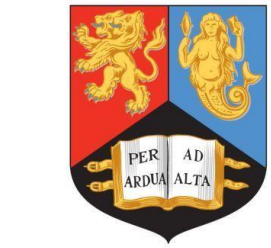


UNIVERSITY OF BIRMINGHAM

ONLINE DYNAMIC OPTIMISATION FOR THE ENGINE MANAGEMENT SYSTEM USING ARTIFICIAL INTELLIGENCE METHODS

BY

ZIYANG LI

A thesis submitted to

The College of Engineering and Physical Sciences of

The University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering

School of Engineering

College of Engineering and Physical Sciences

University of Birmingham

August 2019

ABSTRACT

As the degrees of freedom for engine operations increases, the traditional calibration approaches for the lookup-table-based engine management system become time-consuming and labour-intensive. Besides, the lookup-table-based control method is not robust to system uncertainty and high nonlinearity. Accordingly, artificial intelligence methods are introduced to raise the automation level of the engine management system.

This thesis presents an intelligent non-model-based multi-objective calibration approach using meta-heuristic algorithms. It relies on neither the engine model nor massive experimental data. The experiment studies show that it can automatically locate the optimum engine variable settings to provide the minimum fuel consumption and PM emissions simultaneously with high efficiency.

A proportional-integral-like fuzzy knowledge based controller with the self-adaptive capability and high robustness is developed to regulate the air/fuel ratio for engines. It can reduce the effort to be spent in tuning controller parameters, and improve the system transient response, compared to the conventional lookup-table-based proportional-integral controller.

Subsequently, the controller parameters can be further improved using the intelligent non-model-based transient calibration approach. A better air/fuel ratio transient response is thus achieved.

All the calibration and control strategies have been validated on a real gasoline direct injection engine via the rapid control prototyping platform.

ACKNOWLEDGEMENT

First of all, I would like to express my deepest gratitude to my supervisor, Professor Hongming Xu, for his trust and constant guidance throughout my entire Ph.D. career. Then, I would like to thank my associate supervisors, Professor Miroslaw Wyszynski and Dr. Oluremi Olatunbosun, for their encouragement and help. I would also like to extend my sincere appreciation to Professor Akabar Ghafourian for his valuable comments on my research works. Moreover, I am so grateful for the technical support provided by Stephen Stacey and the engine calibration team at Jaguar Land Rover. They helped me figure out many technical problems.

Special thanks for the support from Engineering and Physical Sciences Research Council (EPSRC), Jaguar Land Rover, ETAS, Shell Global Solutions, Bosch, Rheonik and Cambustion. Without their experimental equipment, technical services, and financial support, I was unable to complete my experiments and Ph.D. studies. I would also like to give great thanks to the engine calibration engineers in Ford and Changan UK R&D Centre for their useful advice on my research.

Many thanks to the laboratory technicians, Carl Hingley, Peter Thornton, Jack Garrod, and Lee Gauntlett, for their maintenance and support with the experimental facilities. In addition, I appreciate the working environment offered by our college. I would also like to acknowledge the guidance and help in thesis writing from Mrs. Janet Hingley. Thanks for her proofreading service for this thesis.

I am greatly indebted to the postdoctoral research fellows in my research group, Dr. He Ma, Dr. Guoxiang Lu, and Dr. Haoye Liu, for their generous guidance and help. I owe my hearty thanks to my colleagues in my research group, Yunfan Zhang, Quan Zhou, Ji Li,

Bin Shuai, Yinglong He, Aawishkar Dharmadhikari, Carlo Coratella, Lewis Parry, Scott Cash, Tawfik Badawy, Sohiel Zeraati Rezaei, and Olalere Rafiu Kayode. Thanks for their concern and help with my study and life.

Finally, I would like to express my heartfelt gratitude to my beloved parents. Thanks for their unconditional support and encouragement during my tough times.

LIST OF PUBLICATIONS

Journal

1. **Ziyang Li**, Ji Li, Quan Zhou, Yunfan Zhang, Hongming Xu*. Intelligent air/fuel ratio control strategy with a PI-like fuzzy knowledge–based controller for gasoline direct injection engines. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 233(8), 2019, 2161–2173. https://doi.org/10.1177/0954407018779180

2. **Ziyang Li**, Quan Zhou, Yunfan Zhang, Ji Li, Hongming Xu*. Enhanced intelligent proportional-integral-like fuzzy knowledge–based controller using chaos-enhanced accelerated particle swarm optimization algorithm for transient calibration of air–fuel ratio control system. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. (Available online since July 2019) https://doi.org/10.1177/0954407019862079

3. He Ma, **Ziyang Li**, Mohamad Tayarani, Guoxiang Lu, Hongming Xu*, Xin Yao. Model-based computational intelligence multi-objective optimization for gasoline direct injection engine calibration. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 233(6), 2019, 1391–1402. https://doi.org/10.1177/0954407018776743

4. He Ma, **Ziyang Li**, Mohamad Tayarani, Guoxiang Lu, Hongming Xu*, Xin Yao. Computational Intelligence Nonmodel-Based Calibration Approach for Internal Combustion Engines. ASME. Journal of Dynamic Systems, Measurement, and Control 140(4), 2017, 041002-041002-9. https://doi.org/10.1115/1.4037835

5. Ji Li, **Ziyang Li**, Quan Zhou, Yunfan Zhang, Hongming Xu*. Improved scheme of membership function optimisation for fuzzy air-fuel ratio control of GDI engines. IET Intelligent Transport Systems 13(1), 2019, 209-217. https://doi.org/10.1049/iet-its.2018.5013

6. Yunfan Zhang, Quan Zhou, **Ziyang Li**, Ji Li, Hongming Xu*. Intelligent transient calibration of a dual-loop EGR diesel engine using chaos-enhanced accelerated particle swarm optimization algorithm. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 233(7), 2019, 1698–1711. https://doi.org/10.1177/0954407018776745

7. Quan Zhou, Yunfan Zhang, **Ziyang Li**, Ji Li, Hongming Xu*, Oluremi Olatunbosun. Cyber-Physical Energy-Saving Control for Hybrid Aircraft-Towing Tractor Based on Online Swarm Intelligent Programming. IEEE Transactions on Industrial Informatics 14(9), 2018, 4149-4158. https://doi.org/10.1109/TII.2017.2781230

8. Yunfan Zhang, Guoxiang Lu, Hongming Xu*, **Ziyang Li**. Tuneable model predictive control of a turbocharged diesel engine with dual loop exhaust gas recirculation. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 232(8), 2018, 1105–1120. https://doi.org/10.1177/0954407017726944

9. Ji Li, Quan Zhou, Yinglong He, Bin Shuai, **Ziyang Li**, Huw Williams, Hongming Xu*. Dual-loop online intelligent programming for driver-oriented predict energy management of plug-in hybrid electric vehicles. Applied Energy 253, 2019, 113617. https://doi.org/10.1016/j.apenergy.2019.113617

10. **Ziyang Li**, Quan Zhou, Yunfan Zhang, Hongming Xu*. Intelligent non-model-based calibration approach for internal combustion engines using the chaos-enhanced accelerated particle swarm optimization algorithm. (Prepare to submit)

Conference Paper

1. Guoxiang Lu, Yunfan Zhang, Hongming Xu*, **Ziyang Li**, Ben Neaves. Multiple Model Predictive Control for Diesel Engines with Dual Loop Exhaust Gas Recirculation. 3rd Biennial International Conference on Powertrain Modelling and Control, PMC 2016, Loughborough, UK.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AFR	Air/Fuel Ratio
ANN	Artificial Neural Network
APC	Advanced Propulsion Centre
APSO	Accelerated Particle Swarm Optimisation
aTDC	After Top Dead Centre
BMEP	Brake Mean Effective Pressure
BSEC	Brake Specific Energy Consumption
BSFC	Brake Specific Fuel Consumption
bTDC	Before Top Dead Centre
BTHE	Brake Thermal Efficiency
CAD	Crank Angle Degree
CAN	Control Area Network
CAPSO	Chaos-enhanced Accelerated Particle Swarm Optimisation
CFD	Computational Fluid Dynamics
CO	Carbon Monoxide
DoE	Design of Experiment
DRNN	Diagonal Recurrent Neural Network
ECU	Engine Control Unit
EGO	Exhaust Gas Oxygen
EGR	Exhaust Gas Recirculation
EIA	Energy Information Administration
EMS	Engine Management System
EVC	Exhaust Valve Closing
FKBC	Fuzzy Knowledge Based Controller
GA	Genetic Algorithm
GDI	Gasoline Direct Injection
GTE	Gas Turbine Engine
HC	Hydrocarbons
HCCI	Homogenous Charge Compression Ignition
I/O	Input/output
IAE	Integral of the Absolute Error
IAU	Integral of the Absolute Output Signal

ICE	Internal Combustion Engine
IMEP	Indicated Mean Effective Pressure
ISFC	Indicated Specific Fuel Consumption
IVO	Intake Valve Opening
LDV	Light Duty Vehicle
LIN	Local Interconnect Network
MAF	Mass Air Flow
MIMO	Multiple-Input and Multiple-Output
MOP	Multi-objective Optimisation Problem
MPC	Model Predictive Control
NMPC	Non-linear Model Predictive Control
NNMPC	Neural Network Model Predictive Control
NOx	Oxides of Nitrogen
OBD	On-Board Diagnostic
OECD	Organisation for Economic Co-operation and Development
OEM	Original Equipment Manufacturer
OSELM	Online Sequential Extreme Learning Machine
PCCI	Premixed Charge Compression Ignition
PFI	Port Fuel Injection
PID	Proportional Integral Derivative
PM	Particulate Matter
PMm	Particulate Matter Emission Mass
PMn	Particulate Matter Emission Number
PSO	Particle Swarm Optimisation
RAM	Random Access Memory
RCP	Rapid Control Prototyping
RDE	Real Driving Emissions
RNN	Recurrent Neural Network
ROM	Read-only Memory
SI	Spark Ignition
SPEA	Strength Pareto Evolutionary Algorithm
SQP	Sequential Quadratic Programming
TCP/IP	Transmission Control Protocol / Internet Protocol
TTL	Transistor-Transistor Logic

TWC	Three-way Catalytic Converter
VCP	Variable Compression Ratio
VEGA	Vector Evaluated Genetic Algorithms
VVT	Variable Valve Timing

LIST OF SYMBOLS

Symbol	Description	Unit
$\hat{\dot{m}}_o$	The estimation of the air charge mass rate	kg/h
$\dot{m}_a$	The intake air mass flow	kg/h
$\dot{m}_{fb}$	The feedback regulation fuel injection signal	kg/h
$\dot{m}_{fc}$	The fuel mass flow entering the cylinder	kg/h
$\dot{m}_{ff}$	The feedforward reference fuel injection signal	kg/h
$\dot{m}_o$	The air mass flow at the cylinder inlet port	kg/h
$\dot{m}_{th}$	The air mass flow through the throttle	kg/h
A_0	The initial external archive	
A_t	The external archive at t^{th} iteration	
$BSFC_{max}$	The maximum brake specific fuel consumption	g/kwh
$BSFC_{min}$	The minimum brake specific fuel consumption	g/kwh
$G_{(t)}^{i}$	The global best location in the swarm until t^{th} iteration	
H_{ref}	The reference value	
K_{cr}	The proportional gain when a steady-state oscillation occurs	
K_d	The derivative gain of the PID controller	
K_i	The integral gain of the PID controller	
K_i^c	The continuous integral gain	
K_i^d	The discrete integral gain	
K_p	The proportional gain of the PID controller	
K_p^c	The continuous proportional gain	
K_p^d	The discrete proportional gain	
K_u	The fuzzy control scaling factor	
$\overline{N}$	The external archive size	
P_0	The initial population	
PMm_{max}	The maximum particulate matter emission mass	µg/cc
PMm_{min}	The minimum particulate matter emission mass	µg/cc

PMn_{max}	The maximum particulate matter emission number	n/cc
PMn_{min}	The minimum particulate matter emission number	n/cc
P_{cr}	The period of the sustained oscillation	s
P_t	The population at t^{th} iteration	
T_d	The derivative time constant	s
T_e	The torque	Nm
T_i	The integral time constant	s
T_m	The intake manifold temperature	℃
V_C	The cylinder volume	m^3
V_m	The intake manifold volume	m^3
e_p	The error signal	
e_v	The rate of the change of error signal	
$p_{(t)}^i$	The ith particle's personal best location at t^{th} iteration	
p_m	The manifold pressure	Pa
u_{PI}	The output signal of the PI controller	
u_{PID}	The output signal of the PID controller	
u_{fb}	The feedback fuelling signal	
$v_{(t)}^i$	The ith particle's velocity at t^{th} iteration	
$x_{(t)}^i$	The ith particle's location at t^{th} iteration	
α_0	The initial randomness parameter	
λ_{ref}	The target λ value	
σ_i^k	The spatial distance (Euclidean distance) between the individual i and its k^{th} neighbouring individual	
Δu_{PI}	The incremental control output of the PI controller	
$BSFC$	Brake specific fuel consumption	g/kwh
$D(i)$	The density value of individual i	
E	The error signal between the reference and the process output	
EVC	Exhaust valves closing timing	Crank Angle Degree
$F(i)$	The fitness value of individual i	

H	The actually measured signal	
IVO	Intake valves opening timing	Crank Angle Degree
$Injection\_t$	Injection timing	Crank Angle Degree
N	The population size	
PMm	Particulate matter emission mass	μg/cc
PMn	Particulate matter emission number	n/cc
Q	The weighting factor on the control signal	
R	The ideal gas constant	
R	The weighting factor on the error signal	
$R(i)$	The raw fitness of individual i	
$Rail\_p$	Rail pressure	MPa
$S(i)$	The strength value of individual i	
$Spark\_t$	Spark timing	Crank Angle Degree
T	The sampling time	s
T	The maximum number of the generations	
W	The correlation factor	
X	The input signal of the PI-like FKBC	
f	The non-linear function between the variable settings and the engine responses	
$floor$	The function to acquire the integer part	
k	The discrete time	
$mod1$	The function to acquire the decimal part	
r	The random number	
s	The independent variable of the Laplace transform	
t	The iterative loop	
u	The real-time control signal	
w	The inertia weight	
$x(t)$	The solution for t^{th} iteration	
α	The scaling factors for the cognitive behaviour	
α	The randomness parameter	
αr	The randomization component	

β	The scaling factors for the social behaviour	
β	The attraction parameter	
γ	The control parameter of the randomization component	
ε	The random action	
η	The volumetric efficiency	
θ	The throttle position	
λ	The air to fuel equivalence ratio	
μ	The membership value of the PI-like FKBC	
ω	The engine speed	rpm

CHAPTER 1 INTRODUCTION

Since 1859, when French engineer J. J. Étienne Lenoir made the first two-stroke internal combustion engine (ICE), and 1876, when German engineer Nikolaus A. Otto built a four stroke spark-ignition engine which is similar to what we use today (Heywood, 1988), ICEs have been retaining the most common power source for powertrains during the past century. Even though the sustainable transport modes like hybrid vehicle, electric vehicle and fuel cell vehicle are developed rapidly nowadays, the power storage, manufacturing cost and universal performance are still limiting their wide utilization. Meanwhile, researchers from different fields including mechanics, chemistry, electronics and materials have made significant contribution to ICEs. Hence, the ICEs are allowed to still remain the leading role as the power supplier for both conventional and hybrid powertrains (García et al., 2019; García and Monsalve-Serrano, 2019; Huang et al., 2019).

1.1 Background and Motivation

According to the latest International Energy Outlook 2017 announced by the U.S. Energy Information Administration, the world energy consumption will increase 28% from 2015 to 2040. Liquid fuels, mostly petroleum-based fuels, are still remaining the largest energy supplier. The outlook is presented in Figure 1.1.

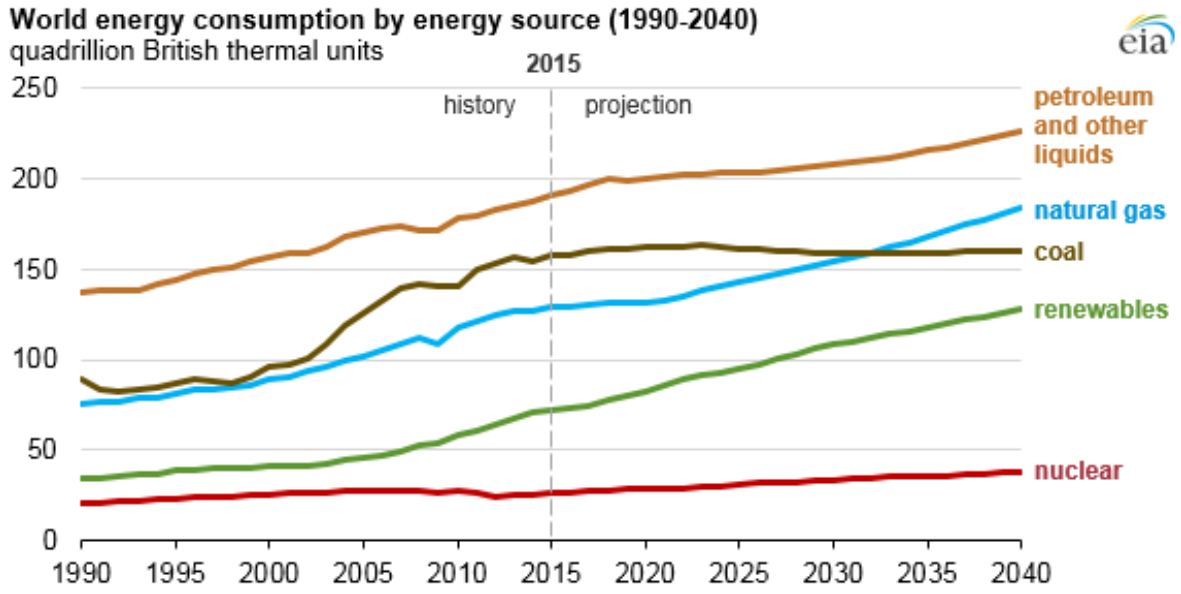


Figure 1.1 Outlook of the world energy consumption by energy source (EIA projects 28% increase in world energy use by 2040 - Today in Energy - U.S. Energy Information Administration (EIA), 2017)

The latest World Oil Outlook 2018 announced by the Organization of the Petroleum Exporting Countries indicates that, the transportation, especially the road transportation, accounted for 57% of the global oil product consumption in 2017, with a demand of 55.8 mb/d. The latest BP Energy Outlook 2019 edition also indicates that, the global transportation is still dominated by oil. As Figure 1.2 shows, the oil will share 85% of the transportation fuel consumption by 2040. Meanwhile, almost all the transportation (>99.9%) is powered by internal combustion engines (ICE) – land and marine transport by reciprocating ICEs and air transport by jet engines (Kalghatgi, 2018). There are around 1.2 billion passenger cars and 380 million commercial vehicles in the world and these numbers will keep rising, almost entirely from the non-OECD countries like China and India (Kalghatgi, 2018; 2018 Outlook for Energy).

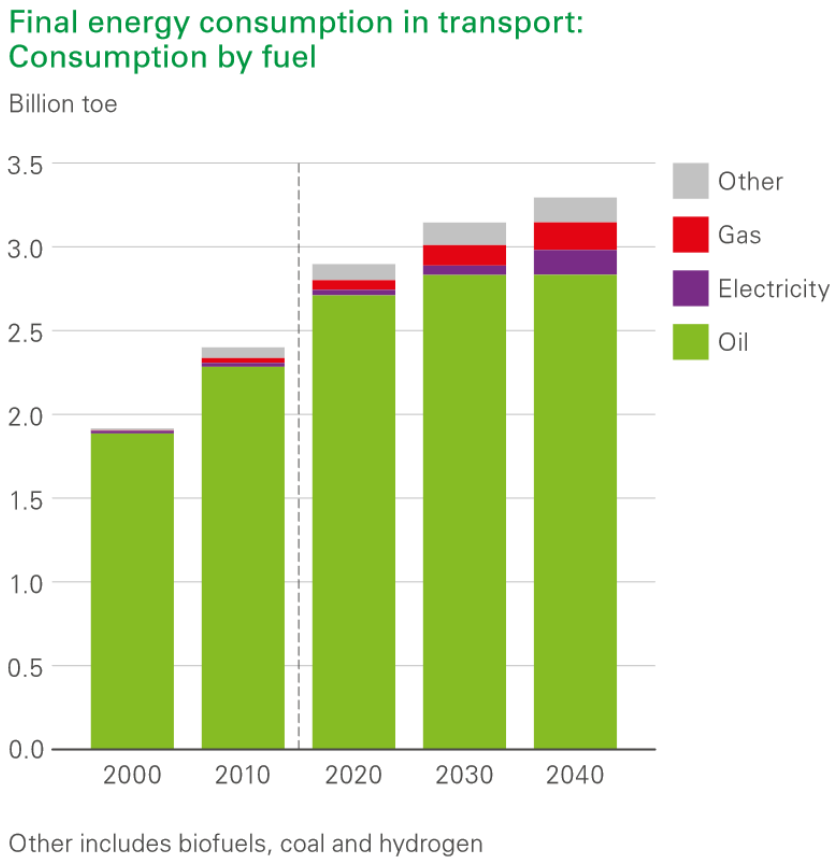
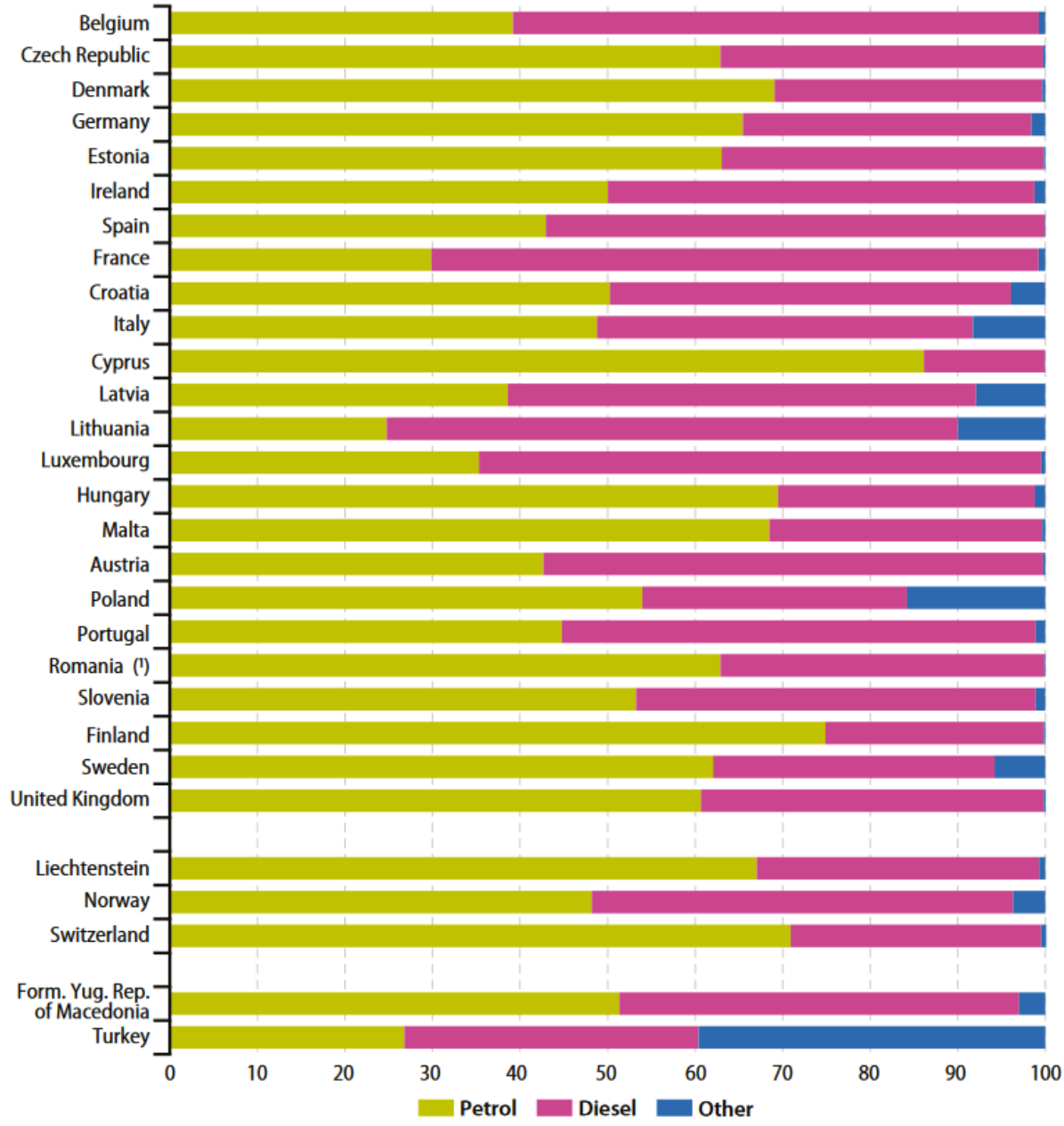


Figure 1.2 Final energy consumption in transport: consumption by fuel (BP Energy Outlook: 2019 edition)

From the Eurostat's latest energy, transport and environment indicators 2018 edition, more than half of the passenger cars were petrol-driven in 15 of 24 European Union member states in 2016 as the statistical data shows in Figure 1.3. In UK, the petrol still made up the majority of the passenger cars' fuel consumption. The "Other" category indicates the type of energy other than petrol and diesel, including the liquefied petroleum gas, natural gas, electricity, nuclear, biofuels and other alternative energy sources. Poland and Turkey are two of the largest consumers in the world of liquefied petroleum gas as a road fuel.



Note: Data not available for Bulgaria, Greece, Netherlands and Slovakia.
(1) 2015 data instead of 2016.

Source: Eurostat (online data code: road_eqs_carpda)

Figure 1.3 Passenger cars by fuel type, 2016 (%) (Energy, transport and environment indicators 2018 edition – Eurostat, 2018)

There is much interest in electric vehicles in recent years and many governments have announced the desire to eventually ban the cars powered by ICEs, although it is not clear if the intention is to ban all ICEs or ban the vehicles powered by ICEs only without any electrical assistance. After all, ICEs are still employed in the hybrid powertrain systems (García and Monsalve-Serrano, 2019; Huang et al., 2019; García et al., 2019).

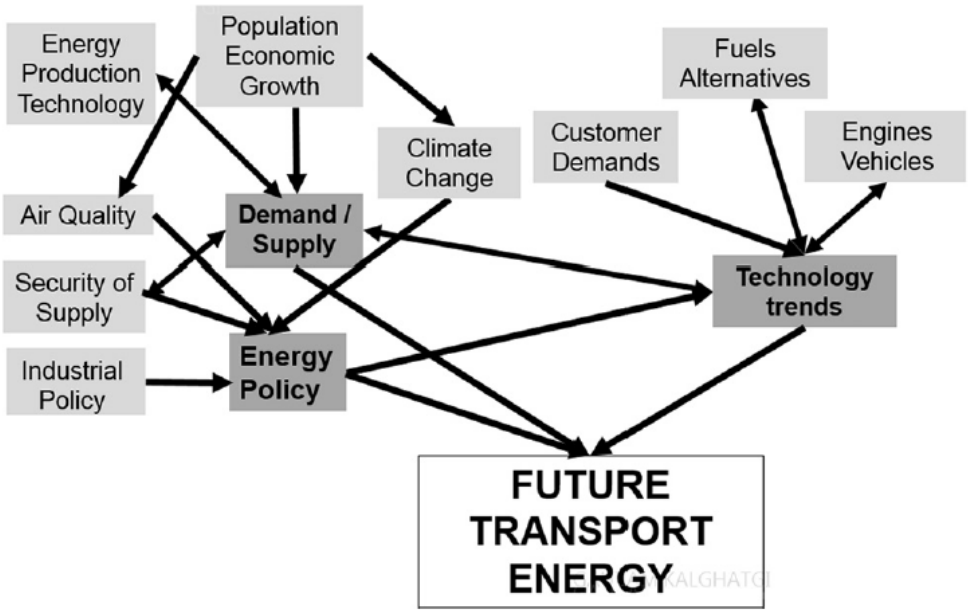


Figure 1.4 Evolution of the transport energy determined by the complex interplay among various drivers (Kalghatgi, 2018)

The evolution of the transport energy depends on the complex interplay among various drivers in Figure 1.4. All the alternative power sources are developed from a very basic level and their growth speed is limited by a variety of factors. Therefore, reliable predictions show that even by 2040 around 90% of the transportation will still be powered by petroleum-driven combustion engines (2018 Outlook for Energy: A View to 2040; World Oil Outlook 2018; Annual Energy Outlook 2019 with projections to 2050).

All in all, although these alternative and renewable energy sources are predicted to fulfil the energy demands in the future, it is expected that fossil fuels are still maintaining the leading roles among these power suppliers, especially for the road transportation, due to the security, storage, budget, technology and other issues. In addition, whether as a fuel-powered vehicle or as a hybrid-powered vehicle, it still requires an ICE for propulsion, which will consume fuel, result in emissions, take time and labour for calibration. Therefore, the engine management system can still be expected to advance considerably in the upcoming decades (World Oil Outlook 2018).

1.2 **Development of Gasoline Engines**

Gasoline engines are widely used for passenger cars because of their advantages as low emission, simple structure, light weight, low cost of construction, smooth operation as well as high rotational speed. During the past decades, the researchers and engineers from different fields have made considerable contribution to the modern gasoline engines. The development about the modelling and control of spark ignition (SI) engines is shown in Figure 1.5:

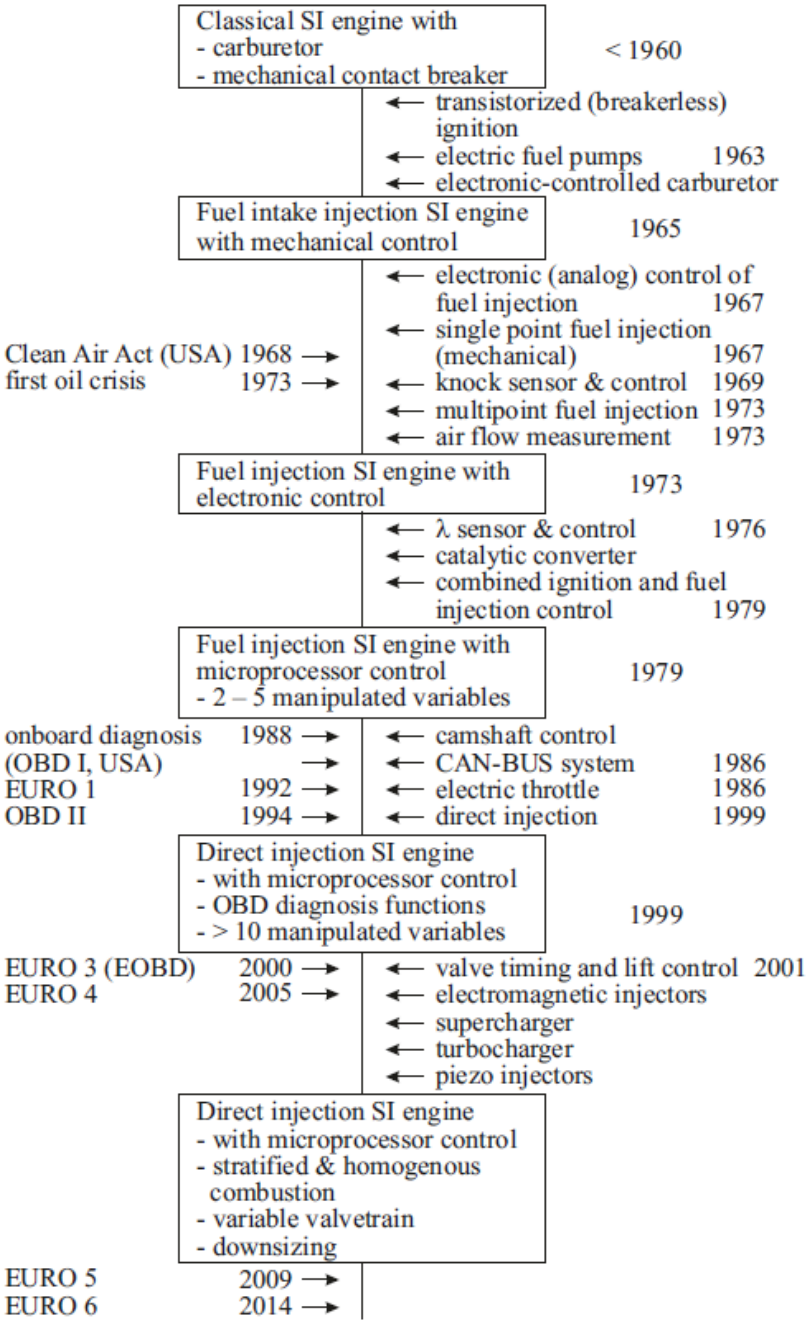


Figure 1.5 Development about the modelling and control of SI engines (Guzzella & Onder, 2010)

Gasoline direct injection (GDI) engines entered the car market in the late 1990s and have been widely employed for passenger cars during the past decades due to the benefits as stoichiometric fuel mixture, high thermal efficiency and potential reductions of HCs and

NOx emissions (Shuai et al., 2018). It is predicted that the global proportion of GDI engines will overtake the proportion of port fuel injection (PFI) engines by 2020 in order to meet the increasingly stringent regulations (Shuai et al., 2018; Xu et al., 2015). Obviously, GDI engines still have great potential to achieve additional improvements in respect of the thermal efficiency and pollutant emissions. Thus, they will be significantly evolved in the next few years (Shuai et al., 2018).

Based on the summary from 2018 World Oil Outlook, the efficiency of gasoline engines for passenger cars has been increased from below 20% in previous decades to around 35% on average today. Further technological advances should retain the gasoline engines as the most economical and dominant transportation technology in the long-term, especially in developing countries (2018 World Oil Outlook). Besides the improvement of fuel efficiency, the cleaning of exhaust gas has also be significantly improved by advanced techniques. Nowadays, the pollution of exhaust gas (mainly NOx and unburnt hydrocarbons) from a gasoline engine is almost zero (2018 World Oil Outlook).

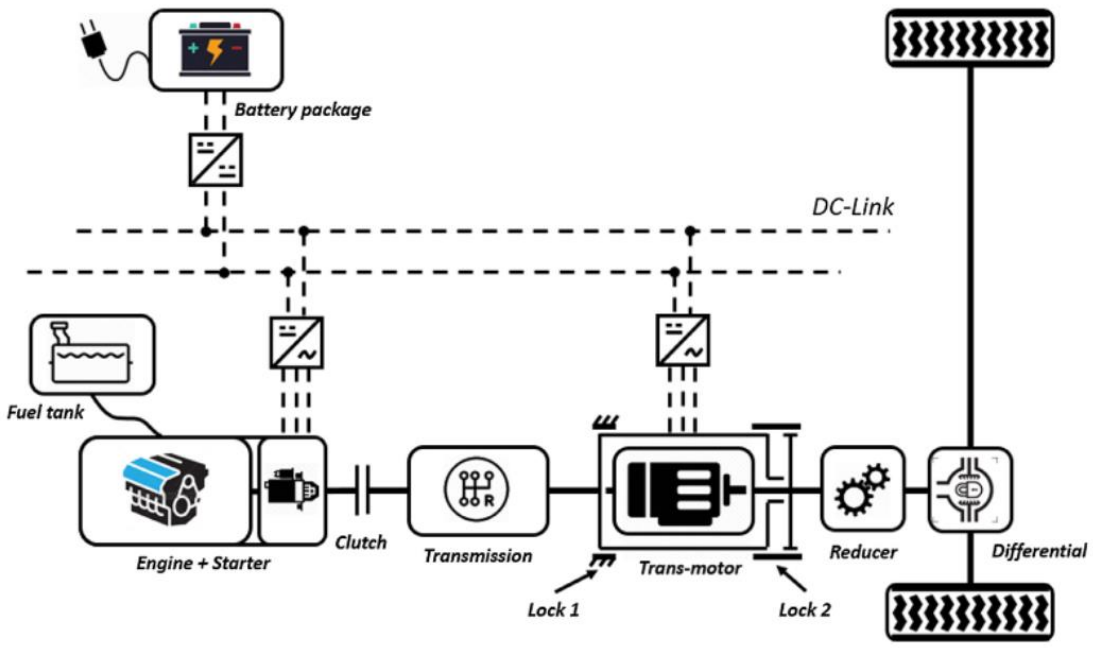


Figure 1.6 Architecture of a series-parallel PHEV powertrain (Li et al., 2019c)

The hybrid electric vehicles (HEV) have been rapidly developed during the past decades because they are regarded as cleaner and greener vehicles. HEVs use ICEs in combination with one or more electric motors connected to a battery pack as a secondary energy supplier providing propulsion to the wheels either together or separately. Thus, they are a compromise between an ICE-powered vehicle and an electric vehicle. They attempt to reduce the emissions of an ICE-powered vehicle, while significantly increasing the mileage and overcoming the shortcomings of an electric vehicle (M. Sabri et al., 2016). The plug-in hybrid electric vehicle (PHEV) is the most popular and advanced HEV. It can exclusively use the electric mode for a certain distance before it switches to a normal HEV, and its batteries can be charged by the outlets (M. Sabri et al., 2016). As for the gasoline-electric HEVs, they have a gasoline engine like the one on most gasoline vehicles. But the gasoline engine on a HEV is smaller with higher efficiency. The architecture of a series-parallel PHEV powertrain can be found in Figure 1.6.

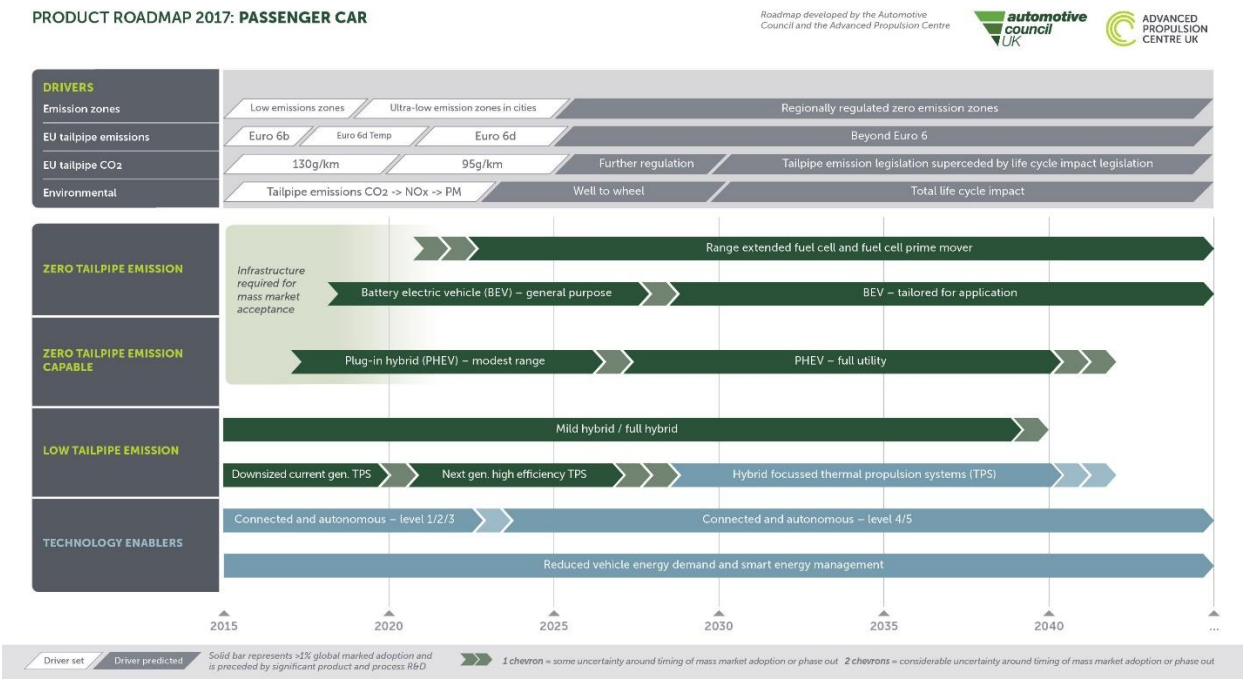


Figure 1.7 UK passenger car roadmap (The Roadmap Report Towards 2040: A Guide to Automotive Propulsion Technologies - UK Advanced Propulsion Centre (APC), 2018)

The latest 2018 passenger car roadmap (shown in Figure 1.7) from the UK Advanced Propulsion Centre (APC) also acknowledges that “thermal propulsion systems will transition from the sole propulsion device to part of a hybrid system” and “the ICEs for passenger cars are expected to evolve through more ambitious downsizing with higher levels of boost, advanced valve train control and more precise injection and breathing strategies to achieve higher efficiencies”. Since ICEs are still widely adopted by HEVs, there is no reason to ban the application and development of ICEs (2018 World Oil Outlook).

1.3 **Challenges for Gasoline Engines**

Basically, there are two main problems restricting the further development of ICEs: fuel consumption and emissions. Although the three-way catalytic converter is able to convert 99% of the HC, NOx and CO emissions to N_2 and H_2O, the wide application of GDI engines still results in non-negligible PM emissions (Zhang et al., 2018; Raza et al., 2018). Hence, in consideration of the human health and environment, the regulations on pollutant emissions are becoming more and more stringent. Besides, since the fuel supply and demand situation becomes increasingly strained, the fuel economy has to be further improved.

The emission legislation for gasoline engines in European Union was firstly proposed in 1992. Table 1.1 shows the details of the European Union emission standards for light-duty gasoline powered passenger cars:

Table 1.1 European emission standards for gasoline powered passenger cars (Category M)*, g/km

Tier	Date (Type Approval)	Date (First Registration)	CO	THC	NMHC	NOx	HC+ NOx	PM	PN [#/km]
Euro 1†	July 1992	January 1993	2.72 (3.16)	-	-	-	0.97 (1.13)	-	-
Euro 2	January 1996	January 1997	2.2	-	-	-	0.5	-	-
Euro 3	January 2000	January 2001	2.3	0.20	-	0.15	-	-	-
Euro 4	January 2005	January 2006	1.0	0.10	-	0.08	-	-	-
Euro 5a	September 2009	January 2011	1.0	0.10	0.068	0.060	-	0.005 **	-
Euro 5b	September 2011	January 2013	1.0	0.10	0.068	0.060	-	0.0045 **	-
Euro 6b	September 2014	September 2015	1.0	0.10	0.068	0.060	-	0.0045 **	6×10^{11} ***
Euro 6c	-	September 2018	1.0	0.10	0.068	0.060	-	0.0045 **	6×10^{11}
Euro 6d-Temp	September 2017	September 2019	1.0	0.10	0.068	0.060	-	0.0045 **	6×10^{11}
Euro 6d	January 2020	January 2021	1.0	0.10	0.068	0.060	-	0.0045 **	6×10^{11}

* Before Euro 5, passenger vehicles > 2500 kg were type approved as light commercial vehicles N_1 Class I
** Applies only to vehicles with direct injection engines
*** 6×10^{12}/km within first three years from Euro 6b effective dates
† Values in parentheses are conformity of production limits

As shown in Table 1.1, all the engine pollutant emissions have to be regulated stringently. Since 2015, the Euro 6 emission standard imposes a stricter legislation on PM emissions and has separate restrictions for the PM number and mass for GDI engines specifically. It has definitely led to extra challenges for GDI engines.

At the same time, to satisfy the continually varying regulations on pollutant emissions and to further reduce the fuel consumption, a lot of new techniques have been applied for modern gasoline engines, such as the intake air boost, stratified combustion, Variable Valve Timing (VVT), and GDI as mentioned above. Based on the latest Roadmap Report Towards 2040 announced by UK Advanced Propulsion Centre (APC), the GDI engine is expected to be effectively downsized before 2020 in coordination with the hybrid powertrain system for passenger vehicles. The downsized GDI engine should be equipped with higher levels of boost, advanced valve control system and more precise injection and breathing strategies to achieve higher efficiencies before 2025. To further reduce the pollutant emissions, newer architectures and novel combustion cycles are required before 2030. However, these techniques make the engine calibration task become a tough multi-objective optimisation problem. In addition, the engine transient operations have become the majority in the real driving scenarios. Fortunately, the engine control unit (ECU) with more powerful computational capability and greater storage is also developed in order to cope with the increasing number of engine sub-systems and the increasing degrees of freedom. Thus, the intelligent and efficient engine calibration and control strategies are highly demanded.

1.4 **Calibration and Control Strategies for Engines**

The engine management system (EMS), also known as the ECU, is an electronic device, fundamentally a computer or microprocessor, which can process the actual measurements from the multitude of sensors located on the engine, operate hundreds of the major control modules, deliver millions of lines of the code, perform millions of calculations every second, and determine the corresponding instructions to each actuator

in real-time, in order to achieve the very best engine performance, emissions and fuel economy. Controlling the engine is always the most processor-intensive job on any vehicles and that is all down to the EMS. Thus, EMS is the brain and the smartest system for today's vehicles, as it is responsible for the control of entire engine working process. In fact, EMS is also where the most advanced engine technologies are found.

The EMS uses a microprocessor to measure and process input parameters from the engine in real-time, then uses a lookup table stored in a microcontroller chip to yield the precomputed output values. The subsequent systems will compute these outputs dynamically. This kind of EMS is only friendly to those who know the system details very well. Besides, such system which relies on the precomputed values is only optimal for the idealised engines and operating conditions. As the engines or operating conditions vary, the system is less able to compensate.

The engine variable settings stored in these lookup tables have to be determined very carefully by engineers in both steady status and transient status, in order to fulfil the legislative standards and give the engines their best performance to meet all the driver’s expectations. This procedure is defined as engine calibration, one of the most important steps of the vehicle development.

The task of engine calibration becomes extremely challenging even for the experienced engineers, requiring a great deal of time and human labour together with expensive equipment and various consumable items, since more and more sub-systems have been introduced to modern engines. The complexity of the calibration arises not only from the large number of adjustable parameters, but also from the trade-off relationships among different objectives, essentially making the calibration a multi-objective optimisation

problem (Millo et al., 2018). In addition, due to the ever-changing emission regulations and the fierce competition in the automotive industry, the trade-off relationship between the calibration efficiency and accuracy is becoming more and more important. For instance, a BOSCH engine management system consists of tens of thousands of variables to be calibrated, and there are more than 12,000 pages of the calibration documents. Generally, it takes over 18 months with at least £1,000,000 to calibrate a new engine for commercial application (International-Limited-of-China-Yuchai 2007).

Thus, a rapid and efficient engine calibration strategy has to be developed.

Meanwhile, with an increasing number of adjustable parameters, the conventional model-based calibration and control strategies cannot satisfactorily address the higher dimensional problems. A highly accurate model can ensure that the calibration results are reliable for the real engine, whereas a poorly constructed model can lead to unfavourable variable settings (Wong et al., 2018a). However, to generate a relatively accurate engine model requires a lot of extra cost, time and labour, while none of the models can be completely accurate as the real engines.

Hence, a non-model-based or online calibration strategy is highly recommended.

Since the modern engine's control parameters grow substantially, it is found that the conventional lookup-table-based control strategy has shown many defects especially in the face of the highly non-linearity and engine parameter uncertainty (Li et al., 2019a).

Therefore, an online self-adaptive engine control strategy with high robustness and self-adaptive capability is necessary.

Robustness refers to the ability to perform effectively while the variables or assumptions are altered. A controller is acknowledged to be robust if it can handle variability and

remain effective under a different set of assumptions. Unlike the conventional one-off lookup-table-based control strategy, the online self-adaptive control strategy can optimise the engine performances continuously by itself in real-time without the need of following the predefined calibration maps. In other words, this strategy is a supplementary technique for the non-model-based engine calibration, because it allows engine to adapt to the change of environmental conditions automatically and always be operated for its best performance based on the feedback signals from the on-board diagnostic (OBD) instruments or sensors in real-time. Thus, it can help engineers get rid of massive engine calibration works and save a great deal of money for automotive companies. Meanwhile, the engine can be sold to customers without detailed pre-defined calibration maps in the future.

Moreover, the proportion that the engine transient operation takes in the on-road or real-driving tests keeps rising. However, due to the increasing complexity of the modern engines, the engine transient calibration and control become more and more difficult. Thus, an online multi-objective engine optimisation approach will benefit to not only the steady-state engine calibration, but also the transient engine optimisation and control.

Accordingly, an online transient calibration approach cooperated with the self-adaptive control strategy should be developed vigorously.

All in all, the future engine calibration approach is supposed to be developed from the manual model-based calibration to the automatic non-model-based calibration. And the control framework is supposed to be shifted from the fixed-gain-based offline static control to optimum-gain-based online self-adaptive control as Figure 1.8 shows. Hence, there is still plenty of space for researchers and engineers to develop advanced controllers and

optimisers, aiming to further improve the engine management system and engine performance.

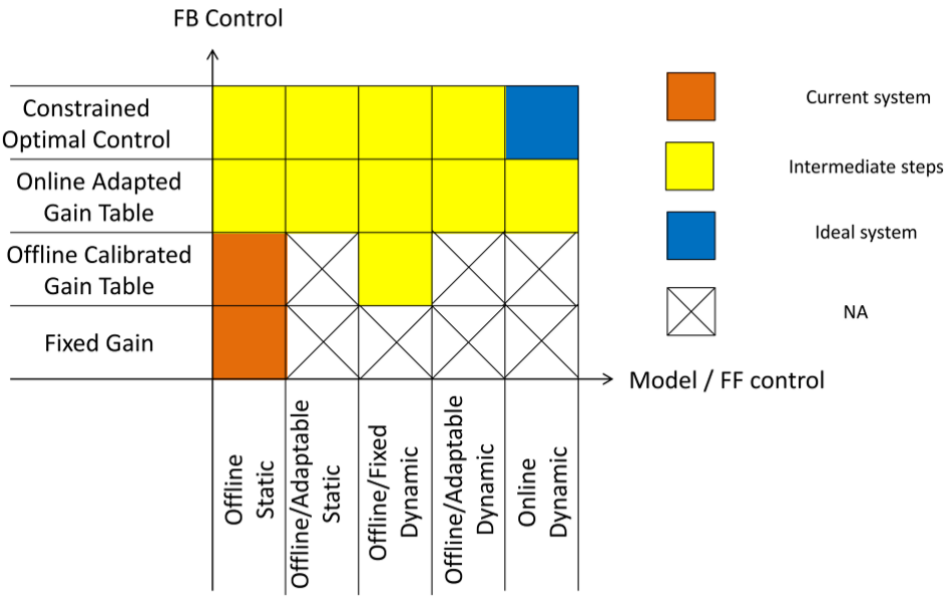


Figure 1.8 Control and calibration for real world fuel economy and emissions (Advanced Control Methods - Ford Research & Innovation Centre - The Future Powertrain Conference 2017)

As a preliminary exploration of the online dynamic optimisation for the engine management system, the studies in this thesis will cover the developments of both steady-state and transient engine multi-objective calibration approaches as well as the engine self-adaptive control strategy. All of these strategies are developed towards the “ideal system” stage, i.e. the blue area in Figure 1.8, in order to achieve the online dynamic optimisation with the constrained optimal and adaptive controller gains.

1.5 **Objectives and Approaches**

The main objective of this thesis is to optimise the GDI engine's management system through artificial intelligence methods. Specifically, this thesis is aimed at improving the GDI engine's performance, minimizing its fuel consumption and pollutant emissions, and reducing the time and labour costed for calibration, by employing the multi-objective meta-heuristic algorithm-based non-model-based calibration approach and the fuzzy-knowledge-based self-adaptive controller. The detailed objectives are listed below:

- To establish an engine test bench at the University of Birmingham for the validation of the novel engine calibration and control strategies.
- To develop a rapid and intelligent non-model-based calibration approach, which is able to automatically find the optimum engine variable settings to achieve the best engine performance based on the multi-objective meta-heuristic algorithm.
- To apply the intelligent non-model-based calibration approach for both the steady-state and transient engine calibration tasks, and to validate the calibration results via experiment studies.
- To implement the fuzzy-knowledge-based self-adaptive control strategy in the Air/Fuel ratio control system of a GDI engine, and to validate the controller's transient behaviour via experiment studies.

