UNIVERSITY OF
LIVERPOOL

Nonparametric Robust Control Methods for Powertrain Control

Thesis submitted in accordance with the requirements of the

University *of* Liverpool for the

Degree of Doctor in Philosophy

by

Shiyu Zhao

January 2011

Acknowledgments

I would firstly like to thank Tom Shenton specifically for his guidance and support during this work. I can't make this work possible without his supervision and help. I would also like to thank all of the other PhD students: Paul, Jack, Chris, Nick, Ahmed, Ke, Zongyan and Ming-Yen for their helpful discussions. Particularly I'd like to thank Derek Neary for his technical assistance with the experimental setup.

Secondly, I would like to thank my Mum and Dad, grandparents and maternal grandparents, my whole family and anyone else I have forgotten for their continued support and encouragement.

Finally, I'd like to thank my loving wife Zhe (Isabel) for all her devotion and support.

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 2006 and October 2009.

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:

- S. Zhao, A. Abass and A.T. Shenton, Nonparametric design of robust linear controllers and their experimental application to idle control, 11th IASTED International Conference on Control and Applications, Cambridge, July 2009
- A. Abass, S. Zhao and A.T. Shenton, Nonparametric Driveline Identification and Control, First International Conference on Intelligent Systems, Modelling and Simulation, ISMS 2010, Liverpool, January 2010.
- S. Zhao, P.B. Dickinson and A.T. Shenton, Decoupled Torque-AFR Control by Frobenius H_∞ Feedback and a λ Estimator, 8th UKACC International Control Conference, Coventry, September 2010
- S. Zhao, A.T. Shenton, A Nonparametric Method for Mixed Sensitivity Frobenius norm H_∞ Controller Design, 8th UKACC International Control Conference, Coventry, September 2010

Shiyu Zhao
January 2011

Abstract

With the rapid increase in the amount of vehicles all over the world, the emissions, particularly CO_2, are inevitably increasing dramatically. This causes the significant greenhouse gas effects and other environmental pollution. It is unrealistic to restrict the consumers' possession of more vehicles, or to reduce the fuel consumption and emissions by sacrificing the output power of the engine. New legislations requires vehicle manufacturers to produce more fuel efficient and less polluting vehicles. To reduce the emissions of vehicles and increase their fuel economy, advanced powertrain control techniques are required to improve the performance of vehicle. To this end many mechanical functions in automotive systems can now be replaced by and intergrated with electrical and electronic devices, to allow precise control and that can thereby give significantly improved performance. Existing industrial techniques to achieve a high performance control are based on feedforward mappings which require considerable and costly development time.

To achieve high performance for the powertrain control performance, in this thesis the nonparameric robust control approach to controller design is investigated and applied to powertrain system experimentally. The nonparametric approach allows the designer to work directly with experimentally obtained data and has the potential to give more accurate results by avoiding the approximation necessary with parameter fitting in the parametric method. Both single-input-single-output (SISO) and multiple-input-multiple-output (MIMO) (nonparametric) H_∞ optimisation techniques are developed and investigated. The first technique developed in this thesis is a novel SISO nonparametric H_∞ optimal controller design technique. This technique is successfully applied to the automotive engine idle speed control problem using air bleed valve (ABV) as the input. The second technique is a novel MIMO H_∞ Frobenius norm optimal controller design technique with its extension to a Hadamard weighted decoupling controller design technique. This technique is successfully applied to the MIMO air-fuel ratio (AFR) and torque automotive engine control problem using ABV and fuel pulse width (FPW) as the inputs.

Coupled with the MIMO H_∞ Frobenius norm, Hadamard weighted, decoupling

controller design technique, a novel calibration approach to design a fast acting feedback air-fuel ratio and torque controller is developed for the port fuel injection (PFI) gasoline engine by feeding back estimated air-fuel ratio and estimated torque. The air-fuel ratio and torque estimators are established by both an algebraic Nonlinear Auto Regressive eXogenous (NARX) and by a Neural Network (NN) method. A fast linear feedback system is identified from the engine with the estimators attached on-line and by fixing the engine speed. A robust multivariable decoupling controller is obtained based on this feedback system. The proposed approach using the H_∞ Frobenius norm controller is shown to provide performance consistent with current standards but with considerable reduce implementation effort than existing feedforward methods.

Contents

List of Figures

Chapter 1

Introduction

1.1 Automotive Engine Control

With the development of electronic technology, more and more mechanical functions on automotive are replaced by and intergrated with electrical and electronic devices. Now all automotive engines subject to emissions legislation are controlled by Engine Management System (EMS), a type of Electronic Control Unit (ECU), which measures engine transducers and issues the control commands to the engine actuators.Although such electronic control systems for automotive engines are applied in industry for many years, the modern automotive engine still has a large potential to improve its performance, such as reducing the fuel consumption and emission of pollutant, increasing safety, improving the drivability and comfort[34]. The improvements rely on the further development of electronics technology, automotive mechatronics and control design techniques. This thesis focuses on the automotive powertrain control problems of the spark ignition (SI) internal combustion (IC) engine, the motivation of the thesis is to investigate the novel modelling, control design techniques which might improve the performance for the modern automotive powertrain system. The engine control system implemented in EMS is a complex system which is related to many inter-related control problems. As the key variable in the engine control system, the engine torque output is required to satisfy the different requests on the EMS including internally generated request such as engine start, catalyst heating or externally generated request such as driver's request, transmission or vehicle dynamic control[28]. The air-fuel ratio (AFR) is another important variable in Engine Management System. Most of the time, the air-fuel ratio is required to be maintained at or close to the the stoichiometric ratio for air/fuel mixture. Besides the torque and air-fuel ratio control, many other control problems are handled in the engine control system. Some of the principal control problems in powertrain control system, including torque and air-fuel ratio control problems, are discussed in the following.

1.1.1 Torque Control

The torque controller is a central controller of the EMS.Precise torque control is not only to satisfy the driver's request and improve the drivability but also to enhance the driving safety by satisfying the vehicle dynamic control requirement in modern vehicles. Because the EMSs cannot measure the engine power or torque directly, the torque production is usually estimated from other variables. One challenge in the torque control problem is thus how to estimate a precise torque value. Figure 1.1 shows a simplified engine torque model which is employed in the Bosch ME7 EMS. In this model, the engine torque is estimated by using look up tables of air charge entering the cylinder, the engine speed, the λ value (air-fuel ratio) and the ignition timing.

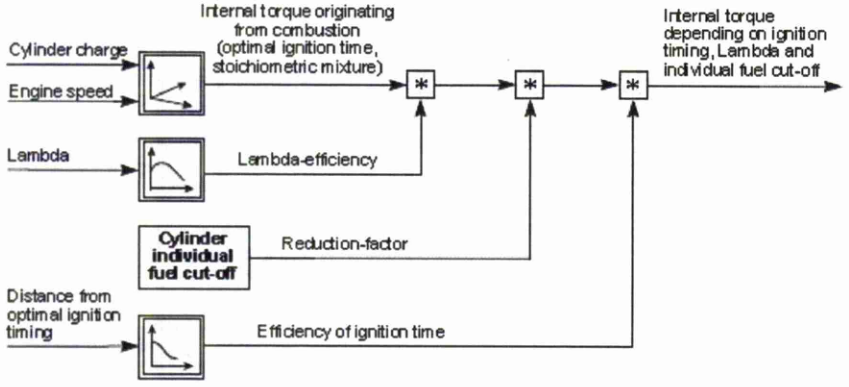

Figure 1.1: Simplified torque model[28]

The main requirement of the engine torque controller is to track and respond to demand with agilely across a large operation range. While the requirement for steady-state accuracy is not high, the pure feedforward compensation can not give good tracking capability in a large operation range. Practically however, a feedforward compensator for air charge coupled with a feedback controller can satisfy the main requirements of torque control problems.

1.1.2 Air-Fuel Ratio Control

The air-fuel ratio control is another central management task of the EMS. The three-way catalytic (TWC) converter requires the air-fuel ratio to be operated close to stoichiometric level to ensure sufficient reduction of NOx, CO and HC efficiently. The ideal stoichiometric level for the air-fuel mixture is a mass ratio of $14.7:1$[54]. The exhaust gas oxygen sensor situated in the exhaust pipe is usually used to measure the air-fuel

ratio. Historically vehicles are equipped a heated exhaust gas oxygen sensor (HEGO) switching type sensor, which is a very nonlinear device. An alternative sensor is the universal exhaust gas oxygen sensor (UEGO), which is a linear device. Although the UEGO sensor is more expensive, it gives a significant improvement for air-fuel ratio control.

Generally the oxygen sensor is mounted at the confluence point in the exhaust. Accordingly, there are large time lags due to the exhaust gas transport between the fuel injector and the oxygen sensor. This is the biggest challenge of air-fuel control problem. In the internal combustion engine, the air-fuel ratio is mainly determined by upstream air flow and fuel mass. The information on air flow can be obtained from the opening percentage of the air bleed valve (ABV) or the throttle angle of an electronic throttle, the manifold absolute pressure (MAP) or the mass air flow (MAF).

To overcome the issue of time delays, feedforward compensator is generally required to be applied. The accurate feedforward compensator for fuelling is determined by the air flow entering the engine, the measured air-fuel ratio and some other variables such as engine speed. The feedforward control scheme gives a fast response for air-fuel ratio control, however it requires a huge manpower effort on 'calibration' by the control engineers. Besides the feedforward control scheme, the feedback controller is also applied to zero the steady state error. When the HEGO system is used this is implemented as a limit cycle controller.

1.1.3 Idle Speed Control

The idle speed controller is an important component of the powertrain control system. Nowadays the overcrowded traffic causes the vehicles to consume up to 30% of their fuel at idle speed[50]. To reduce the fuel consumption, the idle speed controller is required to control the idle speed of engine at as a low speed as possible. The literature [70] indicates that when the idle speed is reduced from 800 to 650 rpm can make fuel saving of approximately 24% at idle. Meanwhile another requirement for the idle speed controller is to reject any external disturbance, such as electrical loads on the alternator or air conditioning pump and power steering pump, to minimise the variation of engine speed and avoid engine stall[46].

In the spark ignition internal combustion engine, the engine speed is influenced by three variables, which are the mass of air entering to the engine, the amount of fuel injected to the engine and the spark ignition timing. In the idle speed problem, the mass of air is determined by the opening percentage of the air bleed valve (ABV) in old low cost vehicles or opening angle of electronic throttle valve (ETV) in modern advanced vehicles. The mass of fuel is determined by the time of the fuel injection.

The spark ignition time is determined by the advance angle for the spark event. The controller designer can thus control the idle speed by these three potential channels.

Actually, the air entering the engine is the primary channel to regulate the engine speed. Although the fuel injection can also control the engine speed, in a port-fuel-injection (PFI) engine the control effort is slow due to the time delay and lags caused wall wetting dynamics and induction to power stroke delay. Accordingly the ABV or ETV is usually used to control the idle speed of the engine, while the fuel mass is determined by the air-fuel ratio requirement. Additionally, the optimal setting of spark advance (SA) for maximum best torque (MBT) is often used to help the engine speed to respond to demand faster by forming a two input one output idle speed controller.

1.1.4 Combustion Control

The combustion in the spark ignition internal combustion engine is required to be controlled to ensure the maximum use of the energy available so as to obtain maximum efficiency of the engine. The combustion control relies on the timing of the ignition. For example, if ignition of the charge is too early in the compression stroke, the resulting charge expansion is also early, therefore the work done is not maximised and energy is lost to heat in the cylinders; if ignition of the charge is too late, that causes the poor use of the energy available in charge, as energy is lost through heating the exhaust gases. Accordingly the optimal setting for the spark timing can make higher for a efficiency engine operation.

The ignition timing conventionally described in terms of the spark (ignition) advance (angle) before the top dead centre (TDC) of the crankshaft. It has been found that the optimal spark advance (SA) for a particular engine coincides more or less with a fixed peak pressure position (PPP)[45]. Accordingly the optimal setting for spark advance can be determined by feeding back the peak pressure position if a pressure sensor is fitted to the engine. Because the spark advance and peak pressure position are best measured based on a crank angle basis, the combustion control must be operated by a crank based event triggered discrete system, not in a sample time based discrete system. Except in expensive vehicles pressure sensors are not generally available however.

In the common port-fuel-injection (PFI) engine, the spark advance is optimised according to the operating condtions including engine speed, load, knock prevention. In gasoline direct injection (GDI) engines, the igniting timing requirement is different due to the two operation modes of the GDI engine - homogeneous charge mode and stratified charge (ultra lean burn) mode.

1.1.5 Knock Control

Engine knock, also called pinging or pinking etc., is a phenomenon which occurs due to uncontrolled combustion in the spark ignition internal combustion engine. Normally the air-fuel charge should be ignited by the spark plug only with the optimum igniting timing setting. However the pre-ignition of the charge can happen prior to the flame front initiated by the spark plug, caused by the sufficiently high pressure and localised high temperature on the cylinder walls sometimes[45]. In the GDI engine, the higher pressure required for the direct fuel injection is more likely to cause engine knock.

To detect knocking, piezoelectric accelerometers on the engine block are typically used. The basic control strategy is to retards the SA or reduce the turbocharger boost pressure once the knocking is detected; when the knocking has disappeared, the SA is gradually ramped to the optimal setting or the turbocharger boost pressure increase [77]. This simple adaptive control technique is the most commonly used in knock control.

1.2 Overview of thesis

In this thesis a nonparametric design methodology for automotive engine control system is presented. The remainder of the thesis is sectioned into appropriate parts and these are detailed here along with a summary of these chapters.

Chapter 2

This chapter gives an overview of the system modelling, controller design process and implementation of embedded systems. First, the different modelling methods, physical modelling, grey-box and black-box modelling, are discussed and compared. The black-box modelling (system identification) approach is introduced. The procedure of system identification, including experiment design, data collection, model selection, structure selection, parameters calculation and model validation, are also summarised.

Next, the basic methods of robust control theory are introduced. The underlying concepts of robust control, such as uncertainty, robust stability and sensitivity functions are discussed. Then the standard H_∞ control problem is presented. The design methods based on H_∞ Riccati equation technique, H_∞ loop shaping technique, Linear Matrix Inequalities (LMIs) technique and H_∞ optimisation over analytic functions are introduced.

Finally, a new rapid prototyping system for embedded systems using SIMULINK

model based programming software is introduced. Then the language based programming method is compared to that of the rapid prototyping programming method. The advantage and characteristics of the rapid prototyping methods are presented. The basic procedures involved in the rapid prototyping are introduced.

Chapter 3

In this chapter the basic setup of the experiment work is introduced. The main equipment is a Ford Zetec 1.6 litre PFI SI engine connected to the University of Liverpool low inertia dynamometer. The characteristics of the engine and dynamometer are explained. The sensors and actuators installed on engine are described. Then the interfacing and connections of the dSPACE rapid prototyping system associated those hardware are also presented.

Chapter 4

This chapter presents the parametric and nonparametric SISO controller design approaches based on H_∞ optimization over analytic functions. First, the H_∞ optimal control problem is presented. Then, the concept of internal stability is introduced. The interpolation conditions for internal stability and the corresponding interpolation method are summarised. The closed loop system can thus be ensured internally stable by parametrising the complementary sensitivity function T with that interpolation function during optimisation. Another idea which also makes the system internally stable is to optimise using the control effort sensitivity function U as the optimisation variable when the plant is stable. This method can then be applied in a nonparametric design approach. The optimisation method over analytic functions (analytic on the unit disk or in the right half plane) for the scalar case can solve the H_∞ optimisation for the 'disk problem' and 'mixed sensitivity' optimisation problem. Then the parametric controller design procedure for the disk problem and the mixed sensitivity problem are then presented. A simple example of a peak pressure position (PPP) control problem is illustrated. The results of a single sensitivity design (disk problem optimisation) and mixed sensitivity optimisation design are then compared. Based on the analytic function optimisation method, a novel nonparametric controller design approach is proposed. The detail and design procedure for the disk problem optimisation and the mixed sensitivity optimisation are presented. A nonparametric design example is illustrated for a nonparametric driveline model.

Chapter 5

In this chapter the parametric and nonparametric H_∞ optimal controller design approaches discussed in last chapter are applied to the idle speed control problem. First, the modelling of engine model is presented. For this purpose the nonparametric identification Blackman-Tukey method is introduced. The system transmission from ABV to engine speed is identified by an autoregressive model with exogenous inputs(ARX)/autoregressive moving average model with exogenous(ARMAX) approach and by the nonparametric Blackman-Tukey method. The results are discussed and compared. Then the controller design procedures for the parametric and nonparametric disk problem and mixed sensitivity optimisation are presented. The simulation results of the different controllers are compared. The experimental test results of several controllers based on different design methods are thus presented. The performance and robustness are compared for different controllers.

Once the controller is designed a novel rapid prototyping approach is used to implement the controller in an embedded system. First, a new fast programming tool for PIC/dsPIC target devices is described. The software is associated with MATLAB/SIMULINK, and the graphical SIMULINK model can be used directly for programming. Then the industry standard CAN-bus communication protocol is described. The idle speed controller is implemented in a CAN-bus based control system based on dsPIC33F microcontroller. The controller can thus be considered as a part of a distributed control system. The programming procedure for the dsPIC using the SIMULINK model based programming tool is presented. The configuration for the dsPIC's oscillator, ADC module, ECAN module and DMA module are discussed in detail. The SIMULINK model of the whole program is also displayed. Then the experimental results are presented.

Chapter 6

This chapter presents a multivariable H_∞ Frobenius norm optimal control method based on analytical optimisation. Similarly, the internal stability conditions for multiple-input-multiple-output (MIMO) systems are introduced. The interpolation constraints for internal stability are presented. A parametrisation method of interpolation functions for simple poles/zeros is developed. A MIMO controller design approach using H_∞ optimisation over analytic functions for the vector case was developed. The matrix mixed sensitivity problem is considered for the MIMO design case. That approach works by minimising the H_∞ Frobenius norm instead of the currently used largest singular value H_∞ norm. An extension for designing a novel decoupling controller for a MIMO system is described. The Hadamard (element-by-element) product can be

applied to the multiplication of the weighting function and the sensitivity function in the controller design, so that each element of the sensitivity transfer matrix can be constrained separately by its own weighting function. The closed loop transfer matrix T for a decoupled system is diagonal matrix. When using the Hadamard product weighting non-diagonal elements of the sensitivity matrix T or matrix S can be constrained to zero or close to zero, meanwhile the diagonal elements of the sensitivity matrix T or matrix S can be constrained by regular weighting functions. The parametric and non-parametric design procedures using interpolation function are developed and presented respectively. A design example of a multivariable torque and air-fuel ratio control problem is presented. A novel Nonlinear Auto Regressive eXogenous (NARX) air-fuel ratio estimator using fuel pulse width (FPW), manifold absolute pressure (MAP) and engine speed with time delay removed is introduced, such that the air-fuel ratio is predicted prior to the measured air-fuel ratio. Because the very large time delay is thus removed, this makes feedback control scheme possible for the air-fuel ratio channel. The nonlinear identification for air-fuel ratio estimator is detailed. The identification for a 2×2 linear system from air bleed valve (ABV) duty and FPW to air-fuel ratio prediction and engine torque obtained from the engine with the estimator attached on-line and by fixing the engine speed is then presented. The linear model is then used to design the feedback controllers by H_∞ Frobenius norm optimal decoupling method and 2 Riccati method. Both the simulation and experimental test results for the decoupling controller and the H_∞ Riccati controller are compared.

Chapter 7

This chapter gives an alternative novel calibration method for multivariable air-fuel ratio and engine torque control system. Again, the control objective and requirements of MIMO AFR and torque system are introduced. In this chapter, the Neural Network (NN) technology is introduced and employed for the air-fuel ratio and engine torque estimator. Both estimators are established with time delays removed. A series-parallel nonlinear autoregressive network with exogenous inputs(NARX) using FPW, MAP, engine speed and measured air-fuel ratio are developed for air-fuel ratio estimator. Because no power/torque sensors are available to the EMS of conventional equipped vehicles. A torque estimator is established by a feedforward multi-layer perceptron(MLP) network using air-fuel ratio, MAP and engine speed in this work. The feedback controller based on this estimator is found to be faster. The detailed procedures for establishing the torque/AFR NN estimators are presented. As with the work in the previous chapter, a 2×2 linear system for the engine coupled with nonlinear estimator from ABV and FPW to engine torque prediction and air-fuel ratio prediction is thus established. The detailed experimental test are carried out to assess tracking

capability, disturbance rejection capability and random driving cycle performance and robustness. The experimental results of the H_∞ Frobenius norm optimal decoupling design approach and the 2 Riccati method are compared.

Chapter 8

This chapter concludes and summarises all previous chapters. The advantages and disadvantages of parametric and nonparametric H_∞ optimisation technique are presented. Suggestions for improving the H_∞ optimisation technique, developing AFR and torque model and extending the implementation of the CAN-bus networked system to more complicated applications are discussed and suggested.

1.3 Contributions of thesis

The elements of novelty in this thesis are:

- A novel nonparametric H_∞ optimal controller design approach based on the analytic function optimisation method is developed. This approach is to find the internally stable controller by solving the H_∞ optimisation problem for the mixed sensitivity problem and the disk problem. Generally, the optimal solution can be found in terms of the complementary sensitivity function T. In the more common parametric case, the optimal solution for T is coupled with Newton interpolation functions which ensure that the closed loop system internally stable. A new nonparametric design approach is developed based on optimisation for control effort sensitivity function U. It was found that the nonparametric controller designed based on a nominal (window sized) model gives a significant robustness in time delay.
- A novel nonparametric multivariable H_∞ optimal controller design approach using the H_∞ Frobenius norm instead of the maximum singular value norms based on the analytical optimisation method is developed. An interpolation method for simple poles/zeros of MIMO case is developed. In parametric design, the interpolation functions can be determined by the unstable poles or zeros of plant. In nonparametric design, the interpolation functions can be formed by the frequency response of the determinant of the transfer function matrix when the plant is a stable non-minimum phase system. A novel extension of MIMO H_∞ optimal controller design method is the decoupling MIMO controller design approach. Through using Hadamard product, a diagonal or quasi-diagonal matrix for the closed loop transfer matrix T and matrix S can be obtained. This may make the closed loop system decoupled.

- A novel calibration method for the AFR and torque control system in PFI SI engine is developed. Instead of the traditional feedforward look-up table modelling method for AFR and torque control, a nonlinear black-box parameter identification method is used to produce the estimators for AFR and torque. The algebraic NARX and NN approach are both used to establish an AFR estimator with time delays removed. It was found that the series-parallel NARX network which used measured AFR gives a more accurate prediction. The time delay for the torque estimator established by a feedforward MLP network is also removed. The experimental results show the feedback control system using such predicting AFR and torque values is faster the traditional method. The H_∞ decoupling controller is applied to the MIMO AFR and torque estimator and engine system. Comparisons show that the decoupling controller is more decoupled than the traditional Riccati controller as required.

- The implementation of an embedded system for a SISO idle speed controller using a new rapid prototyping method for embedded system is demonstrated. The controller designed and simulated in MATLAB/SIMULINK can be implemented to the target device directly. The idle speed controller is implemented in a CAN-bus based control system. This setup was shown to provide significant robustness to network time delay when implemented with the analytical optimisation based H_∞ controller. This type of distributed controller can thus be employed in more industry applications where delays in the CAN network occur because of network overloading.

Chapter 2

Literature Review

2.1 Introduction

In automotive powertrain control applications, a wide scope of techniques are used such as system modelling, controller design and hardware implementation, etc. The process of modelling, control and implementation are three main stages for general control system establishment. In this chapter some modelling, controller design and hardware implementation techniques are summarised for automotive powertrain control problem.

The automotive powertrain system is a comprehensive system which includes many linear and nonlinear parts. To establish a precise engine model, both linear or nonlinear system modelling and identification techniques must be considered. In section 2.2, the various approaches for system modelling using 'black-box' linear and nonlinear identification techniques are summarised. In section 2.3, the fundamentals of robust feedback control techniques are introduced. The optimal H_∞ theory is also summarised in this section.

2.2 Engine System Modelling

In the 80s-90s of the last century, computer technology developed quickly. System modelling technology benefited from that progress and also move forward quickly. Many novel system modelling methods were developed, especially for the more complicated nonlinear systems. As one of the most important tasks of the modern control techniques, advanced system modelling technology is used in more and more advanced forms in the automotive industry. Performance such as precise control of torque output and fuel emission minimisation for automotive engine has been increased greatly in the last two decades.

Most controller design methods are based on a representative model of the physical system. Obviously, the accuracy of a controller produced from such model is directly related to the quality of the model. Generally, modelling techniques can be divided into two main approaches, physical modelling and black-box modelling. The physical model is modelled based on physical theory and understanding of the candidate system. While, the black-box modelling depends on the experimental test and data collection from the physical system [93]. To identify a black-box model, the physical system should be excited with a suitable testing signal in a particular range, and the corresponding output data should be measured and collected. The data recorded is then used to build a mathematical model to represent the system dynamics. A third approach is called grey-box modelling, which is a bridge between the physical modelling and black-box modelling. In this method, the model is a mix of the two types of modelling methods, where the physical models are partially represented by system identification and partially modelled based on the physical theory [83]. Each of the three methods are appropriate in different circumstances.

To establish the engine models for control, two typical modelling approaches are often used. They are continuous mean value models (MVM) and discrete event models (DEM). The DEM is based on the discrete engine cycle events whereas the MVM is not. So that the MVM are used for relatively slow changing processes, whereas the DEM are used for models which is crank-based periodic events, such as combustion modelling [34].

2.2.1 Physical Modelling

The physical model is also called a 'white-box' model. This model is obtained from the analysis of mechanisms or processes for the physical system by using some basic physical principles or laws such as Newton's law, or the 1st or 2nd laws of thermodynamics. To create a physical model, the dynamics of the system should be understood clearly, the physcial parameters also should be known with a degree of certainty. But in particular applications, such as the internal combustion (IC) engine control case, there are a very large number of physical processes occurring simultaneously, including kinematics, thermodynamics and fluid dynamics. Therefore it is very difficult to obtain an accurate physical model suitable for control. In [34] the techniques for physical modelling with particular application to powertrain models are introduced and described in detail.

There are some advantages to this type of modelling. One is that it provides insight into the system. This can help the researchers to choose the appropriate control laws. Another advantage as described in [44] is that once a satisfactory model has been generated this should be adaptable to similar systems by means of changing basic

values. Whereas a black-box identified model is only relevant for the system from which the data was collected and the operating range over which it is identified. An entire physical model for spark ignition (SI) port fuel injection (PFI) engine model is given in [44]. However the parameter for any particular engine remains to be selected for this model.

2.2.2 Black-box Modelling

The black-box model is generally obtained as dynamic identification model. The black-box modelling approach is to build a mathematical model of a system from the observed behaviour of the physical system. The advantage of identification models is that the researchers do not require a deep physical understanding of the candidate system, because the physical phenomenon of the system is represented in the input-output data.

The identification model of a system is developed by collecting the appropriate exciting signals (input) and the observation of the response (output). The relationship between the input and output data can then be determined by a number of linear and nonlinear identification techniques. The details of system identification techniques can be found in [57], [58] and [76], and for nonlinear identification in [79]. The standard processes for system identification can be summarised in the following procedures.

Design of test signal

To capture the dynamics of the system, the system should be excited by a test signal. The test signal is an input signal for the candidate system. The black-box model describes the relationship between the test signal and the output behaviour of the system. A good test signal is a prerequisite for a successful system identification. So that the frequency spectrum, amplitude of the signal should be considered and chosen carefully to construct a proper test signal. A good test signal should have the proper frequency spectrum and amplitude to excite the full desired frequency range and sensitivity to the magnitude of the system.

For linear identification, the test signal can be a two-level pseudo random binary sequences (PRBS) signal, because they can excite a wide band of frequencies and have near white noise autocorrelation. For nonlinear identification, a multi-level test signal is often used. This test signal can excite not only the relevant frequencies in the system but also the magnitude at these frequencies. The technique for choosing test signal are discussed in [25] and [32].

Data Collection

Once the system is excited by the test signal, the output data which contains system dynamics can be captured. The sampling frequency for data capturing is very important as the sampling frequency is a limitation to the frequency which can be captured. Only the frequencies at half the sampling rate and lower being identifiable. Generally the sampling frequency is selected at around 10 times the system bandwidth which is suggested by [76].

The data should be processed further before identification. Usually the data is collected at a high sampling rate to capture data in a high enough frequency range, while the noise, such as sensor noise, can be filtered out by passing the higher frequency data through a low pass filter. The data can thus be down sampled to the desired discrete sampling rate to remove any large outliers that may also affect the quality of the model. A description of a systematic sampling strategy can be seen in [32]. Once it is filtered, the candidate data can be divided into two parts, one is for the identification and another one is for the validation of the resulting model. For linear identification, the trends or mean values of the data should be removed.

Selection of Model

There are many types of models that can be chosen by practitioners for the different applications. Usually, a linear model is considered first. If the linear model cannot satisfy the performance requirement, since some nonlinear behaviour of the system can not be presented by the linear model, a nonlinear model must be considered. To establish a linear or nonliear model many identification techniques can be applied. In linear identification, the model types of Output Error (OE), Box Jenkins (BJ), Auto Regressive eXogenous (ARX), Auto Regressive Moving Average eXogenous (ARMAX) and so on are often used [58]. In nonlinear identification, similar types of the models are used, such as Nonlinear Output Error (NOE), Nonlinear Auto Regressive eXogenous (NARX), Nonlinear Auto Regressive Moving Average eXogenous (NARMAX), Nonlinear Box-Jenkins (NBJ), Neural Networks (NN) [68]. The models most commonly used for identification tend to be the autoregressive model, which means the output of the model is based on a combination of previous inputs and outputs. This type of model is a parametric model which is based on regressors and is a function of the corresponding coefficients or parameters. Besides the parametric model, the nonparametric model is also an important type of model used in identification. A typical nonparametric model is the frequency response model from the Blackman-Tukey method which is based on spectral analysis. Recently some new modelling concepts have been introduced, such as Neural Networks (NN), Fuzzy and Neuro-Fuzzy (NF) models. These are increas-

ingly popular methods of producing effective identification and control in a nonlinear manner.

Selection of Structure

The selection of the model structure is one of the most important and difficult procedures in the system identification. For linear identification, the main task of the structure determination is to select the orders of the linear functions because the regressors of the functions are standard linear ones. For nonlinear identification, the selection of the structure is more complicated. For Neural Network models, the selection of structure consists of the choice of the number of neurons and layers, and the connection of nodes and layers. In the nonlinear autoregressive models, such as a NARX model, the regressors can be either linear or nonlinear. The nonlinear regressors can be combinations of the controlled inputs and measured outputs with the different discrete delays between the inputs and outputs. So usually there is a large number of possible candidate regressors for any nonlinear identification. The structure of nonlinear autoregressive models is selected by choosing suitable regressors from a collection of many, the designer needs to assess if there is any significant relationships between the regressors and the measured output. Now many regressor selection methods have been researched and developed. There is a good review paper for the various techniques of regressor selection to be found in [36], with more specific algorithms given in [4], [86] and [85]. Summarizing the selection of structure; the basic principle is to choose the lowest number of parameters possible whilst still satisfactorily characterising the system dynamics to reduce the complexity of the model. At the same time the risk of over parametrisation and rank deficiency must also be considered.

Calculation of Parameters

Once the structure is determined, the parameters, such as for each regressor of a autoregressive model or for each neurons of a Neural Networks model, should be calculated. There are many algorithms available to obtain the unknown parameters. The basic idea for finding parameters is to minimise the error between the predictions of the mathematical model and the measured physical behaviour. To minimise the error, the mathematical optimisation method is often used. In system identification, the most common technique is the ordinary least squares (OLS) method:

$$\theta = (X^T X)^{-1} X^T y \tag{2.1}$$

where θ is a matrix of parametrised regressors, X is a matrix of regressor data and y is the observed output data [58], [76]. As the most common form of least squares fitting,

this method is computationally quick to use and is easily adapted to handle nonlinear regressors. Besides the OLS method, some other methods are also used in system identification, especially in nonlinear identification. Direct search methods and gradient based algorithms such as Newton, quasi-Newton, steepest descent, with penalty function approaches are usually used in nonlinear local optimisation. Other advanced algorithms such as genetic algorithm (GA) are usually used in nonlinear global optimisation. The parameter calculation methods for system identification are summarised and classified as linear optimisation, nonlinear local optimisation and nonlinear global optimisation in [68].

Model Validation

Once the model is obtained, it has to be validated with the measured output data that has not been used in the identification of the model (known as unseen data). Of course, a visual inspection of the predicted output data against that measured output data will show you the quality of the fit produced straight away. But to give a quantitative understanding of the model quality, statistical fit assessments obtained by mathematical calculation can be used. The most popular method is to examine the square of the residuals:

$$R^2 = 1 - \frac{\sigma^2\varepsilon((t))}{\sigma^2(y(t))} \tag{2.2}$$

where $\sigma^2\varepsilon((t))$ is the variance of the model residuals and $\sigma^2(y(t))$ is the variance of the system output [58], [64]. Therefore when $R^2 = 1$, that means the model can recover 100% of the real system.

2.3 Robust Feedback Control

Optimal robust control theory was first developed in 1980s. Before that, classical control theory was studied in the 1940s and 1950s, well established in the 1960s and 1970s. At that time, controller designers focused mainly on the frequency domain performance. Because there are some differences and variation between the mathematical model and the actual system, sometimes a controller with good performance in simulation may show a bad performance in actual testing or even the controller may not work at all and be unstable. So control systems are required to tolerate model uncertainty and external disturbance. That tolerance or redundancy can be considered as the 'robustness'. Based on the classical frequency domain techniques, such as based on the Nyquist diagram[69], Bode diagram[5] and root-locus method[24], the 'robustness' can be assessed in terms of gain margins (GM) , phase margins (PM) and vector margin (VM) or more generally the distance of the open loop frequency locus from the critical

-1 point. To ensure that the system has a good robustness for its uncertainty, controller designers must make the necessary trade off between system performance and robustness. In the early days, the designers kept a robustness tolerance by maintaining a certain degree by GM and PM, then checking whether the performance is good or not through tests like step impulse responses. So the design processes based on those techniques are complex and iterative. The current robust control theory has considered uncertainty as a part of the design. The aim of robust control design is to obtain a controller which can maintain the system performance in a certain degree in spite of the plant's variation or uncertainty and external disturbance. In automotive engine control applications, the uncertainty of the plant may be due to many reasons, such as a wide variation of manufacturing tolerances, varying environmental conditions and varying wear rates across the entire life cycle of the engine.

A basic unity gain negative feedback system is shown in Figure 2.1. P and K represent the plant and the controller respectively. The controller is required to keep the output y at the desired level r. To do that, the output y is fed back to compare with the desired reference r. The error signal e between two values enters the controller K. Then the controller generates the input signal u for plant P to compensate the error. Ideally a feedforward controller would be the perfect controller [80]. The feedforward can be the exact inverse of the plant if it could be inverted as a stable system. However the perfect model which is exact and invertible is rarely realisable due to the difficulties of accurate modelling to non-minimum phase (NMP) characteristics and to unknown disturbances. A feedback controller is required to reject any external disturbance and has to be 'robust' to the uncertainty of plant modelling. Some linear robust controller design techniques: H_∞ control, H_∞ loop shaping, linear matrix inequalities (LMI) and H_∞ optimisation over analytic functions are described in this chapter.

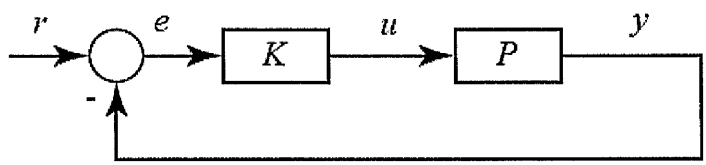

Figure 2.1: Feedback control loop

2.3.1 Uncertainty and Robust Stability

There are several different types of description for uncertainty. The feedback loop for a type of system with uncertainty is shown in figure 2.2. P_0 is the unperturbed or

nominal plant, Δ is the unknown uncertainty, W is the envelope or boundary of that uncertainty. The perturbed plant P can be modelled as the nominal plant P_0 with uncertainty Δ and boundary W

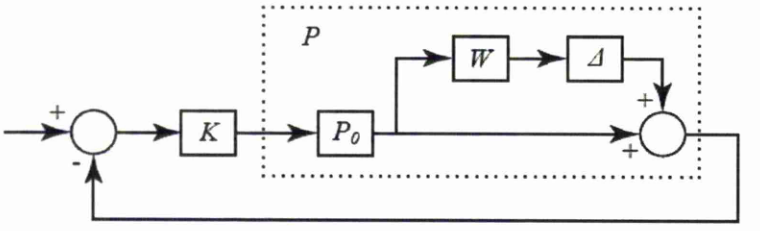

Figure 2.2: Feedback loop of uncertain system

The perturbed plant P can the nbe described in mathematical form as:

$$P = [1 + \Delta W]P_0 \tag{2.3}$$

This modelling description for uncertainty is called *multiplicative dynamic uncertainty*[71]. To determine the stability of systems with such uncertainty, Zames' small gain theorem[97] is used. Consider the interconnection shown in figure 2.3. Assume that Δ is stable and $\|\Delta\|_\infty < 1$, then the interconnection is stable if $\|M\|_\infty \leq 1$. Similarly, the stability of

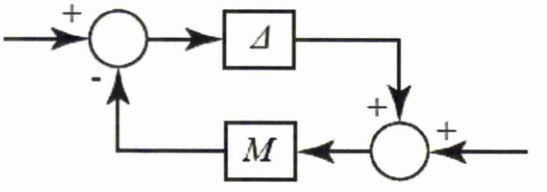

Figure 2.3: Small gain theorem

the feedback loop shown figure 2.2 can also be judged by small gain theory. Consider the Δ in figure 2.2 to be the same as the Δ in figure 2.3; simplify the block diagram from figure 2.2 to figure 2.3, then $W\frac{P_0K}{1+P_0K}$ in figure 2.2 is equivalent to M in figure 2.3. If $\|\Delta\|_\infty < 1$; Δ and W are stable, then feedback loop is stable if $\left\|W\frac{P_0K}{1+P_0K}\right\|_\infty \leq 1$. The Δ is the bounded uncertainty, it is unknown except for its norm. The variation in the plant, external disturbance, even some nonlinearities can be modelled as the uncertainty.

Besides the description of multiplicative dynamic uncertainty, there are some other types of uncertainty descriptions. They can similarly be formulated as mathematical models and robust stability conditions can also be found using Zames' small gain theorem. The uncertainty description and corresponding robust stability conditions are listed in the Table 2.1.

Perturbed plant	Robust Stability Condition
$P = [1 + \Delta W]P_0$	$\|W\frac{P_0K}{1+P_0K}\|_\infty \leq 1$
$P = P_0 + \Delta W$	$\|W\frac{K}{1+P_0K}\|_\infty \leq 1$
$P = P_0[1 + \Delta WP_0]^{-1}$	$\|W\frac{P_0}{1+P_0K}\|_\infty \leq 1$
$P = P_0[1 + \Delta W]^{-1}$	$\|W\frac{1}{1+P_0K}\|_\infty \leq 1$

Table 2.1: The uncertain plant and corresponding stability condition[71]

The functions $\frac{P_0K}{1+P_0K}$, $\frac{K}{1+P_0K}$, $\frac{1}{1+P_0K}$ and $\frac{P_0}{1+P_0K}$ in the Table 2.1 are called the sensitivity functions and which are also the different transmissions in the closed loop system. These will be discussed in the following section.

2.3.2 Sensitivity Functions

A typical unity gain negative feedback system is shown in figure 2.4, assuming a single-input-single-output (SISO) system. $K(s)$ and $P(s)$ represent the controller and the

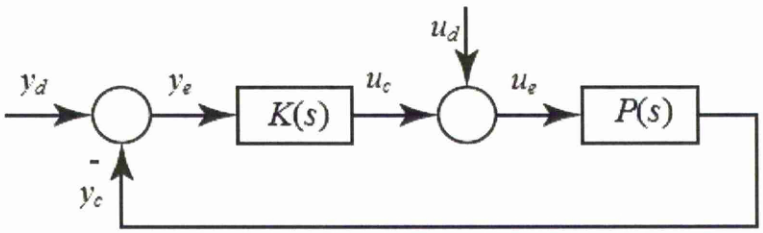

Figure 2.4: Extended disturbance system

plant respectively. There are four unique closed loop transmissions between the different inputs and outputs. In the multiple-input-multiple-output (MIMO) case, there are six transmissions which will be discussed in the following chapters. These transmissions are also called sensitivity functions. The transmissions of those input-output pairs completely determine the system dynamics and controller performance. The H_∞ control approach is a typical robust control technique. In the H_∞ control problem, the frequency response of the sensitivity functions are constrained by the weighting functions to achieve the desired system performance and robustness requirements. The most commonly utilised being primary, complementary and control effort sensitivities.

Primary Sensitivity Function

The primary sensitivity function is the closed loop transmission from the system input or demand signal, denoted by y_d, to the tracking error signal, denoted by y_e. In figure 2.4, the transfer function of this transmission is:

$$y_e = \frac{1}{1 + P(s)K(s)}y_d \tag{2.4}$$

where s denoting the Laplace variable. The primary sensitivity is defined by:

$$S(s) = [I + P(s)K(s)]^{-1} \tag{2.5}$$

The primary sensitivity function S is mainly used for the tracking or regulating the closed loop transient response performance. Usually the common disturbances are the low frequency signals. To track the demand signal despite the disturbance, the error signal must be zero. So the primary sensitivity function S must be kept small particularly at the low frequency. The gain-phase margin, which is the minimum distance from the locus of open loop transmission $L(s) = P(s)K(s)$ to the -1 point, is also related to the primary sensitivity. For the SISO case, gain-phase margin m is expressed as $m = \inf_\omega |1 + L(j\omega)|$. So the gain-phase margin constraint on the primary sensitivity function S is $|S(j\omega)| \leq 1/m$ for all frequencies. According to the robust stability condition the following constraint should be satisfied in robust (H_∞) control design.

$$\|W_S(s) \cdot S(s)\|_\infty \leq 1 \tag{2.6}$$

where $W_s(s)$ is the weighting function for the primary sensitivity function, which is established from the system requirements mentioned above.

Complementary Sensitivity Function

The complementary sensitivity function is described as the closed loop transmission from the system input or demand signal y_d, to the system output signal, denoted by y_c. In figure 2.4, the transfer function of this transmission is:

$$y_c = \frac{P(s)K(s)}{1 + P(s)K(s)} y_d \tag{2.7}$$

The complementary sensitivity is defined by:

$$T(s) = P(s)K(s)[I + P(s)K(s)]^{-1} \tag{2.8}$$

The complementary sensitivity function characterises the closed loop system output to variations in the demand signal. This sensitivity function can be considered as the sense of the system robustness. To ensure the system is robust to the system uncertainty, the complementary sensitivity should has a 0dB gain at low frequency due to requirement of DC gain and roll off to the higher frequencies as the uncertainty of the model becomes greater[80]. In robust controller design, the weighting function chosen for complementary sensitivity should again constrain the target function. Accordingly in robust (H_∞) control design the following constraint should be satisfied.

$$\|W_T(s) \cdot T(s)\|_\infty \leq 1 \tag{2.9}$$

where $W_T(s)$ is the weighting function for the complementary sensitivity function obtained from the system requirements.

Control Effort Sensitivity Function

The control effort sensitivity is the transmission from the input or command signal y_d, to the controller output signal, denoted by u_c. The transfer function of the input-output pair is:

$$u_c = \frac{K(s)}{1 + P(s)K(s)} y_d \tag{2.10}$$

Then the control effort sensitivity is defined by:

$$U(s) = K(s)[1 + P(s)K(s)]^{-1} \tag{2.11}$$

This function determines the amount of control activity at the plant output per amount of demanded input. Generally, the control activity must be limited because the plant actuator easily saturates and also too much activity soon wears out the actuator. Therefore in the robust controller design, controller's output should have a limit on the magnitude and rate. To ensure the controller does not exceed the limit of the actual possible control action, the control effort sensitivity should below a certain value. So the weighting function for this sensitivity often be taken as constant gain at all frequencies. Additionally to avoid the high frequency noise, the control effort sensitivity is better having a small gain at high frequencies. This can be also established a weighting function. Accordingly in robust (H_∞) control design the following constraint should be satisfied.

$$\|W_U(s) \cdot U(s)\|_\infty \leq 1 \tag{2.12}$$

where $W_U(s)$ is the weighting function for the control effort sensitivity function obtained from the system requirements.

Plant Input Disturbance Sensitivity Function

The plant input disturbance sensitivity is the transmission from the disturbance input u_d, to the system output signal, which is denoted by y_c. The transfer function of the input-output pair is:

$$y_c = \frac{P(s)}{1 + P(s)K(s)} u_d \tag{2.13}$$

The plant input disturbance sensitivity is defined by:

$$V(s) = P(s)[1 + P(s)K(s)]^{-1} \tag{2.14}$$

This sensitivity can be considered as the sense of the disturbance for plant. In H_∞ control design the following equation should be solved.

$$\|W(s) \cdot V(s)\|_\infty \leq 1 \tag{2.15}$$

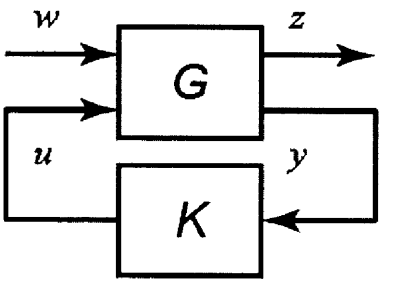

Figure 2.5: General loop block for H_∞ problem

2.3.3 H_∞ Control

One of the most popular robust control design initially approach is the use of H_∞ feedback control theory. This theory was also developed from the 1980s, initially formulated by Zames [98]. The background to the topic can be found in [49], [48] and [101]. In H_∞ control theory, the induced H_∞ norm of $G(s)$, where $y(s) = G(s)u(s)$, is defined as follows [49]:

$$\|G\|_\infty = \sup_\omega \frac{\|y(j\omega)\|_2}{\|u(j\omega)\|_2} \tag{2.16}$$

where y is the output and u is the input of a system. Thus $\|G\|_\infty$ is the supremum of the ratio between input u and output y. In control engineering, the H_∞ norm is described as the peak amplitude of the frequency response or the maximum singular value of the frequency response matrix in MIMO system. A common formulation is the block control problem. Figure 2.5 shows the general loop block diagram for the H_∞ control problem. The sub-optimal controller is the controller which ensures $\|T_{zw}\|_\infty < \gamma$ where γ is a pre-determined value. In another words, the ∞-norm of closed loop transfer function T_{zw} from the disturbance input w to the controlled output z should remain below than a prescribed level across all frequencies.

The block G in figure 2.5 is the augmented system, which can be described in partitioned matrix form as:

$$\begin{bmatrix} z \\ y \end{bmatrix} = \left[\begin{array}{c:c} G_{11} & G_{12} \\ \hdashline G_{21} & G_{22} \end{array}\right] \begin{bmatrix} w \\ u \end{bmatrix} \tag{2.17}$$

where the G, comprised of G_{11}, G_{12}, G_{21} and G_{22} used for controller synthesis. To show the controller synthesis method in detail, we examine a design example, which is a multiple sensitivity problem with primary, control effort and complementary sensitivity function constraints. Now the augmented system G can be written as

$$G = \left[\begin{array}{c:c} W_S & -W_S P \\ 0 & W_U \\ 0 & W_T P \\ \hdashline I & -P \end{array}\right] \tag{2.18}$$

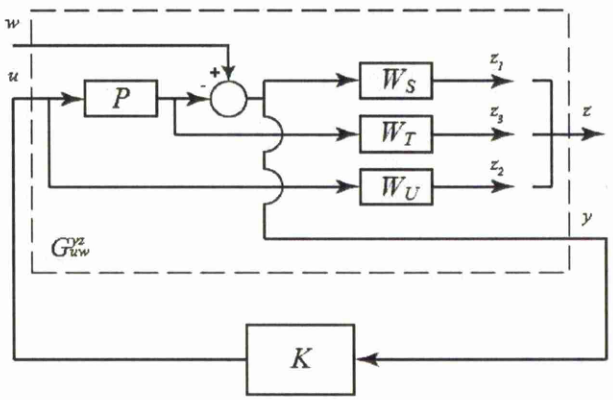

Figure 2.6: Augmented system for H_∞ problem

So that,

$$\begin{bmatrix} z1 \\ z2 \\ z3 \\ y \end{bmatrix} = \left[\begin{array}{c|c} W_S & -W_S P \\ 0 & W_U \\ 0 & W_T P \\ \hline I & -P \end{array}\right] \begin{bmatrix} w \\ u \end{bmatrix} \tag{2.19}$$

where P is the plant; W_S, W_U and W_T are weightings for primary, control effort and complementary sensitivity functions respectively; I is the identity matrix. The corresponding augmented system block diagram of a typical negative feedback system is shown in Figure 2.6. As mentioned before, our aim is to solve the closed loop transfer function $\|T_{zw}\|_\infty < \gamma$. It can be formulated in terms of G using lower fractional transformation (LFT) F_l:

$$T_{zw} = F_l(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21} \tag{2.20}$$

Then our aim is to solve for the controller subject to a constraint on the LFT formation using γ

$$\|F_l(G, K)\|_\infty < \gamma \tag{2.21}$$

In the previous multi-sensitivity example with primary, control effort and complementary sensitivities, the closed loop transfer function (matrix) inequality can be written as:

$$\|T_{zw}\|_\infty = \|F_l(G, K)\|_\infty = \left\| \begin{bmatrix} W_S S \\ W_U U \\ W_T T \end{bmatrix} \right\|_\infty < \gamma \tag{2.22}$$

This inequality is equivalent to the robust stability conditions listed in the previous section. Thus the inequality 2.22 guarantees the stability of the system by the constraint $T_{zw} < \gamma$. To satisfy the robust stability conditions, if the closed loop system is stable, the ∞-norm of the weighted sensitivities should be below than 1, that is $\|T_{zw}\|_\infty$ or $\|F_l(G, K)\|_\infty < 1$ in this case. To solve such an inequality and thus obtain the controller K, the most popular solver is by the algebraic 2 Riccati equations. Now there are many

commercially available software packages, such as that of the MATLAB Robust Control Toolbox [2], which can be used to synthesise the controllers quickly and accurately.

