UNIVERSITY OF
LIVERPOOL

Optimal Test Signal Design and Estimation for Dynamic Powertrain Calibration and Control

Thesis submitted in accordance with the requirements of the
University *of* Liverpool for the
Degree of Doctor in Philosophy

by

Ke Fang

November 2012

Statement of Originality

This thesis is submitted for the degree of Doctor in Philosophy in the Faculty of Engineering at the University of Liverpool. The research project reported herein was carried out, unless otherwise stated, by the author in the Department of Engineering at the University of Liverpool between October 2008 and October 2011.

No part of this thesis has been submitted in support of an application for a degree or qualification of this or any other University or educational establishment. However, some parts of this thesis have been published, or submitted for publication, in the following papers:

- K. Fang and A.T. Shenton, Optimal Input Design for Dynamic Engine Mapping, 6th IFAC Symposium Advances in Automotive Control, Munich, Germany, 2010
- K. Fang and A.T. Shenton, Optimal Test Signal Design for the Identification of Dynamic Engine Model, 1st International Conference on Powertrain Modelling and Control, Bradford, UK, 2012
- A.T. Shenton and K. Fang, IC Engine Dynamic Calibration and Control by Inverse Optimal Behaviours, 1st International Conference on Powertrain Modelling and Control, Bradford, UK, 2012
- K. Fang and A.T. Shenton, Optimal Input Design for Dynamic Model Prediction Accuracy, 15th IFAC Workshop on Control Applications of Optimization, Rimini, Italy, 2012
- K. Fang and A.T. Shenton, Constrained Optimal Test Signal Design for Improved Prediction Error, *submitted* to IEEE Transaction on Automation Science and Engineering, October, 2012

Ke Fang
30th November 2012

Abstract

With the dramatic development of the automotive industry and global economy, the motor vehicle has become an indispensable part of daily life. Because of the intensive competition, vehicle manufacturers are investing a large amount of money and time on research in improving the vehicle performance, reducing fuel consumption and meeting the legislative requirement of environmental protection. Engine calibration is a fundamental process of determining the vehicle performance in diverse working conditions. Control maps are developed in the calibration process which must be conducted across the entire operating region before being implemented in the engine control unit to regulate engine parameters at the different operating points. The traditional calibration method is based on steady-state (pseudo-static) experiments on the engine. The primary challenge for the process is the testing and optimisation time that each increases exponentially with additional calibration parameters and control objectives.

This thesis presents a basic dynamic black-box model-based calibration method for multivariable control and the method is applied experimentally on a gasoline turbocharged direct injection (GTDI) 2.0L virtual engine. Firstly the engine is characterized by dynamic models. A constrained numerical optimization of fuel consumption is conducted on the models and the optimal data is thus obtained and validated on the virtual system to ensure the accuracy of the models. A dynamic optimization is presented in which the entire data sequence is divided into segments then optimized separately in order to enhance the computational efficiency. A dynamic map is identified using the inverse optimal behaviour. The map is shown to be capable of providing a minimized fuel consumption and generally meeting the demands of engine torque and air-fuel-ratio. The control performance of this feedforward map is further improved by the addition of a closed loop controller. An open loop compensator for torque control and a Smith predictor for air-fuel-ratio control are designed and shown to solve the issues of practical implementation on production engines.

A basic pseudo-static engine-based calibration is generated for comparative purposes and the resulting static map is implemented in order to compare the fuel consumption and torque and air-fuel-ratio control with that of the proposed dynamic calibration method.

Methods of optimal test signal design and parameter estimation for polynomial models are particularly detailed and studied in this thesis since polynomial models are frequently

used in the process of dynamic calibration and control. Because of their ease of implementation, the input designs with different objective functions and optimization algorithms are discussed. Novel design criteria which lead to an improved parameter estimation and output prediction method are presented and verified using identified models of a 1.6L Zetec engine developed from test data obtained on the Liverpool University Powertrain Laboratory. Practical amplitude and rate constraints in engine experiments are considered in the optimization and optimal inputs are further validated to be effective in the black box modelling of the virtual engine. An additional experiment of input design for a MIMO model is presented based on a weighted optimization method.

Besides the prediction error based estimation method, a simulation error based estimation method is proposed. This novel method is based on an unconstrained numerical optimization and any output fitness criterion can be used as the objective function. The effectiveness is also evaluated in a black box engine modelling and parameter estimations with a better output fitness of a simulation model are provided.

Acknowledgments

I would like to show my gratitude to my supervisor Dr. Tom Shenton for his guidance and support in the completion of this work. It was not possible to finish this project without his great help. I owe a thanks to Dr. Paul Dickinson for his indispensable assistance in the experimental setup. I would also like to thank all my friends: Shiyu Zhao, Ahmed Abass, Zongyan Li, Mingyen Chen and Ziyun Ding for their advice in discussions.

Thanks to my wife Meirong and my parents for their inspiration and encouragement during the undertaking of this work and thanks to ORSAS and Hong Kong Graduate Association Awards for their financial support.

Contents

List of Figures

Abbreviation

ABV	Air-Bleed Valve
ACF	Auto-Correlation Function
AFR	Air-Fuel Ratio
AIC	Akaikes Information Criterion
APRBS	Amplitude modulated Pseudo-Random Binary Sequence
ATDC	After Top Dead Centre
ARMAX	AutoRegressive Moving Average with eXogeneous inputs
ARX	AutoRegressive with eXogeneous inputs
BDC	Bottom Dead Centre
BIC	Bayesian Information Criterion
BJ	Box-Jenkins
BTDC	Before Top Dead Centre
DoE	Design of Experiment
ECU	Engine Control Unit
EGR	Exhaust Gas Recirculation
EMS	Engine Management System
FPE	Final Prediction Error
FPW	Fuel Pulse Width
GA	Genetic Algorithm
GDI	Gasoline Direct Injection
GTDI	Gasoline Turbocharged Direct Injection
HEGO	Heated Exhaust Gas Oxygen
IC	Internal Combustion
INJ	INJected fuel mass
IP	Interior Point
MIMO	Multiple-Input-Multiple-Output
MISO	Multiple-Input-Single-Output
MLE	Maximum Likelihood Method
MSE	Mean Squared Error
NARX	Non-linear AutoRegressive with eXogenous inputs
NN	Neural Network
OE	Output Error
OLS	Ordinary Least Square
PEM	Prediction Error Method
PFI	Port Fuel Injection
PRBS	Pseudo Random Binary Sequence
PS	Pattern Search

PSD	Power Spectral Density
RBS	Random Binary Signal
RPM	Revolutions Per Minute
RT	Real Time
SA	Spark Advance
SAN	Simulated ANnealling
SEM	Simulation Error Method
SI	Spark Ignition
SISO	Single-Input-Single-Output
SQP	Sequential Quadratic Programming
TDC	Top Dead Centre
TRR	Trust Region Reflective
TWC	Three Way Catalyst
UDRN	Uniformly Distributed Random Number
UEGO	Universal Exhaust Gas Oxygen

Chapter 1

Introduction

In recent years advanced technologies have been introduced to further reduce the fuel consumption and emissions of vehicles. These technologies require complex and expensive engine calibration work. With traditional hardware-based calibration methods, the experimental time increases significantly with additional calibration parameters and may not include important transient characteristics of the system.

Dynamic models and dynamic model-based calibration are thus being investigated, which are able to capture the dynamic behaviour and possibly decrease the cost of calibration by a reduction of set-points and settling time. Dynamic models can also incorporate data-smoothing into the model structure and integrate the calibration and control processes. As more calibration work is carried out on models rather than the real engine the requirement for model quality is essential. In this thesis methodologies of experiment design and model estimation are accordingly proposed to improve the accuracy of identified dynamic models required for calibration optimisation and system identification.

1.1 Advanced Technologies of Gasoline Engines

The gasoline engine has always been the most widely used type of engine in the automotive industry since the first development of the car. Although its performance has been constantly improved by continuous research over decades, there is still a large potential for further improvement by using model-based control technologies [2]. Advanced automotive engine technologies are being increasingly implemented in order to satisfy the customer demands on fuel economy and also the legislative requirements on scheduled emissions. Many new technologies have already been made commercially available and utilized in production. These are summarized in the following sub-sections.

1.1.1 Variable Valve Timing

The inlet and exhaust valves control the amount of air flow going into or out of the cylinder therefore the control of valves has a significant influence on the combustion and volumetric efficiency and so the resulting engine performance. In gasoline engines, the valves are driven by a camshaft which is normally connected to the crankshaft through the timing belt, and the opening and duration are determined correspondingly. For early engines in which the phasing of the camshaft was fixed, it was not possible to alter the timing under changing operating conditions so that the engine performance and fuel economy were necessarily a tradeoff between low-load low-speed conditions and high-load high-speed conditions. For instance a long opening at low engine speed will result in low fuel efficiency and increased emission since the fuel may leave the combustion chamber without a full combustion. Conversely it will be beneficial at high speed because of the less restriction on the air flow [4]. Moreover, the requirement for high-power during a drastic changing in speed cannot be well satisfied by traditionally fixed valve timing which was designed for optimal performance in high speed and high load conditions for maximum power. In recent decades the optimization tends to focus on low speed and low load because of the requirement for fuel efficiency and emission evoked by the concerns for oil supply and environmental protection.

Variable Valve Timing (VVT) refers to technologies which have the ability to adjust the scheduled valve timing flexibly in order to meet the desired performance in the various operating regions. These technologies have been implemented by many automobile companies and can be classified into four categories based on the controlled valves: phasing the inlet or exhaust valves only; phasing the inlet and exhaust valves equally or independently. To realize the variable timing, a mechanism that provided more than two cam profiles on the camshaft was proposed firstly in the Honda VTEC[5]. The driving cam was switched alternatively according to the engine speed. More recently, technologies of VVT for camless engine have been developed [6, 7]. The valves are directly controlled by an electromagnetic or hydraulic approach. This approach allows a continuous control depending on key control references such as torque and engine speed hence it is capable of obtaining optimal engine performance in different driving conditions. Nevertheless both electromagnetic and hydraulic valves need additional energy which will correspondingly reduce the fuel efficiency. The real-time control required by these various schedules raises the requirement for accurate and fast data collection for model development to support the more complex control system.

VVT has an effect which can reduce the fuel consumption and emissions. Normally the optimal timing of the inlet valves helps to increase volumetric efficiency so that the maximum torque for otherwise fixed parameters in the whole operating region can be improved which in turn increases the efficiency and fuel economy [8]. Effects from the timing of the exhaust valves contribute to the exhaust gas recirculation (EGR) which reduces the generation of

CO and NOx [9]. A detailed review and analysis of various strategies for VVT control is presented in [10, 11].

1.1.2 Gasoline Direct Injection

The technology of gasoline direct injection (GDI) has been an important innovation in automobile powertrain design in the last decade. In conventional port fuel injection (PFI) engines, the fuel injector is located in the inlet manifold outside the inlet valve of each cylinder. The injected fuel is firstly mixed with the air stream and then vaporized in the inlet port by the impact with the top surface of the inlet valve and then enters the combustion chamber with the opening of the valve. One of the associated disadvantages is that a fuel puddle is formed in the inlet port, also known as wall wetting which will compromise the accurate control of the fuel delivery and thus the fuel economy. Furthermore the resulting delay in fuel delivery may lead to misfire or rich combustion especially in cold-start [12].

GDI has totally solved the issue of wall wetting by injecting the fuel directly to the cylinder. Although the associated time between injection and ignition for mixture preparation is considerably reduced, the fuel spray can be well atomized within the time limit by using a high pressure injector. The amount of fuel in each combustion event thus can be accurately measured and controlled in different working conditions of the engine and excessive fuel supply is avoided. GDI provides the potential for implementing more complex control methodology. As the timing of GDI is independent of the valve timing, the engine management system (EMS) allows for multi-combustion models: stratified charge and homogeneous charge. Stratified charge is selected in low-speed low-load conditions in which the engine often experiences a constant speed or deceleration. A small amount of fuel is injected at the end of compression stroke so that the lean mixture is away from the cylinder wall when ignition happens. By reducing the wall heat loss, the thermal efficiency is significantly improved and the fuel economy enhanced accordingly. However, since the lean burn causes emission issues, a stoichiometric air-fuel ratio is required in most conditions. The fuel is injected in the intake stroke and the homogenous mixture leads to an exhaust gas which can be effectively converted by the catalyst.

Besides the major advantage in fuel economy, the merit of GDI is extended to emission control since the rich air-fuel ratio (AFR) caused by the generation of a fuel puddle in the cold-start is avoided; It is also beneficial in improving the transient response as less acceleration-enrichment for the puddle is required. A comprehensive comparison of PFI and GDI engine and control strategies of GDI combustion is documented in [13] and [14, 15].

1.1.3 Turbocharger

Volumetric efficiency refers to the ratio of the air on real fuel-air mixture inducted into the cylinder in each combustion event to that of the naturally aspirated engine at almost zero engine speed. It is a key criterion of the performance of internal combustion engines since it affects the maximum achievable power in a unit of a given capacity. Devices such as superchargers and turbochargers induct compressed air flow, also known as forced induction, to the cylinder and hence the associated allowable mass of fuel increases and more power can be generated in each combustion.

The power supply required for the associated additional compression is the major difference between a supercharger and turbocharger. A supercharger is directly connected and driven by the engine mechanically so that it has natural advantages of quick response to the working condition and a reliable power supply. Nevertheless since a part of the generated power needs to be used to maintain the running of the charger, this system may have relatively low efficiency [16]. On the other hand, a turbocharger is driven by the energy of the exhaust gas which was not utilized although it will increase the back-pressure. This system is composed of a turbine and compressor. The exhaust gas delivered into the turbine is controlled by a waste gate which is capable of diverting the gas away from the compressor. The boost-pressure of the intake manifold is thus regulated and the risk of damaging the engine due to effects such as knock can be consequently reduced. Turbochargers provide a significantly enhanced power in high speed conditions however they work much less efficiently at low speed conditions because the amount of exhaust gas is insufficient to spin the compressor to boost. Another challenge of this system is the turbo lag. Due to the basic mechanism of the turbocharger, the time required to generate the boost results in a time delay in the response to changes in working conditions. Correspondingly undesirable drivability issues might be caused in any accelerations.

A twincharger is a compound system composed of supercharger and turbocharger, which can solve the defects of each type of forced induction system effectively. This technology has been successfully implemented in several types of production car, such as the 125 kW 1.4 litre turbocharged stratified injection engine [17], but the disadvantages of the high cost of the components and the requirement for extremely accurate control raise new barriers to their adoption.

1.2 Engine Calibration Methodologies

Along with the development of mechanical and electronic technologies, the methodologies of engine calibration and control are also experiencing rapid developments. As an essential stage

of engine development, the engine calibration determines optimal settings for the best overall engine performance in the various operating conditions. In early times, calibrations were directly carried out on the engine while in modern times the development is moving towards model-based or simulation-based methods because of the rapidly increasing complexity.

Conventional calibration methods which have been used worldwide are based on pseudo-static testing on the real engine over the entire operating range. Since the experiments are conducted directly on the engine, the results obtained from the test bed are considered accurate and reliable enough for implementations on current production vehicles. Nevertheless this method has also been criticized for its inefficiency in testing and optimization [18]. In general, all inputs need to be swept in order to find the optimal point at each operating point and therefore a large number of experiments is unavoidable. Due to the nature of steady-state testing, it is necessary to wait for the output response to reach a steady state which in turn further increases the required experimental time. Moreover with the development of advanced engine technologies, more engine parameters and variables including valve timing and waste gate timing become controllable and the associated dimension of experiments increases significantly.

Model-based calibration methods have been developed to reduce the cost of experiment [19]. A global model or local models are identified from engine data in the operating regions and then used as a replacement of the real engine for offline calibration and optimization which takes the majority of the online calibration burden out of the engine test bench and into a PC. The accuracy of models is a crucial factor since it significantly affects the effectiveness of optimal settings for the controllers which should be robust to the uncertainty in the models. The benefit in reduced experimental cost from employing model-based calibration has popularised these methods which mainly include static and dynamic model-based calibration methods.

1.2.1 Static Modelling and Mapping

Figure 1.1 demonstrates a typical static model-based calibration approach. “Minimap” points denote representative local operating points. Local tests are made at each point and steady-state data is collected for the identification of static models. Subsequent experiments for determining the optimal settings are carried out on the resulting mathematical models and local optimal settings are used to form calibration maps for the whole operating region. In general the derived model is able to generate the simulated result in a short time hence the settling time required in engine tests can be substantially reduced. As the data for analysis is recorded in the steady state, the transient behaviour of the system is neglected so that the overall performance might be compromised when the driving condition changes abruptly,

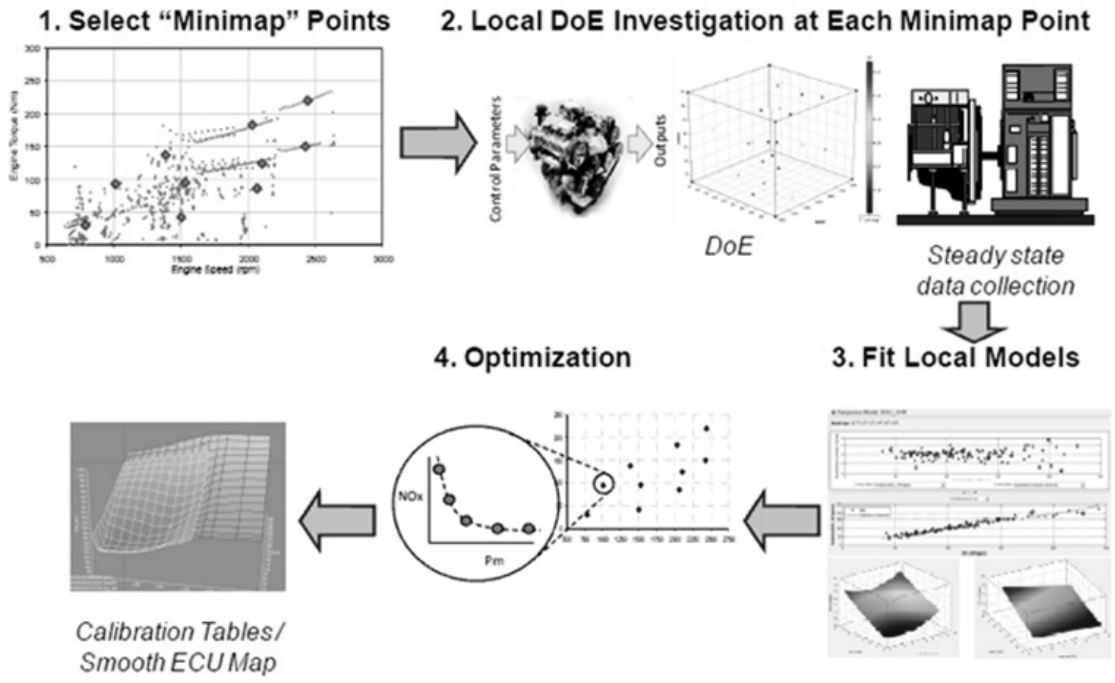


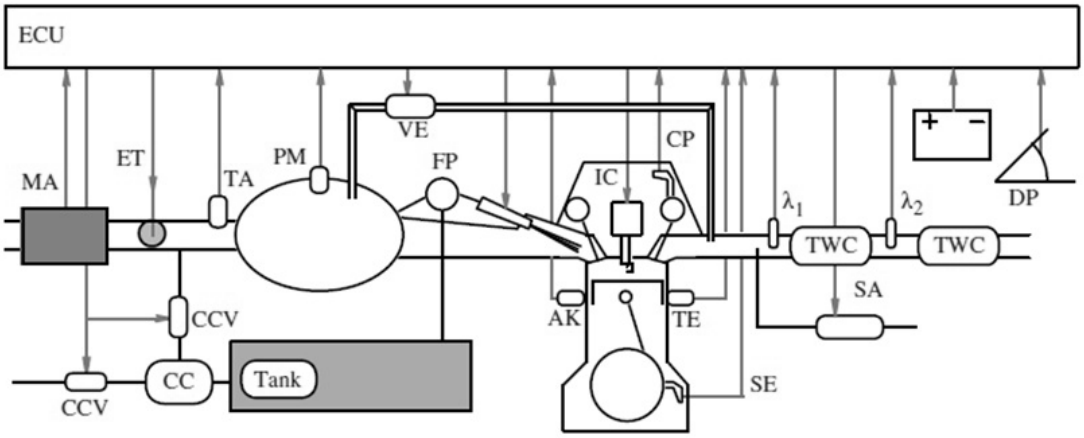
Figure 1.1: A procedure of model-based static calibration [1]

such as acceleration or deceleration.

1.2.2 Dynamic Modelling and Mapping

To capture the important dynamics of the system, dynamic modelling and mapping can be employed. In the design of experiments (DoE), test signals should be appropriately designed in order to excite the system dynamics and the input-output data are collected for model estimation. Dynamic models describe the system behaviour by using the current and past values of inputs and outputs so that they are capable of describing the transient response of inputs and outputs. This approach also gives a potential for removing the burden of selecting operating points and local testing at each point since a well designed dynamic model using a clustering algorithm is able to simulate outputs at different operating points with good accuracy [20]. In the dynamic optimization, it may be possible to use the optimal settings to identify a model which would interpolate and extrapolate to predict the optimal values across a dense set of operating points.

Modelling and control of dynamic systems have been studied by many authors [21, 22, 23]. The extensive applications in engine calibration are well documented in [24].



AK	knock sensor	PM	manifold pressure sensor	VE	EGR valve
CP	camshaft sensor	ET	electronic throttle	SA	secondary air valve
IC	ignition command	TA	intake air temperature sensor	TWC	3-way catalyst
MA	air mass-flow sensor	TE	cooling water temperature sensor	ECU	controller
SE	engine speed sensor	CC	active carbon canister	CCV	CC control valves
FP	fuel pressure control	$\lambda_{1,2}$	air/fuel ratio sensors	DP	driver pedal

Figure 1.2: A schematic configuration of a PFI IC engine [2]

1.3 Engine Control

The obtained optimal settings are used to construct a calibration map in the form of look-up tables and are stored in an engine control unit (ECU). The EMS collects the inputs from engine sensors, searches for the stored settings and controls the actuators in real time to produce the optimal performance. Figure 1.2 illustrates a simplified hardware configuration of a PFI Internal Combustion(IC) engine.

The entire control system often includes a large number of feedforward and feedback control loops that are used to satisfy increasing performance requirements. The best engine performance in terms of smooth response and powerful output with the least fuel consumption has always been the top requirement of customers and automotive manufacturers. Meanwhile legislation for environmental protection encourages the technological progress to address the requirement of reducing vehicle emission. These requirements can only be satisfied by the use of new electronic and mechanical automotive mechanism and control technology.

1.3.1 Torque Control

Engine torque is a vital characteristic of engine performance, which represents the power generated in a fuel combustion for a given speed. In a production car it is closely related to the achievable maximum vehicle speed and acceleration. On a test bed, the torque is measured

by a coupled dynamometer. On a production car, however it is not usually possible to measure torque directly except with very expensive test instrument and normally it is estimated by a model obtained from offline experiments. Engine torque is determined by the combustion of the air-fuel mixture in the cylinders consequently it can be controlled in two ways. As the throttle angle affects the intake air flow which in turn determines the allowable fuel injection and the mass of mixture, control of throttle is a common and effective approach of regulating engine torque in the spark ignition (SI) engine although it generates additional pumping losses. Originally, the acceleration pedal was mechanically connected to the throttle so that the driver could directly adjust the torque in a simple and quick manner. Alternatively, now an electronic throttle control converts the signal of the pedal position into a desired power output and the ECU will select or calculate coordinated optimal settings of all related actuators accordingly. This technique provides a more flexible and precise control. However the transient response might be slower than the conventional mechanical control since the throttle plate is adjusted by filtered signals derived from feedback controllers in the ECU.

Combustion control is the major factor in transferring the chemical energy of the fuel into kinetic energy therefore it also has a critical influence on engine torque. In a spark-ignition IC engine, the spark advance (SA) angle needs to be optimized for maximum efficiency of the combustion. The ignition causes an increase in in-cylinder pressure which creates the piston work. A very early spark in the compression stroke will waste the energy required to push the piston and will unnecessarily heat the cylinder wall. On the other hand, more energy will be lost in the gas out of the cylinder in the exhaust stroke rather than being used to accelerate the crankshaft if a too late spark occurs [3].

The main task of torque control is for the generated torque to track the desired torque profile, in which both accurate steady-state values and rapid transient responses are commonly required. Since the speed is relatively slowly changing and this is perceived by the driver, the torque and power control are equivalent from the driver's perspective. In practice the need for high steady-state accuracy of torque is not great since the driver can manually compensate the error by feedback compensation using the accelerator pedal. An open loop control with an acceptable settling time may have the potential of satisfying the requirement for good torque control.

1.3.2 Fuel Control

According to the ECU map settings, for any specific demand of torque the fuel consumption may vary and so the best fuel economy must be obtained by an optimized fuel controller. As mentioned above, the timing of spark is an essential factor since it determines the location of maximum burning rate and maximum work rate in the engine cycle. On the other hand,

advanced engine technologies are introducing more factors in the fuel optimization. Variable valve timing promotes the fuel efficiency by controlling the inlet valve. The overlap between intake and exhaust valve is extended by early intake valve opening and consequently the burnt high pressure and temperature gases will be pushed back to the intake manifold and sucked into the cylinder in the next cycle and therefore the pumping losses are reduced. Early valve closing happens when the desired amount of mixture is introduced in the cylinder so that the required work for pumping is minimized. As gasoline direct injection (GDI) technology can eliminate the fuel puddle in the conventional PFI engine, the compensation for the fuel film dynamics is not needed. The flexible injection time control in GDI provides a further potential for fuel reduction in low-speed low-load condition. The exhaust gas recirculation and turbocharger can also contribute to the fuel efficiency by reducing the cylinder volumetric capacity through utilizing the energy of the exhaust gas.

Idle speed specifies a special low-speed low-load condition in which the fuel is consumed only to prevent the stall of the engine. In order to reduce the fuel consumption, the rotational speed of engine is expected to be as low as possible but still capable of maintaining the smooth engine performance whilst still operating the ancillaries. Since approximately one third of the fuel is consumed in idle speed because of the traffic congestion [25], the development of an efficient idle speed controller will make a significant contribution to the fuel economy.

1.3.3 Air-Fuel Ratio Control

An effective and efficient after-treatment system is essential to satisfy the increasing legislative requirement for the reduction of emissions. The converting efficiency of the three way catalyst (TWC) is mainly affected by the AFR and the AFR is normally desired to be stoichiometric, usually about 14.7:1 or $\lambda = 1$, to ensure the optimal performance of the TWC [26]. As shown in Figure 1.3, for a SI engine the main poisonous substances, NOx, CO and HC of vehicle exhaust can be majorly filtered by the TWC only if the λ is in a narrow window around 1. Hence the air-fuel ratio control is the most important feedforward and feedback control for the regulation of emissions.

In common practice, an oxygen sensor is placed in the collective exhaust pipe before the TWC. Instead of directly measuring mass of air and fuel in the mixture, this sensor measures the proportion of oxygen and the AFR is determined accordingly [3]. A second λ sensor positioned after the TWC as in Figure 1.2 is used to monitor the efficiency of the TWC. A typical widely used oxygen sensor is the heated exhaust gas oxygen (HEGO) sensor. Although the HEGO sensor benefits from its low cost, its output voltage is quite nonlinear to the AFR. The resulting output voltage changes drastically around stoichiometric while it becomes much less sensitive to lean and rich AFR. Thereby the measuring capability is

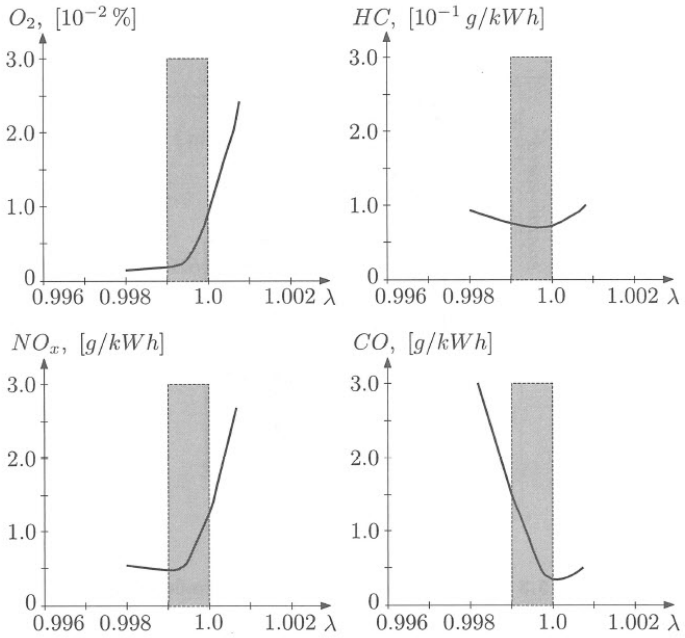


Figure 1.3: Engine emissions after the TWC with different λ [3]

relatively poor for control purpose and its use is limited to limit-cycle control. An alternative choice is the universal exhaust gas oxygen (UEGO) sensor which is capable of measuring the AFR linearly across a wide range and therefore it is generally preferred in the test bench since linear control technologies can be applied. However the high price of UEGO affects its implementation in production cars.

As can be seen from Figure 1.3, the emission rates of the exhaust gas varies and the high conversion efficiency is achieved only within a very small window around the stoichiometric point therefore a precise control of AFR in static and transient situations is required. To design a feedforward controller, the dynamics of the intake air-path must necessarily be estimated in order to predict a suitable fuel flow in the next engine cycle corresponding to the related engine signal e.g. throttle position and SA. However for the purpose of reducing the steady-state offset to an acceptable limit, developing a model with required global accuracy in the operating region is excessive time consuming. In order to eliminate the steady-state error, closed loop control can be employed. However the biggest challenge of adapting feedback control is the long time delay caused by the transport delay associated with delivering the raw exhaust gas from the actuator, which is usually the fuel injector, to the λ sensor. Consequently a combined feedforward-feedback control will be needed to overcome the defects of each control method. With the implementation of certain advanced engine technologies, the control system design can be simplified. For instance in conventional PFI engines, the influence of fuel puddle and the time delay between the fuel injection and inlet valve opening needs to be compensated in the estimation of air-path dynamics while the resulting difficulties

of modelling can be reduced by using the GDI technology.

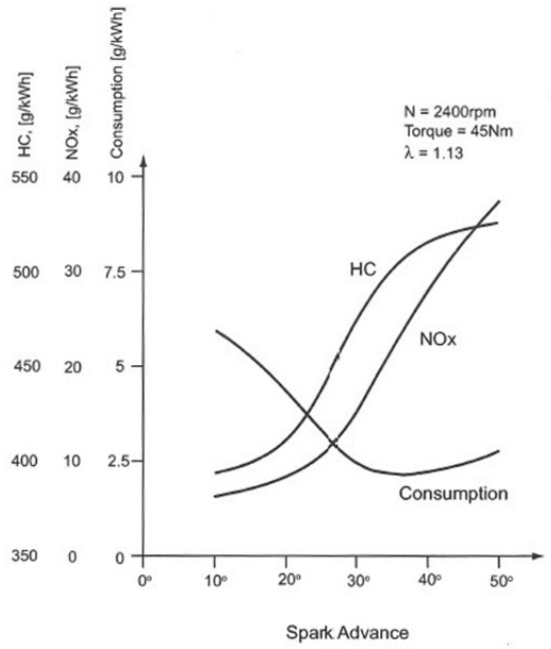


Figure 1.4: Emission and fuel consumption in SA sweeping [3]

Since the engine control system is complex and multi-objective oriented, some actuators are often included in different control loops that have conflicting effects . In these cases, a large amount of calibration work in searching for compromise solutions is considered necessary. Figure 1.4 shows a map of fuel consumption and emission with respect to SA at a specific operating point [3]. The optimal SA has to be a trade-off between the two control objectives.

Besides the three main control requirements, other demands related to safety and drivability such as knock control also should not be neglected. Detailed descriptions of the engine and engine control systems can be found in [2, 3, 27].

1.4 Motivations and Objectives

The motivations and objectives of this thesis can be summarized by two aspects:

1.4.1 Dynamic Model and Calibration

As stated previously, the main challenge in the engine-based calibration process is the expensive experimental cost and time. The acknowledged means of addressing this problem is the increased use of model-based methodologies in the calibration process since models can

generally run much faster than the engine in real time. Because of the increased importance of transients in the calibration, e.g. forthcoming drive cycles with significant transient components, dynamic models (which relate the current output to the past values of input and output data) are likely to feature more within the calibration process to enable improved transient optimisation and the minimisation of transient emissions in particular. Existing static testing is time-consuming since it requires the test-bed to settle to steady-state conditions. The development of dynamic models using system identification methods has the potential to reduce the associated time and cost. In this thesis, a method of dynamic model-based calibration is proposed in order to minimise the fuel consumption with constraints on engine torque and AFR and its control performance is compared to that of a conventional hardware-based static calibration.

1.4.2 Optimal Design of Experiments

For model-based calibrations, the accuracy of the models is the most important factor which determines the online performances of calibrations. More accuracy can generally be obtained by increased experimental testing, however this is expensive in time and resources, and requires significantly increased effort unless a careful design of experiments is determined. To further effectively improve the quality of models, improved methodologies for the design of experiments are required to be developed. Design of experiment methodologies are well developed for static based modelling. However relatively few techniques have been developed for the development of dynamic models and the related problem of optimal test-signal design for dynamic testing has received little attention in the last few decades. The optimisation of test-signals for dynamic model development is difficult because it requires computational expensive optimisations. In this thesis, the influence of optimal test input design and optimal parameter estimation methods for model accuracy is investigated and new methods developed. A new efficient objective function for use in optimal input design is proposed which has the potential to significantly reduce the computational cost and a simulation error based estimation method which is suited to the calibration applications is developed for the associated estimation of simulation models.

1.5 Overview

Chapter 2 presents a general procedure of system identification and controller design including a brief discussion of the general methodologies in each step. Real systems can be conveniently classified as white box or black box models according to how much a prior knowledge of the system is available, and can be furthermore identified as prediction models or simulation models based on the requirements of the selected approach to controller design.

Methodologies of choosing the input signal, model structure and estimator are introduced for developing accurate models and their effectiveness are evaluated by means of validating models using criteria regarding the error between measured output and estimated output. Approaches to data pre-processing in order to reduce the affect of the stochastic of data logging are discussed.

Chapter 3 gives the experimental setup of a 1.6L Zetec real engine and a 2.0L GT-DI virtual engine for the implementation of proposed identification, calibration and control methodologies in this thesis. Details and features of the engines are described together with how the main actuators and sensors are installed and how the control interfaces, D-space and WAVE are connected. As the experiments are conducted in different operating regions, the real engine is coupled with a low inertia dynamometer in order to apply various loads to restrict the engine speed to required ranges. In the virtual engine, additional sub-models are developed to simulate the in-cylinder combustion and road load.

Chapter 4 discusses an implementation of the optimization in test signal selection. The proposed iterative procedure of optimal input design is based on an assumption that a relatively accurate initial model of the real system can be developed. Signals with wide frequency content and experimental constraints are recommended for the identification of any initial model to overcome the disadvantage of the unknown frequency range of the system. Optimal input design is classified into two main types according to the objective of the optimization. The first type of criteria is based on the parameter variance/covariance. The effectiveness of A-optimal and D-optimal criteria are discussed and a weighted A-optimal criterion is proposed for inputs of different scales. An illustrative example is given to evaluate the efficiency of various optimization algorithms. The second type of criteria is based on the minimization of output prediction error. A method of selecting the objective signal is proposed and a new criterion is derived from an adaptation of I-optimal criteria, which leads to a considerably improved computational efficiency. Practical constraints in the design of identification experiment are studied and their influences on optimal input design are demonstrated. The effectiveness of input design is firstly assessed by applications on a known system which was obtained by experimental engine data. An implementation on black box modelling, which is that of identifying a torque model of a virtual engine, is discussed subsequently with the purpose of exploring its feasibility in industrial applications. A preference-based optimization of input selection is investigated and the method required in designing an optimal input for estimating two MISO models as components of a MIMO model in one experiment is exhibited. The validation of model quality is performed statistically by examination of multiple cases, which is consistent with the statistical theory employed in parameter and output estimation.

Chapter 5 describes the approach to choose an estimation method according to the model types. As a type of prediction error methods (PEM), the ordinary least square (OLS) is suit-

able to estimate parameters of prediction models and also can be adapted to approximately estimate simulation models. A simulation error method (SEM) which minimizes the error between the measured output and simulated output is proposed especially for estimating simulation models. The selection of algorithms for the unconstrained optimization is discussed and the SEM is demonstrated giving better model accuracy than the PEM by examples of identifying parametric models of a known system and a unknown system.

Chapter 6 introduces a basic engine-based static calibration on the virtual engine aiming to minimize the fuel consumption with constraints on desired torque and stoichiometric air-fuel-ratio. The torque and λ are regulated by feedback controllers and the SA is swept across a safe range to find the optimal value. Steady-state values of outputs and inputs at the optimal settings are recorded in local tests. The static tests are carried out at each operating point and the results form a look-up table accordingly. The static map is validated online and demonstrated to be effective in the low-speed low-load region. The performance of the resulting static map is utilized as the basis for comparison to the dynamic calibration.

Chapter 7 presents a dynamic model-based calibration with the same control objectives as the static calibration. Dynamic models of torque and λ are used to replace the real engine in the calibration. The process of identifying polynomial models includes advanced DoE methodologies for optimal input design and parameter estimation in order to improve the model quality. Another model type, the recurrent neural network model which can represent system nonlinearity conveniently is also employed to develop the models. Optimal settings of calibration parameters are obtained by a constrained numerical optimization. Since long data sequences need to be optimized, various optimization methods are proposed and one particular method named the segment method is selected to improve the computational efficiency. The optimal data obtained on the dynamic models are examined on the black box virtual engine and additional constraints are applied to improve the consistency of regulated torque and λ. After removing the time delay, inverse optimal data is utilized to identify three dynamic models of injection flow, SA and throttle position and the models are proved to be capable of producing desired torque and λ in the operating region with minimized fuel consumption. Feedback PI controllers are designed to corporate with the feedforward map with the purpose of reducing the steady-state offset. An open loop compensator of engine torque is developed and implemented in the closed control loop since it is not feasible to apply torque sensors in production cars. By using a Smith predictor, the effect of the significant time delay caused by transportation in the λ control loop is reduced. A discussion on the results of applying both the dynamic map and static map are given based on an analysis of the fuel economy and the output responses of torque and λ.

Chapter 2

Literature Review

2.1 Introduction

System identification estimates mathematical models by statistic methods with the purpose of representing real dynamic systems. Figure 2.1 depicts a general procedure of system identification and in this figure we address that the system identification can be conducted iteratively by using methodologies of input signal design, model structure selection and parameter estimation. Popular technologies of each step in the procedure are introduced in this chapter. The steps of input design and parameter estimation are improved by our proposed methodologies in this work and will be introduced in later chapters.

2.2 System Modelling

2.2.1 Prediction and Simulation Model

$$\hat{y}(t) = f\left[u\left(t\right), \ldots, u\left(t-1\right), \ldots, u\left(t-m\right), y\left(t-1\right), \ldots, y\left(t-n\right)\right] \tag{2.1}$$

$$\hat{y}(t) = f\left[u\left(t\right), \ldots, u\left(t-1\right), \ldots, u\left(t-m\right), \hat{y}\left(t-1\right), \ldots, \hat{y}\left(t-n\right)\right] \tag{2.2}$$

Equations (2.1) and (2.2) show typical models for the purpose of prediction and simulation respectively, where m and n denote the maximum time delay of input and output respectively and the values can be determined arbitrarily or according to methods of regressor selection. In prediction, the previous values of input u and output y are collected from the real system and the value of the current output is estimated accordingly. However in simulation, only values of previous inputs are required from the system, whereas the values of previous output are estimated from the simulation [28].

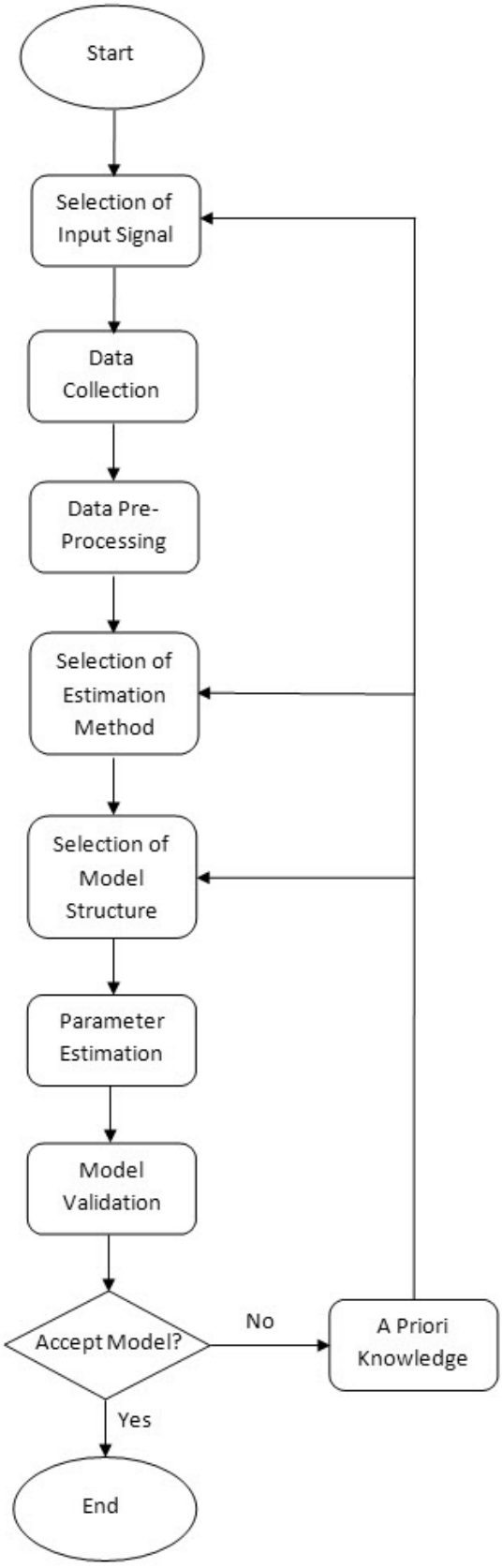


Figure 2.1: A general procedure of system identification

The type of models developed should be determined by the planned application. Prediction models can be implemented for online control problems and require online measurement of the output. Simulation models typically have feedback components from themselves. Although the issue of model stability needs to be in-depth considered in the process of identification, simulation models have been widely used in offline control and optimization tasks because of their independence of system output measurement.

2.2.2 White Box and Black Box Model

The term white box model usually refers to physical systems where the internal mechanisms and processes are available to inspect, e.g. models of a single pendulum system or a serial circuit. The model structure can be acquired and understood by analysing the inner components and logic using relevant principles and laws of physics, e.g. Newton's law and Ohm's law. Parameters should be known with a high degree of certainty, e.g. mass and resistance. As the physical causality of inputs and outputs is clearly exhibited, white box modelling provides a deep insight into the real system. Another advantage as mentioned in [29], is that once a satisfactory white box model is obtained, it can be easily adapted to similar systems by means of slightly modifying the model structure or parameters, while the black box models are only reliable for the very system and operating range over which they are identified and a considerable amount of trial and error test is needed when adapted even to similar systems.

In [2], the techniques for physical modelling with particular application to powertrain models are introduced and described in detail. A typical engine in-cylinder thermodynamics model is given in [30] and a typical kinematics model in [31]. However, in an IC engine, a large number of complex physical processes including the kinematics, thermodynamics and fluid dynamics occur simultaneously. Therefore obtaining an accurate white box model can be extremely difficult and time consuming.

In contrast to a white box model, a black box model is a system which can only be characterised in terms of its input and output. To develop a black box model, system identification methods can be utilized to identify an appropriate structure and parameters from analysis of the input-output sequence collected from the real system [32]. The limitation of the black box modelling is that the reliability of the identified model may degrade with the expansion of the operating region. Nevertheless, it is still considered as an efficient and fast approach for dynamic engine modelling since a physical understanding of the internal mechanisms is not absoluately required for many purposes. Details of system identification methods can be found in [33, 34, 35].

2.3 Input Signal Design

The input signal is crucial to system identification since it should effectively excite the dynamical behaviour of the system in the operating region of interest. Therefore, the selected input cannot be too simple or weak, as maybe a cosine signal with small amplitude for example. In black box modelling, as a prior information of the system is not available initially, banded white noise signals or signals which are generated from a filtered white noise source are often favoured because they contain a wide range of frequency content. In practice, a so called pseudo-random binary sequence (PRBS) can be ideal for linear system identification. It can be adjusted to any demanded binary level and the output limited correspondingly. As the behaviour of nonlinear systems is more complicated, the collected data must contain significantly more information, whereas binary signals cannot fully excite the nonlinear behaviour of a system and may lead to loss of identifiability [36]. Therefore, multi-level signals, such as amplitude modulated pseudo-random binary sequence (APRBS) and random walk sequence, can be chosen for nonlinear identification.

As stated in the Nyquist-Shammon sampling theorem, a signal can be identified only if its maximum frequency is less than half of the sampling rate. It is also generally suggested that an adequate sampling rate should be around 10 times the possible bandwidth of the system [34], or in practice it can give us 4-6 samples within the rise time of the system. However, it might be beneficial to use a higher sample rate so that the user can identify models according to different experimental requirements by down sampling.

After the initial estimation, optimal test signal design could be conducted with the obtained prior knowledge of the system in order to excite the dynamics better. Consider the case of discrete nonlinear dynamic system model in input-output form expressed by a combination of nonlinear input-output regressors which are linear in the parameters, together with a white Gaussian noise term:

$$\begin{aligned} y(k) &= \sum_{i=1}^{N} H_i(\theta) f_i(u(k), \ldots, u(k-d_u), y(k-1), \ldots, y(k-d_y)) \\ z(k) &= y(k) + \epsilon(k) \end{aligned} \tag{2.3}$$

where k is the time index, $u(k)$ is a $p \times 1$ input vector at time k and $y(k)$ and $z(k)$ are undisturbed and disturbed $q \times 1$ output vectors at time k, H is a smooth parameter function term, f is a smooth input-output function term with maximum delays d_u and d_y in u and y respectively, N is the number of regressors in the model structure and $\epsilon(k)$ is a $q \times 1$ noise vector at time k, in which each individual entry $\epsilon_j(k)$ has zero mean and covariance σ_j^2

2.3.1 Information Matrix and Cramer-Rao Law

An optimal input is required to excite the system dynamical behaviour to maximise the data information in the experiment. If the input is deterministic, the data information content is determined by the Fisher information matrix [37]:

$$M \equiv E_{Y|\theta}\left(\frac{\partial \ln p(Y|\theta)}{\partial \theta}\right)^T\left(\frac{\partial \ln p(Y|\theta)}{\partial \theta}\right) \tag{2.4}$$

where Y is the output sequence and θ is the vector of parameters. If the model is expressed in equations of (2.3), M can be given by [38]:

$$M = \frac{1}{\sigma^2}\sum_{t=1}^{N}\left[\frac{\partial y(t)}{\partial \theta}\right]^T\left[\frac{\partial y(t)}{\partial \theta}\right] \tag{2.5}$$

where $y(t)$ is the output at a time instant, σ is the variance of noise and N is the length of output sequence. In optimal input design, the objective is thus generally taken as finding an input u which maximises the information content of the data, based on some measure of the information matrix M. The partial derivatives in the elements of the sensitivity matrix $\frac{\partial y(t)}{\partial \theta}$ are the output sensitivities, which can be determined from the input-output form by solution of:

$$\begin{aligned}\frac{\partial y(k)}{\partial \theta_i} &= \sum_{j=1}^{N}\frac{\partial H_j(\theta)}{\partial \theta_i}f_j + \sum_{j=1}^{N}\sum_{l=1}^{d_y}H_j(\theta)\frac{\partial f_j}{\partial y(k-l)}\frac{\partial y(k-l)}{\partial \theta_i}\\ \frac{\partial y(1)}{\partial \theta_i} &= a\end{aligned} \tag{2.6}$$

where a is the initial condition vector of the output sensitivity terms. The output sensitivities indicate the influence of each parameter on the model output. A small change in the parameter will have a considerable influence on the model output, provided the output sensitivity is high. While if it is low, the model output may not have a distinguishable change even for large parameter changes.

The accuracy of parameter estimation is determined by the covariance matrix of the estimated parameter vector $\hat{\theta}$ where according to the Cramer-Rao law [39, 40]:

$$\begin{aligned}cov(\hat{\theta}) &\equiv E\left[\left(\hat{\theta}-E[\hat{\theta}]\right)\left(\hat{\theta}-E[\hat{\theta}]\right)^T\right] \geq M^{-1}\\ &= \left\{\frac{1}{\sigma^2}\sum_{t=1}^{N}\left[\frac{\partial y(t)}{\partial \theta}\right]^T\left[\frac{\partial y(t)}{\partial \theta}\right]\right\}^{-1}\end{aligned} \tag{2.7}$$

According to [41], an unbiased estimator, such as an ordinary least square estimator is said to be efficient if its covariance is equal to the Cramer Rao lower bound. As the covariance matrix of an efficient estimator is related to the Cramer-Rao lower bound which is determined

by the information matrix, optimising the data information corresponds to optimising the parameter covariance. A scalar measure of the information matrix such as $tr(M^{-1})$ (A-optimal) or $-\ln(det(M))$ (D-optimal) is accordingly favoured as the performance index to be optimised in the test signal design [42].

2.3.2 Optimation Algorithms and Design Criteria

In early developments, optimal test-signal methods were based on the use of local optimisation techniques. Goodwin [43] presented optimal excitation signal design for discrete nonlinear system identifications based on steepest-descent and conjugate-gradient methods. Mehra [44] developed an optimal input obtained using a Riccati equation method for continuous linear system identification. Kalaba and Spingarn [45, 46] employed quasi-linearisation and Newton-Raphson methods to solve an associated boundary value problem in the nonlinear case. These algorithms employ an analytically obtainable gradient to determine a local minimum. With the advent of successful global optimum algorithms, Lejeune [47] used a generalized simulated annealing for heuristic optimization of experiment design and showed its increasing effectiveness for larger models. Reeves and Wright [48] used genetic algorithms in an experimental design perspective and compared these with the current alternative methods. Later improvements in global numerical algorithms and globally optimised DoE have subsequently lead to a significant reduction in required experimentation time [49, 50, 51].

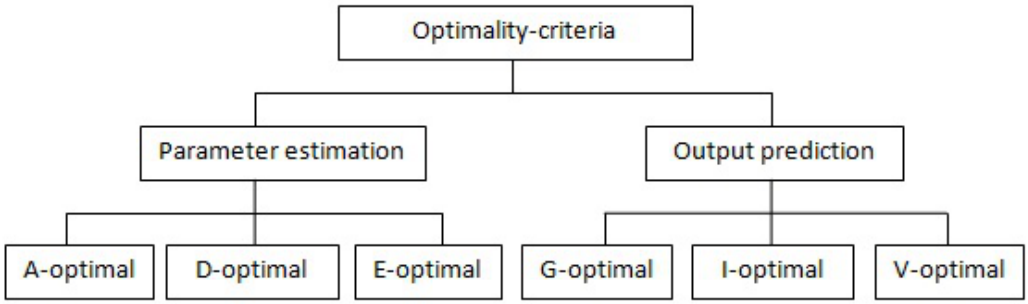


Figure 2.2: Overview of DoE optimality-criteria

For both information-theoretic and tractability reasons, many optimality-criteria for the design of experiments (DoE) are concerned with the variance of parameters. A-optimal designs minimise the trace of the inversed information matrix. Aoki and Staley [52] and Nahi and Napjus [53] used A-optimality as a criterion since it leads to a quadratic optimisation problem which is numerically tractable. E-optimality, which maximises the minimum eigenvalue of the information matrix, was used by Heiligers [54] in weighted polynomial regression. D-optimality minimise the determinant of the information matrix. Mehra [55] found that an important advantage of D-optimality is that it is independent of scale changes in the pa-

rameters and linear transformations of the output. Zaglauer and Delflorian [56] developed a Bayesian modification of the D-optimal design for use in dynamic engine testing which avoids bias towards the experimental boundaries. State-of-the-art dynamic testing procedures for industrial application were presented by Schreiber et. al. [57] who proposed and employed the use of independent Pseudo-Random-Multilevel signals in combination with D-optimal amplitudes for MISO engine testing. Figure 2.2 shows a taxonomy of current DoE criteria.

Now, many real systems are so complex that they cannot be easily identified using white box approaches where the model structures have physical meanings. For instance, in an internal combustion engine, a large number of complex physical processes including the kinematics, thermodynamics and fluid dynamics interact simultaneously in 3D, and so obtaining an accurate white box model can be extremely difficult and time consuming or simply not feasible. On the other hand in black box modelling, an appropriate structure and parameters must be obtained from an iterative analysis of input-output sequences collected experimentally from the real system. Because the model structures are not unique in black box modelling, assessments which are based only on the variance of the parameters become less meaningful and output prediction accuracy is a more appropriate criterion.

Optimality-criteria applied in DoE for black box modelling have however, generally been based on the variance of output predictions. Thus G-optimal criteria minimise the maximum variance of the predicted values. Wong and Cook [58] have discussed the conditions for the equivalence of D and G-optimal criteria and addressed the issue of constructing G-optimal designs when the errors are not homoscedastic. Lizama and Surdilovic [59] designed G-optimal experimental test signals for identification of system dynamics. I-optimal optimization minimises the mean variance of estimators over the operation space while V-optimal assesses optimality over a reduced set of specific points selected from the operation space. Kapelle [60] shows the advantages of I-optimal designs over the more conventional designs used in industry, arising since it has a narrower confidence limits on predictions. Debusho and Haines [61] discussed V and D-optimal designs for linear regression models with a random intercept term.

A further issue is that any practical experimental design must also take into account the constraints on the allowable experimental conditions. Algorithms using penalty functions and Lagrange multipliers have thus been used to incorporate constraints into the optimal cost function. Typical constraints that might be met in system identification practice have been studied by Goodwin [43], who applied input and state amplitude constraints to the optimal test signal design for nonlinear system identification. Ng et. al. [62] discussed the achievable estimation accuracy with constrained input and output variance and also presented a method of optimal input design for parameter estimation for an autoregressive model with constrained output variances, which caused very little computational burden [63]. Forsell and Ljung [64]

gave an explicit solution to an experiment design problem in identification for control with constraints on both the input and output power. Morelli and Klein [65, 66, 67] discussed techniques that can be applied to either linear or nonlinear dynamical systems with practical constraints imposed on input and output amplitudes.

2.4 Data Pre-Procession

After the experimental data is collected from the real system, it may not be a good choice to fit them to the model immediately. Deficiencies of the data, if present, will affect the identification and therefore data pre-procession is recommended. Typical defects and amending methods are discussed in the following section.

2.4.1 Dealing with Offsets

Offset denotes steady-state bias of the data. The experimental data often describes two types of relationship between input and output: the effect on the output of varying the input and the resulting output when the input is a constant. As most real systems are nonlinear, the purpose of identifying a linear model is typically for describing the output response for small deviations from a physical equilibrium. However since a pure linear model generally does not include a term of constant, the offset can not be precisely presented. For this reason, offsets should be removed to avoid their disadvantageous influence on the identification. If an additional static experiment is feasible, the offset can be removed by setting the input close to that for the desired operating point and subtracting the input and corresponding steady-state output from the raw data. In an offline application, the process can be done by subtracting the mean values of input and output from each sample.

2.4.2 Dealing with Outlier Points and Missing Data

Outlier Points represent abnormal data as a result of mistakes in measurement or special issues in experiments, for instance spark knock in an IC engine. These bad points can be determined by plotting the data or analysing the residue. To manage outliers, a simple solution is to split the data sequence into sections, reject the section with outliers and merge the rest of the data. In situations when it is hard to find a segment of clean data, the outliers can be considered as missing points. Missing input can be considered as unknown parameters while missing output can be considered as irregular sampling. [34] and [68] include more details and further discussions.

2.4.3 Dealing with Disturbance

It is well known that the relation between input and output data from a linear system will not be affected by implementing both the input and output data through the same filter. Any disturbance or frequency content beyond the band of interest can be weakened or eliminated from the collected data by filtering. High frequency or low frequency disturbance can be removed by low pass or high pass filtering respectively, the corresponding frequency content of the data will be however filtered as well. A feasible approach is to employ a band stop filter, provided that the frequency range of the disturbance is known. On the other hand, a band pass filter can be very suitable if the identification is made within a specific frequency band. Although building an additional noise model has an equivalent effect to remove disturbance [69], filtering is often considered as a better alternative approach since it will not affect the model structure.

2.5 Selection of Estimation Methods

Consider a discrete linear polynomial dynamic model:

$$A(q)Y = B(q)U + \epsilon \tag{2.8}$$

where q is the time shift operator and A and B are polynomials in q^{-1}

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_m q^{-m} \tag{2.9}$$

$$B(q) = b_1 q^{-1} + b_2 q^{-2} + \ldots + b_n q^{-n} \tag{2.10}$$

The system can be presented in the full regression form at the sample instant t by:

$$\begin{aligned} y(t) = & -a_1 y(t-1) - a_2 y(t-2) \ldots - a_m y(t-m) \\ & + b_1 u(t-1) + b_2 u(t-2) + \ldots + b_n u(t-n) + \epsilon(t) \end{aligned} \tag{2.11}$$

where m and n are the time delays in output and input. A simplified regression form can be given as:

$$y(t) = x(t)\theta + \epsilon(t) \tag{2.12}$$

where $\theta = [-a_1, -a_2, \ldots, -a_m, b_1, b_2, \ldots b_n]$ is the vector of parameters and $x(t) = [y(t-1), y(t-2), \ldots, y(t-m), u(t-1), u(t-2), \ldots, u(t-n)]$ is the vector of regressors.

The above equation is a general expression of auto-regressive with exogeneous inputs (ARX) model type. The parameters θ are linear but the regressors can be in nonlinear form. To estimate parameters of the model, many parametric methods have been developed [34]. According to the characteristic of $\epsilon(t)$, whether uncorrelated or correlated, the ordinary least square method or the instrumental variable method can be applied.

2.5.1 Ordinary Least Square Method

The least square method was originally developed by Gauss and Legendre in the early 19th century. The objective is to minimise the sum of the squared error between the measured output and the predicted output. Taking equation (2.11) as an example:

$$\begin{aligned} Y &= X\theta + \epsilon \\ \hat{Y} &= X\hat{\theta} \end{aligned} \tag{2.13}$$

where θ denotes a $(m+n) \times 1$ vector of true parameters, Y denotes a $N \times 1$ vector of measured output and N denotes the length of the output sequence. $\hat{\theta}$ and $\hat{Y}$ denote the vectors of estimated parameters and the predicted output respectively. The ordinary least square (OLS) method attempts to minimize:

$$\begin{aligned} J(\hat{\theta}) &= \frac{1}{2}\epsilon^T\epsilon \\ &= \frac{1}{2}(Y-\hat{Y})^T(Y-\hat{Y}) \\ &= \frac{1}{2}(Y-X\hat{\theta})^T(Y-X\hat{\theta}) \\ &= \frac{1}{2}(Y^TY - \hat{\theta}^TX^TY - Y^TX\hat{\theta} + \hat{\theta}^TX^TX\hat{\theta}) \end{aligned} \tag{2.14}$$

This function can be optimized by numerical search approaches iteratively or analytically solved as follows:

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta}|_{\theta=\hat{\theta}} &= -X^TY + X^TX\hat{\theta} = 0 \\ X^TX\hat{\theta} &= X^TY \\ \hat{\theta} &= (X^TX)^{-1}X^TY \end{aligned} \tag{2.15}$$

OLS estimators have the properties of unbiased, efficient and consistent estimation. The global minimum of equation (2.14) can be found efficiently and unambiguously. However, the estimation may be inconsistent if the noise ϵ is correlated. Advanced methods based on modifications of the OLS have been developed to overcome the inconsistency problem.