1.6 **Thesis Outline**

There are seven chapters in this thesis. The outline of each chapter is listed as follows:

Chapter 1: Introduction

This chapter offers the background and motivation of this thesis, including the transportation energy scenarios, the development and challenges of gasoline engines, as well as a general introduction for the engine calibration and control strategies. The research objectives, approaches and thesis outline are also presented.

Chapter 2: Literature Review

This chapter is formed by the reviews of the publications related to this thesis, including the engine multi-objective calibration strategies and control strategies, the numerical simulation of spark ignition engines, the artificial intelligence methods, as well as the Air/Fuel Ratio (AFR) control system.

Chapter 3: Experimental Set-up

This chapter introduces the details of the experimental set-up, including the hardware and software. As for the hardware, the specifications of the engine employed in this thesis is listed at first, followed by the details of the dynamometer-based test bench, Rapid Control Prototyping (RCP) test hardware, and engine performance measurement devices. In terms of the software, the MATLAB/Simulink-based controller design, RCP associated software and experimental data processing methods are presented.

Chapter 4: Intelligent Non-model-based Calibration Using Artificial Intelligence Algorithms

This chapter introduces the intelligent non-model-based calibration approaches using the chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm and strength Pareto evolutionary algorithm 2 (SPEA2), which can optimise the engine's brake specific fuel consumption (BSFC), particulate matter emission number (PMn) and mass (PMm) by automatically adjusting the engine variable settings: intake valves opening (IVO) timing, exhaust valves closing (EVC) timing, spark timing, injection timing and rail

pressure. The real-time optimisers have been developed based on MATLAB/Simulink and implemented on a V6 GDI engine test bench through Rapid Control Prototyping (RCP) technology to validate their performance.

Chapter 5: Self-adaptive Air/Fuel Ratio Control Strategy with a PI-like Fuzzy Knowledge Based Controller

This chapter presents a new concept of the proportional-integral-like (PI-like) fuzzy knowledge based controller (FKBC) with self-adaptive capability and high robustness. The proportional term, integral term and output signal of the PI-like FKBC are non-linear functions of the input signals and self-tuned in real-time. The PI-like FKBC is implemented on a V6 GDI engine to regulate the AFR. Its transient performance is evaluated through experiment studies.

Chapter 6: Transient Calibration for the PI-like Fuzzy Knowledge Based Controller of the AFR Control System Using a CAPSO Algorithm

This chapter introduces an enhanced intelligent PI-like FKBC using the CAPSO algorithm to automatically define the optimum controller parameters. An alternative time-domain objective function is applied for the transient calibration program without the need for a prior selection of the search-domain. The real-time transient performance of the enhanced controller is investigated via experimental studies on the AFR control system of a V6 GDI engine.

Chapter 7: Conclusions and Future Work

This chapter summarizes all the achievements found in this thesis, and provide some ideas and advises for the future work.

CHAPTER 2 LITERATURE REVIEW

This chapter is formed by the reviews of publications relevant to the “online dynamic optimisation for the engine management system using artificial intelligence methods”. The contents include the major developments of the engine multi-objective calibration and control strategies, the major developments of the numerical simulation of SI engines, the applications of artificial intelligence methods, and the major developments of the AFR control system.

2.1 Engine Multi-Objective Calibration Strategy

The calibration is to determine a combination of engine variable settings which is able to achieve the best trade-off among conflicting engine performances and particular requirements. The fuel consumption and pollutant emissions are the most classic objectives. Performing the calibration demands a fully automated engine control and measurement platform, including the engine ECU, calibration tools, dynamometer and its controller, measurement devices, and various test cell equipment that need to be coordinated. Based on different purposes, the engine calibration can be divided into two major categories. One is the steady-state engine calibration, which is applied for calibrating engine steady status performance and revising the calibration maps for gradually varied conditions, including the engine’s service life and the long term change of the external environment such as the environmental temperature, pressure and humidity (Ma, 2013). The other category is the transient engine calibration, which is used for achieving the excellent engine transient performance. It is also supposed to automatically adapt to the driver’s driving behaviours on-board, such as stop-start,

acceleration and braking in the future (Ma, 2013). The automation level of both two calibration categories should be further improved based on the multi-objective meta-heuristic algorithm, in order to save time, manpower and budget spent on massive engine calibration works.

2.1.1 Model-Based Calibration Using Design of Experiment

In terms of the conventional static manual calibration approach, engineers have to determine the optimum variable settings manually through hundreds of trial-and-error experiments. To be specific, engineers have to adjust the engine variable settings manually based on their experience until the best engine performances are achieved. Then, the corresponding engine variable settings will be stored in the ECU as the calibration maps (Wong et al., 2018a). This manual calibration approach may be still viable when a relatively small group of variables and objectives are involved. However, the rapid development of modern engines has resulted in an increasing complexity of powertrain systems. Due to the large quantity of adjustable variables and the high degrees of freedom, the burden of trial-and-error experiment for the fully manual calibration approach has increased exponentially which requires plenty of consumable items together with remarkable time consumption and manpower. This conventional calibration approach is also constrained by the combinatorial exploration of the parameter space. In addition, when the engine gets aged or meets a wider range of operation conditions, its performances will definitely vary. Then, the engineers have to revise the calibration maps manually again. All of these facts make the manual calibration approach impractical even for the experienced engineers. To address this challenge with a satisfying expenditure of the costs and time, the model-based calibration approach has

been proposed and fully developed during the recent years, which is able to significantly shorten the time required for various calibration tasks (Jiang et al., 2012). As the name implies, the model-based calibration approach is to identify the engine variable settings on the basis of a mathematical engine model and then to verify them on a real engine.

The Design of Experiment (DoE) method is first employed by the model-based steady-state calibration. It is underpinned by statistical modelling of the engine responses, using test data collected at fixed engine speed / load operating conditions (Kianifar, et al., 2016). Rather than running a large quantity of experiments, the DoE method begins with some representative combinations of parameter settings and optimally distributed measurements. Thus, much fewer experiments are demanded to capture the engine performance effectively (Wong, et al., 2018b). In other words, the main purpose of the DoE method is to collect the maximum possible information with the minimum measurement effort. The number of the experiments should be minimised but can still obtain the sufficient data to generate a statistical model with satisfying fidelity to show the engine responses to variations (Jiang, et al., 2012).

The DoE methods commonly used for the steady state engine mapping experiments include Optimal DoEs (e.g. D-Optimal and V-Optimal) and space-filling DoEs (Kianifar, et al., 2016). The space-filling DoEs, particularly the Optimal Latin Hypercube (OLH) designs, have been increasingly used, given that they enable more flexibility to characterise the engine behaviour over a wider design space (e.g. over the whole engine speed-load space), with no prior knowledge required regarding the type of model which is adequate to represent the engine response trends (Kianifar, et al., 2016). On the basis of DoE, the data measured from selected engine tests are then used to construct statistical models using data-driven modelling methods (such as the non-physical models

listed in section 2.3). The calibration is actually carried out on these data-driven models until the optimum results are obtained. As a result, the resources for measurement including the fuel, time and labour can be significantly reduced, but the fact behind this calibration method is still the manual calibration on engine models.

Many commercial calibration tools have been developed in recent years. CAMEO is a commercial tool developed by AVL LIST GmbH for the engine calibration. A mathematical engine model is automatically generated by the CAMEO modelling module based on the experimental data measured in the DoE tests. Then, the calibration process based on the generated engine model will be conducted. The ECU maps are automatically built by the map generators from the optimisation results (AVL CAMEO All-In-One Powertrain Calibration). However, this approach needs experienced calibration engineers to design the corresponding experimental traces by DoE. Even though fewer experiments are required by DoE to construct an engine model, the increasing number of variables and objectives still leads to excessively complicated experimental tests followed by the unpredictable measurement time (Ma et al., 2017). Thus, a highly automated and intelligent engine calibration strategy is desired to replace the manpower and improve the efficiency of the model-based calibration approach. In order to achieve this objective, it is essential to employ a simplified control-oriented mean value model as a virtual engine, together with an efficient online optimisation algorithm which can find the optimal variable settings automatically and incrementally based on the local engine models.

2.1.2 Model-Based Calibration Using Multi-Objective Optimisation Algorithm

The complexity of the calibration task arises not only from the large number of adjustable parameters, but also from the trade-off relationships among different objectives, essentially making the calibration a complex multi-objective optimisation problem. Thus, the multi-objective meta-heuristic algorithms have been introduced to the calibration works by providing a generation-by-generation instruction and optimisation (Yang, 2014). These algorithms can take place of the manpower to get closer and closer to the optimum solutions as they progress.

From the point of view of developing the real-time efficient calibration strategy, only the main features indicating the engine performances are desired. Thus, the zero-dimensional mean value engine model based on the Wiebe function is widely used as a real-time virtual engine. For example, Ma et al. have constructed a semi-physical and semi-statistical GDI engine mean value model as the control-oriented model. It consists of the engine geometry model, gas exchange model and engine performance evaluation model, which are all generated based on the physical equations. But its combustion model is obtained by a series of empirical formulae, i.e. Wiebe functions (Ma et al., 2019). Details for the mean value model will be introduced in section 2.3.3.

Meta-heuristic algorithms, also termed nature-inspired evolutionary algorithms, have been successfully utilized to resolve the multi-objective problems for industry. They involve the trial-based meta-heuristic search processes, and adopt the natural selection concept among species to locate the optimum solutions within the constraint (Kaveh and Dadras, 2017; Netjinda et al., 2015). Each variable or input for the optimisation process is represented by an individual agent. After the iteration started, the individuals in each generation are evaluated based on a fitness function defined by users. At the end of each

iteration, the fittest individuals are picked out to regroup a new generation of individuals. Then, the locally optimum results are located generation by generation. After a certain number of iteration loops, the globally optimum results are supposed to be found. In addition, thanks to the rapid development of CPU storage and calculation ability, the meta-heuristic algorithms can be embedded with the model and locate the calibration results more efficiently. The meta-heuristic algorithms used for the engine calibration will be introduced in details in section 2.4.

A great many of studies have been carried out on the model-based calibration using the meta-heuristic algorithms during the past two decades. A multi-objective optimisation strategy with Pareto-based genetic algorithm (GA) has been developed (Deb et al., 2014). It is able to optimise the brake-specific energy consumption (BSEC), brake thermal efficiency (BTHE), and volumetric efficiency for a single-cylinder diesel engine with hydrogen as dual fuel. However, this strategy uses a single adjustable parameter. Then, Millo et al. have used GA for the multi-input and multi-output (MIMO) calibration of a diesel engine to minimise the brake specific fuel consumption (BSFC), combustion noise, brake specific emissions and exhaust temperature (Millo et al., 2018). However, although it can improve the automation level for the engine calibration, the time consumption is still high and the GA is introduced unclearly. Moreover, Ma et al., 2019 develop an intelligent model-based multi-objective calibration approach using the strength Pareto evolutionary algorithm 2 (SPEA2) (Ma et al., 2019). A simplified control-oriented GDI engine mean value model is employed. This approach can locate multiple optimum engine variable settings (spark timing, injection timing, intake valves opening (IVO), exhaust valves closing (EVC), and rail pressure) to minimise the BSFC and PM emissions (including both PM number and mass) at a particular operating point with good accuracy and speed. In

each iterative cycle, the SPEA2 optimisation approach is performed to search for the best variable settings. The iteration is repeated until the globally optimum variable settings of that operating point are found. Thus, the calibration process can repeat with multiple operation conditions to cover the entire engine map. To verify the effectiveness of the proposed approach, the calibration results are validated by experiments on a real engine test bench. The workflow is shown in Figure 2.1.

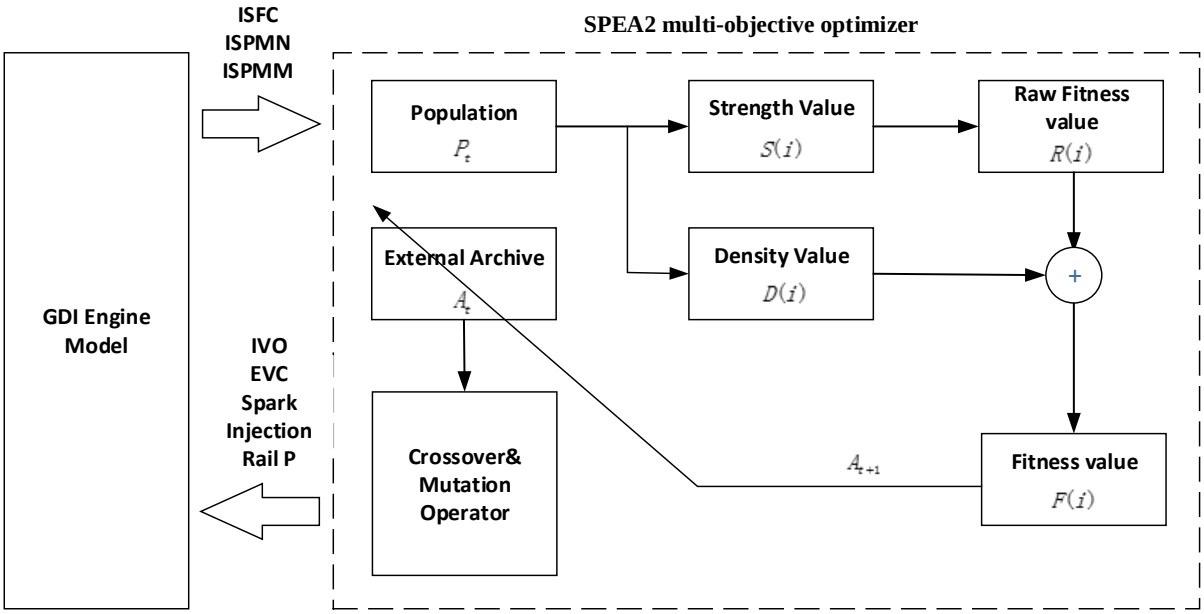


Figure 2.1 Workflow of the intelligent model-based multi-objective calibration approach using the SPEA2 (Ma et al., 2019)

However, with the increasing complexity of modern engines, the conventional model-based calibration approach cannot satisfactorily address the issues of higher dimensionality problems and transient operations. Because the calibration is conducted on models instead of real engines, the reliability of the calibration results for the real engines significantly relies on the accuracy of the engine models. But it is almost impossible to generate a sufficiently accurate control-oriented engine model with the

desirable computational speed. Meanwhile, the control-oriented engine models, mostly zero-dimensional mean value models, which neglect most of the engine dynamic phenomena, are poor at imitating the engine fast transient behaviours. Moreover, the model-based calibration approach is not robust enough, which means that engineers have to generate a new model for a new engine and repeat the calibration jobs again. Thus, an intelligent online dynamic, or termed non-model-based, calibration approach is demanded.

2.1.3 Non-model-based Calibration Using Multi-Objective Optimisation Algorithm

Compared with the model-based calibration approach, the non-model-based calibration approach is able to skip the procedures of DoE planning, experimental measurements, engine model construction and validation, as Figure 2.2 shows. Accordingly, a computational intelligence non-model-based calibration approach has been developed by Ma et al. (Ma et al., 2017). The SPEA2 is applied to this automatic engine calibration process without the need of an engine model. Both of the model-based approach and non-model-based approach will cost time to locate the globally optimum results because of the operation of the algorithm. But the non-model-based approach can save the time spent on modelling (from many months to several weeks, depending on the complexity of the model). Thanks to the development of the rapid control prototyping (RCP) and external ECU bypass technology, a Simulink-based calibration optimiser embedded with the algorithm is developed and allowed to communicate with a real GDI engine on-board in real-time. It overcomes the shortcomings of the model-based calibration approach listed in section 2.1.2. Experimental results have verified that this non-model-based calibration approach is able to automatically find the optimum engine variable settings

(spark timing, injection timing, IVO, EVC, and rail pressure) to minimise the fuel consumption and PM emissions, with high efficiency and good accuracy. It does not rely on either the engine model or massive test bench experimental data. Once the optimiser is generated and the target system is able to communicate with the optimiser, the optimiser will treat the system as a “black-box” system and locate the optimum variable settings automatically and efficiently.

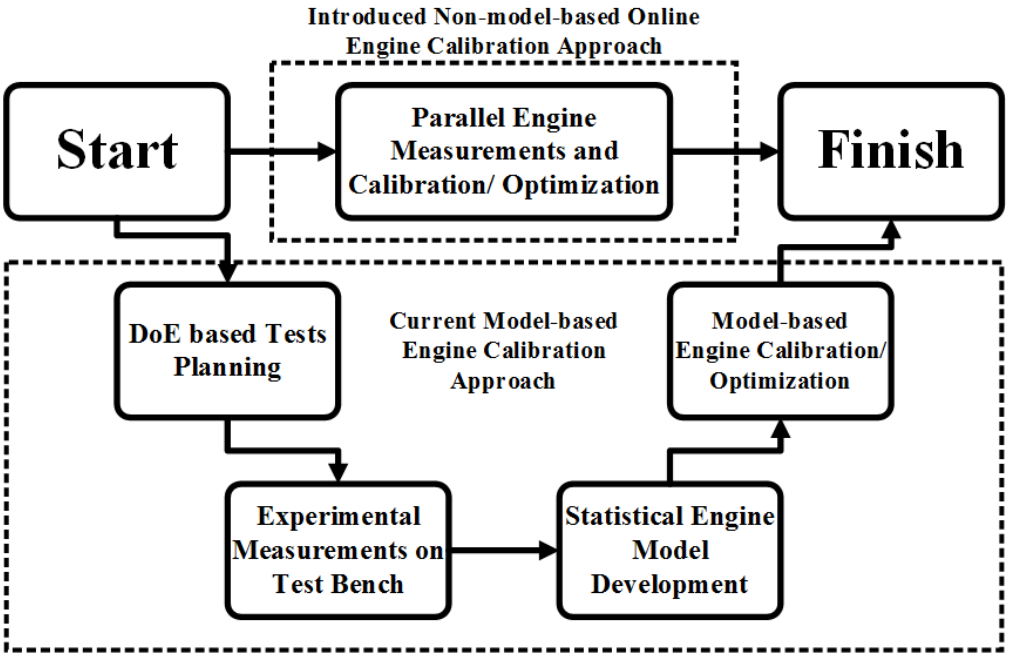


Figure 2.2 Comparison between the model-based calibration and the non-model-based calibration (Ma et al., 2017)

However, although the calibration process is totally automatic, the time consumption should be further reduced and the calibration results should be further improved. To achieve these targets, a more efficient online optimisation algorithm must be employed within this framework of non-model-based calibration, in order to further improve the automation level of engine calibration.

2.1.4 Steady-state Multi-objective Calibration Strategy

For an engine, when the certain environmental or operating conditions change, it takes a while for the engine performance to stabilize and reach its final status. As for the steady-state calibration, the time domain is regarded as “infinity”. Only the final engine performance in steady status is taken into consideration and its corresponding engine variable settings will be stored in the ECU.

During the past decades, a lot of studies have been concerned with the steady-state multi-objective engine calibration, but most of them still focus on the model-based calibration approach. Jiang et al. have used the DoE model-based calibration approach to define the optimal settings for spark advance, AFR and intake VVT across the interested operating region (speed and load), to minimise the BSFC of a 2.0L in-line 4 cylinder gasoline engine (Jiang et al., 2012). Kianifar et al. have improved the model-based engine calibration using an integrated sequential DoE strategy to manipulate the injection timing, fuel rail pressure and variable camshaft timing variables in order to minimise the CO_2 and particulate number emissions of a V8 GDI engine (Kianifar et al., 2013).

2.1.5 Transient Multi-objective Calibration Strategy

The transient response can be treated as the system’s instant behaviour against the input change or disturbance. Thus, compared with the steady-state calibration which focuses more on the final results, the engine transient calibration considers not only the final engine steady-state performance, but also the process of how the system reacts against the inputs dynamically with less settling time and less oscillation. The duration during which the system keeps adopting the changes is called the transient state or response.

The current engine transient calibration is still a model-based calibration and replies on the steady-state calibration results with the interpolation method for transient operations. This kind of calibration approach cannot really capture the transient driver's behaviours, but approximately simulate the transient performances by fitting the steady-state calibration results. Thus, the effect of this transient calibration approach still heavily relies on the accuracy of the model and calibration maps. Although an accurate engine model and well-trained steady-state calibration results are able to improve the transient calibration results, they are still significantly limited by the model-based steady-state calibration methods and cannot be further implemented for the higher dimensionality optimisation problems (Ma, 2013). Therefore, it is expected to develop an online dynamic optimiser which is able to capture the engine transient performances (e.g. AFR) corresponding to the driver's transient behaviours continuously in real-time, in order to improve the reliability of calibration results and save engineers from masses of calibration works.

Montazeri-Gh et al. have applied the particle swarm optimisation (PSO) algorithm for tuning the gains of a gas turbine engine (GTE) fuel control system (Montazeri-Gh et al., 2012). The fuel flow is optimised to reduce the response time during the engine acceleration and deceleration, and reduce the fuel consumption as well. Then, Park and Choi have used a multi-objective Pareto method with the model-based control to optimise the dual-loop EGR split strategies, and thereby minimised the NOx and fuel consumption for a diesel engine through the step transient analysis (Park and Choi, 2016). Moreover, Yu et al. have used the PSO algorithm to optimise the acceleration controller of a gas turbine aero-engine in the transient state (Yu, et al. 2017). The simulation results show that the optimised acceleration controller can track the acceleration command rapidly and

accurately. However, all of these methods are still validated through simulation studies on engine models. Their performances on real engine transient operations are hard to be evaluated.

2.2 Engine Control Strategy

In order to promote the automation level, reduce the cost for calibration, and improve the system transient response, more and more intelligent control strategies have been developed and implemented on gasoline engines. Among these control strategies, the robust adaptive control method gets the most attention, which allows the controller to be easily parameterized and allows engineers to get rid of the numerous calibration works. There are two categories of the control strategies, i.e. the model-based control methods and the non-model-based control methods.

2.2.1 Model-based Control Method

In the model-based control system, the data transmission is usually discrete, and the control signal is first transmitted to a prediction model, which will optimise the control signal before it is sent to the actual control system. Even though the engine management system is very complex, its internal dynamics are actually interrelated and often limited within a scope. For example, the opening of throttle and the VVT must be limited in a given range. Thus, most control problems in the automotive industry are stated as the constrained multi-objective control problems. The model predictive control (MPC), also termed the moving horizon control or receding horizon control, has been widely adopted to resolve the multi-objective control problems for many years (Zhang et al., 2017).

The MPC was first developed in the 1980s for the control of complex linear multiple-input multiple-output (MIMO) systems with constrained variables. Initially, it was only suitable for solving slow dynamic systems. Nowadays, after years of development, it has become a common control method in many fields and has been extended to solve many non-linear and fast dynamic systems (Del Re et al., 2010).

The MPC is widely used in the feedforward control. It formulates the control problem as a MIMO control task and yields an approximate solution based on the dynamic prediction model while satisfying the constraints. The approximate solution will be very close to the actual optimum solution at each moment. The MPC sorts the optimised control signals and outputs an optimum one to regulate the cost function to its minimum, maximum or a specific value under the constraints. The MPC can only optimise the current timeslot repeatedly while considering the impact on the future timeslot. When the actual response of the plant is taken into account, the closed-loop control can thus be achieved (Del Re et al., 2010). The classic structure of an MPC implementation can be found in Figure 2.3. It consists of three main blocks: the system model, the dynamic optimiser, and the evaluation block with the cost functions and constraints.

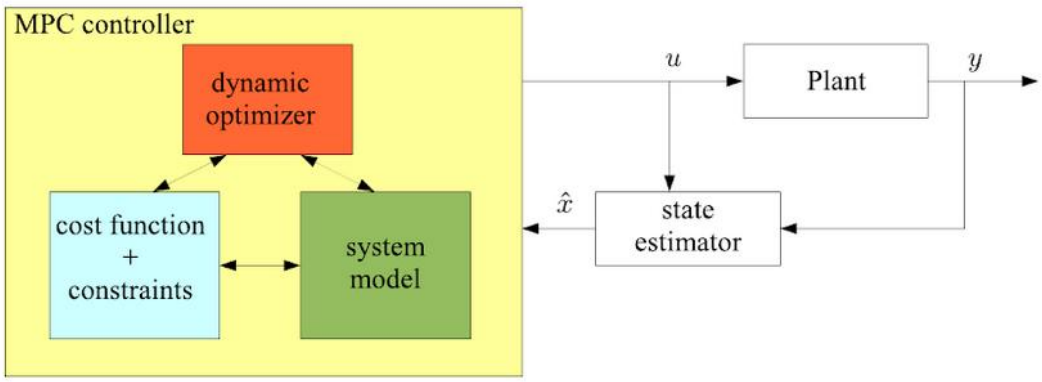


Figure 2.3 A general MPC control loop (Del Re et al., 2010)

The performance of an MPC controller depends on the accuracy of the internal prediction model. However, since the conventional linear MPC cannot directly solve the non-linear systems, the internal prediction model for a non-linear system has to be linearized first before implementation. Accordingly, the entire internal prediction model will be divided into a set of state-space identified models which are trained separately, in order to deal with the wider range of engine operations (Borhan et al., 2015). Meanwhile, each state-space identified model is assigned to an MPC controller. Thus, a single MPC controller is converted to multiple MPC controllers. At runtime, the individual MPC controllers are switched by the particular engine operation conditions (Pahasa and Ngamroo, 2015). Based on this framework, the MPC controller can achieve a relatively better tracking performance and prediction accuracy (Zhang et al., 2018; Koli et al., 2018; Zhu et al., 2016). Nevertheless, the multiple MPC controllers cannot assign the sub MPC controllers to individual systems effectively for the transient changes of engine operating conditions and high nonlinearities (Zhang et al., 2018).

To overcome the disadvantage that the linear MPC cannot handle the non-linear system directly, a concept of non-linear MPC (NMPC) was also proposed (Lee et al., 2011). The NMPC controller can achieve a better trajectory tracking performance for non-linear applications by using the non-linear prediction model, which makes it possible to output the prediction as a non-linear function of the current state and input trajectory. However, the computational burden gets higher as well. To improve the computational efficiency for real-time NMPC applications, the Sequential Quadratic Programming (SQP) is employed as the solver (Zhu et al., 2016). SQP is an iterative method to optimise the nonlinearly constrained optimisation problems (Torrisi et al., 2016; Albin et al., 2015). Although it is very computationally demanding, the NMPC controller has been widely employed for the

control of ICEs and the enhanced transient behaviours are also observed (Torrisi et al., 2016; Albin et al., 2015; Lee et al., 2011).

An innovative idea has been proposed in recent years which combines the MPC with an artificial neural network (ANN). This is because the ANN model is able to approximate the multi-dimensional input-output relationships among selected variables, which can be compatibly implemented to generate the internal prediction model and solve the online optimisation problem (Pan and Wang, 2012; Yan and Wang, 2014). Therefore, the neural network model predictive control (NNMPC) is developed for many applications, including AFR control of SI engines (Shi et al., 2015), combustion control of HCCI engines (Janakiraman et al., 2016), air path control (Albin et al., 2015; Albin et al., 2017), and coordinated control over the throttle and waste-gate of turbocharged gasoline engines (Hu et al., 2018).

The model-based control has two major inherent drawbacks. Firstly, the reliability of the prediction solutions on real engines significantly depends on the accuracy of internal prediction models. Meanwhile, the internal prediction models have to be of low complexity in order to get the reasonable computational speed. Thus, it’s almost impossible to generate a sufficiently accurate model which can capture all the engine transient and non-linear behaviours while still satisfying the computational speed for an efficient online optimisation. How to figure out the trade-off between the reasonable computational speed and the reliable control solution is still the main challenge for the model-based control. Secondly, the model-based control approach is not sufficiently robust. Hence, it is not sensible to expect that one predictor is universal for any models and any engines. A new model-based controller has to be generated and trained for a new plant or under new operation conditions.

The MPC was usually employed for the higher level supervisory control scheme when the sensors failed to offer the measurements in real-time. The higher level supervisory control scheme is supposed to output a precise control signal for the plant rapidly, because it has to give the following individual controllers a precise overall process instruction and allow the integration of operations between individual controllers. However, the transmission delay caused by sensors used to be a considerable problem and only the effective estimator can mitigate the consequence. Nowadays, thanks to the remarkable development of sensors and ECU, the engine dynamics can be reflected in real-time. Thus, only a general feedforward control loop is essential. By contrast, the feedback control loop becomes more and more important, after all it is the key to compensate and correct the error for the control system. In the circumstances, the robust and self-adaptive capabilities are more aligned with the request by feedback control loop, especially for the underlying or lower level direct control scheme. Accordingly, the non-model-based controllers have attracted the attention of researchers to cope with the modelling uncertainties and high nonlinearity.

2.2.2 Non-model-based Control Method

Because the high complexity, system uncertainties, unknown dynamics, external disturbances, high nonlinearity and various transient behaviours are widespread in the modern engines, it is almost impossible to generate a sufficiently accurate control-oriented engine model with the favourable computational speed. Thus, the non-model-based controllers have been widely developed during the past decades.

The non-model-based control method, also simply called motion-based control technique, does not rely on any models and utilizes the system real-time feedback signals as the core inputs. It purely depends on the system states (motion), and assumes that certain variables are unknown but constrained (Saied et al., 2019). Users just need to find the response of the close-loop system and then adjust the controller parameters based on that, even though the system models are unavailable. Therefore, the non-model-based control method is capable of high robustness and fault-tolerant capability.

With the development of the rapid control prototyping (RCP), the non-model-based control strategies are applied in automotive industry. Generally, the RCP tool provides a means for the rapid development and testing of the control strategy, simply using the MATLAB/Simulink-based controller. The controller is implemented on the real machine with multiple I/O interfaces to control the real-world systems in real-time. Then, engineers are allowed to develop, test, verify and refine both the traditional and novel control laws, and evaluate the response of a particular system based on the actual target system instead of a model. Meanwhile, engineers can deploy the control system to the hardware easily without coding complex software. The controller designed and implemented on the RCP tool gathers the input signals, executes the program, computes the algorithms, and commands the outputs to the connected actuators. Engineers can accomplish the non-model-based control on a real-world target system once an actual test bench has been set up in the laboratory. Moreover, the RCP decreases the development time by giving engineers an opportunity to verify and validate the controller continuously during the earliest possible stages. It owns high control fidelity with low-latency.

The simplest and basic non-model-based control approach is the proportional–integral–derivative (PID) control, which was first proposed in the early 1900s. It is still the most

commonly used control algorithm in the control field and automotive industry so far, owing to its numerous benefits, including the relatively simple structure, good stability and security, easy implementation and low-cost maintenance (Civelek et al., 2016; Guzmán et al., 2018; Razvarz et al., 2019). The PID controller retains its strategic position in the industry and non-model-based control field (Saied et al., 2019). More than 95% of the closed-loop control systems in the industry are based on PID control, while PI controllers accounting for the majority (Kumanan and Nagaraj, 2013; Guzmán et al., 2018). That is because the PI controller only needs to identify two parameters, which definitely simplifies the control system while maintaining desirable control results (Dogruer and Tan, 2018). Thanks to its simple structure, even the operators without abundant control experience can tune the PID controller. In contrast, many modern advanced non-model-based controllers are very complicated in structure, while providing a slight improvement.

The transformation function of a PID controller can be expressed as:

$$u_{PID}(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)E(s) = K_p E(s) + K_i \frac{1}{s} E(s) + K_d s E(s) \qquad (2.1)$$

where $u_{PID}(s)$ is the control signal and $E(s)$ is the error signal; K_p is the proportional gain; T_i and T_d are the integral time constant and the derivative time constant respectively; $K_i = K_p/T_i$ is the integral gain and $K_d = K_p * T_d$ is the derivative gain; s is the independent variable of the Laplace transform. The effect of each gain is: the proportional term generates an overall control action proportional to the error signal, and shortens the rise time; the integral term reduces the steady-state error by low-frequency compensation until the steady-state error is eliminated; the derivative term improves the transient response with high frequency compensation and reduces the overshoot and settling time.

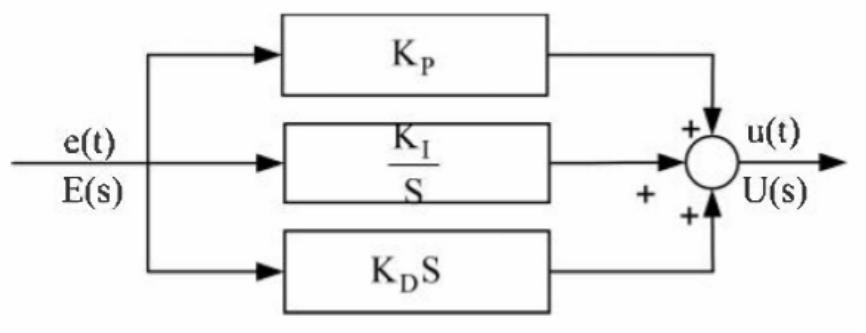


Figure 2.4 Block diagram of the PID controller (Meshram and Kanojiya, 2012)

In recent years, many studies have found that tuning the derivative term K_d has no significant effect and may even degrade the stability for the transient or instant control system. Thus, many operators decide to abandon the derivative term (Li et al., 2006; Ang et al., 2005). This is another reason why the PI controller is even more popular than the PID controller. For the optimum performance, K_p, K_i and K_d have to be defined and adjusted corporately. Due to the different control plants, most PID controllers are tuned on-board.

The Ziegler-Nichols tuning method is widely used to determine the initial parameters of a PID controller. It begins with a low/zero value of K_p. Then, increase it until a steady-state oscillation occurs. Note this gain as K_{cr}, and the sustained oscillation with period P_{cr} (measured in seconds as Figure 2.5 shows). Finally, calculate the K_p, T_i and T_d based on K_{cr} and P_{cr} as the following table shows:

Table 2.1 Ziegler–Nichols tuning method: gain estimator chart

Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$0.833P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

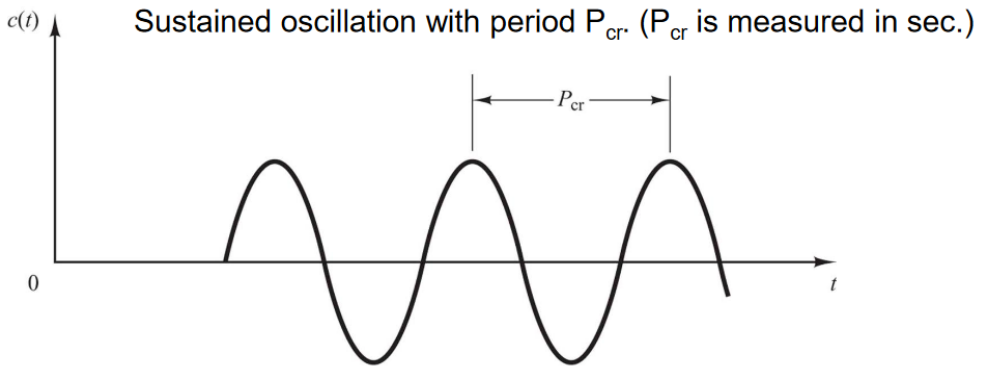


Figure 2.5 Sustained oscillation with a period P_{cr} for the Ziegler–Nichols tuning method (Meshram and Kanojiya, 2012)

Generally, the conventional lookup-table-based PID/PI controller has three major defects making it impossible to be further applied for more practical dynamic systems.

Firstly, it is very troublesome to determine the controller parameters for a given system (Song et al., 2017). Ziegler-Nichols tuning rules can only generate a series of approximate values as reference points followed by the hand-tuning method through "guess and check" for each engine operating condition one by one until fine parameters, i.e. the relatively good proportional, derivative and integral gains, are obtained (Yadav et al., 2016). Thus, it is time-consuming and labour-intensive.

Secondly, the calibrated controller gains for each operating condition will be stored in a lookup table. The controller has to work in accordance with the lookup table and perform with the constant proportional, derivative and integral gains under a certain operating condition (Yadav et al., 2016). Hence, the lookup-table-based PID/PI controller is not robust. Due to the lack of self-adaptive property, the look-up table has to be re-calibrated again for different control plants or operating conditions. This drawback also results in the relatively poor performance of the lookup-table-based PID/PI controller at the non-

calibrated operating conditions, because the fact behind it is still the linear interpolation method, which comes to the third defect (Yadav et al., 2016).

The third major defect is that, even though the PID/PI controller can effectively regulate the linear time-invariant systems, its performance with the highly non-linear systems, time varying dynamic systems, and complex systems with uncertain parameters or unclear mathematical relationships is still unclear, because the controller parameters at the non-calibrated operating conditions are obtained through the linear interpolation method as mentioned above rather than systematically or adaptively determined (Civelek et al., 2016; Song et al., 2017).

Thus, the existing methods have to be promoted considering its fault-tolerant capability, self-adaptive capability, simplicity, robustness and effectiveness. In the circumstances, a single traditional PID controller cannot meet the demand and it has to combine with other artificial intelligence techniques. In recent years, many studies use the optimisation algorithms to automatically generate the optimum parameters for PID controllers (Belkadi et al., 2017; Asgharnia et al., 2018; Sravan Bharadwaj et al., 2018; Xiang et al., 2019). The parameters of the PID/PI controller are continuously and automatically assigned by the on-board control algorithm based on the settling time of the regulating object and the program of the algorithm. Compared with the Ziegler-Nichols tuning method, it is observed that the algorithm-tuned PID controller can reduce the calibration cost and result in better performance. At the same time, many studies combine the PID controller with the fuzzy control method and improve its robustness and self-adaptive capability (Dettori et al., 2018; Verma et al., 2018; Rodríguez-Castellanos et al., 2018). The fuzzy knowledge based control method is another advanced artificial intelligence non-model-based control strategy which will be introduced in section 2.4.4.

2.3 Numerical Simulation of SI Engines

This section will review the numerical simulation methods of SI engines over the past decades and give the reasons why it is almost impossible to generate a sufficiently accurate control-oriented engine model with the favourable computational speed.

The modelling of ICEs is usually performed to accomplish the following goals: designing new engines, designing control systems, calibration of parameter settings, predicting engine's performance without testing when the testing is highly expensive or too time-consuming, and getting the data which is difficult (or even impossible) to measure through testing (Shamekhi and Shamekhi, 2015). The main duty of the model is to indicate the effects of system (i.e. engine) inputs on its outputs, both accurately and in real-time.

The numerical modelling has been employed to study the working process of ICEs and verify the engine optimisation or control strategies for a long history, which can date back to the 19th century. Since the 1960s, the engine numerical simulation has made a breakthrough thanks to the rapid development of computers (Suzuki, 1997). Although various modelling approaches have been developed, the trade-off between the model accuracy and the computational speed is still a challenge, which has already constrained their wider applications in the automotive industry. The numerical models can be classified by different methods. This section will introduce three major categories of the engine numerical models, i.e. multi-dimensional model, quasi-dimensional model, and zero-dimensional model respectively.

2.3.1 Multi-dimensional CFD Models

The multi-dimensional computational fluid dynamics (CFD) modelling approach is the most accurate, complete and detailed modelling methodology as it is generated based on the laws of thermodynamics and physical equations of the energy transfer.

Nevertheless, due to the high complexity of the engine mesh generation and the massive number of sensible cells involved in the combustion, to simulate an ICE by CFD modelling may cost many weeks (Ma, 2013; Lucchini et al., 2015). Given that the computational time increases with the system volume discretizes, the high computational cost and the large amount of data for calculation become two burdens for the purpose of engine control. Although researchers made considerable efforts to reduce the simulation time for the CFD model (Lucchini et al., 2015), it still cannot keep up with the demand of the real-time controller development.

Thus, the best use of the CFD modelling is for more detailed studies concerning the limited conditions, or particular features (e.g. flow dynamics through valves, fuel injection, mixture formation, combustion process, in-cylinder flow and turbulence development), or a specific engine component (e.g. a single cylinder or a manifold), or to support the theory and model development instead of the real-time controller development (Verhelst and Sheppard, 2009). In other words, it usually provides only the component level details (such as pressure drops, flow distributions, fuel and air mixing), but cannot provide a system level perspective (Millo et al., 2011). Moreover, the processes running in a modern ICE are so non-linear and complicated that it is impossible to model all of them through the CFD approach.

2.3.2 Quasi-dimensional Multi-zone Models

The quasi-dimensional models and zero-dimensional models are all used for analysing the combustion process by thermodynamic theory and regardless of the fluid dynamic process (Ma et al., 2008; Perini et al., 2010). They are both classified as the thermodynamic models, where the equations constituting the basic structure of the model are based on the conservation of mass and energy, and they are only dependent on time (resulting in ordinary differential equations) (Verhelst and Sheppard, 2009). But the combustion model of the zero-dimensional engine model is developed by the empirical formulae or curve fitting. Whereas, the quasi-dimensional combustion model introduces a chemistry-based sub-model to describe the combustion process. In addition, the zero-dimensional model considers the in-cylinder mixture as homogeneous, while the quasi-dimensional model divides the combustion chamber into several separate zones and the parameters for different zones are different (Stiesch, 2003).

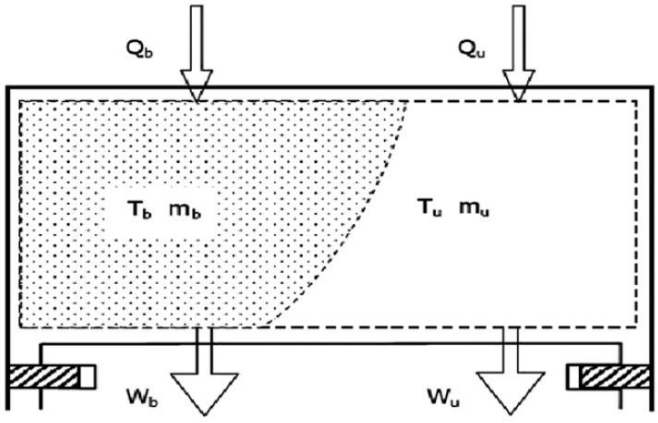


Figure 2.6 Two-zone combustion chamber (Ma et al., 2008)

The two-zone model is the most popular and simplest quasi-dimensional model which considers the chamber (or in-cylinder mixture) as two parts, i.e. the burned zone and the

unburned zone, divided by the flame front surface as shown in Figure 2.6. For the sake of simplicity, it is assumed that no exchange occurs between the two zones. The subdivision into two zones implies two different temperatures, while each one is assumed as constant within its zone. But the in-cylinder pressure is homogeneous throughout the cylinder. Moreover, the blow-by flow to crevices is neglected (Perini et al., 2010). Speaking of the burned zone, the temperature gradient is significant because the combustion occurs at different times. To resolve this problem, the burned zone can be divided into more sub-zones. This results in a multi-zone model.

The multi-zone model is usually designed to simulate the combustion process and pollution formation in detail, because the multi-zone model takes into account the spatial distribution and temperature inside the cylinder (Xue and Caton, 2012). Thus, it is possible to simulate the detailed in-cylinder air-fuel distribution and gas components. Many studies have verified that the multi-zone model is able to simulate the engine combustion progress, specifically the heat transfer, IMEP, thermal efficiency, in-cylinder pressure, and predict the pollutant emissions with good accuracy (Nazoktabar et al., 2018; Bissoli et al., 2016; Nobakht et al., 2011). However, the detailed physical mechanisms will also increase the model complexity, involve more time-consumption for simulation, and impair the real-time performance remarkably, without improving the simulation results obviously. In this situation, the multi-zone model is not encouraged to be widely applied for practical applications in the automotive industry.

2.3.3 Zero-dimensional Mean Value Models

The zero-dimensional mean value model, also termed the single-zone model, is a pure thermodynamic model based on the transformation of mass and energy. It completely ignores the spatial differences of each in-cylinder variable and simplifies the combustion process as a heat addition process (Ma et al., 2008). The engine component such as cylinder and manifold is simply treated as a series of control volumes, and the gas mixture is assumed as uniform state. Hence, the in-cylinder pressure, constituent and temperature are all regarded as homogeneous states and some particular in-cylinder processes, e.g. the vaporization of fuel, are thus neglected (Kumar et al., 2013). To a certain extent, the zero-dimensional model can be perceived as a summary or foundation of the other numerical models (Guzzella and Onder, 2010).

The rate of heat addition of the zero-dimensional mean value model is usually obtained by empirical formulae, e.g. Wiebe functions (Ma et al., 2008). This mass burning rate has to be empirically defined for every engine operating condition on the basis of prior engine experiments or experience, and it is not expressed in terms of physical quantities (e.g. fuel properties, engine geometry, etc.), resulting in the problematic extrapolation to other operating conditions (Verhelst and Sheppard, 2009).

This kind of model is usually used to examine the engine overall performance parametrically (Ma et al., 2008). When such data are available or require little extrapolation, this can be the best approach, as it avoids modelling of the in-cylinder processes. In effect, it works back from a known result and hence should provide unrivalled accuracy in predictions (Verhelst and Sheppard, 2009). For example, by using the mean value engine model, researchers have investigated the variable valve timing (VVT), overexpansion, and variable compression ratio (VCR) characteristics of the

Atkinson cycle engine (Murtaza et al., 2016). Another case in where mean value engine model is frequently applied is the manifold's modelling (Wahlström and Eriksson, 2011).

Because the mean value model is capable of low computational cost and simple structure, it can reduce the complexity of modelling, while maintaining a physical description of the main phenomena in order to achieve the best compromise between the detail level and the computational requirement. Besides, the time in a mean value model is an independent variable without the discrete cycles of engines (Guzzella and Onder, 2010). Since the model-based calibration and control systems do require a control-oriented model, the mean value model becomes the most prevalent simulation platform for the engine investigation and controller design (Wang et al., 2018; Theotokatos et al., 2018).

Figure 2.7 presents a mean value model architecture of a SI engine with major sub-systems as throttle body, intake manifold, gas exchange, fuel injection, wall-wetting, torque generation and engine inertia. Beyond these sub-systems, the dynamic compensation and correction involving the time delay, fluid delivery delay and temperature should also be considered to promote the model accuracy. Nevertheless, all of these states in the mean value models are determined and may vary on a case by case basis (Tan, 2015). Therefore, for the modern engines with a significantly increasing number of sub-systems, the building of a mean value model is still a tough job.

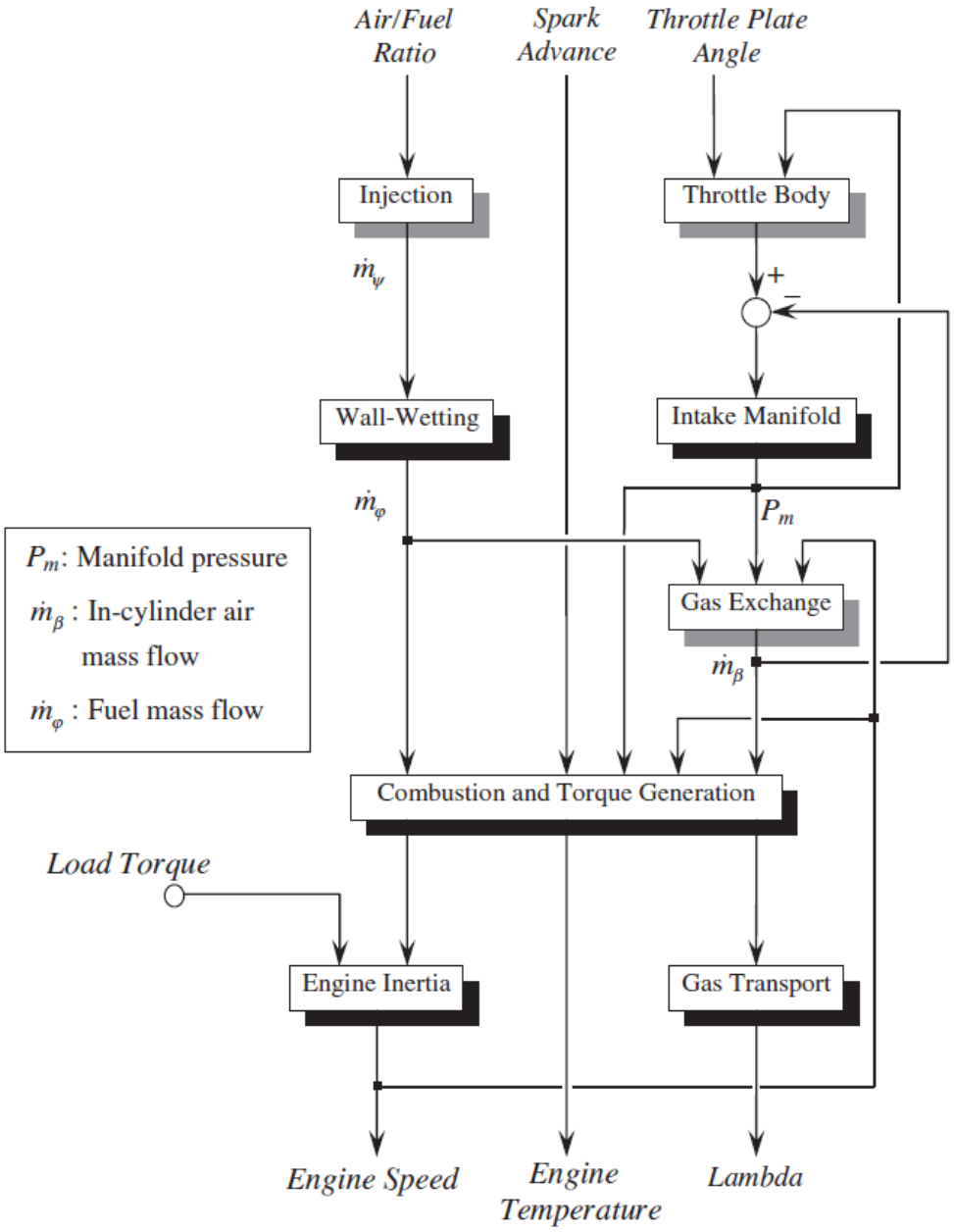


Figure 2.7 Mean value model architecture of a SI engine (Guzzella & Onder, 2010)

From the early 1990s, when the mean value model was first proposed (Hendricks, 1991; Müller et al., 1998) until now, this model has been suffering from several essential downsides. As introduced above, it considers the whole cylinder to be uniform without spatial distribution, and the burn rate profile which is necessary for the model has to be

obtained by other empirical formulae (e.g. Wiebe functions), which require extensive works of the experimental calibration. A great number of experiments with various engine operating variables have to be done to calibrate the associated parameters in Wiebe functions, in order to simulate the burn rate (Ma et al., 2008).

Besides, even though the mean value model is supposed to improve the response speed over other engine models, the fast response is achieved at the cost of losing the model accuracy. Therefore, this kind of model is relatively weak at indicating engine performances and pollutant emissions. In order to overcome the limitations, some researchers use the thermodynamic principles to govern the physical processes, and develop the thermodynamics-based mean value model (Lee et al., 2013; Lee and Jung, 2016), which replaces the input-output data relationship (e.g. Wiebe functions) based combustion subsystem. This action significantly improves the model precision, because the simulation of combustion is based on physical laws instead of nonphysical data relationships. But the computational speed overly decreases due to the thermodynamic calculations. Briefly speaking, while the CFD models are accurate but too slow, the mean value models are rapid but not accurate.

2.3.4 Zero-dimensional 'Black-box' Models

The zero-dimensional black-box model, also known as the data-driven model, is a non-physical model which follows a quasi-static approach based on the experimental steady-state maps. The relationships between the model inputs and outputs are completely set up on the basis of the experiment data. The zero-dimensional black-box model with rapid computational speed and superior real-time capability is propitious to simulate the

complex system with limited understanding of its inner structure. For example, due to the complexity of physical engine models, zero-dimensional black-box models have been widely employed to analyse the power, fuel consumption and pollutant emissions on driving cycles, where the transient performances can be simulated quite smoothly by a sequence of stationary states (Rahimi-Gorji et al., 2017; Kihas and Uchanski, 2015).

