Automotive Combustion Modelling and Control

Peter Fussey

New College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2014

I dedicate this work to my father, who has provided inspiration throughout.

Acknowledgements

I would like to thank Liz, Will and Jack for their support with the 'hard sums' throughout my studies.

I am grateful for the support of my colleagues at Oxford, my supervisor David Limebeer for our wide ranging discussions, Giacomo Perantoni and Matthew Arthington. Together with the help and guidance from my industrial supervisor Andy Noble and my colleagues at Ricardo.

Finally, I would like to thank the EPSRC and Ricardo UK for their support of this work through the CASE studentship.

Automotive Combustion Modelling and Control

Peter Fussey, New College, University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2014

This thesis seeks to bring together advances in control theory, modelling and controller hardware and apply them to automotive powertrains. Automotive powertrain control is dominated by PID controllers, look-up tables and their derivatives. These controllers have been constantly refined over the last two decades and now perform acceptably well. However, they are now becoming excessively complicated and time consuming to calibrate. At the same time the industry faces ever increasing pressure to improve fuel consumption, reduce emissions and provide driver responsiveness. The challenge is to apply more sophisticated control approaches which address these issues and at the same time are intuitive and straightforward to tune for good performance by calibration engineers.

This research is based on a combustion model which, whilst simplified, facilitates an accurate estimate of the harmful NO_x and soot emissions. The combustion model combines a representation of the fuel spray and mixing with charge air to give a time varying distribution of in-cylinder air and fuel mixture which is used to calculate flame temperatures and the subsequent emissions.

A combustion controller was developed, initially in simulation, using the combustion model to minimise emissions during transient manoeuvres. The control approach was implemented on an FPGA exploiting parallel computations that allow the algorithm to run in real-time. The FPGA was integrated into a test vehicle and tested over a number of standard test cycles demonstrating that the combustion controller can be used to reduce NO_x emissions by over 10 % during the US06 test cycle.

A further use of the combustion model was in the optimisation of fuel injection parameters to minimise fuel consumption, whilst delivering the required torque and respecting constraints on cylinder pressure (to preserve engine integrity) and rate of increase in cylinder pressure (to reduce noise).

Abstract

The objective of this research is to bring together advances in combustion modelling, control theory and controller hardware and apply them to automotive powertrains.

Automotive powertrain control is dominated by PID controllers, look-up tables and their derivatives. These controllers have been constantly refined over the last two decades and now perform acceptably. However, they are now becoming excessively complicated and consequently time consuming to calibrate. At the same time the industry faces ever increasing pressure to improve fuel consumption, reduce emissions and provide driver responsiveness. The challenge is to apply more sophisticated control approaches which address these issues and at the same time are intuitive and straightforward to tune for good performance by calibration engineers.

In parallel, there has been steady progress in the application of robust, optimal and model predictive control approaches to applications including fuel cells and vehicle dynamics. In particular, the progress has been in making these approaches suitable for fast non-linear, constrained dynamic systems such as an automotive powertrain. In addition to the progress in control approaches, there are increases in processing power and improved modelling techniques allowing more sophisticated models to run in real time.

This research starts with a review of internal combustion engines and their control in Chapter 1. The review also covers the state of the art of the application of new control paradigms to engine control. From this high level perspective, the research was narrowed down to the control of the combustion process itself. Combustion is an area which has been extensively studied over many decades but has seen little integration with the engine control systems.

A combustion model targeted at an engine control unit is developed in Chapter 2, maintaining sufficient detail to predict key parameters such as engine emissions whilst adopting a structure to reduce computational complexity. The model is applied to a direct injection Diesel engine, although many of the principles may also be applied to the more recent Gasoline combustion systems. The model assumes that as the in-cylinder

spray develops, it forms a time-varying heterogeneous combustible mixture. The model is zero dimensional, using a probability distribution to describe the evolution of the mixing process and captures the variation of the air fuel ratio within the spray.

The combustion model is used in a predictive controller which is developed in simulation in Chapter 3 and transferred to embedded control hardware for vehicle testing in Chapter 4. The implementation considers the model structure and the reduction of execution time through parallel computing. The model and optimisation are implemented on a System on a Chip which has a micro-processor and an FPGA on the same piece of silicon. The controller was run in a $2\,l$ Diesel passenger car and demonstrated that a predictive model can be made to run fast enough to be used in a real-time control algorithm and to optimise the NO_x during transient manoeuvres. The combustion controller was then run over the EUDC and US06 drive cycles where it was shown to significantly reduce the NO_x emissions (by over $10\,\%$ on the US06 cycle). The combustion controller demonstrated the potential to calibrate the engine out emissions rather than indirect quantities such as EGR valve position or intake manifold pressures.

During this work, additional applications of the control orientated combustion models were considered; as an analysis tool for engine development in combination with a 1D unsteady CFD model or in the non-linear optimisation of fuel injection parameters. The latter is studied in an optimal control framework in Chapter 5 through a novel application of a general optimisation toolbox, $GPOPS - II$, to the combustion model input parameters. The injection timing and duration were adjusted to minimise fuel consumption whilst delivering the required torque and respecting constraints on the maximum cylinder pressure and rate of cylinder pressure rise. The maximum cylinder pressure is important for engine structural integrity and the rate of cylinder pressure rise influences engine combustion noise.

In conclusion, this work has developed a new control orientated combustion model and demonstrated a fresh approach to engine control showing how model predictive control may reduce calibration effort and engine emissions.

Acronyms

1D	One-dimensional
3D	Three-dimensional
AFR	Air-to-Fuel Ratio
AHRR	Apparent Heat Release Rate
ARM	Advanced RISC Machines, a semiconductor design company
CI	Compression Ignition
CFD	Computational Fluid Dynamics
DPF	Diesel Particulate Filter, used to capture soot in exhaust gas
ECU	Engine Control Unit
EGR	Exhaust Gas Recirculation
EOI	End of Injection
EUDC	Extra Urban Drive Cycle
EVC	Exhaust Valve Closing
FPGA	Field Programmable Gate Array
GPU	Graphics Processor Unit
HDL	Hardware Description Language
IMEP	Indicated Mean Effective Pressure
IVO	Inlet Valve Opening
KE	Kinetic Energy
LNT	Lean NO_x Trap, used to capture NO_x emissions in exhaust gas
MEP	Mean Effective Pressure
MIMO	Multi Input Multi Output
MPC	Model Predictive Control
MVEM	Mean Value Engine Model
NEDC	New European Drive Cycle
NLP	Non-linear Programming
PDF	Probability Density Function
PID	Proportional Integral Derivative controller
PM	Particulate Matter
PWA	PieceWise Affine model
QP	Quadratic Program
SISO	Single Input Single Output
SMD	Sauter Mean Diameter, used for droplet sizes in sprays
SOC	Start of Combustion
SoC	System on a Chip
SOI	Start of Injection
SRM	Stochastic Reactor Model
TR	Trust Region
VGT	Variable Geometry Turbocharger
VHDL	Very high speed integrated circuit Hardware Description Language

Notation

β	coefficient of evaporation according to the d^2 law
β	ratio of pre-mixed and total mass burnt (Chapter 5)
δ	segregation factor for β - PDF distribution
$\tilde{\epsilon}$	dissipation rate of mean specific kinetic energy
γ	ratio of the specific heat coefficients c_p/c_v
λ	air fuel ratio normalised to stoichiometric air fuel ratio
ϕ	fuel to oxidiser ratio normalised to the stoichiometric ratio ($\phi = 1/\lambda$)
ρ	density
c	concentration
CA	Crank Angle [degrees]
c_v, c_p	specific heat coefficients for constant volume and pressure processes
d	diameter of droplet in spray [m]
d_0	initial diameter of droplets in the spray, the Sauter Mean Diameter [m]
J	cost function
k	total turbulent kinetic energy density
$\tilde{K}$	mean specific turbulent kinetic energy
m	mass [kg]
N	number of particles
N_{eng}	engine speed [rev/min]
p	pressure [Pa]
P	PDF probability [0-1]
q	mass of fuel injected in one time-step [kg/s]
Q	cumulative heat release [J]
R	reaction rate
S	spray length [m]
t	time [s]
T	temperature [K]
Tq	torque [Nm]
V	volume [m^3]
x	burn fraction
X	EGR ratio
Y	mass fraction
Z	mixture fraction

Subscripts

a	air
b	burnt
cyl	cylinder
$diss$	dissipation
e	equilibrium
$evap$	evaporated
f	formed or fuel or frozen
g	gas
$igndelay$	ignition delay
l	liquid
nom	nominal
o	oxidised
par	parcel of fuel
s	spray
$soot$	soot
v	vapourised

Modifiers

$[x]$	concentration of x [ppm]
$\hat{x}$	estimate of x
$\bar{x}$	mean of x
x^*	optimum value of x
$\overline{x'^2}$	variance of x

Contents

Chapter 1

Introduction

1.1 Powertrain control in automotive applications

An automotive powertrain typically consists of an internal combustion engine with air intake, exhaust and fuelling systems, and in the case of hybrid electric vehicles, batteries and electric machines. This study will consider conventional powertrains with just the internal combustion engine systems, controlled with an electronic Engine Control Unit (ECU).

There are many motivations for developing automotive powertrains: emissions legislation, CO_2 emissions reduction requirements, piece cost, noise and vibration and driveability. This results in many new air, fuelling and aftertreatment systems being added to the base engine. To support these developments, the ECUs are designed and developed to maximise the advantages gained from new hardware. In parallel, the development of the ECUs is subjected to the various requirements of automotive controllers: wide climatic ranges, unpredictable driver demands, long operating life, mass production and so on.

Emissions legislation (see Figure 1.1) and CO_2 emission reductions (with corresponding fuel economy improvements) are driving the development of the increasingly complex systems, for example: multiple turbochargers, multiple exhaust gas recirculation (EGR) routes, cylinder pressure feedback and a variety of exhaust aftertreatment systems.

Future legislation is expected to continue to reduce the emissions limits and also revise the test procedures. Over recent years, there has been concern that emissions levels, in particular NO_x emissions, measured in real-world driving differ from those measured on the legislated test cycles. In response to this concern, the European Commission established a Real-Driving Emissions working group which is developing a new test procedure based on measurements in vehicle using a portable emissions measurement

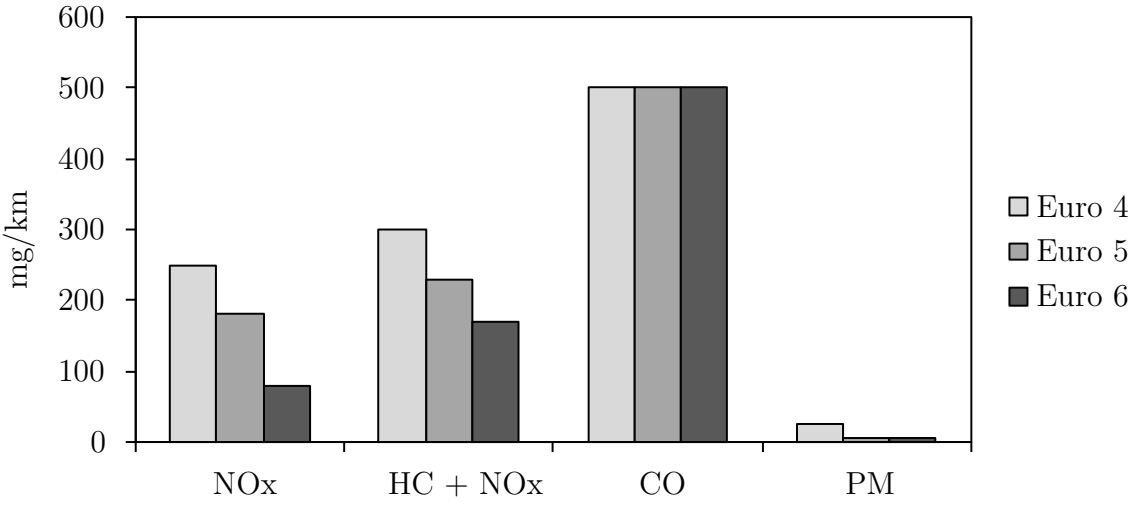

Figure 1.1: Overview of European Emissions Legislation [2] (Diesel passenger cars).

system. This change is likely to increase the emphasis on the reduction of NO_x emissions produced during transient manoeuvres.

At a high level, the vehicle powertrain responds to the driver's pedal request and delivers torque that alters the vehicle speed. The driver then observes the vehicle speed and closes the loop with the pedal request to regulate the vehicle speed (Figure 1.2). Fuel is consumed and harmful gases (emissions) are produced.

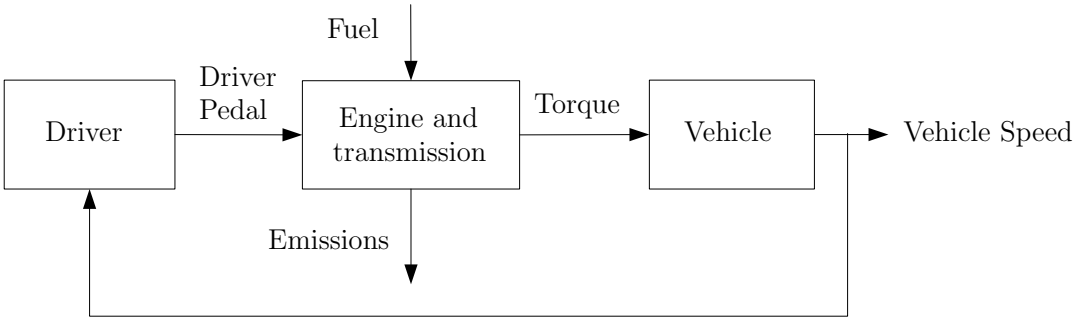

Figure 1.2: Overview of an automotive powertrain.

Internal combustion engines can be grouped into either spark ignition (SI) or compression ignition (CI) engines. This study will mainly consider compression ignition (also known as Diesel) engines as they are currently the area of most active development and have more complex systems. In many cases, the approaches developed for Diesel engines will also be applicable to SI engines.

In Figure 1.2, within the boxes containing the engine, transmission and vehicle there are a number of control units which control quantities such as: the flow of air, quantity and timing of the fuelling, gear selection, torque requests to improve vehicle stability

and so on. By focusing on the engine control unit (ECU), the inputs and outputs can be summarised as (Figure 1.3):

- Turbocharger (Turbo) - the intake manifold boost pressure is generated by a turbo, which is either a Variable Geometry Turbocharger (VGT), or a fixed geometry turbo fitted with a bypass valve (wastegate). The control of the boost pressure is achieved through changing the geometry of the turbo, or by moving the wastegate;
- Exhaust Gas Recirculation (EGR) - exhaust gas can be fed back into the intake system to reduce emissions, the EGR is controlled with a valve and may also have a cooling circuit with bypass to control the EGR temperature;
- Fuelling system - a high pressure fuel system is controlled by the ECU which can vary the injection timing, the number of injections, the durations and the fuel injection (or rail) pressure;
- Sensors - typically include: driver inputs, airflow into the engine, engine speed, temperatures and pressures with a possibility of oxygen sensors in the exhaust and a limited number of position feedback sensors on the actuators.

The ECU provides the link between the driver pedal and the actuators with inputs from the sensors. Note, the engine torque and many of the emissions are not usually measured in vehicle (see Figure 1.3).

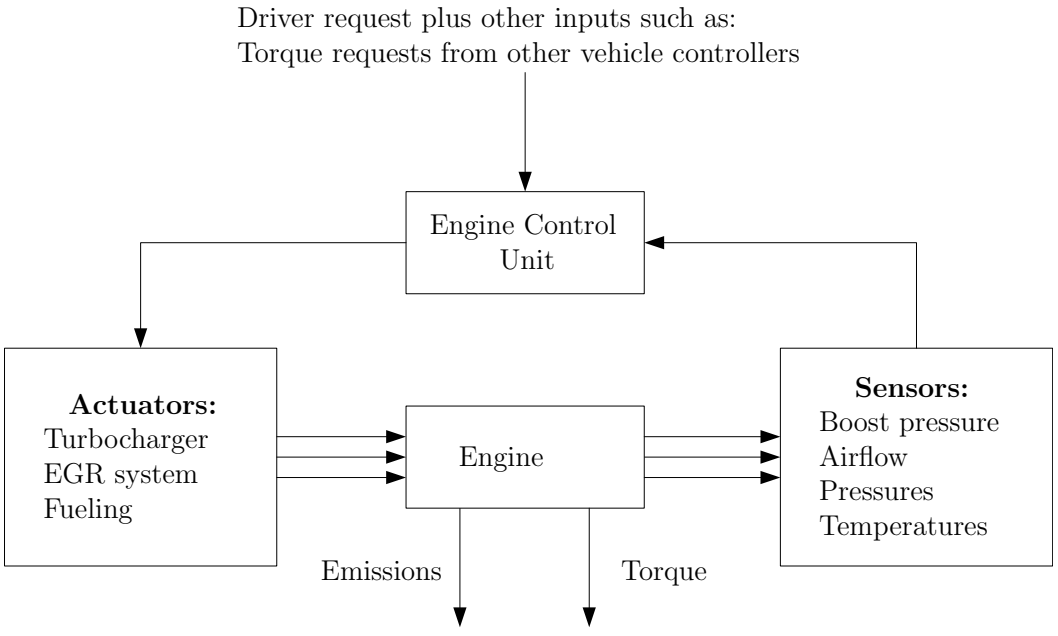

Figure 1.3: Overview of the Engine Control Unit.

1.2 Engine processes

Figure 1.2 shows that at a high level the powertrain responds to the driver by converting fuel to regulate the vehicle speed, whilst also producing emissions. At the centre of the powertrain is the engine and at the centre of the engine is the cylinder where the fuel is combusted and the energy released. Therefore, the introduction to the engine processes starts with the combustion itself.

1.2.1 Combustion

Compression Ignition combustion is a complex process involving the compression of an air and fuel mixture in the cylinder which ignites when it reaches a sufficiently high temperature. The mixture in the cylinder often includes fresh air and burnt exhaust gases (EGR), which are reintroduced into the cylinder to reduce engine emissions. Fuel is injected directly into the cylinder at high pressures (1500 to 2000 bar). Controlling the timing of the subsequent ignition is critical, though difficult to predict since there are mixed phase fluids being compressed, followed by a multitude of combustion or oxidation reactions as the fuel is burnt. The fuel itself is a mixture of hydrocarbons with a variety of combustion characteristics so the overall combustion is typically a superposition of many chemical reactions.

Combustion is affected by the composition of the gases coming into the engine (which is typically covered by the air path control) and the fuelling control which includes the injection timing, pressure and quantity.

Combustion processes have been studied for many years with techniques such as laser imaging (using optical cylinders). These facilities allow the combustion to be visualised and analysed. The experimental work has been combined with detailed modelling of the chemical mixing processes in the jet, as summarised in [12] and [31], and illustrated in Figure 1.4.

The combustion itself is typically characterised by four stages, see Figure 1.5 with the pollutant formation summarised in Figure 1.6.

- Ignition delay - An initial delay from start of injection (SOI) to the start of combustion (SOC). The SOC is identified as a change in gradient on the heat release curve;

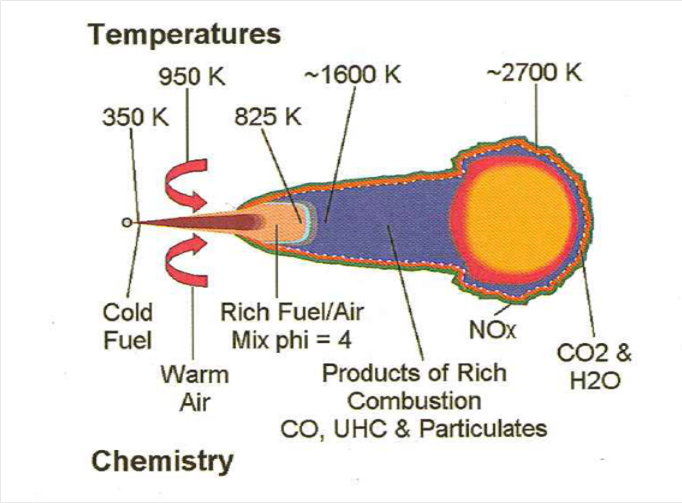

Figure 1.4: Summary of the burning process [31]. Reprinted with permission from SAE Paper No. 1999-01-0509 ©1999 SAE International. Further use or distribution of this material is not permitted without prior permission from SAE.

- Pre-mixed combustion - The fuel that has mixed with the air during the ignition delay period all burns at once leading to a rapid heat release;
- Mixing controlled combustion - Following the pre-mixed combustion, the combustion is controlled by the rate at which the mixture becomes available for burning. This is controlled by many processes including the evaporation of the fuel and the turbulent mixing of the spray;
- Late combustion - A slower phase of combustion as the remaining fuel and combustion products continue to burn, rate slows as cylinder volume expands and the temperature falls.

Another way to visualise the Diesel combustion process has been proposed in [10], which maps the different combustion processes (ignition, soot formation[1] and NO_x formation) with respect to the local in-cylinder equivalence ratio (ϕ) versus local temperature (T) characteristic (see Figure 1.7). The equivalence ratio ϕ is the fuel to air ratio, normalised to the stoichiometric fuel to air ratio. 'Local' quantities are a function of location within the cylinder, as opposed to bulk quantities which are averaged over the cylinder. The base characteristic of emissions generation versus ϕ and T can be calculated from chemical reaction simulations. The ϕ versus T profile for the fuel spray as it is injected

[1]Note, strictly speaking, particulate matter (PM) emissions from combustion engines include both soot (carbon) and hydrocarbons absorbed onto the soot particles. This thesis only considers soot since this is the key constituent of the PM emissions and existing models predominantly consider only soot emissions.

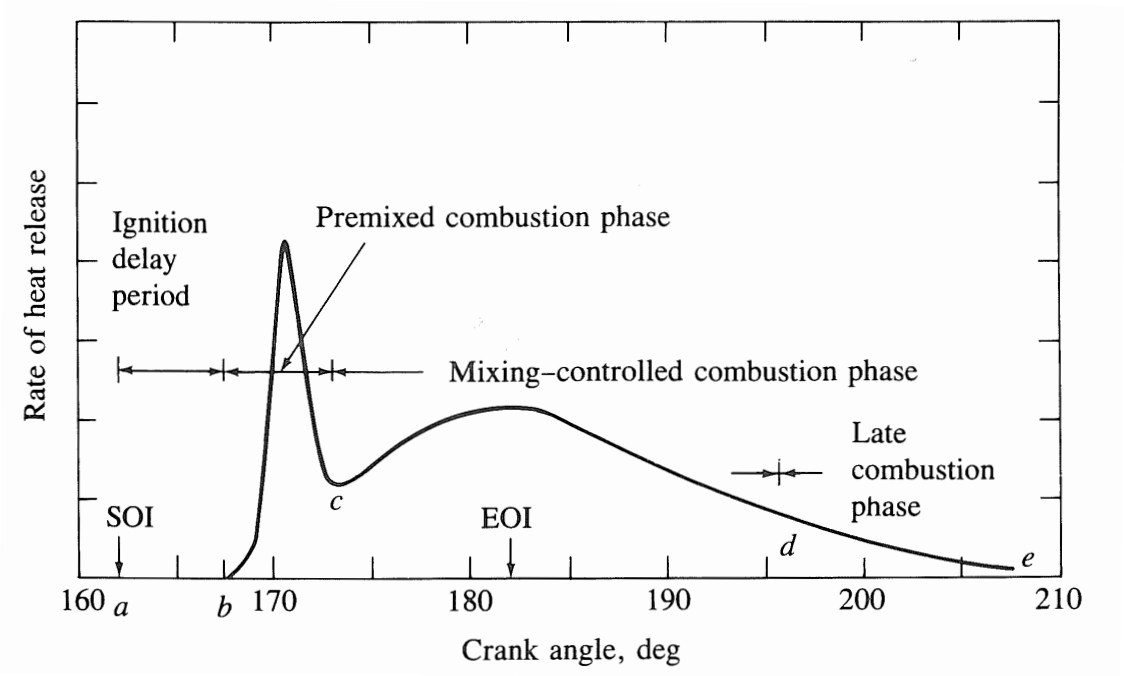

Figure 1.5: Direct Injection Diesel heat release, [44], where EOI = End of Injection.

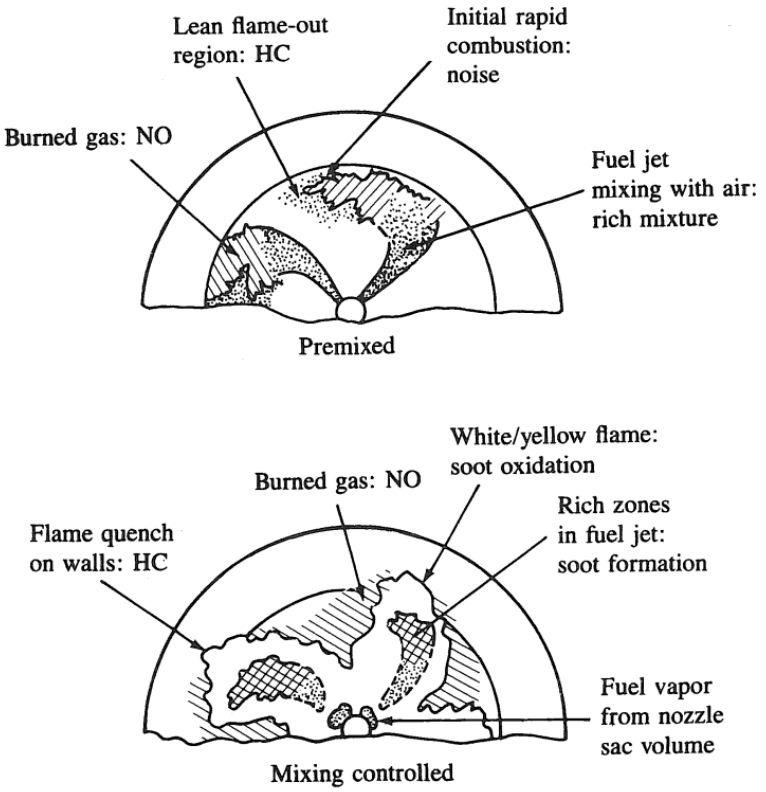

Figure 1.6: Direct Injection Diesel pollutant formation, [44].

into the cylinder can then be superimposed onto this reaction characteristic. One way to do this is to take the output from 3D CFD simulations and plot ϕ versus T for each cell across the spray, [10], note the value of ϕ is taken from each cell just before combustion. A more qualitative explanation is given in [74] where a path for the spray on the ϕ versus T axes is proposed, as illustrated in Figure 1.8, providing the control engineer with direction in how to reduce harmful emissions.

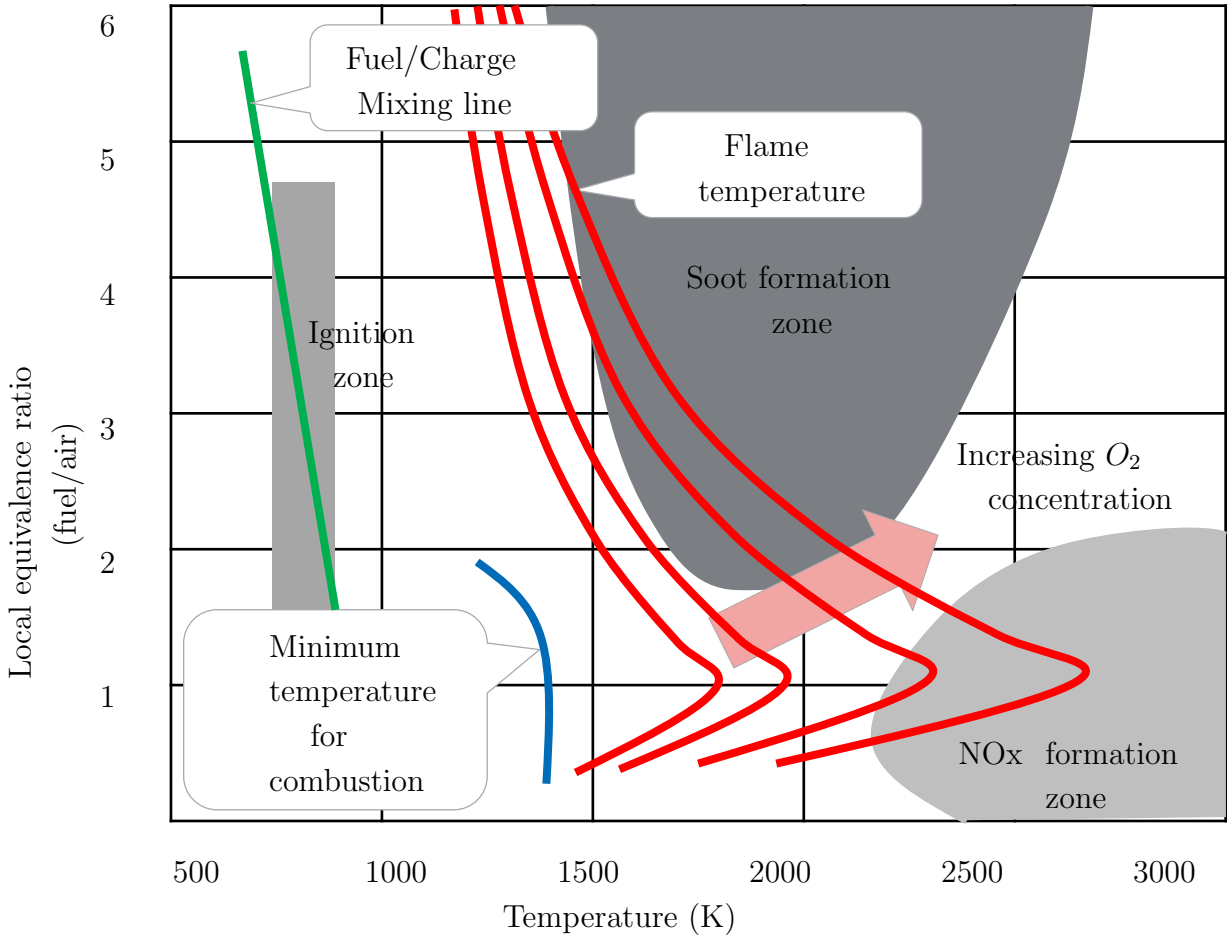

Figure 1.7: Local equivalence ratio versus temperature, after [10] and [74].

The control of combustion has largely been open-loop with the following exceptions:

- Cylinder pressure feedback - using sensors in the cylinder to feedback the combustion pressure or combustion events (for example ion current), for which there are very few production examples, eg [42];

- Oxygen sensor feedback - oxygen sensors (also known as lambda sensors) in the exhaust system can provide feedback of the exhaust gas mixture at the end of the combustion process (ie not during combustion) and are also an averaged signal due to the mixing of gases from different cylinders. Note, lambda is used in internal

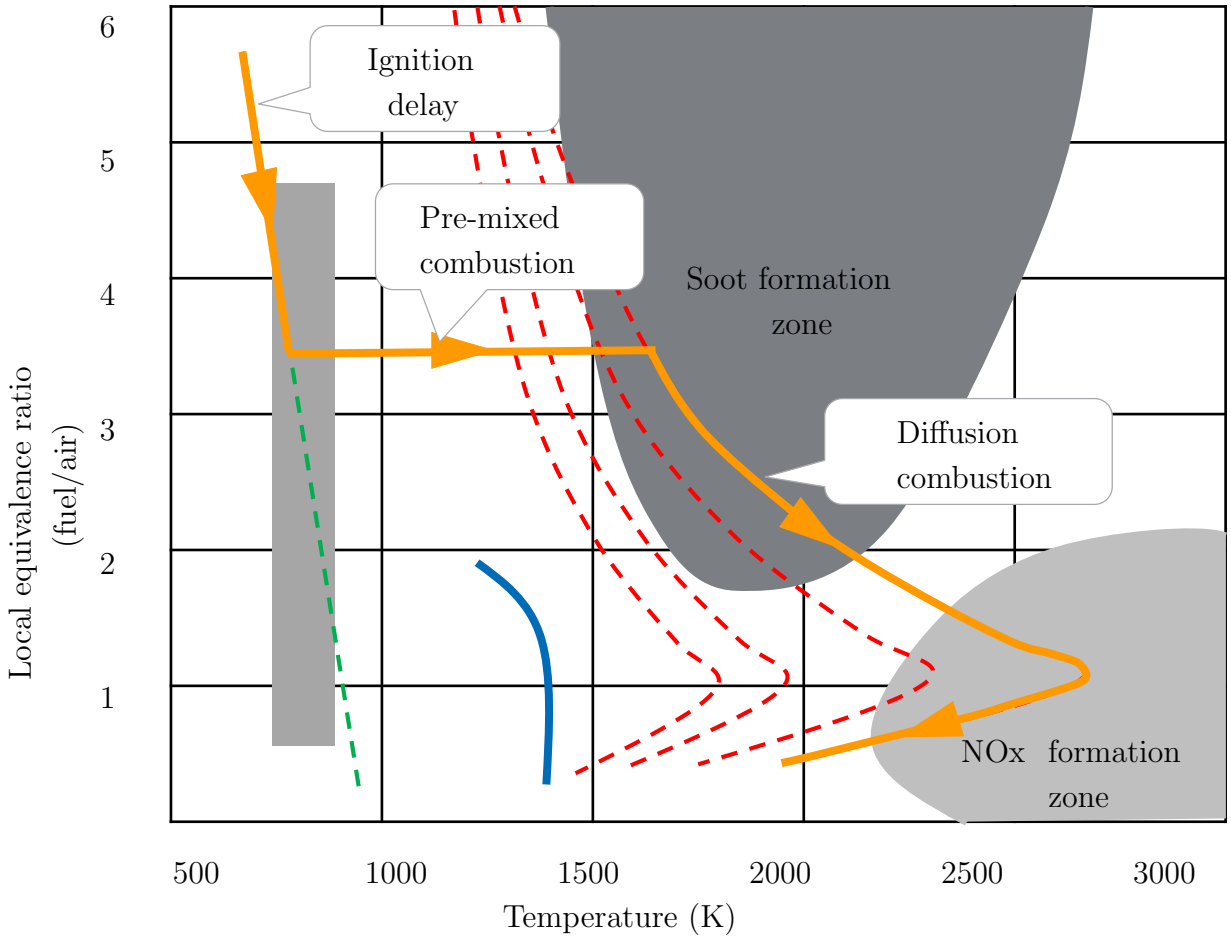

Figure 1.8: Local equivalence ratio versus temperature, tracing the mixture during combustion, after [74].

combustion engine development to mean air to fuel ratio normalised to stoichiometric air to fuel ratio, where the stoichiometric air to fuel ratio is the balanced air-to-fuel ratio from the combustion reaction.

For open-loop control, the fuelling and airpath target set-points are calibrated on an actual engine and in vehicle. The calibration process takes detailed emissions, noise and fuel consumption measurements on test-benches to optimise the set-points off-line. The set-points are then stored as look-up tables, typically versus engine speed and requested torque from the engine, with corrections for factors such as altitude, climatic conditions, aging, transient driving conditions and so on. This control gives rise to two challenges:

- The set-points are based on steady-state calibrations, whilst the engine generally operates in a transient manner;
- All of the look-up tables require extensive calibration activities which is expensive.

The reduction in the number of look-up tables is a current area of research, for example [11].

1.2.2 Air System

There are air and fuel systems outside the cylinder which are controlled by the ECU so that they deliver the gases and fuel into the cylinder at the right temperature, pressure and with the right chemical makeup (principally O_2 concentration). A typical engine system is presented in Figure 1.9 and includes the following components:

- Variable Geometry Turbocharger;
- Intercoolers;
- Multiple EGR routes with coolers and bypasses;
- Oxygen sensors in exhaust;
- Pressure, mass air flow and temperature sensors in intake system.

We will now consider each of these elements in turn and in more detail. The variable geometry turbocharger is introduced to improve low-speed response by closing down nozzles to speed up the airflow. At high speeds the nozzles are closed to restrict the flow. These characteristics give rise to a 'DC gain reversal'.

Intercoolers are used to reduce the temperature of the gases going into the cylinder, this has benefits in terms of combustion, and by increasing the density of the gases, allows more air to be drawn into the cylinder.

Exhaust Gas Recirculation (EGR) has a number of effects on combustion, see [49]. The main benefit for Diesel engines comes from the dilution of the oxygen in the inlet gases. By replacing oxygen with burnt gases, the oxygen concentration is reduced, which reduces the flame temperatures and hence this reduces the NO_x emissions. If the EGR is not cooled some of this benefit is lost. If too much EGR is used then there is not enough oxygen for complete combustion and particulate matter may be produced. The different EGR routes, before or after the turbines, can affect the overall engine efficiency and responsiveness, for example, with a longer EGR route, there is a greater distance from the EGR valve to the cylinder which leads to delays in the system response.

Lambda sensors in the exhaust system are mainly used to control the air-to-fuel ratio in the exhaust system, for example, during the rich operation to regenerate Lean NO_x Traps. In addition, they can be used to adapt the calibration parameters in the ECU, for example the injector characteristics which may drift during the lifetime of the engine or be subject to production variation.

Pressure and temperature sensors in the intake system are used to feedback outputs to the ECU (boost pressure and mass air flow), or to provide inputs for state observers.

Once the target set-points for the airpath have been calibrated, the controller needs to control the turbo and EGR system to deliver to these targets. These systems are coupled multi-input multi-output (MIMO) systems with different timescales and the boost system is further complicated by 'DC gain reversal'. These concepts are explained below.

A turbo has three main components, rotating at high speed on a common axis: the turbine, the compressor and the shaft. The turbine is in the exhaust flow and extracts energy from the exhaust gases. The compressor is in the intake system and increases the pressure in the intake manifold. The shaft connects the turbine and compressor. The boost pressure can be controlled by varying the geometry of the turbine (VGT), which affects the efficiency of the turbine and hence energy being taken from the exhaust. Increasing the energy being extracted will increase the rotational speed of the system, increasing the speed of the compressor and hence the intake manifold boost pressure

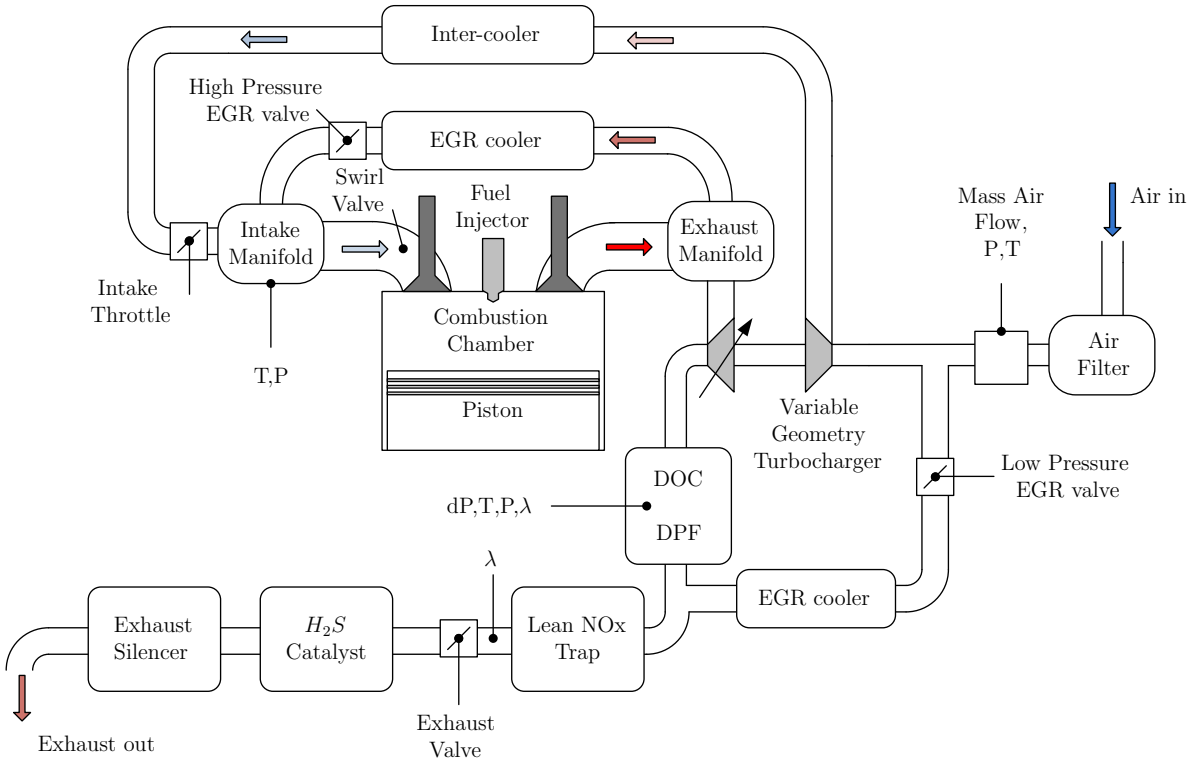

Figure 1.9: Example production engine, after [42], including a Diesel Oxidation Catalyst (DOC), Diesel Particulate Filter (DPF), Lean NO_x Trap and delta pressure (dP), pressure (P), temperature (T) and lambda (λ) sensors.

downstream of the compressor. The VGT can therefore be used to control the pressure of the intake gases into the engine.

However, the VGT also affects the pressure in the exhaust manifold. The exhaust pressure affects the flow of EGR, since the EGR flow is a function of the pressure drop across the EGR system (the exhaust manifold is typically at the upstream end of the EGR circuit). As a result, the VGT can also affect the EGR flow and the control of the EGR and boost pressure are coupled.

Another factor to consider is the different dynamic responses of the boost and EGR systems. The turbo has a longer time constant than the EGR due to the rotational inertia of the turbine and compressor and this can impact the design of the control algorithms.

1.3 State of the art: Automotive control

Automotive control is dominated by PID controllers with additional features such as feed-forward terms, gain scheduling and anti-windup. With such a complex system combined with extensive validation testing requirements and cost pressures, automotive

control development has historically taken an evolutionary route preferring additional look-up tables to a new control approach.

That said, automotive research and development teams have been exploring more modern control methodologies including:

- Model predictive control;
- Modelling and system identification;
- Hardware and software acceleration.

There has been a drive to reduce calibration effort and introduce instead a systematic development process, [11], [32] and [81]. The solutions discussed by these authors use models of the engine in different ways. Reference [11] includes models of the engine in the controller and uses these to optimise the control actions, [32] uses a model of the intake system to estimate the oxygen concentration and uses this as the controlled variable, whilst [81] uses Piecewise Affine (PWA) models of an air system together with explicit MPC to control a Diesel engine.

The following sections begin with an overview of the structure of a typical current production control unit and then summarise the current state of the art in each of the areas of research and development areas identified above.

1.3.1 Current production engine control hardware

An overview of a typical production engine controller hardware is shown in Figure 1.10. The key components are the micro-processor, memory, sensor input circuits, communications interfaces, output driver circuits for injectors and actuators.

The micro-processor technology is moving to multi-core micro processors for the higher end control units, for example the Infineon Tri-core family.

1.3.2 Current production engine control algorithms

Production engine control algorithms are developed in parallel with the introduction of new engine systems. The current 'state of the art' production engines are typified by multiple turbochargers, multiple EGR circuits, multiple coolers and aftertreatment systems (Oxidation catalysts, Diesel Particulate Filters and NO_x aftertreatment). An example is the VW engine described in [42], which has a VGT, twin EGR circuits (one either side of the turbine) and a Lean NO_x trap.

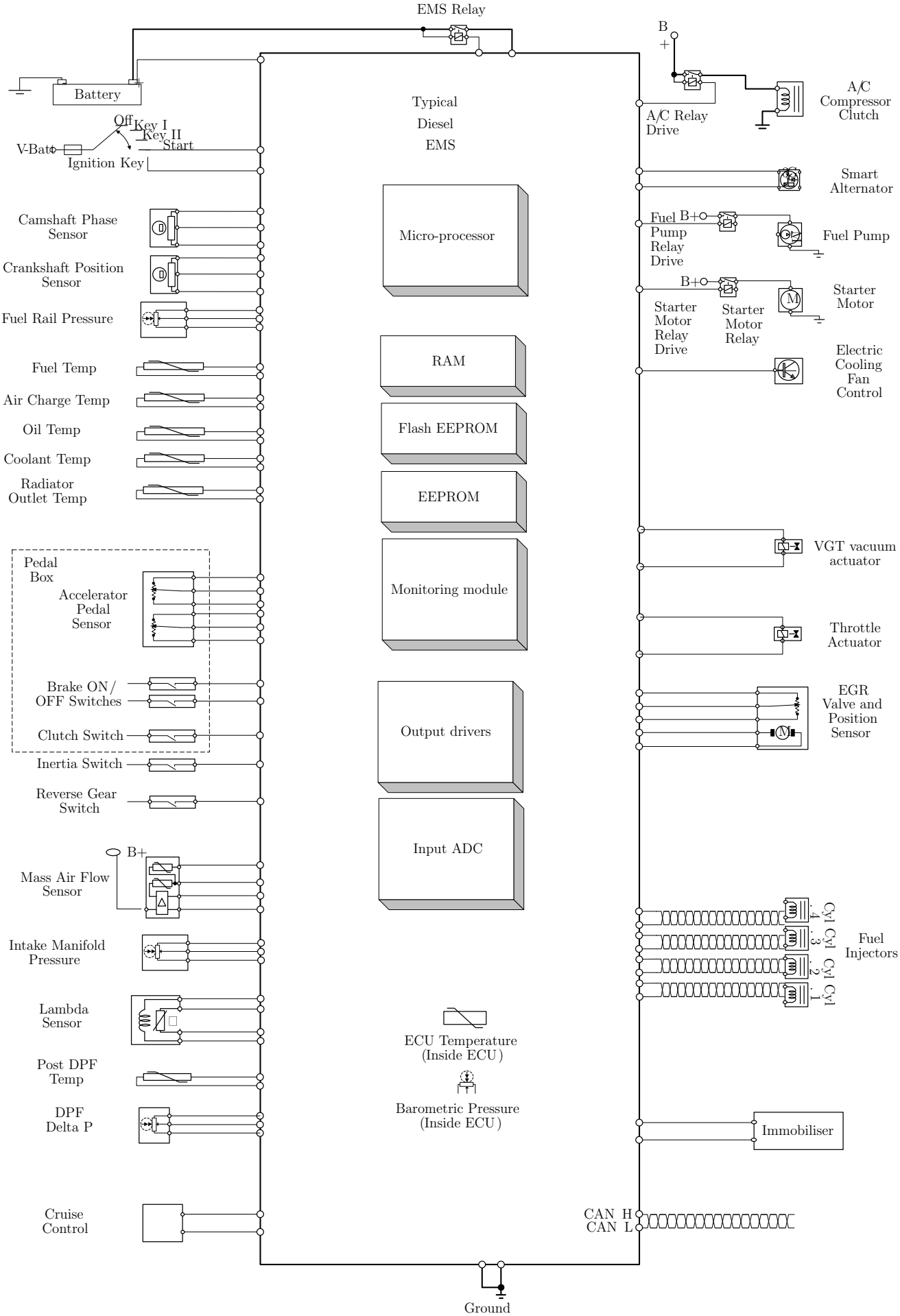

Figure 1.10: The main components of an Electronic Diesel Control unit and connections to the 'external world', after [29].

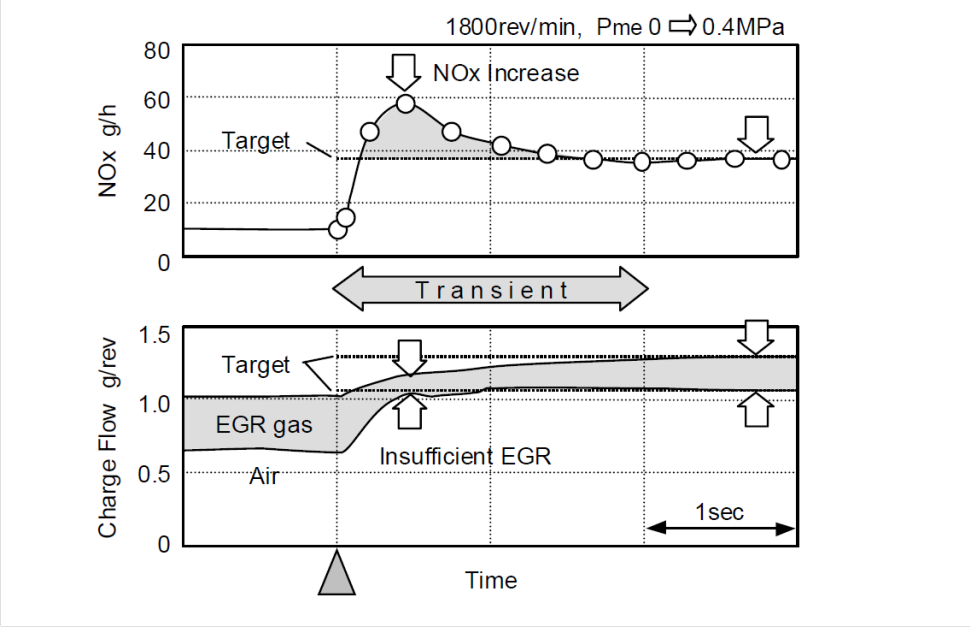

Figure 1.11: EGR control during transients [32].

Air system control has been the subject of research for many years. An overview of advanced, multivariable control methodologies and an assessment of their performance is presented in [89], from which it is concluded that the actual set-points for the controllers are more important than the control approach selected. A control approach for the EGR is derived in [66] that uses the link between NO_x emissions and flame temperature, and then flame temperature and oxygen concentration, to give a control approach based on oxygen concentration. This gave improvements in the transient control of NO_x emissions. The improvement occurs during transients where turbo lag causes a problem with the classical approach of regulating mass airflow with the EGR system and the boost pressure with the VGT. During a transient, the response to the EGR control is faster than the boost response, resulting in insufficient EGR, Figure 1.11.

A similar approach was demonstrated in [20] using measured exhaust lambda. This approach was seen to deliver significant improvements in transient NO_x emissions, explained through a link from exhaust lambda to intake oxygen concentration.

1.3.3 Model predictive control

Modern control approaches use optimisation to increase the range of operation of the ECU and to reduce the amount of specific calibration effort required - since by including a model of the plant being controlled, the controller can adapt to different boundary conditions. The model based approaches have other benefits such as being designed to

handle multivariable problems and being able to take constraints (both actuators and outputs) into account.

Control approaches using models and forward predictive optimisation are often referred to as Model Predictive Control (MPC), which uses a model to predict different control scenarios and optimise the selection of the control input on-line, see [58] and [77]. This approach was initially adopted by the process industry where controllers operate with relatively low update rates allowing sufficient time to carry out on-line calculations. The early work in MPC was split between applications in industry [58] and academic studies looking at proving stability and robustness, [61], that the industrial applications were demonstrating. With the continuous increase in computational power coupled with a more rigorous academic base, MPC has begun to find more applications with faster control update rates and in particular, automotive applications.

The application of MPC to automotive problems is reviewed in [24]. The conclusions from this book suggest that MPC has significant challenges in terms of processor and memory loadings, robustness in automotive applications and requirements for a high degree of skill in order to use and calibrate these systems. The authors go on to conclude that since the model is at the heart of MPC, this should be the main area for development. Automotive applications of MPC are still a research technology with few applications and little published work relating to production controllers.

The automotive applications of MPC tend to be restricted to specific examples in combination with limited physical testing. The examples can be grouped by application area:

- Idle stability control - a classic automotive control problem which has been studied for many years. It only affects one operating condition and so is a good candidate for exploring the application of MPC. Since the problem is well understood, it is also useful for exploring some of the issues associated with implementation into automotive control units;

- Turbocharged Diesel engines - a classic MIMO air path control problem that has yet to be fully addressed, should take advantage of the constraint handling characteristics of MPC;

- Fuel Cell control - whilst not an internal combustion engine, fuel cells are being developed for automotive applications and will use automotive control units with

similar processor/memory requirements to ECUs. In addition they have some characteristics which are similar to internal combustion engines, for example, they are required to control an oxygen concentration within constraints.

- Lean NO_x Trap (LNT) regeneration - aftertreatment components that trap emissions require regular regeneration (or cleaning). In the case of a LNT this is achieved by managing the engine-out emissions to be rich. These conditions allow the chemical release of the NO_x, which is then converted to N_2. The controller needs to have a target composition for the exhaust gases, which minimises additional fuel consumption whilst allowing the LNT to regenerate.

MPC applied to idle stability control

Idle stability control is a classical challenge for SI engines where an increase in load, for example an air conditioning compressor coming on, causes the engine idle speed to drop and potentially stall the engine. The engine speed is regulated by a combination of throttle, fuel injection and spark timing and presents a challenging problem due to time delays and constraints.

An example of idle speed control using MPC is presented in [19], where a linearised model of the engine response is derived. The constraints are on engine speed and throttle angles. The states are estimated with a Kalman filter.

On-line optimisation is still too computationally demanding for an automotive control unit so the authors used an explicit solution to the MPC optimisation problem. The solution to the optimisation problem is stored as a Piecewise Affine function, with the control output being a function of the current state and the reference values. This is then stored in the control unit so the controller can look up the solution rather than repeatedly solving the associated optimisation problem. The MPC algorithm was developed in simulation and then implemented in vehicle, demonstrating higher performance than the baseline PID based controller, whilst being within the capabilities of an automotive control unit.

Whilst this approach has benefits in terms of processor power and memory requirements, more complex problems can rapidly run into memory constraints. Secondly, by carrying out the optimisation off-line (ie. before the engine runs) there is no opportunity to update, or adapt, the models during production or during the life of the vehicle.

MPC applied to turbocharged Diesel engine control

The application of MPC to turbocharged Diesel applications has been approached in a number of different ways: for example [81] takes a non-linear model which is approximated by a Piecewise Affine surface to allow an explicit implementation of MPC to be developed for the control of the EGR and VGT actuators. A Kalman filter was used for state estimation and engine-out NO_x emissions. The algorithm was successfully run on an automotive microprocessor, MPC555, controlling a 2.2 litre Diesel engine. The authors acknowledge the challenge of limited ECU memory availability and propose further research directions that will reduce the memory requirements in future implementations.

A second example, [50], is an application to a heavy duty Diesel engine. The model here is a dynamic six-state model, obtained through system identification, with four control inputs: crank angle for start of injection, fuel injection duration, EGR and VGT actuator positions, and five measured output variables: engine torque, angle of 50% burn, rate of cylinder pressure rise, engine out NO_x and engine out soot. Whilst these quantities were measured in this case, they are not typically available to automotive control units and would therefore require estimation. The MPC algorithm is developed using the standard MPC toolbox from Matlab and is only demonstrated around a fixed operating point. The MPC algorithm was shown to work at one operating point, and the authors conclude with proposals for extending their algorithm to several linear models to cover a wider operating range.

A third example, [79] considers a sequence of identified linear models to cover the state space. The locally linear models are used in the application of model predictive control approach. This example also attempts to tackle the conflicting dynamic responses for different actuators through a multi-level predictive control model.

MPC applied to Fuel Cell control

A series of papers are summarised in [88], where an eight-state, linearised model of the Fuel Cell is used in an on-line optimisation routine to control the currents within the system to avoid starving the cathode of oxygen. This paper presents an analysis of the impact of the prediction horizon on the stability of the resulting system, a longer horizon leads to improved stability which is traded off against a corresponding increase in computational burden.

MPC applied to LNT control

The application of MPC to exhaust aftertreatment is studied in [47]. A model of a lean NO_x trap (LNT) is built up from physical principles and simplified to 3 states: stored oxygen fill ratio (x_1), stored NO_x fill ratio (x_2) and LNT temperature (x_3). MPC is applied to the control of the fuelling during a rich event, when the LNT is being emptied. The cost function is of interest as it seeks to minimise the additional fuel consumed ($\dot{m}_{extrafuel}$) divided by the NO_x released (r_{NOxrel}) which gives a specific NO_x reduction quantity, together with the weighted NO_x left at the end, $t = N$, (weighting factor k). Rather than the difference between the outputs and set-points which is commonly used:

$$J = min \left[\left(\frac{\sum \dot{m}_{extrafuel}}{\sum r_{NOxrel}} \right) + kx_2(N) \right] \tag{1.1}$$

This is an initial step in applying MPC to aftertreatment control and the authors conclude by outlining the next phase which is to include the triggering of the regenerations as an outer supervisory control loop.

MPC summary

MPC has been applied to various automotive applications, with the majority based on linear or linearised models. The linearisation has been necessary to enable the MPC algorithms to run in real time, however this does present major challenges to the widespread application to automotive problems which are frequently non-linear, require fast control update rates and have limited computational resources available.

1.3.4 Engine modelling

The importance of the system modelling has been alluded to in Section 1.3.3. Historically, the subject of engine performance modelling has been approached from the perspective of three different requirements: physical design of the engine, performance and emissions optimisation during development and control system design. Engine models have been exchanged between these different disciplines, so it is appropriate to start the discussion on modelling with an overview of the commonly used models in each of these areas.

- In the design phase, detailed models are used to predict the performance and influence the physical design of the engine often before any engine hardware exists. Models that are used for these purposes include:

  - 3D Computational Fluid Dynamics (CFD) models which are used for detailed port flows, manifold distribution, in-cylinder combustion and any system with 3 dimensional characteristics;
  - 1D unsteady CFD simulations are used for tuning the physical performance parameters such as turbocharger selection, valve timings, manifold runner lengths and acoustic development of the air system tuning components such as resonators and exhaust system geometry;
  - Coupled chemical simulations looking at problems such as the performance of exhaust aftertreatment systems and combustion processes;
  - Thermal models looking at the integration of the various heat exchangers into the underbonnet package space.

- During development of the engine, in addition to re-running the simulations above, simulations can be used to explore performance and emissions tradeoffs over test cycles. These simulations typically use simpler identified models based on initial prototypes to run over longer timescales.

- Control system development can use either models based on physical principles or models identified from measurements:
  - Air-system 1D and 0D models are used to explore, test and develop air-system control algorithms. The 1D models can be carried over from the development phases, although they may be optimised for fast running and capturing specific control responses. The 0D models are commonly known as Mean Value Engine Models (MVEM) [49];
  - Identified models use sensors to update estimates of the system characteristics.

We will now look in more detail at the modelling approaches for the engine processes described in Section 1.2, Combustion and air systems.

Combustion modelling

Combustion is a complex problem that includes the following thermodynamic processes: gaseous compression, viscous air motion, turbulent diffusion, fuel spray penetration, droplet breakup and evaporation, air entrainment, chemical kinetics and so on. The

latest 3D CFD simulations seek to include all of these features and they can be used to estimate exhaust emissions.

In parallel, simpler models of combustion have been considered for many years. Early work, [69], proposed a multi-zone model of combustion, Figure 1.12, coupled with extended Zeldovich mechanisms to predict NO_x concentration for stoichiometric burning. An update of earlier work, with laser measurements is presented in [31], further updates with the latest combustion technology, low temperature combustion and NO_x reburning mechanisms are presented in [12]. Whilst these references look at the modelling processes, they consider models which are too complex for implementation on an ECU.

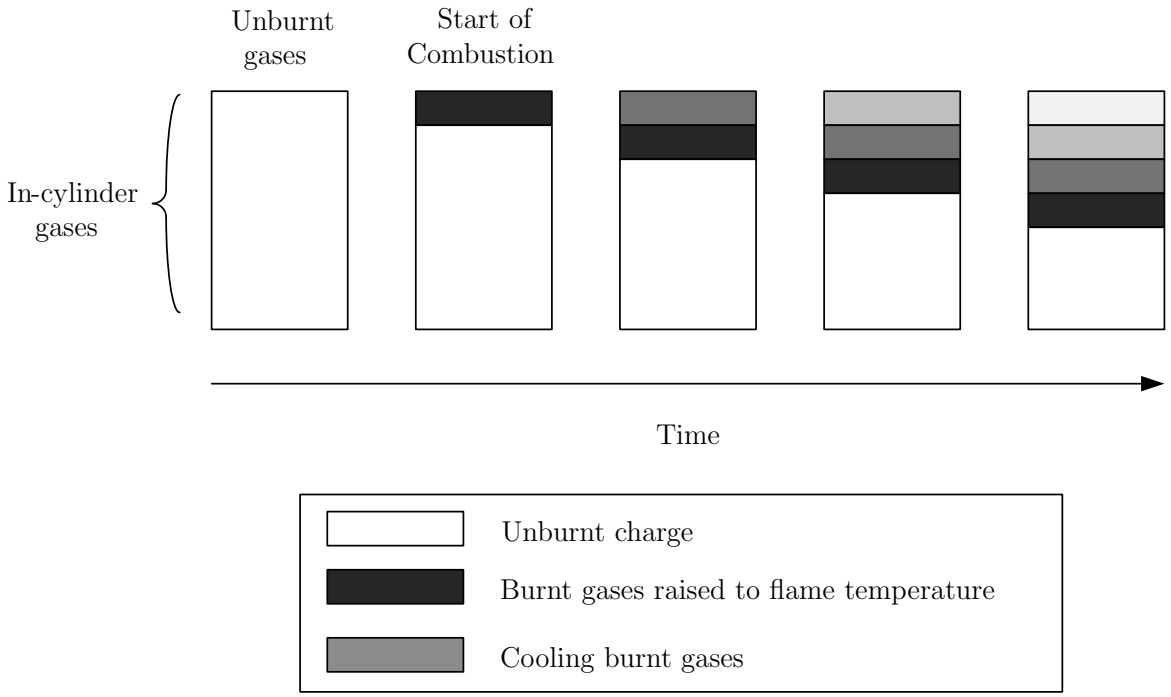

Figure 1.12: Combustion model used to calculate temperature profiles, after [69]. Bars indicate distribution of gas temperature during combustion, starting from unburnt gases. Each parcel of burnt gas is raised to flame temperature and then cools according to a polytropic relationship.

