control accuracy is proportionally influenced by the accuracy of the future CAC estimation, the CAC prediction mismatch computed as $\varepsilon = \frac{CAC_{pred} - CAC_{actual}}{CAC_{actual}} \cdot 100\%$ is a good metric for the control system performance.

Process noise introduced in the system has the following covariance Q:

$$Q = \mathrm{cov}\{w_n\} = diag\left(\mathrm{cov}\{w_{ET,n}\}, \mathrm{cov}\{w_{IM,n}\}, \mathrm{cov}\{w_{im.meas,n}\}\right)$$

$$\operatorname{cov}\{w_{ET,n}\} = diag(0, 1.95e\text{-}6, 0)$$

$$\operatorname{cov}\{w_{IM,n}\} = diag(2.5e\text{-}3, 1e\text{-}2)$$

$$\operatorname{cov}\{w_{im.meas,n}\} = [1e\text{-}4]$$

The measurement noise is characterized by the covariance R:

$$R = \operatorname{cov}\{v_n\} = diag\left(\operatorname{cov}\{v_{ET,n}\}, \operatorname{cov}\{v_{IM,n}\}, \operatorname{cov}\{v_{im.meas,n}\}\right)$$

$$\operatorname{cov}\{v_{ET,n}\} = 7.62e\text{-}5$$

$$\operatorname{cov}\{v_{IM,n}\} = 1e-2$$

$$\operatorname{cov}\{v_{im.meas,n}\} = 2.5e-3$$

The extended and state-dependent Kalman filters (EKF and SDKF) were used for the noisy measurements filtering and reconstruction of the state. The simulation setup uses the throttle actuator, the throttle flow and the intake manifold two-state models. The cylinder air charge (CAC) is used within the feedforward (FF) controller (see section 5.3). The accuracy of the CAC prediction is used as a benchmark of the control performance. The FF controller inputs are either direct measurements of intake manifold pressure, indicated throttle position and intake manifold temperature or estimates of these variables obtained from the EKF or SDKF. Also, for fair comparison, a test where the intake manifold temperature is supplied by an open-loop observer is carried out. The SDKF results are compared with the direct measurements approach results. The simulation tests are carried out with the engine parameters taken from the driving cycle data shown in Figure 6-2. The controller inputs that are either estimates or direct measurements are compared with the actual intake manifold and throttle states. The results of this comparison are presented in Figure 6-3.

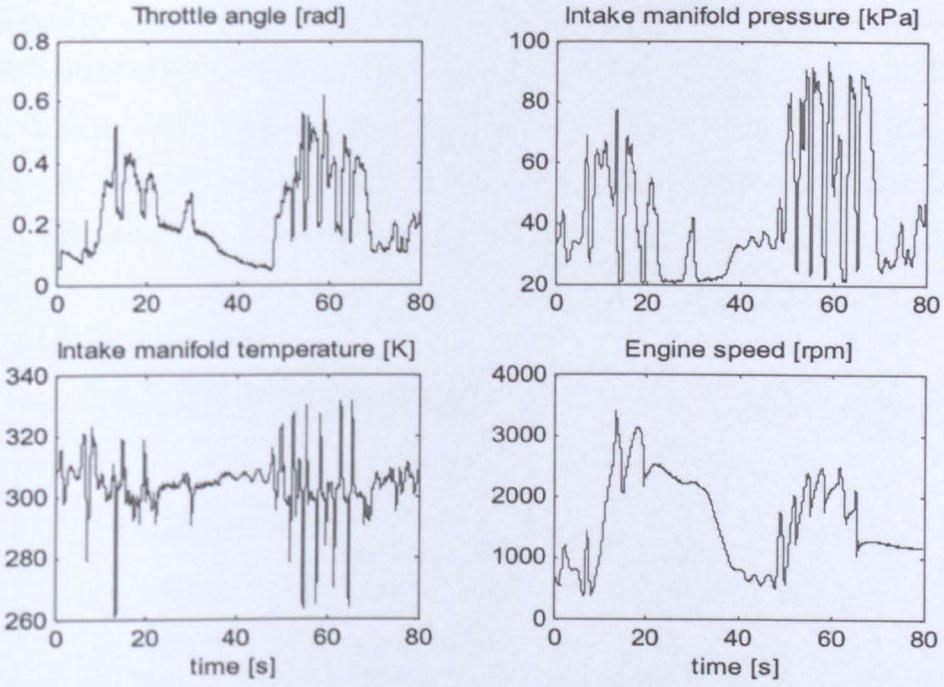

Figure 6-2: Simulation engine parameters

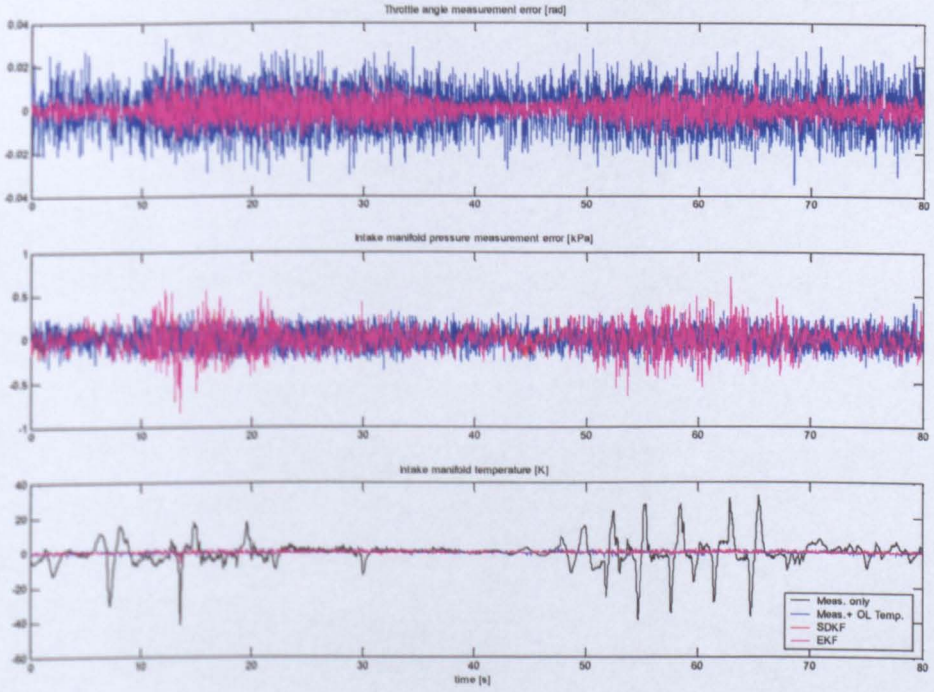

Figure 6-3: FF controller input parameters: direct measurements and Open-Loop observer

The most significant improvement is visible in the throttle angle and intake manifold temperature estimates. The pressure estimate during transients is less accurate than a direct measurement. This may be a result of the assumed noise covariance for process and measurement noise. The time plot presented in Figure 6-3 provides only an indication of the filtering efficiency. More information is obtained from analysis and comparison of integrated absolute error and squared error values for the signals in Table 6-1.

Simulation setup	$\sum\lvert\alpha_{actual}-\alpha_{controller}\rvert$	$\sum(\alpha_{actual}-\alpha_{controller})^2$	$\sum\lvert P_{im,actual}-P_{im,controller}\rvert$	$\sum(P_{im,actual}-P_{im,controller})^2$
Direct measurements	59.2848	0.6478	**6.9491e+2**	**88.0750**
Direct measurements + OL estimator	59.2848	0.6478	**6.9491e+2**	**88.0750**
SDKF	28.8459	0.1587	8.8323e+2	1.5804e+2
EKF	**28.7192**	**0.1573**	9.2527e+2	1.7474e+2

Simulation setup	$\sum\lvert T_{im,actual}-T_{im,controller}\rvert$	$\sum(T_{im,actual}-T_{im,controller})^2$
Direct measurements	4.0395e+4	5.1528e+5
Direct measurements with OL estimator	4.3142e+3	3.5022e+3
SDKF	**2.7874e+3**	**1.6566e+3**
EKF	3.2856e+3	2.7266e+3

Table 6-1: Simulation results: error signal parameters – 80 seconds, US06

The results in Table 6-1 indicate that the SDKF/EKF methods provide better results with respect to throttle angle and temperature than the direct measurement methods with and without open-loop estimation. The only exception is that the SDKF/EKF method results are slightly worse for the intake manifold pressure. It may be difficult to conclude which strategy provides better overall result. Fortunately, the most effective indicator of 'method efficiency' is the cylinder air charge prediction. As was described before, the feedforward (FF) controller used in the simulation employs a throttle actuator model for future throttle trajectory prediction. This trajectory prediction is used by FF controller to generate the cylinder air charge (CAC) prediction. The 6 events prediction is compared with the actual cylinder air charge being an internal variable of the simulated intake manifold. The

accuracy of the CAC prediction over the simulation time of 80 seconds is evaluated based on the integrated absolute and squared error value. The traces showing the cylinder air charge signal precision are given in Figure 6-4. Additionally the error signal statistics are presented in Table 6-2.

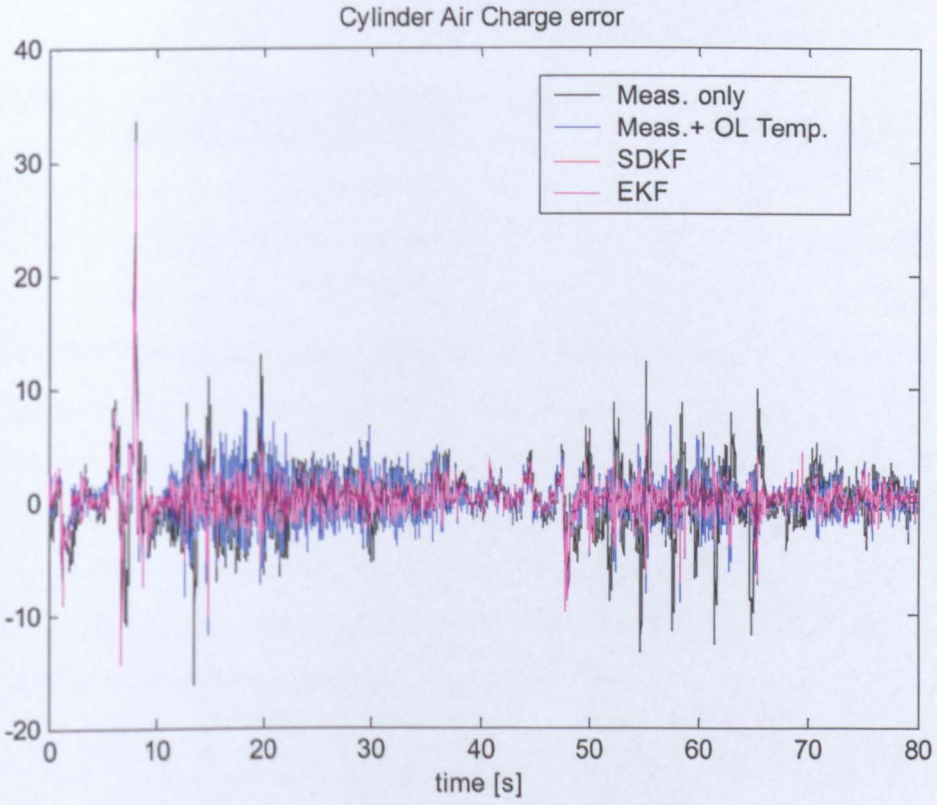

Figure 6-4: Cylinder Air Charge estimation error [%]

Simulation setup	$\sum\lvert CAC_{actual} - CAC_{predicted}\rvert$	$\sum(CAC_{actual} - CAC_{predicted})^2$
Direct measurements	15.376e+3	6.1437e+4
Direct measurements with OL estimator	9.9426e+3	2.5435e+4
SDKF	7.1240e+3	1.5913e+4
Recursive EKF	**7.1006e+3**	**1.5817e+4**

Table 6-2: Cylinder air charge error signal parameters

The results indicate that the extended Kalman filter provides the best accuracy. The state-dependent Kalman filter, however, gives very similar results.

Results obtained using the estimation methods when compared to direct measurement methods indicate that improved prediction of cylinder air charge prediction is achievable. This improved CAC prediction, of course, directly results in significantly improved air-fuel ratio control precision.

6.3.2 Parameter variation analysis simulation setup

The results presented in the previous section indicate improved cylinder air charge (CAC) prediction accuracy when either extended or state-dependent Kalman filters (EKF, SDKF) are used. The derivation of the EKF (or SDKF) is based on the assumption that the process and measurement noise signals are stochastic. This method of modelling – especially for the model mismatch represented by the process noise is not always correct. In this section parameter variation is introduced in the intake manifold and throttle model. Additionally, sensor gain errors are introduced for throttle position, intake manifold pressure and temperature measurements. The system diagram including an indication of the point of introduction of the parameter variation/uncertainty is shown in Figure 6-5.

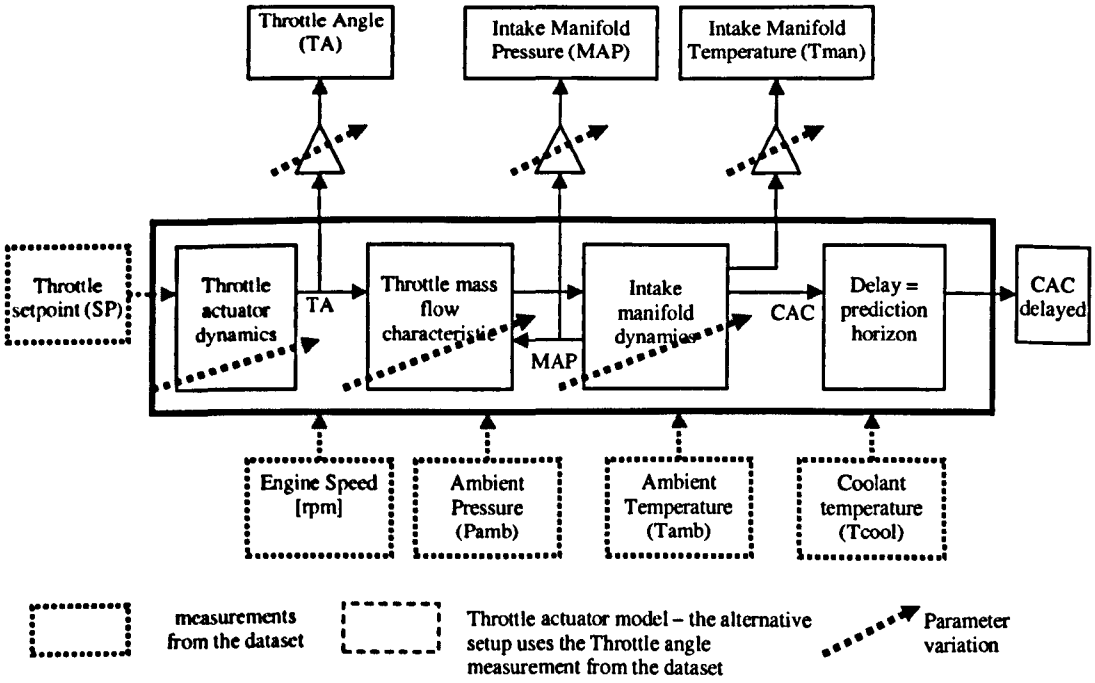

Figure 6-5: Engine simulation block – parameter variation

A Monte-Carlo parameter variation analysis was conducted in a similar way to that in [96]. The parameters that were determined to be subject to variation are displayed in Table 6-3. A truncated Gaussian probability density is assumed within the parameter variation limits.