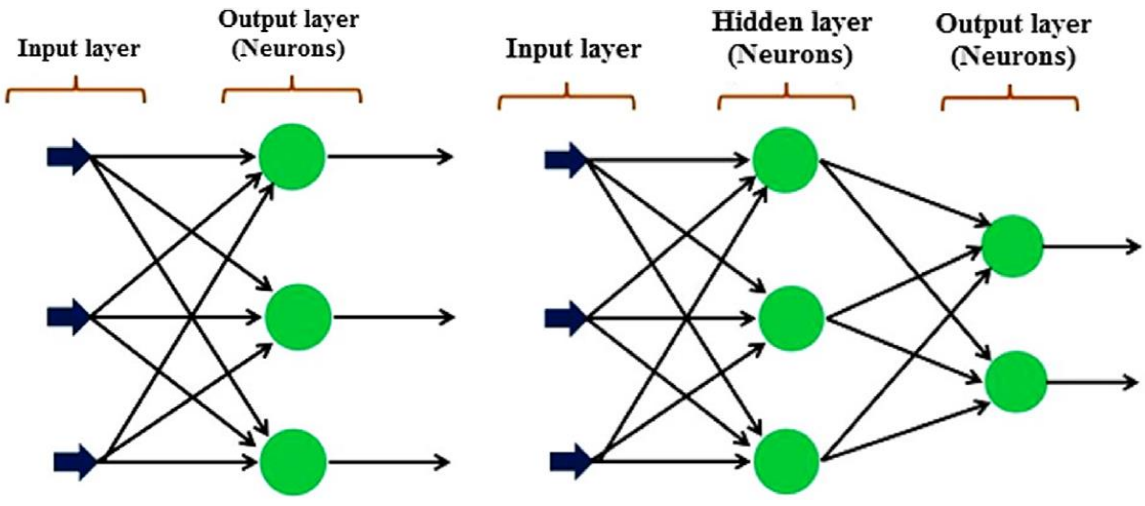
There are two classic black-box data-driven modelling methods. One is the artificial neural network (ANN) model, and the other one is the polynomial regression model. Both of them demand a mass of empirical and experimental data.

Even though the polynomial regression model can reflect relatively precise relationships between the inputs and outputs for engine steady-state operations, it cannot simulate the non-linearity in a global engine model very well, simply because the polynomial model is likely to result in the measurement errors caused by noise and outliers. By contrast, the ANN model is capable of capturing the non-linear behaviours much better than the polynomial regression model. Meanwhile, the ANN model needs less experimental data to construct the model than the polynomial regression model. Thus, the ANN model is usually regarded as a more effective data-driven modelling approach (Shamekhi and Shamekhi, 2015). However, training the ANN always requires plenty of data and skills to avoid overfitting and measurement errors, even for experienced users (Turkson et al., 2016).

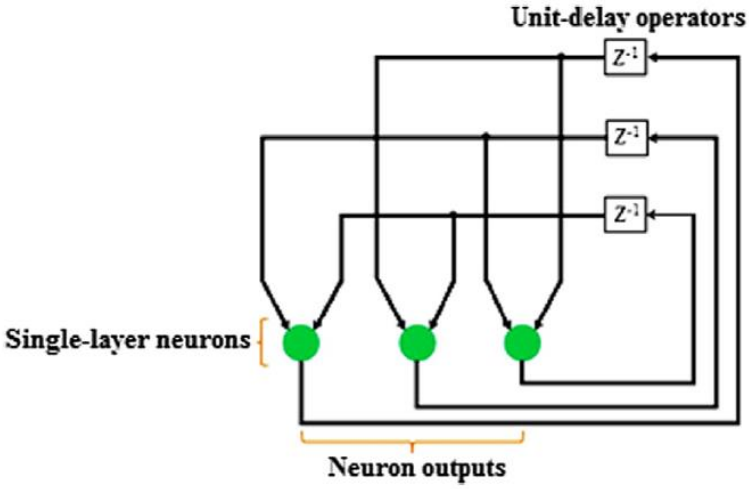
The development of ANN is inspired by the biological nerve system. Thus, an ANN usually contains a single neuron or multiple interrelated neurons just as the biological nervous systems. After the training, an ANN is able to approximate the input-output relationships and respond to a range of given inputs without any specific mathematical input-output

relationships. Basically, there are three parameters to construct a classic ANN model: the number of layers, the weight values and the bias values.

The ANN can be classified as two categories in general: feedforward neural network and recurrent neural network (RNN). Then the feedforward neural network can be further divided as single-layer perceptron network, multi-layer perceptron network and radial basis function network. Meanwhile, the RNN also comes in many variants, including the fully recurrent type, the Elman networks, the Jordan networks, the Hopfield networks as well as many other variants (Turkson et al., 2016). Compared with the feedforward neural network, the RNN can use its internal memory to process the past input and output values for the future prediction (Arsie et al., 2010).



(a) (b)



(c)

Figure 2.8 Artificial neural networks: (a) Single-layer feedforward neural network; (b) Multi-layer feedforward neural network; (c) Single-layer recurrent neural network (Turkson et al., 2016)

The ANN models have been widely used to resolve industrial problems and a well-trained ANN model can match many requirements for the engine calibration task, including the capability to handle the high input dimensionality; flexible optimisation of model parameters to avoid overfitting; reduction of the time and cost for calibration as just a few Design of Experiment (DoE) tests are required for feasible measurements; and the potential of online automatic optimisation without manual intervention (Turkson et al., 2016). Apart from the engine calibration, the ANN models have also been employed in many areas for SI engines, including the validation for novel control strategies, the application of On-Board Diagnostics (OBD), the identification of complex systems by multi-layer feedforward networks, and the identification and characterization of dynamic systems using single-layer RNNs with multiple inputs. Overall, the development of RNN makes the implementation of the neural networks relatively easier and widespread.

However, it requires a large amount of expertise to select a properly trained neural network for a specific application, especially the selection of architecture, because it varies for different cases. For example, the number of inputs required by different neural networks and different applications varies and should be determined on a case by case basis (Turkson et al., 2016). In addition, the ANN is born with two major defects. On one hand, when the system to be modelled becomes more complicated, the prediction efficiency and accuracy will decline accordingly. On the other hand, a non-physical model should never be expected to be as robust as a physical model, since the compatibility of a non-physical model is quite poor. Compatibility refers to the ability of a model to be compatible with multiple applications. A model is acknowledged to be compatible if it is designed for a certain application but can be run by other applications or under various operating conditions. The performance of a non-physical model, e.g. an ANN model, is constrained by its scope of training data. If the operation condition is far beyond the training scope or the model is supposed to be used for a new case, the model will have to be rebuilt completely. Thus, the non-physical model is not the best choice for control-oriented studies. The compatibility and robustness of the ANN can be improved by introducing the dynamic neural networks. But, the dynamic ANN has another two major drawbacks. Firstly, the required training time will be much longer. Secondly, the simulation program is more likely to get stuck because the number of hidden neurons will increase remarkably as the system to be modelled becomes more complicated (Shamekhi and Shamekhi, 2015). Moreover, since the non-physical model has ignored most of the engine dynamics, it is unable to simulate the fast transient performance precisely, such as the real-time response to engine transient acceleration and braking as well as the turbocharging system response (Millo et al., 2011).

2.3.5 Comparisons of Different Modelling Approaches

Actually, for the purpose of the development of control and calibration strategies, most modelling approaches mentioned above are capable of indicating the engine performance and pollutant emissions with desirable accuracy when they are well trained. However, the time costed by constructing and training the models is unpredictable. Besides, the computational speed for different models is significantly different as the Figure 2.9 shows.

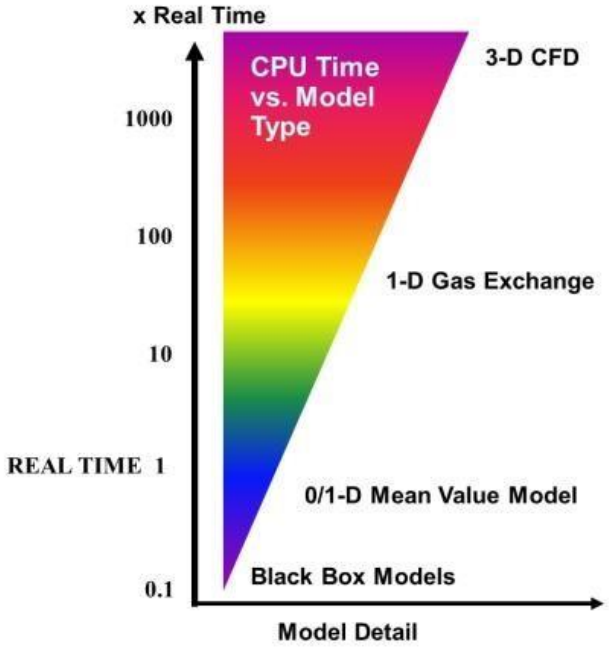


Figure 2.9 Comparison of different engine modelling methodologies concerning model detail and computational speed (Millo et al., 2011)

As can be seen from Figure 2.9, the time consumption of the CFD model can be thousands of times longer than that of the mean value engine model. As mentioned above, while CFD models are accurate but too slow, the mean value models are rapid but not accurate. Thus, the CFD and other detailed physical models are more suitable for the

study of engine geometry design, fuel and air fluid dynamics, and other theoretical researches. In the view of developing the engine management system, only the main features of engine performance are desired, such as IMEP, ISFC, torque etc. Therefore, the mean value engine model and black box model are selected as the control-oriented models.

However, none of these models can completely and accurately replace the real engines. With the constantly rising number of control parameters in modern engines, the use of models cannot satisfactorily address the issues of higher dimensionality problems. A highly accurate model can ensure that the calibration and control strategies are reliable for the real engine, whereas a poorly constructed model can lead to useless works. Then, the further tuning has to be requested by engineers to revise and improve those unfavourable works. The accuracy and generalization of the engine models play an extremely important role and increase the burden for engineers as well. Therefore, the best solution to achieve the trade-off among reliability, compatibility and computational speed is to develop the intelligent online dynamic, or termed non-model-based, calibration and control strategies.

2.4 Artificial Intelligence Methods

As mentioned above, the conventional model-based calibration and control strategies have reached their limitations as they are time-consuming and labour-intensive. Meanwhile, the calibration and control strategies for modern ICEs have to be faced with the trade-off among multiple objectives, input saturation, short transient behaviours, uncertainties and nonlinearities. In order to break through the limitations and fulfil various

engine operation requirements, the conventional engine management system has to be combined with artificial intelligence methods, including multi-objective meta-heuristic algorithms and fuzzy knowledge based control strategy. Thanks to these artificial intelligence methods, the automatic, intelligent, rapid, robust calibration and control strategies for both engine steady-state and transient performance are able to be developed.

2.4.1 Multi-objective Meta-heuristic Algorithms

As introduced in section 1.4, the engine management system or ECU is brain of an engine which manages a variety of actuators by reading the sensors and using multidimensional performance maps. The entire engine management system can be divided into a great many of sub-systems, including the speed control, charge control, AFR control, sparking control, fuel supply control and so on. Accordingly, the engine calibration is also conducted on a series of sub-systems for massive adjustable variables. The engine steady-state calibration has to consider the engine performance, fuel economy and polluted emission jointly. Whereas, for the engine transient calibration and control, both the process and results have to be evaluated. All of these make the design of engine management system become a complex multi-objective optimisation problem (MOP). The trade-off relationships among various optimisation objectives and various constraints are difficult to achieve (Tayarani-N et al., 2015).

Actually, the MOP is much more widespread in industrial applications than the single-objective optimisation problem, because most engineering systems are designed with several criteria or objectives, and these objectives are in conflict with each other. In other

words, the adjustment of one objective may significantly affect other objectives in most practical cases. It is really difficult to optimise all the objectives simultaneously. Therefore, the artificial intelligence methods are widely adopted for MOPs in order to automatically achieve a compromise among all the objectives. Generally, a MOP contains a variety of objectives, accompanied by a series of inequalities and equality constraints. It can be formulated mathematically as:

minimise / maximize / regulate

$$f(x) = \big(f_1(x), \dots, f_i(x)\big), i = 1,2, \dots, n \tag{2.2}$$

subject to:

$$\begin{cases} g_j(x) \leq 0, j = 1,2, \dots, J \\ h_k(x) \leq 0, k = 1,2, \dots, K \end{cases} \tag{2.3}$$

where n in Equation (2.2) is the number of optimisation objectives; $f_i(x)$ is the objective function evaluated at x and $i = 1,2, \dots, n$; $x = \{x_1, \dots, x_p\}$ is a vector of decision variables and p is the number of decision variables; $g_j(x) = j^{th}$ is the inequality constraint evaluated at x; while $h_k(x) = k^{th}$ is the equality constraint evaluated at x.

Broadly speaking, the meta-heuristic algorithms are a kind of searching method based on the bionics. They imitate the mechanisms of biological behaviours in nature, design the intelligent computing systems, and solve the complex computational problems based on the iterative calculations, by randomly or approximately randomly searching for the optimum solutions within the non-linear complex spaces. In addition, meta-heuristic algorithms can even work effectively in situations where no reasonable details of the objective functions versus decision factors exist. Moreover, given that most meta-heuristic algorithms are population-based algorithms, their diversity allows them to be

more likely to find the globally optimum solutions and achieve the trade-off among various fitness criteria. These population-based meta-heuristic algorithms are also named as global searching meta-heuristic algorithms. Thus, the meta-heuristic algorithms are very effective for resolving the multi-objective optimisation problem with multiple constraints.

The general approach using meta-heuristic algorithms for a multi-objective optimisation problem involves populating a number of feasible solutions, aggregating the individual solutions into locally or globally optimal solutions, and among these are a set of final solutions with best fitness values. To be a little bit more precise, multi-objective meta-heuristic algorithms follow the fundamentals of single-objective meta-heuristic algorithms. The optimisation process begins with a random set of individuals that are going to be constantly updated from generation to generation. Each individual is a potential solution which is usually represented by a vector of floating-point values. During each successive generation, each individual is evaluated based on the measurement of its corresponding system performance signals and a fitness function in reference to the desired objectives (evaluative criteria). At the end of each successive generation, individuals with higher fitness values will be retained. Whereas, individuals with lower fitness values will be ranked behind and may be discarded at any time. Hence, a new set of prospective generation of solutions are introduced, through the selection of the best individuals from the previous generation up to now and a series of transformations. The number of individuals remains constant for each successive generation. The optimisation process will be repeated until the pre-set number of iterations or the boundary conditions are reached (Ma, 2013).

Since the vector evaluated genetic algorithm (VEGA) was invented by Schaffer in 1985, a great variety of meta-heuristic algorithms have been developed in succession during

the past few decades. Each of these algorithms is based on different natural or biological behaviours, has its own characteristics, and has found its foothold in the industrial applications (Tayarani-N et al., 2015). The genetic algorithm (GA) and the particle swarm optimisation (PSO) algorithm are two major categories of the evolutionary computation-based meta-heuristic algorithms. The GA was proposed by American scientist John Holland in 1960. It is inspired by the concept of evolutionism, i.e. the survival of the fittest in natural selection, through mutation and crossover. Then, the PSO was introduced by American researchers Kennedy and Eberhart in 1995. It is intended for simulating the flocking behaviours of social organisms in groups, such as bird and fish schooling. Since these algorithms were developed, they have been widely applied for a broad range of subjects, such as the engineering, ecology, economics, sociology and so on. Nowadays, thanks to their good applicability and effect, the meta-heuristic algorithms have been introduced to the automotive industry. In author's research, these two classic meta-heuristic algorithms, GA and PSO, are applied for the design and optimisation of the engine management system.

2.4.2 The Fundamental Generation of Meta-heuristic Algorithms

a) Genetic Algorithm

The GA is a well-known meta-heuristic algorithm for solving both constrained and unconstrained optimisation problems. It is based on the concept of biological evolution. The GA-based optimisation process begins with a population of random individual solutions and repeatedly modifies the populations. At each successive generation, the GA selects the fittest individuals from the current population to be the parents and uses them to produce the offspring for the next generation. Over successive generations, the

better population will replace the worse population and "evolves" toward an optimum solution (Câmara, 2015). The offspring is calculated based on three stages: selection stage, crossover stage, and mutation stage. The selection stage is accomplished in accordance with the dominance relationship, i.e. one individual is regarded to dominate the others when it is no worse in all objectives or strictly better in at least one objective (Câmara, 2015). Thus, the GA is able to effectively resolve both single-objective and multi-objective optimisation problems. A simple pseudo code of GA can be expressed as:

Generate an initial population;
Evaluate fitness value of individuals in the population;
Do:
    Select parents from the population;
    Recombine (mate (crossover and mutation operators)) parents to produce children;
    Evaluate fitness of the children;
    Replace some or all of the population by the children;
Repeat until a satisfactory solution has been found.

Figure 2.10 gives an example for the dominance relationship in respect of an optimisation problem with two objectives:

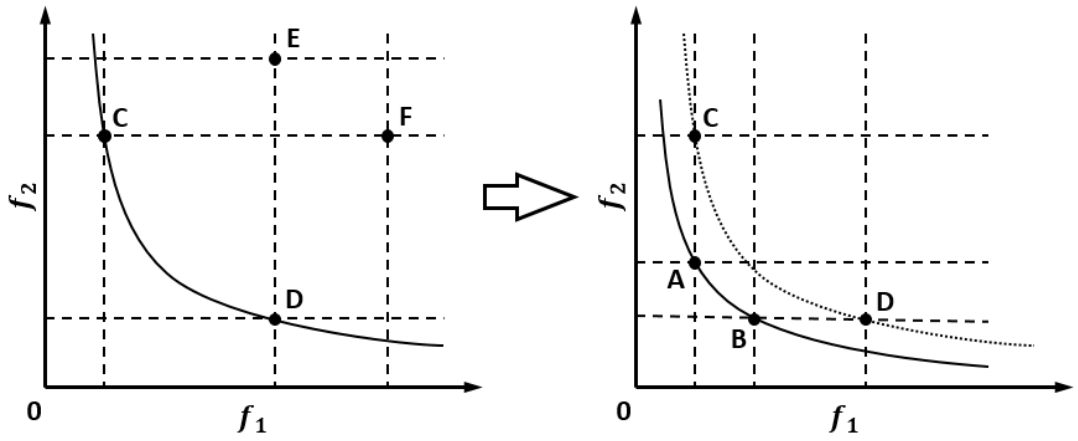
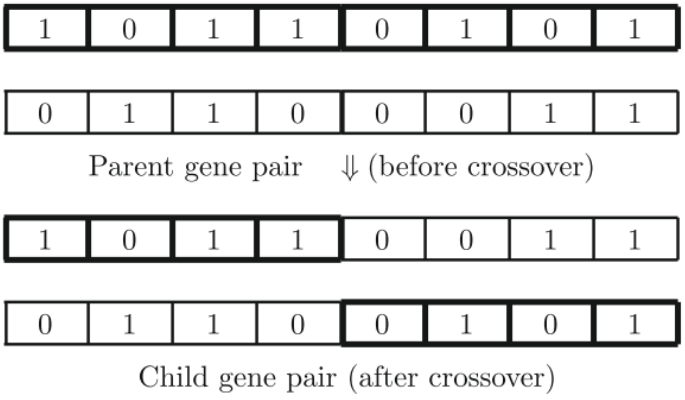


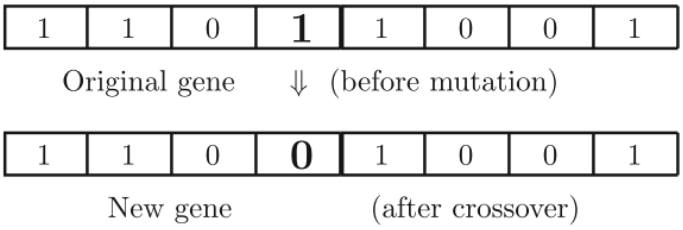
Figure 2.10 An example for the dominance relationship in respect of two optimisation objectives (Ma, 2013)

As can be seen from Figure 2.10, individual C dominates individuals E and F, while individual D dominates individuals E and F. Individuals C and D are not dominated by each other and they are both non-dominated solutions. The boundary generated by the set of all non-dominated solutions is called the Pareto-optimal front. After the iteration, individual A dominates individuals C and D while individual B dominates individuals C and D. Now, individuals A and B do not dominate each other and they generate the new Pareto-optimal front. So in other words, over successive iterations, the solutions on the Pareto-optimal front are the optimisation results. Figure 2.10 uses four points to interpret the dominance relationship, but the Pareto-optimal front is actually generated by the set of all non-dominated solutions, namely more points.

The crossover and mutation stages for the re-production of offspring at each iteration are actually random operations as shown in the figure below:



(a)



(b)

Figure 2.11 An example for the crossover and mutation of the GA: (a) crossover; (b) mutation (Yang, 2014)

The crossover and mutation can also be expressed mathematically as:

$$\begin{cases} \textbf{crossover}: \quad x(t+1) = floor\left(\dfrac{x(t)}{10^k}\right) + 10^{floor(\log(x(t)))} \cdot mod1\left(\dfrac{x(t)}{10^k}\right) \\ \textbf{mutation}: \quad x(t+1) = floor\left(\dfrac{x(t)}{10^j}\right) + 10^{floor(\log(x(t)))} \cdot mod1\left(\dfrac{x(t)}{10^j}\right) + \varepsilon \end{cases} \tag{2.4}$$

where $x(t)$ is the solution for t^{th} iteration; $floor$ is the function to acquire the integer part; $mod1$ is the function to acquire the decimal part; ε is the random action.

A lot of studies have employed the GAs to optimise the engine performance. For example, Liu et al. use GA to optimise the injection system and combustion chamber shape for a diesel/natural-gas dual fuel engine in order to minimise the fuel consumption and

emissions (Liu et al., 2018). Shibata et al. use GA to calculate the optimal heat release shape for a supercharged single-cylinder diesel direct injection engine, in order to reduce the combustion noise by multiple fuel injections while maintaining the thermal efficiency (Shibata et al., 2019).

b) Particle Swarm Optimisation

The PSO is another most popular meta-heuristic algorithm, simulating the foraging or flocking behaviours of birds and fishes. It is similar to the GA and moves a swarm of individuals “flying” within the searching space to look for the potential solutions. The PSO-based optimisation process is also initialised with a swarm of random particles. Each particle in the swarm represents a potential solution. Just like a bird in its flock, it is assigned with a location and a velocity. During each iteration, the fitness value of each particle at its current location will be calculated and compared with its previous best fitness value. If the current fitness value is better, then it becomes the new personal best location. As mentioned above, the diversity of the individuals allows them to convey their information with each other and affect their next action. After comparing the personal best location with each other, the global best location will be obtained and updated after each iteration. Meanwhile, the velocity of each particle is calculated and updated concerning its current velocity, its personal best location in the swarm, and the global best location up to now. Thus, the location and the velocity of the particles will be constantly updated, until the final globally best solution is found or the termination criteria are met (Slowik and Kwasnicka, 2018; Netjinda et al., 2015).

The velocity of each particle is adjusted in each iteration based on equation:

$$v_{(t+1)}^{i} = Current\ motion + Cognitive\ component + Social\ component$$
$$= wv_{(t)}^{i} + \alpha r_1\left(p_{(t)}^{i} - x_{(t)}^{i}\right) + \beta r_2\left(G_{(t)}^{i} - x_{(t)}^{i}\right) \quad (2.5)$$

The cognitive component indicates the perception that how each particle finds its best location, while the social component refers to the perception of the globally best location that the entire swarm is able to find after sharing each other's searching results. In the above equation, $x_{(t)}^{i}$ represents the ith particle's current location at t^{th} iteration, while $x_{(t+1)}^{i}$ indicates the i th particle's future location at $(t+1)^{th}$ iteration. $v_{(t)}^{i}$ is the i th particle's current velocity, while w is the inertia weight in respect of how the current velocity will affect the future velocity. $p_{(t)}^{i}$ indicates the ith particle's personal best location at t^{th} iteration, while $G_{(t)}^{i}$ is the globally best location in the swarm up to now. Besides, α and β are the scaling factors for the cognitive behaviour and social behaviour respectively. They are used to adjust the weight of the particle's current best location and the globally best location. r_1 and r_2 are both the random number generated uniformly within the range [0, 1] in order to make full use of the entire searching space and promote the probability of finding the globally optimal solution.

Then, the movement of any particle is calculated by adding the velocity to its current location as the following equation shows:

$$x_{(t+1)}^{i} = x_{(t)}^{i} + v_{(t+1)}^{i} \quad (2.6)$$

Based on above equations, Figure 2.12 shows how the velocity and position update in particle swarm optimisation:

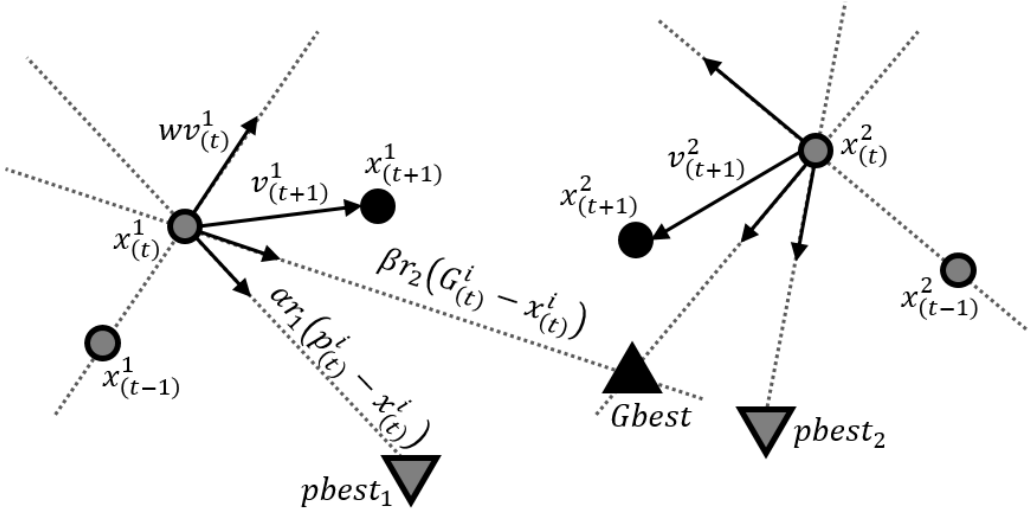


Figure 2.12 An example for the velocity and position updates of the PSO (Bryson et al., 2016)

A simple pseudo-code for the PSO is given below:

Create a swarm with particles;
Initialize the position and velocity of each particle randomly;
Calculate the fitness value of each position;
Calculate the locally and globally best position for particles;
Repeat:
  Update the velocity of each particle;
  Update the position of each particle;
  Calculate the fitness value of each particle;
  Update the locally and globally best position for particles;
Until stopping conditions true.

The PSO algorithm has many benefits, such as simple program, easy operation, fast convergence speed, low computational cost and so on (Deng et al., 2019). Therefore, it is easier to be conducted than other meta-heuristic algorithms and has been widely used to solve the multi-objective optimisation problems for engines. For example, Keshavarz

et al. used the PSO algorithm to calibrate the turbocharger efficiency and the heat transfer multiplier for a spark ignition engine transiently (Keshavarz et al., 2018). Wu and Gao used the PSO algorithm to calibrate various parameters for a diesel engine with multiple injections (Wu and Gao, 2018). The experimental results show that PSO could improve the calibration efficiency and results with faster convergence speed, better accuracy and better reliability.

2.4.3 The Improved Generation of Meta-heuristic Algorithms

a) Strength Pareto Evolutionary Algorithm and Strength Pareto Evolutionary Algorithm 2

The strength Pareto evolutionary algorithm (SPEA) is developed by Swiss computer scientists Eckart Zitzler and Lothar Thiele in 1999 (Zitzler and Thiele, 1999). It is a milestone in the history of GAs. SPEA is designed based on the fundamental GA theory mentioned above and upgraded techniques to achieve multiple Pareto-optimal solutions in parallel.

The SPEA utilizes not only a population as mentioned in the previous section, but also an external archive to store and maintain the non-dominated solutions, i.e. a set of Pareto-optimal solutions. This external archive is separated from the regular population during the entire optimisation process. At the beginning of the optimisation process, a random population and an empty archive are initialized. After the evolutionary process is started, all the Pareto-optimal solutions will be copied to the external archive. Meanwhile, any dominated solutions will be eliminated from the archive over successive generations. If the number of the Pareto-optimal solutions in the archive

exceeds a given maximum capacity, the archive size will be reduced by deleting further archive members through means of clustering. Then, fitness values of solutions in both the current population and the external archive are calculated. Solutions with better fitness values will be sent to the mating pool. Unlike the other fundamental GAs, the SPEA determines the dominance relationship by how many individuals in the population are dominated by each non-dominated solution in the archive. Hence, solutions in the archive are much more likely to be picked to the mating pool than any other individuals in the population. Eventually, crossover and mutation operations are applied. A new generation of solutions (offspring) is produced and the optimisation process starts over again.

However, studies have found that the SPEA may accidently delete non-dominated solutions and get into a constant fitness value which limits the diversity of the optimisation results (Abido, 2006). To make up for the above shortcomings, the strength Pareto evolutionary algorithm 2 (SPEA2) is developed as an advanced version of SPEA (Zitzler et al., 2001).

When calculating the fitness values, the SPEA2 considers not only the dominance relationships but also the density of solutions, in order to sort the Pareto-optimal solutions in more details. When the number of the Pareto-optimal solutions in the archive exceeds its maximum capacity, the SPEA2 will apply the K-nearest neighbour density distribution method to assign the exceeded solutions and maintain the diversified solutions over the current Pareto-front. Meanwhile, when deleting the undesirable solutions from the archive, the means of clustering is improved by an interception strategy in order to better retain the boundary solutions. Moreover, the size of the external archive for SPEA2 is fixed. Specifically speaking, if the number of the

Pareto-optimal solutions in the archive is less than its maximum capacity, the SPEA2 will fill up the archive with dominated individuals (Zitzler et al., 2001). Thanks to the above improvements, the SPEA2 is allowed to provide diversified solutions while maintaining the convergence speed.

The settings of SPEA2 will be introduced in details in section 4.2.2 on the basis of the specific calibration task. It has been employed for the calibration of both HCCI engine and GDI engine. Simulation and experimental studies have verified its improvements on both fuel consumption and pollutant emissions (Ma et al., 2015; Ma et al., 2017; Ma et al., 2019).

b) Accelerated Particle Swam Optimisation and Chaos-enhanced Accelerated Particle Swarm Optimisation

As for the PSO algorithm, in order to further improve its convergence speed, the accelerated particle swam optimisation (APSO) algorithm has been developed by Yang (Yang, 2008). As introduced in section 2.4.2, the conventional PSO algorithm considers both the personal best location and the global best position to determine the movement of particles. The involvement of the personal best location aims to enhance the diversity of the solutions. For the purposed of simplifying the program and improving the convergence speed, the personal best location can be replaced by a randomization function. Then, the diversity of the searching behaviours can be maintained while only the global best solutions need to be considered (Gandomi et al., 2013; Yang et al., 2017). Accordingly, a simpler formula for the velocity vector can be expressed as:

$$v_{(t+1)}^{i} = v_{(t)}^{i} + \alpha r_{(t)}^{i} + \beta\left(G_{(t)}^{i} - x_{(t)}^{i}\right) \tag{2.7}$$

where $\alpha r_{(t)}$ is the randomization component which is used to replace the cognitive component. Actually, the effect of this velocity function is essentially the same as the Eq. (2.5) but without the cognitive component. To increase the convergence speed even further, the location of each particle can be updated through a single step:

$$x_{(t+1)}^{i} = (1-\beta)x_{(t)}^{i} + \beta G_{(t)}^{i} + \alpha r_{(t)} \tag{2.8}$$

This equation simplifies the movement without the requirement of velocity. Hence, the APSO algorithm can result in faster convergence while retaining the same order of convergence. The randomization component $\alpha r_{(t)}$ here allows the solutions to break away from the fetters caused by any local optimal results. r is a random number drawn from [0, 1] while α should be selected in respect of the scales of the variables of interest. $\beta \in [0, 1]$ is the attraction parameter which is employed to regulate the convergence speed. Compared with the conventional PSO algorithm, the APSO algorithm requires only two parameters (α, β), which definitely simplify the program. It is worth mentioning that, to reduce the randomness over successive iterations, a monotonically decreasing function can be used to replace the $\alpha r_{(t)}$ (Gandomi et al., 2013; Yang et al., 2017).

The conventional APSO algorithm sets β as a constant because some studies have verified that it could allow algorithm to work effectively. However, a varying β value in each iteration is actually advantageous, which can achieve a better trade-off relationship between the convergence speed and solution diversity (to avoid being stuck in local optimum results). Thus, the chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm which combines the APSO algorithm with the chaotic mapping method has been developed (Gandomi et al., 2013; Yang et al., 2017).

The CAPSO algorithm employs the chaotic mapping method to adjust β value after each iteration. The logistic mapping method is a classic chaotic mapping method (Quan et al., 2017). The β value is thus updated following the equation:

$$\beta_{(t+1)} = \varphi \cdot \beta_{(t)} \cdot (1 - \beta_{(t)}) \tag{2.9}$$

The chaotic mapping method can produce some accidental changes during the process of the particle movement and randomly select some poor solutions to participate in the next iteration, in order to help the particles avoid being trapped by locally optimum solutions. It is like the mutation stage for the GAs.

The settings of the CAPSO algorithm will be introduced in details in section 4.2.3 on the basis of the specific calibration task. The CAPSO algorithm has been widely employed for the intelligent calibration and control in the vehicular industry because of its simple program, easy implementation and fast convergence speed. Quan et al. applied the CAPSO algorithm for the component sizing of a series hybrid electric powertrain (Quan et al., 2017). Zhang et al. used the CAPSO algorithm for the transient calibration of the air-path control of a diesel engine (Zhang et al., 2018). These studies verify that the CAPSO algorithm can find better globally optimal solutions with faster convergence speed and higher reliability.

2.4.4 Fuzzy Knowledge Based Control

In the control theory, the fuzzy knowledge based controller (FKBC) is a well-known non-model-based control technique which is able to fix all the problems listed in the section 2.2.2 (Wang et al., 2015; Mendes and Neto, 2015). It imitates the human behaviours, follows the genetic “if-then” rules, stays independent of the system structures, and makes

decisions under different circumstances. It can work efficiently when an accurate model is not available (Liu et al., 2017; Askari et al., 2017). The other main advantages of the FKBC includes: improvement of the system stability; high robustness; simple structure; easy tuning; no requirement for the historical data and training; high self-adaptive capability; excellent tracking performance; be able to figure out the problem of the black-box system (Liu et al., 2017; Athari and Ardehali, 2016). Compared with other advanced but complicated control strategies, FKBC can be utilized and understood easily by operators even without abundant experience or knowledge about the target control plants (Liu et al., 2017; Shahnazi, 2016). Meanwhile, FKBC can present more smooth results than common control methods in reason (Senouci and Mellouk, 2016). The essence of the FKBC is the remarkable combination of a fuzzy adaptive controller and a robust compensator (Liu et al., 2017). It gives FKBC the pioneering control effect on unknown non-linear systems.

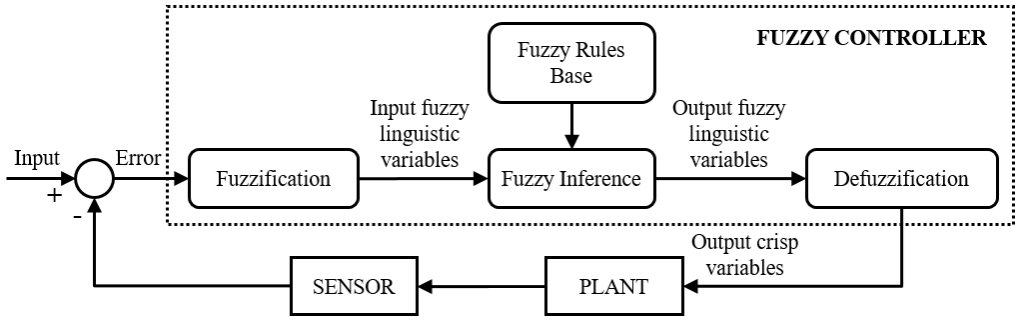


Figure 2.13 Structure of a FKBC (Lu et al., 2017)

The concept of the fuzzy control is proposed by Zadeh in his famous study “Fuzzy Sets” in 1965. After years of development, the widely used structure of a FKBC can be found in Figure 2.13. Basically, there are four stages of the FKBC: fuzzification, fuzzy inference, fuzzy rules base, and defuzzification. The fuzzification stage translates the real scalar

values into the fuzzy linguistic variables based on the input membership functions. The fuzzy rule base stores the genetic “If-Then” rules with conditions and their corresponding conclusions. The fuzzy inference is the stage of “making decisions”. It constructs the output fuzzy linguistic variables based on the fuzzy rules and input fuzzy linguistic variables. Finally, the defuzzification stage converts the output fuzzy linguistic variables into the numerical outcome which is the actual action to the target system. The most common and effective method for the defuzzification is the centre of gravity method. The output signal provided to the target system is the direction and the next-step move distance for each control signal (Liu et al., 2017; Askari et al., 2017).

The FKBC system has been developed from the single variable system (Salehi and Shahrokhi, 2012) to the multiple variable systems (Cui et al., 2019; Zhang et al., 2019). Based on the number of input variable, the FKBC can be classified as one-dimension, two-dimensions, and multiple-dimension controllers. Among these, the two-dimension FKBC is the most widely used FKBC, where the error signal e_p and the rate of the change of error e_v are input words:

$$\begin{aligned} e_p(k) &= H_{ref}(k) - H(k) \\ e_v(k) &= \frac{e_p(k) - e_p(k-1)}{T} \end{aligned} \tag{2.10}$$

where k is the actual discrete time, T is the sampling time, H is the actually measured signal, and H_{ref} is the reference value.

During the recent decades, the FKBC has been widely used for the engine control system. Cheng et al. use the FKBC scheme to determine the appropriate positions of variable geometry turbocharger vanes and exhaust gas recirculation valve of diesel engines in real-time (Cheng et al., 2018). According to the simulation results, the FKBC can improve

the engine's efficiency and the total turbo efficiency while reducing the NOx and soot emissions compared with the conventional PID controller. Mu et al. develop an adaptive FKBC for the non-linear throttle control of GDI engines (Mu et al., 2018). The simulation studies verify that the proposed controller can regulate the throttle opening to track the input rapidly and precisely with favourable robustness and reduced buffeting. The settings of the FKBC will be introduced in details in section 5.3 on the basis of the specific control task.

2.5 Air/Fuel Ratio Control System

The Air/Fuel Ratio (AFR) control system is one of the most important control modules in the EMS since the AFR has to be maintained within a narrow band around the stoichiometric value (14.7) for both steady and transient operations, in order to achieve the optimal trade-off relationship among the power output, fuel economy and the efficiency of three way catalytic converter (TWC) (Heywood, 1988; Stone, 2012). Short excursions of the AFR can be tolerated but long periods of oscillations will impair the thermal efficiency, catalyst efficiency, and even affect the system security. A rich AFR mixture will result in terrible fuel economy and increased emissions, while a lean AFR mixture will cause a much hotter burn, potentially hot enough to melt pistons and spark plugs, and damage the engine's internals. Thus, an effective AFR control system is highly demanded to maintain the stability and security of an engine.

Generally, the AFR control system includes both feedforward and feedback control loops. The feedforward control loop is supposed to generate a reference injected fuel mass in order to avoid time delay and large deviation of AFR. Meanwhile, the feedback control

loop should regulate the AFR upstream of the TWC as close as possible to the stoichiometric value, based on the feedback signal measured by the Exhaust Gas Oxygen (EGO) sensor, also termed lambda sensor (Ashok et al., 2016). The scheme of AFR control system in SI engines can be found in Figure 2.14.

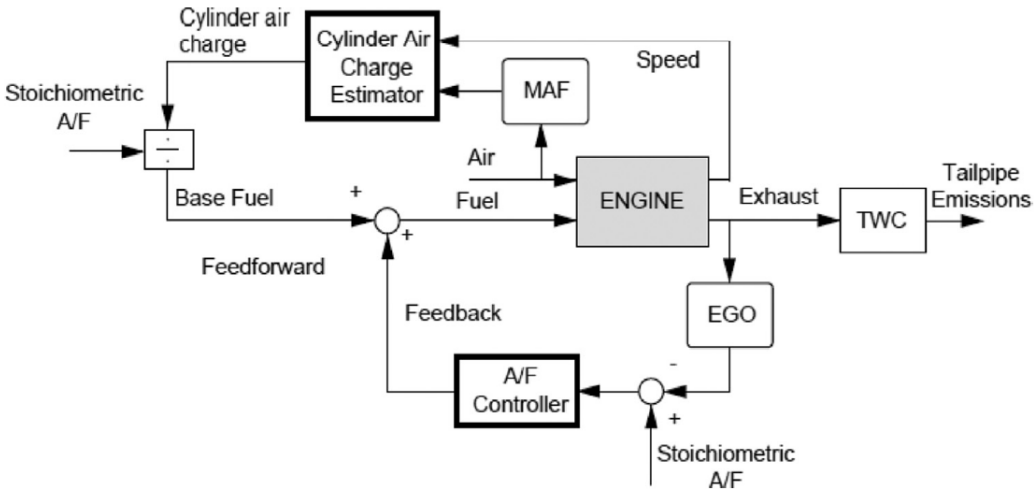


Figure 2.14 Air fuel ratio control module with the feedforward and feedback elements in a SI engine (Ashok et al., 2016)

To improve the transient performance of the AFR control system, many researchers have worked on the feedforward control loop by employing the MPC strategy but neglected the development of the feedback control loop. As for the MPC, a prediction model of the air and fuel path dynamics combined with appropriate compensations is essential. The controller calculates a reference injected fuel mass based on the intake air flow in advance as per the driver torque demand. The prediction model is trained by means of experimental study during the calibration phase of the engine development. Wang et al. first apply the MPC based on an adaptive neural network model for the AFR control (Wang et al., 2006). Two years later, Wang et al. use the adaptive radial basis function network to develop a parameter estimation strategy for the AFR control (Wang and Yu,

2008). Then, Zhai et al. improve the MPC method by investigating the Diagonal Recurrent Neural Network (DRNN) in order to improve the robustness (Zhai et al., 2010). Moreover, Sardarmehni et al. propose an improved non-linear MPC method based on Multi-Layer Perceptron (MLP) neural network resulting in smoother variations of the injected fuel (Sardarmehni et al., 2013). In recent years, Wong et al. utilize the Online Sequential Extreme Learning Machine (OSELM) to reduce the AFR deviation and overshoot (Wong et al., 2016). However, as mentioned in previous sections, the neural networks have many limitations and drawbacks for their further applications. Firstly, they are too complicated to be embedded into a microcontroller. Secondly, the physical meaning of each parameter is not clear enough. Furthermore, these approaches usually require a lot of training to obtain the optimal results. In terms of the MPC strategy, it is difficult to predict the non-linear relationships inside the controller and its transient performance is not optimistic. In addition, it also costs too much time to train the models. To simplify the feedforward control loop, many researchers estimate the injected fuel mass by using the mass air flow (MAF) sensor, which measures the intake air mass directly (Ashok et al., 2016). This method is widely used especially for GDI engines since the fuel is injected directly into the cylinder and the fuel film can be ignored (Ashok et al., 2016).

In the actual engine operations, due to the inaccurate calculation of the air charge, the modelling deficiencies, the variations of environment and fuel composition, as well as the mechanical issues, there will be errors in the feedforward control loop. Then, the feedback control loop will compensate the unavoidable errors by adding or reducing the fuel injection quantity based on the lambda sensor readings. Hence, the feedback control loop is the key to ensure the accuracy of AFR control system. Speaking of the feedback control loop, most of the commercial EMSs still use a lookup-table-based PID controller to

regulate AFR because of its simple structure, easy implementation, no request of a model and many benefits listed in section 2.2.2. But the lookup-table-based PID controller works on the basis of a gain-scheduling approach. As mentioned in section 2.2.2, this control method is time-consuming for calibration and is not robust to highly non-linear systems, time varying dynamic systems, and the particularly complex systems with uncertain parameters or unclear relationships (Li et al., 2019a; Song et al., 2017). In this case, the AFR control system becomes more and more challenging due to the variety of engine operating regimes, complexity of physical and chemical processes inside the engine, uncertainties, disturbances, and non-linear dynamics. But it also becomes a perfect platform to validate the novel transient calibration and control strategies. Accordingly, further studies should simplify the AFR control system by employing a self-adaptive, fault-tolerant and effective control strategy.

2.6 Summary

This chapter reviews the publications in respect of the engine calibration and control strategies, engine numerical simulation methods, artificial intelligence methods, and AFR control system. As the thesis title “online dynamic optimisation for the engine management system using artificial intelligence methods” indicates, the findings from the literature reviews support the motivations for the following areas:

1. “Online dynamic optimisation”:

   Due to the high complexity, high degrees of freedom, system uncertainties, unknown dynamics, unknown disturbances, high nonlinearities and short transient behaviours, it is almost impossible to generate a sufficiently accurate control-

oriented engine model with the favourable computational speed. The conventional model-based calibration and control strategies, whose effect replies on the engine models, are reaching their limitations. Thus, the online dynamic optimisation methods, including the non-model-based calibration strategy and non-model-based self-adaptive controller are highly demanded.

2. “Engine management system”:

   The engine management system consists of numerous lookup-tables. The model-based calibration strategies for these lookup-tables are time-consuming, labour-intensive and not robust. Thus, the non-model-based calibration strategy is highly demanded to automatically locate the optimum calibration results with higher efficiency for both steady-state and transient operations.

   The conventional lookup-table-based PID/PI controller, which works linearly based on the calibration results stored in the lookup tables, is also reaching its limitation. The advanced non-model-based control strategy with self-adaptive capability, high robustness and simple structure should be developed to reduce the calibration cost and improve the control performance.

3. “Artificial intelligence methods”:

   The artificial intelligence methods can be employed to improve the automation level for the engine management system and reduce the cost for calibration. The SPEA2 and CAPSO algorithms can be used for the automatic multi-objective calibration process. The fuzzy knowledge based control method can be used to improve the self-adaptive capability and robustness for the non-model-based controller.

The AFR control system becomes more and more challenging due to the variety of engine operating conditions, complex processes inside the engine, parameter uncertainties, and non-linear dynamics. It requires an efficient and effective non-model-based feedback controller.

CHAPTER 3 EXPERIMENTAL SET-UP

This chapter is to introduce the experimental set-up, including the hardware and the software. A GDI engine test bench has been constructed at the University of Birmingham for the development and validation of the engine calibration and control methodologies. Due to the missing hardware, missing software and various problems appearing on the test bench before and during the experimental studies, the engine test bench, especially the ETAS RCP test platform, has undergone a series of re-configuration, re-calibration, components' repair and replacement by author with the technical supports from Jaguar Land Rover Ltd., ETAS Ltd., technicians of university laboratory as well as previous PhD students and research fellows.

The whole test bench, including the engine, dynamometer, accelerator pedal, measurement devices and other conditioning systems, is controlled by the ETAS system. It is tasked with the test bench control and the data acquisition. The author's main job is to re-configure the test bench and re-calibrate the associated parameters to cope with new engine test tasks.

3.1 Experimental Platform

This section is to introduce how to set up an experiment test in order to validate the proposed control and calibration methods, based on the ETAS RCP system (both hardware and software) as well as the engine test bench.

3.1.1 Investigation Approach

The figure below shows the basic process for the development of new control and calibration methods.

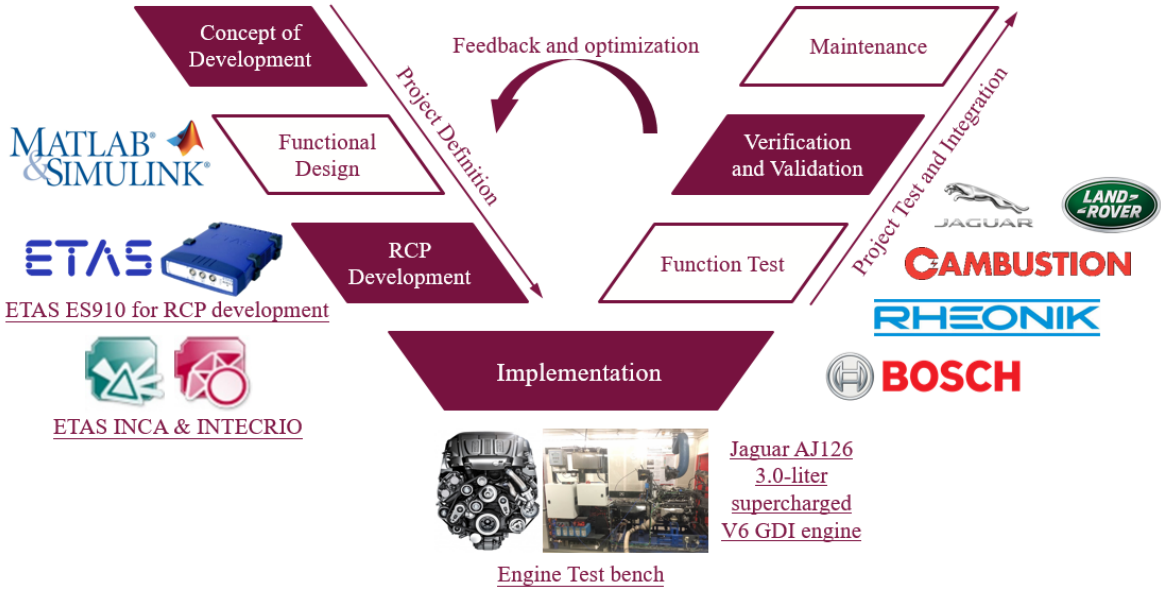


Figure 3.1 Process for the development of new control and calibration methods

The project definition is almost based on the software, including the concept of development, functional design based on the MATLAB/Simulink, and RCP test development based on the ETAS INTECRIO and INCA. Next step is the implementation of these new control and calibration methods on the real engine test bench, followed by the function test, validation, as well as the maintenance. The control and calibration methods will be further revised based on the feedback from the experimental tests.

3.1.2 Operation Platform

As for the RCP test platform, the data transfer between the ETAS hardware and software is shown below:

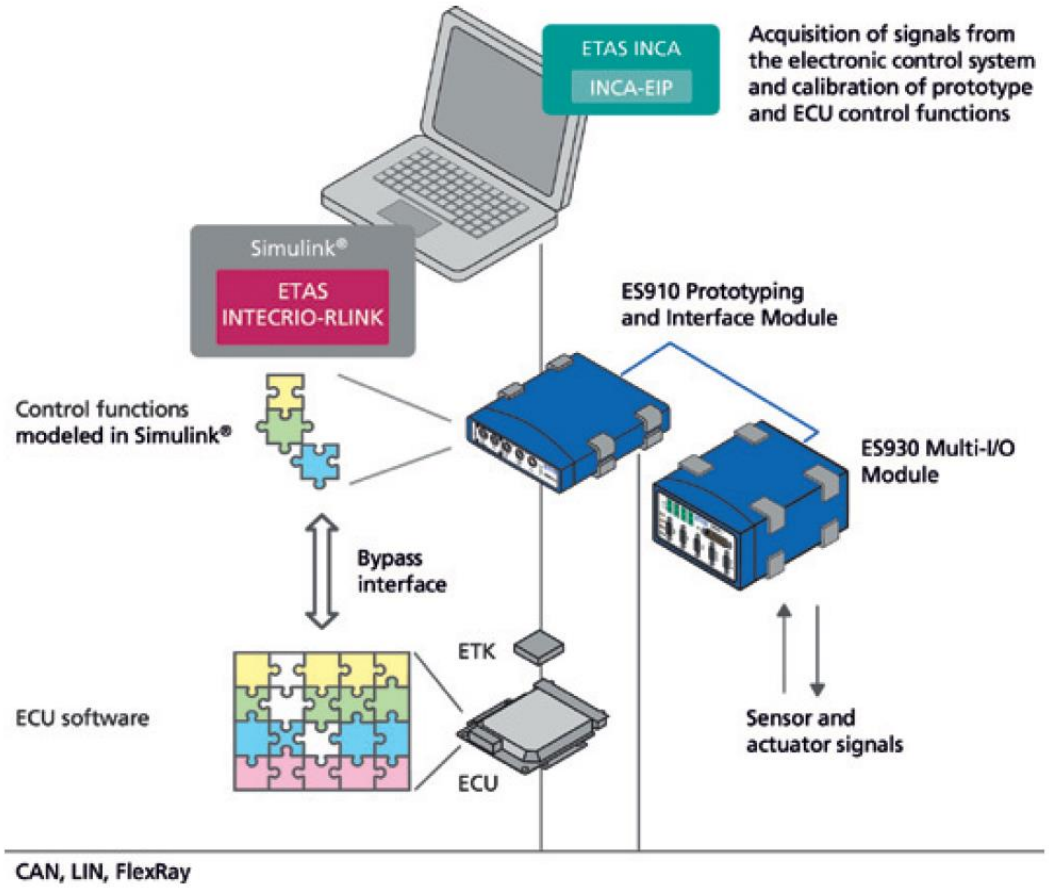


Figure 3.2 Data transfer between the ETAS hardware and software (ETAS, 2012)

The ES910 is a real-time prototyping processor which achieves the signal communications between the PC and the engine. In terms of the optimisation for the engine management system, the ES910 will operate the modified engine management system through the ETK bypass interface while running the unmodified engine management system through the ECU. Speaking of the external sensors, their analogue signals are measured by the ES930 multi-I/O module first. Then, the ES930 will transform the analogue signals to digital signals and send them back to the ES910.

