UNIVERSITY OF
BIRMINGHAM

DESIGN OPTIMISATION AND REAL-TIME ENERGY MANAGEMENT CONTROL OF THE ELECTRIFIED OFF-HIGHWAY VEHICLE WITH ARTIFICIAL INTELLIGENCE

BY
QUAN ZHOU

A THESIS SUBMITTED TO
THE COLLEGE OF ENGINEERING AND PHYSICAL SCIENCE OF
THE UNIVERSITY OF BIRMINGHAM

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering
School of Engineering
College of Engineering and Physical Science of
The University of Birmingham

May 2019

ABSTRACT

Targeting zeros-emissions in transportation, future vehicles will be more energy-efficient via powertrain electrification. This PhD research aims to optimise an electrified off-highway vehicle to achieve the maximum energy efficiency by exploring new artificial intelligence algorithms. The modelling study of the vehicle system is firstly performed. Offline design optimisation and online optimum energy management control methodologies have been researched. New optimisation methods are proposed and compared with the benchmark methods. Hardware-in-the-Loop testing of the energy management controller has been carried out for validation of the control methods. This research delivers three original contributions:

1) Chaos-enhance accelerated particle swarm optimisation algorithm for offline design optimisation is proposed for the first time. This can achieve 200% higher reputation-index value compared to the particle swarm optimisation method.

2) Online swarm intelligent programming is developed as a new online optimisation method for model-based predictive control of the vehicle energy-flow. This method can save up to 17% energy over the rule-based strategy.

3) Multi-step reinforcement learning is researched for a new concept of ‘model-free’ predictive energy management with the capability of continuously online optimisation in real-world driving. It can further save at least 9% energy.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere acknowledgement to my supervisor Prof H. Xu for his guidance and help in my academic research, career development and daily life. I appreciate all his contributions of time, ideas, funding, and networking that enables me to achieve the research targets. I also want to thank my associate supervisor Dr O. Olatunbosun. I am grateful to the help from Prof H. Williams who has greatly inspired me on optimisation method development. I would also like to thank Prof A. Ghafourian, for his suggestions through my annual progress reviews. Sincerely acknowledgement to my PhD examiners, Prof J. Wang (Warwick) and Dr M. Castellani, for their valuable suggestions on thesis correction and ideas for my future research.

I gratefully acknowledge the university's PhD scholarship and the Innovate UK funding (No.102253) which financed my PhD research. I would like to express my acknowledgement to the project partners: Textron Equipment UK and Hyperdrive Innovation. I really appreciate their contributions on equipment, finance, and engineers' time. Sincerely acknowledgement should also be express to Prof F. Eastham (Bath), who was the coordinator of the project. I would also like to thank Dr Y. Wang (Ford), Dr J. Qiao (CHANGAN), Dr B. Wang (SAIC), Dr C. Lv (NTU), and Dr Z. Zhao (Loughborough) who had provided me with lots of valuable suggestions.

I am grateful for the research fellows who have provided guidance in my day-by-day research; they are Dr W. Zhang, Dr G. Lu, and Dr H. Ma. Great thanks go to the technicians and engineers who have supported me on experiment setting-up, they are P. Thornton, C. Hingley, L. Gauntlett, and F. Knoll (ETAS). This thesis was partly proofread by Janet's Proofreading Service for the conventions of grammar, language and spelling. I do appreciate the help from Mrs J. Hingley.

I am thankful to my friends in the UK: Dr C. Wang, Dr Y. Zhang, B. Han, Dr S. Cash, Dr S. Rezaei, Dr H. Liu, Dr A. Sahu, Z. Li, A. Dharmadhikari, J. Li, L. Parry, Y. He, B. Shuai, Z. Zhang et al.

I would like to give my most special thanks to my fiancée, Miss Xiaomeng Shen, for her understanding and encouragement. I do appreciate her one-year accompany in the UK and spiritual support at most of the time.

Finally, I would like to express my sincere gratitude to my parents and family members from the bottom of my heart for their unconditional support. I express my sincere gratitude to my Uncle Dr Zhige Zhang, Aunt Dr Wenbing Che, and Cousin Yike Zhang for their help and support in the UK.