2-Riccati Equations Approach

In 2 Riccati equations approach to solve the equation 2.22, the plant G should be represented as following state-space realization

$$G = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \tag{2.23}$$

Then the following additional conditions are assumed to be guaranteed in advance as well:

- (A, B_2) is stabilisable and (C_2, A) is detectable.
- (A, B_1) is stabilisable and (C_1, A) is detectable.
- $C_1^T D_{12} = 0$ and $B_1 D_{21}^T = 0$
- D_{12} has full column rank with $D_{12}^T D_{12} = I$ and D_{21} has full row rank with $D_{12} D_{12}^T = I$

If these assumptions have been made for the plant, the two Riccati equations are then to be solved. They are:

$$A^T X_\infty + X_\infty A + C_1^T C_1 + X_\infty(\gamma^{-2} B_1 B_1^T - B_2 B_2^T) X_\infty = 0 \tag{2.24}$$

$$A^T Y_\infty + Y_\infty A^T + B_1 B_1^T + Y_\infty(\gamma^{-2} C_1^T C_1 - C_2^T C_2) Y_\infty = 0 \tag{2.25}$$

Then solve for $X_\infty \geq 0$ with the first Riccati equation, which provides a minimum phase (MP) solution; solve for solved for $Y_\infty \geq 0$ with the second Riccati equation, which provides a stable solution. Then the central controller K_c can be parametrized as

$$K_c = \left[\begin{array}{c|cc} A_\infty & -Z_\infty L_\infty & Z_\infty B_2 \\ \hline F_\infty & 0 & I \\ -C_2 & I & 0 \end{array}\right] \tag{2.26}$$

where

$$\begin{aligned} A_\infty &= A + \gamma^{-2} B_1 B_1^T X_\infty + B_2 F_\infty + Z_\infty L_\infty C_2 \\ F_\infty &= -B_2^T X_\infty \\ L_\infty &= -Y_\infty C_2^T \\ Z_\infty &= (I - \gamma^{-2} Y_\infty X_\infty)^{-1} \end{aligned}$$

Mixed Sensitivity Problem

The so-called mixed sensitivity problem is to formulate the two sensitivities S and T with their weighting functions into a single H_∞ norm cost function. The problem is then expressed as:

$$\left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty \leq 1 \tag{2.27}$$

where W_S and W_T are weighting functions for sensitivity S and T respectively. This problem is called the standard mixed sensitivity problem. Due to the constraints on sensitivity S and T over the full frequency range, solving mixed sensitivity problem ensures satisfaction of the basic requirements of performance and robustness. However this cost function cannot be solved exactly easily. Designers usually solve the alternative approximate equation:

$$\left\| |W_S S|^2 + |W_T T|^2 \right\|_\infty \leq \frac{1}{2} \tag{2.28}$$

This alternative formulation is equivalent to the formulation:

$$\left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty \leq \frac{1}{\sqrt{2}} \tag{2.29}$$

and so is slightly conservative compared to equation 2.27.

2.3.4 LMI approach

To solve the the H_∞ (sub-optimal) problem, an alternative method is Linear Matrix Inequalities (LMI) approach. Instead of using Riccati equations the LMI approach are formulated as three matrix inequalities, which are derived from 2 Riccati equations. Similarly, the positive definition solution is used to parametrise sub-optimal controller. Actually, the LMI approach solve the 2 Riccati problem in that singular solutions. Many control problems can be solved by LMI approach, such as H_∞ and H_2 control problem; H_∞/H_2 optimal control problem; pole placement problem; robust predictive control problem. Currently MATLAB Robust Control Toolbox [2] has integrated the LMI approach as its function. Designers can solve such problems easy and quickly. The details of LMI approach can be found in [74].

2.3.5 H_∞ Loop Shaping

The loop shaping technique is to find a controller shaping the open loop system $L = KP$ to satisfy the specific requirements. In H_∞ control theory the closed loop transmissions like S or T are bounded or shaped, but the loop shaping technique is concerned for

the open loop transmission. Because the sensitivity functions can also be represented in terms of the open loop system L by:

$$\begin{aligned} S(s) &= [1 + L(s)]^{-1} \\ T(s) &= 1 - S(s) = 1 - [1 + L]^{-1} \\ U(s) &= K(s)[1 + L(s)]^{-1} \\ V(s) &= P(s)[1 + L(s)]^{-1} \end{aligned}$$

the weighting functions for those sensitivities can be translated to the shaping requirements for the open loop system L. In SISO control problems, the typical shaping requirements are:

- $|L(j\omega)| > 1$ for high loop gain at low frequency.
- $|L(j\omega)| < 1$ for low loop gain at high frequency.
- $L(j\omega)$ to satisfy the Nyquist criterion.

More requirements for MIMO system can be found in [63]. In the MIMO problem, the requirements are given in terms of the maximum and minimum singular values of open loop transmission across the full frequency range.

Recently, a synthesis technique combining the classical loop shaping and H_∞ robust stabilisation was developed by McFarlane and Glover [63]. They proposed pre and post-compensators W_1 and W_2 for plant P, so the plant is shaped by W_1 and W_2 initially giving:

$$P_s = W_2 P W_1 \tag{2.30}$$

Assuming that the plant G has a normalised left coprime factorization $P_s = M_s^{-1} N_s$, where N_s and M_s are stable rational transfer functions, the closed loop system is robustly stable if and only if the nominal (unperturbed) feedback system is stable and

$$\gamma = \left\| \begin{bmatrix} K_s \\ I \end{bmatrix} (I - P_s K_s)^{-1} M_s^{-1} \right\|_\infty \leq \frac{1}{\epsilon} \tag{2.31}$$

where the ϵ is the stability margin. The robust stabilising problem is then to find the maximum stability margin ϵ_{max} [63], which is:

$$\epsilon_{max}^{-1} = \gamma_{min} = \left\{1 - |||[N_s M_s]|||_H^2\right\}^{-\frac{1}{2}} = (1 + \rho(XZ))^{\frac{1}{2}} \tag{2.32}$$

where $||.||_H$ denotes the *Hankel* norms, ρ denotes spectral radius (maximum eigenvalue), X and Z are the unique positive definite solutions for 2 Riccati equations (plant realised by state-space form (A,B,C,D)):

$$\begin{aligned} (A - BS^{-1}D^T C)Z + Z(A - BS^{-1}D^T C)^T - ZC^T R^{-1} CZ + BS^{-1}B^T = 0 \\ (A - BS^{-1}D^T C)^T X + X(A - BS^{-1}D^T C) - XBS^{-1}B^T X + C^T R^{-1} C = 0 \end{aligned}$$

where $R = I + DD^T$,$S = I + DD^T$. Then the central controller K_s can be formulated from solutions X and Z. The controller K can be obtained with pre and post-compensator W_1 and W_2 as:

$$K = W_1 K_S W_2 \tag{2.33}$$

Because the open loop specifications are easy to formulate, the loop shaping technique is attractive and employed in many applications.

2.3.6 H_∞ Optimisation Over Analytic Functions

The general Riccati based H_∞ control method solves the H_∞ sub-optimal problem. To obtain an optimal solution for the H_∞ problem, the $\gamma = ||T_{zw}||_\infty$ should be minimised. To do this, a direct method is the γ-iteration which is to find the lowest γ using a binary search. Most H_∞ optimal control methods including γ-iteration are based on the state-space realization for both plant and controller. Helton [40] proposed a method which is an analytic function optimisation method which solves H_∞ optimal problem in terms of the frequency response function[40].

The analytic function optimisation method is an extension of an older approach called the 'disk method'. The so-called 'disk problem' can be described as follows: consider a circular function on frequency domain

$$\frac{1}{R(e^{j\theta})^2}|K(e^{j\theta}) - H(e^{j\theta})|^2 \leq 1 \tag{2.34}$$

where $\theta = 2\pi(i-1)/N$, N is the number of θ, $i = 1, ..., N$. $e^{j\theta}$ can be transformed to or from $j\omega(e^{j\theta} \leftrightarrow j\omega)$. At each frequency, the constraint function is bounded by a circular shape in the complex plane. The $R(e^{j\theta})$ and $K(e^{j\theta})$ can be considered as the radius and centre of the circle. $H(e^{j\theta})$ is a function bounded in this circle. The weighted sensitivity of H_∞ control problem can be formulated as this type circular function, for example $|1 - T(e^{j\theta})|^2 \leq W_S^{-1}(e^{j\theta})^2$. The H_∞ optimisation for disk problem can be formulated as follows:

$$\gamma = \inf_{H \in H_\infty} \sup_{\theta} \frac{1}{R(e^{j\theta})^2}|K(e^{j\theta}) - H(e^{j\theta})|^2 \tag{2.35}$$

where the minimizer $\gamma \leq 1$. The H_∞ optimisation for disk problem is accomplished by solving the Nehari problem for H, The so-called Nehari problem [67] is to find the shortest distance τ of $k \in L_\infty$ to H_∞:

$$dist(k, H_\infty) : \tau = \inf_{f \in H_\infty} ||k - f||_\infty \tag{2.36}$$

To form a standard Nehari problem, equation 2.35 can be reformulated by finding the spectral factor α for $R^2 = \alpha \cdot \alpha$, then $k = K/\alpha$, $f = H/\alpha$. The shortest distance τ can

be found by computing the negative Fourier coefficients of k to form the Hankel matrix H_k and solving Hankel approximation problem, meanwhile the corresponding solution for $f \in H_\infty$ is also obtained. Helton and Schwartz presented such method which solves Nehari problem for disk problem by means of a solution of the Nehari problem as a frequency response function[42].

As an extension of the disk method, analytic function optimisation method can be used to solve the non-disk problem, such as the standard mixed sensitivity problem. To solve more general problems, the 'disk iteration' algorithm [39]is used in analytic function optimisation method. This method constructs a Taylor expansion for the objective function. Through completing the perfect square in Taylor expansion, a quasi-disk problem is obtained. Then solving the Nehari problem for this quasi-disk problem using the method mentioned above, a local minimizer and corresponding descent direction can be found. The Taylor expansion and perfect square form are introduced in the following. As the disk iteration is adequate for both scalar and vector case, this algorithm is introduced in vector case. Assume that the current solutions guesses are $f^k = f_1^k, \ldots, f_N^k$ which are analytic on the unit disk at the k-th iterative step. The N is the number of the variables, $N = 1$ for scalar case. Assume that the performance function is $\Gamma(e^{j\theta}, z)$, then at k-th iteration the value of the performance function is $\Gamma(e^{j\theta}, f^k)$. Consider a new guess $f^k + h^k$, let $\Gamma(e^{j\theta}, f^k + h^k)$:$C \to R$. Then the Taylor approximation of $\Gamma(e^{j\theta}, f^k + h^k)$ is:

$$\Gamma(e^{j\theta}, f_l^k + h_l^k) = g^k + \sum_{l=1}^{N} 2Re(a_l^k h_l^k) + \sum_{l=1}^{N} t_l^k \left|h_l^k\right|^2 \tag{2.37}$$

Where:$g^k = \Gamma(e^{j\theta}, f^k)$, $a^k = \partial\Gamma/\partial z(e^{j\theta}, f^k)$, $t = \partial^2\Gamma/\partial z\partial\overline{z}(e^{j\theta}, f^k)$. To construct a disk type problem, the Taylor approximation of the Γ function should be converted to obtain a 'perfect square' quadratic in equation 2.37. First, the Taylor approximation is converted to:

$$\delta_m^k = \inf_{h \in H_\infty} \sup_{\theta} \left[g^k - \sum_{l=1}^{N} \left|a^k\right|^2 / t_l^k + \sum_{l=1}^{N} t_l^k \left|\overline{a}_l^k / t_l^k + h_l^k\right|^2 \right] \tag{2.38}$$

where δ_m^k is the minimizer. Then the Taylor approximation can be converted to a perfect square form by introducing λ:

$$\lambda = \inf_{h \in H_\infty} \sup_{\theta} \sum_{l=1}^{N} \left[t_l^k / \left(\delta_m^k - g^k + \left|a^k\right|^2 / t_l^k \right) \right] \cdot \left|\overline{a}_l^k / t_l^k + h_l^k\right|^2 \tag{2.39}$$

When $\lambda = 1$, equation 2.39 is same as equation 2.38. The disk iteration has an outer loop which moves from f^k to f^{k+1} and an inner loop which searches δ_m^k to find a λ which is close enough to one, the corresponding δ_m^k is the local minimizer γ in this (outer) loop. By solving the Nehari problem same as disk problem, the descent direction $h_l^k \in H_\infty$

is found. A line search scheme is applied to the disk iteration, the estimates f_l^k are tested with the pre-set stop criterion. If f_l^k satisfies the criterion, stopping loop, current estimates f_l^k are the optimal solutions; otherwise f_l^k are updated as $f_{l+1}^k = f_l^k + h_l^k$ until the optimal solutions satisfy the stop criterion[38]. The details of the disk iteration algorithm can be found in [39].

Chapter 3

Experimental Setup

3.1 Introduction

In this chapter of the chapter the experimental engine and the corresponding hardware/software setups are introduced. The experimental work were carried out in the powertrain control laboratory at the University of Liverpool. The experimental engine was modified to allow the users to control the air bleed valve (ABV), spark and fuel injector. Various transducers were installed to the experimental engine. The output data, such as engine speed, air fuel ratio, etc., is collected by a standard desktop PC running with the calibration software. The details of configurations for the experimental engine are discussed in the following.

3.2 Engine and Dynamometer Specification

In the University of Liverpool powertrain testing lab, the experiment engine is a Ford's 1.6 litre Zetec petrol engine. This is a convetional four-stroke, four cylinders, 16 valves, double overhead cam (DOHC), Multiple-point Fuel Injected (MPFI) and water cooled Spark Ignition (SI) internal combustion (IC) engine. Each of the four cylinders have two inlet valves and two exhaust valves. The air flow is delivered by ABV; the fuel is injected by the electronic port fuel injectors. The specifications, dimensions and components of the engine are shown in Table 3.1.

Originally, the experimental engine was calibrated for idle speed control testing. For this study, the engine has been modified. So that the actuators - air bleed valve, spark, fuel injector can be all controlled. The control signal for ABV is a pulse width modulation (PWM) signal, the opening of ABV can be adjusted by changing the duty cycle of the PWM signal in % units. The fuel injector is swithed by a pulse signal, high level signal for injector open, low level signal for injector off. So the fuel mass

Cylinder Bore		76.0 mm
Connecting Rod Length		136.2 mm
Compression Ratio		10.3:1
Cubic Displacement		1597cm3
Stroke Length		88 mm
Firing Order		1-3-4-2
Exhaust Valve Opens		64^o B.T.D.C.
Exhaust Valve Closes		12^o A.T.D.C.
Inlet Valve Opens		22^o B.T.D.C.
Inlet Valve Closes		12^o A.T.D.C.
Maximum Torque		138 Nm at 3500 rpm
Maximum Power		90 PS
Idle speed		880 rpm
Fuel Unleaded RON		95 petrol

Table 3.1: Engine Specifications

can be adjusted by changing the fuel pulse width (FPW) signal in μs units. The extra load (brake torque) can be applied through a low inertia DC dynamometer which is a 7.5 kW DC motor and coupled with the engine's crankshaft rigidly. The dynomoeter can work under the voltage control mode or speed control mode. Several sensors are installed on the engine, such as engine speed in RPM units, in-cylinder pressure, mass air flow (MAF), manifold absolute pressure (MAP), torque and etc. Figure 3.1 shows a schematic diagram of the the engine and the key sensors.

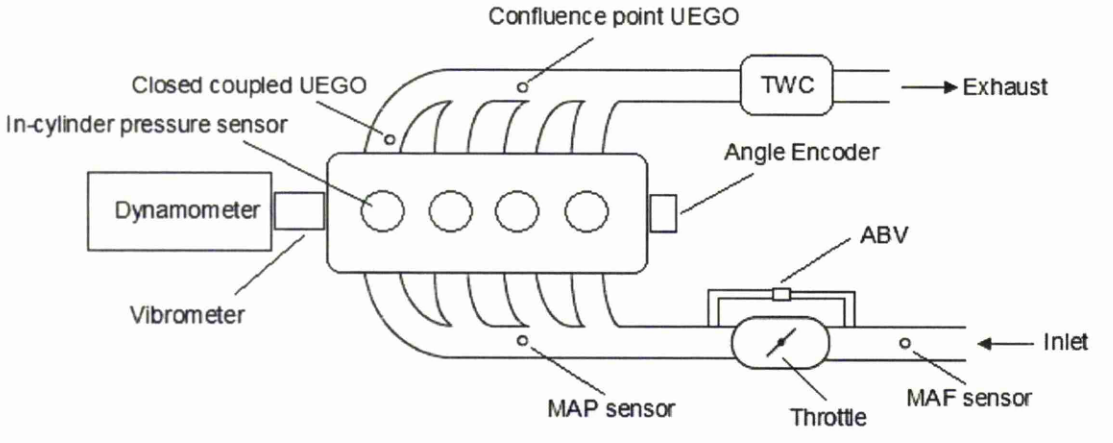

Figure 3.1: Schematic of the the engine and the key sensors

3.3 Engine Sensors

3.3.1 Angle encoder

The optical angle encoder is mainly used to determine the crank angle of the engine. The angle encoder is placed directly onto the crank shaft. It outputs two signals: one pulse per 1^o and one pulse per 360^o. The 1^o pulses are transmitted to the DS4001 card which generates a hardware interrupt for the dSPACE tasks. So that dSPACE tasks is triggered by each 1^o pulse. In another word, the algorithms and models created in SIMULINK are event based, this is important for the engine calibration research. The 360^o pulse are synchronised pulse at the TDC of cylinder 1, checking if any 1^o pulses have been missed and resetting TDC in dSPACE if a discrepancy occurs.

3.3.2 Air-Fuel Ratio (Oxygen) Sensors

There are two types of the Oxygen sensors installed on the engine. One is a heated exhaust gas oxygen sensor (HEGO), and another one is the universal exhaust gas oxygen (UEGO) sensors. The HEGO sensor is used for emission control for the Ford's EEC IV EMS. The additional UEGO sensor is also installed at the confluence point where is a typical region for mounting an oxygen sensor. Also known as wideband sensors or lambda sensors, the (linear) UEGO sensor provide a more accurate air-fuel ratio (AFR) determination than the narrowband (nonlinear) HEGO sensor. But compare to the HEGO sensor, the UEGO is more expensive. The Bosch wideband O^2 sensor is used in this work. It provides an output voltage proportional to the AFR. For the gasoline port fuel injection (PFI) engine, the air to fuel (by mass) ratio is 14.7:1 at stoichiometric conditions. This ratio of air-fuel mixture can also be described in terms of λ:

$$\lambda = \frac{AFR_{actual}}{AFR_{stoichiometric}}$$

where λ is the ratio of measured air-fuel ratio relative to stoichiometric air-fuel ratio.

3.3.3 MAP and MAF Sensors

A manifold absolute pressure (MAP) sensor is fitted in a low pressure (less than 1 bar atmospheric) hose on the inlet manifold. The manifold pressure information can give an indication of the mass of air entering the cylinders. Usually the MAP signal can be used for torque control, because the MAP signal is correlated highly to the load applied to the engine. When the throttle is opened widely or the high load is applied to the engine, the low pressure (between 0 and 1 bar atmospheric) decreases; when throttle

is opened smaller or the low load is applied to the engine, the low pressure increases. Additionally, the MAP signal can be used for fuelling control, because the request of fuel is directly determined to the mass of air entering the cylinders. It was found that the MAP signal integrated over 180^o in identification gives most suitable values when used for information analogous to engine load.

An alternative air mass sensor is a mass airflow (MAF) sensor which provides the information of the air mass entering the engine upstream of the throttle.

3.3.4 Torque and Engine Speed Sensors

A torque Vibrometer (Torquemaster TM112 200Nm rating) torque transducer is mounted on the engine/dynamometer coupling. It provides analogue torque outputs for dSPACE ADC card. A tachometer coupled to the Vibrometer torque transducer is used for measuring the engine speed. The optical angle encoder can also be used to measure the engine speed by counting the crank angle per unit time.

3.4 Engine Input

3.4.1 Air Bleed Valve

In this work, the air bleed valve (ABV) is used to replace the throttle to regulate the air flow. The ABV is a throttle bypass valve that is used for controlling idle speed. The ABV is driven by the pulse width modulation (PWM) signal at a frequency of approximately 300Hz. The opening percentage is directly related to the duty cycle of the driving PWM signal, as compared to the conventional throttle, which is cable driven, the ABV is electronic valve. The ABV can not only be considered as a electronic throttle but also a throttle position sensor. In this work, the ABV is not only used to control the engine (idle) speed but also to control the air-fuel ratio and torque of the engine.

3.4.2 Fuel Injection

The engine used in this work is a port fuel injection (PFI) engine. Fuel is injected from the injector toward the inlet ports of the cylinders before the inlet valve opens. Then the fuel in the manifold evaporates and mixes with air. When the inlet valve opens, the air fuel mixture is drawn into the cylinders. Because the fuel injected before the inlet valve opens, this causes a puddle of fuel in the inlet manifold, not all fuel evaporates and drawn into the cylinders. Such fuel puddle dynamics is important for

fuelling control strategy.

The amount of fuel injected into the engine is determined by the time period of the injection. The fuel injector is driven by the fuel pulse width (FPW) signal, which controls the time of the injector stays open. The fuel injected to the engine can be controlled by two variables: injection-start timing and injection fuel pulse width. Because In PFI engine the fuel entering the cylinder relies on the fuel puddle dynamics, the injection-start timing is not critical. In air-fuel ratio control application of the PFI engine, the injection-start timing is fixed commonly, the fuelling is controlled the FPW signal only.

3.5 Calibration Hardware and Software

Figure3.2 shows the hardware connection between the experimental engine, dSPACE unit, dsPIC microcontroller and the desktop PC. The ABV, fuel injector and spark of the engine are controlled by a Ford production EEC IV EMS. However they can also be switched to be controlled by a powerstage unit which is connected to dSPACE. All signals measured from engine's transducers are transmitted to the dSPACE unit. The control signals are issued from dSPACE to the engine.

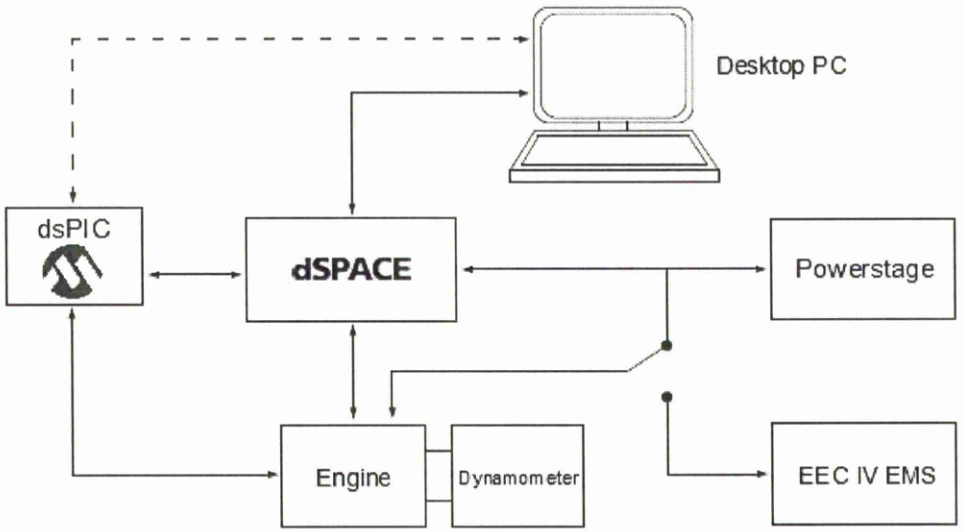

Figure 3.2: Hardware setup

The components of dSPACE used in this work include DS1005 real-time interface

card which is the main processing unit of the dSAPCE unit; DS4001 digital outputs card; DS4002 digital input/output (I/O) card; DS2003 analogue to digital conversion (ADC) card; DS4302 CAN interface card. The hardware of dSPACE unit cooperates with the ControlDesk and MATLAB/SIMULINK software running on a desktop PC. The ControlDesk is a fast prototyping software for engine calibration or some other control designing application. This software provides the established SIMULINK models corresponding to the dSPACE hardware units. To design and simulate the powertrain control system, the algorithms and models can be created by the MATLAB/SIMULINK model associated with these dSPACE units' models off-line. The whole SIMULINK models can be compiled to C code and dSPACE code through Real-Time Workshop and the special compiler for dSPACE. Then, the code can be implemented to the dSPACE directly from PC, which can run the tasks in real time. The ControlDesk also provides the on-line graphical interface and data collecting tool for the users to monitor and capture the experimental data.

To implement the controller in embedded system, the dsPIC33F microcontroller from Microchip is used. It is a 16-bit digital signal processor (DSP) and mounted to a dsPICDEM 1.1 Plus demonstration board. The dsPIC is programmed using the Microchip's target device SIMULINK blocksets which has been introduced in chapter 2. Besides this software, the MPLAB IDE development software and MPLAB ICD 2 debugger/programmer are also used for debugging or programming.

3.6 CAN-bus Network Configuration

In this section the configurations for a CAN-bus network which allows the communication between embedded system and engine are presented. The idle controller implemented in the embedded system is to control engine speed via the CAN-bus. The details of controller design and implementation will be discussed in the Chapter 5.

Based on the engine and electronics configuration in the University of Liverpool Powertrain Lab, this CAN-bus based control system is outlined as follows. First, the engine speed is measured by an analog sensor, the engine speed signal is an analog signal which is to be converted by dsPIC's ADC module; the converted digital engine speed signal is fed back to the idle speed controller; the resulting control signal for ABV duty will be transmitted to dSPACE via CAN-bus; the ds4001 I/O card of dSPACE is then used to output a PWM signal driving the ABV actuator to regulate the engine speed. As an existing configuration, the dynamometer is controlled by dSPACE. It can apply the extra load to simulate the disturbance for the engine during experimental test. Figure 3.3 shows the control system loop.

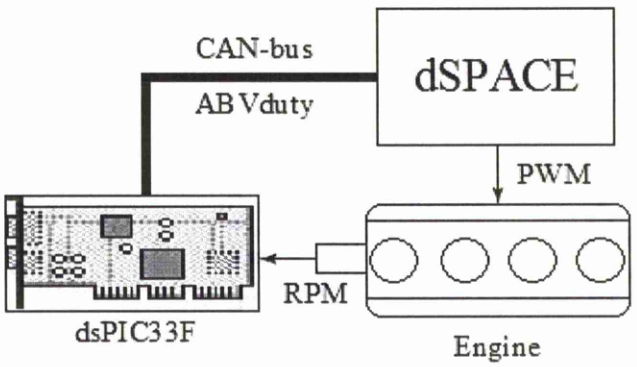

Figure 3.3: Idle speed control system

3.6.1 Fast Prototyping for Embedded System

In this application, a novel fast prototyping method for embedded system was also used. This method is based on SIMULINK blocksets and models. Through converting the SIMULINK blocksets and models to machine code directly, the programming for embedded system is essentially a graphic based process. This method is easier and faster than the conventional language based programming method. In this work, the dsPIC33FJ256GP710 from Microchip corporation was used. The SIMULINK blocksets for dsPIC30/33 from Microchip was used.

Basic Setup For Programming Tool

To setup the fast prototyping tool three steps are carried out. First, the Simulink blockset for the target device is installed. The blockset for dsPIC provided by Microchip was used in this work. After installation, the blocksets for dsPIC is found in the Simulink library. Designers drag the blockset to the Simulink model for programming directly.

The blocksets provided by Microchip can be converted to C code through the Matlab Real-Time Workshop (RTW) software. To compile the C code to the machine code (HEX) of dsPIC, the specific compiler for translating C to HEX code must be used. In this work, the C30 compiler for dsPIC from Microchip was used. After the SIMULINK blockset was installed, the C30 compiler was also installed. The compile command and parameters for C30 can be setup in the Real-Time Workshop option through entering the 'Configuration Parameters' of SIMULINK. From this option, software designers also can check if the C compiler is installed correctly.

After the SIMULINK blockset and C compiler were installed the programmer which is used to code to the target device is installed. Because dsPIC microcontroller was

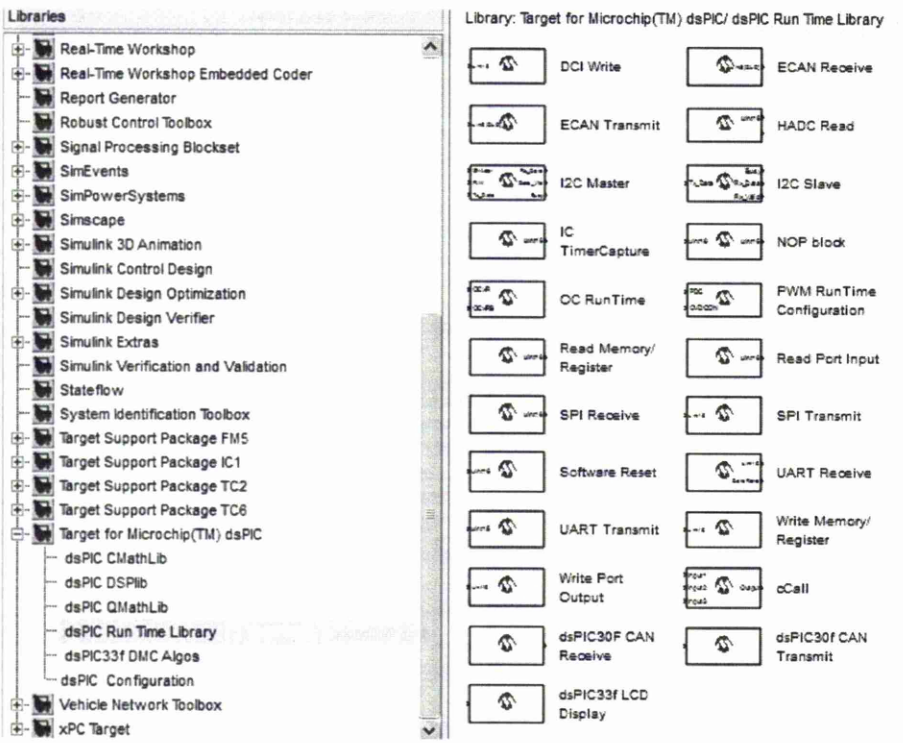

Figure 3.4: The target blockset in Simulink library

used, the MPLAB IDE which is also from Microchip was selected as the assembler. The MPLAB IDE software is a powerful development tool for embedded applications employing Microchip's PIC and dsPIC microcontrollers[15].

Create A Model Using SIMULINK Blockset

Once the fast prototype software is installed, designers can program for the dsPIC using SIMULINK directly. Creating a SIMULINK model using specific blocksets as mentioned above is equivalent to writing a program for the target device. The blocksets provided by Microchip can do; the configuration for modules and peripheral I/Os of dsPIC; the running time read/write operations for Special Function Registers (SFRs); the special calculations for the Digital Signal Process (DSP) engine, etc.. Those blockset and some SIMULINK's default blocksets can be used in the programming according to the different applications.

3.6.2 CAN-bus Protocol

In this work, the control signal of the idle speed controller was transmitted by CAN-bus. In the following part, an outline of CAN-bus is presented.

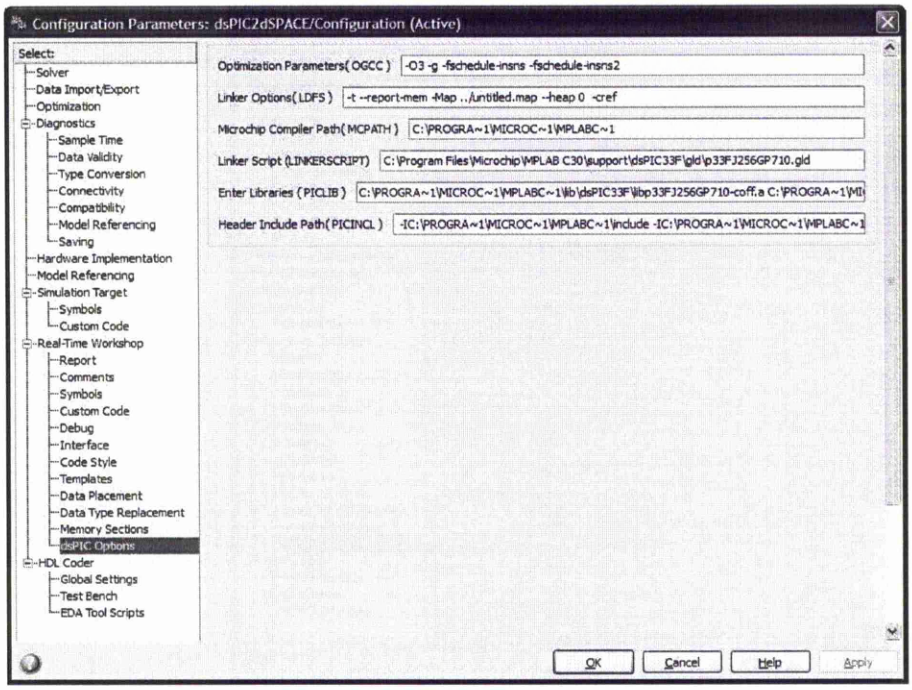

Figure 3.5: Configuration for C30 compiler

CAN-bus Communication

'CAN' is short for 'Controller Area Network', which is a serial communications protocol developed by Robert Bosch GmbH initially. The official standardisation of the CAN protocol was first published by Society of Automotive Engineers (SAE). The current version of CAN is ver2.0, which was released in 1991. The CAN-bus is a standard data bus for vehicle systems in the automotive industry. Many electronic devices and actuators onboard vehicles are now connected by the CAN-bus. As a serial type protocol, the data transmission for CAN-bus is more efficient than a parallel communications protocol. The data is packed as a 'message' or 'frame' transmitted bit by bit over the CAN-bus. The bit rates for the CAN-bus is up to 1 Mbit/s. Each frame starts with a Start of Frame (SOF) bit and stops at an End of Frame (EOF) bit. The sequence of bit fields in each frame is: SOF bit→Arbitration filed→Control field→Cyclic Redundancy Check(CRC) field→Acknowledgement(ACK) field→EOF bit.

According to the different purpose of the message, the frames are divided into several types: data frame, remote frame, error frame, overload frame, etc.. The data frame and remote frame are the most important frames. The data frame contains the data transmitted from one node on the bus to another node. The remote frame is transmitted from one node on the bus to another node to request the data be transmitted back to the original node. In the CAN 2.0B protocol, there are two typs of data frame: standard data frame and extended data frame. In the standard data frame, the arbitration

filed contains the 11 bit identifier; the control filed contains the 4 bits data length code(DLC); the data filed contains 8 bytes of data. In the extended data frame, the arbitration filed contains the 29 bit identifier.

To transmit the data, the complete data frame is stored in the message buffer and transmitted according to the transmission priority. There are 32 message buffers which are located in DMA RAM for data transmitted or received on the CAN-bus. Message buffers 0-7 can be used for either transmitting or receiving buffers. But message buffers 8-31 can only be configured for receive operation.

To receive the data, the received message is stored in the message assembly buffer first. The identifier of the message is compared with 16 acceptance filters. If the identifier completely matches any acceptance filter, a DMA transfer request is sent from the ECAN module to the DMA Controller, the received message is then transferred to the corresponding Message buffer.[30]

3.6.3 Configurations of dsPIC

In the idle speed control application, the dsPIC33FJ256GP710 was installed on the microchip dsPICDEM 1.1 Plus demo board. All required wiring work was carried out on the demo board. The software configuration of the dsPIC are detailed as follows.

Basic Configuration for Simulink Model

First of all, a Simulink model file was created. In this model, 'config' blocksets were added prior to any other blocks. To translate the Microchip's blockset to C code, Microchip provide the system target file - 'dsPIC_stf.tlc' for the Real-Time Workshop software. Programmers should set this system target file correctly before compilation. Advance programmers can browse the set the correct target file in the Real-Time Workshop option of 'Configuration Parameters' in the Simulink model. For beginners, Microchip provides a simple way to do that. The designers can also double-click the 'config' blockset to set the correct '.tlc' file for the system target file option in Real-Time Workshop.

Oscillator Configuration

This configuration was carried out in the 'dsPIC33fxx Main' blockset. On the dsPICDEM 1.1 Plus demo board, a 7.37MHz crystal oscillator is provided for dsPIC. Then in the 'dsPIC33fxx Main' blockset, the input frequency (FIN) should be '7372800'(Hz). To easily program for a CAN-bus module the instruction cycle FCY=16MHz (16MIPS) was

selected for this work. Because the FCY=FOSC/2, the system clock FOSC=32025600Hz was to be obtained equally. To obtain that required frequency from the 7.37MHz crystal, the PLL must be enabled. The dsPIC33F series integrates a complex PLL block. The structure is displayed in figure 3.6. By means of the block system shown in that

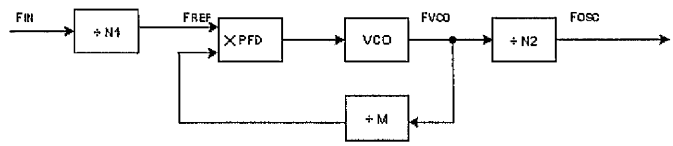

Figure 3.6: dsPIC33F PLL block diagram[16]

figure, the FIN inputting for Phase Frequency Detector (PFD) is divided by pre-PLL Clock Divisor (N1=2,3,...,33), then the Voltage Controlled Oscillator (VCO) output frequency (FVCO) is scaled down by means of a Feedback Divisor (M=2,3,...,513) for feeding FVCO back to the PFD, FVCO = M×FREF. Finally, FVCO is divided by post-PLL Clock Divisor (N2=2,4&8). In dsPIC33F, the PFD input frequency (FREF) is set between 0.8 to 8.0MHz and FVCO is set between 100 to 200MHz. Thus the system clock FOSC is limited to below 80MHz. The calculation of FOSC is:

$$FOSC = FIN \times \frac{M}{N1 \times N2} \tag{3.1}$$

Because 7.37MHz is in the medium-speed range (3-10MHz), the primary oscillator was set to the medium-speed oscillator (XT) Mode. From the Primary Oscillator Clock Source option, 'Primary Oscillator with PLL: Medium-Frequency Mode (XT PLL)' was also selected. It was found that when FIN = 7372800Hz, M = 139, N1 = 4 and N2 = 8, the system clock FOSC=32025600Hz and FCY=16012800Hz ≈16MHz. In 'dsPIC33fxx Main' blockset, the value of FCY can be set by the programmer directly, and there is no need to write the HEX code for divisors (N1,N2& M)register. The main loop time for the microcontroller was then set, at $30ms$ as determined by the controller design requirement. 'Timer 1' was selected as the time source for counting loop time. The detailed setup is shown in figure 3.7 In this work, the 'watchdog' mode and 'code protection' mode were not enabled. More details for configuration of watchdog and code protection can be found in [18], [12] and [10].

Analog-to-Digital Converter Configuration

Because the engine speed signal is an analog signal measured by the speed sensor which is mounted on the engine's crank shaft, the input analog signal must be converted to a digital signal for the dsPIC. So the ADC module of dsIC33F was used.

Corresponding to any variation of engine speed, the sensor returns an analog voltage

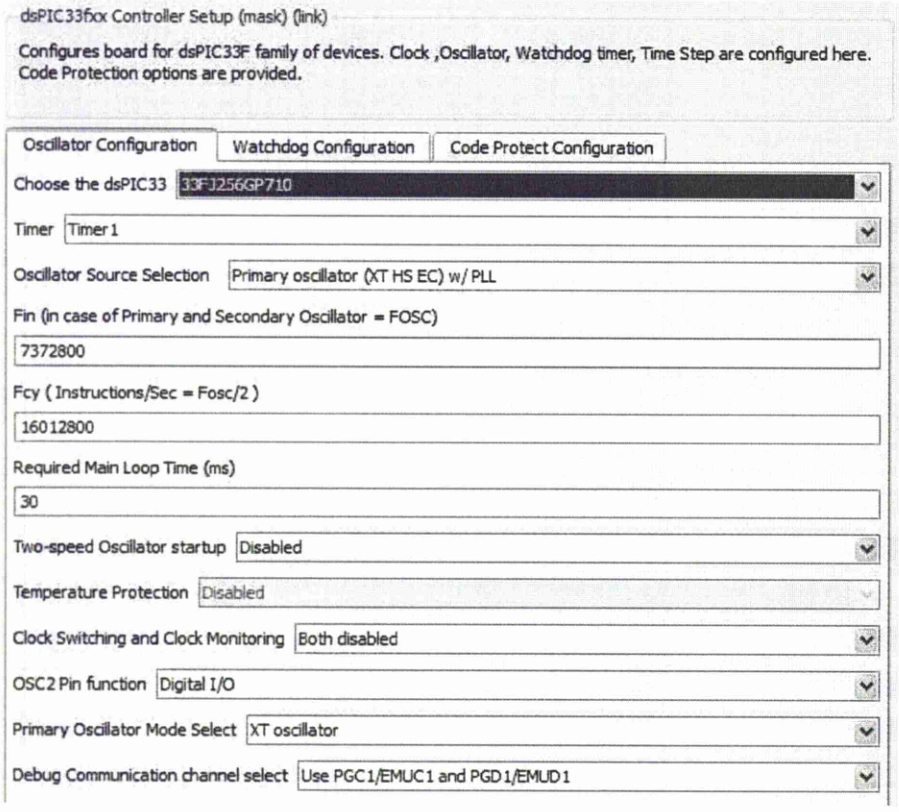

Figure 3.7: Oscillator configuration

signal, where 10 volts analog signal is equivalent to 5000rpm approximately. For the idle speed control problem, the nominal engine speed is 880rpm. As required, the maximum deviation for the regulated idle speed is ±320rpm. Therefore the maximum engine speed is around 1200rpm which is equivalent to 2.4 volts. As mentioned before the operating voltage of the dsPIC33F is 3.3 volts(positive supply for peripheral I/O pins VDD=3.3 volts, ground reference VSS=0). For the analog I/O pins, a positive supply is AVDD=VDD±0.3 volts; ground reference AVSS=VSS±0.3 volts. So the maximum engine speed which gives a safe input voltage for dsPIC33F's ADC module is about 1800rpm. Obviously, the engine speed in the idle case hardly ever to achieves 1800rpm and is never over 3000rpm even the ABV is fully opened. Accordingly the analog signal of engine speed did not need to be scaled down and the demo board was wired to connect with the engine speed sensor's output directly.

In programming, the 'ADC Config' blockset provided by Mirochip was dragged to the SIMULINK model to config are the ADC module. The dsPIC33FJ256GP710 incorporates 2 ADC modules. Because the engine speed took only one (analog) input in this case, only one ADC module was needed. Thus 'ADC 1' was selected, and the 'ADC Operational Mode' was also selected as the '12-bit, single-channel ADC' mode for higher resolution and 'AN1' was selected as the analog input in the ADC Config

blockset. As the engine speed is a positive number, the output data format of ADC was selected as 'Integer(Dout=0000 dddd dddd dddd)'. In 12-bit single channel mode, only the channel CH0 can be used by the programmer. In 'Select Channels' option, 'convert CH0' was chosen.

The voltage references for AD conversion are the voltages used to compare with the input voltages, the ADC module outputs the digital numbers which represent the percentage comparing results of input voltages and references voltages. In 10-bit mode, the maximum digital number returned by the ADC module is 1023 (which represents 100%); in 12-bit mode, the maximum number is 4095 (which represents 100%). For the dsPIC33F series chip's ADC module, the voltage reference high (VREFH) and the voltage reference low (VREFL) can be supplied from the internal references AVDD and AVSS voltage rails or the external references VREF+ and VREF- input pins[17]. The voltage references VREFH/VREFL can be supplied from different combination of AVDD/AVSS and VREF+/VREF-.

VCFG<2:0>	VREFH	VREFL
000	AVDD	AVSS
001	VREF+	AVSS
010	AVDD	VREF-
011	VREF+	VREF-
1xx	AVDD	AVSS

Table 3.2: Voltage reference selection

where the VCFG<2:0> are the Voltage Reference Configuration bits in the ADC Control register. For language based programming, programmers should configure the voltage reference by writing the binary code of Table 3.2 to corresponding configuration bits. For fast prototype programming, programmers only need to select the correct options in the ADC Config blockset. According to the circuit (hardware) configuration for ADC module on the dsPICDEM 1.1 Plus demo board, the ADC Config blockset in the 'Convertor Voltage Reference Configuration' option was configured as 'ADREF+=Avdd;ADREF-=Avss'.

The detailed configuration of the ADC module is displayed in figure 3.8. In 'Alternate Input Sample Mode Select' option, sample A was selected. Then the 'Positive Input Selection for Channel 0 Sample A' was selected as 'AN1'. The rest of the options: 'Positive Input Selection for Channel 0 Sample B','Positive Input Selection for Channel 1/2/3 Sample A' and 'Positive Input Selection for Channel 1/2/3 Sample B' were not used, and so were left as they were. One sample/conversion operations before DMA address increment was selected. The system clock was chosen as the 'ADC Conversion Clock Source', the 'ADC Conversion Clock Select bits'(ADCS) was set as 3. Thus the

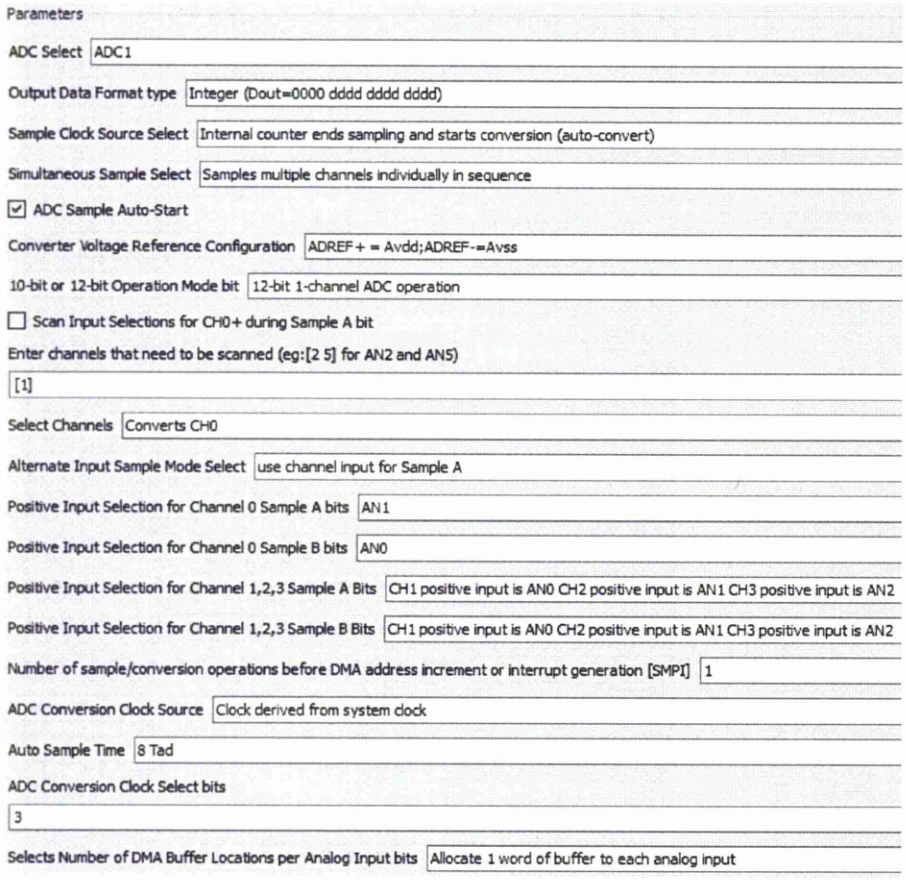

Figure 3.8: ADC module configuration

ADC conversion clock is set as:

$$TAD = TCY \times (ADCS + 1) = (1/16M) \times 4 = 250ns \tag{3.2}$$

The 'Sample Time' for ADC conversion was set as 8 TAD. Then the ADC conversion time is:

$$TCONV = 8 \times TAD = 2000ns \tag{3.3}$$

Corresponding to the ADC conversion time, the (sample) conversion rate is 500 ksps which is the fastest available in 12-bit mode[60]. By Comparison with the main loop time, the ADC conversion is thus seen to be fast enough.

ECAN Configuration

In this model, the ECAN module was used to transmit the control signal to the engine. The ECAN module supports CAN 1.2, CAN 2.0A, CAN 2.0B Passive and CAN 2.0B Active protocol[13]. To configure the ECAN module, 'ECAN Config' blockset was used. The dsPIC33FJ256GP710 has 2 CAN modules. 'ECAN 1' was used in this application.

The Master Clock of the CAN module was set to the same as the instruction frequency, FCAN=FCY. The CAN was run under Normal Operation mode. 8 buffers in DMA RAM were selected. Next the bit timing was set.

As the CAN is a serial protocol, the measure of transmission speed is determined by the number of bits per second transmitted on the CAN bus (nominal bit rate). The Nominal Bit Time=1/Nominal Bit Rate. There are four time segments in a bit time: Synchronization Segment, Propagation Segment, Phase Segment 1 and Phase Segment 2. The Bit Time = Sync Segment + Propagation Segment + Phase Segment 1 + Phase Segment 2. The Figure 3.9 shows the bit time segment.

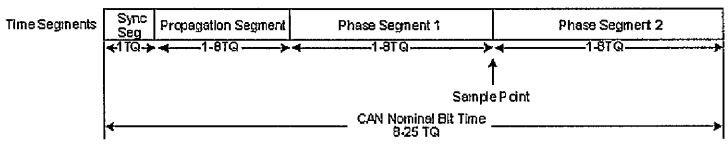

Figure 3.9: Time segment for transmission of each bit[14]

Synchronisation Segment synchronises the different nodes connected on the CAN bus; Propagation Segment compensates for any time delay due to the bus line or the transceivers connected on that bus; Phase Segment 1 compensates for errors due to phase shift in the edges by lengthening the time segment; Phase Segment 2 compensates for errors due to phase shift in the edges by shortening the time segment[14]. The units for those segments is Time Quantum (TQ), which can be calculated by baud rate (FBAUD):

$$FTQ = N \times FBAUD \tag{3.4}$$

where N is the total number of time quanta (8 TQ$\leq$N$\leq$ 25 TQ). The point between Phase Segment 1 and 2 is 'sample point' where the CAN bus is sampled for message once or three times. The times of sampling is controlled by the 'Sample CAN bus Line bit' in ECAN Baud Rate Configuration Register. In this work, 'bus line sampled once' was selected. The Bit Timing of CAN for this work can be calculated by the required Time Quantum (TQ). The Baud Rate Prescaler (CiCFG1<BRP>) is calculated as:

$$CiCFG1(BRP) = \frac{FCY}{2 \times FTQ} - 1 \tag{3.5}$$

The Baud Rate Prescaler must be a integer between 0(00 0000) to 63 (11 1111), the corresponding TQ are listed in Table 3.3.