2.5.2 Instrumental Variable Method

Submitting equation (2.15) to (2.13), we obtain:

$$\begin{aligned} \hat{\theta} &= \theta + (X^TX)^{-1}X^T\epsilon \\ &= \theta + (\frac{1}{N}X^TX)^{-1}\frac{1}{N}X^T\epsilon \end{aligned} \tag{2.16}$$

where X^TX/N converge to $E[x(t)^Tx(t)]$ and $X^T\epsilon/N$ converges to $E[x(t)^T\epsilon(t)]$. It is clear that if ϵ is correlated, $E[x(t)^T\epsilon(t)]$ is not zero hence $\hat{\theta}$ will not converge to θ. To solve this

problem, a model of the error can be incorporated so that:

$$\epsilon(t) = C(q)\gamma(t) \tag{2.17}$$

where $C(q)$ is the developed model and $\gamma(t)$ is an uncorrelated error. Another approach is to replace the OLS estimator by an instrumental variable estimator:

$$\hat{\theta}_{IV} = (Z^T X)^{-1} Z^T Y \tag{2.18}$$

where Z is a matrix which is related to X in the sense that

$$\begin{aligned} Z^T X/N &\rightarrow E[z(t)^T x(t)] \\ detE[z(t)^T x(t)] &\neq 0 \\ Z^T \epsilon/N &\rightarrow E[z(t)^T \epsilon(t)] \equiv 0 \end{aligned} \tag{2.19}$$

In practice, the conditions of equation (2.19) are difficult to check and thus consistent estimation cannot be guaranteed in general. However, the matrix Z can be constructed by using the acquired data from an OLS estimated model:

$$\hat{A}(q)\hat{y}(t) = \hat{B}(q)\hat{u}(t) \tag{2.20}$$

The tth row of Z is then given by

$$z(t) = [-\hat{y}(t-1), -\hat{y}(t-2), \ldots, -\hat{y}(t-m), u(t-1), \ldots, u(t-n)] \tag{2.21}$$

2.5.3 Maximum Likelihood Method

Another general method of parameter estimation is the maximum likelihood method (MLE) which is originally developed by R.A. Fisher[70]. By the OLS, the parameter values which produce the most accurate output prediction can be obtained while by the MLE, the obtained parameter values are most likely to generate the observed output data. The MLE is more related to probability theories. For observations $Y = y(1), y(2), ...y(N)$, the joint likelihood function is given by:

$$f(\theta|y(1), y(2), ...y(N)) = f_y(\theta; Y) \tag{2.22}$$

A MLE estimation is achieved by finding values of θ that maximize the likelihood function $f_y(\theta; Y)$ [71]:

$$\hat{\theta}_{ML}(Y) = \arg\max f_y(\theta; Y) \tag{2.23}$$

It should note that under the additional assumption that the errors are normally distributed, the OLS estimator is identical to the maximum likelihood estimator [72].

The MLE estimator possesses statistical properties such as consistency and efficiency. The MLE estimator converges to the true value in probability and it achieves the Cramer-Rao lower bound as the sample size increases to infinity [73]. However, it is claimed have no optimum properties for finite samples because this estimator can be heavily biased with small samples and the likelihood function might be unknown if the samples do not follow a general distribution such as the normal distribution [74].

2.6 Model Structure Selection

2.6.1 Linear Polynomial Model Structure

The discrete-time linear polynomial model has been a popular model structure for discrete systems. A general discrete-time linear polynomial model can be described as:

$$y(t) = G(q)u(t - n_k) + H(q)\epsilon(t) \tag{2.24}$$

where $G(q)$ represents the model of plant, $H(q)$ represents the model of the noise, $\epsilon(t)$ is assumed to be a white noise and n_k is the time delay between inputs and outputs. According to the different possible selections of numerators and denominators in $G(q)$ and $H(q)$, the model structure can be categorized into 4 common types [75].

ARX model

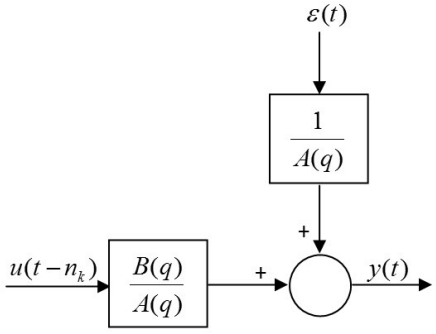


Figure 2.3: Structure of ARX model

The structure of an ARX model is:

$$y(t) = \frac{B(q)}{A(q)}u(t - n_k) + \frac{1}{A(q)}\epsilon(t) \tag{2.25}$$

where

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_{na} q^{-na} \\ B(q) &= b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{nb} q^{-nb+1} \end{aligned} \tag{2.26}$$

The auto-regressive with exogenous inputs model, as shown in Figure 2.3 and equation (2.25) is a simplified case of equation (2.24), where AR represents the term $A(q)y(t)$ and X represents $B(q)u(t)$. The model of the noise does not have any flexible term but is completely determined by the dynamics of the model of the plant. It is suitable for describing a system where the noise is caused by stochastic of the plant.

ARMAX model

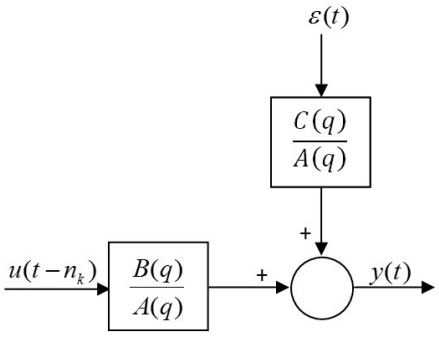


Figure 2.4: Structure of ARMAX model

The structure of an ARMAX model is:

$$y(t) = \frac{B(q)}{A(q)}u(t-n_k) + \frac{C(q)}{A(q)}\epsilon(t) \tag{2.27}$$

where

$$\begin{aligned} A(q) &= 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_{na}q^{-na} \\ B(q) &= b_1q^{-1} + b_2q^{-2} + \ldots + b_{nb}q^{-nb+1} \\ C(q) &= c_1q^{-1} + c_2q^{-2} + \ldots + c_{nc}q^{-nb+1} \end{aligned} \tag{2.28}$$

The autoRegressive moving average with exogeneous inputs (ARMAX) model can be considered as an expansion of the ARX model. An independent polynomial $C(q)$ for the noise, referred to as the MA term is designed for additional flexibility in the noise dynamics. The noise can be used to represent the uncertainty of the plant and input disturbances.

Output Error Model

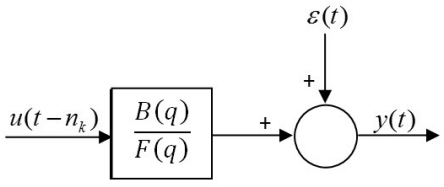


Figure 2.5: Structure of OE model

The output error model structure is:

$$y(t) = \frac{B(q)}{F(q)}u(t-n_k) + \epsilon(t) \tag{2.29}$$

where

$$\begin{aligned} B(q) &= b_1q^{-1} + b_2q^{-2} + \ldots + b_{nb}q^{-nb+1} \\ F(q) &= f_1q^{-1} + f_2q^{-2} + \ldots + f_{nf}q^{-nf+1} \end{aligned} \tag{2.30}$$

ARX and ARMAX models are typical error equation model structures where the transfer function of noise $H(q)$ is affected by the denominator of the plant transfer function. In an output error (OE) model, the noise is directly added to the output without going through the dynamics of the plant and often refers to a pure error in the measurement of output.

Box-Jenkins model

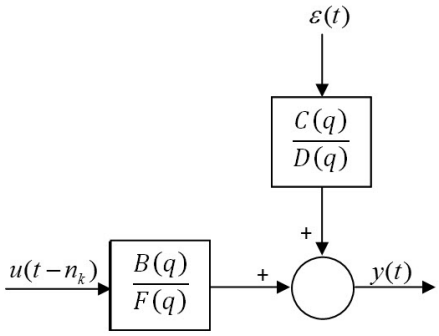


Figure 2.6: Structure of BJ model

The Box-Jenkins model structure is:

$$y(t) = \frac{B(q)}{F(q)}u(t-n_k) + \frac{C(q)}{D(q)}\epsilon(t) \tag{2.31}$$

where

$$\begin{aligned} B(q) &= b_1q^{-1} + b_2q^{-2} + \ldots + b_{nb}q^{-nb+1} \\ C(q) &= c_1q^{-1} + c_2q^{-2} + \ldots + c_{nc}q^{-nc+1} \\ D(q) &= d_1q^{-1} + d_2q^{-2} + \ldots + d_{nd}q^{-nd+1} \\ F(q) &= f_1q^{-1} + f_2q^{-2} + \ldots + f_{nf}q^{-nf+1} \end{aligned} \tag{2.32}$$

Compared to the three types of models above, the Box-Jenkins model has the most complicated structure. It gives more freedom in noise modelling and the models of plant and noise are completely independent.

The above structures of linear models are expressed in the form of transfer function or alternatively described by polynomial regressor as in equation (2.11). The nonlinearity of system therefore can be represented by nonlinear regressors with linear parameters.

2.6.2 Determination of Model Regressors

In nonlinear black box modelling, it is always difficult to determine what regressors should be included in the model structure. Besides trial and error approach, a mathematical method based on hypothesis testing and correlation analysis has been popular. Consider a model:

$$Y = \theta X + \epsilon \tag{2.33}$$

where $Y = [y(1), y(2), \ldots, y(N)]$, $X = [x(1), x(2), \ldots, x(N)]$ and ϵ is assumed to be a white Gaussian noise with zero mean and variance σ^2. To determine the model structure, in the first step it is necessary to construct a pool of candidate regressors, including all linear and nonlinear terms which are to be considered. The correlation between a candidate regressor and measured output is calculated by:

$$r_{jz} = \sum_{t=1}^{N} \frac{[x_j(t) - \bar{x}_j[y(t) - \bar{Y}]]}{\sqrt{S_{jj}S_{yy}}} \tag{2.34}$$

$$\bar{X}_j = \frac{1}{N}\sum_{t=1}^{N} x_j(t) \tag{2.35}$$

$$S_{jj} = \sum_{t=1}^{N}[x_j(t) - \bar{x}_j]^2 \tag{2.36}$$

$$S_{yy} = \sum_{t=1}^{N}[y(t) - \bar{Y}]^2 \tag{2.37}$$

where N is the length of data sequence and x is the candidate. The regressor which has the highest correlation will be added to the model structure and regressor matrix X and the estimated parameter updated correspondingly. The influence of the added regressor will be examined by

$$F = \frac{SS_R(\hat{\theta}_j|\hat{\theta}_m)}{s^2} = \frac{SS_R(\hat{\theta}_{m+j}) - SS_R(\hat{\theta}_m)}{s^2} > F_{in} \tag{2.38}$$

$$SS_R = \sum_{t=1}^{N}[\hat{y}(t) - \bar{Y}]^2 = \hat{\theta}X^TY - N\bar{Y}^2 \tag{2.39}$$

$$s^2 = \frac{1}{N-P}(Y - X\hat{\theta})^T(Y - X\hat{\theta}) \tag{2.40}$$

where $SS_R(\hat{\theta}_{m+j})$ is the sum of the squared variations of the predicted output after the jth regressor is added into the regressor matrix which has m terms. P is the number of regressors in the model including the one added in the current iteration. The new regressor will be accepted, provided $F > F_{in}$, where F_{in} is a value which is predefined according to the desired confidence level. However, because of the relationship between the new regressor and regressors that have already been selected, the importance of each regressor might be

affected. A backward elimination is added in order to reassess the regressors and remove the redundant ones.

$$F = min\frac{SS_R(\hat{\theta}_m) - SS_R(\hat{\theta}_{m-j})}{s^2} < F_{out} \tag{2.41}$$

where $SS_R(\hat{\theta}_{m-j})$ denotes the sum of the squared variations of the predicted output after the jth regressor is removed from the regressor matrix. Then the relevant variable should be updated as follows for the next iteration.

$$x_{j(i+1)} = x_{j(i)} - \hat{\beta}X_i \tag{2.42}$$

$$\hat{\beta} = (X_i^T X_i)^{-1} X_i^T x_{j(i)} \tag{2.43}$$

$$Y_{i+1} = Y_i - \hat{\theta}_{i+1} X_{i+1} \tag{2.44}$$

The whole process continues until no candidate satisfies F_{in} or the predicted output meets the required accuracy.

2.7 Model Validation

With the purpose of measuring the quality of identified model, the model should be tested against various validation signals to find out if it is good enough to describe the real system.

2.7.1 Validation Signals

Basically the validation signal might be selected as the same type of signal as the signal used for the identification. The data collected from the system can be divided and the first half sequence used as an identification signal while the second half as a validation signal. Moreover it is convincing to repeat the validation with signals generated by different seeds. Before applying validation signals to the model, it is beneficial to determine if these validation signals are independent. Correlation refers to a statistical relationship between two sets of data. For a statistically efficient test, the validation test signals should be uncorrelated which indicates there is no tendency for the values of one signal to increase or decrease with the values of the second signal. To test the qualification of validation signals, the correlation coefficient is consequently determined for each pair of signals and is given by:

$$\begin{aligned} r_{U_i,U_j} &= \frac{cov(U_i, U_j)}{\sigma_{Ui}\sigma_{Uj}} \\ &= \frac{E(U_iU_j) - E(U_i)E(U_j)}{\sqrt{E(U_i^2) - E^2(U_i)}\sqrt{E(U_j^2) - E^2(U_j)}} \end{aligned} \tag{2.45}$$

where U_i and U_j, $i \neq j$ are the distinct inputs. Besides evaluating the correlation between validation signals, it is also important to measure the correlation of the signal value at

different sample instants. For dynamic identification, each input is a sequence time series. The correlation of two inner elements with lag l can be calculated by:

$$r_l = \frac{\sum_{t=1}^{N}[u(t) - \bar{U}][u(t-l) - \bar{U}]}{\sum_{t=1}^{N}[u(t) - \bar{U}]^2} \tag{2.46}$$

2.7.2 Validation Criteria

For a white box model where the true model structure and parameters are available, a natural validation is to compare the estimated parameters and their covariance with the values obtained from prior knowledge. Although generally, in the experimental case true model structure and parameters are unknown, the model quality can however be measured by implementing a batch of validation signals to the identified model and checking how well the simulated outputs matches the measured experimental outputs. For this purpose, a scalar function of the error between estimated output and measured output, as a mean squared error (MSE) can be selected.

$$MSE(Y, \hat{Y}) \quad = \quad \frac{||\hat{Y} - Y||^2}{N} \tag{2.47}$$

where $Y = [y(1), y(2)..., y(N)]^T$ is the measured output matrix and $\hat{Y} = [\hat{y}(1), \hat{y}(2)..., \hat{y}(N)]^T$ is the simulated model output. Since the scales of output and corresponding prediction error differ between models, the MSE cannot solely represent the degree of model quality without a comparison to the measured output. Thus a criterion which relates the error to the output of the same model as a percentage is desired to evaluate the quality of model. A multiple correlation coefficient R^2 function is employed to measure the output fitness given by:

$$R^2(Y, \hat{Y}) \quad = \quad 1 - \frac{||\hat{Y} - Y||^2}{||Y - \bar{Y}||^2} \tag{2.48}$$

where $\bar{Y}$ is the mean of Y. If the system is precisely excited which gives $Y \neq \bar{Y}$, the feasible value of R^2 is from $-\infty$ to 1. It is obvious that $R^2 = 1$ indicates a perfect model. It is worth noting that the achievable maximum value of R^2 is dependent upon the specific modelling problem. This fact means that there is no universal value of R^2 which is considered to be acceptable and so the measure of fit is relative and an acceptance criteria must be judged on a case by case basis.

Besides the MSE and R^2, the final prediction error (FPE), Akaike's information criterion (AIC) and Bayesian information criterion (BIC) are also widely used in system identification

and they are given by [76]:

$$FPE = N \cdot \ln(MSE) + N \cdot \ln\left[(N + n_\theta)/(N - n_\theta)\right] \quad (2.49)$$

$$AIC(\rho) = N \cdot \ln(MSE) + \rho \cdot n_\theta; \rho > 0 \quad (2.50)$$

$$BIC = N \cdot \ln(MSE) + n_\theta \cdot \ln(N) \quad (2.51)$$

where N denotes the length of the data sequence, n_θ denotes the number of regressors in the model and ρ denotes the weighting factor. The FPE, AIC and BIC criteria are closely related. They can be used to evaluate the quality of the parameterized model by means of measuring the output prediction error. The complexity of the model, which refers to the number of the regressors in the model, is taken into account by these criteria and a model with less regressors is considered having better quality since in practice the engineers usually prefer a model of the system which is as simple as possible.

The criteria of R^2, FPE, AIC and BIC are all related to the MSE. The advantage of R^2 is that it can represent the quality of model in percentage since the prediction error is compared with the variance of measured output. However in the process of model structure selection, the model which includes all possible regressors generally gives the best R^2. To overcome this problem, the FPE, AIC and BIC criteria can be employed since they penalize the complexity of the model.

In the experiment design, the effects of model structure selection, input design and parameter estimation on the model quality are interrelated. Since the optimal input design and parameter estimation are studied in this thesis, the model structure is fixed in order to eliminate the influence from model structure selection. Therefore the MSE and R^2 criteria are selected to validate the estimated models in this work.

2.8 Artificial Neural Networks

The term neural network (NN) originally refers to a network of biological neurons that are linked together to realize a specific biological function. The artificial neural network is used to present a mathematical model composed of artificial neurons. Similar to the biological NN which can perform various physiological behaviours, the artificial NN is capable of representing very complex nonlinear models. In recent years the NN has been successfully used in model fitting, clustering and pattern recognition. Currently its implementation in dynamic modelling of automobile systems is being extensively investigated [77, 78].

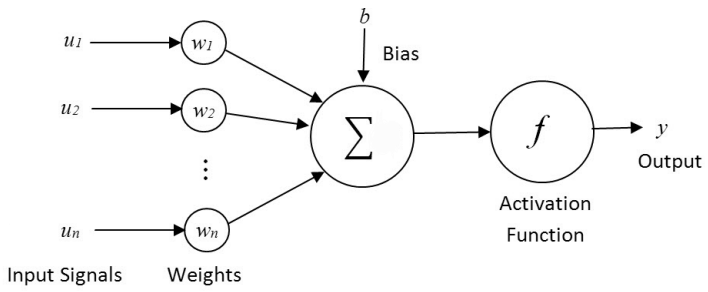


Figure 2.7: Schematic of a neuron

2.8.1 Structure Selection of NN Models

Neuron

Conventional polynomial models are composed of regressors and parameters while in NN models the neuron is the basic component. Figure 2.7 shows a general construction of a neuron. Each channel of the input signals is weighted and summed then a bias is added to the product which is used to feed an activation function. The output of the neuron can be expressed by the equation:

$$y = f\left(\sum_{i=1}^{n}(w_i u_i) + b\right) \tag{2.52}$$

Layer

As shown in Figure 2.8, the simple neurons are linked in parallel to form a layer which is able to represent more nonlinearities. By using the output of the current layer as the input of the next layer, more layers can be added and a comprehensive network is formed accordingly. The output of a multi-layer NN model is given by:

$$y = f_k\left(W_k f_{k-1}\left(W_{k-1} f_{k-2}\left(\ldots f_1\left(W_1 u + b_1\right) + \ldots + b_{k-2}\right) + b_{k-1}\right) + b_k\right) \tag{2.53}$$

The last layer is named the output layer and the other layers are named hidden layers.

To identify a NN model, the model structure should be determined firstly. Basically the number of layers, the number of neurons in each layer and the type of activation function should be pre-determined and for specific types of models such as time series models, the time delay of input and output should also be given. As mentioned by Cybenko [79], a NN model with one hidden layer can represent most systems if sufficient neurons and testing time is available. The purpose of adding more hidden layers is usually for a quick convergence. However according to Priddy [80], designers should try to build NN models within one or two

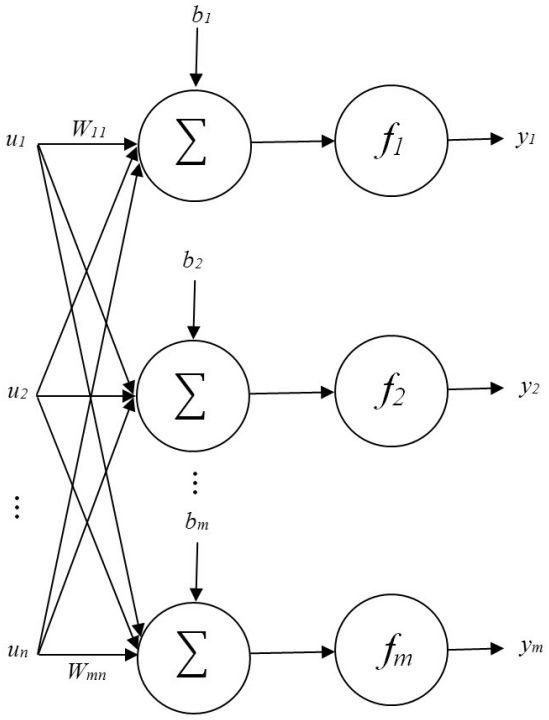


Figure 2.8: Schematic of a single layer

hidden layers since the best performance of training, in other words the best learning, can be obtained when the model is the simplest. He also suggested that the number of neurons should be minimized and the optimal number could be determined by the validation results. A list of common activation functions is given in [81], the selection of activation function is affected by the pre-knowledge of the system and the intended training algorithm. For instance the linear activation function is less useful if the system is expected to be nonlinear and the function must be differentiable if backpropagation algorithms are employed.

2.8.2 Training, Validation and Testing

The term "training" in NN models has a similar meaning to "identification", or more specifically "parameter estimation" in polynomial models. It refers to the process of adjusting the weights and biases of neurons to give the best output performance. In general, training methods can be classified into: supervised and unsupervised learning methods. The supervised method, e.g. backpropagation method, utilizes the error between the desired output and the model simulated output to adapt the weights and bias until a stopping criterion is satisfied. On the other hand, the desired output is not available in the unsupervised learning method. The model output is fed to an adaptation function which represents a general behaviour and the NN model is adjusted according to the output of the adaptation function. Practically

most NN models can be trained by supervised methods and the applications of unsupervised methods are mainly in self-organized map and adaptive resonance theory [82, 83].

The obtained NN models by training may fit the training data very well, however they might not be able to simulate the unseen data accurately and this problem is called "over-fitting". "Over-fitting" generally occurs if there are too many parameters in the model while the data length for identification is relatively short. To evaluate whether a model is "over-fitting", another set of data can be used for validation, e.g. dividing the whole data into two parts for identification and validation respectively. If the model fits the validation data to the same degree as fitting the identification data, this model is considered not "over-fitting".

The definition of validation data in NN models is different from that in polynomial models. In the training of NN models, multiple candidates of trained networks could be obtained. Validation data is used to find the best network which minimizes the error testing against the validation set. In other words, the validation does not further adjust the model structure but only verifies that any increase in accuracy over the training data set actually causes an increase in accuracy over an unseen data set. Therefore the validation error can be used as a stop criterion for training to prevent "over-fitting" of the training data [84].

The chosen network is eventually assessed by test data and its performance reported. To ensure the generalization of the NN model, the test data should not have high independence of the training and validation data for an unbiased estimation [85].

2.9 Conclusions

In this chapter, we present a detailed literature review on system identification and validation. The general procedure of system modelling and relevant DoE methodologies are introduced. The information theory, various optimal criteria and established research on these criteria are studied. Popular structures for polynomial models, methods for model structure selection and parameter estimation are also reviewed. It is found that the input selection, model structure selection and parameter estimation have significant influences on the model quality therefore the optimal input design and estimation method for simulation model are selected as the research interests of this thesis and are discussed in later chapters.

Moreover, the differences between prediction models and simulation models and various validation criteria for evaluating the model accuracy are also introduced to support the selection of model type and validation criteria in the following chapters. Basic features of artificial Neural Network are discussed and the performances of polynomial models and NN on engine modelling are compared in the chapter of dynamic model-based calibration.

Chapter 3

Experimental Setup

3.1 Introduction

The experimental work in this thesis includes experiments on a real engine which is connected to a dynamometer and a virtual engine which is presented by a WAVE-RT model. The methodology of optimal test signal design and the simulation error method is developed based on the real engine. The identification and control methodologies of dynamic calibration developed in this thesis are intended to be applicable to real engine hardware. However for the development of these techniques an engine simulation package is used. This has the advantage of all engine simlators in that, it reduces experimentation cost, is repeatable and unaffected by any external disturbance, such as humidity, atmosphere pressure and temperature, which in the real engine experiment could compromise the results. To make the experiments repeatable, simulation models built by Ricardo 1D WAVE software are used instead of the real engine. A WAVE model of an EcoBoost 2.0-Litre GTDI engine was provided by the Ford Moter Company. Engineers at Ford have been using this model, as a replacement for the real engine, for initial stage tests of some developed control methods. Appropriately designed and validated WAVE models are recognized as giving simulation results with dynamics which closely match those of the real engine. For this reason the effectiveness of the optimal input design and dynamic calibration is examined by WAVE.

In the first two sections of this chapter, the characteristics of the 1.6 Litre Zetec engine and related software and hardware configurations for the experiments are introduced. A WAVE model of a 2.0 Litre GTDI engine is then presented and a specification of the virtual engine and relevant components is given. Section 5 discusses the procedure of adapting the WAVE model for close-to-real-time applications in the Mathworks Simulink environment. This, real-time (RT) model is further modified in order to meet experimental requirements. Essential actuators and sensors of both real engine and virtual engine are introduced in section 6 and a road load model for determining a speed profile is then devlopedd.

3.2 Real Engine Specification

Table 3.1: Specification of Zetec 1.6L real engine

Number of cylinders	4
Strokes per cycle	4
Engine type	Spark ignition
Cylinder bore	76mm
Stroke length	88mm
Connecting Rod Length	136.2mm
Compression Ratio	10.3mm
Maximum torque	138Nm at 3500 RPM
Maximum power	67kW
Idle speed	880 RPM

The experimental engine in the University of Liverpool powertrain control lab is a conventional port fuel injection gasoline spark ignition Ford 1.6 Litre Zetec engine as specified in Table 3.1. In low-speed low-load experiments, the throttle position is fixed and the air is delivered by air bleed valve (ABV). Each cylinder has two intake valves and two exhaust valves and the valve timing is controlled by dual overhead camshafts. The electronic port fuel injectors thus inject the fuel before the opening of intake valves. The EMS is a control unit for air delivery, fuel timing and spark timing. In production vehicles, pre-defined control strategies are saved in EMS. For this thesis the engine is modified so that the designed control signals can be transferred from software and hardware interface to the engine directly.

3.3 Real Engine Experiment Configuration

Figure 3.1 illustrates the configuration of the engine and its related instrumentation. The crankshaft of the engine is coupled to a low inertia DC electric generator engine dynamometer for measuring the engine torque and power. In experiments of torque control, the dynamometer often acts like an extra load for absorbing power generated by the engine and the amount of load produced from the dynamometer is regulated by a voltage control signal.

In this thesis, signals recorded by sensors on the real engine are sampled every degree, in other words the data collection is crank angle based. The crank angle of the engine is measured by the angle encoder located on the crankshaft. The encoder generates a pulse every 1 degree and the pulse is then delivered to the D-space and triggers the data collection in D-space tasks. The frequency of collection can be multiples of every 1 degree which allows for down sampling the data into engine events such as every stroke or engine cycle. The encoder also generates a pulse every 360 degree. The purpose of the 360 degree pulse is to ensure no missing of 1 degree pulse occurs and to reset the crank angle if any inconsistency

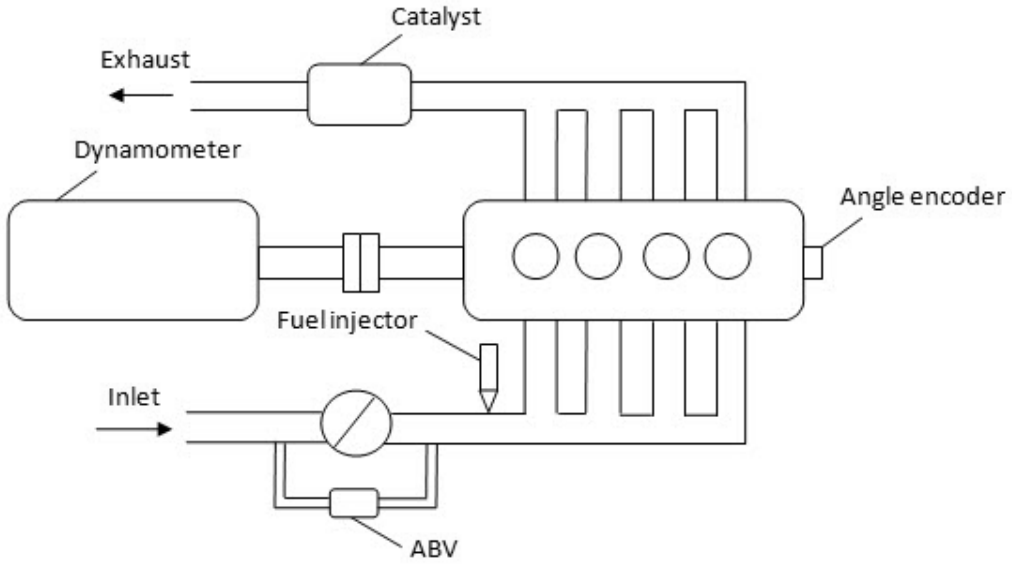


Figure 3.1: A schematic of the engine setup and key instrumentation

occurs. A schematic diagram of the hardware and software configuration for the engine experiments is shown in Figure 3.2. Originally, engine outputs such as engine speed, AFR and temperature are delivered to the EMS which is in charge of controlling all engine inputs such as ABV and FPW according to the embedded control strategy. In engine experiments, the outputs can alternatively be transmitted to a D-space unit. The D-space will take over the control authority of any interested parameters from the EMS and the Power stage is used as an electronic amplifier to boost control signals from D-space in order to power the corresponding inputs. The D-space hardware cooperates with a PC for data processing. The Control Desk software running on the PC is interface software used for date logging and real time monitoring. Real time engine data can be recorded and converted into MAT files which are readable by MATLAB. MATLAB/SIMULINK with the Real-Time workshop add-on package is used to develop controller and test signals by analysing the data offline and then to generate models for implementation. The established models are complied into C code and applied to Control Desk. By building a proper layout for the complied SIMULINK model, online graphical control and monitoring of the engine can be achieved by Control Desk.

3.4 WAVE Virtual Engine

Ricardo WAVE is a commercial software package which principally evaluates 1D flow to simulate and analyse the system behaviour, such as air-path dynamics, fuel injection mass, manifold pressure and piston position in the engine and related parts. It provides a fully

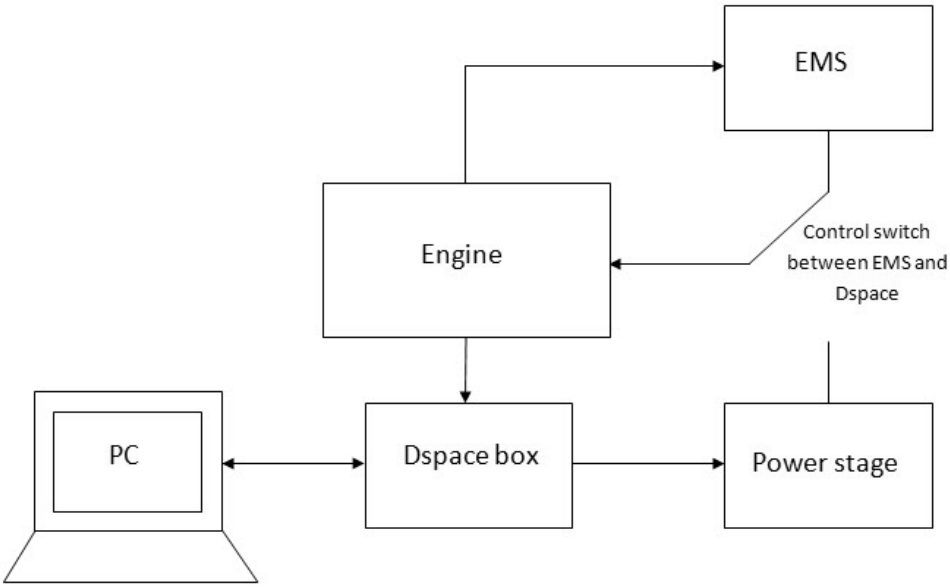


Figure 3.2: Hardware and software configuration of engine experiments

integrated treatment of time-dependent fluid dynamics and thermodynamics by means of a one-dimensional formulation which enables performance simulations to be carried out based on virtually any intake, combustion and exhaust system configuration.

Figure 3.3 shows an example of simulating a single cylinder system by WAVE. The components of the system at the top are modelled and connected as the block diagram in the middle. All parameters such as initial condition and geometry can be set up accurately by element panels at the bottom.

Table 3.2: Specification of GTDI 2.0L virtual engine

Number of cylinders	4
Strokes per cycle	4
Engine type	Spark ignition
Cylinder bore	87.5mm
Stroke length	83.2mm
Clearance height	0.5mm
Piston surface area	6448.89mm^2
Connecting rod length	156.6mm
Compression ratio	9.9
Wrist pin offset	0.8mm

A WAVE model of the 2.0 Litre GTDI Ford engine was provided by the Sponsoring Company (Ford Motor Company) and it is illustrated in Figure 3.4. This is a pressure charged, 4 cylinder, 4 stroke, 16 valve spark ignition engine. A geometric specification of the engine is

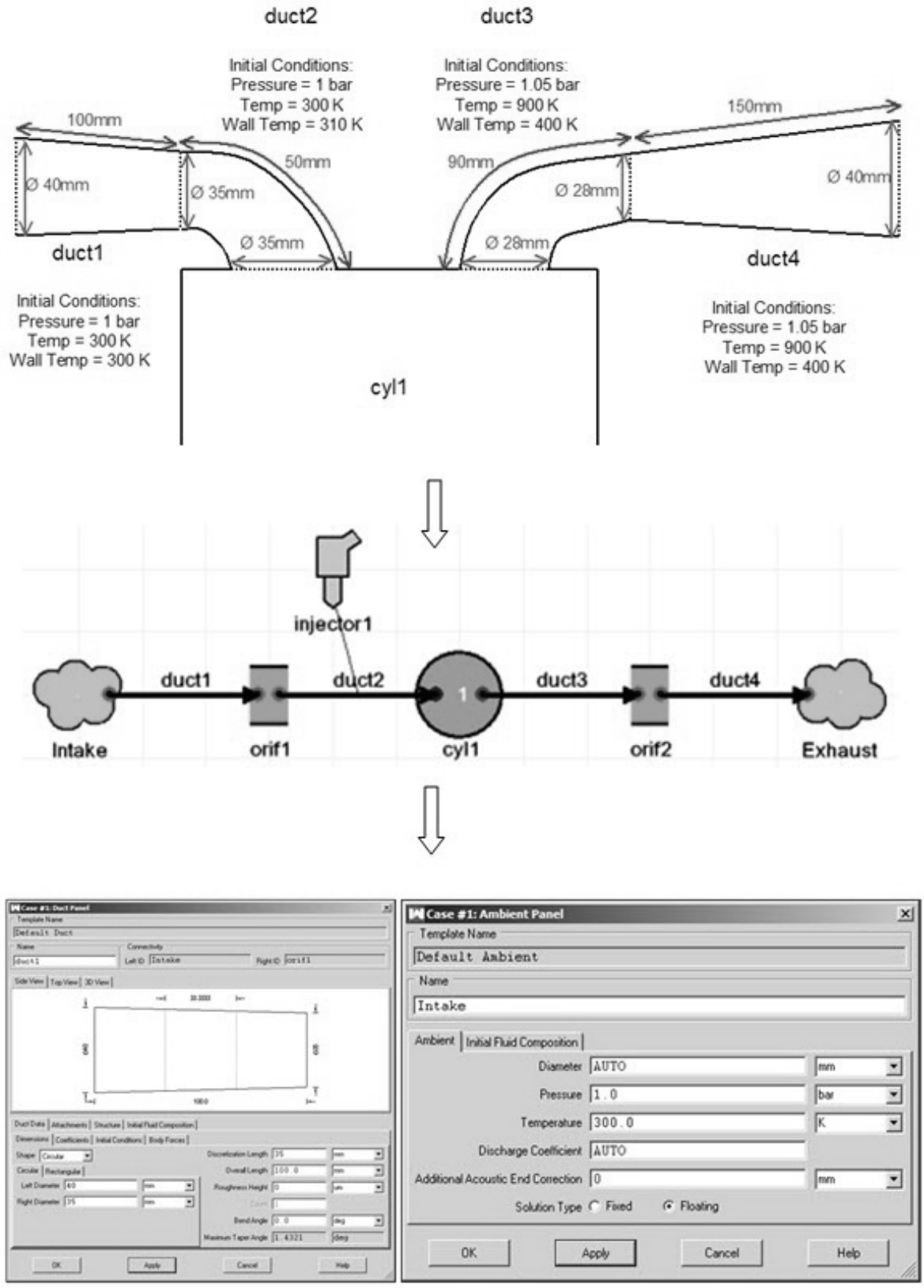


Figure 3.3: An example of simulating a cylinder by WAVE

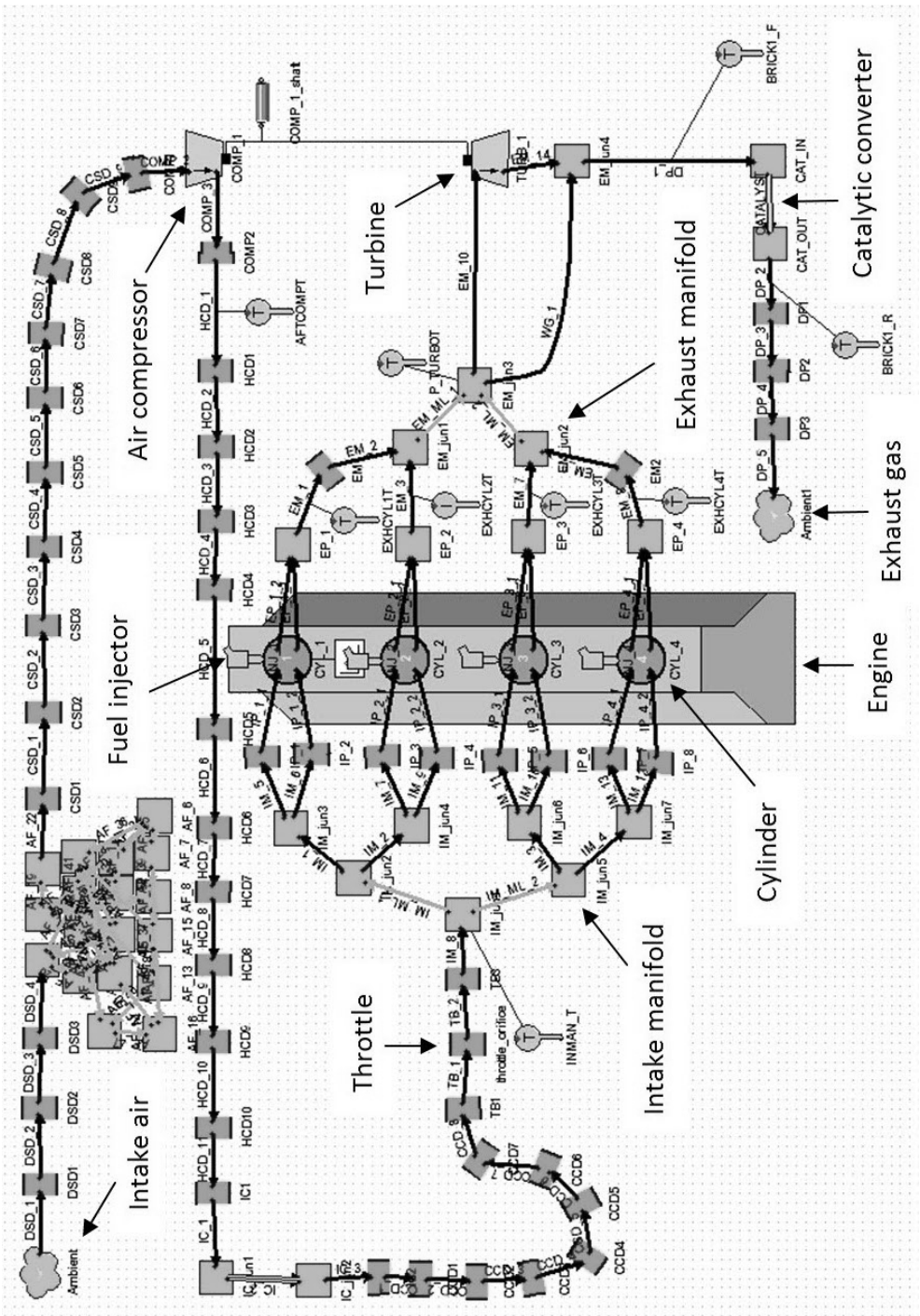


Figure 3.4: WAVE virtual engine

given in Table 3.2. The air enters through a filter which prevents solid particulates going into the engine, where it may cause mechanical wear and oil contamination. The compressor then increases the density of the charge which allows for higher volumetric efficiency for a given engine size. This allows higher peak torque and power to be achieved, whilst benefiting from increased fuel economy at part load conditions. The compressed air is next cooled and then goes through the throttle to the intake manifold. The engine is equipped with a high pressure common rail, direct injection system which simulates the gasoline being injected directly into the in-cylinder air-charge. On the exhaust side of the engine, the turbine generates boost by using the high temperature exhausted gas which improves the thermodynamic efficiency of the engine. Finally, a catalytic converter will convert the toxic by-products in exhaust gas to less toxic substances.

Besides the general engine geometric model, sub-models such as heat transfer, conduction and combustion should also be defined adequately. The combustion sub-model of the provided 2.0L virtual engine is a SI Wiebe model as shown in Figure 3.5. The rate of fuel mass burned in thermodynamic calculations is described by the SI Wiebe function [86]. This type of combustion model is designed by Ricardo and the required combustion parameters of this virtual engine are provided by Ford. The combustion model for the 2.0L engine was not available for reasons of commercial confidentiality and so combustion data for a similar 3.0L engine was obtained and used. The related parameters were provided including the combustion duration and location of 50% burn point. The mass of fuel left in the cylinder was calculated by a simple S-curve function and its burning rate represented as the first derivative of this function [86].

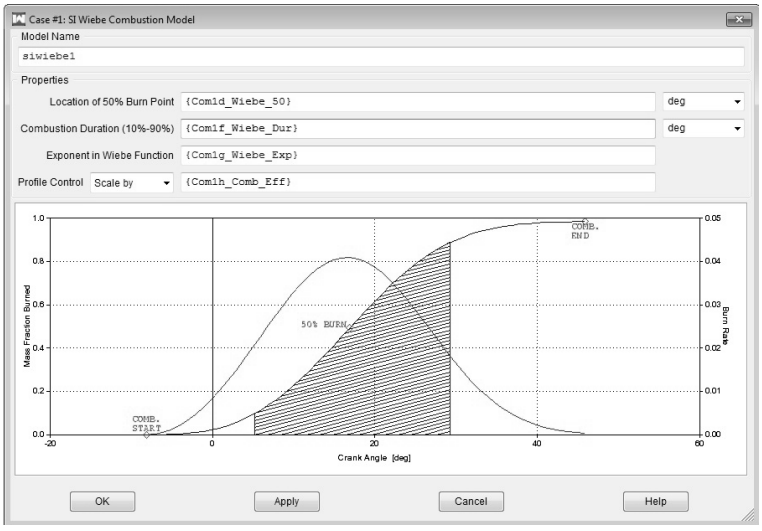


Figure 3.5: SI Wiebe combustion model

In real engine experiments, the stochastic behaviour of variables such as temperature,

pressure, moisture and even errors of the actuators and sensors will compromise the results and make experiments unrepeatable. In the WAVE model, however, not only the environmental parameters are pre-determined but also the sub-models are completely regulated. The entire operating space is categorized into cases and saved in a constant table. The WAVE model is always running under a specific case where all related parameters have been regulated.

3.5 WAVE-RT Model

WAVE-RT is a simplified real-time simulation software version of WAVE which provides a useful interface for connecting between WAVE and conventional control system developing packages, such as MATLAB/Simulink. Besides compiling the geometry from WAVE, the operating parameters and environmental parameters are also compiled appropriately as well. However in practice it is necessary to be able to adjust these parameters in WAVE-RT according to different experimental conditions. In order to do that, actuators should be placed on variables that need to be controlled and the corresponding responses observed by sensors, as shown in Figure 3.6. In the compilation procedure, firstly the WAVE model is converted to C code which includes all necessary information from the WAVE model, such as environmental parameters, actuator and sensor characteristics and details of components. Users can modify the content of C code directly rather than the WAVE model. A WAVE-RT block in SIMULINK will be appointed to the C code and present actuators and sensors as an input-output block, as shown in Figure 3.7.

Figure 3.8 illustrates a fully developed Simulink WAVE-RT model by Ford for this thesis. Besides the main RT block which is directly compiled from the provided Ricardo WAVE model, other sub-models should be designed and cooperated in order to simulate engine behaviours precisely in different operating regions. In the WAVE model, the inputs of the spark advance (SA), burn duration and Wiebe exponent are chosen to simulate the fuel combustion in the cylinders. The SA affects the combustion phasing and can be controlled independently in the RT model. As suggested by Ford, the Wiebe exponent can be fixed at 2.5 for low-load low-speed work. However since the burn duration is causal, it should be determined by a function of speed, load, SA and valve phasing. A sub-model for the burn duration is thus provided by Ford to generate a sensible input to the main RT block. The units of input and output are also converted appropriately.

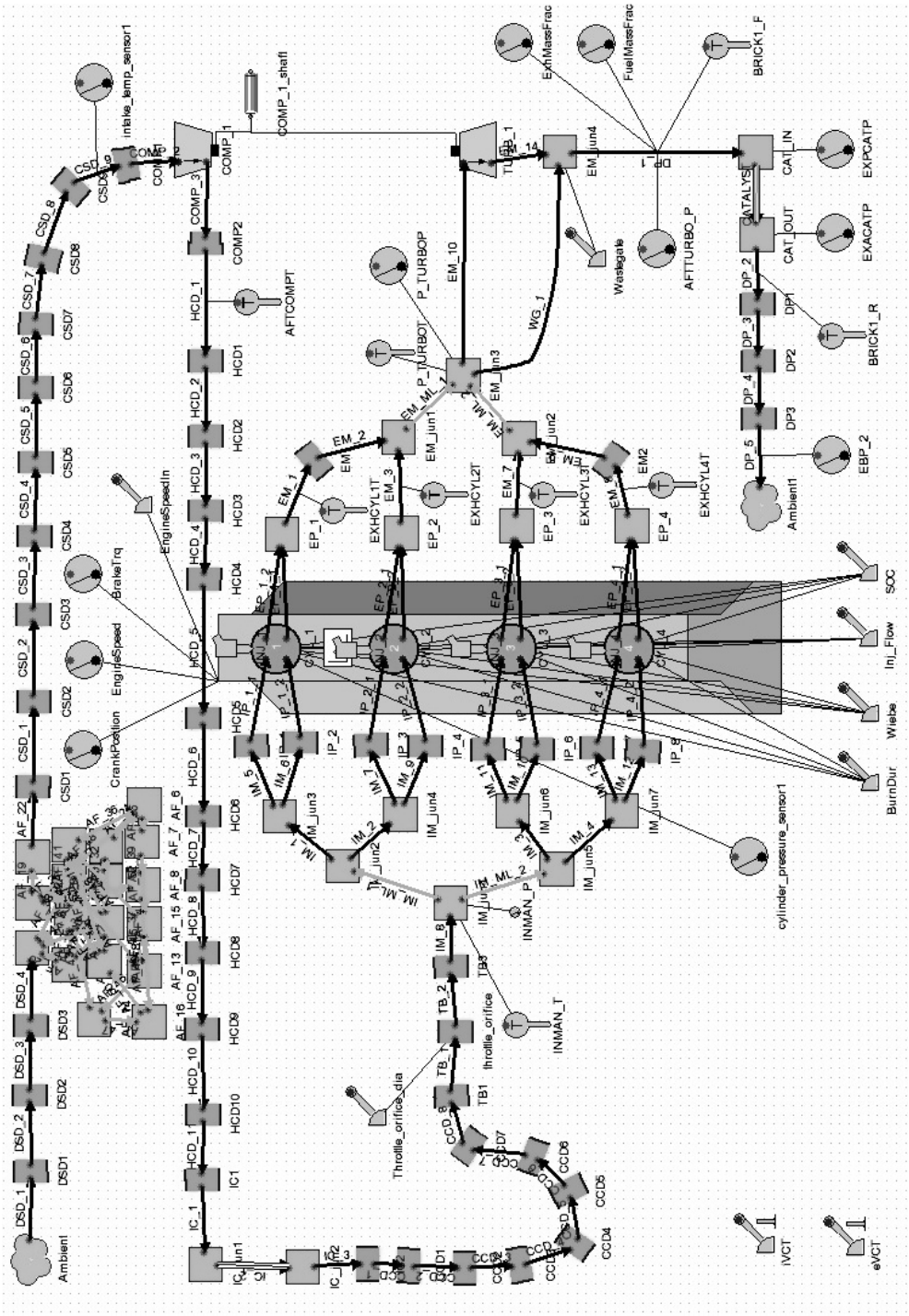


Figure 3.6: WAVE virtual engine with sensors and actuators

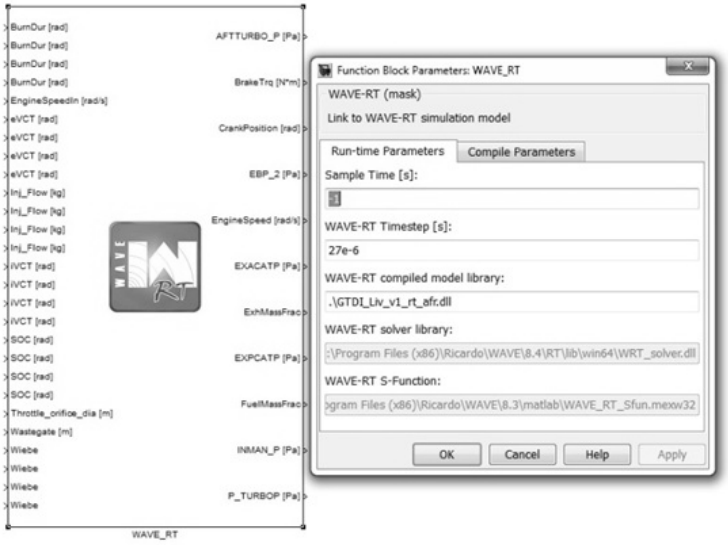


Figure 3.7: WAVE-RT block

3.6 Actuators and Sensors

Throttle Position Actuator

In the IC engine, the throttle position actuator refers to a valve that is located before the intake manifold. The throttle valve directly controls the amount of air going into the intake manifold and has an indirect influence on the engine torque and air-fuel-ratio. The throttle valve in WAVE is modelled as an orifice with an adjustable diameter. A sub-model is used in Simulink to relate the angle of the throttle butterfly valve to a representative orifice diameter.

ABV Actuator

Besides the throttle, the ABV is an alternative path for inlet air flow in the 1.6 Litre engine. The unexpected transient air dynamics resulting from drastic change of throttle position can be eliminated quickly by means of adjusting the ABV. In low-speed low-load Zetec engine experiments, the ABV has a large authority in regulating the amount of inlet air flow whereas the throttle position cannot be electronically controlled.

Spark Advance Actuator

Spark-ignition timing is a crucial factor of engine performance. Inappropriate spark timing will not only affect the fuel consumption and emissions but also bring noise and vibration to the engine. At the end of the compression stroke, the spark advance actuator will control the

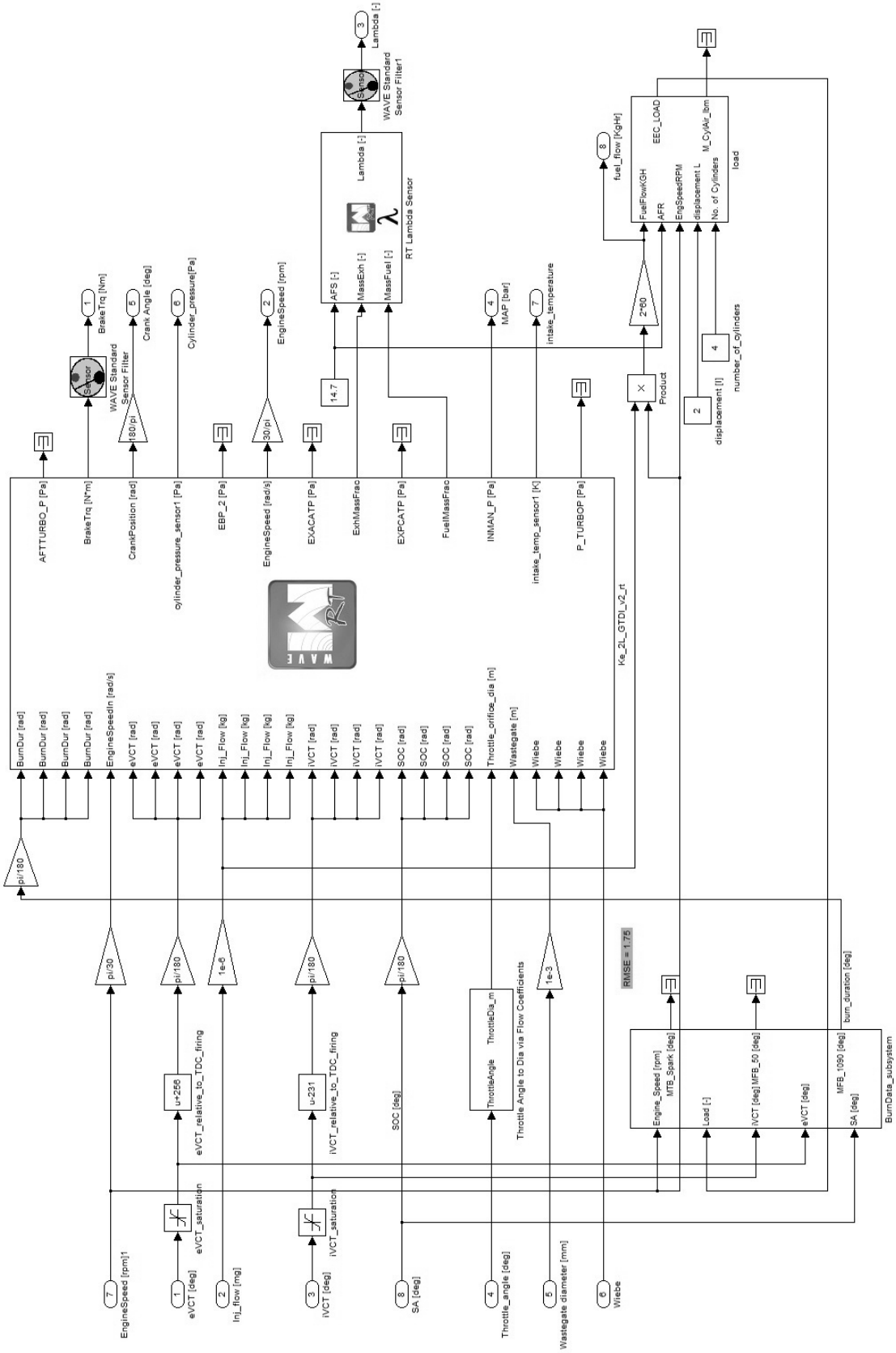


Figure 3.8: Adapted WAVE-RT model of the virtual engine

angle relative to top dead center where the ignition occurs in order to optimize the behaviour of the engine.

Fuel Injection Actuator

The fuel injection event occurs once every 4 strokes per cylinder in the 1.6 Litre and 2.0 Litre engine. The actuator regulates the amount of fuel sprayed into the cylinder in each injection event. Specifically, there are two variables which can be controlled: start of injection and fuel pulse width (FPW). The first variable can be determined by the crank angle when the injection is started and the second variable by the length of time the injector stays open from the injection start angle. In the real engine experiment, the FPW is under control so as to regulate the mass of fuel injected, whilst the fuel mass in each injection can be directly adjusted in the virtual engine.

Engine Speed Actuator/Sensor

In real engine experiments, the engine is typically coupled to a dynamometer. Various engine speeds can be achieved by controlling the load generated by the dynamometer. In the WAVE-RT model, the simulated engine will be considered as connected to a dynamometer with no dynamics which therefore produces desired engine speed instantaneously. Therefore, for transient simulations the user needs to ensure that an appropriate speed profile is input to the model, otherwise unrealistic loading and speed dynamics can result. One particular benefit of testing at fixed engine speeds is that it is very convenient for developing static maps since the engine speed is often an index of the operating space. For simulations representing the engine in a vehicle, the engine speed is a result of the engine inputs, any braking and the properties of the vehicle, such as the mass, inertia and the road conditions. Therefore for calibration purposes a speed profile based on these parameters can be considered appropriate, and engine accelerations faster than the vehicle can be neglected. A road load model which is used to generate sensible speed profiles will be introduced in the later section.

Intake and Exhaust Valve Actuator

The intake and exhaust valve timing is decided by the camshaft phasing. In the 1.6 Litre Zetec engine, the phase of the camshaft is uncontrollable. The intake valve opens and closes at 22 degree BTDC and 12 ATDC while the exhaust valve opens and closes at 64 degree BTDC and 12 degree BTDC. Twin Independent Variable Camshaft Timing (VCT) is a feature of the 2.0 Litre GTDI engine. The VCT changes the valve timing by rotating the camshaft slightly from its initial orientation, which results in the camshaft timing being advanced or retarded.

The camshaft timing is adjusted depending on factors such as engine load and engine speed. This technology is applied to both intake and exhaust valve independently. It allows for more optimum engine performance, reduced emissions, and increased fuel efficiency compared to engines with fixed camshafts.

Wiebe Actuator and Burn Duration Actuator

The Wiebe exponent and burn duration are essential parameters of a combustion model. Similarly to engine speed in the WAVE-RT model, the user of the model should provide reasonable inputs of these two parameters in order to prevent infeasible in-cylinder combustion. As discussed above, the burn duration is determined by a combustion sub-model and the Wiebe exponent can be selected from a speed-load table supported by Ford.

Waste Gate Actuator

A waste gate is a valve that regulates the amount of exhaust gas which enters the turbocharger, which in turn controls the resulting boost. The boost varies with the pressure and temperature of the exhaust gas which is related to the engine speed. As an engine can only accommodate a given amount of boost, this valve should thus be adjusted according to the manifold pressure. At higher boost the wastegate will be opened wider in order to divert more of the gases away from the turbine. Two further constraints relate to the maximum turbocharger speed and preventing compressor surge. At part load operation the wastegate valve can be fully opened to simulate conditions closer to a normal aspirated engine.

Engine Torque Sensor

As a basic specification of an engine, engine torque represents the power that is transmitted from the engine to the car, to produce the acceleration. In real engine experiments, the engine is coupled to a dynamometer then the instantaneous shaft torque can be measured. However, in every engine cycle, the instantaneous torque will reach the peak in the combustion stroke but be used in other strokes in order to move the piston. For calibration and control work in this thesis, only the average shaft torque is of interest and therefore it is necessary to filter this signal, though the peak torque could be of interest in an engine calibration if maximum instantaneous torque loads are constraints.

AFR Sensor

AFR is the proportion of mass of air to mass of fuel in the mixture, which is important in the amount of oxidation in the combustion of the fuel. The sensor is located before the catalytic converter. In practice, the sensor will measure the oxygen or hydrocarbon in the residue mixture and calculate the mass of air and fuel accordingly. The mixture is called stoichiometric if the fuel is burned completely with all the oxygen in air. The stoichiometric AFR is 14.7:1 for gasoline but only feasible in an ideal situation. Since the value of stoichiometric AFR is different by the types of fuel, a relative measurement of AFR is commonly used:

$$\lambda = \frac{AFR_{measured}}{AFR_{stoichiometric}} \tag{3.1}$$

where $\lambda > 1$ represents a lean combustion and $\lambda < 1$ represents a rich combustion. The measured AFR signal should also be filtered since it varies largely during the 4-stroke cycle and only the mean value is necessary for determining catalytic converter performance.