Quan Zhou

May 2019

TABLE OF CONTENT

LIST OF PUBLICATIONS

Journal Papers (10)

1. ***Quan Zhou***, Wei Zhang, Scott Cash, Oluremi Olatunbosun, Hongming Xu*, Guoxiang Lu. “Intelligent sizing of a series hybrid electric power-train system based on Chaos-enhanced accelerated particle swarm optimisation”, *Applied Energy*, Vol. 189 (1), 2017, Pages 588-601

2. ***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*. Vol. 14 (9), 2018, Pages 4149-4158

3. ***Quan Zhou,*** Ji Li, Bin Shuai, Yinglong He, Ziyang Li, Yunfan Zhang, Hongming Xu*, “Multi-step reinforcement learning for model-free predictive energy management of electrified off-highway vehicles”, *Applied Energy*, Vol. 255 (1), 2019. https://doi.org/10.1016/j.apenergy.2019.113755

4. 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 optimisation Algorithm”, *Proceedings of the*

Institution of Mechanical Engineers, Part D: Journal of Automotive Engineering. Vol. 233(7), 2019, Pages 1698-1711.

5. Scott Cash, **Quan Zhou**, Oluremi Olatunbosun, Hongming Xu*, Sean Davis, Robin Shaw. “A new traction motor sizing strategy for an HEV/EV based on an overcurrent-tolerant prediction model”, IET Intelligent Transport Systems. Vol. 13(1):168, 2019.

6. Scott Cash, ***Quan Zhou***, Oluremi Olatunbosun, Hongming Xu, Sean Davis, Robin Shaw, “Development of a series hybrid electric aircraft pushback vehicle: a case study”, Engineering, Vol. 2019 (11), pages:33-47

7. Ji Li, ***Quan Zhou***, Ziyang Li, Yunfan Zhang, Yinglong He, Bin Shuai, Hongming Xu*, “Dual-loop online intelligent programming for driver-oriented predict energy management of plug-in hybrid electric vehicles ”, *Applied Energy*, Vol. 253 (1), Nov 2019.

8. 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*. https://doi.org/10.1177/0954407019862079

9. Ziyang Li, Ji Li, ***Quan Zhou***, Yunfan Zhang, and Hongming Xu*, “Intelligent air/fuel ratio control strategy with a PI-like fuzzy knowledge-based controller for GDI engines”, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering.* Vol. 233 (8), 2019, Pages 2161-2173.

10. Ji Li, Ziyang Li, ***Quan Zhou***, Yunfan Zhang, Hongming Xu, “An improved scheme of membership function optimisation for fuzzy air-fuel ratio control of GDI engines”, IET Intelligent Transport Systems. DOI: 10.1049/iet-its.2018.5013

Patent (1)

11. ***Quan Zhou***, and Hongming Xu*, “Vehicle Power Management System and Method”, PCT/GB2019/051729.

Conference Papers and Presentation (3)

12. ***Quan Zhou***, Scott Cash, Bin Shuai, Yinglong He, Ji Li, Hongming Xu*, “Model-based Swarm Intelligent Optimisation for Sizing and Energy Management of Hybrid Electric Vehicles”, *2018 Powertrain Modelling Conference*, Loughborough

13. Scott Cash, ***Quan Zhou***, Hongming Xu* and Oluremi Olatunbosun, “A robust fuzzy logic field-oriented control scheme for hybrid and electric vehicles”, *2018 Powertrain Modelling Conference*, Loughborough

14. Ziyang Li, ***Quan Zhou***, Yinglong He, Ji Li, Hongming Xu, “Artificial Intelligence Based Calibration and Control for Future Engines”, *UnICEG September Meeting 2018*, Birmingham

LIST OF ABBREVIATIONS

AC	Alternating Current
AI	Artificial Intelligence
ANN	Artificial Neuronal Network
APC	Advanced Propulsion Centre
APSO	Accelerated Particle Swarm Optimisation
BP	Battery Package
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAN	Control Area Network
CAPSO	Chaotic-enhanced Accelerated Particle Swarm Optimisation
CO2	Carbon Dioxide
DC	Direct Current
DoE	Design of Experiments
DP	Dynamic Programming
DQN	Deep Q Networks
EA	Evolutionary Algorithm
EGU	Engine Generator Unit
EM	Electric Motor