A two-zone model proposed by [28] has unburned and burned zones coupled with the extended Zeldovich mechanism to predict the NO_x emissions. This approach relies on a combination of physically based models and identified characteristics such as ignition delay and radiative heat losses to keep the computation effort as low as possible. However this approach does not include a detailed consideration of the fuel injection into the cylinder and the subsequent mixing.

Air system modelling

The Mean Value Engine Model (MVEM) of the air system has been used extensively as a tool for control system development. Parts of these models have been embedded into control systems and in these cases the model has been typically used to estimate states or values which are not measured, for example, [66] estimates EGR rate and [42] uses an air system model with parameter adaption to control a twin EGR air system.

The MVEM has also been used as a starting point for several academic studies into engine control, for example it is used as a basis for deriving a linear parameter varying model (LPV) which was then used to explore robust control theory in [49]. A simplified 7 state MVEM was reduced to a 3 state model in [48] that was then used as a basis for an optimal control design exercise. The optimal controller is demonstrated on a full simulation model and then experimentally on a engine test bench.

Modelling summary

The control engineer faces a dilemma when preparing models for use with embedded control systems, they need to run quickly and yet be accurate enough to serve the desired purpose. There is a wealth of complex models for engine systems, however the challenge is in selecting the correct approach and the subsequent optimisation to facilitate the development of model based control strategies.

1.4 State of the art: Hardware and software acceleration

In common with other areas of computer science, the automotive control unit capabilities have been steadily increasing. However, the long lifecycles and extensive validation required for automotive controllers have restricted the advances to approximately a doubling of performance every five years, compared to Moore's law which predicts the doubling the number of CPU transistors every two years, [53]. The performance is typically measured in terms of clock speeds and memory, both of which are increasing.

The use of Field Programmable Gate Arrays (FPGA) in automotive applications has been growing. The performance of FPGAs has been combined with improvements in the toolchain, VHDL-AMS, which allows modelling of dynamical systems at a high level without requiring detailed low-level coding experience [84]. Another example of toolchain advances is in the HDL coder toolbox from Mathworks^{TM} which allows the

user to select individual blocks within Simulink^{TM} models for coding in VHDL whilst the remaining blocks are coded in C. The tool automatically configures the interface between the FPGA and the micro-controller. A review of the toolchain and challenges of integrating these two different approaches in the same model is presented in [63].

The flexibility offered by FPGAs has been exploited in [93], to set up a rapid prototyping engine controller. The controller is developed to control the fuel system and then MPC was applied to control the temperature of the exhaust gases by varying the start of injection timing and duration of the main injection. The MPC algorithm was run on a separate PC whilst the FPGA looked after the detailed control of the fueling.

The use of FPGAs for accelerating the MPC algorithms has been studied in [56]. The authors demonstrate a parallel implementation of MPC on a FPGA chip with a 25 MHz clock achieving performance comparable to that achieved on a Pentium 3 GHz PC.

Graphics processor units (GPU) are also of interest for accelerating computations and may be suited to the solution of MPC problems. Potential challenges associated with GPUs include a non-deterministic scheduler and lack of maturity of embedded solutions. For engine applications, it is essential that the optimisation is bounded and deterministic since the engine will not wait for the next control input. GPUs have been used in PC applications where they can be called directly from Matlab for example. However for this research, we need deterministic embedded solutions to control an engine.

The software solvers for MPC have also been considered for optimisation, in particular [90] investigated improvements to the minimisation algorithms to take advantage of the particular structure of the MPC problem. This research has been successful in solving MPC control problems 100 times faster than generic solvers, allowing the application of MPC to be extended to systems at kilohertz update rates.

1.5 Thesis outline and objective

Current engine control approaches tend to be based largely on steady-state calibrations, with an ad-hoc treatment of the transient responses of the systems. Some progress has been made through the control of oxygen concentration or EGR rate, however there is still scope for a more complete solution considering the transient manoeuvres as a whole.

The application of MPC to engine control problems has met with limited success due to the challenges of limited memory and processor power within the engine control units. Early applications have sought to linearise models or apply piecewise affine models to

approximate the non-linear characteristics of the engine. The linearisation of the engine model often limits the benefits to be gained from MPC since the linearisation process introduces approximations and more calibration stages. The applications of MPC have centred around the aftertreatment system which has a relatively slow response (allowing larger time-steps for the MPC implementation) and the air system control which features a faster response system and corresponding challenges in implementation. For the air system, the explicit off-line MPC has been applied with some success. Moving to the combustion process at the centre of the internal combustion engine, there has been little research carried out in applying model based approaches to controlling the combustion processes.

The objective of this research is to investigate a new approach for engine control, applicable to transient manoeuvres. This is addressed by developing 'control oriented' models of combustion processes and the subsequently integrating these models into both a model predictive approach for engine control and a model based optimisation of the combustion process.

An outline of the thesis is presented in Figure 1.13.

1.6 Related publications

The work presented in this thesis has been presented at a number of conferences and in one journal paper:

- The 2012, 11th International Symposium on Advanced Vehicle Control, [34],
- The 2012 IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling, [33], and
- The 2014 SAE World Congress, [35].
- The SAE International Journal of Engines, [36], published July 2014.

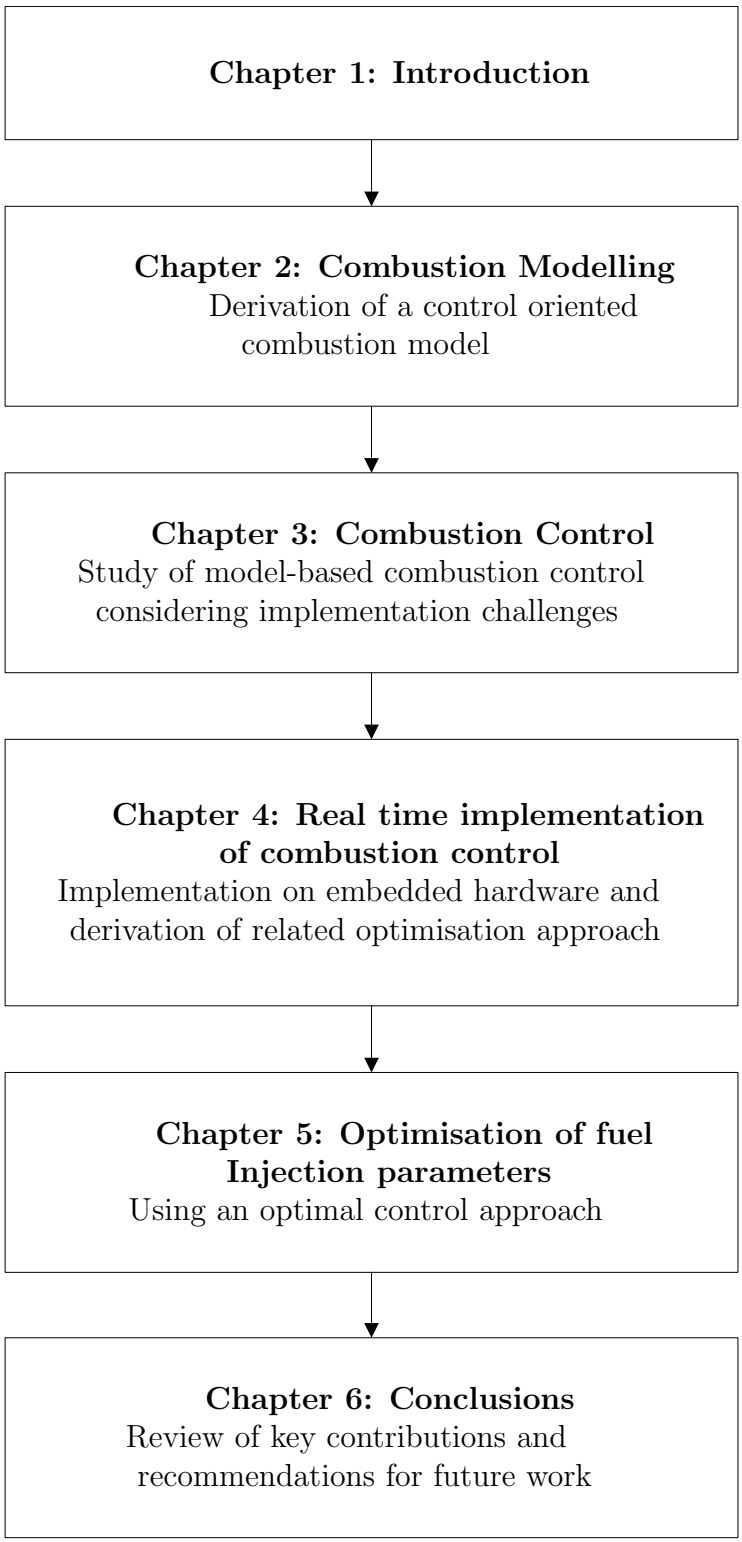

Figure 1.13: Thesis outline

Chapter 2

Combustion modelling

2.1 Objective for combustion model

The objective of this chapter is to develop a control orientated model of the internal combustion engine combustion processes which can be included in a predictive controller and embedded within an automotive control unit.

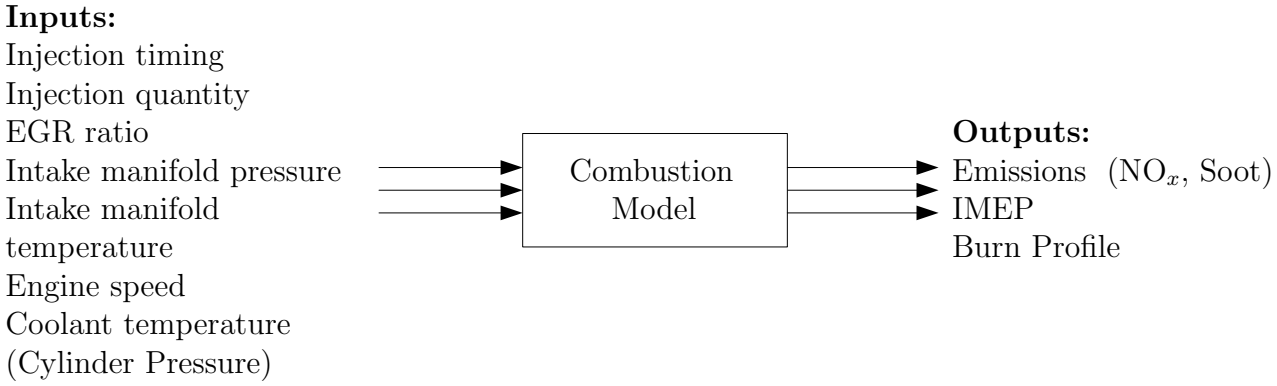

Figure 2.1: Input and outputs for combustion model

It is appropriate to define the requirements for combustion modelling at this stage to help judge which modelling approach is best suited to this research. The control orientated model should have inputs, which are available in a typical engine control unit, as shown in Figure 2.1. The cylinder pressure sensor may or may not be standard in the future, so this research will consider the case without cylinder pressure sensors as being more general. Should cylinder pressure sensors be available in the future, they may be used to provide additional information to the models.

A control orientated model should predict the response of the combustion system to the key control inputs. Whilst it is desirable to estimate the responses accurately, the trends are of more value than the absolute values. This means that the validation will be

based on the response of the combustion system to varying inputs rather than absolute predictions of emissions.

2.2 Literature study: Combustion modelling

Combustion processes in internal combustion engines have been studied for many decades, [44] and [76]. As outlined in Section 1.2.1, these processes are complex and would appear too challenging to model. The combustion measurements have used techniques including high-speed photography and laser measurements. The analysis of these measurements has given rise to a suite of models commonly known as phenomenological models which identify the controlling physical and chemical processes and describe them with a combination of physically based models and empirical relationships. The phenomenological models range from simple, empirical based zero dimensional models through to three dimensional computational fluid dynamics models (3D CFD), which enable the simulations to capture specific local conditions within the cylinders. For example, if emissions of soot are required, the spatial resolution of the combustion process can be significant.

Care must be taken to ensure that the empirical parts of these models are continually updated to reflect changes in engine technology, for example combustion system design and injection profiles. This has been particularly relevant in the last 10 years with the advent of multiple injections and highly pre-mixed combustion systems.

The physical processes which are either modelled or estimated empirically include the following:

- Fuel spray into cylinder;
- Entrainment of air into spray and turbulent mixing;
- Evaporation of the fuel;
- Chemical reactions - combustion and formation of harmful emissions;
- Piston motion, valve opening and closing;

The inputs to the model can be divided into two groups by considering an engine cycle, see [45]. During the induction stroke, a mixture of fresh air and recirculated exhaust gas (EGR) is drawn into the cylinder. Once the intake valve closes, at inlet valve closing (IVC), the mixture is 'frozen' and the actuators which control the air mixture, or

air path, have no more control authority. This leaves just the fuel injection parameters as the remaining control inputs, since most Diesel engines have fuel injection directly into the cylinder. The air path actuators can be considered 'slow' since there is typically a delay dynamic between the actuator and the air being drawn into the cylinder, whilst the fuel path actuators can be considered 'fast' since they act on the current cycle and also have limited influence on subsequent cycles (assuming little internal EGR). This representation is illustrated in Figure 2.2.

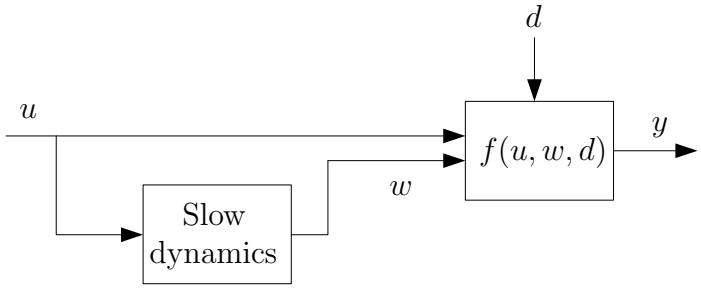

Figure 2.2: The inputs to the combustion model can be divided into 'fast' inputs which act directly on the current engine cycle and 'slow' inputs which introduce a slow dynamic.

The model of combustion is made up of several chemical, physical and thermodynamic processes which can be grouped into two phases: compression and combustion. The compression phase extends from IVC through to the start of injection (SOI) and the combustion phase is from SOI through to the exhaust valve opening (EVO). The combustion phase can be broken down further into several stages: ignition delay, pre-mixed combustion and mixing controlled combustion, Section 1.2.1. The processes can be modelled with differential equations of states (variables used to represent the system), initial conditions (which may depend on preceding processes) and a termination, or transition, condition to pass onto the next process. Figure 2.3 shows the consecutive processes for the combustion model.

A convenient form of the model would be to have n phases (in this case four phases: compression, ignition delay, pre-mixed combustion and mixing controlled combustion) and a state vector, x, which includes all the state variables. The i_{th} phase starts at angle θ_i and is characterised by a differential equation of the states $\frac{dx}{d\theta} = f_i(x, \theta)$. The initial conditions, $x(\theta_i)$, are defined by continuity between each phase. The transitions defined by transition equations $h(x_i) = 0$.

The combustion model can be represented as a sequence of differential equations with the initial conditions of the first phase dependent on the 'slow' input variables (EGR

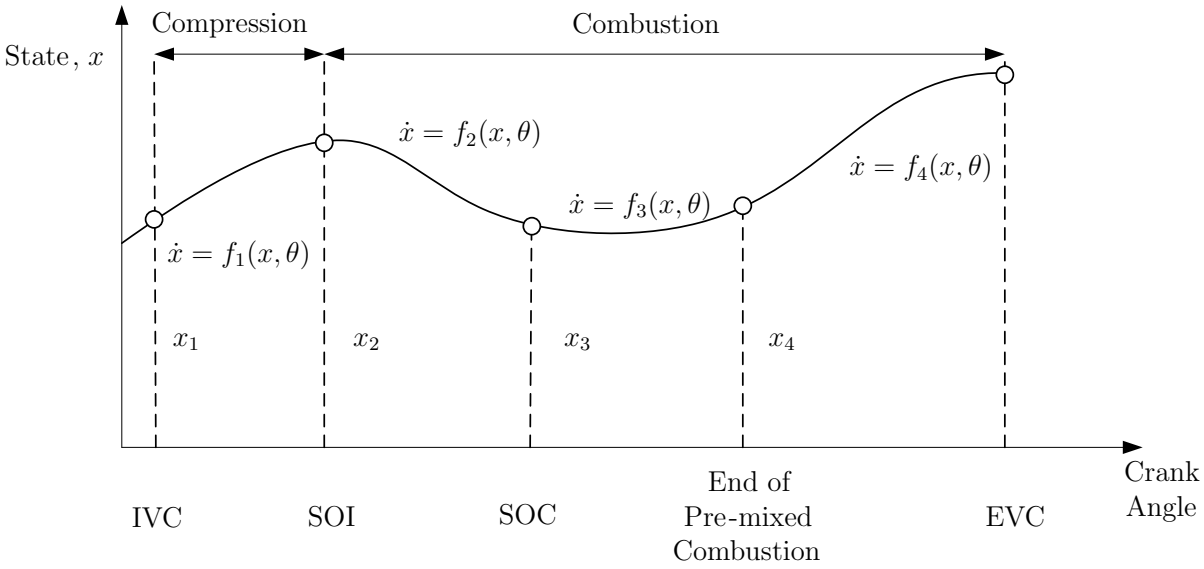

Figure 2.3: The model can be divided into separate processes, each with its own differential equation relationships, initial conditions and termination condition.

rate and boost pressure), with the remaining initial conditions defined by continuity. The first transition is defined by the 'fast' input variable (start of injection) with the remaining transitions defined by transition equations.

The formulation of the differential equations is now considered by reviewing a selection of different modelling approaches - starting with the simplest and moving on to the more complex simulations.

2.2.1 Wiebe heat release model

The Wiebe heat release model originates from early work carried out in Russia by Ivanovitch Wiebe, [39], who derives a general macroscopic reaction rate from the theory of chain reactions. In general, reaction rates are presented as either the mass burn fraction, x, or its derivative, the burn rate. A non-dimensional burn fraction x can be defined as follows:

$$x(t) = \frac{N_0 - N(t)}{N_0} \tag{2.1}$$

where $N(t)$ is the number of molecules of the main reactants at time t and N_0 is the initial number of molecules of the main reactant.

At the start of the combustion as each molecule reacts, it creates highly reactive intermediate species comprising free atoms and radicals. If there are sufficient numbers of these reactive species, then the combustion starts. Wiebe postulated that the reduction

in the number of molecules of the main reactants $N(t)$ is proportional to the increase in the number of reactive species $N_e(t)$:

$$-\frac{dN(t)}{dt} = n \cdot \frac{dN_e(t)}{dt} \tag{2.2}$$

where n is a constant.

He also assumed the rate of change of reactive species was proportional to $N(t)$ and time to the power m, which he called the combustion characteristic exponent.

$$\frac{dN_e(t)}{dt} = k \cdot N(t) \cdot t^m \tag{2.3}$$

Substitution of (2.3) into (2.2) followed by integration gives:

$$N(t) = N_0 e^{\frac{-nkt^{m+1}}{m+1}} \tag{2.4}$$

Substitute (2.4) into (2.1) gives:

$$x(t) = 1 - e^{Ct^{m+1}} \tag{2.5}$$

where $C = -nk/(m+1)$. Equation (2.5) is known as the Wiebe heat release model.

An example application of this model for the simulation of Diesel combustion is detailed in [92]. Here the overall thermal effects of the combustion process are summarised by an apparent heat release rate (AHRR). The heat release from the engine combustion has been calculated from experimental cylinder pressure diagrams. However, the experimental results are of limited use for simulation of new engines and conditions, so the authors have correlated a number of algebraic relations for the AHRR to the main control parameters such as ignition delay and equivalence ratio. Whilst this is based on a limited amount of engine data, these relations have subsequently been used extensively and found to be useful in engine design and development over many years.

The correlation is carried out by taking a standard Wiebe function, and correlating it to the measured combustion profile. The Diesel combustion is simplified into two stages, premixed and diffusion combustion so two correlation functions are used, one for each stage. They are both assumed to begin at the start of combustion and the heat release is calculated from the sum of the two functions.

2.2.2 Mixing controlled combustion

The introduction of high pressure common rail injection systems led to the requirement to update empirical combustion models that had been based on lower injection pressures. In [22] the authors derive two functions from measured data. The first function is derived from the assumption that the heat release should be proportional to the available fuel. This proves inadequate to describe the combustion processes being studied so they call on the observation that heat release can be dependent on the charge turbulence and add a simple model of turbulence which is correlated to the heat release. They call this a Mixing Controlled Combustion relationship for calculating the rate of heat release:

$$\frac{dQ}{dt} = C_{mod} f_1(m_f^a, Q) f_2(k, V) \tag{2.6}$$

where C_{mod} is an empirical constant, f_1 and f_2 are empirical functions of the available fuel, m_f^a, the cumulative heat release, Q, the total turbulent kinetic energy density, k, and the cylinder volume, V.

The function f_1 is derived from the assumption that the rate of heat release should be proportional to the available fuel. The available fuel can be written as the mass of fuel available, m_f^a, minus the fuel which has been burnt. The fuel which has been burnt is proportional to the cumulative heat release, Q. Hence f_1 is a function of m_f^a and Q.

It is well known that turbulence plays a strong part in defining the combustion processes and the authors of [22] seek to introduce a turbulence term to the heat release relationship. In Diesel engines, there are three potential sources of turbulence:

- From the air motion as it enters the cylinder, known as swirl flow;
- From the air motion as the cylinder approaches top dead centre and forces the air into the combustion bowl, known as squish flow;
- From the fuel injection spray interacting with the air in the cylinder;

In the engine studied in [22], the swirl and squish kinetic energies (KE) were estimated to be approximately 50 times smaller than the KE of the fuel spray and therefore neglected. The turbulent energy in the spray is assumed to be a function of the injected fuel, ie $k = f(m_f^a)$. The turbulence model then needs to consider dissipation of the KE into the mixture. For a zero-dimensional model, the authors recognise that the traditional approaches for estimating the KE dissipation are too complex and they propose

that the reaction rate is proportional to the square root of the turbulent KE density, k. They also propose that the collision of the molecules is proportional to the density and therefore refer the dissipation to a characteristic length, in this case the cube root of the cylinder volume. Finally they propose an exponential form for f_2 to be consistent with experimental observations:

$$f_2(k, V) = e^{C_{rate} \frac{\sqrt{k}}{\sqrt[3]{V}}} \tag{2.7}$$

where C_{rate} is a constant.

2.2.3 Pre-mixed and mixing controlled combustion

The introduction of Common Rail injection systems has allowed greater flexibility in the fuel injection profiles, allowing multiple injections per engine cycle. The multiple injections gave new heat release profiles which in turn drove updates to the combustion models. In [14], a pre-mixed combustion model is detailed and added to the mixing controlled combustion. The pre-mixed combustion model is shown in Figure 2.4 and has the following stages:

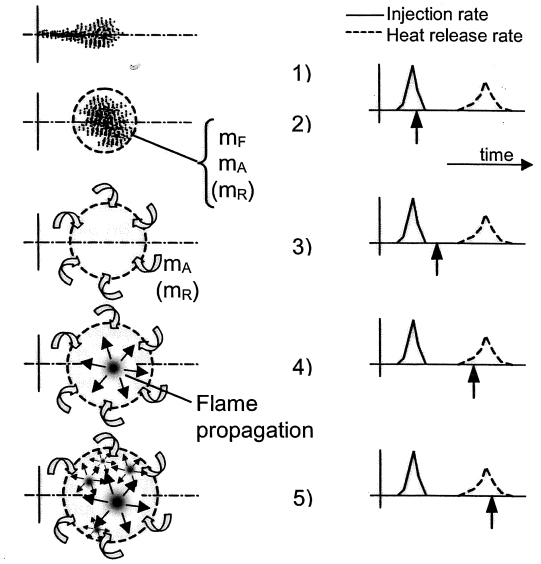

Figure 2.4: Time sequence of pre-mixed combustion model [14]. Reprinted with permission from SAE Paper No.2000-01-2933 ©2000 SAE International. Further use or distribution of this material is not permitted without prior permission from SAE.

- The fuel from each injection forms one pre-mixed zone;
- The spray entrains air into this zone;

- At the end of the injection, more air is mixed into the zone by the turbulence in the cylinder;
- Combustion starts at one location and the heat release increases with respect to time;
- Combustion is limited by mass of fuel and reduces with respect to time.

This model also includes a consideration of residual gases, either left in the cylinder or recirculated from the exhaust (EGR).

2.2.4 Simple spray model and Arrhenius reactions

The next model introduces a degree of physical modelling by combining a simplified one-dimensional spray model and a zero-dimensional burn model. This approach is described in [23] and can be used to calculate the burn rate with multiple injections.

A spray model based on conservation equations is used to give an algebraic equation for the penetration of the spray, which leads to a calculation of the entrained air. The aim of the spray modelling is to establish how much fuel has been mixed below a certain equivalence ratio. The paper proposes an approach to calculate the fraction of fuel below a certain equivalence ratio by coupling average spray values with the steady spray description.

Comparison to spray models from complex 3D CFD simulations show these approximations are reasonable. However it should be highlighted that these are for a large Diesel engine where the spray does not hit the walls or piston.

The combustion of the fuel is assumed to follow the stages described in Section 1.2.1; following the ignition delay, there is pre-mixed combustion, spray combustion and a burnout phase. The total heat release is the sum of the individual burn rates from each of these stages.

The authors of [23] propose the pre-mixed combustion follows a temperature-dependent Arrhenius law:

$$\frac{dQ}{dt} = C_{pre} m_{pre}^{a} e^{\frac{-kT_i}{T}} Q_{LHV} (t - t_{SOC})^2 \tag{2.8}$$

where Q is the heat release, C_{pre} is a constant associated with pre-mixed combustion, m_{pre}^{a} is the mass of fuel available for pre-mixed combustion, k is a constant, T_i is the

activation temperature, T is the temperature, Q_{LHV} is the lower heating value (referring to the case where the H_2O in the reaction products is a vapour), t is time and t_{SOC} is the time at the start of combustion. The quadratic term containing the elapsed time was added to the more familiar Arrhenius law following comparison to measured data and was based on an approach used in Gasoline engines where the combustion may be controlled by the progress of a flame front.

During the spray combustion the heat release is simply proportional to the fuel available for combustion - ie the fuel that is at the correct equivalence ratio. During the burnout combustion, the spray is no longer present and a simpler mixing controlled model is used.

To summarise, this approach includes the modelling of the different stages of combustion and a physical representation of the spray. It has been validated on two different engines for multiple injections. However the treatment of the mixing is at a macro level and there is no EGR, making the model in its current form less suitable for prediction of emissions.

2.2.5 Stratified spray model with detailed chemical reactions

A more detailed spray model has been proposed in [26] which allows for the spray to be considered as a stratified mixture of fuel and air. This is in contrast to the previous model, Section 2.2.4, which had a spray with two zones, above and below a critical equivalence ratio. The stratified spray model has then been coupled with detailed chemistry models of the key species within the cylinder which are stored in the form of look-up tables. The model starts with a similar spray penetration model as in Section 2.2.4, with additional detail regarding the evaporation model of the fuel and the turbulent mixing.

The spray is considered as a turbulent reactive flow which has been studied in detail in [73] where the flow is described using probability density functions (PDF) to give the overall evolution of the mixing processes. In [73], a solution based on Monte Carlo simulations is proposed, which can be computationally intensive and is described in Section 2.2.6. In the model proposed in [26] the PDF is modelled by assuming a characteristic shape, the β distribution, which is characterised by two parameters, α and β. The β distribution is defined by:

$$P(Z;\alpha,\beta) = \frac{Z^{\alpha-1}(1-Z)^{\beta-1}}{B(\alpha,\beta)} \tag{2.9}$$

where Z is the independent variable (in the spray example this is the mixture fraction, see Equation 2.22) and $B(\alpha, \beta)$ is the Euler β function defined by:

$$B(\alpha, \beta) = \int_0^1 y^{\alpha-1}(1-y).^{\beta-1}dy \tag{2.10}$$

Further details of the β distribution can be found in [13], [41] and [91].

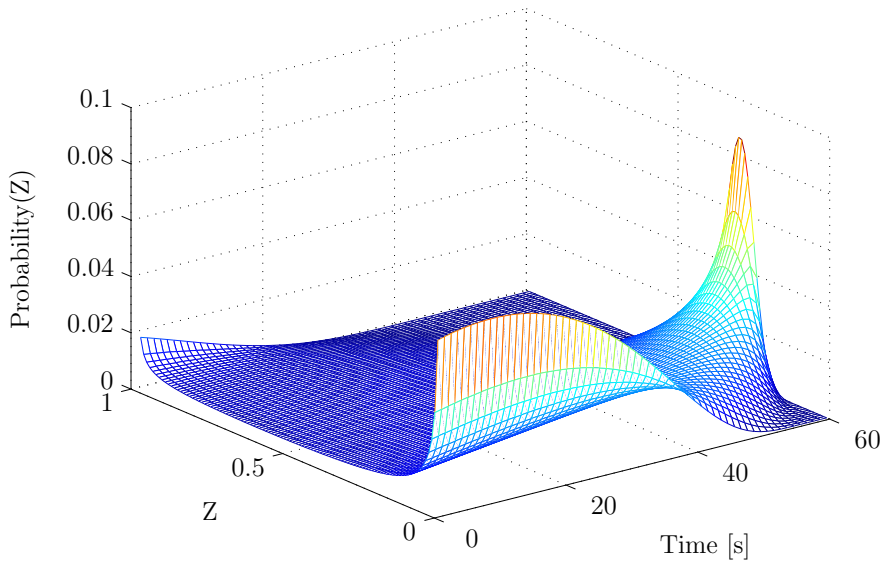

Figure 2.5: An example β distribution showing the evolution from two peaks at $t = 0$ (corresponding to pure fuel and pure air) through to the mixed state with the majority of the spray has the fuel/air ratio of the uniformly mixed gases. Where Z is the mixture fraction variable.

Stratification of the mixture is achieved efficiently by assuming that the distribution of the fuel and air mixture follows a given β distribution. An example of the evolution of a β distribution is given in Figure 2.5, where a distribution starts with peaks at 0 and 1 corresponding to pure air and pure fuel respectively. As time progresses it mixes and ends up with the peak at 0.3 which corresponds to a mixture of 30 % fuel. The mixture fraction variable, Z, is defined in Equation 2.22 where $Z = 1$ corresponds to pure fuel and $Z = 0$ corresponds to pure air with values in between corresponding to mixtures of fuel and air.

Returning to the combustion approach used in [26], the combustion reactions are shown to depend on five variables: initial pressure, initial temperature, equivalence ratio, EGR ratio and a progress variable which details progress through the combustion. An off-line calculation is used to populate a look-up table by varying each of the first four variables across its operating range and for each combination, a detailed chemical

kinetics program is run to give the evolution of the progress variable with time. The reaction rates of each of the key species is tabulated and saved in a large look-up table ($\approx$ 13000 operating points) which is accessed by the simulation at each time-step.

The stratified spray model is combined with the tabulated chemistry data to calculate the overall heat release. The validation included comparison to 3D CFD simulations and experimental data operating both in conventional combustion modes and also in Homogeneous Charge Compression Ignition (HCCI) mode.

2.2.6 Spray model with stochastic reactor model (SRM)

The spray model can be extended further by using a Monte Carlo approach for solving the turbulent mixing problem and solving the chemical reactions on-line. This complex simulation approach has been developed by several teams with differing assumptions and implementations.

The basis for these approaches can be found in [73], where the PDF is introduced as a representation for turbulent reacting flows. The author derives an important equation for the evolution of the joint PDF which shows that the equations for the reaction, pressure gradient, buoyancy and transport are closed and do not need to be approximated, leaving the mixing terms which are modelled with a stochastic approach.

An application to Diesel sprays was introduced in [27] which gives more detail on the Monte Carlo implementation followed by validation to constant volume experiments. The simulation was run with 600 cells each with 1000 particles.

The Monte Carlo mixing models were then combined with the chemical kinetic models to give a Stochastic Reactor Model (SRM) which is currently the state of the art simulation approach for combustion modelling, see [83] for a comprehensive explanation of this approach.

2.2.7 Emissions prediction

The key emissions from Diesel engines are the nitrogen oxides (NO_x) and particulate matter (PM). The NO emissions are a often significant proportion (typically 80 to 90 % in [44]) of the NO_x emissions and are used in this study as an estimate of the NO_x emissions. In this thesis, the predicted emissions are referred to as NO emissions and measured emissions are NO_x emissions. Particulate matter is made up from soot (carbon

particles) together with absorbed hydrocarbons. In the literature, the emissions models are aimed at estimating the soot emissions so this approach is adopted in this work.

NO emissions

The prediction of the emissions from combustion requires two coupled models, the turbulent combustion and the chemical reactions. Emissions models have a broad range in complexity, from the simple model based on the extended Zeldovich equations, see [44] and (2.11), through to fully coupled CFD models with chemical mechanisms. The latter are very expensive in terms of computational effort and are therefore outside the scope of this study. Simplifications to the fully coupled model have been studied in literature, for example the Flame Prolongation of Intrinsic low dimensional manifolds (FPI) discussed in [71] and the tabulated chemistry in [26]. These have produced significant reductions in computation effort, however they are still expensive for an embedded application.

During combustion there are many different chemical species. The complexity of the model is often defined by how many of these reactions are modelled and simulated at once, for example the authors in [94] study a mechanism with 19 reactions. It is generally accepted, see [91], that the NO emissions are produced by four groups of equations: the 'thermal' or 'Zeldovich' NO mechanism, the 'prompt' or 'Fenimore' NO mechanism, the N_2O route and the fuel bound nitrogen route. The Zeldovich equations (2.11) are known as the thermal NO mechanism because the first reaction has a very high activation energy and is only active at higher temperatures. At lower temperatures it is the limiting step in this mechanism.

$$\begin{aligned} O + N_2 &\rightleftharpoons NO + N \\ N + O_2 &\rightleftharpoons NO + O \\ N + OH &\rightleftharpoons NO + H \end{aligned} \tag{2.11}$$

The prompt mechanism results from the radical CH which reacts with the nitrogen in the air producing HCN which then reacts further to produce NO. This reaction is more active in rich conditions and lower temperatures which means it is usually less significant than the Zeldovich mechanism for Diesel combustion, which is usually lean. The N_2O route and the fuel bound routes are insignificant contributors to total NO for Diesel combustion, the fuel bound route being of more interest to the combustion of coal.

Taking the Zeldovich equations and applying to a *heterogeneous* mixture in the combustion chamber offers a potential improvement over just using homogeneous values, indeed whilst the Zeldovich equations are often cited, the temperature used in calculating the rates is rarely discussed. In particular, the choice of whether to use a bulk temperature across the combustion chamber or a local temperature distribution which would capture the heterogeneous nature of some combustion modes. In [69] a model of local temperatures is proposed and the concentrations tracked for each mass parcel, allowing the model to capture the high local temperatures around the flame and hence improve the NO emission predictions, see Figure 1.12.

Soot emissions prediction

The soot emissions models, which are considered here, are restricted to simplified models which are numerically efficient. Soot is principally formed from incomplete combustion in rich areas of the spray and most of the soot formed during the combustion is subsequently oxidised before the exhaust valve opens. Many of the approaches in literature are based on the work presented by Hiroyasu in [46] that considers both the soot formation and soot oxidation processes. From an understanding of the principal controlling factors for the soot emissions, the two processes are assumed to follow Arrhenius equations given in Equation 2.12. The difference between the formation and oxidation gives the rate of change of soot and can be integrated to give a prediction for the soot emissions produced by the engine. Since most of the soot is oxidised before exhausting, this model can be difficult to calibrate and is sensitive to the relative difference between the reaction rates.

$$\begin{aligned} \frac{dm_{soot}}{dt} &= \frac{dm_{sootf}}{dt} - \frac{dm_{sooto}}{dt} \\ \frac{dm_{sootf}}{dt} &= A_f \cdot m_f \cdot p^{0.5} \cdot e^{\frac{E_{sf}}{RT}} \\ \frac{dm_{sooto}}{dt} &= A_o \cdot m_{soot} \cdot \frac{p_{O_2}}{p} \cdot p^{1.8} \cdot e^{\frac{E_{so}}{RT}} \end{aligned} \tag{2.12}$$

where m_{soot} is the mass of soot, m_{sootf} and m_{sooto} are the mass of soot forming and oxidising respectively, E_{sf}, E_{so}, A_f and A_o are constants taken from [43], p is the cylinder pressure and p_{O_2} is the partial pressure of O_2.

An update to the oxidation reaction from Equation 2.12 was described in [43], using an empirical relationship called the Nagel and Strickland-Constable (NSC) approxima-

tion which was based on oxidation experiments of carbon graphite in varying atmospheric conditions, [65].

$$\frac{dm_{sooto}}{dt} = \frac{6 \cdot W_{NSC} \cdot m_{soot}}{\rho_{soot} \cdot D_{nom}} \tag{2.13}$$

where W_{NSC} is the NSC soot mass oxidation rate per unit surface area, D_{nom} is the nominal spherical soot particle size and ρ_{soot} is the soot density.

A simpler model is proposed in [15], where the soot formation and oxidation processes are described in three phases: formation, equilibrium and oxidation, see Figure 2.6. The formation phase is where the soot formation dominates and the cumulative soot is approximated as a linear function with time. During the equilibrium phase, formation rate is approximately equal to oxidation rate and the cumulative soot is constant. In the oxidation phase, an exponential curve is used to calculate the final soot at the end of the cycle. The exponential curve has normalised input variables including engine speed, EGR ratio and rail pressure. The model has 12 constants which are fitted to test data.

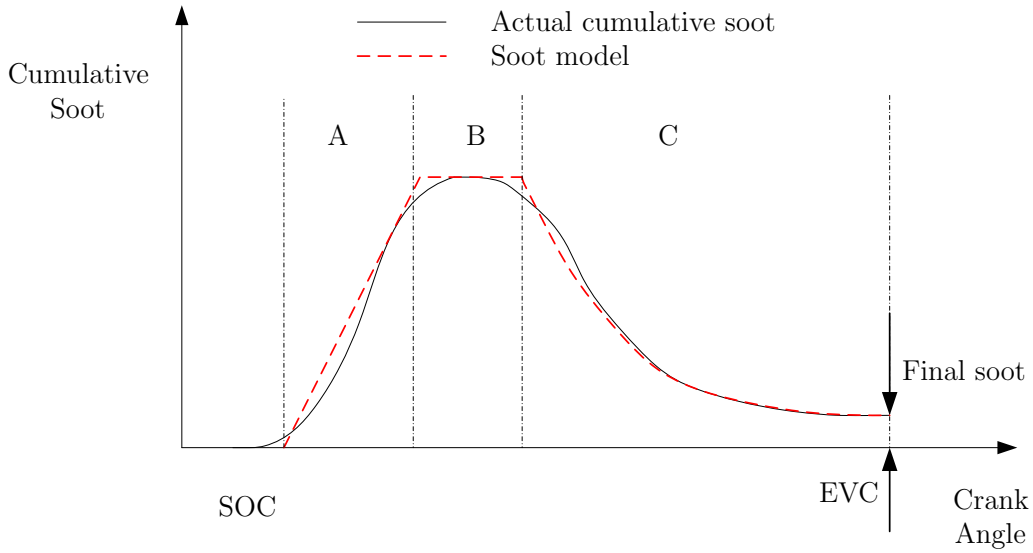

Figure 2.6: Simplified soot model after [15] with three phases: formation (A), equilibrium (B) and oxidation (C).

Further simplification can be made to give a 'black box' model which has a base soot map versus engine speed and load, calibrated at steady state operating points, [82]. The transient soot estimation is calculated by observing that there are six variables which influence soot and may be assumed to be mutually independent. With this assumption, the partial derivative for each variable is multiplied by the deviation at that point and added to the base soot map, as shown in Equation 2.14.

$$c = c_0 + \sum_{k=1}^{6} \frac{\partial c_0^k}{\partial z^k}(z^k - z_0^k) \tag{2.14}$$

where c is the estimated soot, c_0 is the value from the base soot map, z are the independent variables and z_0 is the value of the independent variable at the steady state operating point.

2.2.8 Summary of combustion models

Combustion models can be classified according to the approaches used for the different components of the model, see the table in Figure 2.7. The requirements for the model for this project, Section 2.1, have been used to select the simplest approach for each component that meets the requirements. It should be noted that all the approaches have many approximations.

- Spray model based on mass and momentum conservation - Whilst more complex models (outside this survey) have been developed, eg 3D CFD, most of the models studied here use models based on mass and momentum conservation;
- Evaporation model using the d^2 law approach - Avoids an iterative solution of evaporation equilibrium on droplet surface;
- Mixing model using β - PDF - Computationally efficient approach for capturing mixing of air and fuel vapour;
- Combustion model using Arrhenius and Magnussen laws - The chemical kinetics based approaches either require many reactions and species to be modelled or the reactions to be parameterised in look-up tables, both of which are less suited to on-line implementation. A simpler Arrhenius law is used to capture the ignition delay and pre-mixed combustion and the Magnussen law (related to turbulence) is used to capture the mixing controlled combustion. These approaches being sufficient to estimate the heat release to provide an input to the emissions prediction;
- Emissions model for NO based on extended Zeldovich reactions;
- Emissions model for soot based on Hiroyasu approach.

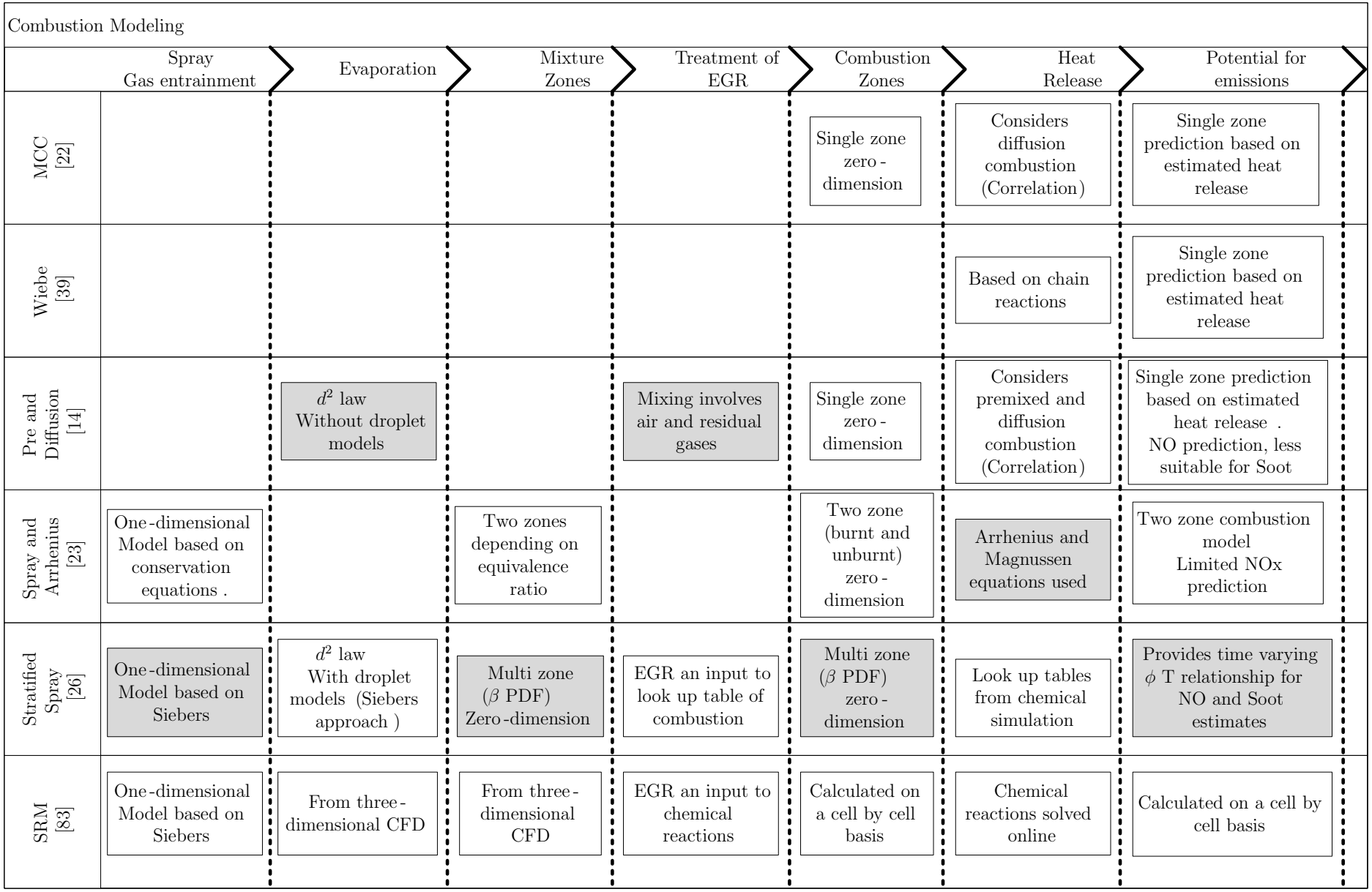

Figure 2.7: Summary of combustion modelling, with the approach adopted here highlighted in grey.

2.3 Combustion model development

In this section we will build up the combustion model components which were selected in the previous Section 2.2.8. A flow diagram of the total model is provided in Figure 2.8. The definition of each component starts with a description of the input and output variables, with additional details included in Appendix A.

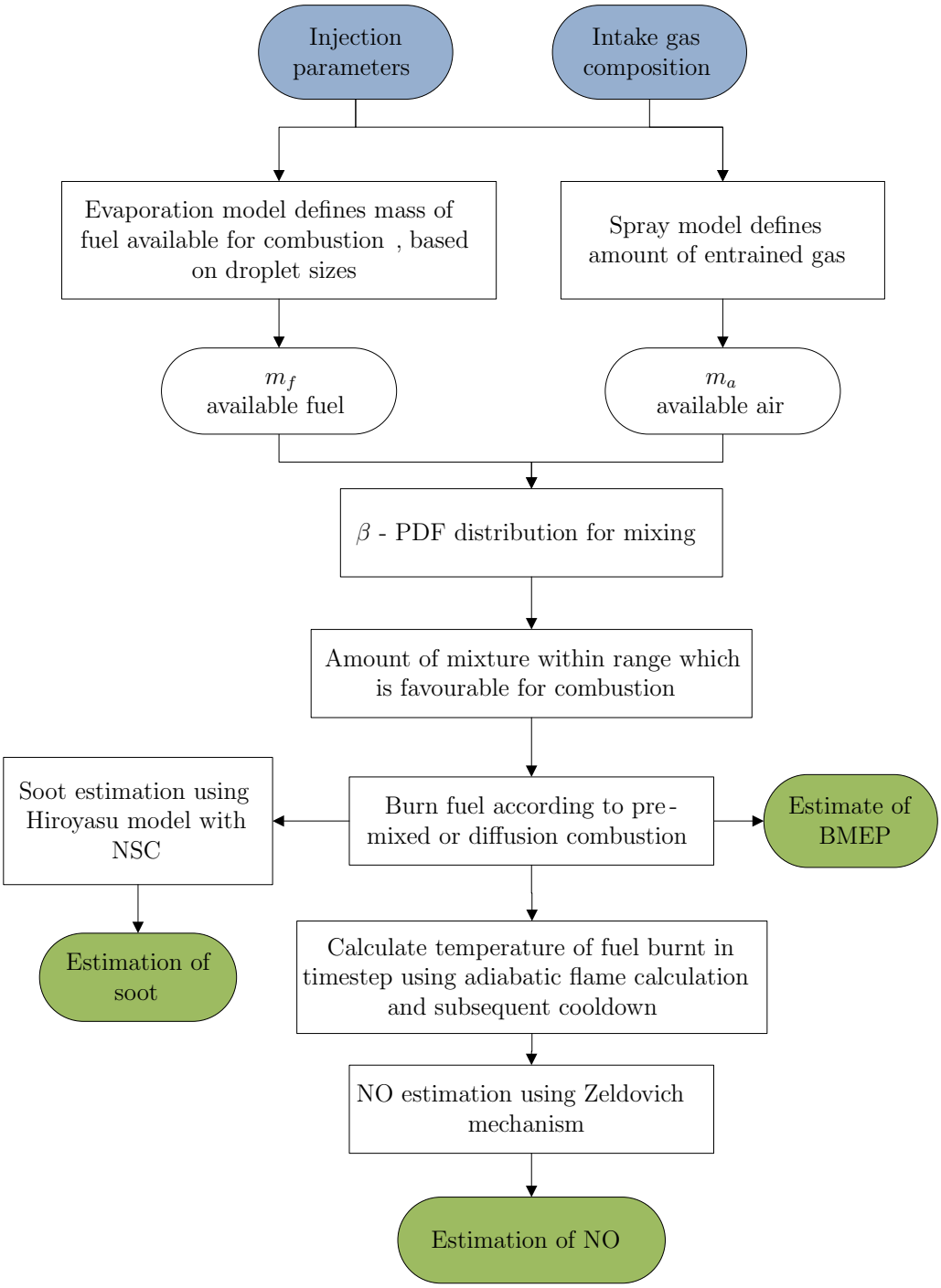

Figure 2.8: Combustion model flow chart. Blue boxes are inputs and the green boxes are the outputs from the model.

2.3.1 Spray model

Interface

Input: Injection parameters

Output: Mass of gas entrained into spray

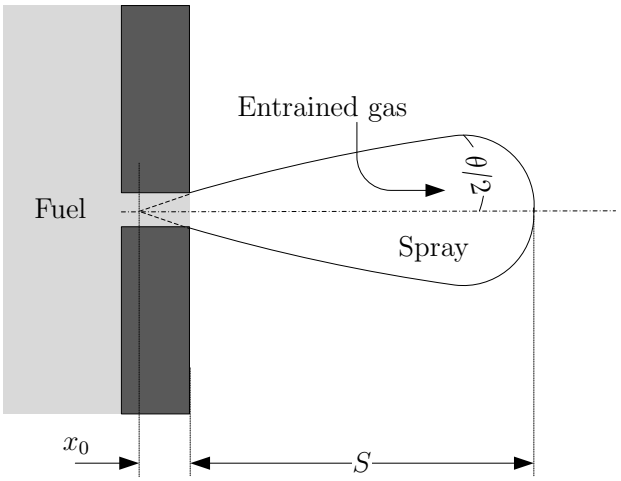

Figure 2.9: Spray schematic

The majority of the combustion models reviewed in Section 2.2 use the non-vaporising spray penetration model proposed by [64] that calculates the spray length and volume as a function of time as gas is entrained into the spray. A schematic for this model is included in Figure 2.9. This model is derived from mass and momentum conservation equations which are rearranged in Appendix A.1 to give the expressions for the spray length S and volume V_s.

$$S = \frac{\tilde{t}}{\left(1 + \tilde{t}^{\frac{n}{2}}\right)^{\frac{1}{n}}} \cdot x^{+} \tag{2.15}$$

where x^{+} is a length scale used to non-dimensionalise distance, see Equation A.7, $\tilde{t}$ is the non-dimensional time coordinate and n is a constant set to 2.2 to match measured spray data.

Assuming a cone shaped spray, the volume of the spray is given by (2.16).

$$V_s = \frac{\pi}{3} \cdot tan^2\left(\frac{\theta}{2}\right) \cdot \left[(S + x_0)^3 - x_0^3\right] \tag{2.16}$$

where x_0 is the additional length of the spray within the injector as shown in Figure 2.9 and θ is the measured spray angle, see [64].

The preceding equation has been derived for injection into a constant pressure vessel. For application to an internal combustion engine, where the fuel is injected into a varying volume with combustion, the differential form of (2.16) is used to relate the change in volume to the change in S via the chain rule:

$$\frac{dV_s}{dt} = \frac{\partial V_s}{\partial S} \cdot \frac{dS}{dt} \tag{2.17}$$

The amount of gas entrained into the spray, m_g, is calculated by integrating the rate of change of V_s multiplied by the density of the entrained gas ρ_a, assuming the increase in volume is the spray is due to the entrained gases.

$$\frac{dm_g}{dt} = \rho_a \cdot \frac{dV_s}{dt} \tag{2.18}$$

Known simplifications with this approach include the assumption that wall impingement of the spray is not significant. This is an important consideration for smaller Diesel engines where the spray is likely to hit the piston or walls. Secondly, the assumption of a non-vaporising spray has been found to over-predict the length of the spray, for a comparison with more recent spray models see [60].

2.3.2 Evaporation

Interface

Input: Injection profile

Output: Mass of vaporised fuel

The approach selected for implementation is based on the d^2 law, see [25] for derivation. The d^2 law states that the rate of change of the diameter squared is a constant, which can also be written as:

$$d^2 = d_0^2 - \beta \cdot t \tag{2.19}$$

where d is the current droplet diameter, d_0 is the initial droplet diameter, β is a constant of order $10^{-6}\,\mathrm{m}^2/\mathrm{s}$ and t is the time since injection.

The evaporation is implemented by discretising the fuel injection into parcels of fuel at crank angle steps. Each parcel of fuel is tracked during the evaporation. The injected fuel starts with an initial droplet diameter for all droplets being given by its Sauter Mean Diameter (see Equation A.2). The droplet diameter reduces over time through

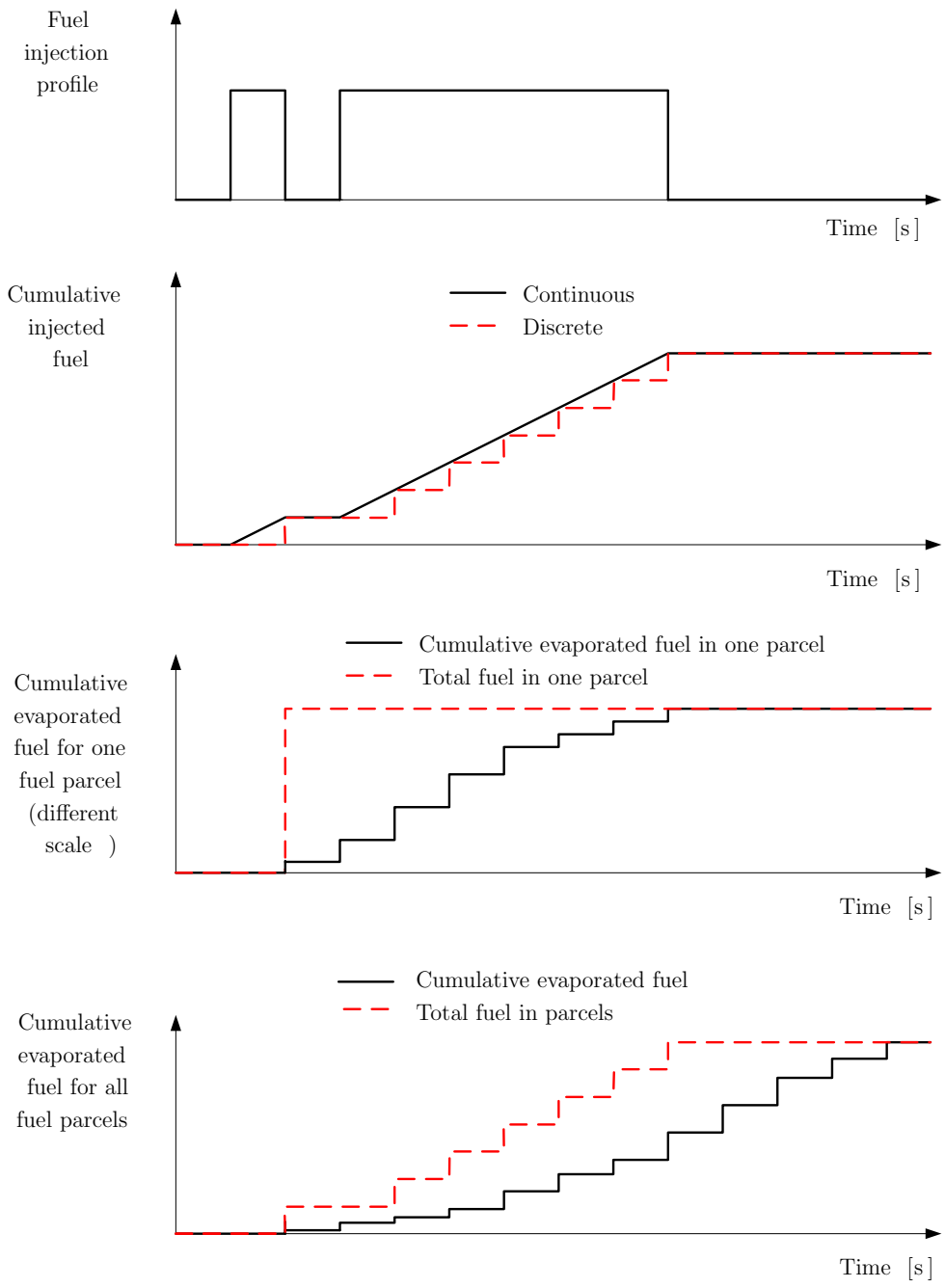

Figure 2.10: Evaporation model.

evaporation according to (2.19). The model uses (2.19) to calculate the remaining mass of liquid fuel, and calculates the mass of vaporised fuel m_{fv} by subtracting from the initial mass of fuel in the fuel parcel according to:

$$m_{fv} = m_{f0} - m_{fl} \tag{2.20}$$

where m_{f0} is the initial mass of fuel injected in the fuel parcel and m_{fl} is the mass of liquid fuel.

The fuel parcels are tracked over time and the total vaporised fuel calculated by summing over all the fuel parcels. This process is illustrated in Figure 2.10.

Equation 2.20 is discretised in Appendix A.2 to give Equation A.24 which is repeated here for convenience:

$$\mathbf{m_{fv}} = \mathbf{T}(\mathbf{m_{parv}}, \mathbf{0}) \cdot \mathbf{Z_{inj}} \tag{2.21}$$

where $\mathbf{T}(\mathbf{m_{parv}}, \mathbf{0})$ is a Toeplitz matrix with $\mathbf{m_{parv}}$ as the first column and zeros as the first row, $\mathbf{m_{parv}}$ is a vector of vapourised fuel from one fuel parcel and $\mathbf{Z_{inj}}$ is a vector defining the fuel injection profile.

2.3.3 Mixing

Interface

Input: Mass of entrained gas, vaporised fuel and burnt fuel

Output: Distribution of mixed gas and fuel

The β-PDF approach has been adopted, as outlined in [26], with a modification to include the distribution of the burnt gases within the spray. The approach in [26] is based on a non-combusting mixture model which includes combustion of the mixture through detailed chemical models and the introduction of a progress variable c to define the overall progress through combustion. The detailed chemistry models are unsuitable for consideration in this project due to their memory size and processor power requirements, so a simpler combustion model has been used, see Section 2.3.4. The modification to the β-PDF approach is new and incorporates the burnt gases into the mixture distribution.

This section starts with an introduction to the β-PDF approach, followed by an overview of the approach for tracking the evolution of the PDF as adopted by [26], where the authors derive an expression for the rate of change of the variance of the PDF. Finally, the variance equation is updated to account for the combusting mixture.

β-PDF description of mixing fluids

It is common to describe the mixing of fuel and oxidiser flows by introducing a mixture fraction variable, Z, which is defined by:

$$\begin{aligned} Y_f &= Z \cdot Y_{f,0} \\ Y_o &= (1 - Z) \cdot Y_{o,0} \end{aligned} \tag{2.22}$$

where Y_f is the fuel mass fraction at time t, $Y_{f,0}$ is the fuel mass fraction in the initial fuel stream (often equal to one), Y_o is the oxidiser mass fraction at time t and $Y_{o,0}$ is the oxidiser mass fraction in the initial oxidiser stream.