In this work, a Baud Rate = 1Mbps was selected to ensure that the transmission of the control signal is fast enough. The configuration of bit timing can be determined by this value. The FBAUD = 1 Mbps was selected in this work. Assuming that the number of time quanta N = 8, then FTQ = 8×1 MHz. Thus the Baud Rate Prescaler

BRP<5:0>	Time Quantum (TQ)
00 0000	TQ = 2 × 1 × 1/FCAN
00 0001	TQ = 2 × 2 × 1/FCAN
00 0010	TQ = 2 × 3 × 1/FCAN
⋮	⋮
11 1111	TQ = 2 × 64 × 1/FCAN

Table 3.3: Baud Rate Prescaler

= 16M/(2×8M)-1=0, TQ was scaled from FCAN by the prescaler of 2 × 1 × 1/FCAN. The number of time quanta for each segment were allocated as follows: Synchronisation Segment is a constant and was set to be 1 TQ; if the sample point was at 75% of Nominal Bit Time, Phase Segment 2 = 25% of Nominal Bit Time = 2 TQ; if the Propagation Segment = 2 TQ; then Phase Segment 1 = 8-1-2-2 = 3 TQ. The Synchronization Jump Width (SJW) which should be no longer than the Phase Segment 2 was set as 1 TQ. The detailed configurations for the ECAN module are displayed in figure 3.10.

Figure 3.10: Configurations of ECAN module

Because the CAN module was used only to transmit signals in this work, the CAN filters and Masks were not configured.

DMA Configuration

The Direct Memory Access (DMA) controller is a new feature for 16-bit dsPIC microcontrollers. The DMA controller allows data transferred between peripheral data registers and data space SRAM (DMA RAM) directly by passing the CPU[11]. In this work, we need the DMA RAM to be working as message buffers for the ECAN module to support transmission of CAN messages. According to this requirement, the configurations of the DMA controller are as follows: First the data transfer size was selected as 'word'. The transfer direction was from DAM RAM to CAN, 'Read from DMA RAM address-write to peripheral address' was selected. The DMA Peripheral was selected as 'ECAN1Transmission'. The Start Address Offset was set as 100. The peripheral address register was chose as 'C1TXD'. The detailed configurations are shown in figure 3.11. Besides ECAN transmission, the DMA can be configured for ECAN reception

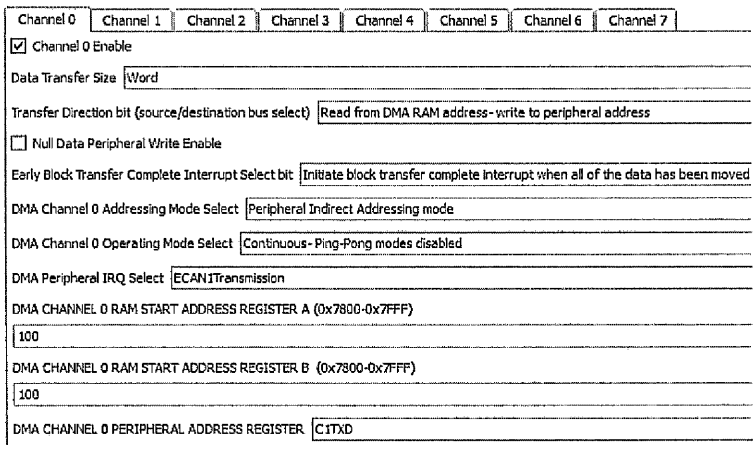

Figure 3.11: Configurations of DMA controller

followed by [29].

Chapter 4

H_∞ Optimal Control by Analytical Optimisation

4.1 Introduction

The H_∞ optimisation control problem can be solved by Helton and Merino's analytic function optimisation method. Using this analytical optimisation, Helton and Merino developed the 'split frequency disk method' for controller design, which solves the multiple sensitivity bounding problem by optimising the different sensitivity functions each in a different frequency range. In this chapter, the analytical optimisation method is also used to solve the standard mixed sensitivity problem. Nonparametric design approaches are then developed which extend Helton and Merino's split frequency disk method and mixed sensitivity optimisation method. All the methods are then applied to powertain control problems.

The H_∞ optimisation problem was first presented by Zames [99]. As mentioned in the previous chapter, the optimal (minimised) value of $\gamma = \|T_{zw}\|_\infty$ in H_∞ control problem is difficulty to obtained. The problem is accordingly solved as $\|T_{zw}\|_\infty < 1$ rather than by directly obtaining a minimising value for $\|T_{zw}\|_\infty$; this problem is named the H_∞ sub-optimal problem. Distinctly, the H_∞ optimal control problem is to find the $\gamma^* = \inf \gamma$ $(\gamma > 0)$ and a corresponding internally stabilizing controller such that the closed loop transfer function $T_{zw}(s)$ satisfy $\|T_{zw}\|_\infty < \gamma$. There are many H_∞ optimisation methods that have been established, such as γ-iteration, J-spectral factorization. Most of them apply to the state-space form and use a two-Riccati formulae. A H_∞ optimisation method based on optimisation frequency response functions was well established by Helton and Merino whose approach is to formulate the weighted sensitivities as a general supremum minimisation problem over the space of analytic functions [39]. The analytic function optimisation method is an iterative approach similar to the common optimisation methods of steepest descent method and Newton's

method. A quadratic Taylor series expansion and a line search strategy are employed in the analytic function optimisation method. An optimal (minimum) value of the objective function can be found by determining the descent direction in each iteration loop. In the conventional descent algorithm or Newton's method the determination of descent direction is based on inverse Hessian and gradient in a quadratic Taylor series expansion of the objective function, while in the analytic function optimisation method, the descent direction is calculated by solving the Nehari problem. Such that the descent direction and the optimal solution is ensured to be analytic on the right half plane (RHP) or in H_∞ space as required.

In this chapter the H_∞ optimal control technique based on analytic functions optimisation method is described. In the section 4.2, to satisfy the internal conditions, an interpolation method given by Helton and Merino [40] is summarised. As a basic requirement for robust feedback control design, the internal stability conditions should be guaranteed during controller design procedure. The method to ensure the internal stability used in nonparametric design is also presented. In the section 4.3 the smoothing and transfer function approximation techniques which are used in H_∞ optimsation are discussed. In the section 4.4, the parametric H_∞ optimal control design approaches - 'split frequency disk' method and the mixed sensitivity optimisation method are presented. In the section 4.6, the nonparametric H_∞ optimal control case based on the 'split frequency disk' method and the mixed sensitivity optimisation method are presented. The design procedures of the parametric and nonparameteric case are both summarized. A simple parametric design example of peak pressure position (PPP) control problem and a simple nonparametric design example of driveline control problem are provided in the section 4.5 and 4.7 respectively. The design procedures and simulation results are presented.

4.2 Internal Stability

The concept of the internal stability of system is introduced first in this section. The internal stability condition and mathematical interpolation method will be summarised. The parametrisation of the interpolant was given by Helton and Merino [40].

4.2.1 Internal Stability Conditions

Consider the closed loop unity negative feedback system of figure 4.1, the constitutive equations are:

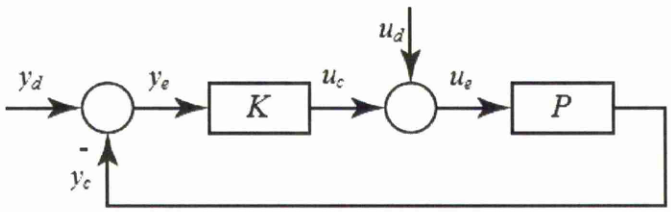

Figure 4.1: Negative unity feedback disturbance system block diagram

$$y_c = Pu_e \tag{4.1}$$

$$u_c = Ky_e \tag{4.2}$$

$$y_e = y_d - y_e \tag{4.3}$$

$$u_e = u_d + u_c \tag{4.4}$$

For the SISO case, there are four transmissions between the different inputs and outputs pairs.

$$y_d \rightarrow y_e \text{ or } u_d \rightarrow u_e : S = \frac{1}{(1+PK)} \tag{4.5}$$

$$y_d \rightarrow y_c \text{ or } u_d \rightarrow u_c : T = \frac{PK}{(1+PK)} \tag{4.6}$$

$$y_d \rightarrow u_c, u_e : U = \frac{K}{(1+PK)} \tag{4.7}$$

$$u_d \rightarrow y_c, y_e : V = \frac{P}{(1+PK)} \tag{4.8}$$

These four transmission are also the sensitivity functions as mentioned in the previous chapter. All four transmissions are required stable for internal stability of the closed loop system. A equivalent internal stability condition, which is more convenient to implemented, is usually employed in a controller design algorithm. The closed loop system is internally stable if the following statements are satisfied[40],[101]:

1. The closed loop transfer function T or the primary sensitivity S is stable.
2. Any unstable pole of the plant is not cancelled by a unstable zero of the controller.
3. Any unstable zero of the plant is not cancelled by a unstable pole of the controller.

As shown in the example at the beginning of this section, the fundamental cause of an unstable system is pole-zero cancellation between plant and controller. Avoid the pole-zero cancellation is naturally help to ensure the internal stability. According to this internal stability condition, some practical methods established to guarantee the

internal stability, such as interpolation approach with the closed loop transmissions. Generally, the interpolation function contains the unstable poles or zeros of the plant. If sensitivity function T is parametrised with the interpolant, the unstable pole-zero cancellation between plant and controller can be avoided, then the internal stability is also guaranteed. The interpolation constraints and parametrisation of the interpolation function for optimisation subject to an internal stability condition is descried in the following sections.

4.2.2 Interpolation Constraints

Recall the internal stability condition as mentioned before, the closed loop system is internally stable if the sensitivity function T is stable and there is no unstable pole-zero cancellation in $KP(L)$. This condition leads to an interpolation constraint for SISO case which is given by [80]. Assume that plant P has the simple unstable poles at $s = p_1, \ldots, p_n$ (n is the number of poles),if the sensitivity function S or T is stable, then the closed loop system is internally stable when the following equality is satisfied:

$$T(p) = 1, \quad S(p) = 0 \tag{4.9}$$

To prove it, assume $S(p) = 0$. Because $S = 1/(1+L)$, $1/(1+L(p)) = 0$. Then $L(p) = \infty$, that means unstable poles p are contained in KP. The p are the unstable poles of plant P, so that controller K does not cancel the unstable poles of plant P[101]. Accordingly if the sensitivity function S or T is stable, the closed loop system is internally stable.

Assume that plant P has the unstable zeros at $s = z$, $z = z_1, \ldots, z_m$ (m is the number of zeros), if the sensitivity function S or T is stable, then the closed loop system is internally stable when the following equality is satisfied:

$$T(z) = 0, \quad S(z) = 1 \tag{4.10}$$

To prove it, assume $T(z) = 0$. Because $T = L/(1 + L)$, $L(z)/(1 + L(z)) = 0$. Then $L(z) = 0$, that means unstable zeros z are contained in KP. The z are the unstable zeros of plant P, so that controller K does not cancel the unstable zeros of plant P[101]. Accordingly if the sensitivity function S or T is stable, the closed loop system is internally stable.

It is to be noted that the case for high-multiplicity RHP poles and zeros of P is more complicated and involves derivatives of T. We will assume simple RHP poles and zeros. The conclusions on the need for a parametric model with this approach still apply though.

To ensure that the equalities 4.9 and 4.10 are satisfied , the interpolants A and B

are introduced to formulate the sensitivity function T [40]:

$$T(s) = A(s) + B(s)T_0(s) \tag{4.11}$$

where function A, B and T_0 are the stable transfer functions. Then optimising over T_0 constrains that T satisfy equations 4.9 and 4.10. Clearly, the sensitivity function T is also stable. For the interpolants A and B, the interpolation constraints [40] can be described as follows. Assume that a plant P is a proper real rational function, with the unstable poles $p_1,\ldots,p_n$, and the unstable zeros $z_1,\ldots,z_m$. If the following statements are satisfied, the unstable pole-zero cancellation in KP can be avoided.

$$A(p_l) = 1 \quad (l = 1,\ldots,n) \tag{4.12}$$

$$A(z_l) = 0 \quad (l = 1,\ldots,m) \tag{4.13}$$

and

$$B(p_l) = 0 \quad (l = 1,\ldots,n) \tag{4.14}$$

$$B(z_l) = 0 \quad (l = 1,\ldots,m) \tag{4.15}$$

From those constraints, the interpolation functions A and B should contain the unstable poles and zeros for the plant P to avoid the unstable pole-zero cancellations. If the internal stability constraints are satisfied and sensitivity function T or S is stable, the closed loop system is then internally stable. It is to be noted that if this interpolation method is employed in the controller design, the function T_0 should be calculated first, then combined with function A and B to formulate the sensitivity function T.

4.2.3 Parametrisation of Interpolation Function

Before giving a parametrisation for the interpolation function, a general parametrisation method for a rational function should be introduced[40]. Let F be a proper transfer function, which can be represented by the summation of several rational functions:

$$F(s) = c_0 + c_1\frac{s - z_1}{s - a} + c_2\frac{(s - z_1)(s - z_2)}{(s - a)^2} + \ldots + c_{n-1}\frac{(s - z_1)\ldots(s - z_{n-1})}{(s - a)^{n-1}} \tag{4.16}$$

where c_0, c_1, ..., c_{n-1} are the constant. This equation is called Newton's representation for function $F(s)$ [40]. To parametrise a suitable and stable interpolant, the number a on the denominators of the functions is required to be a complex number with positive real part. Assume that the following set of equalities are given in advance:

$$F(z_1) = v_1,\ F(z_2) = v_2,\ \ldots,\ F(z_n) = v_n \tag{4.17}$$

where $v_1, \ldots, v_n \in C$. If we set $s = z_1$, $s = z_2$, $\ldots$, $s = z_n$ in equation 4.16, the constant $c_0, c_1, \ldots, c_{n-1}$ can then be obtained by solving the system of equations:

$$
\begin{aligned}
& c_0 = v_1 \\
& c_0 + c_1 \frac{z_2 - z_1}{z_2 - a} \ldots = v_2 \\
& \vdots \\
& c_0 + \ldots + c_{n-1} \frac{(z_n - z_1) \cdots (z_n - z_{n-1})}{(z_n - a)^{n-1}} = v_n
\end{aligned}
$$

Using the idea of Newton's representation form and the equabilities of equations 4.12 to 4.15, the interpolant A and B can be parametrised. The sensitivity T also can be given by multiplying with the parametrised interpolants.

4.2.4 Nonparametric Method

In many practical applications in industry, the plant is stable already. So in the case of a stable plant, the internal stability of the closed loop system depends on the stability of the control effort sensitivity function $U = K/(1 + KP)$. To prove this, let U and P be stable. Because U and P are stable, $T = PU$ is stable; T is stable, then $S = 1 - T$ is stable; As S is stable, sensitivity $V = PS$ is stable. So all four transmissions are stable, the closed loop system is internally stable. It is easy for the designer to design the controller by calculating sensitivity U only for the system when the plant is stable. This method is potentially attractive because determination of zeros and poles is not needed. So this method can be applied to the controller design method which uses a nonparametric plant.

4.3 H_∞ Optimal Control By analytical optimisation

In this chapter, the controller design method based on Helton and Merino's analytical optimisation method is introduced. By using analytical optimisation, the H_∞ optimisation for both disk problem and mixed sensitivity problem can be solved. Then, an extension of nonparametric H_∞ optimisation for disk problem and mixed sensitivity problem is proposed. Before presenting those controller design approach, some particular considerations in controller design procedure is presented. To design a controller based on analytical optimisation, a performance function $\Gamma(\omega, f(j\omega))$ should be given by designers. Then the aim of optimisation is to find $\gamma^* \geq 0$ and $f^*(j\omega) \in A_{RHP}$ such that

$$
\gamma^* = \inf_{f \in A_{RHP}} \sup_{\omega} \Gamma(\omega, f(j\omega)) = \sup_{\omega} \Gamma(\omega, f^*(j\omega)) \tag{4.18}
$$

where A_{RHP} denotes the set of functions analytic on the right half plane. γ^* is the optimal value or a minimizer, $f^*(j\omega)$ is the corresponding optimal solution. The function Γ is the performance function given by designers, Generally, the performance function Γ can be formulated as the weighted sensitivities of H_∞ control problem. For example, the performance function for disk problem can be written as: $\Gamma = R^{-2}(j\omega)|K(j\omega) - f(j\omega)|^2$.

4.3.1 Frequency Response Function

To optimise the performance function Γ which is established in terms of disk problem or mixed sensitivity problem, the discrete frequency response function of Γ should be obtained. For a continuous transfer function $f(s)$, the direct method is to replace the s by $s = j\omega$ (Laplace transform). A vector consisting of a finite number or grid of sample frequencies, called the frequency vector is constructed. A frequency response vector of the transfer function can then be calculated based on this frequency vector. In this work, a frequency vector generation method using a *linear fractional transform* equation [41] is used as suggested by Helton and Schwartz. The equation is defined as:

$$\omega_n = b/tan(\frac{n\pi}{N}) \tag{4.19}$$

where, $n = 0, 1, ..., N-1$ and b is the *sample width parameter*. This gives half the grid points of the frequency vector [ω_1,..., ω_n] in the frequency range of $[-b, b]$. The designer can emphasize any particular frequency range by selecting the parameter b.

To compute the solution, it is easier to work with the functions on the unit disk. The coordinates for the continuous frequency response function will be changed by converting the continuous transfer function to a equivalent discrete transfer function such as $f(j\omega) \Rightarrow f(e^{j\theta})$. So that the continuous transfer function is analytic on right half plane, then the discrete equivalent is required to be analytic on the unit circle in the optimisation. It is to be noted that this transform is for numerical calculation only. In control engineering theory, the discrete transfer function is analytic out of the unit circle when the continuous equivalent is analytic on right half plane. The transforms in numerical calculation and control engineering theory are different. Corresponding to the *linear fractional transform* equation, the transform is:

$$e^{j\theta} = \frac{j\omega - b}{j\omega + b} \tag{4.20}$$

where $\theta = \frac{2\pi i}{N}$, N is the number of frequencies, $i = 0, ..., N-1$. If the performance function is given by a discrete transfer function $f(z)$ which is analytic on the unit disk, the frequency response function can be calculated by evaluating $f(e^{j\theta})$.

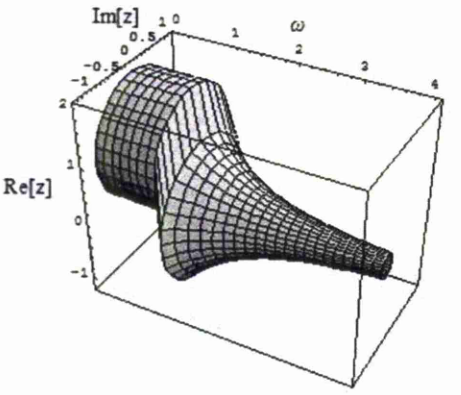

Figure 4.2: Discontinuous Single Disk Inequality

4.3.2 Smoothing for Weighting Function

To solve H_∞ optimisation problem treated as a disk problem, the disk type boundary (weights for sensitivity) is usually switched in different frequency range to satisfy multiple performance and robustness requirements as shown in figure 4.2. The disk type boundary is often discontinuous. This will cause some problems during optimisation, such as bad convergence. To overcome the issues, the single disk inequality should be 'smoothed'. In analytical optimisation method, Blackman-Harris-Nuttall window function will be used. The window function is defined as follows:

$$W(n) = 0.35875 - 0.48829\cos\left(\frac{2\pi n}{N-1}\right) + 0.14128\cos\left(\frac{4\pi n}{N-1}\right) - 0.01168\cos\left(\frac{6\pi n}{N-1}\right) \tag{4.21}$$

where N is the number of the frequency points of the discontinuous frequency response function. n is a counter, which is an integer starting from 0 to $N-1$. The smoothing algorithm used in the design approach is as follows:

$$\mathcal{F}^{-1}(\mathcal{F}(f) \times [W^k]^T) \tag{4.22}$$

where the $\mathcal{F}$ denotes the discrete Fourier transform; $\mathcal{F}^{-1}$ denotes the inverse discrete Fourier transform; f represents the frequency response function; The positive integer k is a parameter chosen by the designers. With increasing k, the function f will become increasingly smooth. Not only for disk problem but also for general mixed sensitivity problem, that function can help the optimal minimizer γ to converge well. Of course, the smoothing process will change the shape of the weightings. When the designer wants to smooth the performance function, the trade off between the smoothness increasing and

retaining the shape of original constraint should be considered. When the performance function formed by sufficient smooth weights is optimised by the analytic function optimisation algorithm, a well convergent γ and corresponding optimal solution will be then obtained.

4.3.3 Transfer Function Approximation

Once the optimal solution is obtained, the transfer function of the optimal solution should be obtained from frequency response for controller implementation purpose. Generally, the optimal solution is given as the closed loop transfer function T. So an important requirement for fitting frequency responses is that the obtained transfer function should be stable. To guarantee that, the Caratheodory-Fejer (CF) approximation will be used in the controller design procedure. This provides a near-best stable rational approximation method for both real or complex functions. For the design procedure, the complex rational CF method[33] of approximating the analytic function on the unit disk is used. The MATLAB codes for both complex and real CF approximation used are those published by Trefethen[84]. Besides the CF approximation, an alternative method is using Levi, Dennis and Schnabel's fitting theory. Basically, that method is to fit the frequency response function as a complex curve using Levi's fitting algorithm[56]. An algorithm improved by Dennis and Schnabel [19] using the Gauss-Newton method for iterative search has been found able to give more accurate results. And the algorithm is available in the MATLAB Signal Processing Toolbox[62] as the functions named *invfreqs* and *invfreqz*, they are used for fitting the frequency responses with continuous and discrete transfer function respectively. The resulting transfer function is also stable.

4.4 Parametric Design Approach

As mentioned, the H_∞ optimisation of disk problem and mixed sensitivity problem can be solved by analytic functions optimisation method. That yields the practical controller design approaches. In this section, the controller design approaches based on optimisation of disk problem and mixed sensitivity problem are introduced.

4.4.1 Split Frequency Disk Approach

The weighted sensitivity function T can be formulated as the disk problem. A robust controller can be found by obtaining the optimal solution for T through solving H_∞ optimisation problem of disk problem. The controller design method based on this

idea was established by Helton and Merino, named 'split frequency disk approach'. The design procedure of the split frequency disk method is summarised in this section.

Assuming that the function A and B are the interpolation functions for T. To ensure the internal stability for the closed loop system, the sensitivity function T should be formulated with interpolation function as $T = A+BT_0$. Then the performance function for the split frequency disk method is:

$$\Gamma = \frac{1}{R^2(j\omega)} |K(j\omega) - (A(j\omega) + B(j\omega)T_0(j\omega))]|^2 \tag{4.23}$$

where K and R are the centre and radius function. T_0 is the optimisation variable. The H_∞ optimisation for disk problem is accomplished by solving the following minimax problem using analytic function optimisation method:

$$\gamma = \inf_{T_0 \in H_\infty} \sup_{\omega} \frac{1}{R^2(j\omega)} |K(j\omega) - (A(j\omega) + B(j\omega)T_0(j\omega))]|^2 \tag{4.24}$$

The radius function R is the boundary for sensitivity T. Because the disk problem formulated one disk type boundary, the single sensitivity problem in H_∞ control can be solved by disk approach naturally. To satisfy multiple performance and robustness requirements in disk approach, different weights can be selected as the boundary R in different frequency range. Normally, the requirements such as track error should be satisfied in low frequency range; the requirements such as bandwidth and closed loop roll off should be satisfied in high frequency range; the requirements such as gain-phase margin and peak magnitude should be satisfied in all frequency range. The trade off between several different requirements is determined by designers from their practical experiences.

Design Procedure

The detailed design procedure is summarised in the following[40].

1. Determine the right half plane (RHP) zeros and poles from the plant P.
2. Generate the interpolation functions A and B based on RHP zeros and poles.
3. Select the desired subject sensitivity function for design, commonly this is T.
4. Determine the Radius Function R and Centre Function K from performance and robustness requirements.
5. Establish a disk type performance function Γ based on Radius Function R and Centre function K with the interpolation functions A and B, expressing T as $T = A + BT_0$.

6. Select the desired frequency range and generate frequency vector.

7. Compute the discrete frequency response for performance function using linear fractional transform.

8. Smooth the constraints (weights).

9. Solve minimax problem for performance function Γ by analytical optimisation.

10. Obtain the minimised γ^* and corresponding optimal solution for T_0. If the γ does not converge well, change the performance and robustness requirements or change the smoothing parameters slightly or change the number of frequency points.

11. Obtain the transfer function for T_0 from optimal solution(frequency response), then obtained the optimised sensitivity function T.

12. Calculate the controller K from $K = \frac{T}{(1-T)P}$.

4.4.2 Mixed Sensitivity Optimisation Approach

In this section the standard mixed sensitivity problem solved by analytical optimisation method is introduced.

Mixed Sensitivity Problem

The mixed sensitivity problem in a design method is to constrain two sensitivities S and T simultaneously. This problem is generally solved with H_∞ Riccati design. The standard mixed sensitivity problem can provide both satisfaction of performance and robustness due to the constraints on sensitivity S and T over full frequency range. Because the limitations of Riccati method, they require only a single norm term to be optimised. To solve this problem the two constraints for S and T are combined into a single H_∞ norm cost function, and the design requirement is then expressed as:

$$\left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty \leq 1 \tag{4.25}$$

where W_S and W_T are weighting functions for sensitivity S and T respectively. This cost function can't be solved easily. To solve the problem practically an important alternative approximate equation which is close to the original standard problem is used:

$$\left\| |W_S S|^2 + |W_T T|^2 \right\|_\infty \leq \frac{1}{2} \tag{4.26}$$

This alternative formulation is equivalent to the formulation that is:

$$\left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty \leq \frac{1}{\sqrt{2}} \tag{4.27}$$

Obviously, if equation 4.26 is satisfied, the original standard mixed sensitivity inequality is satisfied too although slightly conservatively. Generally, the equation 4.26 can also be described as $\left\| |W_S S|^2 + |W_T T|^2 \right\|_\infty \leq 1$ by changing weighting functions.

Optimisation for Mixed Sensitivity Problem

Helton and Merino's analytical optimisation method can also solve the mixed sensitivity problem. For optimisation purposes, the performance function based on mixed sensitivity problem is formulated as:

$$\Gamma = |W_S(j\omega)(1 - T(j\omega))|^2 + |W_T(j\omega)T(j\omega)|^2 \tag{4.28}$$

The H_∞ optimisation for mixed sensitivity problem is accomplished by solving the following minimax problem using analytic function optimisation method:

$$\gamma = \inf_{T \in H_\infty} \sup_{\omega} |W_S(j\omega)(1 - T(j\omega))|^2 + |W_T(j\omega)T(j\omega)|^2 \tag{4.29}$$

Through solving this minimax problem, the optimal minimizer γ^* and corresponding optimal solution for the sensitivity function T can be found. To solve the optimisation problem for the performance function of equations 4.28,the performance function is converted to a quasi-disk type problem through using Taylor expansion and completing a 'perfect square' quadratic form. Then the descent direction $\in H_\infty$ can be found by solving Nehari problem for a quasi-disk type problem. Next, the line search scheme is employed to find the optimal minimizer and solution.

Design Procedure

The controller design procedure based on mixed sensitivity optimisation is discussed as follows. To design a controller using analytical optimisation of mixed sensitivity problem, the parametrisation of the interpolation function for internal stability problem also should be considered. Assuming that the function A and B are the interpolation function parametrised by RHP zeros and poles of plant. Then in controller design procedure, the performance function is formulated as:

$$\Gamma = |W_S(j\omega)[1 - (A(j\omega) + B(j\omega)T_0(j\omega))]|^2 + |W_T(j\omega)(A(j\omega) + B(j\omega)T_0(j\omega))|^2 \tag{4.30}$$

where T_0 is the optimisation variable. The detailed design procedure is presented in the following:

1. Determine the RHP zeros and poles from the plant P,
2. Generate the interpolation functions A and B based on RHP zeros and poles.

3. Determine the weighting function W_S and W_T for sensitivity S and T based on performance and robustness requirements.

4. Establish the performance function Γ of the standard mixed sensitivity problem with the interpolation function, expressing T as $T = A + BT_0$.

5. Select the desired discrete frequency range and generate frequency vector.

6. Compute the discrete frequency response for performance function using linear fractional transform.

7. Smooth the weighting functions W_S and W_T

8. Solve minimax problem for performance function Γ by analytical optimisation.

9. Obtain the minimised γ^* and corresponding optimal solution for T_0. If the γ does not converge well, change the weightings W_S and W_T or change the smoothing parameters slightly or change the number of frequency points.

10. Obtain the transfer function for T_0 from optimal solution(frequency response), then obtained the optimised sensitivity function T.

11. Calculate the controller K from $K = \frac{T}{(1-T)P}$.

It is to be noted that the standard mixed sensitivity problem is solved here, but some other similar problems also can be solved by analytic function optimisation/disk iteration algorithm, such as:

$$\left\| \begin{bmatrix} W_S S \\ W_T T \\ W_U U \end{bmatrix} \right\|_\infty \leq 1 \tag{4.31}$$

4.5 Parametric Design Example

4.5.1 Peak Pressure Position Control

As an example of the application of controller design approaches based on the isk approach and the mixed sensitivity optimisation approach, the peak pressure position (PPP) control problem of the IC engine is considered. Generally the peak pressure position is controlled through regulating the spark advance (SA) before the top dead centre (TDC) in the IC engine. An optimal controller using spark advance is required to give a fixed peak pressure position, which is approximately about 16^o after the top dead centre [45]. Since this position is known to maximise the engine efficiency, and so give what is called the maximum best torque (MBT) can be obtained.

4.5.2 Split Frequency Disk Approach - Single Sensitivity

In this design example, a simple ARX model was used. This simple structure of model has been found effective in several previous studies[7], [91]. The model was given by Ward [91]. To keep enough accuracy, four significant digits are retained for the parameters of the transfer functions. The transfer function of plant is:

$$P = \frac{0.6729s - 20.41}{s + 20.6} \tag{4.32}$$

Due to the RHP zero at $s = 30.3314$, the interpolation functions A and B were parametrised as:

$$A = \frac{s - 30.3314}{s + 2}, B = \frac{s - 30.3314}{s + 2}$$

During optimisation the closed loop system was then constrained as internally stable by expressing the complementary sensitivity function T as: $T = A + B \cdot T_0$ and using T_0 as optimising variable.

Basically, the SA to PPP control problem is an error tracking problem. In the IC engine, the PPP should track the desired optimal position. As introduced in the previous chapter, the error tracking control problem is accomplished by bounding the primary sensitivity function S. For simplicity, only error tracking performance requirement was considered in this design. Accordingly only a single constraint was considered in the design, the performance function Γ for this problem could be also established easily, the centre function was thus specified by $K = 1$ for all frequencies, the radius function was equivalent to the inverse weighting of the primary sensitivity, $R = W_S^{-1}$, for all frequency range. As mentioned in Chapter 2, S must be kept small at low frequency to force the error signal to be zero. The basic shape of W_S^{-1} is a high pass filter. As the engine in practice is always subject to some perturbations and effectively never reaches a frequency range lower than $0.001 rad/s$, the W_S^{-1} is selected to roll off around $0.001 rad/s$ with decreasing frequency. Then a simple weighting function for the primary sensitivity was taken as:

$$W = \frac{s + 1}{s + 0.001} \tag{4.33}$$

The performance function was formulated as:

$$\Gamma = \left(\frac{s + 1}{s + 0.001}\right)^2 \cdot \left|1 - \left(\frac{s - 30.3314}{s + 2} + \frac{s - 30.3314}{s + 2} \cdot z\right)\right|^2 \tag{4.34}$$

where z is unknown and represent the optimisation variable T_0. To obtain the frequency response vector of the performance function Γ, 256 positive frequency grid points were generated by using linear fractional transform equation (equation 4.19) with parameter $b = 0.1$. analytical optimisation method was used to calculate the optimal γ and corresponding optimal solution for T were obtained. The transfer function of T was

found by fitting to the frequency response of the solution using CF approximation, and was found as:

$$T = \frac{-0.002736s^3 + 0.07922s^2 + 0.1127s + 0.06901}{0.02714s^3 + 0.1072s^2 + 0.1404s + 0.06886} \tag{4.35}$$

Sensitivity function U and V were then calculated to check the internal stability of the closed loop system. These were found as:

$$U = \frac{-0.1498s^3 - 3.295s^2 - 4.4s - 2.567}{s^3 + 3.952s^2 + 5.173s + 2.538} \tag{4.36}$$

$$V = \frac{0.7407s^4 - 21.78s^3 - 20.4s^2 - 20.85s + 0.1122}{s^4 + 24.5600s^3 + 86.61s^2 + 109.1s + 52.29} \tag{4.37}$$

The sensitivity U and V were thus established as being stable. The closed system is internally stable. The controller K was then computed from $K = \frac{T}{(1-T)G}$ and found to be:

$$K = \frac{-0.1361s^3 - 2.993s^2 - 3.997s - 2.332}{s^3 + 0.9384s^2 + 0.9277s - 0.004993} \tag{4.38}$$

Figures 4.3 and 4.4 show the Bode magnitude of the sensitivity function S and T. In the Bode magnitude plot of S, the broken line represents the inverse weighting for primary sensitivity function. From that figure, the sensitivity function S is bounded by the weighting W in desired frequency range. Although the sensitivity function T was not bounded by any constraints or weighting function, it still rolls off well in high frequency range. To analyse the time domain performance, the unit step response was simulated in MATLAB/SIMULINK. Figure 4.5 shows the result of step response. From both assessment frequency domain analysis (Bode plots) and time domain analysis (step response), the controller design was seen to be successful. The desired sensitivity function S is bounded well, and the closed system is internally stable. The simulation of the step response shows that the closed loop settled in about 6s and the steady-state error is effectively zero.

4.5.3 Mixed Sensitivity Approach

The mixed sensitivity approach was also applied to the PPP control problem too. In this design procedure, both sensitivity S and T were constrained simultaneously. For weighting of S, the limits of $1.5^{\pm 1}$ open loop gain and ± 5 samples delay with the open loop transmission were considered. It indicated that S should be less than $4dB$. For weighting of T, 5% overshoot limit and high frequency roll-off were considered. It indicated that T should be less than $6B$ and roll off a round $1000rad/s$ (as the engine never achieved up to $1000rad/s$). Then the weighting function for sensitivity S and T

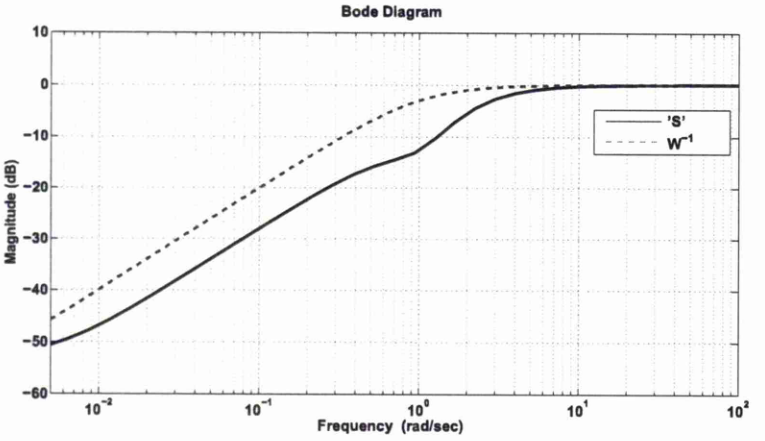

Figure 4.3: Bode magnitude of sensitivity function S

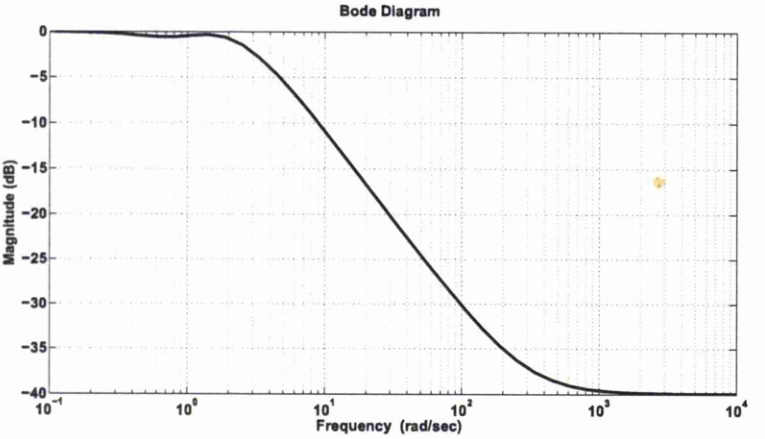

Figure 4.4: Bode magnitude of sensitivity function T

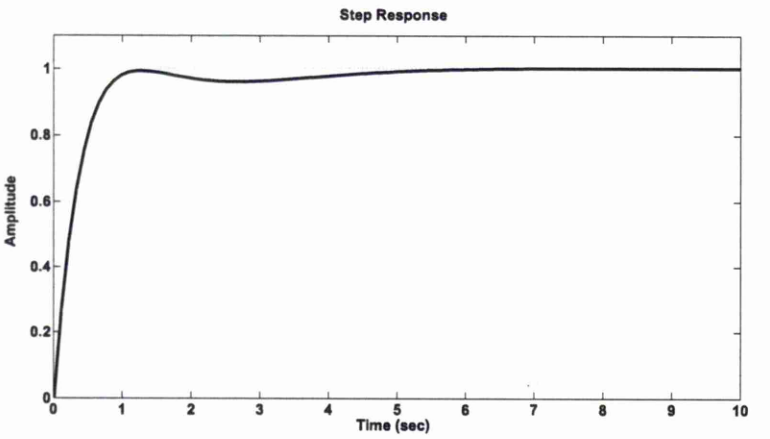

Figure 4.5: Simulation of the unit step response

were:

$$W_S = \frac{s + 1.57}{1.57s + 0.001} \tag{4.39}$$

$$W_T = \frac{0.2s + 1}{0.001s + 2} \tag{4.40}$$

The performance function was formulated as:

$$\Gamma = \left(\frac{s + 1.57}{1.57s + 0.001}\right)^2 \cdot \left|1 - \left(\frac{s - 30.3314}{s + 2} + \frac{s - 30.3314}{s + 2} \cdot z\right)\right|^2 + \left(\frac{0.2s + 1}{0.001s + 2}\right)^2 \cdot \left|\frac{s - 30.3314}{s + 2} + \frac{s - 30.3314}{s + 2} \cdot z\right|^2 \tag{4.41}$$

where z is unknown and represent the optimisation variable T_0. To obtain the frequency response vector of the performance function Γ, 256 positive frequency grid points were generated by using the linear fractional transform equation (equation 4.19) with parameter $b = 0.1$. Using analytical optimisation method, the optimal γ and corresponding optimal solution for T were obtained. The transfer function of T was found by fitting the frequency response of solution using CF approximation, and found to be:

$$T = \frac{-7.121e-6s^4 - 0.002s^3 - 0.1172s^2 + 5.259s + 10.22}{0.0004918s^4 + 0.07641s^3 + 2.957s^2 + 10.73s + 10.24} \tag{4.42}$$

The sensitivity functions U and V were then calculated to check the internal stability of the closed loop system.

$$U = \frac{-0.02152s^4 - 7.139s^3 - 695.3s^2 - 1.25e4s - 2.097e4}{s^4 + 155.4s^3 + 6012s^2 + 2.182e4s + 2.081e4} \tag{4.43}$$

$$V = \frac{0.6826s^5 + 86.57s^4 + 951.2s^3 - 1.201e5s^2 - 2.271e5s - 865.6}{s^5 + 176s^4 + 9213s^3 + 1.457e5s^2 + 4.703e5s + 4.289e5} \tag{4.44}$$

The sensitivity functions U and V were thus established as stable. The closed system was thus itself established as internally stable. The controller K was computed from $K = \frac{T}{(1-T)G}$ and found to be:

$$K = \frac{-0.02121s^4 - 7.037s^3 - 685.4s^2 - 1.232e4s - 2.067e4}{s^4 + 157.2s^3 + 6161s^2 + 1.097e4s + 41.8} \tag{4.45}$$

Figures 4.6 and 4.7 show the Bode magnitude of the sensitivity function S and T. In the Bode magnitude plot of S, the broken line represents the inverse weighting for both the primary sensitivity and complementary sensitivity. From that figure, the sensitivity function S and T are seen to be bounded by the weightings well in the desired frequency range. Again to analyse the time domain performance, the unit step response was simulated in MATLAB/SIMULINK. Figure 4.8 shows the result of step response. The simulation of the step response shows that the controller designed by the

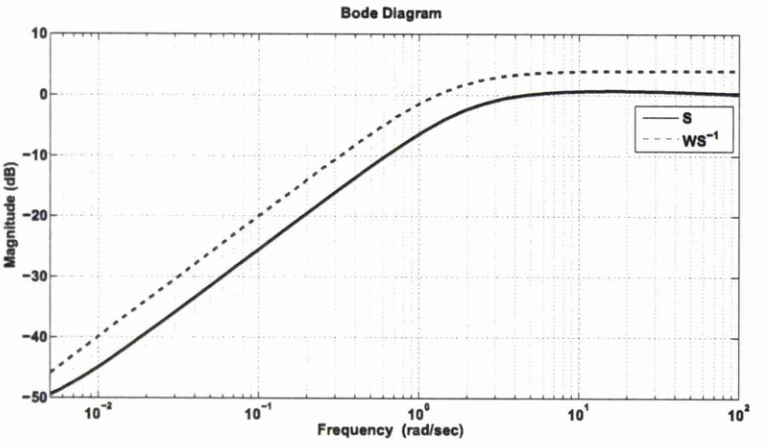

Figure 4.6: Bode magnitude of sensitivity function S

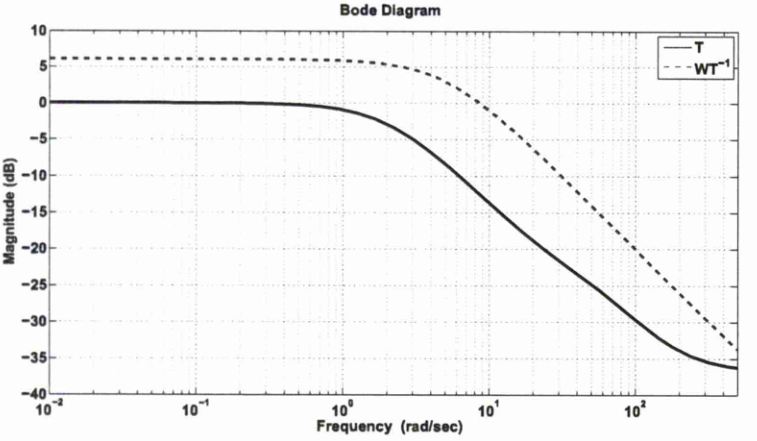

Figure 4.7: Bode magnitude of sensitivity function T

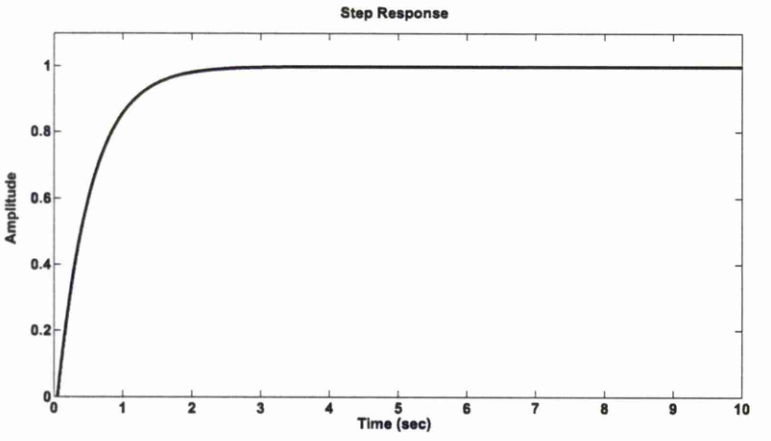

Figure 4.8: Simulation of the unit step response

mixed sensitivity approach responds to the demand more quickly than the controller designed by the disk approach. The settling time is about 3s and no overshoot for the control of mixed sensitivity method. The zero overshoot is due to the 'peak magnitude' constraint accomplished by bounding the sensitivity function T.

4.6 Nonparametric Design Approach

The nonparametric controller design approach is a method which calculates controller based on nonparametric plant. Because nonparametric plant requires minimal assumptions on model structure, it gives a more accurate representation of a physical system than a parametric plant. Accordingly, nonparametric design techniques are potentially attractive because they may be expected to provide improved performance and robustness benefits. In this section, a novel nonparametric design technique combined with the H_∞ optimisation is developed by using analytic function optimisation for both the disk problem and mixed sensitivity problem.

As the analytic function optimisation method is based solely on system frequency response function (nonparametric function), a nonparametric design approach based on nonparametric plant data only is also possible. The nonparametric model of the plant can be identified as a frequency response function using the nonparametric identification techniques (this will be discussed in the next chapter). The nonparametric design techniques proposed here are extended from the parametric split frequency disk method and the mixed sensitivity method. The main difference between the parametric and the nonparametric design approach is that the poles or zeros are not available directly from a nonparametric plant model. The interpolation approach for guaranteeing internal stability is thus not available for nonparametric design. However, the internal stability of the closed loop system can be determined by the stability of the sensitivity function U when the plant P is stable. So a nonparametric deign approach is available when plant P is stable.

In nonparametric design, the performance function is established by the sensitivity functions U and corresponding weightings only. Because the sensitivity function $U = \frac{K}{1+PK}$, the sensitivity function T can be expressed as $T = \frac{PK}{1+PK} = PU$. In nonparametric design, the performance function can be established by replacing T with PU in the performance function of parametric design for both the disk method and mixed sensitivity method. The sensitivity function U is thus selected as the optimisation variable in design.

4.6.1 Nonparametric Split Frequency Disk Approach

Assuming that the plant P is in such a frequency response function form. Accordingly, the performance function in nonparametric split frequency disk method is established as follows:

$$\Gamma = \frac{1}{R(j\omega)^2} |K(j\omega) - P(j\omega)U(j\omega)|^2 \tag{4.46}$$

where R and K are the Radius and Centre function respectively, the sensitivity function U is the optimisation variable. Reformulating equation 4.46 to a circular type function gives:

$$\Gamma = \frac{|P(j\omega)|^2}{R(j\omega)^2} \left| \frac{K(j\omega)}{P(j\omega)} - U(j\omega) \right|^2 \tag{4.47}$$

The nonparametric H_∞ optimisation for disk problem is accomplished by solving the following minimax problem using analytic function optimisation method:

$$\gamma = \inf_{U \in H_\infty} \sup_{\omega} \frac{|P(j\omega)|^2}{R(j\omega)^2} \left| \frac{K(j\omega)}{P(j\omega)} - U(j\omega) \right|^2 \tag{4.48}$$

After that the optimal solution for the sensitivity function U is found, controller K is then obtained.

Design Procedure

The detailed design procedure of the novel nonparametric split frequency disk method is listed as follows:

1. Identify the nonparametric plant P with the desired frequency points.
2. Select sensitivity function U for design.
3. Determine the Radius Function R and Centre function K from the performance and robustness requirements.
4. Establish a disk type performance function Γ with nonparametric plant P, Radius Function R and Centre function K.
5. Smooth the constraints (weights).
6. Solve minimax problem for performance function Γ by analytical optimisation.
7. Obtain the minimised γ^* and corresponding optimal solution for U. If the γ does not converge well, change the performance and robustness requirements or change the smoothing parameters slightly or change the number of frequency points.
8. Obtain the transfer function for U by fitting to the optimal solution (frequency response).

9. Calculate the controller K from $K = \frac{U}{1-PU}$.

4.6.2 Nonparametric Mixed Sensitivity Optimisation Approach

To developed the nonparametric design technique for mixed sensitivity problem, the sensitivity function T is still replaced by $T = PU$. The mixed sensitivity problem can be reformulated as:

$$\left\| |W_S(j\omega)(1 - P(j\omega)U(j\omega))|^2 + |W_T(j\omega)(P(j\omega)U(j\omega))|^2 \right\|_\infty \leq 1 \tag{4.49}$$

Then the perfromance function in nonparametric design can be formulated as:

$$\Gamma = |W_S(j\omega)(1 - P(j\omega)U(j\omega))|^2 + |W_T(j\omega)(P(j\omega)U(j\omega))|^2 \tag{4.50}$$

where sensitivity function U is the optimisation variable. The nonparametric H_∞ optimisation for disk problem is accomplished by solving the following minimax problem using analytic function optimisation method:

$$\gamma = \inf_{U \in H_\infty} \sup_{\omega} |W_S(j\omega)(1 - P(j\omega)U(j\omega))|^2 + |W_T(j\omega)(P(j\omega)U(j\omega))|^2 \tag{4.51}$$

Once that the optimal solution for the sensitivity function U is found, controller K is thus obtained.

1. Identify the nonparametric plant P with the desired frequency points.
2. Select sensitivity function U for design.
3. Determine the weighting functions W_S and W_T from the performance and robustness requirements.
4. Establish a performance function Γ for mixed sensitivity problem with nonparametric plant P and weighting functions W_S and W_T
5. Smooth the constraints (weights).
6. Solve minimax problem for performance function Γ by analytical optimisation.
7. Obtain the minimised γ^* and corresponding optimal solution for U. If the γ does not converge well, change the performance and robustness requirements or change the smoothing parameters slightly or change the number of frequency points.
8. Obtain the transfer function for U by fitting to the optimal solution (frequency response).
9. Calculate the controller K from $K = \frac{U}{1-PU}$.