Manifold Absolute Pressure Sensor

The manifold absolute pressure (MAP) is a basic measurement for fuelling control. It indicates the pressure of the air in the intake manifold and when coupled with the engine speed can be used to estimate the air-charge entering the cylinders. The sensor responds very quickly to changes in the air pressure therefore it proves an informative signal. When the MAP and engine speed are provided, the ECU will in turn adjust the amount of intake air by throttle and determine the optimum fuel enrichment for combustion.

3.7 Road Load Model

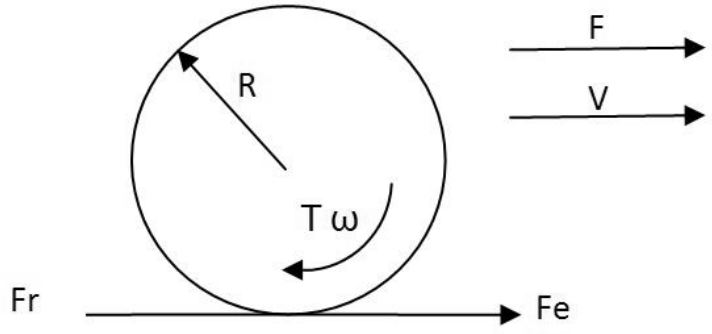


Figure 3.9: Forces on a wheel in motion

Figure 3.9 shows the wheel of a car in motion with driving force F and instantaneous

Table 3.3: Parameters derived by Mondao vehicle experiments

Vehicle mass	1200kg
Tyre radius	0.3m
Vehicle inertia	108kg·m^2
A	34.7
B	0.289
C	0.01705
Basic gear	4.06:1
1st gear	3.417:1
2nd gear	2.136:1
3nd gear	1.448:1
4nd gear	1.028:1
5nd gear	0.767:1

velocity V, where F is determined by the force generated by the engine F_e and resistance F_r. There are various types of resistance but these can generally be categorized into forces that are dependent or independent of the velocity. The resistance could be approximated by the empirically determined function by [87]:

$$F_r = A + Bv + Cv^2 \tag{3.2}$$

Assuming that F_e and F_r are both applied to the wheel, the torque generated by the engine T_e, and the engine rotational speed ω_e can be converted to torque on the wheel T_w and wheel rotational speed ω_w by:

$$T_w = T_e G_b G \tag{3.3}$$

$$\omega_w = \frac{\omega_e}{G_b G} \tag{3.4}$$

where G_b denotes the basic gear ratio and G denotes the selected gear ratio. Accordingly the equation of motion can be expressed as:

$$\begin{aligned} J\dot{\omega}_w &= T_w - F_r R \\ &= T_w - (AR + B\omega_w R^2 + C\omega_w^3 R^2) \end{aligned} \tag{3.5}$$

A road load model is then designed based on the equations above, with an additional system which selects the gear automatically according to the current vehicle speed. This gives a representative speed profile of an engine in vehicle and is implemented for dynamical calibrations in a later chapter. All required parameters are obtained from Mondeo vehicle experiments in the Powertrain laboratory and listed in Table 3.3.

Figure 3.10 illustrates the constructed Simulink model of the auto-gear selection subsystem. The generated engine torque is transferred into the torque on the wheel according to equation (3.3) and the angular velocity of the wheel is obtained after the integrator according

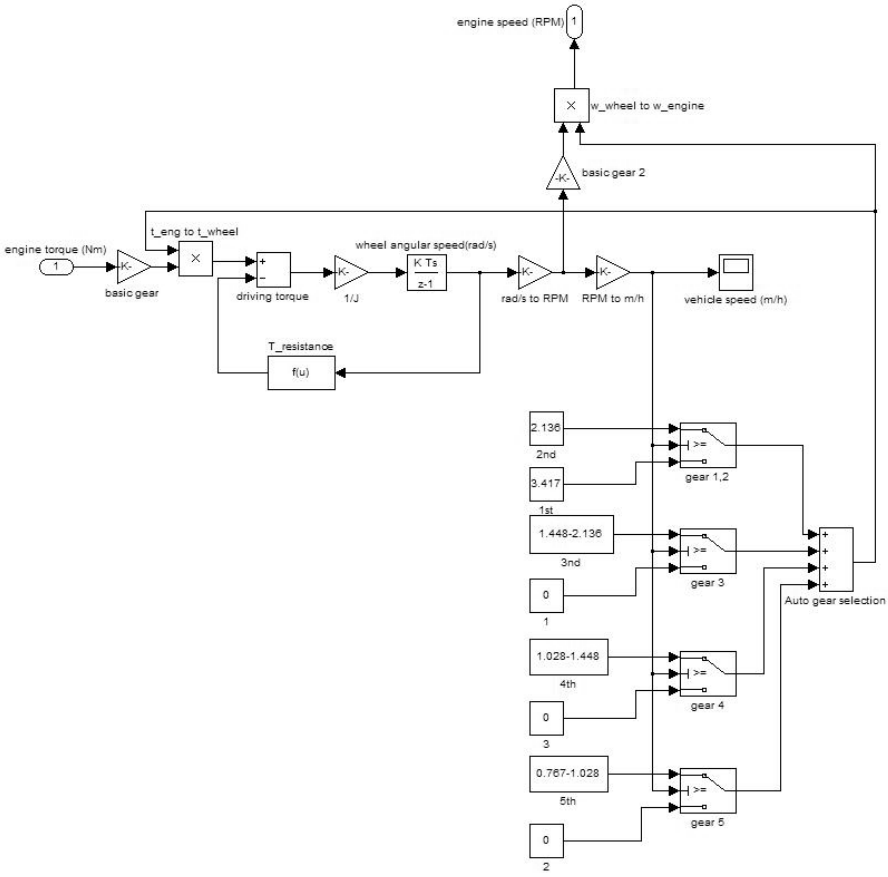


Figure 3.10: Simulink model of the autogear subsystem

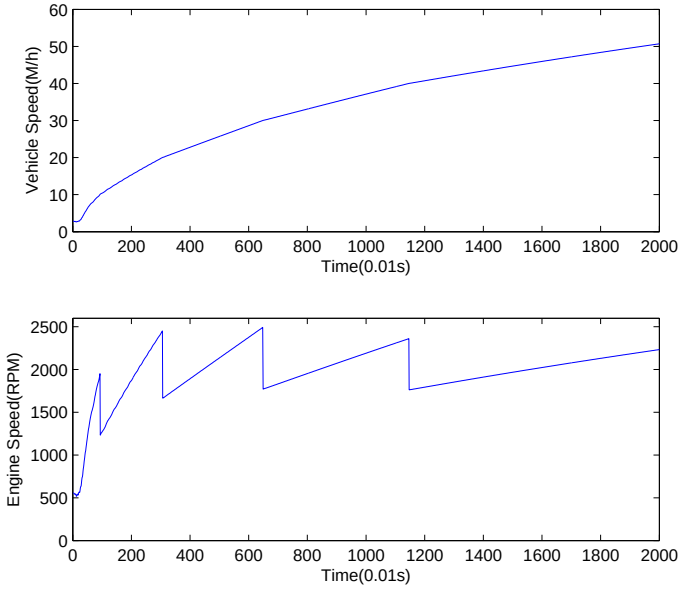


Figure 3.11: Simulated vehicle speed and engine speed in acceleration

to equation 3.5 and then converted to vehicle speed. The auto-gear selection is carried out by comparing the current vehicle speed to change speed from 10 mph to 50 mph. A profile of simulated engine speed and vehicle speed during the acceleration are demonstrated in Figure 3.11. With an engine torque provided as 100 Nm, the vehicle speed increases from 0 to 50 mph within 20 sec. The engine speed increases with the vehicle speed but experiences a drastic reduction when the gear is switched to a higher gear which is due to the instantaneous change of the gear ratio in equation (3.4). The acceleration of the vehicle decreases when a higher gear is selected. The results are sensible because from equation (3.3) if the engine torque remains constant, the torque provided at the wheel changes proportionally to the selected gear.

3.8 Methodology and Research Plan

The proposed methodologies in the following chapters are evaluated by engine experiments. In Chapter 4, the method of optimal input design is firstly tested on the known systems that are obtained from the 1.6 Liter Zetec real engine experiments such as the torque model and AFR model. Non-optimal inputs and optimal inputs designed by conventional design criteria and the proposed new criterion are applied to the known systems with constraints and then the identification results e.g. the estimated parameters or the predicted output are compared to the true values of the known systems in order to validate the improvement on model accuracy. On the other hand, the 2.0 Liter virtual engine is selected as an unknown system and an initial model is obtained by system identification using non-optimal inputs. The optimal inputs designed based on the initial model are applied to the virtual engine and an updated model is developed accordingly. By comparing the accuracy of the models obtained in the first and second iteration, the effectiveness of the optimal input design for the identification of black box systems can be proved.

In Chapter 5, the performance of the proposed simulated error based estimation method is examined by the identification of simulation models for the virtual engine. The data for identification is collected from the virtual engine and the traditional prediction error method and proposed simulation error method are employed respectively for the parameter estimation with the same model structure. The superior performance of the SEM can be shown if the corresponding model is more accurate.

In Chapter 6, a basic hardware-based steady state calibration is conducted on the virtual engine for the optimization of fuel economy with constraints. Operating points at the low-speed low-load region are selected and local tests are carried out accordingly. The derived local optimal settings are assembled to form a calibration map and the control performance of the map is demonstrated.

In Chapter 7, dynamic models of the torque and AFR responses on the virtual engine are developed and then an approach of dynamic model-based calibration is proposed. The numerical optimization is applied to the surrogate models with torque and AFR constraints in order to find the optimal input-output behavior. A feedforward controller is then designed by an inverse identification of the optimal data and its performance on the virtual engine is shown. Moreover in order to evaluate the effectiveness of the dynamic model based-calibration, the performance of the controller is compared to the calibration map obtained in Chapter 6.

3.9 Conclusions

The experiments on the real engine are setup in order to provide data for identifications of black box model. A standard 1.6L Zetec engine is coupled with a low inertia dynamometer which provides a controllable load with extra sensors added for monitoring and collecting engine responses. D-space is utilized to manipulate the inputs of interest and it is advantageous to allow the EMS to control the rest of the actuators. Data samples are collected each degree as determined by the angle encoder and the specific resolution can be adjusted as demanded.

The virtual EcoBoost 2.0-Litre GTDI engine is used as a black box model for validating the proposed methods in this thesis. A WAVE model is assembled according to the detailed specification of the real GTDI engine and adapted to a WAVE-RT model in Simulink for ease and speed of execution. Sub-models such as a road load are developed for experimental requirements. Compared to the real engine, the virtual engine has the advantages of low experimental time and cost, providing a repeatable process and ease of adaptation. Consequently it is a suitable plant to test the proposed methods which are designed with various objectives and need to be validated statistically.

Chapter 4

Optimal Input Design for System Identification

4.1 Introduction

Many industrial applications of nonlinear system identification, such as in aircraft systems and automotive engine calibration, require high efficiency of data capture, high model prediction accuracy and protection from the exceedence of operational limits. Dynamic design of experiment (DoE) methodologies are accordingly sought to address these requirements for nonlinear dynamic experimental testing [34] [38] [88]. In recent decades, three aspects of DoE have been addressed:

(1) Optimization algorithms
(2) Optimality criteria design
(3) Experimental constraints

In this chapter, firstly a general survey of optimization is given and popular optimization algorithms for nonlinear optimization are introduced and compared. Section 3 indicates the approach of applying technologies for optimization to input design. A systematical procedure of optimal input design with constraints is presented. The generation and selection of non-optimal inputs for initial model estimation and the obtained original model are discussed in section 4 and 5. In section 6, two well-known criteria for minimization of parameter covariance, A-optimum and D-optimum, are applied and tested. A new criterion which weights the parameter variance by the square of output sensitivity terms are proposed as a further development of A-optimum methods and evaluated and found to be effective. In criteria for output prediction, a proposed criterion based on a simplified calculation of output covariance is illustrated to be more effective than I-optimum and G-optimum. Since the criteria with regard to parameter or output covariance are all expectation based, the effectiveness of op-

timal inputs is validated statistically. Applications with additional practical constraints and influence of disturbance are discussed in the end.

4.2 Methodology of Optimization

Optimization procedures are means of selecting a set of elements from the feasible candidate sets with the purpose of optimizing some characters of a system. Mathematically, a general optimization problem refers to maximizing or minimizing the value of a scalar objective function by searching for appropriate values of arguments in the feasible region. It has the form of:

$$\begin{aligned} &\arg\min f(x) \\ &x \in S \end{aligned} \tag{4.1}$$

where $f(x)$ is the objective function, x is the argument and S is the feasible region. The feasible region can be restricted by equivalence and inequivalence constraint functions:

$$\begin{aligned} a_i(x) &= 0, \qquad i = 1, 2, \ldots, m \\ b_i(x) &< 0, \qquad i = 1, 2, \ldots, n \end{aligned} \tag{4.2}$$

Optimization problems can be solved either by direct search or indirect search methods. Direct search methods only utilize the values of objective function and constraint function in each iteration. In the feasible space of arguments, variables move from the current position to nearby positions in all directions with an adjustable step size until a smaller function value is founded. Direct search is very suitable if the objective function and constraints function is extremely complex so that an analytical expression of functions is not available. However, it is relatively difficult to converge and a bigger computing burden results. Indirect search algorithms often determine the step size and direct the search with the help of a calculated local gradient. However, the objective function and constraint function should be differentiable or can be approximated as differentiable functions.

Unconstrained optimization and constrained optimization

The unconstrained optimization problem is a simplified case where the argument x is not restricted. Many effective indirect search algorithms such as deepest decent algorithms, Newton algorithms and conjugate gradient algorithms have been developed for unconstrained minimization. In the case of constrained minimizations, the optimization problem is converted into an unconstrained minimization using various types of penalty function or approximation and is then treated by appropriate unconstrained algorithms.

Linear optimization and nonlinear optimization

If the objective function and constraint are both linear, i.e. satisfying:

$$f(ax_1 + bx_2) = af(x_1) + bf(x_2) \tag{4.3}$$

the optimization problem is called a linear optimization (programming) and given as:

$$\begin{aligned} &\min_x \ a^T x && \\ &\text{subject to}: \ b_i^T x = c_i, && i = 1, 2, \ldots, m \\ &\qquad\qquad d_i^T x < e_i, && i = 1, 2, \ldots, n \end{aligned} \tag{4.4}$$

otherwise it is recognized as a nonlinear optimization. Many efficient and reliable algorithms have been developed for linear programming, e.g. primal-dual interior-point method for large-scale linear programming and the active-set algorithm or simplex algorithm for medium-scale problems. Nonlinear optimization consists of convex and non-convex optimization. Figure

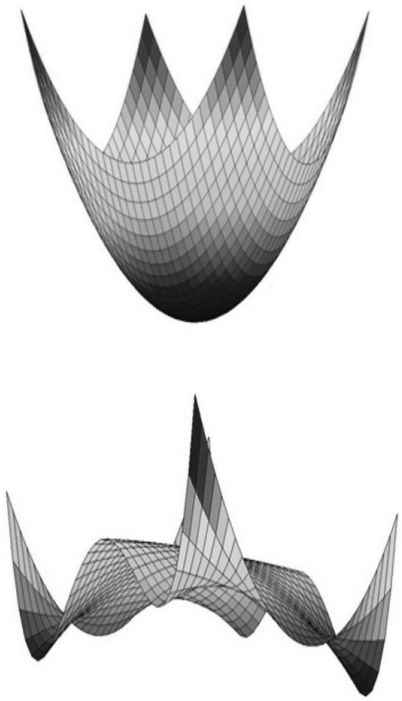


Figure 4.1: Schematic of the convex (top) and nonconvex (bottom)optimization

4.1 shows examples of convex and non-convex objective functions. A general optimization problem of the form of equations (4.1) and (4.2) is considered to be convex if the functions $f, a_1, ..., a_m, b_1, ...b_n : R^n \to R$ are convex. Since variables in a practical optimization problem could be numbered in hundreds, it is generally too difficult to plot the figure with respect to variables and function value. However, the convexity can be evaluated mathematically. A

function is convex if for any two point x_1 and x_2 in the feasible region and any $a, b \in [0, 1]$, the following inequality is satisfied:

$$f(ax_1 + bx_2) \leq af(x_1) + bf(x_2) \tag{4.5}$$

where $a + b = 1$. In a convex minimization, any local minimum (if it exists) must be a global minimum and the global minimum must exist if the functions are strictly convex. Technologies for convex optimization are not as mature as for linear optimization, however algorithms, such as the interior point algorithm, have proved to be effective in practice.

Non-convex optimization which involves multiple local minimums is the most difficult problem in optimization and there is not an effective algorithm that can generally solve all non-convex optimizations. Compromises have to be accepted in methods that attempt to solve non-convex problems. Local optimization methods seek an objective function value that is optimal in a neighbouring area rather than in the whole feasible space. Local optimization is relatively fast but is done at the expense of losing the global accuracy and reliability. The result of local optimization is considerably affected by the initial values of variables and information concerning the difference in magnitude between local and global optimum is not provided. In global optimization, analytical algorithms are often not applicable. Direct search and genetic algorithms can be used but require a very long experimental time. Although a global optimum still cannot be guaranteed, global algorithms have a stronger ability to avoid converging to a local optimum. Therefore the selection of methods for non-convex optimization is actually a compromise between accuracy and efficiency.

4.3 Optimization Algorithms

The objective of optimization is to approach the optimum value iteratively from the start point. Optimization algorithms have been developed by various methods of choosing direction and step length. Popular optimization algorithms for local and global optimization are surveyed as follows:

Trust region reflective algorithm

In an unconstrained minimization problem minimizing $f(x)$, optimization algorithms seek a proper step s from the current position x by various approaches for a smaller updated function value $f(x + s) < f(x)$. In the trust region reflective (TRR) algorithm, the objective function f is approximated by another function $g(x)$, which is often quadratic, within a subspace of the region of f around the current position x. This subspace is named the trust region R. Another minimization problem thus arises which is to find the minimum value of $g(u)$ within

the trust region:

$$\min_{u} \; q(u), u \in R \tag{4.6}$$

If $f(x+u) < f(x)$, the current position is updated to $x+u$ and the trust region is enlarged and the procedure is repeated until the function value converges; If $f(x+u) \geq f(x)$, the current position is not moved and the trust region is contracted and the step in equation (4.6) is repeated [89].

Sequential quadratic programming

Sequential quadratic programming (SQP) is one of the most popular methods for constrained optimization. Considering the general optimization problem in equation (4.1), the Lagrangian function given by:

$$L(x, \lambda, \sigma) = f(x) - \lambda^T a(x) - \sigma^T b(x) \tag{4.7}$$

where λ and σ are Lagrange multipliers. The principle idea of SQP is to solve this problem by working out a sequence of approximated subproblems. At the current position x_k, a subproblem is formed by a quadratic approximation of the Lagrangian function:

$$\begin{aligned} \min_{x,\lambda,\sigma} \; & L(x_k, \lambda_k, \sigma_k) + \nabla L(x_k, \lambda_k, \sigma_k)^T d + \frac{1}{2} d^T H_k d \\ \text{subject to}: \; & a(x_k) + \nabla a(x_k)^T d < 0 \\ & b(x_k) + \nabla b(x_k)^T d = 0 \end{aligned} \tag{4.8}$$

where H_k is the Hessian of the Lagrangian function and d is the search direction. The solution of the subproblem is used to find the position of the next point x_{k+1}. This iterative process is done in such a way that the sequence x converges to a local minimum [90].

Interior point algorithm

The interior point (IP) algorithm is a method for linear and nonlinear convex optimization. It translates the general form of equation (4.1) into an equality constrained form given by:

$$\begin{aligned} \min_{x,s} \; & f_(x, s) = \min \; f(x) - \mu \sum \ln(s_i) \\ \text{subject to}: \; & a(x) = 0, \quad b(x) + s = 0 \end{aligned} \tag{4.9}$$

where $\sum \ln(s_i)$ denotes a barrier function and the slack variable s_i is a positive value for restricting the logarithmic term. μ denotes a barrier parameter. Since μ converges to zero, the solution of equation (4.9) will approach the solution of equation (4.1). Therefore an optimization problem with equality and inequality constraints is reduced to an equality constrained problem. Both the Newton and conjugate gradient methods can be utilized to solve the approximated equality optimization problem [91].

Pattern search

The Pattern search (PS) algorithm is one of the popular direct search methods which can be used in functions that are not continuous or differentiable. This algorithm approaches an optimal solution iteratively without any assistance of the gradient or higher order derivative of the objective function. In each iteration, directions of search and corresponding sequencing, called patterns, are decided firstly. The variables move from the current position towards the first determined direction with a specified step. After that the function value at the updated position is computed and if the obtained value is smaller than the previous one, it is recognized as a successful poll. The new position becomes the current position of the next iteration and the step size is doubled. If the poll failed, variables will be moved along other available directions in order with the same step size and then with a reduced size until a successful poll occurs. Alternatively, pattern search can calculate the function values in all feasible directions then move to the position where the function value is the smallest. However as the feasible directions increase exponentially with the number of variables, the complete directional search only fits for optimizations with small amount of variables [92].

Although pattern search may not be as efficient as other gradient based deterministic algorithms, it has a unique merit. In non-convex optimization, gradient based algorithms converge to a local minimum because the reposition of variables is guided by the gradient and the gradient approaches zero at the local minimum then the process ends. However, the reposition of variable of pattern search is determined by an adjustable step size of arguments, which means the variable can move from one cone to another, provided that the function value at the position on the new cone is smaller than the current value. Pattern search hence has a capability of giving a global optimum.

Genetic algorithm

The genetic algorithm (GA) is inspired by the evolution theory of Darwin. It is capable of solving local optimization and global optimization based on the procedure of natural selection. Unlike most gradient-based deterministic algorithms, the genetic algorithm can be used to solve problems which have discontinuous or undifferentiable objective function. As a stochastic algorithm, it generates a population of solutions at each iteration and selects the best one, while most other stochastic methods operate on a single solution. The procedure of the genetic algorithm can be briefly described as follows [93]:

Initialization: Initially a random population, composed of many individual solutions, is produced as parents of the first generation. A proper size of the population is essential to the optimization result since an extremely large size will occupy most system resource and an insufficient one may omit the global optimum. Generally the random production takes

place in the entire feasible region, whilst when prior knowledge is available, the production of population can be manually restricted to a particular sub-region for higher probability of finding the optimal value. The quality of initial generation is improved correspondingly.

Selection: In each generation, all individual solutions are measured by a fitness function. The solutions which have better fitness have stronger probability to be selected as "parents" to breed the next generation. However, the selection is not solely guided by the fitness because it may lead the algorithm to quickly converge to a local minimum rather than a global minimum if low fitness solutions are completely omitted.

Regeneration: The selected individual solutions in the current generation are used to produce new solutions for the next generation, by following the rules of crossover and mutation [94]. The new generation resulting from the process of selection, crossover and mutation is different from the initial generation and is likely to have better fitness because the individual solutions are produced by the best "parents". The process of selection and regeneration continues until a stopping criterion is satisfied.

Simulated annealing

The simulated annealing (SAN) algorithm belongs to the family of stochastic probabilistic methods. It is inspired by annealing in metallurgy which minimizes the internal energy by means of heating and slowly cooling the metal.

Initially, a state point S is randomly generated in the feasible space and a temperature T is given. Then a new state S' is produced whose position is based on a probability distribution of the temperature and the corresponding value of the objective function is updated subsequently. The increment of objective function value from S to S' is calculated and the new point is accepted if it causes a lower objective. Nevertheless, even if it raises the objective, S' can still be accepted with a certain probability in order to avoid approaching a local minimum. In the next iteration, the temperature is adjusted according to the annealing schedule and a similar process is implemented to the new state point S' or S if no point is accepted [95].

The simulated annealing algorithm is independent of the initial state. Theoretically it converges to the global optimum with the probability of 1, but the demanded experimental time to achieve a good probability of SAN is often extremely long and can even exceed that for a full search in the entire region.

The three local optimization algorithms, the TRR, SQP and IP all require a second derivative of the Lagrangian function. These second variational methods are claimed to have superior convergence rate than first variation methods such as the deepest descent and

Table 4.1: Features of optimization algorithms

	TRR	IP	SQP	PS	GA	SAN
Global optimization				√	√	√
Input bound	√	√	√	√	√	√
Linear equality constraints	√	√	√	√	√	
Nonlinear constraints		√	√	√	√	
Gradient based	√	√	√			
Direct search				√		
Stochastic Algorithm					√	√

conjugate gradient method [96]. It is also worth noting that although global optimization algorithms have the capability of finding the global optimal value, they can be easily trapped at a local optimum. All of the global algorithms compromise between the convergence rate and the extent of the global optimum. Therefore parameters of global algorithms should be selected appropriately in different applications. In complex practical work, although no algorithm can guarantee a global optimum within a finite time, it is still favourable for a solution which satisfies the specific requirements to be found without knowing the existence of a better solution.

Optimization algorithms mentioned above are provided as Matlab functions by the MATLAB Optimization toolbox and are utilized in the optimal input design work in this thesis. Characteristics of the algorithms are listed as in Table 4.1. The toolbox is able to approximate the gradient as necessary and the stopping criteria are given ready for the specification of users. In this thesis, the specification of stopping criteria in a certain optimization problem is kept unchanged between different algorithms in order to fairly compare their effects.

4.4 Iterative Optimal Input Design with Experimental Constraints

An iterative procedure of constrained optimal input design is illustrated in Figure 4.2. In the first iteration, an initial model needs to be identified by non-optimal inputs and corresponding outputs collected from the system. The constraints in the current iteration can initially be conservative and then gradually approach the ultimate experimental constraints in subsequent iterations. The initial conditions for optimization, such as the initial values of inputs can be adapted from the non-optimal signals. The objective function and optimization algorithm used for all iterations need to be designed appropriately for the accuracy of the optimization and the efficiency of computation. Generally, the optimization algorithm is selected according to the convexity of the function and available time for experiment. Resulting optimal inputs will be applied to the system and another model identified subsequently. The model will be

updated and the procedure is repeated until a model with acceptable goodness is determined using optimal signals obtained with non-conservative constraints.

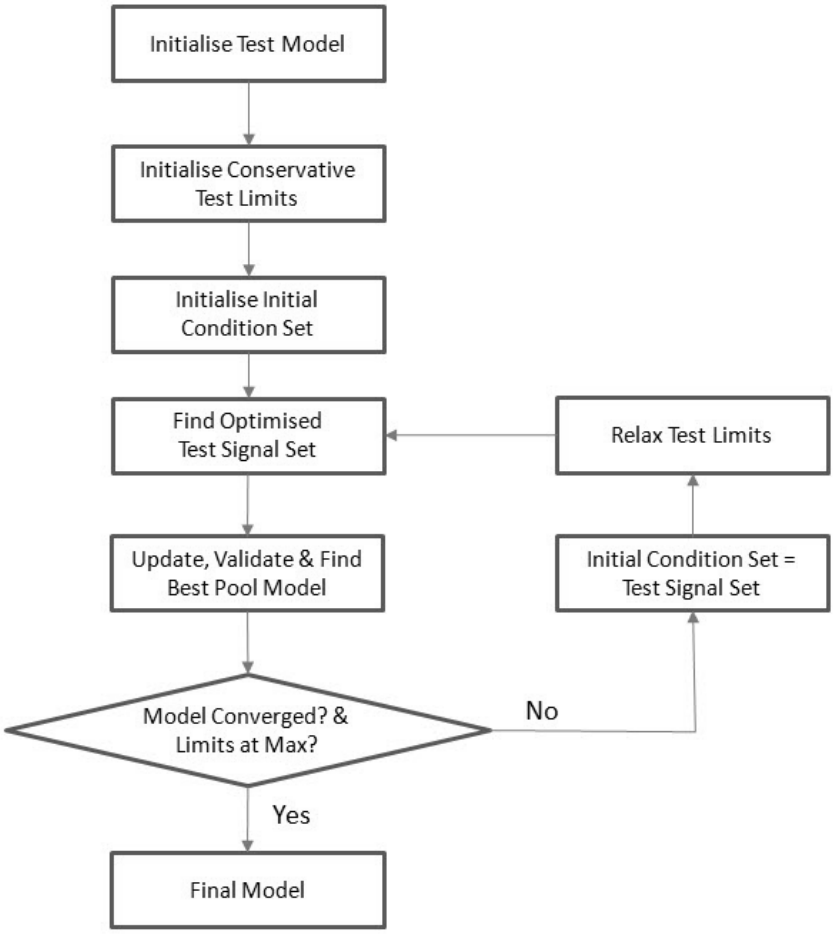


Figure 4.2: Flow chart of the iterative process of optimal input design

4.5 Input Selection for Initial Identification

The method of optimal input design and system identification should be conducted iteratively for the sake of improving the model quality gradually. The selection of inputs for the initial identification hence becomes extremely important in order to give a good start which may reduce the number of iterations processed for an acceptable model. Since any prior knowledge of a black box model is not generally available initially, the input for initial identification should be able to excite the dynamics of most systems. Commonly used inputs are introduced as follows.

4.5.1 White Noise Signal

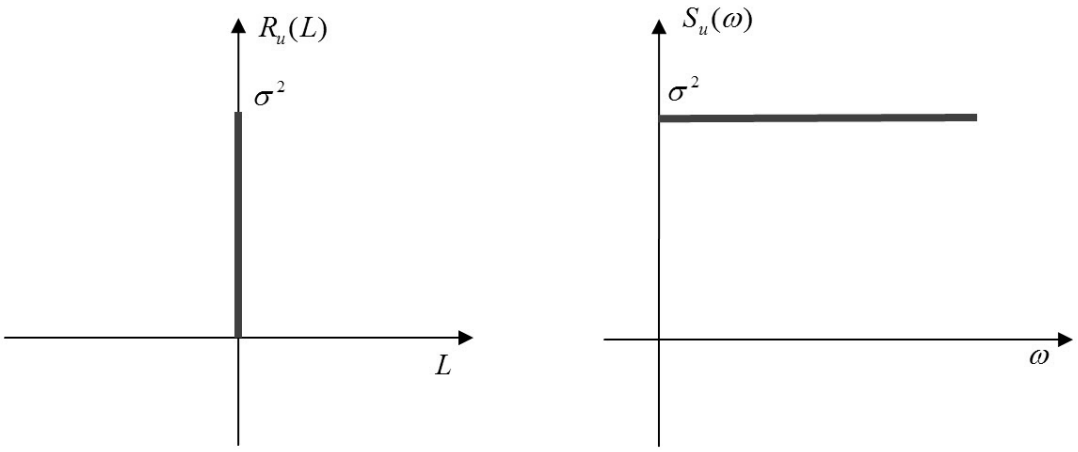


Figure 4.3: ACF $R_u(L)$ and PSD $S_u(\omega)$ of an ideal white noise

A white noise signal is a random signal with zero mean, an impulse like auto-correlation function (ACF) $R_u(L)$ and a constant power spectral density (PSD) $S_u(\omega)$ as shown in Figure 4.3. Mathematically it can be expressed as:

$$\mu_u = E[u] = 0 \tag{4.10}$$

$$R_u(L) = \sigma^2 \delta(L) \tag{4.11}$$

$$S_u(\omega) = \sigma^2 \tag{4.12}$$

$$\delta(L) = \left\{ \begin{array}{l} 1 \ when \ L = 0 \\ 0 \ when \ L \neq 0 \end{array} \right\} \tag{4.13}$$

where L denotes the time delay of the signal and ω denotes the frequency. A PSD is the Fourier transform of the ACF, which presents how the power of a signal is distributed with frequency. For systems without prior frequency domain knowledge, a white noise signal could be an ideal identification signal since it has a flat PSD where the power of input is evenly distributed at any frequency.

An ideal white noise signal is however not realizable in practice. In this thesis a uniformly distributed random number (UDRN) block is used to approximate a white noise signal. It has the merit that the maximum and minimum value of the signal can be defined according to the experimental input amplitude constraints. Figure 4.4 shows a simulated discrete white noise signal and corresponding ACF. The main features of an ideal white noise signal are clearly demonstrated though values of autocorrelation at non-zero delay points are slightly disturbed.

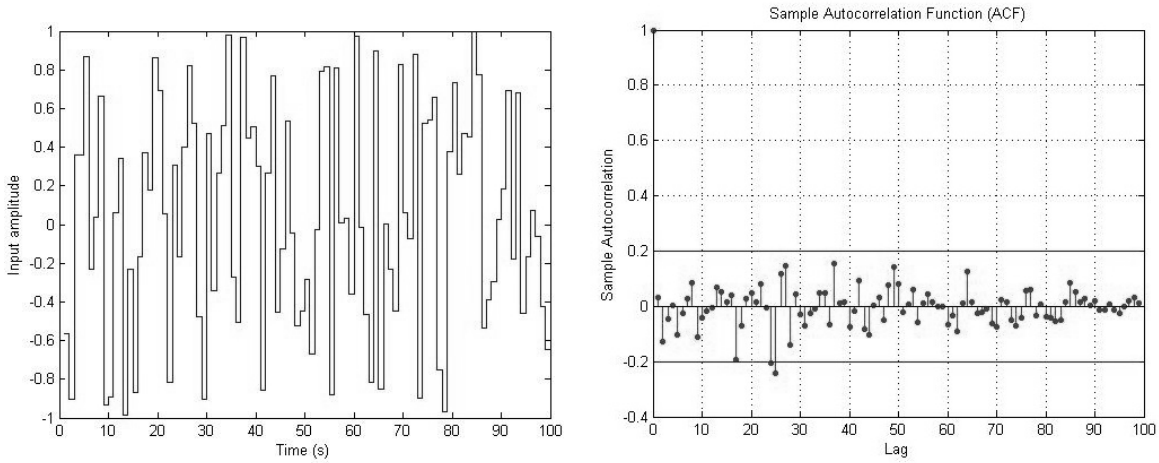


Figure 4.4: Simulated white noise and ACF

4.5.2 Pseudo Random Binary Signal

A discrete Random binary signal (RBS) is a stochastic signal which has 2 levels $\pm\sigma$ and the value switches from one level to the other at any time interval λ. The AFC and PSD of an RBS is given as follows:

$$R_u(L) = \left\{ \begin{array}{cc} \sigma^2\left(1-\frac{|L|}{\lambda}\right) & when\ |L|<\lambda \\ 0 & when\ |L|\geq\lambda \end{array} \right\} \tag{4.14}$$

$$S_u(\omega) = \sigma^2\lambda\left(\frac{sin\frac{\omega\lambda}{2}}{\frac{\omega\lambda}{2}}\right)^2 \ when\ 0\leq|\omega|\leq\frac{\pi}{T_0} \tag{4.15}$$

where T_0 is the sample time. Figure 4.5 depicts a typical RBS and its ACF. The ACF and

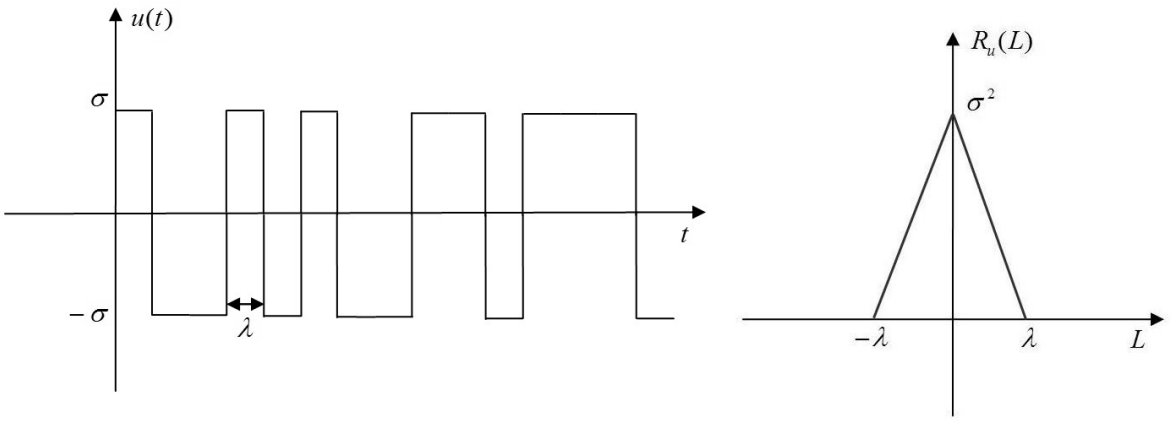


Figure 4.5: Discrete random binary signal and corresponding ACF

PSD of a discrete random binary signal can be very similar to those of a white noise signal if the time interval is infinitely small. A unique advantage of RBS is that it delivers the largest

amplitude density for any amplitude-constrained input. It is considered more informative than other signals for linear system identification since the constrained amplitude range is utilized in the most efficient way.

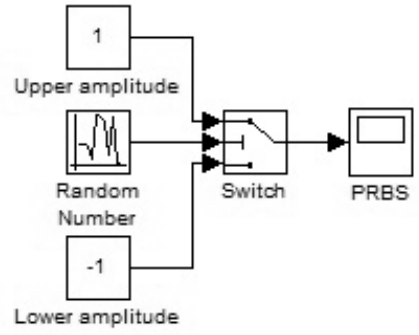


Figure 4.6: A Simulink generator of PRBS

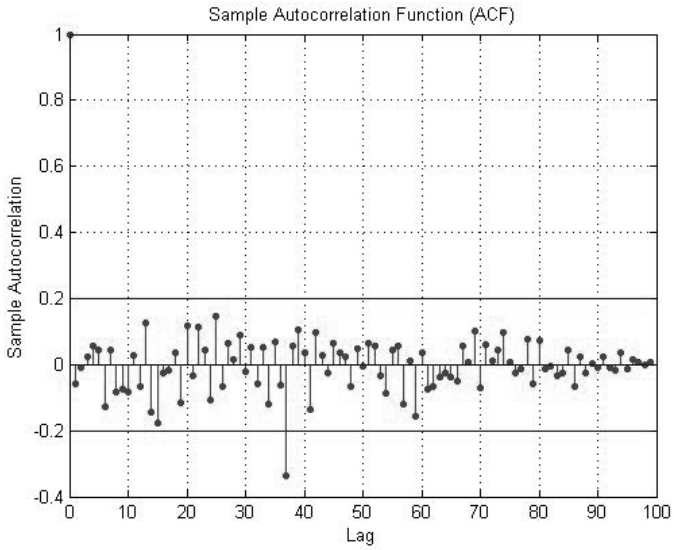


Figure 4.7: ACF of PRBS

An ideal RBS is completely stochastic therefore it cannot be generated by computers which are deterministic devices. A periodic signal, pseudo random binary signal (PRBS) which has a very similar ACF to RBS is often used in practical work. PRBS can be generated by the well-known means of a shift register circuit [34] and the digit of the register determines the length of period. In this thesis a PRBS is converted from a random number signal by restricting the random value to two pre-determined levels as shown in Figure 4.6. The random number block generates a zero mean signal as a reference signal then the 2 level value can be selected according to the result of comparing the reference signal to zero. For a 32 bit system, the length of the period of the generated random number is 2^{32} so that the period of the PRBS is sufficient long and the ACF of the simulated PRBS in Figure 4.7 is very similar

to a pure RBS and white noise.

4.5.3 Amplitude-modulated Pseudo Random Binary Signal

For nonlinear system identification, perturbation signals should have multi-level values over the input range in order to excite the nonlinear dynamics, thus the amplitude of PRBS (APRBS) needs to be modulated. In a difference from white noise signals, the number of levels of APRBS is pre-defined. However with increasing signal levels, it gradually approaches white noise and the amplitude density decreases correspondingly. The whole input range is divided equidistantly and a random number signal is generated to select the pre-determined level at each time.

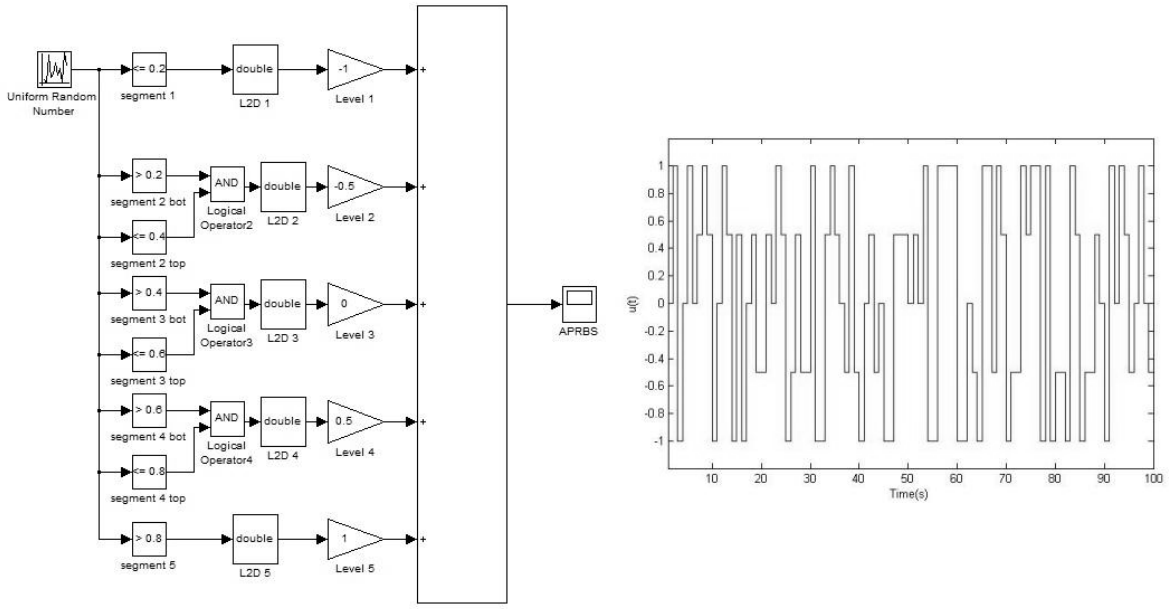


Figure 4.8: A Simulink generator of APRBS

Figure 4.8 shows an APRBS generator assembled in Simulink and the APRBS produced. The entire range of the random number is split equally into segments and values of multi-level are determined accordingly then APRBS thus produced.

4.5.4 Random Walk Signal

In practical experiments where rate constraints are required, a white noise signal is not appropriate because the value of change in the input is generated randomly. Although amplitude levels of APRBS are pre-determined, the value is still selected stochastically so that it cannot prevent a drastic step from the current level to the next. A random walk signal is composed of a sequence of discrete steps with fixed step length, whilst the direction of the step is stochas-

tic. Compared to other signals, a random walk signal sweeps over the input range slowly due to the fixed step length so that the amplitude density is relatively low. It is however not a serious drawback if the data sequence for identification is sufficiently long.

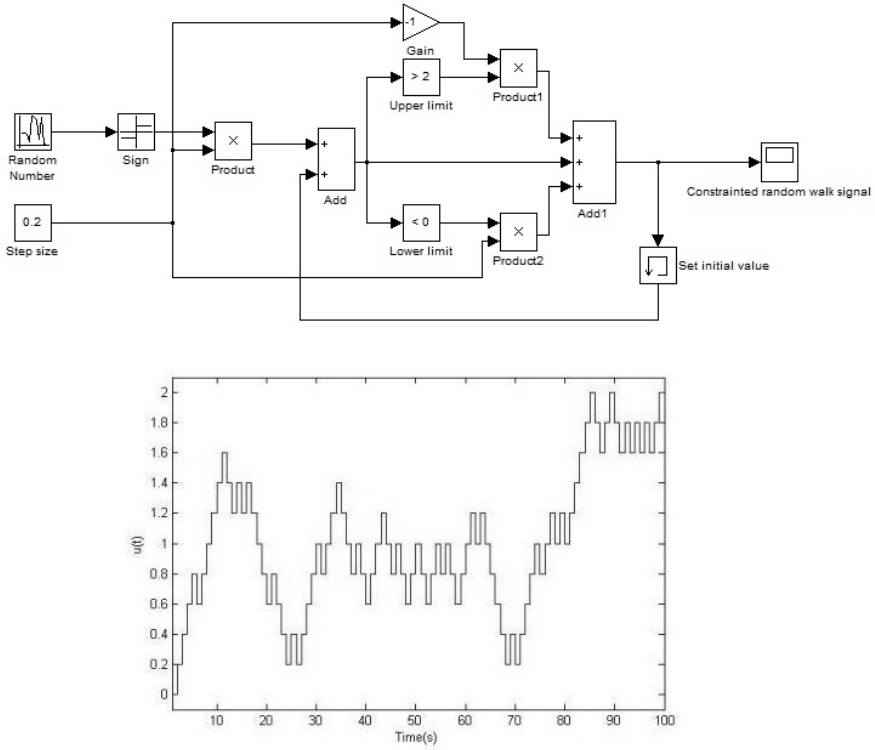


Figure 4.9: A Simulink generator of amplitude constrained random walk signal

A random walk signal and its generator is shown in Figure 4.9. The random walk signal is generated by an initial value and an increment in each step. The step size is fixed and the sign of increment is randomly chosen. To constrain the amplitude of the signal in a desired range, the sign of the next increment is changed by reversing the direction if the value of the signal at the current step exceeds pre-determined boundaries.

These types of signals are recommended for initial estimations of unknown systems. In engine calibration, the selection of signal for initial engine model identification should be determined by the behavior of the system in the interested operating region, for example a torque model in the idle speed region or an AFR model in the high speed high load region. If the system behavior is expected to be linear, a PRBS signal can be employed because it is constrained in amplitude and has the largest amplitude density for any amplitude-constrained input [34]. However it is not recommended to identify a nonlinear system since it may cause a problem in the identifiability [36]. Comparing to a PRBS signal, a white noise signal has a smaller amplitude density. It is generally used to identify a nonlinear system or a system without any prior knowledge because it has multi-level values over the input range to excite

the nonlinear behaviours of the system. A random walk signal is used if there is a practical rate constraint on the inputs. To overcome the disadvantages of PRBS and white noise signal, an APRBS signal can be selected to identify the system and the multi-level values of this signal can be adjusted to give a larger amplitude density without losing the identifiability.

4.6 MISO Engine Model Identification

The system to be identified is a 3×1 nonlinear MISO torque model of a 1.6 Litre port fuel injection Zetec engine with FPW (u_1), ABV (u_2) and engine speed (u_3) as inputs. In order to avoid very high frequency noise, the inputs for the nonlinear torque model identification were collected every stroke (180°). During the experiment, the other controllable parameters are fixed, e.g. the SA is fixed to be 20° before TDC.

As discussed above, white noise signals are suitable to perturb ABV and FPW in order to excite the nonlinear dynamics. Both amplitude and time interval should be considered to generate a proper white noise test signal. The amplitude of inputs should be sufficient for a representative torque response in the desired operating space without engine stall. In this engine experiment, input amplitude constraints have been established as:

$$\begin{aligned} 2000\mu s &< u_1 < 6000\mu s \\ 40\% &< u_2 < 60\% \end{aligned} \tag{4.16}$$

and the resulting engine speed (u_3) for system identification is between 1000 to 2000 RPM.

In the engine experiment, the D-space hardware can read signals from the PC and quickly adjust engine inputs such as FPW, ABV and SA to be demanded values. However the engine speed cannot be controlled by the same approach because it is actually a consequence of many other parameters. Therefore the engine is connected to a low inertia dynamometer and the load applied by the dynamometer is used to restrict the engine speed to the desired range.

For the purpose of reducing the required experimental time for the optimal input design and statistical validation, the data length is down sampled into 100 points. The IV method is employed for parameter estimation because data collected from the engine might be corrupted

by a correlated disturbance. The selected model structure is shown as follows:

$$\begin{aligned} y(t) &= \theta_1 + \theta_2 u_1(t) + \theta_3 u_1(t)u_2(t-10) + \theta_4 u_1(t)u_3(t) + \theta_5 u_1(t)u_3(t-3) \\ &\quad +\theta_6 u_1(t-10)u_2(t-10) + \theta_7 u_1(t-10)u_3(t-9) + \theta_8 u_1(t-10)^2 + \theta_9 u_2(t-10)^2 \\ &\quad +\theta_{10} u_2(t-10)u_3(t-9) + \theta_{11} u_3(t)^2 + \theta_{12} u_3(t-2)u_3(t-4) + \theta_{13} u_2(t-10)u_3(t) \\ &\quad +\theta_{14} u_1(t-10) + \theta_{15} u_3(t)u_3(t-5) + \theta_{16} u_3(t-1)u_3(t-10) \\ &\quad +\theta_{17} u_2(t-10)u_3(t-5) + \theta_{18} u_3(t-2)u_3(t-9) + \theta_{19} u_3(t-2)u_3(t-6) \\ &\quad +\theta_{20} u_1(t-6)u_2(t-7) + \theta_{21} u_1(t-6)u_1(t-10) + \theta_{17} u_2(t-10)u_3(t-5) \\ &\quad +\theta_{22} u_1(t-6)u_1(t) + \theta_{23} y(t-10) \\ z(t) &= y(t) + \epsilon(t) \end{aligned} \tag{4.17}$$

The estimated parameters are:

$$\begin{aligned} \theta &= [\theta_1, \theta_2, ...\theta_{23}] \\ &= [64.17, -0.025, 0.062, -4.06 \times 10^{-7}, 2.73 \times 10^{-6}, -0.0069, 8.5 \times 10^{-7}, 1.54 \times 10^{-6}, \\ &\quad -327.7, 0.0016, -1.52 \times 10^{-5}, -4.22 \times 10^{-8}, 0.069, -0.014, -1.56 \times 10^{-6}, \\ &\quad -2.25 \times 10^{-6}, 0.029, 5.67 \times 10^{-7}, 6.4 \times 10^{-7}, 0.0021, 8.71 \times 10^{-7}, -1.36 \times 10^{-6}, 0.15] \end{aligned}$$

with $cov(\epsilon) = \sigma^2 = 80$. The sample time is taken as 0.1 sec.

Conventional methods of optimal input design have been developed with an assumption that the model structure of the real system is known. However, a true model structure of the engine mechanisms discussed in this thesis is not available. Therefore the original model obtained by initial identification is considered as the "real" system and is used to test the optimal input for the purpose of proving that the optimal signal is effective. In later sections the optimal signal is implemented on a real system with unknown structure to demonstrate its suitability for industrial applications.

4.7 Optimal Input Design for Improved Parameter Estimation

For white box model where the model structure of the real system is known, it is the parameter estimation that determines the accuracy of the identified model. The accuracy of the estimated parameters can be expressed in terms of its statistical property such as covariance and bias. In most of the research on optimal input design, the optimization is simplified to minimize the parameter covariance because it is assumed that an unbiased efficient estimation method is used.

4.7.1 Information Matrix and Cramor-Rao Bound

From equation (2.4), the information matrix is given by:

$$M \equiv E_{Y|\theta}\left[\left(\frac{\partial \ln p(Y|\theta)}{\partial \theta}\right)\left(\frac{\partial \ln p(Y|\theta)}{\partial \theta}\right)^T\right] = -E\left(\frac{\partial^2 \ln p(Y|\theta)}{\partial\theta\partial\theta^T}\right)$$

The log-likelihood function is in the form:

$$\ln p(Y|\theta) = -\frac{1}{2\sigma^2}\sum_{t=1}^{N}\epsilon(t)^T\epsilon(t) - \frac{N}{2}\ln|\sigma| - \frac{Nn_0}{2}\ln(2\pi) \tag{4.18}$$

where $\epsilon(t) = z(t) - y(t)$. The last two terms in equation (4.18) are independent of the parameter θ thus the first and second gradient of the likelihood function are obtained as [38]:

$$\frac{\partial \ln p(Y|\theta)}{\partial \theta} = \frac{1}{\sigma^2}\sum_{t=1}^{N}\frac{\partial y^T(t)}{\partial\theta}\epsilon(t) \tag{4.19}$$

$$\frac{\partial^2 \ln p(Y|\theta)}{\partial\theta\partial\theta^T} = -\frac{1}{\sigma^2}\sum_{t=1}^{N}\frac{\partial y^T(t)}{\partial\theta}\frac{\partial y(t)}{\partial\theta} + \frac{1}{\sigma^2}\sum_{t=1}^{N}\frac{\partial^2 y(t)}{\partial\theta\partial\theta^T}\epsilon(t) \tag{4.20}$$

and the entries of these vectors are:

$$\frac{\partial \ln p(Y|\theta)}{\partial \theta_i} = \frac{1}{\sigma^2}\sum_{t=1}^{N}\frac{\partial y^T(t)}{\partial\theta_i}\epsilon(t) \qquad i,j = 1,\ 2,\ ...,\ p \tag{4.21}$$

$$\frac{\partial^2 \ln p(Y|\theta)}{\partial\theta_i\partial\theta_j} = -\frac{1}{\sigma^2}\sum_{t=1}^{N}\frac{\partial y^T(t)}{\partial\theta_i}\frac{\partial y(t)}{\partial\theta_j} + \frac{1}{\sigma^2}\sum_{t=1}^{N}\frac{\partial^2 y(t)}{\partial\theta_i\partial\theta_j}\epsilon(t) \tag{4.22}$$

The simplification of the second gradient can be made by neglecting the second term in equation (4.20) which is computationally expensive to obtain. Therefore the Fisher information matrix is simplified to [38]:

$$M = -E\left(\frac{\partial^2 lnp(Y|\theta)}{\partial\theta\partial\theta^T}\right) \approx \frac{1}{\sigma^2}\sum_{t=1}^{N}\left(\frac{\partial y(t)}{\partial\theta}\right)^T\left(\frac{\partial y(t)}{\partial\theta}\right) \tag{4.23}$$

where N denotes the length of discrete data set. To determine the data length, the allowable experimental time and the asymptotical property of the Cramer-Rao lower bounds should be considered. The low bounds decrease with increasing data length, however with a fixed sample time, collecting more data requires a large amount of experimental time and a heavy computing burden in the stage of optimization. The information matrix is calculated over the entire data sequence. Therefore if any scalar function of M is selected as the criterion for comparison in order to compare the effectiveness of different inputs, the data length should be kept the same.

The goodness of the information matrix can be measured statistically as [41]:

$$J = E_\theta\phi(M) \tag{4.24}$$

where ϕ is a scalar function of M. Practically this criterion can be simplified by evaluating $\phi(M)$ at suitably chosen parameter values.

Input design with a constant power as input constraint has been explored by many authors [97] [98] [99]. Nevertheless, since the constant power can be obtained by various combinations of data length and maximum allowable input amplitudes, it is not suitable to implement the constant power as the only input constraint if a scalar function of the information matrix is used as criterion. All the optimal inputs designed in this thesis have the same data length and input amplitude constraints as a basic limit.

According to equation (2.6), the output sensitivity equations are obtained as:

$$\begin{aligned}
\frac{\partial y(t)}{\partial \theta_1} &= 1 + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_1} \\
\frac{\partial y(t)}{\partial \theta_2} &= u_1(t) + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_2} \\
\frac{\partial y(t)}{\partial \theta_3} &= u_1(t)u_2(t-10) + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_3} \\
&\vdots \\
\frac{\partial y(t)}{\partial \theta_{22}} &= u_1(t-6)u_1(t) + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_{22}} \\
\frac{\partial y(t)}{\partial \theta_{23}} &= y(t-10) + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_{23}} \\
\frac{\partial y(1)}{\partial \theta} &= [0 \quad 0 \quad \dots \quad 0]^T
\end{aligned} \tag{4.25}$$

Equation (4.23) indicates that the inputs have a nonlinear influence on M, regardless of whether the original model is nonlinear or not because the matrix is nonlinear of the output sensitivity.

The Cramer-Rao law states that the variance of any unbiased estimator is no smaller than M^{-1}:

$$cov(\hat{\theta}) \geq M^{-1} \tag{4.26}$$

where the theoretical lower limit for the covariance of estimated parameters will be achieved if an unbiased efficient estimator is utilized. In equation (4.26), the diagonal elements of M^{-1}, S_{jj}, represent the achievable minimum value of parameter variances and the square roots of the elements are called Cramer-Rao lower bounds which are the standard deviations of estimated parameters:

$$S(\hat{\theta}_j) = \sqrt{S_{jj}} \tag{4.27}$$

The Cramer-Rao lower bound depends on the inputs and a pre-determined model structure with parameters but is independent of the parameter estimation method. Therefore if a priori knowledge of the system is available, it is worthy of using optimal input design to minimize the theoretical minimum variance before estimating the parameters.

4.7.2 Statistical Properties of Parameter Variance

In equation (2.29), the output is disturbed by a white noise term. Because of the existence of noise in the output, results of parameter estimation will be different even if the same inputs are applied to the system. Subsequently the parameter variance which is affected by the input, noise and estimation method is used to measure the error between the true parameter and estimated parameter in a probabilistic way.

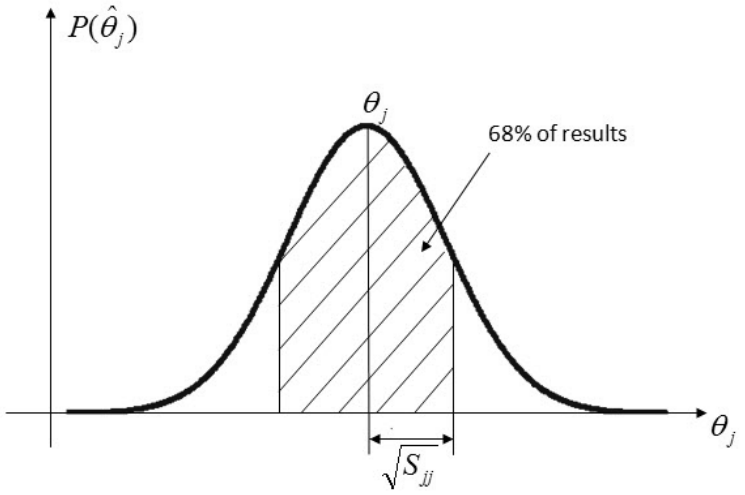


Figure 4.10: Normal distribution of estimated parameter $\hat{\theta}_j$

Assuming the output of the system follows a normal distribution, then Figure 4.10 demonstrates the relationship between the true parameter, the estimated parameter by OLS and the parameter variance [41]. The square root of parameter variance $\sqrt{S_{jj}}$ represents the standard deviation of the estimated parameter and it is shown that the estimated parameter should be in this window with a probability of 68%. A smaller standard deviation indicates that the estimated parameter has a higher probability of approaching the true parameter. The optimal input which minimizes the parameter variance therefore leads to probably more accurate parameter estimation results than non-optimal inputs.

4.7.3 Design of A-optimal Criterion

Minimizing the lower bound of parameter covariance corresponds to maximizing the information matrix which can be measured by various criteria. The A-optimal criterion is a traditional criterion which seeks to minimize the sum of variances of the estimated parameters. It is given by minimizing the trace of the inverse of the information matrix in the form of:

$$J_A = \mathrm{tr}(M^{-1}) \tag{4.28}$$

Table 4.2: Parameter variance by UDRN input and optimal input

	$cov(\hat{\theta})_{UDRN}$	$cov(\hat{\theta})_{op}$
θ_1	0.000282	0.000108
θ_2	0.000284	0.000109
θ_3	0.000285	0.000108

The objective function can easily be proved to be non-convex according to equation (4.5) using two different PRBS inputs.

Example 1

Consider a linear dynamic SISO system:

$$y(t) = \theta_1 u(t-1) + \theta_2 u(t-2) + \theta_3 u(t-3) + \epsilon(t) \tag{4.29}$$

where $\epsilon \sim N(0,1)$ and $\theta = [35.4,\ -0.08,\ 2.6]$. As a basic example for demonstrating optimal input design, the true parameter of the system is assumed known and is chosen as the initial parameter values for the design. Thereby the iterative procedure is only carried out once. Standard optimal designs with initial parameter estimation and more iterations are shown in later sections.