This means that at $Z = 1$, the mixture is at the same concentration as the initial fuel stream (often pure fuel) and at $Z = 0$, the mixture is at the same concentration as the initial oxidiser stream. In our case, the initial oxidiser stream can often have a proportion of re-circulated exhaust gas, X_r, which reduces the concentration of oxygen. In which case, the mass of air is given by:

$$m_a = (1 - X_r) \cdot m_g \tag{2.23}$$

The spray can then be described by associating a probability to Z. From [40] it was found that the standard beta probability distribution (β - PDF) is suitable for describing the mixture distribution in the mixing spray at a point in time. The β - PDF is a normalised statistical tool used to represent the probability $P(Z)$ of a variable Z which is defined between 0 and 1, see Equation 2.9. The distribution is described by two variables; the overall mean, $\overline{Z}$, and the variance of the distribution, $\overline{Z'^2}$, defined below.

$$\overline{Z} = \frac{1}{m_s} \int_{m_s} Z|_{dm} dm \tag{2.24}$$

$$\overline{Z'^2} = \frac{1}{m_s} \int_{m_s} (Z|_{dm} - \overline{Z})^2 dm \tag{2.25}$$

Or in terms of the probability distribution:

$$\overline{Z} = \int_0^1 Z \cdot P(Z) \cdot dZ \tag{2.26}$$

$$\overline{Z'^2} = \int_0^1 (Z - \overline{Z})^2 \cdot P(Z) \cdot dZ \tag{2.27}$$

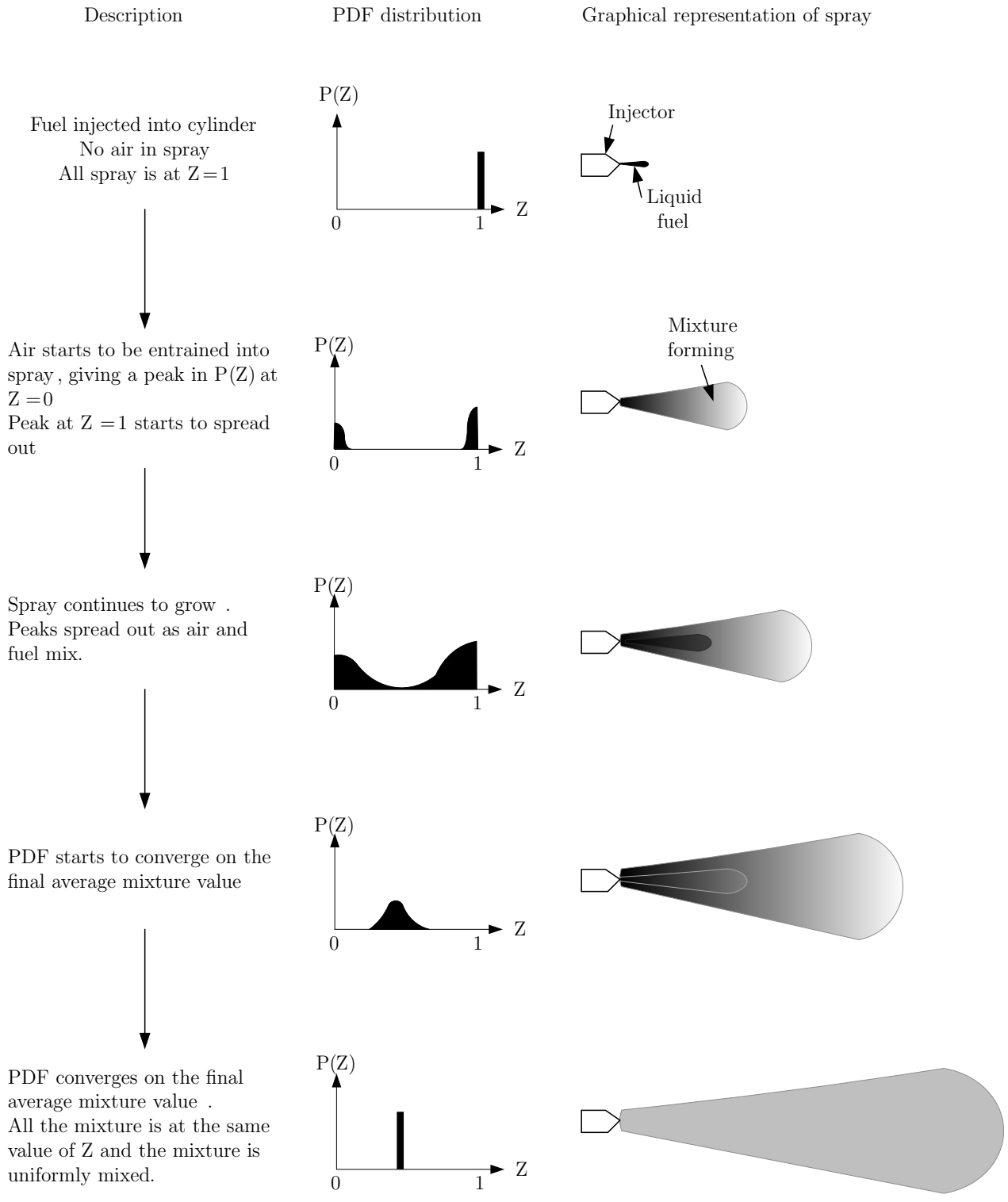

Figure 2.11: Evolution of spray with associated PDF

A normalised variance or segregation factor, δ, is introduced to allow easier interpretation of the distribution, calculated as the variance is divided by the maximum variance. The mixing process starts with $\delta = 1$ which corresponds to a completely unmixed spray through to $\delta = 0$ which corresponds to a completely homogeneous spray.

$$\delta = \frac{\overline{Z'^2}}{\overline{Z'^2}_{max}} \tag{2.28}$$

where $\overline{Z'^2}_{max}$ is the maximum value of the variance, which is equal to $\overline{Z} \cdot (1 - \overline{Z})$.

Non-combusting mixing process

Applying the β - PDF distribution to mixing process reduces the simulation of the mixing process to just the calculation of the mean and variance for the mixing of the fuel and gas. The mean of the distribution can be calculated from the overall composition of the spray given by the mass of the entrained gas in the spray, m_g, and the available vaporised fuel, m_f, derived in previous Sections 2.3.2 and 2.3.3.

$$\overline{Z} = \frac{m_f}{m_f + m_g} \tag{2.29}$$

The variance is calculated by taking the derivative of (2.27) and gathering terms to give an equation for the rate of change of the variance, (2.30) (see [26] for proof). The terms can be gathered into the contribution from the turbulence (I), the entrained mass (II) and the evaporation process (III).

$$\begin{aligned} \frac{d\overline{Z'^2}}{dt} &= \underbrace{-2 \cdot C_{diss} \cdot \frac{\overline{Z'^2}}{\frac{\tilde{K}}{\tilde{\epsilon}}}}_{I} + \underbrace{\frac{1}{m_s} \cdot \left(\overline{Z}^2 - \overline{Z'^2}\right) \cdot \frac{dm_g}{dt}}_{II} \\ &\quad + \underbrace{\frac{1}{m_s} \cdot \left(\overline{Z}^2 \cdot \left(1 - \frac{1}{Z_s}\right) + \left(Z_s - \overline{Z}\right)^2 \cdot \frac{1}{Z_s} - \overline{Z'^2}\right) \cdot \frac{dm_f}{dt}}_{III} \end{aligned} \tag{2.30}$$

Combusting mixing process

A combusting mixture needs to include the impact of the burnt gases created during combustion which modifies the evolution of the β-PDF distribution. As combustion progresses, the two processes of consuming fuel and producing burnt gases are captured in the model. The consumption of the fuel is straightforward to model as the mass of fuel can be reduced each time-step, which has the effect of reducing $\overline{Z}$ over time (2.29).

There are several potential approaches for modelling the burnt gases, this study considered whether they should stay within the 'spray' or leave the 'spray'. The study also considered how these choices would affect the subsequent combustion. During the modelling activities both approaches were investigated with the latter being favoured as it is more in-line with the literature which describes the gases around the spray being entrained into the spray,[12], [31] and [68]. In this model therefore, we have assumed that the burnt gases leave the spray and are available to be re-entrained back into the spray as shown in Figure 2.12.

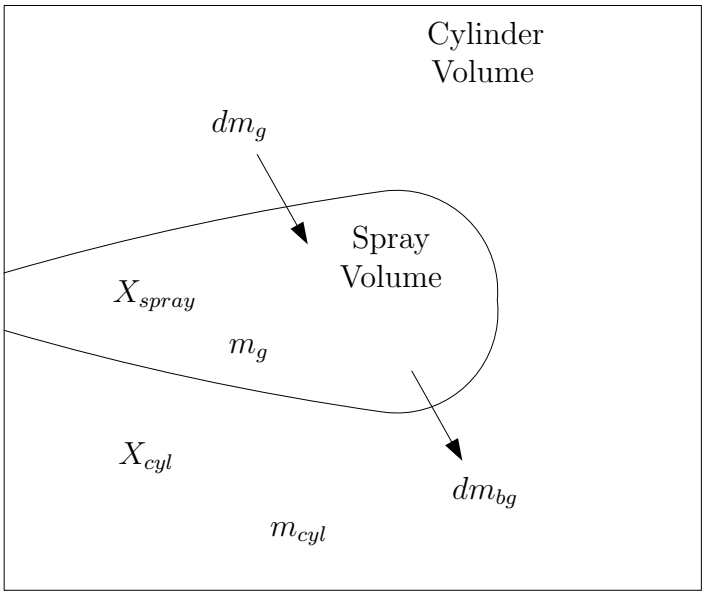

Figure 2.12: Evolution of spray showing burnt gases leaving the spray

The burnt gases are included in the gas to be entrained by augmenting the EGR ratio, X_{cyl}, to include the additional burnt gas burnt in the last time step, dm_{bg}. Where the EGR ratio, X, is defined as the mass of burnt gas divided by the total mass of gas. The EGR ratio in cylinder during combustion at time step $(i+1)$ is given by:

$$\begin{aligned} X_{cyl}(i+1) &= \frac{m_{EGRcyl}(i) + dm_{bg}(i+1)}{m_{cyl}(i+1)} \\ &= \frac{X_{cyl}(i) \cdot m_{cyl}(i) + dm_{bg}(i+1)}{m_{cyl}(i+1)} \end{aligned} \tag{2.31}$$

The corresponding EGR ratio within the spray during combustion:

$$\begin{aligned} X_{spray}(i+1) &= \frac{m_{EGRspray}(i) + dm_{EGRspray}(i+1)}{m_g(i+1)} \\ &= \frac{X_{spray}(i) \cdot m_g(i) + X_{cyl}(i+1) \cdot dm_g(i+1)}{m_g(i+1)} \end{aligned} \tag{2.32}$$

The impact of these processes on the estimation of the variance has been derived in Appendix A.3, resulting in an updated version of Equation 2.30 with the new terms (IV) given below:

$$\begin{aligned}\frac{d\overline{Z'^2}}{dt} &= \underbrace{-2 \cdot C_{diss} \cdot \frac{\overline{Z'^2}}{\frac{\tilde{K}}{\tilde{\epsilon}}}}_{I} + \underbrace{\frac{1}{m_s} \cdot \left(\overline{Z}^2 - \overline{Z'^2}\right) \cdot \frac{dm_g}{dt}}_{II} \\ &+ \underbrace{\frac{1}{m_s} \cdot \left(\overline{Z}^2 \cdot \left(1 - \frac{1}{Z_s}\right) + \left(Z_s - \overline{Z}\right)^2 \cdot \frac{1}{Z_s} - \overline{Z'^2}\right) \cdot \frac{dm_f}{dt}}_{III} \\ &+ \underbrace{\frac{1}{m_s} \left(\overline{Z'^2} \frac{\sum dm_{bg}}{dt} - \int_0^{Z_s} (Z - \overline{Z})^2 \cdot \frac{dm_{bg}(Z)}{dt} dZ\right)}_{IV} \end{aligned} \tag{2.33}$$

Integration numerically gives an estimate of the variance which can then be used in (2.28) to calculate the segregation factor, δ. Figure 2.13, shows a typical evolution of the mean and variance during the mixing process with the evolution in terms of mixture fraction and equivalence ratio which is more commonly used to describe fuel and air mixtures. Note, these values are for the spray and not the whole combustion chamber.

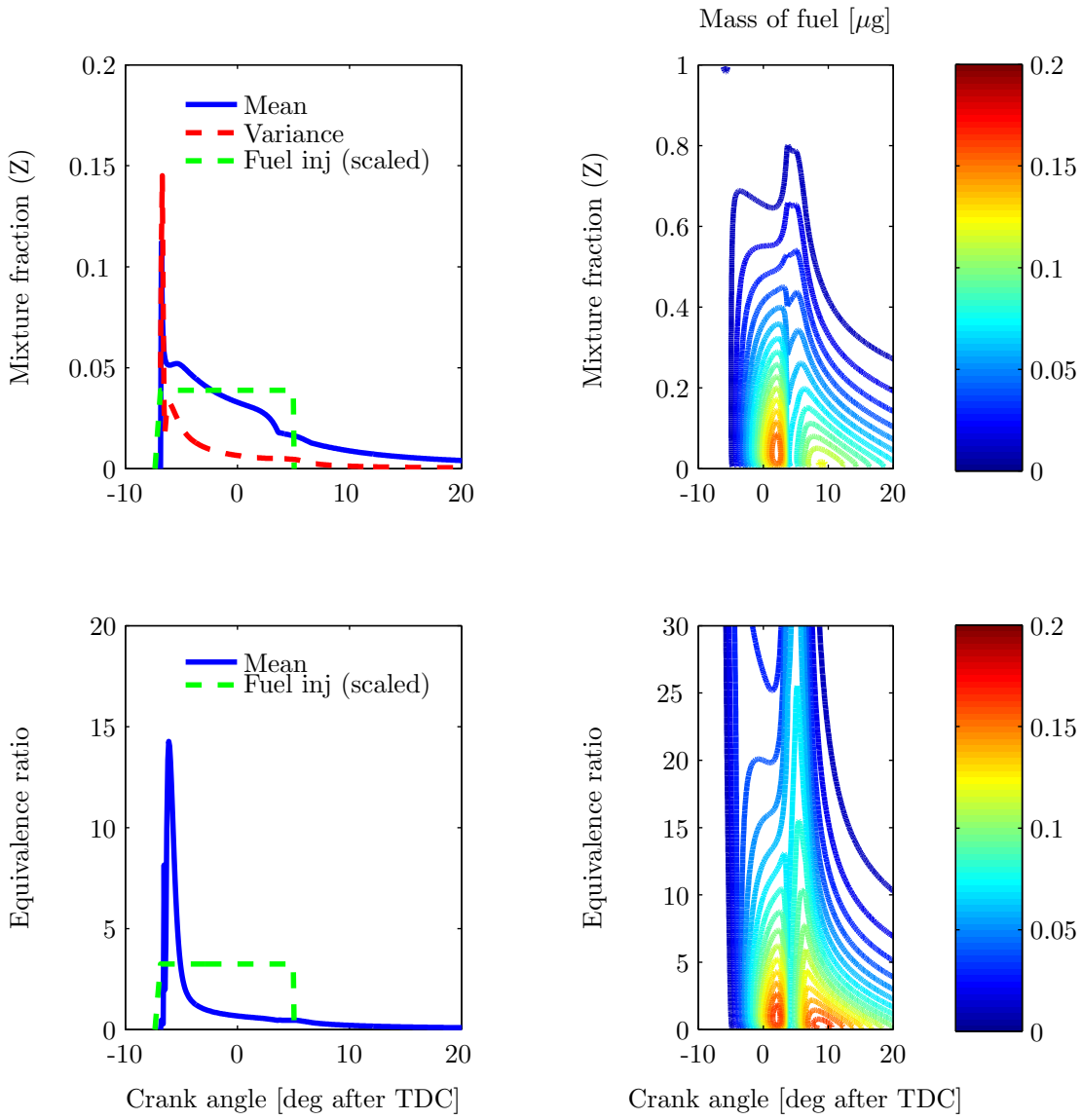

Figure 2.13: Upper plots show the evolution of the mean and variance during combustion plus the the distribution of fuel in the spray in terms of mixture fraction. The lower plots show the same data with respect to equivalence ratio.

2.3.4 Combustion

Interface

Input: Distribution of mixed air and fuel

Output: Temperature prediction in spray and engine torque

In general Diesel combustion has four phases (see Section 1.2.1): ignition delay, pre-mixed combustion, mixing controlled combustion and late combustion. In this model we will combine the last two as they can be modelled with a similar approach. Note: The mixing controlled combustion can also be split in two parts: while the injection is continuing and after the end of injection. This work will treat these two parts in the same way though a potential improvement would be to model both. The models used here are based on the work in [23] and [72].

Ignition Delay

Ignition delay is defined as the time from the start of injection to the point where the heat release increases significantly. This appears as a change in gradient in the heat release curve. Prior to ignition, physical and chemical processes transform the fuel and air mixture. The physical processes include the atomisation and vapourisation of the fuel, chemical processes include the formation of radicals. The formation of radicals can be formulated using the standard Arrhenius equation. When there are sufficient radicals, their reactions release energy to start the combustion process.

The ignition delay is calculated by integrating the formation of radicals and imposing a threshold beyond which combustion can be said to have started. This is implemented by using an Arrhenius equation for the generation of the radicals which is integrated up to start of combustion. The threshold can be calibrated for a number of examples and then the calibrated model can be applied to other situations to estimate ignition delay.

Pre-mixed combustion

During the ignition delay period, a quantity of air and fuel is mixed and is ready for combustion. Once the combustion has started, there is typically a period of combustion where the air and fuel which has been mixed during the ignition delay period, is burnt. This is known as pre-mixed combustion and has been shown to follow an Arrhenius formation, with different parameters to the ignition delay reactions.

$$r_{pre} = C_{pre} \cdot c_f \cdot c_o \cdot e^{\frac{-kT_i}{T}} \tag{2.34}$$

where r_{pre} is the reaction rate in kg/m^{3}s, C_{pre} is a constant, c_f is the concentration of fuel, c_o is the concentration of oxygen, k is a constant, T_i is an activation temperature and T is the temperature. The concentrations are defined as follows:

$$c_f = \frac{m_f}{V_s} \tag{2.35}$$

$$\begin{aligned} c_o &= \frac{m_o}{V_s} \\ &= \frac{0.232 m_a}{V_s} \end{aligned} \tag{2.36}$$

where 0.232 is the mass fraction of oxygen in air.

In [72], an additional quadratic term containing the elapsed time since the start of combustion, $(t - t_{SOC})^2$, is applied to (2.34), as described in Section 2.2.4 to give the full expression for the pre-mixed combustion:

$$r_{pre} = C_{pre} \cdot c_f \cdot c_o \cdot e^{\frac{-kT_i}{T}} (t - t_{SOC})^2 \tag{2.37}$$

Mixing controlled combustion

Once the pre-mixed mixture has been burnt, the combustion moves to what is known as diffusion combustion. During diffusion combustion the chemical reaction rate in combustion chamber conditions is much higher than the rate at which the fuel and oxygen would otherwise react. It can be assumed that the rate of fuel oxidation is determined by the rate of mixing of fuel vapour and gas and therefore by the local turbulent energy. From experience with gasoline combustion systems and based on the work of Magnussen [59], the authors in [22] have developed an expression for the rate of combustion, r_{mcc}, which is a function of turbulence:

$$r_{mcc} = C \cdot c_r \cdot \frac{\epsilon}{k} \tag{2.38}$$

where C is a constant, c_r is the either the fuel concentration or the oxygen concentration depending on how rich the mixture is at that point, ϵ is the dissipation rate and k is the turbulent kinetic energy density. In [21] the authors introduce an approximation for ϵ:

$$\epsilon = \frac{k^{1.5}}{2l} \tag{2.39}$$

where l is a characteristic length.

Substitution into (2.38) and using $\sqrt[3]{V}$ as a characteristic length gives:

$$r_{mcc} = C_{mcc} \cdot c_r \cdot \frac{\sqrt{k}}{\sqrt[3]{V}} \tag{2.40}$$

where C_{mcc} is a constant that requires fitting to data.

The model implemented here currently assumes a constant value for the kinetic energy density, k, however there are a number of expressions which could be employed to refine the model further.

2.3.5 In-cylinder conditions

Interface

Input: Mass of fuel burnt

Output: Bulk cylinder conditions and local temperature distribution

This section describes how the heat released from the combustion processes affects the in-cylinder thermodynamic conditions, resulting in the increased cylinder pressure, bulk cylinder temperature and local cylinder temperatures.

Heat release

The gross heat release can be derived from the rate of combustion calculations using the fuel lower heating value, Q_{LHV} [44]:

$$Q_{gross} = m_{fb} Q_{LHV} \tag{2.41}$$

Heat loss to the cylinder is estimated using the Woschni correlation, [44], for the heat loss coefficient:

$$Q_{loss} = h_c \cdot (T_{wall} - T) \tag{2.42}$$

$$h_c = 3.26 \cdot B^{-0.2} \cdot p^{0.8} \cdot T^{-0.55} \cdot w^{0.8} \tag{2.43}$$

where h_c is the heat transfer coefficient, B is the cylinder bore, p is the cylinder pressure (in kPa), T is the gas temperature, w is the average cylinder gas velocity and T_{wall} is the cylinder wall temperature.

Cylinder pressure

In the literature, there are many derivations of the heat release from cylinder pressure data, whereas for this model we have now calculated the heat release and need to calculate the pressure. This is carried out by re-arranging the equations for calculating the heat release. Applying the first law of thermodynamics (the change in internal energy is equal to the heat into the system minus work done) to the cylinder and assuming an ideal gas gives:

$$\frac{dQ_{net}}{dt} = p\frac{dV}{dt} + mc_v\frac{dT}{dt} \tag{2.44}$$

where c_v is the specific heat of the gas at constant volume and Q_{net} is the net heat release, $Q_{net} = Q_{gross} - Q_{loss}$.

Further manipulation gives:

$$\frac{dQ_{net}}{dt} = \frac{\gamma}{\gamma - 1}p\frac{dV}{dt} + \frac{1}{\gamma - 1}V\frac{dp}{dt} \tag{2.45}$$

where γ is the ratio of specific heats, c_p/c_v. Rearranging (2.45) gives:

$$\frac{dp}{dt} = \frac{\gamma - 1}{V}\frac{dQ_{net}}{dt} - \frac{\gamma p}{V}\frac{dV}{dt} \tag{2.46}$$

which can be integrated to give the cylinder pressure, p. The average cylinder temperature can be calculated assuming an ideal gas.

Flame temperature estimation

The temperature model can be extended to give a multi-zone temperature history. The temperature model is based on the model proposed in [69] and Figure 1.12 where the parcel of fuel burnt in each time-step, m^i_{fb}, tracked over time. Initially m^i_{fb} is raised to the flame temperature (derived assuming an adiabatic flame temperature and implemented with a look up table). The temperature reduces from the initial flame temperature as energy is radiated to the cylinder. It has been found that the temperature reduction can be parameterised simply as a function of lambda (see [57]). The temperature of each parcel then evolves according to a polytropic relationship with γ a function of bulk temperature [44]. The temperature distribution across the mixture evolves as shown in Figure 2.14.

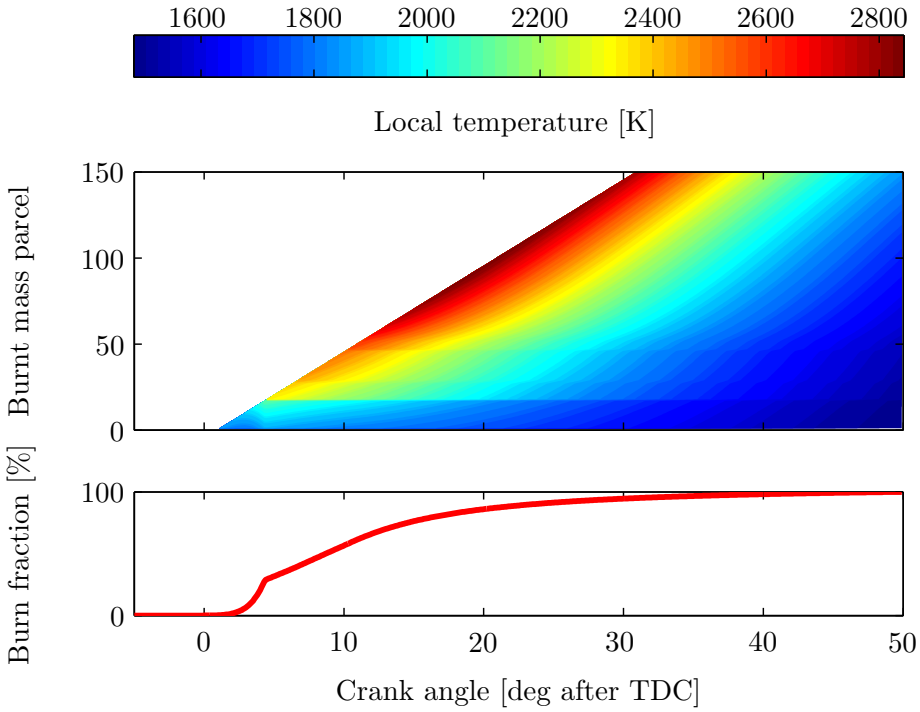

Figure 2.14: Upper plot shows the local temperature distribution in the spray versus crank angle and burnt mass parcel. The burn fraction profile is included in the lower plot for reference.

Flame temperatures depend on the combustion timing (which affects initial temperature and pressure) and the local mixture. They also rise and fall as the pressure rises and falls during the combustion and as the gases mix. The mass distribution during combustion is superimposed on top of these effects to give complex results.

An example is included in Figure 2.15 which illustrates is the effect of EGR on the mixture distribution and subsequent flame temperatures. For the case without EGR, the pre-mixed combustion burns quicker and consumes all the air within the spray. As more fuel is injected, the combustion rate is limited by the amount of entrained gas and the mixing of this within the spray. For the case with EGR, the rate of combustion is slower, since the oxygen concentration is lower. This results in a later and more progressive burn. The flame temperatures and subsequent NO emissions are reduced by the combination of a later burn and lower oxygen concentration.

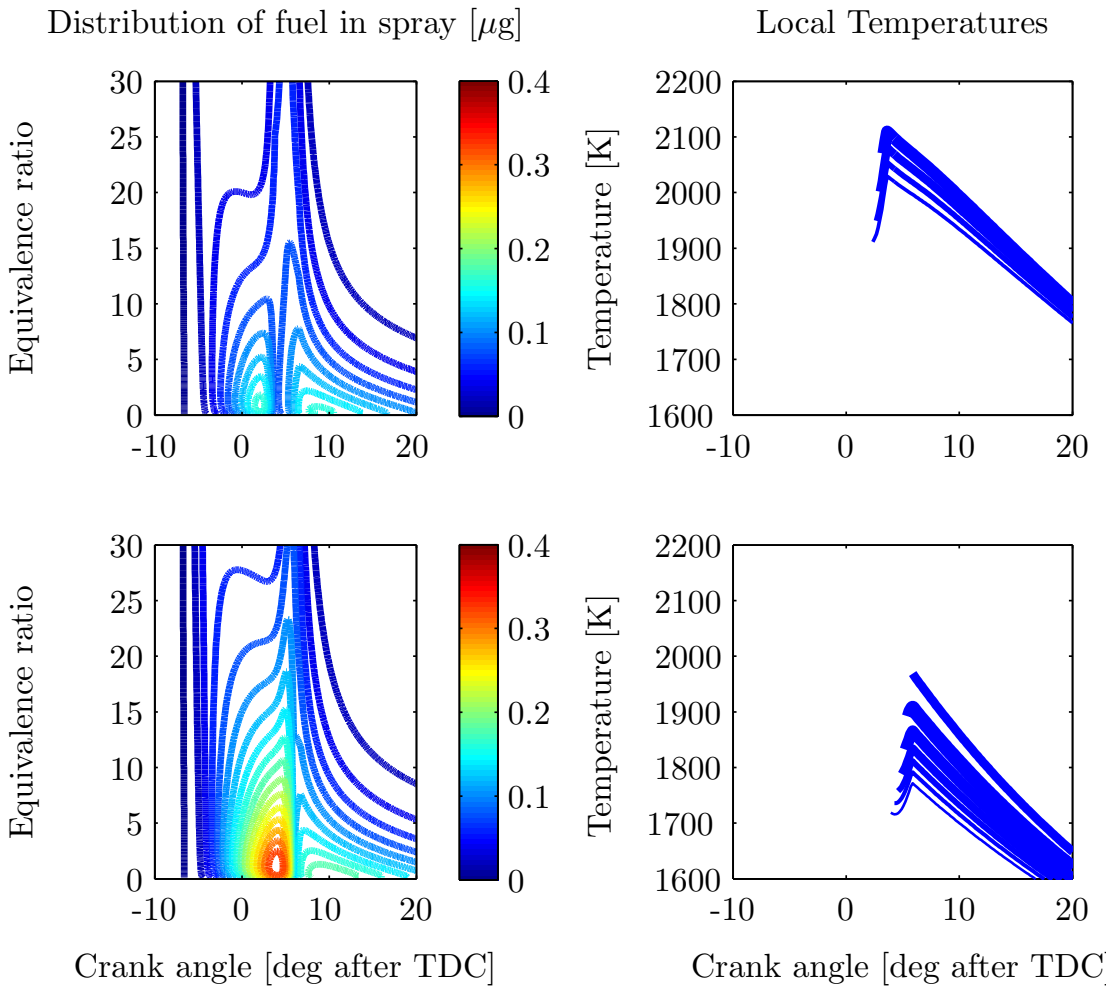

Figure 2.15: Mixture evolution and local temperatures by the control orientated model with 0% EGR (upper) and 30% EGR (lower). The left plots show the distribution of fuel as a function of equivalence ratio and crank angle. The right plots show temperatures of individual mass parcels with line width proportional to the mass in each parcel.

2.3.6 NO_x emissions prediction

Interface

Input: Local temperature and mixture distribution

Output: Cumulative NO_x emissions

The emissions estimate is based on the extended Zeldovich mechanism, see [44], that is used to estimate the rate of change of NO concentration, [NO]:

$$\frac{d[NO]}{dt} = \frac{2R_1(1-\alpha^2)}{1+\alpha R_1/(R_2+R_3)} \tag{2.47}$$

where α is the ratio of [NO] to the equilibrium concentration, $[NO]_e$, and R_x are reaction rates. The reaction rates are calculated from the forward rate constants, k_x^+, associated with each reaction, multiplied by the concentrations of the reacting species. In an internal combustion engine, the flame reaction zone is very thin whilst the burnt gases can be compressed further after combustion. This usually leads to the NO formation in the burnt gases dominating the NO formed during combustion and therefore the combustion can be assumed to be decoupled from the NO formation, [44]. The concentrations of O, N_2, N, O_2 and OH can therefore be approximated by the equilibrium concentrations which leads to the following expressions for the reaction rates:

$$\begin{aligned} R_1 &= k_1^+[O]_e[N_2]_e \\ R_2 &= k_2^+[N]_e[O_2]_e \\ R_3 &= k_3^+[N]_e[OH]_e \end{aligned} \tag{2.48}$$

where the values of k_1^+, k_2^+ and k_3^+ are taken from the Bowman kinetic data, [44].

Equilibrium concentrations of the key species are tabulated as a function of temperature and oxygen concentration. Equation (2.47) is integrated to give the NO concentration, [NO], as a function of time. The variation of the equilibrium concentration with pressure has been neglected since this variation was found to be less significant.

The [NO] is calculated for each burnt mass parcel as a function of time. A typical evolution of [NO] versus time for different mass parcels is shown in Figure 2.16. This figure shows the characteristic increase in [NO] with high cylinder temperatures and subsequent reduction as the chemical reactions reverse, until the chemistry is 'frozen'

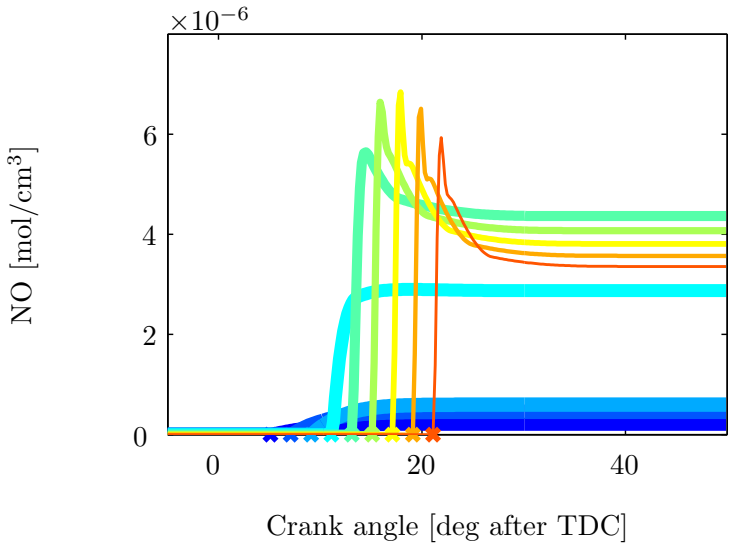

Figure 2.16: The predicted evolution of NO concentration for different parcels of burnt mass. Colour indicates the crank angle when each parcel was burnt, thickness of line indicates mass in each parcel

by the cooling during expansion and mixing with cooler gases within the cylinder. The average exhaust NO mass fraction can be calculated according to (2.49):

$$\{\overline{NO}\} = \int_0^1 \{NO\}_f \, dx_b \tag{2.49}$$

in which $\{NO\}_f$ is the final frozen mass fraction of mass parcel which was burned when the burn fraction was x_b. The burn fraction is defined as the mass burnt at a point in time divided by the total mass burnt.

2.3.7 Soot emissions prediction

Interface

Input: Local temperature and mixture distribution

Output: Cumulative soot emissions

At each time-step, the soot formation and oxidation reactions, characterised in (2.12) and (2.13), are applied across the mixture. The advance over the usual implementations is that the effect of the mixture distribution on the soot calculation is included, as shown in Figure 2.17.

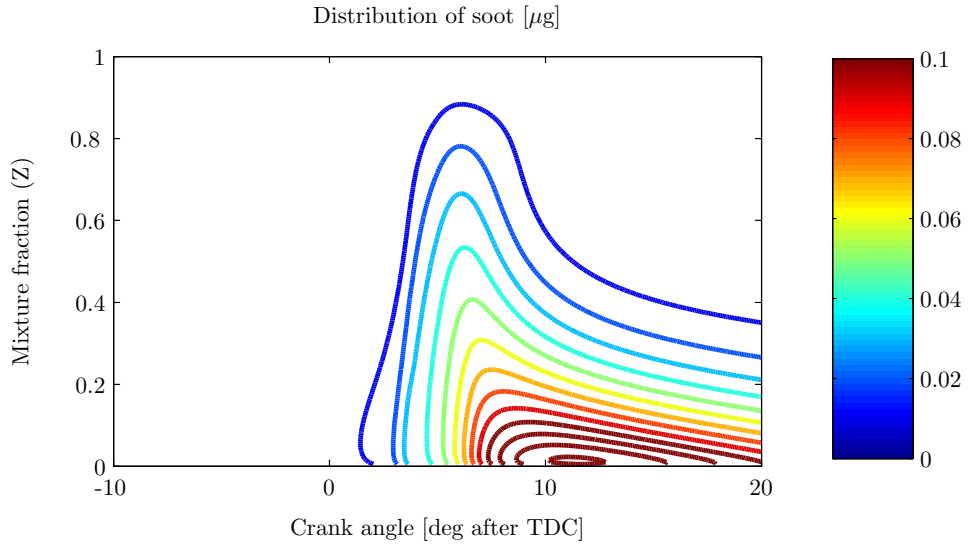

Figure 2.17: The predicted evolution of soot concentration as a function of mixture fraction.

2.4 Model calibration and validation

The physical parameters within the model were calibrated from reference data sets. The model also includes a number of parameters for which no reference or suggested data was available. These parameters were calibrated by comparing simulated model outputs to measurements of the corresponding output at a limited number of test-points. This process was in two phases: firstly calibrating the the bulk cylinder properties and secondly the emissions since the bulk cylinder properties are independent of the emissions models.

The model was then validated across a wider set of data to demonstrate its robustness.

2.4.1 Measured data

The model has been calibrated with data from a detailed model of the mixing and chemical reactions (Stochastic Reactor Model, SRM) and measured data from a Ricardo single cylinder engine running at Brighton University. The engine was run at over 100 test-points with varying input parameters. There are three key-points (KP1, KP2 and KP3), which refer to specific speeds and loads and 3 full load test points. At the key-points the engine is run with a range of EGR ratios and injection timings. At the full load points the injection timing is varied. The calibration data set is presented in Figure 2.18. The SRM model used was developed by CMCL innovations [4], see Appendix A.4 for more details of this simulation.

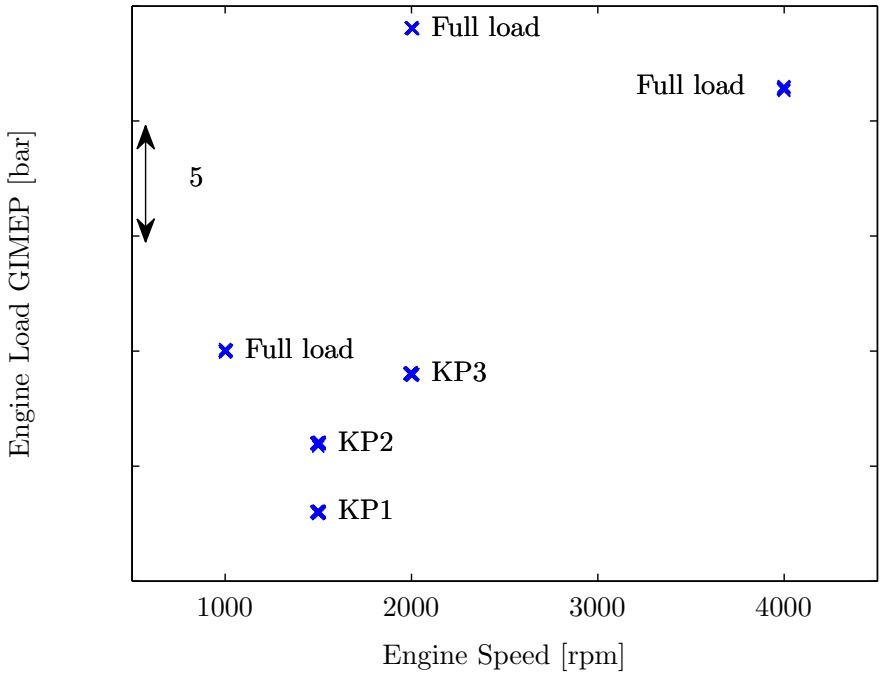

Figure 2.18: Speed and load values for the calibration data set.

To calibrate the engine model, the engine, the SRM model and the control orientated model were all run at the same test-points using the same input data. The outputs from the models were then compared to the outputs from the engine.

2.4.2 Model calibration

The calibration started by considering the cylinder pressures and heat release data at KP2 for the single cylinder engine, the SRM model and the control orientated model, see

Figure 2.19. Parameters considered when calibrating the model included the combustion and Arrhenius coefficients and the polytropic coefficients for the initial gas composition.

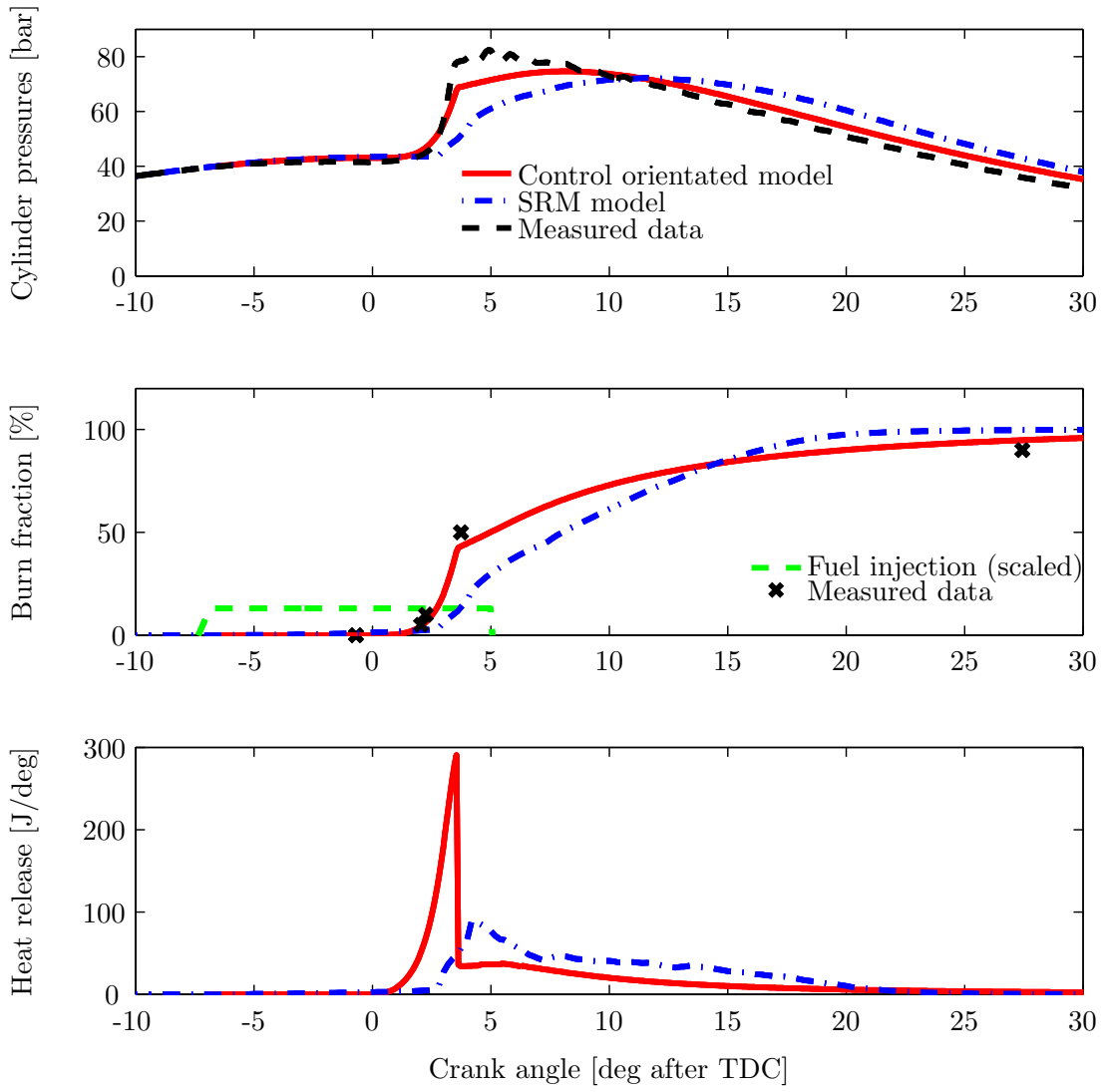

Figure 2.19: A comparison of the measured and predicted in-cylinder results for KP2. The upper plot shows the cylinder pressure, the middle plot shows the burn fraction and the lower plot shows the heat release rate.

The calibration process for the local temperatures considered the mixing rate of the burnt gases for the flame temperature evolution. The NO model is based on physical quantities so only required calibration of one parameter, a scaling factor introduced to calibrate to the test-point with maximum NO_x. The soot emission model required calibration of the formation and oxidation constants, using the test-point with the maximum soot, with the remaining coefficients taken from [43].

2.4.3 Model validation

Once the model was calibrated at the calibration points, the validation was extended to the full data set. The full data set results are presented as a correlation plot where the solid line assumes equal measured and predicted values, while the dashed lines give a representative tolerance band. See Figure 2.20 for the correlation of start of combustion and 50 % burn angles where the 50 % burn angle corresponds the crank angle at which $x_b = 50\,\%$.

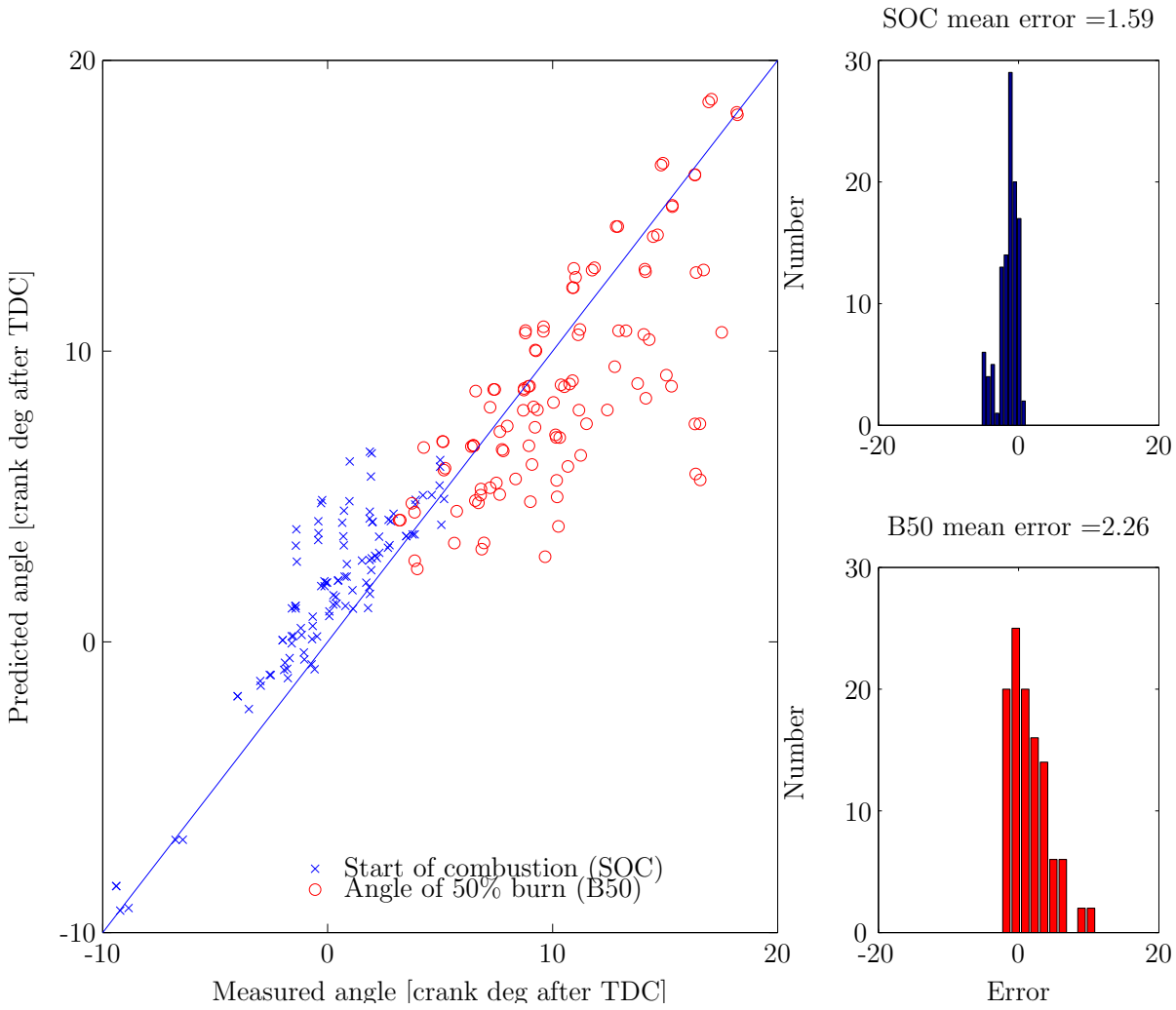

Figure 2.20: Correlation of the start of combustion (SOC) and 50 % burn angles for the validation data set, with distributions of errors for the two angles and mean of the absolute error given in the right hand plots.

The final stage in the validation is to compare the measured and predicted emissions. Since this project is principally concerned with emissions trends, an emissions index has been used for the validation. The emissions index is calculated as the predicted emission, normalised by the peak emissions from the validation test data. In addition, the validation data set includes many points with very low emissions. The full data set

is included in Appendix A.5 with the comparison to measured data for the points with significant emissions included in Figure 2.21.

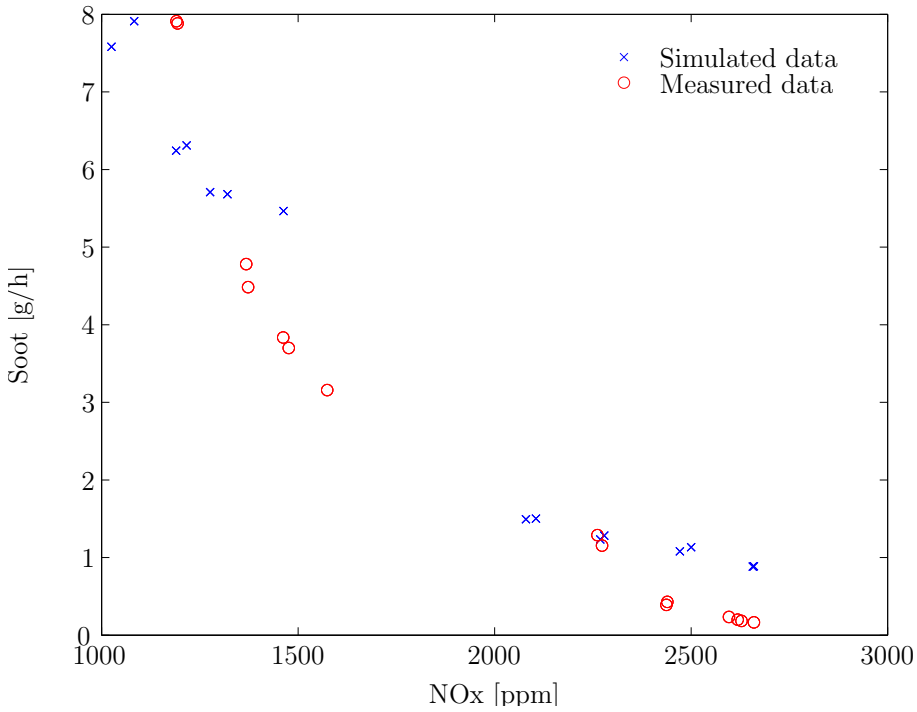

Figure 2.21: A comparison of the NO_x versus soot tradeoff for the measured and predicted data with significant emissions.

The response of the combustion system to varying inputs is the priority, rather than absolute predictions of emissions. Figure 2.22 shows the measured and predicted responses to EGR ratio and start of injection. These plots show the model is capturing the correct trends despite having an offset to the absolute emissions.

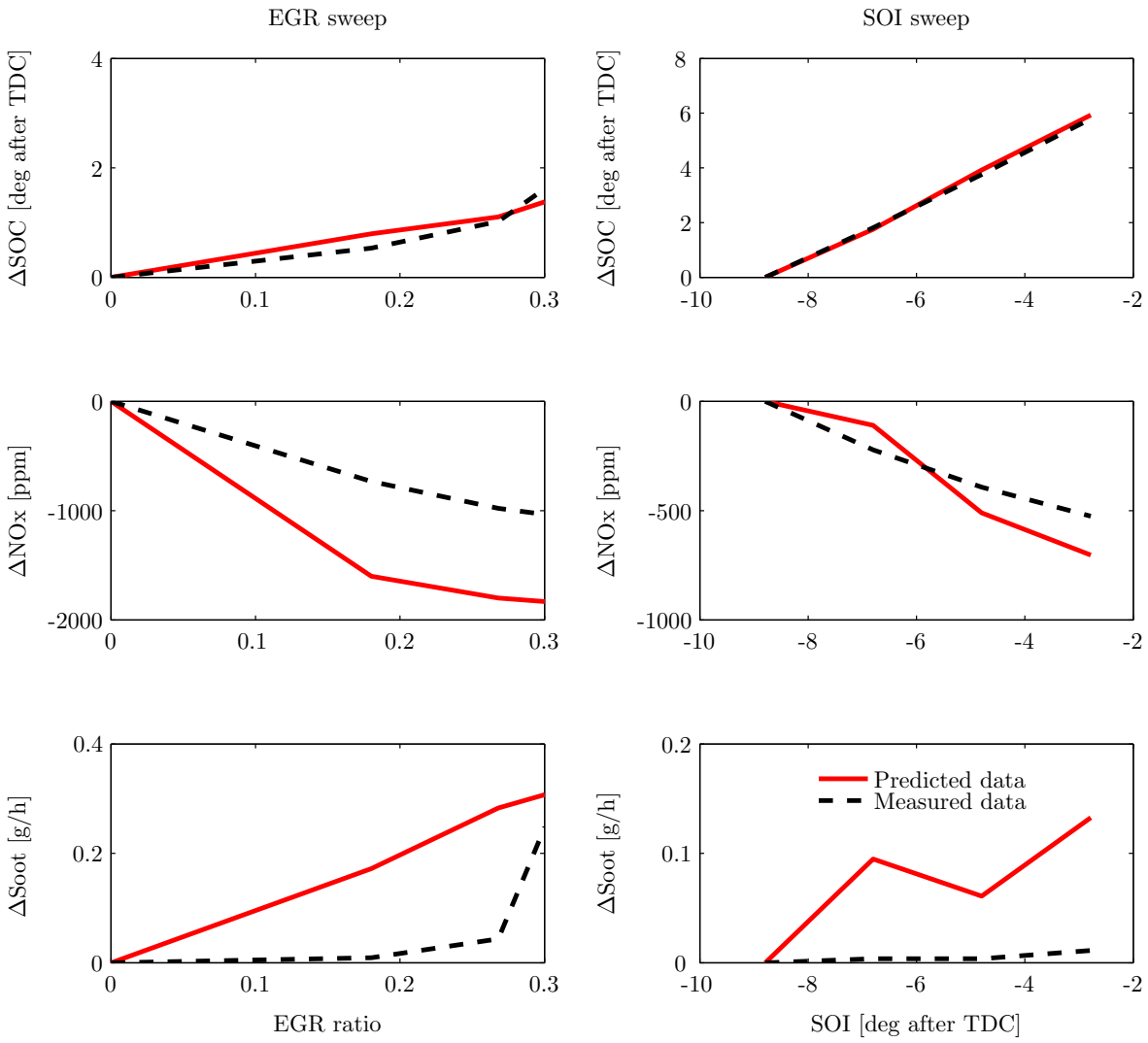

Figure 2.22: A comparison of the change in start of combustion (SOC), NO_x and soot for the measured and predicted data as a function of EGR ratio (left) and start of injection (right).

2.5 Conclusions: Control orientated combustion model

This chapter has described the development of a new combustion model which retains sufficient detail to capture the mixing process whilst reducing the computational burden associated with earlier approaches. The new areas of development include the combination of the β-PDF mixing approach with a simplified combustion model. This provides an interesting insight into the combustion process which may prove useful outside the control development scope of this work, such as the desktop calibration of fuel injection parameters.

The validation data shows a good correlation of the combustion processes and a reasonable representation of the emissions predictions, satisfying the original requirements to capture the response of the combustion system to the key inputs. The validation also suggests that for this particular combustion system, the model has captured the key inputs and therefore provides an upper baseline for the complexity required for control orientated combustion modelling. Future work could take this model and progressively simplify, whilst checking the responses, to create a fully optimised model. In particular, creating a model of differentiable functions as described in Figure 2.3 would provide more opportunities for optimal control formulations.

Chapter 3

Combustion control

3.1 Objective for combustion control

The control of the combustion process is achieved by varying the fuel injection pressure, the fuel quantity and the injection timing together with the state of the gases coming into the engine (boost pressure and EGR ratio). Without direct cylinder pressure measurements, the combustion is controlled in an open-loop manner, with set-points for the fuelling and inlet gases calibrated (optimised) at steady-state operating points on engine dynamometers. During transient operation, the set-points move in accordance with the driver inputs (typically engine torque and speed) with some transient modifications for features such as smoke limitations. However, the operating conditions of the engine during transients are often different to those used for calibrating the steady-state set-points. This chapter investigates whether it is possible to reduce the pollutant emissions from a Diesel engine by optimising the control of the combustion process, with the optimisation algorithms targeted at real-time on-line calculation.

3.2 Literature study: Combustion control and optimisation

3.2.1 Combustion control

Combustion control is often associated with the control of combustion phasing rather than controlling emissions directly. Reference [45] cites a close correlation between combustion phasing and pollutant emissions that avoids the expensive prediction of emissions.

In [67], models are used to estimate the intake gas O_2 concentration and corresponding the ignition timing. The former is controlled by the EGR system and the latter is a function of the estimated O_2 concentration and injection timing. The ignition model

includes the spray behaviour, heat exchanges and oxidation reactions. The start of ignition is defined as the time at which the gas temperature passes a threshold. The ignition timing model is used to estimate the ignition timing for three *injection* timings. The target injection timing is calculated by fitting a quadratic polynomial to these three points and evaluating the required injection timing to give the desired ignition timing. The authors demonstrate this model running in a prototype controller and assess the potential production implementations by capturing the control actions in a Neural Network, avoiding the real-time calculation of the model and optimisation.

In [45] a combustion model is developed as a series of differential equations with transitions from one combustion phase to another. The transitions are triggered as integrated state variables from the differential equations exceed thresholds. The model is used to calculate a Crank Angle corresponding to $x\,\%$ of fuel burnt, CA_x. The authors pose the problem in terms of fast (fuel path) and slow (air path) systems. The CA_x parameter is presented as a function of the fast and slow states: $CA_x = f(x_{slow}, x_{fast})$. During a transient, the slow states deviate from the set-point state and CA_x will differ from the set-point value. Noting set-point values with a bar, the error in CA_x is given as follows:

$$CA_x - \overline{CA_x} = f(x_{slow}, x_{fast}) - f(\bar{x}_{slow}, \bar{x}_{fast}) \tag{3.1}$$

Defining the tracking errors for the states as $\delta x = x - \bar{x}$, one obtains:

$$CA_x - \overline{CA_x} = f(\bar{x}_{slow} + \delta x_{slow}, \bar{x}_{fast} + \delta x_{fast}) - f(\bar{x}_{slow}, \bar{x}_{fast}) \tag{3.2}$$

The control approach involves modifying δx_{fast} such that the error in CA_x is zero. This is achieved using a Taylor expansion and sensitivity analysis. The real time implementation of this approach is achieved by noting that the modification of δx_{fast} can be scheduled on engine speed and engine load and stored as a two dimensional look-up table.

A third approach to combustion control is presented in [87], where estimates of the PM and NO emissions are used in a feedback control architecture. The estimates are based on extended Kalman filter observers and emission sensors. The authors consider three inputs to the engine: swirl valves (which can vary the swirl of the gases in the combustion chamber by closing one of the intake ports), start of injection timing and burnt gas (or EGR) ratio. They observe that the swirl valves have a strong influence

on PM emissions but a small impact on NO emissions and consequently propose a PI controller for the PM emissions using the swirl valves. The NO emissions are controlled with the SOI and EGR ratios. The authors note that the validity of this approach is limited to operating ranges where the sensitivities of the PM and NO emissions on the manipulated variables allow for the control signals to be split in this way. The motivation for this work was to reduce the safety margins used in calibrating engines, facilitating improvements in other areas such as fuel consumption.

The first two approaches have shown that the emissions may be reduced by regulating the burn angle during the transient, these approaches use models to predict combustion and regulate to set-points which have been calibrated in advance. The third approach regulates emissions (again to pre-calibrated set-points) directly during the transient, but does not use a model of emissions.

3.2.2 Optimisation and Non-linear Programming

The optimisation of engine emissions in a combustion controller requires the solution of a Non-linear Programming (NLP) problem. A good introduction to NLP problems together with practical examples is given in [16].

This thesis considers direct methods which can be divided into two broad groups, sequential and collocation methods. Collocation methods can require more computational resources and are studied later in Chapter 5 where the calculations are carried out off-line. Sequential methods are considered here for real-time computation and they evaluate the cost function $J(u)$ by integrating the state equation for different values of u. An optimisation algorithm selects the next value of u and the optimisation continues until a minimum of $J(u)$ is found. Several broad classes of iterative solvers have been developed including: simplex methods, discrete grid methods and trust region methods, see [75] for a summary of these methods which include a search algorithm and a termination criteria.

Trust region methods involve starting with an estimate of the optimum, $\hat{u_0}$. A least squares approach is used to fit a model to the cost function, $\hat{J}$, in the vicinity around $\hat{u_0}$ (the trust region) and the minimum of this model is calculated. The cost function, J, is calculated at the minimum of the model and compared to the cost function which the model estimated, $\hat{J}$. If J and $\hat{J}$ are sufficiently close then the model is deemed a good fit to the original function and the model can be trusted. The trust region can be

expanded in this case. If J and $\hat{J}$ are not close enough the trust region is contracted. The process is repeated until the minimum is found within a tolerance. This approach can lead to a rapid convergence with relatively few function evaluations, however it is still an iterative algorithm and the number of iterations to achieve convergence is not bounded, a requirement for real-time implementation.

The question of convergence is discussed widely, for example in [85], [52] and [37], where the direct methods are shown to converge on the global minimum. However, the number of iterations may be prohibitive for the real-time application considered here.

In [17], a trust region approach was developed for a problem similar to the problem defined in Section 4.1. The authors developed an algorithm to minimize J which was a quadratic function of the output of a detailed simulation model, $y(u)$. The authors simplified the problem by using a linear model, however instead of modelling the objective function, they applied the linear model to the output of the simulation model, $y(u)$. The minimum of J can now be calculated algebraically from this model.

The current research is an advance over this prior work because the trust region approach is applied to a parallel processing architecture and methods have been investigated to reduce the number of iterations.

3.3 Simulation environment

A representative engine model has been provided by Ricardo UK for the simulation of engine responses. This model is based on a 2 litre engine with a Variable Geometry Turbocharger (VGT) and an EGR circuit from the exhaust manifold to the intake manifold, see Figure 3.1. The engine model is a Mean Value Engine Model (MVEM) and captures the air-path response whilst being able to run quickly (faster than real-time), [86]. The combustion and emissions model from Chapter 2 has been added to the MVEM model to create a full plant model for assessing control algorithms.

The simulation environment is completed with a baseline controller for the fuel and air paths. The baseline controller applies feed-forward control with PI action on the EGR rate and feed-forward control for the injection timing. The simulation environment, including baseline controller, is illustrated in Figure 3.2.