The software ETAS INCA installed in the PC offers an operation interface for the communications between the ES910 and the engineer. After the Simulink-based controller is converted to the executable program and integrated with further prototyping functions by the software ETAS INTECRIO, the INCA is able to access the controller functions and operate the measurement and calibration.

The ETAS hardware and software will be introduced in details in the next sections.

After integrating the ETAS RCP test platform within the test bench, the whole experimental platform is presented as follows:

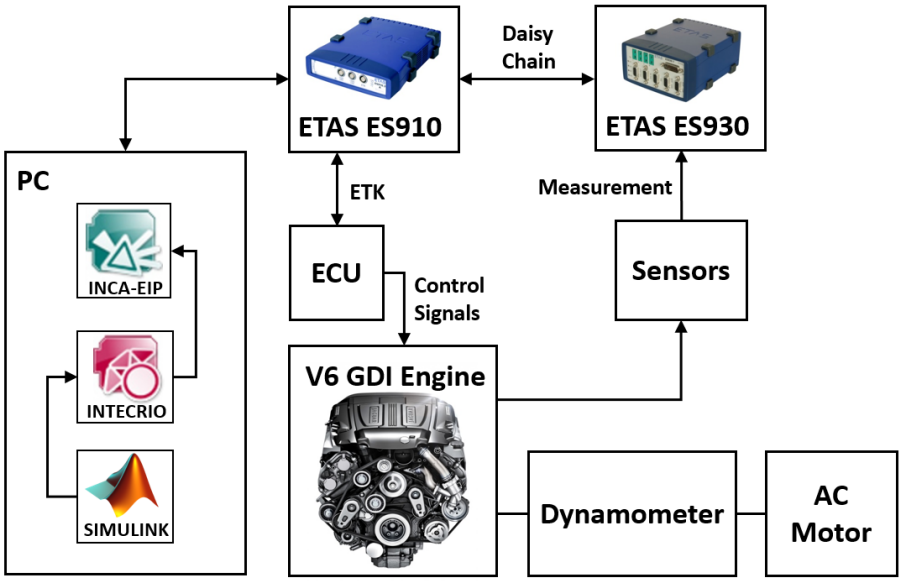


Figure 3.3 Data transfer between the ETAS RCP test platform and the test bench

Generally, the test bench in this thesis supplies an engine installed with a dynamometer test cell powered by an AC motor. Additionally, the test bench includes an additional ECU, sensors and actuators that request dedicated I/O signals to measure and control. Accordingly, the ETAS RCP test platform is applied to accomplish the communication among various devices.

As mentioned above, the signals from sensors are measured by the ES930 and then sent to the ES910 through Daisy Chain as the feedback signals for the controller. The controller operates the functions in the ES910 and sends the updated control signals to the corresponding engine sub-systems or actuators. Because of the ETK bypass cable, the control signals from the ES910 can be applied on the engine directly and independently of the availability of ECU resources. As mentioned above, the ES910 will operate the modified engine management system through the ETK bypass interface while running the unmodified engine management system through the ECU. Accordingly, the modified engine management system can be validated to see if it is better than the original engine management system.

The engine's coolant system is critical because it can affect the security of the experiment and the accuracy of the measurement data. On one hand, an effective coolant system can ensure that the engine is fully warmed up while avoiding the overheating. On the other hand, it can improve the stability of the data over a long term operation and thus improve the reliability of the repeated experiments. The current engine management system still uses a PID controller to adjust the coolant flow valve and control the intake time and mass flow of the coolant liquid. Typically, the operating temperature of the engine coolant is set to 90 °C. When the engine is just started or runs at a low temperature, the coolant does not circulate, but the engine oil will absorb the extra heat from the engine and shorten the engine warm-up time. When the temperature of the engine coolant liquid is over 90 °C, the coolant liquid will all flow into the coolant system. In addition, in order to avoid accidents caused by the engine overheating, when the coolant temperature exceeds 120 °C or the cylinder block temperature exceeds 110 °C, an alarm will be triggered and the engine will automatically turn to the idling mode. At the idling mode, the

coolant remains circulating to cool down the engine components. In terms of all the experiments listed in this thesis, the engine has been fully warmed up.

As for the fuel supply system, the gasoline is supplied by an external fuel line at ambient pressure. The electric low pressure pump will boost the pressure of the gasoline from the ambient pressure at the pipe outlet to 3-5 bar. The pressure regulator then controls the fuel pressure to reach a threshold in order to provide sufficient pressure for the high pressure pump. The high pressure pump is a mechanical pump driven by a camshaft that can boost the injection pressures up to 120 bar. The rest of the fuel will return to the fuel tank through the pressure relief valve.

3.2 Experimental Hardware

As for the hardware, the GDI engine specifications, the dynamometer-based test bench, the RCP test hardware, and the engine performance measurement devices are introduced.

3.2.1 Engine Specification

As introduced in Chapter 1, the GDI engine with advanced boost technology has been widely employed and will take the majority of the market for passenger cars. A Jaguar AJ126 3.0L V6 supercharged GDI engine (shown in Figure 3.4) is offered by our partner Jaguar Land Rover and employed in this study. It has been equipped by Land Rover Discovery, Jaguar F-Pace, Range Rover Velar and many vehicles since 2014, which makes the experimental investigations in this thesis more practical. The engine specifications can be found in Table 3.1. The engine is equipped with a VVT system.

Then, the intake and exhaust cam timing are allowed to be altered within 60 CAD. Besides, the high compression GDI technology is applied for the fuel injection system, combined with a supercharger to boost the pressure of intake air in order to increase the power. The benefits of GDI technology have been introduced in Chapter 1, including stoichiometric fuel mixtures, high thermal efficiency and reductions of HCs and NOx emissions.

Figure 3.4 Jaguar AJ126 3.0L V6 GDI engine

Table 3.1 Engine specifications

Engine Type	Jaguar V6 GDI	**Compression Ratio**	10.5
Displacement Volume	2995 cc	**Peak Power**	246 kW
Max Engine Speed	6600 rpm	**Peak Torque**	450 NM
Bore	84.5 mm	**Intake Valve Timing**	Variable
Stroke	89 mm	**Exhaust Valve Timing**	Variable
Rod	154 mm	**Standard Euro Emissions**	Euro 6
Fuel	ULG95		

3.2.2 RCP Test Hardware

The deployment of RCP technology has allowed engineers to develop the ECU functions in a more efficient and reliable collaborative manner. New concepts can be developed in Simulink and tested on-board the real engine / vehicle, far ahead of the actual release for the implementation of a new ECU function to the ECU vendor (ETAS, 2012). One crucial stage in the ECU function development is the early verification and correction of errors. It is able to avoid the time-consuming and labour-intensive trouble-shootings during the software development. In addition, the testing phase on-board the real engine / vehicle offers engineers ample opportunities to precisely calibrate the parameter settings of controllers under the real-world conditions. Meanwhile, the sufficient calibration data can be easily collected in this phase for the further development of ECU functions (ETAS, 2012). Thus, the RCP operation platform has significantly increased the efficiency of the controller development and has reduced the costs typically associated with projects. The RCP test hardware in this thesis is supported by ETAS Ltd. A general introduction for the RCP test hardware employed in this thesis is listed below.

The ES910 is a powerful real-time processor with high computational performance. Its prototyping and interface module is compatible with all common ECU interfaces in a compact package. It can work effectively in a variety of development environments. The ES910 is used in conjunction with the INCA-MCE (Measurement and Calibration Embedded) module to ensure the fast exchange of measurement and calibration data between the ECU and the test bench. The CAN interface ensures the connection of the ES910 module to the engine, while the ETK interface allows the ES910 module to operate synchronously with the production ECU. This is also known as the rapid prototyping bypass operation. The ETK interface can meet the high real-time requirement and

separate the function model and the ECU. Then, the prototyping of new functions can be operated independently of the availability of ECU resources. Hence, the bypass mode can achieve the mapping of costly and demanding control structures in the early period of development, or migrate the techniques to an alternative ECU from other suppliers (ETAS, 2012). The specifications and main function blocks of ES910 are listed as follows:

Table 3.2 Specifications of the ES910

Environment	
Temperature range	-40 °C to +50 °C / -40 °F to +122 °F (operation)
	-40 °C to +85 °C / -40 °F to +185 °F (storage)
Power supply	
Input voltage	6 V to 32 V DC
Power consumption	14 W (typically) / 20 W (max.)
Stand-by current	< 25 mA
Host interface	
Ethernet connection	10/100/1000 Base-T
Protocol	TCP/IP
IP address	Dynamic
Simulation target	
Main processor	NXP PowerQUICC™ III MPC8548 with 800 MHz clock Double precision floating point unit
Memory	
RAM	512 MByte DDR2-RAM (400 MHz clock)
Flash	64 MByte Flash
ECU and bus interfaces	
ETK	1 Channel
ECU	1 Channel
CAN	2 Channels
LIN	2 Channels

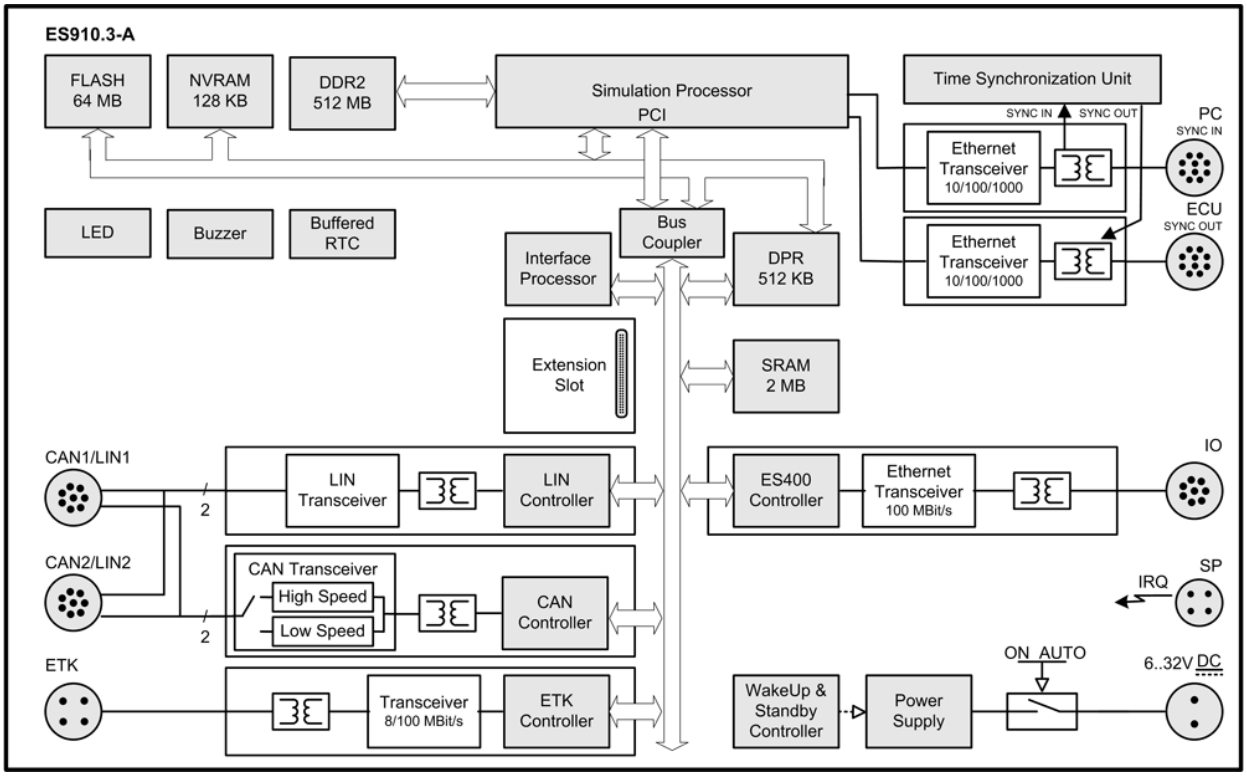


Figure 3.5 Main function blocks of the ES910 Prototyping and Interface Module

The ES930 is a powerful and flexible multi-I/O module featuring numerous input and output channels, which is used to sample the sensors utilizing traditional 0-5 V analogue to digital sampling appropriately. The ES930 multi-I/O module extends the functionalities of the ES910 rapid prototyping and interface module, in order to analyse and control the sensors and actuators directly within a given function model. It actually serves as a compact measurement module and transmission module to communicate with the particular hardware at the test bench. The specifications and main function blocks of ES930 are listed as follows:

Table 3.3 Specifications of the ES930

Analog input channels	
Number of channels	8
Measuring range, resolution	±1 V, ±10 V, ±60 V for differential input voltages, 16 bit
Digital input channels	
Number of channels	4
Input voltage	0 V to 5 V, TTL-compatible
Counter width, resolution	32 bit, 15 ns
Analog output channels	
Number of channels	4
Output voltage, resolution	0 V to +10 V, 14 bit
Output current	±4 mA (max.)
Digital output channels	
Number of channels	6
Output voltage	TTL-compatible
Counter width, resolution	32 bit, 15 ns
Sensor supply	
Number of channels	4, assigned to sensors or input channels
Output voltage	+5 V to +15 V DC or "Off", configurable per channel, resolution <10 mV
Output current	50 mA (max. per channel at 5 V) 30 mA (max. per channel at 15 V, all channels loaded simultaneously)
Host interface	
Ethernet interface	100Base-T
IP-Address	Dynamic
Power supply	
Operating voltage	6 V to 32 V DC

Environment	
Operating temperature range	-40 °F to +158 °F / -40 °C to +70 °C
Supported by ETAS software for configuration, control, or data recording	
General	ES93x Configuration Tool V1.3.0 and higher
Measurement and calibration (MC)	INCA V7.0.0 and higher with INCA Add-On ES93x V1.3.0 and higher
Rapid Prototyping (RP)	INTECRIO V4.0 and higher

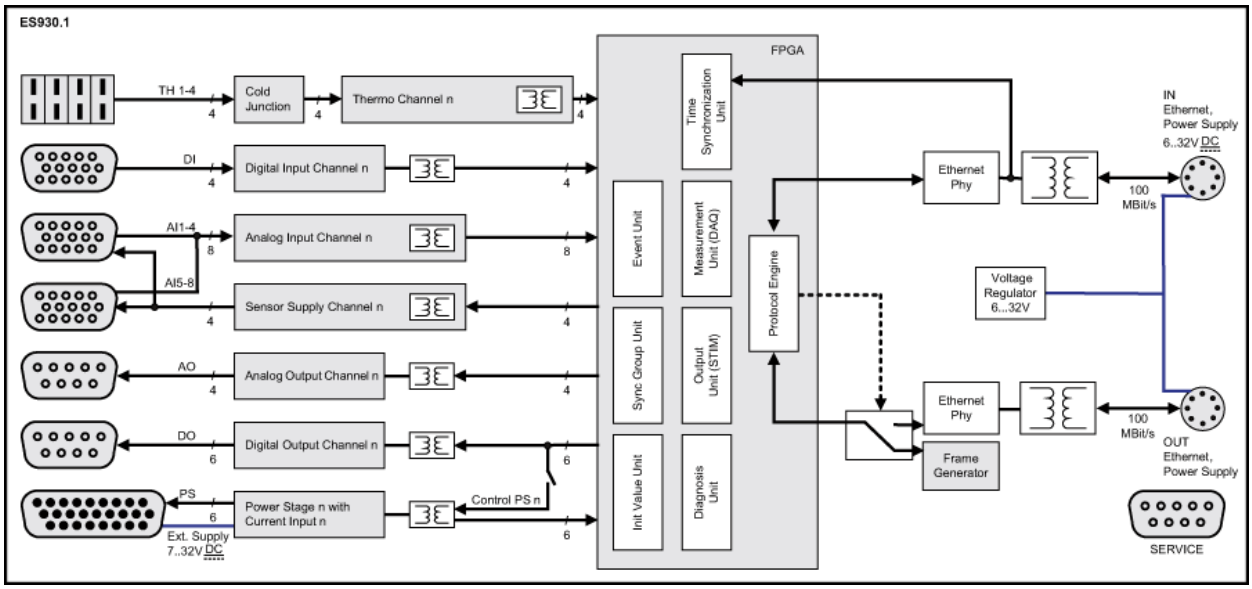


Figure 3.6 Main function blocks of the ES930 multi-I/O module

3.2.3 Engine Performance Measurement Devices

The main engine performance measurement devices employed in author's study include fuel flow meter, PM emission analyser and lambda sensor.

a) Fuel flow meter

A Rheonik Coriolis fuel flow meter system is applied on the test bench to measure the fuel consumption during the calibration process. It will provide the continuous fuel consumption signal in real-time. As a Coriolis-flow type meter, it separates the gas bubbles in the fuel to improve the measurement accuracy and guarantees a transient fuel flow measurement. The complete fuel flow meter system consists of a sensor, a transmitter and a connection cable. The sensor employed in this thesis is Rheonik RHM 03 as shown in Figure 3.7 and its specifications are given in Table 3.4. The transmitter is Rheonik RHE 08 as shown in Figure 3.8, whose specifications can be found in Table 3.5.

Figure 3.7 Rheonik RHM 03 Sensor

Table 3.4 Specifications of the Rheonik RHM 03 Sensor

Max Allowance Pressure	1224 bar / 17752 psi
Temperature Range	-196 up to +350 ℃
Nominal Flow	6 kg/min
Max Flow	12 kg/min
Min Flow	0.1 kg/min

Mass Flow Uncertainty	≤0.10%
Repeatability	≥0.05%

Figure 3.8 Rheonik RHE 08 transmitter

Table 3.5 Specifications of the Rheonik RHE 08 transmitter

Measurements	Mass Flow Density Temperature
Functions	Batch / Filling Control Standard Density Volume Flow % Concentration Brix
Outputs	2 Analog 1 Frequency / Pulse 3 Digital Status
Inputs	2 Digital
Display	LCD

The sensor has precision vibrating tubes inside, which are powered by an electro-magnetic drive system. And the electro-magnetic drive system itself is powered by the transmitter. The precision vibrating tubes vibrate at their natural frequency. As the liquid flows through the vibrating tubes, it causes inertia and leads to additional deflection. This deflection is electronically recorded by two velocity sensors and a high-precision electronic time lag sensor. This measurement is proportional to the mass flow and will be converted to a physical unit using a signal processor in the transmitter.

All of the functions in the fuel flow meter are constantly monitored during the operation. If any interruptions occur, the error messages will be immediately displayed. The Rheonik Coriolis fuel flow meter will output the analog fuel consumption signal to the ES930 of the RCP platform in real-time. As for the accuracy, the mass flow uncertainty is less than 0.10% and repeatability is better than 0.05%.

b) DMS 500

As the Euro 6 emission standard imposes stricter restrictions for the PM concentration and weight separately specific for GDI engines, both metrics of particles must to be minimised. Hence, the DMS500 Fast Particle Analyser (as shown in Figure 3.9) developed by CAMBUSTION Ltd. becomes the perfect equipment for measuring both the PM concentration and weight in this thesis.

Figure 3.9 DMS500 Fast Particle Analyser

Table 3.6 Specifications of the DMS500 Fast Particle Analyser

Particle Size Range	5nm - 1µm (5nm - 2.5µm option)
Max Sample Temperature	800°C
Minimum Sample Pressure	600 mb
Sample Flow Rate	8 slpm (1 µm range) at 0 °C + 100 kpa
Analogue Outputs	4 @ 10 Hz; software configurable
Analogue Inputs	4 @ 10 Hz; software configurable
Output Data Rate	10/sec – 1/min
Max Concentration	1×10^{11} N/cc
Controlling Computer	Windows PC (by Ethernet)

Since 2002, the DMS500 Fast Particle Analyser has become one of the most widely used particulate instruments for research labs, universities and the majority of vehicle OEMs worldwide. It combines electrical mobility measurements of particles with sensitive

electrometer detectors, and offers simultaneous outputs of particle size, number concentration and mass as well as their distributions in real-time.

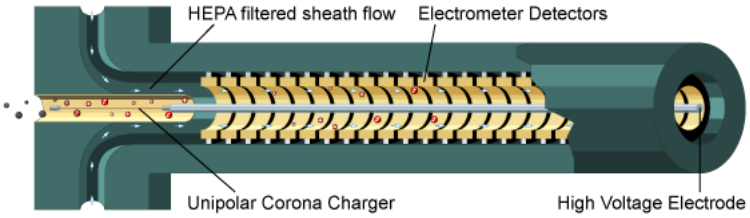


Figure 3.10 Classifier of the DMS500

The DMS500 uses a high voltage discharge to charge each particle proportional to its surface area. Charged particles are introduced into a classifier shown in Figure 3.10 with a strong radial electrical field. This field causes particles to drift through a sheath flow toward the electrometer detectors. Particles are detected at different distances down the column, depending upon their aerodynamic drag/charge ratio. Outputs are processed in real-time at 10 Hz to provide spectral data and other metrics.

The DMS500 offers the measurement of the particle size distribution from 5nm to up to 2.5μm for both solid particles and liquid droplets. The previous study (Tan et al., 2014) has verified that 10s are totally sufficient for DMS500 to get stable PM concentration and weight measurements. It can offer high resolution for even longer scanning times. The DMS500 is controlled remotely by a Windows PC via Ethernet. The user interface includes the real-time data-logging of the spectrum. The data summary tool developed by CAMBUSTION Ltd. helps users get the rid of complex data processing. The DMS500 data files, which are readable with MS Excel and MATLAB, contain all summary and spectral data without the requirement of post-processing. For the calibration works in this

thesis, the DMS500 will output the analog PM concentration and mass signals separately to the ES930 of the RCP platform in real-time.

c) Lambda sensor

A Bosch lambda (oxygen) sensor LSU 4.9 (shown in Figure 3.11) is employed on the test bench to measure the continuous AFR in real-time from the exhaust. The LSU 4.9 is a wide-band lambda sensor developed by BOSCH Ltd., which provides precise measurements over a wider measuring range for λ, from 0.65 to ∞. It can thus permit precise control concepts for not only the stoichiometric mode, but also the lean or rich mixtures.

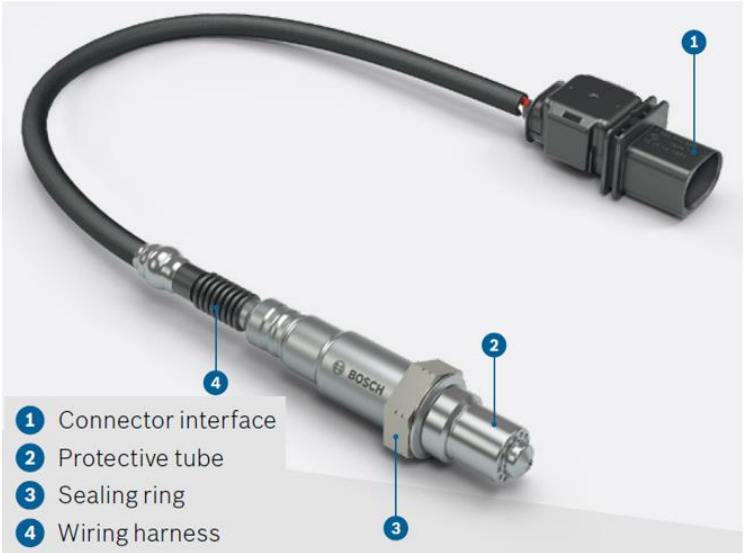


Figure 3.11 BOSCH LSU 4.9 lambda sensor

The LSU is a planar sensor which integrates the measuring unit and heater into the same sensor element. On one hand, it is able to eliminate the effects caused by the temperature of the exhausts on its functions. On the other hand, it allows the LSU to attain its operation

state rapidly in the shortest time. The heating time is usually between 5 and 20 seconds according to different types and operating conditions. Besides, the LSU has a double-walled protective tube that provides the thermal shock protection for gasoline engines. Its structure is shown in the figure below.

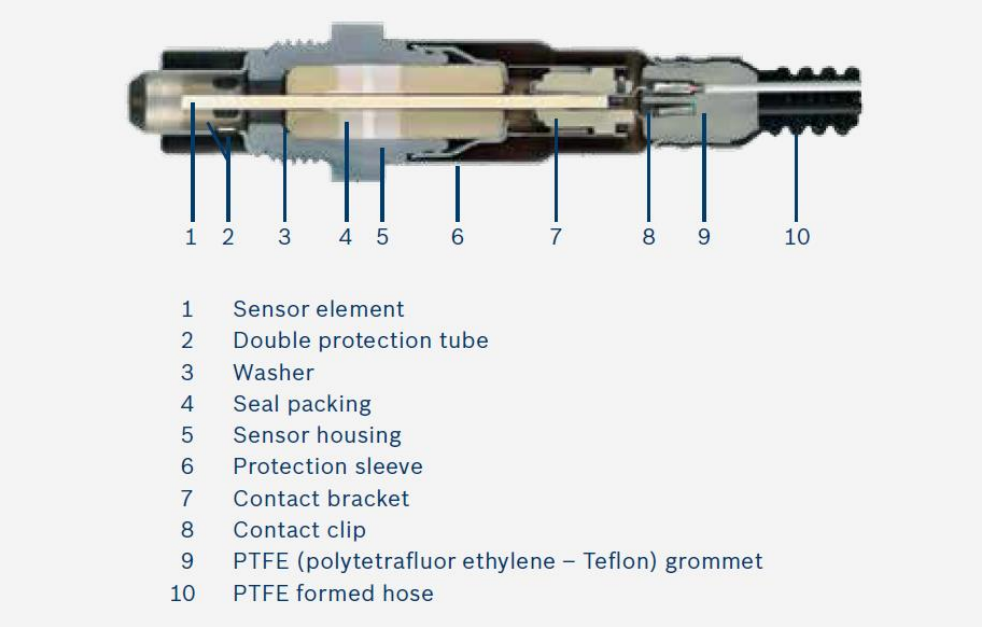


Figure 3.12 Cross-section through the LSU wide-band lambda sensor

Table 3.7 Specifications of the lambda sensor

Lambda Control Range	λ = 0.65 to air
Max. Exhaust-Gas Temperature (Sustained)	≤ 930 °C
Peak Temperature	≤ 1,030 °C
Type	Nernst sensor
Heater Voltage Requirement	11 V
Trimming Resistor	inside connector casing

On modern engines, the lambda sensor is located on the exhaust pipe, upstream and downstream of the catalytic converter. One side of the sensor element electrode is in contact with the exhaust gas, while the other side is exposed to the surrounding air which is used as the reference for measuring the residual oxygen content. After the LSU measures the residual oxygen content of the exhaust gas, it compares this residual oxygen content to the surrounding air oxygen content and identifies the rich mixture ($\lambda < 1$) or lean mixture ($\lambda > 1$). Then, it generates a feedback signal corresponding to the oxygen content of the exhaust gas and sends it to the ES930 of the RCP test platform. Thanks to the rapid dynamics and high precision of LSU, even a slight deviation in mixture formation can be detected and corrected before the negative disruption takes place. This guarantees the long-term measurement accuracy, save the fuel consumption by saving the operating time, and reduce the demand for the catalytic converter's storage capacity for oxygen.

3.2.4 Dynamometer-based Engine Test Bench

In this thesis, the rotation speed of the engine is adjusted by a Froude EC38 eddy current dynamometer as shown below:

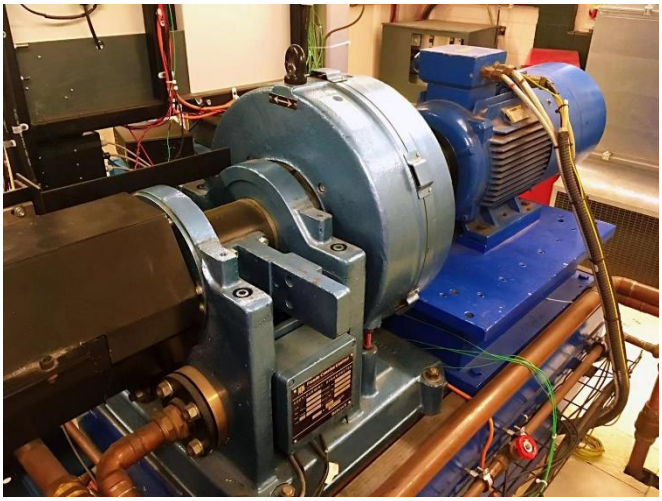


Figure 3.13 Froude EC38 eddy current dynamometer

The technical specifications of the Froude EC38 eddy current dynamometer can be found in Table 3.8. This type of dynamometer offers excellent stability for engine steady state mapping works (endurance tests), and some low-transient type tests.

Table 3.8 Specifications of the dynamometer

Model	Froude EC38 Eddy Current Dynamometer
Maximum Power	165 kW
Maximum Torque	477 Nm
Maximum Speed	8000 RPM
Moment of Inertia	0.101 kg·m^2
Torque Measurement Accuracy	±0.25% of full rated torque of dynamometer
Speed Measurement Accuracy	±1 RPM

The dynamometer is powered and controlled by a DSG Series 2000 control system. The dynamometer has two operation modes: speed mode and torque mode. In the speed mode, the engine rotation speed is regulated and maintained by the dynamometer directly,

and then the engine adjusts the injected fuel according to the throttle position to achieve a specific torque. In the torque mode, the torque is regulated and maintained by the dynamometer directly and the engine is then adjusted to a desired speed. The relationship between the torque and speed applied to the dynamometer and the electrical signal it releases is calibrated and registered by the dynamometer supplier. The torque or speed required for the dynamometer is controlled by the DSG Series 2000 connected with the ETAS system. The speed mode is chosen in this thesis because of its fast dynamic performance and small fluctuation.

3.3 Experimental Software

In terms of the software, the MATLAB/Simulink-based controller design, the RCP associated software and the experimental data processing methods are presented.

3.3.1 Design of Controller

Nowadays, the control functions are widely defined and modified in the MATLAB/Simulink environment. Thus, the controller embedded with proposed control and calibration functions is firstly designed and generated by the MATLAB/Simulink.

It is well-known that MATLAB is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. It has been widely used for data analytics, signal and image processing, and system design, owing to its fast coding capability, rich toolboxes and useful plotting tools. It can also interface with programs compiled in other programming languages. Thanks to the MATLAB, the implementation of artificial intelligence becomes much easier.

The Simulink, as an additional package of MATLAB, adds graphical programming environment for the design, modelling, simulation and analysis of multi-domain dynamical systems and embedded systems. With the Simulink, a complex controller can be generated by a set of customizable graphical programming blocks, integrated with the rest of MATLAB scripts. Hence, the Simulink has become the most popular programming tool for the automatic control and signal processing in the automotive industry.

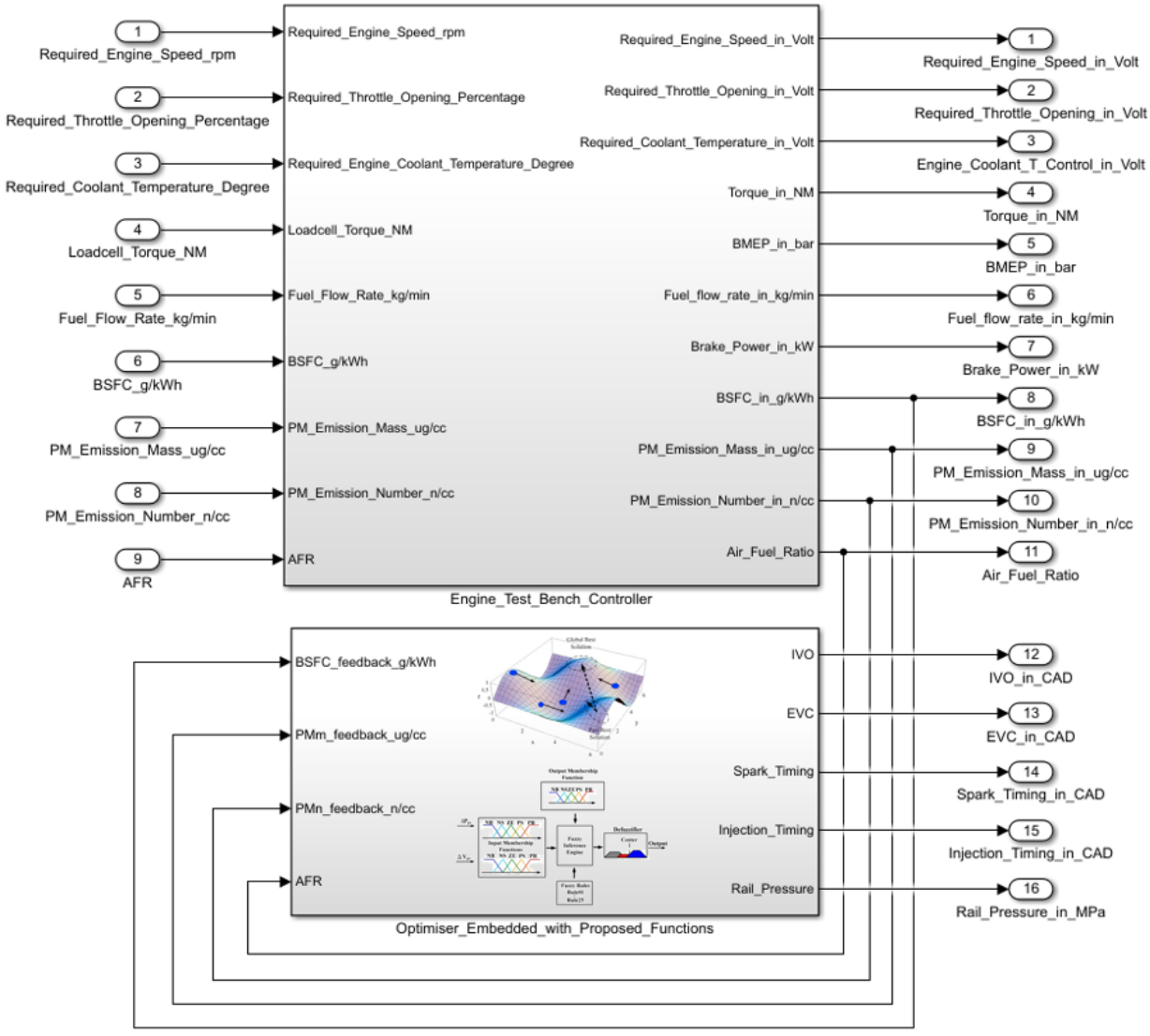


Figure 3.14 Structure of the controller generated by MATLAB/Simulink

Figure 3.14 gives an example of the controller model constructed by MATLAB/Simulink. Basically, the controller designed in this thesis consists of two major parts. One is the engine test bench controller with two main functions: firstly, to send the required engine operation condition signals (e.g. required engine rotation speed or throttle opening) to the test bench; secondly, to collect the measurement signals and display them in the INCA operation interface. The other one is the optimiser embedded with the proposed artificial intelligence functions. It uses the engine measurement signals as feedback signals, operates the programs and outputs the corresponding optimal control signals to the test bench.

3.3.2 RCP Test Software

The software employed for the RCP testing in this thesis includes the ETAS INTECRIO and ETAS INCA.

The software ETAS INTECRIO can easily identify the MATLAB/Simulink-based controller model and convert it into the C-code program, which is suitable for the real-time execution on the compact ES910 prototyping and interface module for RCP test purposes (ETAS, 2013). Then, the ETAS INTECRIO offers users an integration environment to set up RCP configurations rapidly by linking the input and output signals of the controller to the corresponding ECU bypass interfaces, sensors and actuators on the test bench (ETAS, 2012). Thus, the controller model can be embedded directly into the existing RCP configurations and engine management systems. To meet different hardware configurations and development requirements, the ETAS INTECRIO provides a variety of on-board prototyping test targets and integration environments to achieve different RCP functions in conjunction with the ES910 module (ETAS, 2013).

As mentioned in section 3.2.2, the ES910 module is synchronized with the production ECU by using the ETK bypass interface. Therefore, the execution of the proposed control functions on the prototyping target can be separated from the ECU operation and run in parallel (ETAS, 2013). The complete electronic control of the engine is accomplished by using the software ETAS INCA integrated with its INCA Experimental Target Integration Package (INCA-EIP) add-on. It is able to operate the modified engine management system with the bypass function through the ES910 module while running the unmodified engine management system through the ECU.

The ETAS INCA can access not only the function model on-board the ES910 module, but also the existing engine management system in ECU. It provides a one-click-build virtual control prototype with the executable programs transformed from the Simulink model by the ETAS INTECRIO. Then, the proposed control functions can be operated as executable programs running on PCs (ETAS, 2013). The virtual control prototype works on the ES910 module in the same way as the engine management system does in a production ECU. But this virtual control prototype can be loaded independently by the INCA and shared with other INCA users without revealing its underlying model (ETAS, 2013).

The ETAS INCA can display the synchronous measurement signals from the ES910 module and ECU in the same interface. It also allows the measurement and calibration values of the control functions on the ES910 module to be displayed and adjusted in real-time within the same interface (ETAS, 2013). Meanwhile, users can adjust the sampling frequency to cope with different application requirements. Thus, the engine operating conditions and the mapping of the control function inputs can be easily modified by the users (ETAS, 2013). As a result, operations are shifted from the engine test bench to the

computer. And the new ECU functions can be validated through the real-world engine testing under particular conditions and corresponding experimental results can be recorded as required from various devices (ETAS, 2012).

All in all, the ETAS INTECRIO and INCA form the toolchain for implementing the RCP testing. The workflow to generate an RCP test task using the ETAS INTECRIO and INCA can be found in Figure 3.15.

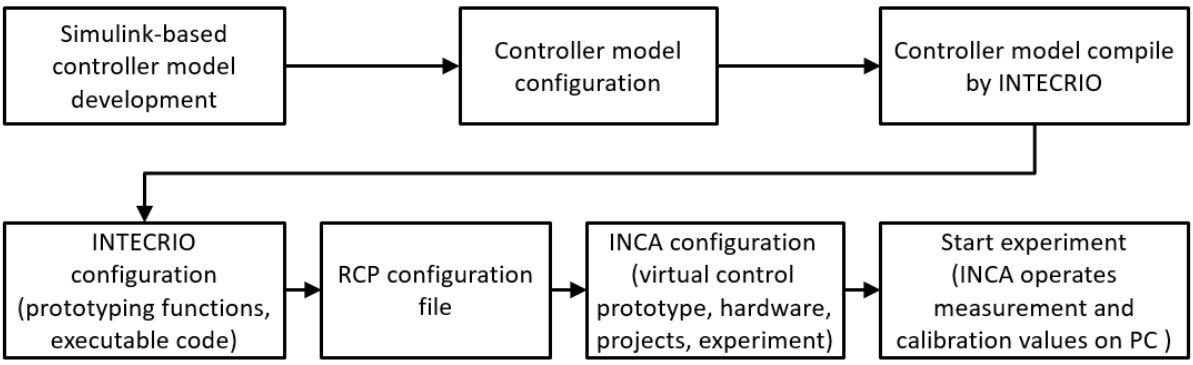


Figure 3.15 Workflow to generate an RCP test task

As the Figure 3.15 shows, the Simulink-based controller model is compiled as the real-time C-code first for the RCP target by the ETAS INTECRIO. The INTECRIO prototyping integration platform combines the controller model with further prototyping functions, builds the interface description file and executable code for the specific RCP system. This allows the ETAS INCA to access the control function parameters for calibration and measurement. Eventually, The ETAS INCA is the platform for data acquisition and recording (Li, et al., 2018).

3.3.3 Operation Interface

With the software ETAS INCA, the experimental operations can be shifted from the engine test bench to the computer.

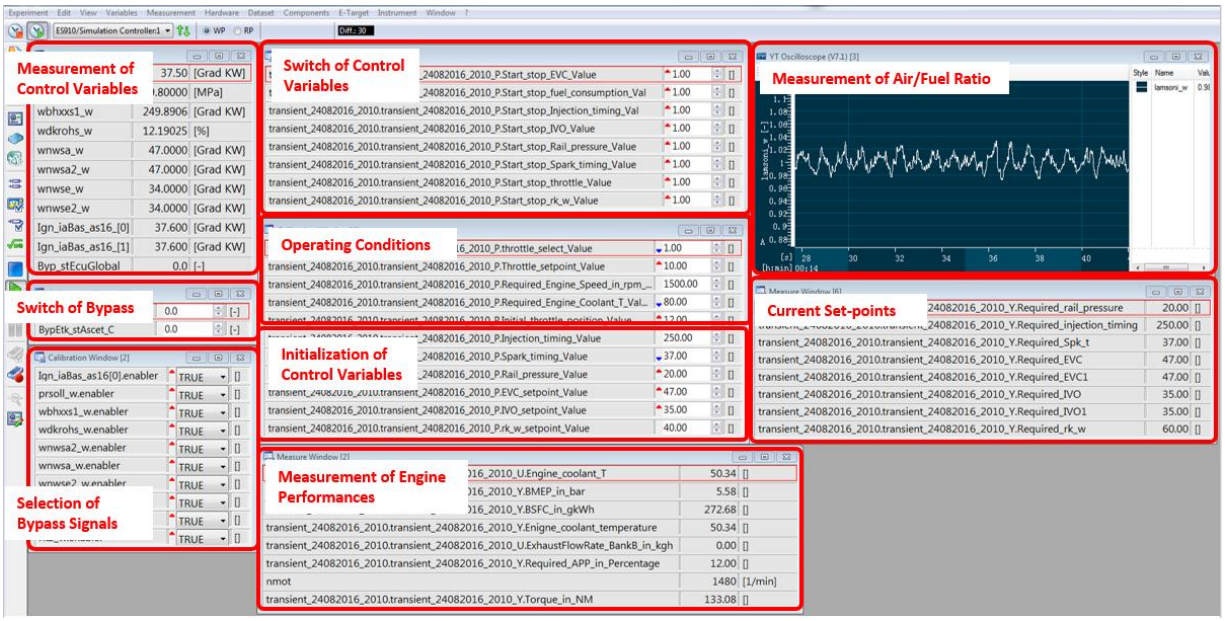


Figure 3.16 ETAS INCA operation interface

As can be seen from the Figure 3.16, the INCA offers users an operation interface which includes all the measurement and calibration data, together with the switches of operations and the engine operating condition configurations. Thus, the calibration data can be easily modified via the operation interface. And the corresponding experimental results can be easily measured and recorded from various devices in real-time.

3.4 Experiment Plan

In this thesis, both the steady-state and transient engine tests are conducted on the engine test bench. The test on the dynamometer-based test bench is different from that

on the real vehicle. The engine speed is adjusted and maintained by the dynamometer within the range of 1200 to 2000 rpm, which is a typical speed range for experimental studies of modern light-duty gasoline engines. Meanwhile, the engine load is adjusted by the throttle position (throttle opening). In terms of the transient engine tests, the adjustment of the operation conditions is achieved by a step-change of the engine speed or throttle position. After the step change occurs, the engine performance signals may oscillate and need a period of time to stabilize again. This period of time is called the settling time. And the overshoot is the increased amplitude of a signal exceeding its steady-state value. The settling time varies based on different tests and the overshoot of the signals should be constrained at the same time. The sampling frequency is 100 Hz in real-time. The details of the experiments for each study are presented in the following chapters separately.

3.5 Summary

The experimental set-up is introduced in this chapter, which consists of the GDI engine test bench, ETAS RCP test platform, and engine performance measurement devices. With the ETAS RCP test platform, the operation and validation of the calibration and control strategies are accomplished using a computer. The investigation approach and experiment plan are also introduced.

CHAPTER 4 INTELLIGENT NON-MODEL-BASED CALIBRATION APPROACH USING ARTIFICIAL INTELLIGENCE ALGORITHM

Part of the study presented in this chapter has been published in the journal “Proceedings of the IMechE, Part D: Journal of Automobile Engineering” (Ma et al., 2019) and the journal “ASME Journal of Dynamic Systems, Measurement and Control” (Ma et al., 2017). As the second author of both two publications, the author of this thesis constructed the optimiser, configured the test platform, designed the experiments, collected the data and wrote the publications. In this chapter, two intelligent non-model-based calibration approaches using SPEA2 and CAPSO algorithm separately have been developed, which can optimise the engine’s brake specific fuel consumption (BSFC), particulate matter emission number (PMn) and mass (PMm) by automatically adjusting the engine variable settings (IVO, EVC, spark timing, injection timing and rail pressure) without the requirement of an engine model or massive experimental data.

A general introduction to the background of this study is presented first. Then, the calibration task is formulated as a multiple-objective optimisation problem with constraints. Besides, the Simulink-based real-time optimisers are developed and implemented on a V6 GDI engine test bench through the ETAS RCP platform. The experimental results are shown in the following section, and the comparisons among the proposed calibration approaches and the baseline calibration approach are given concerning the calibration efficiency and results. The final section presents a summary.

4.1 Introduction

As introduced in the first two chapters, the engine calibration task becomes extremely challenging even for the experienced engineers involving the time, labour and other consumable items, because of the increasing number of sub-systems introduced to the engine management system (EMS). Meanwhile, the increasing complexity of the EMS also makes it difficult to achieve the trade-off relationship among various objectives and the computational speed. As a result, the calibration task has become a multi-objective optimisation problem.

The conventional calibration approach has been developed from the manual setting through hundreds of trial-and-error experiments to the model-based calibration (Ma et al., 2019; Men et al., 2018; Grasreiner et al., 2017). However, as introduced in section 2.1.2, the use of engine models has many inevitable drawbacks. Meanwhile, the reliability of calibration results for the real engines significantly relies on the accuracy of engine models. A poorly constructed model will result in undesirable settings which require the further adjustment by engineers (Wong et al., 2018a).

Thus, the intelligent non-model-based calibration approach is developed in this chapter. The comparison between the model-based calibration approach and the non-model-based calibration approach can be found in Figure 2.2. As for the calibration in the industry, not only the final optimum results but also the time consumption should be taken into consideration. This chapter aims to develop an efficient and economical non-model-based calibration approach to further improve the automation level of the multi-objective calibration. For this purpose, an efficient online optimisation algorithm has to be employed. Accordingly, the SPEA2 and CAPSO algorithm are selected as the most advanced meta-heuristic algorithms and global optimisation methods introduced in section 2.4.3. This

automatic engine calibration process does not need an engine model or massive experimental data.

Three possible and representative cases have been studied for the calibration in this chapter. The PM emissions and BSFC are the objectives and the engine variable settings are constrained. Accordingly, the calibration case becomes a multi-input and multi-objective optimisation problem. The proposed non-model-based calibration approach is very flexible and robust. This chapter aims at presenting the concept and methodology. The variables and objectives can be replaced based on different cases and requirements.

4.2 Methodology

4.2.1 Forming the Optimisation Problem

The experimental studies are all conducted on the 3-litre supercharged V6 GDI engine test cell described in Chapter 3 at the University of Birmingham for the part load steady-state calibration. The non-model-based calibration optimisers embedded with the algorithms will generate a number of calibration variable settings to study their effect on fuel consumption and PM emissions.

The CO, THC and NOx emissions after the three-way catalyst converter have met the Euro 6 emission legislation. The AFR is maintained as 14.7 for the stoichiometric combustion. As mentioned in Chapter 1, the fuel consumption and PM emissions are still among the main objectives for calibration. Thus, the engine calibration task is formed as a multi-objective optimisation problem with three objectives and five variable settings in this chapter. Namely, the optimiser should identify the variable settings IVO, EVC, spark

timing, injection timing and rail pressure to minimise the brake specific fuel consumption (BSFC), PM emission number (PMn) and PM emission mass (PMm).

In addition, from the view of calibration, mapping the variable settings at infeasible engine operating conditions would be a waste of time. Therefore, the combustion stability and exhaust gas temperature act as non-linear constraints for the calibration in this chapter. On this premise, all the variable settings have to be constrained within the reasonable range as Table 4.1 shows.

Table 4.1 Adjustable range of each engine parameter

	IVO (aTDC)	**EVC (aTDC)**	**Spark timing (bTDC)**	**Injection timing (bTDC)**	**Rail pressure (MPa)**
Min	-25CAD	0CAD	-10CAD	250CAD	15
Max	35CAD	50CAD	50CAD	400CAD	20
Minimum sample interval	1 CAD	1 CAD	1 CAD	10 CAD	1MPa

The engine calibration thus becomes a “black-box” multi-objective optimisation problem with constraint variables:

$$\begin{cases} BSFC = f_1(IVO,\ EVC,\ Spark\_t,\ Injection\_t,\ Rail\_p) \\ PMn = f_2(IVO,\ EVC,\ Spark\_t,\ Injection\_t,\ Rail\_p) \\ PMm = f_3(IVO,\ EVC,\ Spark\_t,\ Injection\_t,\ Rail\_p) \\ min\ J = min\ (BSFC + PMn + PMm) \\ -25\ CAD\ \leq IVO \leq 35\ CAD,\ aTDC \\ 0\ CAD\ \leq EVC \leq 50\ CAD, aTDC \\ -10\ CAD \leq Spark\_t \leq 50\ CAD, bTDC \\ 250\ CAD \leq Injection\_t \leq 400\ CAD, bTDC \\ 15\ MPa \leq Rail\_p \leq 20\ MPa \end{cases} \tag{4.1}$$

where f_1, f_2 and f_3 represent the unknown non-linear functions between the variable settings and the engine responses. The BSFC, PMn and PMm are continuously measured online in real-time as the feedback signals throughout the entire calibration process. Figure 4.1 presents the main structure of the calibration task. The engine test bench controller with the optimiser constructed by MATLAB/Simulink can be found in Figure 3.14.