4.7 Nonparametric Design Example

4.7.1 Nonparametric Driveline Control

A nonparametric design example is demonstrated in this section based on a nonparametric plant given by Abass [1]. This nonparametric plant was identified from a white-box driveline (SIMULINK) model. Usually the driveline model is a stable model, such that the nonparametric design approach is possible to be implemented. Due to the nonlinear clutch and driveline backlash characteristics the driveline model was a nonlinear model. The nonparametric model was established through the nonparametric frequency response identification using short time Fourier transforms. To model the nonlinearity of the system, several different input-output data pairs were captured. Then a nominal model was obtained by finding the centre of the uncertain disk at each frequency. The uncertain disk at each frequency is the smallest disk containing all the complex responses at that frequency [80]. This approach then allows the nonlinearities to be treated as the unstructured uncertainties of the linear model in the design. A simplified model which was also identified without backlash was used to compare. Figures 4.10 and 4.9 show the nonparametric models for the driveline model with or without backlash. The nonparametric model with backlash was established based

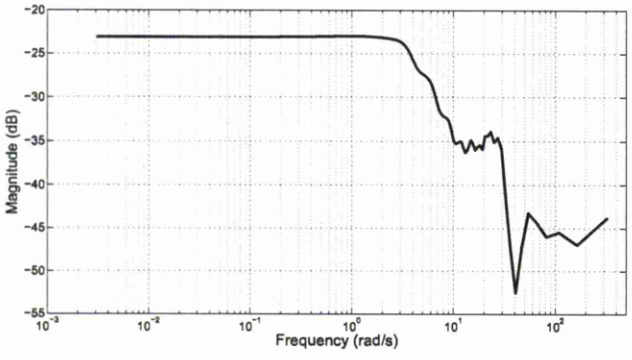

Figure 4.9: Bode magnitude of the nonparametric model with backlash

on the frequency vector which is generated by the linear fractional transform equation (equation 4.19) with parameters $b = 1$ and $N = 1024$. The nonparametric model without backlash was established based on the frequency vector which is generated by the parameters $b = 1$ and $N = 512$.

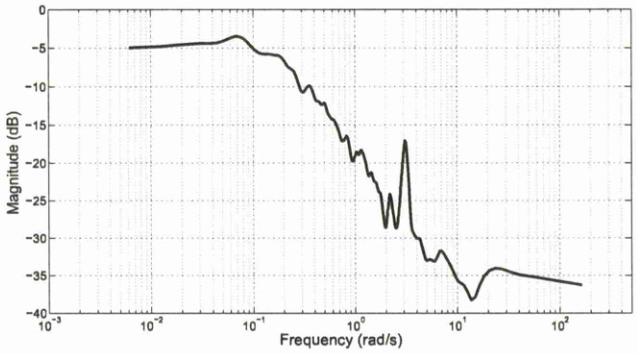

Figure 4.10: Bode magnitude of the nonparametric model without backlash

4.7.2 Nonparametric Design by Mixed Sensitivity Approach

In this design example, the controller was obtained by the mixed sensitivity optimisation approach. The performance function was formulated as follows:

$$\Gamma = |W_S(j\omega)(1 - P(j\omega)U(j\omega))|^2 + |W_T(j\omega)(P(j\omega)U(j\omega))|^2 \tag{4.52}$$

where the weighting function $W_S = \frac{s+1.57}{1.57s}$ and $W_T = \frac{0.2s+1}{2}$. Through using analytical optimisation method, the minimax problem for performance function Γ was solved. The optimal solution for U was thus found. Figures 4.11 and 4.12 show (inverse) weightings and the (nonparametric) frequency response of sensitivities S and T, which were calculated from the optimal solutions U and the nonparametric plants. From these the figures, the unsmoothed frequency responses of S and T basically satisfy the constraints of the weighting functions. Then the transfer function of controller K was found by fitting to the frequency response of sensitivity U, S or T using MATALB's *invfreqs* function. The controller for the nonparametric model with backlash was found to be:

$$K_{nl} = \frac{3.939s^3 + 442.5s^2 + 1.815e4s + 4.233e4}{s^3 + 21.23s^2 + 2103s + 12.41} \tag{4.53}$$

The controller for the nonparametric model without backlash was found to be:

$$K_l = \frac{-19.6s^3 - 256.2s^2 + 953.3s + 212.6}{s^3 + 39.11s^2 + 179s + 2.761} \tag{4.54}$$

Figure 4.13 shows the simulated unit step responses using the controllers from the two models (with and without backlash). Due to the driveline backlash, the overshoot for the nonparametric controller based on the model with backlash is higher than the nonparametric controller based on the model without backlash. The designs of both two nonparametric controllers are seen to be successful from the simulations. The nonparametic approach was validated.

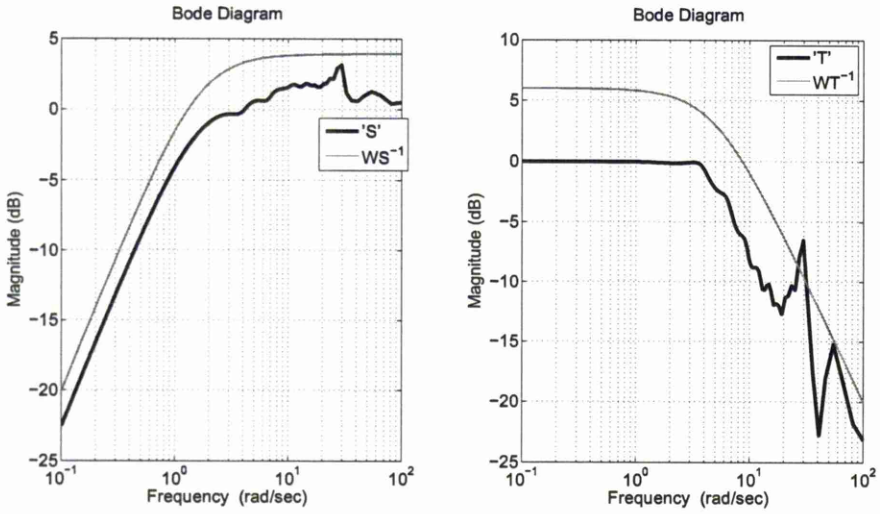

Figure 4.11: Frequency response of S and T based on the nonparametric model with backlash

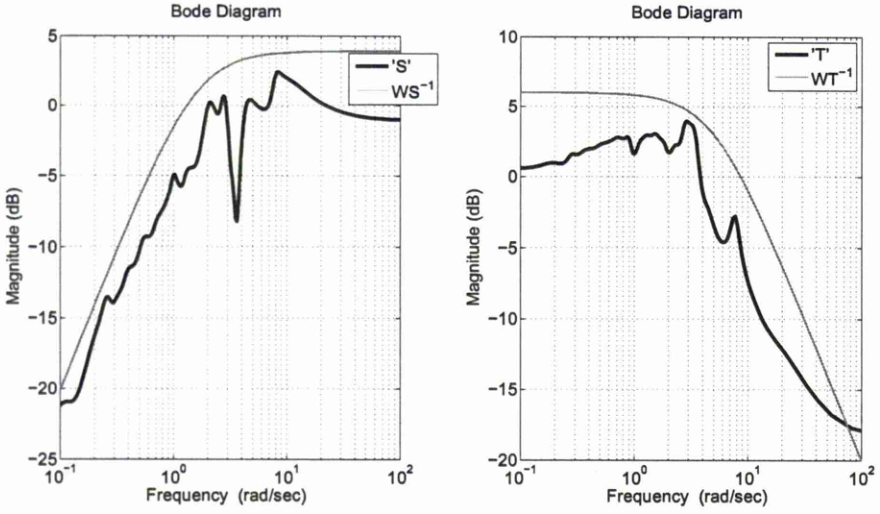

Figure 4.12: Frequency response of S and T based on the nonparametric model without backlash

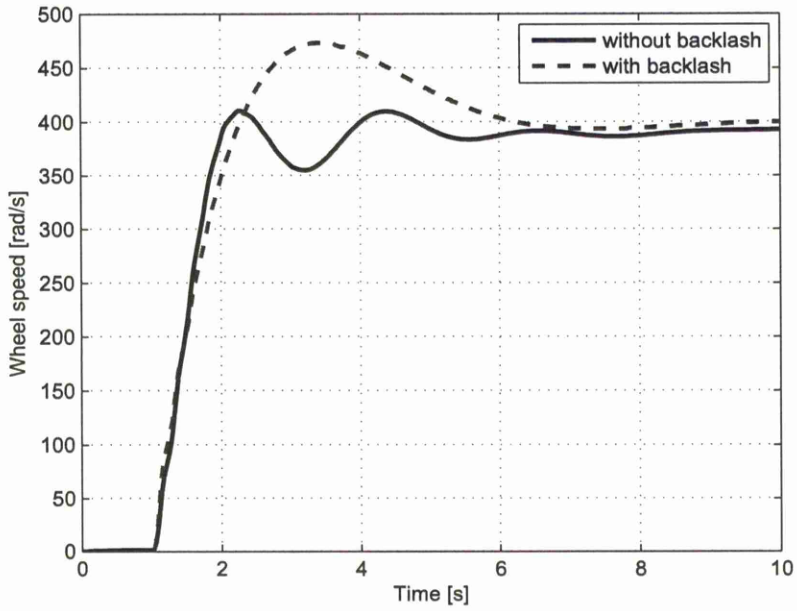

Figure 4.13: Unit step responses of the model with backlash and without backlash

4.8 Conclusion

In this chapter a nonparametric H_∞ optimal control technique based on Helton and Merino's analytic functions optimisation method was developed. Due to the requirement of the robust control design, the concept and conditions of internal stability were presented. Based on internal stability conditions, a interpolation method given by Helton and Merino was introduced. The parametrisation of the interpolation function were summarized. For nonparametric design, the internal stability condition by optimising sensitivity U when plant is stable was also presented.

Then the parametric H_∞ optimal control design procedures based on the 'split frequency disk' method and the mixed sensitivity optimisation method were presented. Basic design step such as construct the performance function, generating the frequency vector and corresponding frequency response, smoothing for the frequency response function and transfer function fitting to the frequency response presented. The nonparametric control design approach, in which the performance functions were formulated in terms of the sensitivity U for the disk problem and the mixed sensitivity problem, was developed and presented. The design procedures for nonparametric design were detailed.

A simple design example of peak pressure position (PPP) control problem of IC engine was illustrated for parametric design technique. The design procedures with disk

method and mixed sensitivity method were presented respectively. The bode diagram shows that the desired sensitivity functions are bounded well for both two controller design methods. The simulation results of step responses shows that both 'split frequency disk method' and mixed sensitivity optimisation method are applied to PPP control problem successfully. For nonparametric design case, a example of driveline control problem was demonstrated. The design procedures using mixed sensitivity optimisation method was presented. The frequency response of the sensitivity S and T was displayed. Although the frequency responses of S and T are unsmoothed, the Bode magnitude plot shows that both S and T satisfy the weightings. In the next chapter, the nonparametric modelling method will be introduced and applied to a nonparametric controller design of idle speed control problem. The comparison of parametric and nonparametric controller design will be presented in detail.

Chapter 5

Nonparametric Controller Design for Idle Control with CAN-bus Networked Embedded System

5.1 Introduction

Nonparametric (NP) design techniques are potentially attractive because they require minimal assumptions about model structure and can be obtained directly from experimental test data. Accordingly they may be expected to give a more accurate representation of a physical system than a more model based approach and may be expected to provide improved performance and robustness benefits. In this section, the nonparametric design techniques are incorporated with nonparametric identification and applied to the idle speed control problem. In section 5.2, the linear nonparametric identification method is summarised. Then parametric and nonparametric identifications with engine dynamometer data are presented in section 5.3. The robust controller design by analytical optimisation and the performance of the controllers designed from both parametric and nonparametric models were compared and investigated in section 5.4. The nonparametric controller is implemented in a CAN-bus based embedded system in section 5.5. The fast prototyping of the controller as an embedded system and the experimental results for the resulting controller are also presented in section 5.5.

5.2 Linear Nonparametric Modelling

A linear time-invariant model can be described by its frequency response or by its corresponding impulse response. Nonparametric methods aim to determine these functions by direct techniques without selecting a certain model structure. Nonparametric methods can be applied in the time domain by using impulse-response testing, step-response

testing and correlation analysis or in the frequency domain by using Fourier analysis and spectral analysis.

5.2.1 The Impulse Response Function

Time-domain correlation analysis is a nonparametric estimate of the transient response of dynamic systems, which computes the impulse response $g(\tau)$ from the data. The response $y(t)$ to an input $u(t)$ is taken as the convolution of the impulse response $g(t)$ with the input plus an additive noise term $v(t)$, as in

$$y(t) = v(t) + \int_{\tau=0}^{\infty} g(\tau)u(t-\tau)d\tau \tag{5.1}$$

An estimate of the impulse response is then obtained from an estimate of the cross covariance $R_{yu}(\tau)$ [57]. Correlation analysis thus assumes a linear system and does not require a specific model structure.

5.2.2 Covariances and Cross-covariance

For N data samples, the cross-covariance R_{yu} between any two signals u and y resp. is given by

$$R_{yu}(\tau) = \frac{1}{N}\sum_{t=1}^{N} y(t+\tau)u(t) \tag{5.2}$$

The cross-covariance R_{yu} between input and output signals u and y resp. and the auto-covariances R_{uu} and R_{yy} of the input and output signals u and y resp. are then determined from (5.2).

5.2.3 Short Time Fourier Transform (STFT)

The function to be transformed by STFT is multiplied by a window function W_M which is nonzero for only a finite (relatively short) period of time τ. The Fourier transform of the resulting signal is taken for each position as the window is slid along the time axis, resulting in a two-dimensional representation of the signal. Thus the transform of a cross-correlation function R_{yu} is

$$\Phi_{yu}(\omega) = \sum_{\tau=-M}^{M} R_{yu}(\tau)W_M(\tau)e^{-i\omega\tau} \tag{5.3}$$

where $W_M(\tau)$ is called the lag window with width M. The choice of width of the lag window, is of course, very dependent on the total number of the available data points N. Tranforms $\Phi_{uu}(\omega)$ and $\Phi_{yy}(\omega)$ of the auto-correlation functions R_{uu} and R_{yy} follow.

In some text books $\gamma = N/M$ is used instead of M. Hence, a small γ corresponds to a large width of window with respect to the total number of data points [57].

5.2.4 Frequency Response Function

The frequency-response function $G(e^{i\omega})$ and the output noise spectrum $\Phi_{vv}(\omega)$ are computable from the covariances and cross-covariance of the input $u(t)$ and output $y(t)$ using the Blackman-Tukey approach as follows [59]:

$$G(e^{i\omega}) = \frac{\Phi_{yu}(\omega)}{\Phi_{uu}(\omega)} \tag{5.4}$$

$$\Phi_{vv}(\omega) = \Phi_{yy}(\omega) - \frac{|\ \Phi_{yu}(\omega)\ |^2}{\Phi_{uu}(\omega)} \tag{5.5}$$

5.3 Plant Identification and Uncertainty Modelling

We consider an application of the nonparametric method approach to the SISO automotive idle speed control problem. This SISO idle speed control problem requires a compromise between maintaining a low engine speed for improved fuel consumption and an ability to reject disturbances. The disturbance rejection must reduce the magnitude of the maximum speed dip or flare in a transient and have a small settling time without excessive oscillation to minimise noise and vibration and avoid possible engine stall. The idling engine experiences unpredictable torque loads from the power-assisted steering hydraulics, air conditioner unit, fan motors, or other electrical alternator loads. The idle speed is controlled by feeding back the engine speed measured at the flywheel to the air bleed (air-bypass) valve (ABV). An effective controller design method that has been shown effective is based on an identified model of the ABV to engine speed frequency response function. A set of such frequency response functions is obtained, by identifying the system at different base loads and speeds and then the set is expressed as a nominal model subject to multiplicative uncertainty to characterise the overall system.

For this study a PRBS test signal was applied to the system, and both parametric and nonparametric models were identified based on the same testing data using ARMAX and the Blackman-Tukey [59] frequency domain method resp.. For the parametric identification, a sample based ARMAX model was obtained with 0.001s sampling time for data collection. Then, the sampling time was down sample to $T_s = 0.03$s which is optimal for design. A third order model with 7 sample time delays was thus obtained as

$$P(s) = \frac{(1334s + 1.753e004)e^{-7T_sS}}{s^2 + 3.187s + 3.758} \tag{5.6}$$

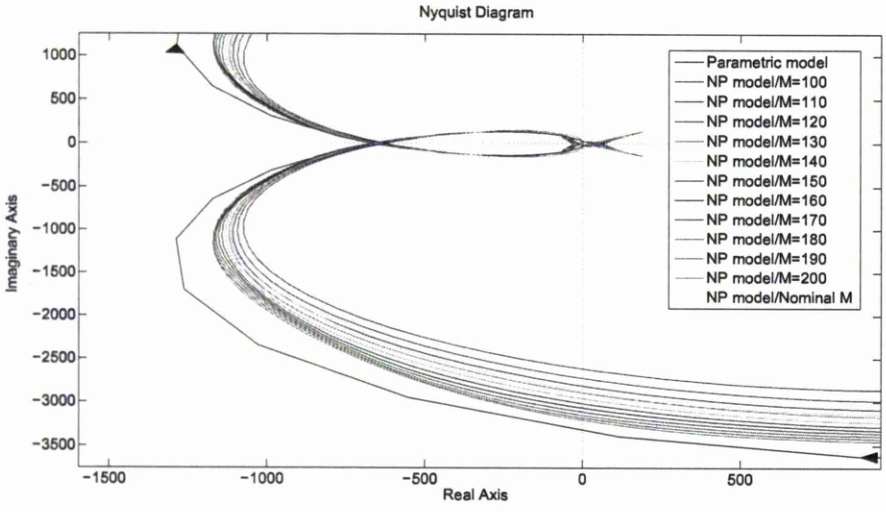

Figure 5.1: Nyquist plots of parametric and nonparametric models

The Blackman-Tukey identification was performed with Windowed Fourier Transforms (WFTs) using the Hamming window [59] of $0.5(1+\cos(i\pi/M))$; $i = 1, \ldots, N_t$ where N_t is the number of samples of time data, for the window sizes of $M = 100, 110, 120,$ $130, 140, 150,$ $160, 170,$ $180,$ 190 and 200. In this identification, $N_w = 128$ frequency grid points were used for the Windowed Fourier Transforms (WFTs). The frequency grid points are generated by the following *linear fractional tansform* equation [41]:

$$\omega_n = b/tan(\frac{n\pi}{N_w}) \tag{5.7}$$

where, $n = 0, 1, ..., N_w - 1$ and b is the *sample width parameter.* That gives half of the the ω_n grid points in the frequency range of $[-b, b]$. Figs, 5.1, 5.2, 5.3 show the Nyquist and Bode plots of parametric(P) and nonparametric(NP) identification(Id) models

In identification, different window sizes lead to different frequency responses, $G(j\omega)$ of the plant. In the design method of this chapter the set of responses are accordingly represented as a nominal frequency response subject to unstructured multiplicative uncertainty, that is at each frequency ω, the uncertain $G(j\omega)$ is represented by a disk in the complex plane centred at a nominal $G_0(j\omega)$ with radius $\Delta(\omega)|G_0(j\omega)|$, where $\Delta(\omega)$ is the real valued multiplicative uncertainty. An approach is taken to achieve this with minimum unstructured uncertainty. The disk at each frequency is thus the smallest disk containing all the complex responses at that frequency [80]. The nominal model $G_0(j\omega)$ is obtained by finding the centre of the uncertain disk at each frequency. Figure 5.4 shows a typical uncertainty disk of the $G(j\omega)$ and the center $G_0(j\omega)$ of this disk.

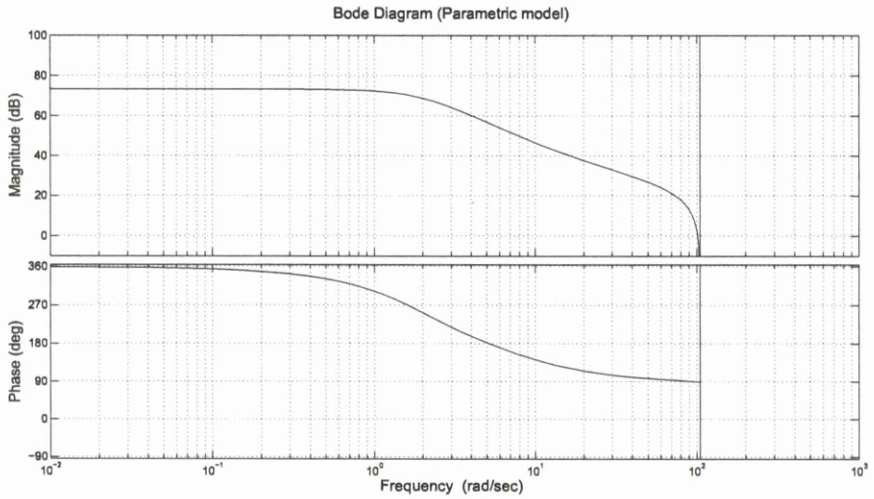

Figure 5.2: Bode plots of parametric model

5.4 Robust Controller Design by Analytical Optimisation

In this section, the parametric and nonparametric controllers are developed by analytical optimisation method based on the parametric and nonparametric plant presented in last section

5.4.1 Controller Design Requirements

For idle-speed control typical controller design the performance and robustness requirements are [3] [78]:

1. The engine speed deviation must remain within acceptable limits: $|y| \leq y_{max} = 320rpm$
2. The speed setting time should remain less than $\tau_{max} = 3s$
3. Adequate gain margin and delay-margin; The system should be robustly stable to perturbations in the transmission loop $L(s)$ of $\gamma L(s)e^{-\tau s}$ for all γ and τ such that $2/3 \leq \gamma \leq 3/2$ and $-5T_s \leq \tau \leq 5T_s$
4. The response overshoot should be less than 5 percent.
5. The loop noise should be rejected.

These requirements are translated into specifications in the weighting functions for primary sensitivity S and complementary sensitivity T functions. The details of the

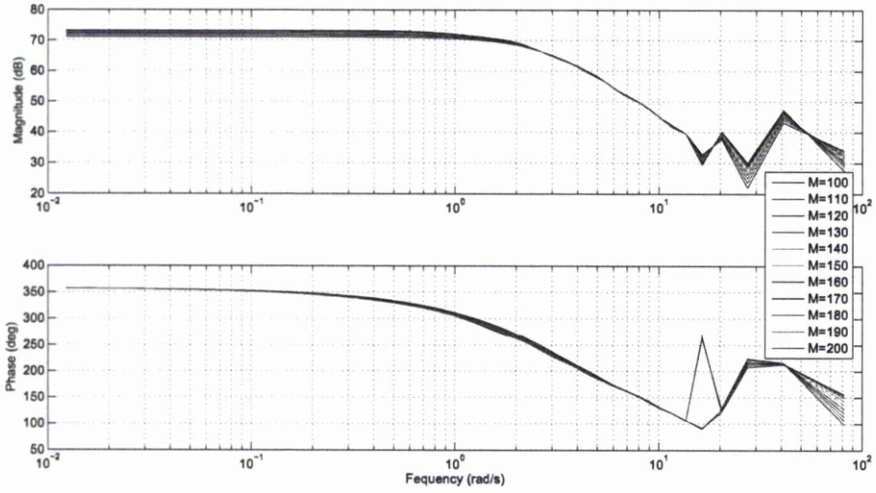

Figure 5.3: Bode plots of nonparametric models

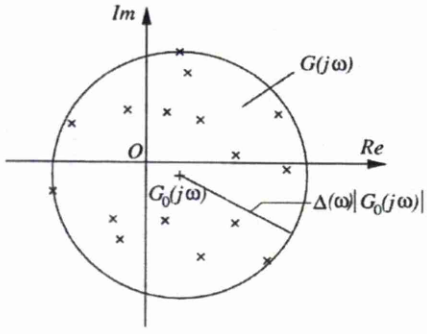

Figure 5.4: The uncertainty disk of the frequency response $G(j\omega)$

determination of the weightings for S and T can be found in [3]. The specifications of the weightings are as follows:

- From requirements 1 and 2, $|S|$ should be below the straight line going through the point $0.01rad/s, -40dB$ and with slope rate of $20dB/$decade at low frequency.
- Requirement 3 indicates that $|S| < 4dB$ at the high frequency range.
- Requirements 4 and 5 indicate that $|T| < 6dB$, have a roll off rate of $20dB/$decade and a small magnitude of approximately 0.1 $(-20dB)$ around the Nyquist rate of $100rad/s$ to ensure robustness to unmodelled system dynamics at high frequency

The corresponding weighting functions on $\omega = [0, \infty)$ for S and T are W_S and W_T and are given by the inverse weights

$$W_S^{-1} = \frac{1.57s}{s + 1.57} \tag{5.8}$$

$$W_T^{-1} = \frac{2}{0.2s+1} \tag{5.9}$$

5.4.2 'Disk method' Design and Simulation Results

These weighting functions may be expressed in the form of the centre and radius functions of the 'disk problem' for formulating a Γ function as follows: For the parametric design:

$$K(\omega) = \begin{cases} 1.0, 0.0 \leq |\omega| \leq 6 \\ 0.0, 6 < |\omega| \end{cases}$$

$$R(\omega) = \begin{cases} |W_S^{-1}|, 0.0 \leq |\omega| \leq 6 \\ |W_T^{-1}|, 6 < |\omega| \end{cases}$$

For the nonparametric design, with $P(j\omega)$ the nonparametric model:

$$K(\omega) = \begin{cases} 1.0/P(j\omega), 0.0 \leq |\omega| \leq 6 \\ 0.0/P(j\omega), 6 < |\omega| \end{cases}$$

$$R(\omega) = \begin{cases} |W_S^{-1}/P(j\omega)|, 0.0 \leq |\omega| \leq 6 \\ |W_T^{-1}/P(j\omega)|, 6 < |\omega| \end{cases}$$

The optimal $\hat{T}$ and $\hat{U}$ may then be found by solving the minimax problem for performance function Γ using analytical optimisation method. to perform a minimisation of the Γ function to obtain $\Gamma \leq 1$. The controller K may then be calculated from:

$$T = \frac{PK}{1+PK} \Rightarrow K = \frac{T}{P(1-T)} \tag{5.10}$$

or

$$U = \frac{K}{1+PK} \Rightarrow K = \frac{U}{1-PU} \tag{5.11}$$

The resulting optimal frequency responses for T in both parametric and nonparametric cases are shown in figure 5.5 and 5.6 respectively. with the constraint envelope determined by the weighting functions. The circular section of the envelope represents the weighted constraints at the different frequencies and the solid line is locus of the optimal sensitivity function T. From these figures it can be seen that the magnitude of optimal T from the NP method was constrained slightly better within envelope than that from the parametric method. This means the controller from the NP method can be expected to meet the initial design requirements better.

Figure 5.7 shows the simulated unit step responses using the controllers from the two methods. The responses of NP method are seen to have a smaller overshoot and superior settling as may be expected from the results of figures 5.5 and 5.6. Also, the overshoot and settling time from the NP method increases with the window size.

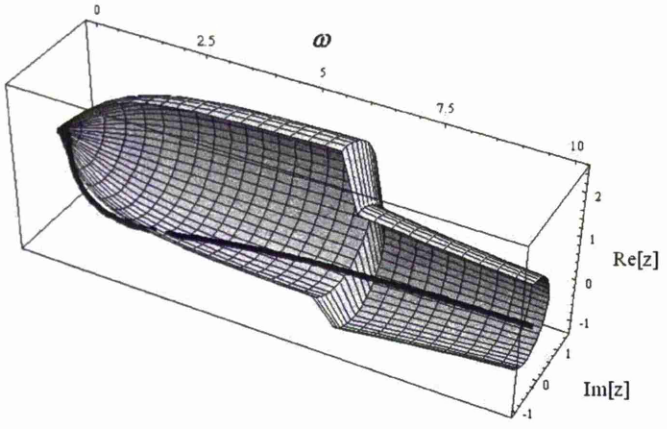

Figure 5.5: Constraint envelope for sensitivity function T (parametric method)

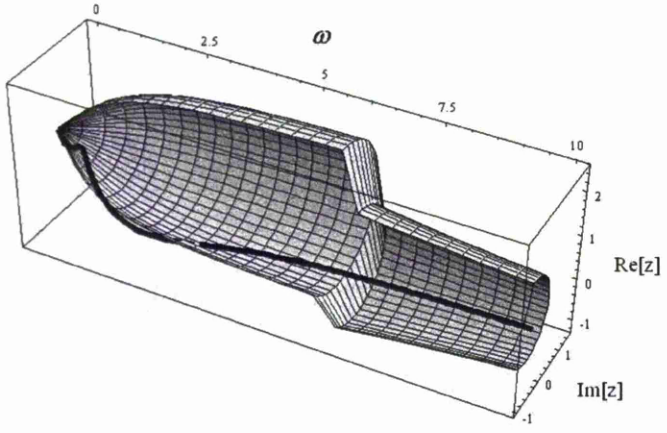

Figure 5.6: Constraint envelope for sensitivity function T (NP method)

5.4.3 'Mixed sensitivity' Design and Simulation Results

Besides solving the disk problem, the standard mixed sensitivity function problem with the constraints on T and S combined at all frequency range can also be solved, where the performance function can be written for both parameteric and NP as follows:
For the parametric design:

$$\Gamma(j\omega) = |W_S(1-T)|^2 + |W_T T|^2, \tag{5.12}$$

For the NP design, with plant $P(j\omega)$ we use:

$$\Gamma(j\omega) = |(W_S/P)(1/P-U)|^2 + |(W_T/P)U|^2, \tag{5.13}$$

Figure 5.8, 5.9 shows the constraints W_S^{-1} and W_T^{-1} for the optimal S and T. The optimal sensitivity functions from the mixed sensitivity method can meet the requirement of the weighting functions.

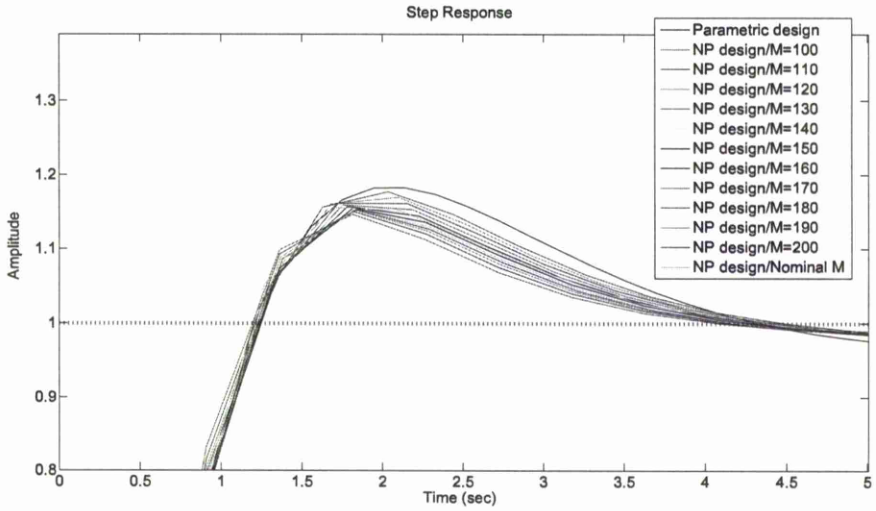

Figure 5.7: Unit step response of closed loop function, disk method

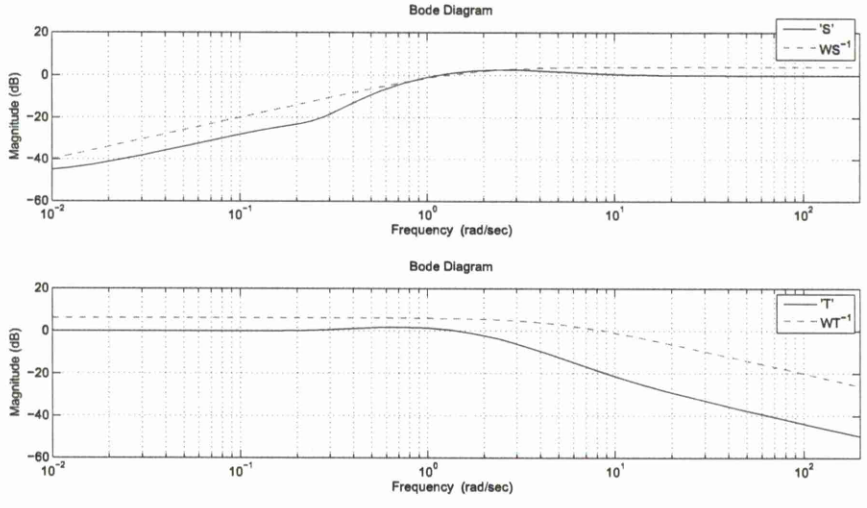

Figure 5.8: Bode magnitude constraints and optimal sensitivity loci of S and T in parametric mixed sensitivity design

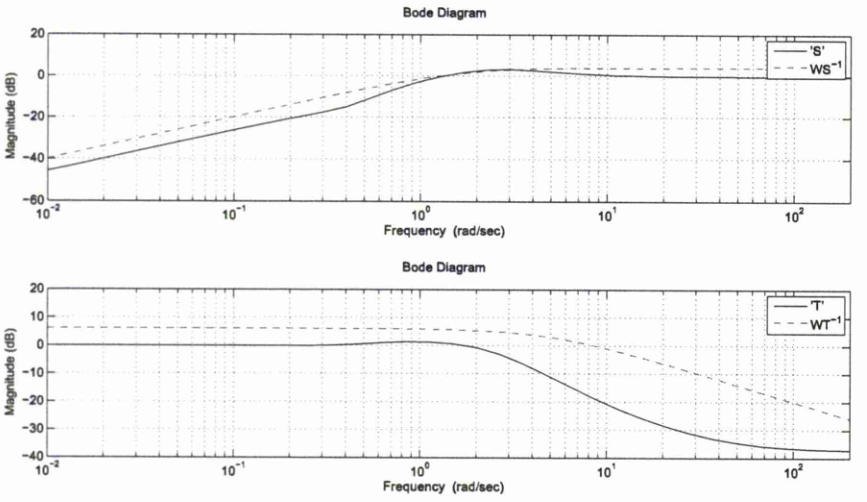

Figure 5.9: Bode magnitude constraints and optimal sensitivity loci of S and T in NP mixed sensitivity design

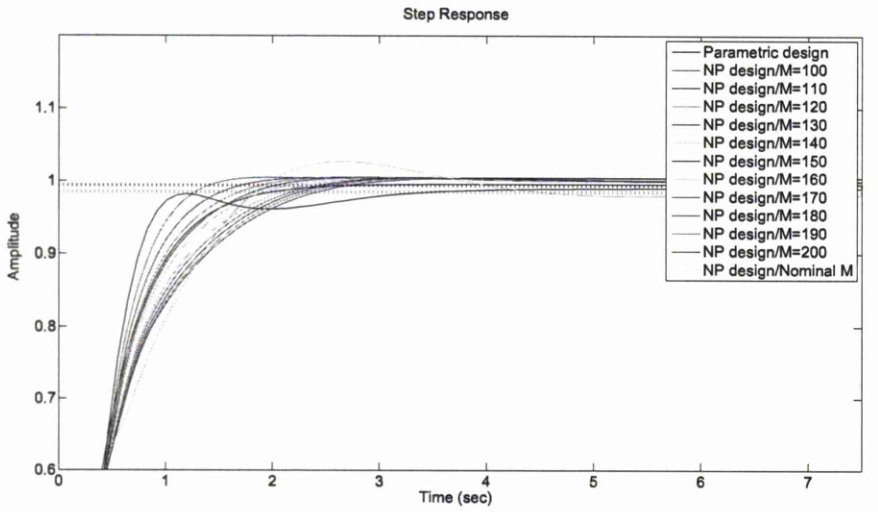

Figure 5.10: Unit step response of closed loop function, mixed-sensitivity functions method

Figure 5.10 also shows the simulated unit step responses of the mixed-sensitivity controllers. Comparing them with the controllers of the disk method, the mixed-sensitivity controllers are seen to have a smaller overshoot, but a slightly bigger steady-state error. Comparing the parametric and NP method, the steady-state error from the NP method is smaller than from the parametric method for most NP models. The settling times of the parametric and NP method are very similar.

Once the optimal T is obtained. The rational functions for S, T and K(controller) can be obtained from a CF(Caratheodory-Fejer) approximation[84], or simply by least square fitting method[56]. The 4th or 5th order rational functions were chosen for the controller to gave a high fitting rate to the optimal solutions.

5.4.4 Experimental Test on Engine

The disturbances rejection capability of the two controllers was experimentally tested. During the tests, a constant base torque load was applied to the engine to which a step reduction or a step increase in torque load was made by the electric dynamometer.

K_{np}	Gain Margin	Delay Margin
M=100	$3.4/10.6dB$	20 delays
M=110	$3.2/10.1dB$	19 delays
M=120	$3.3/10.4dB$	23 delays
M=130	$3.6/11.1dB$	20 delays
M=140	$3.4/10.6dB$	18 delays
M=150	$3.5/10.9dB$	17 delays
M=160	$3.5/10.9dB$	18 delays
M=170	$3.4/10.6dB$	18 delays
M=180	$3.6/11.1dB$	17 delays
M=190	$3.5/10.9dB$	17 delays
M=200	$3.4/10.6dB$	17 delays
Nominal	$3.7/11.4dB$	28 delays
K_p	$3.6/11.1dB$	20 delays

Table 5.1: Robustness test results of NP controllers,disk method

Figure 5.11 and 5.12 show the speed dip and flare responses resulting from the application of the positive or negative step torque load to the engine running with a demand speed of $880rpm$ with both parametric (K_p) and nonparametric (K_{np}) controllers which are designed by the disk method.

Initially, the large positive torque load causes a severe reduction in speed. The engine speed is then restored to the demand speed. Both satisfy the overshoot requirement of $|y| \leq y_{max} = 320rpm$. In experimental testing, the K_{np} with the windows sizes of $M = 100$ to 160 have a similar performance to K_P. In testing, sometimes K_{np} was

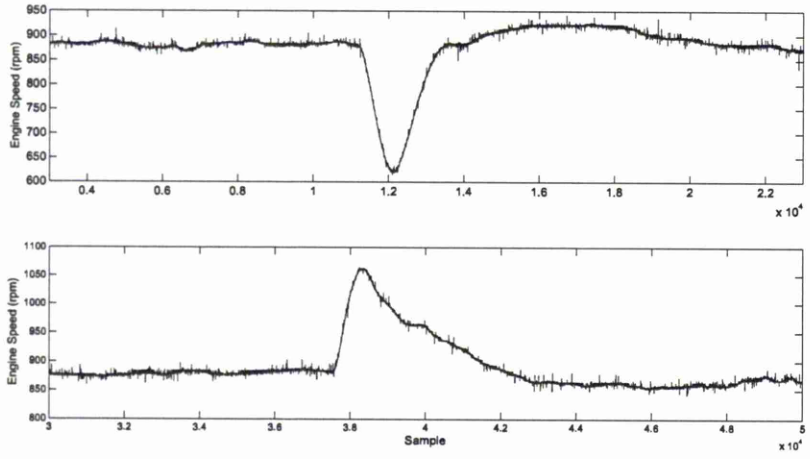

Figure 5.11: Experimental test results, parametric control of disk method

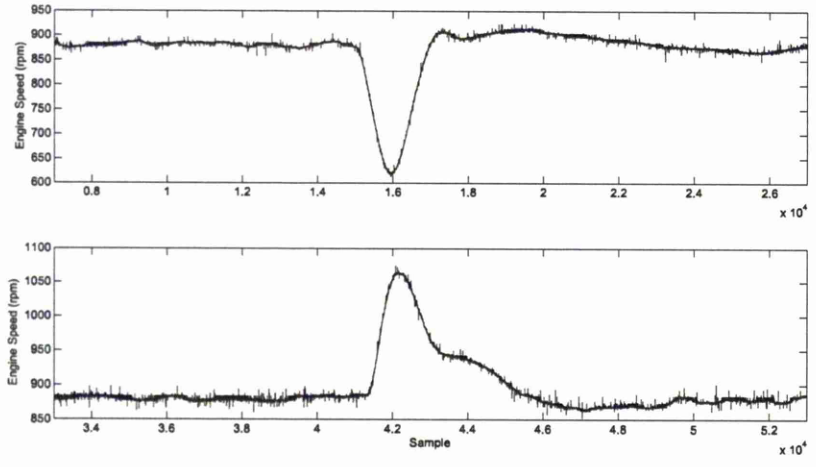

Figure 5.12: Experimental test results, nonparametric control of disk method/M=150

marginally better, othertimes K_P subject to minor environmental changes. For K_{np} with the windows sizes from $M = 170$ to 200, the performances are worse than the K_p with performance worsening with increasing window size. The controller designed from a nominal model has a similar quality performance to the K_{np} with $M = 100$ to 160. When the torque load is released, the nonparametric controller with a window size of $M = 160$ rejected the noise better as may be expected from the results of figures 5.5 and 5.6.

For robustness testing, gains and delays are introduced in series with the controller. The K_p had a gain margin of $3.6/11dB$ and a delay margin of 20 sample time delays. Corresponding to the different window sizes, the K_{np} stabilise up to a gain of 3.2 to $3.6/10dB$ to $11dB$ and 17 to 23 time delays. For the NP controller based on a nominal model, it is robust to the gain of $3.7/11dB$ and 28 time delays. In robustness testing, the nominal model based NP controller is better than K_p entirely. The Table 5.1 shows the robustness test results.

K_{np}	Gain Margin	Delay Margin
M=100	$1.8/5.10dB$	20 delays
M=110	$1.9/5.58dB$	21 delays
M=120	$1.7/4.60dB$	20 delays
M=130	$1.8/5.10dB$	20 delays
M=140	$1.8/5.10dB$	21 delays
M=150	$1.9/5.58dB$	23 delays
M=160	$1.7/4.60dB$	20 delays
M=170	$1.6/4.08dB$	19 delays
M=180	$1.7/4.60dB$	19 delays
M=190	$1.6/4.08dB$	20 delays
M=200	$1.5/3.52dB$	19 delays
Nominal	$1.8/5.10dB$	27 delays
K_p	$1.6/4.08dB$	18 delays

Table 5.2: Robustness test results of NP controllers, mixed sensitivity functions method

Figures 5.14 and 5.13 show the results from the mixed sensitivity controller under the same testing environment. In this experimental testing, the overshoot of the engine speed is smaller than that of the disk method controller, and the overshoot when using the K_{np} decreases slightly with increasing window size. The performance of K_{np} with windows size from $M = 160$ to $M = 200$ are the same or slightly better than that of the K_p. In figure 5.14 with window size is $M = 150$, the performance of the controller is slightly worse than the K_p. The controller designed from a nominal model is also slightly worse than with the K_p. Although the mixed sensitivity method gives better nominal performance than the disk method, a little steady-state error still exists as shown in figure 5.10 and there is some oscillation when the engine speed recovers from

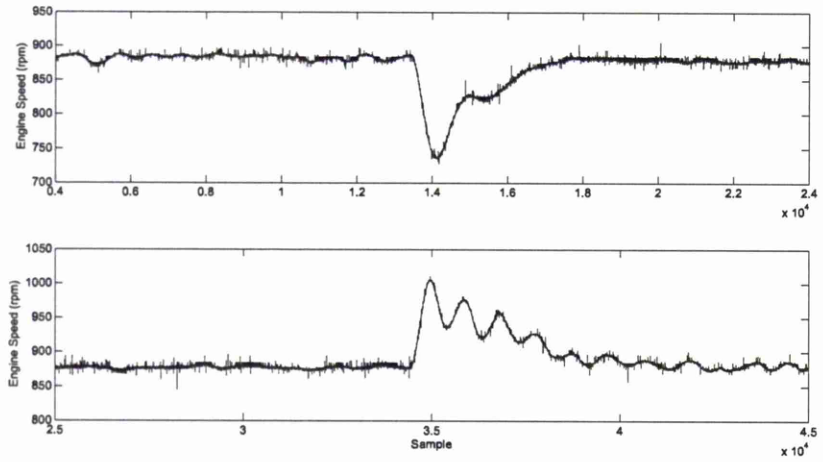

Figure 5.13: Experimental test results, parametric control of mixed sensitivity method

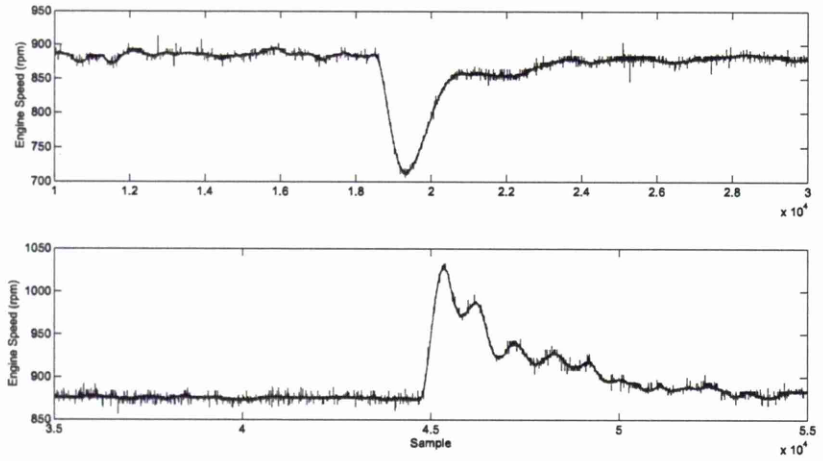

Figure 5.14: Experimental test results, nonparametric control of mixed sensitivity method/M=150

the transient disturbance.

In robustness testing, the K_{np} from the mixed sensitivity functions method is robust to $1.5/3.52dB$ to $1.9/5.58dB$ and 15 to 24 time delays. The K_p is robust to $1.6/4.08dB$ and 18 time delays. The K_{np} based on the nominal model is robust to $1.8/5.10dB$ and 27 time delays, which is the same as the controller based on the nominal model using the disk method. The Table 5.2 shows the complete outcome of the robustness tests.

From the robustness testing, the controllers based on the multiple window nominal model using either the disk method and the mixed sensitivity method have a significantly better robustness in time delay (delay margin) than other controllers.

5.5 CAN-bus Based Embedded System Implementation for Idle Speed Controller

Once the idle speed controller is developed and throughly tested, it can be implemented to the target controller hardware device which in this work was the dsPIC microcontroller from Microchip. For investigative purposes, the idle speed controller was implemented in the dsPIC microcontroller and used to control the engine speed at idle through a CAN-bus. This is the standard data bus in automotive industry, by which the microcontrollers and chips on a modern vehicle communicate. The objective of this work was to build a CAN-bus based controller which can be considered as a part of a distributed control system for the automotive engine. Such distributed control systems may be employed in advanced applications like electric motor/IC engine hybrid systems. Because the CAN-bus can become overloaded with message packets when busy, CAN based controller need to be tolerant to the resulting time delay. The experimental set-up allows the robustness to CAN-bus overloading to be tested experimentally.

5.5.1 Implementation of Controller

When the peripheral modules of the dsPIC were configured, the program including the implementation of the controller was developed. The SIMULINK model representing the program was created as shown in figure 5.15. In figure 5.15, the basic configurations of ADC, CAN and DMA module were set in the corresponding SIMULINK blockset as above. The implementation of the controller and CAN transmission system were set as below.

First, the ADC output data was set as the input for the idle speed controller. To read the output data of the ADC conversion and to write it to the other register, 'ADC Read' blockset was used in the SIMULINK model. For convenience, the output data

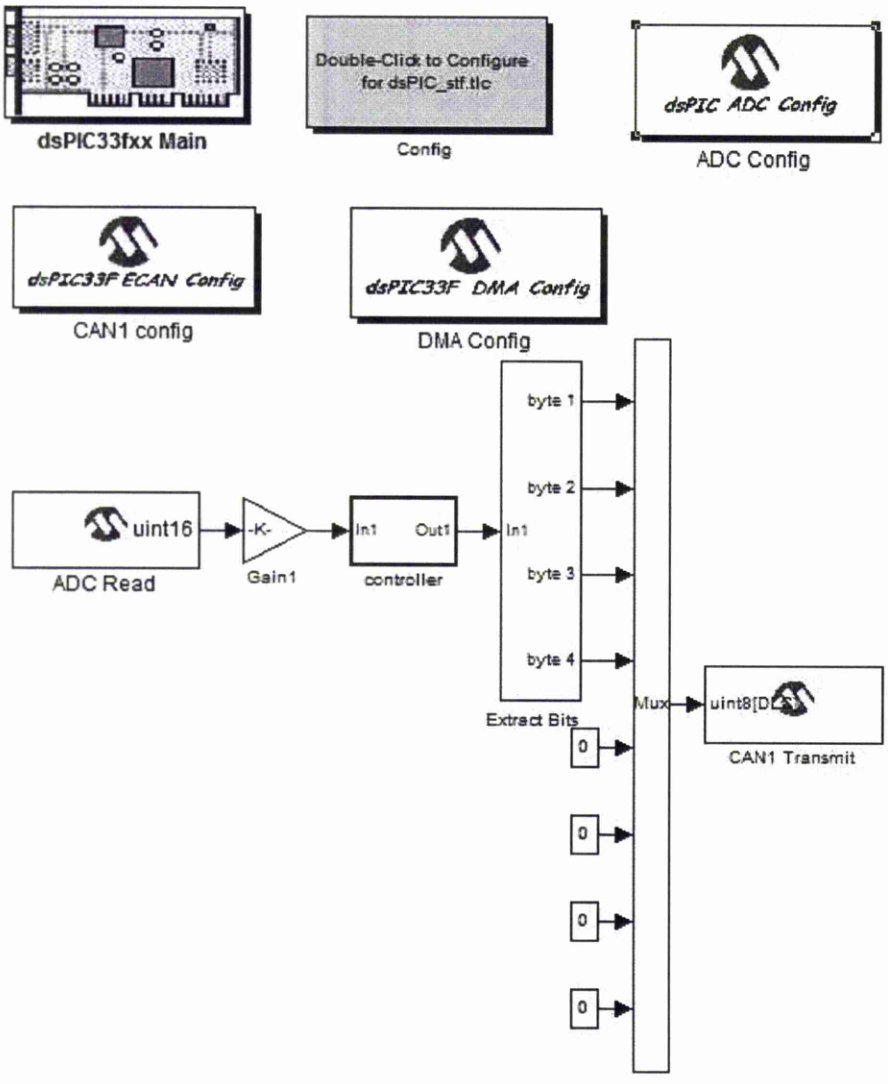

Figure 5.15: SIMULNIK model of idle speed controller

of the ADC can be read in the required data format by selecting 'output data type' for ADC Read blockset. In this work, 'real32' (signal precision) was chosen. The dSPACE ds4302 CAN card can assemble the data for receiving messages automatically up to 32-bit. So 32bit operation in the dsPIC's program could make the data transmission and assembly straight forward. For the application of ABV control, 32-bit provides adequate resolution. As the voltage references is 3.3/0 volts, the maximum output data of ADC 4095 corresponds to 3.3 volts which is equivalent to 1667rpm. To scale the output data of the ADC down to the value of equals the measured engine speed, a 'Gain' block which equals to 1667/4095 was added to the 'ADC Read' block. The engine speed signal was a positive double precision number.