Parameter of interest	Error definition	Parameter variation limit
ζ_T in (6.1)	$\varepsilon_{\zeta_T} = 100\left(\left(\zeta_{T,\text{modelled}} - \zeta_{T,\text{actual}}\right)/\zeta_{T,\text{actual}}\right)$	$\varepsilon_{\zeta_T} = \pm 5[\%]$
ω_T in (6.1)	$\varepsilon_{\omega_T} = 100\left(\left(\omega_{T,\text{modelled}} - \omega_{T,\text{actual}}\right)/\omega_{T,\text{actual}}\right)$	$\omega_{\zeta_T} = \pm 5[\%]$
$C_d \cdot A_{th}$ in (4.3), (4.5)	$\varepsilon_{C_d \cdot A_{th}} = 100\left(\frac{C_d \cdot A_{th\ \text{modelled}} - C_d \cdot A_{th\ \text{actual}}}{C_d \cdot A_{th\ \text{actual}}}\right)$	$\varepsilon_{C_d \cdot A_{th}} = \pm 3[\%]$
h_1 in (4.16)	$\varepsilon_{h_1} = 100\left(\left(h_{1,\text{modelled}} - h_{1,\text{actual}}\right)/h_{1,\text{actual}}\right)$	$\varepsilon_{h_1} = \pm 5[\%]$
h_2 in (4.16)	$\varepsilon_{h_2} = 100\left(\left(h_{2,\text{modelled}} - h_{2,\text{actual}}\right)/h_{2,\text{actual}}\right)$	$\varepsilon_{h_2} = \pm 5[\%]$
$\eta = \eta(N, P_{im})$ in (4.16)	$\varepsilon_{\eta} = 100\left(\left(\eta_{\text{modelled}} - \eta_{\text{actual}}\right)/\eta_{\text{actual}}\right)$	$\varepsilon_{\eta} = \pm 4[\%]$
$T_{im,meas}$ measurement error	$\varepsilon_{T_{im,meas}} = 100\left(\frac{T_{im,meas\,\text{measured}} - T_{im,meas\,\text{actual}}}{T_{im,meas\,\text{actual}}}\right)$	$\varepsilon_{T_{im,meas}} = \pm 2[\%]$
P_{im} measurement error	$\varepsilon_{P_{im}} = 100\left(\left(P_{im\,\text{measured}} - P_{im\,\text{actual}}\right)/P_{im\,\text{actual}}\right)$	$\varepsilon_{P_{im}} = \pm 2[\%]$
α measurement error	$\varepsilon_{\alpha} = 100\left(\left(\alpha_{\text{measured}} - \alpha_{\text{actual}}\right)/\alpha_{\text{actual}}\right)$	$\varepsilon_{\alpha} = \pm 3[\%]$

Table 6-3: The assumed parameters and measurement error variation

The results of the simulations are presented and analyzed below. The histograms based on 2000 simulations are presented (Figure 6-6...Figure 6-13). The analysis of histograms in Figure 6-6 and in Figure 6-7 reveals that the SDKF and EKF filters improve the robustness of the cylinder air charge estimation to the combination of modelling and measurement errors proposed in Table 6-3. Better performance is indicated in any given histogram by a higher number of simulation results (with either absolute or squared integrated error values) occurring at the lower error levels. In each container in the histogram the number of simulation results with the integrated squared or absolute error within the limits is counted. Obviously the best possible result would have all results in the container with the lowest error

limits. The accuracy of the statistical method used here relies on a high number of simulations being carried out. The actual distributions of varied parameters are presented in Figure 6-14.

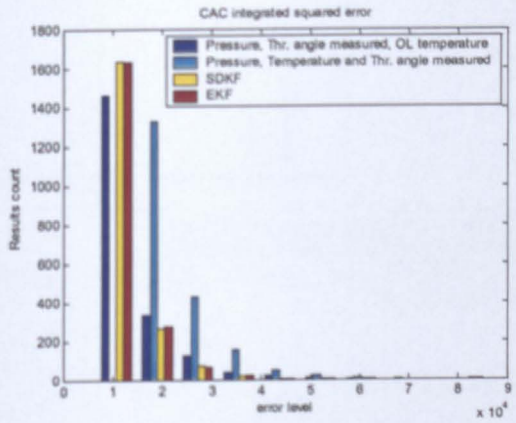

Figure 6-6: CAC int. squared error histogram

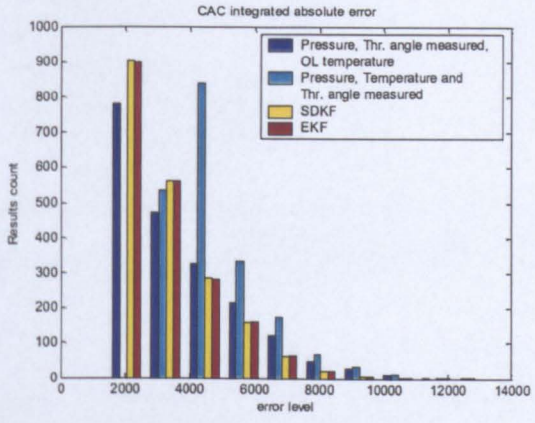

Figure 6-7: CAC int. absolute error histogram

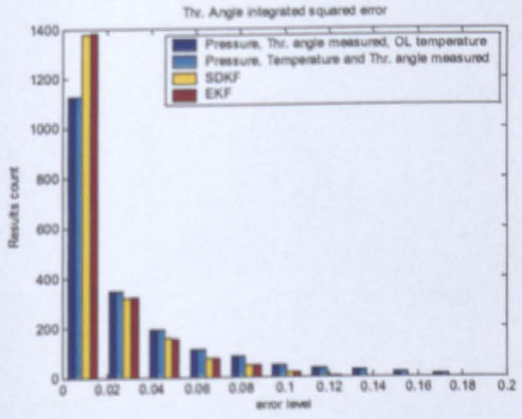

Figure 6-8: Thr. A. int. squared error histogram

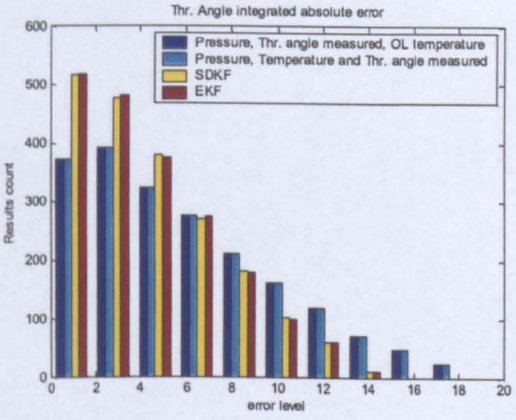

Figure 6-9: Thr. A. int. absolute error histogram

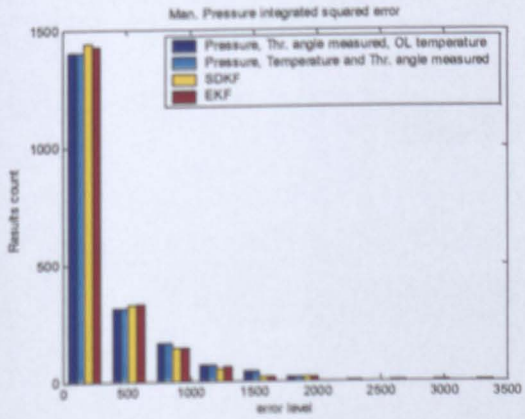

Figure 6-10: Press. int. squared error histogram

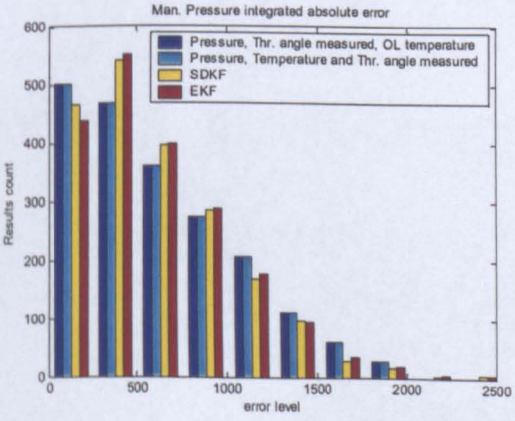

Figure 6-11: Press. int. absolute error histogram

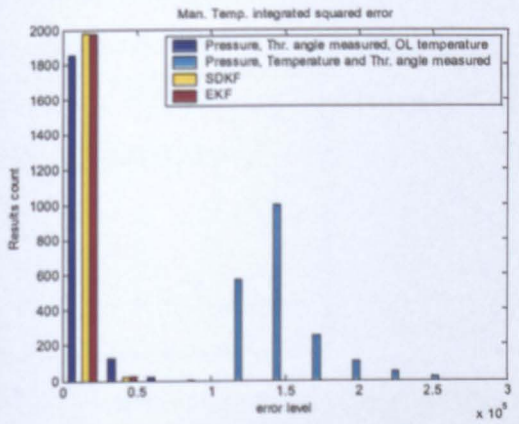

Figure 6-12: Temp. int. squared error histogram

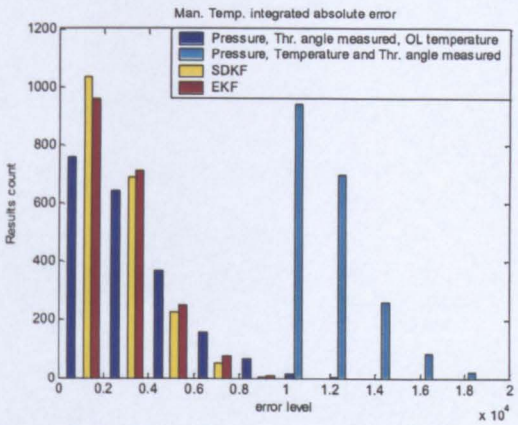

Figure 6-13: Temp. int. absolute error histogram

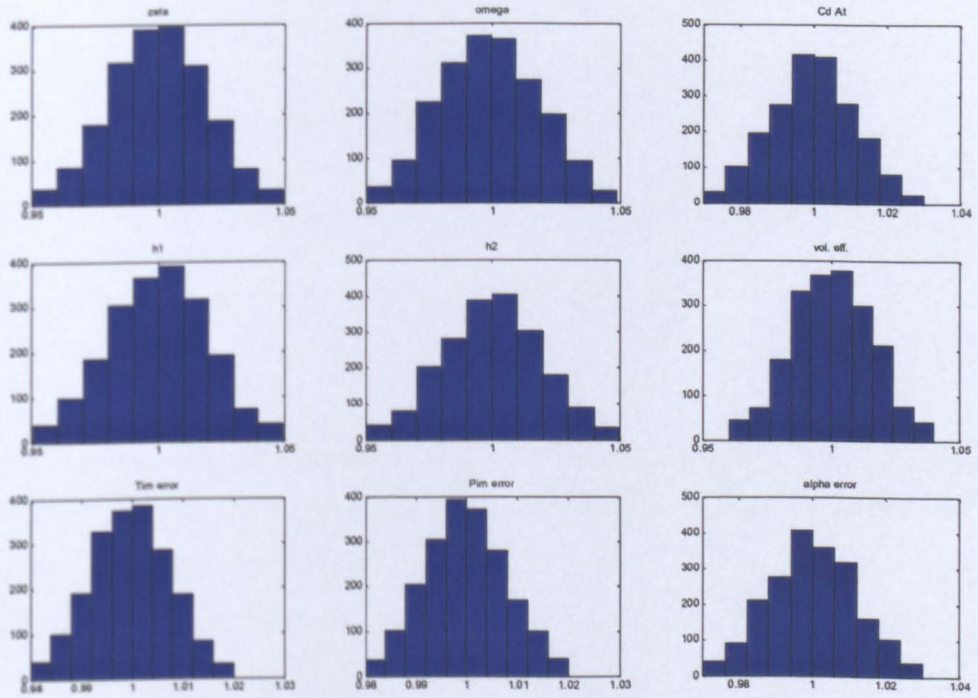

Figure 6-14: Distributions of parameter variations

It can be seen in the histogram for the intake manifold pressure estimation accuracy in Figure 6-10 and Figure 6-11 that filtering does not provide obvious improvement. A similar result for the pressure estimation was noted in section 6.3.1. The mean error values computed based on all of 2000 simulations indicate, however, an improvement for the EKF/SDKF methods over the conventional methods based on

direct measurements. The results are gathered in Table 6-4. The SDKF provides the overall best results in this comparison. This indicates that in the presence of model mismatch, this filtering method is likely to provide the best performance among the considered techniques.

	P_{im}, α, meas., O-L $T_{im,meas}$	P_{im}, α, $T_{im,meas}$ measured	SDKF	EKF
$mean\left\{\sum\left(CAC_{actual}-CAC_{predicted}\right)^2\right\}$	13264.95	23384.67	**11487.33**	11516.52
$mean\left\{\sum\left\|CAC_{actual}-CAC_{predicted}\right\|\right\}$	3646.21	4849.68	**3222.27**	3230.37
$mean\left\{\sum\left(\alpha_{actual}-\alpha_{controller}\right)^2\right\}$	0.02763	0.02763	0.01684	**0.01673**
$mean\left\{\sum\left\|\alpha_{actual}-\alpha_{controller}\right\|\right\}$	5.8347	5.8347	4.3570	**4.3437**
$mean\left\{\sum\left(P_{im,actual}-P_{im,controller}\right)^2\right\}$	300.28	300.28	**283.66**	295.65
$mean\left\{\sum\left\|P_{im,actual}-P_{im,controller}\right\|\right\}$	615.33	615.33	**595.72**	605.96
$mean\left\{\sum\left(T_{im,actual}-T_{im,controller}\right)^2\right\}$	8705.99	49366.81	**4684.20**	5301.60
$mean\left\{\sum\left\|T_{im,actual}-T_{im,controller}\right\|\right\}$	3178.60	12384.92	**2374.85**	2520.68

Table 6-4: Mean values of the integrated errors computed based on 2000 tests

The last comparison of presented methods uses the worst case analysis. The 512 simulations with all possible combinations of parameters assumed to be equal to its extreme values (as in Table 6-3) is carried out. The results are gathered in Table 6-5. The overall best result measured by the CAC error is obtained using the EKF. The SDKF in that respect is just slightly worse and methods without the model based filter provide higher errors. Only the 'worst case' intake manifold pressure estimation error is higher.