EMS	Energy Management System
EU	European Union
EV	Electric Vehicle
FPGA	Field-Programmable Gate Array
FTP	Federal Test Procedure
GA	Genetic Algorithm
GB	Great Britain
GDI	Gasoline Direct Injection
GHG	Green House Gas
GM	General Motors
HCCI	Homogeneous Charge Compression Ignition
HEV	Hybrid Electric Vehicle
HiL	Hardware-in-the-Loop
HWFET	Highway Fuel Economy Test
ICE	Internal Combustion Engine
IoV	Internet of Vehicles
LIN	Local Interconnect Network
LSTM	Long-term-Short-Term-Memory
MCTS	Mont-Carlo Tree Search
MPC	Model-based Predictive Control
NEDC	New European Driving Cycle

NMPC	Nonlinear Model-based Predictive Control
NSGA	Non-dominated Sorting Genetic Algorithm
OEM	Original Equipment Manufacturer
OSIP	Online Swarm Intelligent Programming
PBDC	Push Back Driving Cycle
PC	Personal Computer
PCIe	Peripheral Component Interconnect express
PGS	Planetary Gear Set
PHEV	Plug-in Hybrid Electric Vehicle
PID	Proportional–Integral–Derivative (control)
PMS	Power Management Strategy
PMSM	Permanent Magnet Synchronous Motor
PSO	Particle Swarm Optimisation
PWM	Pulse Width Modulation
R&D	Research and Development
RAM	Random-Access Memory
RDE	Real-world Driving Emissions
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SARSA	State-action-reward-sate-action (algorithm)
SoC	State of the Charge

SPEA	Strength Pareto Evolutionary Algorithms
SQP	Sequential Quadratic Programming
V2I	Vehicle to Infrastructures (network)
V2V	Vehicle to Vehicle (network)
V2X	Vehicle to Everything (network)
WLTC	Worldwide Harmonised Light Vehicle Test Cycle
WLTP	Worldwide Harmonised Light Vehicle Test Procedure
XiL	Supper-hardware-in-the-Loop
RSU	Road Side Unit
MDP	Markov Decision Process
A2N	Average-to-Neighbour
S2T	Sum-to-Terminal
R2T	Recurrent-to-Terminal

LIST OF SYMBOLS

$\boldsymbol{A_h}$	Action history set
C_d	Aerodynamic drag coefficient
β	Attraction factor
I_{bp}	Battery cell current
c_{ems}	The coefficient for the power distribution function
u_{batt}	The control signal of the battery package
L_d	d-axis inductance (mH)
τ_d	Decay rate subjective to system inertia
ρ_f	The density of the fuel
L_{ice}	Displacement of the engine generator
A_{veh_f}	Effective front area
$\dot{v}_f$	Fuel consumption rate
g_{bst}	Global best position
α	Gradient angle

H_f	Heat value of the fuel
h_{cell}	Height of the battery cell
$\boldsymbol{D}$	History sets of state, action and reward
R_{loss}	The internal resistance of the battery
$\boldsymbol{Y}$	The local best cost-function value set
$\boldsymbol{X}$	The local best variable set
$\bar{L_{ice}}$	Lower boundary of L_{ice}
$\bar{c_{ems}}$	Lower boundary of c_{ems}
$\bar{n_{bc}}$	Lower boundary of n_{bc}
m_{veh}	Mass of the aircraft-towing tractor
$\boldsymbol{Q}$	Matrix of Q value
P_{batt_max}	Maximum power of the battery package
P_{egu_max}	Maximum power of the engine generator
Φ	Multi-step learning algorithm
$Num_{opt.}$	Number of achieving the global best result
n_{bc}	Number of battery cells for the battery package

n_{po}	Number of poles in an electric motor
J_{size}	The objective function for component size
J_{loss}	The objective function for energy loss
V_{oc}	Open circuit voltage of battery cell
P_{bp}	Output power provided by the battery package
P_{egu}	Output power provided by the engine-generator unit
λ_{pm}	Permanent magnet flux linkage (Wb)
$x_{i,j}$	Position of the j^{th} particle at i^{th} iteration
η_{f2e}	The power conversion efficiency of fuel-to-electricity
$Loss_{batt}$	The power loss of the battery package
$Loss_{eng}$	The power loss of the engine generator unit
$P_{loss,tm}$	The power loss of the traction motor
P_{link}	Power requirement from the DC link
P_{batt}	Power supplied by the battery package
P_{egu}	Power supplied by the engine generator
L_q	q-axis inductance (mH)