J_A in equation (4.28) is selected as the objective function and the information matrix is given by:

$$M = \frac{1}{\sigma^2} \sum_{t=1}^{N} \begin{bmatrix} u^2(t-1) & u(t-1)u(t-2) & u(t-1)u(t-3) \\ u(t-1)u(t-2) & u^2(t-2) & u(t-2)u(t-3) \\ u(t-1)u(t-3) & u(t-2)u(t-3) & u^2(t-3) \end{bmatrix} \tag{4.30}$$

The desired optimal input is required having a data length of 100 and an amplitude constraint of [-10,10]. Figure 4.11 shows the UDRN signal which is used as the vector of initial values of variables in the optimization and the obtained optimal input. The optimal input looks similar to a PRBS signal since most points of this input are very close to the amplitude limits. It indicates that a binary signal is considered optimal to identify a linear system, as suggested by Ljung [34]. For comparison, their parameter variance evaluated by the information matrix is shown in Table 4.2.

The A-optimal design is proved to be effective since the parameter variance derived from the A-optimal input is considerably smaller, less than 40% of the one derived from the UDRN signal. However, in this example the magnitudes of the diagonal elements of M have a similar scale hence the defect of un-weighted A-optimal design is not exposed. In the next example, the A-optimal design is applied to a more complicated model. The disadvantage caused by different scales of the output sensitivity equations in M, is discussed and an effective weighting function is proposed.

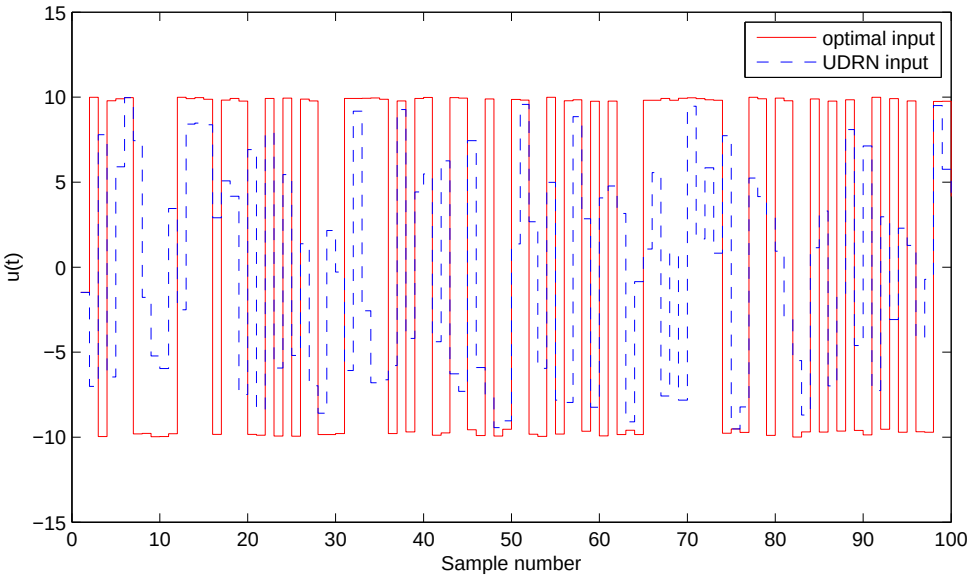


Figure 4.11: UDRN input and optimal input

Example 2

In this example, optimal inputs are designed for the MISO engine model mentioned in Section 4.6. The system is given as the original model in equation (4.17). Bounds of inputs are set as in equation (4.16) and no other linear or nonlinear constraints are implemented. The number of variables is 300 in total, 100 for each input and the initial values of the variables are given by a UDRN input signal.

Various optimization algorithms are tested and the convergence rate is shown in the following figures. Figure 4.12 shows the convergence rate of 3 local optimization algorithms in 50 iterations. The A-optimal criterion is selected as the objective function and the Y axis denotes the value of objective function which is decreasing in iterations. The time required for algorithms to generate 50 iterations are approximately 120 sec. The convergence rates of trust-region algorithm and SQP algorithm are very similar and their function values drop drastically in the 2nd iteration. A reasonable explanation could be that a quadratic approximation is made of the Hessian of the Lagrangian function and then a QP subproblem is generated accordingly in both algorithms. The SQP is generally preferred as it is compatible with nonlinear constraints. The interior point algorithm exhibits a more smooth convergence and reaches the same value of objective function as the others after 30 iterations.

Because of the significant distinctions in principles of the global algorithms, e.g. direct search or indirect search, deterministic or stochastic, the efficiencies of these algorithms evaluated against the number of iterations might not be convincing. Based on the tests of the

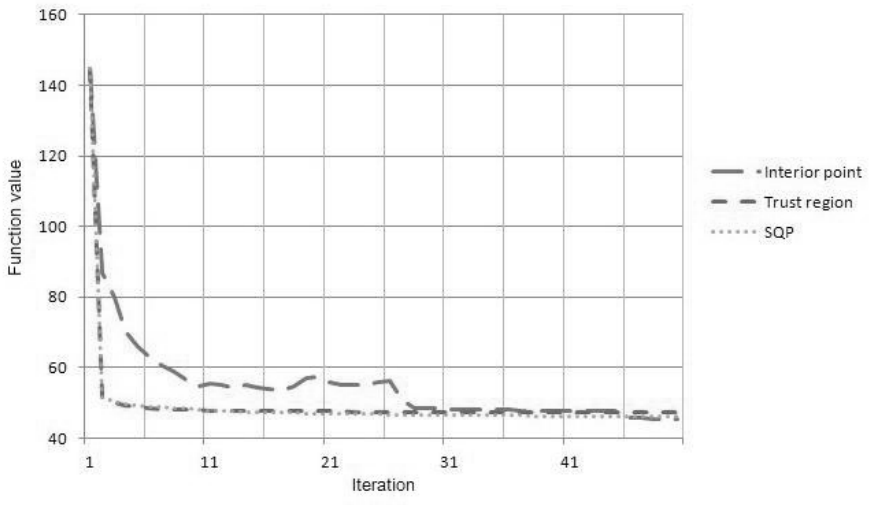


Figure 4.12: Objective function value by local algorithms

numerical optimizations with various algorithms in this thesis, it is found that hundreds of iterations can be generated in a second by the simulated annealing algorithm but very little reduction of the value of objective function obtained in each iteration; the genetic algorithm with a large population size may take minutes for each iteration but the improvement of function value can be remarkable. Moreover, the number of solutions obtained in each iteration is also different. Therefore the optimization results of global algorithms are illustrated separately.

Figure 4.13 shows the results of the SAN algorithm in 6000 iterations which cost about 5 minutes. The lower figure shows the current function value of each iteration. In the first iteration, the temperature is 100 degrees at which point the function value is the largest. With decreasing temperature, the function value reduces accordingly and converges until the temperature reaches 0 degrees. The first annealing is finished in 300 iterations and then the process is carried out again. The upper figure shows the best function value from the start to the current iteration. It can be seen that this algorithm converges very quickly in each annealing and a small function value can be expected in the first annealing. Nevertheless it takes much more time to obtain a smaller value with the increment of iterations.

Figure 4.14 shows the convergence rate of the GA algorithm in generations. In each generation, 20 individual solutions are produced in random positions. The mean and best function values in each generation are plotted and the mean value converges to the best value asymptotically. Figure 4.15 shows the result of mesh size and function value of the PS algorithm. In each iteration of this direct search algorithm, the step length is specified as the mesh size but there are many feasible step directions which are determined by the number of variables. Therefore the function value in the figure represents the best value at the current

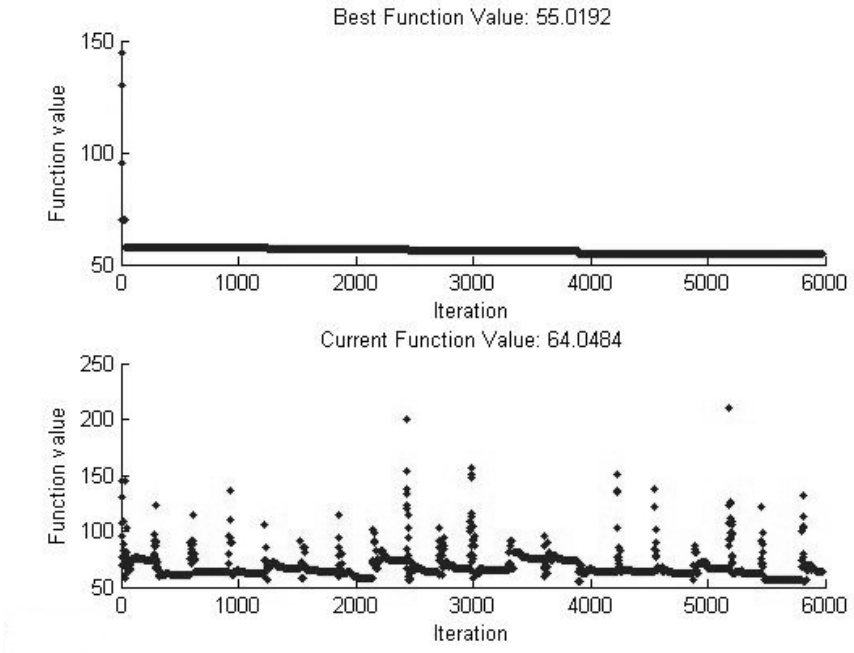


Figure 4.13: Current function value and best function value by simulated annealing

iteration. The function value converges to the optimum with the regulated mesh size by the expansion and contraction factor. The optimal value of objective function can be further reduced by means of using a longer experiment time or by adjusting algorithm parameters appropriately, e.g. mesh size and population size.

In this thesis, since various practical experimental requirements are considered, the objective function will be subjected to different types of constraints which tend to compromise the convexity. Additionally, the experiments are expected to be repeatable for the validation of results. Therefore the pattern search algorithm is selected because it is a global optimization algorithm which is capable of finding the global optimal value and moreover this algorithm is deterministic so that the experiment results can be exactly reproduced with the same initial conditions. Other global optimization algorithms such as the simulated-annealing and genetic algorithm are not employed since a stochastic population is involved, which makes the experiment unrepeatable. Table 4.3 shows the objective function value (J_A) of a UDRN signal and the optimal input. The diagonal elements of M^{-1} represent the low bounds of parameters as shown in column 4 and 6 of Table 4.4.

Table 4.3: Objective function value (A-optimal criterion) of UDRN input and optimal input

	$J_A = \mathrm{tr}(M^{-1})$
UDRN signal	4944.02
Optimal signal	1671.8

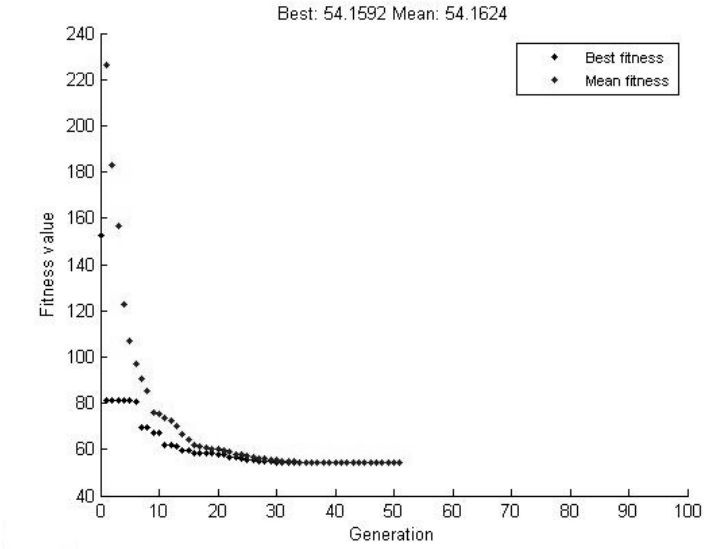


Figure 4.14: Objective function value by genetic algorithm

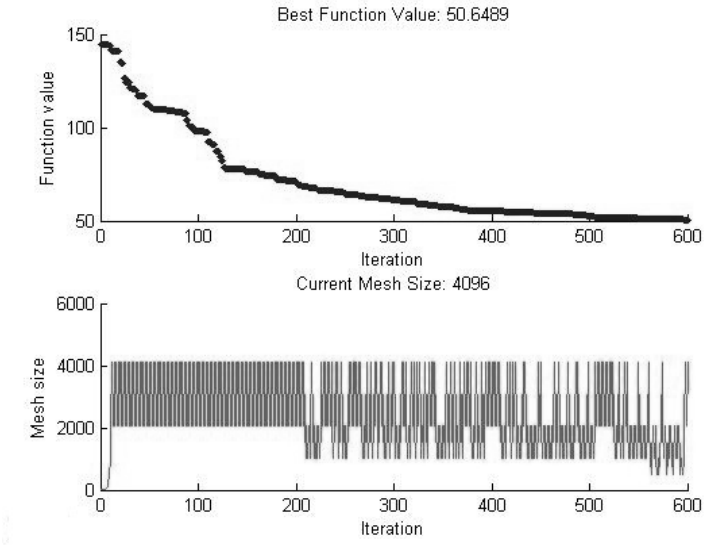


Figure 4.15: Objective function value and mesh size of pattern search

Table 4.4: Estimation results by UDRN input and A-optimal input

	θ	$\hat{\theta}_{UDRN}$	Lower bound (b_1)	$\hat{\theta}_{Aop}$	Lower bound (b_2)	b_2/b_1
1	64.17	77.27	23.16	45.03	16.46	0.71
2	-0.025	-0.034	0.0056	-0.0133	0.0095	1.70
3	0.062	0.070	0.0099	0.0364	0.0167	1.69
4	-4.06×10^{-7}	1.31×10^{-6}	2.39×10^{-6}	7.90×10^{-7}	2.06×10^{-6}	0.86
5	2.73×10^{-6}	2.82×10^{-6}	7.41×10^{-7}	2.31×10^{-6}	7.97×10^{-7}	1.08
6	-0.0069	-0.0034	0.0093	0.0103	0.0163	1.75
7	8.50×10^{-7}	2.42×10^{-6}	1.85×10^{-6}	-1.01×10^{-7}	1.95×10^{-6}	1.05
8	1.54×10^{-6}	1.63×10^{-6}	6.32×10^{-7}	6.887×10^{-7}	8.81×10^{-7}	1.39
9	-327.70	-135.99	94.47	-247.13	54.77	0.58
10	0.0016	-0.027	0.0167	0.0103	0.0185	1.10
11	-1.52×10^{-5}	-1.24×10^{-5}	7.07×10^{-6}	-2.09×10^{-5}	1.15×10^{-5}	1.62
12	-4.22×10^{-8}	3.89×10^{-7}	1.69×10^{-6}	-4.26×10^{-7}	2.26×10^{-6}	1.34
13	0.069	-0.0127	0.041	0.064	0.0523	1.27
14	-0.014	-0.0156	0.0074	-0.013	0.0115	1.55
15	-1.56×10^{-6}	1.40×10^{-5}	8.20×10^{-6}	8.54×10^{-6}	9.82×10^{-6}	1.20
16	-2.25×10^{-6}	-3.63×10^{-6}	1.61×10^{-6}	-1.72×10^{-6}	1.73×10^{-6}	1.07
17	0.029	-0.004	0.0241	-0.0068	0.0321	1.33
18	5.67×10^{-7}	7.53×10^{-6}	2.99×10^{-6}	-3.25×10^{-7}	4.19×10^{-6}	1.40
19	6.4×10^{-7}	-2.35×10^{-6}	1.89×10^{-6}	1.29×10^{-7}	2.24×10^{-6}	1.19
20	0.0021	0.0017	0.0016	0.0039	0.0027	0.96
21	8.71×10^{-7}	4.20×10^{-7}	3.98×10^{-7}	8.34×10^{-7}	4.09×10^{-7}	1.03
22	-1.36×10^{-6}	-1.24×10^{-6}	3.69×10^{-7}	-1.56×10^{-6}	3.72×10^{-7}	1.01
23	0.16	0.0928	0.0727	0.0863	0.0983	1.35

The UDRN input and obtained optimal input are applied to the original model in equation 4.17 and 2 sets of input-output data for identification are recorded. With the predetermined model structure, the results of OLS parameter estimation are shown in column 3 and 5 of Table 4.4. The results in Table 4.3 and 4.4 indicate that although the objective function, the sum of parameter variance is minimized, this cannot ensure that the lower bound of each individual parameter is minimized. As shown in column 7 of Table 4.4, only 4 of 23 individual lower bounds are minimized by the optimal input and the average individual lower bound for the A-optimal input is 122.73% of that of the UDRN signal.

This problem is caused by the complexity of the model structure and the different scales of output sensitivities. For the model shown in equation (4.17), the output sensitivities can

be calculated according to equation (2.6), which are given by:

$$\begin{aligned}
\frac{\partial y(t)}{\partial \theta_1} &= 1 + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_1} \\
\frac{\partial y(t)}{\partial \theta_2} &= u_1(t) + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_2} \\
&\vdots \\
\frac{\partial y(t)}{\partial \theta_{22}} &= u_1(t-6)u_1(t) + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_{22}} \\
\frac{\partial y(t)}{\partial \theta_{22}} &= y(t-10) + \theta_{23}\frac{\partial y(t-10)}{\partial \theta_{23}}
\end{aligned} \tag{4.31}$$

It is shown that in a nonlinear dynamic model, the values of output sensitivities are affected by the regressors and the regressors are often self-related or cross-related e.g. the relation between $u_1(t-1)$ and $u_1(t-3)$ or between $u_1(t-1)$ and $u_1(t-1)u_2(t-3)$. Therefore in an optimization problem, where the objective function is a summation of a few scalar sub-functions of variables:

$$\mathrm{tr}(M^{-1}) = \sum_{i=1}^{n} cov(\theta_i) = \sum_{i=1}^{n} f_i(u_1, u_2, u_3, y) \tag{4.32}$$

Reducing the value of a sub-function f_i by changing the value of the variables may lead to an increased value of another sub-function. Since the sum of individual parameter variance is minimized in A-optimal design, sub-functions with a large scale tend to be over minimized at the expense of increasing the value of those with a small scale.

In practical applications, the input signals are usually normalized before the identification. This transformation is helpful to reduce the influence caused by the different scales of inputs. However in the optimal input design, taking the A-optimal criterion as an example, the objective function is the sum of individual sub-functions (parameter variance) which are directly determined by the output sensitivities. In models without output regressors, the output sensitivities are only determined by the inputs but if any output regressor is included, the sensitivities will also be affected by this term. Therefore in this thesis, the disadvantage of A-optimal criterion is solved by weighting the output sensitivities. The influence of normalizing the inputs in optimal input design will be studied in further research.

4.7.4 Design of Weighted A-optimal Criterion

To solve the problem mentioned above, a weighted A-optimal criterion min $\mathrm{tr}(WM^{-1})$ is proposed. The individual parameter variance can be weighted in accordance with specific experimental requirements, which provides an improved flexibility in the optimal design. Various weighting functions can be designed with prior knowledge of the relative importance

of parameters. Generally, a parameter which is considered to be important should be heavily weighted. In this thesis, the parameters are assumed to have equal importance and a weighted objective function which tends to give reduced variance of each individual parameter is proposed.

According to equation (4.23), the information matrix can be expanded as:

$$M = \frac{1}{\sigma^2}\sum_{t=1}^{N}\begin{vmatrix} \left(\frac{\partial y(t)}{\partial \theta_1}\right)^2 & \frac{\partial y(t)}{\partial \theta_1}\frac{\partial y(t)}{\partial \theta_2} & \cdots & \frac{\partial y(t)}{\partial \theta_1}\frac{\partial y(t)}{\partial \theta_n} \\ \frac{\partial y(t)}{\partial \theta_2}\frac{\partial y(t)}{\partial \theta_1} & \left(\frac{\partial y(t)}{\partial \theta_2}\right)^2 & \cdots & \\ \vdots & \vdots & \ddots & \\ \frac{\partial y(t)}{\partial \theta_n}\frac{\partial y(t)}{\partial \theta_1} & \cdots & & \left(\frac{\partial y(t)}{\partial \theta_n}\right)^2 \end{vmatrix} \tag{4.33}$$

The inverse of M is thus given by:

$$M^{-1} = \frac{1}{\det(M)}\text{adj}(M) = \frac{1}{\det(M)}\begin{vmatrix} C_{11} & & \\ & \ddots & \\ & & C_{nn} \end{vmatrix} \tag{4.34}$$

where adj(M) is the adjoint matrix and C_{kk} is the cofactor. As shown in equations (4.33) and (4.34), the kth diagonal element of M^{-1} is related to all output sensitivities except the kth. Therefore the proposed weighting function weights the individual diagonal elements of M^{-1} with the corresponding squared output sensitivity term:

$$J_{WA} = \text{tr}(WM^{-1}) = \sum_{k=1}^{n} M_{kk}^{-1}\left\|\frac{\partial Y}{\partial \theta_k}\right\|^2 \tag{4.35}$$

where $\left\|\frac{\partial Y}{\partial \theta_k}\right\|^2$ denotes the squared norm of the kth output sensitivity term which is an N$\times$1 vector. Comparing Table 4.5 with Table 4.4, most of the lower bounds are reduced by using the weighted A-optimal criterion and the average b_2/b_1 is 81.37%.

4.7.5 Design of D-optimal Criterion

The D-optimal criterion minimizes the determinant of the inverse information matrix. Compared to the A-optimal criterion, it has the advantage that the scale change of the parameters will not affect its effectiveness. A commonly used form of D-optimum is given by:

$$J_D = -\ln(\det(M)) \tag{4.36}$$

Using J_D as the objective function, the D-optimal input is acquired. As shown in Table 4.6 and 4.7, although the sum of lower bounds of the D-optimum result is larger than that of the A-optimum, the improvement in individual lower bounds is significant. The average individual lower bound for the D-optimal input is 74.39% of that for the UDRN signal and it is 60.61% compared to that for the A-optimal input.

Table 4.5: Estimation results by UDRN input and WA-optimal input

	θ	$\hat{\theta}_{UDRN}$	Lower bound (b_1)	$\hat{\theta}_{WAop}$	Lower bound (b_2)	b_2/b_1
1	64.17	77.27	23.16	88.12	14.15	0.61
2	-0.025	-0.034	0.0056	-0.031	0.0047	0.84
3	0.062	0.070	0.0099	0.070	0.0091	0.92
4	-4.06×10^{-7}	1.31×10^{-6}	2.39×10^{-6}	4.32×10^{-7}	1.47×10^{-6}	0.62
5	2.73×10^{-6}	2.82×10^{-6}	7.41×10^{-7}	3.35×10^{-6}	7.29×10^{-7}	0.98
6	-0.0069	-0.0034	0.0093	0.0062	0.0091	0.98
7	8.50×10^{-7}	2.42×10^{-6}	1.85×10^{-6}	2.97×10^{-6}	1.51×10^{-6}	0.82
8	1.54×10^{-6}	1.63×10^{-6}	6.32×10^{-7}	1.61×10^{-6}	6.51×10^{-7}	1.03
9	-327.70	-135.99	94.47	-319.57	57.31	0.61
10	0.0016	-0.027	0.0167	-0.019	0.011	0.66
11	-1.52×10^{-5}	-1.24×10^{-5}	7.07×10^{-6}	-1.36×10^{-5}	4.70×10^{-6}	0.66
12	-4.22×10^{-8}	3.89×10^{-7}	1.69×10^{-6}	-3.55×10^{-9}	1.56×10^{-6}	0.92
13	0.069	-0.0127	0.041	0.049	0.026	0.63
14	-0.014	-0.0156	0.0074	-0.023	0.0064	0.86
15	-1.56×10^{-6}	1.40×10^{-5}	8.20×10^{-6}	1.49×10^{-7}	4.38×10^{-6}	0.53
16	-2.25×10^{-6}	-3.63×10^{-6}	1.61×10^{-6}	-3.58×10^{-6}	1.41×10^{-6}	0.87
17	0.029	-0.004	0.0241	0.021	0.012	0.50
18	5.67×10^{-7}	7.53×10^{-6}	2.99×10^{-6}	1.61×10^{-6}	2.55×10^{-6}	0.85
19	6.4×10^{-7}	-2.35×10^{-6}	1.89×10^{-6}	7.80×10^{-7}	1.65×10^{-6}	0.87
20	0.0021	0.0017	0.0016	0.0015	0.0026	1.62
21	8.71×10^{-7}	4.20×10^{-7}	3.98×10^{-7}	9.48×10^{-7}	3.87×10^{-7}	0.97
22	-1.36×10^{-6}	-1.24×10^{-6}	3.69×10^{-7}	-1.35×10^{-6}	3.78×10^{-7}	1.02
23	0.16	0.0928	0.0727	0.14	0.073	1.00

Table 4.6: Objective function values of UDRN inputs and optimal inputs

	$J_D = -\ln(\det(M))$	$J_A = \mathrm{tr}(M^{-1})$
UDRN signal	-417.16	9460.7
A-optimal signal	-410.87	3271.1
D-optimal signal	-431.68	4852.7

4.7.6 Validation of Optimal Inputs in Parameter Estimation

In order to demonstrate the statistical effectiveness of optimal inputs on parameter estimation, model identifications by UDRN signals and D-optimal signals are repeated 1000 times. Since the white noise term $\epsilon(t)$ is different in value each time, the estimated parameters are therefore different also. The distribution of $\hat{\theta}_1$ is shown in Figure 4.16 as an example. This illustrates that the $\hat{\theta}_1$ estimated by the D-optimal signal clusters around the initial value $\theta_1 = 64.17$ which is closer than the one obtained by UDRN signal therefore it indicates that an estimated parameter by optimal inputs is probabilistically closer to the initial parameter value.

In the example above, it is assumed that the initial value of parameter is equal to the

Table 4.7: Estimation results by UDRN input and D-optimal input

	θ	$\hat{\theta}_{UDRN}$	Lower bound (b_1)	$\hat{\theta}_{Dop}$	Lower bound (b_2)	b_2/b_1
1	64.17	77.27	23.16	64.35	18.44	0.80
2	-0.025	-0.034	0.0056	-0.0211	0.0047	0.84
3	0.062	0.070	0.0099	0.0571	0.0083	0.84
4	-4.06×10^{-7}	1.31×10^{-6}	2.39×10^{-6}	-2.19×10^{-6}	9.79×10^{-7}	0.41
5	2.73×10^{-6}	2.82×10^{-6}	7.41×10^{-7}	2.74×10^{-6}	4.49×10^{-7}	0.61
6	-0.0069	-0.0034	0.0093	-0.0109	0.0086	0.92
7	8.50×10^{-7}	2.42×10^{-6}	1.85×10^{-6}	2.24×10^{-6}	1.03×10^{-6}	0.56
8	1.54×10^{-6}	1.63×10^{-6}	6.32×10^{-7}	1.91×10^{-6}	8.25×10^{-7}	1.31
9	-327.70	-135.99	94.47	-233.33	67.18	0.71
10	0.0016	-0.027	0.0167	-0.0051	0.01	0.60
11	-1.52×10^{-5}	-1.24×10^{-5}	7.07×10^{-6}	-6.60×10^{-6}	5.92×10^{-6}	0.84
12	-4.22×10^{-8}	3.89×10^{-7}	1.69×10^{-6}	2.10×10^{-6}	1.16×10^{-6}	0.69
13	0.069	-0.0127	0.041	0.0326	0.0317	0.77
14	-0.014	-0.0156	0.0074	-0.016	0.008	1.08
15	-1.56×10^{-6}	1.40×10^{-5}	8.20×10^{-6}	-2.78×10^{-6}	3.66×10^{-6}	0.45
16	-2.25×10^{-6}	-3.63×10^{-6}	1.61×10^{-6}	-1.88×10^{-6}	9.31×10^{-7}	0.58
17	0.029	-0.004	0.0241	0.032	0.0116	0.48
18	5.67×10^{-7}	7.53×10^{-6}	2.99×10^{-6}	-2.12×10^{-6}	1.87×10^{-6}	0.62
19	6.4×10^{-7}	-2.35×10^{-6}	1.89×10^{-6}	-1.06×10^{-6}	1.19×10^{-6}	0.63
20	0.0021	0.0017	0.0016	0.001	0.0023	1.44
21	8.71×10^{-7}	4.20×10^{-7}	3.98×10^{-7}	7.02×10^{-7}	2.19×10^{-7}	0.55
22	-1.36×10^{-6}	-1.24×10^{-6}	3.69×10^{-7}	-1.22×10^{-6}	2.30×10^{-7}	0.62
23	0.16	0.0928	0.0727	0.0754	0.0554	0.76

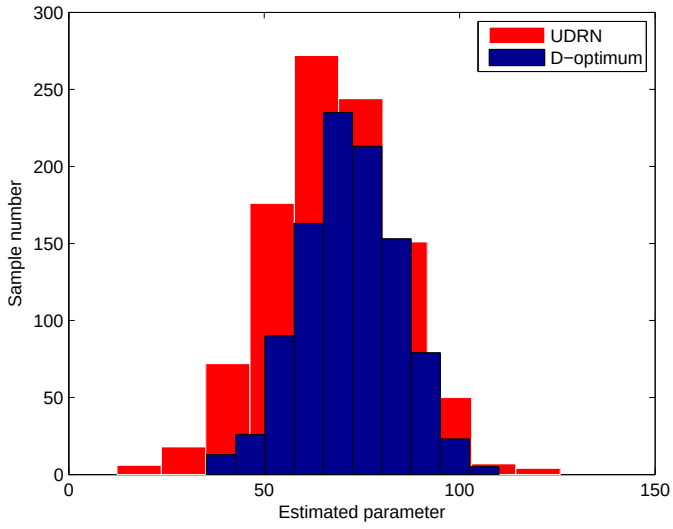


Figure 4.16: Distribution of estimated parameter $\hat{\theta}(1)$

true value. In practice the true parameter value is usually unknown so that the initial value should be determined by a pre-test. The estimated parameter acquired by optimal inputs will

Table 4.8: Iterative parameter estimation with optimal input design

	e
Initial state	2.63
1st iteration	1.11
2nd iteration	0.8
3rd iteration	0.68
4th iteration	0.53
5th iteration	0.5

replace the initial value in the next iteration and the procedure repeated. In the following example, an initial estimation $\hat{\theta}_0$ is derived from a UDRN pre-test and optimal design is carried out iteratively.

$$\begin{aligned}\hat{\theta}_0 \quad = \quad & [74.25, -0.031, 0.068, 7.59 \times 10^{-7}, 2.80 \times 10^{-6}, -0.00466, 2.05 \times 10^{-6}, 1.63 \times 10^{-6}, \\ & -195.34, -0.019, -1.35 \times 10^{-5}, 2.67 \times 10^{-7}, 0.014, -0.015, 9.4 \times 10^{-6}, \\ & -3.24 \times 10^{-6}, 0.0057, 5.53 \times 10^{-6}, -1.47 \times 10^{-6}, 0.0017, 5.65 \times 10^{-7}, -1.30 \times 10^{-6}, 0.11]\end{aligned}$$

Table 4.8 shows the proportion of parameter error in each iteration, which is given by:

$$e = \sum_{i=1}^{n} \left| \frac{\hat{\theta}(i) - \theta(i)}{\theta(i)} \right| \tag{4.37}$$

where $\hat{\theta}$ denotes the estimated value at the current iteration and θ denotes the true parameter value. In this test, the optimal input was designed with θ_0 as the initial conditions in the 1st iteration. The values of parameters were update by system identification using the obtained optimal input and the parameter error was calculated. In the next iteration the updated parameter values were used as the initial conditions in the optimization and the process was repeated for 5 times. As the parameter error becomes smaller gradually by iteration, the estimated parameter is expected to converge to the true value with the iterative optimal input design.

4.8 Optimal Input Design for Improved Output Prediction

As described in last section, optimizations with criteria based on the variance of parameters minimize the lower bound of parameter estimation. Hence the resulting optimal inputs have the effect of giving a parameter estimator with improved accuracy. From the practical point of view, since true parameters of black box models are unknown, it is not feasible to evaluate the effectiveness of optimal input design by directly comparing the estimated value to the true value. Nevertheless, criteria with regard to output prediction can be used to evaluate the accuracy of an estimated model because outputs of black box models can always be measured.

As discussed in the last section, for a system given by equation (4.17), the lower bound of estimated parameter variance is determined by the selected input signal. The objective function for optimization should be a scaler function of M^{-1} which is given by:

$$\begin{aligned} M^{-1} &= E_{Y|\theta}\left[\left(\frac{\partial \ln p(Y|\theta)}{\partial\theta}\right)\left(\frac{\partial \ln p(Y|\theta)}{\partial\theta}\right)^T\right]^{-1} = \left[\frac{1}{\sigma^2}\sum_{t=1}^{N}\left(\frac{\partial y(t)}{\partial\theta}\right)^T\left(\frac{\partial y(t)}{\partial\theta}\right)\right]^{-1} \\ &\leq cov(\hat{\theta}) = E\left[\left(\hat{\theta}-E[\hat{\theta}]\right)\left(\hat{\theta}-E[\hat{\theta}]\right)^T\right] \end{aligned} \tag{4.38}$$

However, the output covariance that needs to be minimized in the output prediction based criteria is given by the form:

$$cov(\hat{Y}) = E\left[\left(\hat{Y}-E[\hat{Y}]\right)\left(\hat{Y}-E[\hat{Y}]\right)^T\right] \tag{4.39}$$

where the predicted output $\hat{Y}$ is affected by the chosen estimation method. Assuming the ordinary least square method is selected for estimation and the OLS parameter estimator is:

$$\hat{\theta} = (X^TX)^{-1}X^TY$$

The parameter covariance and output covariance can be obtained as follows:

$$\begin{aligned} cov(\hat{\theta}) &= E\left[(\hat{\theta}-E[\hat{\theta}])(\hat{\theta}-E[\hat{\theta}])^T\right] \\ &= E\left[\left((X^TX)^{-1}X^T(Y-\hat{Y})\right)\left((X^TX)^{-1}X^T(Y-\hat{Y})\right)^T\right] \\ &= \sigma^2(X^TX)^{-1} \end{aligned} \tag{4.40}$$

$$\begin{aligned} cov(\hat{Y}) &= E\left[\left(X(\hat{\theta}-E[\hat{\theta}])\right)\left(X(\hat{\theta}-E[\hat{\theta}])\right)^T\right] \\ &= XE\left[\left(\hat{\theta}-E[\hat{\theta}]\right)\left(\hat{\theta}-E[\hat{\theta}]\right)^T\right]X^T \\ &= Xcov(\hat{\theta})X^T \end{aligned} \tag{4.41}$$

Therefore the covariance of the predicted output of the input applied for model identification is:

$$cov(\hat{Y}) = \sigma^2X(X^TX)^{-1}X^T \tag{4.42}$$

It is worth noting that equations (4.40), (4.41) and (4.42) are derived under specific pre-conditions i.e. the OLS is employed, the disturbance must be a white noise signal and the input is deterministic. However, the purpose of output error based optimal input design is for practical applications where the system is a black box model so that the pre-conditions cannot be generally guaranteed. Because of this, Mehra [55] proposed a substitution of the covariance of output prediction, which is determined by a first-order expansion of equation (4.39):

$$cov(\hat{Y}) = E\left[\left(\hat{Y}-E[\hat{Y}]\right)\left(\hat{Y}-E[\hat{Y}]\right)^T\right] \geq \left(\frac{\partial Y}{\partial\theta}\right)M^{-1}\left(\frac{\partial Y}{\partial\theta}\right)^T \tag{4.43}$$

4.8.1 Approaches to the Optimization for Output Prediction

If a particular input U_0 is applied to the system and the model of the system is estimated using the designed optimal signal U_{op}, the output covariance between the predicted output $\hat{Y_0}$ and measured output Y_0 can be expressed as:

$$cov(\hat{Y_0}) = \sigma^2 X_0 (X_{op}^T X_{op})^{-1} X_0^T \tag{4.44}$$

where X_{op} is the regressor matrix of the designed optimal signal. X_0 is the regressor matrix of the particular input. Comparing equation (4.42) to (4.44), X_0 and X_{op} have the same structure which is determined by the structure of the initial model. However, U_0 and U_{op} are often different signals so that values of entries of X_0 and X_{op} are not identical. U_0, for which the output prediction error is desired to be minimized, represents the goal of the optimization. U_{op}, by which the model to predict Y_0 is identified, is taken as the signal in the proposed approach to achieve the output error minimisation goal. From the viewpoint of system identification, equation 4.42 should be selected only if the objective of identification is to find a model which gives a minimized output prediction error when using the identification signal itself. Otherwise equation (4.44) should be employed for a model which can accurately predict the output of another specified input U_0. For practical applications, the optimal input design for output prediction can be classified into 2 types:

Optimization for Specific Case

In the simplest case, (equation 4.44) can be applied directly if the objective of the experiment is to predict the output of a particular input accurately. However it lacks practical utility in the real world since the purpose of identifying a model is often to reproduce a type or class of signals rather than a particular one. To solve this issue, a typical input for the specified application can be used as U_0 in the optimal design and the derived optimal input is then able to identify a model which predicts the output in this application with better accuracy than non-optimal inputs. Figure 4.17 shows a closed loop control system and the requirement is to build a model which is qualified to replace the system under the feedback control. In this case the input $e(t)$ over certain sample instants can be used as U_0 and the resulting model may accurately reproduce the output at other sample instants since U_0 is representative of the system behaviour under the specific situation.

Optimization for Global Accuracy

In many applications, the identified model will be utilized for further implementation e.g. as a substitution of the real system in offline controller design therefore the feature of input signal

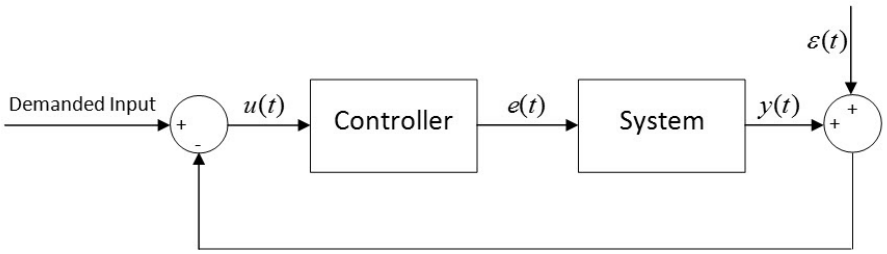


Figure 4.17: U_0 selection in a closed loop control system

to the model in the final implementation is often unknown and consequently it is not feasible to select U_0 as before. A model with global accuracy which gives accurate output predictions against all possible inputs is hence favoured for the compatibility in further design.

A theoretically feasible solution is to explore the entire input space and evaluate the objective function against all possible inputs. A commonly used approach is to represent the input space as several candidate points and the whole input sequence is composed of these points rather than arbitrary values in the input space. The computing burden is remarkably reduced by this approach but the optimization result will be compromised as well with a decreasing number of candidate points. Furthermore, the data length of signal for dynamic system identification tends to be more than hundreds in order to excite the system dynamics. As a result even designing a 100-point 2-level optimal input requires 2^{100} evaluations which demands an extremely long experimental time.

In this thesis, a proposed approach is to choose a signal of a broad frequency content e.g. PRBS, APRBS and UDRN as U_0 and then design the input accordingly. The principle of this method is consistent with a popular identification method which utilizes a white noise or similar signal to estimate a model without any prior knowledge. Because of the wide frequency range of U_0, the identified model can be expected to be globally accurate which will be beneficial for further application. This approach also remarkably reduces the required experimental time. Any design criterion which considers the whole input-output space can then be relaxed to evaluate the sub-space covered by U_0.

4.8.2 Design of I-optimal Criterion

If the purpose of the optimal input design is to accurately predict the output, scalar functions of the output covariance can be selected as the objective function. An objective function designed according to the I-optimal criterion should optimize the sum of the variance of the output prediction over the entire design space and may be simplified to optimize over the sub-space as above. The V-optimal criterion minimises the average function value. In practice, it

is often used as an approximation of I-optimum by giving an averaged value over the space of interest in order to compare the optimization result obtained by other criteria. Therefore it is virtually identical to the I-optimum method in this simplified case.

$$J_I = \sum_{t=1}^{k} cov\left(\hat{y}_0(t)\right) \tag{4.45}$$

where $cov\left(\hat{y}_0(t)\right)$ is the covariance of the predicted output of the objective signal at the time k. Equation (4.44) shows that the dimension of the covariance matrix is identical to the length of the output sequence of selected U_0. In dynamic optimization, the data length tends to be much longer than the number of parameters therefore I-optimal may lead to a very high dimensional optimization problem, which leads to a very high computational burden. Furthermore, in the case when the identification signal U is chosen as the objective signal U_0, it can be proved that the sum of the variance of the entire output sequence is identical to the dimension of the vector of regressors n:

$$\begin{aligned}
\sum_{t=1}^{k} cov(\hat{y}(i)) &= \mathrm{tr}\left(\frac{\partial Y}{\partial \theta} M^{-1} \frac{\partial Y}{\partial \theta}^T\right) \\
&= \sigma^2 \mathrm{tr}\left\{\frac{\partial Y}{\partial \theta}\left(\frac{\partial Y}{\partial \theta}^T \frac{\partial Y}{\partial \theta}\right)^{-1} \frac{\partial Y}{\partial \theta}^T\right\} \\
&= \sigma^2 \mathrm{tr}\left\{\frac{\partial Y}{\partial \theta}^T \frac{\partial Y}{\partial \theta}\left(\frac{\partial Y}{\partial \theta}^T \frac{\partial Y}{\partial \theta}\right)^{-1}\right\} \\
&= \sigma^2 \mathrm{tr}\left(I\right) \\
&= \sigma^2 n
\end{aligned} \tag{4.46}$$

The result indicates that the output variance of any input which is also used to identify the prediction model is a constant and the value of the constant is only determined by the model structure and the covariance of the noise. In optimal input design, the model structure and noise do not change once they are selected, thus if the sum of output variance is chosen as the objective function, the function value will not vary with the variables so that the optimization will fail.

4.8.3 Design of Adapted I-optimal Criterion

According to equation (4.41), the variance of output prediction at data sample instance i can be derived as:

$$\begin{aligned} cov(\hat{y}(t)) &= E\left[(\hat{y}(t) - E[\hat{y}(t)])\,(\hat{y}(t) - E[\hat{y}(t)])^T\right] \\ &= x(t)cov(\hat{\theta})x(t)^T \\ &= [x_1(t), \cdots x_n(t)] \\ &\quad \begin{bmatrix} cov(\hat{\theta}_{11}), & \ldots, & cov(\hat{\theta}_{1n}) \\ cov(\hat{\theta}_{21}), & \ldots, & cov(\hat{\theta}_{2n}) \\ \vdots & \ddots & \vdots \\ cov(\hat{\theta}_{n1}), & \ldots, & cov(\hat{\theta}_{nn}) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \\ &= \sum_{p=1}^{n}\sum_{q=1}^{n} x_p(t)x_q(t)cov(\hat{\theta}_{pq}) \end{aligned} \tag{4.47}$$

where $x(t) = [x_1(t),\ x_2(t),\ \ldots\ x_n(t)]$ is the matrix of regressors at time t. In the thesis, a new optimality criterion based on a weighted trace of the matrix of covariance is proposed, which thereby enjoys similar computational advantages to A-optimality criterion based methods but also approximates the output based approach by the use of only the diagonal elements of the parameter covariance matrix instead of the high dimension output covariance. This use of the diagonal elements will however only be accurate if the regressors are well chosen so the parameter covariances are uncorrelated. In this case, the value of the parameter covariance, $cov(\hat{\theta}_{pq})$ where $p \neq q$, will be very small and the influence of the corresponding term $x_p(t)x_q(t)cov(\hat{\theta}_{pq})$ can be neglected. Therefore the output covariance at data sample instance t will be given by the approximation:

$$\begin{aligned} cov(\hat{y}(t)) &= \sum_{p=1}^{n}\sum_{q=1}^{n} x_p(t)x_q(t)cov(\hat{\theta}_{pq}) \\ &\approx \sum_{p=1}^{n} x_p(t)^2 cov(\hat{\theta}_{pp}) \end{aligned} \tag{4.48}$$

The proposed performance function J_{AI}, to be evaluated over the data of length N is accordingly

$$\begin{aligned} J_{AI} &= \sum_{t=1}^{N}\sum_{p=1}^{n} x_p(t)^2 cov(\hat{\theta}_{pp}) \\ &= \sum_{p=1}^{n}\sum_{t=1}^{N} x_p(t)^2 cov(\hat{\theta}_{pp}) \\ &= \sum_{p=1}^{n} \|x_p\|^2 cov(\hat{\theta}_{pp}) \end{aligned} \tag{4.49}$$

It can be seen from equation (4.49) that for any input, the corresponding output prediction error is affected by both the parameter covariance of the estimated model and the norms of the regressors. The regressor norms appear as a weighting to the variances of the parameter estimates. The weighting of the parameter covariance by the norms of the regressors allows improvement in the output covariance over unweighted parameter variance methods. Moreover if the output covariance is described by the general form in equation (4.43) and an objective signal is determined, the AI criterion correspondingly becomes:

$$J_{AI} = \sum_{p=1}^{n} M_{pp}^{-1} \left\| \frac{\partial Y_0}{\partial \theta_p} \right\|^2 \tag{4.50}$$

where M_{pp}^{-1} denotes the pth diagonal element of the inverse information matrix of the optimal signal and $\frac{\partial Y_0}{\partial \theta_p}$ denote the pth output sensitivity term of the objective signal.

4.8.4 Design of G-optimal Criterion

As a classic criterion, G-optimality searches for a solution which minimizes the maximum function value that can be obtained within a specified variable space. In output prediction based, G-optimal input design refers to a minimization of the maximum variance of the predicted output. It is considered advantageous because the output error can be distributed more evenly over the entire output sequence by this approach. The objective function of G-optimality is the maximum value of the diagonal elements of the output prediction covariance, which is given by:

$$J_G \quad = \quad \max dig \left(cov(\hat{Y}_0) \right) \tag{4.51}$$

4.8.5 Methodology for Statistical Comparison

In much of the literature on optimal test signal design only single cases of the illustrative examples are presented to claim a demonstrated superior performance [43, 100]. However in validating or invalidating any optimisation method for an objective function based on expectation such as in minimised parameter covariance or output covariance presented above, one single good or bad example result is strictly statistically meaningless. Statistical identifications and validations over a significantly sized population of test cases are required. In this work, a pool of models is assembled including models identified with optimal test signals generated with different randomly generated initial conditions. Pools of models are also assembled comprising the models identified using the non-optimal input types, PRBS, APRBS ,UDRN and Random-walk, each again generated with different randomly generated initial conditions. The R^2 is selected as the criterion for validation since the model structure and the data length remain in the same in the tests. A detailed explanation was given in Section

2.7.2. A number of different validation signals are then applied to each of the models in the model pool and the average R^2 of each of the non-optimal models is compared with that of the optimal models.

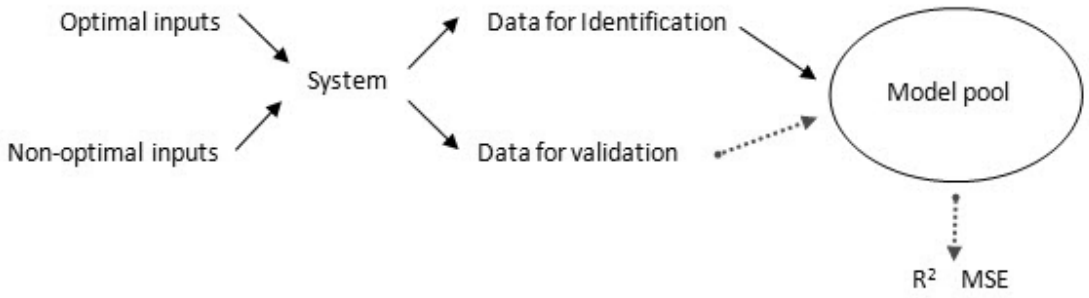


Figure 4.18: Procedure of building model pool and validation

4.8.6 Validation of Optimal Inputs in Output Prediction

In this section, optimal inputs are generated by objective function designed according to the criteria concerning about the output prediction. Models of equation (4.17) are then identified and validated statistically. The process is divided into four steps.

Step 1 Design of objective function

Optimal test signals U_I, U_G and U_{AI} with input amplitude constraints in equation (4.16) are obtained from the minimisation of J_I,J_G and J_{AI} respectively by global optimisation using the pattern search algorithm. An APRBS input is chosen as the U_0 in the criteria. Table 4.9 shows the evaluation of J_I,J_G and J_{AI} performance indices obtained by applying both the optimal signals and also PRBS, APRBS and UDRN inputs to the original model 4.17, each with a sample period of 0.1s, and each of the non-optimal signals scaled to have maxima and minima at the constraint limits of equation (4.16). Figure 4.19 shows examples of the different test signal types.

Comparing with the optimal input in Figure 4.3, the values of optimal inputs generated by the conventional G-optimal I-optimal and our proposed AI-optimal criteria are in multi-levels. It indicates that a binary level signal may not be suitable to identify a nonlinear model as shown in equation (4.17). The time interval (the minimum time period of the input staying in a certain level) of the optimal inputs is 0.1 sec which is the same as the sample time of the model and the values of inputs are changing in the desired range in the time history. Therefore the optimal inputs are considered effective to excite the dynamic behavior of the system. In order to capture more dynamic behaviors of the system, a discrete model with

Table 4.9: objective function values of various inputs against criteria

	U_{PRBS}	U_{APRBS}	U_{UDRN}	U_G	U_I	U_{AI}
J_G	3.58×10^{13}	34.66	133.37	27.37	31.35	39.61
J_I	1.31×10^{15}	1830	4225.6	1727.5	1536.7	1749.6
J_{AI}	3.76×10^{17}	6.86×10^{5}	1.8109×10^{6}	4.56×10^{5}	5.64×10^{5}	4.33×10^{5}

a smaller simple time can be used to represent this system and then the obtained optimal inputs will have a smaller time interval accordingly. However as the sample time is reduced, the length of data collected in the same period of time will increase correspondingly and it will cause heavier computational burden of the optimization.

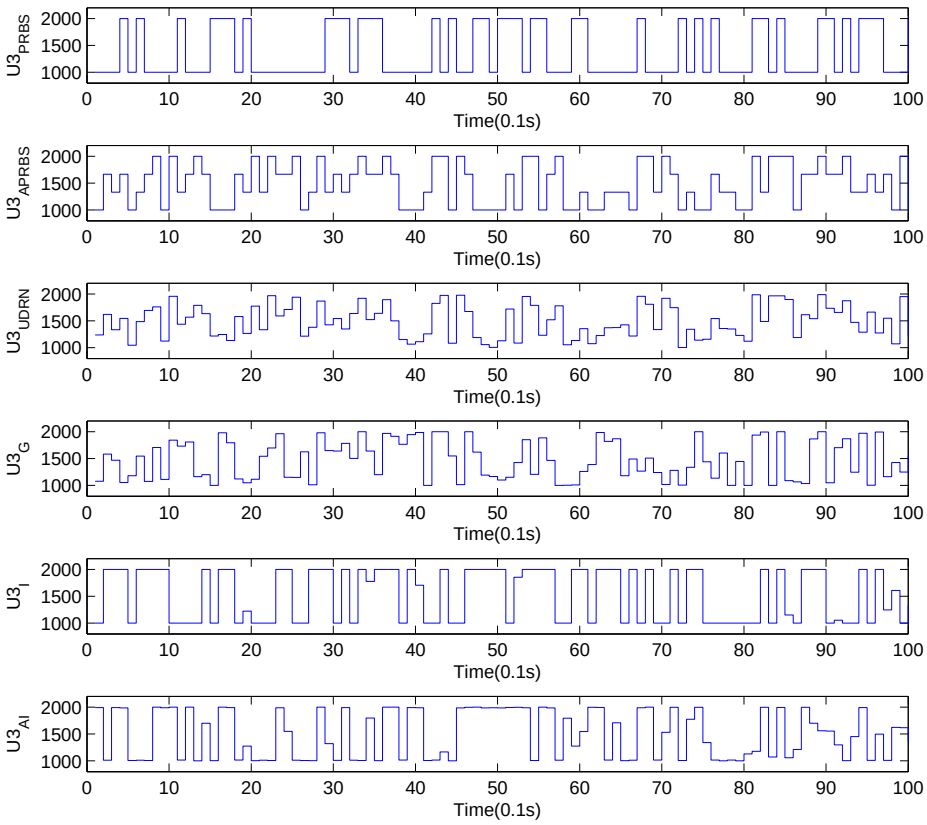


Figure 4.19: An example of test signals of different types

Step 2 Construction of model pool

In order to validate the effectiveness of optimal test signals, models identified using optimal test signals produced by the global optimisation of the J_I,J_G or J_{AI} performance index are compared to those identified by test inputs produced by the PRBS, APRBS and UDRN signals. The test inputs produced by these global ALPS optimizations are repeatable since they are deterministic products of the models and the constraints. The number of models is required to be sufficiently large, usually in the hundreds, to ensure the validation results are statistically significant. This is taken as 10 in this study because of the limit on the available experimental time.

In practice, due to the complexity of the objective function and the limit of numerical searching algorithms, the solution by the global optimization algorithm that runs within an allowable experimental time may converge to a local optimum, in other words, a global optimal value cannot be guaranteed. Therefore, the obtained optimal test signals may vary with the different initial conditions in the global optimisation algorithm. Consequently a set of 10 different optimal test signals U_{AI} optimizing J_{AI} can be assembled by using 10 different initial optimisation conditions. The initial conditions used are UDRN sequences each generated by a unique seed. A pool of 10 models are then assembled by identification with the 10 different optimal test signals U_{AI}. Similarly 10 different optimal test signals U_I or U_G optimising J_I or J_G are used to assemble another two pools of 10 different identified models. For comparison 10 PRBS, 10 APRBS and 10 UDRN each with different seeds are used to identify 10 different models for each signal type.

Step 3 Selection of validation signals

To validate the model pools associated with each type of test signal, a set of validation signals are applied to each model in each of the pools. For each model, the resulting output is compared with the output from the same signal applied to the original system in order to produce an output fitness measured by the R^2 criterion. It should be noted that since the performance indices for J_I,J_G and J_{AI} are based on expectation, any associated optimal test signal is only guaranteed to be superior to a non-optimal test signal in the mean. For a statistically fair test, the validation tests should be uncorrelated. To test the qualification of validation signals, the correlation coefficient in equation (2.45) is consequently determined for each pair of signals as:

$$\begin{aligned} r_{U_i,U_j} &= \frac{cov(U_i,U_j)}{\sigma_{Ui}\sigma_{Uj}} \\ &= \frac{E(U_iU_j)-E(U_i)E(U_j)}{\sqrt{E(U_i^2)-E^2(U_i)}\sqrt{E(U_j^2)-E^2(U_j)}} \end{aligned}$$

Table 4.10: mean correlation coefficients of validation signals

	$\overline{CC_{PRBS}}$	$\overline{CC_{APRBS}}$	$\overline{CC_{UDRN}}$	$\overline{CC_{U_I}}$	$\overline{CC_{U_G}}$	$\overline{CC_{U_{AI}}}$
U_1	0.091	0.079	0.082	0.097	0.132	0.098
U_2	0.092	0.097	0.091	0.082	0.100	0.081
U_3	0.103	0.086	0.097	0.094	0.087	0.095

Table 4.11: mean R^2 of models identified by input constraints

Model	$\overline{R^2_{PRBS}}$	$\overline{R^2_{APRBS}}$	$\overline{R^2_{UDRN}}$	$\overline{R^2_{U_G}}$	$\overline{R^2_{U_I}}$	$\overline{R^2_{U_{AI}}}$
M_{PRBS}	-449.75%	-887.39%	-1417.50%	-817.37%	-388.85%	-646.13%
M_{APRBS}	72.06%	61.49%	47.55%	64.07%	78.30%	66.58%
M_{UDRN}	57.36%	51.84%	42.08%	54.38%	70.68%	54.91%
M_{U_G}	77.37%	65.09%	50.63%	68.11%	81.38%	70.85%
M_{U_I}	79.84%	65.72%	48.49%	67.56%	83.37%	72.35%
$M_{U_{AI}}$	78.73%	66.49%	51.65%	68.19%	82.75%	72.76%

The set of validation signals is then selected so that the correlation between each pair combination is close to 0. In this work, sets of 10 validation signals are assembled for each of the J_I,J_G and J_{AI} optimal input test signals, and the non-optimal PRBS, APRBS and UDRN signals. The 10 signals for J_I,J_G and J_{AI} denoted U_I,U_G and U_{AI}, are obtained by varying the initial conditions of the optimisation by setting these as random sequences. The 10 signals U_{PRBS}, U_{APRBS} and U_{UDRN}, are obtained by varying their seeds. The mean correlation coefficients (CC) of the different validation signal sets are shown in Table 4.10.

Step 4. Validation results

The validation signals are applied to each identified model and Table 4.11 shows validation results measured by the R^2. Since the model identification and validation are repeated for 10 times in each case, the averaged R^2 is shown in the table. Other criteria, such as the distribution of R^2 could also be employed however the mean of R^2 is selected in this thesis since it is easier to be presented in tables. In this table, the y axis denotes models identified by various types of inputs and the x axis denotes the R^2 obtained by applying validation input to estimated models. Since an APRBS input was selected as the U_0 for output space based input design, models identified by optimal inputs should be able to regenerate outputs of APRBS inputs with other seeds more accurately than models identified by non-optimal inputs, including M_{APRBS}. Moreover they are also expected to reproduce all types of outputs better as the U_0 has a frequency content with a large range. These two features are evidenced by the validation results shown the table.

The result highlights the known general unsuitability of PRBS inputs for nonlinear identification [36] since the R^2 of the model identified by PRBS signal is considerably worse

than the others. According to equation (2.48), since the prediction error $\|Y - \hat{Y}\|^2$ is larger than the $\|Y - \overline{Y}\|^2$, the values of the R^2 of PRBS are negative. The R^2 of M_{UDRN} is sensible but the second smallest. A reasonable explanation is that its amplitude density is smaller so that it provides less information over the input range with a fixed data length. Since APRBS has a higher amplitude density than UDRN, M_{APRBS} gives an improved R^2 from 5% to 15%. M_{U_G}, M_{U_I} and $M_{U_{AI}}$ are recognized to be models with the best quality since they give the highest R^2, 3% to 5% further improvement than R_{APRBS} no matter which type of validation signal is selected. Hence the output space based optimal input design is proved to be effective. Since the R^2 of these 3 types of models are not significantly different and the stochastic of the validation needs to be considered, it is premature to give a general conclusion concerning about which input design criterion leads to the most accurate model by judging the R^2. However J_{AI} is suggested as the first choice for optimal input design as it leads to a considerably smaller computing burden than J_G and J_I. Optimal inputs designed according to conventional G-optimal, I-optimal and the proposed AI-optimal criterion were generated using the Matlab Optimization Toolbox with the same number of function evaluations, 50000 in the selected pattern search algorithm. The averaged time to obtain the optimal inputs are 1205s, 1186s and 1032s. The computational speed of the AI-optimal criterion is more than 10% faster than the other criteria.

4.9 Influences of Experimental Constraints and Disturbance

Although maximizing the data information is the general purpose of DoE, excessive long or powerful inputs are not acceptable test signals because the constraints on the practical experimental conditions should be considered [41]. In previous experiments of input design, amplitude constraints on inputs and available experimental time have been taken into account. For nonlinear dynamic systems, two other commonly used constraints are imposed on optimal input design and the effect on output prediction is discussed as follows.

4.9.1 Optimization with Output Amplitude Constraints

Output amplitude constraints are utilized to limit the predicted output in the allowable region so as to prevent undesired dynamics in real systems. Although a model with good quality is required for accurate simulation, it is still sensible to implement a conservative output amplitude constraint in the first iteration.

In the case of systems with true linear behaviour, at the expense of reducing input amplitude, output constraints can be satisfied by directly scaling the input signal. However in general, experimentally investigated systems are usually nonlinear systems and will not

Table 4.12: mean R^2 of models identified by input and output constraints

Model	$\overline{R^2_{PRBS}}$	$\overline{R^2_{APRBS}}$	$\overline{R^2_{UDRN}}$	$\overline{R^2_{U_{AI}}}$
M_{PRBS}	-5.66%	-500.05%	-472.40%	-698.08%
M_{APRBS}	40.61%	49.61%	45.63%	54.67 %
M_{UDRN}	29.69%	42.85%	42.83%	38.00 %
$M_{U_{AI}}$	48.15%	54.47%	49.54%	72.41 %

have associative input-to-output characteristics as linear systems, making tuning the input amplitude much more difficult. Exploring the full extent of the input-output signal envelope with sufficient data information is then a challenging practical problem. Given an accurate output prediction model of the system, an optimised input can explore the maximal input space envelope without violations of the input and output constraints.