The simulation analysis of the accuracy of the cylinder air charge prediction carried out in this section indicated a significant improvement. Robustness was assessed using two different approaches. The presented simulation analysis involved feeding of process and measurement stochastic noise into the simulated intake manifold model. To provide more realistic robustness test environment, the system parameter variation was introduced. The extended Kalman filter (EKF) brought the

best performance. The state-dependent Kalman filter (SDKF) was only slightly worse in terms of integrated squared and absolute prediction errors. In the robustness test (Monte-Carlo) the SDKF provided slightly better performance. One important advantage should not be overlooked. The state-dependent form of the model is simpler than the linearized form. This fact may be important during on-line implementation of the filter. In the next section the fault detection algorithm design will be elaborated and tested.

	P_{im}, α, meas., O-L $T_{im,meas}$	P_{im}, α, $T_{im,meas}$ measured	SDKF	EKF
$\max\left\{\left(CAC_{actual}-CAC_{predicted}\right)^2\right\}$	113872.52	151555.06	102645.97	**102116.67**
$\max\left\{CAC_{actual}-CAC_{predicted}\right\}$	15266.07	17508.07	14455.24	**14413.58**
$\max\left\{\left(\alpha_{actual}-\alpha_{controller}\right)^2\right\}$	0.18340	0.18340	0.12424	**0.12343**
$\max\left\{\alpha_{actual}-\alpha_{controller}\right\}$	18.5219	18.5219	14.5168	**14.4713**
$\max\left\{\left(P_{im,actual}-P_{im,controller}\right)^2\right\}$	**2165.56**	**2165.56**	6409.40	6689.56
$\max\left\{P_{im,actual}-P_{im,controller}\right\}$	**2049.08**	**2049.08**	3497.53	3559.48
$\max\left\{\left(T_{im,actual}-T_{im,controller}\right)^2\right\}$	258522.06	276463.23	**135201.03**	152228.06
$\max\left\{T_{im,actual}-T_{im,controller}\right\}$	21655.53	19771.30	**16400.31**	17267.74

Table 6-5: Maximum error values for the worst case analysis

6.4 The fault detection

Fault detection for the combustion engine was the subject of a number of research papers [98], [99], [100]. Current production systems are based mainly on simple limit and plausibility checks of measured signals. Some simple signal-based methods like frequency analysis are also employed [97]. This however may not be sufficient in a future. The analysis of the intake manifold system is presented in [101], [102]. The hypothesis testing framework based on the system model was used for the intake manifold leaks and sensor faults detection. In this section the non-linear observer-based fault detection for the intake manifold is considered. The fault

detection, isolation and identification is carried out based on the generalized observer scheme.

The system state-space model with faults modelled as an additive signals is given by the equation (6.6). The intake manifold pressure and the temperature sensor faults are modelled as additive perturbations in equations (6.4) and (6.5) respectively. The intake manifold leak is modelled in a similar way to the throttle flow in the state-space model (6.4). The product of the leak area and the discharge coefficient is defined as the unknown input f_{leak}. The pressure sensor fault $f_{P_{im},n}$ is modelled as an additive saturated ramp signal. The signal saturates after 1s at the level of -5kPa. The temperature fault $f_{T_{im,meas},n}$ is also modelled in the same way as an additive signal that saturates after 1s at -20K. These changes simulate the measurement offset. In practice, a quite common situation is when the sensor measurement freezes at certain value. This fault may be simulated by the time-varying additive signal. The features (e.g. saturated ramp) of the sensor fault signal used for the simulation are not exploited by the fault detection algorithm. The results obtained in this chapter may be generalized to faults modelled by any time-varying additive signal of sufficient magnitude. The intake manifold leak is assumed to appear in a more abrupt way. The initial ramp part lasts only 0.2s and the leak fault reaches a value of $2\left[\sqrt{N\cdot m\cdot g}/s\cdot kPa\right]$. This may be seen as a vacuum hose being pulled off the intake manifold [103]. The constant value of the leak fault signal will be important for the fault isolation method employing the variance analysis.

For the work presented here it is assumed that faults are not appearing simultaneously. The Electronic throttle is assumed to be equipped with redundant position sensors and consequently its fault diagnosis may be based on the physical redundancy within its dedicated control system [104]. The fault detection is carried out in three stages (see section 2.4 for details): Failure detection, Isolation and Identification. The first task that is analysed is the residual signals generation presented in the next section.

6.4.1 Residuals generation – Dedicated observer scheme

First and the most important task of the fault detection is residuals generation. On the system diagram in Figure 6-15 three faults: two sensor faults and one system failure are shown. Sensor faults are modelled as additive signals on outputs in equations (6.4), (6.5). The system fault is modelled as an unknown input to the system in the equation (6.4).

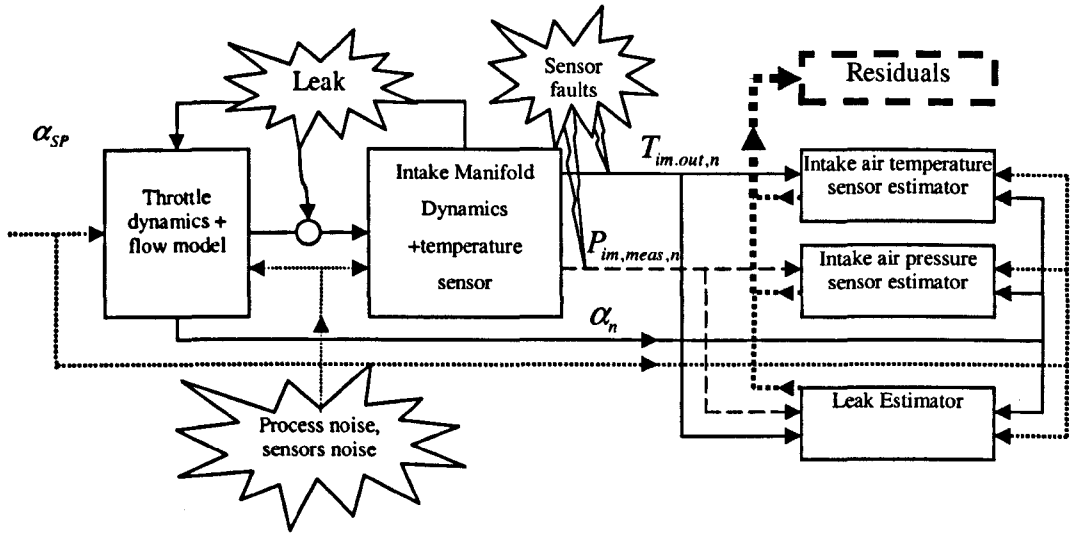

Figure 6-15: System faults and dedicated estimator scheme

The dedicated estimator scheme requires a separate residual generator for each fault. For sensor faults two separate observers are constructed. For the intake manifold leak the fault detection filter with the directional residual is built (see section 2.4.3).

The sensor faults directional residuals generation

Each dedicated observer uses the throttle angle setpoint command α_{SP}, the throttle angle measurement α_n and either the temperature $T_{im.out,n}$ or the pressure

$P_{im,meas,n}$ measurement (Figure 6-15). Each observer carries out diagnosis of only one output signal that is subject to the failure. This provides an isolation of each individual estimator. The intake air temperature sensor estimator detects the temperature sensor failure and is not sensitive to the pressure sensor fault. Similarly the intake air pressure sensor estimator detects the pressure sensor failure and is insensitive to the temperature sensor fault. Under hypothesis that the intake manifold leak is not present, the sensor fault detection is established through analysis of residuals (difference between estimated and measured output) generated by each estimator. The residuals for the fault-free system and the system with fault for pressure and temperature sensor fault are shown in Figure 6-16 and in Figure 6-17 respectively.

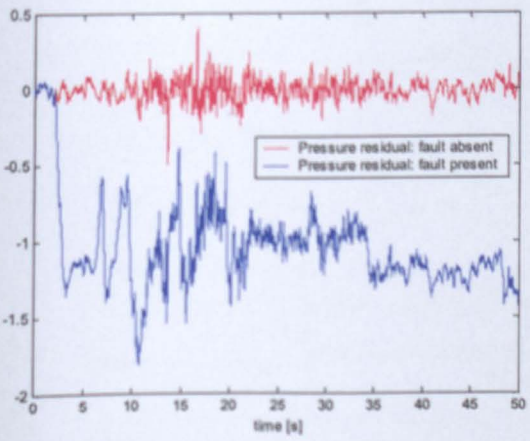

Figure 6-16: Pressure measurement residual

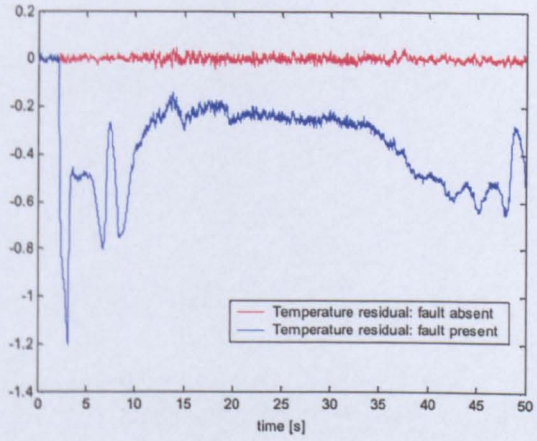

Figure 6-17: Temperature measurement residual

Residual signals throughout this chapter are pre-filtered using the discrete filter $G(z^{-1}) = \frac{0.1z^{-1}}{1-0.9z^{-1}}$. The generated residuals may be related to the fault magnitudes (given at the beginning of section 6.4) that are acting as an additive signals on outputs. The residuals do not reflect the actual level of faults. This is due to the feedback action from residuals to the state estimates and to the model used by the estimator. This feedback action results in the state estimation offset. This problem was analysed in section 2.4.2. The actual and estimated states are shown in Figure

6-18 and in Figure 6-19 for the pressure and the temperature faults respectively. The state estimation bias introduces model mismatch in the state dependent model. That may have a negative impact on fault representation in residuals. It is extremely difficult to analyze this internal feedback in the model due to the complexity of non-linear structure. Note that in Figure 6-16 the value of pressure residual varies with operating conditions (e.g. the intake manifold pressure shown in Figure 6-18 as the state 4) while it is known that the fault magnitude is constant.

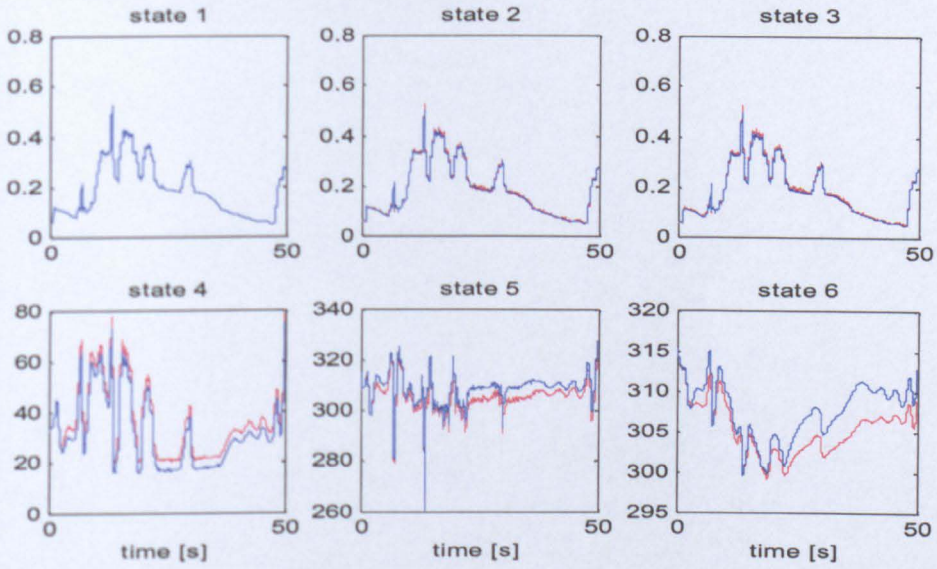

Figure 6-18: State estimates (blue) and actual states (red) for the pressure estimator and sensor fault

The temperature residual in Figure 6-17 also varies with operating conditions. It is important to assess the signal to noise ratio with respect to the residual signals. For situations where the noise results in a significant variation of residuals only faults of higher magnitude may be detected. The detection of faults is facilitated through the analysis of residuals. The threshold tests are the most common fault detection method. The fault is indicated when the value of residual reaches certain pre-defined level. The level must be set such that in the fault-free operating conditions residuals stay below the threshold level. The process noise that may also be regarded as model mismatch is the main factor that determines the robust level of thresholds. It is desired that the fault-free residuals differ significantly from the fault-present values. This allows setting the threshold with a sufficient safety margin. The faults of relatively low magnitude may thus be detected. The Kalman filter (or an observer)

due to feedback action may decrease fault sensitivity of residuals. It must be remembered that Kalman filtering methods rely upon the process and measurement noise being stochastic signals and provide optimal results for disturbances that are characterized by the noise covariances information provided to the filter. The fault enters the system in the same way as the sensor noise. It does not have the same statistical properties and the state update may result in poor residual sensitivity to the system fault.