r_{cell}	The radius of the battery cell
I_{tm_s}	Rated current of traction motor (A)
P_{tm_max}	Rated power of traction motor (kW)
n_{tm_max}	Rated speed of traction motor (rpm)
R_{tm_s}	Rated stator resistance of motor (omega)
V_{tm_s}	Rated voltage of traction motor (V)
SoC_{ref}	Reference SoC
R_i	Reputation index
P_{req}	Required power from the power bus
$\boldsymbol{R_h}$	Reward history set
I_{tm_m}	Rotor inertia of traction motor (kg*m^2)
τ_s	Sampling time
λ	The scale of the reward value
$\boldsymbol{S_h}$	State history set
ρ_{air}	The density of the air
f_{tm_f}	The friction coefficient of the traction motor

f_{veh_f}	The friction coefficient of the vehicle
r_{whl}	The radius of the wheels
Num_{trial}	Total number of the trails in the random repeat test
s_{bp}	The transient state of the battery package
s_{egu}	The transient state of the engine generator
L_{ice}^{+}	Upper boundary of L_{ice}
c_{ems}^{+}	Upper boundary of c_{ems}
n_{bc}^{+}	Upper boundary of n_{bc}
v_{dem}	Vehicle speed demand
vol_{bc}	The volume of a battery cell

LIST OF TABLES

LIST OF FIGURES

Chapter One

INTRODUCTION

Future vehicles are moving towards electrification, connectivity and autonomy. The performance of vehicles will be improved with less fuel consumption and emission through design optimisation in the R&D process and real-time optimum control in real-world driving. This thesis researches on design optimisation and optimal energy management for hybrid electric vehicles by exploring new artificial intelligence algorithms. This chapter provides an introduction of this thesis, including the background, motivations, objectives, and outline.

1.1 Background

1.1.1 Thrusts for vehicle electrification

Numbers of new emission legislations have been launched for reducing CO2 emission in the transportation sector. International energy agency (IEA) announced that the transportation sector produced 7498.6 Million tons of CO2 (with 5695.7 Mt produced by road transportation) accounting 24% of global CO2 emission in 2015 as in Fig. 1-1 (International Energy Agency, 2017). Targeting further CO2 emission

reduction, increasing stringent regulations with new testing procedures will be launched worldwide (Continental Automotive GmbH, 2017). A summary of historical CO2 emissions and future CO2 limitation targets for different world regions is shown in Fig. 1-2. European Commission has proposed its 2020/2021 CO2 emission target based on the NEDC test procedure which was 95 gCO2/km for passenger cars and 147 gCO2/km for light commercial vehicles (European Commission, 2014). They have also proposed to accelerate the uptake of zero- and low-emission vehicles and further 30% CO2 will be reduced by 2030 (European Commission, 2017).

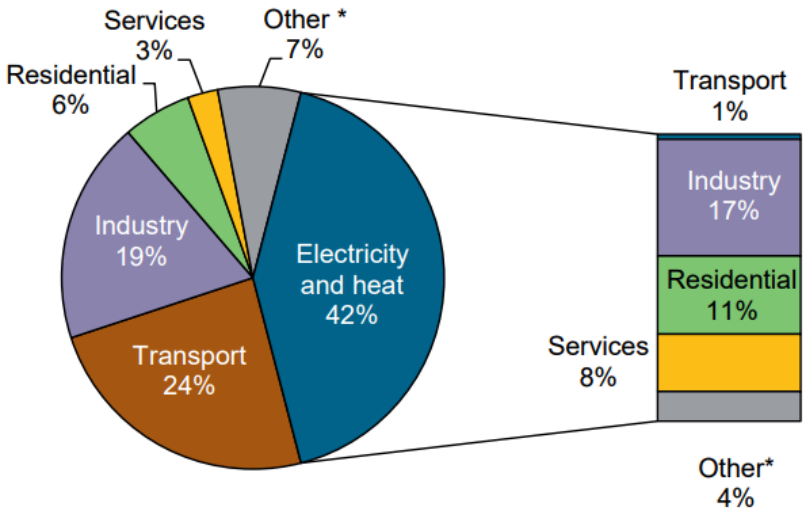


Fig. 1-1 World CO2 emission from fuel combustion by sector (International Energy Agency, 2017)