A typical transient manoeuvre was selected to evaluate the different control approaches. The engine is subjected to a step change in load at a constant engine speed (2500 rev/min). The baseline simulation results are given in Figure 3.3 and show the NO

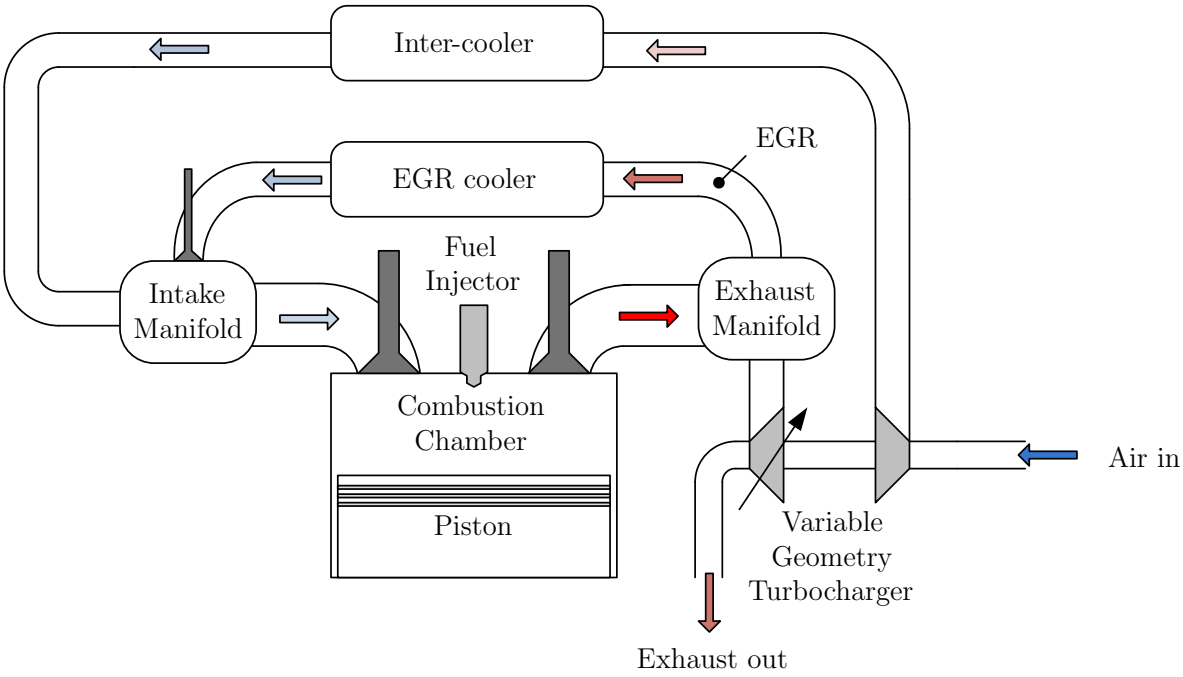

Figure 3.1: Engine schematic for simulation model, including a EGR circuit, an EGR cooler, direct fuel injection, a variable geometry turbo-charger (VGT) and an intercooler.

emissions rising and then falling back during the transient. This relatively aggressive manoeuvre also requires fuel limiting to avoid excessive smoke, which can be seen in the fuelling trace which is delayed in rising to 40 mg/shot whilst the air flow builds up.

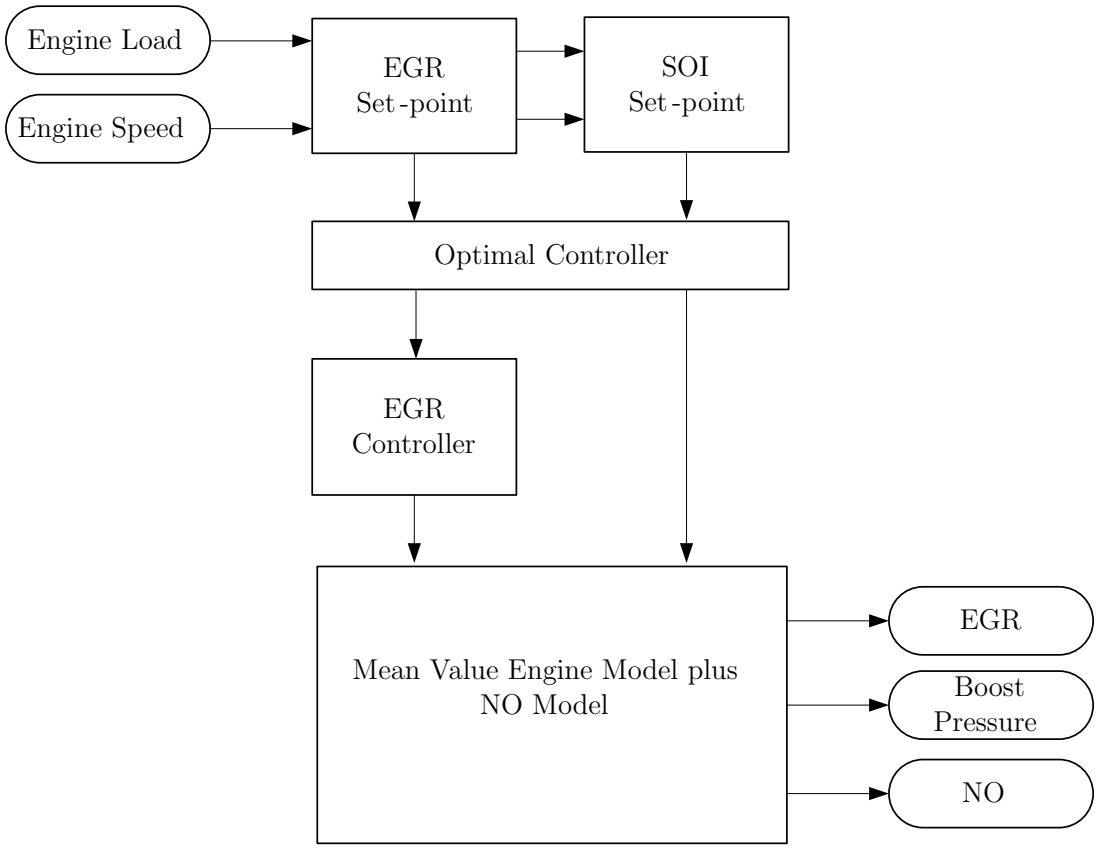

Figure 3.2: Simulation environment for evaluating new algorithms for the fuel and air paths.

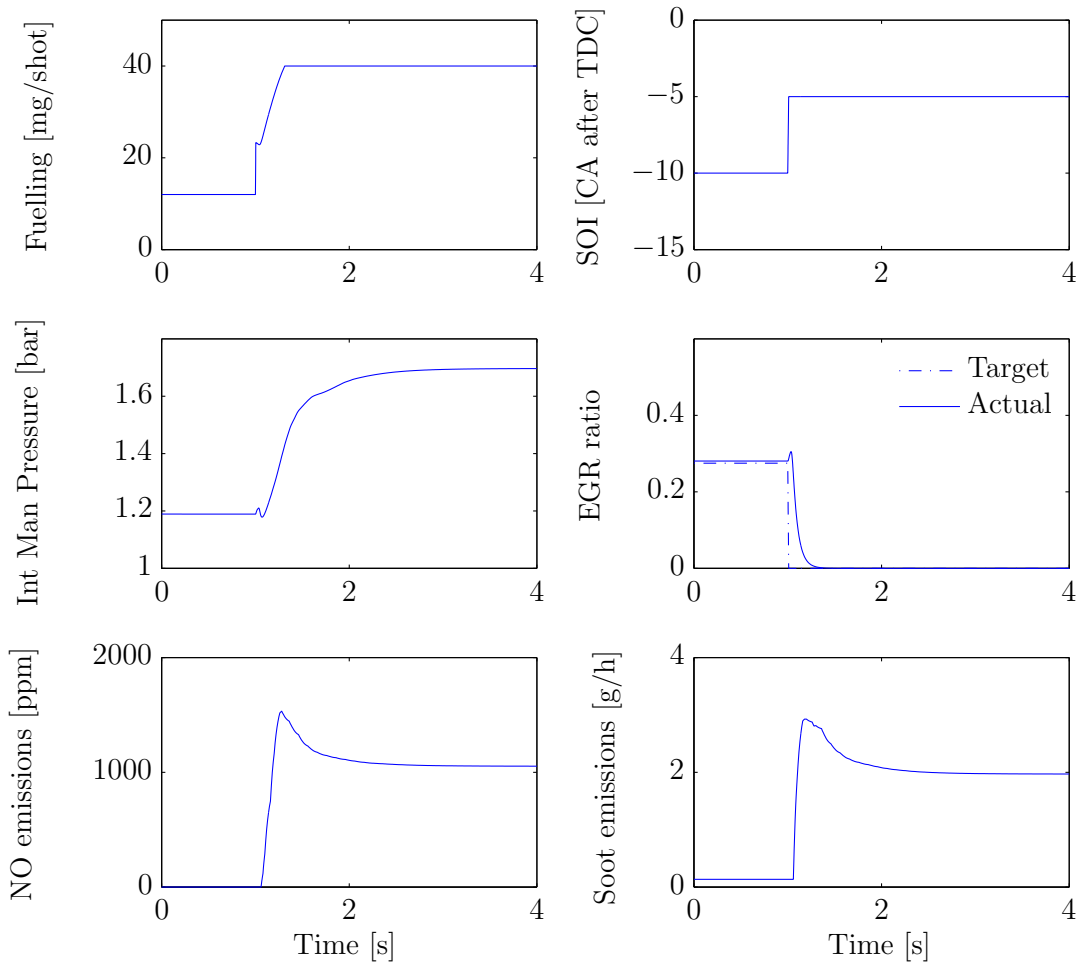

Figure 3.3: Simulation results from the baseline configuration, at a constant 2500 rev/min.

3.4 Controller design

3.4.1 Cost function selection

In Section 2.1, the combustion control variables were divided into slow (air path) and fast (fuel path). The fuel path has a response time of approximately 0.01 seconds, [67], whilst the control of the gaseous mixture entering the cylinder is slower due to the transport delays from the EGR valve to the cylinder (approximately 1 second). Boost pressure control is slower still, and is constrained by the inertia of the turbo-charging system (of the order of a few seconds).

The MVEM response, Figure 3.3, shows that the EGR response for this engine is of the order of 0.25 seconds, and therefore with respect to an engine transient manoeuvre, it can also be considered as 'fast'. Since both controlled variables are 'fast' and have no dynamics, the model predictive controller is simplified to having a control horizon of one step. The control problem is perhaps better described as an on-line optimisation.

The optimisation of engine operation requires a cost function which should be minimised to give the optimal control outputs. The selection of the cost function is the main element of the controller design phase and also needs to align with typical automotive development procedures. In typical engine development programmes, the steady-state performance of the engine is optimised extensively on an engine dynamometer, at many points across the engine operating range.

The requirement for the design of the cost function is therefore to minimise emissions whilst also minimising control actuator deviation from steady-state values. ie when there is a transient delay in air flow, the controller should find a compromise with less emissions with the actuators away from their steady-state positions. When the air flow is close to steady-state conditions, the optimisation should return the actuators to the steady-state positions.

The cost function, J, selected for this study seeks to minimize any increase in NO emissions over the steady-state value plus the difference in control action with respect to the steady-state values.

$$J(\mathbf{u}) = (y(\mathbf{x}, \mathbf{u}) - y(\bar{\mathbf{x}}, \bar{\mathbf{u}}))^2 + Q(u_1 - \bar{u}_1)^2 + R(u_2 - \bar{u}_2)^2 \tag{3.3}$$

where x are the in-cylinder states, y are the outputs (NO emissions), u_1 and u_2 are the control actions and Q and R are weighting factors between the different elements of the

cost function. Steady-state values are indicated with a bar.

Note: This cost function will tend to drive the system to the set-point as the engine states stabilize to steady-state conditions, even if the model predicts a better NO emission with different control settings. This takes advantage of the extensive steady-state calibration, whilst allowing a deviation from these points during a transient when the engine states do not match those at the steady-state operating point.

3.4.2 Search algorithm

This section details the development of a search algorithm which is suitable for on-line calculation. In this application, the processing power available is limited due to cost constraints, requiring a search algorithm which can minimise the cost function in a small number of iterations. Here, initial ideas are explored for parallel implementation with a more detailed description in Chapter 4.

The optimisation needs to cover the whole search space efficiently, whilst being able to converge rapidly on the solution. The approach explored in this work, draws on the trust-region class because they can be configured to search the whole search space. This feature makes them well suited to requirement of a global search with parallel solvers. In the first iteration, the whole search area can be divided into parallel sub-regions and each sub-region can be set as the initial search area. Subsequent iterations can then refine the search. Other benefits of these approaches include: they can be extended to multi-dimensions and are well suited to adopt a starting point from previous iterations (warm starting) which can improve convergence speeds.

This approach differs from the usual trust-region methods because regions which are not initial estimates for the optimum are also examined. This is because the sub-region containing the minimum is not known and due to time constraints, this cannot be established iteratively. Therefore multiple regions are investigated in parallel and this helps provide a wider coverage of the cost function to improve the robustness against finding a local minimum. A second difference to the conventional trust region approach is that the number of iterations has been reduced by only allowing the algorithm to contract the search regions (from the maximum size) rather than iteratively expanding and contracting as the approximations are trusted or not.

The termination conditions adopted here are determined by the real-time requirements for the algorithm, which require a solution at a regular time interval. This means

that the search algorithm has a fixed amount of time for calculation and must deliver the best estimate of the minimum within this time frame, rather than continuing to seek the minimum according to a termination condition. Whilst this may seem a large compromise, the cost function is designed so that even if the search algorithm cannot find a minimum, it will default back to the original calibrated steady-state calibration, thus either improving or preserving the original performance.

The algorithm developed for this work is called the Parallel Trust Region (PTR) search and is analysed in more detail in Appendix B, including a brief analysis of the potential errors in convergence.

3.5 Controller development

The development of the controller starts with the two control variables, Start of Injection (SOI) and Exhaust Gas Recirculation (EGR) treated individually before combining in a multivariable controller optimising both variables over the test manoeuvre. The results are summarised in Figure 3.4 with individual figures for each simulation included at the end of this chapter.

3.5.1 Start of injection (SOI) control

Simulation approach

Objective: To assess the benefit of varying the SOI during the test manoeuvre.

At each timestep, the optimal controller calculates the optimum SOI by minimising the cost function, Equation 3.3, with the SOI range from -14 to -1 degrees after top dead centre (ATDC). The optimum SOI is passed to the plant model and the emissions are calculated for the next timestep. This process is repeated over the test manoeuvre.

To allow comparison to the burn angle control proposed by [45] and [67], the combustion model has been configured to output the angle at which 20 % of the fuel has been burnt. A PI controller has been implemented to regulate the burn angle to the burn angle at the steady-state conditions.

Results and discussion

The results from optimising the SOI control are shown in Figure 3.5 and the results from controlling the 20 % burn angle are shown in Figure 3.6. Figure 3.5 shows that optimising the SOI has partially reduced the NO emissions, whilst the response of the intake

manifold pressure (a measure of the responsiveness of the engine) is unchanged. There is still a peak of NO emissions (reduced relative to baseline) when the SOI controller saturates at 1 degree before TDC. Figure 3.6 shows there is minimal impact controlling the burn angle since the burn angle is already close to the target during the transient and thus the burn angle error is small.

3.5.2 EGR control

Simulation Approach

Objective: To assess the benefit of varying EGR during the test manoeuvre.

At each timestep, the optimal controller calculates the optimum EGR by minimising the cost function, Equation 3.3. The optimum EGR is passed as a target EGR to the EGR controller and the emissions are calculated for the next timestep. This process is repeated over the test manoeuvre.

Results and discussion

The results from optimising the EGR control are shown in Figure 3.7. The difference between the EGR control and the SOI control is that the EGR control impacts the response of the air system, modelled by the MVEM. This can be seen in Figure 3.7 where the intake manifold pressure lags behind the baseline simulation. This then impacts the engine fuelling response since the fuel is now limited by the available fresh air. The NO peak in this case has been removed but at the expense of a slower transient response.

3.5.3 SOI and EGR control

Simulation Approach

Objective: To assess the benefit of varying SOI and EGR during the test manoeuvre.

At each timestep, the optimal controller calculates the optimum SOI and EGR by minimising the cost function, Equation 3.3 with two control variables. This requires a two dimensional optimisation which adds complexity to the previous optimisations, however since the PTR approach can be extended to multi-dimensions, this does not require a new optimisation algorithm. Following the optimisation, the optimum SOI is passed to the plant model and the optimum EGR is passed as a target EGR to the EGR controller and the emissions are calculated for the next timestep. This process is repeated over the test manoeuvre.

Results and discussion

The results from optimising the SOI and EGR control are shown in Figure 3.8. The simultaneous optimisation of the SOI and EGR have reduced the NO peak during the transient with a smaller impact on the engine performance than just EGR control.

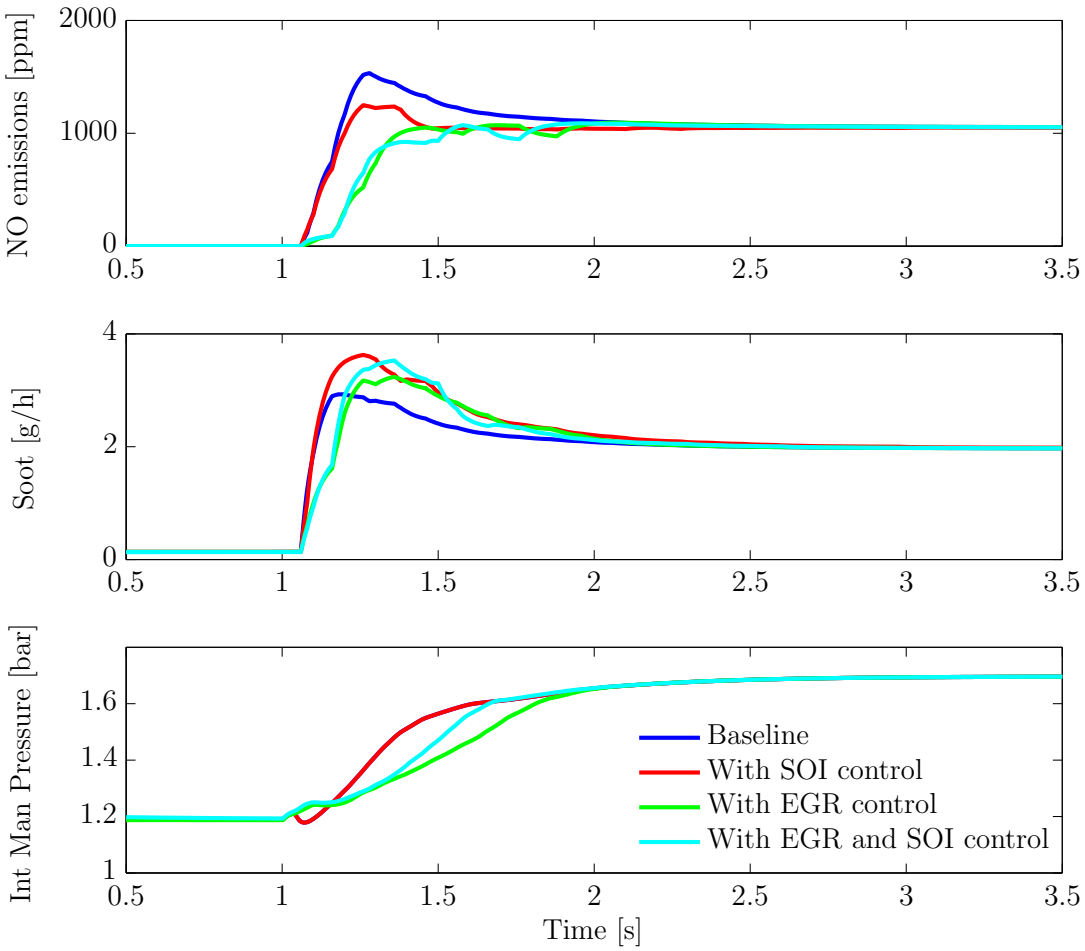

Figure 3.4: A summary of the simulation results from the controller development.

3.6 Conclusions: Combustion control

The simulation results have shown that it is possible to reduce the NO emissions by approximately 30 % by controlling the SOI and EGR during a transient manoeuvre. These control parameters were selected for this study as they are closely related to the combustion timing, which is important when correcting for the delay in air path response during a transient manoeuvre. In addition, the number of parameters were kept to a minimum in preparation for the on-line implementation in an embedded controller.

The other engine parameters, such as rail pressure, intake manifold boost pressure and multiple injections, are likely to have a smaller influence in improving the transient NO emissions and can be optimised off-line, see Chapter 5.

Whilst the NO emissions have been reduced in these simulations, there is a trade-off with the engine response and soot emissions. Increasing the EGR ratio has a direct impact on engine response during transients as exhaust gases are diverted from the turbocharger and fuelling may be limited to avoid incomplete combustion (smoke limiting). The trade-off with engine response is especially relevant for highly pre-mixed engines, such as the engine studied here, where the EGR ratio has a stronger influence on the ignition delay than the start of injection.

The soot emissions have increased as expected, however with the use of Diesel particulate filters, this increase may be less significant than the reduction in NO emissions.

In conclusion, a novel optimisation algorithm with a low number of iterations has been developed for this application, targeting the implementation on an embedded control unit. The next chapter will develop the combustion model and optimisation algorithm for real-time implementation and vehicle testing.

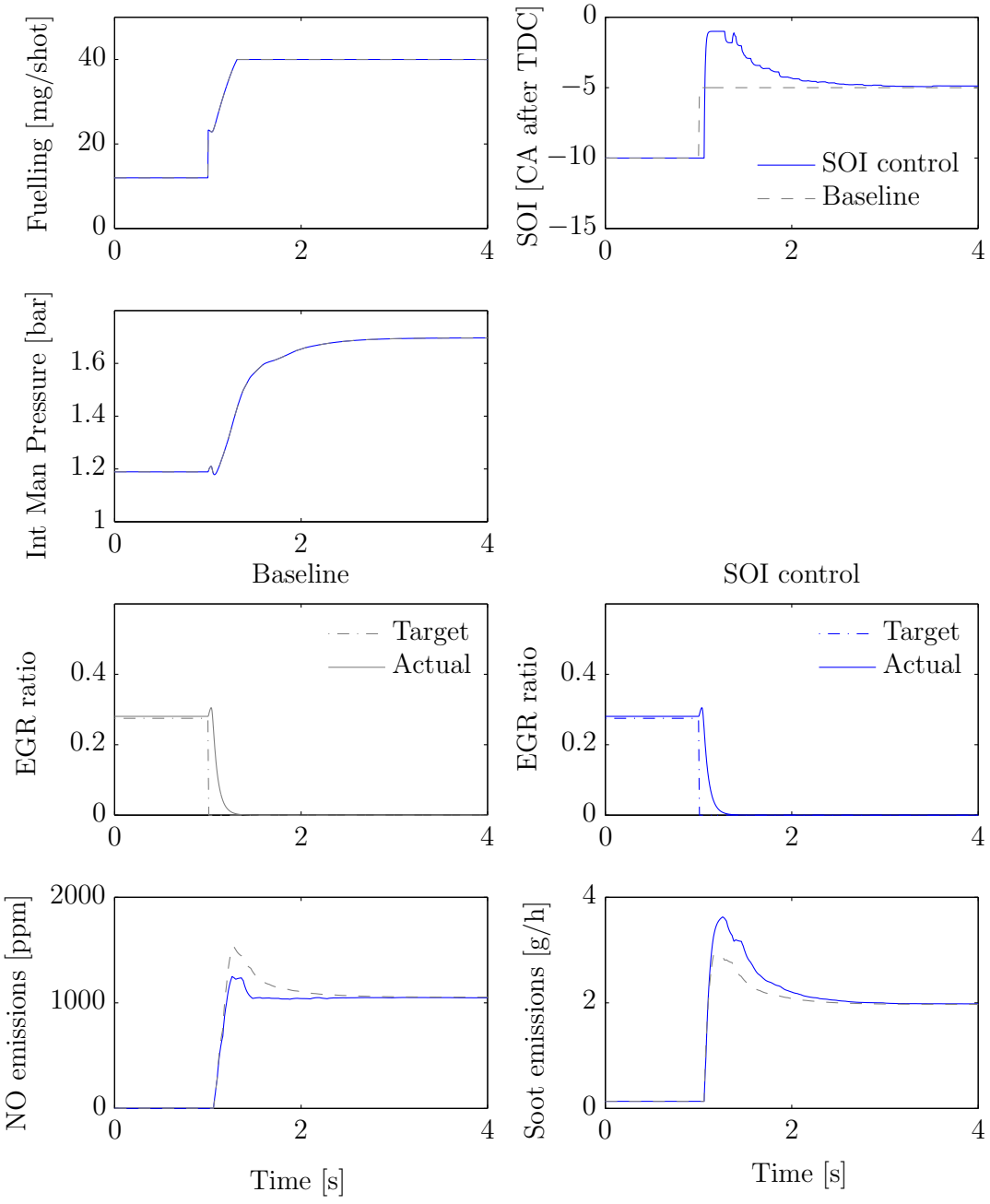

Figure 3.5: Simulation results from a simulation optimising SOI.

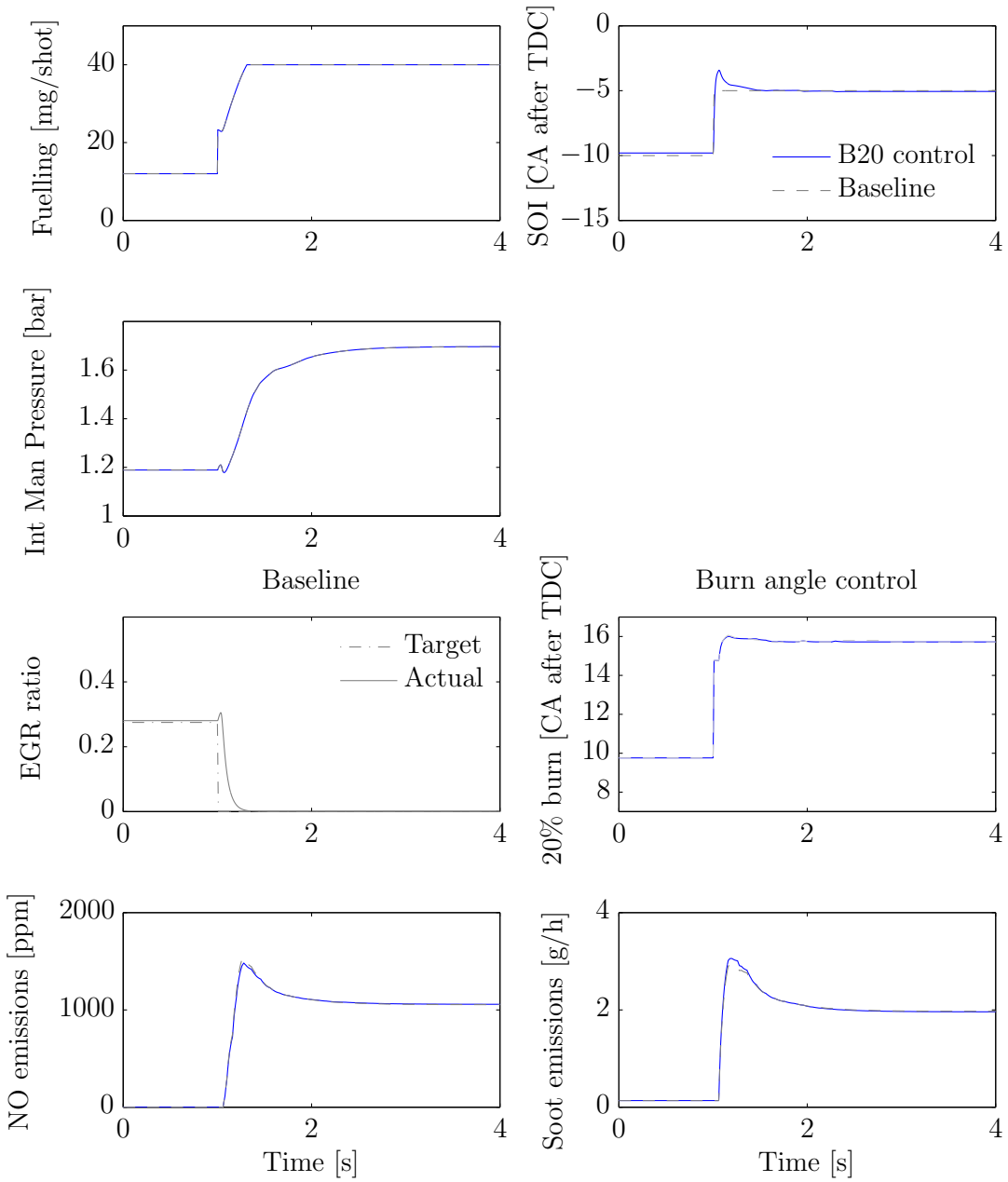

Figure 3.6: Simulation results from a simulation regulating the 20 % burn angle using SOI control.

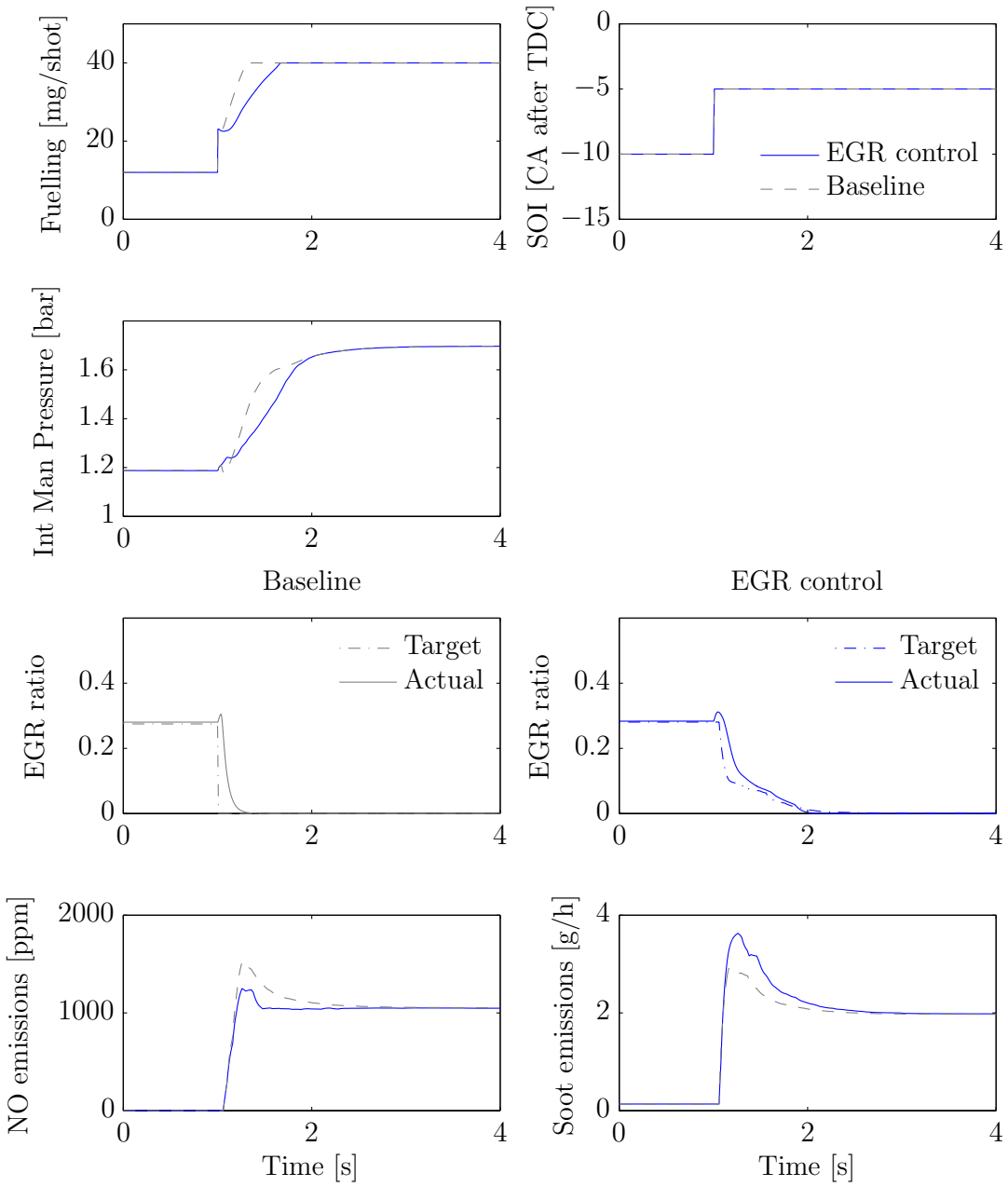

Figure 3.7: Simulation results from a simulation optimising EGR.

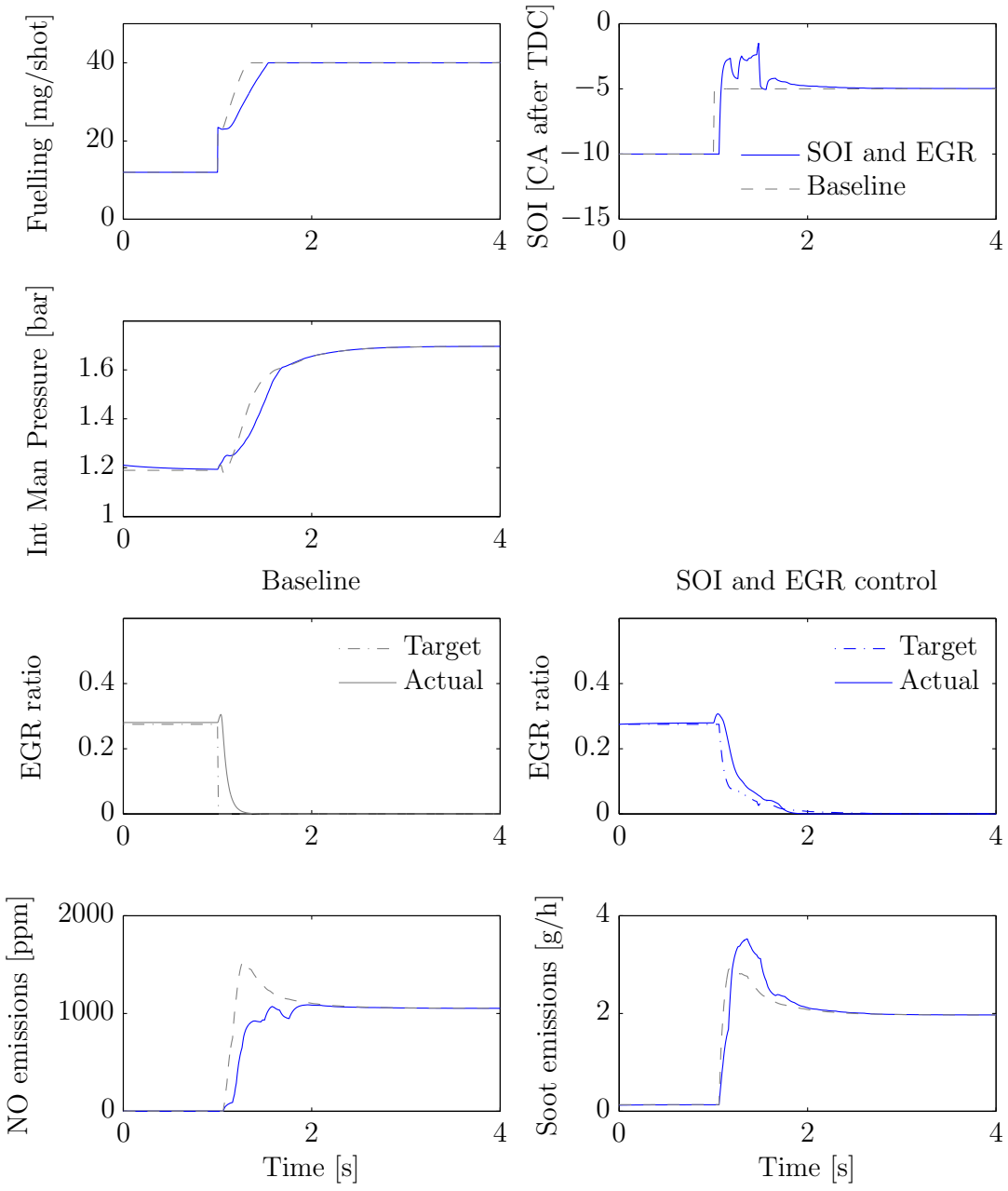

Figure 3.8: Simulation results from a simulation optimising EGR and SOI.

Chapter 4

Vehicle test and development of a predictive combustion controller

4.1 Objective for vehicle test and development

This chapter describes the implementation of the control algorithm described in Chapter 3 onto a computer system which can run in real-time and can be designed into an automotive control system, running in a test vehicle. This computer is often called an embedded control unit since it is embedded into the larger mechanical and electrical powertrain system.

Considerations during the implementation of the combustion controller onto an embedded target include the selection of the controller hardware and the design of the control algorithms to take advantage of the specific hardware.

The combustion model developed in Chapter 2 has been designed to be as simple as possible, however it is still computationally intensive, taking approximately 300 ms for one engine cycle on a 2.5 GHz i5 PC. The road-map for engine control micro processors [54] shows automotive control units operating at approximately 1500 MIPS and 300 MHz in the near future, which is significantly slower than the PC performance. Since conventional processors would not achieve the computational speed required to run the combustion model in real-time, this study investigates the use of parallel processor architectures that are available for use in automotive applications.

In many optimisation problems, the model is often simplified or linearised to derive a mathematical representation of the optimisation surface. In this project, the combustion model has been simplified, whilst still retaining its physical basis. Further simplification through piecewise linear models, identified models or other approaches is not considered further here since one of the goals of this research is to retain the physical model and develop an optimisation approach which operates in real-time.

For the real-time implementation of the combustion model and the optimisation algorithm, the combustion model needs to execute several times faster than real-time. The actual speed is defined by the dynamics of the engine systems. For example, at 4500 rev/min (typically the maximum speed for a passenger car Diesel engine) there is a cycle every 27 ms for each cylinder. This defines the update rate for SOI control at 27 ms (assuming all cylinders have the same SOI). For EGR control, the system response is slower and the update rate for the control of EGR actuators is typically of the order of 100 - 200 ms. Both of these timescales are shorter than the time required to run the combustion model on a PC, so this project has explored control hardware options which are currently available for automotive applications, though not necessarily used in engine control.

Parallel computation architectures are developed for the combustion model to increase the speed prior to application to an optimisation algorithm. A novel optimisation algorithm is required, since the computational time only allows a small number of function evaluations. The validation of the predictive control approach is carried out by implementing the combustion model and optimisation algorithm in an automotive control system and testing in a vehicle driving in an emissions test facility.

4.2 Literature study: Parallel computation

The application of an algorithm (often sequential) to a parallel computing architecture needs to consider both the target hardware configuration and the method for reconfiguring the sequential algorithm to the parallel architecture.

There are many configurations available for parallel computing and are often defined by how the computations are divided up, how the parallel processors communicate and how the memory is shared. The configurations are often summarised using the classification introduced in [30]:

- SISD: single instruction stream, single data stream, as used by the conventional sequential von Neumann architecture.
- SIMD: single instruction stream, multiple data stream. Thousands of low-performance, small-memory processors are controlled in lockstep by a central controller.

- MISD: multiple instruction stream, single data stream, as typified by vector supercomputers such as the CRAY series. The computer paradigm for this process is called pipelining in which data proceeds sequentially in a pipeline through a number of processing stages.

- MIMD: multiple instruction stream, multiple data stream, as typified by a transputer network. Here the processors are independent, each executing its own sets of instructions on its own set of data. MIMD machines may share a common memory or be distributed with each processor having its own local memory.

Computer architectures have moved on since this classification, however this highlights the importance of considering the controller hardware when designing the algorithms and secondly when comparing the relative speeds of parallel and sequential algorithms.

As discussed in Chapter 1.4, hardware platforms for accelerating computations include Field Programmable Gate Arrays (FPGA), Graphics Processor Units (GPU) and multi-core micro processors. GPUs are less suitable for embedded applications whilst FPGAs and multi-core processors are already used in automotive control units. Current multi-core processors have less than ten processors and therefore offer a limited acceleration, whilst the potential of FPGAs is harder to quantify and is the subject of this study. Despite having a slower clock speed compared to micro-processors, FPGAs can be configured to exploit parallelism at various levels. Examples of how they can speed up calculations include: being configured to calculate functions as a pipeline with one calculation being started before the previous one has completed, their instructions can be hard coded and do not need to be fetched from memory and they have efficient tools for accessing memory locations, see [80].

Methods for exploiting parallel computing architectures have been studied extensively and seek to identify independent tasks to be evaluated concurrently, each task requiring approximately equal computational requirements. When considering the configuration of a sequential algorithm for application on parallel computing architectures, there are often three areas of parallelism to consider, see [18] and Figure 4.1: Parallelism across the method in which computations for each step can be solved in parallel, parallelism across the system in which the problem to be solved is broken into separate sub-problems which can be solved in parallel (the sub-problems do not necessarily coincide with the

steps) and parallelism across the steps in which techniques are used to solve the problem simultaneously across a large number of steps.

The model-based engine control problem being studied here has two parts, both of which are candidates for parallelisation: the model of combustion (which typically consists of a system of differential equations) and the optimisation algorithm.

Parallel solution of Ordinary Differential Equations (ODEs)

The application of parallel computing to the solution of ODEs has been the subject of many years of research, see [18] for a review. Opportunities for parallelising the solution are limited because of the inherent sequential nature of the problem and are not explored further in this work.

Parallel computing applied to optimisation

The use of a parallel computing architecture offers significant opportunities to improve the solution of optimisation problems where they can be configured appropriately. The iterative nature of many of the approaches means they still must be re-considered in light of parallel computing. In particular, the solution of an optimisation problem with a low number of iterations and a fixed number of function evaluations at each iteration has so far received little or no attention.

4.3 Development of a optimisation algorithm with few iterations

4.3.1 Pipeline implementation of the PTR algorithm

The PTR search algorithm is suitable for parallel implementation on an FPGA since it can be configured to run in a pipeline. A pipeline is created when a number of function evaluations can be started before the end of the first calculation. This is illustrated in Figure 4.2, based on the PTR algorithm in Appendix B. The first search region is evaluated and analysed while the function is evaluated in the second, third and fourth regions. This allows time for the new search parameters to be calculated for the first region and the subsequent iteration can be started.

The PTR algorithm is now applied to a predictive control problem, starting with an example which is simpler than the complex combustion model. In Appendix C, a Model Predictive Controller (MPC) is implemented with the pipelined PTR algorithm

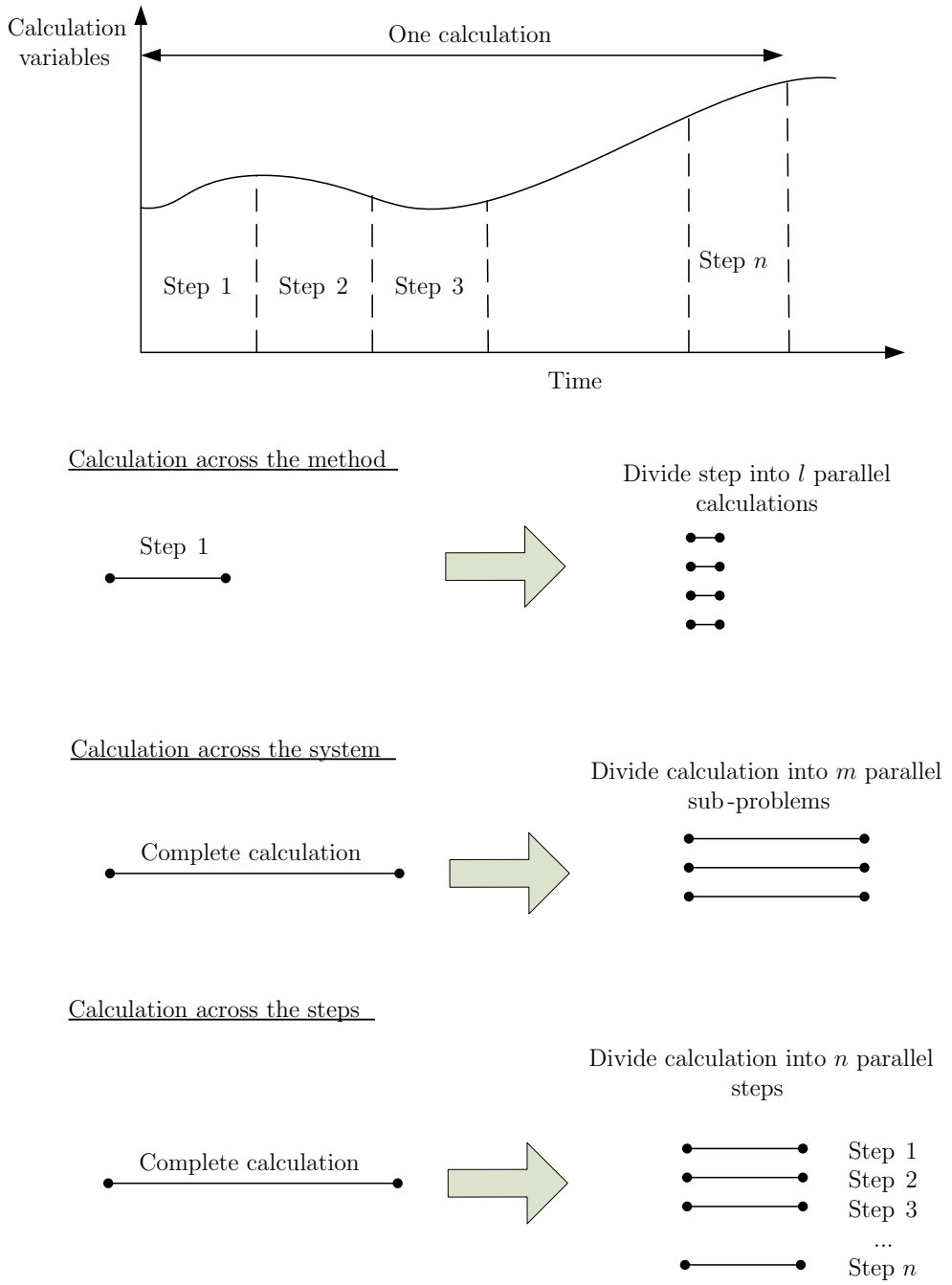

Figure 4.1: An illustration of three areas of parallelism. Consider a calculation which is made up of n steps. Parallelism across the method breaks each of the steps into l computations which can be carried out in parallel. Parallelism across the system breaks the whole calculation down into m separate sub-problems and solves them in parallel. Parallelism across the steps solves n multiple steps simultaneously.

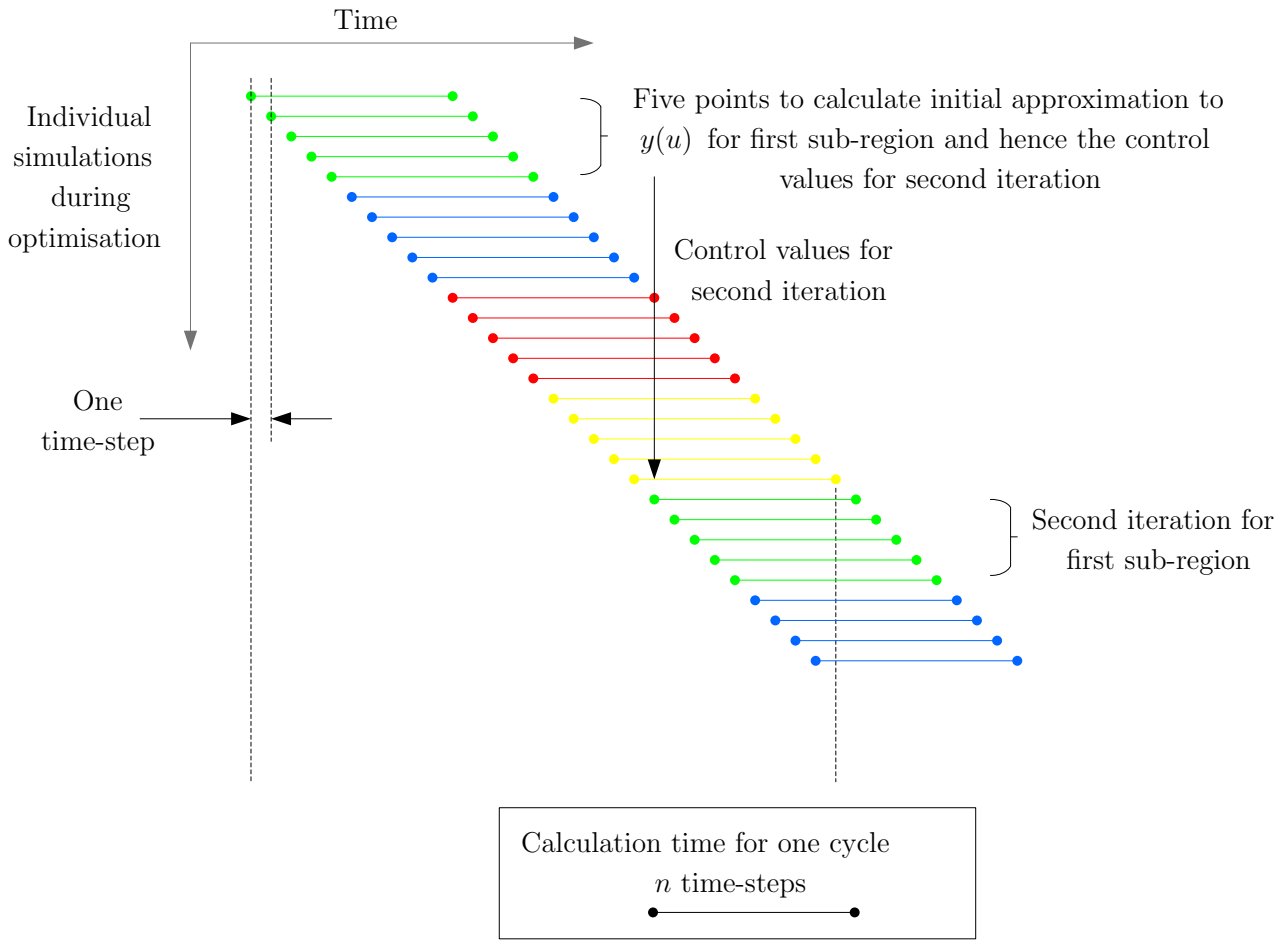

Figure 4.2: Configuration of the pipelined implementation of the PTR algorithm. The four different colours refer to the four sub-regions being assessed, as described in Appendix B, Figure B.2.

for controlling a bicycle. This example also develops the integrated toolchain which can take a graphical modelling language (Simulink) and assign modules between a microprocessor and FPGA.

The pipelined system model is illustrated in Figure 4.3, where the response of the discrete state-space model, Equation 4.1, is calculated over a four step prediction horizon. The control input u is constant across the horizon. The lower plot illustrates the differences between controller time-step, which is the execution time on the real-time controller, and model time-step, which is the simulation step over the prediction horizon.

$$x_{k+1} = Ax_k + Bu_k \tag{4.1}$$

In the MPC bicycle example, this pipelined model is implemented on the FPGA which can predict the system response much faster than a sequential implementation. The PTR in this case is capable of optimising the cost function within three iterations, stabilising the unstable bicycle. This example also allowed the FPGA toolchain to be commissioned prior to implementation of a full combustion model and controller.

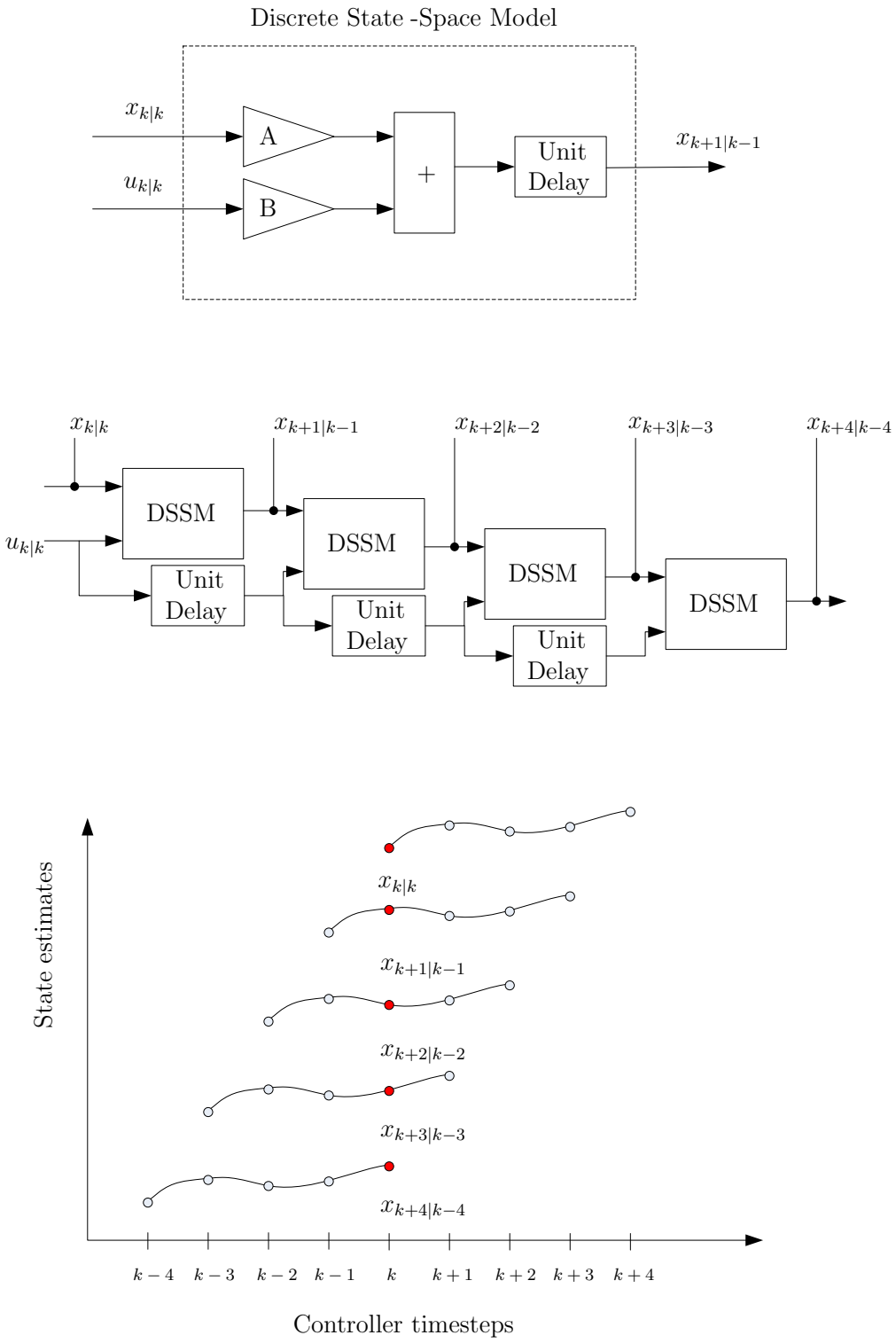

Figure 4.3: Implementation of the discrete state-space model (DSSM) in a pipeline. Where $x_{k+1|k-1}$ refers to the state x at *model* time-step $k+1$, derived from an initial state at *controller* time-step $k-1$. Lower plot shows the calculation of the model states over the prediction horizon, each line relates to a calculation from a different initial state. The highlighted red points are the values along the pipeline at *controller* time-step k.

4.4 Combustion controller refinement

The key components of a real-time implementation have now been developed: a combustion model, an optimisation algorithm and parallel computing approaches. These components are brought together in this section and further refined for application in a vehicle demonstration.

4.4.1 Exploiting parallel computing to reduce model execution time

The combustion model structure, Figure 4.4, can be used to identify opportunities to introduce parallelism into the model implementation. Considering the three areas of parallelism in turn. Parallelism across the steps is not considered due to the sequential nature of the cycle simulations. Parallelism across the system has some potential. Once the main heat release is calculated, the individual emissions calculations and BMEP calculations can be calculated in parallel. Parallelism across the method offers the most potential since the emissions from each parcel of burnt fuel can be calculated in parallel at each timestep.

There are two other considerations in designing the model architecture: communication interfaces between processors and the requirement to have similar operations to parallelise. The size of the communication interfaces should be minimised to avoid excessive delays in the calculation. Considering the dimensions of the variables, Figure 4.4, it can be seen that the model can be divided in two (as shown by dashed line) after the main combustion calculation but before the emissions calculations. At this point the signals are only one-dimensional, and therefore are more suitable for dividing between different processors than the two or three dimensional signals, thus minimising the communication interface.

The second requirement was to identify identical calculations which are carried out repeatedly. The combustion model calculations are not directly suitable for coding in parallel since there are different modes during the combustion and the transitions vary from cycle to cycle. However, the emissions calculations are suitable since the same calculation is required for each timestep and for each parcel of burnt gas.

The resulting design places the upper part of the calculation in a micro-processor with the lower part of the calculation selected for implementation in parallel on an FPGA. The target hardware selected for this project is a System on a Chip which contains both

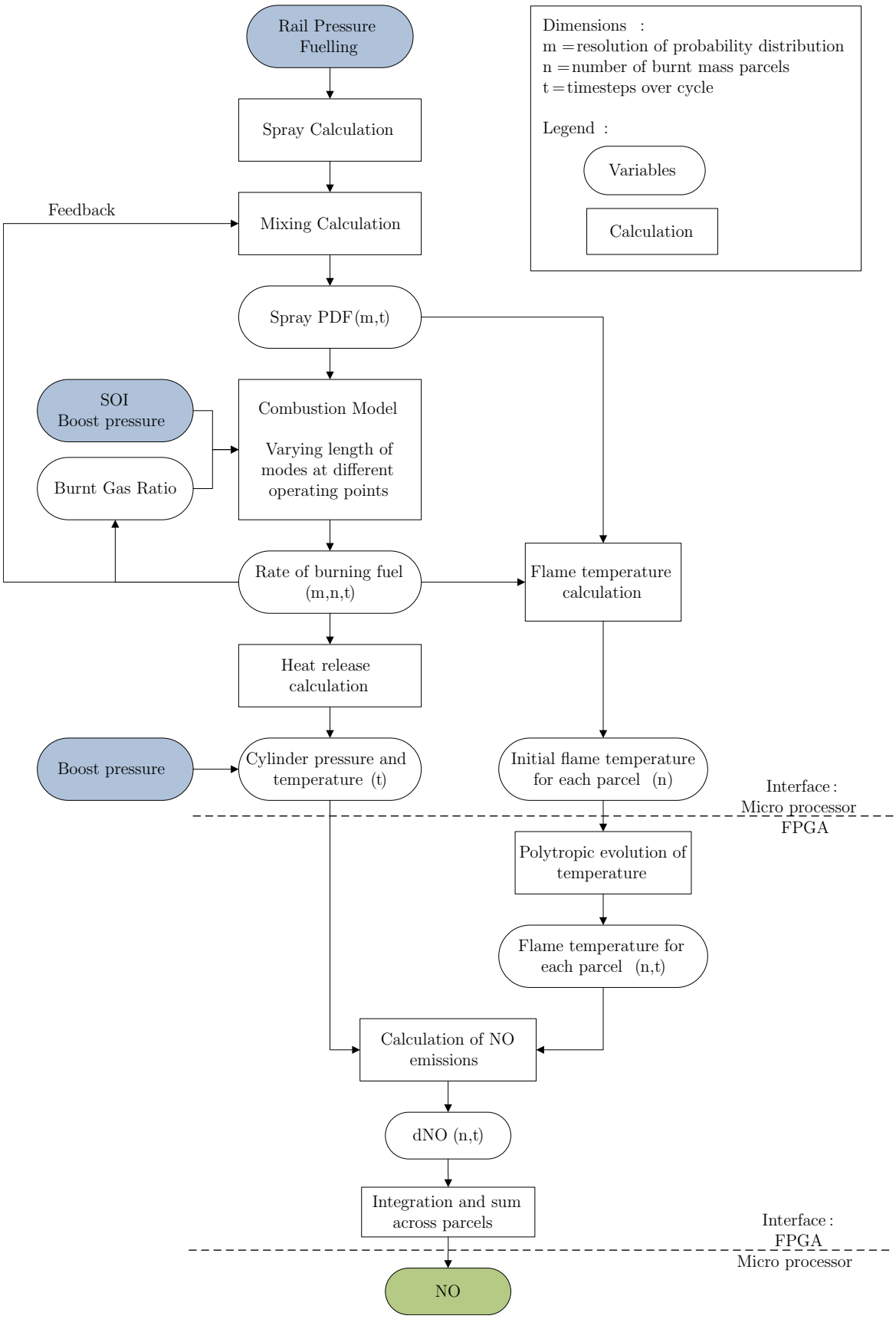

Figure 4.4: Overview of Combustion Model, highlighting an appropriate division between micro processor based calculations and FPGA based calculations. Variables have dimensions in brackets. Blue boxes are inputs and the green box is the output from the model.

the micro-processor and FPGA on the same piece of silicon, minimising communication delays.

4.4.2 Implementation of the emissions calculations on an FPGA

The emissions calculations for each parcel of burnt gas are independent and can be run in parallel, with the number parallel threads being limited by the FPGA selected. In this project, ten instances of the emissions calculation were included. There were more parcels than calculation threads so all of the parcels could not be calculated at once and additional code was developed to manage the data flows. For example if there are 100 parcels then the parallel threads need to be run ten times and the results combined to give the total NO from all the parcels. More details are provided in Appendix D and specifically, Figure D.3.