In the example of output prediction based input design, only amplitudes of inputs are constrained. If an output constraint is added, since it is necessary to maximise signal information, the test signals must be adjusted to satisfy the output limits. Now it can be relatively time consuming to make appropriate PRBS, APRBS and UDRN inputs since the amplitudes of these signals need to be adjusted manually and the nonlinearity may make this process difficult. However, optimal inputs designed with specified input and output constraints can be obtained directly. In the following optimal input design, the input and output constraints are taken as:

$$\begin{aligned} 1800\mu s &< u_1 < 6200\mu s \\ 35\% &< u_2 < 65\% \\ 800RPM &< u_3 < 2200RPM \\ -10Nm &< y < 40Nm \end{aligned}$$

The feasible non-optimal inputs are obtained by trial and error test on the prediction model. Table 4.12 shows the validation results. Compared to other inputs satisfying the output constraints, the optimal input designed by the proposed performance index J_{AI} leads to a better output fitness in R^2, 4%-18% better than the output fitness of the model identified by the second best identification signal. It should note that using the optimal input for system identification can improve the accuracy of the obtained model however it cannot guarantee that the accuracy of the resulting model could always meet the requirement for industrial implementations. Therefore other DoE methods, e.g. model structure selection, can be employed to further refine the model accuracy. However since many different methodologies must be studied in depth and a large amount of experiments needs to be conducted, the further improvement is not discussed in this thesis.

Table 4.13: mean R^2 of models identified by input and rate constraints

Input range	Model	$\overline{R^2_{rw}}$	$\overline{R^2_{U_{rAI}}}$
9 Levels	M_{rw}	-47.68%	-32.53%
	M_{UrAI}	19.18%	-11.49%
5 Levels	M_{rw}	9.23%	5.05%
	M_{UrAI}	32.01%	31.18%

4.9.2 Optimization with Input Rate Constraints

In true linear systems, models can be obtained by local testing, typically with small magnitude PRBS or other binary signals. In the case of nonlinear systems however, binary test input signals will generally be untypical of actual operation and result in significantly poor output fitness and the exceedence of operational limits. The use of binary test signals on nonlinear systems also risks losing the system identifiability [36]. Although APRBS and U-DRN inputs can overcome this issue since they provide values at multi-levels, the input may still experience drastic raise or fall and this is not allowed in experiments which have limits on input rate of changing. In order to prevent a typical system dynamics caused by high input gradients, smooth or rate-limited input signals are usually recommended for nonlinear system identifications with rate constraints.

Rate constrained random-walk inputs (U_{rw}) are obtained from initial values with increments in random directions in each step. In sequentially assembling the test signal, the direction of the next increment is changed by reversing the direction if the values of the signal at the current step exceed the amplitude constraints. In a real experimental based engine identification, the input increment size should be decided according to physical rate limits. For instance, the engine speed cannot increase or decrease too quick because of the load and inertia. For the purposes of this study, the whole input constraint range shown in equation (4.16) is divided into 9 and also 5 parts in order to show the effect on the identification results. An example of random-walk inputs and optimal smooth inputs designed for $J_{AI}(U_{rAI})$ with rate constraints $\triangle u_1 = 500\mu s$, $\triangle u_2 = 2.5\%$ and $\triangle u_3 = 125$RPM are shown in Figure 4.20. Models identified by random-walk inputs (U_{rw}) and smooth optimal inputs (U_{rAI}) are validated by 10 other constrained random-walk signals and 10 other U_{rAI} signals. The validation results for the different input range partitions are shown in Table 4.13. As can be seen, the optimal smooth input designed by J_{AI} always produces the best results.

Besides typical constraints, other linear and nonlinear constraints can be designed and added also. However, the validation result will generally be traded off with any additional constraint since the feasible region of input and output shrinks. Although the region will be subsequently reduced, constraints may lead to more computing burden because they will be converted to additional terms of an unconstrained optimization. Therefore, a subset of

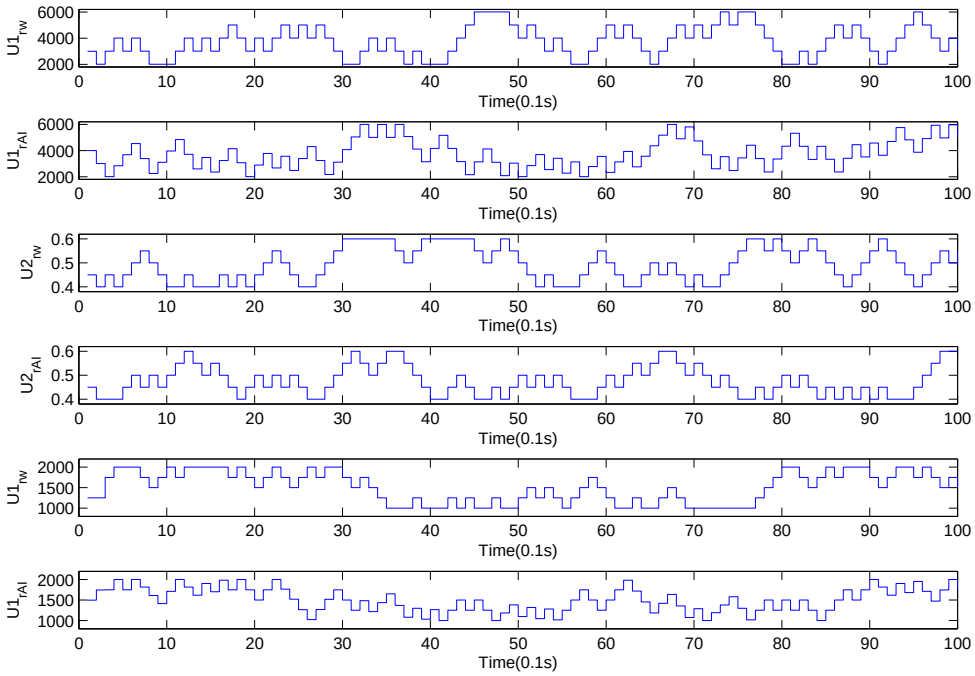


Figure 4.20: An example of the rate constrained random walk signal and optimal signals

relevant constraints should be established and subsequently checked for violation of other constraints before the optimization. Repeated or conflicting constraints need to be removed in order to simplify the optimization and relax the feasible region.

4.9.3 Influence of Disturbance on Optimization

As shown in equation (4.17), the model is disturbed with a white noise signal. Theoretically, according to equation (4.23), if the covariance of disturbance is zero, the data information will be infinitely rich for any input signal hence no input design is needed. On the contrary, optimal inputs are expected to produce better effect than non-optimal inputs in situations where the disturbance has a large covariance.

Tables 4.14 and 4.15 show the validation results of different original models with disturbances for $\sigma^2 = 40$ and $\sigma^2 = 160$. Compared with the experiments tested against the original model with disturbance $\sigma^2 = 80$, the R^2 increases or decreases as expected, however, the relative benefits of optimal inputs are better exhibited since the relative increments in R^2 for the optimal cases become even larger with a stronger disturbance, e.g. the improvement between M_{AI} and M_{APRBS} becomes 8%-25% with $\sigma^2 = 160$. As the disturbance represents the unknown and stochastic nature of the model, it implies that the optimal input design

Table 4.14: mean R^2 of models identified by input constraints with $\sigma^2 = 40$

Model	$\overline{R^2_{PRBS}}$	$\overline{R^2_{APRBS}}$	$\overline{R^2_{UDRN}}$	$\overline{R^2_{U_G}}$	$\overline{R^2_{U_I}}$	$\overline{R^2_{U_{AI}}}$
M_{PRBS}	-249.36%	-636.69%	-1198.01%	-585.38%	-222.17%	-409.60%
M_{APRBS}	84.13%	76.89%	65.69%	78.89%	87.75%	80.38%
M_{UDRN}	76.13%	71.46%	62.60%	73.55%	83.73%	73.86%
M_{U_G}	87.60%	79.55%	68.32%	81.62%	89.88%	83.45%
M_{U_I}	88.64%	79.13%	65.24%	80.45%	90.74%	84.16%
$M_{U_{AI}}$	88.24%	80.04%	68.39%	81.31%	90.51%	84.32%

Table 4.15: mean R^2 of models identified by input constraints with $\sigma^2 = 160$

Model	$\overline{R^2_{PRBS}}$	$\overline{R^2_{APRBS}}$	$\overline{R^2_{UDRN}}$	$\overline{R^2_{U_G}}$	$\overline{R^2_{U_I}}$	$\overline{R^2_{U_{AI}}}$
M_{PRBS}	-205.15%	-456.58%	-701.50%	-444.11%	-182.16%	-321.48%
M_{APRBS}	52.60%	40.67%	27.64%	43.31%	46.31%	62.71%
M_{UDRN}	27.37%	25.41%	19.56%	27.65%	27.26%	49.30%
M_{U_G}	60.80%	45.52%	31.22%	49.18%	67.34%	52.26%
M_{U_I}	65.51%	47.65%	30.23%	49.63%	71.08%	56.42%
$M_{U_{AI}}$	63.41%	48.18%	33.21%	49.90%	69.93%	55.76%

might be also useful for the identification of a black box model in which the initial model cannot perfectly present the true system due to the unavoidable uncertainty.

4.10 Optimal Input Design for Black Box Modelling

In previous sections, the effectiveness of optimal input design is validated by applying this method to identify a known system, an engine model with known model structure and parameter values. There are two main reasons for evaluating the optimal input design initially on a known system.

1. Suitable model equations to optimal input design theories

In the analysis of optimal input design theories, many conclusions are obtained under specific assumptions. For example, the information matrix can be exactly expressed as equation (4.23) only if the system is described in equation (2.3). An engine model can be built as in the form of equation (2.3) in order to satisfy the assumptions of input design. The theoretical effectiveness of the criteria of optimal input design can be verified if the evaluations are conducted on the known system.

2. Known true model structure and parameter

If the true model structure and parameters are available, they can be utilized in the initial model estimation for a relatively accurate model (without a disturbance term in the output) then an optimal input which leads to a model with much better fitness than obtained with a non-optimal input might be achieved without further iterations. Moreover it will be feasible to judge the effectiveness of parameter based design criteria by analysing the error between the true parameter and estimated parameter.

However most practical systems are the black box models therefore the quality of the model is often evaluated by how well it can reproduce the output and so output space based input design criteria are favoured. One potential issue for input design of black box models is that the accuracy of the initial model estimation might be compromised without knowing the true model structure and parameter. However, this can be solved by using suitable regressor, input data and estimation methods. An example of input design for black box modelling and evaluation of its effectiveness is demonstrated in the following.

4.10.1 Initial Model Estimation

For evaluation purposes in this section, the virtual engine is considered as the black box system to be identified and the objective is to identify a torque model with high quality where the inputs are engine speed (u_1), spark advance (u_2) and throttle angle (u_3) with the amplitude constraints:

$$\begin{aligned} 2000RPM &< u_1 < 4000RPM \\ 10^\circ &< u_2 < 30^\circ \\ 2^\circ &< u_3 < 8^\circ \end{aligned}$$

The model structure is selected as the affine model:

$$y(t) = \theta_1 + \theta_2 u_1(t-1) + \theta_3 u_1(t-2) + \theta_4 u_2(t-1) + \theta_5 u_3(t-1) + \theta_6 u_3(t-2) \quad (4.52)$$

The sample time is 0.3 sec and the parameters are estimated by OLS method with a UDRN signal of 200-point length:

$$\hat{\theta} = [\hat{\theta_1}, \hat{\theta_2}, ..., \hat{\theta_6}] = [2.84, -0.0046, -6.96 \times 10^{-4}, -0.045, 3.19, 0.062]$$

Figure 4.21 shows the measured output and simulated output by the identified model. The corresponding R^2 is 92.17% which indicates this affine model is of high accuracy and implies that the relationship between the inputs and outputs is quite linear in this case.

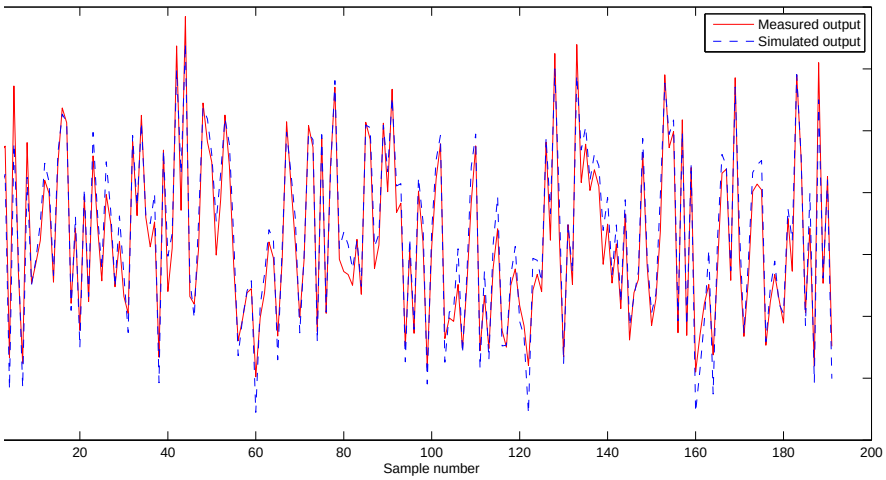


Figure 4.21: Measured output and simulated output of black box torque model

4.10.2 Optimal Input Design and Validation

Selecting a UDRN signal as U_0 and the proposed AI-optimum as the design criterion, the weighting vector is derived according to equation (4.50) as [190, 1.79×10^9, 1.79×10^9, 7.95×10^4, 5.26×10^3, 5.25×10^3]. Figure 4.22 illustrates the difference between a UDRN signal and the optimal input. Although the optimal input looks similar to a PRBS input, it is proved to be more informative than a PRBS input in validation. Using 10 other UDRN signals for validation, models identified by optimal inputs give an averaged result of $\overline{R^2} = 94.01\%$ while models identified by UDRN and PRBS signals only give $\overline{R^2} = 91.50\%$ and $\overline{R^2} = 92.95\%$. Since the optimal input leads to a model which is more accurate than the initial estimated model, it clearly proves the effectiveness of input design for black box modelling and demonstrates the feasibility of implementing input design in practical applications. Moreover since the accuracy of models obtained by the optimal input and PRBS input is better than those by the UDRN input, it indicates that binary signals are more effective than multi-level signals to identify linear systems [34].

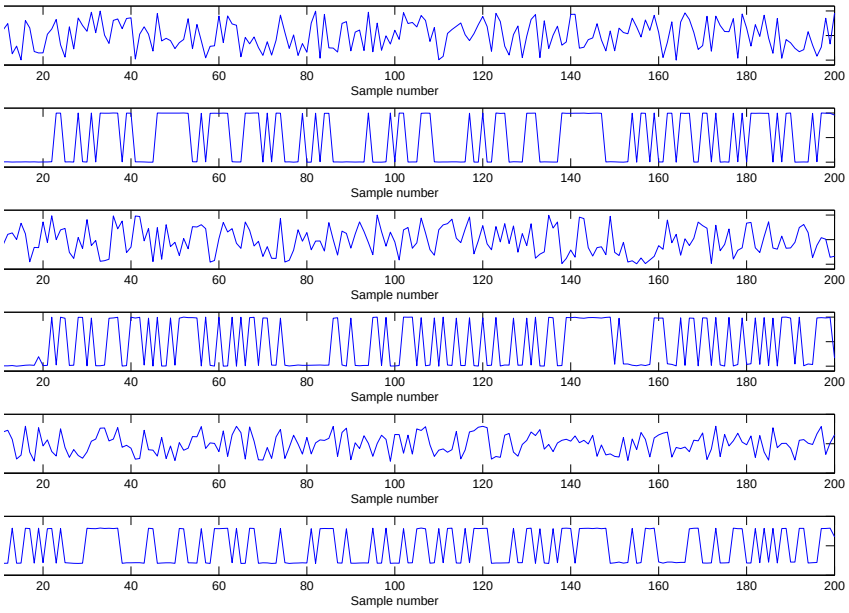


Figure 4.22: An example of UDRN signal and optimal signal

4.11 Optimal Input Design of a MIMO System

Besides the torque model given in equation (4.17), a nonlinear λ model is used as the other component of the 2×2 MIMO model:

$$\begin{aligned} y(t) &= \theta_1 + \theta_2 u_2(t-10)^2 + \theta_3 u_1(t-10)u_3(t-10) + \theta_4 u_1(t)u_3(t-7) \\ &\quad +\theta_5 u_2(t-10)u_3(t) + \theta_6 u_1(t-10)u_2(t-10) + \theta_7 u_1(t)u_3(t-10) + \theta_8 u_1(t)^2 \\ &\quad +\theta_9 u_1(t-10) + \theta_{10} u_3(t) + \theta_{11} u_3(t)^2 + \theta_{12} u_1(t-10)^2 + \theta_{13} u_2(t-10)u_3(t-7) \\ &\quad +\theta_{14} u_3(t-10) + \theta_{15} u_3(t)u_3(t)u_3(t-7) + \theta_{16} u_1(t)u_1(t-10) \\ &\quad +\theta_{17} u_1(t)u_2(t-10) + \theta_{18} u_3(t-7)^2 + \theta_{19} u_2(t-10) + \theta_{20} u_3(t-4) \\ &\quad +\theta_{21} u_3(t)u_3(t-4) \\ z(k) &= y(t) + \epsilon(t) \end{aligned} \tag{4.53}$$

The parameters are:

$$\begin{aligned} \theta &= [\theta_1, \theta_2, ..., \theta_{21}] \\ &= [5.69, 10.75, 1.86 \times 10^{-7}, 1.75 \times 10^{-8}, 0.0046, 0.00031, -2.37 \times 10^{-7}, 5.05 \times 10^{-8}, \\ &\quad -0.00066, 0.0031, -1.28 \times 10^{-6}, 2.27 \times 10^{-8}, 0.003, -8.69 \times 10^{-5}, -1.37 \times 10^{-6}, \\ &\quad -1.50 \times 10^{-8}, -0.0002, 3.22 \times 10^{-7}, -23.36, -0.00023, 3.18 \times 10^{-7}] \end{aligned}$$

with $cov(\epsilon) = \sigma^2 = 0.04$. The sample time is taken as 0.1sec.

Table 4.16: Preference vector of optimal input design for MIMO system

	w_1	w_2
D optimum	477.91	802.84
AI optimum	3945.2	3621.8

Since each MISO model does not include any regressor of the output of the other model, it is feasible to treat this MIMO model as two independent MISO models which can be identified separately therefore two set of optimal input can be design for each of them. However, for the purpose of saving experimental time, we propose an approach to developing a composite objective function by weighting the objectives of two MISO models. One set of optimal test signals can be developed accordingly and used as the identification signal for the two models with the expectation of exciting the behaviours of both models.

To determine the weightings, firstly the magnitudes of the values of sub-objective functions need to be scaled. For a function for which the minimum and maximum value is given, the resulting value can be normalized in $[0, 1]$. However, the minimum values of the sub-objective functions of optimal input design are not provided initially and a large amount of computation will be required in order to find the minimum value of each sub-objective function. Thus in this work a trade-off approach of scaling is proposed. Applying a white noise signal to all of the models, the corresponding absolute value of each sub-objective function, v is computed and used as one component of the weighting factors of the other sub-objectives. In an optimization which has k sub-objectives, the weighting factor of the ith sub-objective w_i is given by:

$$w_i = v_1 v_2 \dots v_{i-1} v_{i+1} \dots v_k \tag{4.54}$$

where v_i is the values of the ith sub-objective function. The catalytic converter converts harmful emission of an gasoline IC engine into less harmful substances. However, it works effectively provided that the λ of the emissions is 1 with a small tolerance of approximately 1%. The accuracy of the λ model is thus considered more important than the torque model and it is weighted relatively by 2:1 for importance in the following experiment.

Optimal input designs with D-optimal criterion and the proposed AI-optimal criterion are carried out in order to minimize the estimated parameters and output prediction of the MIMO model. The determined weights are shown in Table 4.16.

Each type of input design is carried out 10 times with different initial conditions and then utilized for model identification. The model estimated by optimal and non-optimal inputs are compared and the results of statistical validation measured by e and R^2 are shown in Table 4.17 and 4.18. It is indicated that the optimal input design with proper weightings are able to minimize the function value of each individual sub-objective and correspondingly

Table 4.17: Validation results of torque model

	$\overline{e}$	$\overline{R^2_{PRBS}}$	$\overline{R^2_{APRBS}}$	$\overline{R^2_{UDRN}}$
M_{Dop}	2.08	76.54%	64.00%	48.14%
M_{AIop}	2.76	77.80%	65.85%	50.50%
M_{PRBS}	2.45	-449.75%	-892.30%	-1416%
M_{APRBS}	2.24	72.09%	61.63%	47.54%
M_{UDRN}	4.50	57.36%	52.00%	42.08%

Table 4.18: Validation results of λ model

	$\overline{e}$	$\overline{R^2_{PRBS}}$	$\overline{R^2_{APRBS}}$	$\overline{R^2_{UDRN}}$
M_{Dop}	0.31	85.82%	75.10%	59.24%
M_{AIop}	0.34	86.64%	76.63%	61.76%
M_{PRBS}	2.40	-6210%	-11732%	-13961%
M_{APRBS}	0.54	82.75%	73.95%	59.81%
M_{UDRN}	0.59	72.68%	67.05%	55.81%

improved accuracy is obtained in all identified models. Since the required experimental time for the optimization of the composite objective function is close to the time cost of optimizing a single sub-objective function, this approach is more efficient with a large number of sub-objectives. Moreover the model obtained by D-optimum gives the smallest e but the second best R^2. As argued in [55] [101] , the D-optimal criterion is consistent with the G-optimal criterion in principle so that it should also be a sensible criterion for optimization of output prediction. However differently from most output space criteria, the D-optimal criterion does not take the selection of U_0 which is discussed in Section 4.8.1 into account so should not be considered as the best choice of output prediction based input design for black box models.

4.12 Conclusions

Technologies of optimization are implemented for optimal test signal design with the purpose of improving the quality of identified models. An iterative procedure for constrained optimal input design for black box systems is developed. Commonly used excitation signals for initial estimation of models are discussed and a white noise signal is applied in experiments on a 1.6L 4 cylinder SI PFI Zetec engine. An original MISO torque model is identified which is subsequently used as the basis of experiments on optimal input design in this chapter.

Experiments of input design are firstly implemented to a known system for the convenience of comparing the parameters and regulating the disturbance. An implementation on a black box modelling of the virtual engine is given subsequently in order to demonstrate the effectiveness in industrial applications. Various algorithms for optimization are tested for the optimal input design and the deterministic PS algorithm is recognized to be the most

appropriate particularly for reasons of the repeatability.

For the optimization of parameter estimation, A-optimal and D-optimal criteria are employed. The experimental results indicate that A-optimal criterion is effective if the regressors of model have similar scales of magnitude but may lose efficiency if significant diversity exists in scales. However the D-optimum method is not affected and provides more accurate estimation of parameter in all cases. A weighted A-optimum is proposed as an alternative approach to the D-optimum. This criterion weights the parameter variance by corresponding squared output sensitivity terms and gives an estimation with similar accuracy to the D-optimum.

As the true parameters of a black box system is generally unknown, the optimization of output prediction is more suitable for practical applications. Objective functions can be designed according to classic G-optimal and I-optimal criteria. A new criterion based on a minimization of a simplified sum of output error is proposed and illustrated to be the most effective for an improved output prediction since it gives the best computing efficiency.

The statistical validation shows the advantages of optimal inputs in identifying an accurate model for a known system and a unknown virtual engine. In applications of MIMO model identification, methodologies of input design can be applied to generate a set of optimal inputs by minimising a comprehensive objective function which is composed of the weighted values of sub-objective functions. The optimal inputs are effective to improve the accuracy of all sub-models with less computational burden.

The proposed methodology of optimal input design is used in the later chapter of dynamic model-based calibration and control. The optimal inputs are designed to further improve the accuracy of polynomial engine models.

Chapter 5

Selection of Parameter Estimation Methods

5.1 Introduction

The quality of system identification is known to be affected by two main factors: model structure and parameter values. Techniques of DoE such as input design have been developed with the purpose of reducing the error of parameter estimation before selecting the estimator. However, the estimation method does have a significant influence on the estimation results, which thus should be selected sensibly according to the prior knowledge of the system.

In this chapter, the model types which should be determined by eventual application of the model are introduced and estimation methods for different types of models are discussed and subsequently evaluated by examples. A simulation error method is developed from a traditional prediction error method. The proposed estimation method for simulation models is initially demonstrated with an identification of a known system and then applied to identify a black box model of the virtual engine.

5.2 Model Type Selection

Although the most usual application of a model is to forecast the future system output behaviour, there are two types of models that need to be distinguished. As stated in equation (2.1), a prediction model utilizes the input and output of the system to predict the output in one step or k steps ahead while a simulation model in equation (2.2) uses the input of the system and the simulated output of the model to generate the simulated output. In cases where no regressor of the output is included in the model, the prediction model has no difference to the simulation model, e.g. as in finite impulse response model. However, for a

general linear model structure:

$$y(t) = G(q)u(t) + H(q)\epsilon(t) \tag{5.1}$$

the difference between prediction model and simulation model is as illustrated in Figure 5.1. As illustrated in order to run a simulation model, only the input signal is required while the previous output from the system is also needed to run a prediction model.

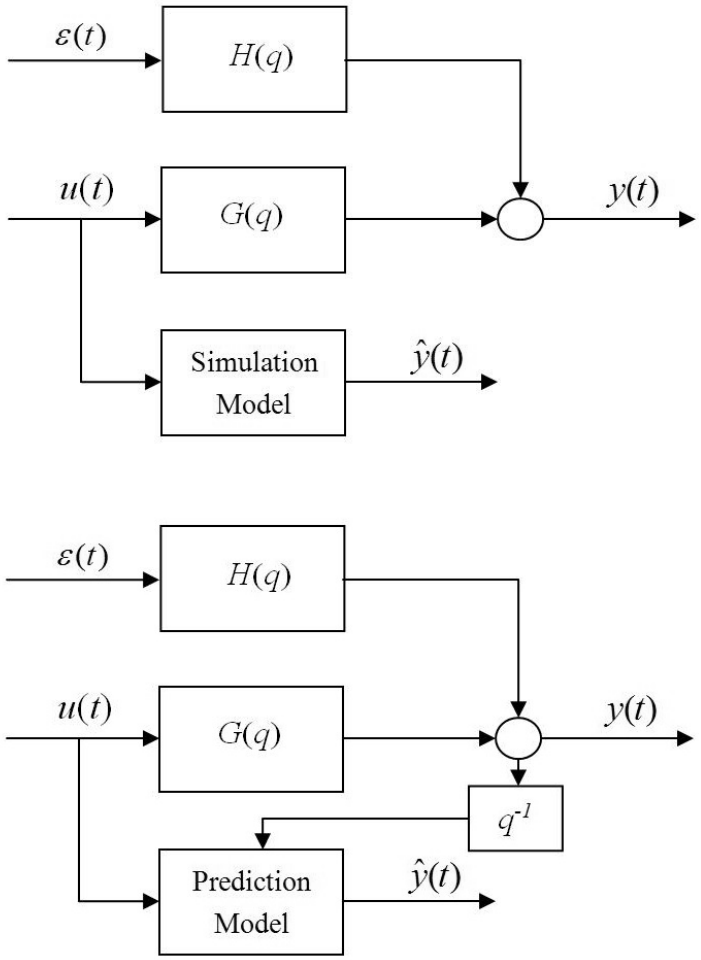


Figure 5.1: Schematic of simulation model and prediction model

Once a prediction model is identified, the output prediction is determined by the input and previous system output. Therefore at the sample instant k, the prediction error $e(k)$ does not affect the prediction results in other sample instants. In other words the predicted output $\hat{y}(t)_{prd}$ has no influence on the predicted output at other sample instants. Although the output of the identified prediction model cannot perfectly match the output of the real system, the inputs of the prediction model which are inputs and delayed outputs of the real system, provide information of the system behaviour online and thus the predicted output can be adjusted to avoid deviating from the measured output at each sample instant. The

prediction model may not have a disturbance term nevertheless the predicted output is still required to have a stochastic part since the measured output is disturbed by the noise of the system. In the process of parameter estimation, the noise in the identification data has an effect on the estimated parameter. When the prediction model is working on the system, even if the system output experiences unexpected disturbance caused by a noise which is very different from the noise in identification, the prediction model can still forecast a relatively accurate output since the information in the new noise is delivered to the prediction model by the delayed system output.

A simulation model can be run fully deterministically once it is identified. The disturbed system output affects the estimation results of the model but is not presented in simulation. Hence a simulation model may not be able to accurately forecast a stochastic system when the system output is disturbed by a different noise from the one that was presented during the identification. The simulated output at the sample instant k, $\hat{y}(k)_{sim}$ is influenced by the input and previous simulated outputs $\hat{y}(1)_{sim}$, ..., $\hat{y}(k-1)_{sim}$. Consequently for a dynamic simulation model, if it cannot perfectly represent the real system, the error between the simulated and measured output would exist from the start of the simulation and would be accumulated in time series. Therefore the simulated output at later sample instant may deviate from the system output significantly.

Compared to the simulation model, the prediction model can generally give a model output which is more accurate if an appropriate estimation method is selected. However, the simulation model has a significant utility because it works independently of the real system. In many practical applications, a simulation model is required to be a substitution of the real system and further design is then developed based on the model and finally implemented on the real system. For instance, a controller for an engine is often initially designed using an accurate offline simulation model.

5.3 Estimation Method for Prediction Model

For a prediction model, the prediction error is the essential measure of the model quality and it can be given by:

$$e(t) = y(t) - \hat{y}(t) \tag{5.2}$$

A well estimated model should thus seek to minimize the prediction error over the identification data. The objective function for prediction error minimization can be a scalar function of the error vector. Since there is considerable flexibility in choosing the objective function, many prediction error methods (PEM) have been developed e.g. least square methods.

As shown in equation (2.14), the OLS method minimizes a quadratic scalar function of prediction error. Assuming no limit on the values of parameters θ and that all parameters are independent of each other, minimizing the prediction error becomes a convex quadratic optimization problem and a unique global solution can be found. An analytical solution of $\hat{\theta}$ is given in equation (2.15), where the matrix $X^T X$ is non singular if the system is precisely excited. The numerical solution approaches the analytical solution with increasing numbers of iterations and the error can be limited to an acceptable range if sufficient iterations carried out.

Example

Consider an affine MISO simulation torque model which is identified from real engine experiment data as a known system:

$$\begin{aligned} y(t) &= \theta_1 + \theta_2 u_1(t-5) + \theta_3 u_1(t-6) + \theta_4 u_1(t-7) + \theta_5 u_2(t-1) \\ &\quad + \theta_6 u_3(t-5) + \theta_7 u_3(t-6) + \theta_8 u_3(t-7) + \theta_9 y(t-1) \\ &\quad + \theta_{10} y(t-2) + \theta_{11} y(t-3) + \theta_{12} y(t-4) \\ z(t) &= y(t) + \epsilon(t) \end{aligned} \tag{5.3}$$

where u_1 denotes ABV, u_2 denotes SA and u_3 denotes engine speed. ϵ is a term of disturbance with zero mean and covariance 0.5. The sample time of this discrete model is 0.1 sec. In this model ϵ is not a real engine input signal but a term which represents the disturbance of the system. In this chapter it is assumed that this disturbance is normally distributed for the ease of using ordinary least square method. The parameter values are:

$$\begin{aligned} \theta = [&-5.11,\ 14.27,\ -50.82,\ 35.74,\ 0.028,\ -0.025, \\ &0.043,\ -0.015,\ 0.20,\ 0.30,\ 0.040,\ 0.25] \end{aligned} \tag{5.4}$$

The inputs are constrained within:

$$\begin{aligned} 42\% &< u_1 < 50\% \\ 16^\circ &< u_2 < 34^\circ \\ 1000RPM &< u_3 < 1800RPM \end{aligned} \tag{5.5}$$

To identify a prediction model corresponding to this known system in equation (5.3), UDRN signals are used and the estimated parameter $\hat{\theta}_{prd}$ is obtained by the OLS method in equation (2.15) as:

$$\begin{aligned} \hat{\theta}_{prd} = [&-1.65,\ 15.8,\ -53.77,\ 34.14,\ 0.021,\ -0.0041, \\ &0.017,\ -0.011,\ 0.24,\ 0.24,\ 0.17,\ 0.21] \end{aligned} \tag{5.6}$$

The predicted output and measured output are depicted in Figure 5.2. It is found that the R^2 of the prediction model is 92.62% and the MSE is 0.62. In this figure the torque values between 750-950 points are negative. This is due to the net engine pumping losses being greater than the power generated from combustion. In other words when the losses from the compression, exhaust and intake strokes are greater than the power generated from the combustion stroke, a negative torque will be obtained.

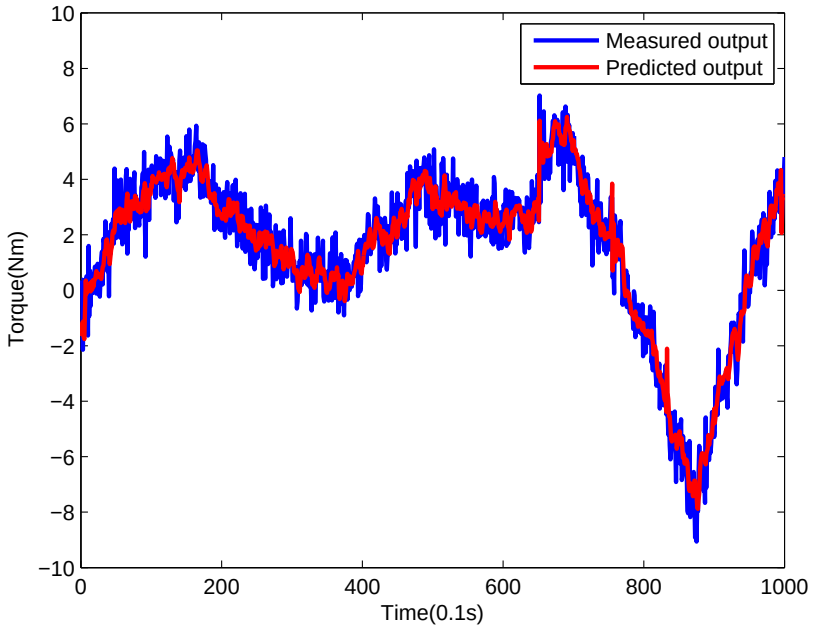


Figure 5.2: Measured output and predicted output

5.4 Estimation Method for Simulation Model

5.4.1 Adapted Prediction Error Method

To estimate the parameters of a simulation model, a simple approach is to build a prediction model with the same model structure then estimate the parameter by the PEM and directly use it for the simulation model. However the PEM is developed explicitly for prediction applications in which accumulated error of simulated output does not exist. Consequently the simulation model obtained by the PEM is estimated at the expense of accuracy.

Using the estimated θ in equation (5.6), a simulation model can be derived and Figure 5.3 illustrates the simulation output and measured output. The R^2 of the simulation model is 88.46% and the MSE is 0.97.

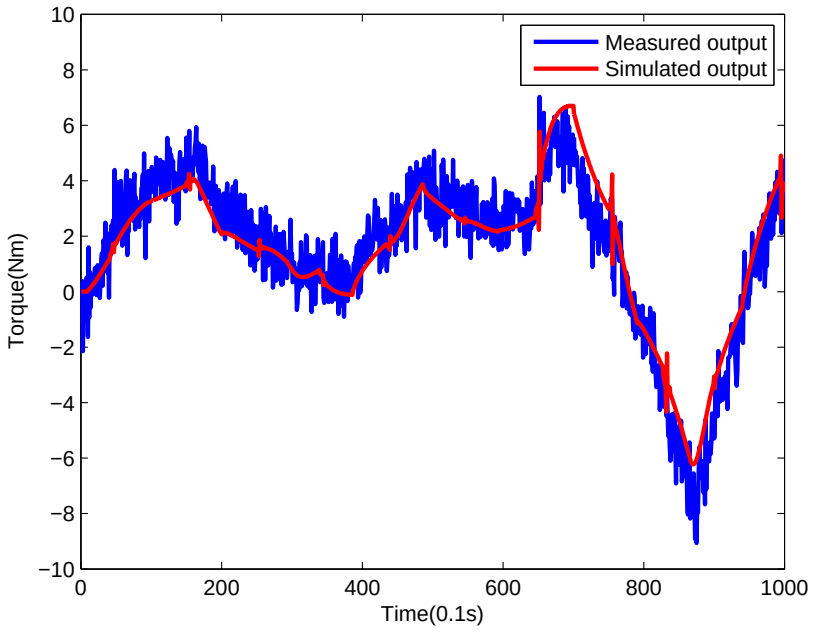


Figure 5.3: Measured output and simulated output by PEM

5.4.2 Simulation Error Method

In this thesis, a simulation error method (SEM) is proposed for the parameter estimation of a simulation model. The SEM is developed based on a modification of PEM in which the matrix of regressors in the output error minimization is amended. Taking the general model in equation (2.11) as an example, for a prediction model the objective function of the optimization is:

$$\min(Y - \hat{Y})^2 = \min(Y - X\theta)^2 \tag{5.7}$$

where

$$Y = \begin{bmatrix} y(p+1) \\ y(p+2) \\ \vdots \\ y(N) \end{bmatrix} \qquad \hat{Y} = \begin{bmatrix} \hat{y}(p+1) \\ \hat{y}(p+2) \\ \vdots \\ \hat{y}(N) \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{m+n} \end{bmatrix} \tag{5.8}$$

$$X = \begin{bmatrix} y(p) & \cdots & y(p-m+1) & u(p) & \cdots & u(p-n+1) \\ y(p+1) & \cdots & y(p-m+2) & u(p+1) & \cdots & u(p-n+2) \\ \vdots & & \vdots & \vdots & & \vdots \\ y(N-1) & \cdots & y(N-m) & u(N-1) & \cdots & u(N-n) \end{bmatrix} \tag{5.9}$$

The objective function can be minimized numerically as a standard convex problem or analytically by OLS in equation (2.15). In this objective function, a sum of squared output error at each sample instant is minimized and every individual error is affected by input and output collected from the system at relevant sample instants.

For a simulation model, the optimization problem is as in equation (5.7) but the regressor matrix X is constructed by:

$$X = \begin{bmatrix} \hat{y}(p) & \cdots & \hat{y}(p-m+1) & u(p) & \cdots & u(p-n+1) \\ \hat{y}(p+1) & \cdots & \hat{y}(p-m+2) & u(p+1) & \cdots & u(p-n+2) \\ \vdots & & \vdots & \vdots & & \vdots \\ \hat{y}(N-1) & \cdots & \hat{y}(N-m) & u(N-1) & \cdots & u(N-n) \end{bmatrix} \tag{5.10}$$

where the vector of simulated output is computed by the input and previous simulation output sequentially. Since the simulation output $\hat{y}$ in X varies during the process of iteratively estimating the parameter by minimizing the error of equation (5.7), the analytical solution which requires every entry of X to be pre-determined cannot be employed for estimation. However this quadratic optimization problem can be conveniently solved by a numerical solution using an appropriate algorithm. The parameter estimated by the PEM is still useful for the identification of simulation model because it can be used as the initial values of the optimization for a reduced experimental time. Using the PS method for optimization, Figure

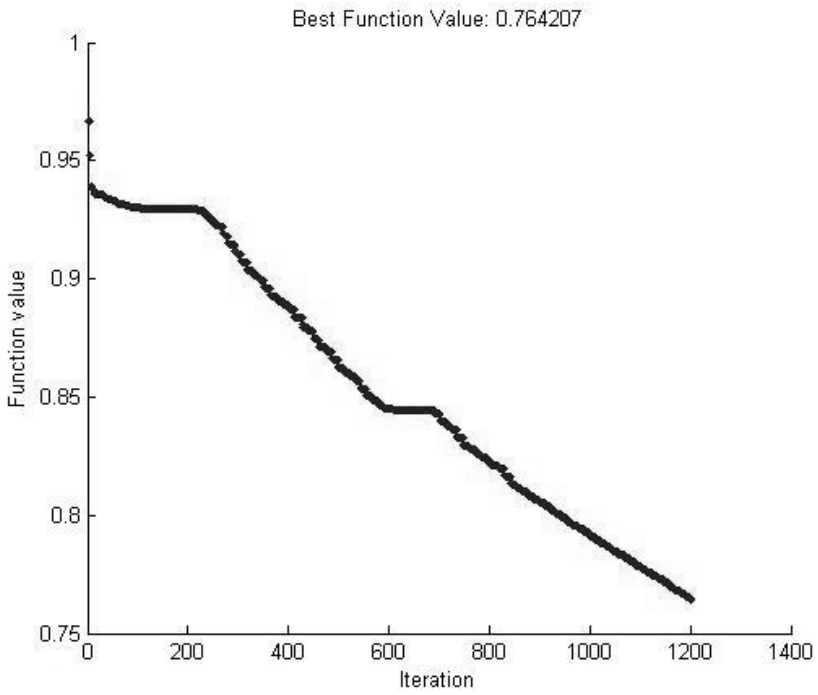


Figure 5.4: Minimized objective function value by Pattern Search method

5.4 shows the further reduced objective function value acquired by the SEM method from 0.97 to 0.76.

As stated above, the optimization of parameter estimation is an unconstrained convex problem which can be solved by local algorithms efficiently. Keeping the stopping criteria the same, two of the unconstrained nonlinear optimization algorithms, line search and Nelder-

Mead (NM) simplex algorithm are tested for comparison. As a basic unconstrained algorithm, the line search firstly attempts to find the descent direction of the objective function and then determines the step size along the direction. In each iteration, a maximum searching interval on the line, called the bracket is determined and subsequently divided into subintervals. The value of the objective function or polynomial interpolation function which is used for approximation is computed at subintervals and the minimum is selected.

The NM simplex algorithm is a direct search method which is independent of the derivative of the objective function. This algorithm constructs a simplex of n-dimensional vector with $n+1$ points. Values of the objective function corresponding to $n+1$ points are computed and arranged in order. The point reflecting the biggest value will be replaced by a new point. Initially the new point can be selected as the centroid of the remaining n points. If the value reflected by the new point is worse than the current worst point, another point will be selected and the procedure is repeated until a better point is found. The simplex is thus modified iteratively and a minimum can be approached.

Table 5.1: Optimized objective function with different algorithms

	Pattern Search	Linear Search	NM simplex
MSE	0.76	0.93	0.51
Time	$46s$	$2s$	$21s$

Table 5.1 shows the optimization result and experimental time of three algorithms. The NM simplex algorithm generates the smallest MSE in a short time. Although the linear search completed the optimization in two seconds, the MSE proves that it is not a suitable choice because of the premature ending of optimization with the same stopping criteria.

The estimated parameter vector acquired by the SEM method with the NM simplex algorithm is:

$$\begin{aligned}\hat{\theta}_{sim} &= [-5.16,\ 21.04,\ -54.10,\ 32.06,\ 0.031,\ -0.0044, \\ &\quad\ 0.019,\ -0.011,\ -0.13,\ 0.28,\ 0.29,\ 0.36]\end{aligned} \tag{5.11}$$

As shown in Figure 5.5, the simulation model obtained by the SEM reproduces the output better with an R^2 of 93.72%.

In order to validation the simulation models, 10 other sets of inputs with constraints in equation (5.5) are applied to the original model in equation (5.4) and simulation models with parameters estimated by $\hat{\theta}_{PEM}$ and $\hat{\theta}_{SEM}$. The MSE and R^2 are shown in Table 5.2. The model with $\hat{\theta}_{SEM}$ is shown to have better accuracy in simulation and thus the SEM is demonstrated to be effective.

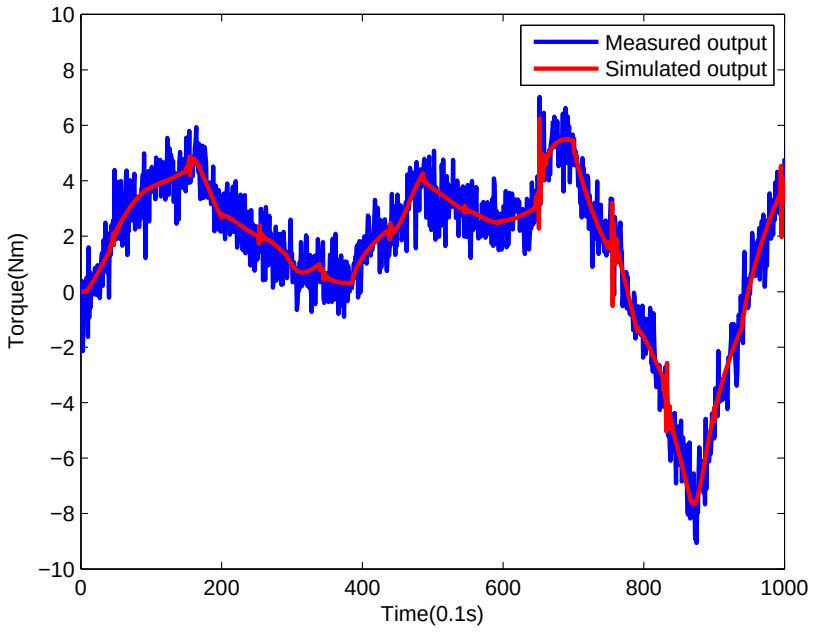


Figure 5.5: Measured output and simulated output by SEM

Table 5.2: Validation results

	$\overline{\text{MSE}}$	$\overline{R^2}$
$M_{\theta_{PEM}}$	1.14	90.83%
$M_{\theta_{SEM}}$	0.53	95.67%

5.5 Parameter Estimation of the Virtual Engine Model

In this section the PEM and SEM are employed to identify a simulation model of a real engine system, rather than of a known system where the model structure and true parameter values are available. The purpose is to demonstrate the effectiveness and compatibility of SEM on parameter estimation of a black box model and also exhibit the SEM in an industrial application.

The virtual engine (RT model) is considered as the real system. The objective of the experiment is to develop a torque model and a λ model which will be used to design controllers by offline approaches and therefore simulation models are favoured. A 3×2 MIMO model is identified by time series data collected from the virtual engine and validated by sets of engine data as well. Assuming each model output is not affected by the other output, the MIMO model can be divided into two 3×1 MISO models. The inputs are selected to be injection fuel mass (u_1), spark advance (u_2) and throttle angle (u_3) and the type of inputs is uniformly distributed random number (UDRN). As discussed in Chapter 4, optimal inputs should be

selected for system identification in order to maximize the data information and improve the model accuracy. However in the following sections we only discuss the benefit of the proposed SEM estimation method in parameter estimation therefore a set of common UDRN signals is employed. The amplitudes of inputs are constrained as follows:

$$\begin{aligned} 0mg &< u_1 < 35mg \\ 5^\circ &< u_2 < 30^\circ \\ 5^\circ &< u_3 < 20^\circ \end{aligned} \tag{5.12}$$

As stated in Section 2.3, the sample time should be selected according to the rise time of the output. In this experiment the sample time is 0.03 sec for both MISO models. Since the sample time is small, the data length should be long enough to capture the dynamics of the system. In this chapter the objective is to develop a better estimation method for simulation models so that the methodology on the selection of data length is not discussed here. The data length of input is selected as 2000 which represent the data recorded in 60 sec. The structure of the torque model is taken as follows:

$$\begin{aligned} y_1(t) = {} & \theta_1(1) + \theta_1(2)u_1(t-1) + \theta_1(3)u_1(t-2) + \theta_1(4)u_1(t-3) \\ & + \theta_1(5)u_2(t-1) + \theta_1(6)u_3(t-1) + \theta_1(7)u_3(t-2) + \theta_1(8)y_1(t-1) \end{aligned} \tag{5.13}$$

Using the same identification signal, parameters estimated by PEM and SEM are listed in Table 5.3.

Table 5.3: Estimated parameters of torque model by PEM and SEM

	$\theta_1(1)$	$\theta_1(2)$	$\theta_1(3)$	$\theta_1(4)$	$\theta_1(5)$	$\theta_1(6)$	$\theta_1(7)$	$\theta_1(8)$
PEM	-4.5	4.56	-5.19	0.94	0.067	1.14	-0.73	0.87
SEM	-2.57	-6.46	14.76	-8.2	0.046	-0.043	0.36	0.92

Two simulation models are established by using θ_{PEM} and θ_{SEM} respectively. The identification signal is applied to these two models and simulated outputs are recorded and compared with the system output in order to evaluate the model accuracy. The output fitness of models is shown in Table 5.4, where the SEM provides a considerable improvement in both MSE and R^2.

Table 5.4: Validation results of torque model

	MSE	R^2
M_{PEM}	113.97	77.78%
M_{SEM}	76.08	83.72%

The identification of the λ model follows the same procedure as above. The model

structure is given by:

$$\begin{aligned} y_2(t) &= \theta_2(1) + \theta_2(2)u_1(t-1) + \theta_2(3)u_1(t-2) + \theta_2(4)u_2(t-1) \\ &\quad +\theta_2(5)u_3(t-3) + \theta_2(6)y_2(t-1) + \theta_2(7)u_2(t-1)y_2(t-1) + \theta_2(8)u_1(t-1)u_3(t-1) \end{aligned} \tag{5.14}$$

Estimated parameters and validation results are shown in Table 5.5 and Table 5.6.

Table 5.5: Estimated parameters of λ model by PEM and SEM

	$\theta_1(1)$	$\theta_1(2)$	$\theta_1(3)$	$\theta_1(4)$	$\theta_1(5)$	$\theta_1(6)$	$\theta_1(7)$	$\theta_1(8)$
PEM	0.41	0.6	-0.6	0.0017	0.0046	0.58	-0.0001	-7.29×10^{-05}
SEM	0.49	0.77	-0.77	0.00088	0.0058	0.46	-0.00056	-0.0002

Table 5.6: Validation results of λ model

	MSE	R^2
M_{PEM}	0.0174	97.68%
M_{SEM}	0.0144	97.7%

Compared to the model of torque, the values of R^2 of the λ model by PEM and SEM are very high and similar. However, the values of MSE still indicate the superiority of the SEM. Generally for a model of good quality, the improvement in estimation accuracy derived by the SEM might be limited. Moreover the objective function of the numerical optimization in the SEM can be selected flexibly, not necessarily to be the squared error between simulated output and system output, according to a specific requirement of the model quality. The estimated model thus has a superior performance in that aspect than using the PEM. This numerical minimization is also favoured since an analytical solution of the objective function is not always available.

The MIMO simulation model is then validated by 10 other sets of signals collected from the virtual engine and the result of averaged MSE and R^2 are shown in Table 5.7

Table 5.7: Validation results of MIMO model

	$\overline{\text{MSE}}$	$\overline{R^2}$
$M_{PEM}(Torque)$	154.97	73.68%
$M_{SEM}(Torque)$	115.90	79.62%
$M_{PEM}(\lambda)$	0.0155	97.15%
$M_{SEM}(\lambda)$	0.0131	97.39%

Based on the results of the tests discussed in this chapter, the accuracy of models estimated the SEM is always better than the PEM method therefore the benefit of the SEM method in parameter estimation for simulation models is proved. For the use in industrial applications, the model accuracy should be further improved by other DoE methodologies such as optimal input design and model structure selection.

5.6 Conclusions

Features of prediction models and simulation models and their practical applications are discussed. Appropriate parameter estimation methods for each type of model are introduced accordingly. An example of LS estimation of the prediction model is demonstrated and also used for identification of the simulation model. The proposed SEM minimizes a quadratic scalar function of output error, which is similar to PEM, nevertheless the estimated output is purely determined by the input and simulated output. The SEM is found to give more accurate parameter estimation than traditional PEM if the intended use of the estimated model is for simulation while the PEM has the drawback of neglecting the possible error accumulation.

The SEM is firstly implemented to an identification of a known torque model which is derived from experimental data from the real engine. In the process of identification and statistic validation, the superior performance of this method is fully displayed by both measurement criteria. Another application of a black box modelling, the virtual engine identification is given subsequently in which the SEM leads to a remarkably improved identification and validation result of the MIMO engine model. It indicates that the SEM can be utilized for the estimation of simulation models in practical applications rather than in purely ideal situations.

In a general practice where the selected simulation model structure has both input and output regressors, it is recommended to start with the LS method for the initial values followed by a SEM estimation.

Chapter 6

Static Calibration and Controller Design

6.1 Introduction

In recent years, the design of control system for modern IC engines is one of the most important steps in the process of engine development. To satisfy the legislative demands of environmental protection and the requirements of manufactures and customers, the major purposes of engine control is to lower the emissions and minimize the fuel consumption with a satisfying engine performance. Because of the nonlinearity of engines and complexity of operating conditions, static look-up table based feedforward controllers are still widely used to realize the control objectives. The whole operating region is represented by a grid of operating points and static calibrations are carried out at each operating point so as to obtain the steady-state settings of related engine calibration parameters. Static maps are thereby formed by the optimal settings of calibration parameters obtained in experimental steady-state testing and utilized to control the engine by the engine management system [3, 2, 102, 103].

The following chapter describes a basic static calibration on the virtual engine for constrained fuel optimization. Firstly the procedure and targets of the calibration are explained and corresponding settings of the RT model are given. The selection of a reasonable operating region according to the simplified virtual engine is discussed. The process of finding optimal settings at an operating point are illustrated and a static map is obtained by testing over the operating region. The effectiveness of the map is then validated on the virtual engine. Because of its known efficiency, the static map is used as the basis of comparison to the dynamic map developed in the next chapter in order to assess the effectiveness of dynamic model based calibration.

6.2 Procedure of Static Calibration

The principle of static calibration is based on the investigation of the steady-state behaviour of the experimental engine over a broad operating region. The optimal settings of calibration parameters that satisfy the control objectives are recorded and then a fixed map is developed for the production engine to choose appropriate settings according to different driver's demand and working conditions. A general procedure of static calibration has been illustrated in Figure 1.1. The first step is to choose representative points in the entire engine operating space and the selection of the grid is then a trade-off between the amount of calibration work and the control performance of subsequent maps. Then local tests are conducted on the engine and steady-state data are recorded in order to develop local models. Local optimal settings are obtained by calibrations at the local models and calibration maps of the entire operating region are derived accordingly.

In any model-based calibration, the accuracy of models is crucial to the effectiveness of the resulting calibration map and many DoE methods may need to be implemented in order to develop static local models with high accuracy. Moreover, the objective of this chapter is to find effective optimal settings of the virtual engine, collect the corresponding optimal engine response and use it as a basis for the comparison with the control performance of the dynamic controller obtained in the next chapter. Therefore a simple static calibration is directly conducted on the virtual engine although the experimental time will be increased due to the hardware-based tests. Standard model-based static calibration is not discussed in this thesis.

6.3 Objectives of Calibration

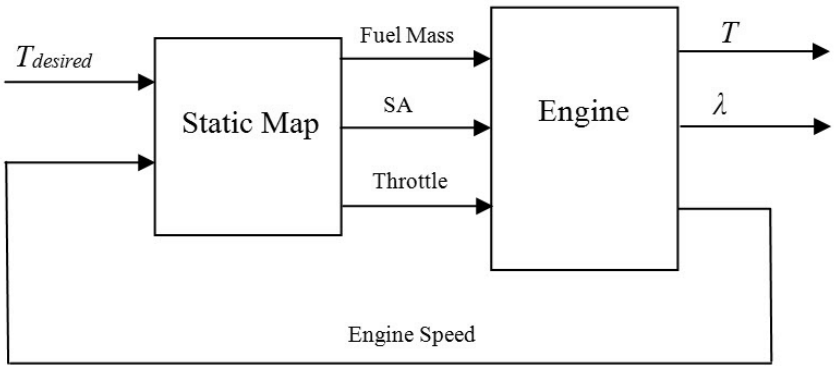


Figure 6.1: Schematic of a calibrated control system

In order to optimize the fuel economy and meet the requirements of engine performance

and emissions, the aims of the proposed simplified gasoline engine calibration are:

1. Track demands for torque
2. Regulate stoichiometric air-fuel ratio
3. Achieve 1. and 2. for the minimum possible fuel mass consumption

To simplify the calibration, requirements of driveability, knock and constraints on emissions and temperatures are neglected in this work. As the desired torque is fulfilled by fuel supply in diesel engines and by air supply in gasoline engines [3], a precise control of throttle position is thus used for the first objective. The AFR presents the ratio of air and fuel in the emission gas therefore it is feasible to incorporate the control of injected fuel mass with the air supply to meet the requirement of stoichiometric λ. For the remaining controllable calibration parameters of this simplified calibration, the SA is the major influential factor in combustion which in turn determines the efficiency of converting the energy in the fuel to engine torque. Correspondingly the resulting calibrated control system receives the desired torque and engine speed as inputs, and controls the injected fuel mass, spark advance and throttle position to achieve minimum fuel consumption subject to the tracking and regulation requirements, as demonstrated in Figure 6.1.

6.4 Design of Experiments

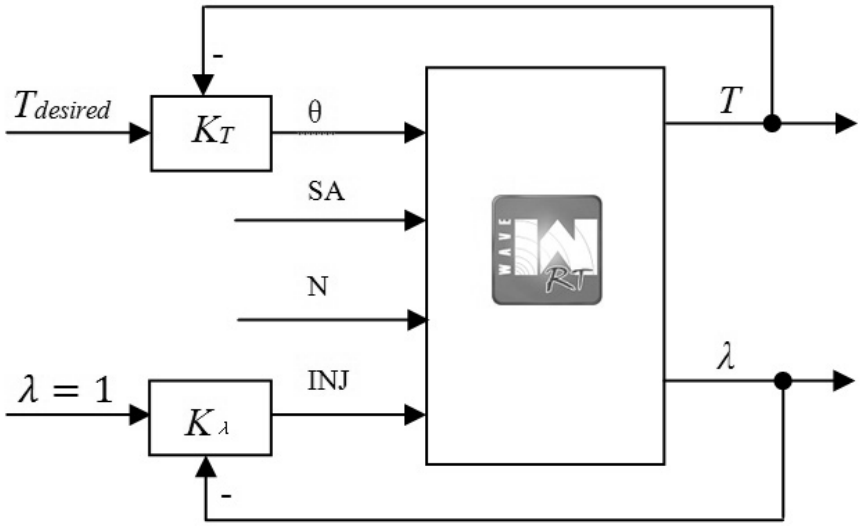


Figure 6.2: Simplified configuration of the WaveRT model for initial development

As discussed in Chapter 3, the experiments are conducted on an RT model, which is a virtual simulation of Ford 2.0L GTDI engine. Various calibrations should be carried out

in order to find out the static map for multi-variable control. With reference to Figure 6.2, firstly two PI feedback control loops are tuned to control the fuel mass INJ to maintain stoichiometric air-fuel ratio and the throttle θ to maintain the desired torque load. The desired torque and stoichiometric λ hence should be the references of the controllers. It is important to know that the selection of PI controller has a crucial impact on the experimental time of the static calibration because before recording the data, it is essential to wait until the output settles down at each operating point. However it would be time costly to find optimal controllers at each testing point so that a compromise has to be made between the time spent on the settling of output and controller design. In this work the PI controllers for desired torque and λ are tuned online and given by:

$$\begin{aligned} K_T &= \frac{0.15 + 0.007s}{s} \\ K_\lambda &= \frac{-20 - 1.5s}{s} \end{aligned} \tag{6.1}$$

In this thesis the operating region is a two dimensional space of torque and engine speed. To adjust the working condition from point to point in the region, the demand of torque is realized by the feedback PI controller while the engine speed is regulated by the applied external load. In real engine tests, the load is often controlled by the coupled dynamometer. Users can adjust the load applied by the dynamometer to achieve the desired speed. In the RT model, an engine speed actuator can be implemented which is able to control the speed directly and instantaneously. At each operating point, the SA is swept over the safe range to find the optimal setting for best fuel consumption.