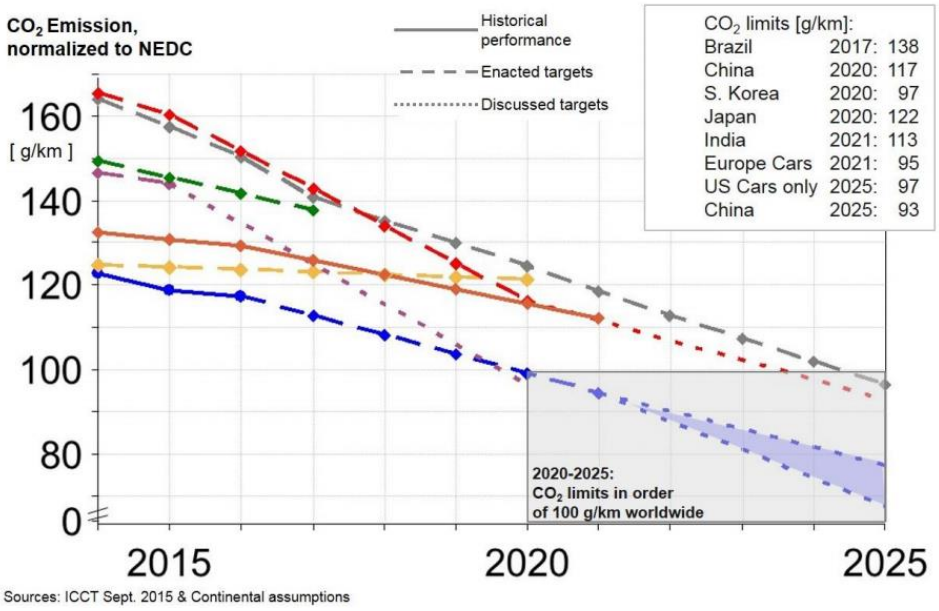


Fig. 1-2 Historic CO2 emissions and target for different world regions

(Continental Automotive GmbH, 2017)

The growing importance of air quality has forced automotive industries (including OEMs and Tier 1 suppliers of passenger cars, bus, commercial vehicle, and off-highway vehicles) considering carbon reduction solutions with low cost. Electrification, which takes full advantages of electric power and electric drive, is the most significate solution towards energy saving and emission reduction. The Passenger Car Products Roadmap from the UK Advanced Propulsion Centre 2017 predicts that “thermal propulsion systems will transit from the sole propulsion device to being part of a hybrid system” and “vehicle connectivity and autonomy on powertrains design will merge” (Advanced Propulsion Centre, 2017). During the past decades, the research of electrified passenger cars has delivered many successful mass-produced vehicle products, e.g. Toyota Prius, Tesla, Nissan Leaf, Jaguar I-

PACE, etc. Electrified cars including mild hybrid, full hybrid, plug-in hybrid, fuel cell hybrid and pure electric vehicles have been widely adopted by the market.

Legislations also focus on heavy-duty vehicles, which have contributed 27% of road transport CO2 emissions and 5% of EU greenhouse gas (GHG) emissions in 2016 (Gregor, 2019). The proposal from EU commission has announced that the average CO2 emission from new heavy-duty vehicles in 2025 should be 15% lower than in 2019, and a further 30% reduction would be achieved in 2030. This promotes the electrification of commercial and off-highway vehicles, especially for logistics vehicles and special utility vehicles working in urban areas.

Many countries around world have proposed their plans to ban petrol and diesel vehicles (e.g. the UK is committed to banning all new petrol and diesel cars/vans from 2040), whereas hybrid electric vehicles (HEVs) are still expected to comprise 45-65% of light-duty vehicles worldwide by 2050 (*Global EV Outlook 2018*, 2018). Hybrid vehicle enjoys longer driving distance and lower life cycle CO2 emission compared to the pure electric vehicle, which is far more complex than conventional vehicles and pure electric vehicles. It incorporates lots of optimisation processes in design and energy management control (Enang and Bannister, 2017a). The design of the hybrid system will consider how the design parameters (including component size and control parameters) affect the system performance (e.g. vehicle power, fuel economy, system cost, size, etc.) and what is the best combination of parameters to

achieve the optimal system performance. Energy management system (EMS) is a complex system for the supervisory control of the hybrid powertrain, which is developed to determine the optimal distribution of energy flow in an HEV to satisfy the driver's demand and achieve maximum energy efficiency (Wang *et al.*, 2017).