In addition to duplicate instances of code where parts of the model can be run in parallel, the implementation also makes use of pipelining the code where one calculation starts before the previous is completed, as described in Section 4.3.1. Again, data management is key to the implementation to ensure that the FPGA is fed with a stream of input data and the results are associated with the correct inputs.

Consider the calculation of emissions on the FPGA as shown on Figure 4.4. The calculation of emissions from a parcel of burnt gas requires calculation of the polytropic evolution of the temperature of the parcel, see Section 2.3.5, and the NO model, Equation 2.47, which was simplified by grouping the constants and reducing the equations to depend on the input variables, resulting in Equation 4.2 for the rate of change in NO concentration, [NO].

$$\frac{d[NO]}{dt} = f_1(T_{par}) \left(\frac{f_2(T_{par}) p_{cyl}^3 - [NO]^2 p_{cyl}}{f_3(T_{par}) p_{cyl} + f_4(T_{par}[NO])} \right) \tag{4.2}$$

where p_{cyl} is the cylinder pressure and f_i are look-up tables as a function of the parcel temperature, T_{par}.

The layout of the emissions model is shown in Figure 4.5 where the initial flame temperature of the parcel, T_{par}^0 and the vector of cylinder bulk temperatures, $\mathbf{T}_{cyl}$, are passed to the polytropic expansion block. This generates a vector of parcel temperatures, $\mathbf{T}_{par}$, which are passed to the d[NO] calculation. The d[NO] calculation is spread over several FPGA clock cycles as a pipelined function.

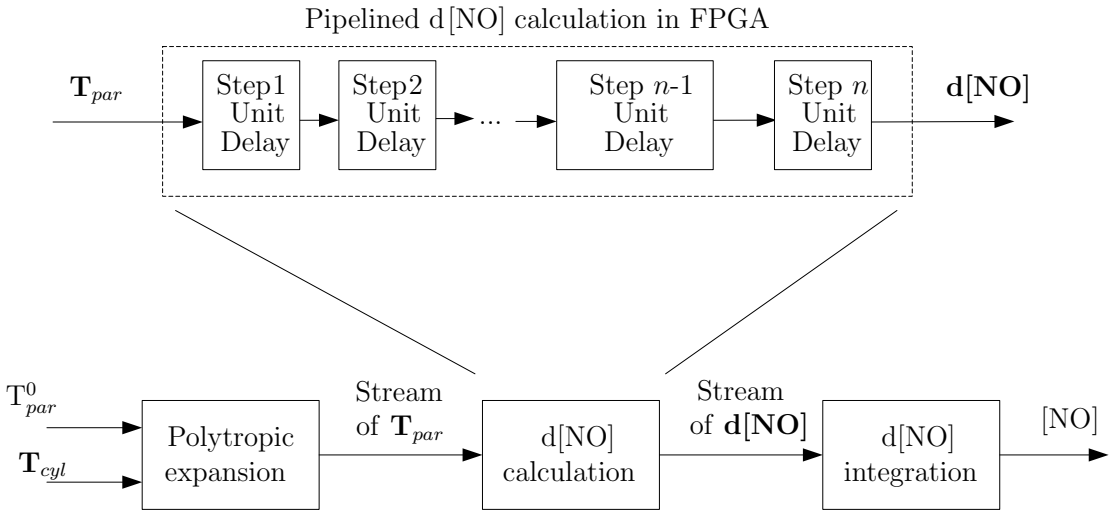

Figure 4.5: Implementation of the emissions model for one parcel of burnt gas. Upper diagram shows the pipelined model for the d[NO] calculation. Lower diagram shows the emissions calculations in the FPGA. Bold quantities are vectors.

4.4.3 Implementation of the optimisation on an FPGA

Once the model had been implemented on the FPGA, the model was calibrated and benchmarked. In particular, the resolutions (discretisation in crank angle and mixture resolution) were reduced as far as possible whilst maintaining accuracy. The resultant model execution speed was approximately 30 engine cycles per second or 30 ms for one cycle simulation, allowing approximately three cycle calculations in the time between EGR control updates (approximately 100 ms). The limiting calculation was the heat release estimation in the micro-processor.

This motivated a simplification of the PTR optimisation to minimise the number of cycle simulations even further. The simplified optimisation has one iteration, with one sub-region and one variable (EGR, since the engine controller only had a bypass hook for the EGR). The optimisation requires four NO simulations: at the steady-state set-point with the air system at steady-state, the steady-state set-point with the current conditions and two other points in the search region. A straight line is fitted to the NO response and used to calculate the minimum of the cost function in the same way as for the PTR algorithm, see Figure 4.6 for a graphical representation of the simplified optimisation algorithm. The heat release calculation on the micro-processor is the limiting calculation for execution speed.

Following experimental investigations, it was found to be beneficial to restrict the range of the control signal search around the set-point. This helped to increase the

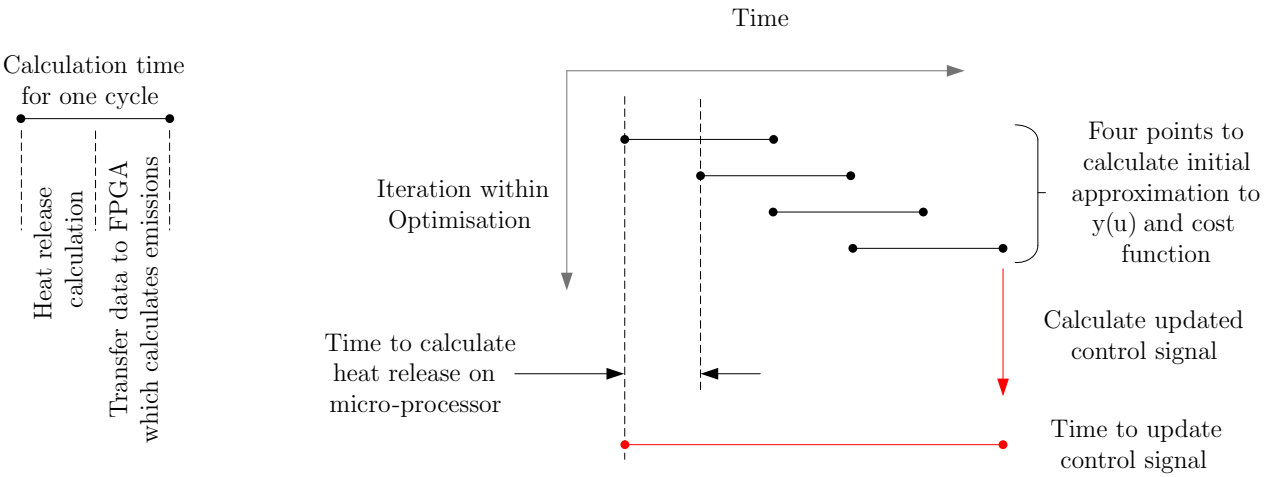

Figure 4.6: A timing diagram for the simplified optimisation algorithm. The left diagram shows the time required to calculate the NO which is made up of two parts: the heat release calculation on the micro-processor and the emissions calculation on the FPGA. The right diagram shows the time for one optimisation cycle, due to the overlap of the iterations, the total time to update the control signal is five times the duration of the heat release calculation, including the delays due to data management.

resolution of the search, since it was a smaller search region, though it did also reduce the authority of the controller.

The simplified optimisation with four cycle simulations takes approximately 150 ms (five times the duration of the heat release calculation), however since the input conditions are taken at the start of the optimisation period, the delay in response can be up to twice the optimisation period. If the input conditions change just after an optimisation has started, that optimisation must complete before the new optimisation can start. The worst case response is therefore approximately 300 ms which is above the upper limit (200 ms) of the target range but is the right order of magnitude to be considered for vehicle demonstration, especially at lower engine speeds.

The calculated values of NO versus EGR are taken into the optimisation algorithm and used to select the optimum value of EGR according to the approach described earlier in this section. A straight line is fitted to the three test points (note it is forced to go through the set-point to ensure convergence to the set-point). The line fit is used to calculate the minimum of the cost function. See Figures 4.7 and 4.8 for the curve fit to the NO emissions and the corresponding cost function. Figure 4.7 is for an operating point with the intake manifold pressure below the steady-state value. Here the optimisation selects an increase in EGR ratio to reduce the NO. Figure 4.8 is for the same operating point, with the intake manifold pressure at the steady-state value. Here the optimisation

selects the steady-state EGR value.

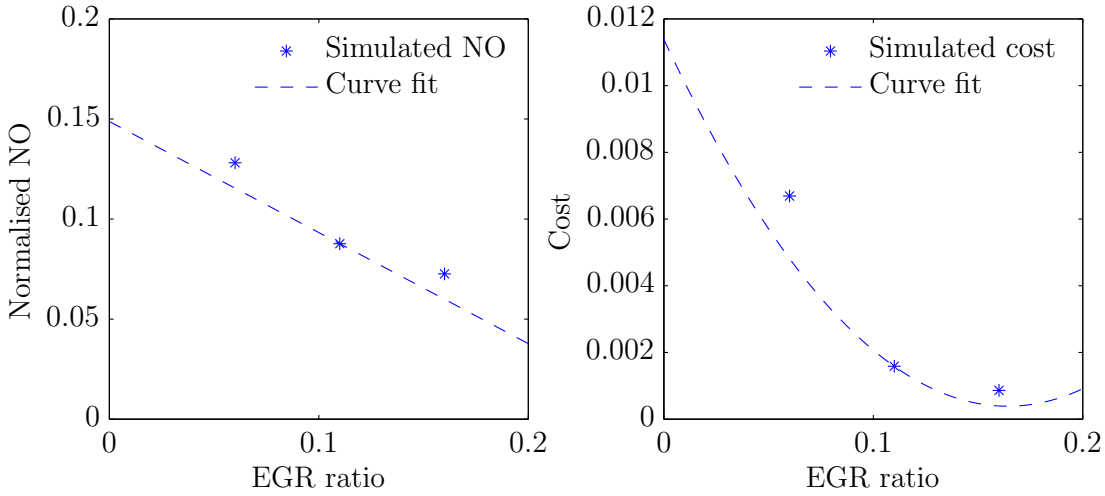

Figure 4.7: Cost and NO for first set-point

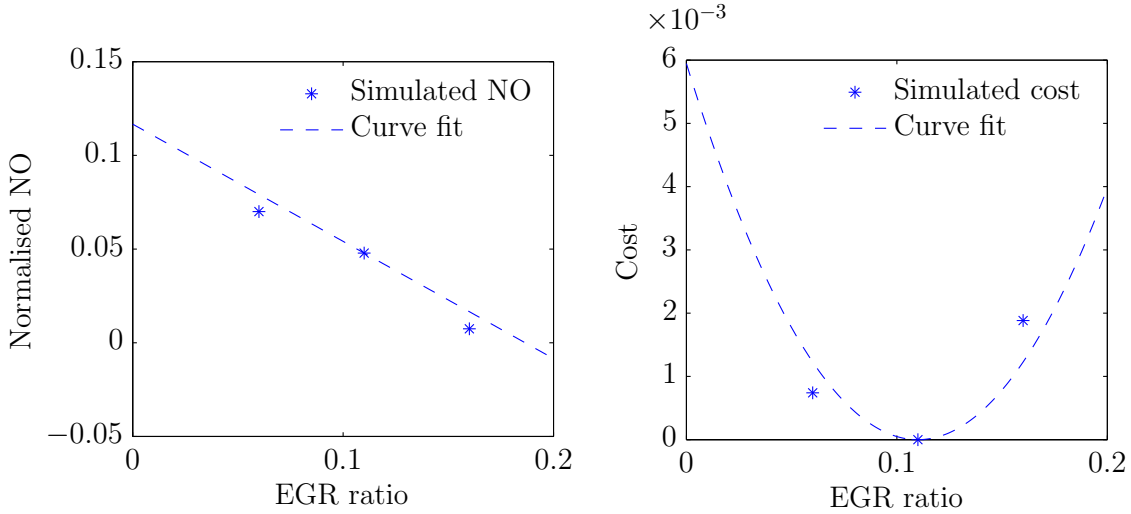

Figure 4.8: Cost and NO for second set-point

4.5 Vehicle testing

4.5.1 Implementation of predictive combustion control in a vehicle

The final stage in the development of the combustion controller was to implement the combustion control algorithm on an FPGA and a micro-processor. A communication interface was added to allow the controller to communicate with the engine control unit via 'hooks' into the production control strategy. The system was validated on the test bench, before integration into the full vehicle control system. The realisation of the combustion

controller in the vehicle brings together the combustion model, optimisation algorithm, communication interfaces and schedulers for the data flows. The layout of the control system is presented in Figure 4.9 with additional details of the vehicle implementation in Appendix D.

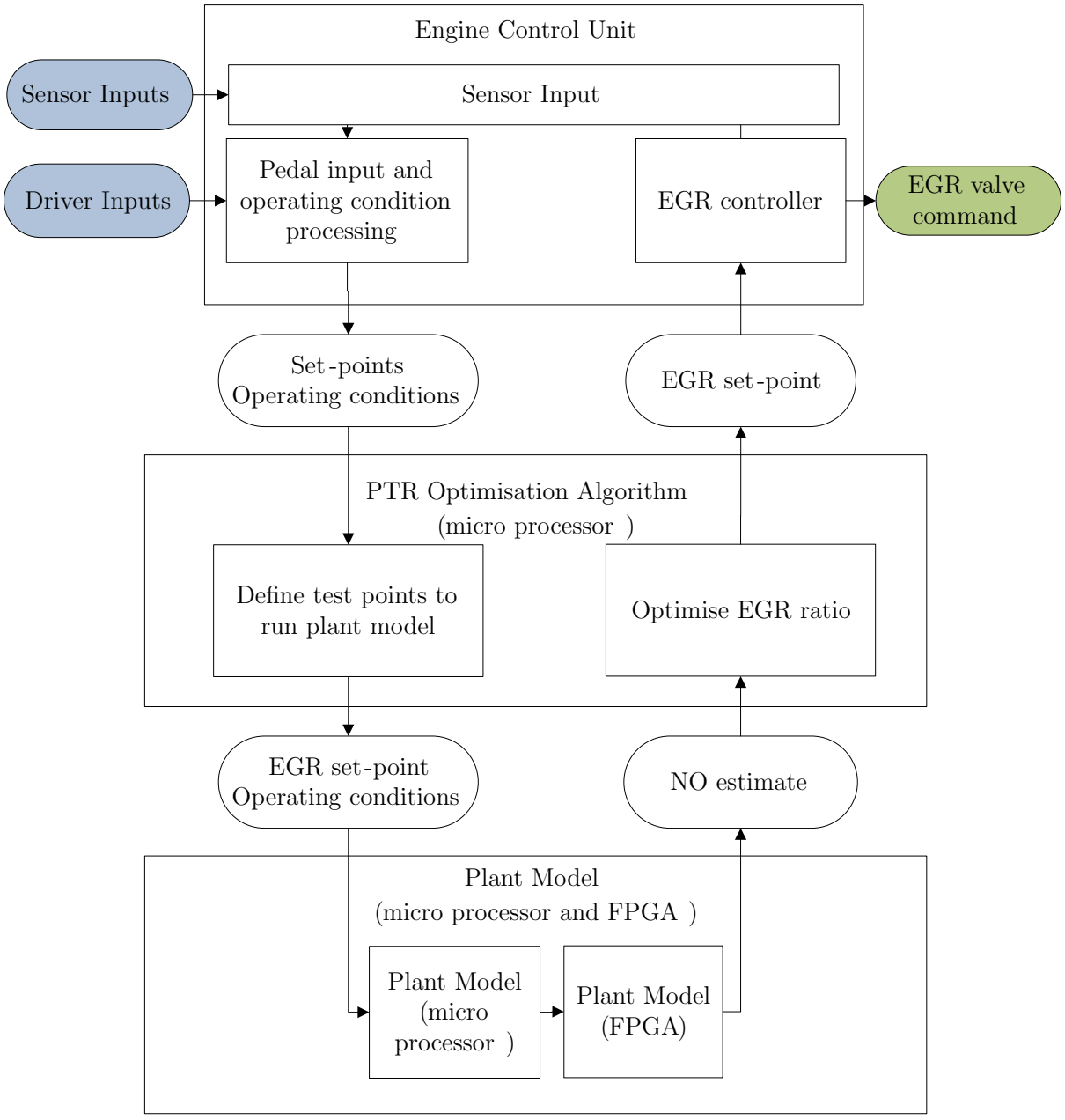

Figure 4.9: Overview of the combustion controller showing integration of FPGA, micro processor and engine control unit.

The vehicle was installed in an emissions test facility and driven over three different transient manoeuvres whilst recording emissions data: a step increase in load at constant speed, the Extra Urban Drive Cycle [1] and the US06 drive cycle [5]. See Figure 4.10 for the vehicle speed and engine load traces from the two drive cycles.

In the vehicle test facility, the vehicle is stationary when being tested, with the vehicle

inertia and aerodynamic drag applied to the powertrain with an electrical machine via a 'rolling road'. A variable speed fan in-front of the vehicle provides an airflow consistent with the vehicle speed to ensure the powertrain is cooled in a representative manner. Additional details of the test facility are given in Appendix D.3.2. The data was logged on several systems: engine control parameters were logged on a computer inside the vehicle, NO_x emissions were logged with a 'MEXA' emissions analyser and the soot emissions were logged with a soot meter. The data was synchronised with respect to recording times, however the reader should bear in mind there are time delays in the exhaust system and logging lines. For example, the time delay between an ECU control action and the measured response in emissions can be seen to be approximately 4 seconds, see Figure 4.11, where the start of the transient is at approximately 3 seconds and the response in NO_x emissions starts at approximately 6 to 7 seconds.

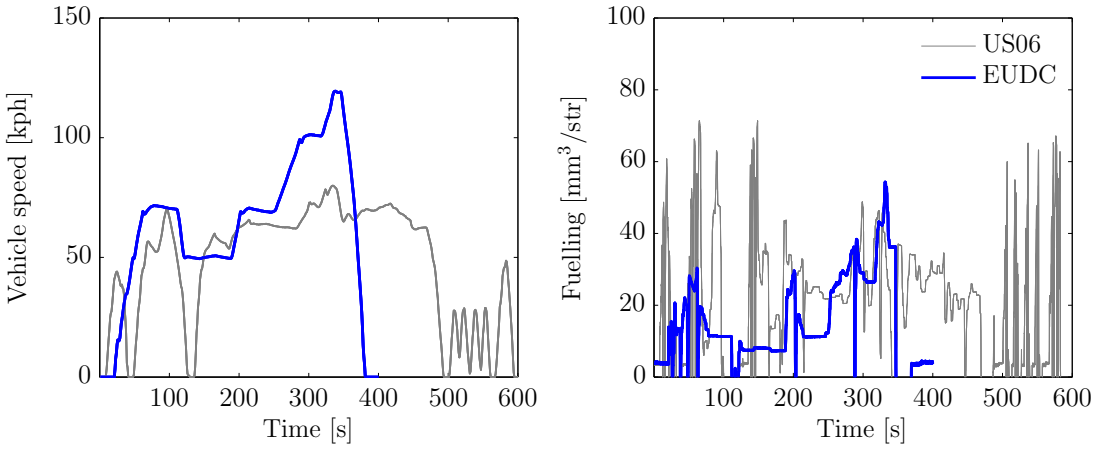

Figure 4.10: A comparison of the two drive cycles: the Extra Urban Drive Cycle (EUDC) and the US06 cycle, which are 400 and 600 seconds long respectively. The left hand plot shows the vehicle speed trace and the right hand plot shows the fuelling trace (which corresponds to the engine load).

4.5.2 Load step tests at constant speed

The first tests were carried out at a constant engine speed. The combustion controller was tested by applying a step increase in the engine load (fuelling), a similar test to the simulation developed in Chapter 3. The engine was held at a constant speed by the dynamometer and the load was varied with the accelerator pedal. During the step increase in load, the air system lagged the fuelling input due to the inertia of the turbocharger. The delayed air system response when combined with the conventional EGR controller, reduced the EGR. This led to a characteristic transient NO_x peak (see Figure 4.11, lower right plot). The load step test was carried out with the conventional EGR controller, followed by the same test with the combustion controller optimising the EGR valve position to minimise NO_x emissions. The combustion controller is able to achieve a significant reduction in NO_x emissions without a corresponding increase in soot. Note, the soot emissions for such a short transient were below the level measured by the analyser.

The results in Figure 4.11 are now reviewed in more detail, starting at the upper left plot. The fuelling (indicative of load request) rises rapidly to the target, with a slight step which is likely to have been caused by a smoke limiting strategy (in a similar fashion to the simulation example in Chapter 3). The upper right plot shows that the conventional EGR controller closes the EGR valve completely. This is the result of a feed-forward strategy which seeks to increase boost pressure as fast as possible and directs all the exhaust flow through the turbocharger. The combustion controller, however, limits the reduction in EGR because it is optimising NO_x and predicts that completely closing the EGR will result in a large NO_x spike.

The middle left plot shows the intake manifold pressure rising quickly in both cases. The result from with the original EGR controller is slightly quicker (as predicted in the simulation in Chapter 3). The middle right plot shows the actual EGR as estimated by the engine control unit. The results from both tests show that the EGR is effectively cut in both cases, with the EGR from the combustion controller returning to the target quicker than the conventional EGR controller. This can be explained by observing that the EGR rate is a function of both the EGR valve position and the pressure drop across the EGR circuit. The pressure drop is the difference between exhaust manifold pressure and intake manifold pressure. At the start of the transient, the intake pressure rises faster than the exhaust pressure which therefore reduces the EGR flow for both tests.

As the transient manoeuvre progresses the exhaust pressure rises and the pressure drop increases. Once the pressure drop has returned, the EGR valve has control authority over the EGR rate. The combustion controller keeps the EGR valve open and therefore the EGR returns quicker than with the conventional EGR controller.

The lower right plot shows the measured NO_x from the engine for the standard EGR controller and the combustion controller. The combustion controller has more EGR which has resulted in a significant reduction of NO_x compared to the EGR controller.

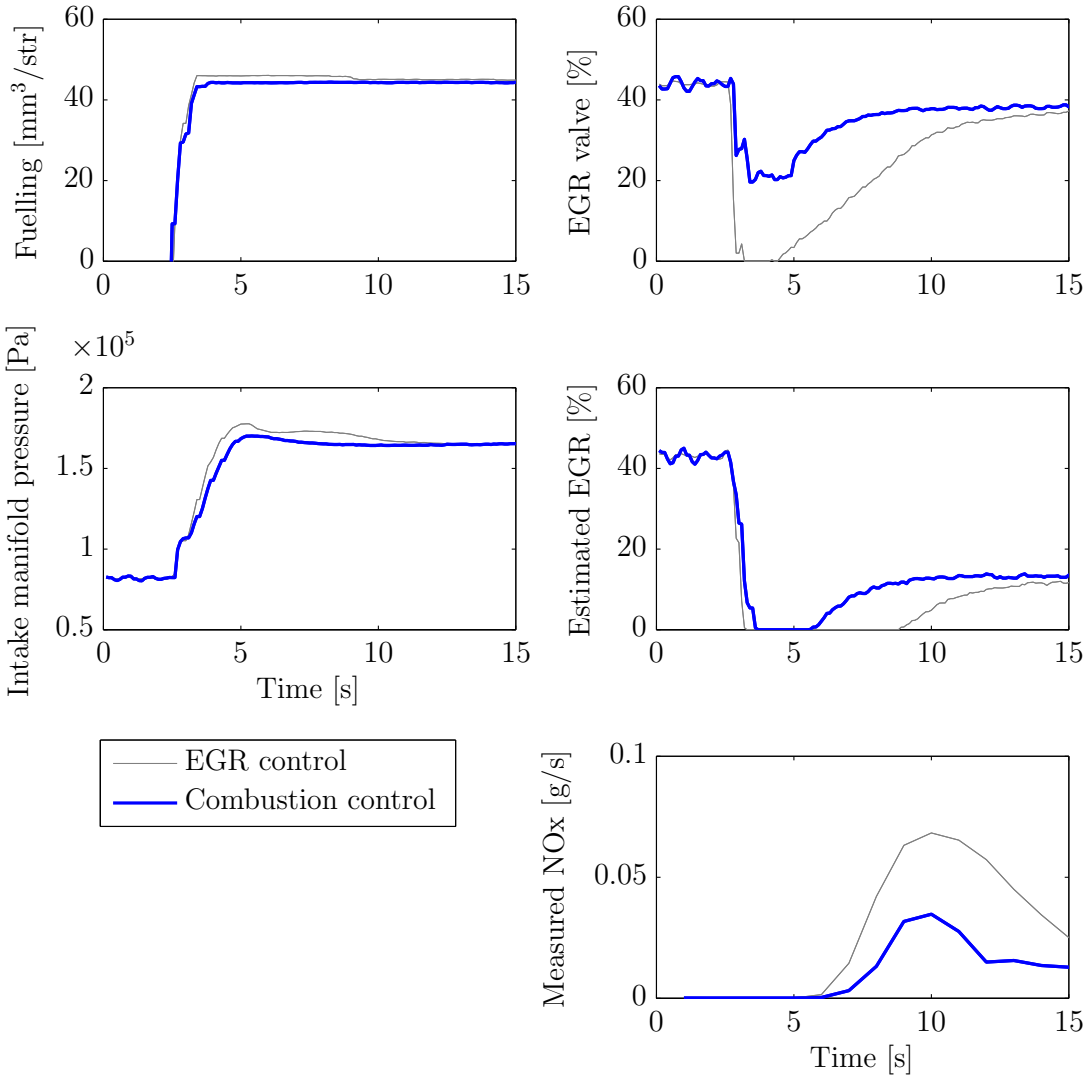

Figure 4.11: Vehicle test results from applying a step increase in engine load at a constant speed (1200 rev/min). A detailed explanation is provided in the text, with the key result being the Combustion controller delivers a significant reduction in NO_x emissions, shown in the lower right plot, by keeping the EGR valve open during the transient.

4.5.3 Transient tests over Extra Urban Drive Cycle (EUDC)

The vehicle was then driven over a typical European drive cycle, the Extra Urban Drive Cycle. This is the high speed part of the New European Drive Cycle and was selected because it has higher load transients than the low speed section. However, this cycle is still relatively gentle, and most of the time, the air system is able to keep up with the demanded boost pressure and corresponding airflow. This means that the system is operating close to the steady-state calibration values and the benefits of combustion control can only be seen during one or two manoeuvres. The overall response is very similar to the conventional EGR controller and therefore the controller was also tested using the US06 test cycle which is more aggressive, see Section 4.5.4. Over the EUDC, the vehicle tests were carried out with a conventional EGR controller and the Combustion controller. The results from these two tests are superimposed in Figures 4.12 and 4.13. Figure 4.12 shows the full cycle, with the data expanded for the higher load transient in Figure 4.13.

The format of the figures with the drive cycle results is as follows, starting at the top left plot and covering each row in turn. The top row shows the vehicle speed trace and the engine fuelling (representative of the engine load) and the second row shows the engine speed and the intake manifold pressure. These four quantities define the state of the engine and vehicle. In particular, it can be seen that the engine and vehicle speed traces are very close between tests, confirming good repeatability by the vehicle driver (an experienced test driver). From the engine speed trace, one can see the gear shifts, indicated by the saw tooth profile as the engine speed falls when the gear is shifted up. The fuelling traces differ slightly at high speed (see Figure 4.13) and may have contributed to the increase in NO_x for the EGR controller. The intake manifold pressure rises to a peak of around 2 bar during the high speed section.

The third row of plots contains the EGR valve signal and the estimated EGR from the ECU. The EGR valve signal reflects any differences in the EGR control strategies. For the EUDC test cycle, EGR is applied virtually throughout as this is not as aggressive as the load step test. There are limited differences between the two tests, for example at 330 seconds (Figure 4.13) where the EGR valve command for the conventional EGR controller goes to zero whilst the combustion controller holds the EGR rate and results in a reduction in the transient NO_x.

The fourth and fifth rows detail the emissions measurements, instantaneous and cumulative respectively, with the NO_x emissions on the left and soot emissions on the right. NO_x and soot measurements are dominated by high speed section and hence this section is expanded in Figure 4.13. Even in the high speed section, the differences between the two controllers are small and it can be concluded that when the engine is operating close to steady-state conditions, the combustion controller gives a similar result to the EGR controller.

Finally, the CO_2 emissions were measured and compared between the two controllers, see Table 4.1. Again the differences between the two controllers are small on this test cycle.

Description	NO_x [g]	Soot [mg]	CO_2 [g]
EGR control	2.22	17.3	964
Combustion control	2.21	18.0	960
Percentage change [%]	-0.5	4.0	-0.4

Table 4.1: Summary results from EUDC test cycle showing the percentage change when using the combustion controller.

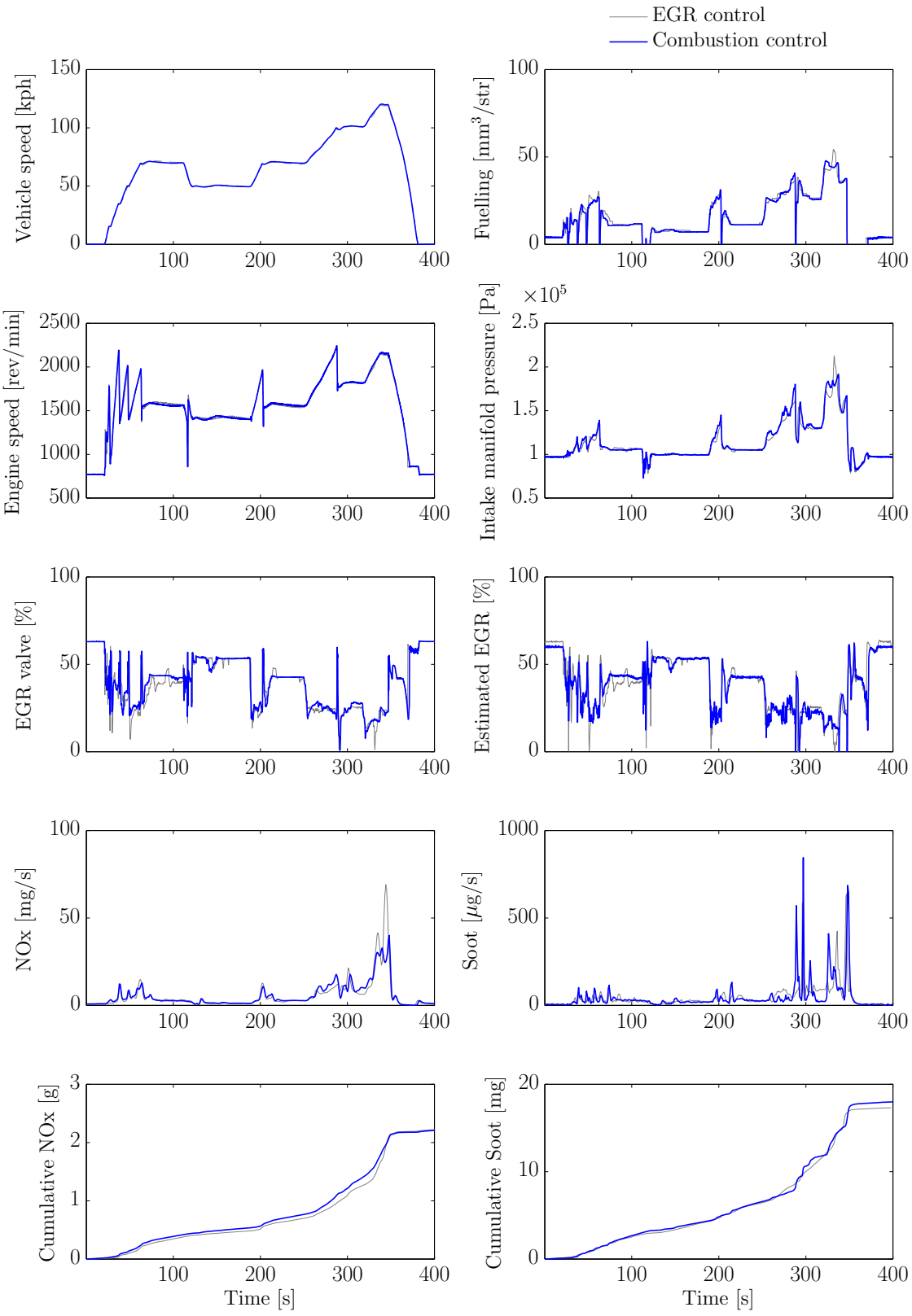

Figure 4.12: Vehicle test results over EUDC test cycle, this plot shows the full cycle. A detailed explanation is provided in the text, with the key result being the controllers deliver very similar results over the standard European test cycle (see the cumulative plots at the bottom of the figure). This is to be expected since the vehicle is driven gently and the engine is operating close to the steady-state set-points.

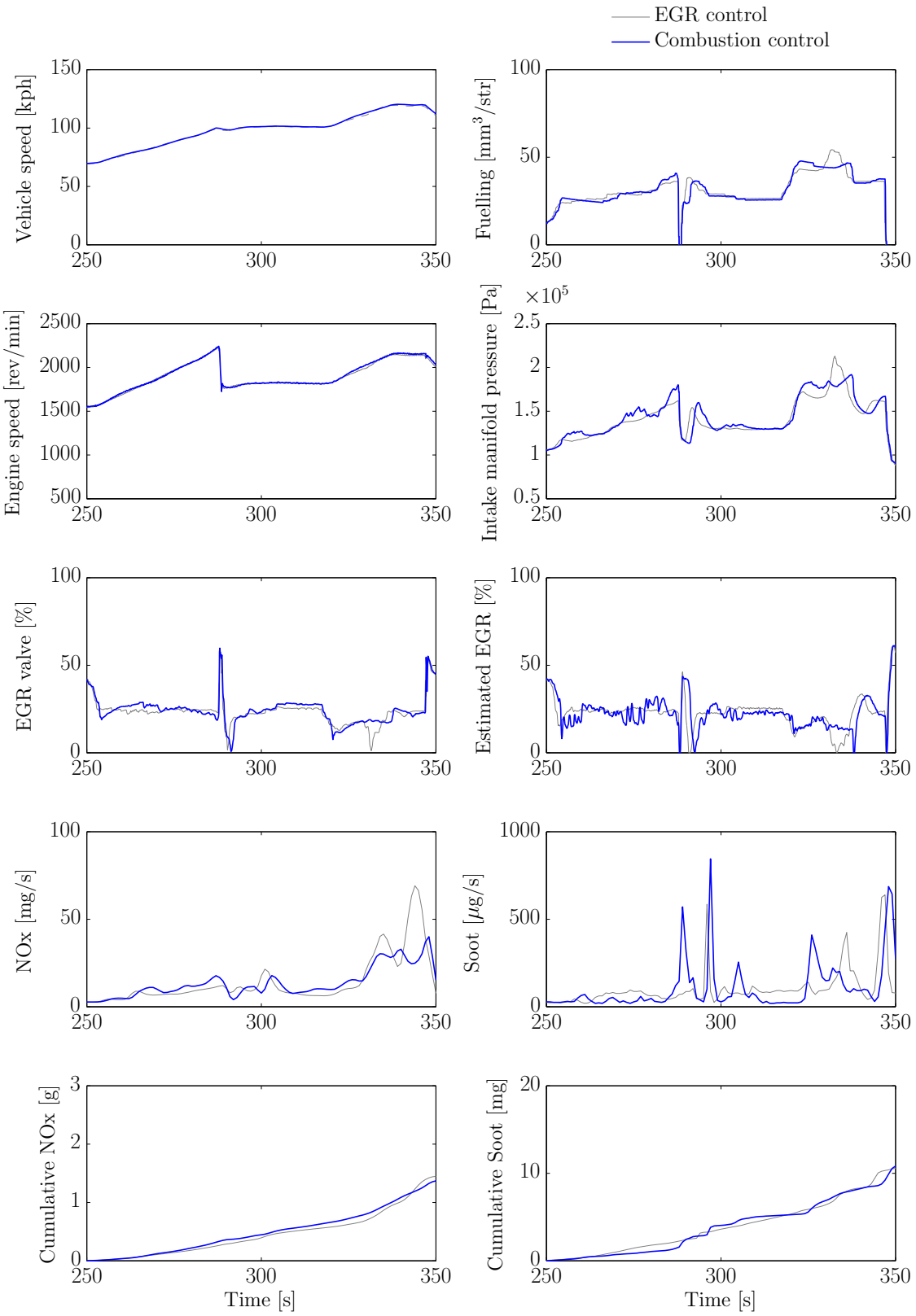

Figure 4.13: Vehicle test results over EUDC test cycle, this plot shows the high speed acceleration phase where the engine load is the highest. Even in this section, the air-system is able to keep up with the engine acceleration and there is little difference between the two controllers.

4.5.4 Transient tests over the US06 test cycle

The combustion controller was evaluated over the US06 test cycle which is a highly transient test cycle. The vehicle tests were carried out with a conventional EGR controller and the combustion controller. The results from these two tests are superimposed in Figures 4.15 to 4.17. Figure 4.15 shows the full cycle, with the data expanded for the initial and final transients in Figures 4.16 and 4.17 respectively.

The cumulative emissions are summarised in Table 4.2.

Description	NO_x [g]	Soot [mg]	CO_2 [g]
EGR control	6.25	58.6	2480
Combustion control	5.41	68.1	2305
Percentage change [%]	-13	16	-7

Table 4.2: Summary results from US06 test cycle showing the percentage change when using the combustion controller.

The format of the plots is the same as for the EUDC results with the vehicle speed, fuelling, engine speed and intake manifold pressure traces in the top two rows. The vehicle speed profile shows the repeated accelerations with little steady-state operation. The fuelling, engine speed and intake manifold pressure values for the US06 test cycle are all significantly greater than the previous tests, see Table 4.3.

	Load step	EUDC	US06
Fuelling [mm^3/str]	45	50	70
Engine speed [rev/min]	1200	2200	3500
Intake manifold pressure [bar]	1.8	1.9	2.5

Table 4.3: Approximate peak values from the different test cycles.

The remaining three rows of Figures 4.15 to 4.17 show the EGR and emissions data which is highly dynamic and best reviewed on the expanded Figures 4.16 and 4.17. Correlating the cause and effect can be difficult on such a transient cycle since in addition to the time delays between the EGR valve movements and the emissions measurements, discussed in Section 4.5.1, there are rapidly fluctuating mass flows and pressures. One approach is to correlate the peaks in emissions with transient manoeuvres where the EGR valve is closed. At these points the NO_x emissions are likely to be high, with no EGR, and the soot emissions may also be high, since the fuelling is high whilst the air system may be delayed, causing a shortage of air for the injected fuel and potential incomplete combustion.

Consider Figure 4.16, where the main peak in NO_x emissions at 23 seconds is probably caused by the estimated EGR being zero at around 20 seconds. The combustion controller can be seen to request more EGR as the transients ramp in at both 10 seconds and 17 seconds. This has probably reduced the NO_x spikes during this transient. In contrast, the soot emissions are greater with the combustion controller. It should be noted that in the case of the soot measurements, there is a larger distance (and hence delay) from the engine to the instruments. The soot peak at 25 seconds is significantly larger for the combustion controller and this could have been caused by the increased EGR at 17 seconds coupled with the slower intake manifold pressure response at around 19 seconds which would indicate a reduced intake of fresh air leading to incomplete combustion.

Over the full cycle and also over the expanded transient manoeuvres, the cumulative plots (at the bottom of the page) and Table 4.2 show that NO_x and CO_2 emissions have been significantly reduced whilst there is a corresponding increase in soot emissions. Diesel engines are known to have a NO_x versus soot trade-off characteristic, see Figure 2.21, where increasing the EGR ratio reduces the NO_x emissions due to the factors described in Section 1.2.2, but also reduces the amount of fresh air which can lead to incomplete combustion and an increase in soot emissions. The NO_x versus soot trade-off characteristic is often referred to as a Pareto curve, which describes the optimum trade-off between a number of variables, see Figure 4.14. Comparing the different transients during the US06 cycle, it can be seen that the first transient has a similar increase in soot compared to the decrease in NO_x whereas the final transient has a smaller increase in soot emissions when compared to the reduction in NO_x emissions.

The inclusion of soot emissions in the cost function would allow the controller to carry out these trade-offs automatically, potentially reducing the NO_x emissions on the final transient whilst leaving the first transient closer to the steady-state set-points. This may be illustrated in Figure 4.14, assuming that these manoeuvres are operating close to the Pareto curve.

The combustion controller offers the development engineer the opportunity to calibrate the emissions trade-off by tuning the relative weights in the cost function, rather than adjusting EGR and boost pressure set-points iteratively to achieve a given trade-off. Allowing the combustion controller to automatically seek out improvements in NO_x emissions where the engine is less sensitive to soot emissions.

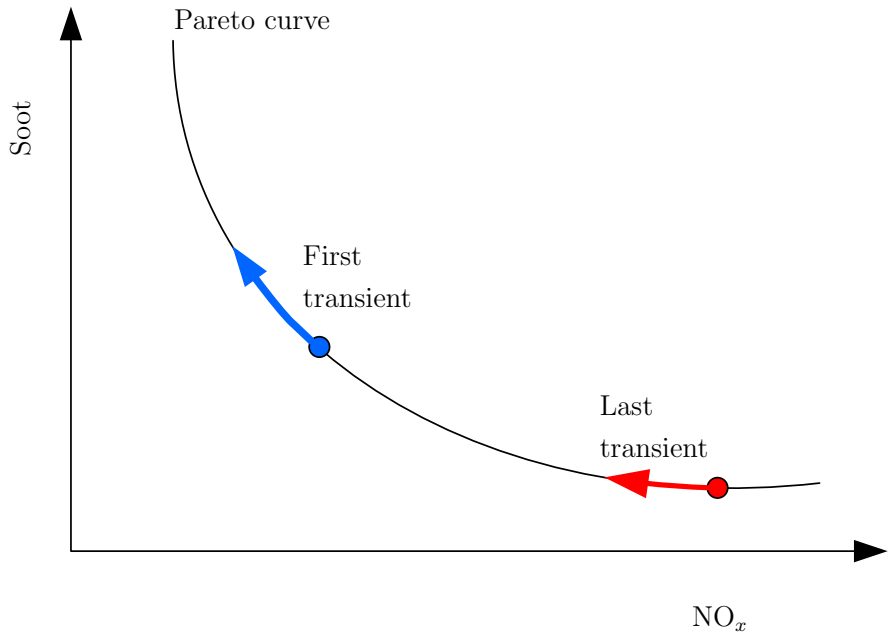

Figure 4.14: The typical soot - NO_x Pareto curve showing the potential locations of the two highlighted transients, blue referring to the first transient and red referring to the last transient. The circles signify the soot-NO_x trade-off for the EGR controller and the arrow shows the response of the engine over the same manoeuvre with the combustion controller. Note: the actual values of the soot and NO_x vary for each manoeuvre.

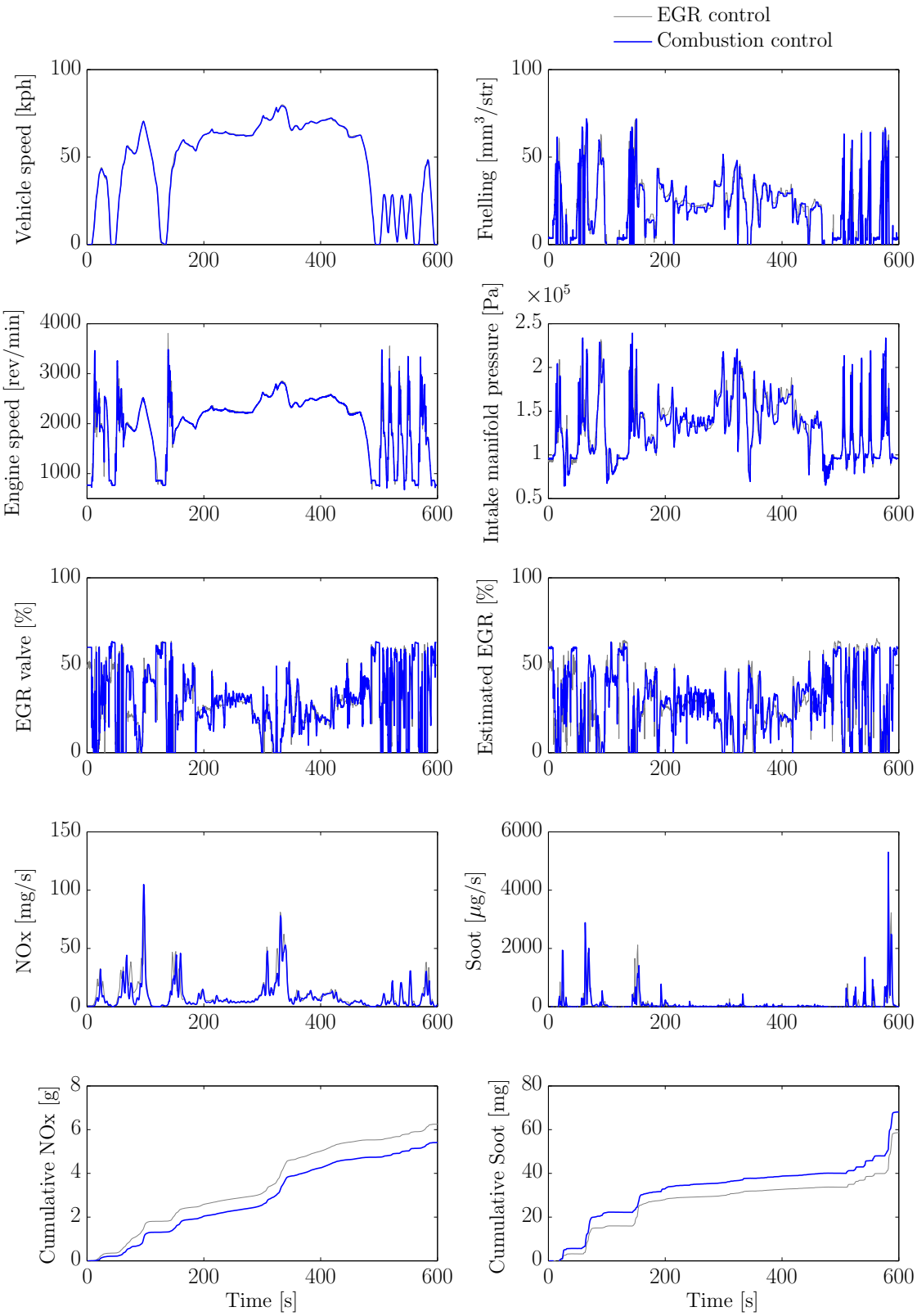

Figure 4.15: Vehicle test results over US06 test cycle, plot showing full cycle. A detailed explanation is provided in the text, with the key result being the combustion controller increases the EGR during transients and reduces NOx emissions by 13 % whilst the soot emissions increase by 16 %.

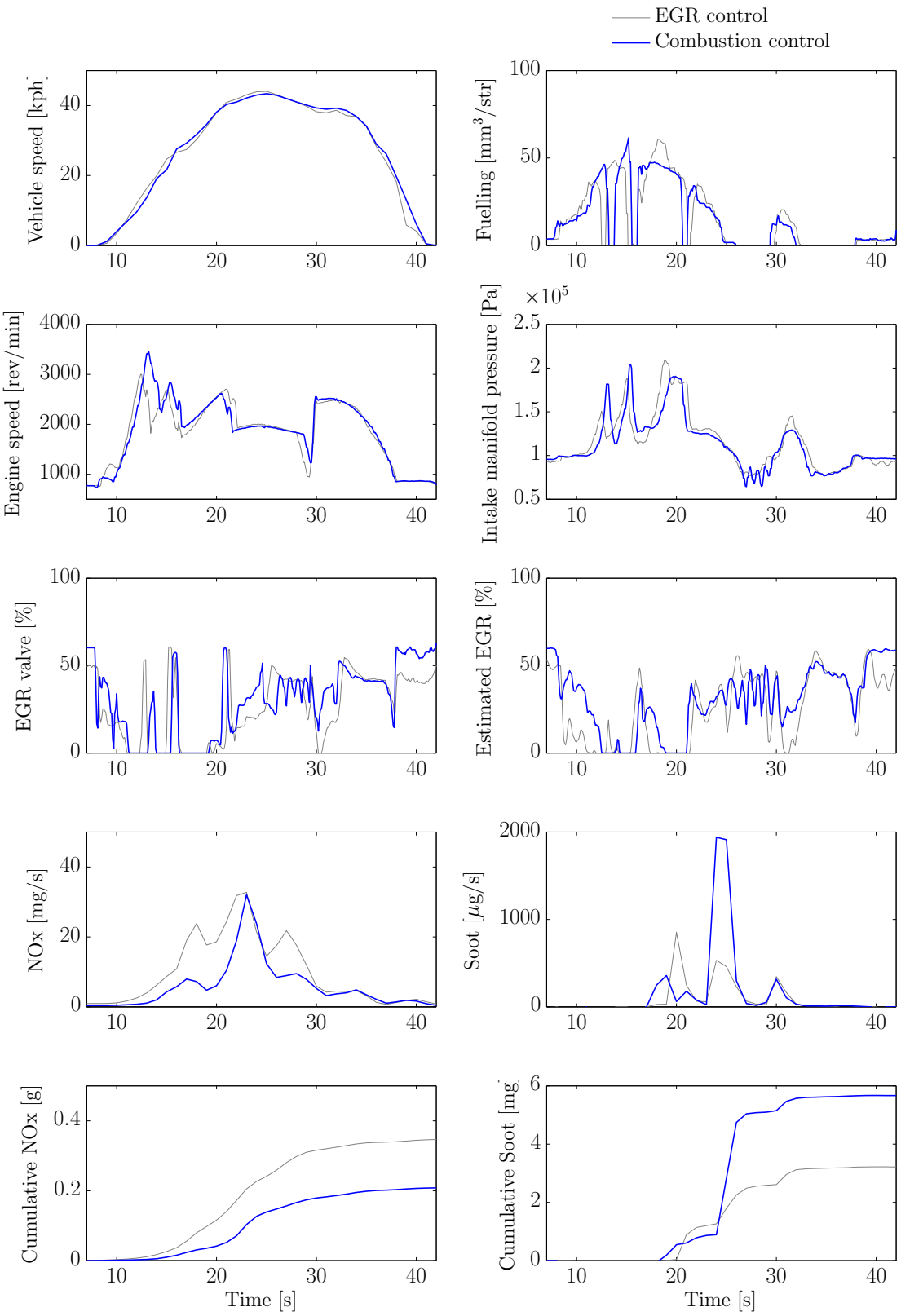

Figure 4.16: Vehicle test results over US06 test cycle, expanding on the first transient manoeuvre. Over this manoeuvre, the combustion controller increases the EGR during transients and reduces NO_x emissions by 40 % whilst the soot emissions increase by 76 %.

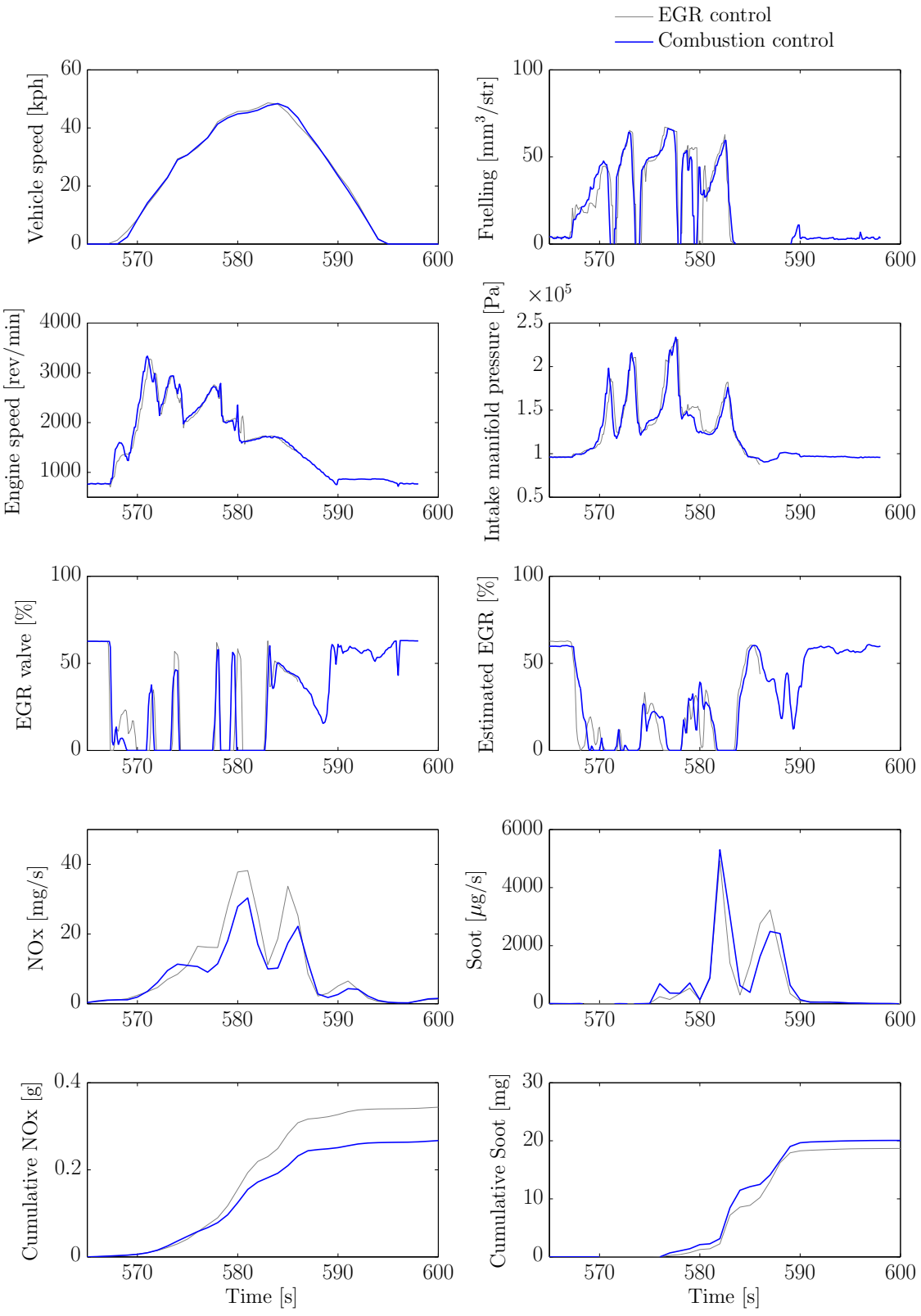

Figure 4.17: Vehicle test results over US06 test cycle, expanding on the last transient manoeuvre. Over this manoeuvre, the combustion controller increases the EGR during transients and reduces NO_x emissions by 22 % whilst the soot emissions increase by 7 %.

4.6 Conclusions: Real-time implementation of combustion control

The vehicle testing has demonstrated that a predictive combustion model can be made to run fast enough to be used in a predictive control algorithm and used to optimise and reduce the NO_x during a load transient. The algorithm was also demonstrated in a vehicle over the EUDC and US06 test cycles. Over the EUDC it gave a similar result to the baseline EGR controller. Over the US06 it reduced the NO_x and CO_2 but increased the soot, effectively moving the soot NO_x trade-off to lower NO_x but higher soot operating points. With the use of Diesel Particulate Filters, the reduction in NO_x may be worth more than the increase in soot, especially if the increase can be minimised as in the last transient manoeuvre from the US06 cycle. It should also be noted that the combustion controller here was implemented with only NO_x emissions in the cost function with only the EGR valve used as an actuator. A next step would be to include the soot model and present the engine developer with a calibration based on emissions trade-offs, rather than, as is the current practice, setting fuel injection parameters, EGR and boost pressure set-points and inferring the impact on emissions. In this case, the controller can be used to seek out points where the NO_x emissions can be reduced with minimal increase in soot.

In this work, the combustion model was calibrated on a different engine to that used for the vehicle testing. This suggests that the two engines may have a similar response, however the performance may be improved with the combustion model calibrated to the actual engine. This was not possible in the scope of this project since it would require extensive engine testing with cylinder pressure measurements, however this could be the subject of future work when the data is available.

The recent advances in integrated development toolchains have been exercised on a complex system whilst providing a relatively straightforward interface to the user. This allows the general Simulink user to exploit the benefits of an FPGA without having to become fully fluent in VHDL, in the same way that Simulink users do not have to be fully fluent in C code to generate embedded code. However, the use of the FPGA does require an understanding of the FPGA characteristics to ensure the generated VHDL is appropriate.

Finally, optimisation algorithms with low numbers of iterations were studied and

developed for this application. The optimisation was demonstrated on an FPGA, controlling an unstable bicycle with a model predictive controller, followed by the implementation of a simplified optimisation for the combustion controller.

Chapter 5

Optimisation of fuel injection parameters

5.1 Objective for fuel injection optimisation

The objective of this chapter is to investigate the use of control orientated models of combustion processes to calibrate fuel injection parameters. The calibration of these parameters involves selecting the timing, quantity, profile and rail pressure and so on, to minimise the cost function in an optimal control problem. One criterion might be minimising the fuel consumption whilst respecting constraints on the maximum cylinder pressure. The resulting optimal parameters can provide a 'desktop' calibration for the fuel injection system prior to engine testing. This approach could then be extended to the calibration of EGR and boost pressure set-points.

Current best practice is to calibrate off-line using experimentally derived stochastic process models that capture the response of the engine to fuel injection parameters. Stochastic process models have been developed over many years to provide a better fit to non-linear engine responses than lower order polynomials, see [78]. However, this approach requires experimental data that may not be available at early stages in the engine development programme.

This chapter starts with a brief overview of optimal control, followed by the application of non-linear optimisation to the Wiebe model (see Section 2.2.1). Once the optimisation approach has been demonstrated with the simple model, the optimisation is extended to a more realistic combustion model incorporating an ignition delay, pre-mixed combustion and diffusion combustion.

5.2 Introduction to optimal control and optimisation approaches

The solution of optimal control problems using general purpose numerical solvers is widespread. These solvers have been applied to a wide range of problems, optimising space flight trajectories, cancer treatment and Formula One cars, see [70] and [55]. The application of these general purpose solvers to new problems, such as combustion control, requires the careful consideration of a number of issues related to: ensuring the problem is compatible with the general purpose solver, discretisation techniques, scaling, treatment of non-smooth functions, the computation of gradients and mesh refinement.

This section starts with the general formulation of the optimal control problem, see [16] for a more detailed explanation.

5.2.1 Definition of an optimal control problem

Consider a non-linear, time varying dynamic system described by the state equation:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{q}, t) \tag{5.1}$$

where $\mathbf{x}(t)$ is a vector of the system states, $\mathbf{u}(t)$ is a vector of control actions and $\mathbf{q}$ is a vector of fixed parameters. The outputs from the system are described by:

$$\mathbf{y} = c(\mathbf{x}(t), \mathbf{u}(t), \mathbf{q}, t) \tag{5.2}$$

over a time period $t \in [t_0, t_f]$, the state trajectory should satisfy state constraints at all times. Defining $\mathcal{X}$ as the set of admissible states, then $x \in \mathcal{X}$. In a similar manner, the control actions are constrained to a set of admissible control actions, $u \in \mathcal{U}$, the outputs are constrained to a set of admissible outputs, $\mathbf{y} \in \mathcal{Y}$, and the parameters are constrained to a set of admissible parameters, $\mathbf{q} \in \mathcal{Q}$.

A cost function can be introduced to define a quantity that can be used to assess the performance of different control actions. The performance can be assessed over the time period and may take a variety of forms depending on the requirements of the control system designer. Examples of performance measures could include; the time taken to move between A and B, the energy required to move between A and B or the noxious emissions generated between A and B and so on. The Mayer form of the cost function is:

$$J = \Phi(\mathbf{x}(t_f)) \tag{5.3}$$

where t_f is the end time (t_f may be constrained or free depending on the problem formulation) and Φ is a scalar function.

The optimal control problem can now be stated as follows, after [51]: Find the admissible control function, $\mathbf{u}^*(t) \in \mathcal{U}$, and parameters, $\mathbf{q}^* \in \mathcal{Q}$, which cause the system, Equation 5.1, to follow admissible state and output trajectories, $\mathbf{x}^*(t) \in \mathcal{X}$ and $\mathbf{y}^*(t) \in \mathcal{Y}$, that minimise the cost function, Equation 5.3; the asterisk refers to extremal or optimal quantities.

From standard texts for optimal control, for example [51], the first-order necessary conditions for optimality are given by the Pontryagin minimum principle:

$$u^* = \arg\min_u \mathcal{H}(x, \lambda, u, t) \tag{5.4}$$

where $\mathcal{H}$ is the Hamiltonian function

$$\mathcal{H}(x, \lambda, u, t) = \lambda^T f. \tag{5.5}$$

and λ is the costate vector. Using this notation, the state (5.1) and costate dynamics are given by:

$$\dot{x} = \nabla_\lambda \mathcal{H} \qquad x(t_0) = x_0 \tag{5.6}$$

$$\dot{\lambda} = -\nabla_x \mathcal{H} \qquad \lambda(t_f) = \nabla\Phi(x(t_f)). \tag{5.7}$$

Equations (5.6) and (5.7), in combination, form a two-point boundary value problem in which the initial state and the terminal costate are specified. In the case where the control u lies interior to $\mathcal{U}$, assuming the existence and continuity of the relevant partial derivatives, (5.4) implies the strong form of the minimum principle:

$$\nabla_u \mathcal{H} = 0 \tag{5.8}$$

The optimal control problem can be solved using either direct or indirect methods [16]. Direct methods discretise the original problem and translate it into a non-linear programming problem, while indirect methods solve numerically the first-order necessary conditions described above.