The four inputs of the RT model under consideration are engine speed N, throttle angle θ, spark advance SA, and the injected fuel mass INJ. The remaining calibration parameters for the model are kept fixed, as follows:

- Fixed ambient conditions
- Inlet valve timing in locked position (MOP = 231° BTDC firing)
- Exhaust valve timing (MOP = 256° ATDC firing)
- Wiebe exponent (2)
- Wastegate actuator (28 mm diameter)

6.5 Selection of Operating Space

To reduce the dimensionality of the model for a first demonstration of the techniques, the effect of the turbocharger was reduced as much as possible by setting the orifice diameter,

representing the waste gate, to 28mm diameter. Both camshafts were set at zero degrees advance, which is consistent with their lock positions (-231 and +256 relative to firing-TDC for the inlet and exhaust cams respectively). In order to mitigate the effect of these constraints, only the low-load low-speed region is considered in this chapter. The typical engine behaviour in the reduced region is reasonably consistent with the open wastegate and locked cam positions.

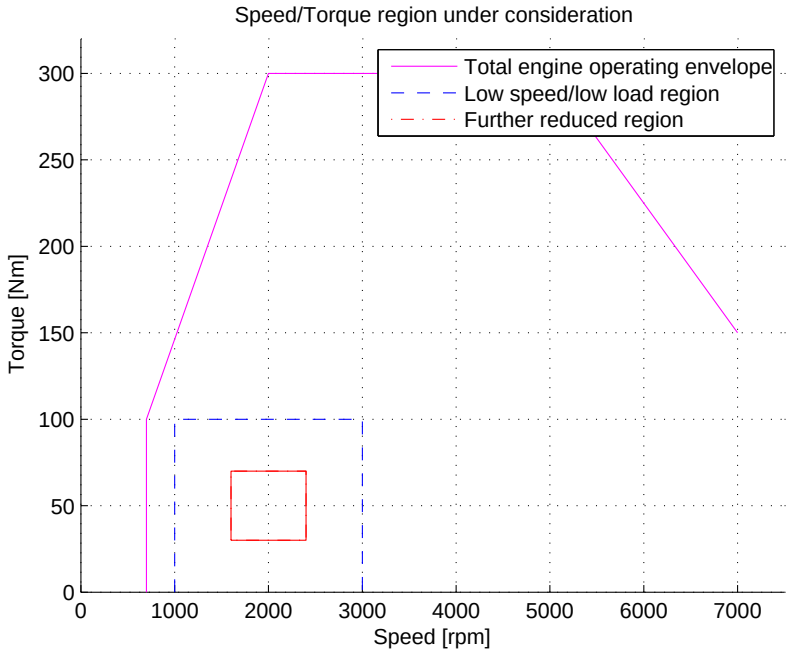


Figure 6.3: Model operating envelope and reduced calibration region

Figure 6.3 indicates the approximate speed/load range for the entire engine operating envelope. A full series of spark sweeps at equally spaced fixed speeds and desired torques of the low-load low-speed region are given by:

$$\begin{aligned} SA &= \{5, 6, ..., 29, 40\} \\ N &= \{1000, 1200, ..., 2800, 3000\} \\ T &= \{0, 10, ..., 90, 100\} \end{aligned} \tag{6.2}$$

The static calibration methodology is firstly carried out at further reduced subset and then expanded to the whole low-load low-speed region.

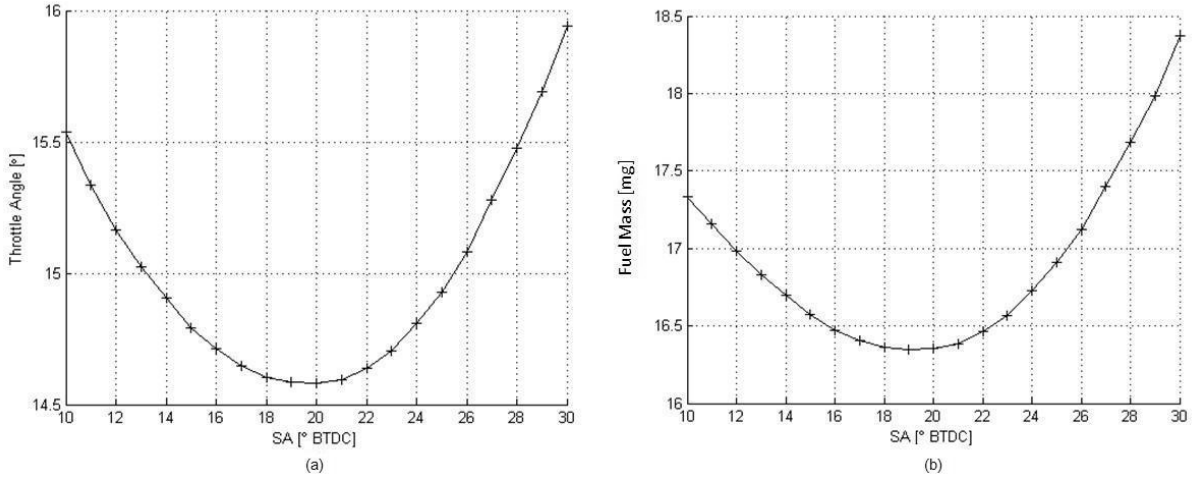


Figure 6.4: Throttle (a) and fuel mass (b) required to maintain torque demand

6.6 Calibration Results

6.6.1 Optimal Setting at Operating Points

For every speed-torque operating point under consideration, the SA is stepped within the allowable range and the throttle and fuel control inputs allowed to settle for 9 sec to achieve the desired static values. After that the data is sampled and averaged over the next one second. An example of the processed WAVE-RT data for a static spark sweep at fixed torque load and speed (70 Nm, 2000 RPM) produced from the current WAVE-RT 2.0L GTDI model is shown in Figure 6.4

The examination of Figure 6.4 reveals that the optimal SA for minimum fuel consumption at this particular speed torque point is 19 deg BTDC. The profiles of SA to θ and to INJ are similar to quadratic curves, which indicates the investigation of SA to optimal fuel economy can alternatively be determined by a numerical convex optimization. Additionally only a 1% increase in fuel consumption spark timing is obtained at 3 degrees away from the optimal value. Assuming that the local test is conducted on an engine model, the requirement of the model quality can be relaxed since a sub-optimal SA within a reasonable tolerance is able to provide a fuel consumption very close to the optimal solution.

6.6.2 Calibration Maps

For the further reduced number of operating points (N = 1600, 1800, 2000, 2200, 2400, T = 30, 40, 50, 60, 70) the minimum fuel consumption at each operating point was obtained

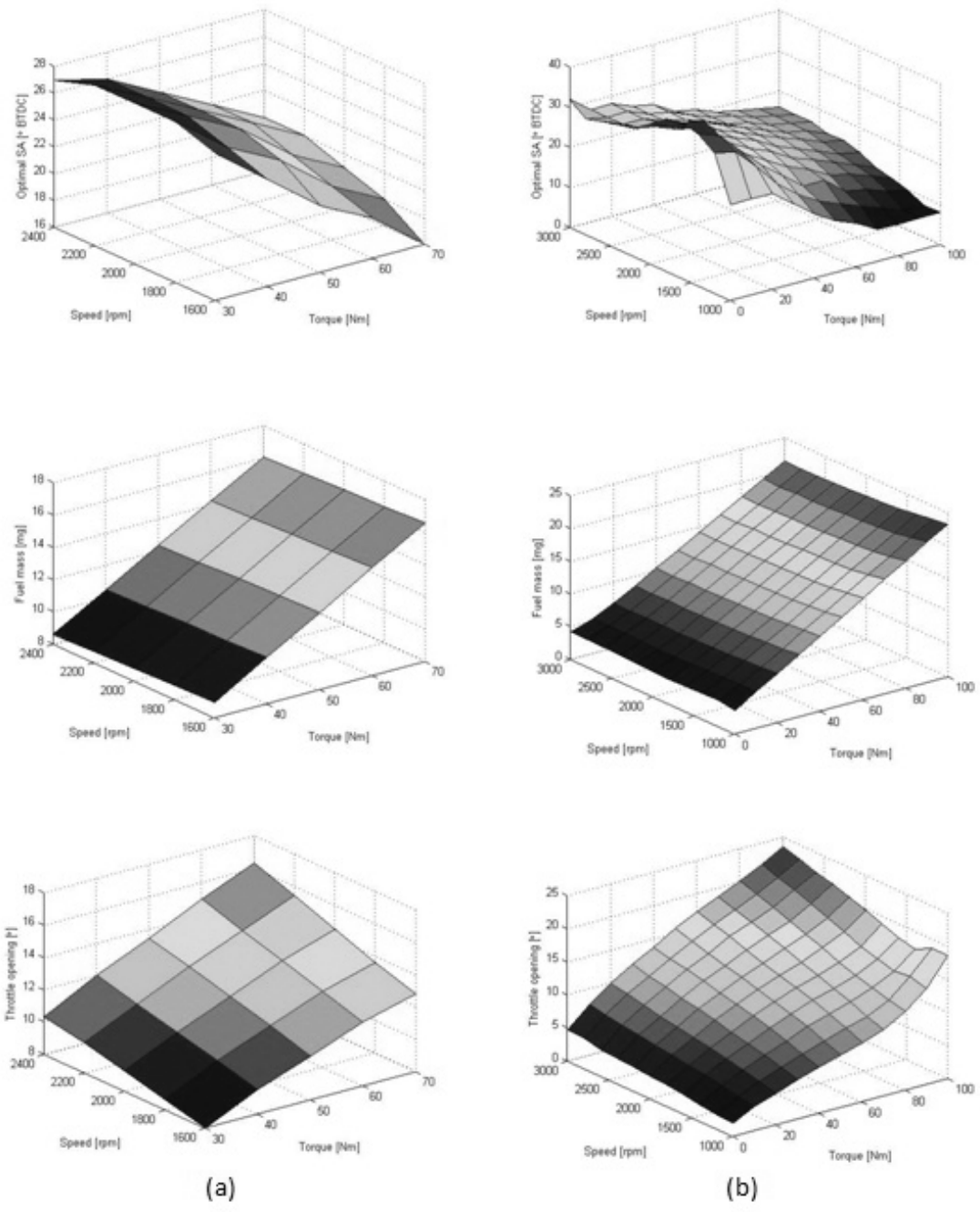


Figure 6.5: Calibration maps of the reduced region (a) and low-speed low-load region (b)

by sweeping the SA and regulating the throttle to maintain the desired brake torque. From these 25 sweeps the optimal SA and corresponding throttle angle and injected fuel mass were obtained. Figure 6.5 shows the resulting optimal maps of the reduced region (a) and also the whole low-speed low-load region (b) which is composed of 121 operating points. It can be seen that the profile of the optimal fuel mass is remarkably flat and devoid of any static nonlinearity. Profiles of optimal SA and throttle angle are relatively linear in the further reduced region while more nonlinearity is exhibited with the expanding of engine speed and torque region as shown in the low-speed low-load region. In general, the values of optimal INJ linearly decrease with the increase of engine speed and the decrease of torque. The increase or decrease of optimal θ is consistent with the engine speed and torque however in the relative high torque region the trend of optimal θ with respect to speed becomes nonlinear. Compared to the other two maps, the map of optimal SA exhibits more nonlinearity. The optimal value increases with the increase of speed and the decrease of torque and a significant nonlinearity is discovered in the low-speed low-load corner of the map.

6.7 Online Validation of Static Map

The obtained static map was connected to the RT model to examine its ability of tracking the torque and λ and minimizing the fuel consumption online. Random number signals with a time interval of 6 sec are applied as the desired torque and engine speed with amplitude constraints which are given by:

$$\begin{aligned} 10Nm < \quad T_{desired} \quad &< 90Nm \\ 1000RPM < \quad N \quad &< 3000RPM \end{aligned} \tag{6.3}$$

A set of engine speed and torque profiles and corresponding optimal INJ, SA and θ are shown in Figure 6.6. At the time instant that the operating point switches, the optimal inputs change immediately according to the static map which works as a feedforward controller. The profile of optimal fuel mass is quite similar to the desired torque, which indicates that providing the AFR remains a constant and the SA is always at its optimal value, the generated engine torque is proportional to the fuel consumption and the proportional ratio would be almost the same at all operating points.

Figure 6.7 illustrates the outputs of torque and λ from the RT model and Table 6.1 shows the characteristic result of controllers. The output responds to the change of operating points quickly with no overshoot and a maximum steady-state offset of less than 1%. λ experiences a spike at the transient switching operating point. The size of the spike and corresponding settling time is closely related to the step size of the torque. Since the engine torque is proportional to the injected fuel mass, in this control system the INJ changes accordingly

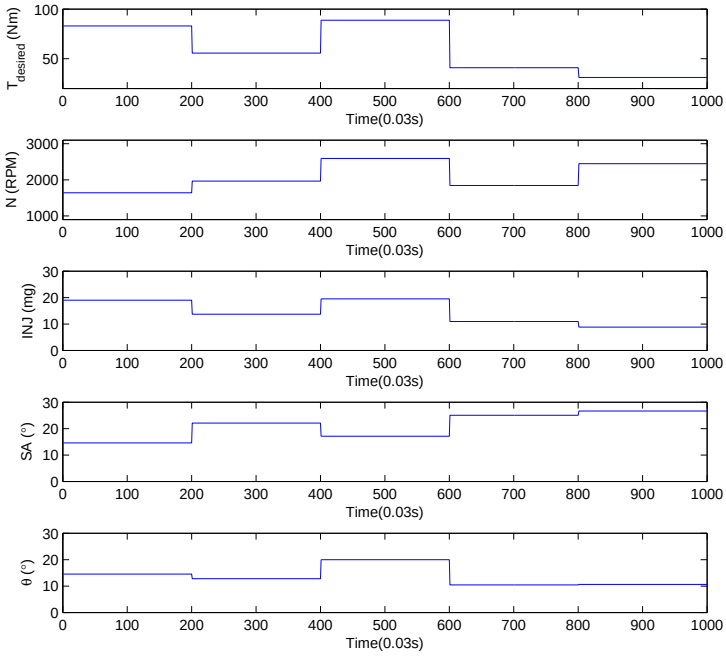


Figure 6.6: Optimal input signals at random operating points

Table 6.1: Measurements of engine output responses

Output	Max settling time	Max overshoot	Max offset
Torque	1.5s	0	1 Nm
λ	2s	0.36	0.01

if the desired torque increase or decrease so that a rich or lean combustion results from the transience which in turn compromises the efficiency of the catalytic converter. The spike is formed due to the interaction between channels of system. With the development of control theory, advanced controllers are able to decouple the relation [104, 105] and have been successfully implemented to solve engine control problems [106, 107]. An alternative approach is using dynamic models instead of static look-up tables since the resulting optimal inputs vary more smoothly and dramatic changes of outputs can be avoid.

To validate the constrained optimization of the fuel economy, a random SA input is used to perturb the RT model instead of the optimal SA input however the inputs of INJ and θ channels are kept at the optimal values. The resulting outputs are compared to optimal inputs in Figure 6.8. Since INJ and θ are kept the same, the amount of fuel mass and air flow mass in each combustion is kept at a fixed ratio therefore λ_{random} is almost identical to λ_{opt}. However T_{opt} is always larger than T_{random} which indicates that the energy generated by combustion can be more efficiently converted to engine torque if the spark ignition occurs at

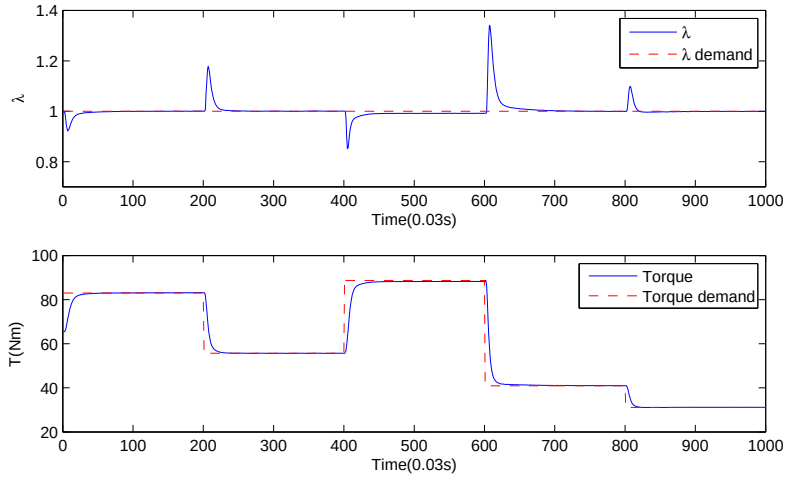


Figure 6.7: Optimal engine outputs at random operating points

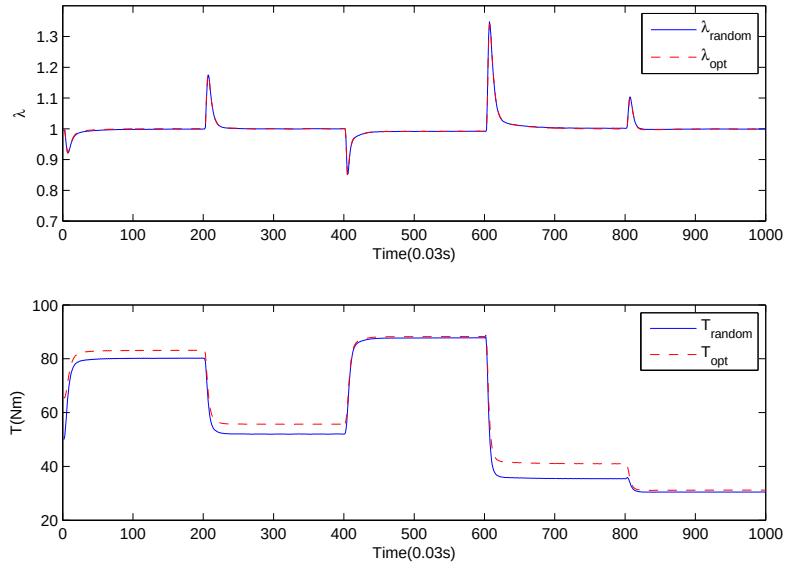


Figure 6.8: Optimal engine outputs with optimal SA and random SA

an optimal angle. Assuming the injected fuel is fully consumed in combustion, it is sensible to express the efficiency of energy conversion over a time period by the total amount of injected fuel and the generated engine torque. Accordingly a measurement of fuel economy is given by:

$$e = \frac{\sum T}{\sum INJ} \tag{6.4}$$

The random SA leads to: $e_{random} = 3.97$ Nm/mg while the optimal SA gives: $e_{opt} = 4.13$ Nm/mg which is 104% of e_{random}. To compare the fuel economy by the e measurement, it is necessary to ensure that the resulting profiles of λ between the two cases are the same otherwise the assessment is meaningless since the requirement on emissions is satisfied to different levels.

By the experiments in this section, it is proved that the obtained calibration map is capable of minimizing the fuel consumption and meeting the requirements on engine torque and λ. Therefore an optimal control performance for the dynamic calibration to compare with is derived in this chapter. This basic hardware-based static calibration is considered to be highly effective as all local tests are conducted on the real system rather than surrogate models. Moreover the time delay between the actuators and sensors can be ignored because only steady-state data are utilized for modelling and control. Any dynamic model based controller has an inherent time delay in control because of the selected dynamic model structure which is composed of delayed input-output regressors. Nevertheless static maps in the SI engine lead to rapid control signals without any significant delay since the relationship between the reference signals and control efforts is determined by time-independent look-up tables.

However as the controllable calibration parameters have been increasing in recent decades, the approach of the hardware-based static calibration is being challenged since the required experimental time and cost will increasing significantly. Model-based static or dynamic calibration methods are thus proposed for less experimental cost and higher efficiency. Although it is not discussed in this thesis, the standard model-based static calibration has been proved to be an effective and efficient approach for engine calibration [108] and related software such as Matlab model-based calibration toolbox [19] has been developed and widely used.

6.8 Conclusions

In this chapter methodologies for conventional steady-state based calibration are introduced and an experiment on engine-based steady-state calibration is conducted to develop a static map which is capable of satisfying the multiple control objectives of torque and λ tracking and fuel optimization. With the purpose of simply demonstrating this method, the operating

region is restricted to the low-load low-speed range. The derived static control map is validated as effective as an accurate feedforward controller which is able to provide rapid output responses and small steady-state offsets. Since the static calibration investigated the engine behaviour at every considered operating point, a detailed prior knowledge of the system is obtained from the tests which is helpful for the subsequent study of dynamic calibration in the following chapter.

Chapter 7

Dynamic Calibration and Controller Design

7.1 Introduction

Since many advanced engine technologies have been introduced to satisfy the increasing requirements of the legislation and market, a number of new calibration parameters are available in modern engines. Engine control systems thus are becoming more and more complex and the associated engine calibration is becoming much more sophisticated. The conventional static calibration process takes an significantly long time to obtain look-up tables with high dimensions and the optimal engine performance might be compromised because the dynamics of system are not addressed. The chapter investigates a way to meet the demands of low cost and high efficiency, by the use of simulation(model)-based calibration which can be incorporated in the calibration process to avoid a significant number of tests at steady-state operating points by using dynamic models and dynamic DoE techniques.

The methodology of simulation-based calibration is firstly carried out on engine models then the controllers developed are implemented and tuned on the real engine. Therefore the quality of models is crucial to the calibration results. The developed models should be capable of accurately regenerating the identification data and also precisely predicting the system behaviour in the operating region of interest. Guzzella [2] and Sun [109] developed a series of engine models for air, fuel and mechanical systems. These models are designed based on physical first principles hence the key parameters can be estimated with a few experiments. With the development of modelling technologies, various types of behaviour-based models have been found to be comparable with the first principle models. Neural network models have been widely investigated for application to automobile industry systems in recent decades. Tan [77] modelled the manifold pressure and mass flow processes with recurrent networks. Saraswati [110] and Xia [111] discussed the reconstruction of cylinder

pressure by NN models. The identification of AFR in the gasoline engine using NN has been studied in-depth by many authors [78, 112, 113]. On the other hand conventional polynomial models are also extensively employed in engine model identification and control [114, 115, 116]. Additionally methodologies of model structure selection, optimal input design and parameter estimation have been developed to improve the accuracy of engine models [117, 118, 119].

An approach to model-based dynamic engine calibration is proposed in this chapter to obtain optimal settings of fuel consumption subject to constraints on torque and λ. In section 2 the principle and general configuration of the calibration process are presented. The whole procedure is classified into 4 steps and a brief description of each step is introduced in section 3. Section 4 details the modelling of engine torque and λ using a DoE approach and estimation technologies. NN models and polynomial models are identified and critically evaluated by output fitness. The selection of objective function, constraints and algorithm for numerical optimization are discussed and an optimization over a fixed length input-output data sequence is given in section 5. A dynamic map developed by an inverse identification of the causal optimal data is presented and the effectiveness of the map is evaluated in section 6. Section 7 demonstrates an approach to further improve the output response using feedback from the virtual engine and open-loop estimators. The computing efficiency of the NN model or polynomial model based optimisation is discussed and an approach of refining the dynamic map is proposed in section 8.

7.2 Basic Model-Based Dynamic Calibration

Figure 7.1 shows a configuration of the basic dynamic gasoline engine calibration and control problem for optimized fuel economy, engine performance and emissions. The whole system is composed of a virtual engine, control model, dynamic map and feedback controllers. The loaded virtual engine to be calibrated is denoted by VE which consists of an RT model and road-load sub-model. In a real production engine, the engine speed is actually affected by load rather than being determined by an engine speed actuator as in the RT model. Therefore it is necessary to provide a reasonable speed profile to the speed actuator in order to simulate the real engine appropriately. Consequently a vehicle-road-load sub-model described in Section 3.7 is employed. By connecting to the sub-model, the virtual engine is simulated as an engine in vehicle. The engine speed is related to the engine torque from the RT model, the resisting load and the load of the vehicle which is determined independently. Additionally the road-load sub-model is able to set different gear ratios hence diverse transient driving cycles can be simulated and examined.

The basic dynamic calibration obtains a feedforward dynamic map and feedback controllers K_T and K_λ to track the engine torque T from the signal $T_{desired}$ and regulates λ to the

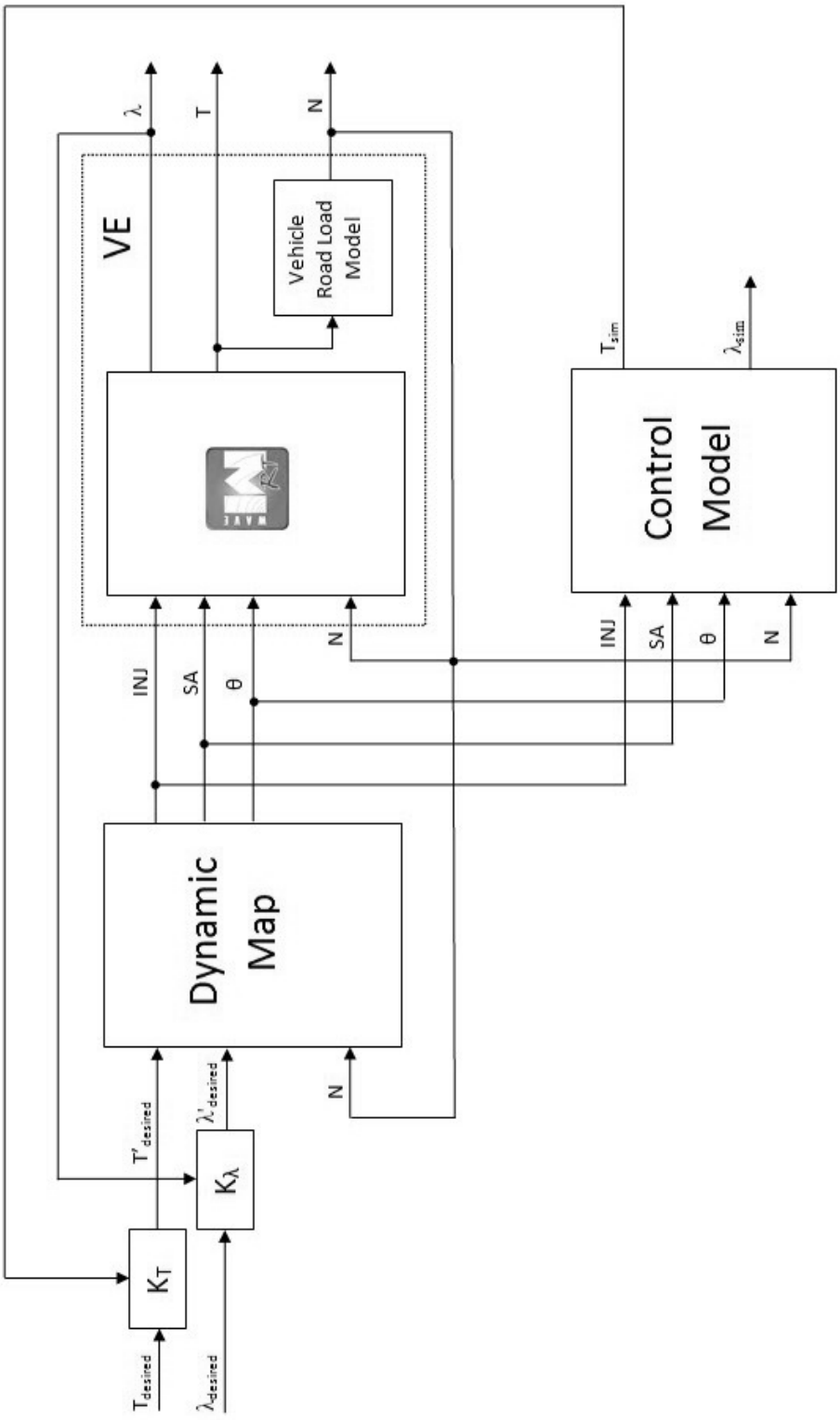


Figure 7.1: Basic dynamic calibration and control configuration

stoichiometric $\lambda_{desired} = 1$ and minimise the fuel consumption subject to these constraints on engine performance and emissions. The vehicle-road-load sub-model then automatically provides feasible engine speeds according to the pre-determined relation between engine torque and speed. Differently from the static map in Chapter 6, the feedforward dynamic map is composed of time invariant dynamic models which are obtained by identification and so the time consuming local experiments at every operating point and smooth curve fitting through these points can be avoided. To design the dynamic map, a constrained optimization of engine inputs over a representative operating region is required in the first place. The obtained optimal data is considered causal and includes rich information of the optimal setting in the desired region so that it is reasonable to acquire the dynamic map by inverse identifications using the resulting engine outputs and the optimal inputs. For the accuracy of tracking desired torque and λ, feedback controllers are utilized to eliminate any offset in the open loop control. However the controllers need to be precisely tuned since the implementation of closed loop control may compromise the settling time of the entire control system. In cases when it is not possible to collect the feedback signals from a production engine, e.g. the engine torque, an accurate model is selected as an open loop estimator which provides a simulated output and the output signals generated by this estimator are used for precise closed-loop control of the real engine.

7.3 The Procedure of Dynamic Calibration and Control

As shown in Figure 7.2, the basic dynamic calibration is consist of in 4 main steps.

1. Identification of Engine Models

In order to reduce the experimental time, the fuel optimization and controller design are carried out offline on engine models. Using the experimental engine test data, a torque model and λ model are identified as:

$$\begin{aligned} T &= T(INJ, SA, \theta, N) \\ \lambda &= \lambda(INJ, SA, \theta, N) \end{aligned} \tag{7.1}$$

As the calibration results obtained by model-based experiments are to be implemented on the real system eventually, the model quality is crucial to the consistency between the model response and engine response. It would be necessary and most effective to apply methodologies of DoE and estimation for the purpose of further improving the model accuracy as much as possible in this first step of calibration.

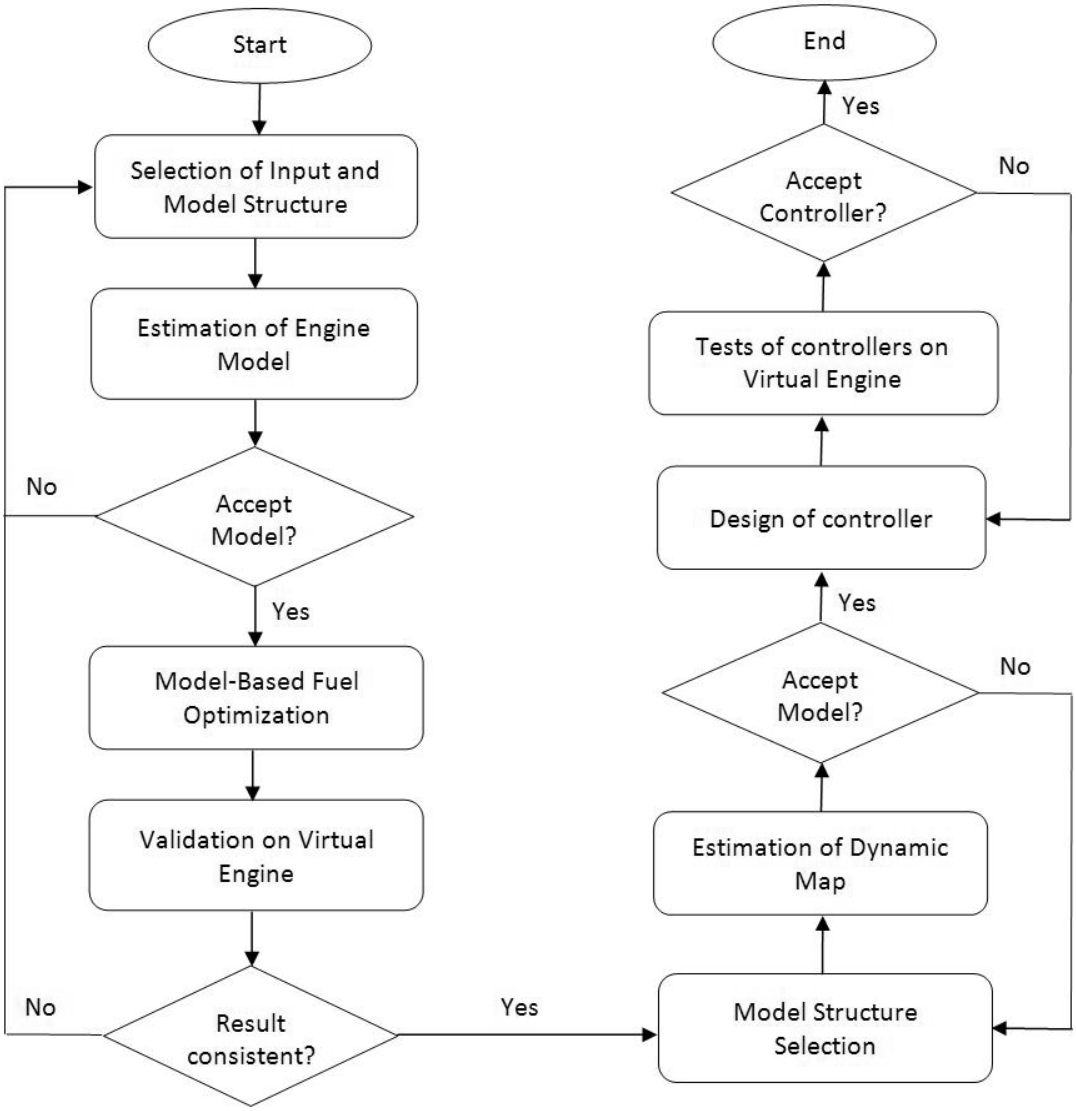


Figure 7.2: The process of dynamic calibration and control

2. Optimization of Fuel Consumption

In this step an optimal causal behaviour between the engine inputs INJ, SA, θ and outputs T, λ, N is investigated by a constrained optimization. The objective function, the consumed mass of fuel (m_f) is given by:

$$m_{fopt} = \min m_f(INJ, N) \tag{7.2}$$

And the corresponding constraints are:

$$\begin{aligned} T(INJ_{opt}, SA_{opt}, \theta_{opt}, N) &= T_{desired} \\ \lambda(INJ_{opt}, SA_{opt}, \theta_{opt}, N) &= \lambda_{desired} \end{aligned} \tag{7.3}$$

The optimization is the start for exploring the optimal settings of the engine control parameters for the overall objectives. It thus has a significant influence on the effectiveness of the dynamic map. Appropriate optimization algorithms and associated settings need to be selected for the efficiency of computation.

3. Identification of Dynamic Map

Inverse models are often used as feedforward compensators to track the desired outputs [120]. Feedback controllers are designed and connected in series with inverse models in order to eliminate the steady-state offset of output response and they can be developed by linear methodologies since the resulting open loop control system has less uncertainty and nonlinearity [121, 122, 123]. In this thesis the following inverse models (causal dynamic maps) identified with the optimal data set have an additional effect of providing minimized fuel consumption so that the three performance objectives of control mentioned in Section 6.3 can be satisfied.

$$\begin{aligned} INJ_{map} &= INJ_{map}(T_{desired}, \lambda_{desired}, N) \\ SA_{map} &= SA_{map}(T_{desired}, \lambda_{desired}, N) \\ \theta_{map} &= \theta_{map}(T_{desired}, \lambda_{desired}, N) \end{aligned} \tag{7.4}$$

The fitness of the inverse identification denotes how well the causal optimal behaviour discovered by the constrained optimization can be presented and utilized with unseen data sets. Although the steady-state offset of torque and λ can be amended by the closed loop control of the next step, it is worth finding a map with the best fit of INJ_{map} and θ_{map} in order to produce the best linearisation of the system for the feedback control design.

4. Design of Closed Loop Control

Because of the sensitive relation between the stoichiometric λ and three-way catalyst efficiency, the allowable offset of the λ control loop is generally required to be less than 1%. Consequently a closed loop λ control is often employed to satisfy this strict requirement. On the other hand the limit on offset of torque is not so strict since the driver is able to adjust the throttle manually if more or less rapid acceleration is desired. Therefore a closed loop control of torque is only necessary if the output error caused by the feedforward controller is significant, above 5%. Despite a torque sensor not being installed in production vehicles, it is nevertheless feasible to obtain a torque model with good quality from powertrain experiments and simulate the engine torque for control action.

7.4 Identification of Engine Models

In general a production engine is operated under various conditions according to different driving cycles. Identifying a model which is universally accurate would be time consuming and often practically impossible. The system identification thus should be control-oriented that is determined by the objectives of the calibration. According to the objective mentioned in Section 6.3, input signals should be able to generate an engine torque in the operating region, 10Nm to 90 Nm and stoichiometric λ. Rather than using a trial and error approach to manually find appropriate inputs to generate the desired outputs, feedback controllers can be used to restrict the engine outputs to safe desired regions as shown in Figure 6.2. The SA is excited with an input amplitude constraint while INJ and θ are determined by closed loop control.

The objective of controller design in this step is different from that of a conventional tracking control hence the principle of refining controller is also different. Generally a feedback controller is developed to precisely track the reference input, as in the controllers produced by the static calibration of this work. In an ideal situation, the corresponding output is expected to have no overshoot, oscillation and short settling time. The design of the controller is thus aimed to minimize these measures as much as possible. However in this stage of the dynamic identification process, the major objective of the control is to regulate the data for identification in the desired region and consequently the tuning work of controllers aiming to optimize the output response can be considerably reduced.

It is worth noting that in order to capture the dynamics of the system, the inputs should be able to excite the system appropriately and the controllers should be adjusted to provided the rapid output responses required. In static calibration it is necessary to validate the effectiveness of controllers against all operating points since the optimal settings of the entire

operating region would be recorded. Nevertheless controllers for dynamic identification are not required to be tested at every operating point because only data along a representative transient profile is used for modelling. This in turn provides more available experimental time for designers to optimize the performance of the controllers. The PI controllers for dynamic engine identification for the RT model were thus obtained by online tuning and are given by:

$$\begin{aligned} K_T &= \frac{0.3 + 0.7s}{s} \\ K_\lambda &= \frac{-20 - 1.5s}{s} \end{aligned} \tag{7.5}$$

7.4.1 Excitation Signals

To identify engine models, $T_{desired}$, $\lambda_{desired}$, engine speed and SA are considered as excitation signals while engine speed, SA, INJ and θ are input signals for the identification. As discussed in Chapter 4, the selection of excitation signals has a significant influence on the accuracy of the system identification. The input should be sufficiently rich to excite the key frequencies of the system and also the nonlinearity. Accordingly the signal must have wide frequency content and include values at multi-levels. To identify nonlinear dynamic models, initially an APRBS signal, which type has the advantages of both amplitude density and frequency content, is employed to excite input channels since it has been demonstrated as a better sub-optimal signal than PRBS and random number signal in Section 4.8.6.

Furthermore the input amplitude and the rate of switching of the input value from one level to another must be decided adequately. To represent the low-speed low-load operating region, the amplitudes of the APRBS inputs are selected as follows:

$$\begin{aligned} 0Nm < \quad & T_{desired} && < 100Nm \\ 0.9 < \quad & \lambda_{desired} && < 1.1 \\ 5^\circ < \quad & \text{SA} && < 30^\circ \end{aligned} \tag{7.6}$$

The range of SA is determined by a knowledge of the system from the previous static calibration and ranges of θ and INJ are determined by $T_{desired}$ and λ. An APRBS signal could vary from a PRBS to a random number signal with the number of input levels changing from two to infinite. The selection of levels is a trade-off between the amplitude density of input and the ability to excite system nonlinearity. According to [124], 5 levels : the lower bound, 1/4 point, middle point, 3/4 point and upper bound are significant levels and are selected for the signal.

From the viewpoint of model identification, the input of engine speed should also be an APRBS sequence. This is feasible in virtual engine experiments since the speed can be instantaneously adjusted by the speed actuator. Nevertheless a profile of actual engine

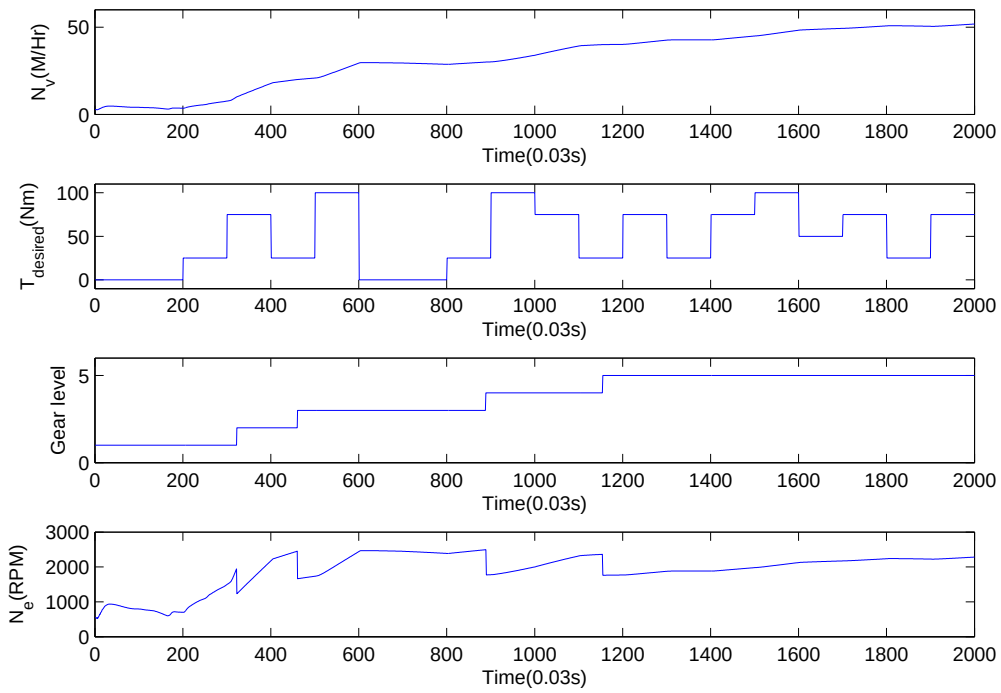


Figure 7.3: A profile of engine speed at the acceleration of vehicle

speed is affected by the comprehensive effect of factors such as vehicle speed, gear ratio and resistance, which cannot be instantaneously controlled by the engine inputs e.g. SA and θ. An arbitrary APRBS input hence becomes less practically meaningful. In this chapter the engine speed is determined by the vehicle-road-load sub-model. An example of the acceleration of vehicle speed(N_v) from 0 to 50 miles/hour with an APRBS engine torque($T_{desired}$) is shown in Figure 7.3, the associated gear level and corresponding engine speed(N_e) are also displayed.

The best time interval for changing the input value between levels depends on the dynamics of the system. If the signal changes too slowly, it would be difficult to capture the dynamics at higher frequency while if changes too fast, the output would have insufficient time to respond fully. It is found that the rise time of the outputs for step change of INJ, SA and θ are approximately: 0.45s, 0.36s and 0.51s. Additionally the time interval of demanded torque and λ must be longer than the setting time of outputs in closed loop control. In this work sets of candidate signals with different time intervals were tried and the time interval was determined as 3 sec for all inputs.

It is acknowledged that the signals of input channels in MISO and MIMO identification must be uncorrelated [125]. Therefore the seeds of the different APRBS inputs need to be different. The determination of input-output sample time and data length involves a

compromise between the prospective model accuracy and computing efficiency. In practice the real-time data can be sampled more frequently than necessary to ensure the high frequency content of the system is adequately recorded. Then the discrete data sequence can be down sampled according to the specific requirements of the modelling such as the determined sample time of the discrete model. Theoretically a long data length is always preferred since it contains more information on the input-output relation however the most efficient data length for identification can be investigated by trials using any prior knowledge of the system. Generally a linear system can be identified with a smaller data length since the linear relationship between input and output can be clearly captured by a limited number of the data points. Moreover the time interval of the input signal should also be taken into account in the selection of the data length and more data points may be required to identify the low frequency mode of the system. In this section, experimental data of one minute of test time was recorded and the sample time was selected as 0.01 sec, which equals to the time of an engine cycle at the speed of 6000 RPM. To identify models working in the low-speed low-load region, the data was further down sampled into 2000-point sequences with a sample time of 0.03 sec.

7.4.2 Neural Network Models of Torque and λ

In recent years the Neural Network has been a popular candidate for system modelling because of its superior ability for describing system nonlinearity [126]. Generally speaking, Neural networks can be classified into static and dynamic categories. In this thesis a specific dynamic recurrent neural network, nonlinear autogressive with exogenous inputs (NARX) network is chosen to represent dynamic systems. The NARX network can be regarded as an extension of the popular time-series linear ARX model and has the advantage of recognising a very large number of nonlinear phenomena in the system. In contrast to other dynamic networks, the current output of the NARX network is not only related to the previous and current input but also the previous output.

The NARX network can be further classified into parallel and series-parallel architecture as illustrated in Figure 7.4 where TDL refers to the time delay. The parallel NARX network is a simulation model in which the delayed output is provided by the feedback of simulated model output. On the other hand, the series-parallel architecture can be set up by using the previous output of the real system as an input when employed in online estimation and control applications [127]. As a prediction model, this network is used in the feedforward architecture however the accuracy of the predicted output is significantly improved if it is possible to measure and provide the previous output of real system as a model input. In the application of this section, the NN model is utilized to replace the real system for offline model based optimization thus the parallel architecture is employed.

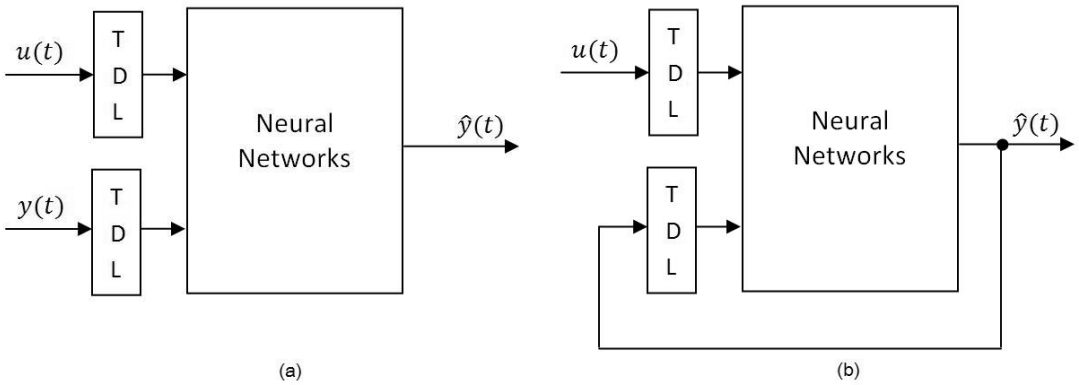


Figure 7.4: Series-parallel architecture (a) and parallel architecture (b)

As suggested by Priddy [80], designers should try to model a real system by a two layer network, a hidden layer and a output layer, since generally additional hidden layers can be presented by more neurons in the first hidden layer. For a reduction of model complexity and computational processing burden, the number of neurons in each layer needs to be determined by downsizing trials and a smaller number which does not result in a significant decrease of model accuracy is preferred. Table 7.1 shows the result of layer size selection. 70% of points in the data sequence were chosen for training, 15% for validating and 15% for testing. Since the data for training, validating and testing are randomly selected in each trial [81], a statistical result is more convincing so that each network was trained ten times with the same data set and the mean result presented.

Table 7.1: Selection of layer size

Layer size		$\overline{\text{MSE}}$	$\overline{\text{MSE}}$	$\overline{\text{MSE}}$
5	1	25.58	51.70	31.15
10	1	20.45	39.01	27.76
20	1	20.32	41.97	27.10
30	1	19.31	35.84	28.75

The maximum delays of input and output are selected to be 5 and 1 so that the equation of the NARX network is given by:

$$\hat{y}(t) = f\left(u(t-1), u(t-2), u(t-3), u(t-4), u(t-5), \hat{y}(t-1)\right) \tag{7.7}$$

Figure 7.5 shows the architecture of selected NARX NN for both torque and λ. The first layer has 10 neurons with a tan-sigmoid activation function and the second layer has 1 neuron with a linear activation function. Levenberg-Marquardt backpropagation [128, 129] is employed as the training algorithm.

Using the settings mentioned above, the NN models were trained 10 times offline using

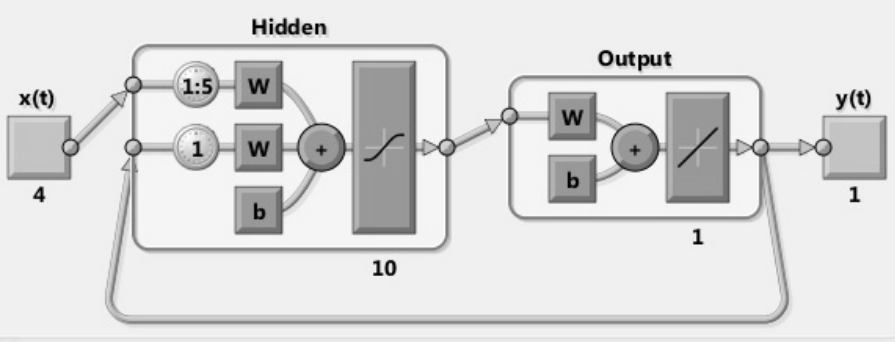


Figure 7.5: The architecture of selected NARX neural networks

Table 7.2: Testing results of neural network M_T and M_λ

	M_T	M_λ
MSE	92.86	0.0251
R^2	91.13%	89.52%

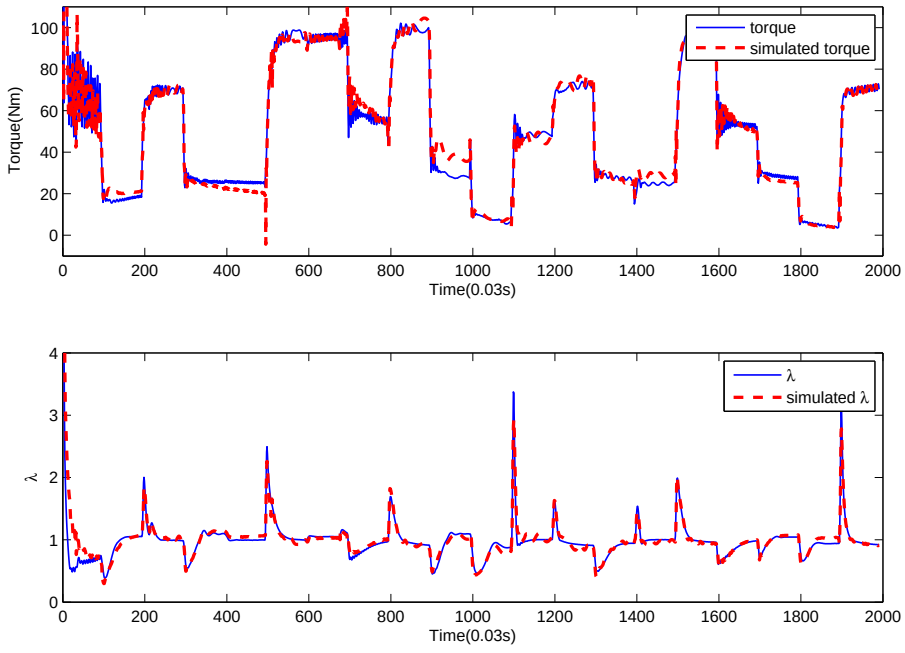


Figure 7.6: Simulated engine outputs and real engine outputs

the identical identification data. The best testing result with respect to MSE and R^2 as criteria are given in Table 7.2. Corresponding models of torque and λ are selected to simulate the identification data. As displayed in in Figure 7.6, the simulated outputs closely matches the real system outputs.

7.4.3 Polynomial Models of Torque and λ

The polynomial model is another competitive candidate for modelling dynamic systems. Models are expressed by algebraic equations in which the relationship between input and output time series is clearly exhibited and it is less time consuming to estimate the parameters than with the NN structure. In many practical applications polynomial models are preferred for the ease of being programmed and low computational burden. In order to find a model that gives a satisfactory output fitness of the identification data, various model structures were tested. To establish the model structure, firstly the linear terms was be added and then the nonlinear quadratic terms. With these trial tests the structures of torque model and λ model are determined as:

$$\begin{aligned} y_T(t) &= \theta_1 + \theta_2 u_1(t-1) + \theta_3 u_1(t-2) + \theta_4 u_1(t-3) + \theta_5 u_2(t-1) + \theta_6 u_2(t-2) \\ &\quad + \theta_7 u_3(t-1) + \theta_8 u_3(t-2) + \theta_9 u_4(t-1) + \theta_{10} y_T(t-1) + \theta_{11} u_1(t-1) u_2(t-1) \\ &\quad + \theta_{12} u_1(t-1) u_3(t-1) + \theta_{13} u_2(t-1)^2 + \theta_{14} u_2(t-1) u_3(t-1) + \theta_{15} u_3(t-2)^2 \\ &\quad + \theta_{16} u_3(t-2)^3 \\ y_\lambda(t) &= \theta_1 + \theta_2 u_1(t-1) + \theta_3 u_1(t-2) + \theta_4 u_2(t-1) + \theta_5 u_3(t-3) + \theta_6 u_4(t-1) + \theta_7 y_\lambda(t-1) \end{aligned} \tag{7.8}$$

Using the same identification data as with the NN models and the PEM estimation method, the parameters obtained are:

$$\begin{aligned} \hat{\theta}_T &= [-8.91, 1.86, 0.58, -1.34, 1.06, 0.18, 0.61, 0.88, -0.0087, \\ &\quad 0.72, -0.02, 2.96 \times 10^{-4}, -0.0244, -0.0068, -0.0177, -1.69 \times 10^{-4}] \\ \hat{\theta}_\lambda &= [0.61, 0.091, -0.10, -1.07 \times 10^{-4}, 0.0083, 0.82, -1.87 \times 10^{-4}] \end{aligned} \tag{7.9}$$

The obtained polynomial models are used to regenerate the identification signal and the evaluated fitness is shown in Table 7.3.

Table 7.3: Testing results of polynomial M_T and M_λ with PEM

	M_T	M_λ
MSE_{PEM}	157.42	0.0431
R^2_{PEM}	84.56%	82.02%

In order to further improve the model accuracy, the proposed DoE methods are employed. The optimal input is designed by using the AI-optimal criterion and used as the signal for

identification. The developed SEM estimation method is selected to estimate the parameters of the model. The derived optimized parameters are then given by:

$$\begin{aligned}\hat{\theta}_T &= [-21.33, 1.43, 0.60, -1.09, 1.38, -0.37, 0.62, 0.88, -0.0016, \\ &\quad 0.77, -0.02, 3.96 \times 10^{-4}, -0.0202, -0.0092, -0.0132, -2.36 \times 10^{-5}] \\ \hat{\theta}_\lambda &= [0.51, 0.15, -0.16, -3.18 \times 10^{-4}, 0.012, 0.64, -6.96 \times 10^{-5}]\end{aligned} \tag{7.10}$$

Table 7.4 shows the resulting improved output fitness.

Table 7.4: Testing results of polynomial M_T and M_λ with SEM

	M_T	M_λ
MSE_{SEM}	125.68	0.0247
R^2_{SEM}	87.99%	89.70%

Over-fitting may occur in any system identification because a model is developed to maximise or minimise a performance index, e.g. minimising the MSE, for the identification data but the accuracy of the model is determined by its performance on predicting or simulating unseen data. An over-fitted model will general give a poor predictive performance on the unseen data therefore the models obtained in Section 7.4.2 and 7.4.3 should be validated by other data sets.

7.4.4 Validation of Engine Models

The results in Table 7.2 and Table 7.4 indicate that the obtained NN models and polynomial models provide good estimations of the identified data set. However, qualified models are expected to have the capability of accurately simulating signals that are uncorrelated to the identification signals. To further validate the models with unseen data, 10 other sets of data with different seeds were collected from the real system and simulated by the models. Figure 7.7 demonstrates an example of engine outputs and simulated outputs by NN and polynomial models. An averaged validation result for these 10 sets is shown in Table 7.5

Table 7.5: Validation results of NN and polynomial M_T and M_λ

	M_{TNN}	$M_{\lambda NN}$	M_{Tpoly}	$M_{\lambda poly}$
$\overline{MSE}$	47.11	0.0254	150.95	0.0277
$\overline{R^2}$	95.41%	88.27%	85.26%	86.65%

The validation results indicate that NN models lead to better output fitness in this application especially in the torque response. In cases where NN models are not available due to limits of industrial practice, nonlinear polynomial models can be employed and methodologies of model structure selection can be used to improve the accuracy of the polynomial

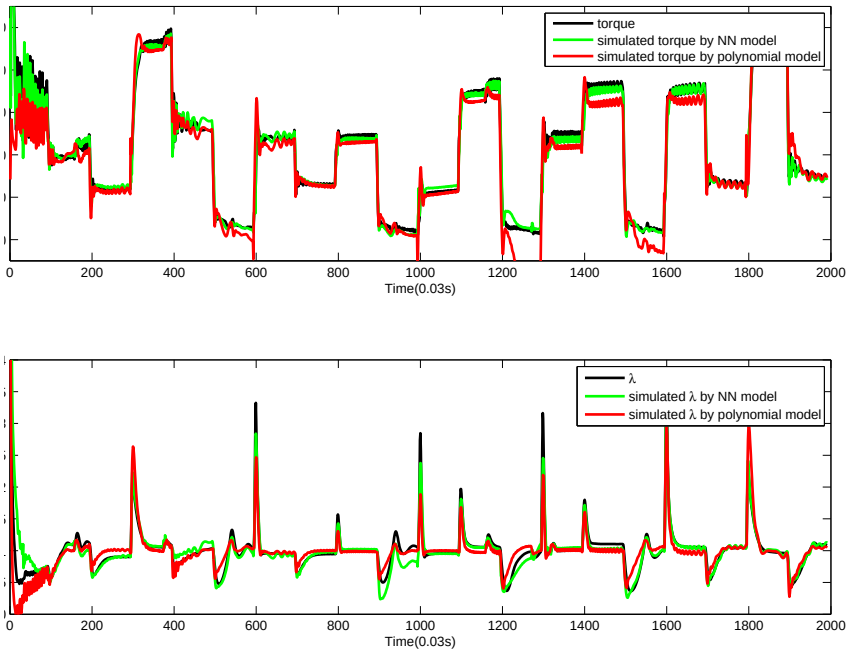


Figure 7.7: Validation of NN and polynomial models

models [76, 130, 131]. In the process of dynamic calibration, the models developed in this step are utilized for offline fuel optimization and controller design. Since the engine models will not be programmed and stored in the ECU, it is feasible to choose NN models to obtain the required high accuracy for use in the subsequent optimization process.

7.5 Neural Network based Fuel Optimization

With the obtained NN models, the next step is using a numerical optimization in order to minimise the fuel consumption and satisfy the constraints on torque and λ. The Matlab Optimization Toolbox [132] is utilized to solve the constrained optimization in this thesis. Options of the optimization program are configured as follows:

7.5.1 Initial Conditions of Optimization

The objective of fuel optimization is to find optimal engine inputs which lead to minimized fuel consumption in the operating region of interest. A set of engine inputs is selected as the initial conditions for the numerical optimization and then the obtained optimal data is used to develop a dynamic map which is able to generate optimal inputs in the whole region. Therefore the initial input signals should excite the system adequately so that the corre-

sponding input-output data includes sufficient information of the optimal system behaviour over the whole desired operating region.

In this work a set of APRBS signals which have the same amplitude constraints and same time interval as the validation signal in Section 7.4.4 and the corresponding engine speed generated by the vehicle-road-load model are applied to the NN models of torque and λ; the data length is selected as 2000 points. The resulting INJ, SA, θ, engine speed are chosen as initial values of inputs and the corresponding simulated torque and λ from the NN models are chosen as constraints of the subsequent model-based fuel optimization.

7.5.2 Design of Objective Function and Constraints

The objective of the constrained optimization of the fuel economy is to minimize the amount of injected fuel over a period and to satisfy the constraints simultaneously. Accordingly the number of fuel injections which are related to engine speed N and the fuel mass in each injection INJ in this period is an essential variable that determines the total fuel mass. The mass of fuel m_f is given as a function of fuel injection and speed by:

$$m_f(kg/hour) = 1.2 \times 10^{-4} \cdot INJ(mg) \cdot N(RPM) \tag{7.11}$$

In a specific optimization, parameters concerned with the working condition such as the desired torque, λ and engine speed are fixed as the initial values. INJ, SA and θ are three variable vectors that will be manipulated to realize the objective of the optimization.