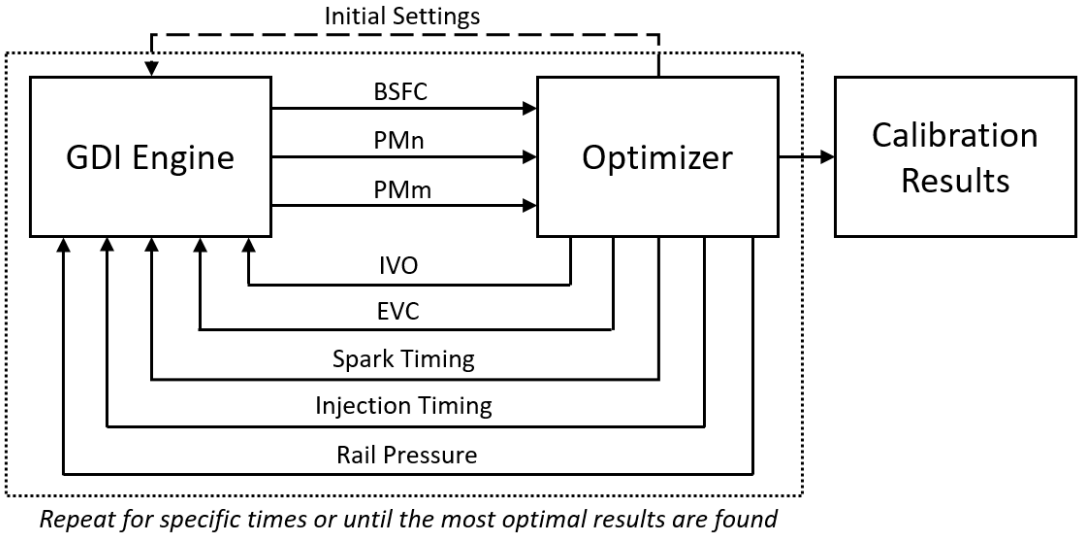


Figure 4.1 Main structure of the calibration task with variable settings and objectives

It should be emphasized that the objectives and constraints are decided on a case by case basis. This chapter divides PMn and PMm as two objectives because the Euro 6 emission legislation has restricted the number and mass of PM emissions separately specific for GDI engines. Basically, ultrafine particles (< 50 nm) dominate to the total number of PM emissions, while fine particles (< 1 μm) generate the major mass of PM emissions. Both of these two metrics have to be minimised from the view of health and environment issues.

4.2.2 SPEA2 Optimiser

Given that the theory and benefits of the SPEA2 have been introduced very clearly in section 2.4.3, this section just presents the specific working steps of the SPEA2 optimiser.

Step 1 Initialization:

Start the engine, and enable the external ECU bypass to allow the engine to run with the modified variable settings offered by the optimiser. Define the population size and the external archive size as N and $\bar{N}$ respectively. The selection for the size of population should consider the trade-off between the time consumption and solution diversity. More individuals may locate better optimisation results, but it will cost more time for each iterative loop. It should be determined on a case by case basis. Based on the author's previous study (Ma et al., 2019), different population sizes 10, 30, and 50 are investigated. It is found that a small population cannot sufficiently cover the searching space and locate the optimum results. In order to ensure the diversity of the variable settings, a population of 50 is selected in this thesis. The archive size is equal to the population size. Thus, the SPEA2 optimiser generates an initial population P_0 with 50 groups of engine variable settings (IVO, EVC, Spk_t, Inj_t and Rail_P) randomly within the ranges shown in Equation (4.1) and sends them to the engine in real-time. Meanwhile, an empty initial external archive $A_0 = 0$ is created for storing non-dominated solutions, and the iterative loop counter is set to $t = 0$. The engine runs with each given group of variable settings for 10s, which is sufficient to allow the measurement readings of the BSFC, PMn and PMm to become stabilized.

Step 2 Fitness assignment:

Calculate the fitness value of each individual in the population P_t and external archive A_t at tth generation (iterative loop). The fitness value contains the raw fitness $R(i)$ and density value $D(i)$. Basically, $R(i)$ represents the strength of individual i's dominators in both archive and population. $D(i)$ is employed to distinguish the individuals with same raw fitness values. Therefore, the non-dominated solution who is farther away from the other non-dominated solutions with the same raw fitness value is retained to improve the diversity.

Specifically, a strength value $S(i)$ is first assigned to each individual in the population and external archive, which indicates the number of solutions that are dominated by this individual. The equation to calculate the strength value of the individual i is given by:

$$S(i) = |\{j \mid j \in P_t + A_t \wedge i \succ j\}| \tag{4.2}$$

where $|\cdot|$denotes the cardinality of a set; + stands for the multiset union; the symbol $\succ$ corresponds to the Pareto dominance relation; and j is the individual that is dominated by i. Then, the raw fitness value $R(i)$ is equal to the sum of the strength values of all the individuals that dominate the individual i:

$$R(i) = \sum_{j \in P_t + A_t \wedge j \succ i} S(j) \tag{4.3}$$

It should be noted that the raw fitness value is to be minimized here. In other words, $R(i) = 0$ indicates the non-dominated solution, while a higher $R(i)$ represents that the individual is dominated by many other individuals.

The density value $D(i)$ of the individual i is calculated based on the k^{th} nearest neighbour method:

$$\begin{cases} D(i) = \dfrac{1}{\sigma_i^k + 2} \\ k = \sqrt{N + \bar{N}} \end{cases} \tag{4.4}$$

where σ_i^k denotes the spatial distance (Euclidean distance) between the individual i and its k^{th} neighbouring individual. k is determined by the population size N and archive size $\bar{N}$. Eventually, the fitness value $F(i)$ of each individual i can be obtained by:

$$F(i) = R(i) + D(i) \tag{4.5}$$

Step 3: Environmental selection:

Copy all the non-dominated solutions (current Pareto-optimal solutions) from P_t and A_t to the archive A_{t+1}.

If the size of A_{t+1} is larger than $\bar{N}$ (archive size), then reduce A_{t+1} using the truncation operator; else fill A_{t+1} by the dominated individuals in P_t and A_t until the archive size becomes equal to $\bar{N}$. In this chapter, $\bar{N} = 50$.

Step 4: Termination:

T is the maximum number of the generations (iterative loops), which is set as 35 in this chapter. If $t \geq T$, then store all the non-dominated solutions in the external archive A_{t+1} and finish the engine test. If $t < T$, then go to step 5.

Step 5: Mating selection and variation:

50 individuals are put into the mating pool from the A_{t+1}. Crossover and mutation operators are then applied to the mating pool. This chapter uses the binary tournament

selection method (Maheta and Dabhi, 2014) to select the parent in A_{t+1}. To be specific, two individuals are chosen at random from the A_{t+1} to compete against each other. The one with better fitness is selected as a parent. Then the engine runs with the derived new 50 groups of engine variable settings. At last, increase t to $t + 1$ and go back to step 2.

Figure 4.2 gives the workflow of the SPEA2.

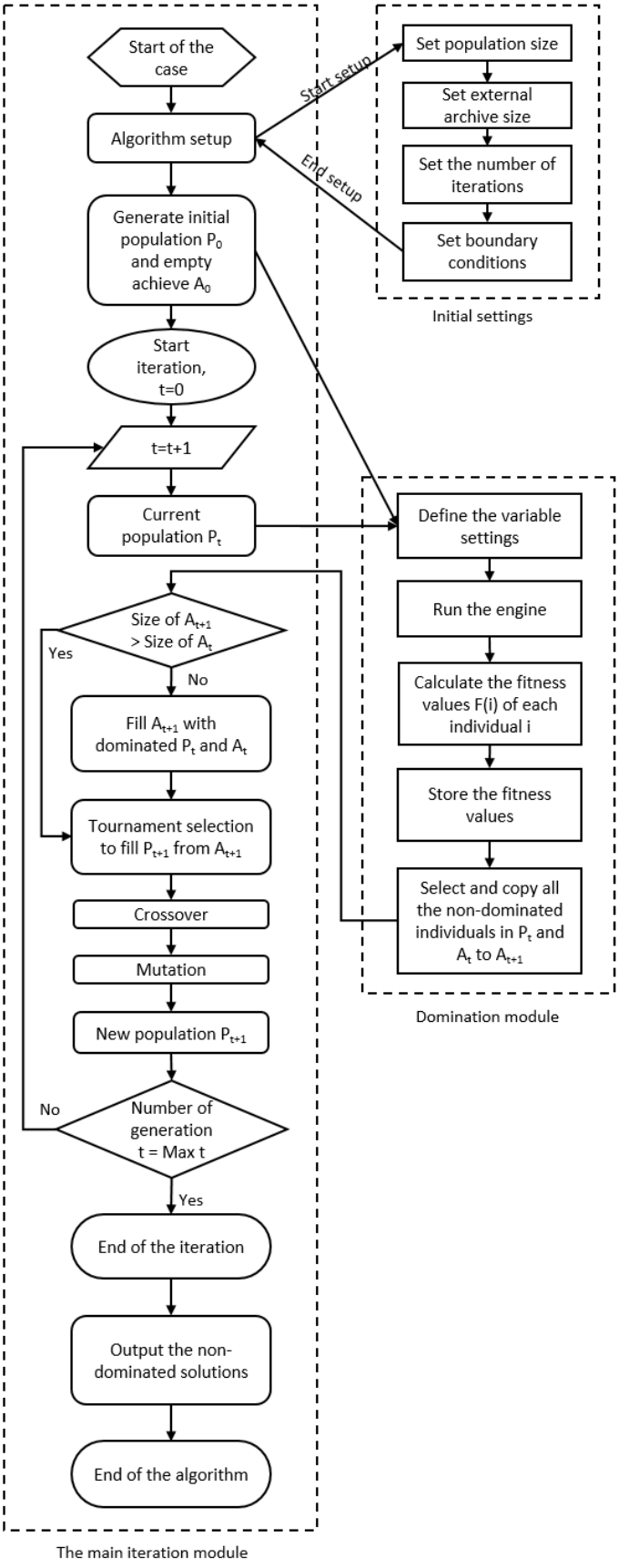


Figure 4.2 Workflow of the SPEA2

4.2.3 CAPSO Optimiser

The theory and benefits of the CAPSO algorithm have been introduced very clearly in section 2.4.3. The location of each particle is updated through a single step:

$$x^i_{(t+1)} = (1-\beta)x^i_{(t)} + \beta G^i_{(t)} + \alpha r_{(t)} \tag{4.6}$$

The randomization component $\alpha r_{(t)}$ here allows the solutions to break away from the fetters caused by any local optimal results. Specifically, $r_{(t)}$ is a random movement within the search-domain for the particle, while α indicates the randomness parameter that would be updated at each iteration. An improvement of the CAPSO algorithm is to reduce the randomness over successive iterations. Thus, a monotonically decreasing function is used to replace the α:

$$\alpha = \alpha_0 \cdot \gamma^t \tag{4.7}$$

Previous studies (Yang, 2014; Gandomi, et al., 2013) have explained that $\alpha_0 \in [0.5, 1]$ as the initial randomness parameter and $\gamma \in [0, 1]$ as the control parameter. γ is usually taken as 0.7 to 0.9 for the standard CAPSO cases. Meanwhile, a larger α_0 is able to achieve a larger α which can make it effective to get rid of the locally optimum results.

As the attraction parameter, $\beta \in [0, 1]$ is used to adjust the convergence speed. For β = 0, the particles will move and update very slowly. Whereas, for β = 1, the particles will converge directly to the current locally optimum results even though they are not the best globally optimum results. The conventional APSO algorithm sets β = 0.5 as a constant because some studies have verified that it could allow the algorithm to work effectively. However, a variable β in each iteration can improve the trade-off between the convergence speed and diversification (to avoid being stuck in locally optimum results).

That is the reason why the CAPSO algorithm employs the chaotic mapping strategy to adjust β value after each iteration. In this chapter, as one of the most classic chaotic mapping strategies, the logistic mapping strategy is applied (Quan et al., 2017). Then, the β value is updated following the equation:

$$\beta_{(t+1)} = \varphi \cdot \beta_{(t)} \cdot (1 - \beta_{(t)}) \tag{4.8}$$

Generally, the CAPSO algorithm consists of three steps.

Step 1 Initialization:

Start the engine, and enable the external ECU bypass to allow the engine to run with the modified variable settings offered by the optimiser. In this chapter, the number of particles in each swarm is selected as 50, the maximum number of iterations is selected as 35, and the search-domain for each variable has been shown in Equation (4.1). Thus, 50 particles, who represent the various engine variable settings, will start from their initial locations generated randomly based on the range of search-domains shown in Equation (4.1). The engine runs with each given group of variable settings for 10s, which is sufficient to allow the measurement readings of BSFC, PMn and PMm to become stabilized.

Step 2 Update for each particle:

The iteration process begins. The value of the fitness function for each particle is calculated based on the real-time measurements. The fitness function is expressed as:

$$\begin{aligned} \min_{IVO,EVC,Spk\_t,Inj\_t\ and\ Rail\_P} J \\ = \text{minimise}\left(\frac{BSFC - BSFC_{min}}{BSFC_{max} - BSFC_{min}} + \frac{PMm - PMm_{min}}{PMm_{max} - PMm_{min}}\right. \\ \left. + \frac{PMn - PMn_{min}}{PMn_{max} - PMn_{min}}\right) \end{aligned} \tag{4.9}$$

where $BSFC_{max}$ and $BSFC_{min}$ indicate the maximum and minimum brake specific fuel consumption found by the particles in current swarm respectively; PMm_{max} and PMm_{min} indicate the maximum and minimum PM emission mass found by the particles in current swarm respectively; PMn_{max} and PMn_{min} indicate the maximum and minimum PM emission number found by the particles in current swarm respectively.

Afterwards, the current locally optimum results corresponding to the most desirable fitness function value are stored. At the end of each iteration, the location of each particle is updated in respect of three indicators, i.e. the particle's current location, the best location in the swarm, and a random factor.

In this chapter, each particle's location is defined as:

$$x_{(t)}^{i} = [IVO_{(t)}^{i}, EVC_{(t)}^{i}, Injection\_timing_{(t)}^{i}, Spark\_timing_{(t)}^{i}, Rail\_pressure_{(t)}^{i}] \quad (4.10)$$

where $t = [1,2,3 \dots N]$ is the index of iterations, N is the number of iterations; $i = [1,2,3 \dots M]$ is the index of particles in each swarm, M is the number of particles in each swarm. $IVO_{(t)}^{i}$, $EVC_{(t)}^{i}$, $Injection\_timing_{(t)}^{i}$, $Spark\_timing_{(t)}^{i}$, and $Rail\_pressure_{(t)}^{i}$ are the engine variable settings for t th iteration and i th particle.

The location of each particle is evolved following the equations:

$$\begin{cases} x_{(t+1)}^{i} = (1-\beta)x_{(t)}^{i} + \beta G_{(t)}^{i} + \alpha r_{(t)} \\ \alpha = \alpha_0 \cdot \gamma^{t} \\ \beta_{(t+1)} = \varphi \cdot \beta_{(t)} \cdot (1-\beta_{(t)}) \end{cases} \quad (4.11)$$

Based on the author's previous researches (Zhang, et al., 2019; Zhou, et al., 2017), α_0 = 0.9 and γ = 0.8 are selected in this chapter. φ and initial $\beta_{(1)}$ are set as φ = 0.4 and $\beta_{(1)}$ = 0.7 respectively to significantly improve the randomization for the generated particles.

Step 3 Stopping conditions reached:

The iteration process is repeated until the pre-set number of iterations or boundary conditions are reached. The final optimum solutions will be obtained from the optimum results of the final iteration.

The workflow of the CAPSO algorithm can be found in Figure 4.3:

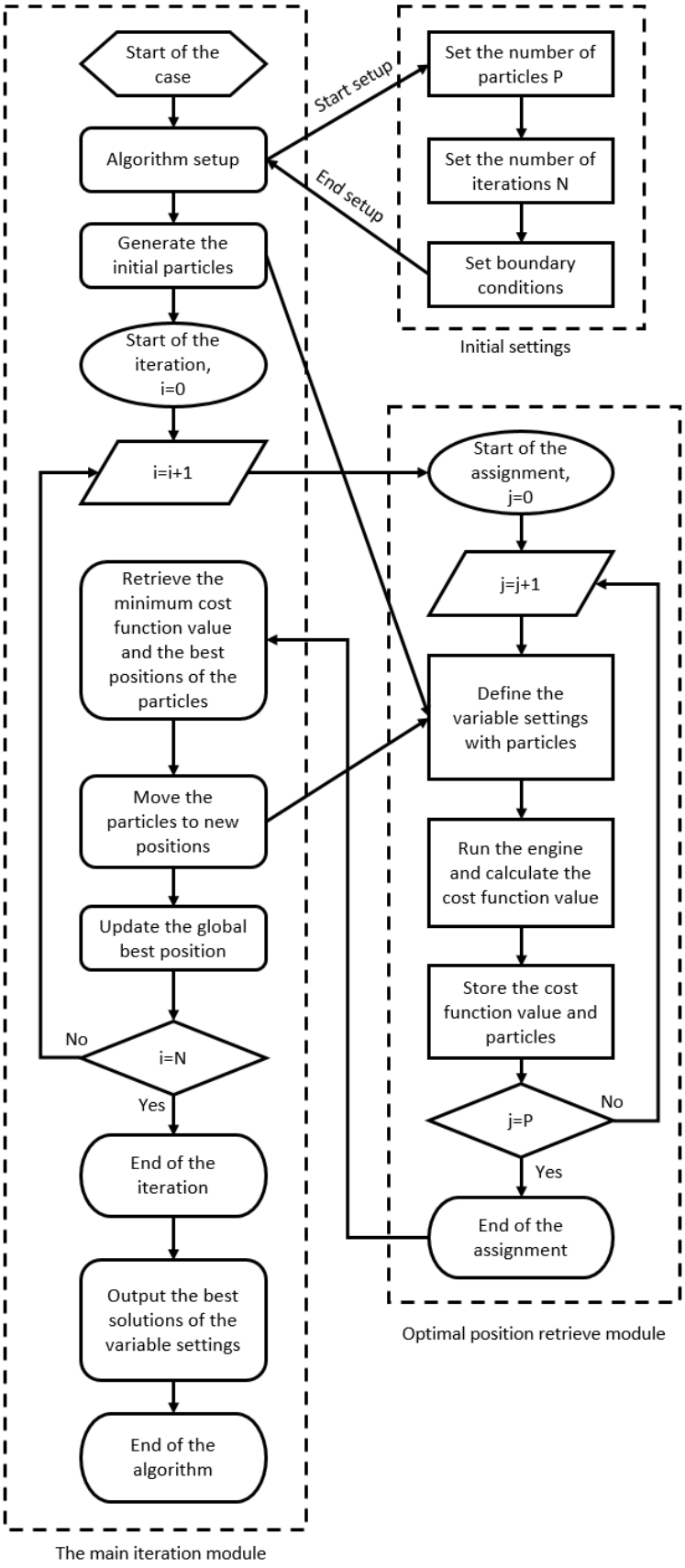


Figure 4.3 Workflow of the CAPSO algorithm

4.3 Experimental Plan

Details about the experimental set-up have been introduced in Chapter 3. Generally, the fuel consumption is measured by the fuel flow meter, and the PM emissions are measured by DMS500. All of these measurements are received by the ES930 and sent to the ES910 as the feedback signals for the calibration. Then, the algorithm-based optimiser embedded in the ES910 will process the program, generate a new group of engine variable settings and send them to the engine in real-time until the globally optimum solutions are found.

More and more meta-heuristic optimisation algorithms have been applied to deal with the engineering problems. Each of them has its own advantages. As the “no free lunch” theorem said (Wolpert and Macready, 1997), it is impossible to discover one algorithm to resolve all kinds of optimisation problems. Different algorithms are inspired by different natural behaviours and designed by different theories. The program, the code, the size of the controller and the number of parameters are all different. From the view of calibration and the feedbacks from engineers, the efficiency (the trade-off between the results and the time consumption) is the priority when evaluating the calibration strategies. Hence, this chapter is going to compare the SPEA2 and CAPSO optimisers based on their convergence speed and optimisation results.

The initial engine variable settings are generated randomly within the range as the Table 4.1 shows. On one hand, it can ensure the diversity of the variable settings. On the other hand, for a brand-new engine, only the constraints of its variable settings can be obtained, while it is impossible to predefine the optimal variable settings at the beginning. Thus, the proposed optimiser is supposed to automatically find the optimum variable settings from scratch, without the guidance by the engineers during the optimisation process.

Given that 10 seconds are enough for the measurements to become stabilized, the engine is defined to run 10 seconds with each given group of variable settings. After 10 s, the calibration process for the first group will stop and get started with the next group of variable settings. Because there are both 50 individuals in each population for SPEA2 and 50 particles in each swarm for CAPSO, 50 groups of variable settings are generated by the optimiser and sent to the engine in each iteration in real-time. Therefore, the calibration process will repeat 50 times and cost $10\,s \times 50 = 500\,s$ (8.33 min) for one iterative generation. After that, the engine variable settings for the next iterative generation will be produced based on the algorithms mentioned above and fed into the engine. The calibration process will repeat 50 times again. 35 iterative loops are applied based on the experimental studies achieved. The calibration process will stop in advance when the optimum results converge. Thus, the whole calibration process will cost no more than $500\,s \times 35 = 17500\,s$ (4.86 h). As for the model-based calibration approach using optimisation algorithm mentioned in section 2.1.2, it will cost the same time for calibration together with weeks to generate a mean value model.

It should be noted that, the number of iterative loops and the number of individuals or particles for algorithms are predefined as the compromise between the solution diversity and the convergence speed. Although more individuals or particles for algorithms can make greater use of the search-domains, the longer time it takes for each iterative loop. Hence, to compare the convergence speed of optimisers, this chapter compares the time consumption required by two optimisers with the same number of individuals and particles.

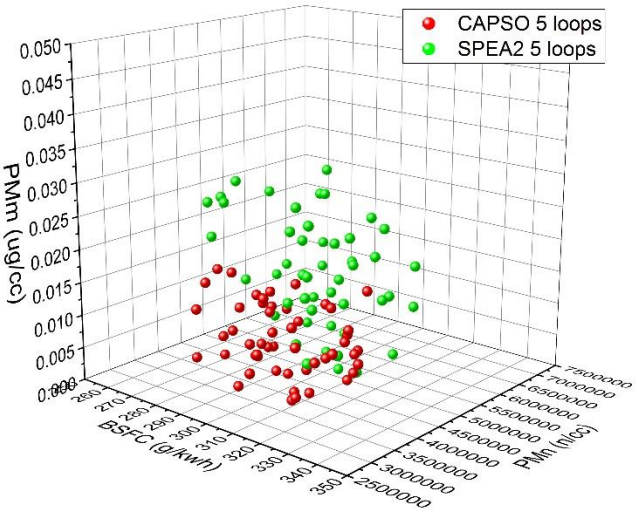
4.4 Results and Discussion

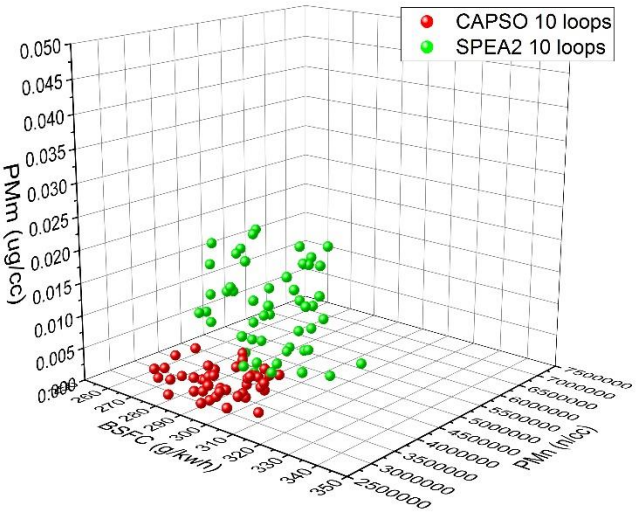
To validate the non-model-based calibration approaches, the calibration results for three different engine operating conditions with: 1) 1500 rpm and 5.5 bar BMEP; 2) 1500 rpm and 8.0 bar BMEP; 3) 1700 rpm and 4.5 bar BMEP are shown here. To evaluate the efficiencies of these calibration approaches, their convergence speed and optimisation results are compared as follows.

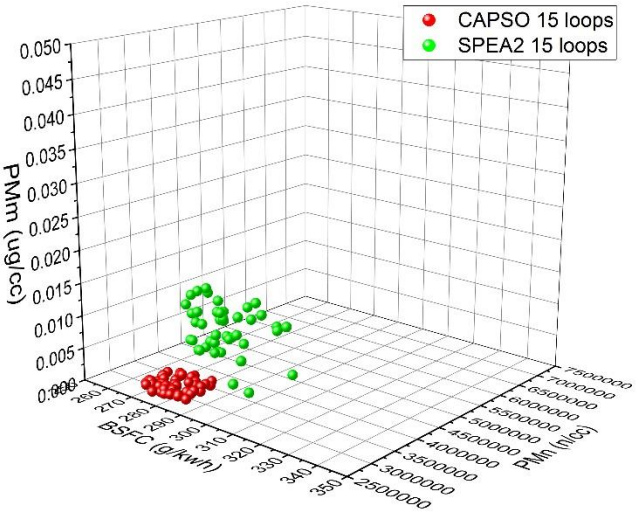
4.4.1 Comparison of Convergence Speed

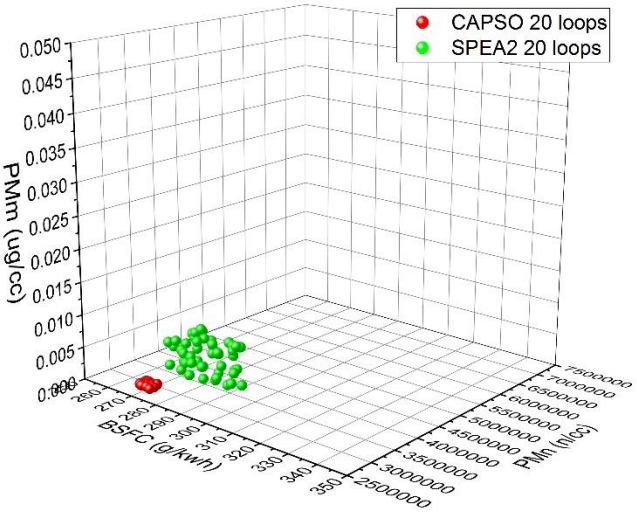
The calibration results found by two optimisers have been shown in Figure 4.4 to 4.6 as the searching process goes by. Since 35 iterative loops are applied, the figures show the optimum results after every five iterative loops. The three-dimensional figures are employed to demonstrate the calibration results given that there are three objectives. And the lower left corner in each figure is the origin of coordinates, which indicates the lower fuel consumption and lower PM emissions. The point who is closer to the lower left corner represents the better calibration results, i.e. locally optimum results. Because each population of SPEA2 has 50 individuals and each swarm of CAPSO algorithm has 50 particles, there are 50 points in each figure and each point represents a solution for engine variable settings.

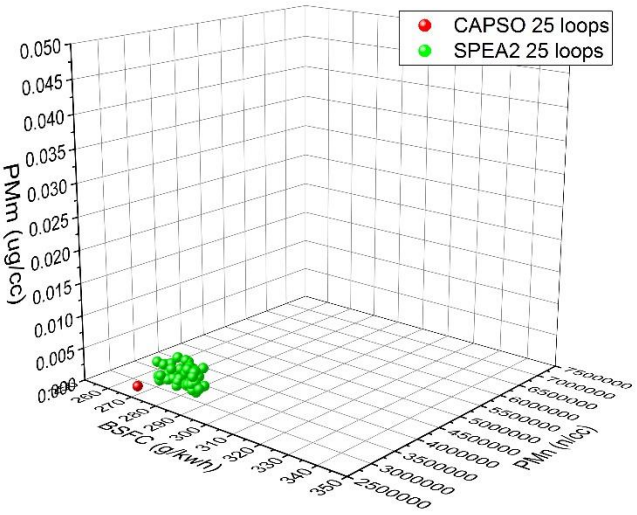
Case 1). 1500 rpm and 5.5 bar BMEP:

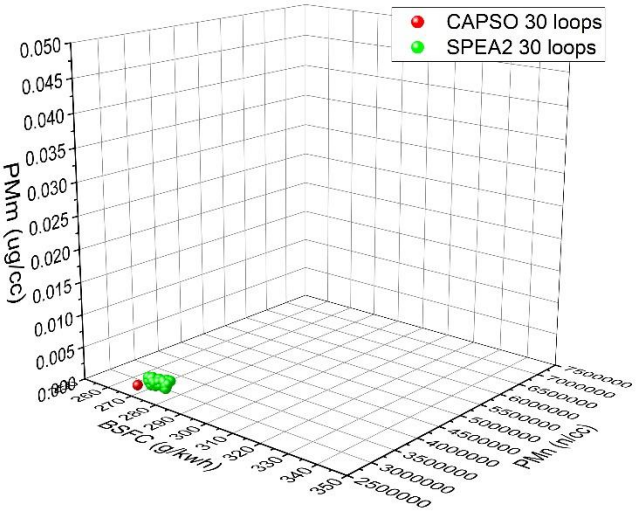







CAPSO 20 loops
SPEA2 20 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)


CAPSO 25 loops
SPEA2 25 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)


CAPSO 30 loops
SPEA2 30 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)

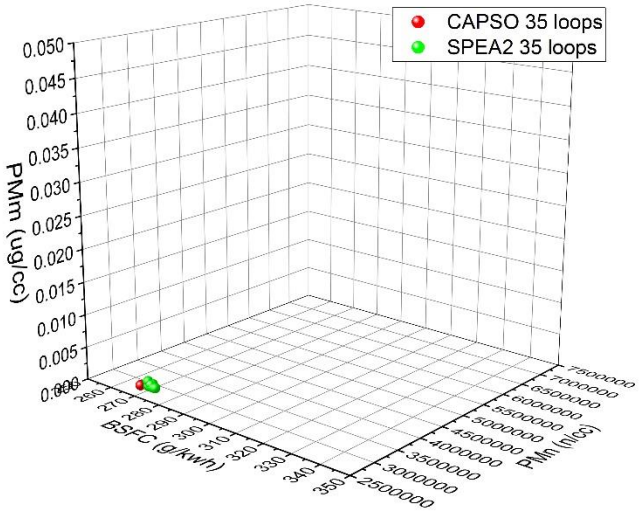
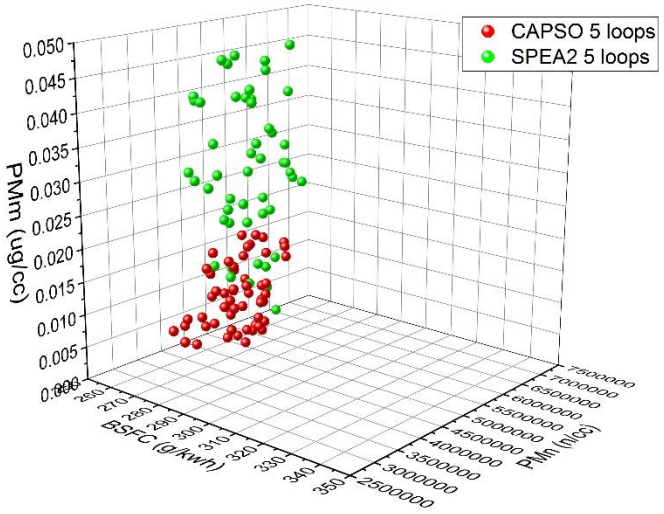
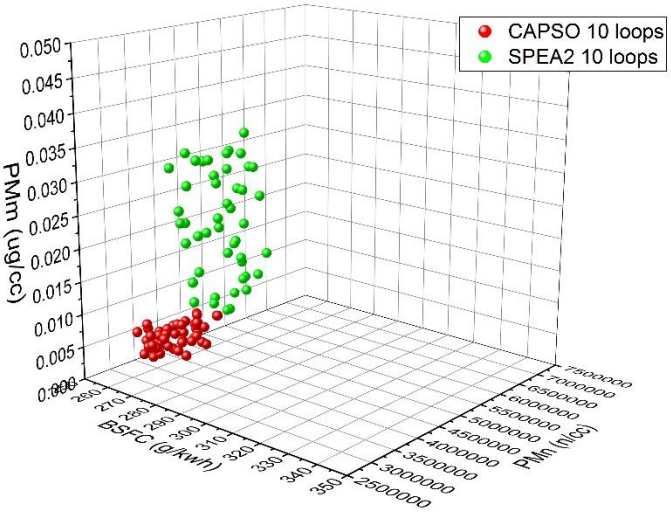
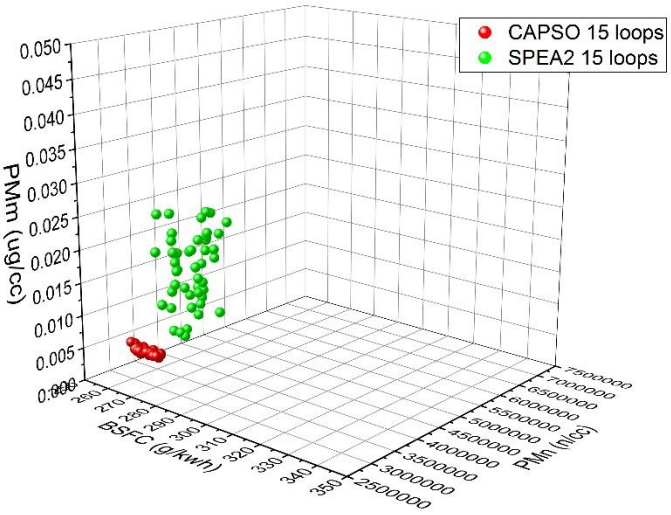


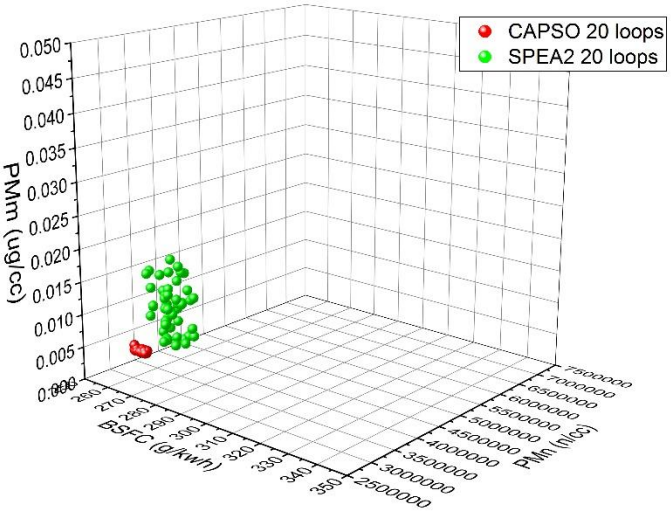
Figure 4.4 Calibration results of the Case 1 after every five iterative loops

Case 2). 1500 rpm and 8.0 bar BMEP:




CAPSO 10 loops
SPEA2 10 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)


CAPSO 15 loops
SPEA2 15 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)


CAPSO 20 loops
SPEA2 20 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)

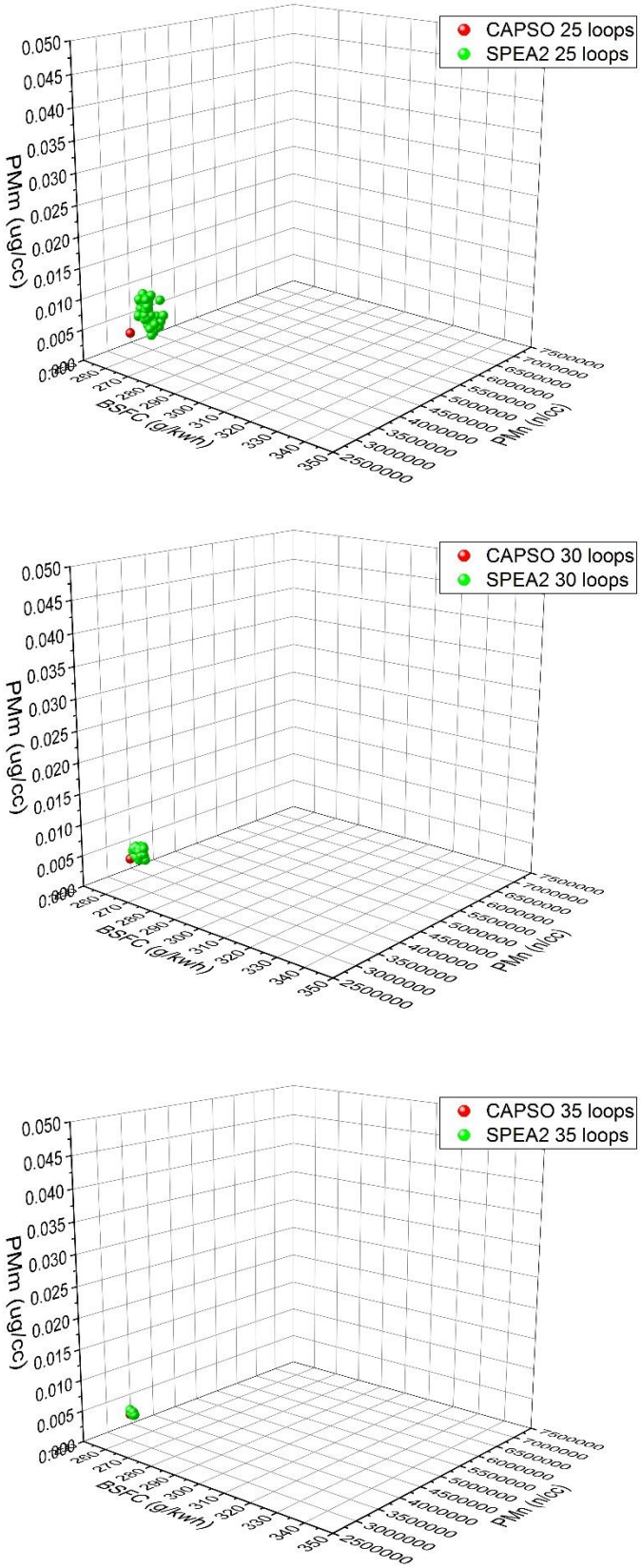
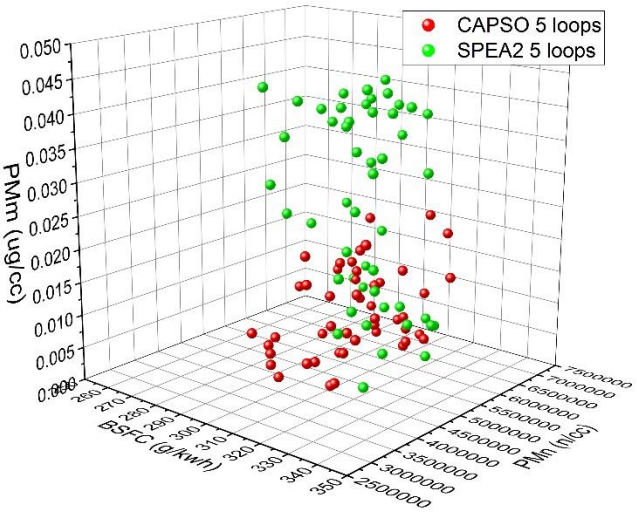
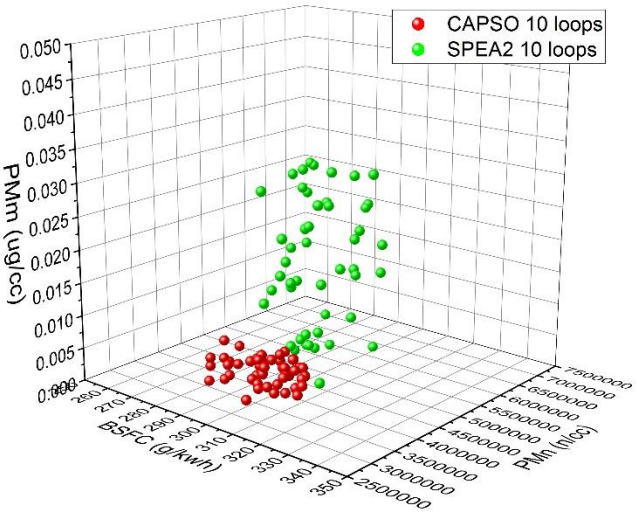
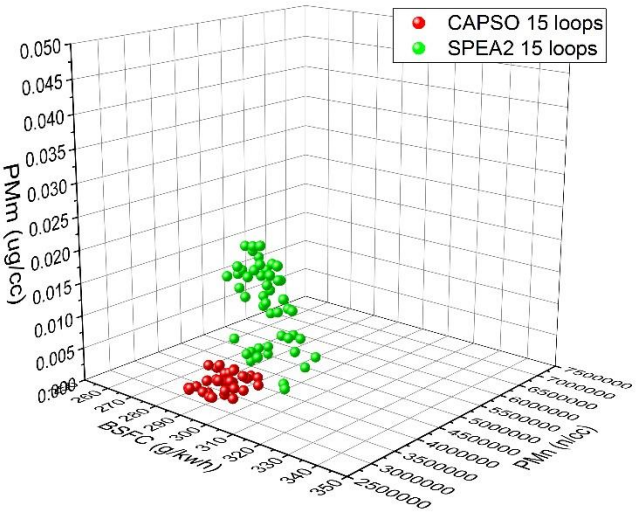


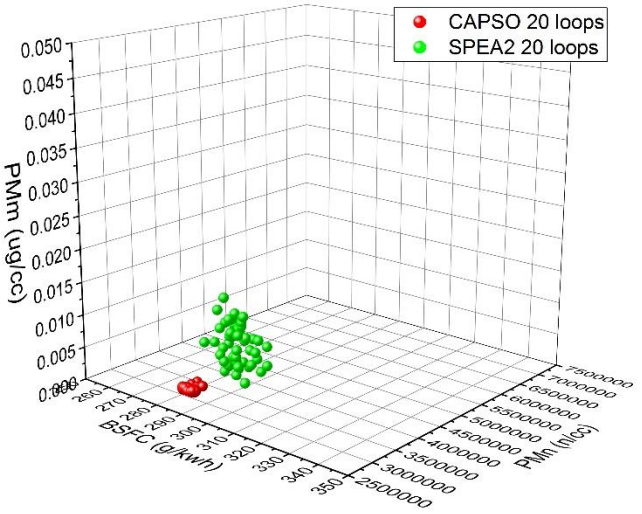
Figure 4.5 Calibration results of the Case 2 after every five iterative loops

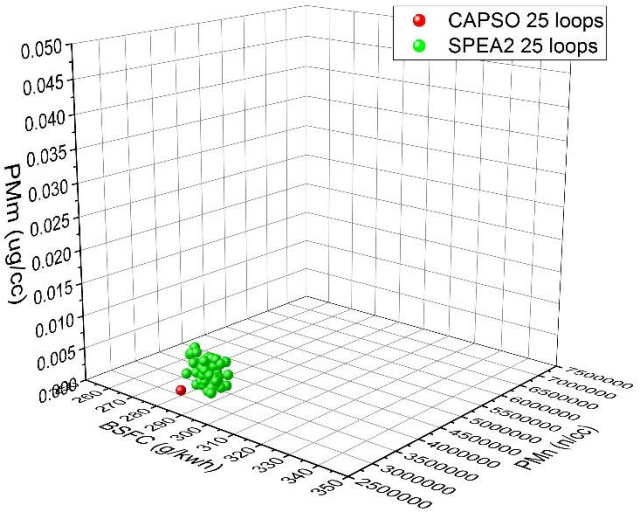
Case 3). 1700 rpm and 4.5 bar BMEP:

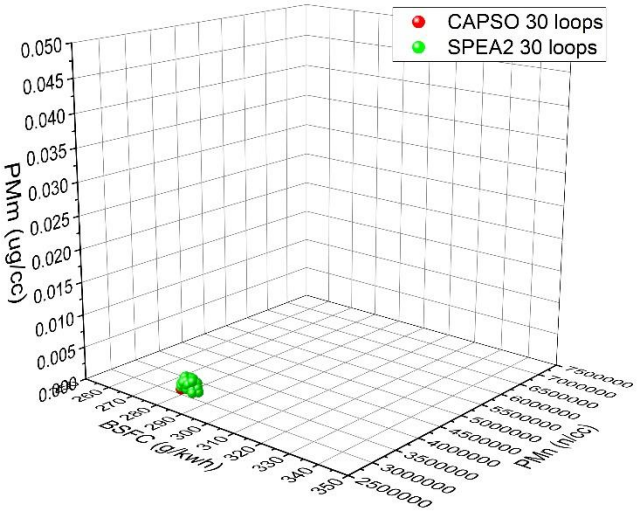







CAPSO 20 loops
SPEA2 20 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)


CAPSO 25 loops
SPEA2 25 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)


CAPSO 30 loops
SPEA2 30 loops
PMm (ug/cc)
BSFC (g/kwh)
BMEP (Pa)

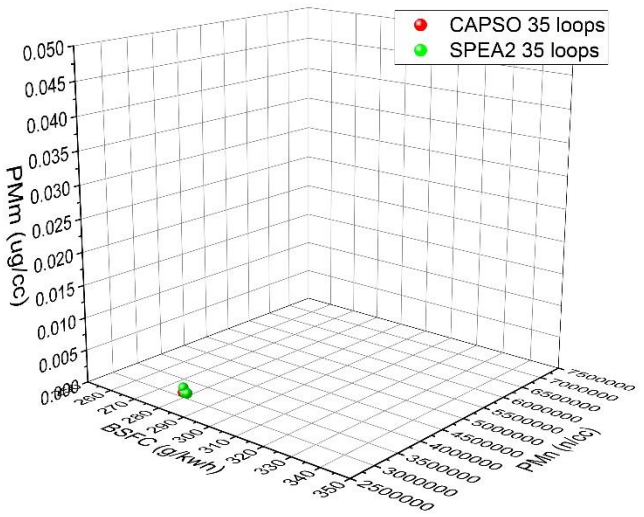


Figure 4.6 Calibration results of the Case 3 after every five iterative loops

As can be seen from the figures, the range of the calibration results becomes narrower as the iteration process goes by. It confirms that the proposed non-model-based calibration approach can automatically find the better calibration solutions. The CAPSO-based calibration results have already converged to the globally best solution within 25 iterative loops. Whereas, the SPEA2-based calibration results cost at least 35 iterative loops to converge to a very small range. Thus, the CAPSO optimiser is able to save at least 28.6% of the time consumption. That is because SPEA2 and CAPSO algorithms are inspired by different natural behaviours and designed by different theories as mentioned above. The most remarkable advantage of the CAPSO algorithm is its fast convergence speed. Once the CAPSO algorithm finds a best solution in the swarm, particles will converge to that solution gradually while part of them keep searching the surrounding areas to retain the diversity. The iteration of SPEA2 is based on the mutation and crossover, which means the “offspring” solutions may be even worse than the “parent” solutions, and it will cost more time until the best “offspring” solutions are generated.

The CAPSO algorithm will output only one globally optimum solution based on the scaling factors or weights predefined by users. Meanwhile, the SPEA2 will output a set of solutions along the Pareto front, because all of these solutions are equally good and do not dominate each other. Accordingly, the SPEA2-based calibration results cannot converge to one solution, but users can select the most desirable solution based on their requirements. From the view of calibration, the CAPSO-based calibration approach is regarded as a more efficient calibration approach, because only one best solution is going to be stored in the ECU. Therefore, the CAPSO algorithm can speed up the calibration process.

4.4.2 Comparison of Calibration Results

Because the engine employed in this chapter is a commercial engine, the default calibration maps offered by the OEM are available to be viewed from the ECU. Thus, they are used as the benchmark for the comparison. In order to evaluate the proposed calibration approaches, comparisons among the results of the proposed calibration approaches and the default ECU maps are shown in Figure 4.7 to Figure 4.9.

Case 1). 1500 rpm and 5.5 bar BMEP:

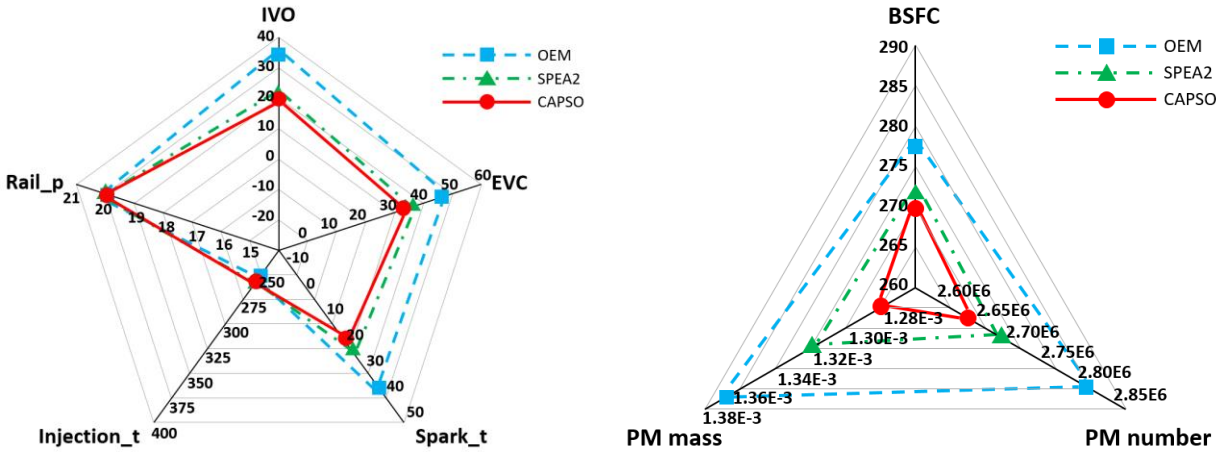


Figure 4.7 Final calibration results of the Case 1

Case 2). 1500 rpm and 8.0 bar BMEP:

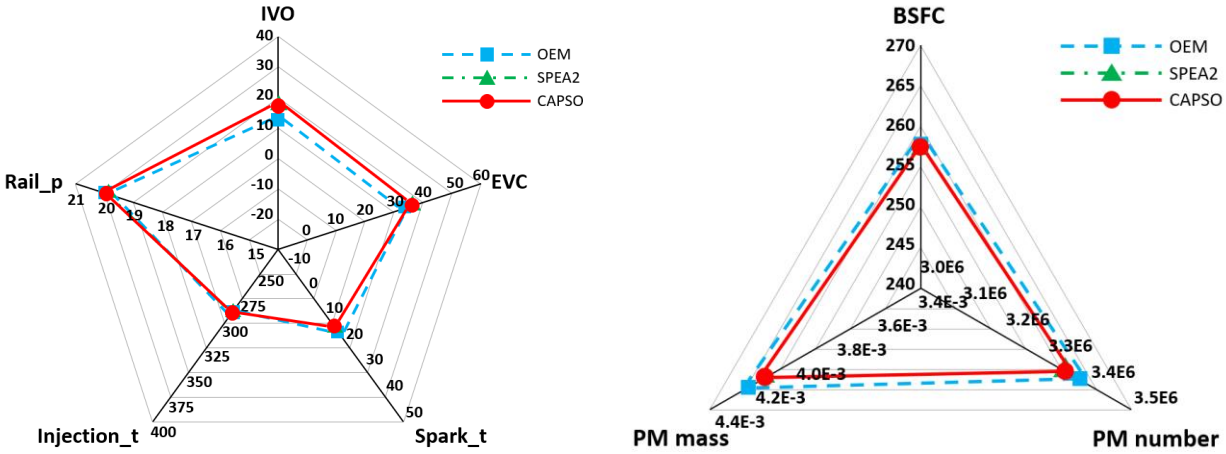


Figure 4.8 Final calibration results of the Case 2

Case 3). 1700 rpm and 4.5 bar BMEP:

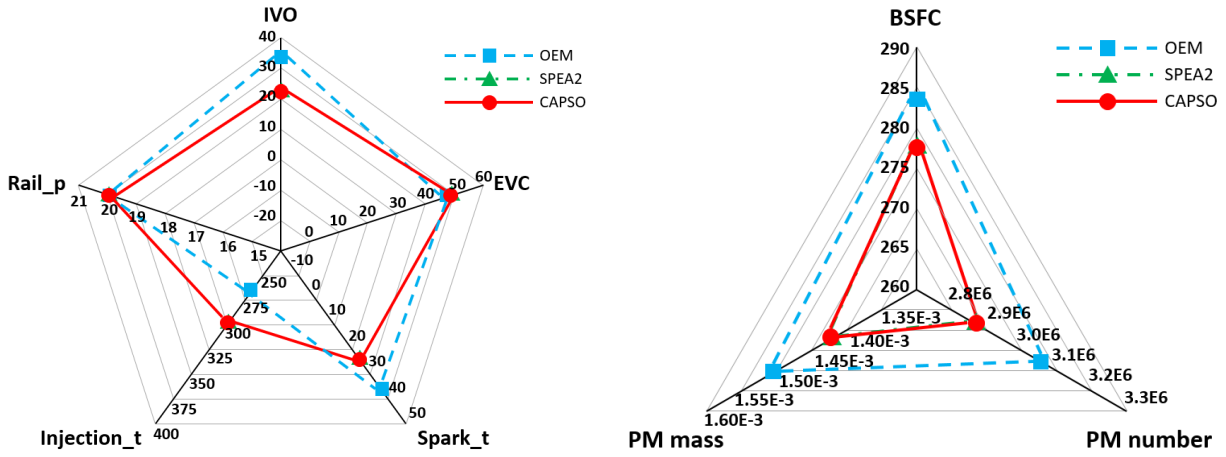


Figure 4.9 Final calibration results of the Case 3

The left-hand figures in Figure 4.7, 4.8 and 4.9 indicate the engine variable settings, while the right-hand figures in Figure 4.7, 4.8 and 4.9 indicate the corresponding engine performances. Because both algorithms involve random operations during the calibration process, it is not fair to compare their performance through a single case. Therefore, both calibration approaches are implemented for six times repeatability tests, and the most representative calibration results found by algorithms after 35 iterative loops are used for

evaluation. The details can be found in Table 4.2. This chapter is aimed at improving the automation and intelligence level of the calibration approach. The variables and objectives can be added and replaced concerning different requirements. Once the proposed calibration approach found the lower BSFC and PM emissions compared to the OEM data, it comes to the conclusion that the proposed calibration approach is able to automatically locate the optimum calibration results.