Then the engine speed signal was fed back to the controller. Because the

dsPIC33FJ256GP710 supports floating point numbers, the controller can be implemented by using the SIMULINK blocksets such as 'transfer function','state space', 'zero-pole-gain', etc. with the decimal format parameters directly. Due to the good robustness found on delay margin, the nonparametric controller based on the nominal model developed using the mixed sensitivity method by analytic function optimisation was implemented. In experimental test, it was found that the implementation using 'zero-pole-gain' blockset worked correctly in the for dsPIC, however other blocksets for transfer function implementation contained a bug and failed to work properly at the time of writing this bug had not yet been corrected by Microchip Corporation. The reason might be due to the particular system target file ('.tlc' file) or C compiler. Besides using the 'zero-pole-gain' blockset, a 'Saturation' blockset was connected in series with 'zero-pole-gain' blockset in series to bound the output of controller between 0(0% for ABV duty) to 1(100% for ABV duty). Those blocksets were built in the 'Controller' subsystem in the model.

After the controller was implemented, the controller's output signal was to be transmitted to dSPACE through CAN-bus. Because data can only be written to the 'CAN Transmit' blockset byte by byte, the 32-bit format data for transmission was divided into 4 bytes. This was done by using SIMULINK's default 'Extract Bits' blockset. The blocks were built in the corresponding subsystem. Using the C language based programming approach however, the data for transmission can be written to the message buffer word by word. This is however a mirror shortcoming for Microchip's target device blocksets. In the 'CAN Transmit' blockset, DLC was set as 4 as should be expected. The data was set to be transmitted as standard format with identifier = 100.

Once the SIMULINK model was complete, the model was able to be converted to HEX code directly in the MATLAB environment. The generated code could than be loaded to the dsPIC by the programmer using the MAPLAB IDE system.

5.5.2 Model Developement for dSAPCE

When the programming for the dsPIC was complete, the corresponding SIMULINK model for dSPACE could be also established. Figure 5.16 shows the SIMULINK model for dSPACE. To receive the data from the dsPIC via CAN-bus, the ds4302 CAN card should be configured first. In 'CAN CONTROLLER SETUP' blocksets, the baud rate, bit time segments was set the same as the setup in dsPIC. In the CAN message receive blockset ('DS4302CAN_RX_B1_C1'), the identifier was set as 100; the data length was set as 4; and the data format was 'float 32'. The output data of the CAN receiver block was connected to a 'switch', a constant value could than be selected by means of

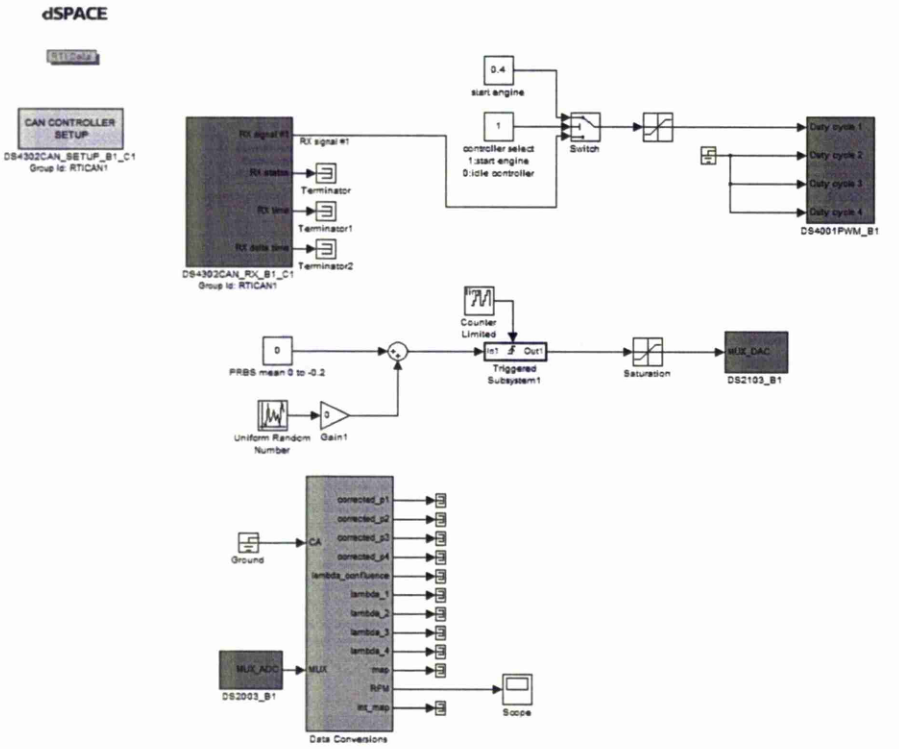

Figure 5.16: SIMULNIK model for dSPACE

the switch to start the engine. Then the data was sent to the ds4001 PWM blockset. In that block, the frequency for the PWM was set as 33Hz. The remaining part of the model was used for other purposes. In particular the model with the ds2103 DAC blockset was used to control the dynamometer; and the model with the ds2003 ADC blockset was used to collect the data.

5.5.3 Experimental Results

Once all software and hardware was completely configured completely, the MS analytic function optimised idle speed controller was implemented in the dsPIC and tested on the engine dynamometer. In experiment, a similar disturbance test was carried out. Figure 5.17 shows the experimental results with disturbance of positive and negative step load as loaded by the dynamometer. The idle speed controller is found to be able to regulate the engine speed around idle and reject the disturbance of extra load as required. Compared to the results from the previous controller implemented directly in dSPACE, it is found that the variation of engine speed controlled by the dsPIC33 controller is large at steady state, while the variation at steady state is very small for the dSPACE controller. The reason might be due to the hardware structure. The dsPIC33 is a 16-bit Digital Signal Processor (DSP), and the ADC module is up to 12-bit. The CPU

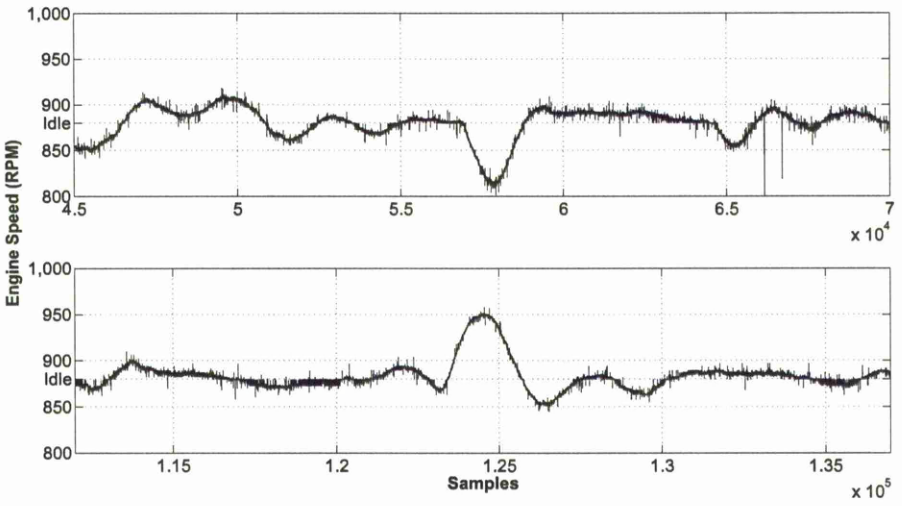

Figure 5.17: Experimental results for the idle speed controller implemented in dsPIC

for dSPACE however is 32-bit and its ADC card is up to 16-bit. The dSPACE is more precise than dsPIC33. The results presented here are nevertheless useful for validation purposes. From the results, the CAN-bus based embedded system implementation of idle speed controller is successful in its overall regulation performance.

5.6 Conclusions

The H_∞ analytical optimisation controller design methods were applied to the design of an idle speed automotive engine controller using the multiple sensitivity-function split-frequency 'disk method' and the mixed sensitivity method. The resulting NP controllers based on different windows sizes were experimentally tested.

The experimental results were compared with those of a parametric design. Experimental testing shows a NP controller design method with an appropriate window size can reject the disturbances and achieve settling similar to the parametric method when using the disk method. If a nominal model based on the centre of the frequency response from different window sizes is used in the NP controller design, a more robust system is obtained than with the parametric design method. If in other applications the nonparametric and parametric models have significantly different frequency responses, the NP method may offer further improved performance.

The nonparametric idle speed controller was implemented in a dsPIC33 target device using a the new fast prototyping system from Microchip Corporation based on SIMULINK's graphical model. The whole control system was also implemented using a CAN-bus communication between dsPIC and dSPACE such as might be implemented

in a hybrid vehicle. The experimental results show the implementation works effectively including with the software/hardware configuration and programming was successful. The nonparametric approach with its increased delay margin was shown to provide a suitable controller for such CAN networks where overload on the bus could cause significant delays.

In future, more applications of the NP design method should be theoretically and experimentally investigated including for MIMO problems to explore their possible advantages over parametric methods. The implementation of MIMO controller in the dsPIC target device is also to be explored. The problem of an NP design method for unstable systems remains an open problem, though is of less importance in automotive applications.

Chapter 6

Multivariable H_∞ Optimal Control by Analytical Optimisation

6.1 Introduction

To design the controller for a multivariable system, a simple way is to design several SISO controllers for each input and output pair. This way is often implemented in industry. However, each output of a multivariable system is usually affected by two or more inputs. For example, the engine speed is normally controlled by the throttle (air path) in Engine Management Systems (EMSs). While the air-fuel ratio control problem was treated as a SISO case historically, and was regulated by fuel path only. Although the effectiveness is weaker than in the air path, the fuel path and spark advance (SA) also affect the engine speed. Although the air path affects the air-fuel ratio less than fuel path, the change in air path still causes the variation of air-fuel ratio. The interaction between several different inputs and outputs forms a multivariable system. For simplicity, those problems still can be treated by using a SISO approach. The output is controlled by one input and the interaction from the other inputs are treated as the disturbances which can be modelled as the uncertainty of the plant in controller design. However, the resulting transient performance for SISO controller may be compromised and poor, as also may be the robustness.

The poor performance of SISO controllers applied to multivariable systems have led to multiple-input-multiple-output (MIMO) design algorithms. Obviously, the relations and interactions between different inputs and outputs can be represented as a matrix of input and output signals. To handle the interactions robustly, the H_∞ MIMO control algorithms develop the control requirements in terms of the maximum singular values of the transfer function matrix. So the system's performance and robustness for such

MIMO design generally produces better outcomes than SISO loop-by-loop design. In MIMO designs, in general, the interactions can be controlled but can not be eliminated. Each output is still affected by some unwanted inputs.

Decoupling Control

To obtain better performance by overcoming the interactions or coupling of a multivariable system, various decoupling control design techniques have been developed. From Morgan and Glover's descriptions[65] [21], the decoupling is concerned with the problem of designing a controller such that specific input-output pairs of the closed-loop system are 'isolated' from each other, i.e., the cross-coupling between the groups is zero. More precisely, a system is decoupled if the closed loop transfer function matrix of the system is block diagonal:

$$C(sK - A)^{-1}B = \begin{bmatrix} H_1(s) & 0 & \dots & 0 \\ 0 & H_2(s) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & H_k(s) \end{bmatrix} \tag{6.1}$$

Many decoupling control design techniques have been developed to solve the block diagonal problem for closed loop transfer function matrix, such as [61],[75],[92]. In these literatures, the presented simulation results indicated those technique can eliminate the cross-coupling between the different inputs and outputs completely. Most of these techniques are based on state-space form algorithm. A novel and simple approach for the design of decoupling linear feedback controller for MIMO plants is proposed in this chapter. The approach proposed is based on Helton and Merino's analytic functions optimisation technique. By allowing the design requirements to be stated in terms of the Hadamard product, the diagonal elements and off-diagonal elements of the the closed loop transfer function matrix are bounded by the weighting matrix individually on an element-by-element basis. The H_∞ Frobenius norm is used to replace the maximum singular value to allow for the element-by-element optimisation.

In this chapter, the proposed decoupling method is applied experimentally to solve the air-fuel ratio/torque control problem which is the basic control strategy (structure) for modern EMSs. In modern vehicles, the driver requests a certain amount of torque by pressing or releasing the accelerating pedal, air and fuel paths then respond to the torque request. This action causes a deviation of air-fuel ratio. However the air-fuel ratio is required to be always maintained at a fixed level, generally at the stoichiometric ratio for the particular fuel. This is a typical coupling problem. The proposed decoupling method is implemented to try to reduce the effect of the torque request on the air-fuel ratio control.

This chapter is organised as follows. First, the internal stability conditions for multivariable case are introduced in section 6.2.1. Then the interpolation constraints to achieve internal stability are presented in section 6.2.2. The interpolation scheme for the controller optimisation is then derived in section 6.2.3. The alternative method to achieve internal stability for nonparametric design is presented in in section 6.2.4. In the section 6.3 the MIMO H_∞ Frobenius norm, Hadamard weighted, decoupling controller design method is developed, both the parametric and nonparametric design approaches are presented. Then the decoupling controller is applied experimentally to the air-fuel ratio/torque control problem. In the section 6.4 a 2×2 system based on a novel nonlinear AFR estimator is introduced, and identifications for nonlinear AFR estimator and linear MIMO system are both detailed. The decoupling controller is then designed and applied to the engine. In section 6.5, the experimental results compared with the Riccati controller are displayed.

6.2 Internal Stability

As in the SISO case, the internal stability of a multivariable system requires that all transmissions of inputs to nodes in the feedback loop are stable. The conditions are normally stated in either state-space form or in terms of rational transfer function representations of the controller and plant. Assume that G is plant and K is controller in a closed loop unity negative feedback system of the figure 6.1: the constitutive

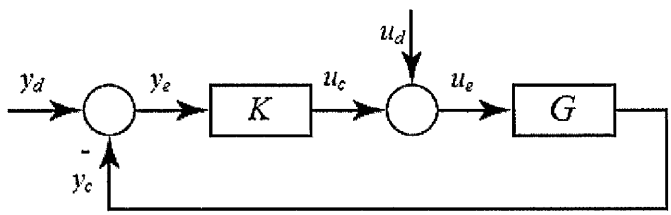

Figure 6.1: Negative unity feedback disturbance system block diagram

equations for that system are:

$$y_c = Gu_e$$
$$u_c = Ky_e$$
$$y_e = y_d - y_e$$
$$u_e = u_d + u_e$$

all closed loop transmissions (sensitivity functions) for MIMO case are:

$$S = [I + GK]^{-1} \tag{6.2}$$

$$T = GK[I + GK]^{-1} = [I + GK]^{-1}GK \tag{6.3}$$

$$KS = K[I + GK]^{-1} = [1 + KG]^{-1}K \tag{6.4}$$

$$SG = G[I + KG]^{-1} = [1 + GK]^{-1}G \tag{6.5}$$

$$\hat{S} = [I + KG]^{-1} \tag{6.6}$$

$$\hat{T} = KG[I + KG]^{-1} = [I + KG]^{-1}KG \tag{6.7}$$

where S is termed the Primary Sensitivity;T is termed Complementary Sensitivity; KS is termed Control-effort Sensitivity; $\hat{S}$ is termed Dual-primary Sensitivity; $\hat{T}$ is Dual-complementary Sensitivity;SG is termed Plant Input Disturbance Sensitivity. Now since the $T = I - S$, stability of $T$is determined by the stability of S. Similarly, the stability of $\hat{T}$ is determined by $\hat{S}$. So the internal stability is established by the stability of the four sensitivity functions:

$$S = [I + GK]^{-1} \tag{6.8}$$

$$KS = K[I + GK]^{-1} = [I + KG]^{-1}K \tag{6.9}$$

$$SG = [I + GK]^{-1}G = G[I + KG]^{-1} \tag{6.10}$$

$$\hat{S} = [I + KG]^{-}1 \tag{6.11}$$

6.2.1 Internal Stability Conditions for Multivariable Case

As in the SISO case, a practical method to ensure the internal stability during optimisation of the controller is an interpolation method. Before deriving the interpolation method, the internal stability constraints are now introduced. As mentioned in the SISO case, the pole-zero cancellation of plant G and controller K is a key issues causing the closed loop to be unstable. Zhou presented an internal stability condition based on the viewpoint of pole-zero cancellations[101]. The description of the condition is as follows: assume that there are no Right Half Plane (RHP) pole-zero cancellations between G and K. In another word, all RHP-poles in G and K are contained in the minimal realizations of GK and KG. Then the closed loop feedback system is internally stable if and only if one of the four sensitivities (equation 6.8 to 6.11) is stable. So that an interpolation method can be derived based on the Zhou's theory, and the main motivation of the interpolation method is to avoid the RHP pole-zero cancellation of G and K.

To avoid the RHP pole-zero cancellation of G and K, the best way is to keep the RHP zeros or poles in the closed loop system transmissions. Two essential results apply when a feedback system is internally stable [96], [80]:

1. If G has a RHP zero at z, then $L = GK$,$T = GK[I + GK]^{-1}$,$SG = [I + GK]^{-1}G$,$L_I = KG$ and $\hat{T} = KG[I + KG]^{-1}$ will have a RHP zero at z respectively;

2. If G has a RHP pole at p, then $L = GK$ and $L_I = KG$ will each have a RHP pole at p, the $S = [I + GK]^{-1}$,$K = K[I + GK]^{-1}$ and $\hat{S} = [I + KG]^{-1}$ will have a RHP *zero* at p.

6.2.2 Interpolation Constraints for Multivariable Case

From Zhou's results, the closed loop system is internally stable if one of the four sensitivities (equation 6.8 to 6.11) and the following interpolation constraints, which were given by Skogestad and Postlethwaite [80], are satisfied. The interpolation constraints are stated for the plant G with the simple RHP zeros and the G with the simple RHP poles respectively.

Plant Has RHP Zero

Assume that G has a RHP zero at $s = z$, and y_z is the output zero direction of z, and u_z is the input zero direction of z. Then the interpolation constraints for sensitivities T and S to avoid the RHP pole-zero cancellations of the GK and KG are:

$$y_z^H T(z) = 0, \quad y_z^H S(z) = y_z^H \tag{6.12}$$

and

$$\hat{T}(z)u_z = 0, \quad \hat{S}(z)u_z = u_z \tag{6.13}$$

where X^H represents the conjugate transpose of X. Because the $\hat{S}(z)u_z = [I + KG]^{-1}u_z = u_z[I + GK]^{-1} = u_z S(z)$, then equation 6.13 can be written to:

$$u_z S(z) = u_z, \quad u_z T(z) = 0 \tag{6.14}$$

Then the interpolation constraints for sensitivity T are:

$$y_z^H T(z) = 0 \ \& \ u_z T(z) = 0 \tag{6.15}$$

For sensitivity S, the constraints are:

$$y_z^H S(z) = y_z^H \ \& \ u_z S(z) = u_z \tag{6.16}$$

To obtain a practical algorithm the interpolation constraints can be expressed as the following representations:

$$T(z) = 0, \quad S(z) = I \tag{6.17}$$

Plant Has RHP Pole

Assume that G has a RHP pole at $s = p$, y_p is the output pole direction of p, and u_p is the input pole direction of p. The interpolation constraints for sensitivities T and S are then:

$$S(p)y_p = 0, \quad T(p)y_p = y_p \tag{6.18}$$

and

$$u_p^H \hat{S}(p) = 0, \quad u_z^H \hat{T}(p) = u_z^H \tag{6.19}$$

In an approach similar to the case above, because $u_p^H \hat{S}(p) = u_p^H [I + KG]^{-1} = [I + GK]^{-1} u_p^H = S(p) u_p^H$, the interpolation constraint for sensitivity T can be written as:

$$T(p)y_p = y_p \ \& \ T(p)u_p^H = u_p^H \tag{6.20}$$

For sensitivity S, the constraints can be written as:

$$S(p)y_p = 0 \ \& \ S(p)u_p^H = 0 \tag{6.21}$$

To obtain a practical algorithm the interpolation constraints can be expressed as the following representations:

$$T(p) = I, \quad S(p) = 0 \tag{6.22}$$

Plant Has RHP Zero and Pole

If the plant has a RHP zero at $s = z$ and a RHP pole at $s = p$, the interpolation constraints can be combined with equation 6.17 and 6.22 to give:

$$T(z) = 0 \ \& \ T(p) = I, \quad S(z) = I \ \& \ S(p) = 0 \tag{6.23}$$

6.2.3 Parametrisation of Interpolation

To ensure S or T to satisfy the interpolation constraints above, the auxiliary interpolation function can be applied. The method depends on numbers and locations of the RHP zeros and poles from the transfer matrices of the plant. The parametrisations of the interpolation function are different in the three cases: plant only has RHP zeros, plant only has RHP poles and plant both has RHP poles and zeros.

Plant Only Has RHP Zeros

The interpolation function is a scalar function if the plant only has RHP zeros at $s = z_1, ..., z_n$. Assume that the sensitivity T is stable, then the closed loop system is internally stable if the sensitivity T satisfies the interpolation constraint of equation

6.17. According to that equation, we have $T(z_1) = 0, ..., T(z_n) = 0$. To satisfy those equalities, sensitivity T should have the same RHP zeros at $s = z_1, ..., z_n$. Introduce a scalar:

$$\alpha = \frac{(s - z_1)...(s - z_n)}{(s + a_1)...(s + a_n)} \tag{6.24}$$

where $a_1, ..., a_n$ are the positive numbers, so α is stable and proper obviously. Then the sensitivity function (matrix) $T(s)$ can be given by:

$$T(s) = \alpha(s) \times T_0(s) \tag{6.25}$$

where $T_0 \in RH_\infty$, so that $T \in RH_\infty$ too. Obviously, equation 6.25 satisfies the equalities of the interpolation constraints $T(z_1) = 0, ..., T(z_n) = 0$.

Plant Only Has RHP Poles

As in the first case, the interpolation function is a scalar function if the plant has only RHP poles at $s = p_1, ..., p_n$. Assuming that the sensitivity S is stable, then the closed loop system is internally stable if the sensitivity S satisfies the interpolation constraint of equation 6.22. From that equation, we have $S(p_1) = 0, ..., S(p_n) = 0$. The transfer function of the sensitivity S also should have the RHP zeros at $s = p_1, ..., p_n$. Similarly, introduce a scalar:

$$\alpha = \frac{(s - p_1)...(s - p_n)}{(s + a_1)...(s + a_n)} \tag{6.26}$$

where $a_1, ..., a_n$ are also the positive numbers, α is stable and proper. Then the sensitivity function matrix $S(s)$ can be given by:

$$S(s) = \alpha(s) \times S_0(s) \tag{6.27}$$

where $S_0 \in RH_\infty$, so that $S \in RH_\infty$ too. Obviously, equation 6.27 satisfies the equalities of the interpolation constraints $S(p_1) = 0, ..., S(p_n) = 0$.

Plant Has RHP Zeros and Poles

If the plant has both RHP zeros $z_1, ..., z_n$ and RHP poles $p_1, ..., p_m$, interpolation function (matrix) is more complicated. Assuming that sensitivity T is stable, the closed loop system is internally stable if T satisfies the equation 6.23. From that equations, we have $T(z_1) = 0, ..., T(z_n) = 0$ and $T(p_1) = I, ..., T(p_m) = I$. Firstly, a similar scalar is introduced:

$$\alpha = \frac{(s - z_1)...(s - z_n)(s - p_1)...(s - p_m)}{(s + a_1)...(s + a_{n+m})} \tag{6.28}$$

where the number $a_1, \ldots, a_2$ are positive. Obviously $\alpha = 0$ at $s = z_1, ..., z_n$ and $s = p_1, ..., p_m$. Then introduce another auxiliary matrix A, which has same the dimension

as the sensitivity function matrix T and each diagonal element of A is a scalar A^*:

$$A = \begin{pmatrix} A^* & 0 & \cdots & 0 \\ 0 & A^* & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A^* \end{pmatrix} \tag{6.29}$$

where

$$A^* = c_1 \frac{(s-p_1)(s-z_1)...(s-z_n)}{(s+a_1)...(s+a_{n+1})} + c_2 \frac{(s-p_2)(s-z_1)...(s-z_n)}{(s+a_1)...(s+a_{n+1})} + ... + c_m \frac{(s-p_m)(s-z_1)...(s-z_n)}{(s+a_1)...(s+a_{n+1})} \tag{6.30}$$

Where $c_1, ..., c_m$ are the unknown coefficients. Obviously $A^* = 0$ at $s = z_1, ..., z_n$, which implies to matrix $A = 0$, Let $A^* = 1$ at $s = p_1, ..., p_m$, then matrix $A = I$ at $s = p_1, ..., p_m$. The sensitivity function (matrix) T can be given by:

$$T(s) = A + \alpha \times T_0(s) \tag{6.31}$$

Because $\alpha = 0$ and $A = 0$ at $s = z_1, ..., z_n$, then $T(z_1) = 0, ..., T(z_n) = 0$; because $\alpha = 0$ and $A = I$ at $s = p_1, ..., p_m$, then $T(p_1) = I, ..., T(p_m) = I$. The interpolation constraints (equation 6.23) are satisfied.

The unknown coefficients $c_1, ..., c_m$ can be calculated from the equalities $A^*(p_1) = 1, ..., A^*(p_m) = 1$. The following equations set are established to compute the $c_1, ..., c_m$:

$$\begin{aligned}
A^*(p_1) = 1 \Rightarrow\ & c_2 \frac{(p_1-p_2)(p_1-z_1)...(p_1-z_n)}{(p_1+a_1)...(p_1+a_{n+1})} + c_3 \frac{(p_1-p_3)(p_1-z_1)...(p_1-z_n)}{(p_1+a_1)...(p_1+a_{n+1})} \\
& + ... + c_m \frac{(p_1-p_m)(p_1-z_1)...(p_1-z_n)}{(p_1+a_1)...(p_1+a_{n+1})} = 1 \\
A^*(p_2) = 1 \Rightarrow\ & c_1 \frac{(p_2-p_1)(p_2-z_1)...(p_2-z_n)}{(p_2+a_1)...(p_2+a_{n+1})} + c_3 \frac{(p_2-p_3)(p_2-z_1)...(p_2-z_n)}{(p_2+a_1)...(p_2+a_{n+1})} \\
& + ... + c_m \frac{(p_1-p_m)(p_1-z_1)...(p_1-z_n)}{(p_1+a_1)...(p_1+a_{n+1})} = 1 \\
& \vdots \\
A^*(p_m) = 1 \Rightarrow\ & c_1 \frac{(p_m-p_1)(p_m-z_1)...(p_m-z_n)}{(p_m+a_1)...(p_m+a_{n+1})} + c_2 \frac{(p_m-p_2)(p_m-z_1)...(p_m-z_n)}{(p_m+a_1)...(p_m+a_{n+1})} \\
& + ... + c_{m-1} \frac{(p_m-p_{m-1})(p_m-z_1)...(p_m-z_n)}{(p_m+a_1)...(p_m+a_{n+1})} = 1
\end{aligned} \tag{6.32}$$

It is easy to find the solutions for the unknown coefficients $c_1, ..., c_m$. The interpolation matrix A can be parametrised by these coefficients.

6.2.4 Nonparametric Method

To implement the previous interpolation method the RHP poles and zeros of the plant are required to be know explicitly. This requires a parametric representation of the

plant. Although zeros are not obviously seen explicitly in the nonparametric plant, them should belong to the determinant of matrix. For the case of a plant having RHP zeros only, the sufficient internal stability constraint requires that the scalar interpolation function α is stable and contain the RHP zeros. For a stable square nonparametric plant $G(j\omega)$, the nonparametric interpolant $\alpha(j\omega)$ can be determined from the determinant of the frequency response matrix.

Beside the interpolation method, an alternative method also can be used to ensure the internal stability. As in the SISO case, the internal stability of the multivariable closed loop system can be determined by the control effort sensitivity KS when the plant is stable. Let us recall the internal stability conditions presented before to prove this. The internal stability of the closed loop system is determined by four sensitivities S, KS, GK and $\hat{S}$. Assuming that the plant G and KS are stable, the sensitivity $T = GK[I + GK]^{-1} = G \times KS$ is obviously stable. Then the $S = I - T$ is stable too. So that the $SG = [1 + GK]^{-1}G = S \times G$ is table. Because $\hat{T} = [I + KG]^{-1}KG = K[I + GK]^{-1}G = KS \times G$, then $\hat{T}$ and $\hat{S} = I - \hat{T}$ are stable. The closed loop system is then internally stable. No poles and zeros are required to be determined in this idea, so this method can be employed in the nonparametric multivariable controller design.

6.3 Multivariable H_∞ Optimal Control Using H_∞ Frobenius Norm

In this section an H_∞ optimal control method allowing element-by-element weighting and thus also decoupling is proposed. As mentioned in the last section, the H_∞ Frobenius norm of the matrix can be optimised by using the analytical optimisation method. So the Multiple-Input-Multiple-Output (MIMO) H_∞ optimal control design technique and its extension to a decoupling design technique presented in this section are based on the optimisation of H_∞ Frobenius norm using the analytical optimisation method.

The main optimisation problem considered in the multivariable case is[40]: Given a performance function $\Gamma(\omega, f_1(j\omega), \ldots, f_N(j\omega))$, where N is the number of the variables, find $\gamma^* \geq 0$ and $f_1^*(j\omega), \ldots, f_N^*(j\omega) \in A_{RHP}^N$ such that

$$\gamma^* = \inf_{f \in A_{RHP}} \sup_{\omega} \Gamma(\omega, f_1(j\omega), \ldots, f_N(j\omega))) = \sup_{\omega} \Gamma(\omega, f_1^*(j\omega), \ldots, f_N^*(j\omega)) \quad (6.33)$$

where A_{RHP}^N are all analytic function vectors analytic in right half plane. γ^* is the optimal value or minimizer, the $f_1^*(j\omega), \ldots, f_N^*(j\omega)$ are the optimal solutions. This problem can also be solved by the analytical optimisation method using 'disk-iteration' algorithm as introduced in previous chapter.

6.3.1 H_∞ Frobenius Norm

The *Frobenius* norm is the norm of a matrix defined as the square root of the summation of the absolute squares of its elements [31]. If $A \in R^{m\times n}$,then:

$$\|A\|_F = \sqrt{\sum_i^m \sum_j^n |a_{ij}|} \tag{6.34}$$

The Frobenius norm also equals to the square root of the square matrix trace:

$$\|A\|_F = \sqrt{Tr(AA^H)} \tag{6.35}$$

whereA^H is the conjugate transpose of A. Assume that there is a singular value decomposition(SVD) of A,$A = USV^T$. If we substitute the singular value decomposition(SVD) of A into the above definition, we have:

$$\|A\|_F = \sqrt{Tr[(USV^T)(VS^TU^T)]} = \sqrt{Tr(SS^T)} \tag{6.36}$$

Let the singular values of matrix S be $\sigma_1\sigma_2...\sigma_n$, where n is rank of A, equation(6.36) can be rewritten as:

$$\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + ... + \sigma_n^2} \tag{6.37}$$

Comparing the Frobenius norm with the Maximum singular value σ_{max}, it is thus easy to show:

$$\sigma_{max} \leq \|A\|_F \leq \sqrt{n}\sigma_{max} \tag{6.38}$$

where $n = rank(A)$. Because the Frobenius norm is greater than the maximum singular value of matrix A, then the maximum singular value can be bounded by the *Frobenius* norm. For the full frequency range, the peak magnitude or maximum singular value of the frequency dependent transfer function matrix can be constrained by bounding the maximum Frobenius norm over all frequencies, which is defined as the H_∞ Frobenius norm of the transfer function matrix. Although the H_∞ Frobenius norm is not the most desirable means of representing the uncertainty, the maximum singular value is not either. Actually the structured singular value (μ) proposed by Doyle is a more desirable and flexible method since it allows modelling of structured uncertainty. However the H_∞ Frobenius norm method can be used to derive a practical MIMO controller design algorithms in the same way as singular value H_∞ methods.

6.3.2 Multivariable Mixed Sensitivity Problem

The controller design method proposed in this section is based on solving the mixed sensitivity problem. The normal closed loop performance and robust stability design requirements are $||W_S S||_\infty \leq 1$ and $||W_T T||_\infty \leq 1$ respectively in the maximum singular

value H_∞ norm. These requirements are frequently conservatively met by the weighted mixed sensitivity optimality criterion coupled with a requirement for internal stability,

$$\left|\left|\begin{bmatrix} W_S S \\ W_T T \end{bmatrix}\right|\right|_\infty < 1 \Rightarrow |||W_S S|^2 + |W_T T|^2||_\infty < \frac{1}{2} \tag{6.39}$$

where W_S and W_T are positive-matrix frequency dependent weighting functions and $||.||_\infty$ indicates maximum singular value $\overline{\sigma}(\cdot)$ over all frequency. Of course, additional constraints on the other closed loop transmissions (sensitivity functions) can be added to achieve other requirements. For example $||W_{KS} KS||_\infty \leq 1$ may be used to limit control effort. A difficulty with using such an approach is the choice of weighting matrix, because the significance of off diagonal terms are difficult to judge. The appropriate choice of weighting matrices is not always clear. As a results, designers often limit themselves to diagonal weighting matrices.

In this section we examine the methods to allow the use of closed loop nominal performance and robust stability design requirements specified using the Frobenius norm, $||.||_F$ by means of the weighted mixed sensitivity optimality criterion coupled with Hadamard (element-by-element) multiplication of the weighting functions and a requirement for internal stability by means of the criterion

$$\left|\left|\begin{bmatrix} W_S * S \\ W_T * T \end{bmatrix}\right|\right|_\infty < 1 \Rightarrow ||W_T * S||_F^2 + ||W_T * T||_F^2 < \frac{1}{2} \tag{6.40}$$

where now W_S and W_T are matrices whose elements are positive frequency dependent weighting functions and $*$ indicates Hadamard (element-by-element) multiplication. This approach is particularly attractive because it localises the designer's choice of weighting function to each element. If the requirement is for a decoupling controller, the closed loop transfer matrix T should be a diagonal matrix. However, it is not possible to replace the off-diagonal element of T by 0 directly and then to compute the controller and retain robustness, since this will change the bounded induced H_∞ norm of T. This approach would result in a non-optimal solution which could even cause the sensitivity of S to become unbounded and the closed loop system to be unstable. To achieve diagonal for a bounded T, the approach using Hadamard multiplication can be readily used. In the design procedure, the off-diagonal elements of T can be constrained to be 0 or at least a very small number near 0 while the diagonal elements are bounded by the usual frequency dependent weighting functions through the ability to directly multiply different scalar weighting function by each element such that the resulting T is a diagonal or a near diagonal matrix.

6.3.3 Parametric Design Approach

Mixed Sensitivity Optimisation with Internal Stability Interpolation

A 2×2 MIMO system is illustrated here to describe the H_∞ Frobenius norm optimisation of the MIMO system. The extension to higher orders (including non-square) is obvious. We wish to determine a K so that the system of figure 6.1 is stable and

$$\begin{aligned}\left\|\begin{bmatrix} W_S * S \\ W_T * T \end{bmatrix}\right\|_F &= ||W_s * [I - T]||_F^2 + ||W_t * T||_F^2 \\ &= \inf_{T_{ij} \in \mathcal{H}_\infty^{1\times 1}} \sup_\omega \{|W_{T11}T_{11}|^2 + |W_{S11}[1 - T_{11}]|^2 + \\ &\qquad |W_{T12}T_{12}|^2 + |W_{S12}[T_{12}]|^2 + \\ &\qquad |W_{T21}T_{21}|^2 + |W_{S21}[T_{21}]|^2 + \\ &\qquad |W_{T22}T_{22}|^2 + |W_{S22}[1 - T_{22}]|^2\} \leq 1 \end{aligned} \tag{6.41}$$

where $i, j = 1, 2$. To determine the feasibility and obtain a potential solution for T the following performance functions should be optimised:

$$\begin{aligned}\Gamma = &|W_{T11}T_{11}|^2 + |W_{S11}[1 - T_{11}]|^2 + |W_{T12}T_{12}|^2 + |W_{S12}[T_{12}]|^2 + \\ &|W_{T21}T_{21}|^2 + |W_{S21}[T_{21}]|^2 + |W_{T22}T_{22}|^2 + |W_{S22}[1 - T_{22}]|^2 \end{aligned} \tag{6.42}$$

or

$$\begin{aligned}\Gamma = &|W_{T11}[1 - S_{11}]|^2 + |W_{S11}S_{11}|^2 + |W_{T12}[0 - S_{12}]|^2 + |W_{S12}S_{12}|^2 + \\ &|W_{T21}[0 - S_{21}]|^2 + |W_{S21}S_{21}|^2 + |W_{T22}[1 - S_{22}]|^2 + |W_{S22}S_{22}|^2 \end{aligned} \tag{6.43}$$

where T or S should be parameterised with the interpolation function. For the case of a plant having RHP zeros, each elements of T are given by:

$$\begin{aligned} T_{11} &= \alpha T_{011} \\ T_{12} &= \alpha T_{012} \\ T_{21} &= \alpha T_{021} \\ T_{22} &= \alpha T_{022} \end{aligned} \tag{6.44}$$

where the $\alpha \in H_\infty^{1\times 1}$ is the interpolation function contains exactly the RHP zeros of plant (and no other RHP zeros). For the case of plant having RHP poles, each elements of S are given by:

$$\begin{aligned} S_{11} &= \alpha S_{011} \\ S_{12} &= \alpha S_{012} \\ S_{21} &= \alpha S_{021} \\ S_{22} &= \alpha S_{022} \end{aligned} \tag{6.45}$$

where the interpolation function the α contains exactly the RHP poles of plant (and no other RHP poles). For the case of the plant having RHP zeros and poles, each elements of T are given by:

$$\begin{aligned} T_{11} &= A^* + \alpha T_{011} \\ T_{12} &= 0 + \alpha T_{012} \\ T_{21} &= 0 + \alpha T_{021} \\ T_{22} &= A^* + \alpha T_{022} \end{aligned} \tag{6.46}$$

Where the parameterisations of $A^* \in H_\infty^{1\times 1}$ and $\alpha \in H_\infty^{1\times 1}$ in equations 6.44, 6.45 and 6.46 were presented in section 6.2. And the parameterisations sufficient to ensure internal stability of section 6.2 implies.

The performance function Γ does not obviously presents a quasi-circle problem. In the disk iteration algorithm the second Taylor approximation of Γ is taken to complete the perfect square form in Γ when the performance function is established. The function Γ is then a quasi-circle type function in the scalar optimisation case; in the multivariable optimisation case, the function Γ is the summation of the Taylor approximation for each variable. And the Γ for multivariable case is also the summation of several quasi-circle type functions, the 'dual extremal' algorithm can still solve that multivariable Nehari problem uniformly. Such that descent direction for each variable can be obtained. The optimal solutions can thus be found by updating the current solutions with the descent directions and testing with the optimality conditions iteratively.

Design Procedure

The detailed design procedure for parametric design is presented in the following.

1. Determine the RHP zeros and poles from the plant G.
2. Generate the interpolation function α and matrix A based on RHP zeros and poles.
3. Select the desired subject sensitivity function for design. For case of plant having RHP zeros only, this is T; for case of plant having RHP poles only, this is S; for case of plant having RHP zeros and poles, this is T.
4. Determine element-by-element weighting function W_S and W_T based on performance and robustness requirements.
5. Establish the performance function Γ of the mixed sensitivity problem with the interpolation function (matrix), expressing T as $T = \alpha T_0$ or $T = A + \alpha T_0$; or expressing S as $S = \alpha S_0$

6. Select the desired frequency range and generate frequency vector.

7. Compute frequency response for performance function.

8. Smooth the weightings.

9. Solve minimax problem for performance function Γ by analytical optimisation

10. Obtain the minimised γ and corresponding optimal solutions. If the γ does not converge well, change the weighting functions matrix or change the smoothing parameter slightly or change the number of frequency points.

11. Obtain the transfer function matrix for T_0 or S_0 by fitting to the optimal solutions, then obtained the optimised sensitivity function T or S.

12. Calculate the controller K from plant G and optimal sensitivity function matrix.

Although the mixed sensitivity problem involving S and T is solved here, other closed loop transmissions (sensitivity functions), such as sensitivity KS, can be added in the performance function to achieve other requirements.

6.3.4 Nonparametric Design Approach

The method presented in the last section is a parametric design method, since it requires explicit determination of the RHP poles and zeros of the plant which implies the use of a parametric plant model. The nonparametric design method is also possible for the multivariable case. As in the SISO case, the key issue in the nonparametric design approach is the guarantee of the internal stability. Unlike the parametric plant, the RHP zeros and poles can not be found explicitly for the nonparametric plant. Two alternative methods for the nonparametric design are presented in the following.

Mixed Sensitivity Optimisation with Interpolation Function

To satisfies the sufficient internal stability constraint presented previously, the scalar interpolation function α should be stable and contain the RHP transmission zeros of G. In general the zeros of transfer function (matrix) for the square system are the zeros of the determinant of the matrix[80]. In nonparametric design, the scalar interpolation function $\alpha(j\omega)$ can be obtained from the determinant of plant $G(j\omega)$ when the plant is stable. In practice when using $\alpha(j\omega)$ obtained from $det(G(j\omega))$ the frequency response may be divided by a stable minimum phase compensating filter $\overline{\alpha}(j\omega)$ where $\overline{\alpha}(j\omega) = |\alpha(j\omega)|$ obtain the all-pass condition to control the relative degree and obtain convergence. Such a filter may be obtained by a Hilbert transform approach.

A small modification with an additional factor $\frac{1}{(s+a)^r}$, $a > 0$, and integer $r \geq 0$ controls the relative degree of T.

Design Procedure

For the common case of stable plant with RHP zeros, the above nonparametric method gives a design procedure which is similar to the parametric design. Expecting the interpolation function should be obtained by calculating the determinant of matrix. The detailed design procedure is presented in the following:

1. Calculate the determinant of plant $det(G(j\omega))$ in the form of the frequency response function.
2. Obtain the nonparametric interpolation function $\alpha(j\omega)$ from the $det(G(j\omega))$.
3. Determine weighting function W_S and W_T based on performance and robustness requirements.
4. Establish the performance function Γ of mixed sensitivity problem using with the interpolation function (matrix), expressing $T(j\omega)$ as $T(j\omega) = \alpha(j\omega)T_0(j\omega)$.
5. Smooth the weightings.
6. Solve minimax problem for performance function Γ by analytical optimisation.
7. Obtain the minimised γ and corresponding optimal solutions. If the γ does not converge well, change the weighting functions matrix or change the smoothing parameter slightly or change the number of frequency points.
8. Obtain the transfer function matrix for T by fitting to the optimal solutions of $\alpha(j\omega)T_0(j\omega)$.
9. Calculate the controller K from plant G and optimal sensitivity T.

Mixed Sensitivity Optimisation for Sensitivity KS

As in the SISO case, the nonparametric optimisation may be based on optimisation with the control effort sensitivity $U = KS$ as the optimisation variable in MIMO case, when plant G is stable. The guarantee of internal stability has been presented in the section 6.2.4. Here, the optimisation of sensitivity U is emphasised only. Because the sensitivity $T = GU$, the optimisation of mixed sensitivity problem can be expressed in terms of the sensitivity U. A 2×2 MIMO system is used to illustrate the idea.

Assuming that the H_∞ Frobenius norm mixed sensitivity requirement is

$$\begin{aligned}\left|\left|\begin{bmatrix} W_S * S \\ W_T * T \end{bmatrix}\right|\right|_F &= ||W_S * [I - T]||_F^2 + ||W_T * T||_F^2 \\ &= ||W_S * [I - GU]||_F^2 + ||W_T * GU||_F^2 \\ &= \inf_{T_{ij} \in \mathcal{H}_\infty^{1\times 1}} \sup_{\omega} \{||W_S * [I - GU]||_F^2 + ||W_T * GU||_F^2\} \leq 1 \quad (6.47)\end{aligned}$$

To determine the feasibility and obtain a potential solution for U the following performance function Γ to be optimised can be expressed as:

$$\begin{aligned}\Gamma = &|W_{T11}(U_{11}G_{11} + U_{21}G_{12})|^2 + |W_{S11}[1 - (U_{11}G_{11} + U_{21}G_{12})]|^2 + \\ &|W_{T12}(U_{12}G_{11} + U_{22}G_{12})|^2 + |W_{S12}[0 - (U_{12}G_{11} + U_{22}G_{12})]|^2 + \\ &|W_{T21}(U_{11}G_{21} + U_{21}G_{22})|^2 + |W_{S21}[0 - (U_{11}G_{21} + U_{21}G_{22})]|^2 + \\ &|W_{T22}(U_{12}G_{21} + U_{22}G_{22})|^2 + |W_{S22}[1 - (U_{12}G_{21} + U_{22}G_{22})]|^2 \quad (6.48)\end{aligned}$$

Design Procedure

The detailed design procedure for optimisation of a mixed sensitivity performance function Γ for sensitivity $U = KS$ is presented in the following:

1. Determine weighting function W_S and W_T based on performance and robustness requirements.
2. Establish the performance function Γ of mixed sensitivity problem in terms of sensitivity $U(j\omega)$ and nonparametric plant $G(j\omega)$.
3. Solve minimax problem for performance function Γ by analytical optimisation.
4. Obtain the minimised γ and corresponding optimal solutions. If the γ does not converge well, change the weighting functions matrix or change the smoothing parameter slightly or change the number of frequency points.
5. Obtain the transfer function matrix for the sensitivity U by fitting to the optimal solutions.
6. Calculate the controller K from plant G and optimal sensitivity function matrix U.

6.4 Design Example for Torque-λ Control Problem

In this section, the decoupling controller design approach based on H_∞ Frobenius norm feedback control technique is experimentally applied to the Air-Fuel Ratio (AFR,

termed as λ) and Torque MIMO control of a spark ignition (SI) internal combustion (IC) engine. This is a typical MIMO control problem. In this problem, the inputs are the duty cycle of Air Bleed Valve (ABV) in % units and Fuel Pulse Width (FPW) in μs units of signal of fuel injector. The whole system is a 2×2 square system.

To apply the decoupling controller design approach to that system, a linear model is required to be obtained. When the engine speed is fixed, the system from ABV/FPW to AFR/Torque is quite linear. The linear model can be obtained at a constant engine speed. Due to the very significant delay between the FPW and measured AFR signal, the linear model, whether as ARX or ARMAX model, is required to be a high order model. A linear MIMO robust controller based on such model is generally also needed to be of a high order controller. Even with the best controller designs the resulting response of AFR signal is very slow. A directly applied linear MIMO controller is not practicable. To overcome this issue, an estimator with the time delay removed in the AFR signal is employed to provide a linear characteristic with reduced time delay. Benefiting from the estimator with time delay removed, the linear robust controller for the engine supplemented with the estimator becomes a form of a predictive controller such that it can regulate the AFR signal very effectively.

In this section, the entire procedure from system identification to controller design and experimental validation for AFR/Torque control system is presented. The simulation and experimental results of the decoupling controller are compared with those of a conventional 'induced H_∞ norm mixed sensitivity controller' developed by the 2-Riccati equation algorithm. For conciseness this method and controller will be referred to as the 'Riccati method' and 'Riccati controller' in the following.

6.4.1 Modelling for λ and Torque Production

This section describes how the nonlinear AFR estimator is incorporated with the 2×2 MIMO engine model. To structure the estimator of the AFR, we require to identify the AFR path as a multiple input single-output (MISO) plant with the time delay removed. Our approach is based on nonlinear black-box modelling. An algebraic Nonlinear Auto Regressive eXogenous (NARX) model structure is used for the AFR estimator. From previous work [20],[72], it has been found that the FPW, integrated-MAP and engine speed (RPM) signals have high correlation with AFR signal. So these signal were used as inputs using the structure shown in figure 6.2. The estimator is then incorporated in a 2×2 negative feedback control system as shown in figure 6.3. The 2×2 MIMO decoupling feedback controller is used to feedback on estimated AFR, however a slower timescale PID controller has been found useful to trim slow drift in the estimator and ensure zero steady-state AFR error. Because of the slow time-scale nature of any drift,

the parameters of the PID compensator are not critical to the system performance and are easily found by on-line tuning.