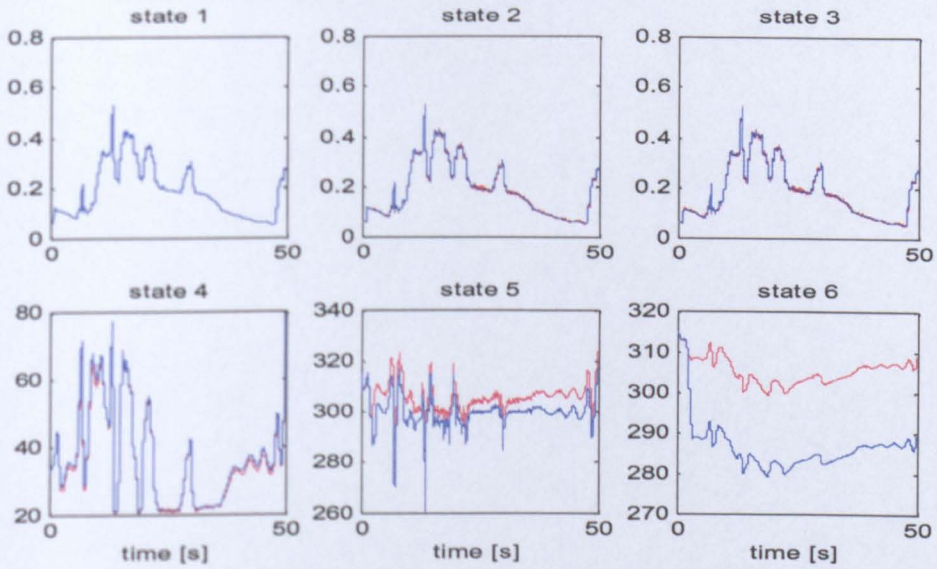

Figure 6-19: State estimates (blue) and actual states (red) for the temperature estimator and sensor fault

The intake manifold system model is open-loop stable. The estimator undesired state estimation offset and lower sensitivity of residuals to faults provided a motivation to use the open-loop estimation. The analysis and justification for use of such approach was given in section 2.4.2. The system may be split in two subsystems. In Figure 6-20 a separate state-dependent Kalman filter for the throttle angle estimation is designed. The throttle angle estimate is used by the open-loop observer to compute the intake manifold pressure and temperature. The residuals are the difference between the estimates and the measurements. The residual for the pressure sensor fault is presented in Figure 6-21. The relative difference in magnitude between fault and fault-free residual is better than for the Kalman filter residuals in Figure 6-16. Also the residuals are less dependent upon operating

conditions. The same conclusion may be reached after analysis of temperature sensor fault detection residuals in Figure 6-22. The Kalman filter counterpart for the same fault magnitude and simulation test cycle is presented in Figure 6-17. This indicates that for the intake manifold sensor fault residuals generation mixed structure (in Figure 6-20) gives the best results. The state-dependent Kalman filter for the throttle angle estimation is combined with the open loop observer for the intake manifold pressure and temperature.

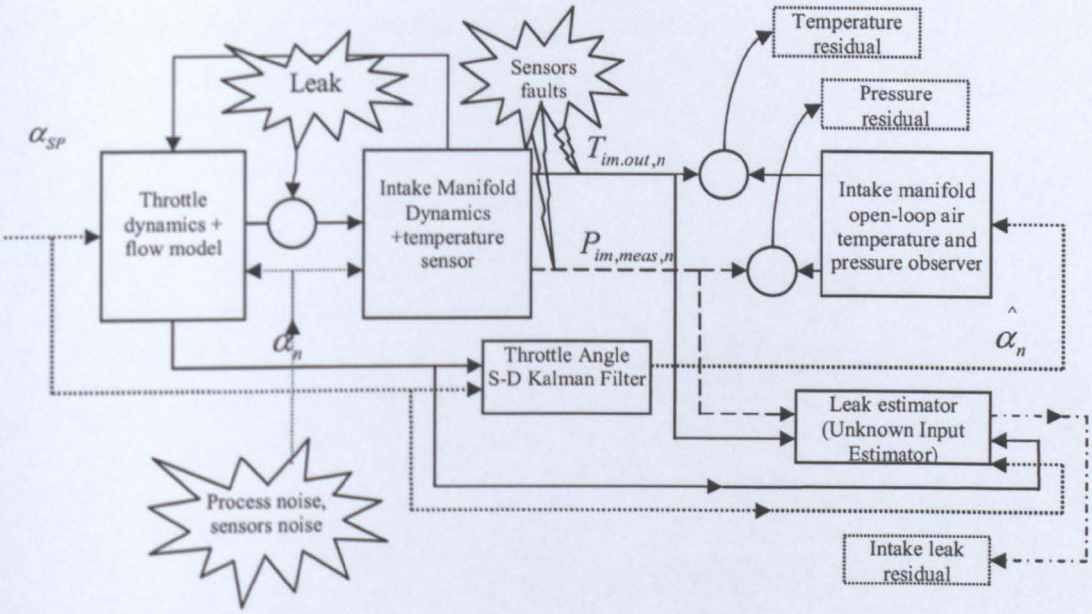

Figure 6-20: System faults and dedicated estimator scheme

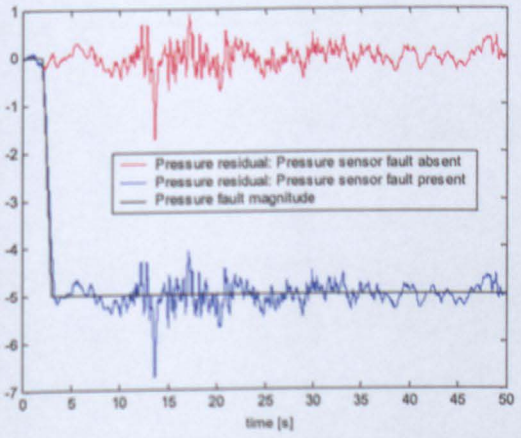

Figure 6-21: Pressure measurement residual – O.L. estimator

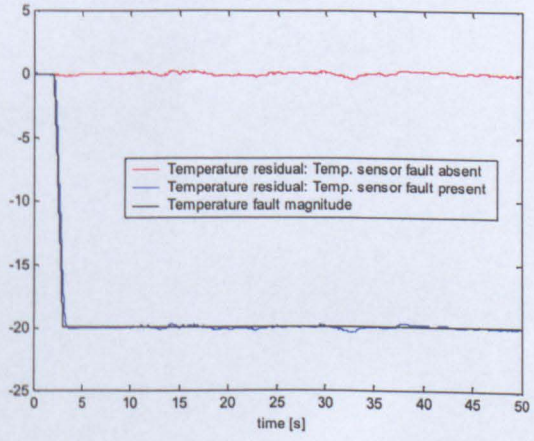

Figure 6-22: Temperature measurement residual – O.L. estimator

The intake manifold leak directional residual generation

The intake manifold leak is estimated by the fault detection filter presented in section 2.4.3. The system diagram with the leak estimator is shown in Figure 6-20. The filter generates a directional residual that is sensitive to the leak. The intake manifold leak directional residual is shown in Figure 6-23. The remaining estimator residuals decoupled from the fault are shown in Figure 6-24. These facilitate unbiased estimation and the process and the measurement noise attenuation. The leak estimation is carried out under the assumption that output (sensor) faults are not present in the system.

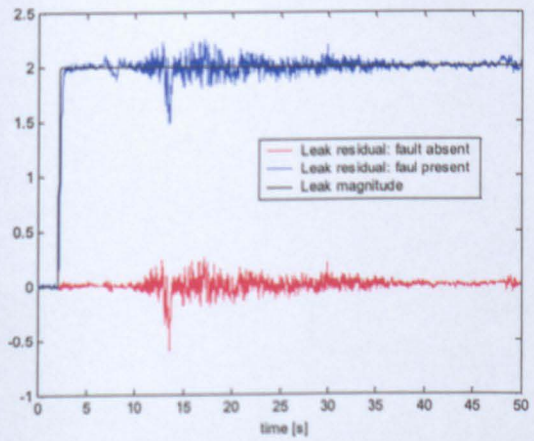

Figure 6-23: The intake manifold leak directional residual

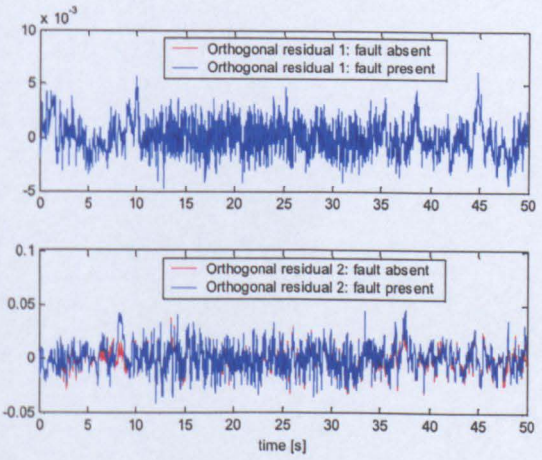

Figure 6-24: The intake manifold leak decoupled residuals

Dedicated estimators sensitivity to other faults

In the analysis the sensor faults residuals were generated with the assumption that the leak is not present in the system. A similar assumption about sensor faults was made for the leak directional residual generation. It is important to assess the result of the leak on the pressure and temperature sensor residuals. Also, the intake manifold leak directional residual reaction to the pressure and temperature fault is important for the design of fault detection logic. Reaction of the pressure and

temperature dedicated residuals to the intake manifold leak is shown in Figure 6-26 and in Figure 6-25 respectively. It should be noticed that the leak entails the change in pressure and temperature residuals levels. It is important to notice that the level for these residuals varies with the operating condition. For the intake manifold it may be difficult to detect the leak when the throttle is wide open. The influence of such failure may, depending on the extent of leak, be negligible. In such a situation the fault detection is not critical. The product of the throttle area and discharge coefficient over the simulation is shown in Figure 6-29. This may be related to the leak $f_{leak,n} = 2$. Over the full simulation the leak is at least 12.5% relative to the throttle area.

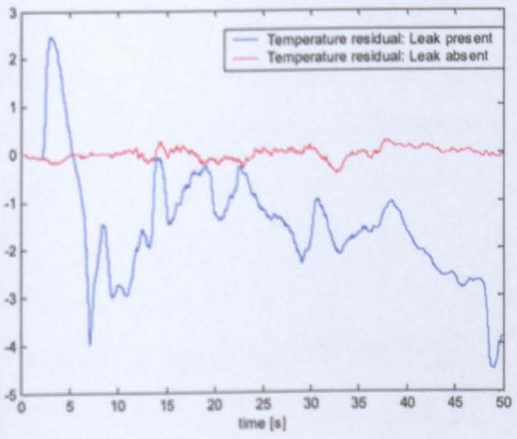

Figure 6-25: Temperature residual response to the leak – Open-Loop observer

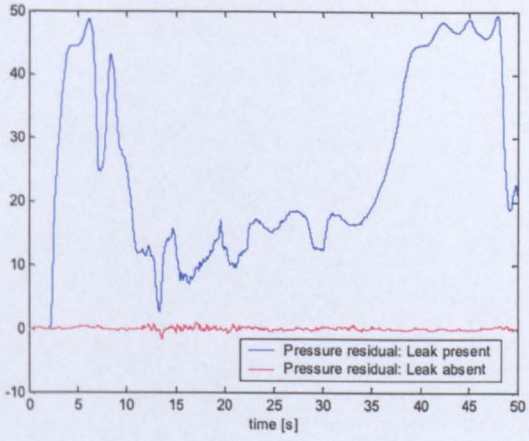

Figure 6-26: Pressure residual response to the leak – Open-Loop observer

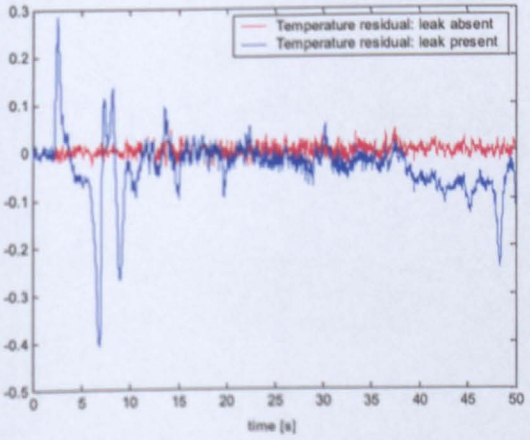

Figure 6-27: Temperature residual response to the leak – State-Dependent observer

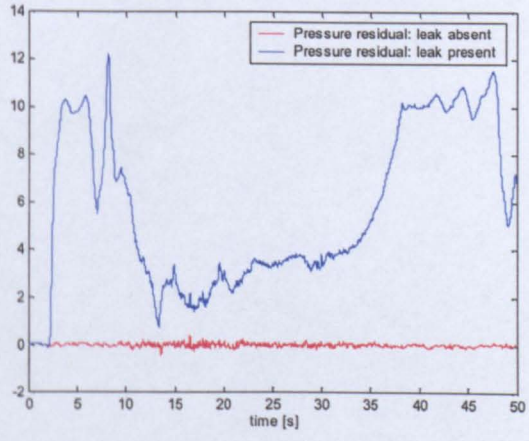

Figure 6-28: Pressure residual response to the leak State-Dependent observer

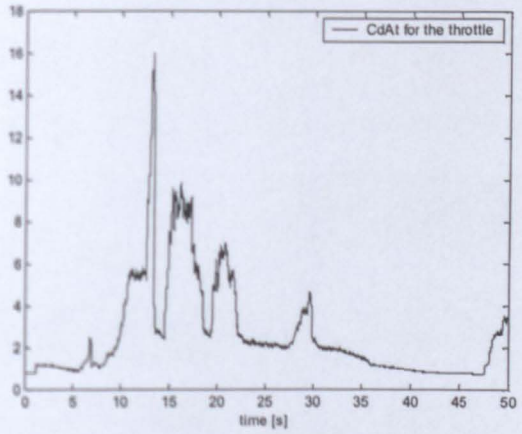

Figure 6-29: Product of throttle discharge coefficient and the throttle area

The intake manifold leak estimation is carried out with the assumption that sensor faults are not present in the system. The residuals are affected by the sensor faults since the pressure and temperature measurements are used by the leak estimator. The pressure sensor fault results in the leak directional residual trace shown in Figure 6-30. For the temperature sensor fault the leak directional residual is plotted in Figure 6-31. It should be noticed that the leak residual strongly relies on the pressure sensor information. This is due to the lower fault detectability index for this output. The temperature sensor fault results in a significantly lower residual change.