1.1.2 Current development of electrification of vehicles

Hybridization is a mainstream technology for vehicle electrification to achieve energy saving and emission reduction in the automotive industry. A hybrid electric vehicle usually equips with at least two power sources, e.g. a combination of internal combustion engine and electric motors.

Hybrid electric vehicles can be classified into four categories according to their degree of hybridization, i.e. micro-hybrid, mild hybrid, full hybrid and plug-in hybrid (Chan, 2007). Taking passenger cars as an example, the electric power for the **micro-hybrid vehicle** is about 2.5kW at 12V, which can contribute to 5%~10% energy saving comparing with conventional engine-driven cars. **The mild-hybrid vehicle** is usually equipped with a 10-20kW motor at 48~200V level which can help to save 20%~30% energy with 20%~30% cost increase. The 48V mild-hybrid is the most commonly used technology for most OEMs (ReportLinker, 2016). **The full hybrid** car is powered by a motor around 50kW with 200-300V power supplier, which has the potential to save 30%~50% energy while the cost would increase by about

30%-40%. **Plug-in hybrid electric vehicle** (PHEV) is mainly refuelled by the power grid, which is designed with all-electric ranges of 30-60km using lithium-ion batteries. The energy efficiency of PHEV can be very high, for example, the BMW X1 plug-in edition (a compact size SUV with 37-mile electric range) can achieve a fuel consumption rate of 1.8L/100km in a combined driving cycle (Padeanu, 2016).

It is also commonly accepted that the hybrid vehicle can be classified into three mainstreams, including series, parallel and power-split (series-parallel).

a) Series hybrid electric vehicles

The series hybrid vehicle has the longest history. Porsche built the first hybrid car using an internal combustion engine to spin a generator that provided power to electric motors located in wheel hubs in the 1890s (Chan, 2007). Many modern auto companies also have their series hybrid products, e.g. Mitsubishi, Volvo and BMW (Enang and Bannister, 2017b).

As shown in Fig. 1-3, the series hybrid is purely driven by the electric motor, and there is no mechanical link between the internal combustion engine and driving wheels. The engine is used to power a generator to transfer fossil energy into electric power to charge the battery or power the electric motor(s). An inverter (or power converter) is used to control the direction of power flow. Normally, there are four working modes for series hybrid vehicle: 1) battery drive; 2) charging only; 3) engine

boosting; and 4) regenerative brake. For the design of a series hybrid powertrain, the maximum engine power can be less than the average power requirement of the vehicle. In the control aspect, the energy management system will ensure the series hybrid vehicle working in battery drive mode at most of the operation hours. Fuel consumption and exhaust emissions would be minimised by making the engine working at its most efficient region (Enang and Bannister, 2017b). The drawbacks of series configuration are mainly concerned on its total system efficiency and the increasing of cost led by a large battery and electric driving system (Hu, Ni and Peng, 2018).

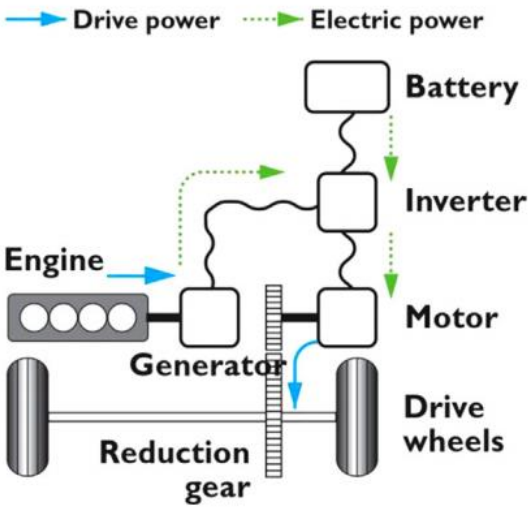


Fig. 1-3 Configuration of series hybrid vehicle (Hu, Ni and Peng, 2018)