5.2.2 Discretisation techniques

For several reasons, a direct pseudospectral numerical method based on Legendre-Gauss-Radau collocation has been applied to the optimisation of the fuel injection control using the general solver, *GPOPS-II*. This approach transforms the timescale to $\tau \in [-1, 1]$ and treats the state and controls as parameters on a finite collocation or implicit integration mesh determined by N Radau points $(\tau_0, \ldots, \tau_N)$ that are roots of $P_{N+1}(-\tau) + P_N(-\tau)$, where $P_N(\tau)$ is the Nth-degree Legendre polynomial. These roots lie in the interval $[-1, 1)$ with one root at $\tau_0 = -1$ always. Using this discretisation, the state is approximated using a basis of Lagrange polynomials, $\mathcal{L}_i(\tau)$:

$$\mathcal{L}_i(\tau) = \prod_{\substack{j \neq i \\ j=0}}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \qquad i = 1, \cdots, N \tag{5.9}$$

giving the following approximation for the state in Equation (5.1):

$$x(\tau) \approx \begin{bmatrix} \mathcal{L}_0(\tau) & \cdots & \mathcal{L}_N(\tau) \end{bmatrix} \begin{bmatrix} x(\tau_0) \\ \vdots \\ x(\tau_N) \end{bmatrix} \tag{5.10}$$

The differential of (5.10) with respect to time, can be written:

$$\begin{bmatrix} \dot{x}(\tau_1) \\ \vdots \\ \dot{x}(\tau_N) \end{bmatrix} \approx \begin{bmatrix} \dot{\mathcal{L}}_0(\tau_1) & \cdots & \dot{\mathcal{L}}_N(\tau_1) \\ \vdots & \ddots & \vdots \\ \dot{\mathcal{L}}_0(\tau_N) & \cdots & \dot{\mathcal{L}}_N(\tau_N) \end{bmatrix} \begin{bmatrix} x(\tau_0) \\ \vdots \\ x(\tau_N) \end{bmatrix}. \tag{5.11}$$

Consider the matrix in (5.11), if the first column is ignored, the remaining matrix is non-singular with its inverse being a Legendre-Gauss-Radau integration matrix. This discretisation allows the transformation of the optimal control problem from Section 5.2.1 into a non-linear programming problem, drawing on the established relationship, [38], between the Karush-Kuhn-Tucker optimality conditions for NLP problems and the first-order conditions for optimal control, Equation (5.8). *GPOPS-II* calls a general NLP solver, for example IPOPT, to complete the solution.

5.2.3 Computation of gradients

The preparation of the optimal control problem for solution with IPOPT requires the calculation of the derivatives of the objective function and constraints with respect to the states and controls. In this study, a Matlab based automatic differentiation package, ADiGator [70], was used to compute the derivatives.

5.3 Fuel injection parameter optimisation

5.3.1 Problem definition

The optimisation problem can be stated as follows:

Select fuel injection parameters to minimise fuel consumption and emissions whilst delivering the torque requested by the driver. The cylinder pressure should not exceed a given ceiling value (to preserve the engine integrity) and the maximum rate of cylinder pressure rise should also be limited (to reduce radiated noise from the engine which is a function of the rate of change of cylinder pressure).

Formulating as an optimal control problem, the cost function to be minimised is defined as:

$$\min_{\mathbf{q}} q_1 \tag{5.12}$$

where $\mathbf{q}$ is a vector of admissible parameters with $\mathbf{q} \in \mathcal{Q}$. The first parameter, q_1 is also the cost function to be minimised, subject to the dynamic constraint:

$$\frac{dx}{dt} - f(x, \mathbf{q}, t) = 0 \tag{5.13}$$

with path constraints:

$$g(x, \mathbf{q}, t) \leq 0 \tag{5.14}$$

terminal constraint:

$$h(x(t_f), \mathbf{q}, t_f) = 0 \tag{5.15}$$

and initial conditions:

$$x(t_0) = x_0 \tag{5.16}$$

In this study, the cylinder pressure is the only state. The cost function to be minimised is the total fuel injected. The injected fuel is a parameter defined at the start of the cycle and is a 'static' parameter. An alternative implementation could integrate the fuel injected during the cycle, in which case the cost function would include an integral term.

Path constraints include limits on the cylinder pressure and the rate of change of cylinder pressure and apply throughout the engine cycle.

A terminal constraint is introduced to constrain the estimated torque to be the same as the requested torque from the driver.

The initial condition is the cylinder pressure at the start of the simulation, calculated from the motoring curve of the engine, which is in turn based on the intake manifold pressure and composition.

5.4 Optimisation using a Wiebe model

5.4.1 Description of the system dynamics: The Wiebe model

The Wiebe combustion model, see Section 2.2.1, assumes that the burnt mass fraction is of the form:

$$x_b(t) = 1 - e^{C\left(\frac{t-t_{SOC}}{t_d}\right)^{m+1}} \tag{5.17}$$

where C and m are constants and time is normalised to the combustion duration, t_d, and initialised at the start of combustion, t_{SOC}, for $t > t_{SOC}$.

The rate of heat release can be calculated from the derivative of the burnt mass fraction (the burn rate) multiplied by the lower heating value of the fuel, Q_{LHV}, and the mass of fuel injected, q_{inj}:

$$\frac{dQ}{dt} = Q_{LHV} q_{inj} \frac{dx_b}{dt} \tag{5.18}$$

Finally, the cylinder pressure is calculated, neglecting heat losses, using Equation (2.46), repeated here for convenience:

$$\frac{dp}{dt} = \frac{\gamma - 1}{V}\frac{dQ}{dt} - \frac{\gamma p}{V}\frac{dV}{dt} \tag{5.19}$$

The cylinder volume is an algebraic function of time and therefore the state equation can be written in the form:

$$\frac{dx}{dt} - f(x, \mathbf{q}, t) = 0 \tag{5.20}$$

5.4.2 Definition of constraints

In the combustion optimisation, path constraints are applied to both the state (cylinder pressure) and its derivative. The cylinder pressure should not exceed a ceiling value, p_{max}, to preserve the engine integrity, and the rate of cylinder pressure rise should be limited, to dp_{max}, to reduce radiated noise from the engine (which is a function of the rate of change of cylinder pressure):

$$p_{mot} \leq p \leq p_{max}$$
$$\left|\frac{dp}{dt}\right| \leq dp_{max} \tag{5.21}$$

where p_{mot} is the motoring cylinder pressure with no combustion.

The admissible set of parameters, $\mathcal{Q}$, allows additional constraints to be included. For example, injection timing may be constrained to avoid misfire (a phenomenon which is not predicted by the combustion model), however most parameters are not restricted.

A terminal constraint is introduced to ensure that the estimated torque is the same as the driver request. The torque is calculated from the Mean Effective Pressure (MEP), which is calculated from the enclosed area of the pressure - Volume (p-V) diagram. See the lower right plot in Figure 5.2 for an example p-V diagram - note the axes on this diagram are scaled to highlight the differences between the optimisations and therefore only show part of the full p-V diagram. The relationships between the pressure, Volume, MEP and engine torque are given in Equation 5.22, [44].

$$\begin{aligned} MEP &= \frac{1}{V_d}\oint pdV \\ Tq &= \frac{V_d}{4\pi}MEP \end{aligned} \tag{5.22}$$

where p is the cylinder pressure, V is the cylinder volume, Tq is the engine torque [Nm] and V_d is the engine displacement [m^3] (note this expression applies to 4 stroke engines).

5.4.3 Configuration of the optimisation calculation

Definition of discretisation mesh

The continuous problem defined in Section 5.3.1 is discretised over the engine cycle resulting in a large, sparse NLP problem. The *GPOPS-II* toolbox includes an adaptive meshing algorithm which updates the mesh during the optimisation, see Figure 5.1. This is important when considering the combustion optimisation since there is a small region of rapid pressure rise which requires a fine mesh, outside this region the mesh can be much larger. If the whole problem is discretised according to the minimum mesh size, the problem would be significantly larger.

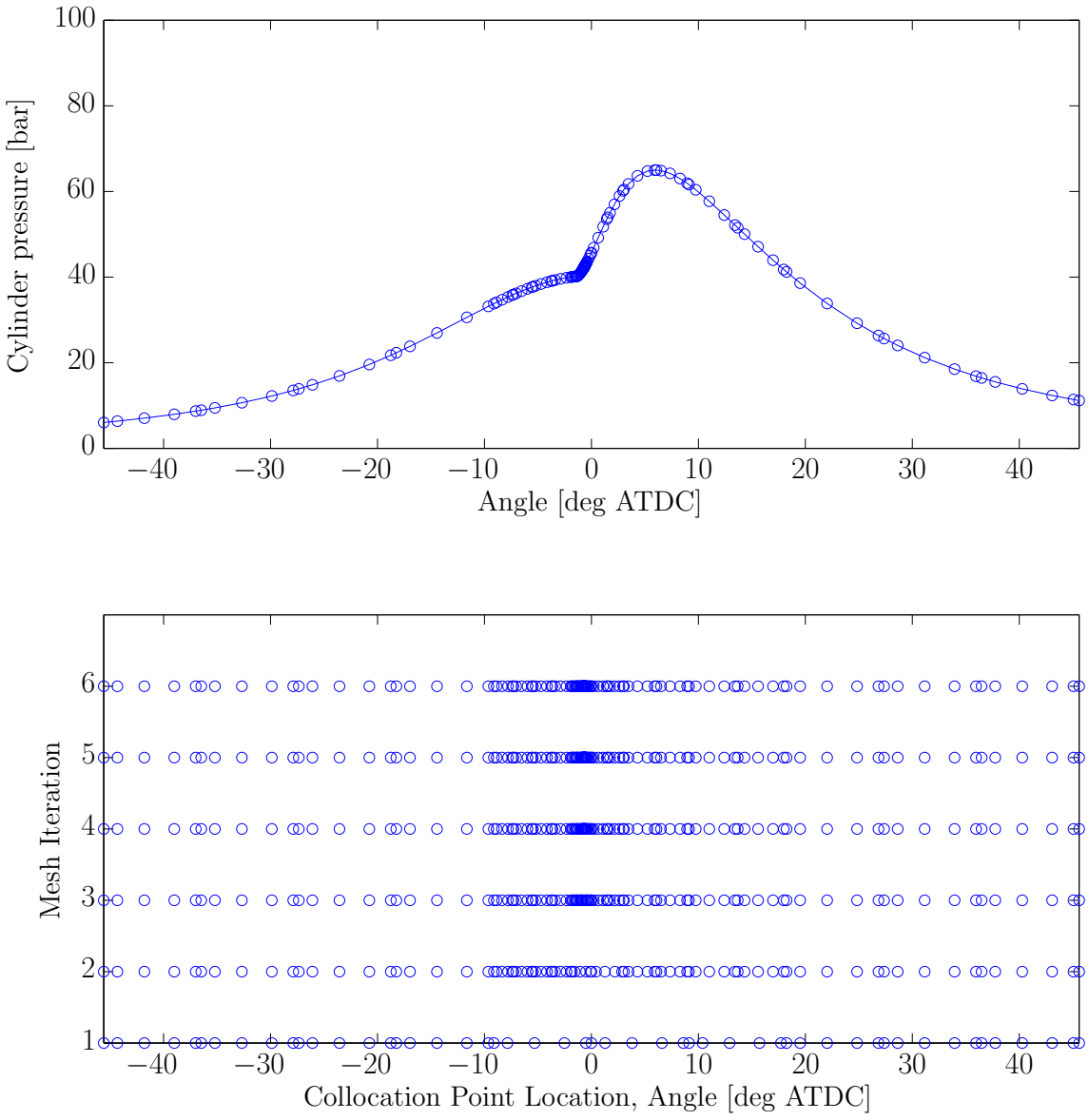

Figure 5.1: Illustration of the adaptive mesh derived during the optimisation.

Smoothing functions

The *GPOPS-II* toolbox is used with a standard NLP solver, in this case IPOPT. Interior point solvers require first and second order derivative information for the functions defining the cost and constraints. In the combustion problem, the Wiebe function is discontinuous, with the burnt mass profile applied during combustion and zero elsewhere. This can be implemented by dividing the problem into multiple phases (before, during and after combustion) or as one phase with continuous approximations for the discontinuities. In this case, the discontinuous function $max(x, y)$ was implemented with the following approximation to allow a single phase to be used:

$$max(x, y) \approx \frac{(x + y) + \sqrt{(x - y)^2 + \epsilon}}{2} \tag{5.23}$$

where ϵ is a 'small' constant.

The smoothness of the function can be assessed by considering the derivative with respect to x:

$$\frac{\partial}{\partial x}\left(\frac{(x + y) + \sqrt{(x - y)^2 + \epsilon}}{2}\right) = \frac{1}{2}\left(1 + \frac{(x - y)}{\sqrt{(x - y)^2 + \epsilon}}\right) \tag{5.24}$$

which is finite even when $x = y$.

Scaling

Scaling can have a significant impact on the performance of optimisation algorithms, affecting convergence rate, termination criteria and numerical conditioning. For instance, the tolerances for convergence may be based on a notion of 'small' quantities but this may not be 'small' for all variables if they are left in their original units. In the combustion optimisation problem, the numerical values of the different quantities in the simulation differ by several orders of magnitudes; pressure is of the order of 50×10^5 Pa whilst the injected fuel is 10×10^{-6} kg.

In general scaling can consist of a multiplication factor and an offset. In this study, a multiplication scaling was sufficient to normalise the variables and was introduced for the mass, length and time dimensions, with other quantities being derived from these scalings according to dimensional analysis. The mass was scaled on a typical injection quantity, the length was scaled on the cylinder dimensions and time was scaled to a timescale representative of the combustion process.

Initial guess

The optimisation is started with an initial 'guess' which is shown in blue on Figure 5.2. The initial guess starts with a fuelling set to 8 mg/inj, the start of combustion at top dead centre (TDC) and a combustion duration of 15 degrees.

5.4.4 Discussion of results

The *GPOPS-II* optimisation is run without constraints on cylinder pressure or rate of increase of cylinder pressure. This results in the 'unconstrained' optimisation of fuel consumption to deliver the requested MEP, see Figure 5.2. The optimum fuel consumption is 7.86 mg/str, a reduction of 2 % compared to the initial guess. The optimisation can be seen to 'push' the pressure - volume trace into the corners to improve the efficiency of the combustion.

The constraints can now be applied to the cylinder pressure, see Figure 5.2, or the rate of change of cylinder pressure, see Figure 5.3. In running the optimisations it was observed that the optimisation can output different combustion timings depending on the tolerance of the optimisation. This is because the MEP is less sensitive around TDC since the rate of change of volume at this point is small so the start of combustion timing has a small impact on the optimum value. The constraints have been set to artificially low levels to illustrate the constrained optimisation at this low load operating point.

Description	Start of Injection [degrees ATDC]	Burn Duration [degrees]	Fuelling [mg/str]	Constraint value
Initial guess	0.00	20.0	8.00	N/A
Unconstrained	0.25	1.0	7.86	N/A
Pressure constraint	-2.28	19.8	7.98	65×10^5 Pa
Pressure rate constraint	-0.83	1.8	7.87	30×10^9 Pa/s

Table 5.1: Results from Wiebe model, all with the same MEP.

The results are summarised in Table 5.1, which shows that optimisation tends towards a short combustion close to TDC. The maximum cylinder pressure constraint tends to delay the combustion, by increasing the burn duration so that the pressure rise is not added to the peak motoring cylinder pressure at TDC. The constraint on the rate of cylinder pressure rise tends to increase the burn duration slightly, to reduce the rate of heat release and hence the rate of cylinder pressure rise. The combustion timing is advanced slightly to achieve the peak cylinder pressure near TDC.

These results can now be applied to the calibration of fuel injection parameters. The combustion timing is closely related to injection timing, though the ignition delay may also vary with injection timing, adding a disturbance to the direct correlation. The combustion duration is harder to relate to the fuel injection parameters of injection timing and duration since there is a non-linear relationship to both which is difficult to invert and secondly, depends on other parameters such as EGR ratio. The results for the Wiebe optimisation provide an optimum burnt mass profile which can be used as a target for the fuel injection calibration exercise.

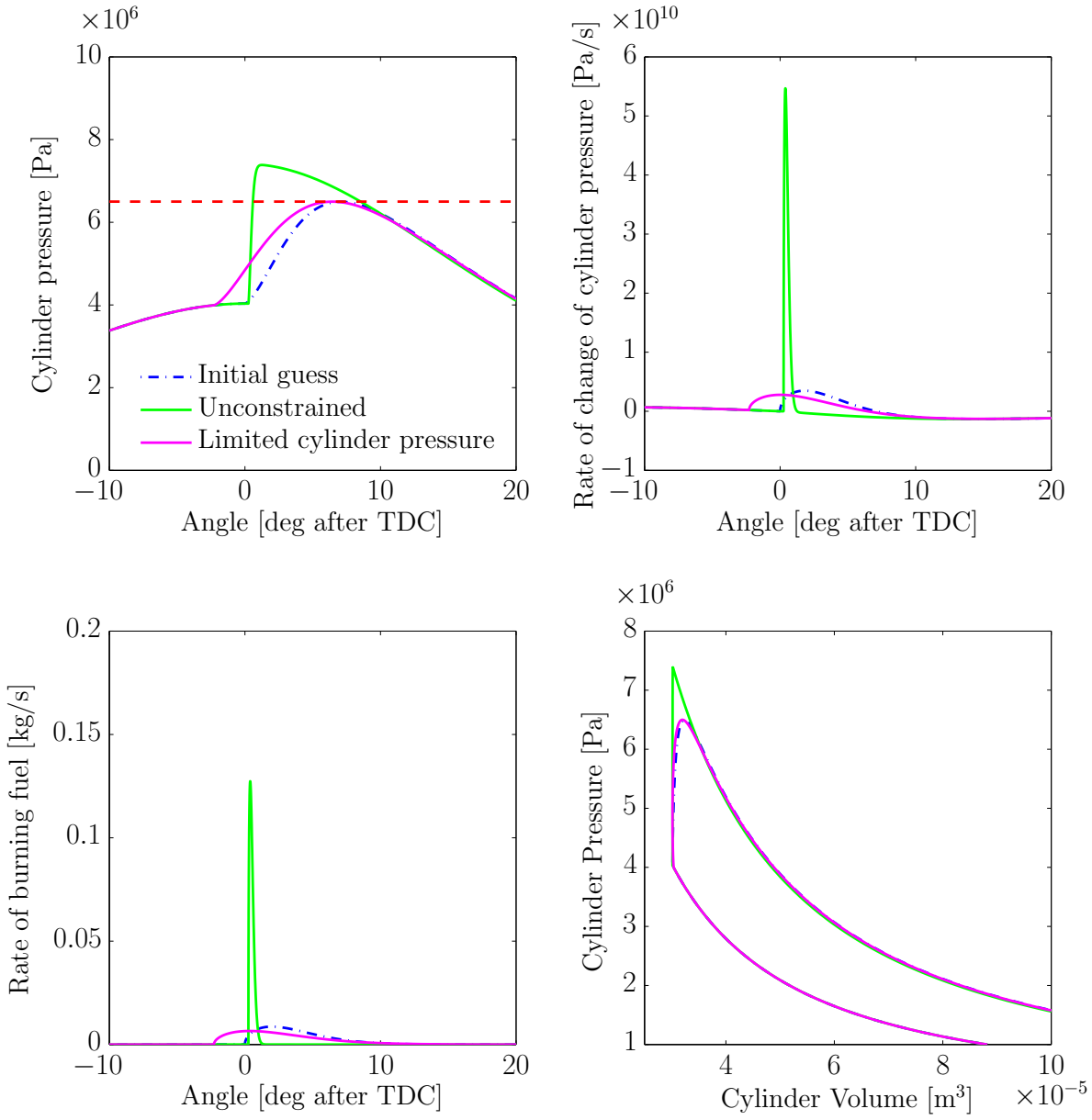

Figure 5.2: Predicted responses for the initial guess, the unconstrained optimisation and the optimisation with a cylinder pressure constraint.

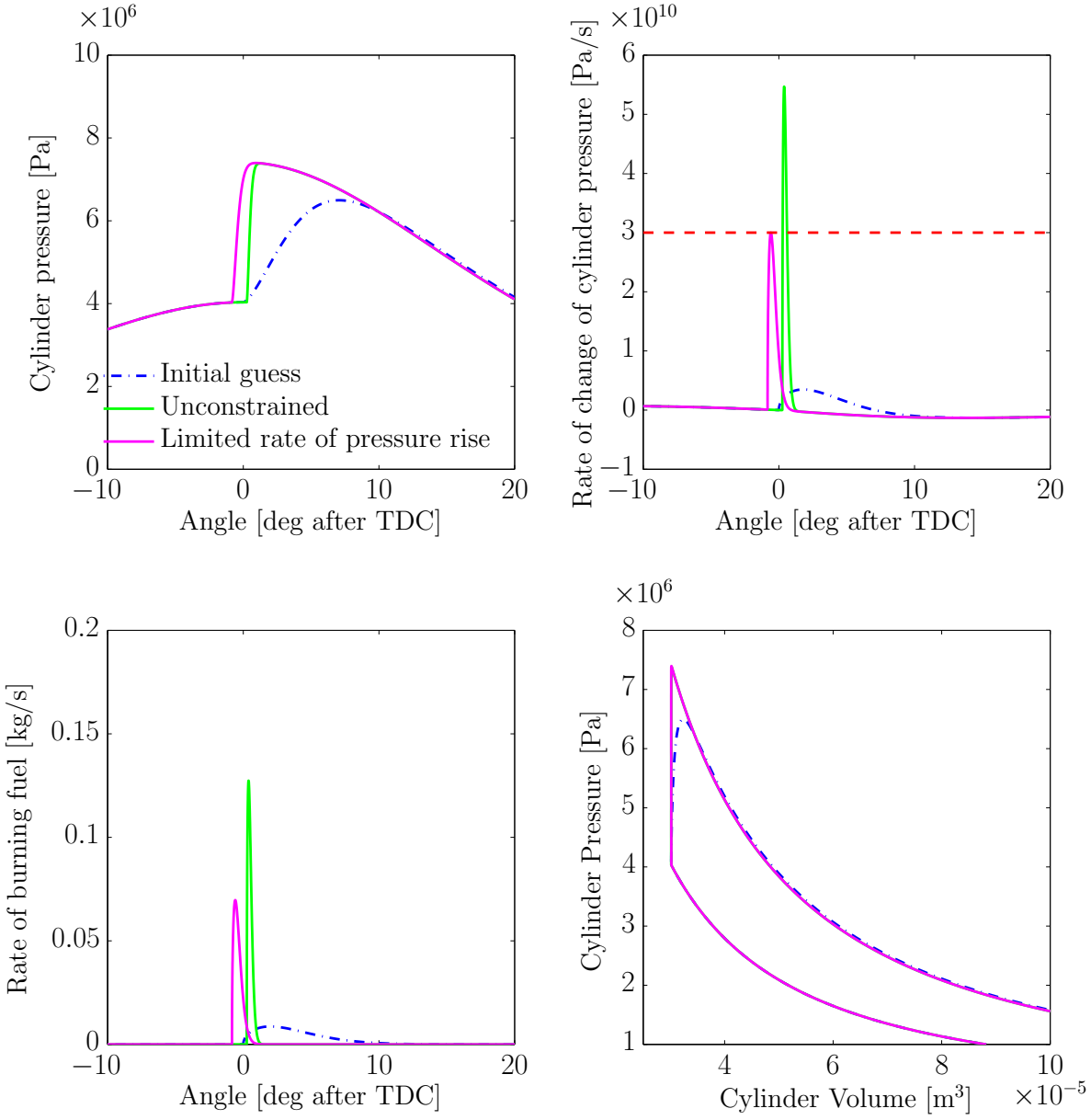

Figure 5.3: Predicted responses for the initial guess, the unconstrained optimisation and the optimisation with a constraint on the rate of cylinder pressure rise.

5.5 Optimisation using a more complex combustion model

5.5.1 The Watson and Pilley model

The combustion optimisation is repeated in this section with a more representative combustion model, the Watson and Pilley model, see [92]. This model was correlated to many combustion profiles and includes both pre-mixed and diffusion combustion phases. Significantly, it also includes an approximation for the ignition delay, so the optimisation parameter in this case is the start of injection (rather than the start of combustion in Section 5.4). An example of the Watson and Pilley model is compared to an example Wiebe model in Figure 5.4.

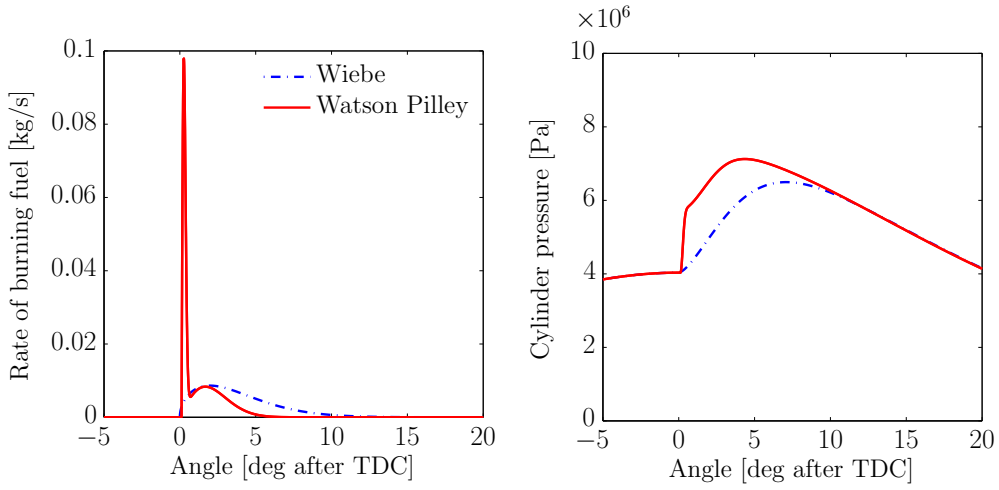

Figure 5.4: A comparison of the Wiebe and Watson Pilley combustion correlations.

The ignition delay is given by:

$$t_{igndelay} = 3.45 e^{\frac{2100}{T_{SOI}}} p_{SOI}^{-1.02} \tag{5.25}$$

where T_{SOI} is the bulk cylinder temperature at the start of injection and p_{SOI} is the cylinder pressure at the start of injection.

The combustion model is based on the ratio between pre-mixed and total mass burnt, β. The authors establish a correlation for β to the ignition delay δ and the equivalence ratio of the trapped gas in the cylinder, ϕ:

$$\begin{aligned} \beta &= \frac{m_p}{m_t} \\ &= 1 - 0.926 \phi^{0.37} \delta^{-0.26} \end{aligned} \tag{5.26}$$

The mass burnt is the weighted sum of the pre-mixed and diffusion burn rates:

$$\dot{m}_t = \frac{m_t}{t_d}\left(\beta \dot{M}_p(\tau) + (1-\beta)\dot{M}_d(\tau)\right) \tag{5.27}$$

where m_t is the total burnt mass, M_p is the pre-mixed burnt mass and M_d is the diffusion burnt mass and τ is non-dimensional time, $\tau = \frac{t - t_{SOC}}{t_d}$, normalised to the combustion duration, t_d, and initialised at the start of combustion, t_{SOC}.

The authors calculated the burnt mass profiles for different engines and different operating points. They correlated measured burn profiles to different functions. The resulting functions are given in Equations 5.28. It can be seen that the diffusion combustion is a Wiebe function.

$$\begin{aligned} M_p(\tau) &= 1 - (1 - \tau^{c_{p1}})^{c_{p2}} \\ M_d(\tau) &= 1 - e^{-c_{d1}\tau^{c_{d2}}} \end{aligned} \tag{5.28}$$

where c_{px} are coefficients for the pre-mixed combustion and c_{dx} are for the diffusion combustion.

The coefficients of the combustion functions were found to be functions of engine speed, N_{eng}, and the equivalence ratio, ϕ, of the trapped gases, see Equations 5.29.

$$\begin{aligned} c_{p1} &= 2 + 1.25 \times 10^{-8}(\delta \cdot N_{eng})^{2.4} \\ c_{p2} &= 5000 \\ c_{d1} &= 14.2\phi^{-0.644} \\ c_{d2} &= 0.79 c_{d1}^{0.25} \end{aligned} \tag{5.29}$$

The burn profile can therefore be expressed as a function of input variables and the optimisation parameters, as required for solving the optimsation problem.

5.5.2 Additional considerations for the Watson and Pilley model

The Watson and Pilley model was implemented with the same approach as described in Section 5.4.3. In addition, the inclusion of the ignition delay presents a challenge for the optimisation since the timing of the combustion profile depends on the state at the start of injection which is a function of the cylinder pressure. Multiple phase

approaches were investigated, however it was found that the problem could be configured in a similar manner to the Wiebe model, Section 5.4, by observing that the ignition delay is a function of the start of injection timing. The ignition delay was calculated for the full range of injection timings and a fourth order polynomial was fitted to the resulting curve, see Figure 5.5, allowing the ignition delay to be calculated directly from the start of injection timing.

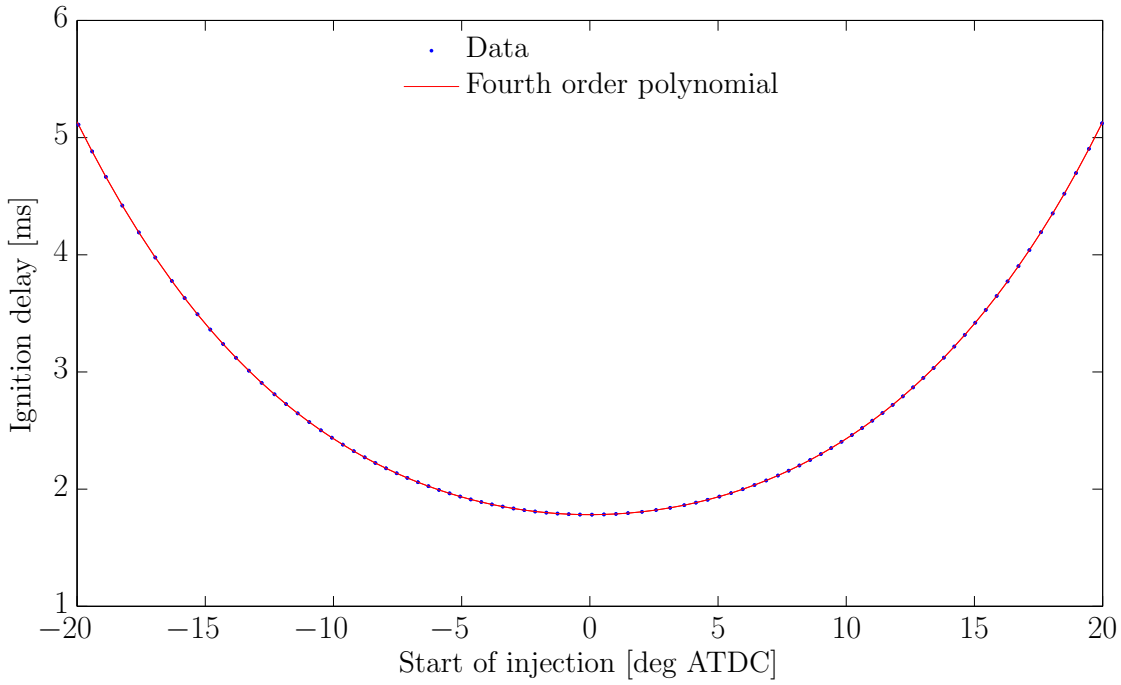

Figure 5.5: Ignition delay as a function of injection timing. Points are calculated according to equation 5.25. Curve is a fourth order polynomial fit.

The burnt mass curve can now be evaluated directly from the calibration parameters: injection timing, burn duration and mass of fuel injected.

5.5.3 Discussion of results

The *GPOPS-II* toolbox is applied to the same optimisation problem as Section 5.4.3, evaluating the initial guess followed by the unconstrained optimum, the constrained cylinder pressure and constrained rate of pressure increase.

The results are summarised in Table 5.2. The unconstrained optimisation tends towards a short combustion close to TDC. The maximum cylinder pressure constraint tends to delay the combustion so that the pressure rise is not added to the peak motoring cylinder pressure at TDC. The constraint on the rate of cylinder pressure rise tends to

increase the burn duration, to reduce the rate of heat release and hence the rate of cylinder pressure rise.

Description	Start of Injection [degrees ATDC]	Burn Duration [degrees]	Fuelling [mg/str]	Constraint value
Initial guess	0.00	15.0	8.00	N/A
Unconstrained	0.29	6.3	7.98	N/A
Pressure constraint	1.12	30.0	8.10	65×10^5 Pa
Pressure rate constraint	-1.06	21.1	8.00	30×10^9 Pa/s

Table 5.2: Results from Watson and Pilley model

5.6 Conclusions

In this chapter, a non-linear optimisation was applied to the calibration of fuel injection parameters. A Wiebe heat release model was used initially to configure the optimisation before moving onto the Watson and Pilley model which has a more realistic heat release profile and parameters with more physical meaning.

The optimal control of fuel injection parameters is a new application of *GPOPS-II* and required some configuration to pose in a suitable form. For example smoothing the heat release characteristic and configuring the combustion model to run in one phase. The cost function used for the optimisation is unusual in that it is not a function of the states or the control actions, but rather one of the initial parameters. Whilst the cost function is relatively simple in this study, the optimal control framework established here will allow future, more complex optimisations to include states such as engine emissions.

As for any new application of a general purpose solver, it is important to validate the results from the optimisation. Firstly, the unconstrained optimum tends to give a short heat release around top dead centre (TDC) as expected, with the optimised Wiebe combustion tending to a pre-mixed combustion profile as found in many Diesel engines.

Considering now the constrained results, there is an interesting difference between the response of the two models to the cylinder pressure constraint. The Wiebe model advances the combustion whilst the Watson and Pilley model retards the combustion. In general, the constraint on cylinder pressure would be expected to retard the combustion so that more heat is released as the cylinder volume is increasing (thus having less impact on cylinder pressure), however this has a detrimental effect on the MEP. This characteristic is seen with the Watson and Pilley model which has a portion of heat released during pre-mixed combustion followed by the further heat release during

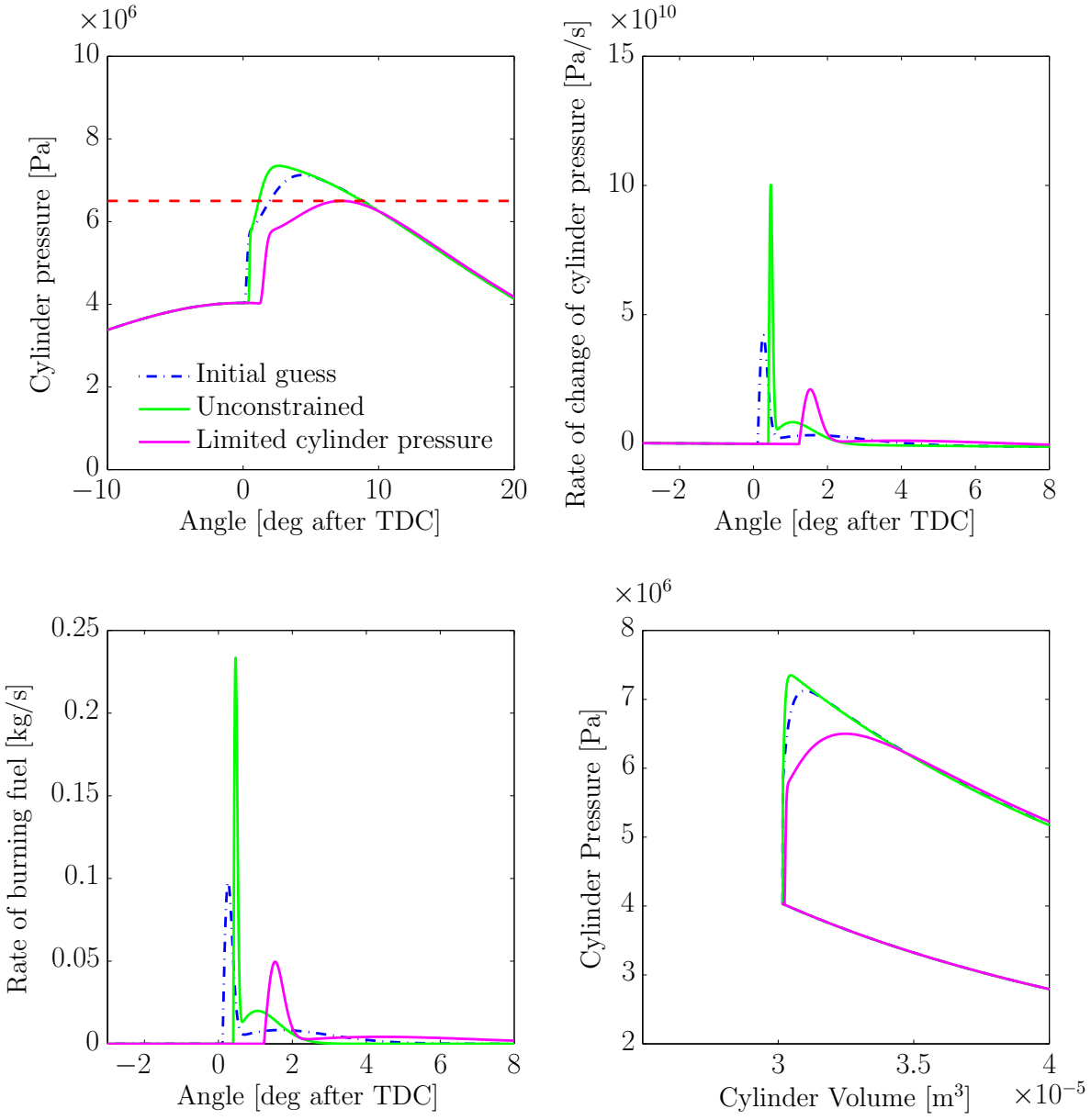

Figure 5.6: Cylinder pressure constraint.

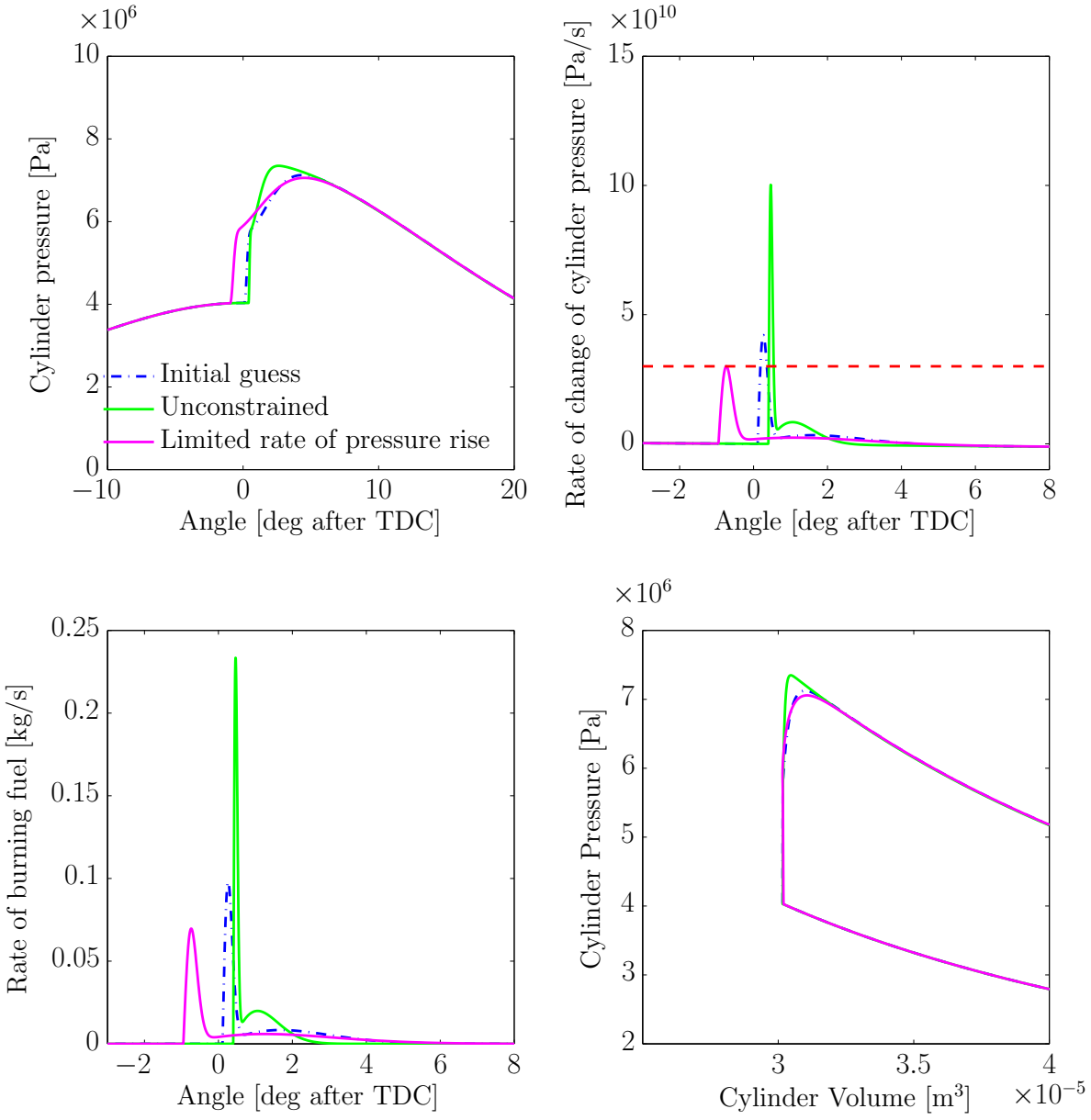

Figure 5.7: Constraint on rate of cylinder pressure rise.

diffusion combustion. The pre-mixed combustion ratio, β is symmetrical around TDC (since it is a function of the ignition delay which is also symmetrical around TDC) so for a given β, the optimisation can choose to advance or retard the combustion relative to TDC. Advancing the combustion will result in a higher cylinder pressure since the diffusion combustion heat release will overlap TDC, whereas retarding the combustion will result in a lower cylinder pressure as the heat is released as the cylinder is expanding. The Wiebe model, without the pre-mixed combustion, can limit the cylinder pressure by advancing combustion and having a slower heat release around TDC which gives a more gradual pressure rise. This gives rise to an earlier peak cylinder pressure, resulting in a higher MEP.

For the rate of pressure rise constraint, both models advance the combustion. For a typical combustion profile with pre-mixed and diffusion combustion the optimisation adjusts the balance between the pre-mixed and diffusion combustion to keep the maximum rate of cylinder pressure rise below the limit. For a given pre-mixed combustion, the optimisation can either advance or retard relative to TDC. Without a constraint on cylinder pressure, the optimisation looks to maximise the MEP by increasing the cylinder pressure and hence advances the combustion, as shown in the Watson and Pilley result.

Future developments could add mulitple injections, by superimposing heat release profiles, to include the impact of pilot injections for example. Moving further ahead, the Watson and Pilley model could be replaced with the full combustion model developed in Chapter 2. This would allow the optimisation to include emissions predictions and directly use engine parameters such as injection duration, rail pressure and EGR ratio.

Chapter 6

Conclusions

The motivation for this work was to develop a new approach to combustion control. Combustion is at the core of an internal combustion engine, yet its control is still based largely around PID controllers that track quantities that are not direct objectives for the development engineer (for example: air flow, EGR flow, intake manifold pressure). In this work, a novel control approach based on a model of combustion, seeks to minimise emissions through on-line optimisation. The results and conclusions draw from work in this thesis and are discussed in detail at the end of each chapter. These conclusions are used to summarise the key contributions and provide suggestions for future work.

Overall, this work provides a fresh approach to engine control and shows how model predictive control may reduce calibration effort and engine emissions.

6.1 Key contributions

Combustion model

A combustion model was developed, taking elements from a variety of approaches, bringing them together in a new formulation. A specific area of development was the combination of the β-PDF mixing approach with a simplified combustion model (Section 2.3.3). This combined the efficient implementation of the spray mixing with an efficient combustion simulation. In addition to the subsequent application to a predictive combustion controller, this model also provides an interesting insight into the combustion processes as it allows access to different graphical representations of combustion.

Parallel trust region optimisation

A novel optimisation approach was developed for the specific requirements of this project - an optimisation with few iterations that must run in real-time. A direct optimisation

based on a trust region search, the parallel trust region (PTR) optimisation (Appendix B), was developed and tested as part of a model predictive controller for a bicycle model (Appendix C). The PTR optimisation was then used to tune engine control parameters to control the combustion in simulation, using a mean value engine model as a plant model.

Vehicle testing

The combustion model was combined with the PTR optimisation and implemented on a prototype control system, based around an FPGA. The control system was integrated into a vehicle control system and demonstrated that a predictive model can be made to run fast enough to be used in a control algorithm and to optimise the NO_x during transient manoeuvres. The combustion controller was set up to minimise NO_x emissions and tested over the EUDC and US06 drive cycles where it was shown to significantly reduce NO_x emissions by over 10 % during the US06 test cycle (Figure 4.15). The combustion controller demonstrated the potential to calibrate the engine out emissions rather than indirect quantities such as EGR valve position or intake manifold pressures.

During the development of the embedded vehicle controller, an integrated toolchain has been exercised with on complex system allowing a combined VHDL and C code implementation to be generated from the same Simulink model. Whilst this allows the general Simulink user to exploit the benefits of an FPGA without having to become fully fluent in VHDL, it does require an understanding of the FPGA characteristics to ensure the generated VHDL is appropriate.

Fuel injection calibration

The development of the control orientated combustion model sparked a new direction of research, using the model to calibrate the fuel injection parameters through the application of a non-linear optimisation to the combustion model input parameters (Chapter 5). The injection timing and duration were adjusted to minimise fuel consumption whilst delivering the required torque and simultaneously respecting constraints on the maximum cylinder pressure and rate of cylinder pressure rise.

6.2 Future work

Additional validation activities

The combustion model has been calibrated to one engine and used to control another engine. Ideally the calibration of the engine would be carried out on the same engine used for the control development. In practice, this will be rarely possible, so it is recommended to build an understanding of the sensitivities of the different model parameters used in the calibration by parameterising to different data sets.

Start of injection optimisation demonstration in a vehicle

The optimisation in vehicle was applied to the EGR system, a logical extension is to apply to both the EGR and start of injection. This would require an engine controller with access to the start of injection and further refinements in the FPGA implementation to achieve an even faster model execution time frame.

NO_x and soot trade-off

The combustion controller in the test vehicle minimised NO_x but particulates were not considered, the next stage would be to include the soot model and include NO_x and soot in the cost function. This would allow the trade-off between NO_x and soot to be balanced through the relative weights in the cost function.

Fuel injection calibration

The approach developed for the calibration of the fuel injection processes could be explored at higher engine loads to allow the optimisation to find the optimum balances between pre-mixed combustion and diffusion combustion at real world engine limits. The framework could also be extended to include the full combustion model, allowing the optimisation to include emissions as well as fuel consumption in the cost function.

Appendix A

Combustion Modelling

A.1 Spray penetration

A detailed derivation for the spray penetration is given in [64] and is summarised here. The spray geometry is given in Figure A.1.

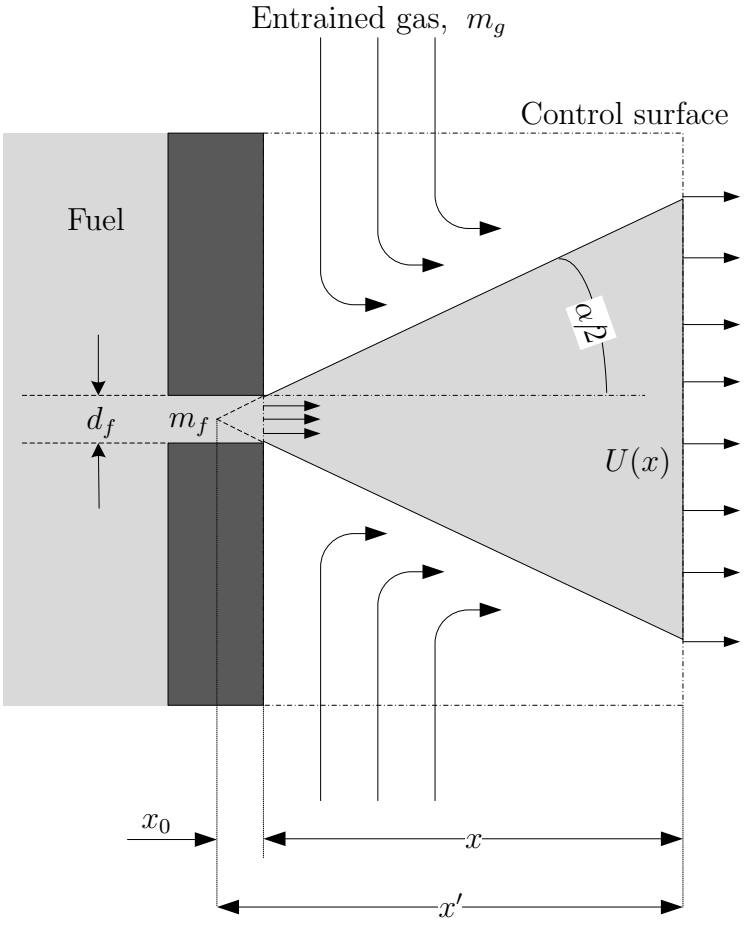

Figure A.1: Spray schematic

The spray length at a given point in time is derived by calculating the spray tip velocity and then integrating this over time to get the length or penetration of the spray. The spray tip velocity is calculated by conserving fuel mass and momentum in the axial

direction across the control volume:

$$\rho_f \cdot A_f(0) \cdot U_f = \rho_f \cdot A_f(x) \cdot U(x) \tag{A.1}$$

$$\rho_f \cdot A_f(0) \cdot U_f^2 = \rho_f \cdot A_f(x) \cdot U(x)^2 + \rho_a \cdot A_a(x) \cdot U(x)^2 \tag{A.2}$$

where ρ_f and ρ_a are the densities of the injected fuel and entrained ambient air respectively. U_f and $A_f(0)$ are the axial velocity and cross-sectional area of the fuel at the exit of the orifice. $U(x)$, $A_f(x)$ and $A_a(x)$ are the spray velocity and cross-sectional areas of the fuel and air at any x location in the spray.

This modelling approach calls on a number of assumptions, as detailed in [64]. In particular there is an assumption that there is a constant injection during the analysis. For the model being developed here, this assumption will be challenged since the model needs to accept any injection profile with several different injections. A number of authors have looked into modelling multiple injections, for example [25] and [68], and tend to consider each injection independently of the other injections. In this work we are looking to produce a simplified model so will assume that injections feed into one spray.

A third equation is derived by considering the cross sectional area of the spray, $A(x)$, which is the sum of the area occupied by the fuel and the area occupied by the air:

$$A(x) = A_a(x) + m \cdot A_f(x) \tag{A.3}$$

where m is a parameter that is either 0 or 1 and is included to allow the impact of neglecting the area occupied by the fuel. Setting m to zero would neglect the cross-sectional area occupied by the fuel.

Eliminating $A_f(x)$ and $A_a(x)$ from equations (A.1) to (A.3) inclusive gives $U(x)$ in terms of the areas, U_f and the densities:

$$U(x) = \frac{U_f \cdot (\tilde{\rho} - m)}{2 \cdot \tilde{A}} \left(\sqrt{1 + \frac{4 \cdot \tilde{A} \cdot \tilde{\rho}}{(\tilde{\rho} - m)^2}} - 1 \right) \tag{A.4}$$

where $\tilde{\rho}$ and $\tilde{A}$ are non-dimensional density and areas respectively:

$$\tilde{\rho} = \frac{\rho_f}{\rho_a} \tag{A.5}$$

$$\tilde{A} = \frac{A(x)}{A_f(0)} \tag{A.6}$$

The authors of [64] have analysed many sprays and have found that they can be conveniently characterised by considering the non-dimensional trajectory for the spray. The non-dimensional length, $\tilde{x}$, and time, $\tilde{t}$, have been calculated using a representative length, x^+, and time, t^+.

$$x^+ = d_f \cdot \sqrt{\tilde{\rho}} \cdot \left(\frac{\tilde{\rho} - m}{\tilde{\rho}}\right) \cdot \frac{1}{tan(\alpha/2)} \tag{A.7}$$

$$t^+ = \frac{x^+}{U_f}\left(\frac{\tilde{\rho} - m}{\tilde{\rho}}\right) \tag{A.8}$$

where α is the spray angle in the 'model' which is related to the commonly measured spray angle θ in Figure 2.9 by $tan\alpha = atan\theta$, [64]. The authors found a value of $a = 0.66$ was found to give a close match between the model and measured data.

The non-dimensional length and time given by:

$$\tilde{t} = t'/t^+ \tag{A.9}$$

$$\tilde{x} = x'/x^+ \tag{A.10}$$

Equation (A.4) simplifies to:

$$\frac{d\tilde{x}}{d\tilde{t}} = \frac{2}{\sqrt{1 + 16 \cdot \tilde{x}^2} + 1} \tag{A.11}$$

Now, assume that $U(x)$ is the velocity of the spray tip, integration of (A.11) from $\tilde{x} = 0$ to $\tilde{x} = \tilde{S}$ where $\tilde{S}$ is the non-dimensional spray penetration $\tilde{S} = S'/x^+$ gives:

$$\tilde{t} = \frac{\tilde{S}}{2} + \frac{\tilde{S}}{4} \cdot \sqrt{1 + 16 \cdot \tilde{S}^2} + \frac{1}{16} \cdot ln\left(4 \cdot \tilde{S} + \sqrt{1 + 16 \cdot \tilde{S}^2}\right) \tag{A.12}$$

The correlation can be split into two phases. Early in the spray, for $\tilde{t}$ less than 1, the jet is dominated by the injected fluid. Whilst later, for $\tilde{t}$ greater than one, the spray is dominated by the entrained air. These two phases are shown as the straight line sections of the spray characteristic from (A.12) in Figure A.2.

Equation (A.12) cannot be rearranged to give $\tilde{S}$ in terms of $\tilde{t}$, however a simple combination of the two phases of the spray gives the following expression for $\tilde{S}$ in terms of $\tilde{t}$.

$$\begin{aligned} \tilde{S} &= \left[\left(\frac{1}{\tilde{t}}\right)^n + \left(\frac{1}{\sqrt{\tilde{t}}}\right)^n\right]^{-\frac{1}{n}} \\ &= \frac{\tilde{t}}{\left(1+\tilde{t}^{\frac{n}{2}}\right)^{\frac{1}{n}}} \end{aligned} \tag{A.13}$$

A value of $n = 2.2$ results in a fit to within 5 percent in the transition region.

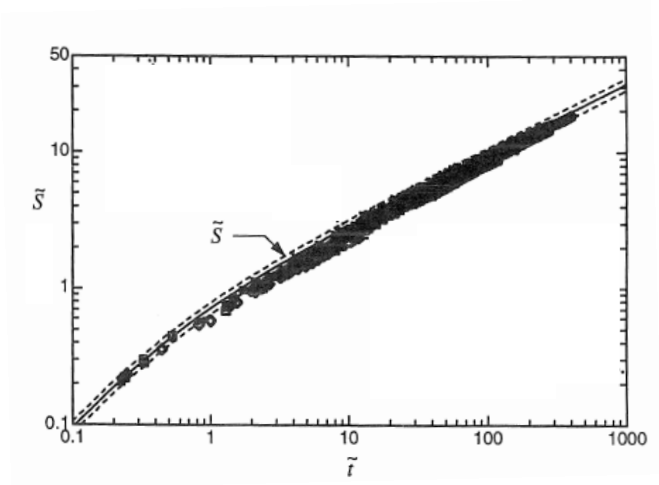

Figure A.2: Vapourising Spray Correlation [64]. Reprinted with permission from SAE Paper No. 960034 ©1996 SAE International. Further use or distribution of this material is not permitted without prior permission from SAE.

The authors of [64] also carried out an investigation into the differences between non-vapourising and vapourising sprays. They noted a reduction in the penetration of the vapourising spray after a $\tilde{t}$ of 0.5. However as can be seen from Figure A.2, the non-vapourising model provides a reasonable estimate of the vapourising spray.

A.2 Evaporation

The evaporation process can be modeled by assuming that the fuel atomises into droplets which all have the same diameter, the Sauter Mean Diameter (SMD) . An empirical formula for the SMD is taken from [44]:

$$d_o[\mu m] = A\Delta p^{-0.135} \cdot \rho_a^{0.121} \cdot V_f^{0.131} \tag{A.14}$$

where A is a constant given by the authors as approximately 24, Δp is the pressure across the injector [MPa], V_f fuel per cylinder per cycle and ρ_a is the density of air [kg/m^3]. The droplets then reduce in size over time according to the d^2 law:

$$d(t)^2 = d_0^2 - \beta \cdot t \tag{A.15}$$

where $d(t)$ is the current droplet diameter, d_0 is the initial droplet diameter, β is a constant of order $10^{-6}\,\mathrm{m}^2/\mathrm{s}$ and t is the time since injection. It is assumed that β does not vary with temperature.

The volume of a droplet is given by, $V = \pi d^3/6$, and therefore the liquid mass of fuel, m_{fl}, at time t is given by:

$$m_{fl}(t) = \frac{n\rho\pi}{6} d(t)^3 \tag{A.16}$$

where n is the number of droplets and ρ is the density of fuel.

The fuel injection is discretised into parcels of fuel injected each time-step. The initial mass of the liquid fuel in the fuel parcel is set to q_{par} and can be expressed as a function of the initial droplet diameter, d_0:

$$q_{par} = \frac{n\rho\pi}{6} d_0^3 \tag{A.17}$$

where the subscript *par* refers to the fuel parcel. Now, substitute into (A.16) to obtain:

$$m_{parl}(t) = q_{par} \left(\frac{d(t)}{d_0} \right)^3 \tag{A.18}$$

The vaporised fuel, m_{parv} at time t is the difference between the initial mass of liquid fuel and the mass of liquid fuel at time t:

$$\begin{aligned} m_{parv}(t) &= q_{par} - m_{parl}(t) \\ &= q_{par} \left(1 - \left(\frac{d(t)}{d_0} \right)^3 \right) \end{aligned} \tag{A.19}$$

Discretisation with time-step δt gives the following evolution of the droplet diameter for the first three time-steps:

$$\begin{aligned} d(1)^2 &= d_0^2 - \beta \cdot \delta t \\ d(2)^2 &= d_0^2 - 2\beta \cdot \delta t \\ d(3)^2 &= d_0^2 - 3\beta \cdot \delta t \cdots \end{aligned} \tag{A.20}$$

and the following evolution of the mass of vaporised fuel:

$$\begin{aligned} m_{parv}(1) &= q_{par}(1-(d(1)/d_0)^3) \\ m_{parv}(2) &= q_{par}(1-(d(2)/d_0)^3) \\ m_{parv}(3) &= q_{par}(1-(d(3)/d_0)^3)\cdots \end{aligned} \tag{A.21}$$

The vaporised fuel from each fuel parcel is summed to give the total vaporised fuel, m_{fv}, noting that a new fuel parcel is injected each time-step.

$$\begin{aligned} m_{fv}(1) &= m_{parv}(1) \\ m_{fv}(2) &= m_{parv}(2)+m_{parv}(1) \\ m_{fv}(3) &= m_{parv}(3)+m_{parv}(2)+m_{parv}(1)\cdots \end{aligned} \tag{A.22}$$

This can be presented in matrix form:

$$\begin{bmatrix} m_{fv}(1) \\ m_{fv}(2) \\ m_{fv}(3) \\ \vdots \\ m_{fv}(N) \end{bmatrix} = \begin{bmatrix} m_{parv}(1) & 0 & 0 & \cdots & 0 \\ m_{parv}(2) & m_{parv}(1) & 0 & \cdots & 0 \\ m_{parv}(3) & m_{parv}(2) & m_{parv}(1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{parv}(M) & m_{parv}(M-1) & m_{parv}(M-2) & \ddots & \vdots \\ q_{par} & m_{parv}(M) & m_{parv}(M-1) & \ddots & \vdots \\ q_{par} & q_{par} & m_{parv}(M) & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{par} & q_{par} & q_{par} & \cdots & q_{par} \end{bmatrix} . \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

where M is the number of steps required to evaporate a fuel parcel and N is the number of fuel parcels. Rewriting in a compact form:

$$\mathbf{m_{fv}} = \mathbf{T}(\mathbf{m_{parv}}, \mathbf{0}) \cdot \mathbf{1} \tag{A.23}$$

where $\mathbf{T}(\mathbf{m_{parv}}, \mathbf{0})$ is a Toeplitz matrix with $\mathbf{m_{parv}}$ as the first column and zeros as the first row.

Multiple injections can now be added by replacing the vector of ones, $\mathbf{1}$, with a vector, $\mathbf{Z_{inj}}$, which allows a fuel injection profile to be created.

$$\mathbf{m_{fv}} = \mathbf{T}(\mathbf{m_{parv}}, \mathbf{0}) \cdot \mathbf{Z_{inj}} \tag{A.24}$$

In this formulation, the evaporation is not dependent on combustion and (A.24) can be calculated up front.