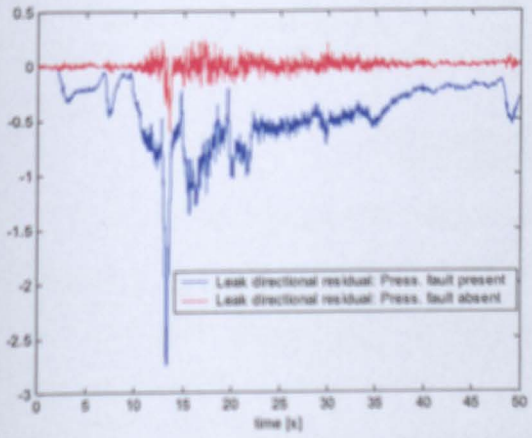

Figure 6-30: Leak directional residual resulting from the pressure sensor fault

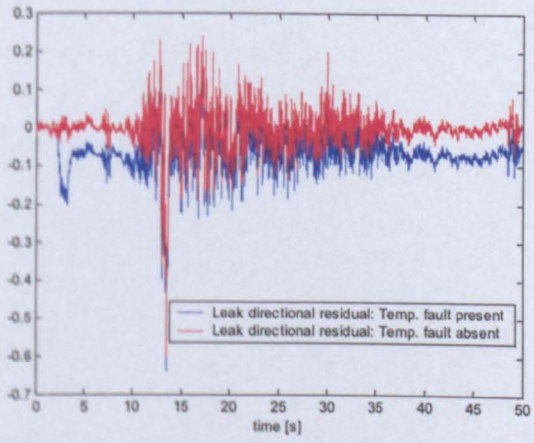

Figure 6-31: Leak directional residual resulting from the temperature sensor fault

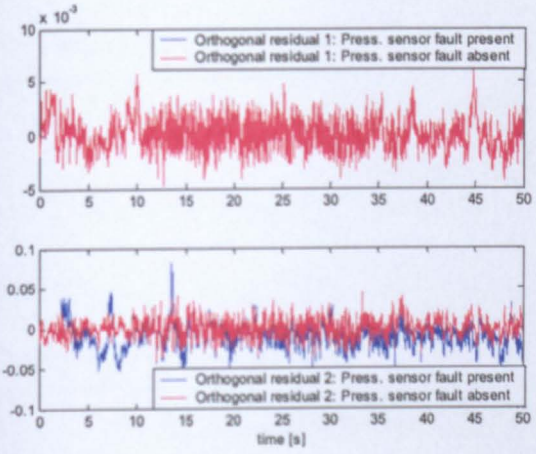

Figure 6-32: Leak orthogonal residuals response to the Pressure sensor fault

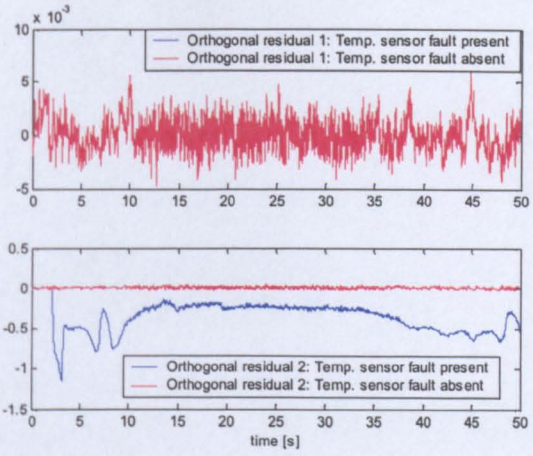

Figure 6-33: Leak orthogonal residuals response to the Temperature sensor fault

6.4.2 Determination of thresholds

The residuals generation is the first step towards fault detection. For stochastic or uncertain systems the residuals are not zero for the fault-free system. The threshold for residuals must be established. If residuals are within the pre-defined boundaries the system is assumed to be fault-free. The residual(s) exceeding threshold(s) indicate the fault in the system. It is important to define thresholds in a way that the system noise or uncertainty does not trigger the fault alarm or system reconfiguration. In practice thresholds should be based on the information about extreme values for the fault-free system. The formal analytical derivation of thresholds for complex system like the intake manifold may not be possible in practice. It may also be too conservative if a multiple models approach or worst case analysis is used.

In the work presented in this chapter, thresholds are determined based on the simulation of the system and the fault-free trace of residuals shown in Figure 6-34 (also in Figure 6-21, Figure 6-22 and Figure 6-23). A safety margin was also included to improve robustness of the design. For the pressure sensor residuals, the threshold of $\pm 2[kPa]$ was determined. For the temperature sensor residual the threshold is set to $\pm 0.5[K]$. The intake manifold leak residual in Figure 6-23 is

assumed to stay between $\pm 0.7\left[\sqrt{N \cdot m \cdot g}/s \cdot kPa\right]$ for the fault-free system. The fault in the system is detected if the value of any of residuals violates the threshold.

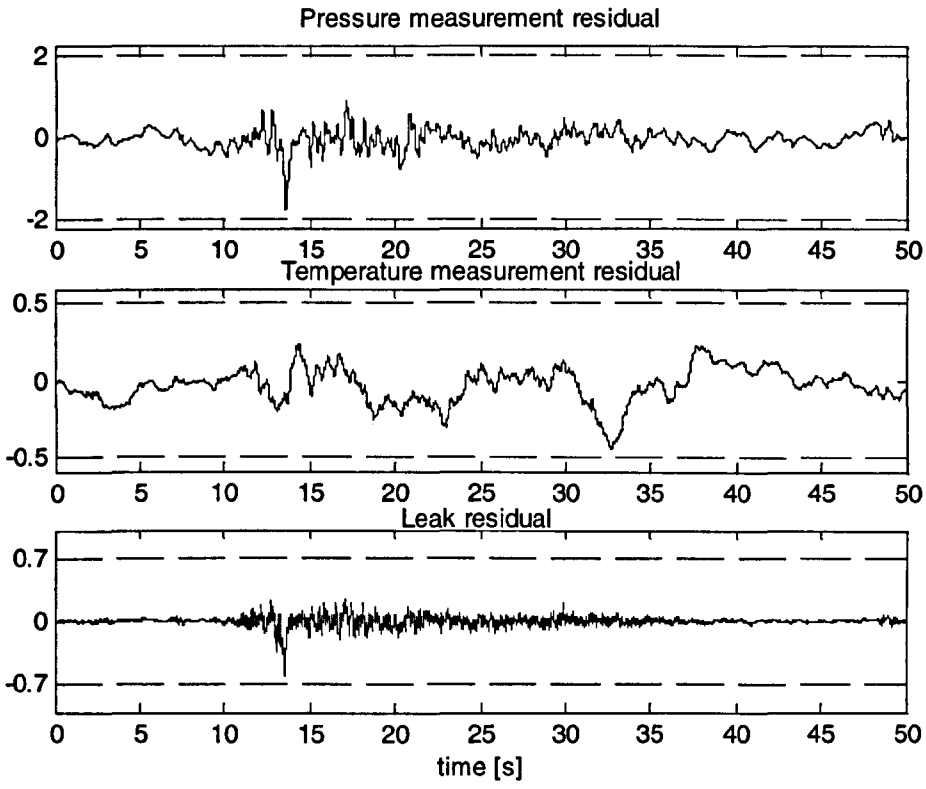

Figure 6-34: Fault-free residual trace for thresholds determination

6.4.3 Fault isolation and identification

The faults occurring in the system result in residuals violating the thresholds. The threshold violation indicates that the fault is present in the system but does not locate the fault. The fault isolation method based on the generalized observer scheme uses the table with 'fault signatures' (see section 2.4.4). The pressure sensor fault causes its dedicated residual to violate the threshold and the temperature sensor residual remains unchanged. The temperature sensor fault moves the residual over the threshold while the pressure sensor residual remains within the usual limits. The intake manifold leak results in the pressure sensor and temperature sensor directional residuals moving over the thresholds. The fault signatures Table 6.6 is extracted from the analysis of the residuals behaviour presented in section 6.4.1. Note that the leak

directional residual does not provide much of the information for the fault isolation system. The logical value 1 denotes that the threshold is violated, 0 denotes that the residual is within the limits. The 'X' value denotes that both 0 and 1 are possible to occur.

	Pressure sensor fault	Temperature sensor fault	Intake manifold leak
Pressure directional residual over threshold	1	0	1
Temperature directional residual over threshold	0	1	1
Leak directional residual over threshold	X	X	1

Table 6.6: Fault signatures table

The fault isolation is carried out with the assumption that only one fault occurs at the time. Due to the intake manifold dynamic nature the thresholds established for pressure, temperature and leak directional residuals are not violated simultaneously. The timing is also influenced by the system noise and the driving pattern, which interacts with the fault detection system. The time window that allows checking which thresholds are violated must be established. For the 90 degrees event based sampling and the allowed fault detection lag the number of events for the algorithm time window is established. For this work in order to achieve good robustness properties a 100 events window was chosen. This results in the delay of e.g. 1.5[s] for 1000[rpm] or 0.3[s] for 5000[rpm]. This delay may be reduced in some cases with the method that will be presented in sequel. An alternative robust method will rely on statistical analysis and a wide window is required to increase reliability.

Threshold-based method

The threshold-based fault isolation method relies upon the Table 6.6 and uses Boolean logic and tests for residuals. The logical test may be structured in a way that the fault isolation lag is reduced. The test must detect zeros that identify the type of the fault in Table 6.6. In case of the intake manifold leak all three residuals violate

the thresholds. If three threshold violations are detected the conclusion that the leak is present in the manifold is made. If within the permitted data analysis window either of residuals does reach the threshold by the end of the lag period, it is checked if it was the pressure or the temperature residual. This completes the fault isolation procedure. The simulation results of the fault detection and isolation are shown in sequel. The pressure sensor fault detection is shown in Figure 6-35. The fault in the system is detected when the pressure sensor directional residual moves below the threshold. Within 100 events from that moment the temperature sensor residual stays within the threshold range. This gives the signature of the pressure sensor fault. The temperature sensor fault detection is shown in Figure 6-36. In a similar way to the pressure sensor, the fault is first detected and finally isolated. The fault signature reveals the temperature sensor fault. The intake manifold leak detection is shown in Figure 6-37. The procedure is faster than for sensor faults. The event when the fault is isolated is determined by the moment when all directional residuals move over their thresholds.