Table 4.2 Detailed comparison among the representative results of the proposed calibration approaches and the default ECU maps through six times repeatability tests

Case		IVO (CAD)	EVC (CAD)	Spk_t (CAD)	Inj_t (CAD)	Rail_p (MPa)	BSFC (g/kwh)	PMn (n/cc)	PMm (μg/cc)
1	OEM	35	47	37	250	20	278	2.80E+06	1.37E-03
	SPEA2	21	36	21	255	20	272	2.68E+06	1.32E-03
	Improvement						**2.2%**	**4.3%**	**3.6%**
	CAPSO	20	33	18	255	20	270	2.63E+06	1.28E-03
	Improvement						**2.9%**	**6.1%**	**6.6%**
2	OEM	13	35	14	289	20	258	3.38E+06	4.20E-03
	SPEA2	18	37	12	289	20	257	3.33E+06	4.15E-03
	Improvement						**0.4%**	**1.5%**	**1.2%**
	CAPSO	18	37	12	289	20	257	3.33E+06	4.15E-03
	Improvement						**0.4%**	**1.5%**	**1.2%**
3	OEM	35	47	39	265	20	284	3.07E+06	1.51E-03
	SPEA2	23	50	25	300	20	278	2.89E+06	1.43E-03
	Improvement						**2.1%**	**5.9%**	**5.3%**
	CAPSO	23	50	25	300	20	278	2.89E+06	1.43E-03
	Improvement						**2.1%**	**5.9%**	**5.3%**

As can be seen from the figures and the table above, the calibration results obtained by two algorithms are very close to each other and they are both better than the default ECU settings. As for the case 1, it is found that the CAPSO-based calibration approach can find the engine variable settings for lower fuel consumption and lower PM emissions compared to the SPEA2-based calibration approach. In terms of the case 2, both two algorithms result in the same optimum calibration results and they are very close to the default ECU settings. As for the case 3, the algorithm-based calibration approaches still find the same optimum results. But they are different from the default ECU settings and can reduce the fuel consumption as well as PM emissions. The improvements of the fuel consumption, PM emission number and mass are up to 2.9%, 6.1% and 6.6% respectively over the default ECU settings.

4.5 Summary

In this chapter, the intelligent non-model-based calibration approach for ICEs has been developed. To validate the non-model-based calibration approach by a V6 GDI engine test bench, the Simulink-based optimisers using the SPEA2 and CAPSO algorithms have been developed. They were applied to the real engine via RCP and external ECU bypass technology.

The calibration task was formed as a multi-objective and multi-variable problem, i.e. to optimise the BSFC and PM emissions (PMn and PMm) through tuning the engine variables: IVO, EVC, spark timing, injection timing and rail pressure. The experiment results show that:

- The intelligent non-model-based calibration approach can automatically find the optimum engine variable settings from scratch. It does not rely on the engine model or massive experimental data. It is a more efficient and economical way to calibrate an engine.
- The intelligent non-model-based calibration approach can even find the variable settings for lower BSFC (up to 2.9% reduction), lower PM emission number (up to 6.1% reduction), and lower PM emission mass (up to 6.6% reduction), compared to the default ECU settings.
- Compared to the SPEA2-based calibration approach, the CAPSO-based calibration approach is able to find better engine variable settings with faster speed (save at least 28.6% of the time consumption).

The conclusions above show that the developed non-model-based calibration approach has great potential to improve the automation level of the engine calibration. It is a promising approach to help the calibration engineers get rid of the numerous engine mapping tests.

CHAPTER 5 SELF-ADAPTIVE AIR/FUEL RATIO CONTROL STRATEGY WITH A PI-LIKE FUZZY KNOWLEDGE BASED CONTROLLER

The work presented in this chapter has been published in the journal “Proceedings of the IMechE, Part D: Journal of Automobile Engineering” (Li et al., 2019a). This chapter proposes a new concept of the PI-like fuzzy knowledge based controller (FKBC) with the self-adaptive capability, high robustness and rapid development capability to regulate the AFR for GDI engines. The proportional term, integral term, and the output signal of the PI-like FKBC are non-linear functions of the input signals and self-tuned in real-time at operation conditions, which can reduce the effort to be spent in calibrating controller parameters. The experimental studies show that the proposed controller can regulate the AFR at the stoichiometric value with less settling time and less oscillation compared with the conventional lookup-table-based controller for transient operations.

A general introduction to the background of this study is presented first, followed by an introduction to the AFR dynamics. Then, the Simulink-based PI-like FKBC is developed and implemented on a V6 GDI engine test bench through the ETAS RCP platform. The experimental results are shown in the following section, and the comparison between the PI-like FKBC and the conventional lookup-table-based PI controller is given concerning the transient control performances, including the settling time and the oscillation. The final section presents a summary.

5.1 Introduction

As introduced in section 2.5, the AFR should be maintained as the stoichiometric value (14.7) for both steady-state and transient operations, in order to achieve the minimum emissions and the optimal trade-off between the power output and the fuel consumption. If the oscillations of the AFR continue for a long period with dramatic overshoots, they can affect the system security and impair the thermal efficiency as well as the catalyst efficiency. As for the feedback control loop of the AFR control system, the current commercial engine management system uses the lookup-table-based PI feedback controller to adjust the injected fuel mass and regulate the AFR. This method is time-consuming due to the massive parameter calibration works, and it is not robust to engine parameter uncertainties and time varying dynamics. Besides, it cannot work efficiently for the highly non-linear engine systems (Li et al., 2019a).

Therefore, the FKBC is applied in this chapter to optimise the AFR feedback control scheme. The main advantages of the FKBC includes: improvement of the system stability; self-adaptive capability; high robustness; fault-tolerant capability; simple structure; easy tuning; no requirement for historical data and training; excellent tracking performance; be able to figure out the problem of the black-box system. The PI-like FKBC is expected to reduce the effort for the parameter tuning and calibration compared to the conventional lookup-table-based PI controller. The proposed controller is built by the MATLAB/Simulink and co-operates with the ETAS RCP facilities for the real-time implementation. Its transient performance is investigated on a production GDI engine via various operating conditions (Li et al., 2019a).

5.2 Air/Fuel Ratio Dynamics

5.2.1 Cylinder Air Charge Dynamics

Generally, the structure of the GDI engine with the AFR control system after the supercharger is shown in Figure 5.1. It is representative of the GDI engine today and will be representative over the next decade.

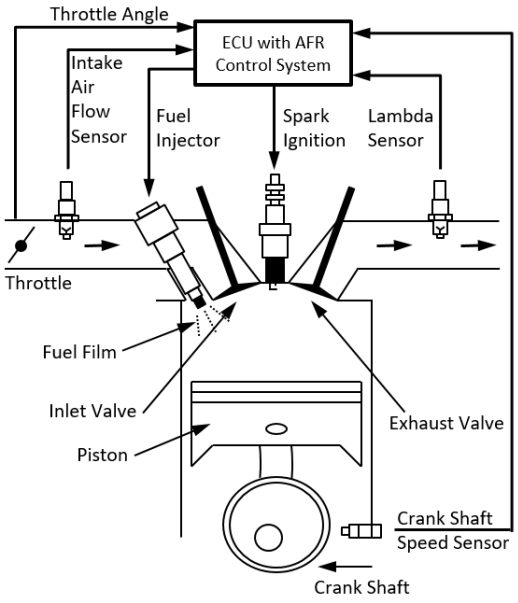


Figure 5.1 Structure of the GDI engine with the AFR control system

In the air charge dynamic subsystem, the throttle position (θ) and the engine speed (ω) will decide the air mass flow through the throttle ($\dot{m}_{th}$). Then, the intake air mass flow $\dot{m}_a$ is calculated based on $\dot{m}_{th}$ and the air mass flow at the cylinder inlet port ($\dot{m}_o$) (Li et al., 2019a):

$$\dot{m}_a = \dot{m}_{th} - \dot{m}_o \tag{5.1}$$

The derivation of the manifold pressure (p_m) is estimated through the ideal gas law $PV = MRT$ (Li et al., 2019a):

$$\dot{p}_m = \frac{RT_m}{V_m} \cdot \dot{m}_a = \frac{RT_m}{V_m} (\dot{m}_{th} - \dot{m}_o) \tag{5.2}$$

where R represents the ideal gas constant, T_m and V_m are the intake manifold temperature and volume respectively (Li et al., 2019a).

$\dot{m}_o$ is decided from the air intake speed-density principle. It can be described by the intake manifold pressure and engine speed (Li et al., 2019a):

$$\dot{m}_o = \frac{V_C \eta}{4\pi R T_m} p_m \omega \tag{5.3}$$

where V_C represents the cylinder volume, η is the volumetric efficiency, ω is the engine speed (Li et al., 2019a).

5.2.2 AFR Control Plant

For the GDI engine, the fuel is injected from a common rail and injector into the cylinder directly and immediately. Because of the mechanical structure of the GDI engine, the wall wetting effect could be ignored. Thus, this chapter assumes that the mass of fuel from the injector is equal to that into the cylinder. Here, $\dot{m}_{fc}$ denotes the fuel mass flow entering the cylinder. Then, the AFR is defined as (Li et al., 2019a):

$$AFR = \frac{\dot{m}_o}{\dot{m}_{fc}} \tag{5.4}$$

Based on these formulae, $\dot{m}_{fc}$ and AFR are defined as the output and input of the controller respectively. The objective is to generate the $\dot{m}_{fc}$ signal to ensure the measured AFR is close to the target AFR in the presence of uncertain parameters. The

$\dot{m}_{fc}$ signal contains the feedforward reference fuel injection $\dot{m}_{ff}$ and the feedback regulation fuel injection $\dot{m}_{fb}$ (Li et al., 2019a):

$$\dot{m}_{fc} = \dot{m}_{ff} + \dot{m}_{fb} = \frac{\hat{\dot{m}}_o}{AFR_{ref}} + \dot{m}_{fb} \tag{5.5}$$

where $\hat{\dot{m}}_o$ is the estimation of the air charge mass rate $\dot{m}_o$ in advance of the injection delay. Therefore, the feedforward reference fuel injection is decided by the target AFR_{ref} and the air charge estimation $\hat{\dot{m}}_o$. At the same time, the feedback regulation fuel injection $\dot{m}_{fb}$ corrects this estimated value based on the real-time AFR measured by the EGO (lambda) sensor. Due to the uncertain parameters, a PI-like FKBC for the feedback fuel injection control with the self-tuning capability is designed in the next section (Li et al., 2019a).

The workflow of the AFR control system is presented in Figure 5.2 (Li et al., 2019a).

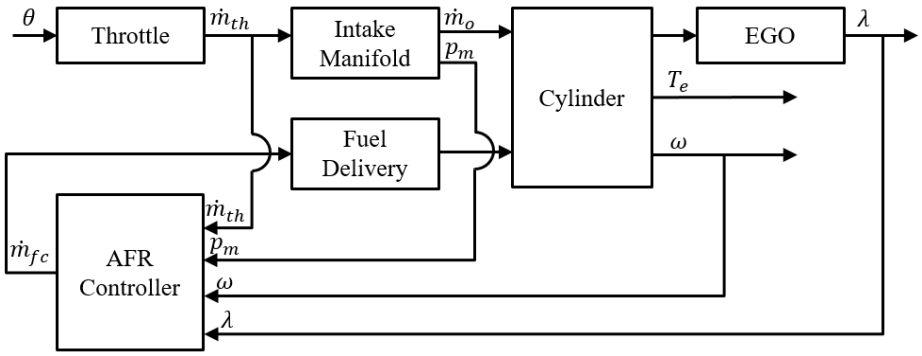


Figure 5.2 Workflow of the AFR control system

As can be seen from the Figure 5.2, T_e is the torque generated by the engine. The AFR is expressed as the air to fuel equivalence ratio, lambda (λ). λ is the feedback signal for the AFR controller in this chapter. It is the ratio of the actual AFR to the stoichiometry for a given mixture (Li et al., 2019a).

$$\lambda = \frac{AFR}{AFR_{stoich}} \tag{5.6}$$

where λ = 1.0 means the AFR has been regulated as the ideal stoichiometric value (Li et al., 2019a).

5.3 Design of the PI-like FKBC

A flowchart of the design procedures for the PI-like FKBC developed in this chapter is shown in Figure 5.3 (Li et al., 2019a).

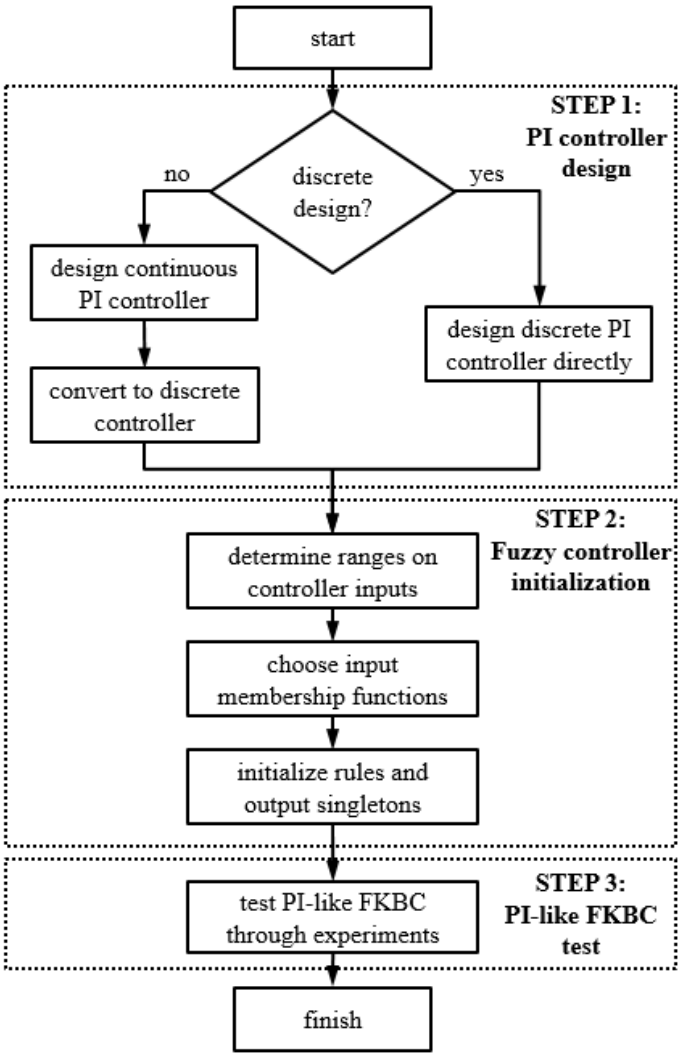


Figure 5.3 Design procedure of the PI-like FKBC

There are three basic steps. In the first step, the PI-like FKBC is converted from a conventional linear PI controller. The second step is the fuzzy controller initialization, including three stages: fuzzification stage, control rule base formulation, and defuzzification stage (Li et al., 2019a). Finally, the PI-like FKBC is tested through experiments.

5.3.1 Construction of the Conventional Discrete PI Controller

The output of a conventional PI controller is (Li et al., 2019a):

$$u_{PI}(s) = \left(K_p^c + \frac{K_i^c}{s}\right) E(s) \tag{5.7}$$

where $E(s)$ is the error signal between the reference and the process output; K_p^c and K_i^c represent the continuous proportional gain and integral gain respectively; $u_{PI}(s)$ is the output signal of the PI controller. In this chapter, the equation should be converted to the discrete format through the bilinear conversion (Li et al., 2019a):

$$s = \frac{2(z-1)}{T(z+1)} \tag{5.8}$$

where T is the sampling time. Equation (5.7) can be rewritten as (Li et al., 2019a):

$$u_{PI}(z) = \left(K_p^c - \frac{K_i^c T}{2} + \frac{K_i^c T}{1 - z^{-1}}\right) E(z) \tag{5.9}$$

Let

$$\begin{cases} K_p^d = K_p^c - \dfrac{K_i^c T}{2} \\ K_i^d = K_i^c T \end{cases} \tag{5.10}$$

where K_p^d and K_i^d are the current discrete proportional gain and integral gain respectively (Li et al., 2019a).

After using the inverse z-transform to convert the equation back to a discrete-time domain, Equation (5.9) can be rewritten as (Li et al., 2019a):

$$u_{PI}(k) - u_{PI}(k-1) = K_p^d[e(k) - e(k-1)] + K_i^d T e(k) \tag{5.11}$$

After dividing the equation by T, Equation (5.11) can be rewritten as:

$$\Delta u_{PI}(k) = K_p^d e_v(k) + K_i^d e_p(k) \tag{5.12}$$

where

$$\begin{cases} \Delta u_{PI}(k) = \dfrac{u_{PI}(k) - u_{PI}(k-1)}{T} \\ e_v(k) = \dfrac{\mathrm{e}(k) - \mathrm{e}(k-1)}{T} \\ e_p(k) = e(k) \end{cases} \tag{5.13}$$

More exactly, $\Delta u_{PI}(k)$ is the incremental control output of the PI controller, $e_p(k)$ stands for the error, and $e_v(k)$ stands for the rate of the change of error. The equation can be rewritten as (Li et al., 2019a):

$$u_{fb} = u_{PI}(k) = u_{PI}(k-1) + T\Delta u_{PI}(k) \tag{5.14}$$

The conventional lookup-table-based PI controller is designed based on the Equation (5.14), where u_{fb} is the output signal of the PI controller, namely the feedback fuelling signal. Then, the AFR control problem can be mathematically formulated as (Li et al., 2019a):

$$\begin{cases} \min\limits_{K_p^d, K_i^d} IAE = \sum\limits_{k=1}^{n} T \cdot |e(k)| \\ e(k) = \lambda_{ref}(k) - \lambda(k) \end{cases} \tag{5.15}$$

where $\lambda_{ref}(k)$ is the target λ value; $\lambda(k)$ is actual λ value; IAE is the integral of the absolute error (Li et al., 2019a).

However, as introduced in section 2.2.2, the conventional lookup-table-based PI controller cannot meet the requirements for the further development of modern engine control systems. On one hand, this controller usually costs lots of time for the parameter calibration and it is not robust. The Ziegler-Nichols rules are widely used to generate a series of approximate values as reference points followed by the hand-tuning method through "guess and check" for each engine operating condition. The controller has to work based on the constant proportional and integral gains stored in the lookup table under a particular operating condition. The controller gains at non-calibrated conditions are calculated using the linear interpolation method based on the calibrated values. On the other hand, the lookup-table-based PI controller is unable to perform well for the highly non-linear systems, time varying dynamic systems, and the particularly complex systems with uncertain parameters (Li et al., 2019a).

5.3.2 Construction of the PI-like FKBC

In terms of designing the PI-like FKBC, the item $T\Delta u_{PI}(k)$ in Equation (5.14) is replaced by the fuzzy control item $K_u \Delta u_{PI}(k)$, so the equation becomes (Li et al., 2019a):

$$u_{fb} = u_{PI}(k) = u_{PI}(k-1) + K_u \Delta u_{PI}(k) \tag{5.16}$$

where K_u is a fuzzy control scaling factor that will be determined later. A larger K_u factor can reduce the convergence time but raise the oscillation of the system. Whereas, a smaller K_u gain will lead to the less sensitive response. Equation (5.16) is the final equation for the design of the proposed discrete PI-like FKBC (shown in Figure 5.4) (Li et al., 2019a).

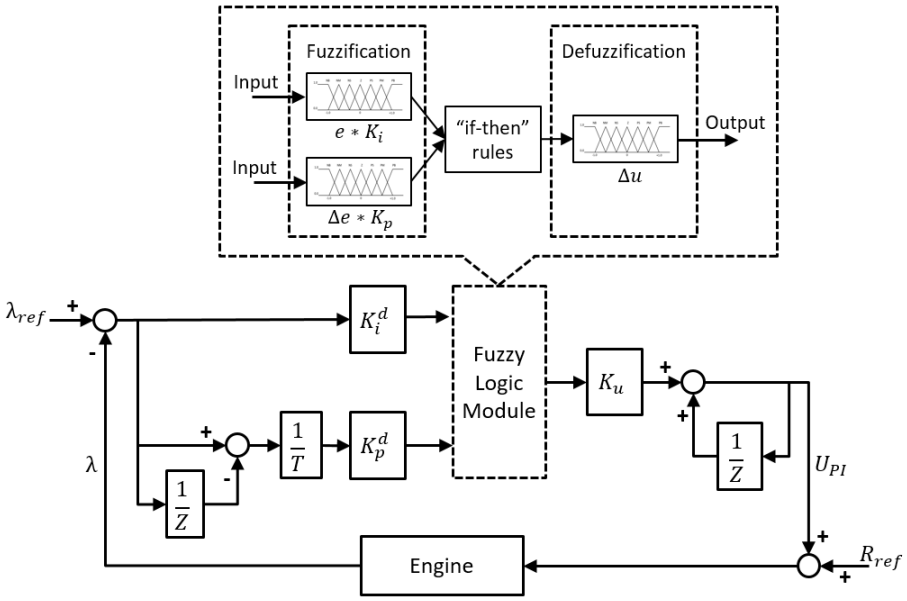


Figure 5.4 Block diagram of the PI-like FKBC

The PI-like FKBC in this chapter employs two input signals, i.e. the error signal $e_p(k)$ multiplied by K_i^d, and the rate of the change of error signal $e_v(k)$ multiplied by K_p^d. For the AFR control, the $e_p(k)$ value is the gap between the target λ value and the actually measured λ value. The individual inputs $X_1(k)$ and $X_2(k)$ can be derived as follows (Li et al., 2019a):

$$X_1(k) = K_i^d\big(\lambda_{ref}(k) - \lambda(k)\big) \tag{5.17}$$

$$X_2(k) = K_p^d \left(\frac{\Delta\lambda(k) - \Delta\lambda(k-1)}{T} \right) \tag{5.18}$$

where λ_{ref} is the desired λ value. It is a constant value equal to 1.0 (Li et al., 2019a).

In the meantime, $\Delta u_{PI}(k)$ in Equation (5.16) is defined as the single output term, which is the function in $[X_1(k), X_2(k)]$ (Li et al., 2019a).

Then, the input and output variables should be specified as the fuzzy set values. In this chapter, the membership functions for the input signals $X_1(k)$ and $X_2(k)$, as well as the output incremental signal $\Delta u_{PI}(k)$ are defined over [-1, 1]. Figure 5.5 gives the simplest triangular membership function used for input signals and output signal with the height 1, which occurs at the points -1.00, -0.65, -0.33, 0.00, 0.33, 0.65, and 1.00. This is the simplest method to design the membership function in order to reduce the time spent in designing the controller (Li et al., 2019a).

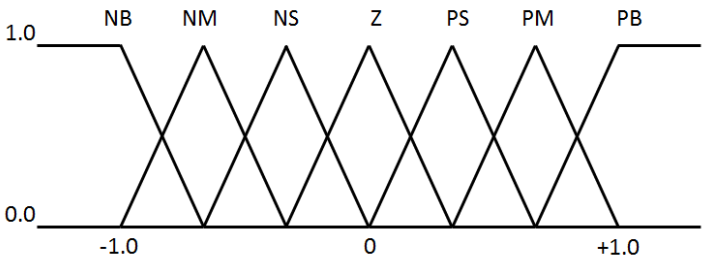


Figure 5.5 Triangular membership function for the inputs and output of the PI-like FKBC

Table 5.1 Fuzzy associative matrix for the excitation control

		X1						
		NB	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
	NB	PB	PB	PB	PM	PS	PS	Z
	NM	PB	PB	PM	PS	PS	Z	NS
	NS	PB	PM	PS	PS	Z	NS	NS
X2	**Z**	PM	PS	PS	Z	NS	NS	NM
	PS	PS	PS	Z	NS	NS	NM	NB
	PM	PS	Z	NS	NS	NM	NB	NB
	PB	Z	NS	NS	NM	NB	NB	NB

The output signal is defuzzified by the set of fuzzy rules shown in Table 5.1. Seven fuzzy sets are applied as follows: NB: Negative Big; NM: Negative Medium; NS: Negative Small; Z: Zero; PS: Positive Small; PM: Positive Medium; PB: Positive Big. Each term of the seven fuzzy sets has an associated membership function as: $[\mu_{NB}\ \mu_{NM}\ \mu_{NS}\ \mu_{Z}\ \mu_{PS}\ \mu_{PM}\ \mu_{PB}]$. Actually, each membership function is a map from the real line to the interval [0, 1] (Li et al., 2019a).

5.3.3 Fuzzification

In the fuzzification stage, the transformations of the inputs $X_1(k)$ and $X_2(k)$ into the setting of fuzzy sets are done. The inputs $X_1(k)$ and $X_2(k)$ are first normalized by scaling them and then converted to the fuzzy sets. In each step k, there are two membership values $\mu_1\big(k_1, X_1(k)\big)$ and $\mu_2\big(k_2, X_2(k)\big)$, where k_1 and k_2 are the scaling factors for the two inputs of PI-like FKBC respectively (Li et al., 2019a).

5.3.4 Control Rule Base

A set of fuzzy decision rules relating the controller inputs to the output are constructed as the Table 5.1 shows, where the 49 rules are used. The rule in term of the column 1 and the row 6 for the PI-like FKBC can be written as (Li et al., 2019a):

IF X_1 is Negative Big, AND X_2 is Positive Medium, THEN the PI output = Positive Small.

This rule can be explained as: if the input X_1 is a member of the negative big and the input X_2 is a member of the positive medium, then the output of the PI-like FKBC is a tendency for the positive small. The 49 rules yield the control actions in order to obtain the acceptable damping (Li et al., 2019a).

Here is an example about how to convert the signals by membership functions. Assuming that X_1 belongs to two membership terms, NB and NM, and the membership value of X_1 belonging to NB is a (a < 1). Then, its membership value belonging to NM is (1 - a). Assuming that X_2 belongs to two membership terms, PS and PM, and the membership value of X_2 belonging to PM is b (b < 1). Then, its membership value belonging to PS is (1 - b). Based on the fuzzy rule mentioned above, if X_1 is NB and X_2 is PM, then the output is PS. Therefore, the associated membership value of the output signal belonging to PS is a × b. Similarly, as for the output signal belonging to PS, the other two associated membership values are a × (1 - b) and (1 - a) × (1 - b). The output signal belonging to Z is assigned with a membership values (1 - a) × b. The sum of four associated membership values is exactly equal to 1. Therefore, the membership value of the output value PS is a × b + a × (1 - b) + (1 - a) × (1 - b), and the membership value of the output value Z is (1 - a) × b.

5.3.5 Defuzzification

The fuzzified result should be transferred to a supplementary signal in this stage. The centre of gravity method is applied here to transform the fuzzy value into a supplementary value. The supplementary value u_{PI} is given by (Li et al., 2019a):

$$u_{PI}(k) = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)} \tag{5.19}$$

where x_i is an output in a discrete domain, $\mu(x_i)$ is its related membership value in the membership function. The defuzzified value is the weighted average of the elements (Li et al., 2019a). For the above example, $\sum_i \mu(x_i) x_i = \big(\mathrm{ab} + \mathrm{a}(1-\mathrm{b}) + (1-\mathrm{a})(1-\mathrm{b})\big) \times \mathrm{PS} + \big((1-\mathrm{a})\mathrm{b}\big) \times \mathrm{Z}$; $\sum_i \mu(x_i) = 1$.

The main characteristics of the PI-like FKBC are summarized as follows (Li et al., 2019a):

- The proposed PI-like FKBC is designed from the classic PI controller through the bilinear discretization formula (Li et al., 2019a).
- It has self-adaptive proportional term, integral term and output signal, which are the non-linear functions of the input signals (Li et al., 2019a).
- The simple triangular membership functions are applied with some easy-understanding “if-then” rules. These fuzzy “if-then” rules are generic type, and they are independent of the system structure (Li et al., 2019a).
- The proposed PI-like FKBC is an artificially intelligent analytic scheme, which is able to work well for the real-time closed-loop control (Li et al., 2019a).
- The initialization of the PI-like FKBC is one-off throughout the entire control process (Li et al., 2019a).

5.4 Experimental Plan

Details about the engine test bench and the RCP platform to validate the PI-like FKBC have been introduced in Chapter 3. The λ signal is detected by the oxygen (lambda) sensor, received by the ES930, and sent to the ES910 as the input signal for the PI-like FKBC. In the view of developing the feedback control strategy, the sensor is identified to provide reliable measurements. The PI-like FKBC program in the ES910 will process the measured λ signal and send the updated fuel injection signal to the corresponding engine injection system in real-time (Li et al., 2019a).

The λ value is expected to be regulated within the range of the stoichiometric value ±1%, i.e. 1.00 ± 0.01 with the disturbance. The sampling time of the discrete PI-like FKBC is chosen to be 0.01 s. To show the advantages of the PI-like FKBC, its response curves are compared with those using a conventional lookup-table-based PI controller. The conventional lookup-table-based PI controller is the one in the unmodified engine management system embedded in the ECU, and the PI-like FKBC is one in the modified engine management system embedded in the ES910. The sampling time of the conventional lookup-table-based PI controller is also selected to be 0.01 s (Li et al., 2019a). This chapter is aimed to promote the self-adaptive capability of the PI-like FKBC. Then, the PI-like FKBC can reduce a lot of time and effort costed by calibrating the K_p^d and K_i^d settings compared to the conventional lookup-table-based PI controller. Three cases have been studied under the random operating conditions within the range of the normal engine operating conditions (Li et al., 2019a). In the vehicular applications, the driver describes a torque demand to the engine management system through the accelerator pedal. This signal is transferred to the AFR control system directly for the air-

mass configuration. Thus, in this chapter, the engine operating condition is adjusted by a step change of the throttle opening.

In the first case, there are four tests with different K_p^d values, i.e. K_i^d value is fixed to K_i^d = 185, while K_p^d values are K_p^d = 10, 30, 50 and 70 respectively. The second case is to test the controller with different K_i^d values, i.e. K_p^d value is fixed to K_p^d = 50, while K_i^d values are K_i^d = 65, 145, 225 and 265 respectively. In each test, both the conventional lookup-table-based PI controller and the PI-like FKBC are operated with the same K_p^d and K_i^d value. A step change of the throttle opening is randomly selected as it is adjusted from 8% to 12% after 4 s with the engine speed of 1400 rpm. These two cases are aimed to compare the adaptive performances of the PI-like FKBC and the conventional lookup-table-based PI controller with the same step change of the operating condition. Once the λ value is maintained within the range of 1.00 ± 0.01 after a certain duration of oscillation, the system is regarded to reach the steady state. In addition, the output scaling factor of the PI-like FKBC is set as K_u = 0.2 (Li et al., 2019a).

The third case is to test the performances of both controllers at the non-calibrated operating conditions. The K_p^d and K_i^d values of both two controllers are set as the Table 5.2 and Table 5.3 show as the initially calibrated conditions based on the estimates from the previous two cases. Two throttle positions are selected as the initially calibrated conditions with the engine speed of 1400 rpm. Therefore, the other throttle positions are the non-calibrated operating conditions. The PI-like FKBC should work effectively at all the non-calibrated operating conditions based on only two groups of scaling factors at initially calibrated conditions which may even be not accurate. The output scaling factor of the PI-like FKBC is still fixed to K_u = 0.2 (Li et al., 2019a).

Table 5.2 K_p^d values at the calibrated conditions

	Throttle opening (%)	
	8	10
K_p^d	30	60

Table 5.3 K_i^d values at the calibrated conditions

	Throttle opening (%)	
	8	10
K_i^d	140	200

It should be noted that, experimental studies have shown that the derivative term cannot improve the response of the AFR obviously. Therefore, this chapter omits the derivative term, and saves the time spent in calibrating the derivative gains as well (Li et al., 2019a).

5.5 Results and Discussion

5.5.1 Performance with Different K_p Values

The response of the λ value from the first case is displayed in Figure 5.6 below:

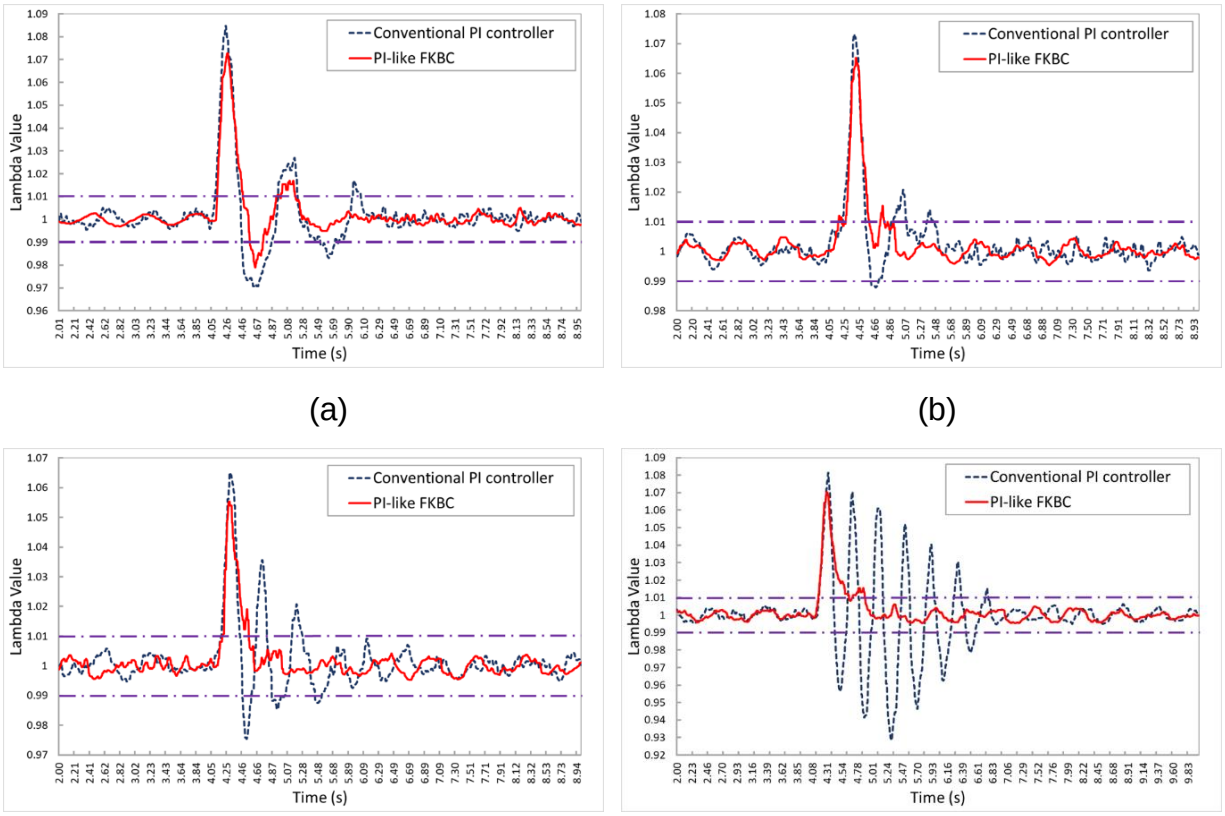


Figure 5.6 **λ** value response with different K_p^d gains: (a) K_p^d=10; K_i^d=185; (b) K_p^d=30; K_i^d=185; (c) K_p^d=50; K_i^d=185; (d) K_p^d=70; K_i^d=185

As shown in Figure 5.6, when $K_p^d = 10$, the conventional lookup-table-based PI controller becomes less sensitive, and costs more time to allow the λ value to reach the steady state, because this K_p^d gain is smaller for the controller. When $K_p^d = 70$, the λ value of the conventional lookup-table-based PI controller has started to oscillate obviously, and it costs more time to reach the steady state, since this K_p^d gain is too large for the controller and results in the unstable system. In contrast, the PI-like FKBC can work effectively with

any K_p^d values. The detailed performance evaluations for the system response are shown in Table 5.4 (Li et al., 2019a).

Table 5.4 Performance evaluations for the controllers with different K_p^d values

K_p^d	Controller	Settling time (s)	Maximum overshoot	Number of the positive damping	IAE
10	Conventional PI	2.14	0.085	3	0.075
	PI-like FKBC	1.25	0.070	2	0.049
	Reduction	**41.59%**	**17.65%**	**33.33%**	**34.67%**
30	Conventional PI	1.49	0.073	3	0.052
	PI-like FKBC	0.93	0.065	2	0.041
	Reduction	**37.58%**	**10.96%**	**33.33%**	**21.15%**
50	Conventional PI	1.65	0.065	3	0.057
	PI-like FKBC	0.63	0.054	1	0.036
	Reduction	**61.82%**	**16.92%**	**66.67%**	**36.84%**
70	Conventional PI	2.82	0.082	7	0.139
	PI-like FKBC	0.93	0.070	2	0.050
	Reduction	**67.02%**	**14.63%**	**71.43%**	**64.03%**

5.5.2 Performance with Different K_i Values

The response of the λ value from the second case is displayed in Figure 5.7 below:

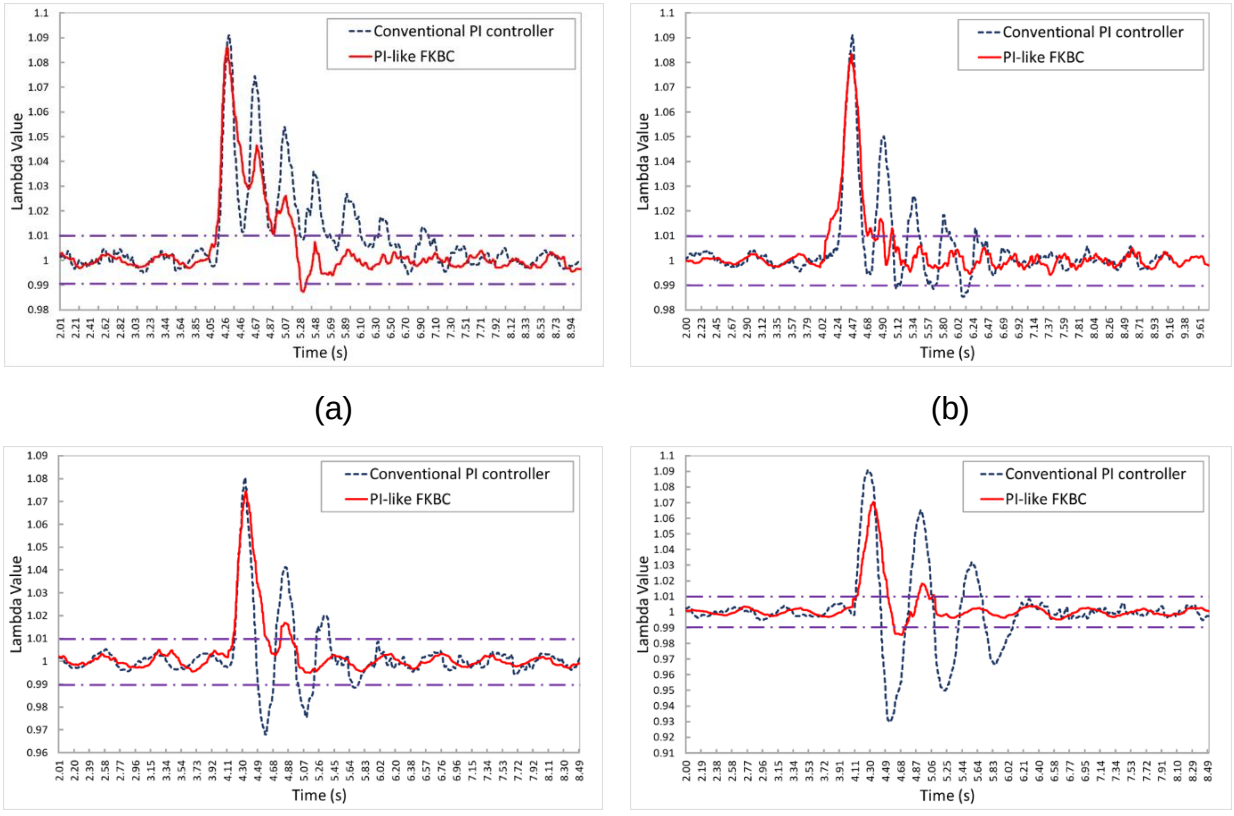


Figure 5.7 **λ** value response with different K_i^d gains: (a) K_i^d=65; K_p^d=50; (b) K_i^d=145; K_p^d=50; (c) K_i^d=225; K_p^d=50; (d) K_i^d=305; K_p^d=50

As shown in Figure 5.7, when $K_i^d = 65$, the conventional lookup-table-based PI controller eliminates the steady-state error very slowly, because this K_i^d gain is smaller for the controller. When K_i^d= 305, the λ value response of the conventional lookup-table-based PI controller starts to oscillate, because this K_i^d gain is bigger for the controller. However, the PI-like FKBC with these K_i^d gains can still work effectively. The detailed performance evaluations for the system response are shown in Table 5.5 (Li et al., 2019a):

Table 5.5 Performance evaluations for the controllers with different K_i^d values

K_i^d	Controller	Settling time (s)	Maximum overshoot	Number of the positive damping	IAE
65	Conventional PI	3.13	0.090	7	0.128
	PI-like FKBC	1.40	0.085	3	0.091
	Reduction	**55.27%**	**5.56%**	**57.14%**	**28.91%**
145	Conventional PI	2.51	0.090	5	0.081
	PI-like FKBC	1.08	0.083	2	0.062
	Reduction	**56.97%**	**7.78%**	**60.00%**	**23.46%**
225	Conventional PI	1.78	0.081	3	0.068
	PI-like FKBC	0.99	0.074	2	0.049
	Reduction	**44.38%**	**8.64%**	**33.33%**	**27.94%**
305	Conventional PI	2.11	0.090	3	0.127
	PI-like FKBC	1.11	0.070	2	0.049
	Reduction	**47.39%**	**22.22%**	**33.33%**	**61.42%**

According to the results above, the performance of the PI-like FKBC is not defined by the initially calibrated constant K_p^d and K_i^d gains, because its output signal is self-tuned in real-time. It can work effectively no matter how good or poor the K_p^d and K_i^d gains are. Furthermore, it can exhibit less settling time and less oscillation than the conventional lookup-table-based PI controller does. In contrast, the performance of the conventional lookup-table-based PI controller totally relies on the constant calibrated K_p^d and K_i^d gains (Li et al., 2019a).

5.5.3 Performance at Non-Calibrated Points

The operating conditions of the third case are changed based on what Figure 5.8 shows. The responses of the λ value, intake manifold pressure, air mass flow into the cylinder, and fuel injection into the cylinder of the third case are displayed in Figure 5.9, 5.10, 5.11 and 5.12 respectively (Li et al., 2019a):

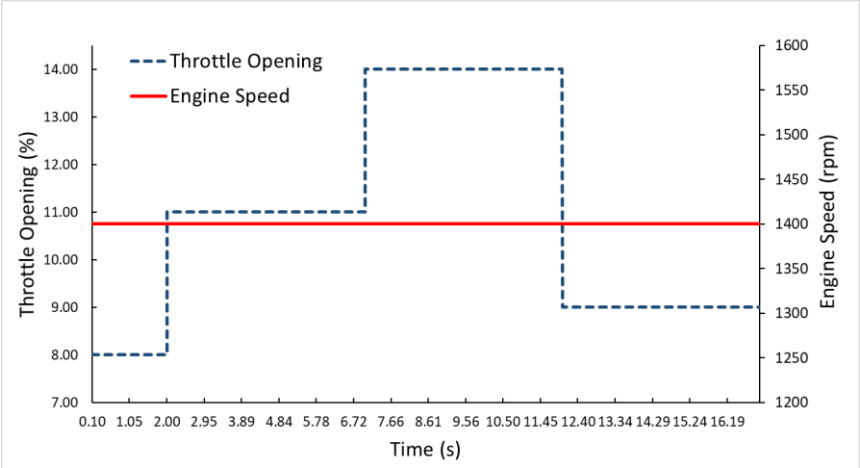


Figure 5.8 Variable engine operating conditions

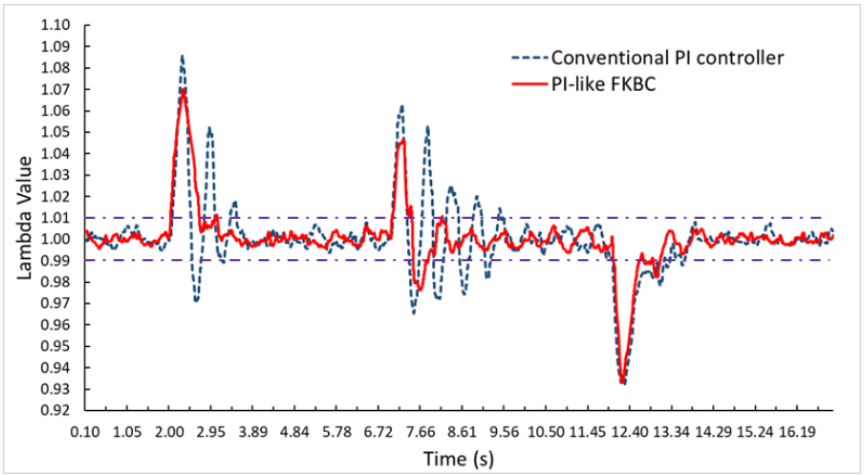


Figure 5.9 λ value response

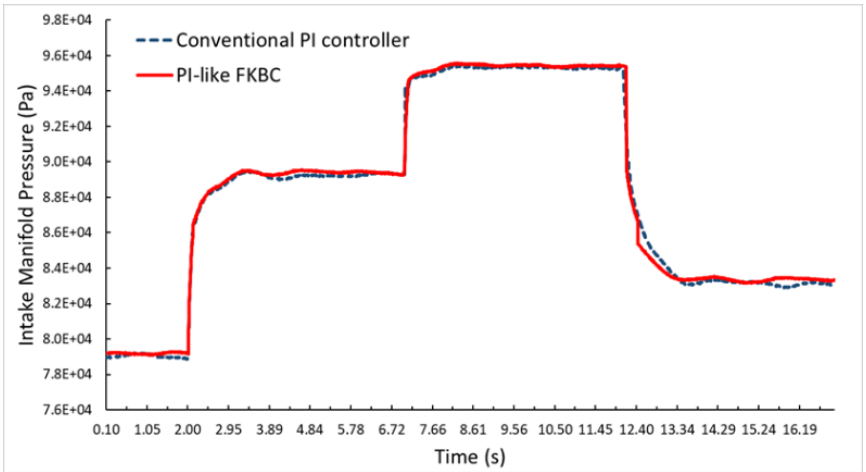


Figure 5.10 Intake manifold pressure (Pa) response

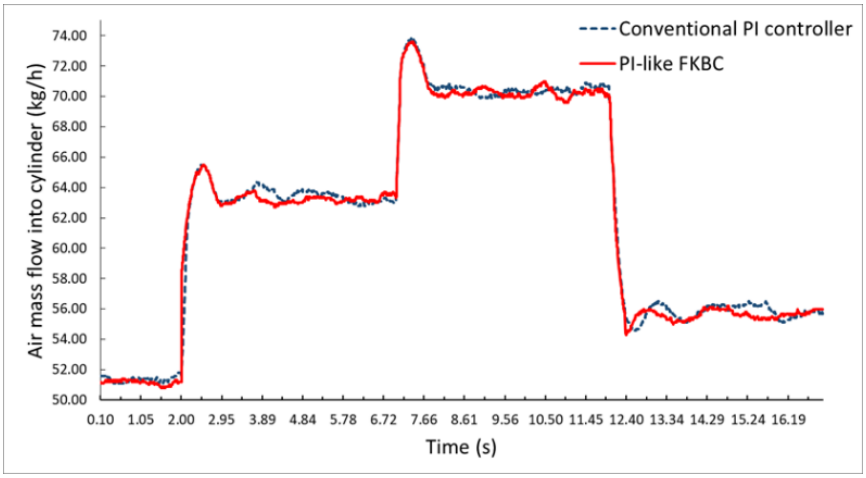


Figure 5.11 Air mass flow into the cylinder (kg/h) response

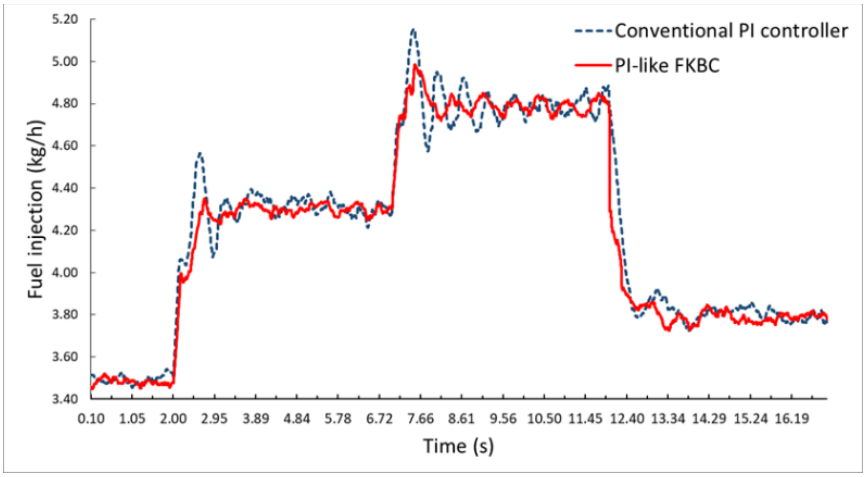


Figure 5.12 Fuel injection (kg/h) response

Since the K_p^d and K_i^d gains of the conventional lookup-table-based PI controller at non-calibrated conditions are calculated using the linear interpolation method based on the calibrated values in Table 5.2 and 5.3, these K_p^d and K_i^d gains lead to the poor damping performance, i.e. larger overshoot and more settling time as Figure 5.9 shows. In contrast, the PI-like FKBC can still reduce much settling time and oscillation, because the controller is self-tuned based on the gap between the target value and the actual value instead of the constant proportional and integral gains. As Figure 5.12 shows, the injected fuel mass increases very fast after the operating condition is changed in order to accelerate the response speed. However, it rises slowly when the **λ** value is getting closer to the target value in case of the significant deviation and oscillation. The detailed performance evaluations for the system response are shown in Table 5.6 (Li et al., 2019a).