A.3 Evolution of the mixture variance

During the mixing process, the variance of the mixture evolves. This section derives a new formula for this process, based on the work of [26] and updated to include the combustion processes used by the model in this project. Specifically, the model here generates burnt gases which was not the case with the model in [26], which tracks the species as combustion progresses but doesn't change the overall mixture.

In this model, the burnt gases leave the spray after combustion and can re-entrained as the spray and combustion progresses.

A.3.1 Preliminaries

The mass averaged variance of the fuel mass fraction Z is defined as:

$$\begin{aligned} \overline{Z'^2} &= \frac{1}{m_s}\int_{m_s} Z'^2 dm \\ &= \int_0^{Z_s} (Z-\overline{Z})^2 \cdot P(Z) \cdot dZ \end{aligned} \tag{A.25}$$

where $\overline{Z}$ is the mean of Z, Z' is the deviation from the mean and Z_s is the maximum value for Z. Differentiating (A.25) gives:

$$\frac{d\overline{Z'^2}}{dt} = \underbrace{\int_0^{Z_s} \frac{d(Z-\overline{Z})^2}{dt} \cdot P(Z) \cdot dZ}_{a} + \underbrace{\int_0^{Z_s} (Z-\overline{Z})^2 \cdot \frac{dP(Z)}{dt} \cdot dZ}_{b} \tag{A.26}$$

In [26] the author demonstrates that term (a) of (A.26) is equal to zero, leaving term (b) to define the evolution of the variance. Adopting the approach from [26], the flows into and out of the spray are used to calculate the impact on the variance. These are broken down into contributions from the entrained mass (at $Z = 0$), evaporated fuel (at $Z = Z_s$) and turbulence, together with a 'catch all' complimentary function. The entrained mass and evaporation contributions are repeated here for the benefit of the reader. The contribution caused by the burnt gases leaving the spray is derived and then the complimentary function is re-evaluated for the updated model.

A.3.2 Entrained mass contribution

Consider the spray entrains a small amount of gas, δm_g, during time δt. The change in probability of the mixture at $Z = 0$ can be calculated as follows:

$$P(0,t+\delta t) - P(0,t) = \frac{P(0,t)m_s + \delta m_g}{m_s + \delta m_g} - P(0,t) \tag{A.27}$$

In the limit, this gives the following expression for the rate of change of the probability function:

$$\begin{aligned} \frac{dP(0)}{dt} &= \lim_{\delta t \to 0} \frac{P(0,t+\delta t) - P(0,t)}{\delta t} \\ &= \frac{1}{m_s}\frac{dm_g}{dt} \end{aligned} \tag{A.28}$$

Substituting Equation A.28 into A.26 (b) gives the following contribution to the change in variance:

$$\begin{aligned} \int_0^{Z_s} (Z-\overline{Z})^2 \cdot \frac{dP(Z)}{dt} \cdot dZ &= \int_0^{Z_s} (Z-\overline{Z})^2 \cdot \frac{dP(Z)}{dt} \cdot \psi(Z) \cdot dZ \\ &= \overline{Z}^2 \cdot \frac{1}{m_s}\frac{dm_g}{dt} \end{aligned} \tag{A.29}$$

where ψ is the Dirac function.

A.3.3 Evaporated fuel contribution

Applying a similar approach for saturated vapour, m_{sat}, being added at Z_s as fuel evaporates into the spray, gives:

$$\frac{dP(Z_s)}{dt} = \frac{1}{m_s}\frac{dm_{sat}}{dt} \tag{A.30}$$

The corresponding contribution to the variance is:

$$\int_0^{Z_s} (Z-\overline{Z})^2 \cdot \frac{dP(Z)}{dt} \cdot \psi(Z-Z_s) \cdot dZ = (Z_s-\overline{Z})^2 \cdot \frac{1}{m_s}\frac{dm_{sat}}{dt} \tag{A.31}$$

The introduction of the saturated vapour at Z_s involves mixing fuel and air. For a given fuel evaporation rate, the fuel is mixed with air taken from the spray. This affects the mixture by removing air at $Z = 0$ and can be included in the evolution of the spray with an additional term for the mass of gas removed by evaporation, m_{evap}.

The contribution to the PDF is given by:

$$\frac{dP(0)}{dt} = \frac{1}{m_s}\frac{dm_{evap}}{dt} \tag{A.32}$$

The corresponding contribution to the variance is:

$$\int_0^{Z_s} (Z-\overline{Z})^2 \cdot \frac{dP(Z)}{dt} \cdot \psi(Z) \cdot dZ = \overline{Z}^2 \cdot \frac{1}{m_s}\frac{dm_{evap}}{dt} \tag{A.33}$$

A.3.4 Burnt mass contribution

During combustion, a mass of fuel $\delta m_{bf}(Z)$ is burnt during time δt. In the mixing model $\delta m_{bf}(Z)$ is a function of Z. The fuel and the associated gas which is burnt with it is removed from the spray. Assuming stoichiometric combustion and an EGR fraction of X_{spray}, this gives the total mass leaving the spray,

$$\delta m_{bg}(Z) = \delta m_{bf}(Z)(1 + \frac{AFR_{st}}{(1 - X_{spray})}), \tag{A.34}$$

where AFR_{st} is the stoichiometric air to fuel ratio.

Consider the change in probability of the mixture at Z, see Figure A.3

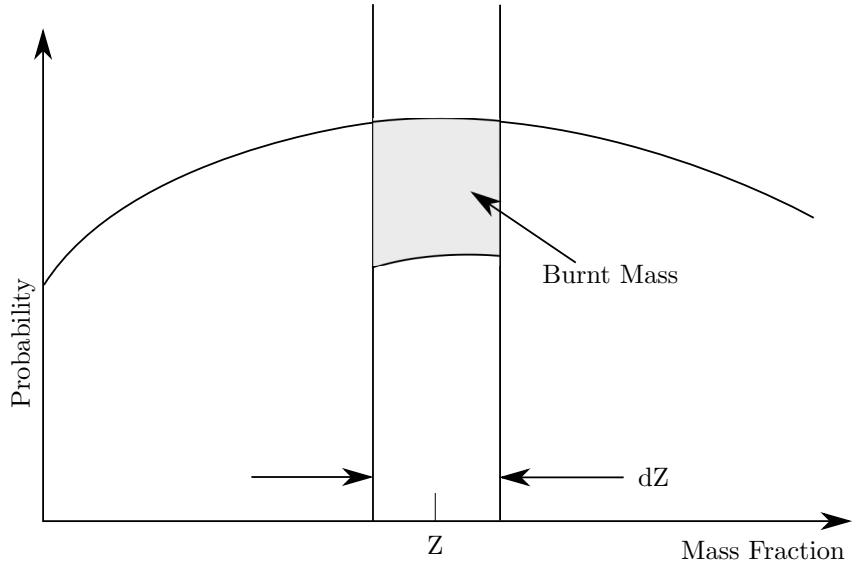

Figure A.3: Illustration of burnt mass leaving probability distribution

$$P(Z, t + \delta t) - P(Z, t) = \frac{P(Z, t)m_s - \delta m_{bg}(Z)}{m_s - \delta m_{bg}(Z)} - P(Z, t) \tag{A.35}$$

The rate of change of the probability function in the limit, as $\delta t \to 0$, is given by:

$$\begin{aligned} \frac{dP(Z)}{dt} &= \lim_{\delta t \to 0} \frac{P(Z, t + \delta t) - P(Z, t)}{\delta t} \\ &= -\frac{1}{m_s} \cdot \frac{dm_{bg}(Z)}{dt} \end{aligned} \tag{A.36}$$

The contribution to the change in variance is therefore:

$$\int_0^{Z_s} (Z - \overline{Z})^2 \cdot \frac{dP(Z)}{dt} dZ = -\frac{1}{m_s} \int_0^{Z_s} (Z - \overline{Z})^2 \cdot \frac{dm_{bg}(Z)}{dt} dZ \tag{A.37}$$

A.3.5 Complimentary contribution

By definition:

$$\int_0^{Z_s} P(Z)dZ = 1 \tag{A.38}$$

With the masses entering and leaving the spray, a complimentary function is introduced to ensure the probability continues to sum to one. The complimentary function is defined as:

$$CF = \frac{1}{1 - \frac{\sum \delta m}{m_s}} \tag{A.39}$$

where $\sum \delta m$ is the sum of the mass out of the spray and $\frac{\sum \delta m}{m_s}$ is the change in probability without correction.

The rate of change of probability due to the complimentary function is given by:

$$\begin{aligned}
\frac{dP_c(Z)}{dt} &= \lim_{\delta t \to 0} \frac{P(Z, t+\delta t) - P(Z,t)}{\delta t} \\
&= \lim_{\delta t \to 0} \frac{CF \cdot P(Z) - P(Z)}{\delta t} \\
&= \lim_{\delta t \to 0} \frac{\sum \delta m}{m_s} \cdot \frac{P(Z)}{\delta t} \\
&= \frac{\sum dm}{dt} \frac{P(Z)}{m_s}
\end{aligned} \tag{A.40}$$

The contribution to the change in variance is therefore:

$$\begin{aligned}
\int_0^{Z_s} (Z - \overline{Z})^2 \cdot \frac{dP(Z)}{dt} dZ &= \frac{1}{m_s} \frac{\sum dm}{dt} \int_0^{Z_s} (Z - \overline{Z})^2 P(Z) dZ \\
&= \frac{1}{m_s} \frac{\sum dm}{dt} \overline{Z'^2} \\
&= \frac{1}{m_s} (\frac{\sum dm_{bg}}{dt} - \frac{dm_g}{dt} - \frac{dm_f}{dt}) \overline{Z'^2}
\end{aligned} \tag{A.41}$$

A.3.6 Turbulence contribution

Turbulence in the combustion chamber reduces the variance in mixture distribution. This is treated with an empirical term in [26], given by:

$$\left. \frac{d\overline{Z'^2}}{dt} \right|_{diss} = -2 \cdot C_{diss} \cdot \frac{\overline{Z'^2}}{\frac{\tilde{K}}{\tilde{\epsilon}}} \tag{A.42}$$

where C_{diss} is an empirical constant, $\tilde{K}$ the mean specific turbulent kinetic energy and $\tilde{\epsilon}$ the dissipation rate of the mean specific kinetic energy.

A.3.7 Variance equation

Combining the contributions to the change in variance; A.42, A.29, A.31, A.33, A.37 and A.41, gives the following equation for the rate of change of variance.

$$\begin{aligned}\frac{d\overline{Z'^2}}{dt} &= -2\cdot C_{diss}\cdot\frac{\overline{Z'^2}}{\frac{\tilde{K}}{\tilde{\epsilon}}}+\frac{1}{m_s}\overline{Z}^2\frac{dm_g}{dt}+\frac{1}{m_s}(Z_s-\overline{Z})^2\frac{dm_{sat}}{dt}+\frac{1}{m_s}\overline{Z}^2\frac{dm_{evap}}{dt}\\ &\quad -\frac{1}{m_s}\int_0^{Z_s}(Z-\overline{Z})^2\cdot\frac{dm_{bg}(Z)}{dt}dZ\\ &\quad +\frac{1}{m_s}(\frac{\sum dm_{bg}}{dt}-\frac{dm_g}{dt}-\frac{dm_f}{dt})\overline{Z'^2}\end{aligned} \tag{A.43}$$

Observing that $\frac{dm_{sat}}{dt}=\frac{1}{Z_s}\frac{dm_f}{dt}$ and $\frac{dm_{evap}}{dt}=\left(1-\frac{1}{Z_s}\right)\frac{dm_f}{dt}$,

$$\begin{aligned}\frac{d\overline{Z'^2}}{dt} &= -2\cdot C_{diss}\cdot\frac{\overline{Z'^2}}{\frac{\tilde{K}}{\tilde{\epsilon}}}+\frac{1}{m_s}[\overline{Z}^2\left(\frac{dm_g}{dt}+\left(1-\frac{1}{Z_s}\right)\frac{dm_f}{dt}\right)+\frac{1}{Z_s}(Z_s-\overline{Z})^2\frac{dm_f}{dt}\\ &\quad -\left(\frac{dm_g}{dt}+\frac{dm_f}{dt}\right)\overline{Z'^2}-\int_0^{Z_s}(Z-\overline{Z})^2\cdot\frac{dm_{bg}(Z)}{dt}dZ+\overline{Z'^2}\frac{\sum dm_{bg}}{dt}]\\ &= \underbrace{-2\cdot C_{diss}\cdot\frac{\overline{Z'^2}}{\frac{\tilde{K}}{\tilde{\epsilon}}}}_{I}+\underbrace{\frac{1}{m_s}\cdot\left(\overline{Z}^2-\overline{Z'^2}\right)\cdot\frac{dm_g}{dt}}_{II}\\ &\quad +\underbrace{\frac{1}{m_s}\cdot\left(\overline{Z}^2\cdot\left(1-\frac{1}{Z_s}\right)+\left(Z_s-\overline{Z}\right)^2\cdot\frac{1}{Z_s}-\overline{Z'^2}\right)\cdot\frac{dm_f}{dt}}_{III}\\ &\quad +\underbrace{\frac{1}{m_s}\left(\overline{Z'^2}\frac{\sum dm_{bg}}{dt}-\int_0^{Z_s}(Z-\overline{Z})^2\cdot\frac{dm_{bg}(Z)}{dt}dZ\right)}_{IV}\end{aligned} \tag{A.44}$$

A.4 Stochastic Reactor Model

A stochastic reactor model was investigated during this research. The model contains a number (calibratable) of particles which track the mixing and subsequent combustion within the combustion chamber, see Fig A.4 for an example distribution of the equivalence ratio across the particles during combustion. The results from this model differ to those from the control orientated model in that they cover the whole combustion chamber (not only the spray) and secondly do not have distinct burnt and unburnt gases (so the equivalence ratio does not evolve during combustion, it only shows how the fuel and air mix in the spray).

This means that the mean equivalence ratio from the SRM model is leaner as the fuel is injected since it is taken across the whole cylinder, whereas the control orientated

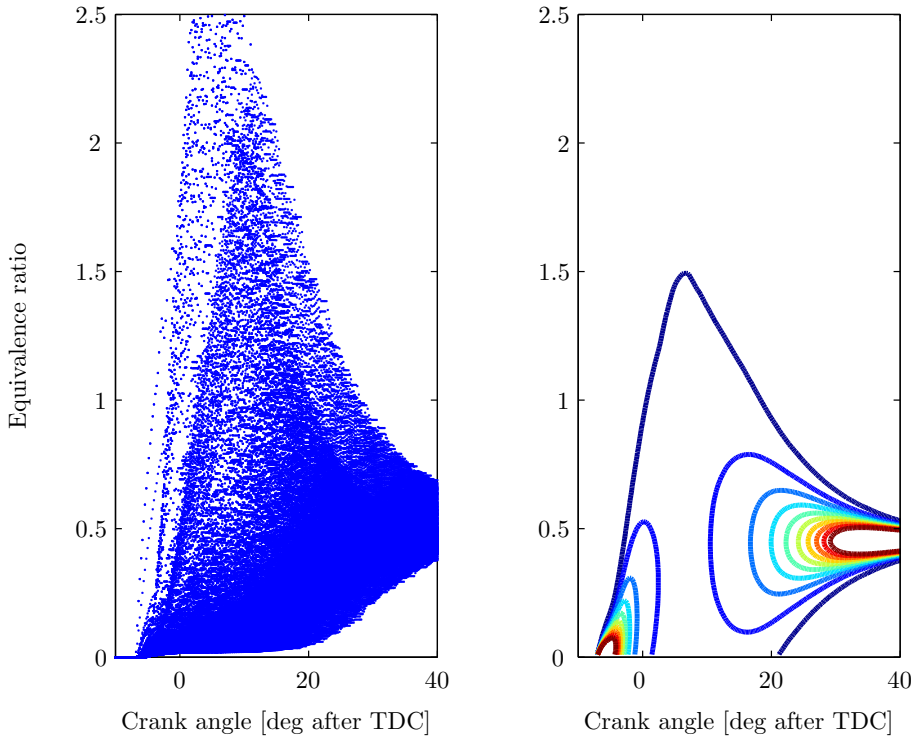

Figure A.4: Distribution of equivalence ratio for SRM particles during combustion (left) and the SRM results fitted to a normal distribution (right).

model only considers the spray, so when the fuel is injected it is a rich mixture. See Fig A.5 to see the mean equivalence ratio compared between the two models. The control orientated mixing has been presented with respect to both the spray and the combustion chamber, with the two curves meeting when the spray volume covers the whole chamber. Comparing the mixing curves with respect to the chamber, it can be observed that the control orientated model mixes slightly quicker than the SRM model. However, since the control orientated model is closer to the measured data, see Fig 2.19, it may be the SRM model which may need more calibration.

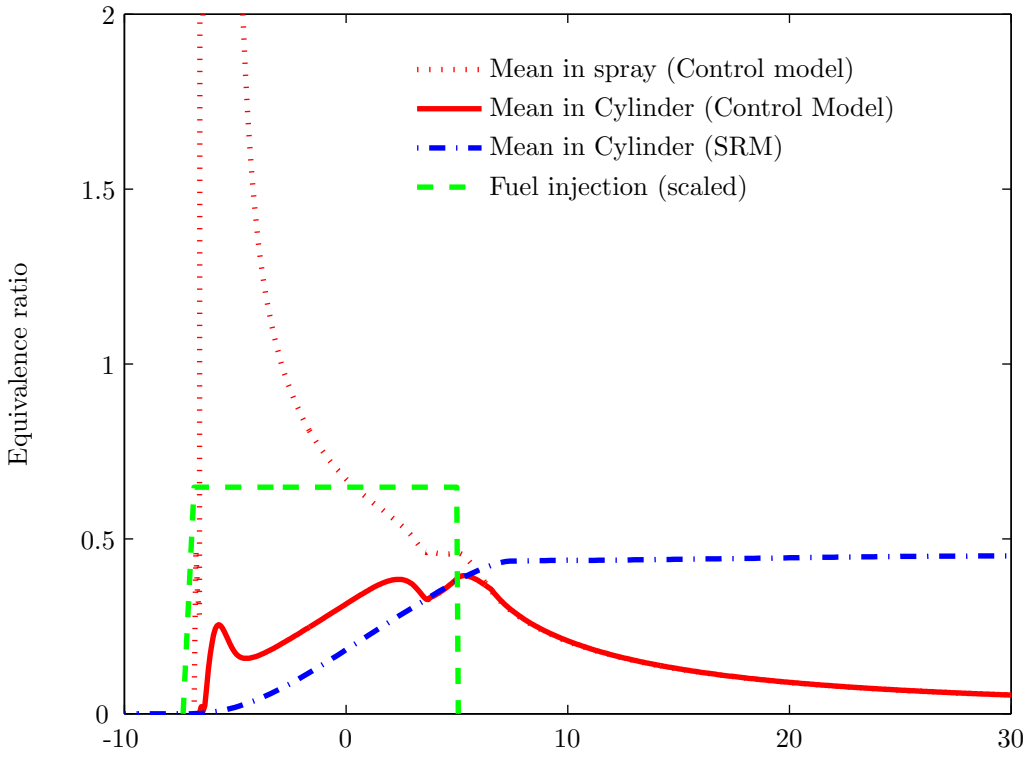

Figure A.5: Comparison of the mean equivalence ratio between the SRM model and the control orientated model.

A.5 Emissions validation for full data set

Figures A.6 and A.7 compare measured and predicted emissions index for all the full validation data set, including the points with very low emissions.

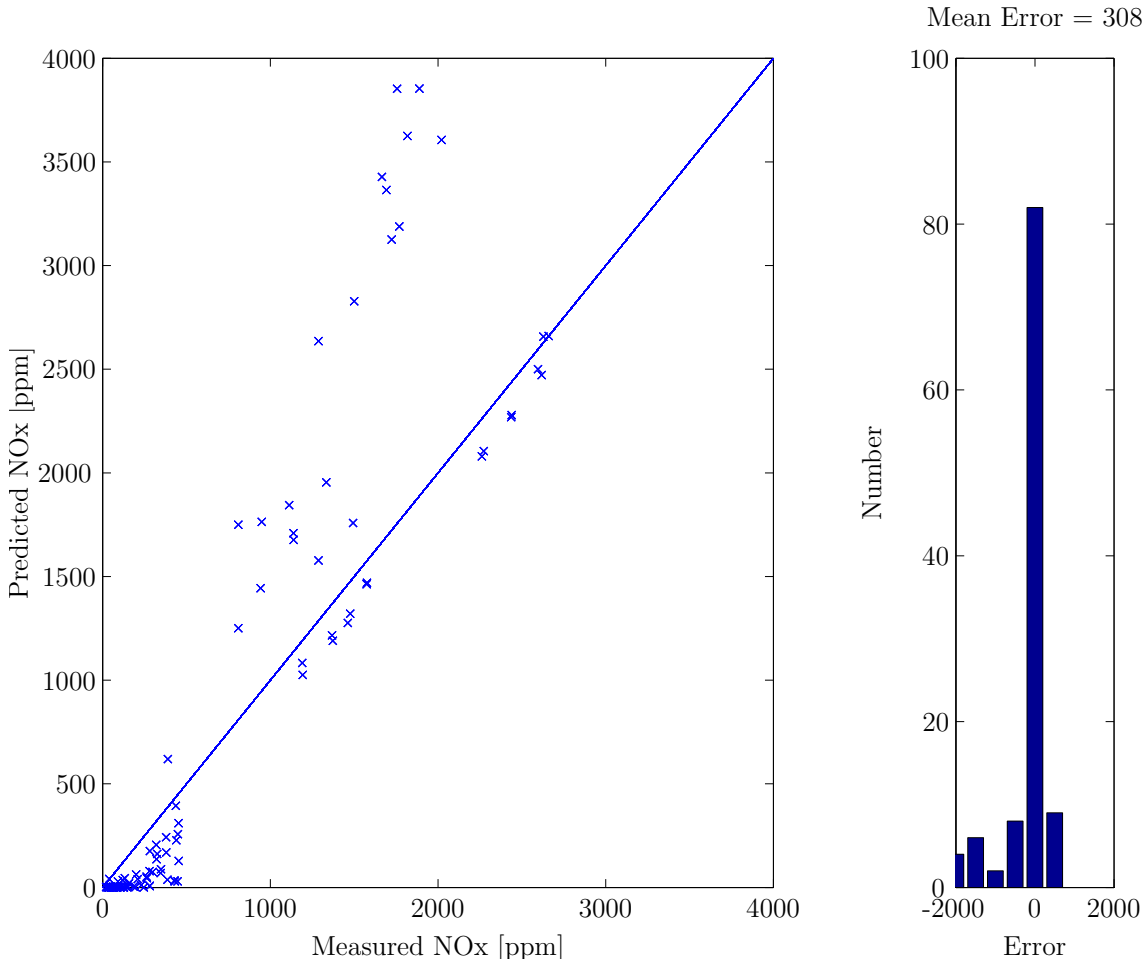

Figure A.6: Correlation of the NO_x emissions for the full validation data set, with distribution of errors in the right plot.

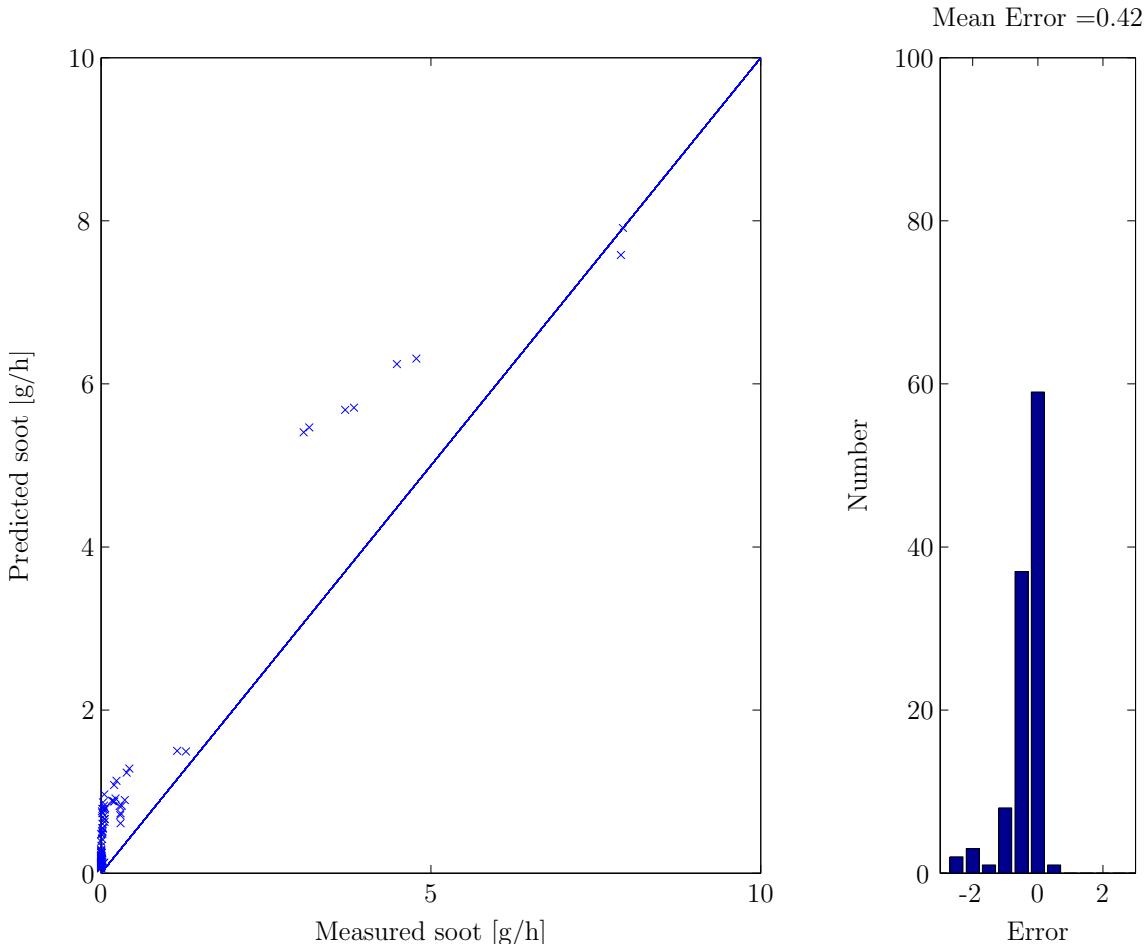

Figure A.7: Correlation of the soot emissions for the full validation data set, with distribution of errors in the right plot.

Appendix B

Parallel trust region search

B.1 An optimisation algorithm using a small number of iterations

B.1.1 Requirements

This Appendix details an iterative search algorithm that is used to find a minimum of a cost function, J. The algorithm is designed for the case where the time for finding the minimum is limited by the computational hardware and required update rate, whilst a significant time is required for the evaluation of the cost function. This gives rise to the situation where only a small number of iterations, n_{iter}, (typically less than five) can be contemplated. To help with the search, parallel computing is available, so that n_{eval} function evaluations can be executed in each iteration.

B.1.2 Definitions

The problem is defined by:

$$\min_{u}(J) \tag{B.1}$$

where:

$$\begin{aligned} J &= f_1(y) + f_2(u) \\ y &= f_3(x, u) \end{aligned} \tag{B.2}$$

and J = cost function, x = current system states, u = system input and y = system output.

B.1.3 Parallel Trust Region (PTR) algorithm

The algorithm is based on an approach known as a Trust Region (TR) search. A typical TR search uses a function (eg polynomial) to approximate the unknown part of the

problem, for example the system output, $y(x, u)$. The cost function is evaluated at a small number of points defining the TR and the results compared to the cost estimated by the approximated function. If the error is small then the TR can be expanded, since the approximation is good, if the error is large then the TR must be reduced until the error reduces.

With parallel computing facilities available, the search can be carried out in several regions at once. The parallel trust region (PTR) algorithm divides the search area into a number of equal regions and fits a simple (in this case linear) function to the underlying system output. The approximation is carried out by evaluating the system output at the mid-point of the sub-region and at the verticies and fitting a linear function to these points. In the two-dimensional case there are five evaluations per sub-region.

The linear function is used to evaluate the minimum of the cost function in each sub-region. This minimum is then used as the centre for the sub-region of the next iteration.

Since the whole search space has already been covered with the multiple regions, the PTR algorithm does not need to check how good the approximation is (since the region cannot be expanded), but rather simply reduces the search region around the estimated minimum. The reduction of the sub-region size is the main tuning parameter for the PTR algorithm.

The PTR algorithm for two dimensions is defined as follows:

1. `divide search region into` $n_{eval}/5$ `sub-regions`
2. $i_{iter} = 0$
3. `while` $i_{iter} < n_{iter}$`, for each sub-region`
4. $i_{iter} = i_{iter}$`+1`
5. `evaluate` y `at the 5 points (centre and each of the four vertices)`
6. `fit a plane to the 5 evaluations of` y
7. `calculate minimum of the cost function,` J `, within the sub-region`
8. `use the minimum as the centre of the new sub-region in the next iteration`
9. `reduce the size of the sub-region by` γ_{iter}

10. `end`

11. `calculate the minimum of` J `from all evaluations of` J

12. `end`

B.2 Worked example

The Goldstein and Price function was used to test and illustrate the optimisation algorithm. This is a non-linear function of two inputs defined as follows:

$$\begin{aligned} y(u_1, u_2) &= \left[1 + (u_1 + u_2 + 1)^2(19 - 14u_1 + 3u_1^2 - 14u_2 + 6u_1u_2 + 3u_2^2)\right] \\ &\quad \left[30 + (2u_1 - 3u_2^2)^2(18 - 32u_1 + 12u_1^2 + 48u_2 - 36u_1u_2 + 27u_2^2)\right] \end{aligned} \quad \text{(B.3)}$$

where $u_1, u_2 \in [-2, 2]$. y is included in the following cost function:

$$J(u_1, u_2) = \frac{y(u_1, u_2)}{\bar{y}} + Q\left\{(u_1 - \bar{u_1})^2 + (u_2 - \bar{u_2})^2\right\} \quad \text{(B.4)}$$

where $\bar{u}$ is a set-point and $\bar{y}$ is the value of y at the set-point. The function y and the corresponding cost function J at the set-point $u_1 = -1$, $u_2 = 1$ are illustrated in Figure B.1, with the minimum of the cost function shown with a green cross.

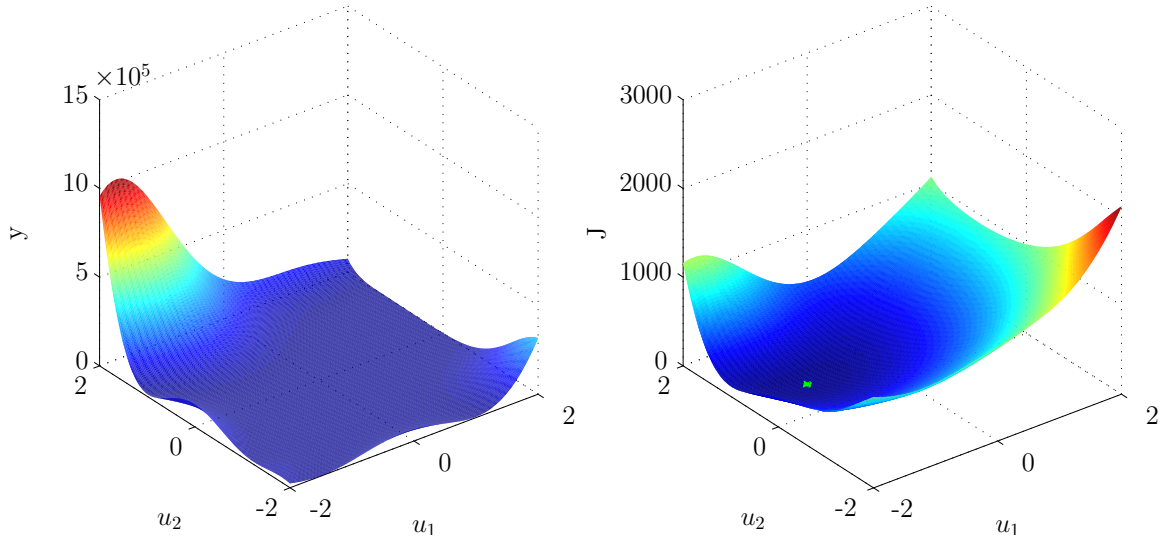

Figure B.1: Goldstein and Price function (left) with corresponding cost function (right). The minimum of cost function is shown with a green cross

The algorithm divides the search space into a number of regions, in this example a two-by-two grid is used with four sub-regions. A linear function is fitted to the the

underlying function y by evaluating y at five points, the centre and each vertex of the sub-region and performing a least squares fit. In this example, this results in four planes fitted to the function y, with corresponding surfaces fitted to the cost function as shown in Figure B.2.

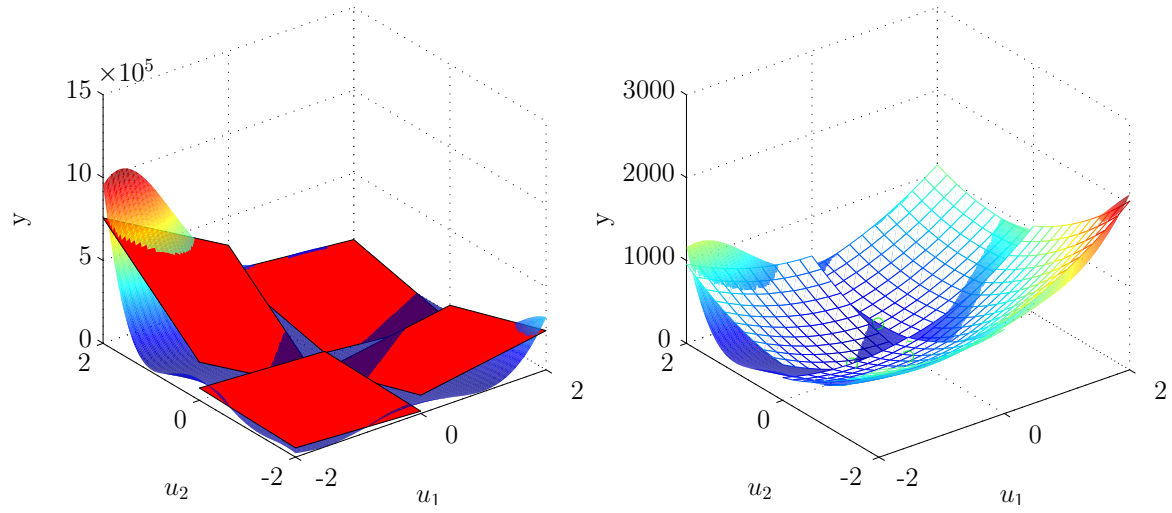

Figure B.2: Goldstein and Price function with fitted planes (left). Corresponding cost function with fitted surfaces (right).

It is not necessary to evaluate the surfaces fitted to the cost function since the minimum can be evaluated directly. Consider the equation of the plane:

$$y(u_1, u_2) = a + bu_1 + cu_2 \tag{B.5}$$

Then the cost function B.4 can be written as:

$$J(u_1, u_2) = \frac{a + bu_1 + cu_2}{\bar{y}} + Q\left\{(u_1 - \bar{u_1})^2 + (u_2 - \bar{u_2})^2\right\} \tag{B.6}$$

Necessary conditions for a minimum of the cost function can be found by setting the partial differentials with respect to u_1 and u_2, equal to zero.

$$\frac{\partial J}{\partial u_1} = \frac{b}{\bar{y}} + 2Q(u_1^* - \bar{u_1}) \tag{B.7}$$

$$u_1^* = \bar{u_1} - \frac{b}{2Q\bar{y}} \tag{B.8}$$

where u_1^* is the value of u_1 at the minimum of J. Similarly for u_2 gives:

$$u_2^* = \bar{u_2} - \frac{c}{2Q\bar{y}} \tag{B.9}$$

These values of u_1^* and u_2^* are set as the centre of the sub-region for the next iteration. The size of the sub-region is reduced at each iteration by a constant factor, γ_{iter}, and the process is repeated.

For this example, three iterations are carried out. The results are illustrated in Figure B.3.

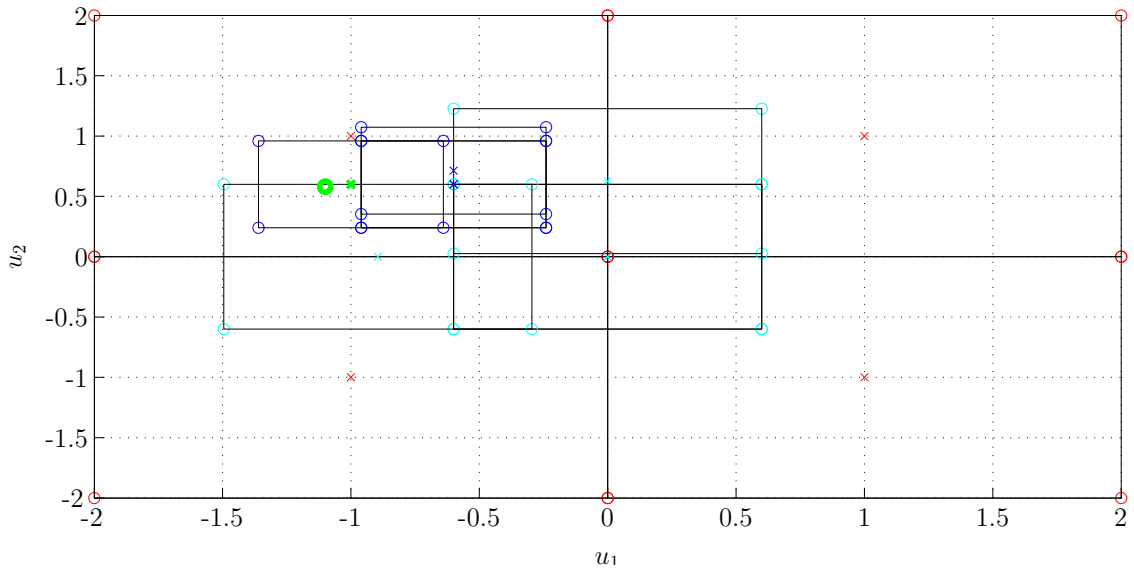

Figure B.3: Sample points for evaluating the function y. Crosses are at the centre of the sub-regions, circles at the vertices of the sub-regions. Red is for the first iteration, cyan is the second iteration and blue is the third iteration. Finally, the green cross is the minimum found by the search algorithm and the green circle is the overall minimum.

By way of comparison, the PTR algorithm is compared to a grid search, with an equi-spaced grid of 8 by 8 points, see Figure B.4. Here it can be seen that the error is greater than that found using the PTR algorithm with fewer iterations.

This comparison is somewhat arbitrary since the equi-spaced search will sometimes be 'lucky' and be close to the minimum and other times will be further away. It is therefore important to assess the PTR algorithm over a wide range of functions. This is achieved here by varying the set-point across the search area and looking at the distribution of errors from the actual minimum. These distributions are also compared to the maximum error when the optimum point is at the middle of the equi-spaced grid. The results from running these different functions is shown in Figure B.5, where it can be seen that the PTR algorithm demonstrates a significant improvement over the equi-spaced search.

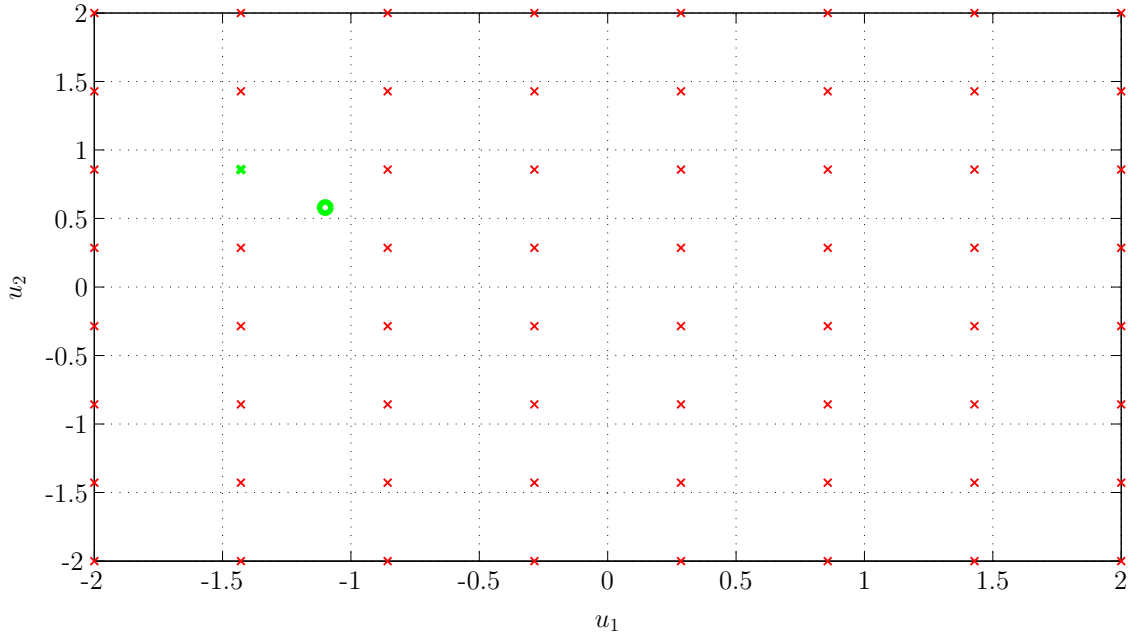

Figure B.4: Sample points for evaluating the function y using an equi-spaced grid. Red crosses are the sample points, the green cross is the minimum found by the search algorithm and the green circle is the overall minimum.

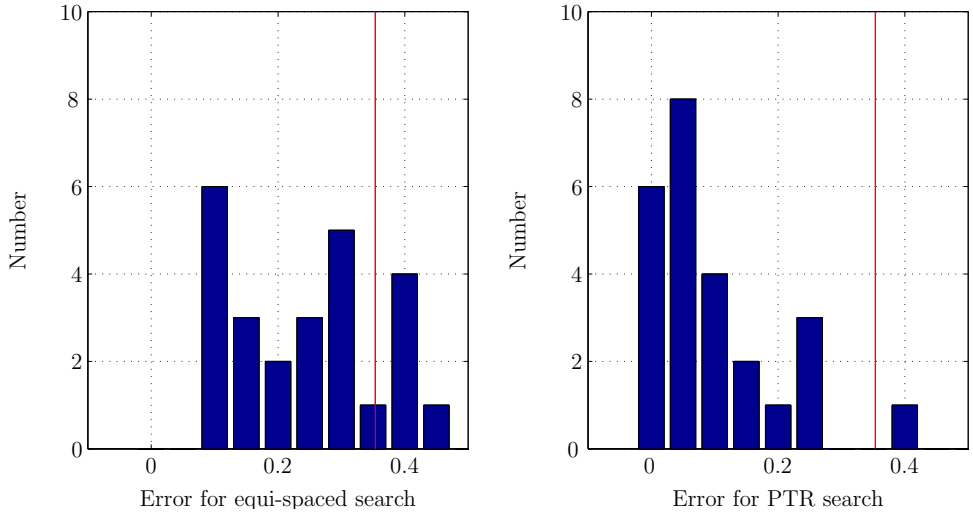

Figure B.5: Distribution of the errors for the equi-spaced search (left) and the PTR search (right), with the error at the centre of the equi-spaced grid shown by the red line.

B.3 Study of PTR parameters

There are a number of parameters associated with the PTR algorithm which require investigation in order to understand their impact on the search algorithm performance. The first of these is the reduction ratio between iterations, γ_{iter}. The search algorithm has been run with a range of γ_{iter} between 0.1 and 0.8, with the results shown in Figure B.6. This shows that there may be an optimum reduction ratio, which for this example is 0.6. Greater than this and the search doesn't reduce the size of the sub-region quick enough to give sufficient resolution. Smaller than this and the inaccuracy of the initial minimum calculation can lead to the updated search area not including the minimum, so the search effectively 'misses' the minimum, see Figure B.7 for the case with $\gamma_{iter} = 0.1$.

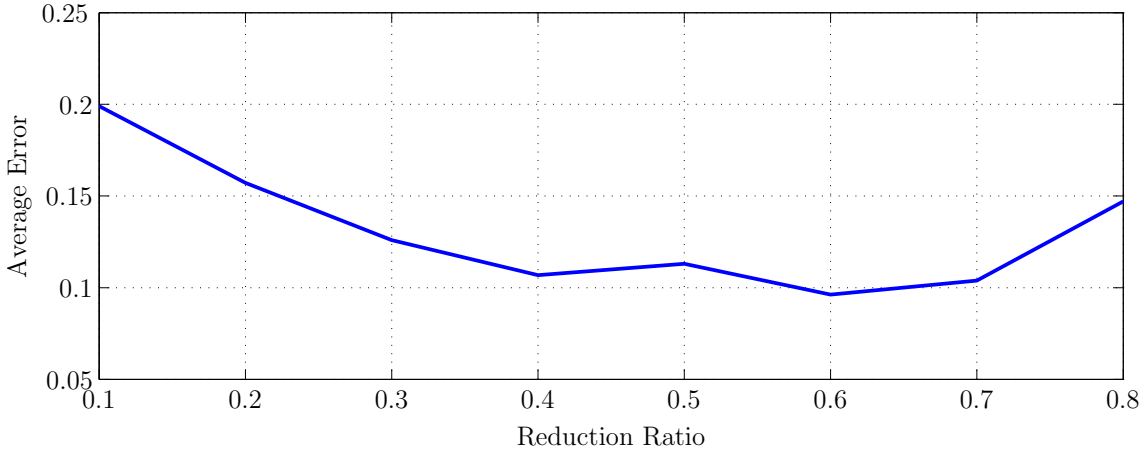

Figure B.6: Average error for different values of reduction ratio, γ_{iter}.

B.4 Error analysis

The worst case error for the PTR algorithm occurs when the actual minimum is the furthest from any of the sample points selected by the PTR algorithm. The maximum value for the worst case error occurs when the second iteration is within one half of the original sub-region - giving the maximum error in u as $l/4$, as illustrated in Figure B.8.

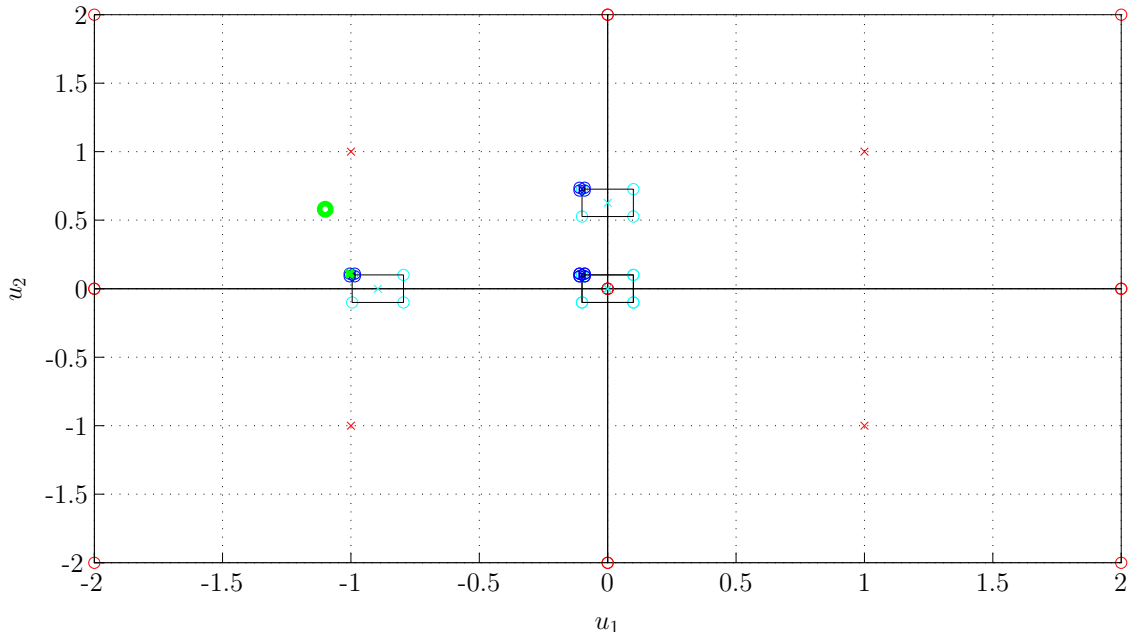

Figure B.7: Sample points for the example with a reduction ratio which is too small, $\gamma_{iter} = 0.1$.

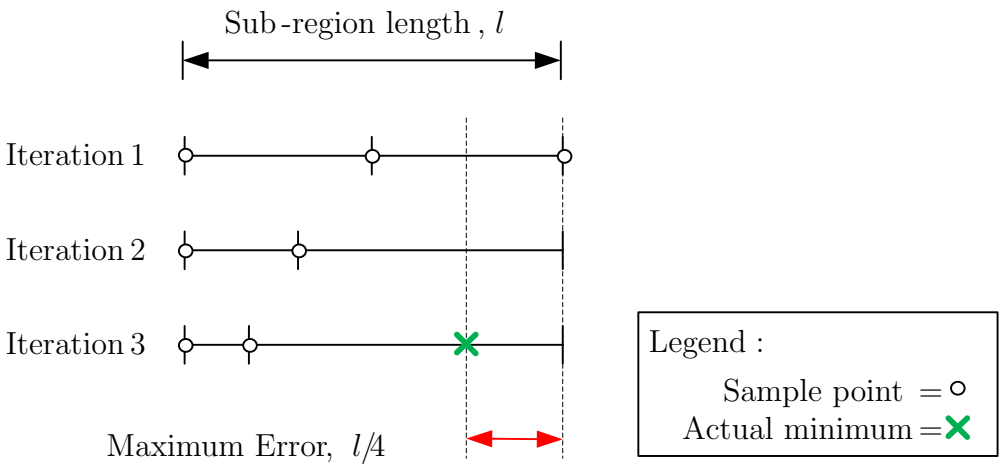

Figure B.8: Analysis of the maximum potential error.

B.5 Refinements

One area of refinement for the PTR algorithm is to consider how to handle the case where the minimum from the first iteration is located on or near the boundary of the sub-region. In this case, the second iteration may have its centre close to the edge of the original sub-region and may then extend outside the original sub-region. This is a likely scenario with the PTR algorithm since there is only one sub-region which has the minimum, so all the other sub-regions will tend to migrate towards that sub-region.

The algorithm designer has the choice of constraining each of the parallel searches within their original sub-regions or allowing the parallel searches to migrate across boundaries (whilst remaining in the initial overall search area). The implementation shown in the example in Figure B.3 has allowed the search to migrate across boundaries. This may improve the overall accuracy by having more samples around the minimum, however it does increase the maximum potential error and also the risk of homing in on a local minimum. This is a trade-off which can be made during the algorithm design.

Appendix C

Implementation of an iterative MPC algorithm on an FPGA

C.1 Definition of an example problem for Model Predictive Control (MPC)

The objective of this Appendix is to apply the PTR algorithm from Appendix B to a predictive control problem and implement it on a FPGA. The example system used for this work is a bicycle model. See [62] for the derivation of the linearised equations of motion of the constant-speed, straight-running nonholonomic bicycle. Equation C.1 describes the motion at speed v with lean angle ϕ and steer angle δ. The inputs to the system are the torques T_ϕ and T_δ about the lean and steering axes respectively.

$$\mathbf{M}_0\ddot{\mathbf{q}} + v\mathbf{C}_1\dot{\mathbf{q}} + \left[g\mathbf{K}_0 + v^2\mathbf{K}_2\right]\mathbf{q} = \mathbf{f} \tag{C.1}$$

where $\mathbf{q} = [\phi, \delta]^T$ and $\mathbf{f} = [T_\phi, T_\delta]^T$, $\mathbf{M}, \mathbf{C}_1, \mathbf{K}_0$ and $\mathbf{K}_2$ are matrices defined by the bicycle geometry, g is gravity and v is the forward speed of the bicycle.

Taking equation C.1, multiplying by $\mathbf{M}_0^{-1}$ and grouping variables $\mathbf{M} = \mathbf{M}_0^{-1}$, $\mathbf{S} = \mathbf{M}_0^{-1}\mathbf{C}_1 v$ and $\mathbf{K} = \mathbf{M}_0^{-1}\left[g\mathbf{K}_0 + v^2\mathbf{K}_2\right]$ gives:

$$\ddot{\mathbf{q}} + \mathbf{S}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{M}\mathbf{f} \tag{C.2}$$

Equation C.2 has been converted to state-space form, Equation C.3, by introducing additional states for the derivatives of $\mathbf{q}$. Using the notation $\mathbf{q} = [q_1, q_2]^T$, the equation can be rewritten using x_1 = lean angle (q_1), x_2 = rate of change of lean angle ($\dot{q_1}$), x_3 = lean angle (q_2), x_4 = rate of change of lean angle ($\dot{q_2}$), $u_1 = 0$, $u_2 = T_\phi$, $u_3 = 0$ and $u_4 = T_\delta$.

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \tag{C.3}$$

where

$$A = -\begin{bmatrix} 0 & -1 & 0 & 0 \\ K_{11} & S_{11} & K_{12} & S_{12} \\ 0 & 0 & 0 & -1 \\ K_{21} & S_{21} & K_{22} & S_{22} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & M_{11} & 0 & M_{12} \\ 0 & 0 & 0 & 0 \\ 0 & M_{21} & 0 & M_{22} \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}.$$

The stability of the bicycle is a function of v. At low speeds and high speeds, the bicycle is unstable with a self stable region at mid-speeds. The test case for this study considers a bicycle travelling in a straight line at v in the high speed unstable region. The cyclist then approaches a corner where they follow a target lean angle. This can be achieved using both lean and steering torques as control inputs. The following sections develop two MPC approaches for this problem.

C.2 'Conventional' Model Predictive Control

This constrained control problem can be addressed using 'conventional' MPC, where a Quadratic Program (QP) is used to solve the optimisation of a cost function over the prediction horizon. The cost function penalises deviations of the predicted controlled outputs, $\hat{y}(k+i|k)$ from a reference trajectory, $r(k+i|k)$ plus the change in control input values, $\Delta\hat{u}$. $(k+i|k)$ corresponds to the evaluation of the function at timestep $k+i$ based on conditions at k.

$$J = \sum_{i=1}^{N_p} \|\hat{y}(k+i|k) - r(k+i|k)\|_Q^2 + \sum_{i=1}^{N_u} \|\Delta\hat{u}(k+i|k)\|_R^2 \tag{C.4}$$

The Matlab MPC toolbox was used to implement this controller, with the following parameters selected for the MPC implementation; a prediction horizon of 0.8 seconds and only one control move. The controllers are constrained to +/- 2 Nm.

Over the test case, the system is stabilised and tracks the lean angle set-point step change to -0.5 rad. The results for this simulation are presented in Figure C.1. This figure also shows the expected counter-steering behaviour where initially the steering angle is positive and then it goes negative as the bicycle settles into the corner.

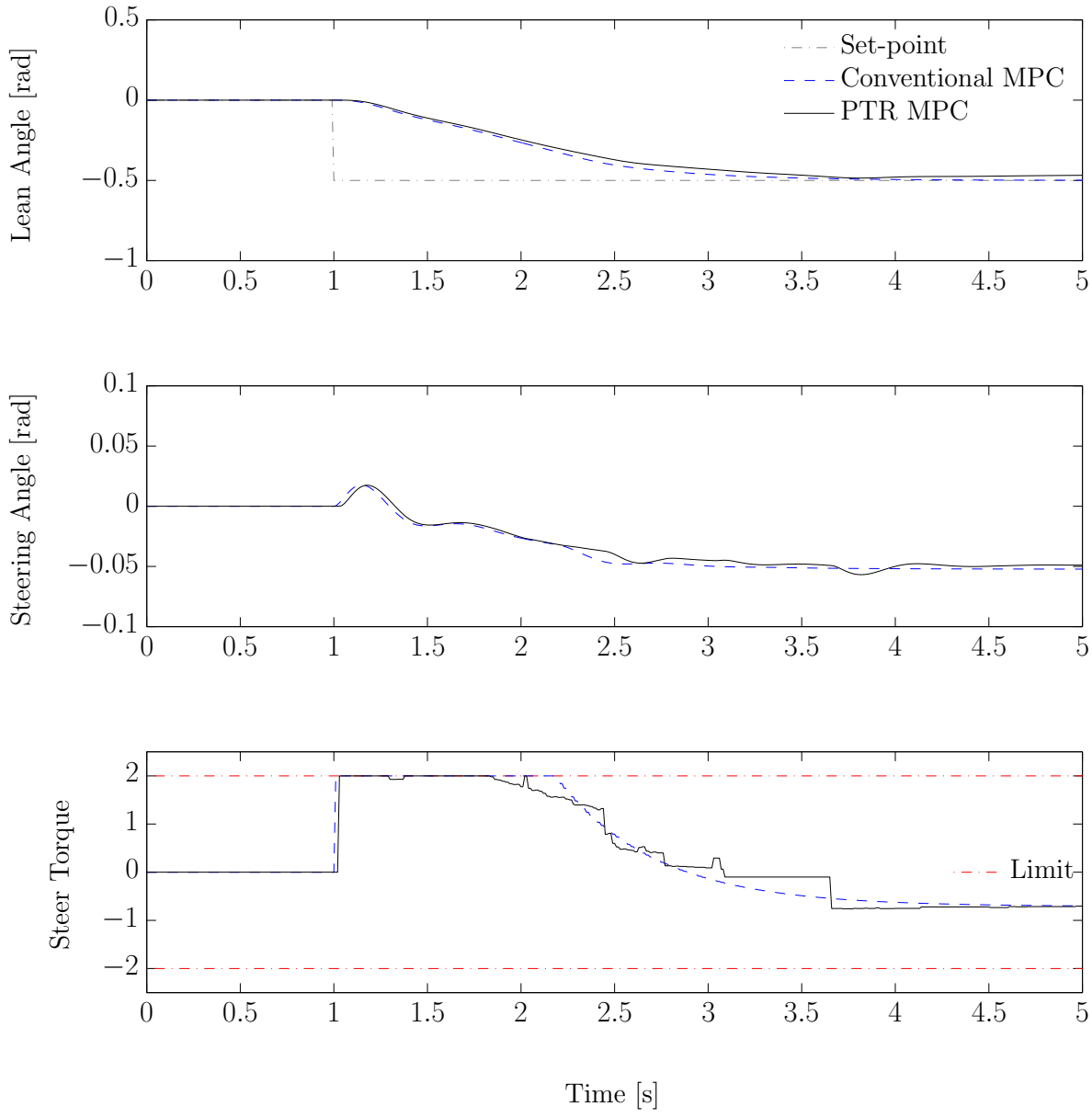

Figure C.1: Simulation results for the 'conventional' MPC and the parallel trust region (PTR) MPC algorithm. The upper two plots show the system outputs and set-point. The lower plot shows the control action associated with these responses.

C.3 MPC using a Parallel Trust Region (PTR) search

An alternative solution is developed here for situations where the computational resources are not sufficient for a QP optimisation and a fast response is required. For the case when the optimisation surface cannot be described mathematically, an iterative approach is needed to find the minimum of the cost function. The PTR search, described in Appendix B, is used here to solve the optimisation in the Model Predictive controller. The overall controller structure is described in Figure C.2.

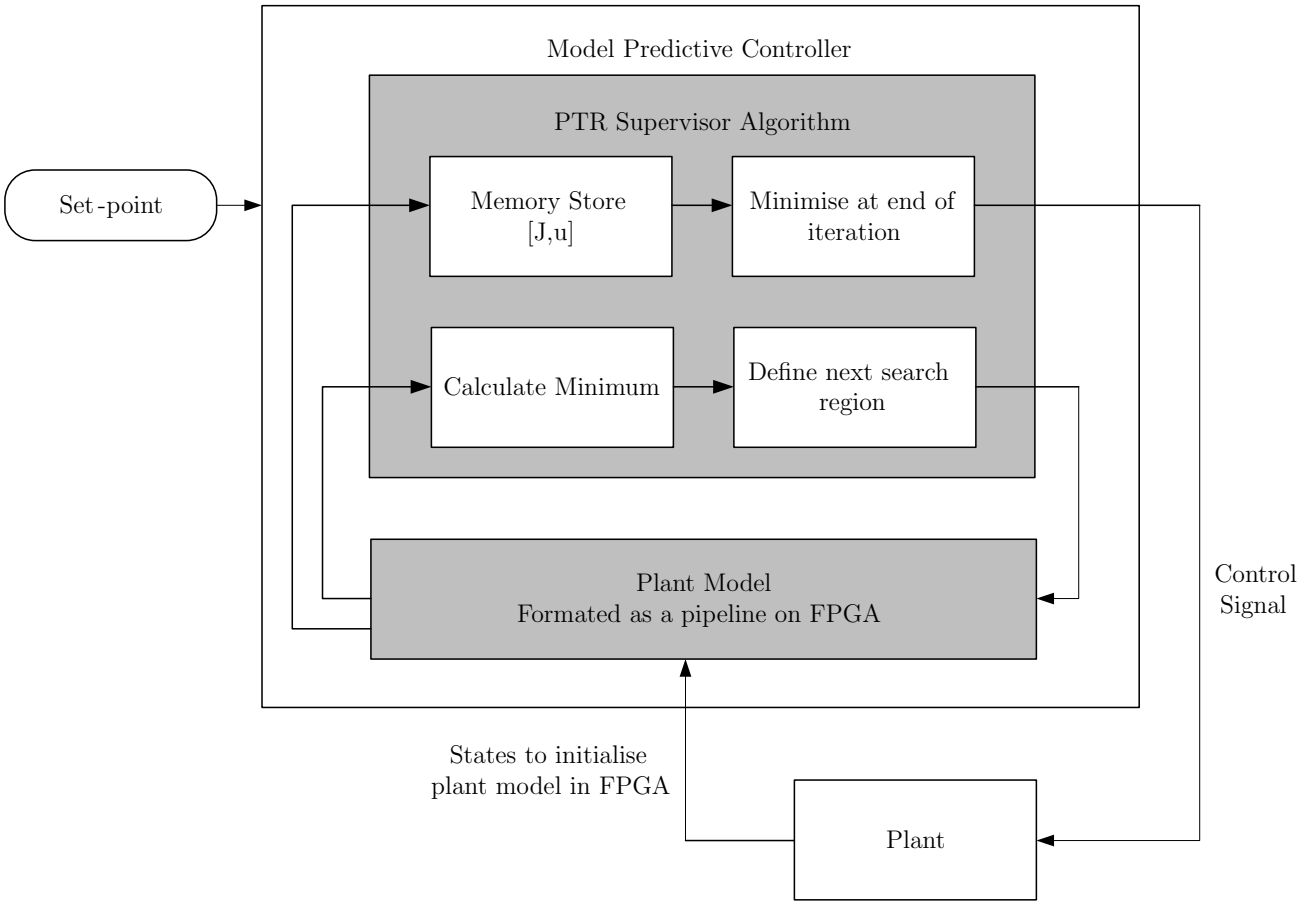

Figure C.2: Architecture for a model predictive controller using a parallel trust region search.