The objective function of the dynamic calibration is different from that of the basic hardware-based static calibration presented in Chapter 6 although the fuel mass in each injection will be involved in both calibrations. Because system dynamics are neglected in that static calibration, the current output is only related to the current input, in other words, the engine parameter settings of an operating point are completely independent from those of other operating points. Therefore minimizing the fuel consumption over a period is equivalent to minimise the m_f at each individual operating point. As the engine speed is fixed at a certain operating point, the objective can be simplified as minimising the INJ at each point. However in dynamic models, the constraint at the current time instant is affected by the INJ of several past time instants. Correspondingly minimizing the m_f at a time instant may compromise the m_f at other instants because of the constraints. The objective of dynamic calibration is therefore to minimize the average m_f, which is expressed in the form:

$$\min \frac{\sum_{t=1}^{n} m_f(t)}{n} = \min \frac{\sum_{t=1}^{n} f(INJ(t), N(t))}{n} \tag{7.12}$$

where INJ(t) is the manipulated variable, $N(t)$ is a pre-determined sequence and n is the length of the discrete time sequence. This objective function for fuel optimisation is a simplification for the purpose of demonstrating the proposed dynamic calibration in this thesis. It needs to be further improved for practical industrial implementations.

To track the desired torque and λ, equality constraints on the output values are applied to the optimization of the form:

$$\begin{aligned} T(INJ(t), SA(t), \theta(t)) &= T_{desired}(t), \qquad t = 1, 2 \ldots, n \\ \lambda(INJ(t), SA(t), \theta(t)) &= \lambda_{desired}(t), \qquad t = 1, 2 \ldots, n \end{aligned} \tag{7.13}$$

Because of the general nonlinearity of the torque model and λ model, these two constraints are treated as nonlinear constraints and will be converted into an unconstrained optimization problem by penalty function. Since the scales of the desired torque and λ are quite different, appropriate weightings must be added for balance otherwise the constraint for the small scale signal will be compromised in the optimization and cannot be appropriately met. Additionally the level of importance of constraints in the experiment could be another factor in the optimisation criterion through the choice of weightings. In this application the requirement for stoichiometric λ is more serious than that for achieving the desired torque because the engine torque can be easily adjusted by the driver nevertheless a 1% error in λ will significantly lower the working efficiency of the catalytic convert. The weighting ratio of torque and λ is arbitrarily chosen as 1:1000 in this optimization in order to ensure that the important λ constraint can be well satisfied. More advanced methods of determining the weighting ratio can be employed, for instance by weighting the scales of the torque constraints and λ constraints. However the results show that the ratio 1:1000 used in this case is effective in balancing the torque and λ constraints.

Inequality constraints on input amplitudes should also be considered to avoid physically unavailable settings. From prior knowledge of the engine and the static calibration result, the inputs are accordingly constrained as:

$$\begin{aligned} 0mg < &\quad \text{INJ} &< 30mg \\ 5^\circ < &\quad \text{SA} &< 30^\circ \\ 0 <^\circ &\quad \theta &< 90^\circ \end{aligned} \tag{7.14}$$

7.5.3 Optimization Algorithms

Since the objective function in equation (7.12) is purely linear, the interior point algorithm can be employed to solve the local optimization problem for this system efficiently. An increasing number of iterations of this algorithm would improve the optimization result at the expense

of increasing the computational time. Therefore a prior knowledge of the optimization from trials is very helpful in selecting the iteration number and this is determined as 50000 for these experiments. In the optimization, each point of the input sequence is considered as a variable, the number of variables is dependent of the length of data and the number of input channels. The time required to process the optimization fully across the full data length would be very long with a large number of variables. Constraints also have a significant influence on the computing work. From equations (7.13) it is learnt that the total number of nonlinear constraint is 4000, twice the data length. The number of variables included in each constraints is determined by the model structure of torque and λ and this is 15 as shown in equation (7.7). In order to process the optimization efficiently, three approaches are studied:

1. Batch approach

In this approach, the variables of the entire data sequence are manipulated in one optimization. The advantage of the batch approach is that the result of optimization, if it is practically possible to achieve it, may be very accurate since all input points are optimized under full constraints. However, there are two major disadvantages with this approach. The first disadvantage is that it does not take the causality into account because it can use information about future behaviour and disturbances to determine current control inputs. The second disadvantage is the high computational demand on memory and computational time. In fact for the data lengths considered in this thesis, this is quite impractical. As described in Chapter 4, the optimization algorithm approaches the optimal value asymptotically with a number of iterations. Therefore 6000 variables subjected to 4000 constraints are processed in each iteration, which means a vast size of memory is required to load and run the optimization and it results in an extremely heave computing burden. In practice it was found that the computer memory that can be utilized by a 32bit version of Matlab is even not enough to execute the optimization.

2. Segment approach

To reduce the computational work in each iteration, a novel segment approach has been investigated which splits the whole data sequence into continuous sub-sequences or segments and optimizations are then performed sequentially on each separate segment, each of which includes much fewer variables and constraints. This method is found by experiment to be practically effective. However since the optimizations of the segments are carried out independently, the data at the connecting points between two sequential segments might not be consistent in end and initial condition and so may not be well optimized overall. In dynamic models, the current output of the objective function or constraint function is often related to

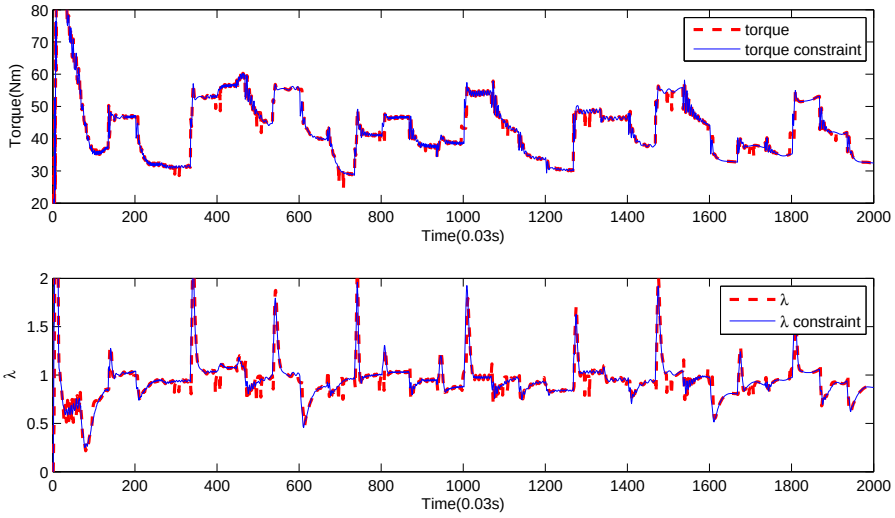


Figure 7.8: The effect of segment approach on output constraints

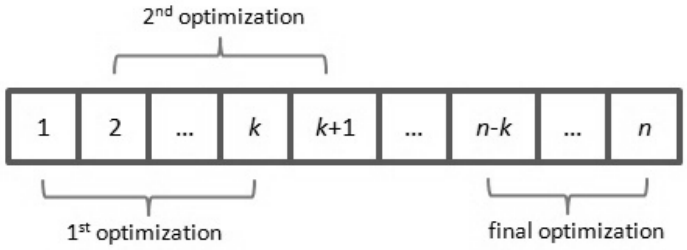


Figure 7.9: A schematic of the predictive horizon approach

the previous values of the input and output. This means that in order to meet the constraints on the first several points of a current torque or λ segment, the last a few points of previous INJ, SA and θ segments will be affected. Therefore the constraints on the previous segment may not be well satisfied.

Figure 7.8 shows an example of the torque and λ constraints in a fuel optimization using the segment approach. The "spikes" in output are caused by the compromised initial condition values at the connecting part between the segments. However, in practice it has been found that these spikes can be reduced by smoothing these connecting points of the data though the smoothing may have a negative influence on the result of the fuel optimization. The segment approach overcomes much of the computational burden problem of the batch approach but retains the disadvantage that the optimization may be non causal.

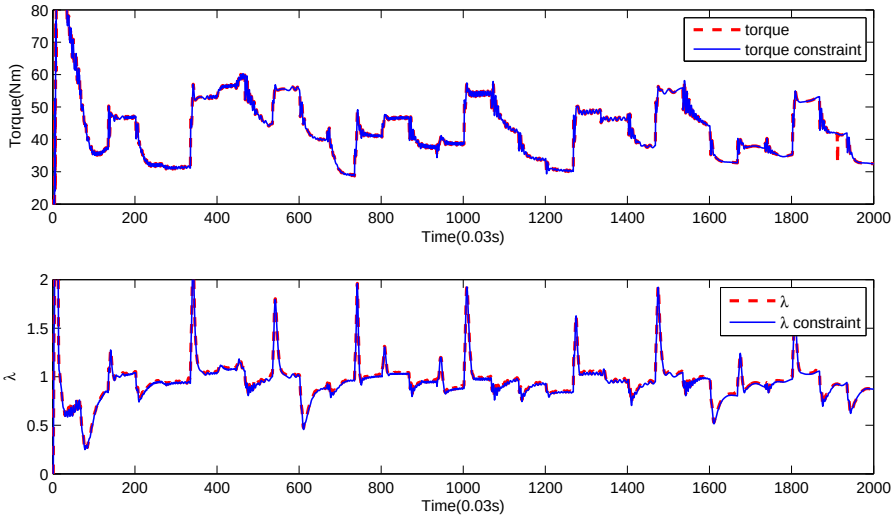


Figure 7.10: The effect of predictive horizon approach on output constraints

3. Predictive Horizon approach

The predictive horizon approach is demonstrated by Figure 7.9. This method is similar to explicit model predictive control (MPC) [133]. A horizon of length k is selected from the start of the entire input sequence and an optimization of the selected data is carried out. Although the whole horizon is optimized, only the optimal value of the first input point will be recorded and then the horizon will move one step forward for the second optimization. This process continues until the horizon reaches the end of the input sequence. Using this approach the constraints can be satisfied very accurately as displayed in Figure 7.10.

The required computational work might be relatively heavy since the optimization of the horizon will be repeated for the entire data length. However it is more feasible than the batch method because of the adjustable length of the horizon. In addition it produces a causal optimisation. To improve the efficiency, the size of the forward step and the number of optimal values recorded in each optimization can be adjusted from 1 to k at the expense of sacrificing the constraints. The predictive horizon approach is identical to the segment approach if the size of the forward step is identical to the length of horizon.

In this thesis the segment method is chosen because of its high computational efficiency. The connecting data of neighbour segments is simply smoothed by using the mean value over a narrow connecting area. Figure 7.11 illustrates the effectiveness of the smoothing. The spike in the outputs around the 50th sample instant is remarkably reduced by smoothing the input data. Although the spikes in the constraints cannot be completely eliminated by means of manually smoothing, in the experiments of later sections it is found that their effects can

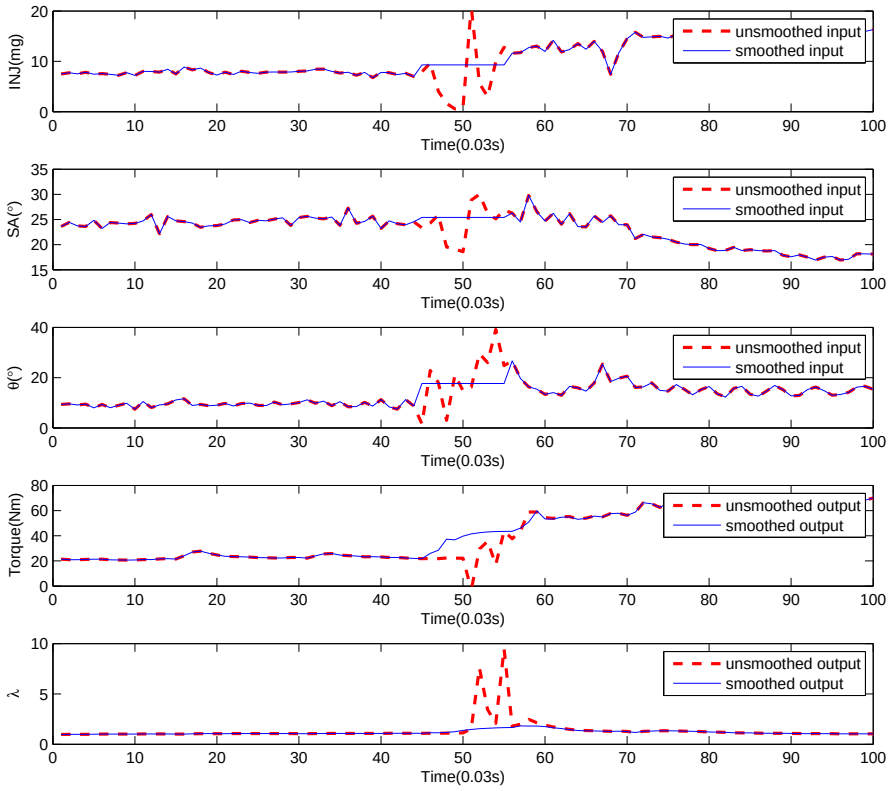


Figure 7.11: An example of the effect by input smoothing on the outputs

be further filtered to a large degree by the inverse compensator which will be used as the dynamic map and also by any feedback controller if this is implemented.

7.5.4 Optimization Results

In the proposed fuel optimization, the whole 2000-point data is evenly divided into 20 segments. Using the interior point algorithm with 50000 numerical iterations in each optimization, the resulting optimal inputs: INJ, SA, θ are shown in Figure 7.12.

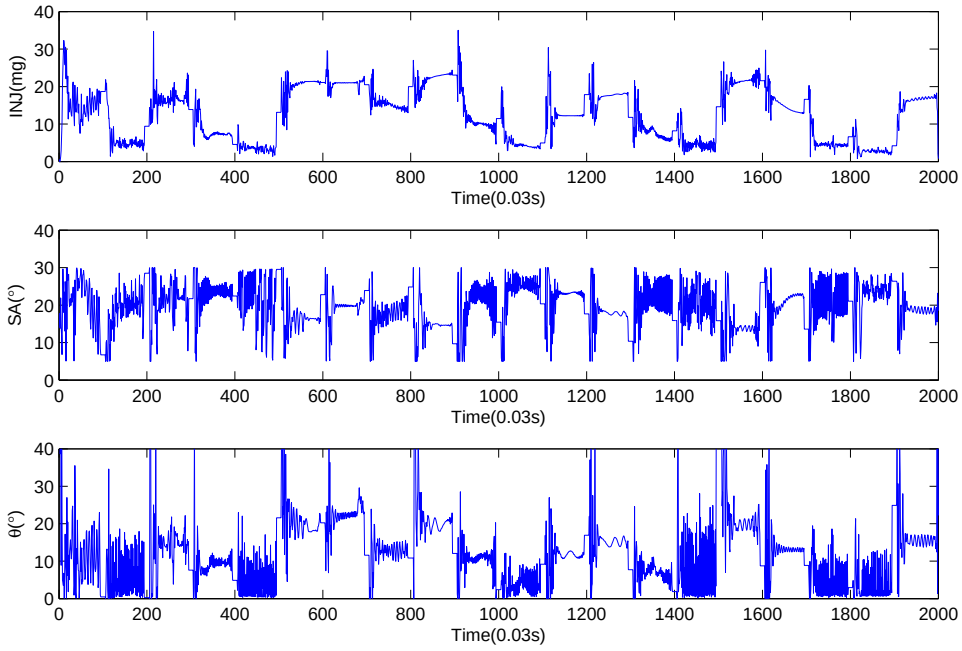


Figure 7.12: Optimal inputs obtained by constrained fuel optimization

The obtained optimal input looks "noisy" since parts of the input values change drastically. This problem is caused by the settings of the optimization. In the numerical optimization, each point of the input is treated as an individual variable and the algorithm will adjust these variables in order to meet the equivalent constraints. Since these variables are considered independent from each other, the resulting optimal input may not be smooth. In order to solve this problem, additional constraints such as input rate constraint can be applied to relate the neighbour points of input in time series. Moreover, in later steps of the dynamic calibration, a feedforward controller will be developed by an inverse identification in order to generate the optimal inputs. Due to the inverse identification, this controller will produce approximated smooth signals rather than the same inputs in Figure 7.12 so that the control efforts which will be applied to the real system are not "noisy".

Comparing to the original fuel consumption of $m_f = 3.54kg/hour$, the optimized fuel consumption is $m_{fopt} = 3.05kg/hour$ and:

$$\frac{m_{fopt}}{m_f} = 86.17\% \tag{7.15}$$

Figure 7.13 demonstrates the demanded constraints and the output responses on the NN

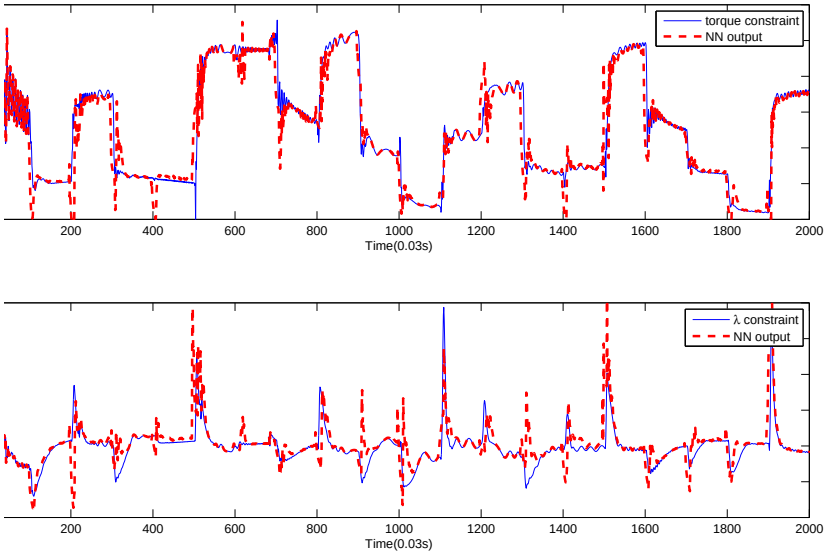


Figure 7.13: Demanded constraints and optimal outputs on NN models

models generated by the optimal inputs. It is found that the constraints were not satisfied too well in segments with the selected iteration number when using the interior point algorithm. Although the error can be reduced by using a larger number of iterations, the computational efficiency of the optimization then deceases accordingly. As closed loop control can be implemented to further regulate inputs to satisfy the constraints in the last stage of the calibration, it is sensible to set the options for the optimization algorithm for the most efficient computation at this stage.

Since the optimal inputs lead to a minimized fuel consumption and generally meet the constraints satisfactorily, the constrained model-based fuel optimization is verified as being effective. The final validation would require that the performance of optimization however should be validated on the real system. The optimal inputs can only be considered practically useful only if the generated outputs of the RT model closely match the outputs of the NN models.

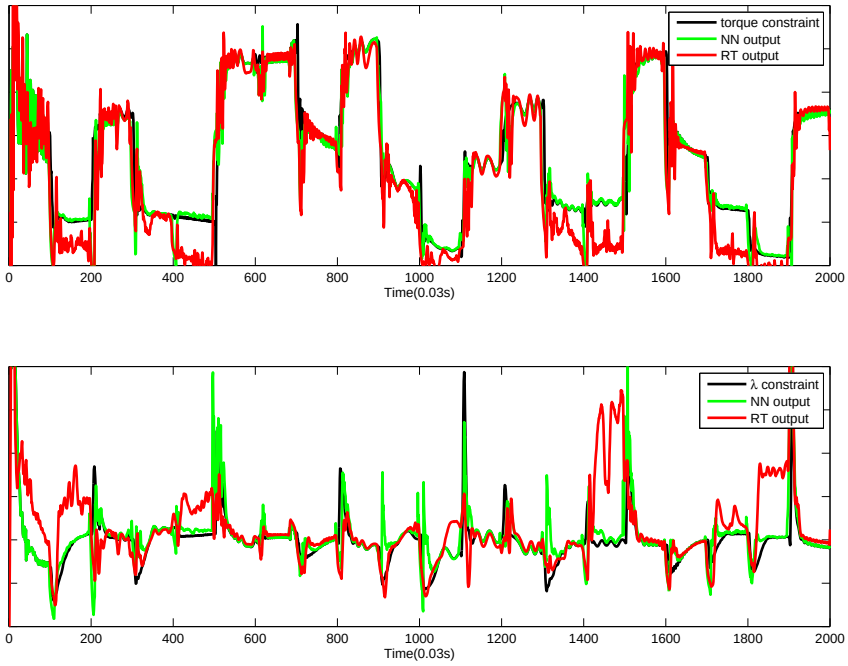


Figure 7.14: Optimal outputs on NN model and RT model

7.5.5 Adaption for Output Consistency

The numerical optimization of the fuel consumption has a significant influence on the performance of the dynamic map which is identified based on the inverse optimal data. In order to achieve a well optimized and reliable result, first of all sufficient iterations in the numerical optimization should be conducted in order to guarantee that a satisfactory causal optimal behaviour of the system can be found. Secondly the identifiability of the obtained engine control optimal inputs should be considered to ensure that the causal system behaviour can be represented by an inverse identified dynamic model. Moreover since the constrained fuel optimization is based on engine models, it is crucial to apply the obtained optimal signals to the real system and evaluate the consistency of the resulting system outputs to the simulated outputs. Figure 7.14 displays the profiles of the optimal outputs collected from the NN models and the virtual engine, where the fitness of the RT output to the demanded constraints is:

$$R^2_{Topt} = 79.96\% \qquad R^2_{\lambda opt} = -222.86\% \tag{7.16}$$

Since the segment method is selected, the NN outputs have spikes at the connecting points but meet the constraints closely at other points. However in the RT outputs the spikes are filtered but large errors exist at points across the whole output sequence. Comparing to the validation results in Section 7.4.4, the fitness of the optimal RT output is remarkably small.

This indicates that the identified NN models could represent the system dynamics accurately if tested with inputs that are similar to the identification signals however the models may not be qualified to simulate the system behavior accurately if tested with the optimal inputs. In the time domain the optimal inputs change much more quickly and drastically than the signal used for identification as shown in Figure 7.12.

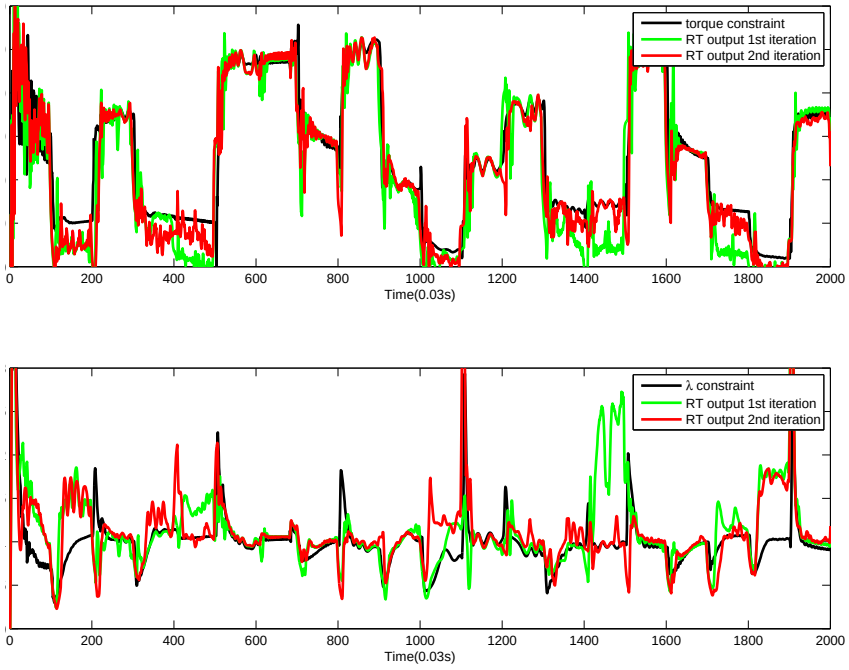


Figure 7.15: Optimal outputs on RT model by iterations

To improve the output consistency two approaches can be employed. The first approach is to reidentify the engine models using the obtained optimal inputs and then to repeat the fuel optimization with the revised models. It is well known a model can easily represent the system behaviour accurately in the validation if the validation signals have similar properties to the identification signals in the time domain and frequency domain.

Assuming the identification signals are chosen as the initial data signals used in the optimization, the resulting optimal inputs which are considered as validation signals must necessarily be different from the initial signals since the inputs must be adjusted to optimize the fuel consumption. However the difference between the signals can be reduced asymptotically by running the identification and optimization iteratively. Figure 7.15 and Table 7.6 illustrate an example of the effect of this approach. In the first iteration there is a large discrepancy between the resulting RT output and the desired constraint while this distinction is significantly reduced in the second iteration and the output fitness is enhanced correspond-

ingly. As no additional constraint is added to the optimization, the search region in each iteration is not further limited. Theoretically the true optimal value is always achievable providing sufficient iterations are conducted in the identification and optimization. Nevertheless the major disadvantage here is the large amount of experimental time required to repeat this comprehensive procedure.

Table 7.6: The fitness of RT output to desired constraints in iterations

	1st iteration	2nd iteration
R_T^2	79.96%	88.16%
R_λ^2	-222.86%	-79.88%

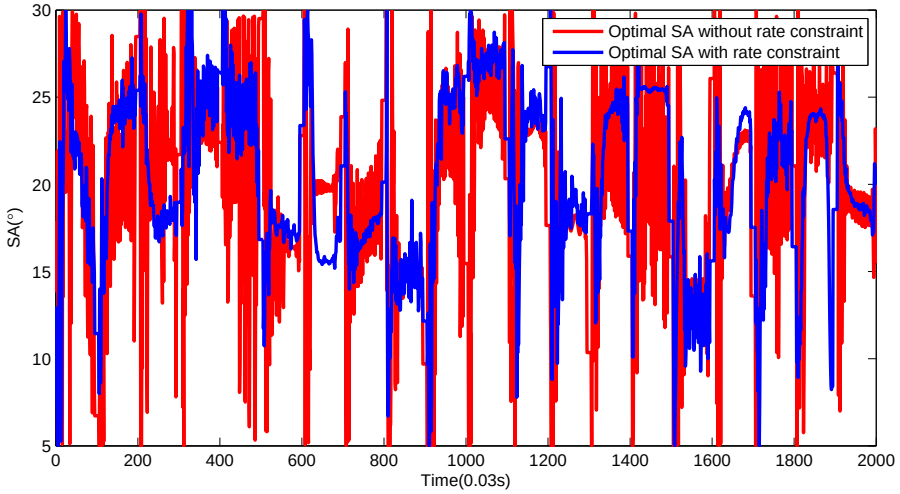


Figure 7.16: Optimal SA obtained with/without rate constraint

Table 7.7: The fitness of RT output to desired constraints with/without rate constraint

	No rate constraint	With rate constraint
R_T^2	79.96%	94.66%
R_λ^2	-222.86%	28.43%

Besides processing the model identification and the fuel optimization iteratively, additional constraints can be applied to the optimization with the purpose of improving the output consistency. The major difference between the identification signal and the optimal signal is the time interval and the rate of change. To compensate for the dissimilarity, a constraint on the rate of input change would be effective however it should be implemented on the inputs selectively since additional constraints may compromise the optimization result and computational time. In the dynamic calibration procedure, the INJ and θ values mainly determine the generated torque and λ and are adjusted by feedforward and feedback

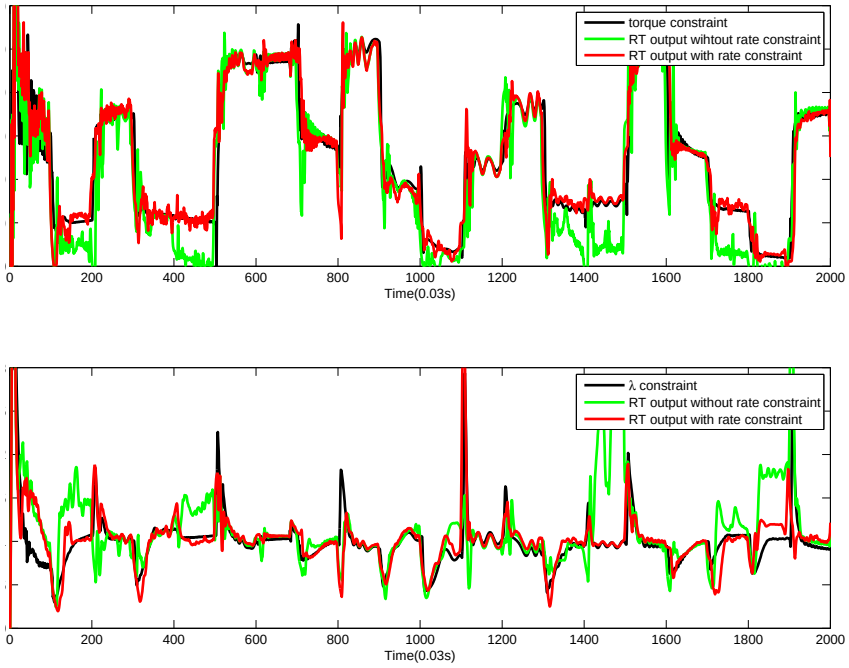


Figure 7.17: Optimal outputs on RT model with/without rate constraints

controllers. On the other hand the SA is only controlled by the feedforward compensator consequently a SA map with good accuracy is crucial for the fuel economy. A rate constraint $\Delta u = u(t) - u(t-1) = 1$ was applied to SA and the optimal signal obtained with and without the rate constraint is shown in Figure 7.16. The corresponding outputs on the RT model are illustrated in Figure 7.17 and the validation result is given in Table 7.7, proving that the rate constraint has a significant effect on the consistency of the output.

7.6 Design of Dynamic Map

In the automotive industry, EMS strategies generally use look-up tables as static maps to control the actuators. In the basic static calibration presented in Chapter 6, 36 different values of SA needed to be tested at each operating point and the settling time for each test is 10 sec. The experimental time required to collect data for 121 operating points is thus 43560 seconds. For the accuracy of static calibration, a larger amount of operating points may need to be mapped which results in an even longer experimental time. To overcome the disadvantage of the cost in experimental time of the static calibration, many authors have attempted to characterise the desired behaviours of system by dynamic models [134, 135]. The methodology of dynamic mapping proposed in this thesis refers to the prediction of the

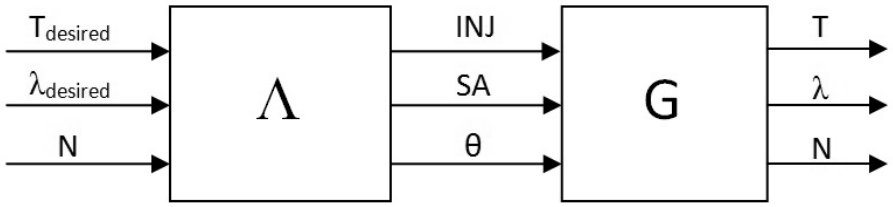


Figure 7.18: A Schematic of feedforward controller

optimal setting of the actuators by using an inverse feedforward controller, the first form of which was initially used by Aquino [136]. Since the optimal inputs obtained in the last section are causal, the feedforward controller can be designed by using the inverted optimal data set. Three inverse MISO models are designed to predict the optimal INJ, SA and θ by using desired torque, λ and engine speed as inputs. This method is quicker in experimental time than the static calibration since only one set of optimal transient data is required for inverse model identification. In order to generate the optimal data, a representative set of initial data which sweeps over the operating region is needed in the optimization. Figure 7.18 depicts the structure of an inverse feedforward controller Λ. After the identification the desired λ input is set to 1 so that the controller predicts the optimal inputs under the stoichiometric condition.

7.6.1 Synchronisation of Optimal Data

To design rapid and accurate feedforward controllers the inherent time delay between inputs and output need to be removed appropriately before modelling to ensure causality of the inverse system. The length of the system time delay can be determined by simple step tests on the system. For an IC engine most of the mechanical, chemical and thermal reactions are combustion based. The time required for every combustion event which is equal to the event of a 720° crankshaft rotation for a 4 stroke 4 cylinder engine. 360° is thus a reasonable choice of sample time. Accordingly the sample time of the NN models obtained in Section 7.4.2 and the following polynomial models is selected as 0.03 sec which equals to the time of a 360 ° rotation at the engine speed of 2000 RPM.

The first type of time delay is caused by the transport delay because of locations of the sensors and actuators. To develop appropriate controllers, the significant delay due to transportation lag should be removed before modelling. For instance since the λ sensor is placed in the exhaust pipe close to the catalyst, a pure time delay is caused by the transportation of the exhaust gas from the exhaust port to the sensor and the output in the input-output data should be advanced accordingly to describe an instantaneous causal reaction. Addi-

tionally the reaction time of sensors to report the experienced output should also be taken into account if it is not much smaller than the sample time of model. In the RT model, it is assumed that the λ sensor is placed at the exhaust port and ideal sensors and actuators with no reaction time are used. Therefore the time delay of transportation and sensor reaction can be ignored with this model.

Since the feedforward controller is composed of delayed and cross-related regressors and is identified with inverse input-output sequences, the other type of time delay is caused by the selected structure of the dynamic controller and the structure of the related dynamic engine models. Assuming m and n are the maximum time delays of the engine model and the controller respectively, in order to generate the same output sequence Y the input of the controller should be Y' which is given by:

$$y'(t) = y(t - m - n) \tag{7.17}$$

7.6.2 Inverse MISO Feedforward Controller Identification

In this step polynomial models are employed to describe the controllers because they are more easily programmed and less resource demanding in the ECU than NN models. The selected model structures of the 3 nonlinear dynamic control maps are:

$$\begin{aligned}
y_1(t) &= a_1 + a_2u_1(t-3) + a_3y_1(t-1) + a_4u_1(t-2)u_2(t-3) + a_5u_2(t-3)y_1(t-1) \\
&\quad + a_6u_3(t-3)y_1(t-1) \qquad (7.18)\\
y_2(t) &= b_1 + b_2y_2(t-1) + b_3u_1(t-3)y_2(t-1) + b_4u_1(t-2)u_3(t-1) + b_5u_2(t-1)y_2(t-1)\\
y_3(t) &= c_1u_1(t-1) + c_2u_1(t-2) + c_3u_2(t-1) + c_4u_2(t-2) + c_5u_2(t-3)\\
&\quad + c_6u_2(t-4) + c_7u_3(t-1) + c_8u_3(t-2)
\end{aligned}$$

where u_1, u_2, u_3 denote desired torque, λ and engine speed and y_1, y_2 and y_3 denote INJ, SA and θ. The optimal data obtained in Section 7.5 are inverted and used as identification signals. The optimal output sequence is shifted 8 steps backwards since the maximum delay of the NN engine models and the polynomial controller are 5 and 3 samples respectively. From the PEM method the estimated parameters are obtained as:

$$\begin{aligned}
\hat{a}_{PEM} &= [1.61, 0.77, 0.54, -4.00 \times 10^{-5}, 0.0176, -0.0039] \qquad (7.19)\\
\hat{b}_{PEM} &= [2.38, 0.90, 2.85 \times 10^{-5}, -4.32 \times 10^{-6}, -0.0012]\\
\hat{c}_{PEM} &= [0.30, -0.14, 0.74, 0.31, 0.092, -0.37, 4.36 \times 10^{-5}, 0.0018]
\end{aligned}$$

And the corresponding R^2 values are:

$$R^2_{INJ} = 73.14\% \quad R^2_{SA} = 25.92\% \quad R^2_{\theta} = 38.89\% \tag{7.20}$$

Using the proposed SEM method, the parameters and improved R^2 are given by:

$$\begin{aligned} \hat{a}_{SEM} &= [0.30, 0.96, 0.0078, -1.04 \times 10^{-5}, 0.0018, -0.0033] \\ \hat{b}_{SEM} &= [5.22, 0.80, -1.68 \times 10^{-4}, -7.55 \times 10^{-6}, -0.0063] \\ \hat{c}_{SEM} &= [0.19, -0.031, 0.094, 0.15, 0.16, 0.11, -0.011, 0.013] \end{aligned} \tag{7.21}$$

$$R^2_{INJ} = 79.29\% \quad R^2_{SA} = 26.94\% \quad R^2_{\theta} = 40.26\% \tag{7.22}$$

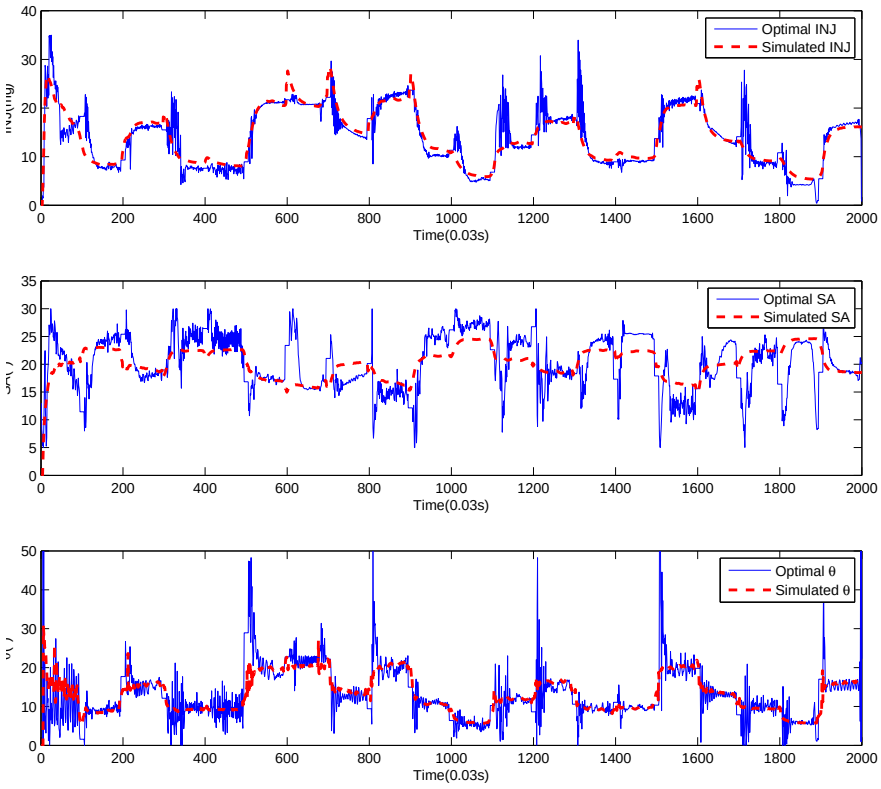


Figure 7.19: Optimal inputs and simulated optimal inputs by inverse models

Although the values of R^2 in equation (7.22) are not as high as the fitness of the identified engine model in Table 7.5, the simulated optimal inputs can still give satisfactory results. As displayed in Figure 7.19 when the control map is implemented, the spikes between segments are substantially reduced and the rest of the original optimal inputs are well matched in general. A satisfactory dynamic map which has the ability of tracking desired torque and λ and minimizing the fuel consumption is therefore composed of 3 dynamic models in equations (7.18) with estimated parameters in equations (7.21). Theoretically the difference between the original and simulated optimal inputs will undermine the control performance of the dynamic map, nonetheless the influence can be reduced by an additional closed loop control.

7.6.3 Offline Validation of Dynamic Map

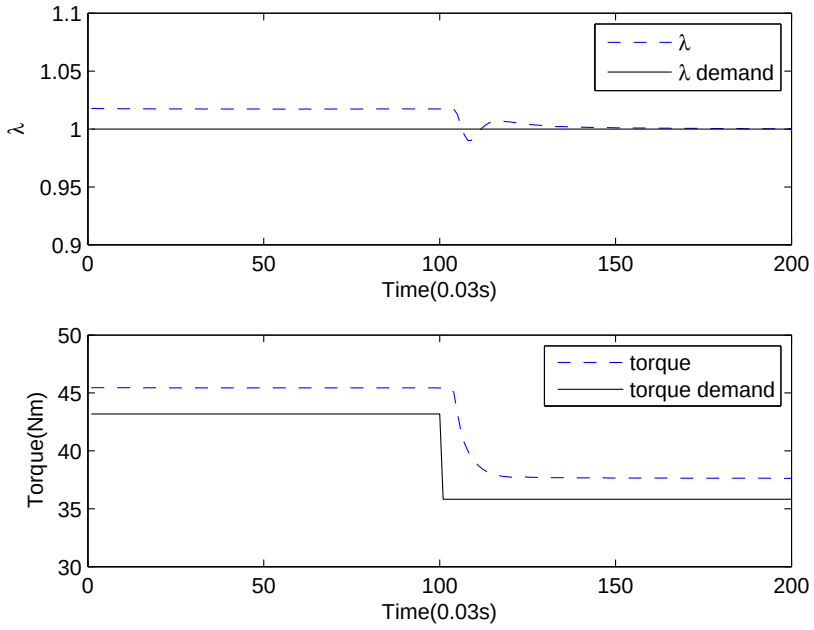


Figure 7.20: Control performance of dynamic map in offline validation

Since the numerical fuel optimization was carried out on the identified dynamic models rather than the RT model, the obtained dynamic map should be firstly validated on the engine models. Figure 7.20 shows the torque and λ response to a step change of demanded torque at the engine speed 2000 RPM. The settling time of torque and λ is less than one second so that the responses are as rapid as those controlled by the static map. However due to the quality of the inverse identified dynamic map, the demanded signals cannot be perfectly generated by this feedforward controller. The steady-state errors of torque and λ are approximately 5% and 2%.

7.6.4 Online Validation of Dynamic Map

The dynamic map is implemented on the virtual engine so as to observe its capability in satisfying the control objectives on the real system. Firstly a random number signal with a time interval of 6 sec is applied as the demanded torque to test the tracking of torque and the interaction with λ while the demanded λ remains stoichiometric. In real engine experiments, there is a limit on the maximum brake torque generated by the low-inertia dynamometer therefore may not be appropriate to apply step change of torque demand larger than $\pm$20Nm. The system behaviour excited by the limited step size of torque tends to be linear. Accordingly it is difficult to test the robustness of the controller against nonlinear

system dynamics excited by dramatic changes of torque. This issue should not be ignored since it is common in drive cycle tests. As a unique benefit of calibrating the virtual engine, it is possible to apply a simulated heavy load and hence the desired torque is selected with a relaxed amplitude constraint from 20Nm to 80Nm. The engine speed N is the other index of the operating region besides the torque. In the test the speed is generated by the vehicle-road-load submodel in order to evaluate the robustness of controllers to diverse torque and speed profiles. The outputs corresponding to the dynamic map and the static map are both plotted in Figure 7.21 for the ease of comparison and the validation signal is identical to the one used for static map online validation in Section 6.7 .

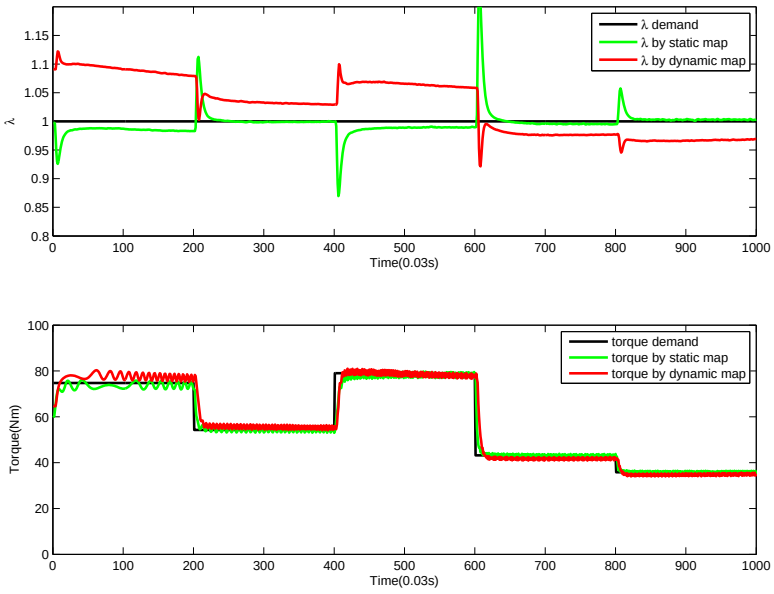


Figure 7.21: Control performance of the dynamic map and static map in online validation

According to equation (6.4), the resulting fuel economy from the dynamic map is: e_{dy}=4.15 Nm/mg, which is the same as the fuel economy from the static map. The tracking of desired torque and stoichiometric λ is displayed in Figure 7.21. Comparing to the static map, the overshoot of λ caused by the dynamic map is considerably shorter. The settling time of torque and λ responses by the dynamic map is as small as that by the static map however the dynamic map leads to larger steady state offsets. The steady-state offset of torque increases from 1Nm to 2Nm and the λ offset increases from 0.01 to 0.1 if the dynamic map is employed. The drawbacks on the control performance of the dynamic map result from the lack of accuracy of the identified engine models and also the inverse models used to obtain the dynamic map. Improving the model quality is therefore the primary solution of the lack of accuracy. Alternatively the offset error can also be reduced by the implementation of an

additional feedback control loop.

7.7 Design of Closed Loop Control

The major advantage of feedforward control is the rapid output response since the reference signal is not affected by the delays in measurement from the real system. However a closed loop control is often employed to reduce the steady-state offset between the desired and measured response. The following section discusses the design of such a closed loop control designed according to different requirements.

7.7.1 RT Model Feedback for Torque and λ Control

Although advanced methodologies for controller design have been proposed by many authors in recent decades, the PI controller is still one of the most widely used controllers in industry because of its simple structure and effectiveness [137]. As the remaining nonlinearity of the dynamic map feedforward compensated system, here composed of the feedforward controller in series with the virtual engine, can be characterised as linear uncertainty, a pair of PI controllers for engine torque and λ are designed according to the parameter-space method [138, 139] in this step. Initially, 5-level APRBS signals are implemented as 5 equally spaced set points of $T_{desired}$ and $\lambda_{desired}$ across the ranges [0.9 1.1] and [10Nm 90Nm]. The frequency responses in 4 channels: $T_{desired}$ to T, $T_{desired}$ to λ, $\lambda_{desired}$ to T and $\lambda_{desired}$ to λ are tested. To excite the dynamic map compensated system, firstly $\lambda_{desired}$ is fixed as a constant from [0.9, 0.95, 1, 1.05, 1] respectively and $T_{desired}$ is selected as an APRBS signal which generates the output response of $T_{desired}$ to T and $T_{desired}$ to λ. On the other hand $T_{desired}$ is fixed at [10Nm, 30Nm, 50Nm, 70Nm, 90Nm] respectively and $\lambda_{desired}$ is selected as an APRBS signal which generates the response of $\lambda_{desired}$ to T and $\lambda_{desired}$ to λ. 100 logarithmically distributed frequencies from 0.01 to 500 Hz are collected and the corresponding Bode plots are shown in Figure 7.22.

To simplify the control problem, the interaction between the λ channel and torque channel which should not be excited during normal stoichiometric operation is ignored in this thesis. The subplot of $T_{desired}$ to T indicates that a change of desired torque will lead to the same scale of change in the measured torque while the subplot of $\lambda_{desired}$ to λ shows that the output of λ is not sensitive to the change of desired λ. However because the desired λ remains stoichiometric in the calibration work, the insensitive $\lambda_{desired}$ to λ response is also of less importance.

Using the derived frequency responses, the parameter-space method produces the profiles of gain margins of 0dB, 6dB and 12dB and phase margins of 30°, 45° and 60° plotted in the

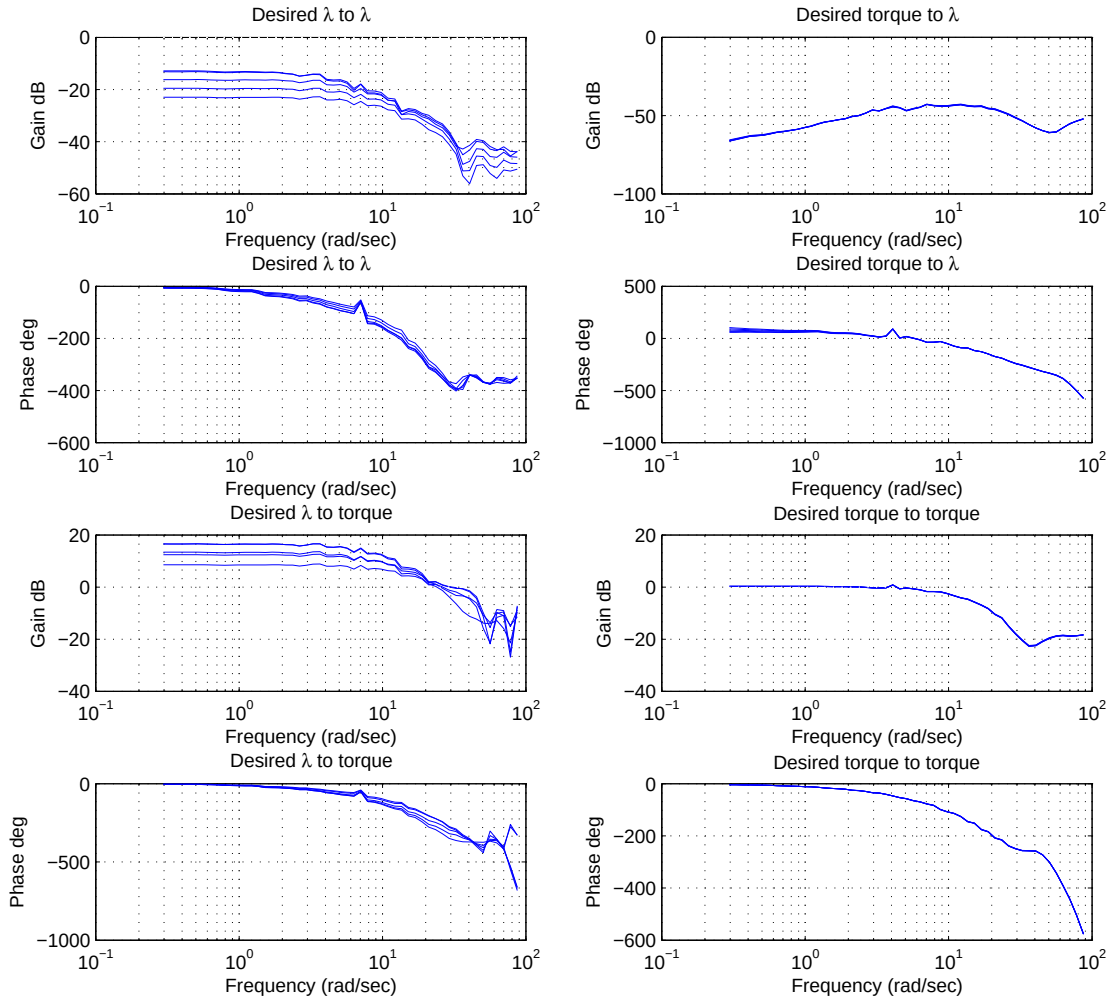


Figure 7.22: The Bode plot of frequency responses in 4 channels

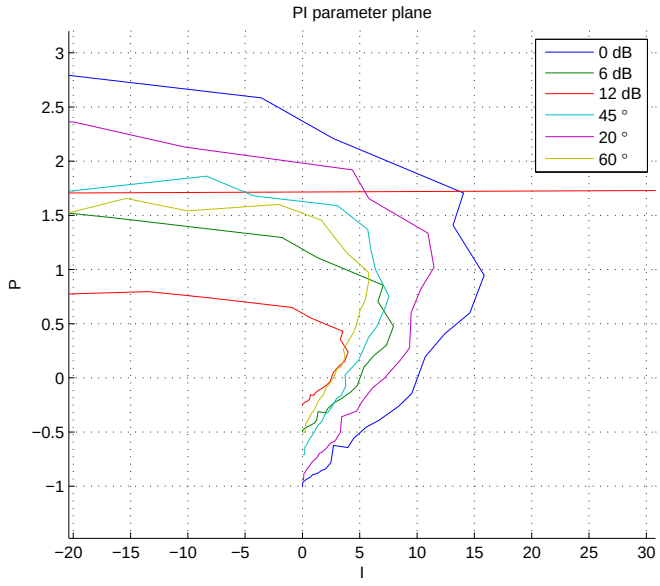


Figure 7.23: The parameter plane of P and I terms

K_p and K_I plane as shown in Figure 7.23. These profiles provide guides for selecting the values of K_p and K_I, yield the feedback controllers:

$$\begin{aligned} K_T &= \frac{2.64 + 0.33s}{s} \\ K_\lambda &= \frac{19.34 + 3.59s}{s} \end{aligned} \tag{7.23}$$

The capability of controllers to track demanded torque and λ is tested and presented in Figure 7.24. Comparing to Figure 7.21, it can be seen that the steady-state deviation is eliminated by the PI controllers. Nevertheless the corresponding outputs settle to the desired value in approximately 1.5 sec which is longer than when using the static map.

7.7.2 Open Loop Compensator for Torque control

As torque sensors are highly expensive, it is not reasonable to install these in a production car and so the implementation of closed loop torque control using feedback data from the measured engine torque becomes infeasible. For agility and steady-state accuracy in the torque control, an effective approach is for the ECU to provide online estimate of the engine torque using an open loop torque estimator and to use feedback of the simulated torque instead of the real engine torque in the closed loop control. Subsequently the resulting control efforts: SA, INJ, θ are then delivered to the inputs of the real engine and the inaccuracy of the pure feedforward torque control could be compensated. Figure 7.25 depicts a structure of the open loop compensator for such a torque control system.

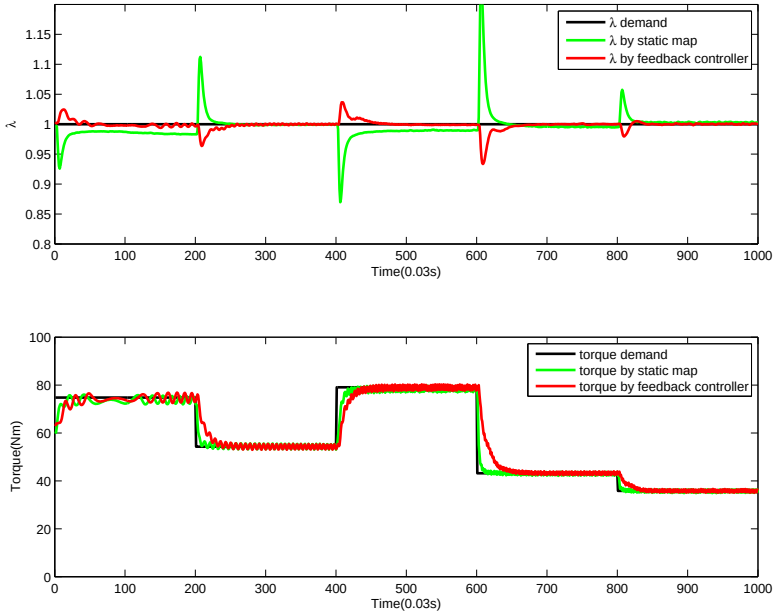


Figure 7.24: Closed loop control performance of PI controllers

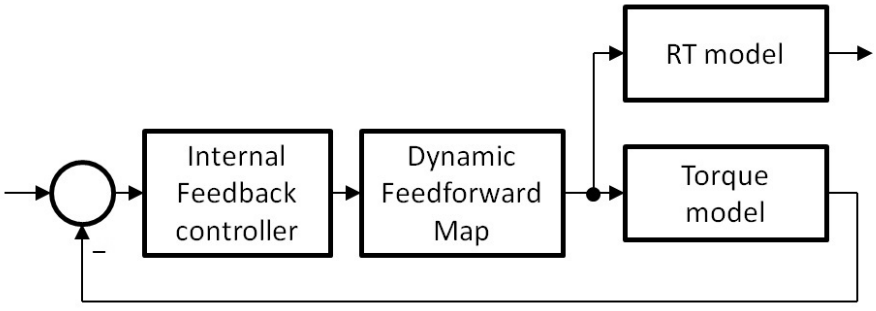


Figure 7.25: An open loop compensator for torque control

The type of model estimator has been extensively discussed by several authors. Static mean value models are chosen in [140, 141] because of their straightforward model structure and rapid output response. Recurrent NN models are selected in [142, 143] for their superior capability in modelling the nonlinearity of dynamic systems. In this chapter a nonlinear dynamic polynomial MISO model is used to implement the estimator. This type of model can represent a wide range of system dynamics by using nonlinear regressors with time delay and additionally its algebraic model structure is simple to program and implement in the ECU. Various engine signals can be used to build the estimator, generally including MAP, SA, N etc. [144]. For consistency with the engine models obtained in Section 7.4, INJ, SA, θ and N are employed.

In practical experiments significant time delays may exist between actuators and sensors due to the limits of the experimental conditions and cannot be physically removed. However, the estimator could still be used to improve the control. By removing the undesired time delay from the input-output data for the identification of the estimator, the obtained estimator is able to predict the output before the measurement from sensors is available. Providing that the output prediction is sufficiently accurate, a controller that is designed based on the estimator can lead to a more rapid output response [143].

Since the torque estimator is intended to replace the virtual engine in the closed loop control, the data for identification is collected from a test in the closed loop control system in Section 7.7.1. The identified polynomial model of the engine torque is given as:

$$\begin{aligned} y_T(t) &= \theta_1 + \theta_2 u_1(t-1) + \theta_3 u_1(t-2) + \theta_4 u_1(t-3) + \theta_5 u_2(t-1) \\ &\quad + \theta_6 u_3(t-1) + \theta_7 u_3(t-2) + \theta_8 y_T(t-1) \end{aligned} \tag{7.24}$$

where the parameters estimated by PEM are as follows:

$$\hat{\theta}_T = [-5.89, 5.67, -1.27, 2.25, -0.55, -1.79, 1.02, 0.89] \tag{7.25}$$

The model quality is validated and the resulting output fitness are: MSE=11.45 and R^2=98.87% By using the proposed methodologies of optimal input design and SEM, the estimated parameters are updated as

$$\hat{\theta}_{Topt} = [16.16, 7.99, -4.08, 4.12, -1.57, 0.53, -1.59, -0.45] \tag{7.26}$$

The output fitness of the updated model in validation is the found to be: MSE_{opt}=11.27 and R^2_{opt}=98.90%. The benefit by using DoE methodologies is limited in this identification because the accuracy of the model obtained by PEM is already very high and the system behavior has been precisely modelled.

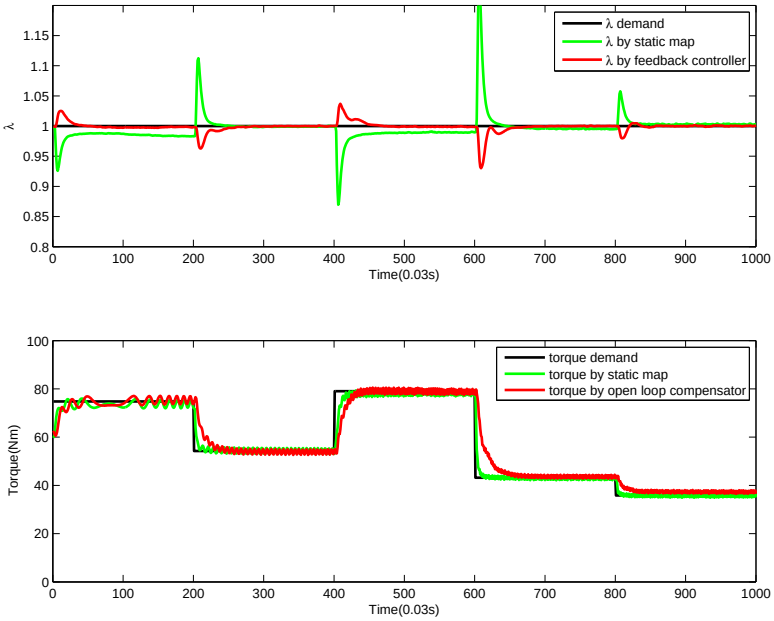


Figure 7.26: Closed loop control performance of an open loop compensator

Using the obtained torque model as the open loop compensator, the control performance is shown in Figure 7.26. Comparing with Figure 7.24, the control performance is seen to be very close except for a slight steady-state offset of torque, of approximately 1Nm, due to the difference between the model and real system.