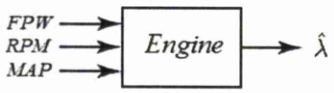

Figure 6.2: Structure of AFR estimator

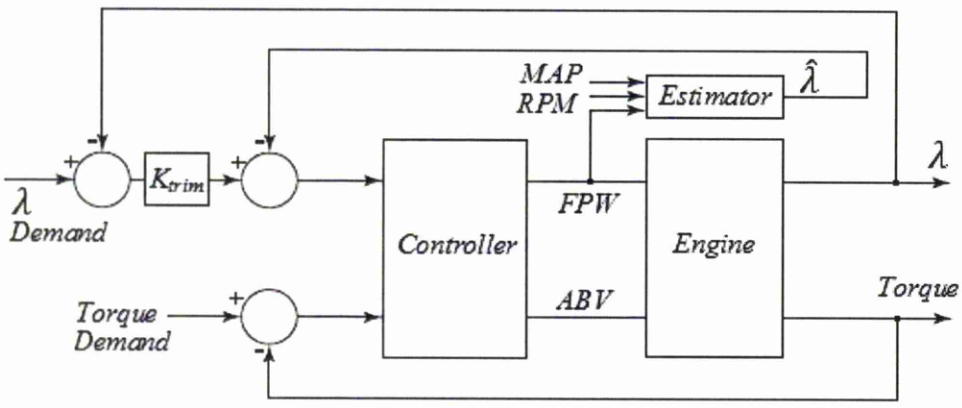

Figure 6.3: Feedback control system diagram

6.4.2 NARX Identification of Nonlinear λ Estimator Using Black-box Method

Experimental Set-up

The experimental work described here was carried out on the low inertia electric engine dynamometer at the University of Liverpool. The experimental engine is a 4-cylinder Ford Zetec 1.6 L 16 valve engine with sequential port fuel injection. For the purposes of the study, the ABV on the engine is treated as an effective electronic throttle. Once ABV is fully opened, the engine speed attains over than $2500RPM$. For the identification work, the dynamometer is run under speed-control mode, so that the speed of dynamometer is controlled by the researcher. As the dynamometer's shaft and the engine's shaft are coupled rigidly, the engine torque production can be measured by an in-line Vibrometer torque transducer. The engine torque production is given by:

$$T_E = T_L + I\dot{\omega} \tag{6.49}$$

where T_E is the engine torque, T_L is the brake load torque, I is the moment of inertia of the rotating shaft and ω is the angular velocity. The system is such that the en-

gine torque production is controlled by the demand of the load (brake torque) of the dynamometer. The maximum load which can be loaded by the dynamometer is approximately 35 to $40N \cdot M$ at 1500rpm. The MATLAB real-time workshop (RTW) was used to interface the dynamometer using a dSpace-Autobox to enable a SIMULINK representation of the control system and estimation system to be applied in real-time.

Black-box NARX Identification of $\hat{\lambda}$ Estimator

To identify the black-box model, the input/output dynamics need to be collected from the engine test. Random binary sequences are used for linear systems identification. For the nonlinear system identification, to test for the amplitude dependencies input signals require more than two levels. Accordingly, scaled uniform random numbers with a DC bias were selected as inputs.

The AFR estimator is a 3×1 nonlinear multivariable system. For identification uncorrelated test signals are necessary for each input [25]. The choice of suitable test signals, requires consideration both in the period and amplitude. An appropriate period of test signal should excite the dynamics of across the full frequency range of the system. From pre-testing, it was found that a 1.2s period is suitable for ABV, 1.5s is suitable for FPW and 5s is suitable for the engine speed. The amplitude of the signal was determined to ensure that the system was sufficiently excited at all speeds and torques but also to avoid engine stall. Accordingly the ABV duty was varied within [40%, 60%], FPW within [2200, 6500] and engine speed within [1000RPM, 2000RPM]. Figure 6.4 shows the typical nonlinear identification test signals applied.

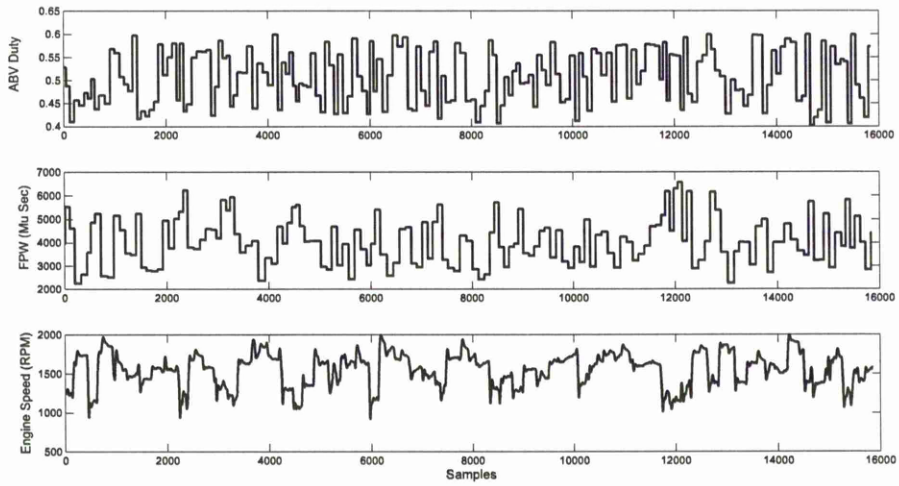

Figure 6.4: Nonlinear identification test signals

In this work, a NARX model is selected to identify the nonlinear AFR estimator. The NARX model is a polynomial function of the current output y_t, which is based on previous and current inputs $u(t), ..., u(t-n)$ and outputs $y(t-1), ..., y(t-m)$. The polynomial include the different linear or nonlinear terms (regressors). The nonlinear terms were allowed to be square or cubic or other power terms, cross-products terms, reciprocal terms and combinations thereof. Due to the different combination of inputs and outputs, many nonlinear terms can be generated to establish the NARX model. To generate a simplified model structure whose output fits the measured output well, the most significant regressors of the candidates were selected to establish the NARX model. Many structure selection methods are available for determining the choice of the regressors, such as the methods based on step and impulse tests, frequency response measurements, correlation analysis and time response data, etc. The survey paper by [36] summarises many of them. The selection method for the regressors used in this work is based on correlation analysis between the current output y_t and the candidate regressors. The method is described in [86].

To obtain the coefficient of each term (regressors) of the NARX model the Ordinary Least Square (OLS) algorithm is used. The coefficients of the regressors matrix or vectors could be solved by minimizing the error between the estimated output and the measured output using the classical least squares solution methods. Neither SVD nor QR algorithms were required for the OLS solution in this work.

In the NARX identification of the AFR estimator the data was collected from the dSpace-Autobox. Initially, the sampling program was triggered by every 4^o of crank angle. But in data processing, the data was down-sampled to every 180^o of crank angle, so that the very high frequency noise could be removed. Half of the dataset is used for identification, and the rest was used for validation. The time delay was removed between candidate inputs and outputs for the resulting estimator model. So that input and output behaviour remains correctly synchronised. To find the time delays between the FPW, MAP, engine speed and AFR signals, a simple step test was used. By fixing other input signals and applying a step demand for FPW, the corresponding response of MAP and AFR signals could be found and reordered easily. The time delays between them can be found more accurately through comparing the step demand of FPW and the response of MAP and AFR. Because the engine speed was controlled by the dynamometer, engine speed signal was not correlated to the FPW signal. But the equivalent time delays could be found from the response of MAP and AFR signal with a step in engine speed. From step tests, it was found that the delay between the FPW signal and AFR signal is approximately 20 samples and the delay between the MAP and FPW signal is approximately 18 sample. The equivalent delays between engine speed and FPW signal is approximately 18 samples. To remove the

delays from the dataset prior to identification and retain synchronisation between input and output signals, the AFR signal was shifted forward 18 samples. Figure 6.5 shows the procedure of data shifting.

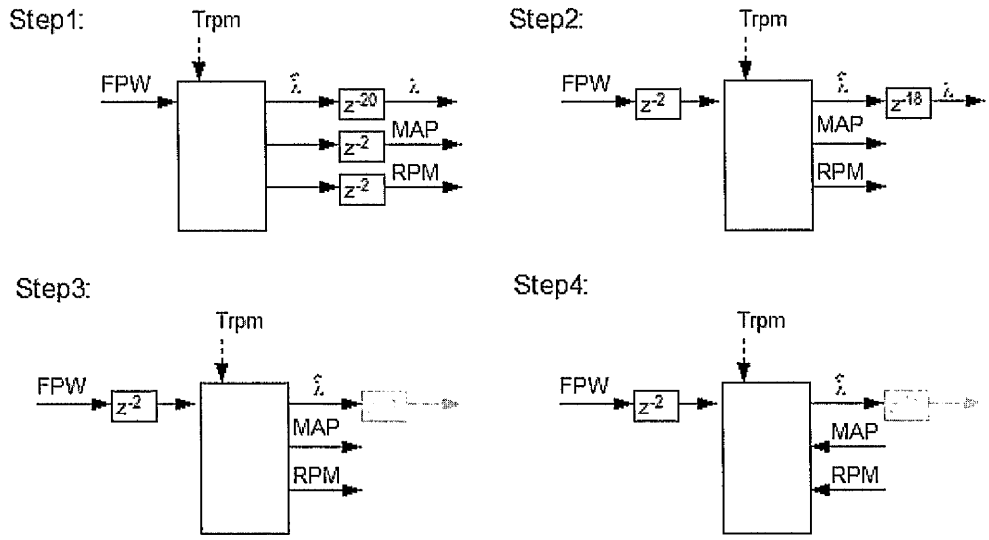

Figure 6.5: Data shifting for nonlinear identification

Once the delays were shifted in the identification datasets, the NARX model could then be calculated. As we are mostly interested in the degree of relationship between the candidate regressors and the current output dataset, the candidate regressors were sorted according to the correlation coefficients. Then the regressors were selected in order of correlation until the required model order was achieved. If a term correlated very highly with a regressor already selected for the model, this regressor was removed from the model. Table 6.1 lists the regressors which are chosen in the identification. The coefficients are calculated by the OLS algorithm, which are also listed in Table.6.1.

Table 6.1: AFR estimator regressor terms and associated coefficients

Regressors		Parameter coefficient
Unity		0.48186
$FPW^{-1}(t-2)MAP^{2}(t)$		-3.2756e-4
$FPW(t-2)MAP^{-1}(t-1)$		6.3190e-4
$FPW^{-1}(t-15)MAP(t-1)$		-3.6015e-4
$RPM(t-4)FPW(t-8)MAP^{-1}(t-1)$		3.3519e-05
$RPM(t-4)MAP^{2}(t)$		2.9634e-07
$RPM^{-1}(t-1)MAP^{-1}(t)FPW^{2}(t-6)$		-13.928
$RPM^{-1}(t-4)MAP^{-1}(t)$		-75.531

Once the NARX model was obtained, the estimated output datasets and measured output datasets were compared to validate the accuracy of the NARX model. Figure 6.6 shows the fit of the nonlinear model from off-line validation. From the data, the R^2 is about 0.9040, indicating that the model recovered 90% of the system dynamics. The NARX model was then validated further by comparing with the actual experimental data on-line. Figure 6.7 shows a comparison of the estimated AFR value of the NARX estimator and the measured AFR value. It is evident that the estimated AFR value is always significantly in advance of the measured AFR value as required.

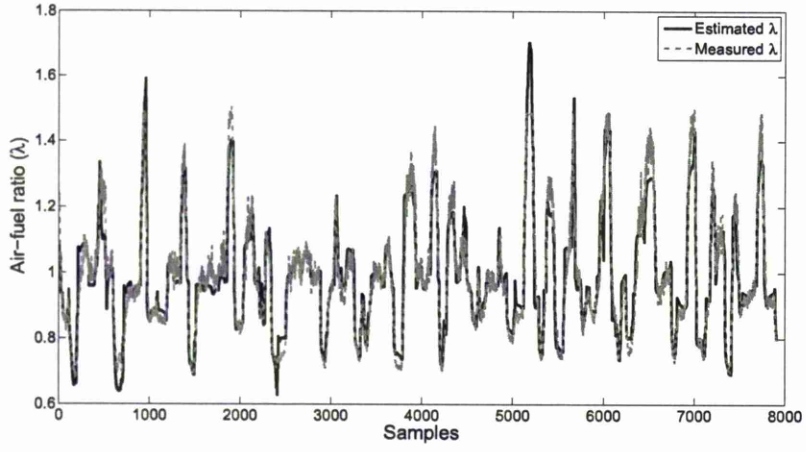

Figure 6.6: Off-line validation of the estimated AFR fit to measured AFR

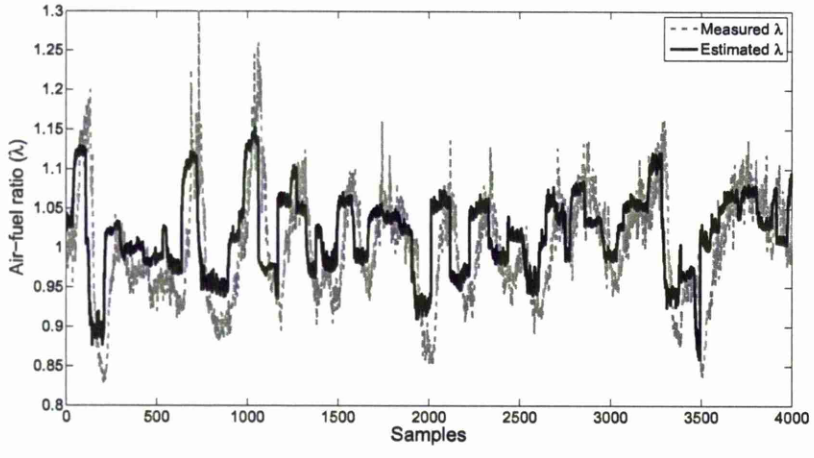

Figure 6.7: On-line validation of estimated AFR value and actual measured AFR

6.4.3 Linear Identification for MIMO Model

The linear robust control design is based on a linear model identified around the engine-estimator system. The model is a 2×2 MIMO model with ABV duty cycle and FPW as inputs and the $\hat{\lambda}$ AFR estimate and engine Torque production as output. As in a similar approach taken for SISO torque control [94], several models of different engine speed were identified. For the purposes of obtaining the identification data the engine speed is regulated at 11 different speed setpoint values with the dynamometer controller set in speed-control mode in increments of 100rpm, from 1000rpm to 2000rpm. The process of developing a nominal mode of those linear models captures the plant nonlinearities remaining, which can be considered as plant uncertainty around a nominal model. For all tests, the dynamometer controller regulated the engine speed within ±50rpm of the nominal.

Pseudo-random binary sequence (PRBS) signals were used to excite a wide range of frequencies. The magnitude of the PRBS ABV duty cycle signals was 0.02(%), which was added to and subtracted from the nominal ABV duty, which depending upon the engine speed was found to vary from 0.4 to 0.49 (40% to 49%). The period of the signal was designed to maximise fit quality and retain high frequency information. From testing, the resulting optimal ABV signal was found to be a PRBS of period 1.0s and for FPW signal a PRBS of period 1.2s. An ARX identification algorithm [57] was used to obtain each of the set-speed models. A 2×2 MIMO model was found by identifying two MISO models, each MISO model having common denominators in its elements.

For the AFR path MISO models, the best trade off between model order and fit quality was found with second order linear transfer functions with about 2 delays . All 11 of the AFR path transfer functions were found to have a fit rate of about 75%. For the Torque production path MISO model, second order linear transfer functions between 2 and 4 delays were found. The fit rate for all the Torque path models was over 60%. The array Bode plots for the eleven 2×2 models are shown in figure 6.8. 6.8 As mentioned the nonlinear system can be described as a family of linear systems centred on a nominal model while the nonlinearities is considered as its uncertainty[94]. The method used for getting a controller which can handle the nonlinearities of the operating frequency range is to obtain a nominal model from different frequency responses. Following Skogestad and Postlethwaite [80], the nominal model is obtained by discretising the frequencies and by finding the minimal radius disk enclosing all responses at each frequency in the complex plane. The centre of each disk represents the nominal model at the corresponding disk frequency, and the radius of the disk is the minimum unstructured uncertainty. The transfer function of the nominal model was obtained

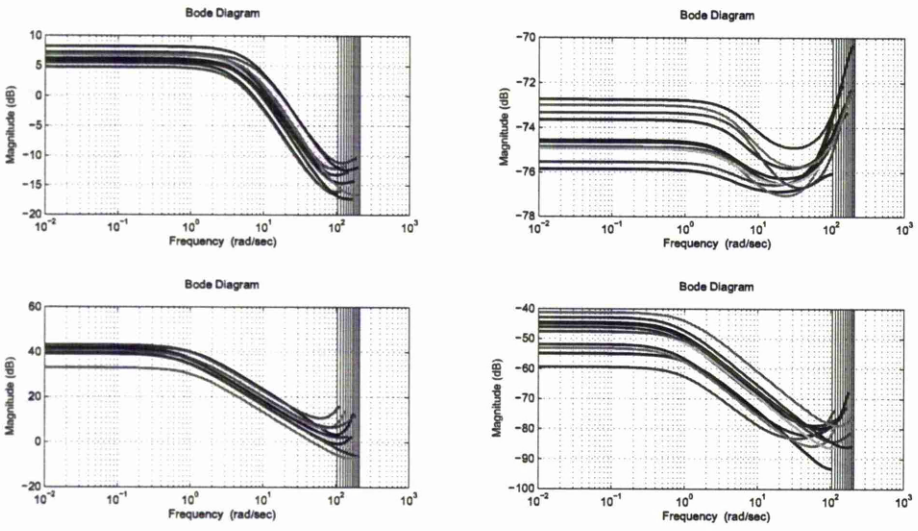

Figure 6.8: Linear ARX models of 2×2 MIMO system

by fitting the discretised frequency response using Levi, Dennis and Schnabel's fitting method [56], [19]. The resulting plant transfer functions expressed as continuous domain models from the two control inputs $[ABV, FPW]^T$ to the two outputs $[\hat{\lambda}, T_E]$ were found to be:

$$G11 = \frac{-1.615s^3 + 490.1s^2 - 4.957e4s + 1.671e6}{s^3 + 259.8s^2 + 1.614e4s + 1.644e4} \tag{6.50}$$

$$G12 = \frac{-6.94e - 5s^3 + 0.01771s^2 - 1.461s + 38.34}{s^3 + 259.8s^2 + 1.614e4s + 1.644e4} \tag{6.51}$$

$$G21 = \frac{-0.2377s^2 - 5s + 1877}{s^2 + 165.6s + 908.9} \tag{6.52}$$

$$G22 = \frac{-0.0002142s^2 - 0.02296s - 0.1537}{s^2 + 165.6s + 908.9} \tag{6.53}$$

6.4.4 Feedback Controller Design

To design the decoupling feedback controller, the multivariable H_∞ Frobenius norm mixed sensitivity optimisation control design technique based on the analytic function optimisation methods developed in this chapter was employed to satisfy the mixed sensitivity requirement:

$$\Gamma = \left|\left| \begin{bmatrix} W_S * S \\ W_T * T \end{bmatrix} \right|\right|_F^2 < 1 \Rightarrow ||W_S * [I - T]||_F^2 + ||W_T * T||_F^2 < 1 \tag{6.54}$$

where $*$ represents element-by-element multiplication (Schur or Hadamaard product) and W_S and W_T are the element-by-element weighting functions for S and T respectively. To enable a fair comparison for the decoupling controller, a H_∞ mixed sensitivity

controller was developed using the H_∞ 2-Riccati equation algorithm from Chiang and Safonov [9]. It solves the standard mixed sensitivity problem over the maximum singular value of transfer function matrix.

Based on the Nyquist-frequency associated with the data sampling, the frequency range from 0.01 to 200rad/s. For the decoupling controller, the frequency grid points for the optimisation, are generated by the following linear fractional transformation equation [41]:

$$\omega = b/tan(\frac{n\pi}{N_\omega}) \tag{6.55}$$

where, $n = 0, 1, ..., N_w - 1$ and N_w is the number of the frequency points. The b is the sample width parameter. $N_\omega = 512$ frequency points were then with $b = 1$ were chosen to cover the main frequency range of the engine.

To establish the performance function, the complementary sensitivity function T was selected and interpolated by the method presented in section 6.2. The sensitivity T was interpolated with scalar function α:

$$T = \alpha T_0 \tag{6.56}$$

where $\alpha \in H_\infty^{1\times1}$ has precisely the same unstable zeros as the unstable transmission zeros of the plant, to guarantee internal stability. The main optimisation problem is to minimise γ for the performance function:

$$\gamma = \inf_{T\in H_\infty^{2\times2}} \sup_\omega ||W_S(j\omega) * [I - \alpha(j\omega)T_0(j\omega)]||_F^2 + ||W_T(j\omega) * \alpha(j\omega)T_0(j\omega)||_F^2 \tag{6.57}$$

where the interpolation function α was parametrised to contain the RHP zero of the plant. From the transfer function of equation 6.50-equation 6.53, the plant of the engine is seen to have three simple zeros in the RHP, no poles in the RHP. The zeros are at $s = 103.977$,$s = 100.00$ and $s = 100.00$. Thus:

$$\alpha = \frac{(s - 103.977)(s - 100)(s - 100)}{(s + 103.977)(s + 100)(s + 100)} \tag{6.58}$$

where the poles are selected to ensure α is stable and achieve an all-pass, 0dB/1 magnitude characteristic to aid solution convergence and control the solution relative degree. Another effect of those poles is to ensure T_0 has the same frequency response magnitude as T, which allows more possible solutions for T_0 to be 'spanned' in the region bounded by weighting function during optimisation. For $T = \alpha \cdot T_0$ clearly, $T(103.977) = T(100) = T(100) = 0$ which satisfies the interpolation constraint 6.17. The system is thus constrained to be internally stable.

To achieve the performance targets of air-fuel ratio and torque control problem, the weighting functions W_S and W_T must be chosen carefully. Commonly, the primary sensitivity is used in the design to achieve the regulation performance targets and the

complementary sensitivity function is used to ensure the robustness. For decoupling the closed loop transfer matrix function should also be an invertible diagonal rational matrix function and the off-diagonal elements of the closed-loop transfer matrix T should be close to zero [65]. In this case, we constrain the off-diagonal element of T to be less than 10^{-5}. A selection of weighting functions based on these performance and robustness requirements such as minimal speed deviation, fast settling time, small overshoot, zero steady-state error, sensor noise, random parameter variation and engine-to-engine variation, etc. resulted in the Hadamard weighting functions:

$$W_S = \begin{bmatrix} \frac{s+1}{2s+0.001} & 10^5 \\ 10^5 & \frac{s+1}{5s+0.001} \end{bmatrix} \text{ and } W_T = \begin{bmatrix} \frac{10s+1}{100} & 10^5 \\ 10^5 & \frac{10s+1}{100} \end{bmatrix}$$

In the design using the Riccati method, corresponding weighting functions chosen as diagonal matrices were taken as:

$$W_S = \begin{bmatrix} \frac{s+1}{2s+0.001} & 0 \\ 0 & \frac{s+1}{5s+0.001} \end{bmatrix} \text{ and } W_T = \begin{bmatrix} \frac{10s+1}{100} & 0 \\ 0 & \frac{10s+1}{100} \end{bmatrix}$$

Once the optimal solutions (frequency responses) were found, the stable transfer function (matrix) of T was obtained by fitting the optimal solutions using *Caratheodory-Fejer* approximation method[33]. The controller K could be calculated from T with plant G easily. An $18th$ order transfer matrix was thus obtained. The following figure shows the comparison of the results of the Riccati method and decoupling method in Bode magnitude plots for sensitivity S and T. From these figures, the diagonal elements are seen to be similar for the Riccati method and analytical optimisation method both in T and S. But the off-diagonal elements of the decoupling method are smaller than those of the Riccati method. In other words, the closed-loop system designed by decoupling method is close to an ideal decoupling system.

6.5 Simulation and Experimental Results

6.5.1 Simulation Results

The step responses of the closed loop system was simulated first. A positive step of $10N \cdot m$ for torque was applied at 1s. Another positive step of 0.1 for AFR was applied at 5s. From figure 6.11, it can be seen that the performance for the Riccati and the optimal decoupling controller are similar. On response to torque disturbance, the decoupling optimal controller is initially slightly faster than the Riccati controller but has effectively the same settling time. The AFR signal takes about 3s to return from a step disturbance and the torque takes about 4s.

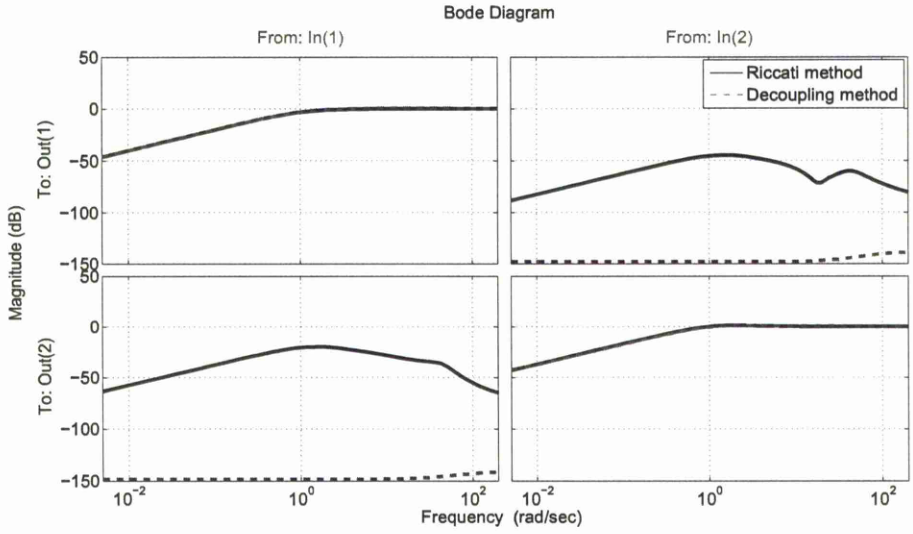

Figure 6.9: Bode magnitude plots for 'S' transfer matrix

A considerable cross-coupling between the torque and AFR channels of the Riccati controller is evident in figure 6.11. Here it is clear that despite the demand for AFR being set at zero for the first 5s, the change in demand for the torque after 1s causes a significant negative deviation in AFR. The decoupling optimal controller on the other hand is seen to effectively achieve exact decoupling with no significant interaction between the two signals.

Results of simulation of the H_∞ Frobenius decoupling design and Riccati designs using the linear models showed that similar disturbance response, tracking performance and robustness was achieved with the two controllers, but that the H_∞ Frobenius decoupling controller achieved effectively complete decoupling. Experimental studies were then conducted to see how effective the controllers with estimator performed experimentally on the dynamometer based engine.

6.5.2 Experimental Results

Figure 6.12 and 6.13 show the experimental test of AFR and torque deviation where the demands are fixed at 1.0 and $61.5N \cdot m$ respectively and engine speed is changed between 1200rpm and 1800rpm in a random drive cycle. It can be seen that the deviation in AFR is within 3% for both controllers and the regulation capability of the two controllers is satisfied on AFR. The deviation in torque is bigger, but it is still less than 5%. The performance of the two controllers are thus similar.

To test the tracking capability for torque control and interaction of torque and AFR, a series of step demands were applied. The torque demands of $\pm 10N \cdot m$ were

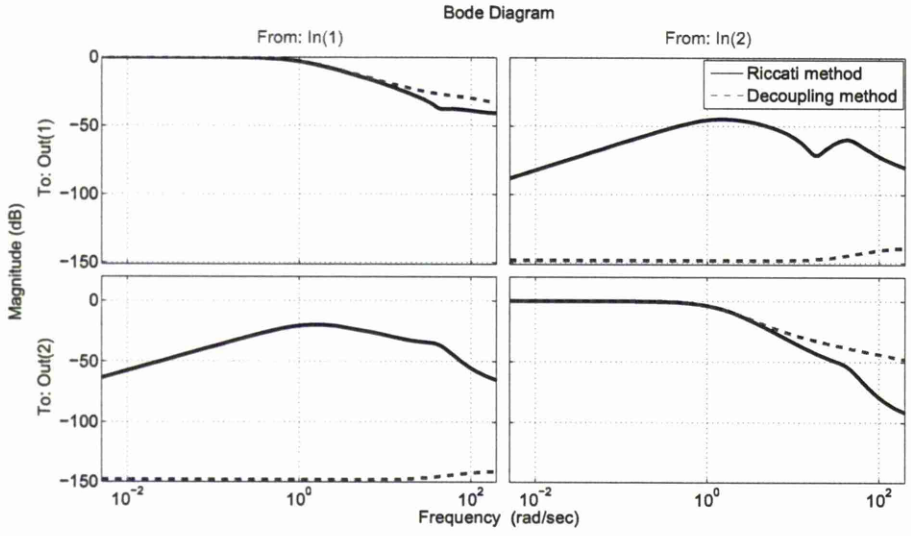

Figure 6.10: Bode magnitude plots for 'T' transfer matrix

applied to the engine, while the engine was operating at a nominal speed of 1500rpm and while the λ demand was fixed at 1.0. The measured torque production for positive and negative step demands are displayed in figures 6.14 and 6.15. The torque can usually achieve 90% of the demand within in about 4s(corresponding to 200 samples). For the $+10N \cdot m$ step demand, the settling time for both Riccati and decoupling controllers are about 4 to 5s(200 to 250 samples) and there is almost no steady-state error. For the -$10N \cdot m$ step demand, the settling time is also about 4 to 5s(200 to 250 samples). The data shows both the Riccati controller and decoupling controller satisfy the design target. The overall performances of the two controllers are similar, however the decoupling controller responds to the step demand a little faster initially, though the settling time is almost same. This is mirrored in the simulation results.

To investigate the interaction of torque and AFR, the change in AFR signal is observed when the positive and negative step torques are loaded. In figures 6.14 and 6.15 when $+10N \cdot m$ torque is loaded, the maximum AFR deviation is 0.03 with the Riccati controller and <0.02 for the decoupling optimal controller. With -$10N \cdot m$ step torque the maximum AFR deviation is over 0.03 for the Riccati controller and 0.015 for the decoupling optimal controller. In simulation it was found that the AFR signal dips when positive torque step demand is applied. In contrast, the AFR signal increases in testing. The likely reason is that the AFR is also influenced by the dynamics of the engine speed which have not been captured in the fixed speed identification data. Once the torque step is applied, the engine speed is inevitably also changed. The problem is accordingly not a simple interaction problem between AFR and torque but an interaction problem of three factors - AFR, torque and engine speed. As the same torque step should affect the engine speed equally, the different performance

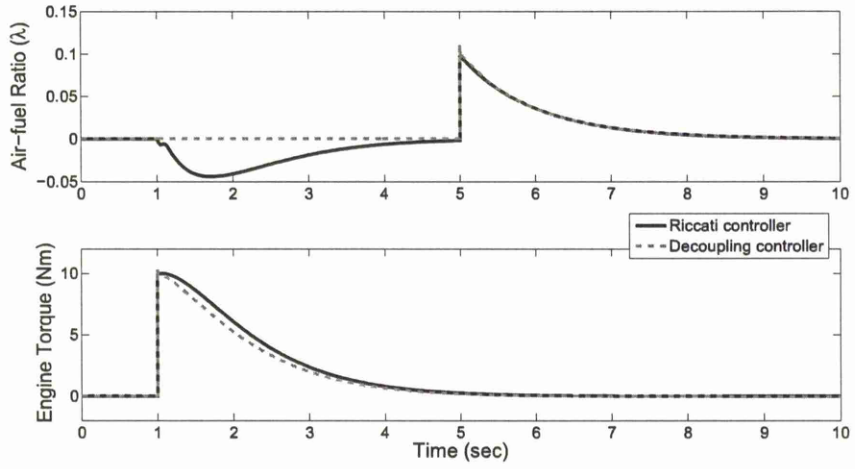

Figure 6.11: Simulation of the step responses

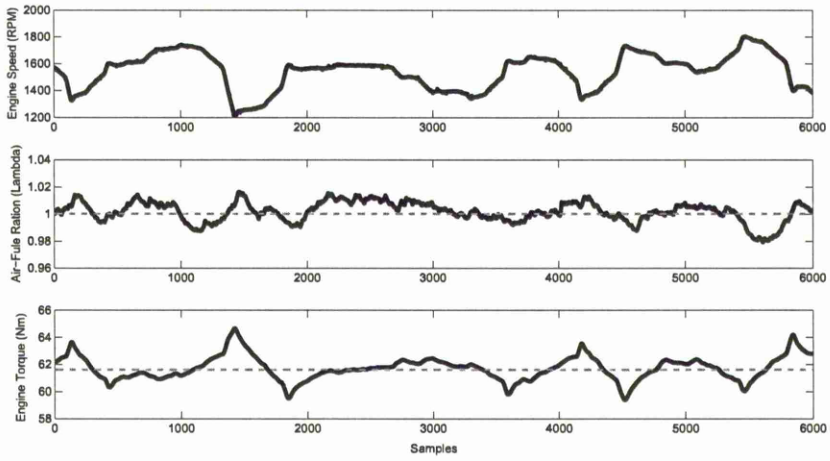

Figure 6.12: Random drive cycle test for Riccati controller

in AFR signal should be due to the difference in the controllers. The decoupling controller provides more decoupling than the Riccati controller not only in the computer simulation but also in the experimental testing. The incomplete decoupling achieved experimentally is believed to be due to the nonlinearity in the controlled system.

For the robustness testing an additional loop gain and loop delay are added until the AFR settling time is about 5s and torque settling is about 7s with a maximum deviation of AFR of 5%, and a maximum deviation for torque of $\pm 2N \cdot m$. Experimental testing then shows the Riccati controller is robust to $5dB$ and 10 time delays, and the decoupling controller to $4dB$ and 10 time delays. Accordingly, the two controllers have similar robustness.

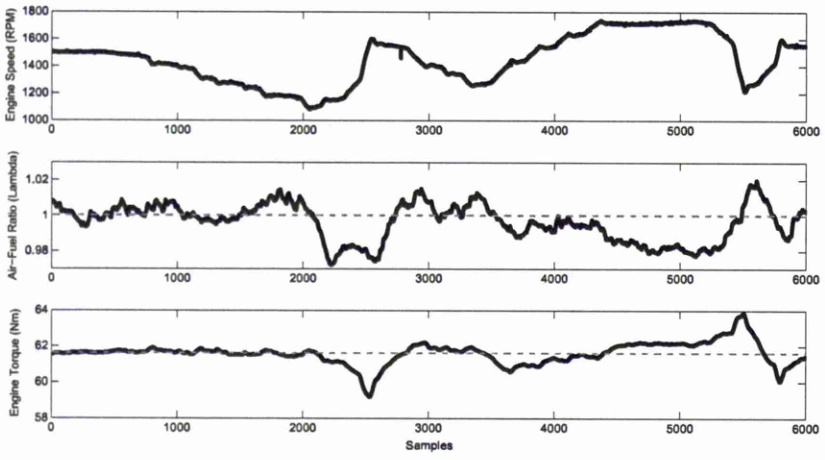

Figure 6.13: Random drive cycle test for decoupling controller

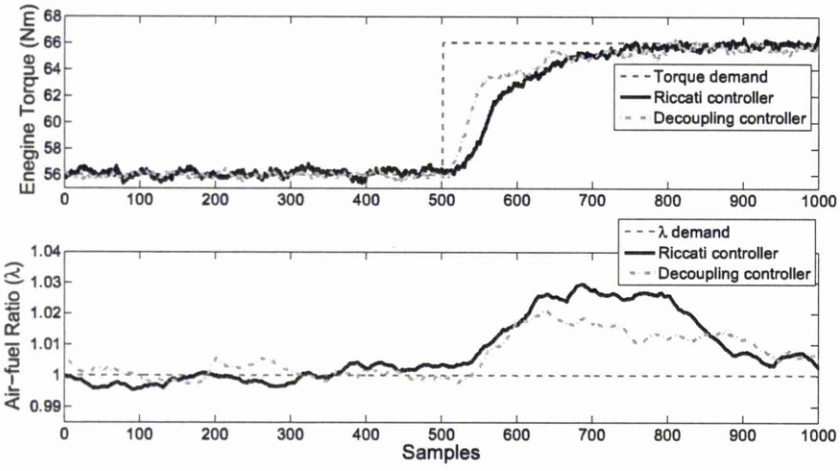

Figure 6.14: Closed loop system response to a $+10N \cdot m$ step demand

6.6 Conclusions

In this chapter the MIMO H_∞ Frobenius norm, Hadamard weighted, decoupling controller design method was developed by using the analytic function optimisation method of Helton and Merino [40]. To guarantee the internal stability, the internal stability conditions and interpolation constraints were presented. The interpolation scheme was derived based on the interpolation constraints. The H_∞ Frobenius norm mixed sensitivity optimisation approach with interpolant guaranteed internal stability was presented, and the corresponding parametric and nonparametric design procedures were detailed respectively.

A novel AFR estimator system for SI engines was presented to remove significant

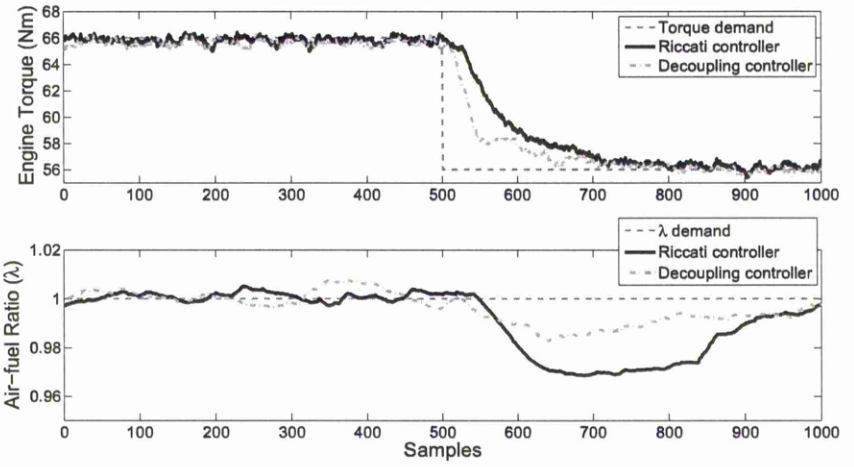

Figure 6.15: Closed loop system response to a $-10N \cdot m$ step demand

time delays in the AFR (downstream UEGO) sensor. A nonlinear black-box parameter NARX identification methodology for the estimator using FPW, MAP and engine speed was developed, and was shown to reducing the significant transport time delay of the UEGO sensor. The new method was a relatively quick and economical way to control AFR of SI engine compared to traditional approaches of feedforward controller using air estimation.

The H_∞ Frobenius norm decoupling design method was proposed for application to the estimated λ and engine torque control problem. The resulting controller was applied experimentally together with the λ estimator. The experimental results were compared with a maximum singular H_∞ norm mixed sensitivity design. Experimental testing shows the performances of the decoupling and Riccati controller are similar with similar robustness. The decoupling optimal controller is, as required, more decoupled than the common Riccati controller in the experimental testing. The incomplete decoupling achieved experimentally is believed to be due to the nonlinearity in the controlled system.

In this work, the MIMO system was established based on feeding back measured torque. Although the controllers worked well in experiment, the controllers are not suitable for implementation in actual Engine Management System (EMS). Since the EMS do not have available the measured of torque since there is no torque transducer implemented in the production vehicle. Further development of torque estimator in the subsequent chapter develops an improved estimator to estimate the torque and so to allow the controller to regulate the torque without feeding back the measured torque signal. Additionally, more experimental testing is then carried out to investigate the tracking and disturbance rejection performance and robustness of decoupling controller.

Chapter 7

Decoupled Torque-λ Control by H_∞ Frobenius Feedback Using A Neural Network Estimator

7.1 Introduction

Torque and fuelling control systems [27] are the basic core of the modern automotive Engine Management Systems (EMSs). To make the driving more efficient and safe, the torque requests not only originate from the driver but also from engine management functions and external subsystems, such as automatic transmission control unit, traction control system and anti-side slip control system, etc. The EMSs are required to track the driver's demand or other requests for torque and regulate the air-fuel-ratio (AFR) despite unpredictable disturbances. Unpredictable disturbances in both torque and air-fuel ratio may arise from road load changes, power-assisted steering demand, and uncontrolled electrical loads and other loads possibly including automatic transmission, air-conditioning units, window motors etc.. In lower cost vehicles with mechanical activated throttle driver's demand is also unpredictable. In such cases pure feedforward compensation is not feasible and feedback control is required to deal with the loads. A basic requirement, particularly with mechanical activated throttle vehicles, is that cross-coupling between load effects is minimised, so that torque load changes do not perturb air-fuel ratio affecting emissions and air-fuel ratio changes do not perturb torque delivery which may be detected by the driver affecting driveability.

The aim of air-fuel ratio control is to keep the air-fuel ratio close to stoichiometric level, which is ideal for complete combustion of the air-fuel mixture and the three-way catalytic (TWC) converter in the exhaust system also operates efficiently. Because the engine operates in a lean condition, the TWC efficiency for NOx drops dramatically and while efficiency for CO and HC raises. This is explained by the resultant abundance

of oxygen from lean operation in the exhaust gas, which in turn oxidises unburned hydrocarbons [34]. To ensure that emission reductions are maximum for NOx, CO and HC, the ideal level for air-fuel ratio where the air fuel mixture is a mass ratio of 14.7 : 1 or a volumetric ratio of around 9500 : 1, [45],[6],[54]. Generally, the air-fuel ratio is measured by λ which is defined as:

$$\lambda = \frac{AFR_{actual}}{AFR_{stoichiometric}}$$

A change in λ of 0.1% around $\lambda = 1$ would double the emissions rate although across transients of engine speed and torque, deviations of 2 to 3% around $\lambda = 1$ over short periods of time are acceptable[54]. The Figure 7.1 illustrates the optimum air-fuel ratio for the operation of a TWC.

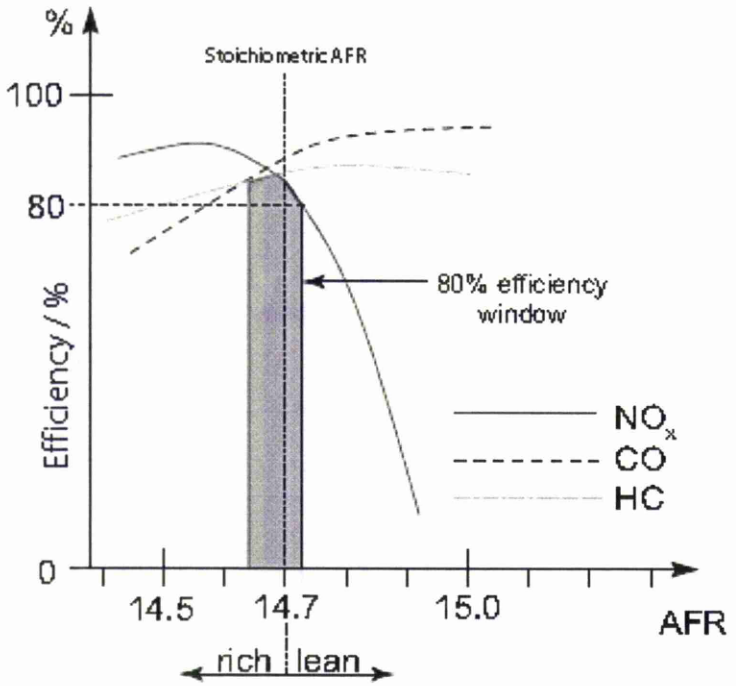

Figure 7.1: Efficiency window of a TWC

This chapter presents a novel method of decoupling the AFR and torque paths in both trucking and disturbance. The new approach which combines the Frobenius H_∞ control of the last chapter with a novel nonlinear AFR and torque estimator is shown to provide accurate fast acting control. The method which is fully systematic thus the potential to eliminate the excessively time consuming calibration required for current feed forward controllers.

The chapter is organised as follows. In section 7.2, the difficulties and requirements for air-fuel ratio and torque control are discussed. Some previous modelling and control

design methods both for air-fuel ratio and torque are reviewed. Alternative modelling methods and control strategies are then proposed. In section 7.3, non-linear modelling and estimation methods for air-fuel ratio and torque are introduced. The parameters and structure used for air-fuel ratio and torque estimation models are determined. The modelling technique chosen for this work is also determined. The fast feedback control scheme for multivariable torque/air-fuel ratio control is proposed and discussed. In section 7.4, a black-box method for air-fuel ratio and torque modelling using Neural Network is first proposed. The Neural Network modelling technique is also introduced and discussed. Then, the experimental setup is detailed. The non-linear Neural Network identification procedure including excitation signal setting, data collection, Neural Network generation, off-line and on-line validation for torque and air-fuel ratio estimation are detailed respectively. In section 7.5, the linear identification for torque and air-fuel ratio estimation is detailed, a 2×2 multiple-input-multiple-output (MIMO) linear model identified with torque and air-fuel ratio estimation is provided. In section 7.6, a decoupling controller based on the 2×2 MIMO linear model is obtained by using the optimal decoupling design technique developed in the last chapter. For comparison, a conventional Riccati controller is also obtained. In section 7.7, the results of a series of experimental tests for tracking and disturbance rejection capability of torque and air-fuel ratio control are displayed, compared and analysed. Section 7.8 concludes the modelling and control method proposed in this work, and comments based on the experimental results are summarised. The possible extension for future work is discussed.

7.2 Modelling, Estimation and Control for Engine Torque and λ

This section describes the requirements and proposed approach of this chapter for combined torque and air-fuel ratio control. Due to the strict requirements of air-fuel ratio control, a fast and accurate controller should be applied to this case. However, the traditional feedback control theory can not easily satisfy the fast requirement for AFR control. Because the air-fuel ratio of a SI engine is detected by oxygen sensors such as a heated exhaust gas oxygen (HEGO) or universal exhaust gas oxygen (UEGO) sensors which is usually located in the exhaust manifold, there are large transportation lags between the input signals of air and fuel and the output of measured air-fuel ratio signal. To obtain a fast controller, a feedforward control strategy is conventionally employed in the Engine Management System (EMS) in industry. The feedforward compensator is used to estimate the air mass entering the cylinders from Manifold Absolute Pressure (MAP) or Mass Air Flow (MAF) sensor and engine speed signals. The fuelling signals such as Fuel Pulse Width (FPW) are the estimated from the air mass infor-

mation. The fuelling signal determines the amount of the fuel injected to ensure the correct air-fuel ratio[34]. In industry, the FPW signal is calculated by using look-up tables which are generated as a number of static operating point mappings for a large number of speeds, loads and temperatures. In the previous works of Dickinson[20] and Rivara[72], the feedforward compensator was generated by identifying dynamic inverse engine models directly. This approach had the advantage that it could potentially to save 'calibration' time when applied in industry. Both inverse algebratic NARMA and NN modelling methods were used in the work. Additionally, Parameter-Space and loop shaping techniques were used to design feedback controllers around the nonlinear compensated system to reject disturbances and correct the steady-state error.

A alternative method proposed here is to generate an estimator model of the air-fuel ratio (measured as λ). By use of the estimator model, the delays are removed and the predicted (or estimated) $\hat{\lambda}$ is significantly less delayed than the UEGO measured λ signal. A feedback control system using the predicted $\hat{\lambda}$ rather than the measured λ is significantly faster than feedback system using the UEGO measured λ. The control behaviour of the feedback system using $\hat{\lambda}$ can also regulate the deviation of measured λ before it measures. An additional motivation for the use of feedback controller in this setup is that it also potentially reduces the 'calibration' time required to produce the much higher accuracy feedforward fuelling maps. Due to the nonlinearities of the air-fuel ratio path, a Neural Network model was used to generate an accurate air-fuel ratio estimator model $\hat{\lambda}$ in this work. The time delays are removed from the estimator model by shifting the identification data signals to synchronise the behaviour of input and output signals.

Agility of response and good tracking capability with low steady-state accuracy are the main requirements for engine torque controllers. As the EMSs can not measure the torque or power directly, the produced torque is required to be estimated by the EMS. Previous work has proposed several methods based on instantaneous torque estimation [73], [8], [52] for diagnosis purposes such as misfire detection, injector faults. For control purposes, the mean value torque is required. The estimator for mean torque requires implementation of a nonlinear function of many variables such as fuel mass in cylinder, air-fuel ratio, engine speed, ignition and injection time, etc [34]. In most research including in [81], [43], [53], static torque models are used. Parameters used to form the estimate are typically MAP, AFR, engine speed and spark advance, etc. [53]. To get a fast controller, a feedforward control strategy is also used for torque. In [47], the feedforward controller was used to estimate Mass Air Flow (MAF) by a static nonlinear function of torque request, AFR and engine speed. The output of the static function provided a MAF request to the additional MAF-torque controller. The dynamic models of the compensated system combined with feedforward static nonlinear function, MAF

controller and engine was identified with conditions of different AFR, engine speed and torque request. A nominal model was then used to design a H_∞ controller for the compensated system. A alternative method used in this thesis and described in this chapter is to identify a nonlinear estimator with time delay removed from the torque path using similar Neural Network technique in the air-fuel ratio path. The motivation for the method proposed here is not only to estimate the torque accurately but also to design a quick acting controller around the estimator with the delays removed.

Although the time delays and some nonlinearities are theoretically removed from the air-fuel ratio and torque path by using estimator, some system nonlinearities still found to exist when the estimator is implemented. The nonlinearities remaining are caused by unmodelled reaction to variation of loads or engine speed. The deal with this remaining nonlinear variation to load and speeds a feedback controller is implemented in this chapter based on modelling the nonlinearity by a set of linear models. Thus to model the nonlinear dynamics of the air-fuel ratio and torque paths at different engine speeds, several linear models can be identified with the estimated air-fuel ratio/torque at different fixed engine speeds to cover the same range of engine speed caused the nonlinearities. The nonlinear dynamics can thus be modelled as 'multiplicative dynamic uncertainty' around to a nominal model [47]. This means that the nominal model with uncertainty can capture the system nonlinear dynamics. As described in previous chapters the effect of 'multiplicative dynamic uncertainty' on the plant may be expressed as:

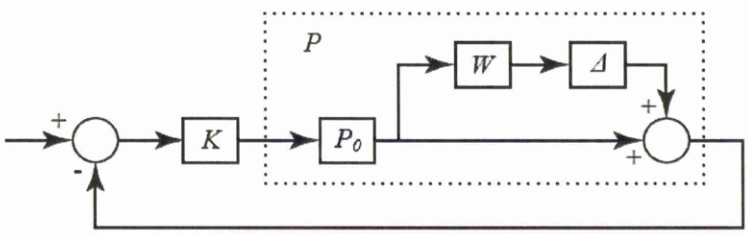

Figure 7.2: The system with multiplicative dynamic uncertainty

$$P = [1 + \Delta W]P_0 \tag{7.1}$$

where P is the perturbed plant, P_0 is the nominal plant and W is the uncertainty weighting function. To bound and normalise the uncertainty, we take:

$$\|\Delta\|_\infty < 1 \tag{7.2}$$

The equivalent robust stability condition is then:

$$\|W_T T\|_\infty < 1 \tag{7.3}$$

where T and W_T are the complementary sensitivity and corresponding weighting function respectively. Through bounding the complementary sensitivity, the uncertainty (system nonlinearities) in the nominal model is handled by the robust linear design. Then a linear robust controller based on the nominal model is robust to the system nonlinearities and this is the approach that will be used in the subsequent work of this chapter.