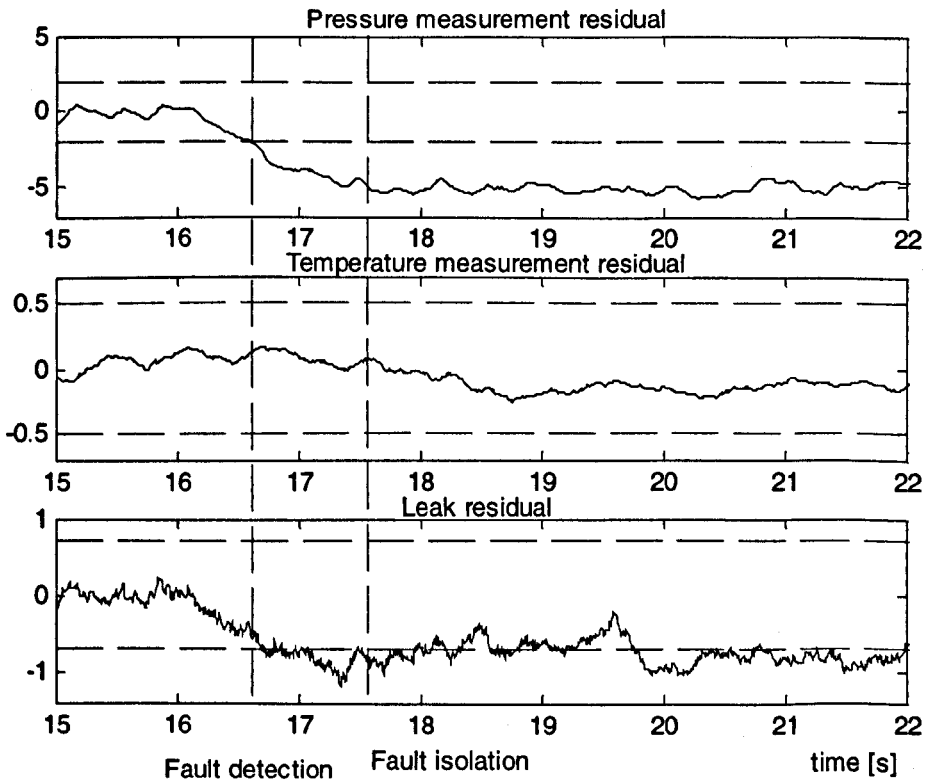

Figure 6-35: The pressure sensor fault isolation

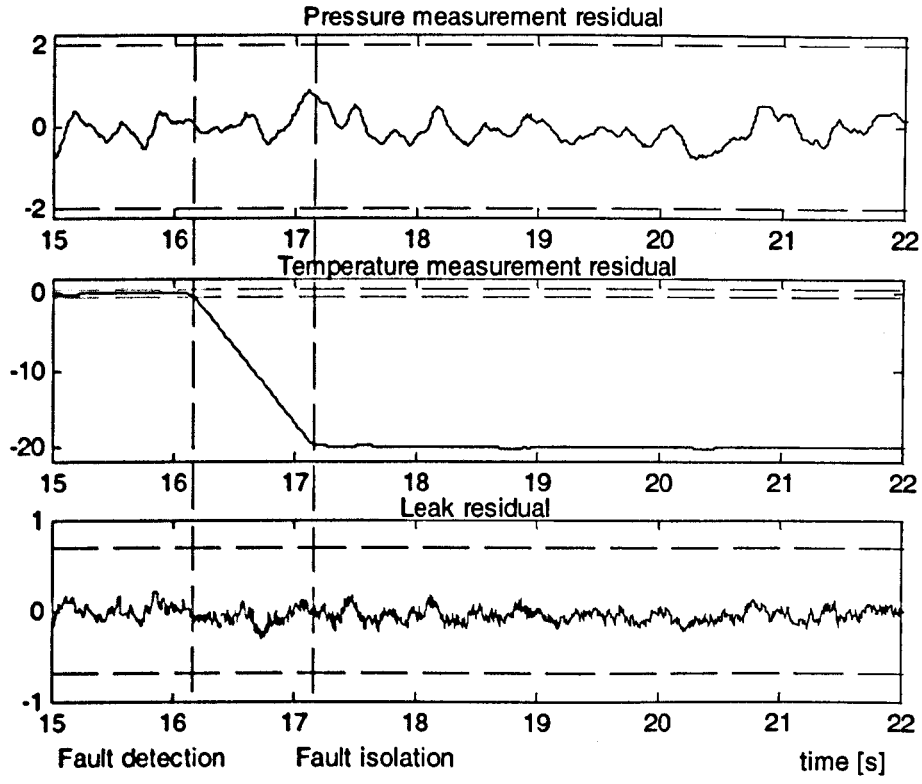

Figure 6-36: The temperature sensor fault isolation

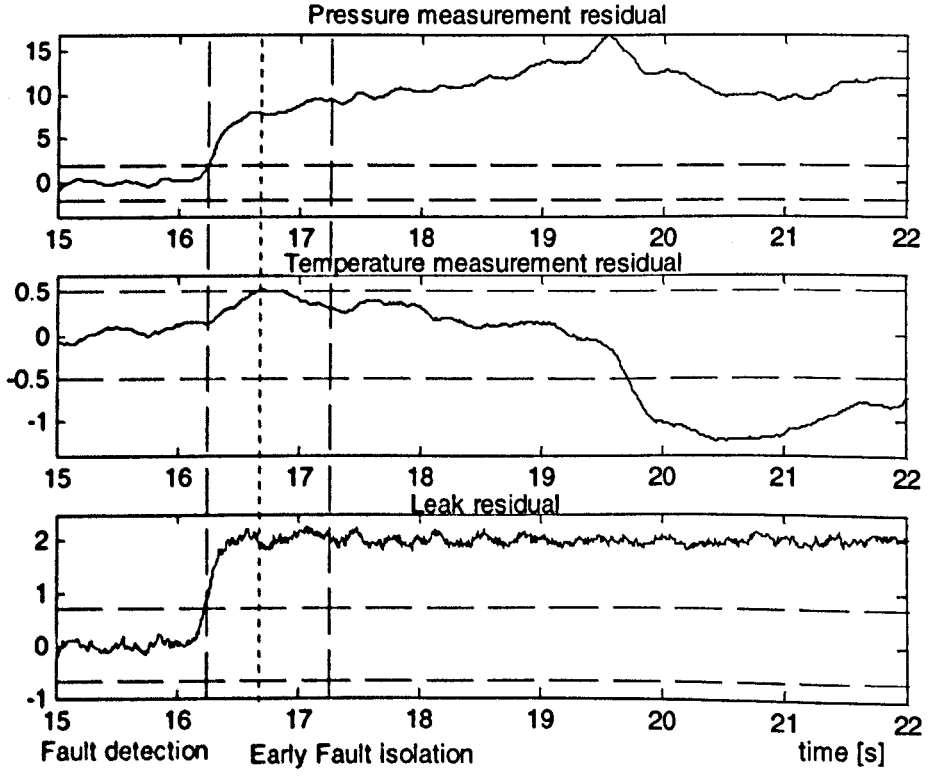

Figure 6-37: The intake manifold leak fault isolation

Threshold method with statistical and expert analysis

The fault isolation methods may be enhanced by incorporating system specific knowledge and statistical analysis. The system knowledge may be used with respect to the pressure sensor fault. The fault detection is triggered by the pressure sensor and leak directional residuals violating the thresholds. The leak residual violating the negative threshold immediately indicates that the pressure sensor fault had occurred. The leak, for naturally aspirated engine, can only be positive. If the leak directional residual is detected to be negative immediate fault isolation is possible. The pressure sensor fault isolation shown in Figure 6-35 may be completed much quicker (i.e. within 0.1 [s] after the fault detection).

To test the fault detection efficiency the following test was carried out. The pressure sensor fault $f_{P_{im},n}$ and the temperature fault $f_{T_{im,meas},n}$ are modelled as ramp signals that saturate after 1 sec. at the level of +5kPa and -20K respectively. Note that during this test the pressure sensor fault will be simulated with positive values to avoid trivial detection when the leak residual becomes negative. The intake manifold leak is assumed to appear in more abrupt way and is modelled as a ramp signal that saturates after 0.2 sec. at the level of $2\left[\sqrt{N \cdot m \cdot g}/s \cdot kPa\right]$. The fault detection speed is shown in Figure 6-38. The time required for the detection of pressure and temperature faults is determined by the time when residuals move over thresholds and the window width – 100 events. Unfortunately due to non-linearities the fault isolation algorithm fails to detect the leak for faults which start to ramp up at 15.1 and 20.1 second. This is due to fact that within the test window the temperature residual does not cross the threshold. Similar problems occurred during the intake manifold fault isolation presented in [102].

A more robust method presented now aims to improve the fault isolation accuracy by using the assumption that the leak area and the discharge coefficient of the leak source are constant. With this assumption the variance of the leak directional residual is monitored to help reliably isolate the fault. For the statistical analysis the number of data points determines the accuracy and the window length will be

increased to 200 events. Within the 200 events window from the moment of the fault detection (see in Figure 6-37) the temperature directional residual only marginally violates the threshold. If the temperature did not violate the threshold within the data window the pressure sensor fault would be isolated. Such a situation is shown in Figure 6-38 for the leak occurring at the time of 15.1 seconds. The pressure sensor fault is wrongly isolated. This clearly is a problem if control system re-configuration was to be used.

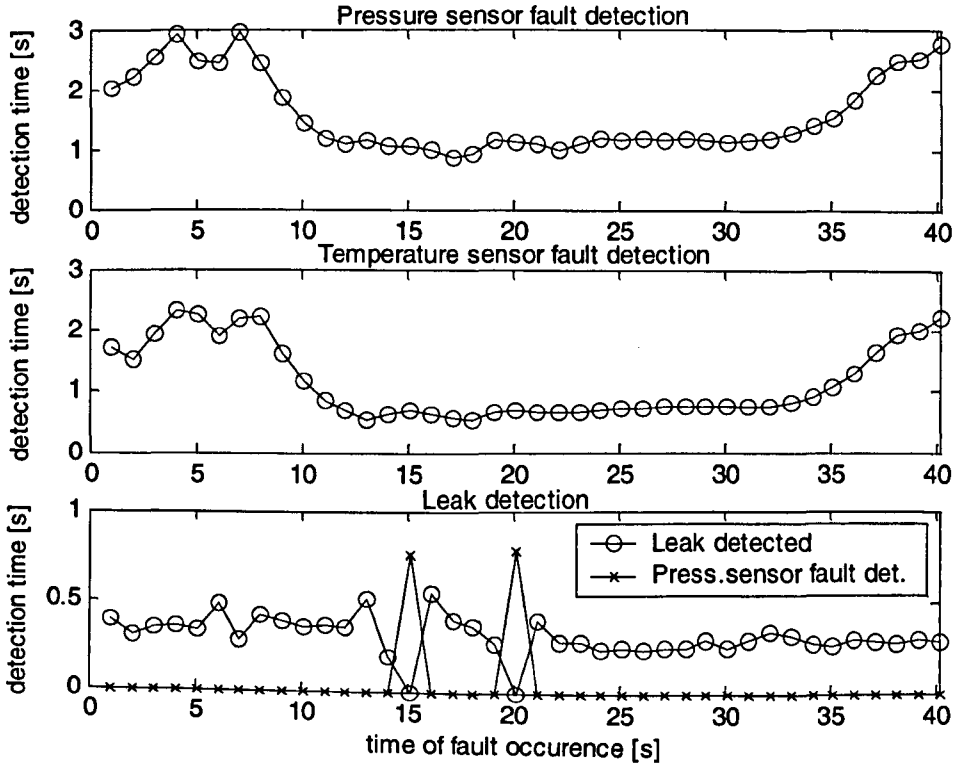

Figure 6-38: Fault detection efficiency: simple threshold method

The monitoring of the variance of the leak directional residual will improve the robustness. The leak directional residual that represents the leak area is assumed to be constant. For that reason it is possible to establish the upper bound of this parameter. The upper bound for the variance of the leak residual was determined based on the test simulation to be 0.04. If the variance of the leak remains below its maximum value, it indicates that there might be a leak in the intake manifold. The data window is shifted forward in time until the temperature residual violates the

threshold which indicates the leak, or until the variance violates its threshold, which indicates the pressure sensor fault.

	Pressure sensor fault	Temperature sensor fault	Intake manifold leak	Extend the data window
Pressure directional residual over threshold	1	0	1	1
Temperature directional residual over threshold	0	1	1	0
Leak directional residual over threshold	X	X	1	1
Leak directional residual variance over threshold	1	X	0	0

Table 6.7: Extended fault signatures table

The fault isolation logic is formulated in Table 6.7. The improved robustness of the method is presented in Figure 6-39. It should be noticed that the robustness of the method comes at the expense of the speed of isolation.

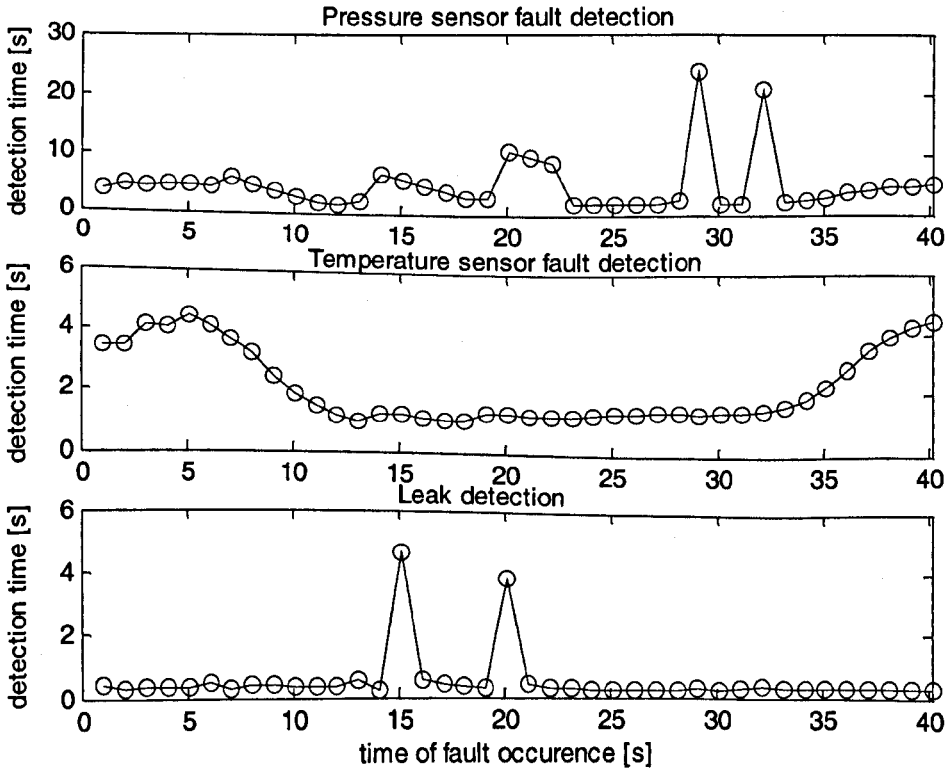

Figure 6-39: The intake manifold faults robust isolation

The accurate isolation is important for the control system re-configuration. For the re-configuration, the identification of the fault extent is essential. The fault identification is only required for the intake manifold leak. The directional residual estimate $\hat{f}_{leak,n}$ provided by the process fault detection filter is used for that purpose. The residual is a combination of the product of the leak discharge coefficient, the leak area estimate and the noise. Since the fault $f_{leak,n}$ is assumed to be constant and the noise is filtered out, the leak estimate may be used by the control algorithm. The system re-configuration will be demonstrated in the next section.

6.4.4 System reconfiguration and the cylinder air charge estimation

The system estimation with the state dependent Kalman filter is combined now with the fault detection, isolation and the identification algorithm. The system diagram with the fault detection functional block and the estimator is shown in Figure 6-40. The state dependent Kalman filter is re-configured in a way that if pressure or temperature sensor faults are detected the faulty sensor is excluded from the system. In case of the intake manifold leak, state estimates obtained from the intake leak dedicated filter are used by the controller.

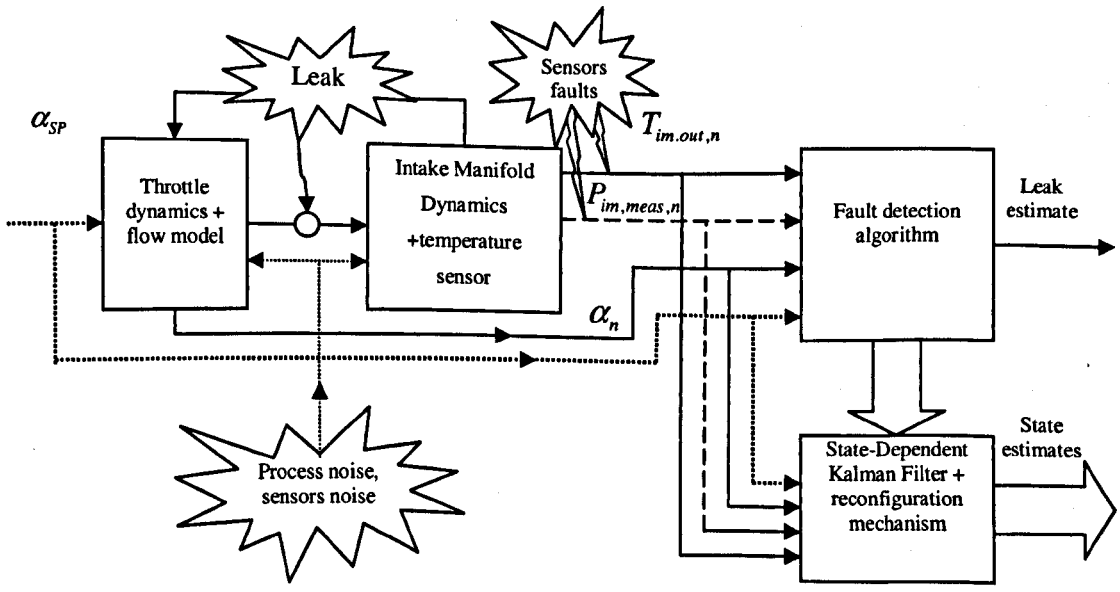

Figure 6-40: Fault detection combined with the state estimator

The system re-configuration results in the improved control when faults are present in the system. A more accurate cylinder air charge prediction without a bias is achieved. The in-cylinder air-fuel ratio comparison for the system with and without the re-configuration is presented in Figure 6-41.