7.7.3 Smith Predictor for λ Control

The control performance of λ in the virtual engine experiments is satisfactory since the feedback signal from the RT model is used as the reference signal for control. However in the real in-vehicle engine, two main practical issues need to be considered. Firstly as discussed in Section 7.6.1, there is a significant time delay of λ in real engine because of the location of the λ sensor and this delay will considerably compromise the control performance. Secondly the steady-state error is required to be small enough to achieve the strict requirement on emissions. An open loop compensator designed with time shifted data can solve the first issue. However since it is actually an open loop control on the λ of the real system, the high demand on steady-state accuracy is difficult to meet.

The control performance can be enhanced by iteratively refining the quality of the estimator. Alternatively a Simth predictor is also capable of improving the performance. To illustrate this for the linear case, assuming a system without extra output delay $G(z)$ and a feedback controller $K(z)$, the corresponding closed loop transfer function is thereby in the

form of:

$$H(z) = \frac{K(z)G(z)}{1 + K(z)G(z)} \tag{7.27}$$

Adding a pure k step time delay to the output, the transfer function of the system should be updated as $G(z)z^{-k}$. In order to obtain an updated closed loop transfer function $\bar{H}(z) = H(z)z^{-k}$, the controller $\bar{K}(z)$ which is named Smith predictor can be design as:

$$\frac{\bar{K}(z)G(z)z^{-k}}{1 + \bar{K}(z)G(z)z^{-k}} = z^{-k}\frac{K(z)G(z)}{1 + K(z)G(z)} \Rightarrow \bar{K}(z) = \frac{K(z)}{1 + K(z)G(z)(1 - z^{-k})} \tag{7.28}$$

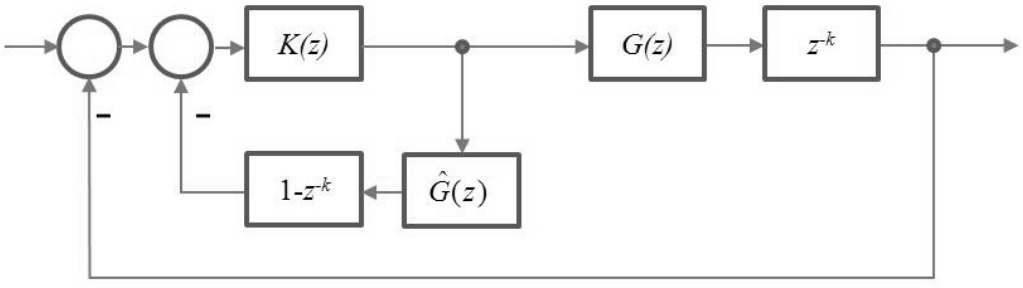


Figure 7.27: A Smith predictor for system with extra output time delay

Practically the real system $G(z)$ is often unknown so that an estimated $\hat{G}(z)$ needs to be used. The Smith predictor is a predictive controller with pure time delay as demonstrated in Figure 7.27. Without the predictor, the controller will regulate the system behaviour by using delayed output information hence the control performance may not be satisfactory. The estimator in the predictor is able to provide predicted output information which can enhance the control performance providing the estimator can represent the system behaviour precisely.

For λ control, the Smith predictor can be adapted as shown in Figure 7.28. The feedback signal from the real system is amended by both the delayed and the non delayed output of the λ estimator. It is shown that if the λ model perfectly matches the λ response of the virtual engine, the internal feedback controller designed with RT model feedback signal can be implemented directly without further tuning.

Adding an extra 5 step λ sensor delay to the RT output and using the same λ PI controller as in the last section, the output response in the extra delayed system is found to be as shown in Figure 7.29. Besides the 5 time delay, a large oscillation occurs which in turn affects the settling time and overshoot, the control performance is thus significantly lowered.

To reduce the influence of the extra output delay on the control performance, an estimator for λ is developed using the same process in Section 7.7.2 by using a time shifted

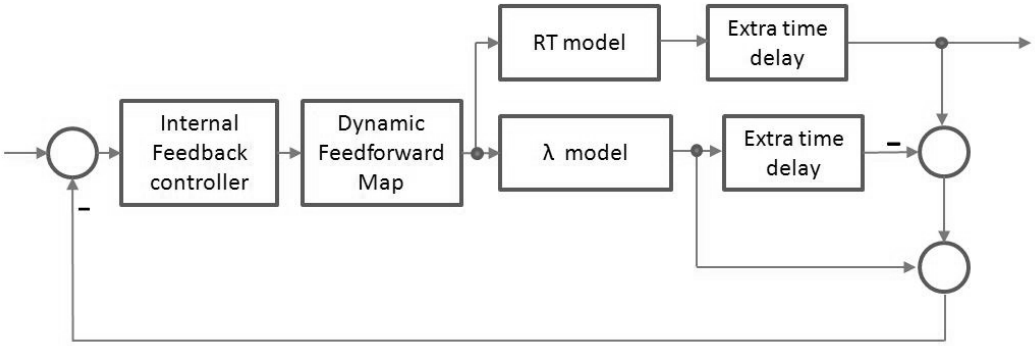


Figure 7.28: A Smith predictor for λ control

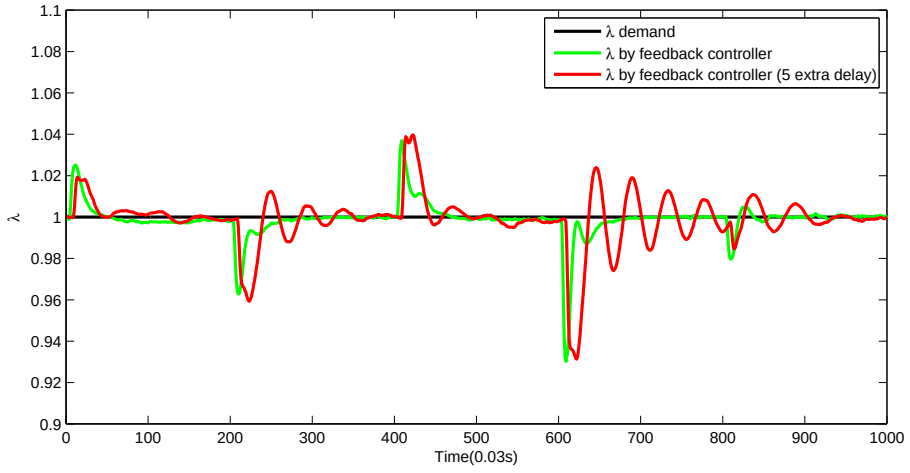


Figure 7.29: Closed loop control performance of PI controllers in delayed system

data. The resulting control performance of the Smith predictor is displayed in Figure 7.30. Compared with the λ response controlled by the feedback signal from the RT model without extra delay, the response in a delayed RT model which is controlled by the Smith predictor has a pure time delay of 5 samples but the main control performance such as the settling time and overshoot is not significantly affected. The benefit of the Smith predictor in the delayed system is thus exhibited. However the difference in shape between the two curves indicates that the accuracy of the estimator can be further improved and a better control performance could then be achieved.

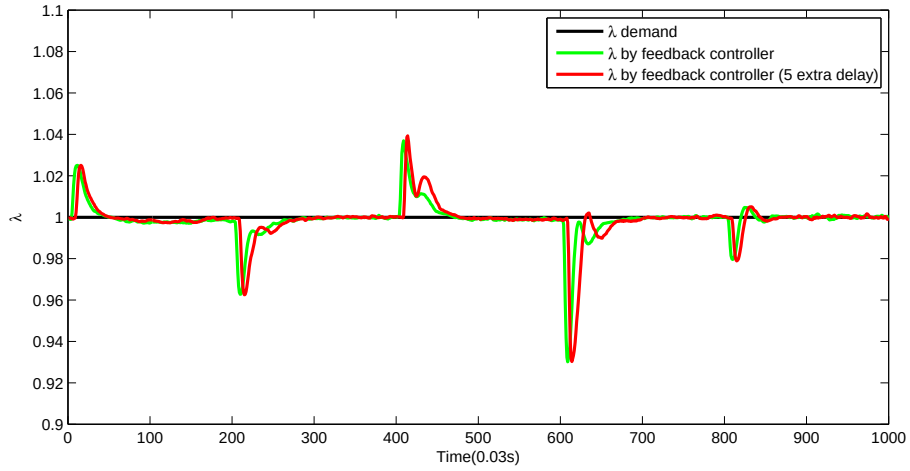


Figure 7.30: Closed loop control performance of Smith predictor in delayed system

7.8 Polynomial Model Based Design

7.8.1 Polynomial Model Based Fuel Optimization

The Neural Network models obtained in Section 7.4.2 were selected for use in the model-based optimization in the previous sections because of their high output fitness. However due to the relative complexity of the NN model structure compared with the simple polynomial models of Section 7.4.3, the simulation by the NN models required within the optimisation is much slower than that using the polynomial models though it is still considerably faster than the alternative of generating the output from the RT model on the real engine. Therefore providing sufficient accuracy can be obtained it is beneficial to replace the NN models by polynomial models in the optimization if a further improvement of computing efficiency is desired.

In the following experiment, the polynomial models of equation (7.8) are employed to simulate the torque and λ constraints while the other settings of the optimization such as the

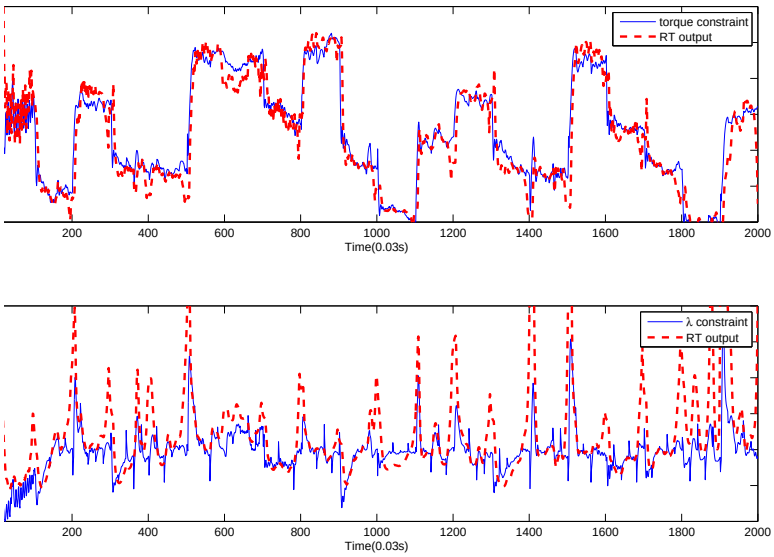


Figure 7.31: Optimal outputs on the RT model (segment length of 100 points)

objective function and the algorithm type remain the same. Figure 7.31 shows the resulting outputs on the RT model when the optimal signals were generated using the segment approach and the length of each segment is 100 points. As mentioned in Section 7.5.3, the spikes appear between segments so that in order to improve the output responses, another optimization was carried out using segments of 500 points. The corresponding output of the whole data length 2000 points is displayed in Figure 7.32. Compared with Figure 7.31, the output fitness is improved since the spikes are significantly reduced.

Table 7.8: The computing time of the model-based numerical optimization

	Segments of 100 points	Segments of 500 points
Neural Network	103340s	163927s
Polynomial model	1442s	1668s

The computing time required for the model-based numerical optimization over the entire data length of 2000 points with various lengths of segments is shown in Table 7.8. The computing time for the polynomial model based optimization is only approximately 1.4% of that for NN model based optimization. Although a longer segment length leads to a longer computing time, improving the output fitness by increasing the length of segments is still a sensible and practical approach because the cost of this is very much offset by the outstanding computing efficiency of the polynomial model based optimization.

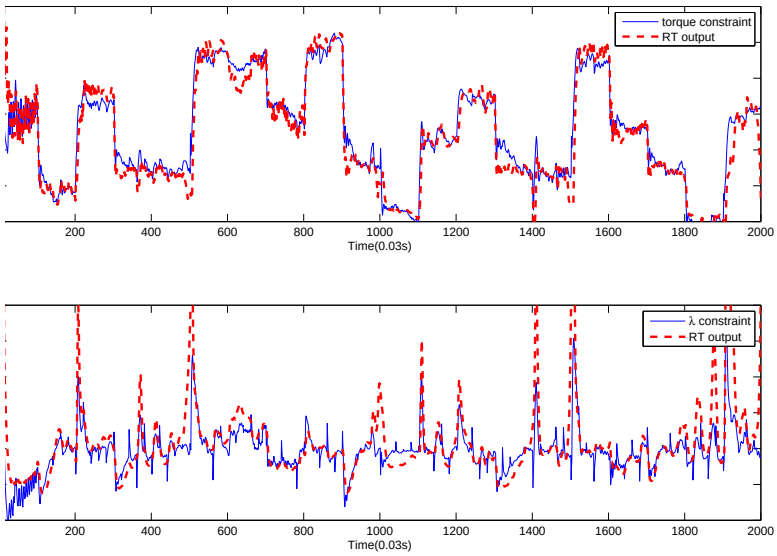


Figure 7.32: Optimal outputs on the RT model (segment length of 500 points)

7.8.2 Iterative Dynamic Map Design

Using the same model structure as in equations (7.18) and the optimal signals obtained in the polynomial model based fuel optimization with the segments of 500 points, the parameters estimated by the PEM method and the corresponding output fitness are given by:

$$\begin{aligned} \hat{a}_{PEM} &= [0.12, 0.0052, 1.03, 0.0043, -0.0054, -3.41 \times 10^{-5}] \\ \hat{b}_{PEM} &= [11.77, 0.66, -7.19 \times 10^{-4}, -2.64 \times 10^{-5}, -0.0057] \\ \hat{c}_{PEM} &= [0.21, -0.061, 9.60, -7.47, 7.70, -4.64, -0.0049, 0.0049] \end{aligned} \tag{7.29}$$

$$R^2_{INJ} = 85.11\% \quad R^2_{SA} = 69.69\% \quad R^2_{\theta} = 58.60\% \tag{7.30}$$

The parameters and improved R^2 of the proposed SEM method are given in equations (7.31) and (7.32). The optimal inputs and simulated optimal inputs from the dynamic map are shown in Figure 7.33. The parameters obtained by the SEM are:

$$\begin{aligned} \hat{a}_{SEM} &= [0.44, 0.0061, 1.08, 0.016, -0.099, -4.60 \times 10^{-5}] \\ \hat{b}_{SEM} &= [16.50, 0.50; -0.0014; -3.21 \times 10^{-5}; 0.0051] \\ \hat{c}_{SEM} &= [0.22, -0.062, 10.17, -10.62, 12.02, -6.44, -0.0058, 0.0059] \end{aligned} \tag{7.31}$$

$$R^2_{INJ} = 93.06\% \quad R^2_{SA} = 70.65\% \quad R^2_{\theta} = 58.66\% \tag{7.32}$$

Following the approach described in Section 7.7.1, a pair of PI controllers is obtained as shown in equation (7.33) and the output responses obtained by using these controllers are

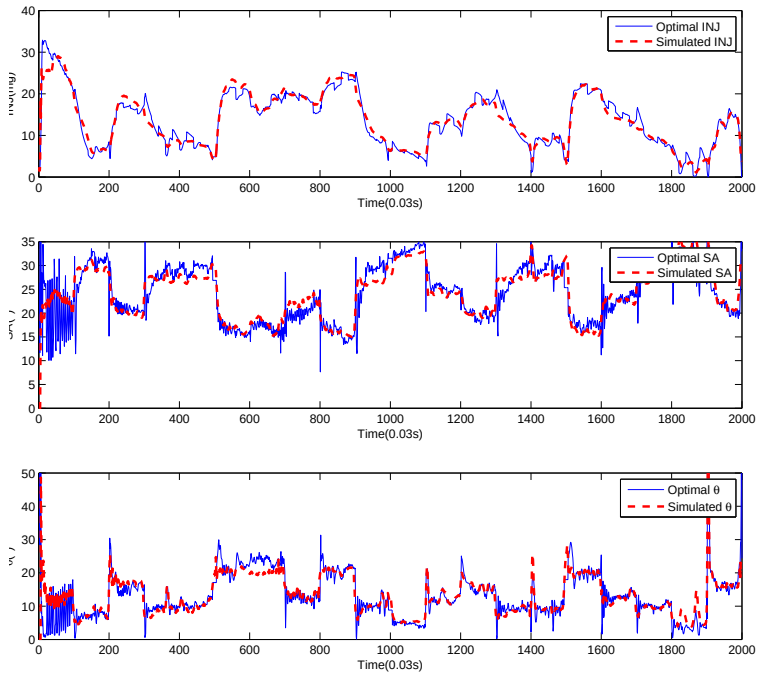


Figure 7.33: Optimal inputs and simulated optimal inputs by inverse models

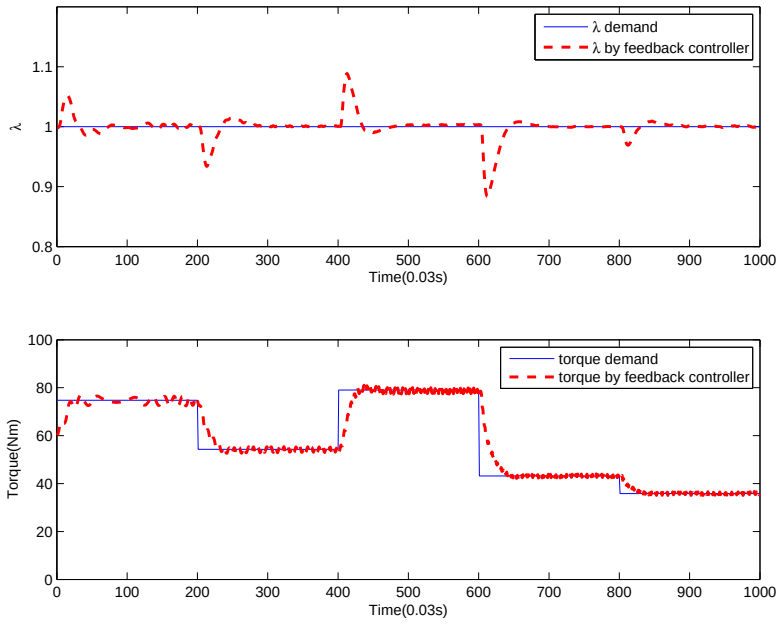


Figure 7.34: Closed loop control performance of PI controllers

shown in Figure 7.34.

$$\begin{aligned} K_T &= \frac{2.64 + 0.33s}{s} \\ K_\lambda &= \frac{2 + 0.5s}{s} \end{aligned} \tag{7.33}$$

Comparing to the feedforward controllers, the major disadvantage of the feedback controllers in this section and Section 7.7.1 is the resulting longer settling time. However we propose an approach to refine the dynamic map using the input-output data collected from the closed loop control system. Since the design of the dynamic map is based on experiments on the developed engine models, the control performance might be compromised when implementing the dynamic map on the virtual engine. Therefore it is sensible to further develop the dynamic map using the data collected in the closed loop control system because the data represents a typical optimal behaviour of the virtual engine.

Using the data from the closed loop control system and the same model structure, the parameters estimated by the SEM are updated as:

$$\begin{aligned} \hat{a}_{SEM} &= [2.83, 0.093, 1.39, 0.078, -1.23, -3.19 \times 10^{-5}] \\ \hat{b}_{SEM} &= [31.31, -0.0049, 0.0012, -8.79 \times 10^{-5}, 0.019] \\ \hat{c}_{SEM} &= [0.096, 0.043, -63.79, 62.99, 74.68, -75.69, 0.16, -0.15] \end{aligned} \tag{7.34}$$

Figure 7.35 shows the control performance of the dynamic map with the parameters as in equation 7.31 and the updated dynamic map with parameters as in equation (7.34). It is clearly illustrated that the static error of λ is reduced to less than 2% from 5% and the error in the torque is reduced to less than 0.5 % which is close to the control performance of the feedback controllers. The approach of refining the dynamic map is thus demonstrated to be effective.

7.9 Conclusions

A dynamic model-based calibration and inverse optimal behaviour based control methodology is presented in this chapter. Dynamic engine models of torque and λ are developed using the prior knowledge of the engine behaviour learnt from the static calibration. The use of NN models and polynomial models are discussed and proposed DoE and estimation methods are used for better model quality. The NN models are initially selected for their superior fitness for the torque model.

A constrained numerical optimization based on the developed engine models is employed to investigate the optimal fuel economy with specified torque demand and the requirement of stoichiometric air-fuel ratio. Assuming that the fuel is completely consumed in combustion,

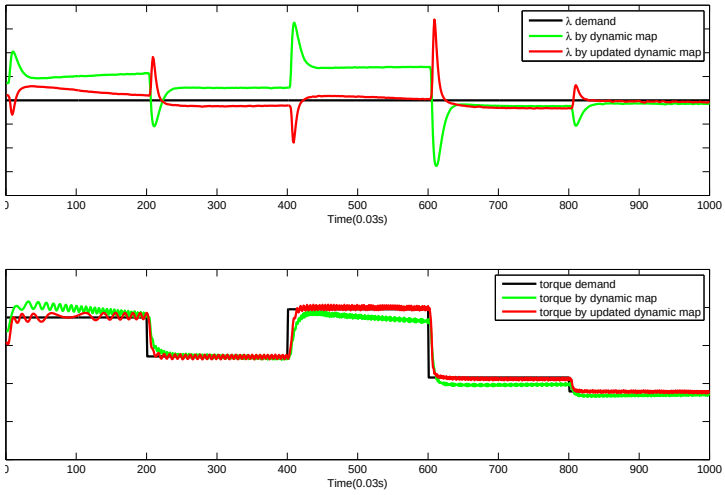


Figure 7.35: Control performance of dynamic maps in online validation

the objective function is selected as the mean of injected fuel mass over a period of time. The constraints are weighted according to their scales and experimental requirements. The segment method is selected to optimize the long data sequence because of its superior computational efficiency. The obtained optimal inputs are applied to the virtual engine with the purpose of validating the consistency between the output from the dynamic model and the RT model. Iterative model identification and fuel optimization, and additional input rate constraints can improve the consistency of output effectively.

The inverse optimal data is used to develop a feedforward dynamic control map. The time delay that is determined by the structure of the engine models and the resulting dynamic map is removed to obtain causality for the inverse identification. The dynamic models in the map are in the form of polynomial structures for the ease of programming in the ECU. The obtained dynamic map is capable of providing an optimized fuel consumption and rapid output response however steady-state offsets are observed in offline and online validation.

A closed loop control is designed to reduce the drawback of effects in the open loop dynamic map control. Simple PI controllers are developed using the frequency responses of torque and λ of the dynamic map feedforward compensated open loop control system and a parameter space design method is employed. The feasibility of direct feedback control in a practical engine implementation is considered. An open loop compensator for torque control and a Smith predictor for λ control are designed accordingly. The combined control system is capable of providing a similar fuel consumption and control performance to the static map.

The computing time of the polynomial model based fuel optimization is proved to be considerably shorter than that of the NN model based optimization. The approach of using

the data collected from the closed loop control system to refine the dynamic map is shown to be effective in improving the control performance of the dynamic map.

Chapter 8

Discussions and Conclusions

8.1 Discussions

In Chapter 4, 5 and 7, a new criterion for optimal input design, a simulation error method for parameter estimation and a dynamic model-based calibration approach are proposed respectively. Benefits of and critical reviews of the developed methods are summarized as follows:

Optimal input design

Since optimal inputs maximize the data information in the collected identification signals, models identified by optimal inputs are more accurate than those identified by non-optimal inputs as shown in Chapter 4. The proposed criterion for optimal input design is a simplification of the established I-optimal criterion. As the objective function is required calculated thousands, or even millions of times in any optimisation algorithm used for dynamic model optimization, the proposed criterion leads to a significant improvement in the computational efficiency of the dynamic modelling objective function. Moreover since in the new measure only terms which have little influence on the output prediction are removed from the computation, the new criterion is capable of improving the model accuracy to virtually the same degree as conventional optimal criteria.

In practical applications, any optimization should be constrained according to the experimental requirements. For instance, assuming the engine speed is one of the inputs, without proper input constraints the resulting optimal input for engine speed may change too quickly to be realized by a low-inertia dynomometer. Similarly the input signals may be too large and liable to cause damage to the test engine or test equipment. Secondly also for the parameter estimation based criteria, it is necessary to weight the diagonal elements in the parameter covariance matrix in order to reduce the influence of output sensitivities in the different scales

of the different dimensions of the signals. Since the output sensitivities are affected by the inputs, normalizing the scales of inputs before the optimization might be another effective approach of solving this problem and it will be tested in further research. Thirdly, the optimal input is found to be capable of improving the model accuracy however the degree of improvement varies in different cases. In systems with a strong disturbance, the improvement is relatively significant while this benefit decreases if the system disturbance is reduced. Therefore to model a system with very low uncertainty, it may not be necessary to design optimal inputs since the improvement on model accuracy would be limited. Nevertheless it is expected that in the physical testing required for engine calibration significant uncertainty would generally be present.

Simulation error based estimation method

A simulation error based estimation method is proposed to estimate parameters for use in the more difficult to establish simulation models (as opposed to on-line prediction models using measured output data). It is demonstrated that the resulting models are more accurate than those identified by prediction error methods such as ordinary least square method. Once established, simulation models are able to generate outputs without requiring the output data from the real system and are therefore favoured for model-based calibration.

In order to ensure the effectiveness of the model-based calibration, the requirement on model accuracy is usually very high. However as the purpose of the examples used in this chapter is just to illustrate the benefit of the SEM comparing to the PEM, some of the resulting models do not meet the requirement on model accuracy for industrial implementations, such as the torque model in Section 5.5. Nevertheless, it is expected that the model accuracy can be further improved to the required standard by other DoE methods such as by model structure selection techniques, but this aspect is not discussed in this thesis.

Dynamic model-based calibration

In Chapter 7, an approach to basic dynamic model-based calibration is demonstrated. The objective is to minimize the fuel consumption with constraints on engine torque and AFR. The engine behaviours are modelled by dynamic models in the form of NN or polynomial types and the obtained models are validated by other data sets in order to eliminate the influence of possible over/under-fitting. A constrained numerical optimization is conducted on the established models and the resulting optimal behaviour is used to design a feedforward controller by inverse identification. The design of an open loop compensator and Smith predictor is introduced in order to further improve the control performance. Comparing the hardware-based steady state calibration in Chapter 6 to the dynamic calibration, it is

shown that the dynamic calibration gives the same improvement on the fuel economy but the experimental time, as represented by the length of data used, is significantly reduced.

In recent decades, many methodologies for steady state model-based calibration, as illustrated in Figure 1.1, have been proposed. Such approaches model the engine behavior at representative operating points and then optimize the settings based on the obtained local models. Because adjacent models may differ significantly a data-smoothing of the resulting control map must be performed to give the smooth response required for good driveability. This smoothing may compromise the optimal steady-state performance. Nevertheless, since in these approaches, the optimisation is carried out on models, the corresponding experiment time can also be reduced. The performance of the dynamic and static model-based calibration methods should be compared in further research to analyse their relative advantages and disadvantages. Additional practical constraints, especially on emission levels, should be applied to the inputs of the numerical fuel optimisation and a more advanced optimization method should be invented to generate smooth optimal inputs. A better feedforward controller is desired in order to reproduce the optimal input-output behaviour obtained by the optimisation algorithm more precisely. The proposed dynamic calibration method should be tested in an aggressive drive cycle, such as the US06 drive cycle, to test its possible advantages in transients.

8.2 Conclusions

This thesis focuses on a development of optimal input design and estimation methods for popular polynomial and NN dynamic models. The optimal test signal and numerical simulation based estimation method are utilized in system identification in order to improve the quality of dynamic models. A dynamic model-based engine calibration and inverse optimal behaviour based control implementing these dynamic models is proposed. Related polynomial and NN model based methods to implement the control are investigated and refined methods proposed.

The conclusions drawn from each chapter in this work are as follows:

- In chapter 4, a general procedure of iterative optimal input design with practical constraints is presented and the influence of the optimal test signal on model estimation accuracy is compared with popular test signals currently used in industry. Signal tests are conduced on a nonlinear MISO polynomial engine torque model developed by experimental data from a 1.6L 4 cylinder SI Zetec Ford engine and the validation is carried out repeatedly for a convincing statistical result. The optimal input is designed by a constrained numerical optimization with the purpose of maximizing the data informa-

tion of the input. Since the Fisher information matrix is capable of measuring the data information, the objective function for optimization is selected as a scalar function of the matrix.

Using the objective function based on the covariance of estimated parameters, the A-optimal criterion is found to be unsuitable if the output sensitive terms are in significantly different scales while the D-optimal criterion and the proposed WA-optimal criteria are always effective in producing an estimate which is close to the true value. For the objective function based on the covariance of output prediction, the I-optimal, G-optimal and proposed AI-optimal criteria are examined. The AI-optimal criterion has a superior computational efficiency and leads to a model with an enhanced output fitness similar to the other criteria.

Various local and global optimization algorithms are discussed and the deterministic pattern search algorithm is selected due to the nonlinearity of the objective function and constraints. An optimal input is also designed with the additional practical constraints arising in experimental engine testing and it is demonstrated to be more useful for identifying systems with large disturbances. Moreover this methodology is shown to be effective in a black box identification of a 2.0L GTDI virtual engine and its potential in industrial practices is thus indicated. By a multi-variable optimization method, an optimal input can be designed to improve the model quality of a MIMO engine model and consequently time consuming tasks to design inputs for each sub-model can be avoided.

- In chapter 5, the differences between prediction models and simulation models are studied and a simulation error method is proposed to estimate parameters of simulation models. In prediction models, the predicted output is affected by both the input and the previous values of system output and the parameters can be estimated using a prediction error method. In the proposed PEM, an analytical solution for minimizing the prediction error is given. The system output is not only used as the reference signal to compute the prediction error but also contributes to the computation of the predicted output. Therefore the principle of this method is consistent to the prediction model.

  Simulation models only use the input to forecast the system output. These can be selected to describe dynamic engine models for offline calibration and controller design in which the plant should work independently of the real system. The PEM can be used to estimate parameters of simulation models at the expensive of compromising the minimization of the simulation error. The proposed simulation error method is developed as a numerical optimization of a selected objective function which is often a scalar function of the simulation error. The SEM is extremely advantageous if it is difficult to obtain an analytical solution of the scalar function. The linear search

method is employed to solve the unconstrained optimization for this method since it shows a higher computational efficiency than other algorithms. The estimation methods are validated by estimating parameters of an established model of the real engine and a black box model of the virtual engine. Compared to the PEM, the model identified by the SEM shows a better output fitness in MSE and R^2.

- In chapter 6, the objectives of steady-state "static" engine calibration are presented in the form of minimizing the fuel consumption and satisfying the constraints on torque and λ. A conventional static engine-based calibration is conducted to realize the objectives by the control of the engine parameters: injection flow, spark advance and throttle angle. The operating space is restricted to a low-speed low-load region for a simple demonstration of this methodology and local tests are carried out at each operating point according to desired engine torque and speed. In each test, the INJ and θ are controlled by feedback PI controllers to enable the engine to produce the stoichiometric air-fuel-ratio and desired torque. The SA is swept in a safe range to find the optimal parameter settings for fuel economy. The signals are applied for 10 sec to reach the steady-state values and therefore the calibration is a time-consuming task. A static control map is composed of optimal parameter settings at each operating point and is used as a feedforward controller by the EMS in a production vehicle. In the online validation, the static map is tested against random signals and it is proved to be able to provide satisfactory control performance in output tracking and minimization of fuel consumption. This map is used to evaluate the effectiveness of the dynamic maps and related compensators obtained in the novel process of the subsequent chapter by comparing the control performance.

- In chapter 7, a novel process of dynamic model-based calibration and inverse optimal behaviour based control is presented for the same control objectives used in that hardware-based static calibration. MISO dynamic models of the virtual engine torque and λ are identified in both the Neural Network and polynomial form. In the process of collecting signals for identification, feedback controllers are attached to the RT model in order to restrict the system outputs to the interested region. The methodologies of optimal test signal and SEM estimation are employed to further improve the quality of the engine models. NN models are initially selected for model-based experiments because of their higher accuracy and the main drawback of implementing NN models to the ECU is avoided since the engine models are only utilized in offline experiments.

  The optimal behaviour of minimized fuel consumption with constraints on torque and λ is determined by a numerical optimization on the NN engine models. The constraints on torque and λ are weighted since their scales are significantly different and the requirement of the stoichiometric λ is most crucial. For a superior computational

efficiency, the entire data sequence is optimized gradually in segments. The resulting optimal inputs are applied to the RT model to test the online performance and the consistency of the optimal outputs between the NN models and RT model are found to be improved by conducting the procedure of model identification and optimization iteratively or optimizing with additional input constraints.

The dynamic map is obtained by inverse identifications of the optimal data. The time delay is removed according to the selected model structure and the quality of the inverse polynomial models are enhanced by the SEM estimation. The dynamic map leads to the same fuel economy as the static map with a compromised static control performance caused by the loss of fitness in the inverse identification and the inconsistency between model and system. However the dynamic calibration process is advantageous because it requires significantly less expensive experimental data. The offset in the dynamic method can be largely reduced by an additional closed loop control. A open loop compensator is developed due to the high cost of implementing torque sensors on production engines and a Smith predictor is employed to reduce the influence of extra λ delay on control performance.

Chapter 9

Contributions and Future Work

9.1 Contributions

The novel contributions of this thesis are summarized in two major areas:

1. Development of methodologies for system identification

- A detailed procedure for constrained optimal input design for system identification is presented. Issues from the initial identification to the final application are discussed in-depth, including the selection of sub-optimal signal, optimization algorithms, practical constraints and objective function design according to the model type, etc.
- A novel weighted A-optimal design criterion for parameter estimation based optimal input design is developed. The major use for parameter estimation based input design is to identify white box models of which the structure is physically determined in advance. In this thesis conventional A-optimal and D-optimal criteria are applied to improve the quality of dynamic engine models by more accurate parameter estimation. The traditional A-optimum is found to be sensitive to the scales of input signals but this disadvantage can be overcome by the proposed weighted-A optimal criterion.
- A novel criterion adapted from I-optimal design for output prediction based optimal input design is presented and the selection of objective signal is studied. Conventional I-optimal and G-optimal are usually utilized in black box modelling in which the aim is to minimize the output prediction error. The proposed adapted-I optimum provides the same effect in optimization when the regressors are well chosen but with a considerably reduced experimental time. An approach to choose the objective signal used in the criterion to improve the global accuracy of identified models is presented.
- A novel application for optimal input design for MIMO systems is proposed. A reference

based method is used to adapt the input design to weighted optimization. An input signal which optimizes the development of two MISO models is generated and the experimental time for obtaining the required data and the accuracy in the resultant modelling is improved accordingly.

- A method of parameter estimation is developed to improve the estimation accuracy of simulation models. The method is adapted from the conventional ordinary least square method by replacing the output of the real system in regressors with those of a simulated output.

- An approach to statistical validation is utilized to evaluate the proposed methods. As statistical theories are fundamental in the methodologies for system identification, one single good or bad example can hardly prove the effectiveness of any of the proposed methods. Accordingly it is sensible to test any obtained models against a variety of signals since statistical assurance is required for the global model accuracy which is desired.

2. Dynamic calibration for multi-variable engine control

- A dynamic model-based calibration method originally proposed by Shenton [145] with the purpose of optimizing the fuel consumption and tracking the desired engine torque and λ is implemented in detail for the first time. This method proves to be a more time-efficient approach than conventional static calibration methods since the tests are carried out on dynamic models with a more limited amount of experimental data. The required feedforward controller for optimized fuel consumption is obtained by a novel approach of inverse causal identification. Feedback controllers are used to further reduce the stead-state offset of tracking. The methodology is validated on a Ford GTDI 2.0L virtual engine and the result of the control is compared favourably with that of a developed static map.

9.2 Recommendations of Future Work

The methodologies proposed in this thesis can be further developed in the following aspects:

Optimal input design for control

The direct objective of optimal input design in this thesis is to enhance the quality of the identified model. Assuming the model is used for control purposes, there is a strong connection between the accuracy of the model and the control performance on the system. The

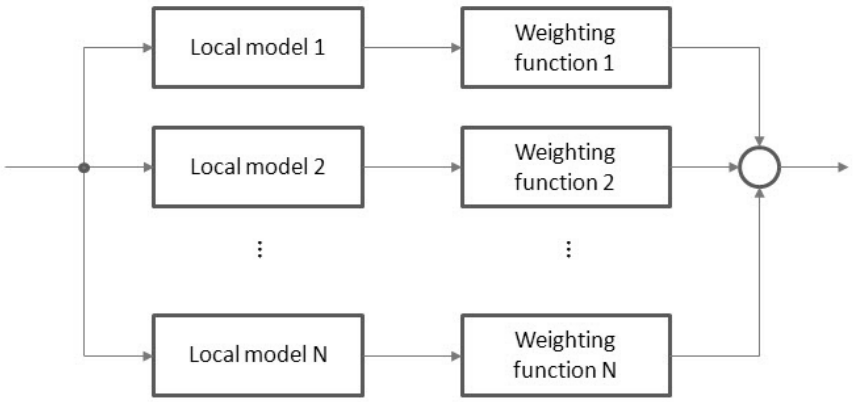


Figure 9.1: A schematic of multi-models

identified model represents the system behaviour with a region of uncertainty and the controller is designed to work stably in the uncertainty region [146]. The objective function of the optimization can be designed to measure the control performance therefore a controller designed using the estimated model is likely to achieve the desired control performance.

Design of multi-polynomial model

With the expansion of the operating region, a single model developed in a reduced region may not describe the system dynamics appropriately. As shown in Figure 9.1, a multi-model is composed of a series of local models and weighting functions that are selected according to the current operating point so that it is capable of representing the system accurately in a larger space by means of independently obtained local models. This approach is also recommended for the inverse identification of the dynamic map with the purpose of improving the model quality since it may also reflect the system nonlinearity more accurately than a single polynomial model.

Dynamic programming

The computational efficiency of numerical optimization in this work can be further improved by dynamic programming. Although the optimization of the entire sequence is solved in segments, each element in the segment is considered as an independent variable therefore the required computing time increases exponentially. The dynamic programming approach divides the optimization into subproblems and the solution of each subproblem is calculated and stored. If the same subproblem occurs in the process of optimizing, the solution can be directly loaded to reduce the computational burden [147]. Furthermore, the dynamic programming also correctly accounts for the controller causality.

Full operating region of production engines with turbocharger

The methodology developed in this thesis are recommended to be applied and validated in many other industrial applications. For further automotive applications, the calibration envelope should be expanded to a fully practical engine operating region from low-speed low-load to high-speed high-load. The control of waste gate of the virtual engine should be enabled to activate the turbocharger. Furthermore, the variable of inlet-outlet valves and EGR valve should also be considered as control inputs. Constraints on emissions over legislated drive cycles should be incorporated into the numerical optimisation. More system nonlinearities are expected in such an extended new application and further challenges to the modelling and control methodologies may be introduced accordingly. The whole approach should also be considered for applications to the diesel engine control.

References

[1] Advanced Engineering Systems Optimisation Research Group. *Automotive Research Centre*. the University of Bradford.

[2] L. Guzzella and C.H. Onder. *Introduction to Modelling and Control of Internal Combustion Engine Systems*. Springer-Verlag, 2004.

[3] U. Kiencke and L. Nielsen. *Automotive Control System for Engine, Driveline and Vehicle*. Springer, 2005.

[4] T. Dresner and P. Barkan. A review and classification of variable valve timing mechanisms. *SAE paper*, 890674, 1989.

[5] Honda Motor Co. Ltd.. *The VTEC Engine*, 1989.

[6] S.H. Park, J. Lee, J. Yoo, D. Kim, and K. Park. Effects of design and operating parameters on the static and dynamic performance of an electromagnetic valve actuator. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 217, 2003.

[7] J.J. Liu, Y.P. Yang, and J.H. Xu. Electromechanical valve actuator with hybrid mmf for camless engine. In *Proceedings of the 17th IFAC World Congress*, July 2008.

[8] S. Nagumo and S. Hara. Study of fuel economy improvement through control of intake valve closing timing: cause of combustion deterioration and improvement. *JSAE Review*, 16,1:13–19, 1995.

[9] E. Sher and T. Bar-Kohany. Optimization of variable valve timing for maximizing performance of an unthrottled si engine-a theoretical study. *Energy*, 27,8:757–775, 2002.

[10] H. Hong, G.B. Parvate-Patil, and B. Gordon. Review and analysis of variable valve timing strategies-eight ways to approach. *Proceedings of the Institution of Mechanical Engineers. Part D, Journal of automobile engineering*, 218:1179–1200, 2004.

[11] T. Ahmed and M.A.A. Theobald. A survey of variable-valveactuation technology. *SAE paper*, 891674, 1989.

[12] W.K. Cheng, D. Hamrin, J.B. Heywood, S. Hochgreb, K. Min, and M. Norris. An overview of hydrocarbon emissions mechanisms in spark-ignition engines. *SAE Technical Paper*, 9332708, 1993.

[13] F. Zhao, D.L. Harrington, and M.C. Lai. *Automotive Gasoline Direct-injection Engines*. SAE International, 2002.

[14] A. Hiromitsu. Combustion control for mitsubishi gdi engine. In *Proceedings of the 2nd International Workshop on Advanced Spray Combustion*, pages 225–235, November 1998.

[15] K. Kazunari and A. Hiromitsu. Control of mixing and combustion for mitsubishi gdi engine. *Proc. JSAE Annual Congress*, 984:35–38, 1998.

[16] J.I. Bae and S.C. Bae. A study on the engine downsizing using mechanical supercharger. *Journal of Mechanical Science and Technology*, 19(12):2321–2329, 2005.

[17] Volkswagen Group. 1.4l tsi engine with dual-charging. *Self-study Programme 359*, 2006.

[18] E. Rask and M. Sellnau. Simulation-based engine calibration: Tools, techniques, and applications. In *Proceedings of the 2004 SAE World Congress*, March 2004.

[19] Mathwork. Model-based calibration toolbox. *Matlab 2010b*, 2010.

[20] M. Karlsson, K. Ekholm, P. Strandh, R. Johansson, and P. Tunestal. Dynamic mapping of diesel engine through system identification. In *Proceedings of American Control Conference*, pages 3015–3020, June 2010.

[21] B. Fabien. *Analytical System Dynamics : Modelling and Simulation*. New York : Springer Science+Business Media, 2009.

[22] N. U. Ahmed. *Dynamic Systems and Control with Applications*. Hackensack, NJ : World Scientific, 2006.

[23] F. Haugen. *Dynamic Systems : Modelling, Analysis and Simulation*. Trondheim : Tapir Academic, 2004.

[24] K. Ropke (ed.). *Design of Experiments (DoE) in Engine Development V*. Expert-Verlag, 2011.

[25] R. Jurgen. *Automotive Electronics Handbook*. McGraw-Hill, 1995.

[26] R. Stone. *Introduction to Internal Combustion Engines Second Edition.* SAE International, 1992.

[27] J.B. Heywood. *Internal Combustion Engine Fundamentals.* McGraw-Hill, 1988.

[28] Mathwork. Simulating and predicting model output. *System Identification Toolbox*, 2012.

[29] E.Hendricks and S.C. Sorenson. Mean value modelling of spark ignition engines. *SAE Technical Papers*, 900616, 1990.

[30] S.H. Chan and J. Zhu. Modelling of engine in-cylinder thermodynamics under high values of ignition retard. *International Journal of Thermal Sciences*, 40:94–103, 2001.

[31] B.C. Chen, Y.Y. Wu, and F.C. Hsieh. Estimation of engine rotational dynamics using kalman filter based on a kinematic model. *IEEE Transactions on Vehicular Technology*, 59:3728–3735, 2010.

[32] J. Sjoeberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31:1691–1724, 1995.

[33] T.Söderstöm and P. Stoica. *System Identification.* Prentice Hall, 1989.

[34] L. Ljung. *System Identification: Theory for the User second edition.* Prentice Hall, 1999.

[35] K.J. Keesman. *System Identification: An Introducation.* Springer, 2011.

[36] I.J. Leontaritis and S.A. Billings. Experimental design and identifiability for non-linear system. *International Journal of Systems Science*, 18, 1987.

[37] F.Y. Edgeworth. On the probable errors of frequency-constants. *Journal of the Royal Statistical Society*, 71:499–512, 1908.

[38] V. Klein and E.A. Morelli. *Aircraft System Identification: Theory and Practice.* American Institute of Aeronautics and Astronautics, 2006.

[39] H. Cramér. *Mathematical Methods of Statistics.* Princeton Univ. Press, 1946.

[40] C.R. Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–89, 1945.

[41] C.G. Goodwin and L.R. Payne. *Dynamic System Identification: Experiment Design and Data Analysis.* Academic press, 1977.

[42] A.C. Atkinson. *Optimum Experimental Designs, with SAS*. Oxford University Press, 2007.

[43] G.C. Goodwin. Optimal input signal for nonlinear-system identification. *Proc. IEE*, 118(7):922–926, 1971.

[44] R.K. Mehra. Optimal inputs for linear system identification. *IEEE Transactions on Automatic Control*, 19(3):192–200, 1974.

[45] R.E. Kalaba and K. Spingarn. Optimal input system identification for nonlinear dynamic systems. *Journal of Optimization Theory and Applications*, 21:91–102, 1977.

[46] R.E. Kalaba and K. Spingarn. *Control Identification and Input Optimization*. Plenum Press, 1982.

[47] M.A. Lejeune. Heuristic optimization of experimental designs. *European Journal of Operational Research*, 147:484–498, 2003.

[48] C.R. Reeves and C.C. Wright. An experimental design perspective on genetic algorithms. In *in Proceedings of Foundations of Genetic Algorithms 3*, 1995.

[49] H. Lan, X. Wang, and L. Wang. Improved genetic-annealing algorithm for global optimization of complex functions. *Journal of Tsinghua University (Science and Technology)*, 42(9):1237–1240, 2002.

[50] A. Amirjanov. A changing range genetic algorithm. *International Journal for Numerical Methods of Engineering*, 61(15):2660–2674, 2004.

[51] X. Hao and W. Pu. An improved genetic algorithm for solving simulation optimization problems. *International Journal of Physical Science*, 6(10):2399–2404, 2011.

[52] M. Aoki and R.M. Staley. On input signal synthesis in parameter identification. *Automatica*, 6:431–440, 1970.

[53] E. Nahi and G.A. Napjus. Design of optimal probing signals for vector parameter estimation. In *Proceedings of IEEE Conference on Decision and Control*, 1971.

[54] B. Heiligers. E-optimal designs in weighted polynomial regression. *The Annals of Statistics*, 22:917–929, 1994.

[55] R.K. Mehra. Optimal input signals for parameter estimation in dynamic systems-survey and new results. *IEEE Transactions on Automatic Control*, AC-19(6):753–768, 1974.

[56] S. Zaglauer and M. Delflorian. Bayesian d-optimal design. In *Proceedings of the 6th Conference Design of Experiments in Engine Development*, 2011.

[57] A. Schreiber, M. Kowalczyk, and R. Isermann. Method for dynamic online identification with integrated determination of operating boundaries. In *Proceedings of the 6th Conference Design of Experiments in Engine Development*, 2011.

[58] W.K. Wong and R.D. Cook. Heteroscedastic g-optimal design. *Journal of the Royal Statistical Society. Series B*, 55(4):871–880, 1993.

[59] E. Lizama and D. Surdilovic. Designing g-optimal experiments for robot dynamics identification. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1996.

[60] W.D.K. Kapelle. Using i-optimal designs for narrower confidence limits. In *Proceedings of Intra-Americas Sea Initiative Research Planning Meeting*, 1998.

[61] L.K. Debusho and L.M. Haines. D- and v-optimal population designs for the quadratic regression model with a random intercept term. *Journal of Statistical Planning and Inference*, 141:889–898, 2011.

[62] T.S. Ng, G.C. Goodwin, and R.L. Payne. On maximal accuracy estimation with output power constraints. *IEEE Transactions on Automatic Control*, 22(1):133–134, 1977.

[63] T.S. Ng, Z.H. Qureshii, and Y.C. Cheah. Optimal input design for an ar model with output constraints. *Automatica*, 20(3):359–363, 1984.

[64] U. Forssell and L. Ljung. Identification for control: some results on optimal experiment design. In *Proceedings of the 37th IEEE Conference on Desision and Control*, 1998.

[65] E.A. Morelli and V. Klein. Optimal input design for aircraft parameter estimation using dynamic programming principles. In *Proceedings of AIAA Atmospheric Flight Mechanics Conference*, August 1990.

[66] E.A. Morelli. Flight test validation of optimal input design using pilot implementation. In *Proceedings of the 10th IFAC Symposium on System Identification*, July 1994.

[67] E.A. Morelli. Multiple input design for real-time parameter estimation in the frequency domain. In *Proceedings of the 13th IFAC Symposium on System Identification*, August 2003.

[68] A. Isaksson. Identification of arx models subject to missing data. *IEEE Transactions on Automatic Control*, 38:813–819, 1993.

[69] G.E.P. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco, 1970.

[70] I.J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47, 2003.

[71] D. Kristensen and Y. Shin. Estimation of dynamic models with nonparametric simulated maximum likelihood. *Journal of Econometrics*, 167, 2012.

[72] F. Hayashi. *Econometrics*. Princeton University Press, 2000.

[73] K. Miura. An introduction to maximum likelihood estimation and information geometry. *Interdisciplinary Information Sciences*, 17, 2012.

[74] NIST/SEMATECH. E-handbook of statistical methods. *http://www.itl.nist.gov/div898/handbook/*.

[75] Mathwork. Identifying input-output polynomial models. *System Identification Toolbox*, 2012.

[76] R. Haber and H. Unbehauen. Structure identification of nonlinear dynamics systems-a survey on input/output approaches. *Automatica*, 26(4):651–677, 1990.

[77] Y. Tan and M. Saif. Nonlinear dynamic modelling of automotive engines using neural networks. In *Proceedings of the IEEE International Control Conference on Control Application*, pages 408–410, 1997.

[78] Z. Hou, Q. Sen, and Y. Wu. Air-fuel ratio identification of gasoline engine during transient conditions based on elman neural networks. *IEEE International Conference on Intelligent Systems and Design Applications*, (0-7695-2528-8/06), 2006.

[79] G. V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals*, 2(4):303–314, 1989.

[80] K.L. Priddy and P.E. Keller. *Artificial Neural Networks: An Introduction*. Bellingham: SPIE, 2005.

[81] H. Demuth, M. Beale, and M. Hagan. *Matlab Neural Network Toolbox User's Guide*. The Mathworks Inc, 2006.

[82] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[83] G. A. Carpenter. Distributed learning, recognition, and prediction by art and artmap neural networks. *Neural Networks*, 10(8):1473–1494, 1997.

[84] I.V. Tetko, D.J. Livingstone, and A.I. Luik. Neural network studies. 1. comaprsion of overfitting and overtraining. *J. Chem. Inf. Comput. Sci.*, 35, 1995.

[85] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[86] Ricardo Software. Sub-models: Si wiebe combustion. *Help Documents*, 2012.

[87] S.D. Angelo and R.D. Gafford. Feed-forward dynamometer controller for high speed inertia simulation. *SAE*, 1981.

[88] O. Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer-Verlag Berlin Heidelberg, 2001.

[89] J.J. More and D.C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 3, 1983.

[90] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, 1987.

[91] R.H. Byrd, M.E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9, 1999.

[92] V.J. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 9, 1997.

[93] M.D. Vose. *The Simple Genetic Algorithm : Foundations and Theory*. MIT Press, 1998.

[94] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[95] R. Azencott. *Simulated Annealing: Parallelization Techniques*. Wiley:New York, 1992.

[96] G.C. Goodwin. *Optimal Inputs and Iterative Methods for Nonlinear System Identification*. PhD Thesis, The University of New South Wales, 1970.

[97] E. Plaetschke and G. Schulz. Practical input signal design. *AGARD-LS-104*, paper, 3, 2011.

[98] E. Plaetschke J.A. Mulder and J.H. Breeman. Flight test results of five input signals for aircraft parameter identification. In *Proceedings of the 6th IFAC Symposium on Identification and System Parameter Estimation*, pages 1149–1154, 1982.

[99] C.A.A.M. van der Linden, J.A. Mulder, and J.K. Sridhar. Recent developments in aircraft parameter identification at delft university of technology - optimal input design. *Aerospace Vehicle Dynamics and Control*, 1, 2001.

[100] C. Jauberthie, F. Bournonville, P. Conton, and F. Rendell. On the convergence of pattern search algorithms. *Aerospace Science and Technology*, 10, 2006.

[101] W.K. Wong. On the equivalence of d and g-optimal designs in heteroscedastic models. *Statistics and Probability Letters*, 25:317–321, 1995.

[102] L. Guzzella and C. Onder. Past, present and future of automotive control. *Control of Uncertain Systems*, 329:163–182, 2006.

[103] J. Cook, J. Sun, J. Buckland, I. Kolmanovsky, H. Peng, and J. Grizzle. Automotive powertrain control: a survey. *Asian Journal of Control*, 8(3):237–260, 2005.

[104] J.W. Helton and O. Merino. *Classical Control Using H-Infinity Methods*. SIAM, 1987.

[105] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley and Sons, 2001.

[106] M. Suzuki, M. Hori, and M. Terashima. Decoupling torque control system for automotive engine tester. *IEEE Transactions on Industry application*, 36:467–474, 2000.

[107] S. Zhao, P.B. Dickinson, and A.T. Shenton. Decoupled torque-afr control by frobenius h-infinity feedback and a lambda estimator. In *United Kingdom Automatic Control Conference*, pages 1–6, 2010.

[108] A. Malikopoulos. *Real-time, Self-learning Identification and Stochastic Optimal Control of Advanced Powertrain Systems*. ProQuest LLC, 2008.

[109] J. Sun, I. Kolmanovsky, J.A. Cook, and J.H. Buckland. Modelling and control of automotive powertrain systems: a tutorial. In *Proceedings of American Control Conference*, pages 3271–3283, 2005.

[110] S. Saraswati. Reconstruction of cylinder pressure for si engine using recurrent neural network. *Journal of Neural Computing and Applications*, 19:935–944, 2010.

[111] Y. Xia, G. Hao, C. Shan, Z. Ni, and W. Zhang. Reconstruction of cylinder pressure of i.c. engine based on neural networks. In *Proceedings of the 1st International Conference on Pervasive Computing, Signal Processing and Applications*, pages 924–927, September 2010.

[112] Y. Zhang, L. Xi, and J. Liu. Transient air-fuel ratio estimations in spark ignition engine using recurrent neural networks. *KES 2007/WIRN 2007, Part II*, pages 240–246, 2007.

[113] I. Arsie, M.M. Marotta, C. Pianese, and M. Sorrentino. Experimental validation of a recurrent neural newtwork for air-fuel ratio dynamic simulation in si ic engines. In *Proceedings of ASME International Mechanical Engineering Congress and Exposition*, pages 127–136, November 2004.

[114] G.C. Luh and C.Y. Wu. Inversion control of non-linear systems with an inverse narx model identified using genetic algorithms. In *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 214, pages 259–271, June 2000.

[115] G. Zito and I.D. Landau. Narmax model identification of a variable geometry turbocharged diesel engine. In *Proceedings of the American Control Conference*, June 2005.

[116] S.A. Billings and S. Chen. The identification of linear and non-linear models of a turbocharged automotive disel engine. *Mechanical Systems and Signal Processing*, 3(2):123–142, 1989.

[117] Z. Li and A.T. Shenton. Nonlinear model structure identification of engine torque and air fuel ratio. In *Proceedings of 6th IFAC Symposium Advances in Automotive Control*, pages 1–6, 2010.

[118] M. Hirsch and L.D. Re. Sequential identification of engine subsystems by optimal input design. *SAE International Journal of Engines*, 2(2):562–569, 2010.

[119] Q.R. Butt. Estimation of gasoline-engine parameters using higher order sliding mode. *IEEE Transactions on Industrial Electronics*, 55:3891–3898, 2008.

[120] F. Hover. Inversion of a distributed system for open-loop trajectory following. *International Journal of Control*, 60:671–686, 1994.

[121] D. Enns, D. Bugajski, R. Hendrick, and G. Stein. Dynamic inversion: an evolving methodology for flight control design. *International Journal of Control*, 59:71–91, 1994.

[122] A.P. Petridis and A.T. Shenton. Non-linear inverse compensation of an si engine by system identification for robust performance control. *Inverse Problems in Engineering*, 8:163–176, 2000.

[123] A.T. Shenton and A.P. Petridis. Nonlinear miso direct-inverse compensation for robust performance speed control of an si engine. *Nonlinear and Adaptive Control*, 281:337–349, 2003.

[124] Q.Y. Du, J.M. Ni, M. Chen, X.M. Zhang, and Y.S. Ping. The application of doe in engine design. In *Proceedings of the 6th Conference on Design of Experiments in Engine Development*, pages 136–153, 2011.

[125] S. Gaikwad and D. Rivera. Control-relevant input signal design for multivariable system identification: application to high-purity distillation. In *Proceedings of the 13th IFAC World Congress*, pages 349–354, 1996.

[126] R.L. Harvey. *Neural Network Principles*. Prentice-Hall, 1994.

[127] K.S. Narendra and K. Parthasarathy. Learning automata approach to hierarchical multiobjective analysis. *IEEE Transactions on Systems, Man and Cybernetics*, 20(1):263C272, 1991.

[128] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2, 1944.

[129] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11, 1963.

[130] H.L. Wei, S. Billings, and J. Liu. Term and variable selection for non-linear system identification. *International Journal of Control*, 77(1):86–100, 2004.

[131] R. Pearson. *Discrete-Time Dynamic Models*. Oxford University Press, 1999.

[132] Mathwork. Optimization toolbox. *Matlab 2010b*, 2010.

[133] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 2002.

[134] K. Ropke, W. Baumann, B.U. Kohler, S. Schaum, R. Lange, and M. Knaak. Engine calibration using nonlinear dynamic modeling. *Identification for Automotive Systems*, 418:165–182, 2012.

[135] F. Liu, G. Amaratunga, N. Collings, and A. Soliman. An experimental study on engine dynamics model based in-cylinder pressure estimation. *SAE Technical Paper*, 2012-01-0896, 2012.

[136] C.F. Aquino. Transient a/f control characteristics of the 5 liter central fuel injection engine. *SAE Technical Paper*, (810494), 1981.

[137] T. Glad and L. Ljung. *Control Theory : Multivariable and Nonlinear Methods*. London: Taylor and Francis, 2000.

[138] V. Besson. *Parameter Space Robust Control for SI Engine Idel Speed*. PhD Thesis, The University of Liverpool, UK, 1998.

[139] J. Ackermann. *Robust Control: The Parameter Space Approach*. Springer, 2002.

[140] A.G. Stefanopoulou, J.W. Grizzle, and J.S. Freudenberg. Engine air-fuel ratio and torque control using secondary throttles. In *Proceedings of Conference on Decision and Control*, pages 2748–2753, 1994.

[141] E. Hendricks. Isothermal vs. adiabatic mean value si engine models. In *Proceedings of IFAC Workshop on Advances in Automotive Control*, pages 363–368, 2001.

[142] N. Rivara. *IC Engine Control by Ionization Current Sensing*. PhD Thesis, The University of Liverpool, UK, 2009.

[143] S. Zhao. *Nonparametric Robust Control Methods for Powertrain Control*. PhD Thesis, The University of Liverpool, UK, 2011.

[144] D. Khiar, J. Lauber, T. Floquet, G. Colinc, T.M. Guerra, and Y. Chamaillard. Robust takagi-sugeno fuzzy control of a spark ignition engine. *Control Engineering Practice*, 15:1446–1456, 2007.

[145] A.T. Shenton. Dynamical calibration. *University of Liverpool, Centre for Engineering Dynamics, Internal Report MES/ATS/INT/110*, 2011.

[146] R. Hildebrand and M. Gevers. Identification for control: optimal input design with respect to a worst-case -gap cost function. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 996–1001, 2003.

[147] R.D. Robinett. *Applied Dynamic Programming for Optimization of Dynamical Systems*. Philadelphia : Society for Industrial and Applied Mathematics, 2005.