Table 5.6 Performance evaluations for the controllers at non-calibrated conditions

Throttle opening step (%)	Controller	Settling time (s)	Maximum overshoot	Number of the positive damping	IAE
8-11	Conventional PI	1.71	0.086	3	0.059
	PI-like FKBC	1.17	0.070	2	0.045
	Reduction	**14.56%**	**18.60%**	**33.33%**	**25.00%**
11-14	Conventional PI	2.72	0.063	5	0.075
	PI-like FKBC	1.19	0.046	2	0.031
	Reduction	**56.25%**	**26.98%**	**60.00%**	**58.67%**
14-9	Conventional PI	2.01	0.068	3	0.061
	PI-like FKBC	1.20	0.067	2	0.041
	Reduction	**40.30%**	**1.47%**	**33.33%**	**32.79%**

The Table 5.6 shows that, the PI-like FKBC is able to reduce the settling time up to 56.25% and reduce the IAE up to 58.67% compared with the conventional lookup-table-based PI controller when damping out the oscillations at the non-calibrated conditions. Thus, the PI-like FKBC can save much time and effort for engineers, since they do not need to calibrate the K_p^d and K_i^d values at each calibrated condition accurately (Li et al., 2019a).

Take a 100-point map (10 engine speeds x 10 throttle positions) for example as Figure 5.13 shows:

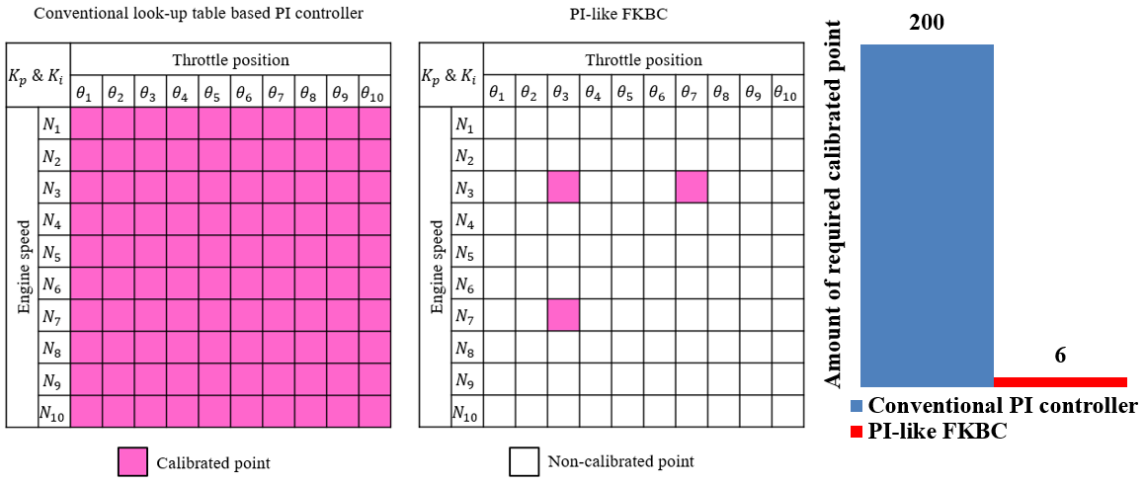


Figure 5.13 Comparison of controllers to calibrate a 100-point map

The conventional lookup-table-based PI controller needs to calibrate 100 K_p gains and 100 K_i gains accurately. Whereas, the PI-like FKBC needs to calibrate only three K_p gains and three K_i gains randomly and approximately to generate an initial trend. The output signal of the PI-like FKBC will be adjusted automatically later based on the gap between the target λ value and the actual λ value, instead of the initially calibrated constant K_p^d and K_i^d gains. More initially calibrated conditions may lead to more reliable performance of the PI-like FKBC. However, since the purpose of developing the proposed

controller is to reduce the time and effort for engineers, the engineers can determine the trade-off between more initially calibrated conditions and less time-consumption by themselves. At least, the PI-like FKBC is more intelligent than the conventional lookup-table-based PI controller (Li et al., 2019a).

5.6 Summary

A discrete PI-like FKBC has been developed for GDI engines to regulate the AFR. The output signal of the PI-like FKBC was self-tuned in real-time according to the fuzzy "if-then" rules and the non-linear defuzzification algorithm. The experimental results have verified the high robustness and self-adaptive capability of the proposed PI-like FKBC. The conclusions drawn from the investigation are (Li et al., 2019a):

- The proposed PI-like FKBC can lead to better transient response behaviour to damp out the oscillations with less settling time and less integral of absolute error (IAE) for various operating conditions, compared with the conventional lookup-table-based PI controller (Li et al., 2019a).
- The proportional term, the integral term, and the output signal of the PI-like FKBC for the AFR control are non-linear functions of the difference between the target and real AFRs. They are self-tuned in real-time and do not rely on the constant calibrated values, so that the PI-like FKBC can work efficiently with even poor proportional and integral gains, and save the effort to be spent on the calibration (Li et al., 2019a).
- With the same proportional and integral gains at the calibrated conditions, the proposed PI-like FKBC can reduce the settling time by up to 67.02% and reduce

the IAE by up to 64.03%, compared with the conventional lookup-table-based PI controller for the transient engine operation (Li et al., 2019a).

- At the same non-calibrated conditions, the proposed PI-like FKBC can reduce the settling time by up to 56.25% and reduce the IAE by up to 58.67%, compared with the conventional lookup-table-based PI controller for the transient engine operation (Li et al., 2019a).
- The proposed PI-like FKBC has a clear and simple structure, which can facilitate its implementation in the field of engineering (Li et al., 2019a).

Overall, the proposed PI-like FKBC can reduce the effort and time to be spent in calibrating controller parameters, and it can regulate the AFR more efficiently for transient operations (Li et al., 2019a).

CHAPTER 6 TRANSIENT CALIBRATION FOR THE PI-LIKE FUZZY KNOWLEDGE BASED CONTROLLER OF THE AFR CONTROL SYSTEM USING A CAPSO ALGORITHM

The work presented in this chapter has been published in the journal “Proceedings of the IMechE, Part D: Journal of Automobile Engineering” (Li et al., 2019b). This chapter proposes an enhanced PI-like FKBC using the chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm to automatically define the optimum controller parameter settings, in order to further improve its transient response behaviour. An alternative time-domain objective function is applied for the transient calibration program, without the need for prior selection of the search domain. The real-time transient performance of the enhanced PI-like FKBC has also been investigated on the AFR control system of a GDI engine.

A general introduction to the background of this study is presented first. Then, the design of the transient optimiser, including its structure and the selection of the objective function, is introduced. The Simulink-based enhanced PI-like FKBC is implemented on a V6 GDI engine test bench through the ETAS RCP platform. The experimental results are shown in the following section, and comparisons among the enhanced PI-like FKBC and the other PI controllers are given. The final section presents a summary.

6.1 Introduction

As introduced in the previous chapter, the PI-like FKBC is developed to replace the conventional lookup-table-based PI controller for the AFR control of GDI engines, in order

to reduce the effort needed for calibration, and cope with the variety of engine operating regimes, parameter uncertainties, and non-linear dynamics. The self-adaptive capability and high robustness of the PI-like FKBC have been validated by experimental studies. However, although the PI-like FKBC can automatically correct the initially calibrated proportional and integral parameters, a more appropriate selection of the initial parameters will lead to a better transient performance. Accordingly, an enhanced PI-like FKBC should be developed based on the automatic transient calibration process, which can still save on the effort needed for calibration and offer the best parameter settings as well. Thanks to the development of the non-model-based calibration approach in Chapter 4, an enhanced self-calibrating PI-like FKBC is developed in this chapter (Li et al., 2019b).

The CAPSO algorithm is employed for transient calibration in this chapter, as it performs generally better than other algorithms. In terms of the swarm intelligence algorithm, the following studies show that its transient performance highly depends on the selection of the search domain and objective function. For most of the algorithm-based transient calibration problems, prior knowledge about the search domain is essential. Otherwise, the optimiser may converge to improper solutions; which may lead to an extremely large overshoot of the control signal, or may cost an unpredictably long time to stabilize. To minimize the impact of various search domains, an alternative objective function is presented in this study. The simulation results show that the optimiser with the proposed objective function can converge to similar calibration results, regardless of the range of the search domain. Eventually, this alternative objective function is used to design the enhanced PI-like FKBC for the AFR control system. The controller is built by MATLAB/Simulink and co-operates with ETAS RCP facilities for real-time implementation.

The transient performance of the enhanced PI-like FKBC is investigated on a production GDI engine via various operating conditions (Li et al., 2019b).

6.2 Design of the Transient Optimiser

6.2.1 Initialisation of the Optimiser

Since the enhanced PI-like FKBC in this chapter is revised based on the PI-like FKBC designed in the previous chapter, details of the controller can be found in the previous chapter and are skipped here (Li et al., 2019b).

As introduced, the scaling terms of the PI-like FKBC are non-linear functions of the gap between stoichiometric and actual AFRs. The scaling gains are not constant values but variable values automatically adjusted in real-time based on the gap between stoichiometric and actual AFRs. However, the transient performance of the PI-like FKBC is closely related to the scaling factors K_p, K_i, K_u; which should be better parameterized with the optimum values at the beginning (Li et al., 2019b).

As for the CAPSO-based calibration in this chapter, each particle's position is defined as (Li et al., 2019b):

$$x^{(i,j)} = [{K_p}^{(i,j)}, {K_i}^{(i,j)}, {K_u}^{(i,j)}] \tag{6.1}$$

where i = [1,2,3…N] is the index of iterations and N is the number of iterations; j = [1,2,3…M] is the index of particles in each swarm and M is the number of particles in each swarm; ${K_p}^{(i,j)}, {K_i}^{(i,j)}, {K_u}^{(i,j)}$ are the scaling factors K_p, K_i, K_u of the PI-like FKBC for i th iteration and j th particle (Li et al., 2019b).

6.2.2 Choice of Objective Functions

To design the CAPSO-based optimiser for transient calibration, the proper search domain and objective function have to be identified first. As the following examples show, different search domains with different objective functions will lead to totally different transient response behaviours. In terms of the objective function, the integral of absolute error (IAE) is one of the most widely used objective functions for the time-domain optimisation, since a good PI-like FKBC should minimize the oscillation of the system's transient response with less settling time and less overshoot (Li et al., 2019b).

To present how the selection of the search domain and objective function will affect the calibration results, this thesis supposes that a PI-like FKBC is to be designed for a time-varying discrete system, as Equation (6.2) shows, and the CAPSO algorithm is supposed to find the optimum scaling factors K_p, K_i and K_u of this PI-like FKBC. The step change of the system is defined as 0 to 1 at the beginning of the running (Li et al., 2019b).

$$\begin{cases} \dot{x_1}(t_n) = x_2(t_n) \\ \dot{x_2}(t_n) = -e^{-0.2t_n}x_2(t_n) - e^{-5t_n}\sin(2t_n + 6)\,x_1(t_n) + u(t_n) \\ t_n = 0, T, 2T, 3T \ldots nT \ (T = 0.01s) \end{cases} \tag{6.2}$$

where u is the control signal and x_1 is the response signal (Li et al., 2019b).

Firstly, the IAE is employed as the only objective. The CAPSO algorithm is implemented several times with different search domains: [0, 10], [0, 100] and [0, 1000]. The optimal results for K_p, K_i and K_u as well as the transient performance of the system are listed as follows (Li et al., 2019b).

Table 6.1 Optimal scaling factors based on different ranges of search domains with IAE as the objective

Objective	Search domain	K_p	K_i	K_u	Settling time (s)	Overshoot of the control signal
IAE	[0, 10]	2.95	1.00	8.98	2.31	8.97
IAE	[0, 100]	5.05	1.01	33.44	1.19	33.01
IAE	[0, 1000]	165.16	167.49	892.54	1.08	537.25

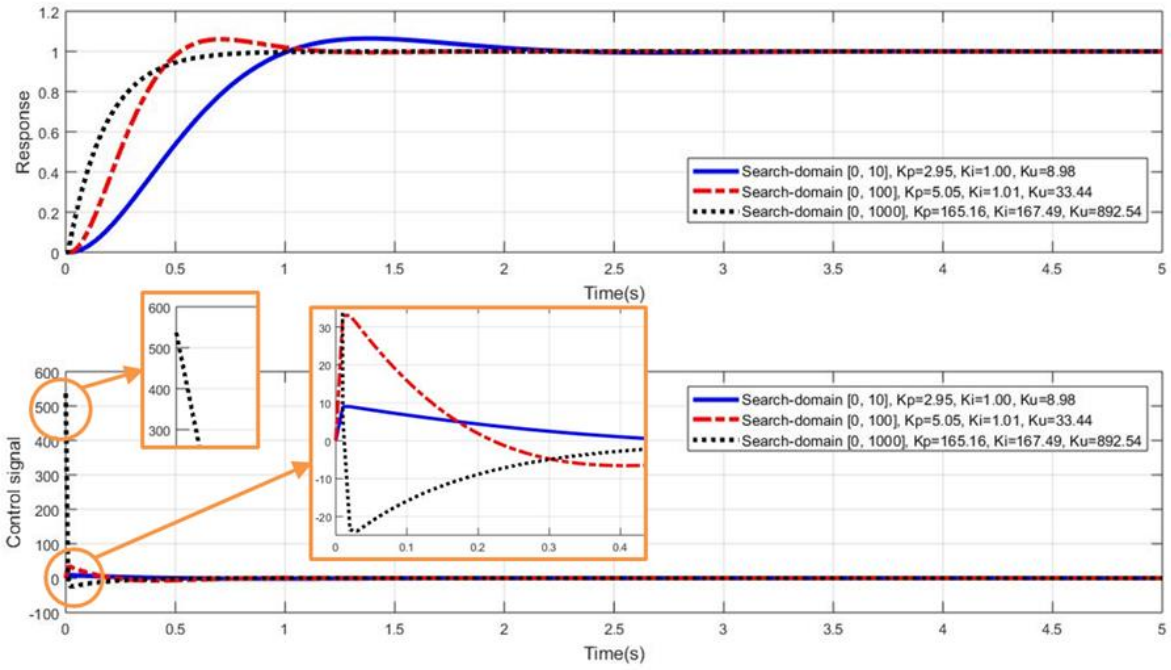


Figure 6.1 Step response of the system based on different ranges of search domains with IAE as the objective

As can be seen from the results, the range of the search domain will lead to different calibration results and affect the transient performance. Although all of these optimum scaling factors can reduce the oscillation of the system, the search domain [0, 10] costs more settling time, whereas search domain [0, 1000] leads to an excessively large overshoot of the control signal. Thus, unless there is sufficient prior information for the

selection of the search domain, the conventional IAE objective function may lead to an unfavourable performance of the control signal and actuator saturation (Li et al., 2019b).

To reduce the impact of the range of the search domain, the next case combines the magnitude of the control signal with the IAE as objectives in order to achieve relatively similar calibration results. Hence, the calibration task is formulated into a multi-objective optimisation problem (Li et al., 2019b):

$$\begin{cases} \min\limits_{Kp,Ki,Ku} J = \text{minimize}\,(IAE + IAU) = \text{minimize}\,(\sum\limits_{k=1}^{n} R \cdot T \cdot |e(k)| + \sum\limits_{k=1}^{n} Q \cdot T \cdot |u(k)|) \\ 0 \leq \mathrm{R} < 1, 0 < \mathrm{Q} \leq 1, \mathrm{R} + \mathrm{Q} = 1 \end{cases} \tag{6.3}$$

where the magnitude of the control signal is indicated by the integral of the absolute control signal (IAU); $e(k)$ is the error between the target value and the actually measured value; $u(k)$ is the value of the real-time control signal. Both of them are discrete time signals; and R and Q are the weighting factors on the error and the control signal respectively (Li et al., 2019b).

The step response of the system with different ranges of search domains using IAE and IAU as objectives is shown in Figure 6.2. R and Q are both assigned as 0.5 in this case. The calibration results are listed in Table 6.2. As the figure shows, relatively similar step responses can be obtained regardless of the ranges of search domains. Even though the settling time becomes a little bit longer, the overshoot of the control signal is reduced effectively. Search domains [0, 10] and [0, 100] achieve similar performances while [0, 10] leads to better trade-off between the settling time and the overshoot of the control signal (Li et al., 2019b).

Table 6.2 Optimal scaling factors based on different ranges of search domains with IAE and IAU as objectives

Objective	Search domain	K_p	K_i	K_u	Settling time (s)	Overshoot of the control signal
IAE+IAU	[0, 10]	2.82	2.47	7.39	4.15	3.00
IAE+IAU	[0, 100]	21.52	20.28	80.15	4.08	3.95
IAE+IAU	[0, 1000]	665.35	682.26	627.78	5.45	95.27

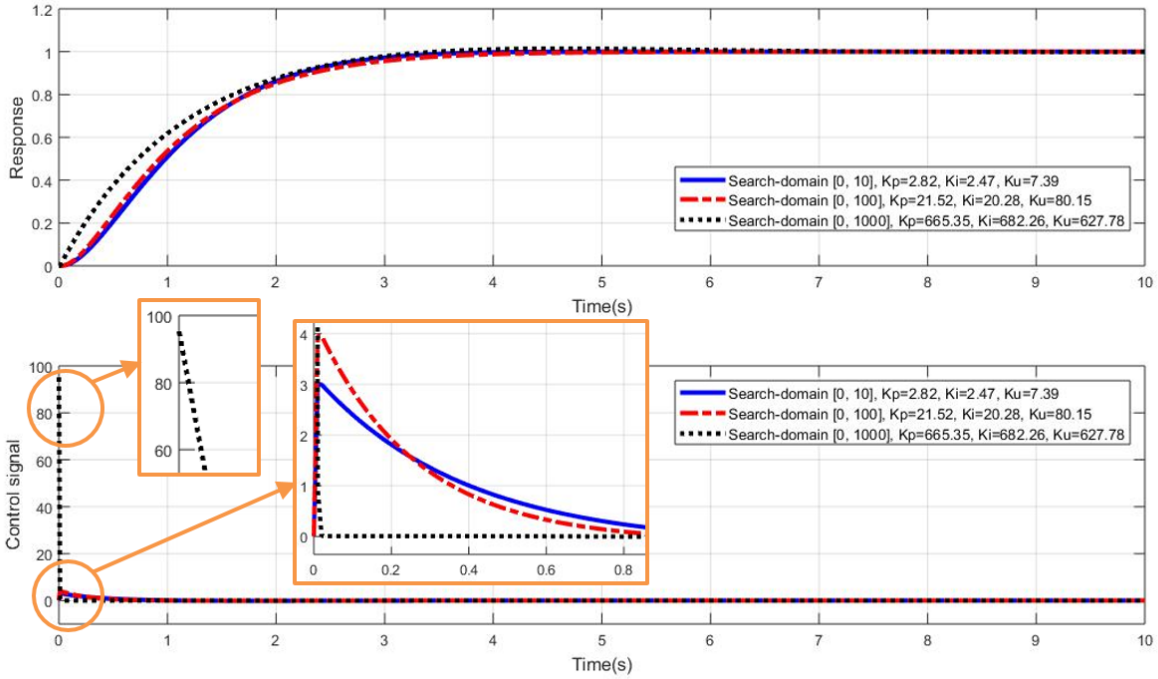


Figure 6.2 Step response of the system based on different ranges of search domains with IAE and IAU as objectives

To present how the different weighting factors R and Q affect the step response, several cases with search domain [0, 10] are tested and shown in Figure 6.3. Meanwhile, the optimum scaling factors are listed in Table 6.3 (Li et al., 2019b).

Table 6.3 Optimal scaling factors based on different R's and Q's with IAE and IAU as objectives

R, Q	K_p	K_i	K_u	Settling time (s)	Overshoot of the control signal
1, 0	2.95	1.00	8.98	2.31	8.97
0.9, 0.1	2.15	1.00	5.38	3.25	5.38
0.8, 0.2	3.74	1.82	6.74	3.67	3.70
0.5, 0.5	2.82	2.47	7.39	4.15	3.00
0.2, 0.8	2.09	3.15	7.74	8.02	2.46
0.1, 0.9	1.09	2.45	7.29	13.03	2.98

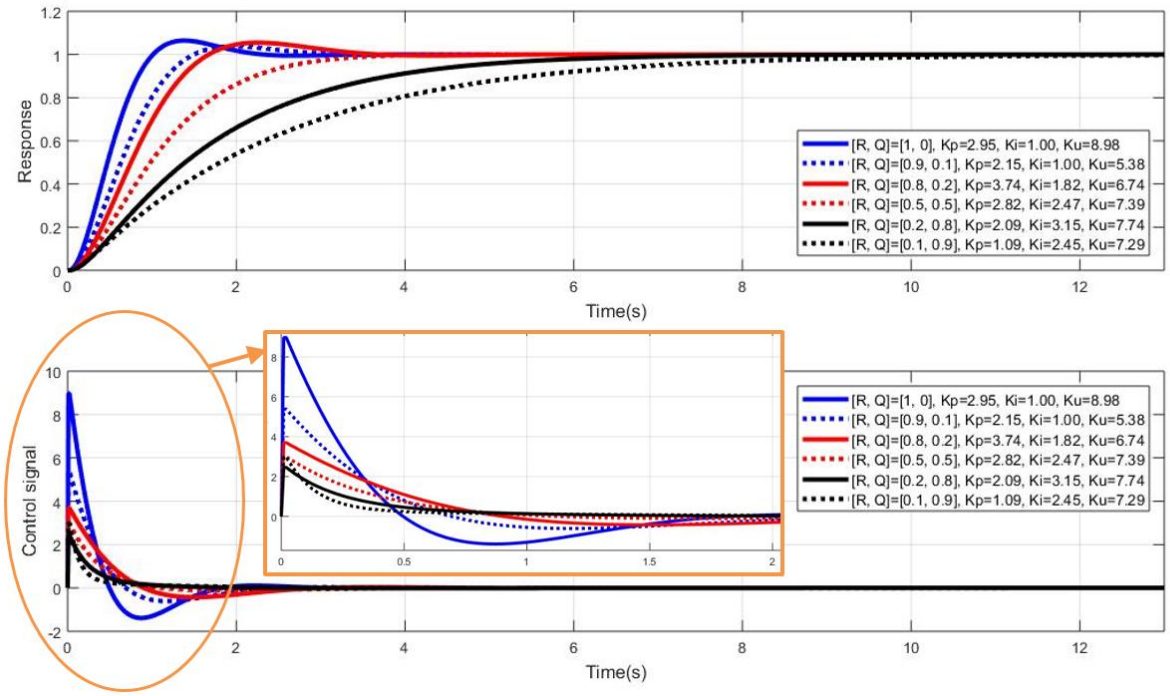


Figure 6.3 Step response of the system based on different R's and Q's with IAE and IAU as objectives

R and Q can be adjusted by users to achieve a desirable trade-off between reducing the settling time and avoiding the extremely large control signal. In this chapter, the values of R and Q are both assigned as 0.5 (Li et al., 2019b).

Considering that there may be a huge gap between the magnitudes of IAE and IAU in practical engineering problems, W is applied as the correlation factor to ensure that the IAE and IAU values fall in the same scale of range, in case that the optimisation process may be more sensitive to the objective with larger magnitude (Li et al., 2019b).

Thus, the final objective function for the optimisation process is shown as (Li et al., 2019b):

$$\min_{Kp,Ki,Ku} J = \text{minimize}\left(\sum_{k=1}^{n} W \cdot 0.5 \cdot T \cdot |e(k)| + \sum_{k=1}^{n} 0.5 \cdot T \cdot |u(k)|\right) \tag{6.4}$$

It is worth mentioning that the IAU is widely employed in practical engineering problems against the extremely large control signal and violent system oscillation (Li et al., 2019b).

6.3 Experimental Plan

6.3.1 Air/Fuel Ratio Control Plant

As the IAE and IAU are employed as the objectives for the CAPSO program, the AFR closed-loop feedback controller is designed as the following figure shows (Li et al., 2019b):

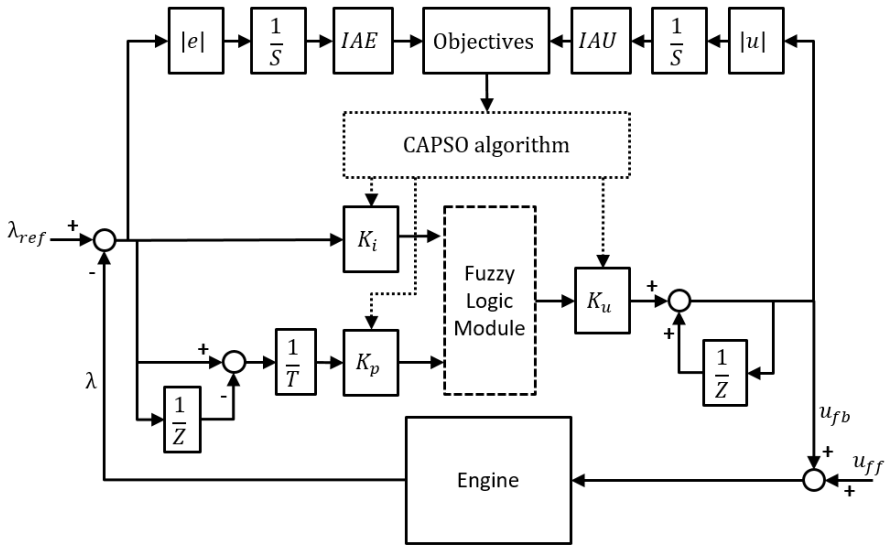


Figure 6.4 Structure of the PI-like FKBC with IAE and IAU as the objective functions

6.3.2 Experimental Procedure

Because the proposed CAPSO algorithm is evolved from the APSO algorithm, both CAPSO and APSO algorithms are implemented for calibration in this chapter and a comparison of these two algorithms is considered. Meanwhile, the algorithm-based PI-like FKBC is compared with the conventional self-adaptive PI-like FKBC, which uses the random scaling factors as the previous chapter introduces, to show the improvement (Li et al., 2019b).

Based on the experimental studies, the order of magnitude of the IAE is 0.01; while it is 1.00 for the IAU. Thus, $W = 100$ is applied in Equation (6.4) to normalize the IAE and IAU with the same resolution (Li et al., 2019b).

Throttle positions of 8% and 12% with an engine rotation speed of 1400 rpm are selected for calibration within the threshold of the engine operating conditions. The engine runs stably with these throttle positions which can guarantee the long-term measurement accuracy and increase the reliability of the repeatability tests. Besides, this chapter is to investigate the transient performance of the controller against the step change of operating conditions rather than the investigation for the performance at steady-state conditions. In the first case, at the point of two seconds from the start of the experiment there is a step change of the throttle position from 10% to 12% with an engine rotation speed of 1400 rpm. In the second case, the throttle position is adjusted from 6% to 8% after 2 s again as the step change with the engine rotation speed of 1400 rpm. The performance of the AFR control system with the calibration results (i.e. optimum K_p, K_i and K_u) is shown in Figures 6.7, 6.10 and Tables 6.4, 6.5 in the following section (Li et al., 2019b).

The last case for throttle positions 11% and 14% with the engine rotation speed of 1400 rpm is to validate the performance of the PI-like FKBC at non-calibration operating conditions. Since the scaling factors of the PI-like FKBC have been calibrated at throttle positions of 8% and 12%, the other throttle positions are regarded as non-calibration operating conditions. The PI-like FKBC should still regulate the signal effectively at all non-calibration conditions, based on the optimum scaling factors at calibration conditions. The details are shown in Figure 6.11 and Table 6.6 (Li et al., 2019b).

For both of the CAPSO and APSO algorithms, the search domain is selected as [0, 10]; 10 particles in each swarm and 20 iterative loops are pre-set for calibration (Li et al., 2019b).

As introduced in Chapter 5, the λ value should be regulated back to 1.00 within 2 seconds. Hence, 5 seconds are enough for the signal to become stabilized. Then, the engine is defined to run for 5 seconds with each given group of scaling factors in each iteration for calibration. The calibration process is switched on when the step change happens. After 5 s, the calibration process for the first given group of scaling factors will stop and run with the next group of scaling factors. Thus, the scaling factors of the PI-like FKBC are continuously and automatically assigned by every 5 s. Because there are 10 particles in each swarm, 10 groups of scaling factors (K_p, K_i, K_u) are generated and sent to the PI-like FKBC in each iteration in real time. Therefore, the calibration process will repeat 10 times and cost 10 × 5 = 50 seconds for one generation of iterations. After that, the scaling factors for the next generation of iterations will be calculated based on the algorithm and fed into the engine. The calibration process will repeat 10 times (50 seconds) again. Given that 20 iterations are selected initially and the calibration process will stop when the

optimum results converge, it costs no more than 20 × 50 = 1000 seconds (16.67 minutes) for the whole calibration process (Li et al., 2019b).

In each case, transient performances of three PI-like FKBCs are tested and compared. The CAPSO algorithm is firstly used to calibrate the scaling factors of the first PI-like FKBC following the above procedures. The optimum results (determined scaling factors) are fed into the first PI-like FKBC and the corresponding real-time engine response is obtained. Then, the APSO algorithm is used to calibrate the scaling factors of the second PI-like FKBC. The third PI-like FKBC is the conventional self-adaptive PI-like FKBC in Chapter 5, where scaling factors are randomly selected rather than initially calibrated (Li et al., 2019b).

The λ value is supposed to be regulated within the range 1.00±0.01 with the disturbance. Once the λ value is maintained within the range 1.00±0.01 after a certain duration of oscillation, it would be regarded as reaching a steady state (Li et al., 2019b).

6.4 Results and Discussion

6.4.1 Performance at Calibrated Points

The APSO-algorithm and CAPSO-algorithm-based optimisation processes of the 12% throttle position are shown as follows (Li et al., 2019b):

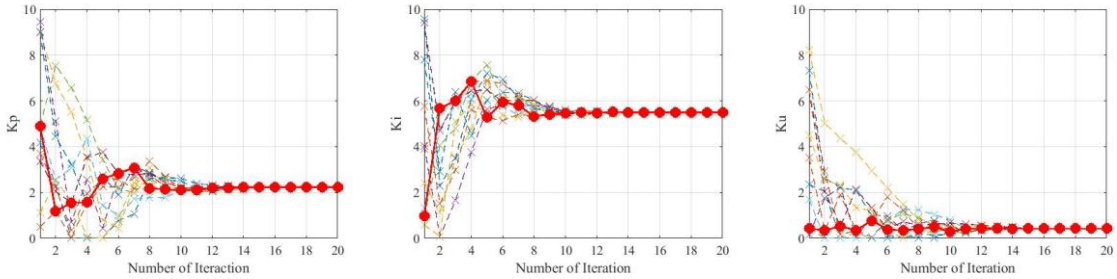


Figure 6.5 Calculated Kp, Ki, Ku factors after 20 iterations for 12% throttle position based on APSO

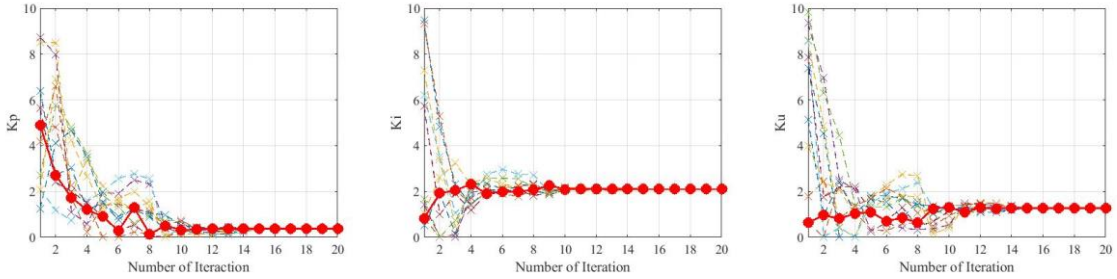
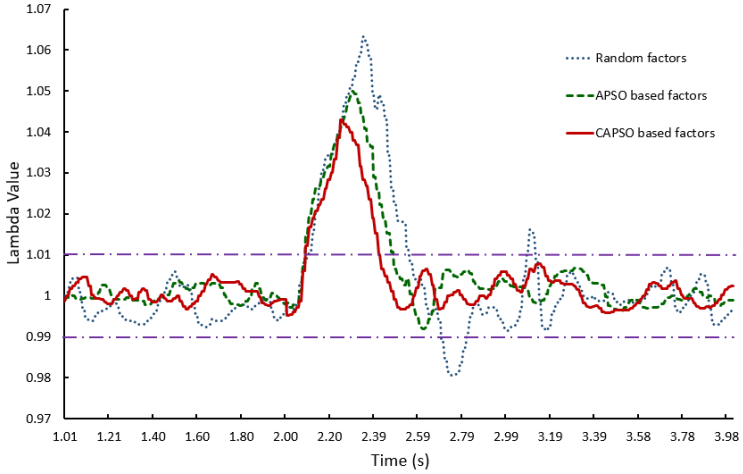
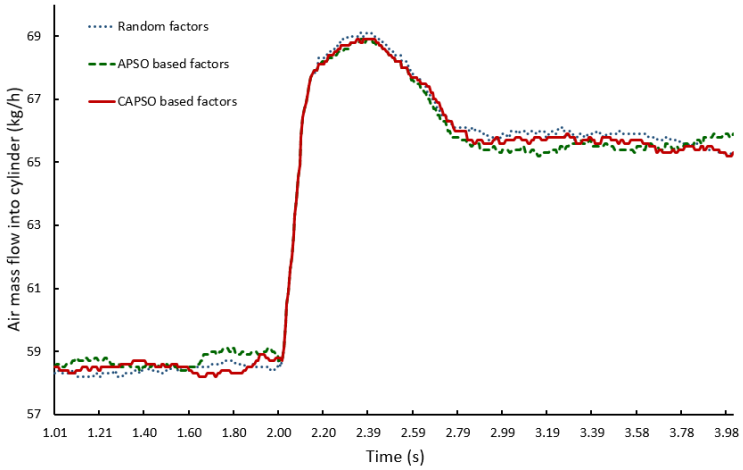


Figure 6.6 Calculated Kp, Ki, Ku factors after 20 iterations for 12% throttle position based on CAPSO

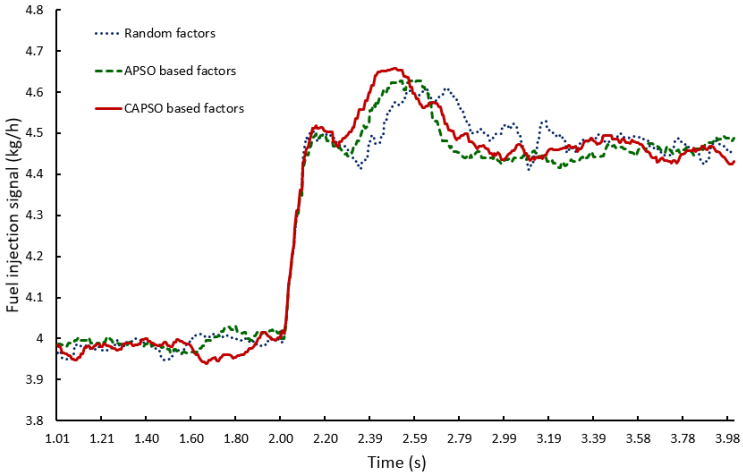
The red points in each figure indicate the locally optimum scaling factors K_p, K_i and K_u which result in the minimum objective function after each iteration. The points in other colours indicate the other scaling factors in the swarm which lead to worse objective function. As can be seen from the figures, both the APSO algorithm and CAPSO algorithm cost 12 iterations, around 10 minutes, to converge to the stable optimum results, i.e. the globally optimum results in the swarm. The time consumption is less than the initially pre-set time. The calibration results and the step responses based on the calculated scaling factors are shown as follows (Li et al., 2019b):



(a)



(b)



(c)

Figure 6.7 Step response based on the calculated scaling factors for 12% throttle position: (a) **λ** value response; (b) air mass flow into the cylinder (kg/h); (c) fuel injection (kg/h)

Table 6.4 Calibration result and performance evaluation for 12% throttle position

Scaling factors	K_p	K_i	K_u	Settling time (s)	Maximum overshoot of λ	IAE	IAU	0.5*IAE +0.5*IAU
Imputed	5.00	10.00	0.20	1.10	0.063	3.143	7.045	5.094
APSO based	2.24	5.50	0.42	0.54	0.050	1.886	7.021	4.453
Reduction				**50.91%**	**20.63%**	**39.99%**		**12.57%**
CAPSO based	0.37	2.10	1.28	0.47	0.043	1.411	7.050	4.231
Reduction				**57.27%**	**31.75%**	**55.11%**		**19.39%**

Obviously, both CAPSO and APSO-based PI-like FKBCs can damp out the oscillation more effectively than the conventional PI-like FKBC; while CAPSO-based PI-like FKBC performs better than APSO-based PI-like FKBC. Compared with conventional PI-like FKBC, CAPSO-based PI-like FKBC is able to reduce 19.39% of the total objective function IAE+IAU; while APSO-based PI-like FKBC could reduce 12.57% (Li et al., 2019b).

The APSO-algorithm and CAPSO-algorithm-based optimisation processes of the 8% throttle position are shown as follows (Li et al., 2019b):

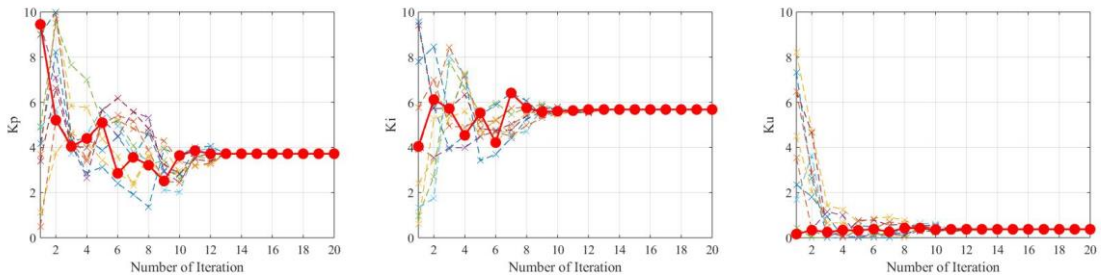


Figure 6.8 Calculated Kp, Ki, Ku factors after 20 iterations for 8% throttle position based on APSO

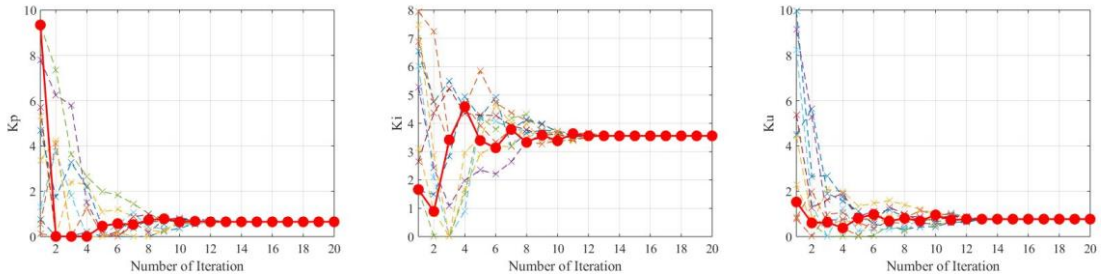
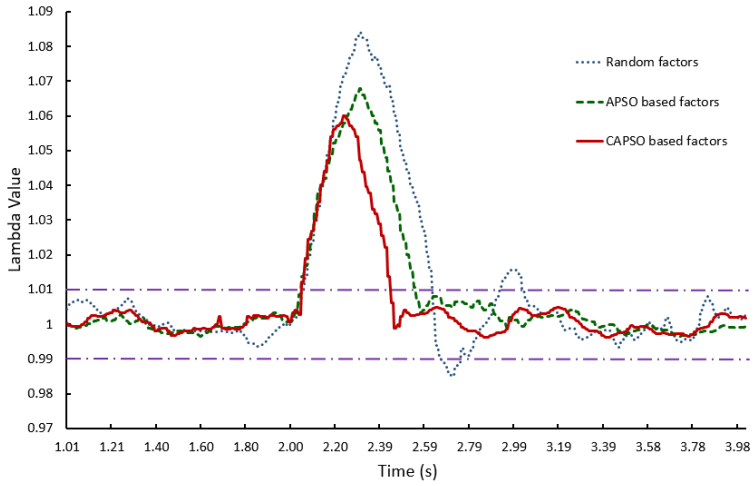
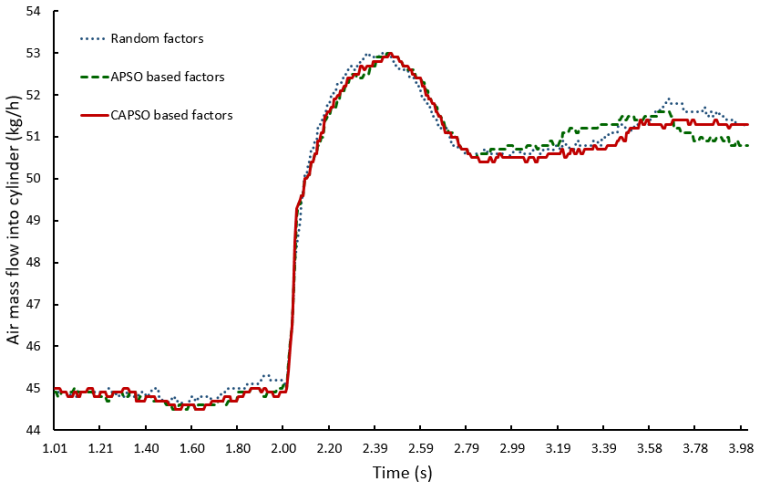


Figure 6.9 Calculated Kp, Ki, Ku factors after 20 iterations for 8% throttle position based on CAPSO

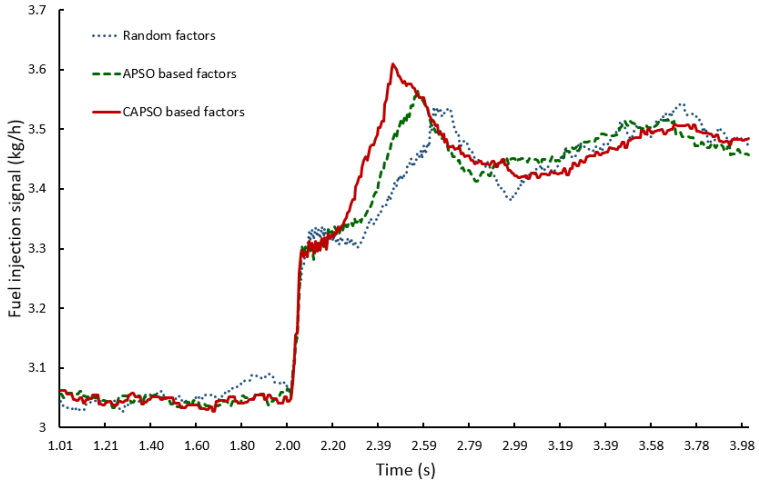
Similarly, the red points in each figure indicate the locally optimum scaling factors K_p, K_i and K_u which result in the minimum objective function after each iteration. The points in other colours indicate the other scaling factors in the swarm which lead to worse objective function. Both the APSO algorithm and the CAPSO algorithm cost 12 iterations, around 10 minutes, to converge to the stable optimal results, i.e. the globally optimum results in the swarm. The calibration results and the step responses based on the calculated scaling factors are shown as follows (Li et al., 2019b):



(a)



(b)



(c)

Figure 6.10 Step response based on calculated scaling factors for 8% throttle position: (a) **λ** value response; (b) air mass flow into the cylinder (kg/h); (c) fuel injection (kg/h)

Table 6.5 Calibration result and performance evaluation for 8% throttle position

Scaling factors	K_p	K_i	K_u	Settling time (s)	Maximum overshoot of λ	IAE	IAU	0.5*IAE +0.5*IAU
Imputed	5.00	10.00	0.20	1.13	0.083	4.973	5.461	5.217
APSO based	3.72	5.68	0.38	0.67	0.068	3.307	5.491	4.399
Reduction				**40.71%**	**18.07%**	**33.50%**		**15.68%**
CAPSO based	0.66	3.56	0.77	0.45	0.058	2.260	5.523	3.892
Reduction				**60.18%**	**30.12%**	**54.55%**		**25.41%**

Likewise, both CAPSO and APSO-based PI-like FKBCs can damp out the oscillation more effectively than the conventional PI-like FKBC; while the CAPSO-based PI-like FKBC performs better than the APSO-based PI-like FKBC. Compared with the conventional PI-like FKBC, the CAPSO-based PI-like FKBC is able to reduce 25.41% of the total objective function; while the APSO-based PI-like FKBC reduces 15.68% (Li et al., 2019b).

As the experimental results show, although both the CAPSO and APSO algorithms could converge to stable scaling factors within 12 iterations, the CAPSO algorithm is able to result in smaller values of the total objective function, namely better controller performance. Since the CAPSO algorithm employs the chaotic mapping strategy with the algorithm, it can improve both the randomization and diversification of the solutions, which allows the optimiser to explore a wider searching space and avoid falling into the trap of locally optimum results (Li et al., 2019b).

However, because both the CAPSO and APSO algorithms generate random values during the calibration process, it is not fair to compare their performance through a single case. Therefore, the statistical analysis approach is implemented to assess the performance of the algorithm through 20 times repeatability tests and statistical analysis. The results are shown in Table 6.6, including the mean value and standard deviation of the total cost function (IAE + IAU) (Li et al., 2019b).

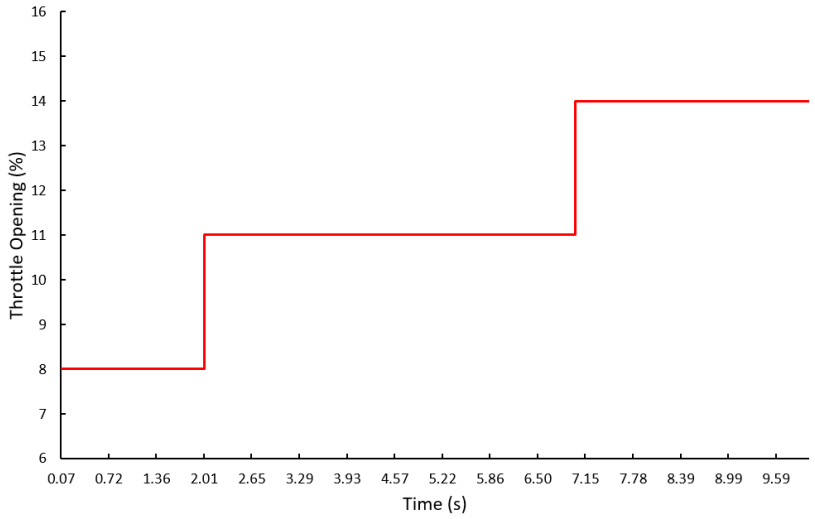
Table 6.6 Mean value and standard deviation of the total cost function using CAPSO and APSO

Throttle position	Optimisation algorithm	Mean value of cost function	Standard deviation of cost function
12%	APSO	4.422	0.139
	CAPSO	4.224	0.101
	Reduction	**4.48%**	**26.88%**
8%	APSO	4.384	0.265
	CAPSO	3.887	0.190
	Reduction	**10.61%**	**28.29%**

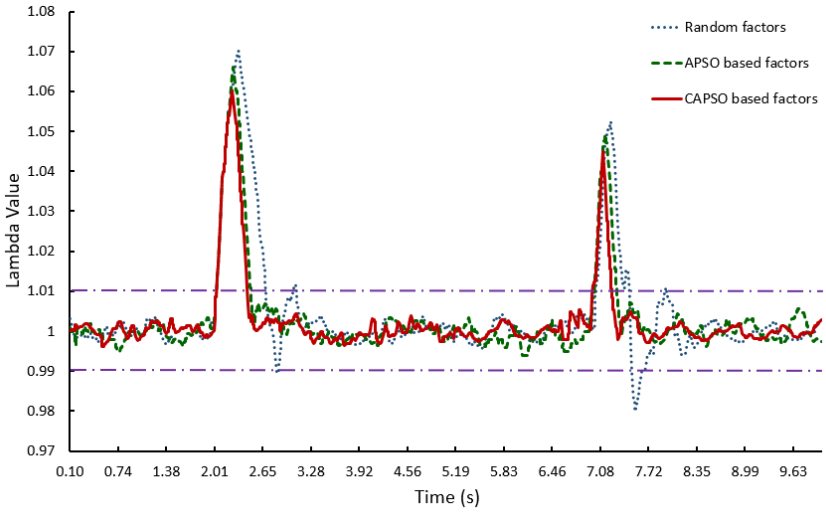
In terms of the mean value of the cost function, the CAPSO approach is able to reduce it by up to 10.61%, compared with the APSO approach. In other words, the CAPSO approach can achieve a better trade-off between reducing the oscillation effectively and avoiding an extremely large control signal. Meanwhile, the CAPSO approach can reduce the standard deviation value by up to 28.29%, compared with the conventional APSO approach. The lower standard deviation value indicates that the optimisation results fall into a narrower range (Li et al., 2019b).

6.4.2 Performance at Non-calibrated Points

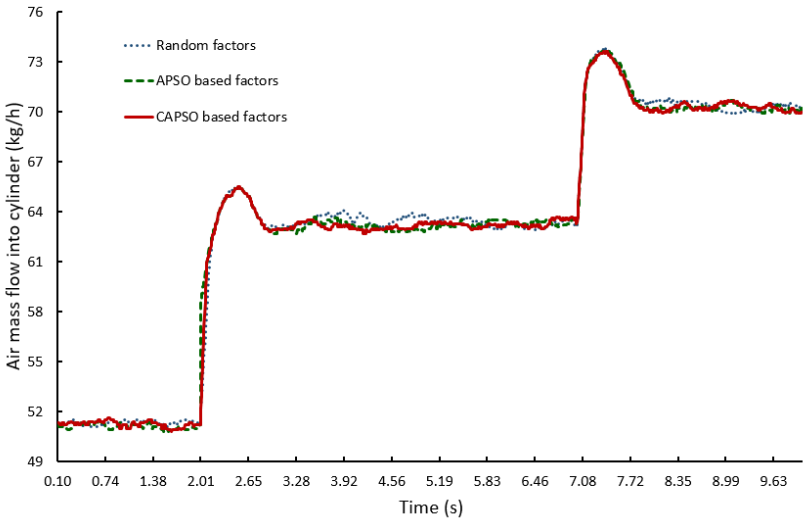
As introduced above, throttle positions of 11% and 14% are selected as non-calibration operating conditions. The transient responses of PI-like FKBCs based on APSO and CAPSO algorithms at non-calibration operating conditions are shown as follows (Li et al., 2019b):



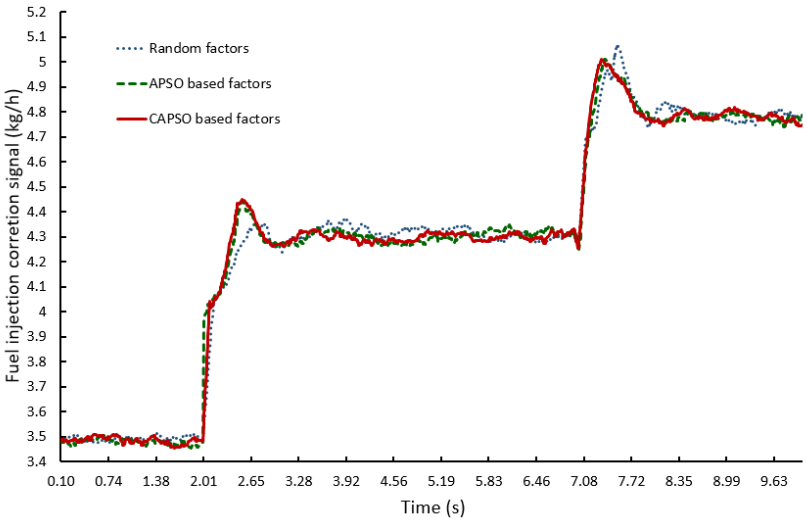
(a)



(b)



(c)



(d)

Figure 6.11 Step response at non-calibration operating conditions:
(a) throttle position step changes; (b) **λ** value response; (c) air mass flow into the cylinder (kg/h); (d) fuel injection (kg/h)

Table 6.7 Performance evaluation for non-calibration operating conditions

Step of throttle position	Controller	Settling time (s)	Maximum overshoot of λ	IAE	IAU	0.5*IAE +0.5*IAU
8%-11%	Imputed	1.11	0.070	4.518	6.527	5.523
	APSO based	0.50	0.066	2.705	6.621	4.663
	Reduction	**54.96%**	**5.71%**	**40.13%**		**15.57%**
	CAPSO based	0.45	0.060	2.275	6.615	4.445
	Reduction	**59.46%**	**14.29%**	**49.65%**		**19.52%**
11%-14%	Imputed	0.96	0.053	2.516	6.562	4.539
	APSO based	0.30	0.049	1.354	6.557	3.956
	Reduction	**68.75%**	**7.55%**	**46.18%**		**12.86%**
	CAPSO based	0.24	0.045	0.904	6.571	3.738
	Reduction	**75.00%**	**15.09%**	**64.07%**		**17.66%**

As can be seen from the experimental results, both CAPSO and APSO-based PI-like FKBCs can damp out the oscillation more effectively than the conventional self-adaptive PI-like FKBC at non-calibration operating conditions; while the CAPSO-based PI-like FKBC performs better than APSO-based PI-like FKBC. Compared with the conventional PI-like FKBC, the CAPSO-based PI-like FKBC is able to reduce the total objective function by up to 19.52%; while the APSO-based PI-like FKBC reduces the total objective function by up to 15.57%. As introduced earlier, it is because the CAPSO algorithm can find better globally optimum results at calibration operating conditions, which will definitely lead to a better transient performance at non-calibration operating conditions (Li et al., 2019b).