Due to the requirements for speed of response to driver demand and environment disturbance, the combined air-fuel ratio and torque controllers should be fast acting. Conventional H_∞ techniques(such as in [47]), and other modern control techniques have been employed on this problem recently. In the work of [53], Fuzzy Logic control was applied to the torque control problem, and experimental results were also given. Dutka presented a Generalized Predictive Control (GPC) design method and gave corresponding simulation results for the multivariable control of air-fuel ratio and torque in [23].

In the new approach described in this through feeding back the estimation of air-fuel ratio and torque without delays, a feedback controller using any general robust linear control algorithm, such as a Riccati method, can give a quick response to both torque and AFR. Another problem focused on this work is the decoupling of multivariable system. To design a decoupling multivariable feedback controller around the estimator, the novel MIMO Frobenius H_∞ norm decoupling optimisation control technique which was developed and described in the previous chapter is used in this chapter. This approach is based on the technique which as previous described, uses disk iteration to solve quasi-Nehari problems [40]. Some control design techniques were developed based on this algorithm in Single-Input-Single-Output (SISO) case and applied experimentally in [100].

7.3 Nonlinear Modelling for λ and Torque

For the proposed control method an accurate estimator model for engine is required. The input and output signals measured on engine are air bleed valve(ABV), fuel pulse width(FPW), integrated manifold absolute pressure (MAP), engine speed, air-fuel ratio and torque. The estimator model for λ and torque must accordingly be based solely on these measured signal. Different parameters were used to estimate the AFR and torque respectively. As in the last chapter, FPW, MAP and engine speed was still used to generate the estimator for AFR. Additionally, the measured AFR was treated as another input of the estimator to create a dynamic model. Similarly to the work of [53], MAP, AFR and engine speed are used to create a torque estimator. The AFR and torque were required to be identified as two multiple-input-single-output

(MISO) models with the time delay removed using a Neural Network technique. The structures are shown in figure 7.3 and 7.4. The estimators are then incorporated in a 2 by 2 negative feedback control system as shown in figure 7.5. The control outputs for feedback controller K are ABV and FPW. In the estimator model $\hat{G}$ which includes the AFR estimator $\hat{G}_1$ and torque estimator $\hat{G}_2$, the estimated value of air-fuel ratio ($\hat{\lambda}$) and engine torque ($\hat{T_E}$) were fed back to the controller K directly. A slower time-scale controller K_{trim} was to trim slow drift in the estimator and ensure zero steady-state AFR error. Because of the slow time-scale nature of any drift, the choice of structure of K_{trim} was not important, and a general PID structure was chosen. The parameters of the PID compensator were not critical to the system performance and were easily found by on-line tuning.

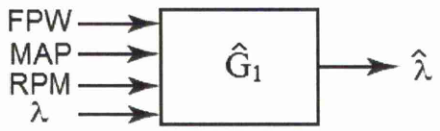

Figure 7.3: structure of λ estimator

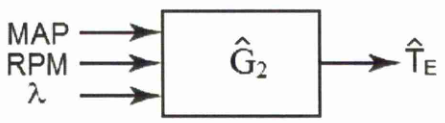

Figure 7.4: structure of torque estimator

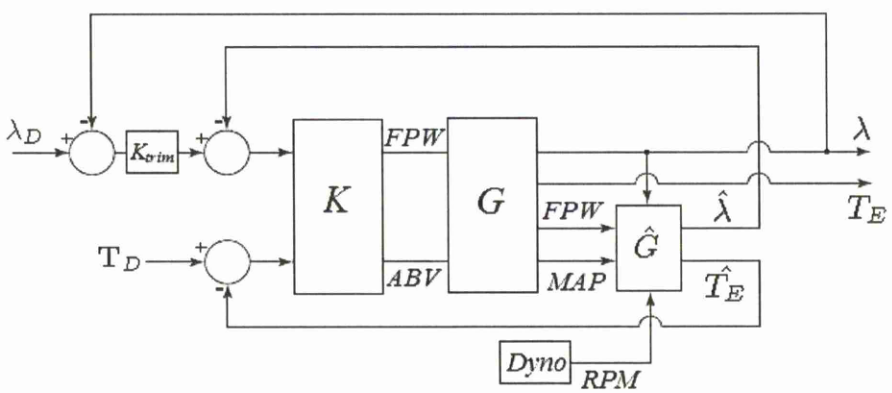

Figure 7.5: Feedback controller structure with estimator

7.4 Neural Networks for Nonlinear λ and Torque Prediction Based On Black-box Method

7.4.1 Neural Networks Technique

For the proposed estimator, the Neural Network is used to model for air-fuel ratio and torque. Basically, Neural Networks come into two classes: feedforward networks and recurrent networks [22]. The feedforward network is a nonlinear function based on its inputs only. A very common type of feedforward network is multilayer perceptron (MLP) networks, which are extensions of a single layer networks. The only difference is that MLP networks contain one or more hidden layers. Compared to a simple single layer network an MLP is a more complex network and can thus fit higher order or more complex functions. Frequently, the activation functions of the hidden layers in MLP networks are tan-sigmoid function; the final layer containing the pure linear activation function. The parameters are assigned to the connections. As an important intelligent computation method and universal function approximation tool, the feedforward network has been applied to a wide range of applications, including curve fitting, pattern classification and nonlinear system identification.[82]. Recurrent networks are also called nonlinear autoregressive network with exogenous inputs (NARX) or feedback neural network which is equivalent to a feedforward network with feedback path as shown in 7.6

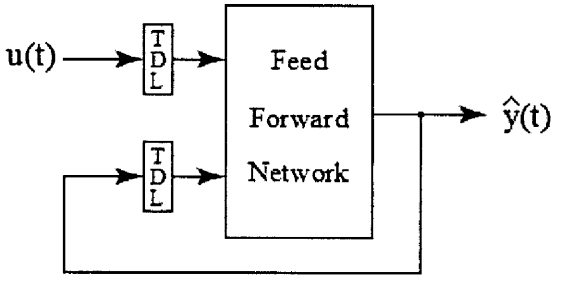

Figure 7.6: Parallel NARX network[66]

The standard Recurrent network is a parallel NARX network, where $u(t)$ represents input, $\hat{y}(t)$represents estimate output, the output is fed back to the input of a feedforward network. Due to the feedback path the Recurrent network is a dynamic model[55]. The recurrent network can work with time delayed signals. Such network can be applied to many more applications especially for delayed system such as in process control. An extension for parallel NARX network is a series-parallel NARX network[66], where the measured output $y(t)$ is treated as a input of the feedforward network as shown in 7.7.

Although it is a feedforward architecture, such network is still a dynamic network. As

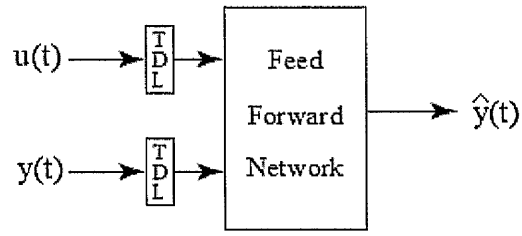

Figure 7.7: Series-parallel NARX network[66]

mentioned, an estimator with time delays removed is established in this work. Both NARX dynamical network and feedforward multilayer perceptrons network are used to create the estimator for air-fuel ratio and torque dynamics based on synchronised input-output data.

7.4.2 Experimental Setup

The experimental work described here was carried out on the low inertia electric engine dynamometer at the University of Liverpool. The experimental engine was a 4-cylinder Ford Zetec 1.6 L 16 valve engine with sequential port fuel injection. The actuators on the engine which can be manipulated by dSpace-Autobox are air bleed valve (ABV), fuel injector and spark. In this work, the control signal for spark ignition was fixed. The ABV on the engine can be considered equivalent to an electronic throttle. For testing purposes the ABV duty cycle is limited to the 10% to 90% range to avoid stall and exceessile loading. With the ABV is fully opened, the engine speed attains slightly over 2500rpm. For the identification work, the dynamometer was maintained running under speed-control mode, so that the speed of the dynamometer was controlled by the researcher. The dynamometer's shaft and the engine's shaft were coupled rigidly. The brake torque can be loaded by the dynamometer to request the torque production from engine. The maximum brake torque which can be loaded by the dynamometer is about 35 to $40N \cdot m$ in the general operating range. The torque production was measured by a Vibrometer Torquemaster TM112 with a 200Nm rating. The Matlab Real-Time Workshop (RTW) software was used to interface to the dynamometer Mentor power amplifier using a dSpace-Autobox to enable a Simulink representation of the control system and estimator system, etc. to be applied in real-time.

7.4.3 Inherent Delay

To create an estimator with time delay removed, the delays between all input and output signals need to be determined. This requires shifting the identification data to produce behaviour synchronised and causal input-output data. To infer the magnitude of the delay accurately, the basic knowledge of the system dynamics should be understood, especially for the air-fuel ratio path. There have been many detailed models of the dynamics between the actuation of fuel injector to the response of air-fuel ratio measured at the oxygen sensors [95], [51], [89], [54]. In some previous work, Wang and Detwiler performed studies of the individual steps of the delay [90], [87], [88]. It was found that the wall wetting effect of the fuel delivery is the most important source for the transient air-fuel ratio excursions [89]. This factor should be considered carefully when inferring the delay. Generally an oxygen sensor is located at a confluence point in the exhaust manifold. There would then be a lag of a few combustion cycles. This lag should also be considered as a contribution to the delay and is determined largely by the volume of the exhaust plenum. In simplified terms, the overall delay t_{total} for air-fuel ratio path is composed of:

1. t_{IPS}, the delay is termed 'induction to power stroke (IPS) delay' which is due to the time from the air and fuel request to the wall wetting, to the inlet valve opening, land in the four stroke, to the exhaust valve opening.

2. $t_{transport}$, the time lag taken for the exhaust gas to travel from the exhaust valve to the UEGO sensor.

3. t_{UEGO}, the time delay caused by the reaction time of the UEGO sensor in reporting the measured air-fuel ratio.

The induction to power stroke delay is a natural feature the IC engine. For the experimental 4-cylinder Ford Zetec engine the IPS delay is approximately two combustion events which is equivalent to:

$$t_{IPS} \approx \frac{2\pi}{\omega_E} \tag{7.4}$$

where ω_E is the engine speed in units of rad/s. The dynamics between the actuation of throttle and fuel injector to the response of torque production faster than the dynamics of the air-fuel ratio path. The overall delay t_{total} for the torque path is basically the same as the induction to power stroke (IPS) delay.

7.4.4 Excitation Signal for Nonlinear Identification

To identify the black-box model, the input/output dynamics need to be recorded during specifically designed identification experiments [57]. From [35], both Gaussian white

noise test signals or pseudo-random multi-level test signals can detect the linear and nonlinear features of a process. Therefore, the random binary sequences are suitable for linear systems since they have an auto correlation similar to that of white noise with a spectrum similar to that of a Gaussian signal. However for nonlinear system, the amplitude dependencies input signals must have more than two levels. Generally speaking, sufficient and suitable excitation signals can excite the output signals well. The dynamics exposed from experiment can represent the original system good enough. The requirements for a good excitation testing signal include a suitable period and amplitude of the signal. The perturbation period for each input signal should be short enough to excite all the dynamics of the system but still be long enough to allow high frequency dynamics and rapid transient responses. Additionally, uncorrelated test signals are necessary for each input [26].

Due to the nonlinearities of air-fuel ratio and torque path, scaled uniform random numbers with a DC bias were selected to be the excitation signal for the identifications of λ and torque estimations. The inputs of the λ estimator were FPW, MAP and engine speed. The FPW signal was the control signal for the fuelling actuator of engine. The excitation of FPW could be achieved by applying a test signal directly to the actuator. The engine speed was controlled by the dynamometer, the excitation of engine speed could be achieved by applying a test signal to the dynamometer. As the ABV and engine speed signals had higher correlation with MAP, the excitation for MAP in the experiment was determined by the sufficient and suitable excitation of ABV and speed of dynamometer. For the torque estimator, the inputs were MAP, λ and engine speed. The MAP λ and engine speed signals depend themselves on the excitations of ABV, FPW and dynamometer.

The suitable period of test signal should excite the dynamics of all across the full frequency range of the system. The load (brake torque) and engine speed signals were restricted due to the requirements to prevent the saturation of the λ signal (λ is usually saturated when $0.5 < \lambda > 1.4$). To overcome this restriction, the identification signal generation has to be mapped such that the saturation of the λ signal reported by the UEGO sensor is prevented, while the exciting signal is still 'random' to expose the engine dynamics especially in the λ signal adequately.

A simple method to satisfy the requirements mentioned above is to map the FPW signal with the MAP signal. By defining an appropriate relation between FPW and MAP signal, the λ signal can be regulated about 1. If the excitation is supplemented by an extra scaled, biased, uniformly random number sequence superimposed upon the FPW signals and based on the relationship, the minor deviations from stoichiometry of the resulting λ signal is still 'randomised'. Meanwhile the saturation of the λ signal is prevented. To determine the required relationship between the FPW and the MAP

Signal	period	range
ABV duty(%)	1.5s	40 to 60
FPW (μs)	1.7s	2200 to 6500
Engine speed(rpm)	5s	1000 to 2000

Table 7.1: Exciting signals for nonlinear identification

signal, dynamic experimental engine data of FPW and MAP was collected. The engine speed was changed from 1000 rpm to 2000 rpm and was set to run at a set of fixed speed points. At each speed point, ABV duty was varied to give different engine torques. The FPW was then adjusted to give the λ value of 1. Both FPW values and corresponding MAP values at each engine speed and torque condition were reordered. The relationship was found by plotting the FPW value and MAP value. It was found that the relationship is almost linear. A static function was chosen to represent the relationship:

$$FPW = 91 \times MAP + 2100 \tag{7.5}$$

Based on the empirical relation of equation 7.5 the identification signal for the FPW was determined. Table 7.1 shows the parameters of the nonlinear identification test signals.

The effect of the excitation signal fluctuations is that the integrated MAP fluctuates between 50 and 100 bar^2, the λ fluctuates between 1.4 and 0.7 and the load (brake torque) fluctuates between -10 and 35 $N \cdot m$. The first 2000 samples of the excitation signals and the resulting output signals of the identification testing are shown in figure 7.8

7.4.5 Neural Networks Employed for λ Estimation

Neural Networks Generation

Once enough identification data was recorded from the dSpace-Autobox, the identification data required to be pre-processed. Initially, the sampling program was triggered at every 4^o of crank angle when dSpace-Autobox collect the data, but the data contain too much high frequency noise. The data was therefore down sampled to every 180^o at crank angle to remove the high frequency noise. The data for identification was divided into three parts. The first part was used for estimation, the second was used for validation and the last part was used as a 'hold-out' dataset for offline validation [37].

In this work, a neural network model was selected to identify the nonlinear λ estimator. As discussed in the previous sections, the aim was to create a predictive on-line

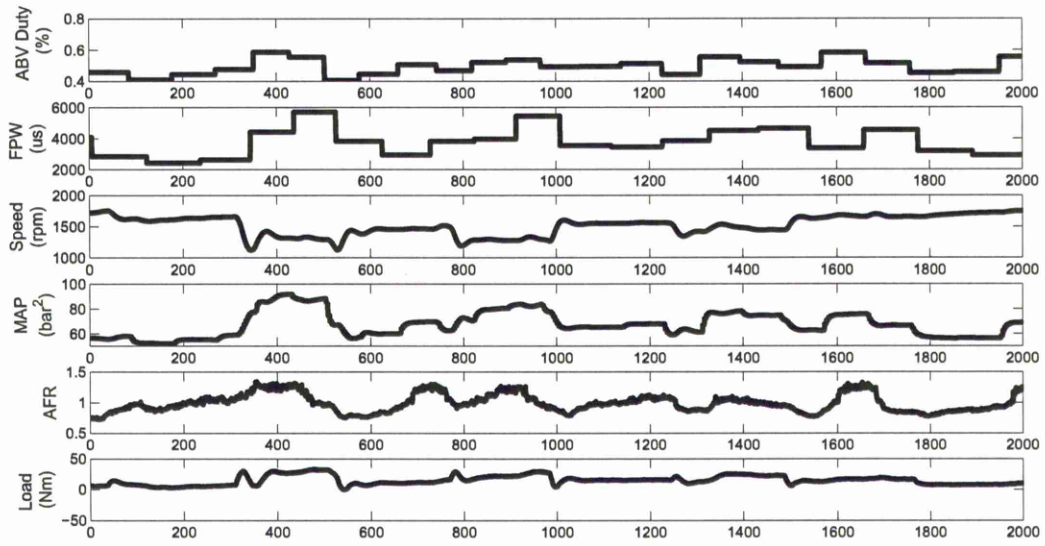

Figure 7.8: Data record of nonlinear identification signals

λ estimator by removing the time delays. Accordingly the time delays between inputs and output were determined and removed first. The large time delays exiting between fuelling input and output of λ were due to the physical transport distance, wall wetting dynamics and sensor and actuator response time. From initial testing, it was found that approximately 20 time delays existed between fuelling signal and λ signal, and approximately 2 time delays existed between the fuelling and MAP signal. As the engine speed was controlled by the dynamometer, it was determined by the signals from ABV or FPW inputs, there was in the theory no correlation between the fuelling and engine speed signal. It was found that about there were approximates 18 delays between λ and engine speed, which was equivalent to 2 delays between FPW and engine speed signal. As mentioned the measured λ is treated as an input of the estimator. During training, the data of λ was used again, it was considered as a variable measured synchronously with FPW and no delays exited between this variable and FPW. Consequently in the pre-processing of the identification data, the λ(treated as output), MAP and engine speed data were shifted forward 20, 2 and 2 delays respectively, FPW and λ(treated as input) were kept as they were. The following figures show the model with different time delays and the model without time delay of the λ estimator. Although there is a feedback path for the measured λ, feedforward architecture is still used for the neural network of λ estimation.The neurons and delays, etc. of the network would then be determined.

For implementation in an EMSs low computation time, a small model size and simple network structure is preferable. To obtain a suitable and simple structure, a network

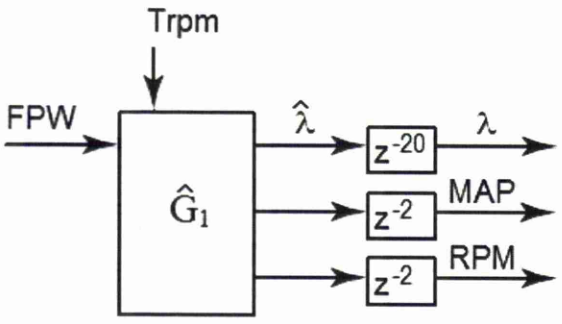

Figure 7.9: The model with time delay

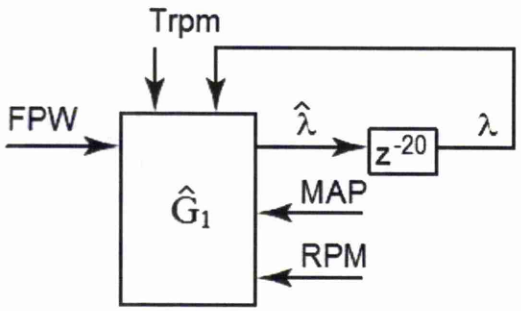

Figure 7.10: The λ estimator without time delay

structure of 3 layers was given initially. Then several networks, with reduced layers and neurons were gradually generated and compared. The comparisons included the r^2 values, the number of effective parameters, and the number of available parameters, the resulting sum of squared errors over the estimation and validation training data. By achieving a compromise between model estimation accuracy, structure complexity and processing speed, a two layer network structure was found suitable to generate the λ estimator.

Figure 7.11 shows the 2 layer multilayer perceptron (MLP) neural network structure, where IW represent input layer weight matrix and LW represent output layer weight matrices. The $f1$ activation functions are tansigmoid functions, $f2$ are pure linear activation functions. The first layer contains 9 neurons. As the network is a single output network, layer 2 contains 1 neuron. Generally, the current and previous data of the inputs which are required for generation of the model can be represented by inputs passing through the specific tapped delay line (TDL) in the neural network. In this network, the 3 inputs pass through 8 TDLs where the output was a vector made up of the input signal at the current and previous input signal. The input-output behaviour of the network is thus of the form:

$$\hat{\lambda(t)} = f(FPW(t), \ldots, FPW(t-8), MAP(t), \ldots, MAP(t-8), \\ RPM(t), \ldots, RPM(t-8), \lambda(t), \ldots, \lambda(t-8)) \quad (7.6)$$

To prevents over-training, this network is trained by 'Bayesian-Regularisation back

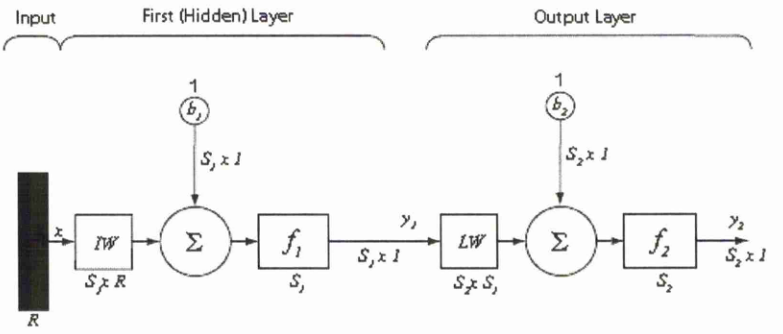

Figure 7.11: Schematic of a Two Layer Network

propagation with early stopping criterion'. 'Early stopping' occurs when the mean squared error of the validation dataset begins to increase indicating the optimal training epoch for that network has been reached.

Off-line Neural Networks Validation

After the training, a candidate network was obtained. Then, the 'hold-out' data was used to validate whether the network estimates the λ adequately. Figure 7.12 shows the comparison of measured λ and estimated $\hat{\lambda}$. To quantify the accuracy of the estimated $\hat{\lambda}$ value, the 'R^2' (coefficient of determination) is calculated. The 'R^2' coefficient is defined as:

$$R^2 = 1 - \frac{\sigma_\epsilon^2}{\sigma_y^2} \tag{7.7}$$

where σ_ϵ^2 represents the variance of the model residuals and σ_y^2 the variance of the system output. Based on the comparison of hold-out data and estimated output the R^2 value was found to be:

$$R^2 = 0.9661 \tag{7.8}$$

This indicates that over 96% of the variance in the measured λ is explained by the estimated $\hat{\lambda}$ from NN output.

On-line Neural Networks Validation

The off-line validation, indicated that the candidate network provided a good estimate of λ. The network was accordingly implemented in dSpace and the validated on the engine. In the on-line validation, the ABV and FPW were perturbed randomly, and the resulting λ and estimated $\hat{\lambda}$ were both recorded. Figure 7.13 shows the comparison

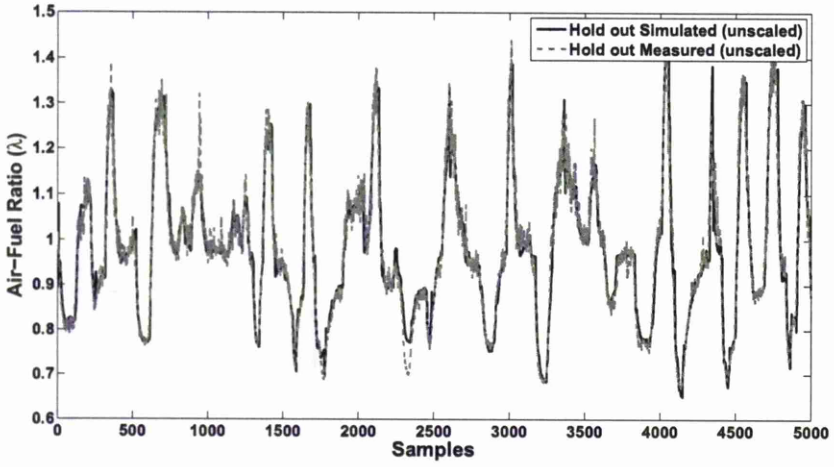

Figure 7.12: Validation of the estimated $\hat{\lambda}$ fit to measured λ(without delay)

of actual λ and estimated $\hat{\lambda}$. From that figure, it is evident that the estimated $\hat{\lambda}$ is

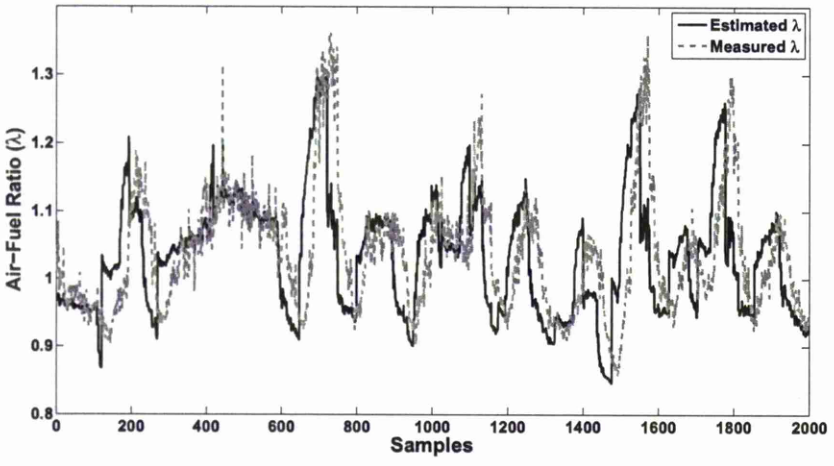

Figure 7.13: Comparison of estimated $\hat{\lambda}$ and actual measured λ

always significantly in advance of the actual measured λ as required.

7.4.6 Neural Networks Employed for Torque Estimation

Neural Networks Generation

As in the process for the λ estimator, the data for identification was divided into three parts. There were used for estimation, validation and off-line validation of a 'hold-out' dataset respectively. Again, a neural network model was selected to construct the torque estimator. Based on a similar idea to that used for the λ estimator, an estimator

with time delays removed for torque was established. As mentioned, the estimation of torque was based on MAP, λ and engine speed signals. As the output of the MAP signal occurred prior to the λ and torque signal, the λ and torque signals should align to the MAP signal once time delays are removed. Because there was no delay between engine speed and torque signal, that is equivalent to 2 delays existing between MAP and the engine speed signal. From ABV signal to MAP a very small delay (of about 2) existed, so the estimator aligning to MAP signal was also delayed 2 sampling periods from ABV and FPW signal. As mentioned, 20 time delays existed between FPW signal and λ signal, so 18 delays existed between MAP signal and λ signal. As almost no delay existed between engine speed and torque signal, this implies 2 delays between MAP and engine speed. The λ, torque and engine speed signal are therefore shifted forward 18, 2 and 2 delays respectively. The following figures show the unsynchronised (with original time delays) model and synchronised (without time delay) model of the torque estimator. In the controller design stage, the estimator is then used as a quick feedback controller.

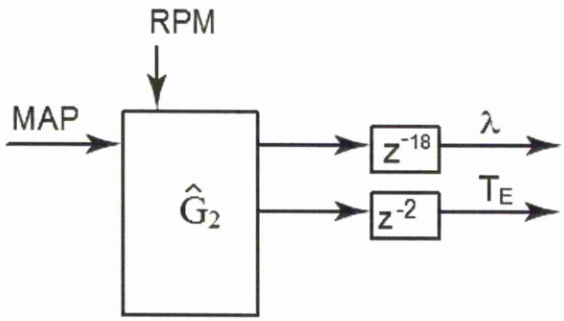

Figure 7.14: The model with time delays

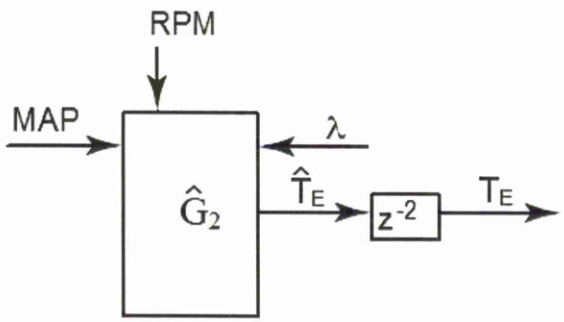

Figure 7.15: The torque estimator without time delay

Similarly, a 2 layer multilayer perceptron (MLP) neural network structure was selected to estimate torque. Figure 7.11 shows that structure. As in the λ estimator, the $f1$ activation functions are tansigmoid functions, and the $f2$ are pure linear activation

functions. While the first layer contains 9 neurons, layer 2 contains 1 neuron. In this network, all 3 inputs pass through 8 TDLs where the output is a vector made up of the input signal at the current time and the previous input signal. The resulting network has the form:

$$\hat{T_E}(t) = f(\lambda(t), \ldots, \lambda(t-8), MAP(t), \ldots, MAP(t-8), RPM(t), \ldots, RPM(t-8)) \tag{7.9}$$

The same criterion as in the λ training, 'Bayesian-Regularisation back propagation with early stopping' is used in the torque model training to prevents over-training.

Off-line Neural Networks Validation

Once a network was obtained by the training the 'hold-out' data was used to validate the network. Figure 7.16 shows the comparison of measured (brake) torque and estimated (brake) torque. The 'R^2' (coefficient of determination) is calculated to quantify the accuracy of the torque estimator. From comparison of measured torque and estimated torque, the R^2 value was determined as:

$$R^2 = 0.9010 \tag{7.10}$$

This indicates that over 90.10% of the variance in the measured engine torque is explained by the estimated torque from NN output.

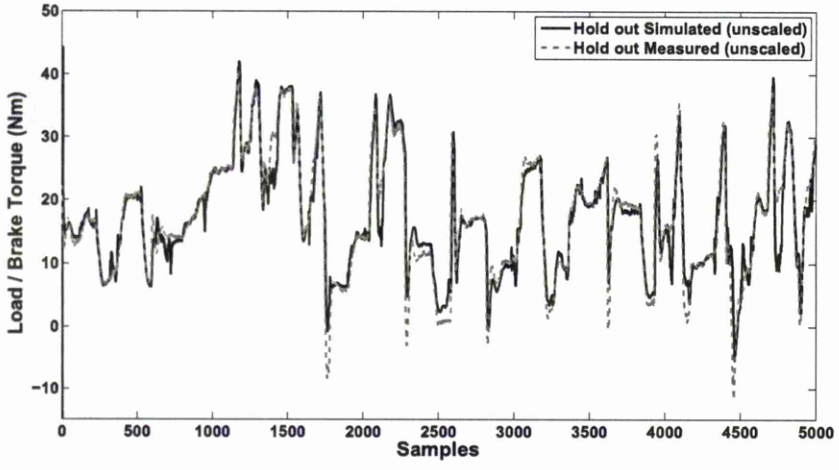

Figure 7.16: Validation of the estimated torque fit to measured torque(no delay)

On-line Neural Networks Validation

After off-line validation on-line validation was carried out. The ABV and FPW were perturbed randomly, the resulting output of estimated (brake) torque and actual mea-

sured (brake) torque were both recorded. Figure 7.17 shows the comparison of actual torque and estimated torque. From that figure, it is evident that the estimated torque

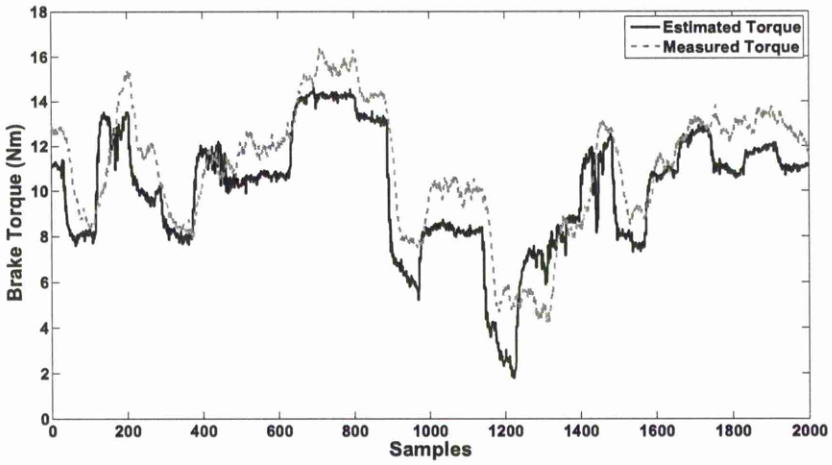

Figure 7.17: Comparison of estimated torque and actual measured torque

occurs slightly prior to the actual torque.

Having obtained the λ and torque estimator, the linear identification is carried out.

7.5 Linear Identification for MIMO Model

The linear robust feedback controller design was based on a linear model identified around the engine-estimator system. The inputs of the MIMO linear model were ABV duty cycle and FPW, the outputs were AFR-estimate and Torque-estimate. As mentioned the system nonlinearities can be captured as 'multiplicative uncertainty' around the nominal model. To model the dynamics at the different engine speed, several linear models were thereby established. The engine speed was regulated at 11 different speed setpoint values in dynamometer speed control mode in increments of 100rpm, from 1000rpm to 2000rpm. For all tests, the dynamometer controller regulated speed within of the nominal.

7.5.1 Excitation Signal for Linear Identification

Generally Pseudo-Random Binary Sequence (PRBS) signals were used to excite a wide range of frequencies in the linear identification. The magnitude of the PRBS ABV duty cycle signals was ± 0.035 (3.5%), which were applied about the nominal value of ABV duty cycle required to maintain the set point, which depending upon the engine speed was found to vary from 0.4 to 0.5(40% to 50%). Although a PRBS test signal similar

used in perturbation of ABV was still used in FPW identification, the resulting FPW exciting signal looked like a random signal due to the FPW signal was mapping to MAP signal. The magnitude of PRBS FPW was set to $\pm 300\mu s$, the nominal value was varied mapping to MAP signal. The chosen FPW exciting signal was varied between $\pm 850\mu s$ at each set-speed and between 2200 to $5000\mu s$ overall. The period of the exciting signal was designed to obtain a good fit quality and retain high frequency information. From testing, the resulting optimal ABV signal was found to be a PRBS of period 1.1s and the period for FPW signal was set to 1.3s. Figure 7.18 shows the test signals used in the linear identification.

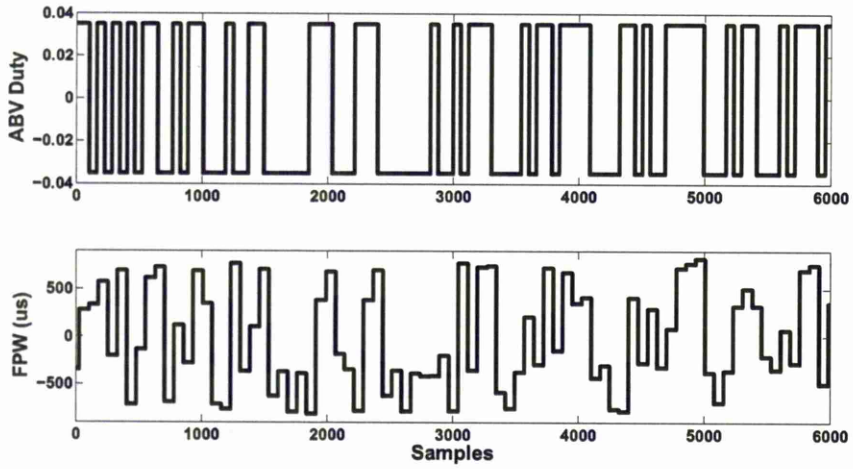

Figure 7.18: Exciting signals of linear identification

7.5.2 ARMAX Identification for MIMO System

An ARMAX identification algorithm [59] was used to obtain each of the set-speed models. A 2 × 2 MIMO model was found by identifying two MISO models, each MISO model having common denominators in its elements. For the air-fuel ratio path MISO models the best trade off between model order and fit quality was found with second order linear transfer functions with delays removed. All 11 of the air-fuel ratio path transfer functions were found to have the fit rate of approximately 73% to 75%. For the Torque production path MISO model, second order linear transfer function between 2 delays were found. The fit rate for all the torque path models was around 70%. The superimposed array of bode plots for the eleven 2 × 2 models are shown in figure 7.19. Once all 11 models were established, a nominal model was obtained from the set of frequency responses. Following Skogestad and Postlethwaite [80], the nominal model was obtained by discretising the frequencies and by finding the minimal radius disk enclosing all responses at each frequency in the complex plane. The centre

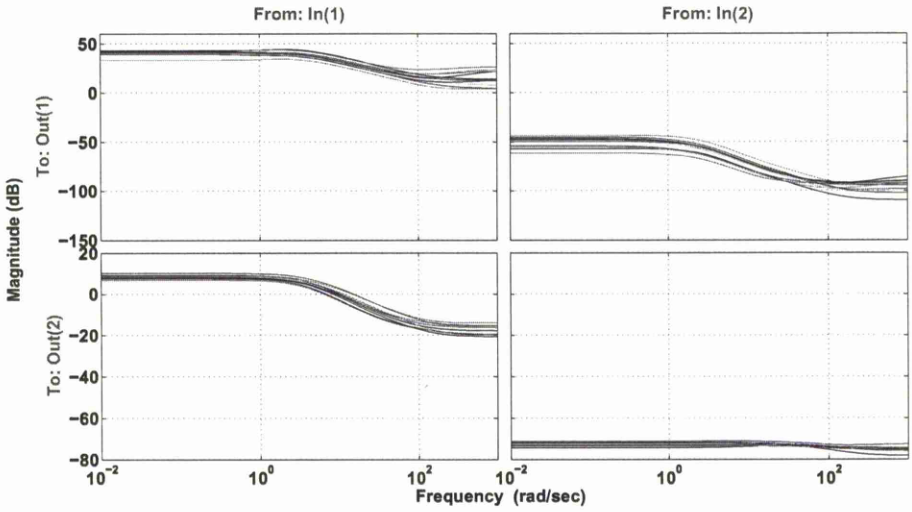

Figure 7.19: Linear models of the 2 × 2 MIMO system

of each disk represents the nominal model at the corresponding disk frequency, and the radius of the disk is the minimum unstructured uncertainty. The transfer function of the nominal model is obtained by using a LS fitting method [56]. The resulting plant transfer functions expressed as continuous domain models from the control inputs $[ABV, FPW]^T$ to the outputs $[\hat{T_E}, \hat{\lambda}]^T$ were found to be:

$$G11 = \frac{5.359s^3 - 1065s^2 + 5.214e4s + 7.229e4}{s^3 + 105.3s^2 + 538.8s + 705.8} \tag{7.11}$$

$$G12 = \frac{-1.07e - 5s^3 + 0.002271s^2 - 0.1333s + 1.317}{s^3 + 105.3s^2 + 538.8s + 705.8} \tag{7.12}$$

$$G21 = \frac{-0.162s^2 - 8.297s + 790.2}{s^2 + 84.68s + 302.4} \tag{7.13}$$

$$G22 = \frac{-0.0001673s^2 - 0.01735s - 0.06139}{s^2 + 84.68s + 302.4} \tag{7.14}$$

7.6 Feedback Controller Design

To design a feedback controller, the multivariable Frobenius H_∞ norm mixed sensitivity (MS) optimisation control design technique which was developed in previous chapters was employed. The aim of the design technique is to minimise the Frobenius H_∞ norm of the constrained transfer matrix. The mixed sensitivity problem with Frobenius norm can be written as:

$$||W_S * S||_F^2 + ||W_T * T||_F^2 < 1 \tag{7.15}$$

where $*$ represents element-by-element multiplication (Schur or Hadamard product), S and T are primary and complementary sensitivity functions respectively, and W_S and W_T are the corresponding weighting functions. To enable a fair comparison with current state of the art techniques, a feedback controller was designed by the 2 Riccati equation H_∞ method [9] as well.

Based on the Nyquist frequency associated with the data sampling, the frequency range of the engine's dynamics was considered from 0.01 to 200 rad/s. For the decoupling controller, the frequency grid points for the optimisation, were generated by the *linear fractional transform* [42]:

$$\omega = b/tan(\frac{n\pi}{N_w}) \tag{7.16}$$

where, $n = 0, 1, ..., N_w - 1$ and b is the *sample width parameter.* $N_w = 512$ frequency points with $b = 1$ were chosen to cover the main frequency range of the engine.

To satisfy the H_∞ Frobenius norm mixed sensitivity problem, the method of Helton and Merino for optimization method over analytic functions for multivariable system was used. The main optimization problem considered here is the determination of γ and $T(j\omega)$ so that:

$$\gamma = \inf_{T \in H_\infty^{2\times 2}} \sup \omega ||W_S(j\omega) * [I - T(j\omega)]||_F^2 + ||W_T(j\omega) * T(j\omega)||_F^2 \tag{7.17}$$

The aim of the optimization is to find the minimum γ and corresponding T which achieves the lowest possible bound of the weighted sensitivity. The minimum γ and optimal solution for T is obtained by solving the optimization problem of equation 7.17 using 'disk-iteration' of Helton and Merino[40]. After solution the feedback controller can be obtained from the optimal sensitivity function T. To ensure the closed loop system is internally stable, the sensitivity function T is interpolated by the method presented in previous chapter. The determination of the interpolant is based on the unstable poles and zeros of plant. In this work, the plant of equations 7.11-7.14 was required as a stable transfer function matrix, which has two RHP zeros at $s = 100$. So the interpolation method chosen for this case was:

$$T = \alpha X \tag{7.18}$$

where $\alpha \in H_\infty^{1\times 1}$ is an interpolation function, which has precisely the same RHP zeros as the unstable transmission zeros of plant. X is an unknown optimal function $\in H_\infty^{2\times 2}$. Due to the RHP zeros, the interpolant α is formulated as:

$$\alpha = \frac{(s-100)(s-100)}{(s+100)(s+100)} \tag{7.19}$$

where the poles are selected to achieve an all-pass, $0dB/1$ magnitude characteristic to aid solution convergence and control the solution relative degree. Another effect of those

poles is to ensure X has the same frequency response magnitude as T, which allows more possible solutions X in the region bounded by weighting function to be spanned during optimisation. The resulting interpolant is found to produce a convergent and internally stable solution for T.

To achieve the performance targets of air-fuel ratio and torque control, the weighting functions W_S and W_T required to be chosen carefully. Commonly, the primary sensitivity is used in the design to achieve the performance targets; the complementary sensitivity function is used to ensure the robustness. In this work, the nonlinearities of the MIMO system were modelled as multiplicative uncertainty around the nominal plant. Then the uncertainty was bounded by the constraint on the complementary sensitivity function. In this work, the performance and robustness requirements include minimal speed deviation, fast settling time, small overshoot, zero steady-state error, sensor noise, random parameter variation and engine-to-engine variation, etc.. Furthermore, the closed-loop transfer matrix function should be an invertible diagonal rational matrix function for the decoupling of the system[65]. Then, the non-diagonal elements of the closed-loop transfer matrix, (the complementary sensitivity function T) should be zero or at least very close to zero. In this work, the non-diagonal elements of T are constrained to be less than 10^{10}. Accordingly, the element-by-element weighting functions W_S and W_T in this case are:

$$W_S = \begin{bmatrix} \frac{s+5}{5s+0.001} & 10^{10} \\ 10^{10} & \frac{s+10}{2s+0.001} \end{bmatrix} \text{ and } W_T = \begin{bmatrix} \frac{5s+1}{0.01s+100} & 10^{10} \\ 10^{10} & \frac{5s+1}{0.01s+100} \end{bmatrix}$$

In the design using the Riccati method, corresponding weighting functions chosen as diagonal matrices are:

$$W_S = \begin{bmatrix} \frac{s+5}{5s+0.001} & 0 \\ 0 & \frac{s+10}{2s+0.001} \end{bmatrix} \text{ and } W_T = \begin{bmatrix} \frac{5s+1}{0.01s+100} & 0 \\ 0 & \frac{5s+1}{0.01s+100} \end{bmatrix}$$

The Frobenius norm for the mixed sensitivity form, weighted primary and complementary sensitivity functions are illustrated in the figure 7.20. Once the optimal solution (frequency response) for T was found, the stable transfer function (matrix) of T is obtained by using Caratheodory-Fejer approximation [33]. Applying the Frobenius H_∞ optimisation it is found that the optimal is given by :

$$T11 = \frac{0.000112s^5 + 1.635s^4 - 313.8s^3 + 1.326e4s^2 + 1.661e5s + 2471}{s^5 + 206.8s^4 + 1.138e4s^3 + 7.171e004s^2 + 1.655e5s + 2483} \tag{7.20}$$

$$T12 = 0 \tag{7.21}$$

$$T21 = 0 \tag{7.22}$$

$$T22 = \frac{0.0007244s^5 + 9.147s^4 - 1232s^3 - 2.282e4s^2 + 4.574e6s + 8.083e7}{s^5 + 277.9s^4 + 2.707e4s^3 + 1.086e6s^2 + 1.654e7s + 8.08e7} \tag{7.23}$$

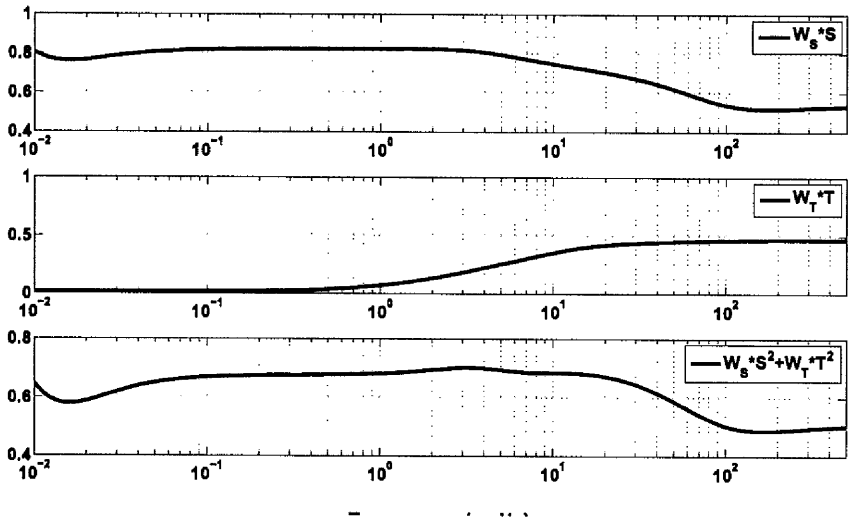

Figure 7.20: Frobenius norm of $[W_S * S]^2 + [W_T * T]^2$, $W_S * S$ and $W_T * T$ resp.

The controller K calculated from T with plant G is found from $K = T[G - TG]^{-1}$ to give:

$$K11 = \frac{\begin{array}{c} 2.086e-5s^7 + 0.3111s^6 + 36.77s^5 \\ +623s^4 + 3696s^3 + 9142s^2 + 8135s + 119.2 \end{array}}{\begin{array}{c} s^7 + 210.1s^6 + 1.271e4s^5 \\ +1.169e5s^4 + 3.567e5s^3 + 3.54e5s^2 - 3062s + 74.79 \end{array}} \tag{7.24}$$

$$K12 = \frac{\begin{array}{c} -8.63e-6s^8 - 0.1096s^7 + 6.797s^6 + 1415s^5 \\ -4.527e4s^4 - 5.162e6s^3 - 3.038e7s^2 + 9.154e8s + 3.585e9 \end{array}}{\begin{array}{c} s^8 + 373.8s^7 + 5.703e4s^6 + 4.214e6s^5 \\ +1.425e8s^4 + 1.823e9s^3 + 6.603e9s^2 + 7.285e9s - 2.155e7 \end{array}} \tag{7.25}$$

$$K21 = \frac{\begin{array}{c} -0.0202s^7 - 300.1s^6 - 1.98e4s^5 \\ +1.206e6s^4 + 2.142e7s^3 + 8.71e7s^2 + 1.043e8s + 1.534e6 \end{array}}{\begin{array}{c} s^7 + 210.1s^6 + 1.271e4s^5 \\ +1.169e5s^4 + 3.567e5s^3 + 3.54e5s^2 - 3062s + 74.79 \end{array}} \tag{7.26}$$

$$K22 = \frac{\begin{array}{c} -4.324s^8 - 5.497e4s^7 + 2.655e6s^6 + 7.461e8s^5 \\ -1.254e10s^4 - 2.772e12s^3 - 5.283e13s^2 - 2.122e14s - 1.968e14 \end{array}}{\begin{array}{c} s^8 + 373.8s^7 + 5.703e4s^6 + 4.214e6s^5 \\ +1.425e8s^4 + 1.823e9s^3 + 6.603e9s^2 + 7.285e9s - 2.155e7 \end{array}} \tag{7.27}$$

The order for the decoupling controller and Riccati controller are similar being an $8th$ order for the decoupling controller, and a $9th$ order for the Riccati controller.

7.7 Experimental Results

7.7.1 Torque Demand Response

In the experimental testing, the tracking capability for torque control and disturbance decoupling and rejection capability for λ control, was tested by inputs which are representative of real-world driving conditions, were tested first. To test the tracking capability for torque controller and interaction of torque and λ, a series of step demands were applied. Due to limits on the maximum brake torque loaded from the dynamometer, torque demands of $+20N \cdot m$ and $-10N \cdot m$ were applied to the engine. The engine was operated at a nominal speed of $1500rpm$ and the λ demand fixed at $\lambda = 1$. The measured torque production for positive step demands using the optimal decouplling controller and Riccati controller are displayed in figures 7.21 and 7.22. As the feedback controllers did not feed back the measured value of torque, the accuracy of the torque control is evaluated and compared by plotting against the resulting but only the output of torque from the estimator estimated torque value in figures 7.21 and 7.22. The corresponding control action of ABV (duty cycle) and fuel injector (FPW, unit:μ second) are plotted in figures 7.23 and 7.24. Form figures 7.21 and 7.22, the torque can usually achieve 90% of the demand within in 3 seconds both for optimal decoupling and Riccati controllers, the overshoot is about 4 $N \cdot m$. Although the actual measured torque production is not fed back to the controller directly, the feedback controller still can regulate the actual torque as the demand required. The steady state error is within $1N \cdot m$ at torque=$70N \cdot m$. That indicates the accuracy of the 2 layer feedforward network for torque estimator is high enough for torque control. Durning the same period, a small deviation of approximately ± 0.05 is observed in λ signal for the optimal decoupling controller, while approximately +0.04/-0.07 is observed for the Riccati controller. Moreover, the λ settles to demand level in 2.5 second both for the optimal decoupling controller and Riccati controller

In the figures 7.23 and 7.24 the control behaviours for ABV and fuel injector are compared. For a $+20N \cdot m$ torque step, the ABV and FPW responds to the step demand quickly. The actuations of ABV for decoupling optimal controller and Riccati controller are basically similar. Both of them achieve the maximum positions of ABV about 1 second. For decoupling controller, FPW signal achieves to the maximum injection timing in about 1.5 second. While for Riccati controller, FPW signal achieves to the maximum injection timing in 1.2 second. Compare the two controllers, the response of decoupling optimal controller is slightly slower than Riccati controller, but the control behaviour of the decoupling controller is more 'smooth'. In experimental results this difference might result in the out come that the maximum deviation of air-fuel ratio controlled by decoupling controller is less than that of the Riccati controller.