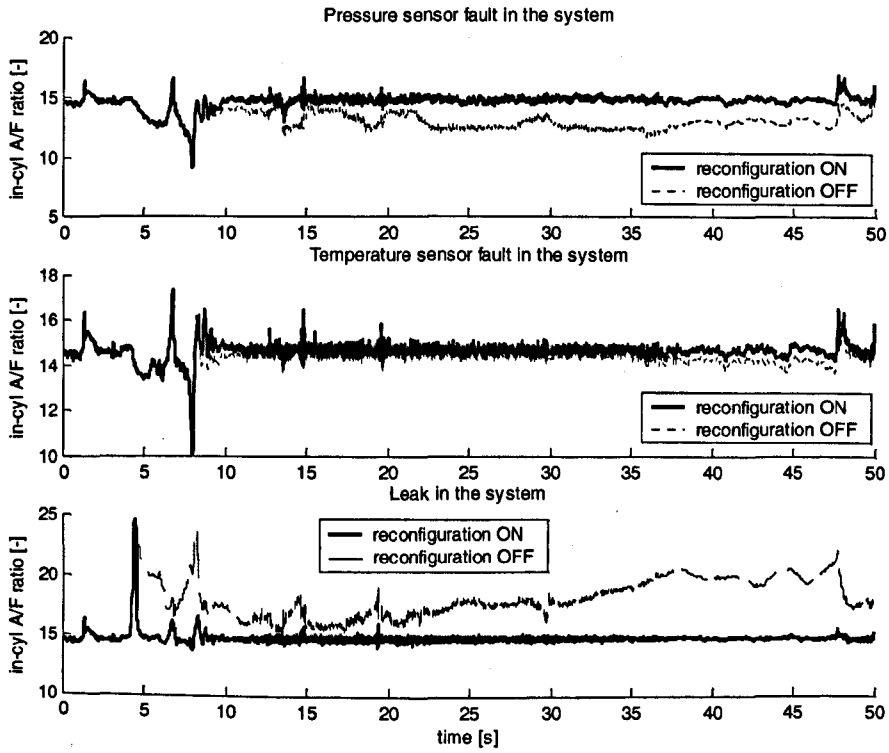

Figure 6-41: CAC prediction error reduction with the system re-configuration

The fault detection, isolation and identification combined with the control system re-configuration provides improved accuracy of the air-fuel ratio control. As a result bias-free estimates of parameters are obtained, the cylinder air charge prediction is more accurate and the feedforward control, based on the fuel-film dynamics model inversion, is improved. The control system provides good control performance in presence of faults. The system reconfiguration is intended to maintain acceptable engine operation to reach the service station. However, if the re-configured control system was capable of maintaining legislated emission levels, the fault could be fixed during the scheduled servicing. The engine management system

could give a fault indication to the driver to let him know that the engine experiences technical difficulties. In the meantime the engine control system would allow a continuation of the journey.

6.5 Summary

A simulation analysis of the accuracy of the cylinder air charge (CAC) prediction was carried out. Robustness was assessed using two different approaches. The first approach involved the feeding of the process and the measurement stochastic noise into the simulated intake manifold model. The improvement in the CAC prediction accuracy was found to be significant. The extended Kalman filter (EKF) provided the best performance. The state-dependent Kalman filter (SDKF) was only slightly worse in terms of the integrated squared and the absolute prediction errors. The second approach introduced the parameter and the measurement gain errors. Monte-Carlo analysis based on a population (2000 with truncated Gaussian probability distribution) of simulations was carried out in order to obtain the CAC error distributions. The EKF and SDKF methods once again provided superior performance compared to direct measurement methods. The SDKF provided the best robustness to errors introduced in the simulation.

Fault detection schemes such as presented later in the chapter are a very important part of the modern control algorithms. For the intake manifold presented solution delivers an unbiased estimation, if combined with the re-configuration. The fault detection and isolation method used in this chapter performed threshold tests for directional residuals. The fault detection was followed by the isolation. To increase robustness, the fault detection signal triggered the fault signature diagnosis algorithm. The set of measurements over the period of time was gathered and the properties of residuals were analysed. The fault isolation may be improved by the introduction of residual statistical analysis. The variance of the leak residual signal was tested. Additionally system specific knowledge was incorporated to improve the robustness of the fault detection.

Chapter 7

Conclusions and Future work

7.1 Summary and conclusion

The thesis presented a number of areas of control engineering theory and practice applied to the spark ignition engine. The motivation for that is a multidisciplinary nature of control engineering. The control engineer must be able to understand the process being subject of control, be able to model it, design the control algorithm and implement it.

In Chapter 2 the basic theory for estimation and more advanced fault detection algorithms were elaborated. The chapter introduced parameter and state estimation methods for linear systems. The theoretical background was briefly presented to give a basis for the non-linear techniques analysis. This led to the estimation methods for the non-linear systems, out of which the extended Kalman filter is probably the most popular. An alternative formulation of the non-linear estimation method is introduced next in a form of the state-dependent Kalman filter. A discrete time formulation of the state-dependent Kalman filter was introduced. This was a basis for the fault detection algorithms development. The dedicated observer scheme for sensor fault detection was outlined next. This was followed by the development of the non-linear fault detection filter based on state-dependent models. The fault detection filter can isolate, to no higher than the output measurements, the number of faults/unmeasured inputs. Two types of residuals are generated by the filter: the orthogonal residual vector decoupled from faults (or

disturbances) and the fault/unmeasured input estimate. The residuals may be used for the fault detection, the isolation and even the identification. For plants with both: process and sensor faults appearing in the system, the methodology for simultaneous detection of both types of faults is described.

In Chapter 3 the design methods for non-linear control algorithms were presented. The algorithms were based on state-dependent models. Two distinct design approaches were analyzed. First the well-known linear-quadratic optimal control theory was used as a basis and its applicability was extended to non-linear systems. The development was based on the state-dependent Riccati equation method and improved its performance with the linear time-varying approximation of the non-linear system. The optimality of the solution was also a subject of consideration and the algorithm that provides the optimal solution to the cost function minimization problem was introduced. The second group of algorithms used the generalized predictive control algorithm as a basis and extended this control technique to non-linear systems using state-dependent models. The time-varying approximation was used together with the predicted control trajectory computed in previous iteration of the predictive control algorithm. The method of recovering the optimal solution of the minimisation problem was presented in the predictive control context. Depending on control system requirements the basic non-linear or their optimal versions may be used. For systems where the cost function and weighting matrices are introduced as tuning parameters, the exact value of the performance index may not be of interest. For some applications where the cost function determines the energy used in the process, achieving the minimum value may be of interest. In such cases, the optimal versions of presented algorithms should be used.

The developments described above are of a theoretical nature. This work was intended to be a preparation for the subsequent application effort.

In Chapter 4 the identification of the combustion engine model was presented. The methodology presented there aimed to use as few parameters supplied by the engine manufacturer as possible. Most importantly, the identification

procedure used the driving cycle data which implies that presented methods may be adopted for the on-line operation. This in turn allows the model adaptation to the gradual changes of the system as a result of wear and tear. The use of the driving cycle data required the specific identification procedure to be developed. The identification task was split in two sub-tasks. The intake manifold with the throttle model was identified first. The exhaust and fuel delivery time delays were then removed from the data and fuel delivery, torque and air fuel ratio models identified. The model complexity determines its accuracy and various model structures were employed and validated.

The engine model was utilized in Chapter 5. A feedforward fuel control algorithm with a built-in predictor was developed and tested in the vehicle using the rapid prototyping hardware. Test results indicated that good performance had been achieved. This also provided an additional test for the identified model accuracy. In addition, by using the throttle actuator model and the throttle setpoint command, more accurate future throttle position is predicted and, hence, more accurate cylinder air charge estimated. The estimated cylinder air charge is used by the controller and more accurate fuel injection command is computed. Further improvements in the air-fuel ratio control are achieved by multi-input multi-output (MIMO) nonlinear control strategies which were tested in simulation. Models of the exhaust gas air-fuel ratio and generated torque are used for the design of the control system. The MIMO control strategy is based on the predictive control approach. The multivariable solution is a trade-off between the accuracy in the regulation of air-fuel ratio and the tracking of the requested torque profile. Considering the complexity of the non-linear engine model, some simplifications based on the specific nature of the problem are made to reduce the amount of computational effort required. Improvements in the air fuel ratio regulation are achieved by decoupling of the accelerator pedal and the throttle opening through the introduction of a simple delay in the application of the throttle command. The length of the time delay may be optimized to allow for the maximum permissible air-fuel ratio excursion during transient operations. Additional improvements in transient torque control are achieved through the magnitude decoupling of the throttle movement and the accelerator pedal position as well. The

compensation of the cylinder air charge response dynamics using simple classical control methods is proposed. The attention is then turned to the multivariable predictive air-fuel ratio and torque control methods. This provides a suitable solution for the torque tracking problem. The predictive method is based on a minimization of the cost function defined in terms of the cylinder air charge only. The amount of fuel injection is then derived from future cylinder air charge prediction that is available through the predictive control algorithm. By changing the weights (i.e. tuning) and the prediction horizon, the required performance of lambda regulation and torque tracking can be influenced simultaneously. Presented algorithms provide significant improvements in the control system performance.

In Chapter 6 the robustness to the modelling inaccuracies, noise and faults are analysed. The simulation analysis of the accuracy of the cylinder air charge (CAC) prediction was carried out. Robustness was assessed using two different approaches. The first approach involved feeding of the process and the measurement stochastic noise into the simulated intake manifold model. The improvement in the CAC prediction accuracy was substantial. The presented non-linear filtering methods were capable of providing a better performance. The second approach benchmarked robustness of the control system to parameter and measurement errors. Monte-Carlo analysis based on a population of simulations was carried out in order to obtain the CAC prediction error distributions. The non-linear filtering algorithms once again provided superior performance compared to direct measurement methods. The fault detection scheme that was presented later in Chapter 6 is a very important part of the modern control algorithms. The legislation requires the engine control system to be equipped with the diagnosis of parts that determine emissions. The fault detection scheme developed for the intake manifold delivered unbiased estimation, if combined with the system re-configuration. The fault detection and isolation method presented used threshold tests for residuals. To increase robustness, the fault detection signal triggered the fault signature diagnosis algorithm. The set of measurements over the period of time was gathered and properties of residuals were analysed. The fault isolation robustness was improved by introduction of the residual

statistical analysis and the system specific knowledge. The control system re-configuration was a natural consequence of the fault isolation.

The range of methods presented in this thesis aims to address most of the control problems associated with the spark ignition engine. The emphasis was put on both theoretical research and the application with a slight shift towards application. The presented work will not provide a solution to all problems encountered during the engine control system development. However it gives a good insight in the application and shows how the model-based techniques may be employed to reduce the development time and effort required.

7.2 Future work

The theoretical and application research results presented in this thesis may be further developed. Especially, when the engine control application is considered there are number of issues that may and should be researched more thoroughly. The results presented in this thesis should be considered as an indication of what are the possible directions in modelling, control, estimation and fault detection methods development. It must be stressed that the developments presented here are not a complete solution for the presented application. The following directions for the future research should be considered:

- Theoretical analysis of non-linear system structures and the optimal state-dependent model parameterization for these systems. It is known that the performance of state-dependent control algorithms depends strongly upon the system parameterization. The guidelines for the model parameterization should be a subject of research
- Stability analysis and real-time implementation of the presented algorithms will be analysed. Rigorous analysis of the stability for non-linear systems is a difficult issue. It may not be possible to formulate stability conditions for general non-linear system structures. However, the stability for a class of

non-linear systems may be considered and rigorous analysis methods developed.

- The engine identification presented in this thesis may be extended to the cold-start conditions. Additional thermal parameters must be added and the estimation carried out. Also, alternative model structures for fuel-film dynamics may be employed to achieve improved accuracy. A more accurate non-linear model of the throttle actuator and an improved torque model will be introduced
- The real-time implementation and tests of multivariable engine control strategies should be attempted. This will require consideration of the computational burden and some simplifications resulting in scheduled control strategies. Also, the optimization based multivariable control strategy with the efficient optimization algorithm will be analyzed
- The implementation of the estimation methods will require careful consideration of the computational burden. This will be a preparation for the real-time implementation of the presented model-based intake manifold filtering methods. The estimation for the full system with the fuelling path should also be subject of research. More comprehensive robustness analysis for the full system should be carried out.
- The estimation and fault detection applied to the intake manifold should be extended to the full engine. The important parameter of interest is the estimate of the wall fuel mass. The solution that is expected to give the best results leads to a separation of tasks between the intake-manifold and wall-fuel estimators and follows the methodology used for the system identification. The main difficulty of considered filtering problem results from nonlinearities and variable time delays in the system.
- The application of the hybrid system control theory [105], [106](i.e. mixed logical and dynamic systems) to the engine control is expected to provide good results. The combination of the logic based controls with other methods, widely used on the engine, provides the motivation. The other feature of hybrid systems that is found in combustion engines is the natural

combination of continuous and discrete event system behaviour. Hybrid systems theory addresses these issues for both: simulation and control.

References

[1] Taylor, J. H., (2001). Modelling & Simulation of Dynamic Systems - a Tutorial, Proc. IMACS/IFAC Fourth International Symposium on Mathematical Modelling and Simulation in Agricultural and Bio-Industries, Haifa, Israel

[2] Wellstead, P. E. (1979). Introduction to Physical System Modelling, Academic Press, London

[3] Ljung, L., (1999). System Identification - Theory for the User, Second edition, Prentice Hall, Saddle River, NJ,.

[4] Soderstrom, T., Stoica, P., (1989). System Identification, Prentice Hall, Hertfordshire

[5] Oussar Y, Dreyfus G., (2001). How to be a gray box: dynamic semi-physical modelling, Neural Netw., vol. 14, no. 9, pp. 1161-1172

[6] Tan, K. C., Li, Y., Gawthrop, P. J. Glidle, A., (1997). Evolutionary Grey-Box Modelling for Nonlinear Systems, University of Glasgow Technical Report CSC-96019

[7] Friedman, J. H., (1991). Multivariate adaptive regression splines, The Annals of Statistics, vol. 19, no. 1, pp. 1-67.