The PTR supervisor sends a series of candidate control actions to the FPGA which calculates the system response over the prediction horizon. The FPGA also outputs the cost associated with each of the candidate control actions. Once the FPGA has evaluated five control actions, covering one sub-region, the outputs are used to select the next iteration of this sub-region. These are fed to the FPGA once the final sub-region has completed its first iteration, see Figure C.3. This configuration is known as a pipeline, where the next input is fed to the plant model before the previous solution has completed.

The five control inputs for each of the four sub-regions and three iterations, as used in Appendix B, give a total of 60 evaluations of the system response. If one evaluation takes n timesteps, then a sequential processor would take $60n$ timesteps, whereas the pipeline configuration only requires $(60 + n)$ timesteps, or a reduction of $(59n - 60)$ timesteps. A simple system such as this with a prediction horizon of four, requires four iterations of the state space model over four computation timesteps, resulting in a reduction of 73%. For more complex systems which take longer to evaluate, the time savings will be greater.

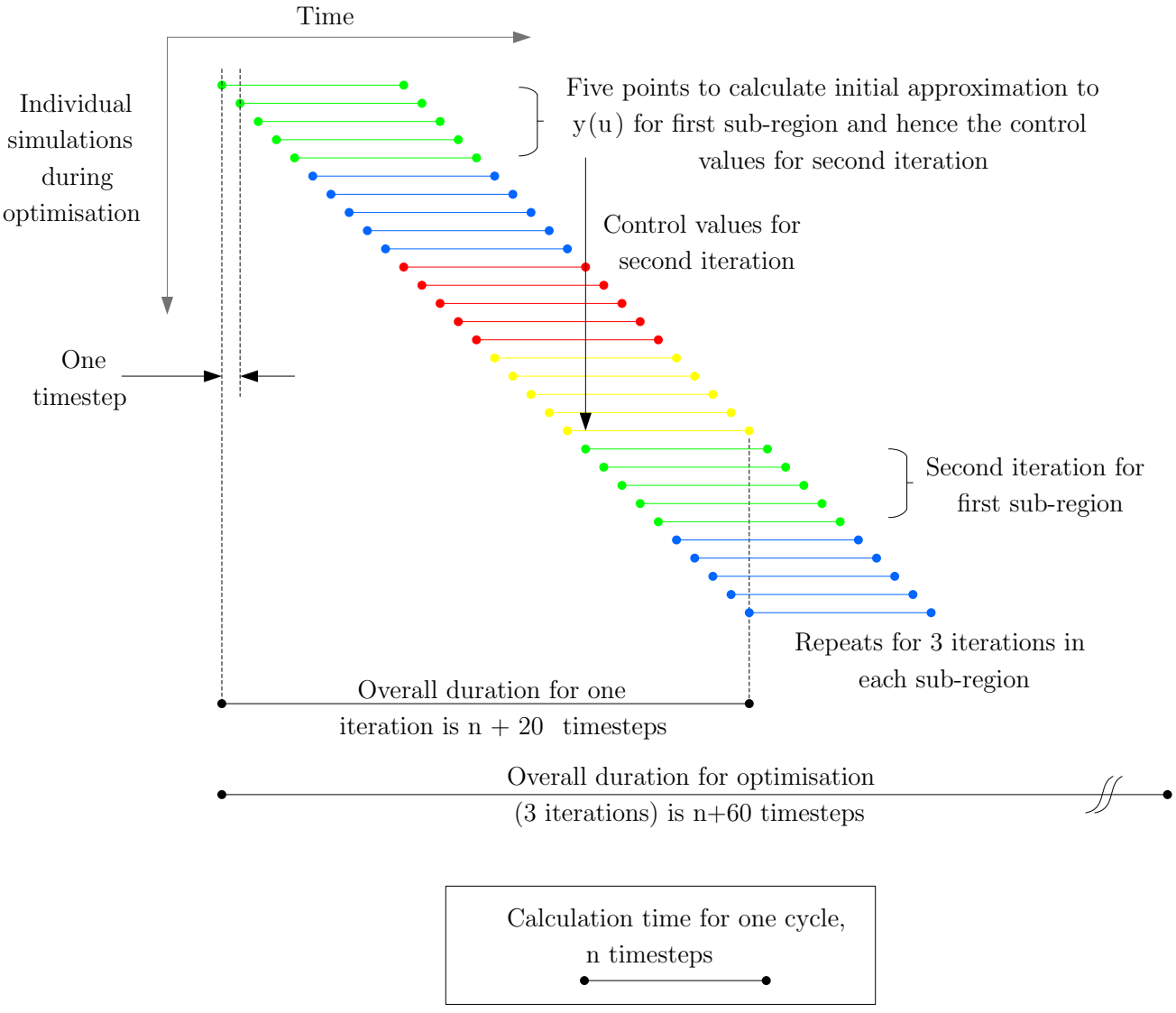

Figure C.3: An illustration of the pipeline configuration for calculating the system response during a parallel trust region search.

The algorithm has been implemented in a System on a Chip (SoC) which allows the PTR supervisor to be coded in C on micro-processor and the pipelined plant model coded in HDL on the FPGA.

The optimisation is illustrated in Figures C.4 and C.5 which show the same data with different time axes. The upper plot shows the candidate control actions at each timestep. These are fed to the FPGA which predicts the system response over the prediction horizon for each control action. The prediction horizon timestep is 0.2 seconds with a four step horizon giving a prediction horizon of 0.8 seconds, whilst the controller update rate is 0.02 seconds. In the middle plot, to illustrate the algorithm, the responses for each of the candidate control actions are plotted at the control update points. The lower plot captures the costs associated with the candidate control actions.

The delay associated with the PTR algorithm response can be seen in Figure C.4. This is caused while the controller evaluates the different control actions. The target is fixed during each optimisation, so if the target changes during a optimisation, the controller has to wait until the next update step before starting the new optimisation. In this case, the set-point changes at 1.0 seconds, midway through an optimisation. The solver needs to complete this calculation and do another calculation before the control output is updated. A fast controller update rate is therefore required to ensure this delay is kept small enough to be acceptable.

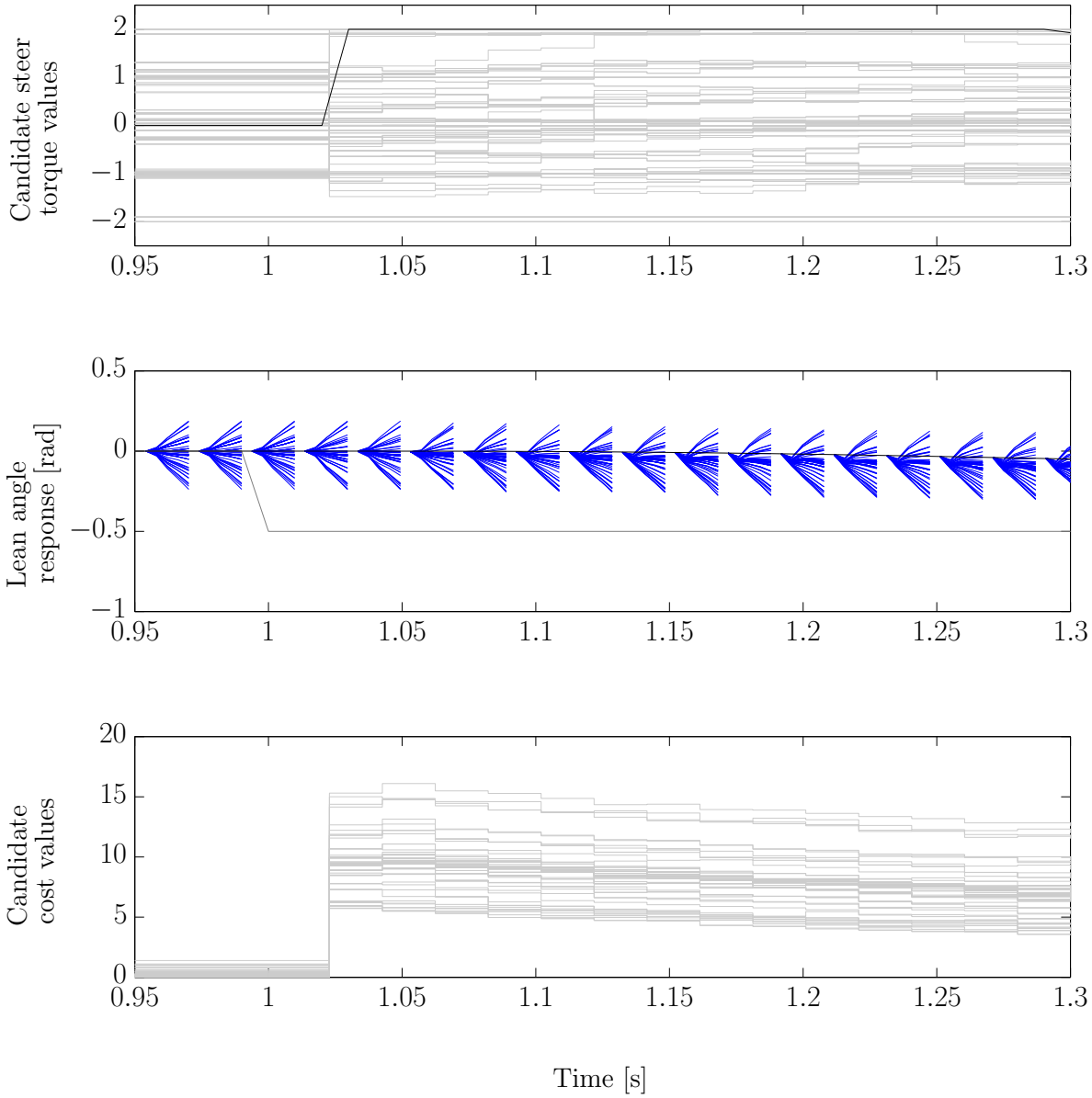

Figure C.4: Detail of the PTR optimisation. Upper plot shows the candidate control actions being assessed at each timestep in grey, with the optimum control action in black. The middle plot shows the simulated responses for each of the control actions in blue over the prediction horizon, at each timestep. Note the prediction horizon (0.8 s) is compressed to fit on the plot at each control update point (0.02 s). The target set-point is shown in grey. The lower plot shows the cost associated with each candidate control action.

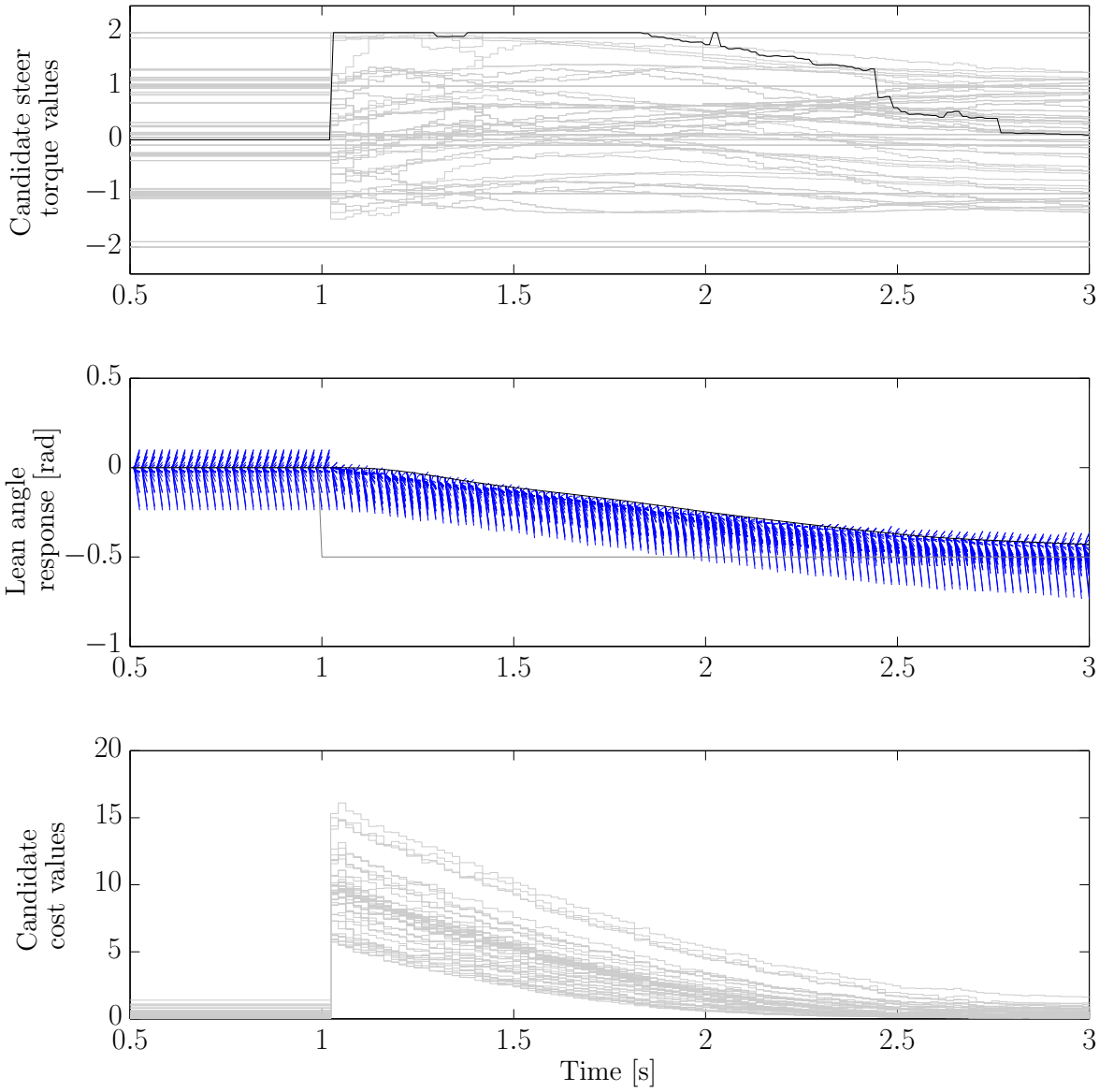

Figure C.5: The same results as Figure C.4, over a longer time period.

C.4 Discussion

The 'conventional' MPC and the PTR MPC have been simulated over the manoeuvre defined in Section C.1 with the results shown in Figure C.1. Both responses converge to the set-points and remain stable. The PTR MPC is more noisy with some jitter in the control signal. This is likely to be due to the lack of resolution resulting from the simplified optimisation. Whilst this is sub-optimal, the jitters do not pass through to the response of the bicycle and there is little difference between the lean angle and steering angle responses of both approaches.

This Appendix has shown how the Parallel Trust Region optimisation algorithm can be applied to a Model Predictive Control problem. The response was compared with the conventional QP solution and found to perform well. The algorithm has been implemented on an FPGA and demonstrated how this hardware can be used to pipeline the simulations, allowing a rapid calculation of system responses.

A key learning from this work has been the significance of the delays in the system. It is important in controlling fast systems to ensure that the control update rate is fast. This may require compromises in the optimisation to ensure the response is fast enough.

The toolchain and control approach have been validated and are suitable for extending to more complex predictive control problems such as Combustion Control.

Appendix D

Vehicle testing

D.1 Overview

The implementation of the model-based combustion control takes the control algorithm from the simulation environment to running an engine in a test vehicle. This process has a number of phases, progressively building and testing the control modules. The algorithms are implemented on an embedded prototype control system. Each module of the control system is tested on a test bench. The control system is commissioned in the test vehicle and finally the control algorithm is tested in a vehicle running on a chassis dynamometer. A chassis dynamometer is a test facility which allows the vehicle to run as if it is on the road and has emissions measuring equipment to monitor the harmful emissions, such as NO_x and Particulate Matter. This Appendix summarises the details of the embedded prototype control system, model simplifications and the test facilities.

D.2 Embedded prototype control system

D.2.1 Introduction

Moving the control algorithms from the simulation environment to embedded hardware is a well known process. It typically involves adding input and output modules to the software before compiling the software to run on the target hardware. This project has three additional complexities;

- the introduction of parallel computing hardware, an FPGA, which is not typically integrated with the model based control environment;
- control algorithms which are divided between a micro-processor and an FPGA adding an additional communication layer and

- the engine control unit interface is via a Controller Area Network (CAN), [3], whilst the FPGA board interface is via User Datagram Protocol (UDP), an internet protocol, requiring an additional communication network with a 'gateway' controller to interface between the two different protocols.

D.2.2 Hardware specifications

FPGA hardware

The FPGA hardware used in this project was the Xilinx Zynq System on a Chip (SoC), which includes both a FPGA and micro-processor on the same piece of silicon. The ZC702 development board, Figure D.1, was selected for this project since it is integrated with the Mathworks^{TM} HDL coder toolchain. Key features of the Zynq SoC include the following:

- Airtex 7 FPGA with 85k Logic Cells;
- Dual ARM®Cortex^{TM}-A9 processor, 866 MHz and
- Gigabit Ethernet communications.

See [7] and [8] for more details.

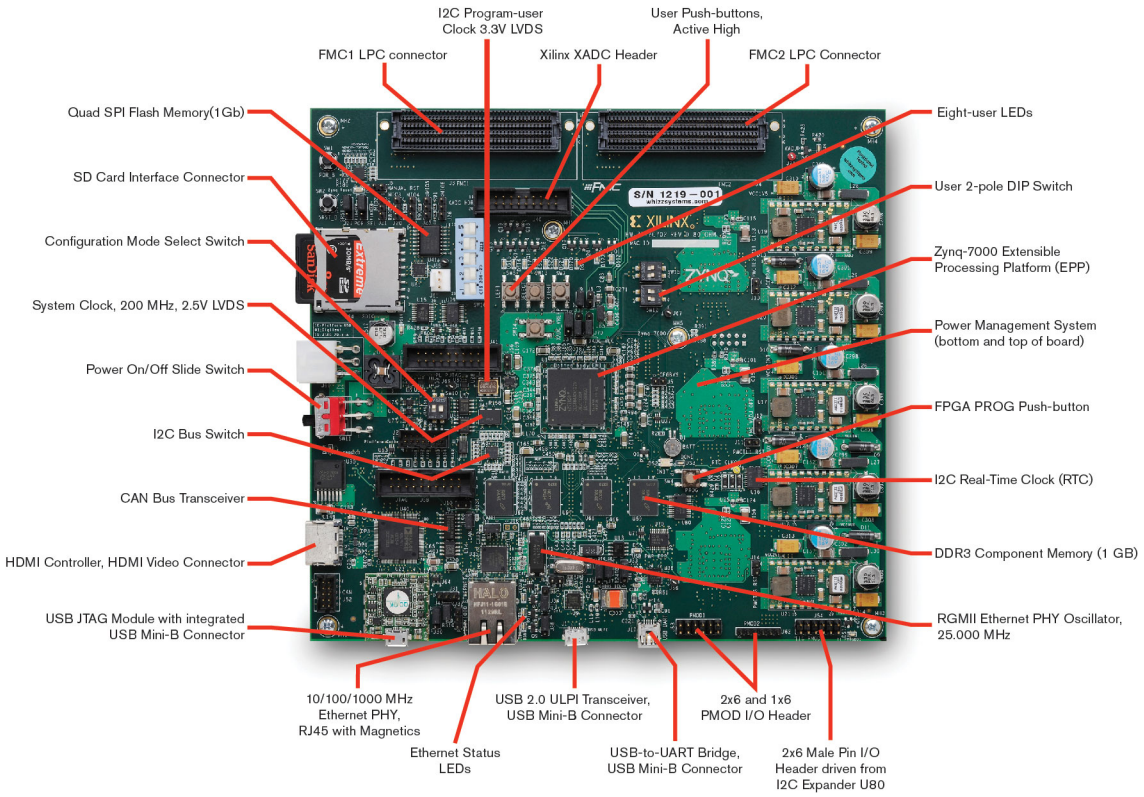

Figure D.1: Xilinx Zynq ZC702 development board, [6].

'Gateway' hardware

The communications gateway from the SoC to the engine controller is provided by a rCube controller, [9]. This is a prototyping control unit which is configured to pass signals between the SoC (UDP) and the engine control unit (CAN).

D.2.3 Control system architecture

The full control architecture is illustrated in Figure D.2, including the three communication buses:

- the AXI protocol between the FPGA and the micro-processor, internal to the Zynq SoC;
- the UDP protocol between the micro-processor and
- the CAN protocol between the rCube and the engine controller.

D.2.4 Data management

The configuration of the AXI bus within the Mathworks^{TM} toolchain does not allow the use of vectorised signals, resulting in many individual signals being passed on the AXI bus. The AXI bus has been tested at increasing data rates and has been found to maintain real time up to 10MHz. The model was re-structured to include this limitation, with data sent across the AXI bus in several data packages.

In the main text, it was explained that the combustion model has been divided into two parts, see Figure 4.4, with the micro-processor calculating the heat release and combustion and the FPGA calculating the NO emissions. The corresponding data management configuration is described below and illustrated in Figure D.3

Data packaging - Micro-processor

The results from one cycle simulation are stored to memory as the simulation progresses and sent to the FPGA at the end of the cycle. This requires two memory locations to store and send the data, with the pointers alternating between memory stores every cycle. The N_{cycle} elements of data are sent to the FPGA in parcels of $N_{parallel}$ points at each micro-processor timestep. After $\frac{N_{cycle}}{N_{parallel}}$ micro-processor timesteps, the full cycle has been transferred and the emissions can be calculated.

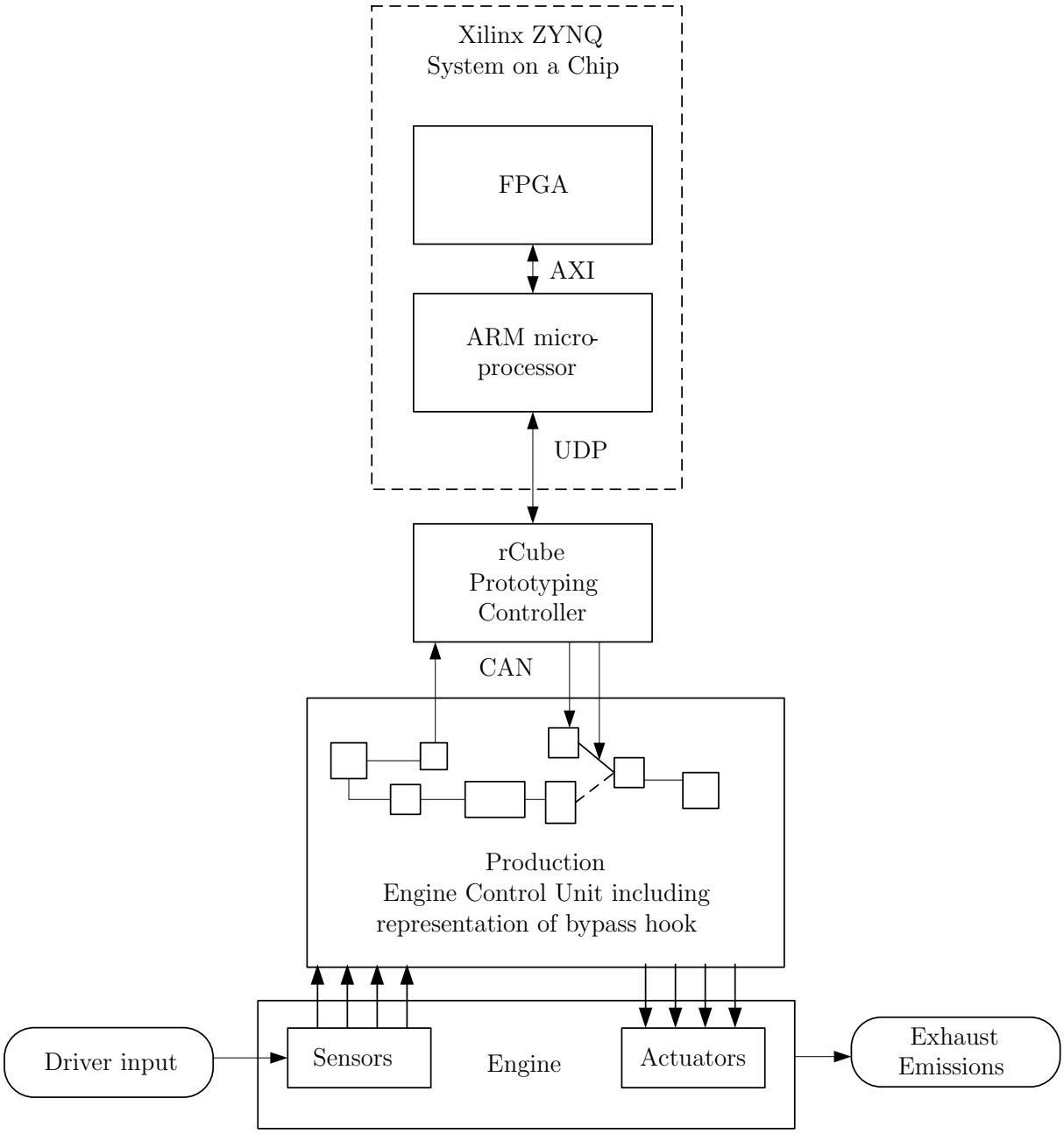

Figure D.2: Controller architecture for vehicle testing.

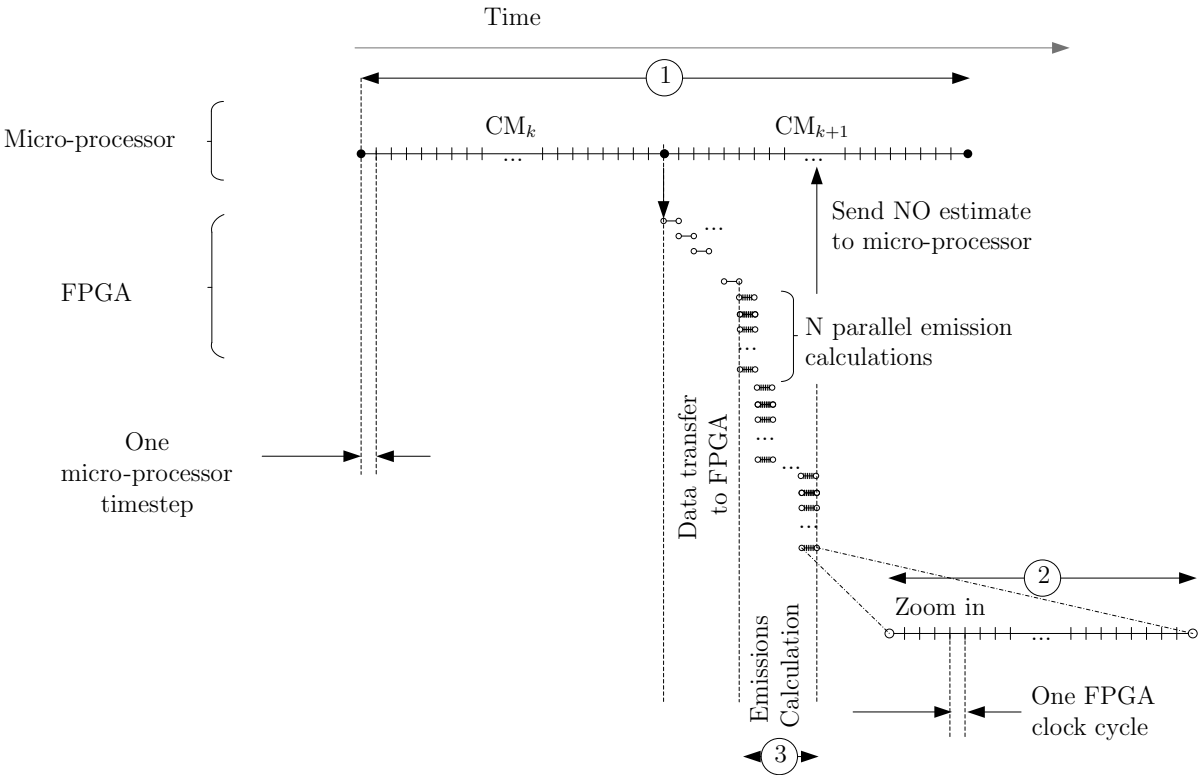

Figure D.3: Data management for combustion model. Timescale (1) is illustrated in Figure D.5, Timescale (2) is illustrated in Figure D.6 (left plot) and Timescale (3) is illustrated in Figure D.6 (right plot).

Data packaging - FPGA

In a similar way, the FPGA also requires data management. Each emissions calculation requires the initial temperature of the mass parcel and the bulk cylinder conditions for all of the cycle. The calculations can only start once the full cycle has been simulated on the micro-processor and transferred to the FPGA (since the bulk cylinder conditions are required for all of the cycle). Once combustion cycle CM_k has completed, the data sent and received by the FPGA over $\frac{N_{cycle}}{N_{parallel}}$ micro-processor cycles. The data is stored in memory in the FPGA and then fed into the NO model which has been set up to calculate $N_{parallel}$ parcels simultaneously over the full combustion cycle. The emissions calculation is carried out every micro-processor timestep.

A co-blocking implementation is used here to ensure synchronisation of the micro-processor and FPGA, this means the micro-processor is halted whilst the FPGA is enabled, and vice versa.

The following example shows the progression of signals through the calculations for a simulation with a constant EGR ratio, engine speed and injection parameters. The simulation starts with the intake manifold pressure at atmospheric pressure.

At $t = 1s$ the manifold pressure is increased to the steady state value, see Figure D.4. The subsequent plots show the key signals within the controller during this simulation.

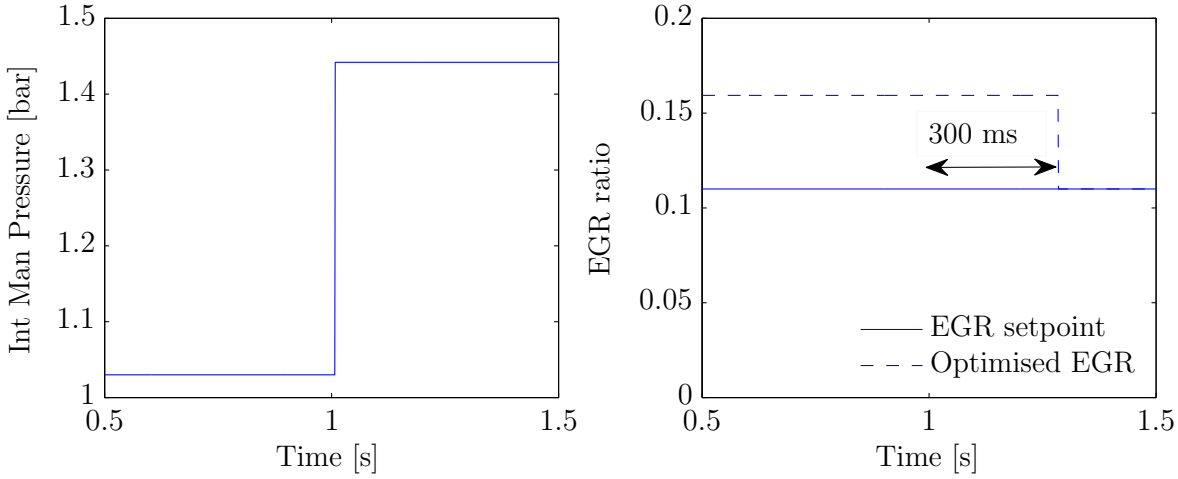

Figure D.4: Simulation input parameters, applied to implemented control algorithm. Right plot shows there is a 300 ms delay between the change in intake manifold pressure and the update to the optimsed EGR.

Figure D.5 shows the outputs from the optimisation supervisor which selects which candidate values of EGR and intake manifold pressure to use. The upper and mid plots show the candidate values of EGR and intake manifold pressure which are simulated by the Combustion Model. The EGR values are selected either side of the set-point. The intake manifold pressure is the actual pressure except for the baseline simulation where the steady state value is used. The lower plot shows the integration of the NO estimates for each candidate simulation. The integration here is the summation of the outputs from the burnt mass parcels.

Figure D.6 shows the signals for the input and outputs to the NO calculation, with different time axis resolution. On the left, a single flame temperature is applied to initialise the NO calculation which also reads in the bulk cylinder temperature over the cycle. In the FPGA, this calculation is carried out 10 times in parallel. The right plot shows the calculations are repeated 10 times with the different flame temperatures, but the same bulk cylinder temperatures. The NO from all 100 calculations is summed to give the NO for a single cycle, see lower plot on Figure D.5.

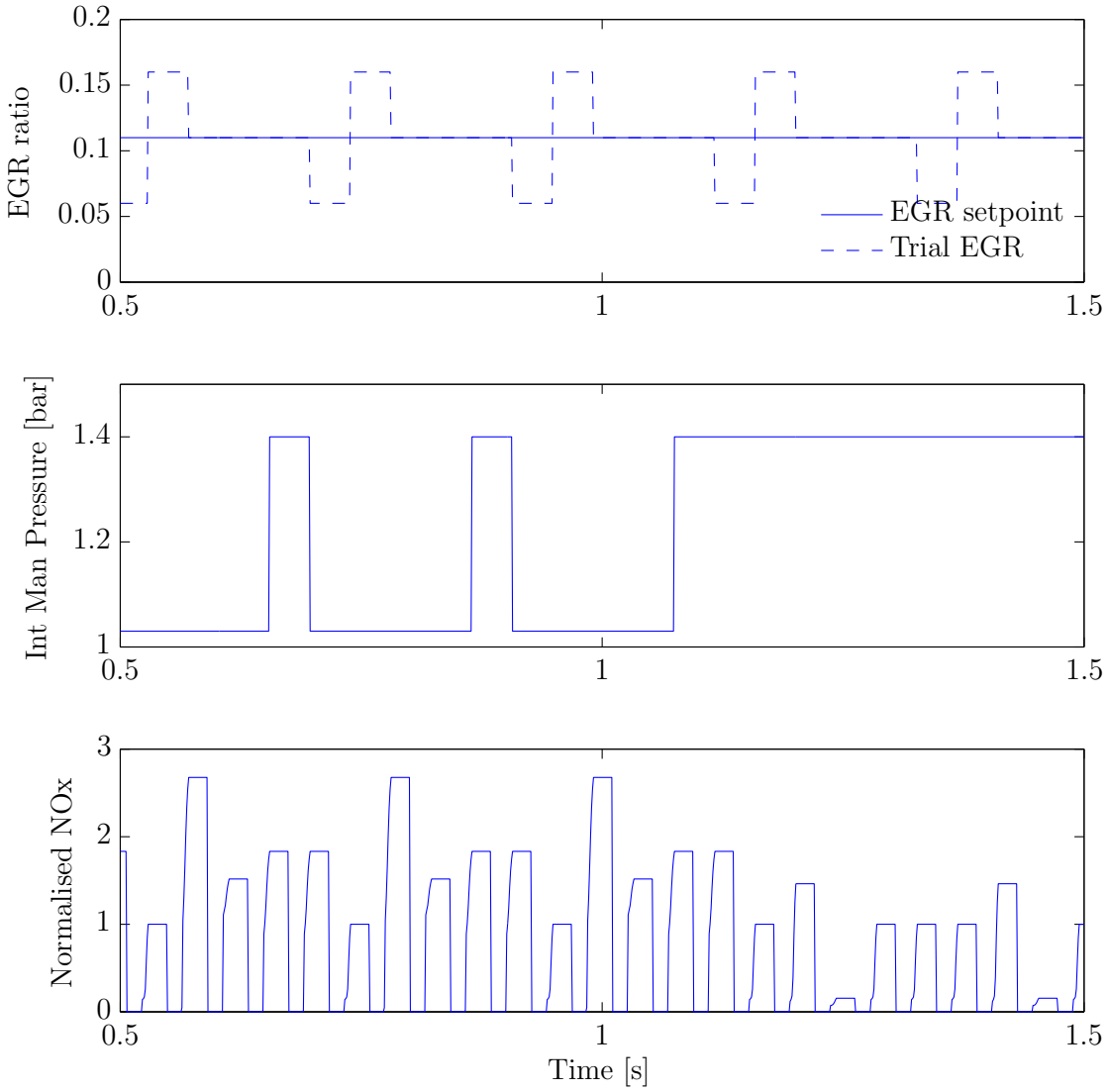

Figure D.5: Data streams within the control unit. The upper and mid plots show the candidate values of EGR and intake manifold pressure which are simulated by the Combustion Model. The EGR values are selected either side of the set-point. The intake manifold pressure is the actual pressure except for the baseline simulation where the steady state value is used. The lower plot shows the integration of the NO estimates for each candidate simulation. See Figure D.3, Timescale (1).

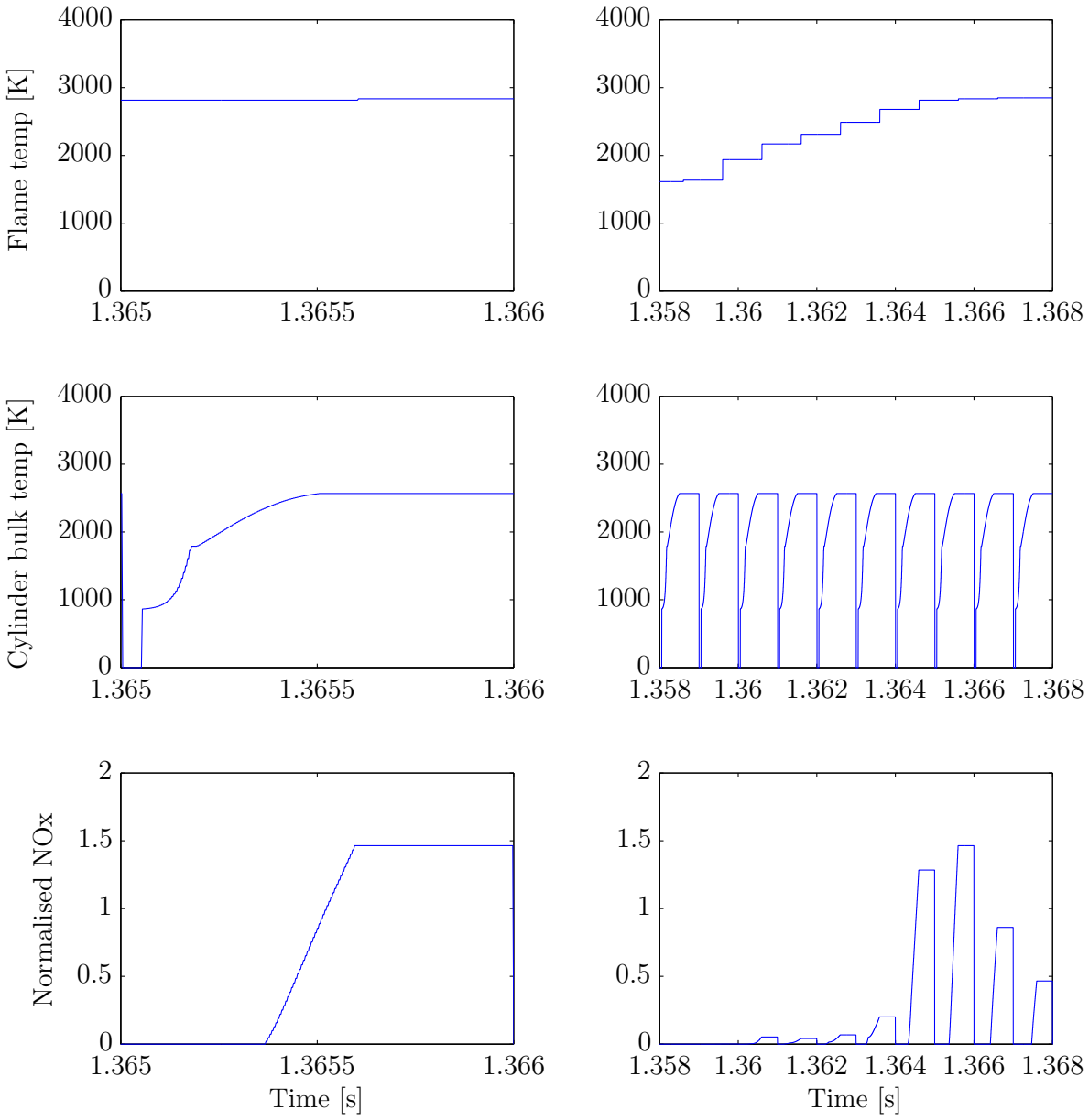

Figure D.6: Inputs and outputs for NO calculations with different time axis resolution: one parcel (left), see Figure D.3, Timescale (2), and one cycle (right), see Figure D.3, Timescale (3).

D.3 Testing

D.3.1 Testbench development

The system communications from the CAN bus through the rCube and into the ARM and FPGA were validated on the testbench. This was carried out using a CANalyser, a tool for generating and receiving CAN messages, and INCA, a tool for calibrating and measuring data within the rCube. The layout of the test bench is shown in Figure D.7.

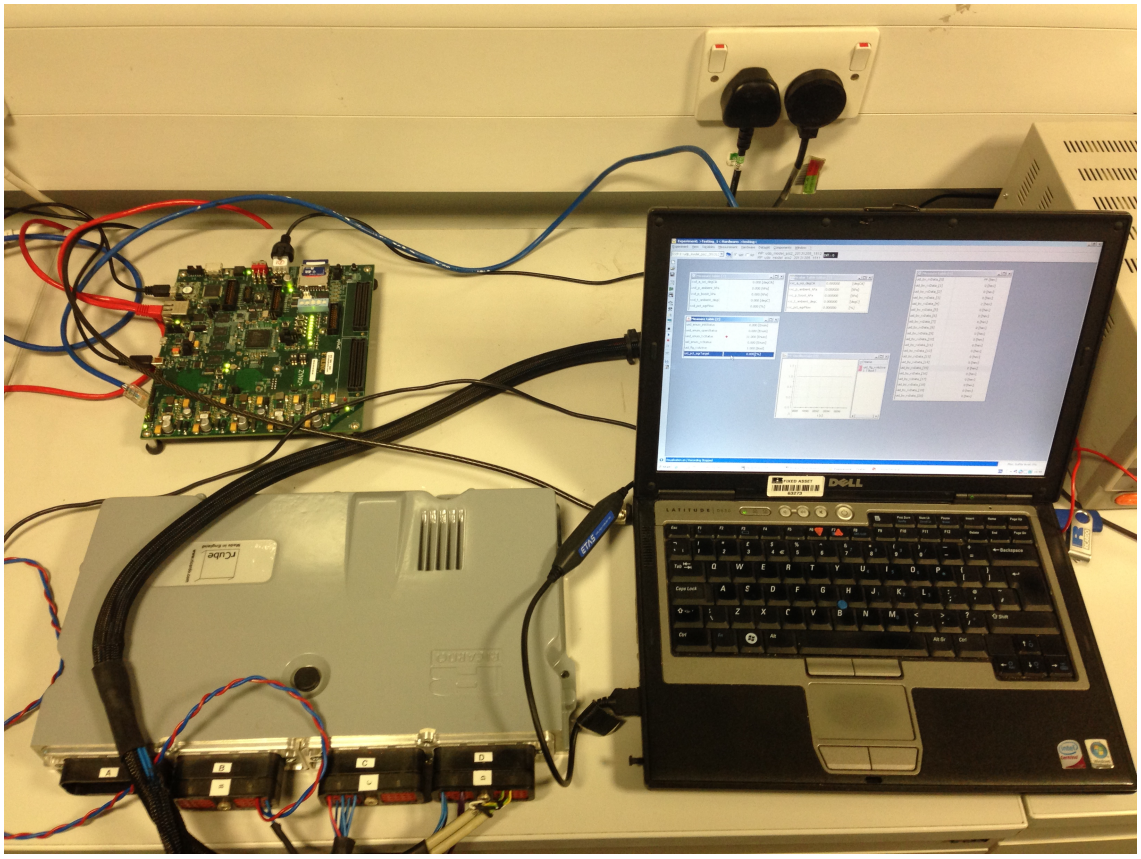

Figure D.7: FPGA and rCube on test bench with laptop running INCA and CANalyser.

D.3.2 Vehicle testing

The rCube and Zynq board were installed in the test vehicle and connected to the engine CAN network. The bypass hooks were enabled and the controller was progressively commissioned, initially with the engine operating at idle to validate communications, moving on to steady state and transient testing on the chassis dynamometer.

Figure D.8: The chassis dynamometer at Ricardo showing the rolling road.

The chassis dynamometer, see Figure D.8, is capable of measuring regulated emissions to all global standards. Analysis capabilities include: multi-bench real-time emissions, EGR level and air fuel ratio, particle size and number characterisation, NO_x and HC speciation. The driver is presented with a speed trace and gear shift points during the tests.

Bibliography

[1] E/ECE/324/Rev.2/Add.100/Rev.3 E/ECE/TRANS/505/Rev.2/Add.100/Rev.3 Regulation no. 101: Uniform provisions concerning the approval of passenger cars powered by an internal combustion engine only, or powered by a hybrid electric power train with regard to the measurement of the emission of carbon dioxide and fuel consumption and/or the measurement of electric energy consumption and electric range, and of categories M1 and N1 vehicles powered by an electric power train only with regard to the measurement of electric energy consumption and electric range.

[2] (EC) 715/2007 on type approval of motor vehicles with respect to emissions from light passenger and commercial vehicles (Euro 5 and Euro 6) and on access to vehicle repair and maintenance information, 20 June 2007.

[3] Road vehicles - controller area network (CAN), 2003.

[4] `http://www.cmclinnovations.com/products/srmsuite`, accessed 7 December 2014.

[5] `http://www.epa.gov/otaq/standards/light-duty/sc06-sftp.htm`, accessed 16 November 2014.

[6] *http://www.xilinx.com/images/product-images/zc702-baseboard-lg.jpg*, accessed 13 July 2014.

[7] *http://www.xilinx.com/products/boards-and-kits/EK-Z7-ZC702-G.htm*, accessed 13 July 2014.

[8] *http://www.xilinx.com/publications/prod_mktg/zynq7000/Zynq-7000-combined-product-table.pdf*, accessed 13 July 2014.

[9] Ricardo quarterly review, Q2, 2004.

[10] K Akihama, Y Takatori, K Inagaki, S Sasaki, and A Dean. Mechanism of the smokeless rich Diesel combustion by reducing temperature. volume SAE 2001-01-0655, 2001.

[11] M Allain, A Kropp, Y Kalish, H Zhang, and C Atkinson. Demonstrating and validating a next generation model- based controller for fuel efficient, low emissions Diesel engines. DEER, 2009.

[12] J Arregle, J Javier Lopez, C Guardiola, and C Monin. *Automotive Model Predictive Control*, chapter On board NOx prediction in Diesel engines: A physical approach, pages 25–36. Springer, 2010.

[13] J Bakosi and J R Ristorcelli. Exploring the beta-pdf in variable-density turbulent mixing. *Journal of Turbulence*, 2010.

[14] C Barba, C Burkhardt, K Boulouchos, and M Bargende. A phenomenological combustion model for heat release rate prediction in high speed DI Diesel engines with common rail injection. volume SAE 2000-01-2933, 2000.

[15] C Barro, P Oberecht, P Lauer, and K Boulouchos. Development of a virtual soot sensor in Diesel engines. volume 75. MTZ, Jun 2014.

[16] J T Betts. *Practical Methods for Optimal Control and Estimation using nonlinear programming.* SIAM, 2010.

[17] R Brekelmans, L Driessen, H Hamers, and D den Hertog. Constrained optimization involving expensive function evaluations: A sequential approach. *European Journal of Operational Research*, 160:121–138, 2005.

[18] K. Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations.* Oxford University Press, 1995.

[19] S Di Cairano, D Yanakiev, A Bemporad, I Kolmanovsky, and D Hrovat. *Automotive Model Predictive Control*, chapter Model Predictive Powertrain Control: An application to idle speed regulation, pages 183 – 194. Springer, 2010.

[20] S K Chen and O Yanakiev. Transient NOx emission reduction using exhaust oxygen concentration based control for a Diesel engine. volume SAE 2005-01-0372, 2005.

[21] F Chmela, M Engelmayer, G Pirker, and A Wimmer. Prediction of turbulence controlled combustion in Diesel engines. Thiesel 2004 Conference on Thermo and Fluid dynamic processes in Diesel engines, 2004.

[22] F Chmela and G Orthaber. Rate of heat release prediction for direct injection Diesel engines based on purely mixing controlled combustion. volume SAE 1999-01-0186, 1999.

[23] F Chmela, G Pirker, B Losonczi, A Wimmer, J M Desantes, and J M Garcia-Oliver. A new burn rate simulation model for improved prediction of multiple injection effects on large Diesel engines. Thiesel 2010 Conference on Thermo and Fluid dynamic processes in Diesel engines, 2010.

[24] L del Re, F Allgower, L Glielmo, C Guardiola, and I Kolmanovsky, editors. *Automotive Model Predictive Control*. Springer, 2010.

[25] A Dulbecco. *Modeling of Diesel HCCI combustion and its impact on pollutant emissions applied to global engine system simulation*. PhD thesis, University of Toulouse, 2010.

[26] A Dulbecco, F A Lafossas, G Mauviot, and T J Poinsot. A new 0D Diesel HCCI combustion model derived from a 3D CFD approach with detailed tabulated chemistry. *Oil and Gas Science and Technology*, 64(2009) No3:259 – 284, 2009.

[27] P Durand, M Gorokhovski, and R Borghi. The PDF equation approach to Diesel spray evaporation computations. volume SAE 960632, 1996.

[28] C Ericson, B Westerberg, M Andersson, and R Egnell. Modelling Diesel engine combustion and NOx formation for model based control and simulation of engine and exhaust aftertreatment systems. volume SAE 2006-01-0687, 2006.

[29] J Lachner et al. *Diesel Engine Management*. Robert Bosch GmbH, 4th edition, 2005.

[30] M F Flynn. Some computer organizations and their effectiveness. *IEEE Trans. Comput.*, 23 (9):948–960, 1972.

[31] P Flynn, R Durrett, G Hunter, A Loye, O Akinyemi, J Dec, and C Westbrook. Diesel combustion: An integrated view combining laser diagnostics, chemical kinetics and empirical validation. volume SAE 1999-01-0509, 1999.

[32] T Fukuma, S Nakayama, Y Harada, and A. Matsunaga. Challenging the vision of calibration free Diesel engine development by means of model based control and automatic optimisation. volume 25 Internationales Wiener Motorensymposium, pages 228 – 243, 2004.

[33] P M Fussey and D J N Limebeer. A control orientated combustion model. volume ECOSM: Workshop on Engine and Powertrain Control, Simulation and Modeling. IFAC, 2012.

[34] P M Fussey and D J N Limebeer. Model predictive combustion control. volume 11th International Symposium on Advanced Vehicle Control. AVEC, 2012.

[35] P M Fussey and D J N Limebeer. Control and constraint of NOx emissions during transient manuvres, based on an engine combustion model. volume SAE 2014-01-1355, 2014.

[36] P M Fussey and D J N Limebeer. Control and constraint of NOx emissions during transient manuvres, based on an engine combustion model. *SAE International Journal of Engines*, July, 2014.

[37] J Gablonsky. *Modifications of the direct algorithm.* PhD thesis, North Carolina State University, 2001.

[38] D Garg, M A Patterson, W W Hager, A V Rao, A V Benson, and G T Huntington. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica*, 46(11):1843–1851, 2010.

[39] J I Ghojel. Review of the development and applications of the Wiebe function: a tribute to the contribution of Ivan Wiebe to engine research. *International Journal of Engine Research*, 11:297–312, 2010.

[40] S Girimaji. Assumed B-PDF model for turbulent mixing: Validation and extension to multiple scalar mixing. *Combustion Sciences and Technology*, 78:177–196, 1991.

[41] S Girimaji. Simulations of diffusion reaction equations with implications to turbulent combustion modeling. Technical Report ICASE report 93-69, NASA, 1993.

[42] J Hadler, F Rudolph, R Dorenkamp, H Stehr, J Hilzendeger, and S Kranzusch. Volkswagen's new 2.0 l TDI engine for the most stringent emissions standards - part 1. *MTZ*, 69, May 2008.

[43] G Hampson and R Reitz. Development of nox and soot models for multidimensional Diesel combustion. *Joint Power Generation Conference, ASME*, 1:187 – 198, 1995.

[44] J B Heywood. *Internal Combustion Engine Fundamentals*. McGraw Hill, 1988.

[45] M Hillion. *Transient combustion control of internal combustion engines*. PhD thesis, Ecole des Mines de Paris, 2009.

[46] H Hiroyasu, T Kadota, and M Arai. Development and use of a spray combustion modeling to predict Diesel engine efficiency and pollutant emissions (part 1 combustion modeling). *JSME*, 26(214):569–575, 1983.

[47] M F Hsieh, M Canova, and J Wang. Model predictive control approach for AFR control during lean NOx trap regenerations. volume SAE 2009-01-0586, 2009.

[48] M Jankovic, M Jankovic, and I Kolmanovsky. Constructive Lyapunov control design for turbocharged Diesel engines. *IEEE transactions on control systems technology*, 8/2:288–299, 2000.

[49] M Jung. *Mean-Value Modelling and Robust Control of the Airpath of a Turbocharged Diesel Engine*. PhD thesis, Cambridge, 2003.

[50] M Karlsson, K Ekholm, P Strandh, R Johansson, and P Tunestal. Multiple-input multiple-output model predictive control of a Diesel engine. IFAC, 2010.

[51] D E Kirk. *Optimal Control Theory: An Introduction*. Dover Publications, 2004.

[52] T Kolda, R Lewis, and V Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385 – 482, 2003.

[53] P Leteinturier. Silicon technologies at the service of powertrain challenges. In *Ricardo Seminar*. IMechE, 2007.

[54] T Lewin. Emissions mission. Automotive Design, March/April 2013.

[55] D J N Limebeer, G Perantoni, and A V Rao. Optimal control of formula one car energy recovery systems. *International Journal of Control*, published online: 23 Apr 2014, 2014.

[56] K V Ling, B F Wu, and J M Maciejowski. Embedded model predictive control (MPC) using a FPGA. IFAC, 2008.

[57] Z Liu, D Ceccarini, A Gangopadhyay, A Baccile, and P Fussey. Patent: US 2010/0223014, Combustion temperature estimation system and method for an engine management system, Filed 2009.

[58] J Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.

[59] B F Magnussen and B H Hjertager. On mathematical modelling of turbulent combustion with special emphasis on soot formation and combustion. 16th International Symposium on Combustion, 1976.

[60] E Mancaruso, L Sequino, and B Valglieco. First and second generation bioDiesels spray characterisation in a Diesel engine. *Fuel*, 90:2870 – 2883, 2011.

[61] D Q Mayne, J B Rawlings, C V Rao, and P O M Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[62] J P Meijaard, J M Papadopoulos, A Ruina, and A L Schwab. Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review. *Proc. R. Soc. A*, 463, 2007.

[63] M Mueller, Hans-Christian Schwannecke, and Wolfgang Fengler. *Technology and Engineering Applications of Simulink*, chapter 6: From Control Design to FPGA Implementation. Intech, 2012.

[64] J Naber and D L Siebers. Effects of gas density and vaporisation on penetration and dispersion of Diesel sprays. volume SAE 960034, 1996.

[65] J Nagle and R F Strickland-Constable. Oxidation of carbon between 1000 - 2000 ^{o}C. In *Proceedings of the Fifth Carbon Conference*, volume 1, page 154. Pergamon Press, 1962.

[66] S Nakayama, T Fukuma, A Matsunaga, T Miyake, and T Wakimoto. A new dynamic combustion control method based on charge oxygen concentration for Diesel engines. volume SAE 2003-01-3181. Toyota Motor Corporation, 2003.

[67] S Nakayama, T Ibuki, H Hosaki, and H Tominaga. An application of model based combustion control to transient cycle-by-cycle Diesel combustion. volume SAE 2008-01-1311, 2008.

[68] S Nakayama, M Ueda, K Inagaki, K Nakakita, and T Fukuma. Toyotas Diesel transient simulation with an innovative combustion model adaptable to the latest combustion concept. In *Haus der Technik Conference on Simulation of In-engine Processes and Turbocharging*, 2007.

[69] D R Nightingale. A fundamental investigation into the problem of NO formation in Diesel engines. volume SAE 750848, 1975.

[70] M A Patterson and A V Rao. Gpops-II. a general-purpose matlab software for solving multiple-phase optimal control problems. University of Florida, May 2014.

[71] C Pera, O Colin, and S Jay. Development of a FPI detailed chemistry tabulation methodology for internal combustion engines. *Oil and Gas Science and Technology*, 64(3):243–258, 2009.

[72] G Pirker, F Chmela, and A Wimmer. ROHR simulation for di Diesel engines based on sequential combustion mechanisms. volume SAE 2006-01-0654, 2006.

[73] S B Pope. PDF methods for turbulent reactive flows. *Prog Energy Combust. Sci.*, 11:119 – 1921, 1985.

[74] M Potter and R. Durrett. High-efficiency clean combustion design for compression ignition engines. DEER, 2006.

[75] M J D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, pages 287–336, 1998.

[76] H Ricardo and J G G Hempson. *The high speed internal combustion engine*. Blackie and Sons Ltd, 1968.

[77] J A Rossiter. *Model Based Predictive Control, A Practical Approach*. CRC press, 2004.

[78] J Seabrook, B Rogers, G Farrow, J Patterson, and S Edwards. Applications of advanced modelling methods in engine development. I Mech E, 2002.

[79] Z Sika, M Valasek, M Florian, J Macek, and M Polasek. Multilevel predictive models of IC engines for model predictive control implementation. volume SAE 2008-01-0209, 2008.

[80] S Sirowy and A Forin. Wheres the beef? why FPGAs are so fast. *Microsoft Technical Report, MSR-TR-2008-130*, 2008.

[81] G Stewart, F Borrelli, J Pekar, D Germann, D Pachner, and D Kihas. *Automotive Model Predictive Control*, chapter 14 Toward a Systematic Design for Turbocharged Engine Control, pages 211 – 230. Springer, 2010.

[82] E Stolting, D Eckardt, E Neumann, and L Bilke. Mathematical on-board engine-out soot emission model. volume 74. MTZ, Oct 2013.

[83] H Su. *Stochastic Reactor Models for simulating direct injection homogeneous charge compression ignition engines*. PhD thesis, Cambridge University, 2010.

[84] D A Teegarden. FPGA design and verification in mechatronic applications. Web, October 2009.

[85] V Tourczon. *Multi-directional search: a direct search algorithm for parallel machines*. PhD thesis, Rice University, 1989.

[86] A J Truscott and B C Porter. Simulation of model-based control algorithms for a variable geometry turbocharged diesel engine. *MTZ*, 1997.

[87] F Tschanz, A Amstutz, C Onder, and L Guzzella. Feedback control of particulate matter and nitrogen oxide emissions in Diesel engines. *Control Engineering Practice*, 21:1809 –1820, 2013.

[88] A Vahidi, A Stefanopoulou, and H Peng. Model predictive control for starvation prevention in a hybrid fuel cell system. American Control Conference, 2004.

[89] M J van Nieuwstadt, I. Kolmanovsky., and P E Moraal. Coordinated EGT - VGT control for Diesel engines: and experimental comparison. volume SAE 2000-01-0266, 2000.

[90] Y Wang and S Boyd. Fast model predictive control using online optimisation. *IEEE Control systems technology Journal*, 2010.

[91] J Warnatz, U Maas, and R W Dribble. *Combustion. Physical and chemical fundamentals, modelling and simulation, experiments, pollutant formation.* Springer, 1996.

[92] N Watson, A D Pilley, and M Marzouk. A combustion correlation for Diesel engine simulation. volume SAE 800029, 1980.

[93] Winward, Deng, and Stobart. Innovations in experimental techniques for the development of fuel path control of Diesel engines. 2010.

[94] T Yoshikawa and R Reitz. Development of an improved NOx reaction mechanism for low temperature Diesel combustion modeling. volume SAE 2008-01-2413, 2008.