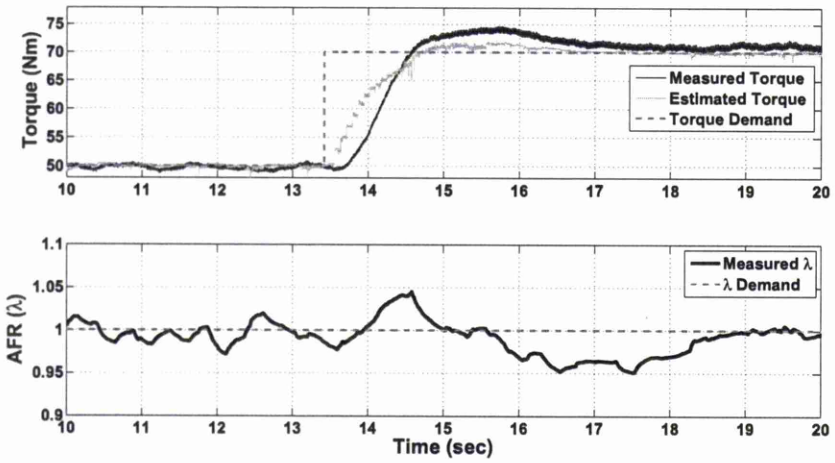

Figure 7.21: Decoupling controller responses to $+20N \cdot m$ torque step demand

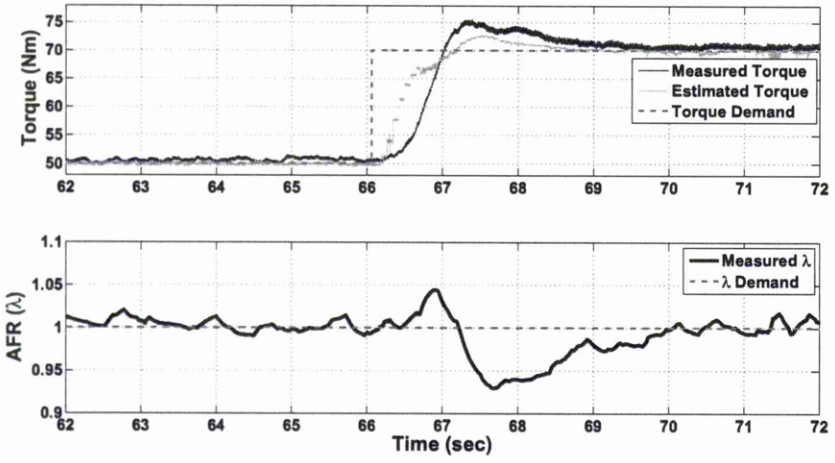

Figure 7.22: Riccati controller system responses to $+20N \cdot m$ torque step demand

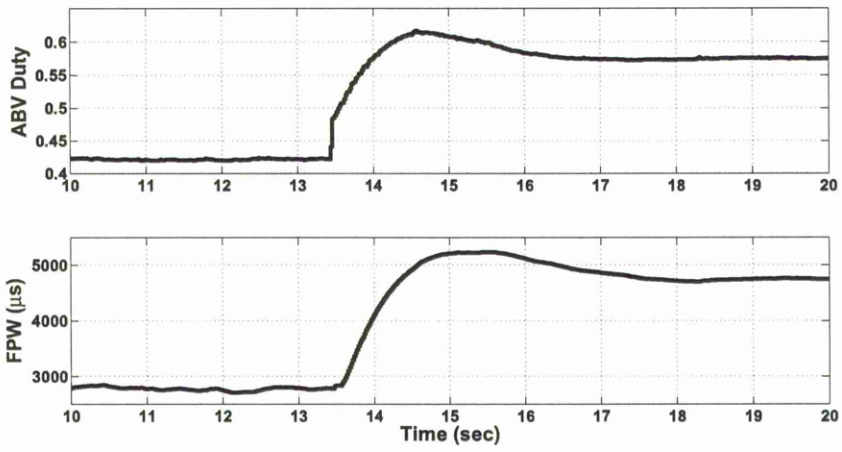

Figure 7.23: ABV duty and FPW response of decoupling controller to $+20N \cdot m$ torque step demands

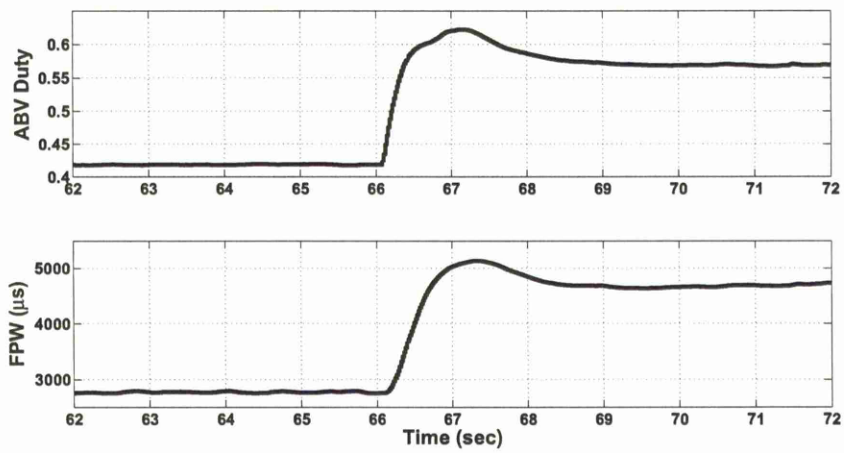

Figure 7.24: ABV duty and FPW response of Riccati controller to $+20N \cdot m$ torque step demands

The responses of -10$N \cdot m$ torque step demand are shown in figures 7.25 and 7.26. The corresponding variation of ABV duty and FPW are plotting in figures 7.27 and 7.28. In this test, the torque settles to the demand similarly for both two controllers. The measured torque can achieve to 90% in about 3 second, and the overshoot is approximately 2 to $3N \cdot m$. Due to the transient variation of torque, λ is perturbed by approximately -0.04/+0.06 for Riccati controller. However, much smaller deviation of λ (± 0.02) is observed for the optimal decoupling controller. The advantage of decoupling controller is clearly seen in the negative torque step demand test.

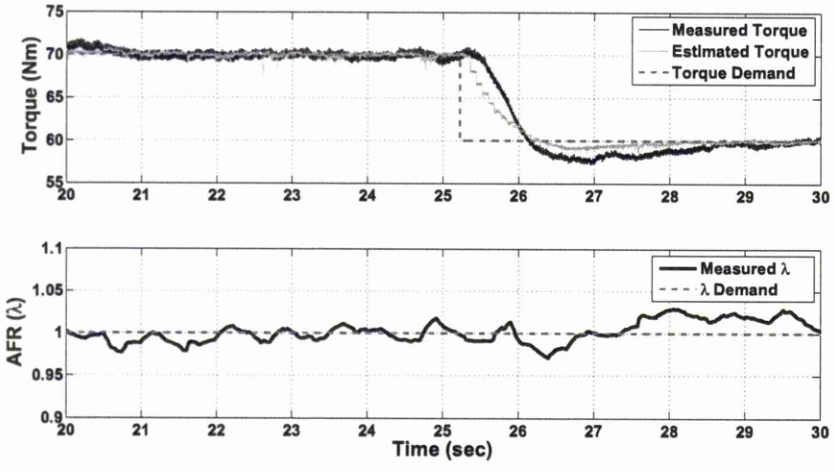

Figure 7.25: Decoupling controller responses to -10$N \cdot m$ torque step demands

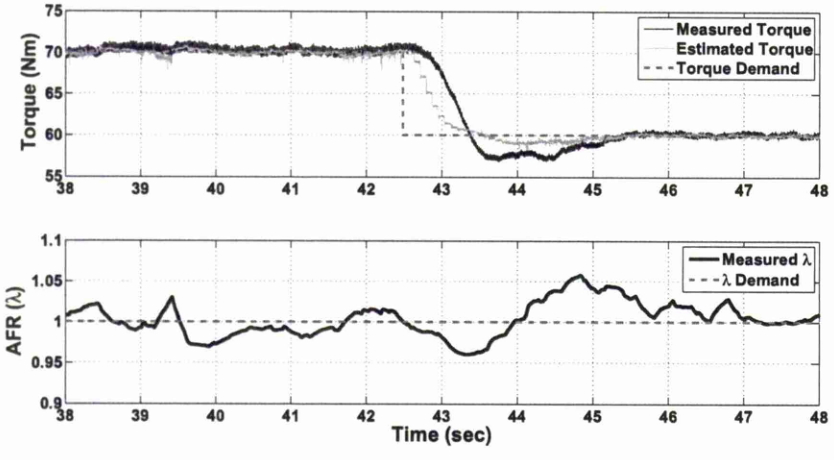

Figure 7.26: Riccati controller responses to -10$N \cdot m$ torque step demands

In figures 7.27 and 7.28 the control behaviours for ABV and fuel injector are compared. For -10$N \cdot m$ torque step, the actuations of ABV for decoupling optimal controller and Riccati controller are similar. Both of them achieve to the minimum positions of ABV in 1 second. The actuations of fuel injector for the two controllers are also similar,

they achieve to the minimum positions of FPW slightly larger than 1 second.

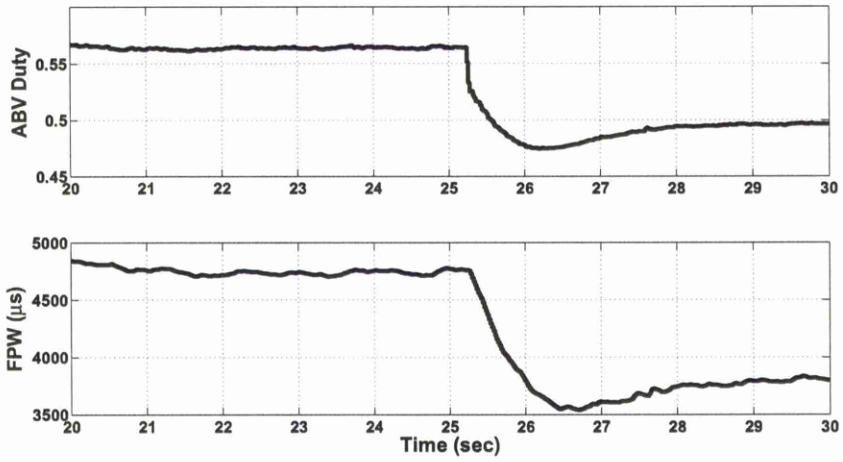

Figure 7.27: ABV duty and FPW of decoupling controller response to -10$N \cdot m$ torque step demands

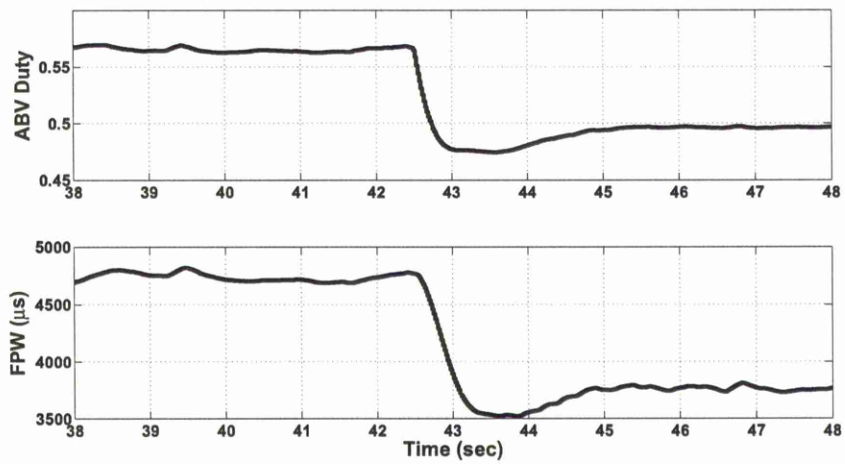

Figure 7.28: ABV duty and FPW of Riccati controller response to -10$N \cdot m$ torque step demands

7.7.2 λ Demand Response

Similar step response tests were carried out on the FPW to λ path to test the tracking capability, which is useful for some particular occasions of IC engine control, such as in lean-burn technology. During these tests the torque demand was set to $60N \cdot m$, and the speed command was set to the constant nominal level of 1500 rpm. The responses to steps of ± 0.1 in λ demand corresponding to lean and rich demand are displayed in figures 7.29 and 7.29. The tracking performance for both two controller are quite

similar. The λ responses to ± 0.1 steps are each very quick. The λ can settle to the demand in 1 second, with almost no overshoot. The interaction with torque due to the step in λ demand is limited to a deviation of just less then $\pm 2N \cdot m$ with each controller. The steady-state error for torque estimator at torque=$60N \cdot m$ is about $0.5N \cdot m$. The change to lean and rich of λ also causes a very small stead-state error about -$0.5N \cdot m$ when λ is lean and $+1.5N \cdot m$ when λ is rich.

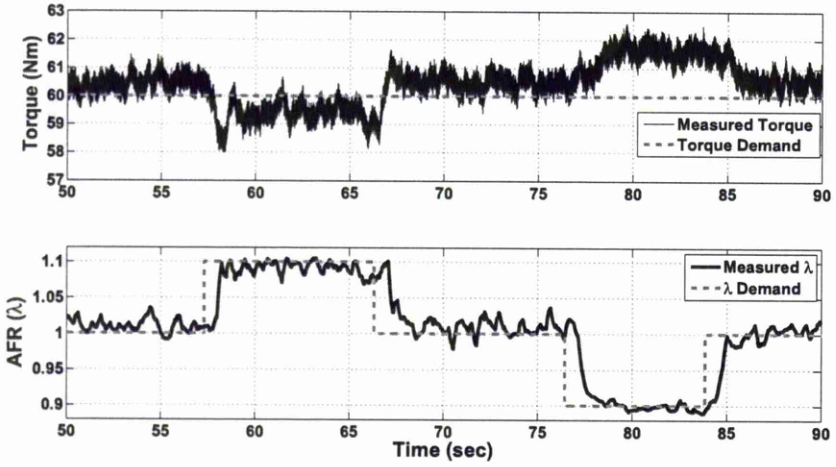

Figure 7.29: Decoupling controller responses to ± 0.1 λ step demands

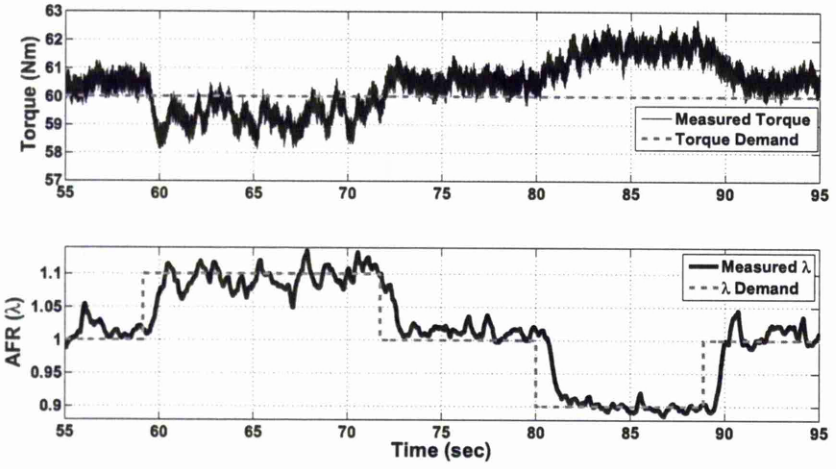

Figure 7.30: Riccati controller responses to ± 0.1 λ step demands

In figures 7.31 and 7.32 the control behaviours for ABV and fuel injector responding to λ step are compared. Although the responses for torque and λ of two controllers are similar, the change of control behaviours of decoupling controller is more quicker than Riccati controller in these figures.

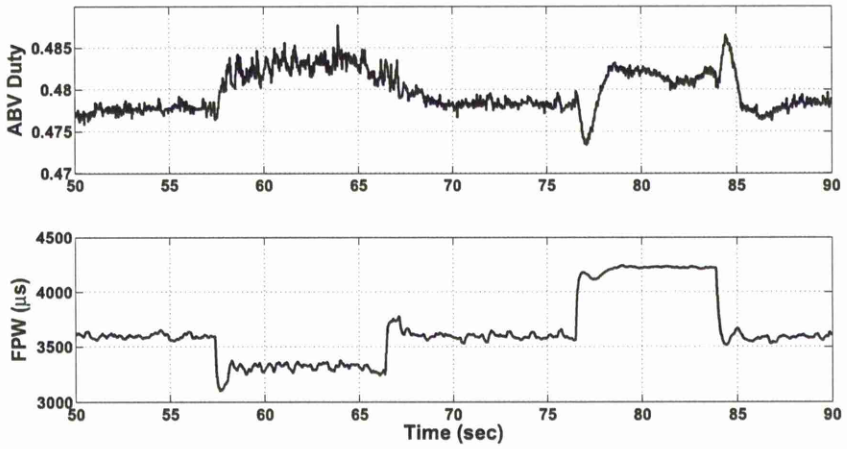

Figure 7.31: ABV duty and FPW of decoupling controller response to $\pm 0.1\lambda$ step demands

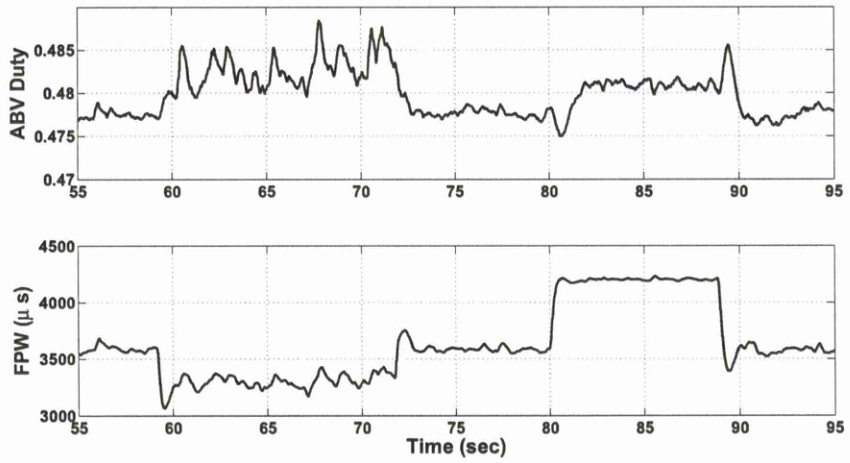

Figure 7.32: ABV duty and FPW of Riccati controller response to $\pm 0.1\lambda$ step demands

7.7.3 Response to Load (Torque) Disturbance

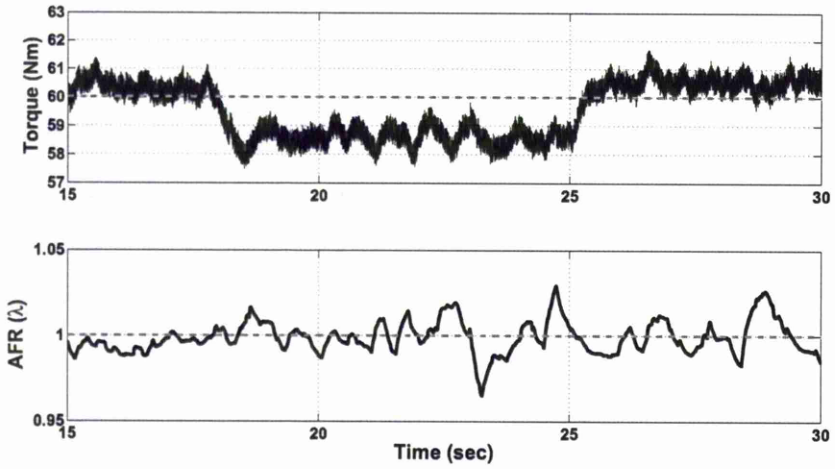

Figure 7.33: Disturbance rejection test for the decoupling controller

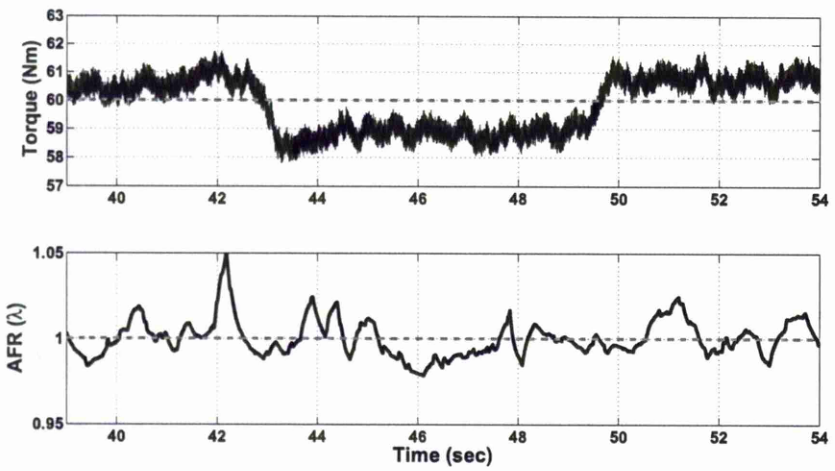

Figure 7.34: Disturbance rejection test for the Riccati controller

Both torque and λ path are required to reject the unpredictable disturbances such as road load changes, power-assisted steering demand and uncontrolled electrical loads. In the test, a resistance which is equivalent to a 400W power electrical device is loaded to the alternator. The responses of torque and λ for both decoupling and Riccati controller are shown in figures 7.33 and 7.34. The results show that such electrical load causes -1.5$N \cdot m$ steady state error at torque=60$N \cdot m$. The maximum deviation of corresponding λ response is about 0.03 for decoupling controller and 0.05 for Riccati controller. The settling time of λ for both two controllers is about 0.5 second.

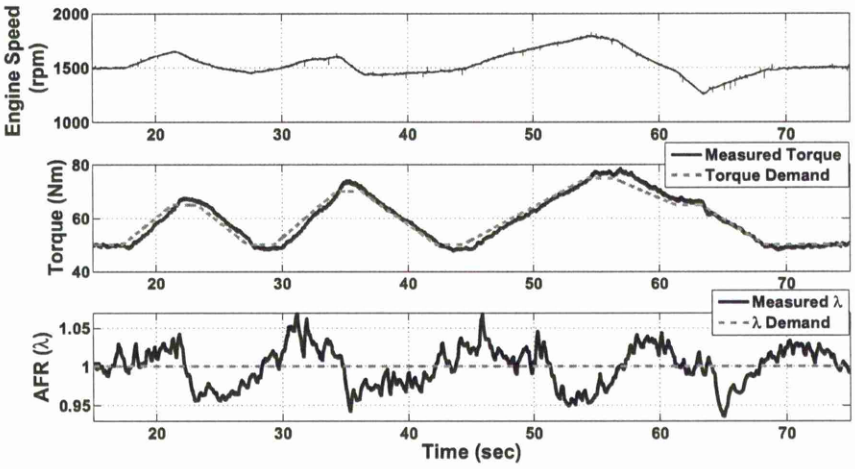

Figure 7.35: Test for decoupling controller in random driving cycle

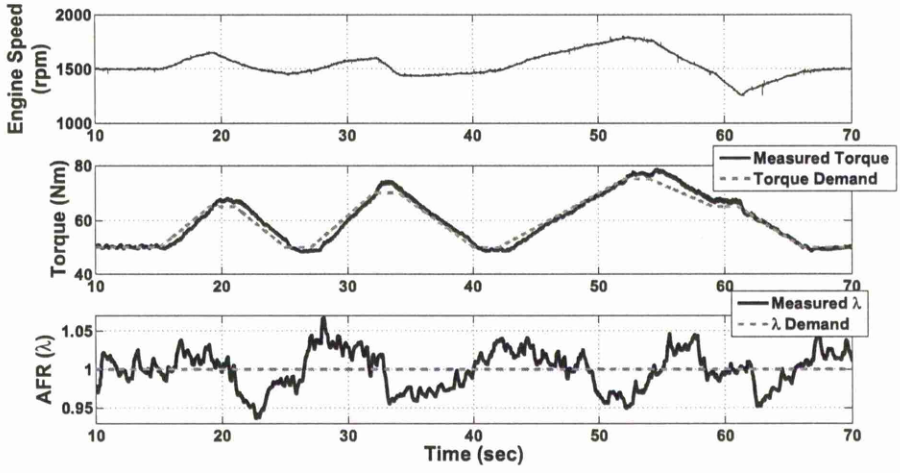

Figure 7.36: Test for Riccati controller in random driving cycle

7.7.4 Random Driving Cycle Test

For comparative purposes a simple random driving cycle was applied to the engine. During test the torque demand and engine speed were set to vary simultaneously. To simulate a particular real driving condition, the engine speed was changed according the torque demand in the driving cycle by means of random input to the dyno controller in speed mode. The situation is similar to acceleration or deceleration of the vehicle. Also in some period of the cycle, the torque demand was increased while the engine speed was decreased, which is similar to the case when vehicle starts to hill climb. In this driving cycle, the torque was changed from approximately 50 to $75N \cdot m$. At the same time, the engine speed was varied between 1250 to 1800rpm. The test results for the Riccati and decoupling controller are shown in 7.36. From those figures, the

deviation of λ can remain within ± 0.03 (3%) for both controllers over than 90% period of the driving cycle. The torque also responds to the demand quickly. That indicates the performance and robustness are similar for two controllers.

7.7.5 Robustness Test

For the robustness testing an additional loop gain and loop delay are added to the controller until the variations of λ and torque response are over than ± 0.4 and $\pm 10 N \cdot m$. The test result shows the Riccati controller is robust to $12dB$ and 20 time delays. And the decoupling controller to 11dB and 20 time delays. But starting from 17 time delays, the variations of torque and λ response at steady state of decoupling controller are quite more than the Riccati controller. The robustness for the two controllers are basically similar.

7.8 Conclusions

A novel nonlinear NN SI engine model for use as an on-line λ and torque estimator was presented. The significant time delays in the downstream UEGO sensor and short time delays in torque path were both removed from the tracking control path by means of the estimators. A fast feedback controller was then designed by feeding back the predicted λ and torque produced by the estimator. A nonlinear black-box Nueral Network identification and training methodology for the λ and torque estimators were developed. FPW, MAP and engine speed were used as additional inputs for λ estimator; λ, MAP and engine speed were used as additional inputs for the torque estimator. The new method is a relatively quick and economical way to control air-fuel ratio and engine torque compared to traditional approaches of feedforward air-fuel ratio control using air estimation presented in [34] and feedforward torque controller using static mapping functions presented in [47].

The MIMO Frobenius norm, Hadamard weighted, decoupling controller design methods developed in previous chapters using the H_∞ analytical optimisation algorithms of Helton and Merino [40] were proposed for application to the estimated λ and torque engine control problem. The controller design methods were applied experimentally together with the torque and λ estimators. The interpolation scheme used with the H_∞ Frobenius norm mixed-sensitivity optimisation approach guaranteed internal stability.

The experimental results were compared with a maximum singular value norm H_∞ mixed-sensitivity Riccati design method. Experimental test showed the performances and robustness of the decoupling and Riccati controllers are similar. The decoupling

optimal controller is, as required, more decoupled than the common Riccati controller, especially in -10$N \cdot m$ torque step test. The incomplete decoupling achieved experimentally is believed to be due to the nonlinearity in the estimator controlled system. Further use of feedforward linearisation filters may allow the experimental decoupling to approach the effectively complete decoupling of the simulated system. Further, the feedforward Neural Network torque estimator also show higher accuracy in experimental test than the static torque model.

Due to the limitation of ABV and the dynamometer, the engine cannot run at a high speed range. The implementation and the further experimental testing the decoupling-estimator based control strategy using an electronic throttle controlled (ETC) engine over the full speed and low range is required in future to justify industrial application to advanced engine technology. Experimental application to ETC technology engines may be expected to give improved results, because the electronic throttle valve responds more quickly and introduces more air flow to the cylinder than the ABV, and so the regulation of air intake is more aggressive.

Chapter 8

Conclusions and Suggestions for Future Work

8.1 Conclusions

In this thesis a nonparametric H_∞ optimal controller design technique for powertrain control application is presented. The nonparametric approach allows the control designer to work directly with experimentally obtained data and has the potential to give more accurate results by avoiding the approximation necessary with parameter fitting in parametric method. The methodology involves both single-input-single-output (SISO) and multiple-input-multiple-output (MIMO) controller design approaches which are mainly applied to the idle speed control problem (SISO case) and the air-fuel ratio/torque control problem (MIMO case).

The following conclusions and contributions, as detailed in previous chapters, can be drawn from this work:

- In Chapter 4, the parametric and nonparametric SISO controller design approaches based on H_∞ optimization over analytic functions were presented. The interpolation conditions for internal stability and the corresponding interpolation method for SISO case were summarised. For parametric design, it was shown that the closed loop system can be ensured internally stable by parametrising the sensitivity function T with that interpolation function during optimisation. For nonparametric design, it was shown that the closed loop can be ensured internally stable by stabilizing the control effort sensitivity function U when the plant is stable. An H_∞ optimal controller design approach based on the analytic function optimisation method was presented in this chapter. Through solving either the H_∞ optimisation problem for the disk problem and the mixed sensitivity problem by analytical optimisation, the optimal solution for the complementary

sensitivity function T or control effort sensitivity function U can be found. In the previously published parametric case, the optimal solution for T is coupled with Newton interpolation functions which ensure that the closed loop system internally stable. Thus an internally stable controller can be calculated from the resulting T. A novel nonparametric design approach is developed here based on optimisation of the control effort sensitivity function U.

The parametric design approach was demonstrated to allow successful design of the controller for a peak pressure position (PPP) automotive engine controller. The simulation results of a single sensitivity - disk method and mixed sensitivity method are presented and compared. The nonparametric design approach was demonstrated to design the controller for a nonparametric driveline model using the mixed sensitivity optimisation method. The simulation results for both parametric and nonparametric design show the designs are successful, the novel developed nonparametric design approach is validated.

- In Chapter 5 a systematic rapid prototyping procedure for idle speed control problem using both a parametric and nonparametric H_∞ optimal controller design approach was detailed. The system transmission from air bleed valve (ABV) to engine speed was identified by both the parametric autoregressive moving average model with exogenous(ARMAX) approach and by the nonparametric Blackman-Tukey method. The nonparametric models were established based on different window sizes by short time Fourier transformation (STFT). The different parametric and nonparametric models were displayed and compared.

  Then the parametric and nonparametric controllers were then designed by the disk method and the mixed sensitivity optimisation method. The simulation results of the different controllers were compared. Experimental testing showed that the transient performance of a nonparametric controller is similar to the parametric controller when using the H_∞ optimisation disk method and mixed sensitivity optimisation methods. From the experimental results, however the nonparametric controller designed using a nominal model based on the centre of the frequency response from different window sizes was found to give a significantly superior robustness in time delay.

  The nonparametric idle speed controller was implemented in a CAN-bus based control system based on dsPIC33F microcontroller using a new fast prototyping method based on SIMULINK's graphical model. The controller can thus be implemented as a distributed control system, such as might be employed in a hybrid vehicle. The experimental results show the implementation work including software/hardware configuration and programming was successful. The nonparametric approach with its increased delay margin was shown to provide a suitable

controller for such CAN-bus networks.

- In the Chapter 6 a novel MIMO H_∞ Frobenius norm, Hadamard weighted, decoupling controller design method was developed using the analytic function optimisation method. From the interpolation constraints for internal stability, a parametrisation method of interpolation functions for simple poles/zeros was developed. A new MIMO controller design approach using this interpolation method and minimising the H_∞ Frobenius norm of the mixed sensitivity problem was thus proposed. Through using the Hadamard (element-by-element) multiplication of the sensitivity functions by the weighting functions, the resulting method was thus found applicable to design a novel decoupling controller using this MIMO H_∞ Frobenius norm optimisation method. The parametric and nonparametric design approaches to the H_∞ Frobenius norm optimisation were developed and presented respectively.

  A nonlinear black-box algebraic Nonlinear Auto Regressive eXogenous (NARX) identification methodology for a novel air-fuel ratio (AFR) estimator with time delays removed was presented. The estimator using MAP and engine speed feedback to allow AFR estimation from fuel pulse width (FPW) input data. The feedback control scheme could be applied to the AFR path to reduce the significant transport time delay of the UEGO sensor. The new method was found to be a relatively quick and economical way to develop a control system to control AFR in automotive engine compared to traditional approaches of a feedforward controller using air estimation of mappings.

  The H_∞ Frobenius norm decoupling design method was applied experimentally to the estimated AFR and engine torque control problem. The simulation and experimental results were compared with a maximum singular value H_∞ norm mixed sensitivity design. In the simulation, the decoupling controller was seen to effectively achieve exact decoupling. The performances and robustness of the decoupling and Riccati controllers were found to be similar in the experimental testing. While the test results showed the decoupling controller is, as required, more decoupled than the Riccati controller, but does not achieve the full decoupling found in simulation.

- In Chapter 7 an alternative novel calibration method for multivariable AFR and engine torque control system was presented. Novel neural network (NN) estimators for both AFR and torque are developed with time delays removed. A NARX network using FPW, MAP, engine speed and measured AFR was established for the AFR estimator. A feedforward multi-layer perceptron(MLP) network using AFR, MAP and engine speed was established for the torque estimator. A 2×2 linear system from air bleed valve (ABV) duty and FPW to estimated AFR

and estimated engine torque was thereby identified. The feedback control was obtained using H_∞ Frobenius norm optimal decoupling design approach.

Detailed experimental testing was carried out to assess tracking capability, disturbance rejection capability, random driving cycle performance and robustness. The experimental results were compared with a maximum singular value H_∞ norm mixed sensitivity Riccati design method. The feedback controllers based on both methods should be more quickly and systematically implemented than the traditional feedforward method. The decoupling controller was also, as required, more decoupled than the Riccati controller, such as in torque step test. While in the random driving cycle test, two controllers showed the similar performance.

8.2 Recommendations for Future Work

- In future, more applications of the nonparametric design method can be theoretically and experimentally investigated including for both SISO and MIMO problems to explore their further possible advantages over parametric methods.
- The nonparametric identification method can be investigated for the MIMO system, such that the nonparametric MIMO controller design approach can be employed practically to the powertrain control applications.
- The current interpolation function for MIMO conroller design case only satisfies the sufficient conditions for internal stability. When using the analytical optimisation method, the resulting solution is thus not necessarily a global optimal solution. New advanced interpolation can be investigated to find the possible improved results.
- Further use of feedforward linearisation filters for current engine system coupled with estimators may allow the dynamical calibration of engines across a much larger operational range. This calibration method could then be apply to the engines equipped with an electronic throttle valve to allow improved control of drive-by-wire vehicles.
- The rapid prototyping system implemented here for dsPIC based CAN-bus network SISO control systems could be established for MIMO control system. Such a system would allow a low cost rapid prototyping system which would be particularly useful for hybrid vehicle development.

References

[1] A. Abass, S. Zhao, and A.T. Shenton. Nonparametric driveline identification and control. In *First International Conference on Intelligent Systems, Modelling and Simulation, ISMS 2010, Liverpool*, January 2010.

[2] G. Balas, A. Packard R. Chiang, and M. Safonov. *MATLAB Robust Control Toolbox 3 User's Guide.* The MathWorks, Inc., 2010.

[3] V. Besson. *Parameter Space Robust Control for S.I. Engine Idle Speed.* PhD thesis, University of Liverpool, 1998. Ph.D.thesis.

[4] S.A. Billings and Q.M. Zhu. A structure detection algorithm for nonlinear dynamic rational models. *International Journal of Control*, 59(6):1439–1463, 1994.

[5] H.W. Bode. *Network Analysis and Feedback Amplifier Design.* Van Nostrand New York, 1945.

[6] Bosch. *Gasoline Engine Management.* SAE International, 1999.

[7] S.D. Carroll. *Control of S.I. Engines Using In-Cylinder Pressure.* PhD thesis, University of Liverpool, 2003. Ph.D. thesis.

[8] J. Chauvin, G. Corde, P. Moulin, M. Castagne, N. Petit, and P. Rouchon. Real-time combustion torque estimation on a diesel engine test bench using time-varying kalman filtering. In *43rd IEEE conference on decision and control, Atlantis, Paradise Island, Bahamas*, 2004.

[9] R. Chiang and M. Safonov. Robust control toolbox-user's guide. *The MathsWorks, Inc.*, 1992.

[10] Microchip Corporation. CodeGuard Security: Protecting Intellectual Property in Collaborative System Design. Technical report, 2006.

[11] Microchip Corporation. Direct Memory Access (DMA) - dsPIC33F FRM. Technical report, 2008.

[12] Microchip Corporation. CodeGuard Security - dsPIC33F FRM. Technical report, 2009.

[13] Microchip Corporation. dsPIC33FJXXXGPX06/X08/X10 Data Sheet. Technical report, 2009.

[14] Microchip Corporation. Enhanced Controller Area Network (ECAN) - dsPIC33F FRM. Technical report, 2009.

[15] Microchip Corporation. MPLAB IDE User's Guide. Technical report, 2009.

[16] Microchip Corporation. Oscillator - dsPIC33F FRM. Technical report, 2009.

[17] Microchip Corporation. Analog-to-Digital Converter (ADC)-dsPIC33F/PIC24H FRM. Technical report, 2010.

[18] Microchip Corporation. Watchdog Timer and Power-Saving Modes-dsPIC33F/PIC24H FRM. Technical report, 2010.

[19] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[20] P.B. Dickinson. *Robust Low-order Control Techniques with Powertrain Applications.* PhD thesis, University of Liverpool, 2007. Ph.D.thesis.

[21] F. Van Diggelen and K. Glover. A hadamard weighted loop shaping design procedure for robust decoupling. *Automatica*, 30(5):831–845, 1994.

[22] G. Dreyfus. *Neural Networks: Methodology and Applications.* Springer, 2005.

[23] A. Dutka, H. Javaherian, and M.J. Grimble. Model-based nonlinear multivariable engine control. In *Proceedings of the American Control Conference, New York, USA*, July 2007.

[24] W.R. Evans. *Control System Dynamics.* McGraw-Hill, 1954.

[25] S. Gaikwad and D. Rivera. Systematic techniques for determining modelling requirements for siso and mimo feedback control. *Journal of Process Control*, 5(4):213–224, 1995.

[26] S. Gaikwad and D. Rivera. Control-relevant input signal design for multivariable system identification: Application to high-purity distillation. In *In Proceedings of the 13th IFAC World congress*, pages 349–354, 1996.

[27] J. Gerhardt, N. Benninger, and W. Hess. Torque based system structure of an electronic engine management system (ME7) as a new base for drivetrain systems. In *FISITA Congress, Paris, France*, volume paper F98T624, July 1998.

[28] J. Gerhardt, H. Höninger, and H. Bischof. A new approach to functional and software structure for engine management systems - BOSCH ME7. *SAE Technical paper*, 49, 1998.

[29] J. Gharoo and Microchip Corporation. AN1249 ECAN Operation with DMA on dsPIC33F and PIC24H Devices. Technical report, 2009.

[30] Robert Bosch GmbH. CAN Specification. Technical report, Postfach 30 02 40, D-70442 Stuttgart, 1991.

[31] G.H. Golub and C.F. Van Loan. *Matrix Computations.* JHU Press, 1996.

[32] G.C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis, Mathematics in Science and Engineering, Volume 136.* Academic Press Inc., 1977.

[33] H. Gutknecht and L.N. Trefethen. Real and complex chebyshev approximation on the unit disk and interval. *Bulletin of the American Mathematical Society*, 8(3), 1983.

[34] L. Guzzella and C. H. Onder. *Introduction to modelling and control of internal combustion engine systems.* Berlin: Springer, 2004.

[35] R. Haber and L. Keviczky. *Nonlinear System Identification: Input-Output Modeling Approach.* Kluwer Academic Publishers, 1999.

[36] R. Haber and H. Unbehauen. Structure identification of nonlinear dynamic systems –a survey on input/output approaches. *Automatica*, 26(4):651–677, 1990.

[37] S. Haykin. *Neural Networks, A Comprehensive Foundation.* Prentice-Hall, 1999.

[38] J. W. Helton and O. Merino. Conditions for optimality over $\mathcal{H}_\infty$. *SIAM J. Control and Optimization*, 31(6):1379–1415, 1993.

[39] J.W. Helton and O. Merino. Novel approach to accelerating newton's method for sup-norm optimization arising in H_∞ control. *Journal of Optimization Theory and Applications*, 78(3):553–578, 1993.

[40] J.W. Helton and O. Merino. *Classical Control Using $\mathcal{H}_\infty$ Methods.* SIAM - Advances in Design and Control, 1998.

[41] J.W. Helton and D. Schwartz. A primer on the $\mathcal{H}_\infty$ disk method in frequency domain design control. Technical report, Lab. for Math. and Statistics, Univ. California, San Diego, 1985.

[42] J.W. Helton and D. Schwartz. Best uniform approximation of vector valued functions by $\mathcal{H}_\infty$ functions. Technical report, Lab. for Math. and Statistics, Univ. California, San Diego, 1990.

[43] E. Hendricks. Isothermal vs. adiabatic mean value si engine models. In *IFAC workshop on advances in automotive control, Karlsruhe, Germany*, 2001.

[44] E. Hendricks and S.C. Sorenson. Mean value modeling of spark ignition engines. *SAE Technical papers*, 1990.

[45] J. Heywood. *Internal Combustion Engine Fundamentals.* McGraw-Hill, 1988.

[46] D. Hrovat and J.Sun. Models and control methodologies for ic engine idle speed control design. *Control Engineering Practice*, 11, June 2002.

[47] G. Ingram, M. Franchek, and V. Balakrishnan. Spark ignition engine torque management. In *Proceedings of American Control Conference, Denver, USA*, June 2003.

[48] B. A. Francis J. C. Doyle and A. Tannenbaum. *A Course in $\mathcal{H}_\infty$ Control Theory.* Springer-Verlag, 1987.

[49] B. A. Francis J. C. Doyle and A. Tannenbaum. *Feedback Control Theroy.* Macmillan Pub. Co., 1992.

[50] R. Jurgen. *Automotive Electronic Handbook.* McGraw-Hill, 1995.

[51] J.L. Kainz and J.C. Smith. Individual cylinder fuel control with a switching oxygen sensor. *SAE Technical Paper*, 1999-01-0546, 1999.

[52] D. Khiar, J. Lauber, T. Floquet, and T. M. Guerra. An observer design for the instantaneous torque estimation of an ic engine. In *IEEE VTS-vehicle power and propulsion (VPP) conference, Illinois Institute of Technology, Chicago, IL, USA.*, 2005.

[53] D. Khiara, J. Laubera, T. Floquetb, G. Colinc, T.M. Guerraa, and Y. Chamaillard. Robust takagi-sugeno fuzzy control of a spark ignition engine. *Control Engineering Practice*, 15:1446–1456, 2007.

[54] U. Kiencke and L. Nielsen. *Automotive Control Systems.* Springer, 2005.

[55] Satish Kumar. *Neural Networks: A Classroom Approach.* Tata McGraw-Hill, 2004.

[56] E.C. Levi. Complex-curve fitting. *IRE Trans. on Automatic Control*, AC-4:37–44, 1959.

[57] Lennart Ljung. *System Identification.* New Jersey: Prentice Hall PTR, 1999.

[58] Lennart Ljung. *System Identification: Theory for The User-2nd edition.* Prentice-Hall, 1999.

[59] Lennart Ljung. *System Identification Toolbox 7.* Natick: The MathWorks, Inc., 2007.

[60] J.M. Madapura and Microchip Corporation. AN1152 Achieving Higher ADC Resolution Using Oversampling. Technical report, 2008.

[61] Maolis and A. Christodoulou. Decoupling in the design and synthesis of singular systems. *Automatica*, 22(2):245–249, 1986.

[62] The MathWorks. *MATLAB Signal Processing Toolbox 6 User's Guide.* The MathWorks, Inc., 2007.

[63] D. McFarlane and K. Glover. A loop shaping design procedure using $\mathcal{H}_\infty$ synthesis. *IEEE Transaction on Automatic Control*, 37(6):759–769, 1992.

[64] J. Miles and M. Shevlin. *Applying Regression and Correlation: A Guide for Students and Researchers.* Sage Publications Ltd., 2000.

[65] B.S. Morgan. The synthesis of linear multivariable systems by state variable feedback. *IEEE Transactions on Automatic Control*, AC-9:405–411, 1964.

[66] K.S. Narendra and S. Mukhopadhyay. Adaptive control using neural networks and approximate models. *IEEE Transactions on Neural Networks*, 8:475–485, 1997.

[67] Z. Nehari. On bounded bilinear forms. *Ann. of Math.*, 65(2):153–162, 1957.

[68] O. Nelles. *Nonlinear System Identification.* Springer, 2001.

[69] H. Nyquist. Regeneration theory. *Bell System Technical Journal*, 11, 1932.

[70] F.E. Coats R.D. Fruechte and C.H. folkerts. Idle speed control for automobiles. In *IEEE 17th Inter-Society Energy Conservation Engineering Conferences*, 1983.

[71] S.S. Ricado and M. Sznaier. *Robust System: Theory and Applications.* J. Willey and Sons Inc., 1998.

[72] N. Rivara. *IC Engine Control by Ionization Current Sensing.* PhD thesis, University of Liverpool, 2009. Ph.D.thesis.

[73] G. Rizzoni and F. T. Connolly. Estimate of ic engine torque from measurement of crankshaft angular position. *SAE Technical paper*, No. 932410, 1993.

[74] E. Feron S. Boyd, L. El Ghaoui and V. Balakrishnan. *Linear Matrix Inequalities In System and Control Theory.* SIAM, 1994.

[75] S.M. Sato and P.V. Lopresti. New results in multivariable decoupling theory. *Automatica*, 7, 1971.

[76] T. Söderstrom and P. Stoica. *System Identification.* Prentice-Hall, 1989.

[77] A.T. Shenton. Automotive applications of control. In *EPSRC/UKACC Postgraduate Workshop-UKAC International Control Conference*, 28-29 Aug. 2006.

[78] A.T. Shenton and V. Besson. Robust performance S.I. engine idle-speed control by a mixed sensitivity parameter-space method.

[79] J. Sjoberg and et al. Nonlinear black-box modelling in system identification: A unified overview. *Automatica*, 31(12):1691–1724, 1995.

[80] S. Skogestad and I. Postletwaite. *Multivariable Feedback Control: Analysis and Design.* J. Wiley, 2005.

[81] A. G. Stefanopoulou, J. W. Grizzle, and J. S. Freudenberg. Engine air-fuel ratio and torque control using secondary throttles. In *In Proceedings of conference on decision and control, Orlando, USA*, 1994.

[82] H.J. Tang, K.C. Tan, and Y. Zhang. *Neural Networks: Computational Models and Applications.* Springer, 2007.

[83] J.H. Taylor. Modelling & simulation of dynamic system - a tutorial proc. In *IMACS/IFAC Fourth International Symposium on Mathematical Modelling and Simulation in Agricultural and Bio-Industries, Haifa, Isreal*, 2001.

[84] L.N. Trefethen. Matlab programs for cf approximation. *Approximation theory V, Academic Press, Inc.*, 1986.

[85] G. Triantos. *NARMAX Modelling and Control With Powertrain Applications.* PhD thesis, University of Liverpool, 2006. Ph.D.thesis.

[86] G. Triantos and A.T. Shenton. NARMAX structure selection for powertrain control. In *IFAC Symposium in: Advances in Automotive Control AAC04, Salerno*, 2004.

[87] D.Y. Wang and E. Detwiler. Electrode dynamic study of exhaust gas oxygen sensors. *Sensors and Actuators B*, 99(2-3):571–578, 2004.

[88] D.Y. Wang and E. Detwiler. Dynamic study of porous coating layers of exhaust oxygen sensors. *Sensors and Actuators B*, 106(1):229–233, 2005.

[89] D.Y. Wang and E. Detwiler. Exhaust oxygen sensor dynamic study. *Sensors and Actuators B*, 120(1):200–206, 2006.

[90] D.Y. Wang, E. Detwiler, and S. Nelson. Dynamic study of louvre shield and exhaust pipe for exhaust gas oxygen sensors. Technical report, 2001.

[91] C.P. Ward. *J-Spectral Factorisation for Automotive Powertrain Controls.* PhD thesis, University of Liverpool, 2009. Ph.D. thesis.

[92] M.S. Wei, Q. Wang, and X.H. Cheng. Some new results for system decoupling and pole assignment problems. *Automatica*, doi:10.1016/j.automatica.2010.02.017, 2010.

[93] P.E. Wellstead. *Introduction to Physical System Modelling.* Academic Press, London, 1979.

[94] P. Higelin Y. Chamaillard and A. Charlet. A simple method for robust control design, application on a non-linear and delayed system: Engine torque control. *Control Engineering Practice*, 12(4):417–429, 2004.

[95] Y. Yildiz, A. Annaswamy, D. Yanakiev, and I. Kolmanovsky. Adaptive air-fuel ratio control for internal combustion engines. In *2008 American Control Conference*, pages 2058–2063, 2008.

[96] D.C. Youla, J.J. Nongiorno, and C.N. Lu. Single-loop feedback stabilization of linear multivariable dynamical plants. *Automatic*, 10(2), 1973.

[97] G. Zames. On the input-output stability of time-varying nonlinear feedback systems part i: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, AC-11, 1966.

[98] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320, 1981.

[99] G. Zames and B.A. Franics. Feedback, minimax sensitivity, and optimal robustness. *IEEE Transactions on Automatic Control*, 28(5):585–601, 1983.

[100] S. Zhao, A. Abass, and A.T. Shenton. Nonparametric design of robust linear controllers and their experimental application to idle control. In *11th* IASTED *International Conference on Control and Applications. IASTED, Cambridge, UK*, pages 121–128.

[101] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice-Hall, 1996.