[8] Liu, G. P., (2001). Nonlinear Identification and Control – A neural network Approach, Springer-Verlag, London

[9] Saez, D., Cipriano, A., Ordys, A. W., (2002). Optimization of industrial processes at supervisory level – Application to Control of Thermal Power Plants, Springer-Verlag, London

[10] Giovanini, L and Dutka, A., (2003), Fault Isolation Filter for Nonlinear Systems, IFAC Safeprocess 2003, Washington, USA

[11] Dutka, A. S., Grimble, M. J., (2004). State-Dependent Riccati Equation Control with Predicted Trajectory, Proceedings of the American Control Conference, Boston

[12] Dutka, A. S., A. W. Ordys, Grimble, M. J., (2005). Optimized discrete-time State Dependent Riccati Equation regulator, Proceedings of the American Control Conference, Portland

[13] Grimble, M. J., A. W. Ordys, (2001). Non-linear Predictive Control for Manufacturing and Robotic Applications, IEEE Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland

[14] Ordys, A. W. and M. J. Grimble, (2001). Predictive control design for systems with the state-dependent non-linearities, SIAM Conference on Control and its Applications, San Diego, California

[15] Dutka, A. S., Ordys, A. W., Grimble, M. J. (2003). Non-linear Predictive Control of 2 dof helicopter model, IEEE CDC proceedings

[16] Dutka, A. S., Ordys, A. W. (2004). The Optimal Nonlinear Generalized Predictive Control By The Time-Varying Approximation, Proceedings of IEEE International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland

[17] Hendricks, E., Jensen, M., Chevalier, A., Vesterholm, T., Problems in Event Based Engine Control, Proceedings of the American Control Conference, Baltimore, June 1994

[18] Kamen, E. W., Su, J. K., (1999). Introduction to Optimal Estimation, Springer-Verlag, London

[19] Grimble, M. J., Johnson, M. A., (1988). Optimal Control and Stochastic Estimation, Theory and Applications, John Wiley & Sons, Chichester

[20] Kalman, R. E., (1960). A new approach to linear filtering and prediction problems, Journal of Basic Engineering, 82, March, pp. 35-45

[21] Chui, C. K., Chen, G., (1991). Kalman Filtering with Real-Time Applications, Springer-Verlag, Berlin Heidelberg

[22] Maybeck, P. S., (1982). Stochastic Models, Estimation and Control, Academic Press, New York

[23] Mracek, C. P., Cloutier, J. R. (1998). Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method, Int. J. of Robust and Nonlinear Control, Vol. 8, pp. 401-433

[24] Cloutier, J. R., Stansbery, D. T., (2002). The Capabilities and Art of State-Dependent Riccati Equation-Based Design, Proceedings of the American Control Conference, Anchorage, AK, USA

[25] Mracek, C. P., Cloutier, J. R., D'Souza, C. A., (1996). A New Technique for Nonlinear Estimation, Proceedings of the 1996 IEEE International Conference on Control Applications, Dearborn, MI, pp. 338-343

[26] Gertler, J. (1988). Survey of model-based failure detection and isolation in complex plants, IEEE Control Systems magazine, Vol. 8, pp.3-11

[27] Frank, P. (1990). Fault diagnosis in dynamic systems using analytical and knowledge based redundancy - a survey and some new results, Automatica, Vol.26, pp.459-474

[28] Frank, P. (1991). Enhancement of robustness in observer-based fault detection, IFAC/IMACS Safeprocess Conference, Baden-Baden, Germany, pp.275-287

[29] Gertler, J. (1991). Analytical redundancy methods in fault detection and isolation, IFAC/IMACS Safeprocess Conference, Baden-Baden, Germany, pp. 9-21.

[30] Gertler, J. (1995). Generating directional residuals with dynamic parity relations, Automatica, Vol. 31, pp. 627-635

[31] Patton, R. J., Chen, J., (1991). A review of parity space approaches to fault diagnosis, IFAC/IMACS Safeprocess Conference, Baden-Baden, Germany

[32] Chen, J. and Patton R. J., (1999). Robust Model-Based Fault Diagnosis for Dynamic Systems, Kluwer Academic

[33] Gertler, J., (1998). Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, New York

[34] Beard, R. (1971). Failure accommodation in linear systems through self-reorganization, Ph.D. dissertation, Cambridge, Massachusetts, USA

[35] Jones, H. L. (1973). Failure detection in linear systems, PhD dissertation, Cambridge, Massachusetts, USA

[36] Massoumnia, M. (1986). A geometric approach to the synthesis of failure detection filters, IEEE Trans. on Automatic Control, Vol. 31, pp. 839-846

[37] White, J. E. and J. L. Speyer (1987). Detection filter desire: Spectral theory and algorithms, IEEE Trans. Autom. Control, Vol. 32, pp. 593-603

[38] Park, J. and G. Rizzoni (1994). A new interpretation of the fault detection filter: Part 1, Int. J. Control, Vol. 60, pp. 767-787

[39] Liu, B. and J. Si (1997). Fault isolation filter design for time-invariant systems, IEEE Trans. Autom. Control, Vol. 21, pp. 704-707

[40] Keller, J.-Y. (1999). Fault isolation filter design for linear stochastic systems, Automatica, Vol. 35, pp. 1701-1706

[41] Clark, R. N., (1989). Fault Diagnosis in Dynamic Systems: Theory and Application, Prentice Hall, New York

[42] Patton, R. J., Frank, P. M., Clark, R. N., editors (1989). Fault Detection in Dynamic Systems, Theory and Application. Control Engineering Series, Prentice Hall, New York

[43] Simani, S., Fantuzzi, C., Patton, J. R., (2002). Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques, Springer-Verlag London

[44] Bendat, J. S., (1998). Nonlinear System Techniques and Applications, Wiley-Interscience, New York

[45] Isidori, A., (1995). Non-linear control systems, 3rd Edition, Springer Verlag, Berlin

[46] Kwakernaak, H. and Sivan, R., (1972). Linear optimal control systems, John Wiley

[47] Pearson, J. D., (1962). Approximation Methods in Optimal Control", Journal of Electronics and Control, vol. 13, pp. 453-465

[48] Burghart, J. H., (1969). A Technique for Suboptimal Feedback Control of Nonlinear Systems, IEEE Transactions on Automatic Control, October, pp.530-533

[49] Cloutier, J.R, D'Souza, C.N. and Mracek, C.P., (1996). Nonlinear regulation and Nonlinear H-infinity Control via the State-Dependent Riccati Equation Technique: Part 1, Theory, Part 2, Examples, Proceedings of the 1st International Conference on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL, USA, pp.117-141,

[50] Cloutier, J. R., (1997). State-Dependent Riccati Equation Techniques: An Overview, Proceedings of the American Control Conference, pp.932-936

[51] D'Angelo, H., (1970). Linear time-varying systems: analysis and synthesis, Allyn and Bacon, Boston

[52] Erdem, E. B., Alleyne, A. G., (2002). Estimation of Stability Regions of SDRE Controlled Systems Using Vector Norms, Proceedings of the American Control Conference, Anchorage 2002, 80-85

[53] Curtis, J. W., Beard, R. W., (2002). Ensuring Stability of State-dependent Riccati Equation Controllers Via Satisficing, Proceedings of the 41st IEEE Conference on Decision and Control, 2645-2650

[54] Youseff, M., A., Nonlinear Predictive Flight Control System Design, PhD Dissertation, University of Strathclyde, Glasgow, UK

[55] Seiler, P., (2003). Stability Region Estimates for SDRE Controlled Systems Using Sum of Squares Optimization, Proceedings of the American Control Conference, Denver, 1867-1872

[56] Anderson, B., and Moore, J., (1971). Linear optimal control, Prentice Hall, Englewood Cliffs

[57] Mohler, R. R., (1991). Nonlinear Systems, Vol. 1, Dynamics and Control, Prentice Hall

[58] Cloutier, J. R., Stansbery, D. T., (2002). The Capabilities and Art of State-Dependent Riccati Equation-Based Design, Proceedings of the American Control Conference, Anchorage, 86-91

[59] Grimble, M. J., (2001). Industrial control systems design, John Wiley, Chichester

[60] Lewis, F. L., (1986). Optimal Control, John Wiley New York

[61] Michalska, H., Mayne D. Q. (1993). Robust Receding Horizon Control of Constrained Nonlinear Systems, IEEE Transactions on Automatic Control, vol. 38, pp. 1623-1633

[62] Huang, Y., Lu, W-M., (1996). Nonlinear Optimal Control: Alternatives to Hamilton-Jacobi Equation, Proceedings of the 35th IEEE Conference on Decision and Control, pp.3942-3947

[63] Chen, H., Allgower, F. (1998). A quasi-infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability, Automatica, Vol. 34, pp.1205-1217

[64] Atherton, D. P., (1975). Non-linear Control Engineering, Van Nostrand Reinhold, London

[65] Richalet, J. A. Rault, Testud J. L., Papon J., (1978). Model predictive heuristic control applications to industrial processes, Automatica, Vol. 14, pp. 414-428

[66] Richalet, J. Industrial applications of the model based predictive control, Automatica, Vol. 29, 1993, No. 8, pp. 1251-1274.

[67] Cutler, C. R. and Ramarker, B. L., (1980). Dynamic matrix control – A computer control algorithm, Proceedings JACC, San Francisco.

[68] Garcia, C. E. (1984). Quadratic/Dynamic Matrix Control of Nonlinear Processes: An Application to a Batch Reaction Process, AIChE Annual Meeting, San Francisco, CA

[69] Clarke, D. W., Montadi, C. and Tuffs, P. S., (1987). Generalized predictive control – Part 1, The basic algorithm, Part 2, Extensions and interpretations, Automatica, vol. 23, 2, 137-148.

[70] Clarke, D. W., and Mounted, C., (1989). Properties of generalized predictive control, Automatica, vol. 25, 6, pp. 859-875.

[71] Tomizuka, M., Rosenthal, D. E., (1979). On the optimal digital state vector feedback controller with integral and preview actions, Transactions of the ASME, 101, pp. 172-178.

[72] Bitmead R. R., Givers M. and Hertz, V., (1989). Optimal control redesign of generalized predictive control, IFAC Symposium on Adaptive Systems in Control and Signal Processing, Glasgow, UK.

[73] Ordys, A. W. and Clarke, D. W., (1993). A state-space description for GPC controllers, Int. J. Systems Sci., vol. 24, 9. pp. 1727-1744

[74] Mosca, E. and J. Zhang, (1992). Stable redesign of predictive control. Automatica, Vol. 28, No. 6, , pp.1229-1233

[75] Kouvaritakis, B. J. A. Rossiter, A. O. T. Chang, (1992). Stable generalized predictive control: an algorithm with guaranteed stability, IEEE Proceedings-D, Vol 139, No. 4, pp. 349-362

[76] Kothare S. L. and M. Morari, (2000). Contractive Model Predictive Control for Constrained Nonlinear Systems", IEEE Transactions on Automatic Control, vol. 45, no. 6, , pp. 1053-1071

[77] Kouvaritakis, B., Cannon, M., Rossiter, J. A., (1999). Non-linear model based predictive control, Int. J. Control, 72, 10, 919-928

[78] Youssef A., Grimble, M.J., Ordys, A.W., Dutka A.S., Anderson, D., (2003). Robust Nonlinear Predictive Flight Control, European Control Conference, Cambridge, UK, September 2003

[79] Lee, Y.I., Kouvaritakis, B. and Cannon, M., (1999). Constrained receding horizon predictive control for nonlinear systems, Automatica, Vol. 38, No. 12, 2002, pp.2093-2102

[80] Bellman, R., (1957). Dynamic programming, Princeton University Press., Princeton, N.J.

[81] Dutka, A., Javaherian, H., (2005). Data Driven Engine Model Identification, General Motors Internal Report

[82] Hendricks, E., Sorenson, S. C., (1990). Mean value modelling of spark ignition engines, SAE paper 9006161

[83] Moskwa, J. J., Hedrick, J. K., (1992). Modelling and validation of spark ignition engines, ASME Journal Dynamics Systems, Measurement and Control, vol. 114, pp. 278-285

[84] Kiencke, U., Nielsen, L., (2000). Automotive Control Systems, Springer-Verlag Berlin Heidenberg

[85] Scattolini, R., De Nicolao, G., Cittadini, M., Rossi, C., Siviero, C., (1997). Modelling internal combustion engines via identification techniques, Proceedings of 5th IEEE Mediterranean Conference on Control and Systems, Paphos, Cyprus

[86] Jones, V. K., Ault, B. A., Franklin, G. F., Powell, J. D., (1995). Identification and Air-Fuel Ratio Control of a Spark Ignition Engine, IEEE Transactions on Control Systems Technology, vol. 3, no. 1, pp 14-21

[87] Chank, C.-F., Fekete, N. P., Amstutz, A., Powell, J. D., (1995). Air-Fuel Ration Control in Spark-Ignition Engines Using Estimation Theory, IEEE Transactions on Control Systems Technology, vol. 3, no. 1, pp 22-31

[88] Hendricks E., Jensen, M., Chevalier, A., Vesterholm, T., (1994). Problems in Event Based Engine Control, Proceedings of the American Control Conference, Baltimore, Maryland, USA

[89] Dutka, A., (2004). Report on Sampling Rate Optimization, General Motors project deliverable

[90] Moskwa, J. J., (1988). Automotive Engine Modelling for Real Time Control, MIT PhD Thesis

[91] Chevalier, A. C. Winge Vigild, (2000). Predicting the Port Air Mass Flow of SI Engines in Air/Fuel Ratio Control Applications, SAE 2000 World Congress, Detroit, Michigan

[92] Chevalier, A., Muller, M., Hendricks, E. (2000). On validity of Mean Value Engine Models During Transient Operation, SAE paper, 2000-01-1261

[93] Li, L., Caldwell, G. E., (1998). Identification of phase shift using cross correlation, Presented at NACOB 98: North American Congress on Biomechanics University of Waterloo, Ontario, Canada, August 14-18, http://asb-biomech.org/onlineabs/NACOB98/177/

[94] Chatfield, C., (1975). The Analysis of Time Series: Theory and Practice, Chapman and Hall, London

[95] Aquino, C. F., (1981). Transient A/F Control Characteristics of the 5 liter Central Fuel Injection Engine, SAE Technical Paper, no. 810494

[96] G. M. van der Molen, General Motors Report on Performance and Robustness of A/F Control against Parameter Variations Component Models and Monte-Carlo Simulation Results, ISC Report, 2004

[97] Kimmich, F., Schwarte, A., Isermann, R., (2005). Fault detection for modern Diesel engines using signal- and process model-based methods, Control Engineering Practice, vol. 13, pp. 189-203

[98] Cho, D., Paolella, P., (1990). Model-based failure detection and isolation of automotive powertrain systems, Proceedings of the American Control Conference, San Diego, California, USA

[99] Rizzoni, G., Azzoni, P. M., Minelli, G., (1993). On-board diagnosis of emission control system malfunctions in electronically controlled spark ignition engines, Proceedings of the American Control Conference, San Francisco, USA

[100] Gertler, J., Costin, M., Fang, X., Hira, Z., (1995). Model based diagnosis for automotive engines – algorithm development and testing on a production vehicle, IEEE Transactions on Control Systems Technology, vol. 3, no. 1, pp 61-69

[101] Nyberg, M., (1999). Model Based Fault Diagnosis: Methods, Theory, and Automotive Engine Applications, Linköping University, PhD thesis No. 591

[102] Nyberg, M., (2002). Model-based diagnosis of an automotive engine using several types of fault models, IEEE Transaction on Control Systems Technology, 10, 5, 679--689

[103] Weeks W. R., Moskwa J. J. (1995) Transient Air Flow Rate Estimation in a Natural Gas Engine Using a Nonlinear Observer, SAE Technical Paper 940759

[104] McKay, D., Nichols, G., Schreurs, B., (2000), "Delphi Electronic Throttle Control Systems for Model Year 2000; Driver Features, System Security, and OEM

Benefits. ETC for the Mass Market" SAE 2000 World Congress, Detroit, Michigan, March 6-9

[105] Lygeros, J., (1996). Hierarchical, Hybrid Control of Large Scale Systems. PhD thesis, University of California, Berkeley, California

[106] Baotic, M., Vasak, M., Morari, M., Peric, N., (2003). Proceedings of the American Control Conference, Denver, Colorado, June