The transient calibration approach is able to cover a wide range of operating conditions. As introduced in section 6.2, the calibration result depends more on the selection of the

objective. For any operating conditions or step changes, once the objectives are defined as the IAE and IAU, the calibration program will repeat running the engine with different groups of scaling factors until the optimum solutions are found to achieve the minimum oscillation and the most stable control signal (Li et al., 2019b).

The experimental studies in section 5.5 have shown that two sets of initially calibrated scaling factors are able to cover all the other non-calibrated conditions. Besides, since the purpose of developing the PI-like FKBC is to achieve a favourable transient performance with minimum effort for calibration, only two sets of scaling factors need to be calibrated as reference. Then, the control signal will be adjusted automatically based on the change of the gap between the target value and the actual value, no matter how large or small. Thus, the PI-like FKBC is able to regulate the AFR effectively with any unknown step changes (Li et al., 2019b). Moreover, a large step change is actually a collection of small step changes.

More initially calibrated scaling factors may lead to not only a more reliable performance, but also more time consumption. Users can choose the trade-off between the number of initially calibrated scaling factors and the time consumption by themselves (Li et al., 2019b).

6.5 Summary

An enhanced PI-like FKBC based on the CAPSO algorithm has been developed for GDI engines to improve the AFR control system. The scaling factors of the PI-like FKBC were calibrated automatically through the CAPSO algorithm. An alternative time-domain objective function was applied for the optimisation program, without the need for prior selection of the search domain. The real-time transient performance of the enhanced PI-

like FKBC is investigated through experimental studies. The conclusions are listed as follows (Li et al., 2019b):

- The enhanced PI-like FKBC based on the CAPSO algorithm can lead to better transient response behaviour to damp out the oscillation with less settling time (up to 75% reduction) and less IAE (up to 64.07% reduction) for various operating points, compared with the conventional self-adaptive PI-like FKBC (Li et al., 2019b).
- Since both the integral of absolute error (IAE) and the integral of absolute output signal (IAU) are employed as objectives, the enhanced PI-like FKBC is able to achieve a better trade-off between reducing the oscillation effectively and avoiding a large overshoot of the control signal. It is able to reduce the total objective function (IAE+IAU) by up to 25.41%, compared with the conventional self-adaptive PI-like FKBC (Li et al., 2019b).
- The enhanced PI-like FKBC based on the CAPSO algorithm outperforms that based on the APSO algorithm. The repeatability tests indicate that the CAPSO-algorithm-based PI-like FKBC is able to reduce the mean value of the objective function by up to 10.61%, compared with the APSO-algorithm-based PI-like FKBC. Meanwhile, the standard deviation of the objective function using the CAPSO algorithm is lower (up to 28.29% reduction) than that using the APSO algorithm (Li et al., 2019b).

Overall, the proposed, enhanced, PI-like FKBC is capable of both self-adaptive capability and non-model-based automatic transient calibration capability. It can reduce the calibration effort and regulate the AFR more effectively for various engine operation conditions (Li et al., 2019b).

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

This thesis presents the achievements of the author's research over the past four years. The online dynamic optimisation for the engine management system is achieved with the help of artificial intelligence methods. The main conclusions are summarized in this chapter, followed by the plans of future work.

7.1 Conclusions

In this thesis, the intelligent non-model-based calibration approach has been developed using the multi-objective meta-heuristic algorithm, and the self-adaptive control strategy for the AFR control system has been developed using the PI-like fuzzy knowledge based controller (FKBC). The conclusions are drawn as follows corresponding to each relevant chapter.

a) The proposed intelligent non-model-based calibration approach using the multi-objective meta-heuristic algorithm is a more efficient and economical calibration approach for engines.

As Chapter 4 shows, the intelligent non-model-based calibration approach for internal combustion engines using SPEA2 and CAPSO algorithms have been developed and implemented on the real engine via RCP and external ECU bypass technologies. The experimental studies have verified that the proposed calibration approach can minimise the BSFC and PM emissions by automatically finding the optimum engine variable settings (IVO, EVC, spark timing, injection timing and rail pressure). Detailed conclusions are listed as follows:

- The intelligent non-model-based calibration approach does not rely on the engine model or massive experimental data. It can improve the automation level of the engine calibration process.
- The intelligent non-model-based calibration approach can even find the engine variable settings for lower BSFC (up to 2.9% reduction), lower PM emission number (up to 6.1% reduction), and lower PM emission mass (up to 6.6% reduction), compared to the default ECU settings.
- Compared to the SPEA2-based calibration approach, the CAPSO-based calibration approach is able to find better engine variable settings with faster speed (save at least 28.6% of the time consumption).

b) The proposed PI-like fuzzy knowledge based controller (FKBC) achieves the self-adaptive AFR control system.

As Chapter 5 shows, a discrete PI-like FKBC has been developed for GDI engines to regulate the AFR. The output signal of the PI-like FKBC was self-tuned in real-time according to the fuzzy "if-then" rules and the non-linear defuzzification algorithm. The experimental studies have verified the high robustness and self-adaptive capability of the proposed PI-like FKBC. The conclusions drawn from the investigations are:

- The proposed PI-like FKBC can lead to better transient response behaviour to damp out the oscillations with less settling time and less integral of absolute error (IAE) for various operating conditions, compared with the conventional lookup-table-based PI controller.

- The proportional term, the integral term, and the output signal of the PI-like FKBC are non-linear functions of the gap between the target and real AFRs. They are self-tuned in real-time and do not rely on the constant calibrated values, so that the PI-like FKBC can work efficiently with even poor proportional and integral gains, and save the effort and time to be spent in calibrating controller gains.

- With the same proportional and integral gains at the calibrated conditions, the proposed PI-like FKBC can reduce the settling time by up to 67.02% and reduce the IAE by up to 64.03%, compared with the conventional lookup-table-based PI controller for the transient engine operation.

- At the same non-calibrated conditions, the proposed PI-like FKBC can reduce the settling time by up to 56.25% and reduce the IAE by up to 58.67%, compared with the conventional lookup-table-based PI controller for the transient engine operation.

- The proposed PI-like FKBC has a clear and simple structure, which can facilitate its implementation for engineering problems.

c) The transient calibration method using the CAPSO algorithm can help the PI-like FKBC achieve better transient performance for the AFR control.

As Chapter 6 shows, an enhanced PI-like FKBC using the CAPSO algorithm for transient calibration has been developed to improve its transient control performance for the AFR control system. The scaling factors of the PI-like FKBC were automatically calibrated through the CAPSO algorithm. An alternative time-domain objective function was applied for the optimisation program, without the need for prior selection of the search-domain. The real-time transient performance of the enhanced PI-like FKBC is investigated through experimental studies. The detailed conclusions are listed as follow:

- The enhanced PI-like FKBC based on the CAPSO algorithm can lead to better transient response behaviour to damp out the oscillation with less settling time (up to 75% reduction) and less IAE (up to 64.07% reduction) for various operating conditions, compared with the conventional self-adaptive PI-like FKBC.
- Since both the integral of absolute error (IAE) and the integral of absolute output signal (IAU) are employed as objectives, the enhanced PI-like FKBC is able to achieve a better trade-off between reducing the oscillation effectively and avoiding large overshoot of the control signal. It can reduce the total objective function (IAE+IAU) by up to 25.41% reduction, compared with the conventional self-adaptive PI-like FKBC.
- The enhanced PI-like FKBC based on the CAPSO algorithm outperforms the one based on the APSO algorithm. The repeatability tests indicate that the CAPSO-based PI-like FKBC is able to reduce the mean value of optimisation objective function by up to 10.61% reduction compared with the APSO-based PI-like FKBC. Meanwhile, the standard deviation of the objective function using CAPSO algorithm is lower (up to 28.29% reduction) than that using APSO algorithm.
- The enhanced PI-like FKBC is capable of both self-adaptive capability and non-model-based automatic transient calibration capability. It can reduce the effort to be spent in calibrating controller parameters and regulate the AFR more effectively for various engine operating conditions.

The achievements in this thesis have been approved by our partners including Jaguar Land Rover, Ford, and Changan UK R&D Centre. The calibration engineers and technical specialists all agree with that, the author's research is very in line with the scope and

objectives of their ongoing projects. The proposed calibration and control strategies have achieved an excellent collaboration between artificial intelligence and engineering applications. They can indeed optimise the engine management system and reduce the cost for the engine calibration process.

7.2 Future Work

The dynamometer-based engine test bench at the University of Birmingham is being upgraded in order to proceed the fully transient engine operations. According to the research in this thesis, which introduces the artificial intelligence methods to figure out the engineering problems, several suggestions for the future work are provided.

- As the future powertrain systems are getting more and more complex, the number of adjustable parameters and optimisation objectives will keep rising. Meanwhile, the test of future vehicles and engines should consider the real-world driving conditions, including the weather conditions, vehicle status, geography information, and driver's behaviours, etc. Thus, they make the calibration work become a many-objective optimisation problem, whose objectives and constraints are far more than the multi-objective optimisation problem. Accordingly, further studies would like to investigate and upgrade the performance of the non-model-based calibration approach for many-objective optimisation problems (including durability, noise and other emissions), in order to better fit the industrial application.
- After the online dynamic calibration method has been developed in this thesis, further studies would like to test its performance with continuous transient operations and different driving cycles. As for the engine calibration, the vehicle driving cycles should

be converted to the engine driving cycles first. The optimiser should be investigated continually through different driving cycles. In addition, further studies are expected to apply the online dynamic calibration method for the real driving emissions (RDE) tests.

- During the first several iterative loops of the algorithm-based calibration process, the generated solutions may be distributed very widely in order to make full use of the search space and they may result in very terrible engine performance. Thus, further studies would like to design a diagnostic module to distinguish these poor solutions and skip them during the automatic calibration process while retaining the convergence performance.
- The FKBC is employed in this thesis to improve the feedback control loop of the AFR control system. Further studies would like to consider the application of FKBC in the feedforward control loop. Meanwhile, further studies are expected to simplify the design procedure of FKBC, making the fuzzy rules and membership functions easier to be understood and constructed.
- Although the calibration maps can be automatically generated by the intelligent non-model-based calibration approach, further studies would like to develop the self-adaptive maps, which can automatically adapt to the change of environmental conditions, engine’s service life, different drivers, and etc. The FKBC are expected to be applied to achieve the rapid identification, clustering and classification of this self-adaptive calibration maps.

Reference

2018 Outlook for Energy: A View to 2040 - Exxonmobil [Web Document], 2018. URL: https://corporate.exxonmobil.com/-/media/global/files/outlook-for-energy/2018-outlook-for-energy.pdf

Advanced Engine Control Algorithms - ETAS [Web Document], 2015. URL: https://www.etas.com/data/RealTimes_2015/rt_2015_1_40_en.pdf

Albin, T., Ritter, D., Abel, D., Liberda, N., Quirynen, R., Diehl, M., 2015. Non-linear MPC for a two-stage turbocharged gasoline engine airpath. In: Proceedings of the IEEE Conference on Decision and Control, Osaka, Japan, pp. 849–856. DOI: https://doi.org/10.1109/CDC.2015.7402335

Albin, T., Ritter, D., Liberda, N., Quirynen, R., Diehl, M., 2017. In-Vehicle Realization of Non-linear MPC for Gasoline Two-Stage Turbocharging Airpath Control. IEEE Trans. Control Syst. Technol. 1–13. DOI: https://doi.org/10.1109/TCST.2017.2724020

Ang, K.H., Chong, G. and Li, Y., 2005. PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology 13(4), 559-576. DOI: https://doi.org/10.1109/TCST.2005.847331

Annual Energy Outlook 2019 with projections to 2050 - U.S. Energy Information Administration (EIA) [Web Document], 2019. URL: https://www.eia.gov/outlooks/aeo/pdf/aeo2019.pdf

Arsie, I., Marra, D., Pianese, C., Sorrentino, M., 2010. Real-Time Estimation of Engine NOx Emissions via Recurrent Neural Networks. IFAC Proceedings Volumes 43(7), 228-233. DOI: https://doi.org/10.3182/20100712-3-DE-2013.00117

Asgharnia, A., Shahnazi, A. and Jamali, A., 2018. Performance and robustness of optimal fractional fuzzy PID controllers for pitch control of a wind turbine using chaotic optimization algorithms. ISA Transactions 79, 27-44. DOI: https://doi.org/10.1016/j.isatra.2018.04.016

Ashok, B., Denis Ashok, S. and Ramesh Kumar, C., 2016. A review on control system architecture of a SI engine management system. Annual Reviews in Control 41, 94-118. DOI: https://doi.org/10.1016/j.arcontrol.2016.04.005

Askari, M.R., Shahrokhi, M. and Talkhoncheh, M.K., 2017. Observer-based adaptive fuzzy controller for non-linear systems with unknown control directions and input saturation. Fuzzy Sets and Systems 314, 24-45. DOI: https://doi.org/10.1016/j.fss.2016.05.004

Athari, M. and Ardehali, M., 2016. Operational performance of energy storage as function of electricity prices for on-grid hybrid renewable energy system by optimised fuzzy logic controller. Renewable Energy 85, 890 – 902. DOI: https://doi.org/10.1016/j.renene.2015.07.055

AVL CAMEO All-In-One Powertrain Calibration – AVL. [Web Document] https://www.avl.com/documents/10138/885957/01814_PA3025E_Cameo_Leporello_Emission_web.pdf/3003168f-8898-44d8-a437-4683d0bcf812

Belkadi, A., Oulhadj, H., Touati, Y., Khan, S.A., Daachi, B., 2017. On the robust PID adaptive controller for exoskeletons: A particle swarm optimization based approach. Applied Soft Computing 60, 87-100. DOI: https://doi.org/10.1016/j.asoc.2017.06.012

Bissoli, M., Frassoldati, A., Cuoci, A., Ranzi, E., Mehl, M., Faravelli. T., 2016. A new predictive multi-zone model for HCCI engine combustion. Applied Energy 178, 826-843. DOI: https://doi.org/10.1016/j.apenergy.2016.06.062

Borhan, H., Kothandaraman, G., Pattel, B., 2015. Air handling control of a diesel engine with a complex dual-loop EGR and VGT air system using MPC. 2015 American Control Conference (ACC), Chicago, IL, 2015, 4509-4516. DOI: https://doi.org/10.1109/ACC.2015.7172039

Bosch Lambda Sensors: Ready to Install from the System Specialist – BOSCH [Web Document], 2019. URL: https://aa-boschap-uk.resource.bosch.com/media/parts/engine_systems__auto_parts_1/starters_alternators__engine_systems_1/gasoline__engine_systems_1/lambdasensor_imagefolder.pdf

BP Energy Outlook: 2019 edition - BP [Web Document], 2018. URL: https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/energy-outlook/bp-energy-outlook-2019.pdf

Bryson, J.T., Jin, X. and Agrawal, S.K., 2016. Optimal Design of Cable-Driven Manipulators Using Particle Swarm Optimization. ASME. Journal of Mechanisms and Robotics 8(4), 041003-041003-8. DOI: https://doi.org/10.1115/1.4032103

Câmara, C., 2015. 1 - Evolution and Evolutionary Algorithms. In: Câmara, C., (eds.) Bio-inspired Networking, pp. 1-30. Elsevier. DOI: https://doi.org/10.1016/B978-1-78548-021-8.50001-6

CAMBUSTION DMS500 fast particulate analyser brochure 2019 – CAMBUSTION [Web Document], 2019. URL: https://www.cambustion.com/sites/default/files/instruments/DMS500/DMS500engine.pdf

Cheng, L., Dimitriou, P., Wang, W., Peng, J., Aitouche, A. 2018. A novel fuzzy logic variable geometry turbocharger and exhaust gas recirculation control scheme for optimizing the performance and emissions of a diesel engine. International Journal of Engine Research. DOI: https://doi.org/10.1177/1468087418809261

Civelek, Z., Çam, E., Lüy, M., Mamur, H., 2016. Proportional–integral–derivative parameter optimisation of blade pitch controller in wind turbines by a new intelligent genetic algorithm. IET Renewable Power Generation 10(8), 1220-1228. DOI: https://doi.org/10.1049/iet-rpg.2016.0029

Cui, Y., Zhang, H., Wang, Y. and Jiang, H., 2019. A Fuzzy Adaptive Tracking Control for MIMO Switched Uncertain Non-linear Systems in Strict-Feedback Form. IEEE Transactions on Fuzzy Systems. DOI: https://doi.org/10.1109/TFUZZ.2019.2900610

Deb, M., Banerjee, R., Majumder, A., Sastry, G.R.K., 2014. Multi objective optimization of performance parameters of a single cylinder diesel engine with hydrogen as a dual fuel using pareto-based genetic algorithm. International Journal of Hydrogen Energy 39(15), 8063-8077. DOI: https://doi.org/10.1016/j.ijhydene.2014.03.045

Del Re, L., Ortner, P., Alberer, D., 2010. Chances and Challenges in Automotive Predictive Control. In: Automotive Model Predictive Control. Lecture Notes in Control and Information Sciences, 402, 1-22, London: Springer. DOI: https://doi.org/10.1007/978-1-84996-071-7_1

Deng, W., Yao, R., Zhao, H., Yang, X., Li, G., 2019. A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. Soft Computing 23(7), 2445–2462. DOI: https://doi.org/10.1007/s00500-017-2940-9

Dettori, S., Iannino, V., Colla, V., Signorini, A., 2018. An adaptive Fuzzy logic-based approach to PID control of steam turbines in solar applications. Applied Energy 227, 655-664. DOI: https://doi.org/10.1016/j.apenergy.2017.08.145

DMS500 for aerosol size, number and mass measurements – CAMBUSTION [Web Document], 2018. URL: https://www.cambustion.com/sites/default/files/instruments/DMS500/DMS500aerosol.pdf

Dogruer, T. and Tan, N., 2018. Design of PI Controller using Optimization Method in Fractional Order Control Systems. IFAC-PapersOnLine 51(4), 841-846. DOI: https://doi.org/10.1016/j.ifacol.2018.06.124

Energy, transport and environment indicators 2018 edition – Eurostat [Web Document], 2018. URL: https://ec.europa.eu/eurostat/web/products-statistical-books/-/KS-DK-18-001

Gandomi, A.H., Yun, G.J., Yang, X.S., Talatahari, S., 2013. Chaos-enhanced accelerated particle swarm optimization. Communications in Non-linear Science and Numerical Simulation 18(2), 327-340. DOI: https://doi.org/10.1016/j.cnsns.2012.07.017

García, A., Monsalve-Serrano, J., 2019. Analysis of a series hybrid vehicle concept that combines low temperature combustion and biofuels as power source. Results in Engineering 1, 100001. DOI: https://doi.org/10.1016/j.rineng.2019.01.001

García, A., Monsalve-Serrano, J., Sari, R., Dimitrakopoulos, N., Tunér, M., Tunestål, P., 2019. Performance and emissions of a series hybrid vehicle powered by a gasoline partially premixed combustion engine. Applied Thermal Engineering 150, 564-575. DOI: https://doi.org/10.1016/j.applthermaleng.2019.01.035

Grasreiner, S., Neumann, J., Wensing, M., Hasse, C., 2017. Model-based virtual engine calibration with the help of phenomenological methods for spark-ignited engines. Applied Thermal Engineering 121, 190-199. DOI: https://doi.org/10.1016/j.applthermaleng.2017.04.046

Guzmán, J.L., Moreno, J.C., Berenguel, M., Moscoso, J., 2018. Inverse pole placement method for PI control in the tracking problem. IFAC-PapersOnLine 51(4), 406-411. DOI: https://doi.org/10.1016/j.ifacol.2018.06.128

Guzzella, L. and Onder, Ch.H., 2010. Introduction to Modelling and Control of Internal Combustion Engine Systems. 2nd Edition, Springer Editor. DOI: http://dx.doi.org/10.1007/978-3-642-10775-7

Hellström, E., Lee, D., Jiang, L., Stefanopoulou, A.G., Yilmaz, H., 2013. On-Board Calibration of Spark Timing by Extremum Seeking for Flex-Fuel Engines. IEEE Transactions on Control Systems Technology 21(6), 2273-2279. DOI: https://doi.org/10.1109/TCST.2012.2236093

Hendricks, E. and Sorenson, S., 1991. SI Engine Controls and Mean Value Engine Modelling. SAE Technical Paper 910258. DOI: https://doi.org/10.4271/910258

Heywood, J.B., 1988. Internal Combustion Engine Fundamentals. New York: McGraw Hill International.

Hirata, M., Ishizuki, S. and Suzuki, M., 2017. Two-degree-of-freedom H-infinity control of combustion in diesel engine using a discrete dynamics model. Control Theory and Technology 15(2), 109–116. DOI: https://doi.org/10.1007/s11768-017-6144-8

Hu, Y., Chen, H., Wang, P., Chen, H., Ren, L., 2018. Non-linear model predictive controller design based on learning model for turbocharged gasoline engine of passenger vehicle. Mechanical Systems and Signal Processing 109, 74-88. DOI: https://doi.org/10.1016/j.ymssp.2018.02.012

Huang, Y., Surawski, N.C., Organ, B., Zhou, J.L., Tang, O.H.H., Chan, E.F.C., 2019. Fuel consumption and emissions performance under real driving: Comparison between hybrid and conventional vehicles. Science of the Total Environment 659, 275-282. DOI: https://doi.org/10.1016/j.scitotenv.2018.12.349

International Energy Outlook 2017 - U.S. Energy Information Administration (EIA) [Web Document], 2017. URL: https://www.eia.gov/outlooks/ieo/pdf/0484(2017).pdf

Janakiraman, V.M., Nguyen, X., and Assanis, D., 2016. Engineering Applications of Artificial Intelligence 48, 106-118. DOI: https://doi.org/10.1016/j.engappai.2015.10.007

Jiang, S., Nutter, D., and Gullitti, A., 2012. Implementation of Model-Based Calibration for a Gasoline Engine, SAE Technical Paper 2012-01-0722. DOI: https://doi.org/10.4271/2012-01-0722

Kalghatgi, G.T., 2018. Is it really the end of internal combustion engines and petroleum in transport? Applied Energy 225, 965–974. DOI: https://doi.org/10.1016/j.apenergy.2018.05.076

Kaveh, A. and Dadras, A., 2017. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. Advances in Engineering Software 110, 69-84. DOI: https://doi.org/10.1016/j.advengsoft.2017.03.014

Keshavarz, M., Kakaee, AH. and Fajri, H. 2018. Journal of the Brazilian Society of Mechanical Sciences and Engineering 40, 358. DOI: https://doi.org/10.1007/s40430-018-1278-2

Kianifar, M., Campean, L., and Richardson, D., 2013. Sequential DoE Framework for Steady State Model Based Calibration. SAE International Journal of Engines 6(2), 843-855. DOI: https://doi.org/10.4271/2013-01-0972

Kianifar, M.R., Campean, F. and Wood, A., 2016. Application of permutation genetic algorithm for sequential model building–model validation design of experiments. Soft Computing 20(8), 3023–3044. DOI: https://doi.org/10.1007/s00500-015-1929-5

Kihas, D. and Uchanski, M. 2015. Engine-Out NOx Models for on-ECU Implementation: A Brief Overview. SAE Technical Paper 2015-01-1638. DOI: https://doi.org/10.4271/2015-01-1638

Koli, R., Arunachalam, H., Zhu, Q., Onori, S., Vahidi, A., Prucka, R., 2018. Non-linear Model Predictive Control of Dual Loop - Exhaust Gas Recirculation in a Turbocharged Spark Ignited engine. 2018 Annual American Control Conference (ACC), Milwaukee, WI, 2437-2442. DOI: https://doi.org/10.23919/ACC.2018.8430853

Kumanan, D. and Nagaraj, B., 2013. Tuning of proportional integral derivative controller based on firefly algorithm. Systems Science & Control Engineering 1(1), 52-56. DOI: https://doi.org/10.1080/21642583.2013.770375

Kumar, S., Chauhan, M.K., and Varun, 2013. Numerical modeling of compression ignition engine: A review. Renewable and Sustainable Energy Reviews 19, 517-530. DOI: https://doi.org/10.1016/j.rser.2012.11.043

Lambda Sensor LSU 4.9 Data sheet - BOSCH [Web Document], 2019. URL: https://www.bosch-motorsport.com/content/downloads/Raceparts/Resources/pdf/Data%20sheet_69034379_Lambda_Sensor_LSU_4.9.pdf

Lee, B. and Jung, D., 2016. Thermodynamics-based mean value engine model with main and pilot injection sensitivity. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 230(13), 1822–1834. DOI: https://doi.org/10.1177/0954407015624525

Lee, B., Jung, D., Kim, Y., van Nieuwstadt, M., 2013. Thermodynamics-Based Mean Value Model for Diesel Combustion. ASME. Journal of Engineering for Gas Turbines and Power 135(9), 091504-091504-9. DOI: https://doi.org/10.1115/1.4024757

Lee, T.-K., and Filipi, Z. S., 2011. Non-linear model predictive control of a dual-independent variable valve timing engine with electronic throttle control. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 225(9), 1221–1234. DOI: https://doi.org/10.1177/0954407011407691

Li, C., Jing, H., Wang, R., Chen, N., 2018. Vehicle lateral motion regulation under unreliable communication links based on robust H∞ output-feedback control schema. Mechanical Systems and Signal Processing 104, 171-187. DOI: https://doi.org/10.1016/j.ymssp.2017.09.012

Li, J., Zhou, Q., He, Y., Shuai, B., Li, Z., Williams, H., Xu, H., 2019c. Dual-loop online intelligent programming for driver-oriented predict energy management of plug-in hybrid electric vehicles. Applied Energy 253, 2019, 113617. DOI: https://doi.org/10.1016/j.apenergy.2019.113617

Li, Y., Ang, K.H. and Chong, G. C. Y., 2006. PID control system analysis and design. IEEE Control Systems Magazine 26(1), 32-41. DOI: https://doi.org/10.1109/MCS.2006.1580152

Li, Z., Li, J., Zhou, Q., Zhang, Y., Xu, H., 2019a. Intelligent air/fuel ratio control strategy with a PI-like fuzzy knowledge–based controller for gasoline direct injection engines. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 233(8), 2161–2173. DOI: https://doi.org/10.1177/0954407018779180

Li, Z., Zhou, Q., Zhang, Y., Li, J., Xu, H., 2019b. Enhanced intelligent proportional-integral-like fuzzy knowledge–based controller using chaos-enhanced accelerated particle swarm optimization algorithm for transient calibration of air–fuel ratio control system. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. DOI: https://doi.org/10.1177/0954407019862079

Liu, H., Li, S., Cao, J. et al., 2017. Adaptive fuzzy prescribed performance controller design for a class of uncertain fractional-order non-linear systems with external disturbances. Neurocomputing 219, 422-430. DOI: https://doi.org/10.1016/j.neucom.2016.09.050

Liu, J., Wang, J. and Zhao, H., 2018. Optimization of the injection parameters and combustion chamber geometries of a diesel/natural gas RCCI engine. Energy 164, 837-852. DOI: https://doi.org/10.1016/j.energy.2018.09.064

Liu, Z., Zheng, Z. and Li, Y., 2017. Enhancing fault-tolerant ability of a nine-phase induction motor drive system using fuzzy logic current controllers. IEEE Transactions on Energy Conversion 32(2), 759 – 769. DOI: https://doi.org/10.1109/TEC.2017.2692528

Lu, X., Liu, M., Liu, J., 2017. Design and Optimization of Interval Type-2 Fuzzy Logic Controller for Delta Parallel Robot Trajectory Control. International Journal of Fuzzy Systems 19(1), 190-206. DOI: https://doi.org/10.1007/s40815-015-0131-3

Lucchini, T., Della Torre, A., D'Errico, G., Montenegro, G. Fiocco, M., Maghbouli, A., 2015. Automatic Mesh Generation for CFD Simulations of Direct-Injection Engines. SAE Technical Paper 2015-01-0376. DOI: https://doi.org/10.4271/2015-01-0376

M. Sabri, M.F., Danapalasingamn, K.A. and Rahma, M.F., 2016. A review on hybrid electric vehicles architecture and energy management strategies. Renewable and Sustainable Energy Reviews 53, 1433-1442. DOI: https://doi.org/10.1016/j.rser.2015.09.036

Ma, F., Wang, Y., Wang, M., Liu, H., Wang, J., Ding, S., Zhao. S., 2008. Development and validation of a quasi-dimensional combustion model for SI engines fuelled by HCNG with variable hydrogen fractions. International Journal of Hydrogen Energy 33(18), 4863-4875. DOI: https://doi.org/10.1016/j.ijhydene.2008.06.068

Ma, H., 2013. Control Oriented Engine Modelling and Engine Multi-Objective Optimal Feedback Control. PhD Thesis. University of Birmingham. URL: https://etheses.bham.ac.uk/id/eprint/4308

Ma, H., Li, Z., Tayarani, M., Lu, G., Xu, H., Yao, X., 2017. Computational Intelligence Nonmodel-Based Calibration Approach for Internal Combustion Engines. ASME. Journal of Dynamic Systems, Measurement, and Control 140(4), 041002-041002-9. DOI: https://doi.org/10.1115/1.4037835

Ma, H., Li, Z., Tayarani, M., Lu, G., Xu, H., Yao, X., 2019. Model-based computational intelligence multi-objective optimization for gasoline direct injection engine calibration. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 233(6), 1391–1402. DOI: https://doi.org/10.1177/0954407018776743

Ma, H., Xu, H., Wang, J., Schnier, T., Neaves, B., Tan, C., Wang, Z., 2015. Model-Based Multiobjective Evolutionary Algorithm Optimization for HCCI Engines. IEEE Transactions on Vehicular Technology 64(9), 4326-4331. DOI: https://doi.org/10.1109/TVT.2014.2362954

Maheta, H.H. and Dabhi, V.K., 2014. An improved SPEA2 Multi objective algorithm with non dominated elitism and Generational Crossover. In: 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, pp. 75-82. DOI: https://doi.org/10.1109/ICICICT.2014.6781256

Mendes, N. and Neto, P., 2015. Indirect adaptive fuzzy control for industrial robots: A solution for contact applications. Expert Systems with Applications 42(22), 8929-8935. DOI: https://doi.org/10.1016/j.eswa.2015.07.047

Meshram, P.M. and Kanojiya, R.G., 2012. Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor. In: IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM - 2012), Nagapattinam, Tamil Nadu, India, pp. 117-121. URL: https://ieeexplore.ieee.org/document/6216102

Millo, F., Rolando, L. and Andreata, M., 2011. Numerical Simulation for Vehicle Powertrain Development. In: Numerical Analysis - Theory and Application. InTech. DOI: https://doi.org/10.5772/24111

Milloa, F., Arya, P., Mallamo, F., 2018. Optimization of automotive diesel engine calibration using genetic algorithm techniques. Energy 158, 807-819. DOI: https://doi.org/10.1016/j.energy.2018.06.044

Montazeri-Gh, M., Jafari, S. and Ilkhani, M.R., 2012. Application of particle swarm optimization in gas turbine engine fuel controller gain tuning. Engineering Optimization 44(2), 225-240. DOI: https://doi.org/10.1080/0305215X.2011.576760

Mu, H., Zhao, Y. and Li, X., 2018. Adaptive Fuzzy Sliding-Mode Control for Piston Stop Position of the GDI Engine. 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, 1598-1603. DOI: https://doi.org/10.1109/CAC.2018.8623805

Müller, M., Hendricks, E. and Sorenson, S., 1998. Mean Value Modelling of Turbocharged Spark Ignition Engines. SAE Technical Paper 980784. DOI: https://doi.org/10.4271/980784

Murtaza, G.G., Bhatti, A.I., Ahmed, Q.Q., 2016. Control-Oriented Model of Atkinson Cycle Engine With Variable Intake Valve Actuation. ASME. J. Dyn. Sys., Meas., Control 138(6), 061001-061001-9. DOI: https://doi:10.1115/1.4032746

Nazoktabar, M., Jazayeri, S.A., Arshtabar, K., Ganji, D.D., 2018. Developing a multi-zone model for a HCCI engine to obtain optimal conditions using genetic algorithm. Energy Conversion and Management 157, 49-58. DOI: https://doi.org/10.1016/j.enconman.2017.12.001

Netjinda, N., Achalakul, T. and Sirinaovakul, B., 2015. Particle Swarm Optimization inspired by starling flock behaviour. Applied Soft Computing 35, 411-422. DOI: https://doi.org/10.1016/j.asoc.2015.06.052

New Blocksets from ETAS for Simulink® - ETAS [Web Document], 2013. URL: https://www.etas.com/data/RealTimes_2013/rt_2013_1_48_en.pdf

Nobakht, A.Y., Saray, R.K., Rahimi, A., 2011. A parametric study on natural gas fueled HCCI combustion engine using a multi-zone combustion model. Fuel 90(4), 1508-1514. DOI: https://doi.org/10.1016/j.fuel.2010.12.026

Pahasa, J. and Ngamroo, I., 2015. PHEVs Bidirectional Charging/Discharging and SoC Control for Microgrid Frequency Stabilization Using Multiple MPC. IEEE Transactions on Smart Grid 6(2), 526-533. DOI: https://doi.org/10.1109/TSG.2014.2372038

Pan, Y. and Wang, J., 2012. Model Predictive Control of Unknown Non-linear Dynamical Systems Based on Recurrent Neural Networks. IEEE Transactions on Industrial Electronics 59(8), 3089-3101. DOI: https://doi.org/10.1109/TIE.2011.2169636

Park, J. and Choi, J., 2016. Optimization of dual-loop exhaust gas recirculation splitting for a light-duty diesel engine with model-based control. Applied Energy 181, 268-277. DOI: https://doi.org/10.1016/j.apenergy.2016.07.128

Perini, F., Paltrinieri, F. and Mattarelli, E., 2010. A quasi-dimensional combustion model for performance and emissions of SI engines running on hydrogen-methane blends.

International Journal of Hydrogen Energy 35(10), 4687-4701. DOI: https://doi.org/10.1016/j.ijhydene.2010.02.083

Product data sheet for planar broadband lambda sensor - BOSCH [Web Document], 2019. URL: https://www.bosch-mobility-solutions.com/media/global/products-and-services/commercial-vehicles/powertrain-solutions/natural-gas/planar-broadband-lambda-sensor/product_data_sheet_lamda_sensor.pdf

Rahimi-Gorji, M., Ghajar, M., Kakaee, AH. et al. 2017. Modeling of the air conditions effects on the power and fuel consumption of the SI engine using neural networks and regression. Journal of the Brazilian Society of Mechanical Sciences and Engineering 39(2), 375-384. DOI: https://doi.org/10.1007/s40430-016-0539-1

Rapid Prototyping of New Control Functions - ETAS [Web Document], 2012. URL: https://www.etas.com/data/RealTimes_2012/rt_2012_2_18_en.pdf

Raza, M., Chen, L., Leach, F., Ding, S., 2018. A Review of Particulate Number (PN) Emissions from Gasoline Direct Injection (GDI) Engines and Their Control Techniques. Energies 11(6), 1417. DOI: https://doi.org/10.3390/en11061417

Razvarz, S., Vargas-Jarillo, C., Jafari, R., Gegov, A., 2019. Flow Control of Fluid in Pipelines Using PID Controller. IEEE Access 7, 25673-25680. DOI: https://doi.org/10.1109/ACCESS.2019.2897992

RHEONIK Coriolis Mass Flow Meters – RHEONIK [Web Document], 2014. URL: https://able.co.uk/media/2014/01/rheonik-transmitter-flow-meter-datasheet.pdf

RHEONIK Coriolis Mass Flowmeter RHM 03 Datasheet – RHEONIK [Web Document], 2018. URL: https://www.rheonik.com/wp-content/uploads/pdf/Rheonik_Coriolis_RHM015L,02L,03L,04L_Datasheet.pdf

Rodríguez-Castellanos, J.E., Grisales-Palacio, V.H. and Cote-Ballesteros, J.E., 2018. A tuning proposal for direct fuzzy PID controllers oriented to industrial continuous processes. IFAC-PapersOnLine 51(4), 657-662. DOI: https://doi.org/10.1016/j.ifacol.2018.06.172

Rosaline, A. D. and Somarajan, U., 2019. Structured H-Infinity Controller for an Uncertain Deregulated Power System. IEEE Transactions on Industry Applications 55(1), 892-906. DOI: https://doi.org/10.1109/TIA.2018.2866560

Saied, H., Chemori, A., El Rafei, M., Francis, C., Pierrot, F., 2019. From Non-model-Based to Model-Based Control of PKMs: A Comparative Study. In: Rizk, R., Awad, M., (eds.) Mechanism, Machine, Robotics and Mechatronics Sciences. Mechanisms and Machine Science 58, pp. 153-169. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-89911-4_12

Salehi, S. and Shahrokhi, M., 2012. Adaptive output feedback tracking controller for a class of uncertain strict feedback non-linear systems in the absence of state measurements. International Journal of Systems Science 43(2), 201-210. DOI: https://doi.org/10.1080/00207721.2010.488755

Sardarmehni, T., Keighobadi, J., Menhaj, M.B., Rahmani, H., 2013. Robust predictive control of lambda in internal combustion engines using neural networks. Archives of Civil and Mechanical Engineering 13(4), 432-443. DOI: https://doi.org/10.1016/j.acme.2013.05.003

Senouci, M.R. and Mellouk, A., 2016. 5 - Dynamic Deployment. Deploying Wireless Sensor Networks, 89-117. DOI: https://doi.org/10.1016/B978-1-78548-099-7.50005-2

Shahnazi, R., 2016. Observer-based adaptive interval type-2 fuzzy control of uncertain MIMO non-linear systems with unknown asymmetric saturation actuators. Neurocomputing 171, 1053-1065. DOI: https://doi.org/10.1016/j.neucom.2015.07.098

Shamekhi, A. and Shamekhi, A.H., 2015. A new approach in improvement of mean value models for spark ignition engines using neural networks. Expert Systems with Applications 42, 5192–5218. DOI: https://doi.org/10.1016/j.eswa.2015.02.031

Shi, Y., Yu, D., Tian, Y., Shi, Y., 2015. Air–fuel ratio prediction and NMPC for SI engines with modified Volterra model and RBF network. Engineering Applications of Artificial Intelligence 45, 313-324. DOI: https://doi.org/10.1016/j.engappai.2015.07.008

Shibata, G., Ogawa, H., Amanuma, Y., Okamoto, Y., 2019. Optimization of multiple heat releases in pre-mixed diesel engine combustion for high thermal efficiency and low combustion noise by a genetic-based algorithm method. International Journal of Engine Research 20(5), 540–554. DOI: https://doi.org/10.1177/1468087418767225

Shuai, S., Ma, X., Li, Y., Qi, Y., Xu, H., 2018. Recent Progress in Automotive Gasoline Direct Injection Engine Technology. Automotive Innovation 1, 95–113. DOI: https://doi.org/10.1007/s42154-018-0020-1

Slowik, A. and Kwasnicka, H., 2018. Nature Inspired Methods and Their Industry Applications—Swarm Intelligence Algorithms. IEEE Transactions on Industrial Informatics 14(3), 1004-1015. DOI: https://doi.org/10.1109/TII.2017.2786782

Song, Y., Wang, Y. and Wen, C., 2017. Adaptive Fault-Tolerant PI Tracking Control with Guaranteed Transient and Steady-State Performance. IEEE Transactions on Automatic Control 62(1), 481-487. DOI: https://doi.org/10.1109/TAC.2016.2554362

Sravan Bharadwaj, C., Sudhakar Babu, T., Rajasekar, N., 2018. Tuning PID Controller for Inverted Pendulum Using Genetic Algorithm. In: Konkani, A., Bera, R., Paul, S. (eds.) Advances in Systems, Control and Automation. Lecture Notes in Electrical Engineering 442, pp 395-404. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-10-4762-6_38

Stone, R., 2012. Introduction to Internal Combustion Engines. 4th edition. Palgrave Macmillan.

Suzuki, T., 1997. The romance of engines (book). Warrendale, PA, Society of Automotive Engineers.

Takahashi, M., Yamasaki, Y., Kaneko, S., Koizumi, J., Hayashi, T., Hirata, M., 2018. Model-Based Control System for Air Path and Premixed Combustion of Diesel Engine. IFAC-PapersOnLine 51(31), 522-528. DOI: https://doi.org/10.1016/j.ifacol.2018.10.114

Tan, C., Xu, H., Ma, H. and Ghafourian, A., 2014. Investigation of VVT and spark timing on combustion and particle emission from a GDI Engine during transient operation. SAE Technical Paper 2014-01-1370. DOI: https://doi.org/10.4271/2014-01-1370

Tayarani-N, M., Yao, X. and Xu, H., 2015. Meta-Heuristic Algorithms in Car Engine Design: A Literature Survey. IEEE Transactions on Evolutionary Computation 19(5), 609-629. DOI: https://doi.org/10.1109/TEVC.2014.2355174

The common rail diesel engine calibration manual - International-Limited-of-China-Yuchai, 2007. URL: http://wenku.baidu.com/view/3241b6232f60ddccda38a014.html [Access from 06.06.2013]

Theotokatos, G., Guan, C., Chen, H., Lazakis, I., 2018. Development of an extended mean value engine model for predicting the marine two-stroke engine operation at varying settings. Energy 143, 533-545. DOI: https://doi.org/10.1016/j.energy.2017.10.138

The Roadmap Report Towards 2040: A Guide to Automotive Propulsion Technologies - UK Advanced Propulsion Centre (APC) [Web Document], 2018. URL: https://www.apcuk.co.uk/app/uploads/2018/06/roadmap-report-26-6-18.pdf

Torrisi, G., Grammatico, S., Smith, R. S., Morari, M., 2016. A variant to Sequential Quadratic Programming for non-linear Model Predictive Control. In: 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, pp. 2814-2819. DOI: https://doi.org/10.1109/CDC.2016.7798688

Turkson, R.F., Yan, F., Ali, M.K.A., Hu, J., 2016. Artificial neural network applications in the calibration of spark-ignition engines: An overview. Engineering Science and Technology, an International Journal 19(3), 1346-1359. DOI: https://doi.org/10.1016/j.jestch.2016.03.003

Verhelst, S. and Sheppard, C.G.W., 2009. Multi-zone thermodynamic modelling of spark-ignition engine combustion – An overview. Energy Conversion and Management 50, 1326–1335. DOI: https://doi.org/10.1016/j.enconman.2009.01.002

Verma, O.P., Manik, G. and Jain, V.K., 2018. Simulation and control of a complex non-linear dynamic behavior of multi-stage evaporator using PID and Fuzzy-PID controllers. Journal of Computational Science 25, 238-251. DOI: https://doi.org/10.1016/j.jocs.2017.04.001

Wahlström, J. and Eriksson, L. 2011. Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 225(7), 960–986. DOI: https://doi.org/10.1177/0954407011398177

Wang, H., Liu, X., Liu, K., Karimi, H.R., 2015. Approximation-Based Adaptive Fuzzy Tracking Control for a Class of Nonstrict-Feedback Stochastic Non-linear Time-Delay Systems. IEEE Transactions on Fuzzy Systems 23(5), 1746-1760. DOI: https://doi.org/10.1109/TFUZZ.2014.2375917

Wang, R., Li, X., Zhang, J. Zhang, J., Li, W., Liu, Y., Fu, W., Ma, X., 2018. Speed Control for a Marine Diesel Engine Based on the Combined Linear-Non-linear Active Disturbance Rejection Control. Mathematical Problems in Engineering 2018, 7641862. DOI: https://doi.org/10.1155/2018/7641862

Wang, S.W., Yu, D.L., Gomm, J.B., Douglas, S.S., 2006. Adaptive neural network model based predictive control for air–fuel ratio of SI engines. Engineering Applications of Artificial Intelligence 19(2), 189-200. DOI: https://doi.org/10.1016/j.engappai.2005.08.005

Wang, S. and Yu, D., 2008. Adaptive RBF network for parameter estimation and stable air–fuel ratio control. Neural Networks 21(1), 102-112. DOI: https://doi.org/10.1016/j.neunet.2007.10.006

Wang, Y., 2017. Control and Calibration for Real World Fuel Economy and Emissions - Advanced Control Methods, Ford Research & Innovation Centre. In: The Future Powertrain Conference 2017. URL: https://futurepowertrains.co.uk/wp-content/uploads/2017/12/Yan-Wang.pdf

Wati, D. A. R. and Hidayat, R., 2013. Genetic algorithm-based PID parameters optimization for air heater temperature control. In: 2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems, Jogjakarta, pp. 30-34. DOI: https://doi.org/10.1109/ROBIONETICS.2013.6743573

Wong, P.K., Gao, X.H., Wong, K.I., Vong, C.M., 2018a. Efficient point-by-point engine calibration using machine learning and sequential design of experiment strategies. Journal of the Franklin Institute 355(4), 1517-1538. https://doi.org/10.1016/j.jfranklin.2017.02.006

Wong, P.K., Gao, X.H., Wong, K.I., Vong, C.M., 2018b. Online extreme learning machine based modeling and optimization for point-by-point engine calibration. Neurocomputing 277, 187-197. DOI: https://doi.org/10.1016/j.neucom.2017.02.104

Wong, P.K., Wong, H.C., Vong, C.M., Xie, Z., Huang, S., 2016. Model predictive engine air-ratio control using online sequential extreme learning machine. Neural Computing and Applications 27(1), 79-92. DOI: https://doi.org/10.1007/s00521-014-1555-7

World Oil Outlook 2018 - Organization of the Petroleum Exporting Countries (OPEC) [Web Document], 2018. URL: https://woo.opec.org/index.html

Wu, D. and Gao, H., 2018. Proceedings of the National Academy of Sciences, India Section A: Physical Sciences 88(1), 121–128. DOI: 121. https://doi.org/10.1007/s40010-016-0320-y

Xiang, Z., Ji, D., Zhang, H., Wu, H., Li, Y., 2019. A simple PID-based strategy for particle swarm optimization algorithm. Information Sciences 502, 558-574. DOI: https://doi.org/10.1016/j.ins.2019.06.042

Xu, H., Wang, C., Ma, X., Sarangi, A.K., Weall, A., Krueger-Venus, J., 2015. Fuel injector deposits in direct-injection spark-ignition engines. Progress in Energy and Combustion Science 50, 63–80. DOI: https://doi.org/10.1016/j.pecs.2015.02.002

Xue, X., and Caton, J. A. 2012. Detailed multi-zone thermodynamic simulation for direct-injection diesel engine combustion. *International Journal of Engine Research 13*(4), 340–356. DOI: https://doi.org/10.1177/1468087411435206

Yadav, S., Verma, S., Nagar, S., 2016. Optimised PID controller for magnetic levitation system. IFAC Papersonline 49(1), 778–782. DOI: https://doi.org/10.1016/j.ifacol.2016.03.151

Yan, Z. and Wang, J., 2014. Robust Model Predictive Control of Non-linear Systems with Unmodeled Dynamics and Bounded Uncertainties Based on Neural Networks. IEEE Transactions on Neural Networks and Learning Systems 25(3), 457-469. DOI: https://doi.org/10.1109/TNNLS.2013.2275948

Yang, D., Liu, Z., Yi, P., 2017. Computational efficiency of accelerated particle swarm optimization combined with different chaotic maps for global optimization. Neural Computing and Applications 28(1), 1245–1264. DOI: https://doi.org/10.1007/s00521-016-2433-2

Yang, M., Li, Y., Du, H., Li, C., He, Z., 2019. Hierarchical Multiobjective H-Infinity Robust Control Design for Wireless Power Transfer System Using Genetic Algorithm. IEEE Transactions on Control Systems Technology 27(4), 1753-1761. DOI: https://doi.org/10.1109/TCST.2018.2814589

Yang, X.-S., 2014. Chapter 1 - Introduction to Algorithms. Nature-Inspired Optimization Algorithms, 1-21. DOI: https://doi.org/10.1016/B978-0-12-416743-8.00001-4

Yu, B., Cao, C., Shu, W., Hu, Z., 2017. A New Method for the Design of Optimal Control in the Transient State of a Gas Turbine Engine. IEEE Access 5, 23848-23857. DOI: https://doi.org/10.1109/ACCESS.2017.2764056

Zadeh, L.A., 1965. Fuzzy sets. Information and Control 8(3), 338-353. DOI: https://doi.org/10.1016/S0019-9958(65)90241-X

Zhai, Y.J., Yu, D.W., Guo, H.Y., Yu, D.L., 2010. Robust air/fuel ratio control with adaptive DRNN model and AD tuning. Engineering Applications of Artificial Intelligence 23(2), 283 -289. DOI: https://doi.org/10.1016/j.engappai.2009.12.006

Zhang, M., Hong, W., Xie, F., Su, Y., Liu, H., Zhou, S., 2018. Combustion, performance and particulate matter emissions analysis of operating parameters on a GDI engine by traditional experimental investigation and Taguchi method. Energy Conversion and Management 164, 344-352. DOI: https://doi.org/10.1016/j.enconman.2018.03.017

Zhang, Y., Lu, G., Xu, H., Li, Z., 2018. Tuneable model predictive control of a turbocharged diesel engine with dual loop exhaust gas recirculation. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 232(8), 1105–1120. DOI: https://doi.org/10.1177/0954407017726944

Zhang, Y., Su, X., Liu, Z. and Chen, C. L. P., 2019. Event-Triggered Adaptive Fuzzy Tracking Control With Guaranteed Transient Performance for MIMO Non-linear Uncertain Systems. IEEE Transactions on Cybernetics. DOI: https://doi.org/10.1109/TCYB.2019.2894343

Zhang, Y., Zhou, Q., Li, Z., Li, J., Xu, H., 2019. Intelligent transient calibration of a dual-loop EGR diesel engine using chaos-enhanced accelerated particle swarm optimization algorithm. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 233(7), 1698–1711. DOI: https://doi.org/10.1177/0954407018776745

Zhou, Q., Zhang, W., Cash, S., Olatunbosun, O., Xu, H., Lu, G., 2017. Intelligent sizing of a series hybrid electric power-train system based on Chaos-enhanced accelerated particle swarm optimization. Applied Energy 189, 588-601. DOI: https://doi.org/10.1016/j.apenergy.2016.12.074

Zhu, Q., Onori, S. and Prucka, R., 2016. Non-linear economic Model Predictive Control for SI engines based on Sequential Quadratic Programming. 2016 American Control

Conference (ACC), Boston, MA, 1802-1807. DOI: https://doi.org/10.1109/ACC.2016.7525180

Zitzler, E. and Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257-271. DOI: https://doi.org/10.1109/4235.797969

Zitzler, E., Laumanns, M., Thiele, L., 2001. Spea 2: improving the strength pareto evolutionary algorithm for multi-objective optimization. In: Giannakoglou, K., Tsahalis, D., Périaux, J., Papailiou, K., Fogarty, T. (eds.) Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, pp. 95–100. CIMNE, Athens. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.22.4617&rep=rep1&type=pdf