b) Parallel hybrid electric vehicles

The first parallel hybrid electric vehicle was designed by H. Piper in 1905, which uses a small electric motor to provide boosting power to its internal combustion engine in

the low-speed zone (Chan, 2007). Both internal combustion engine and electric motor are physically connected to the driving wheels as shown in Fig. 1-4, so that the engine and the motor can work independently or simultaneously to propel the vehicle. This configuration is usually used for micro and mild hybrid vehicles which allow a relevant larger internal combustion engine compared to the series architectures. The parallel configuration is widely used for hybridisation of conventional powertrains with an internal combustion engine.

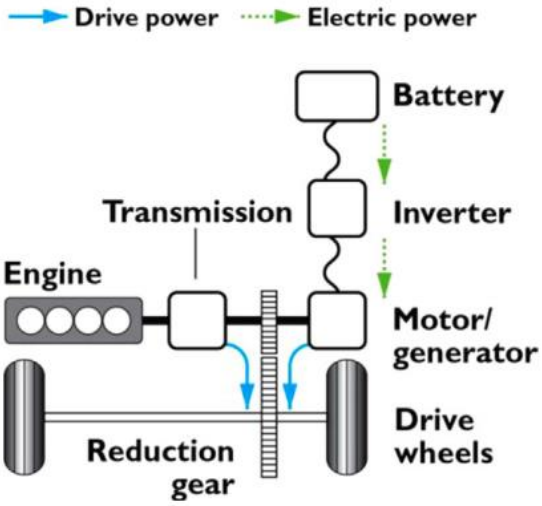


Fig. 1-4 Configuration of parallel hybrid vehicle (Hu, Ni and Peng, 2018)

Based on the position of the electric motor, there are five main powertrain architectures for parallel hybrid vehicles (i.e. P0, P1, P2, P3, and P4) as shown in Fig. 1-5. In P0 architecture (also named belt starter generator architecture), a 48V motor/generator would replace the starter (normally a 12V motor) and connect with the internal combustion engine with a belt. The 48V P0 structure is a mainstream

technology adopted by automotive manufacturers because it combines a relatively low integration cost with considerable benefits in terms of CO2 emissions reduction and dynamic performance boost (X-engineer, 2019). The P1 architecture mounts a motor at the end of the engine crankshaft (before clutch), and it can provide more torque than P0 architecture. P2, P3 and P4 architectures have better energy flow efficiency because the electric motor is integrated with the transmission to allowed higher transmission efficiency. The motor of P2 architecture is allocated before the transmission (after the clutch). P3 architecture arranges its motor at the output shaft of the gearbox (before final reducer) and the motor of P4 architecture is located at the output shaft of the final reducer.

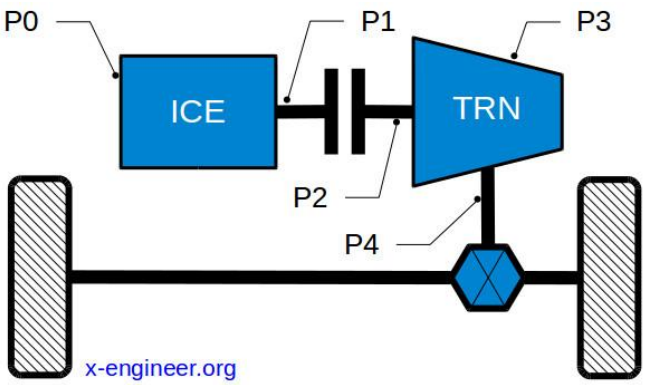


Fig. 1-5 Different architecture for a parallel hybrid vehicle (X-engineer, 2019)

c) Series-parallel hybrid electric vehicles

The series-parallel hybrid architecture was firstly introduced by Toyota Prius in 1997, as a newborn of the hybrid vehicle before a fad-away since the 1920s (Chan, 2007). As shown in Fig. 1-6, the series-parallel architecture contains an internal combustion

engine, a generator, an electric motor, a battery and a power convertor. The engine, generator and electric motor are physically connected with a power split device (normally a planetary gear set) so that this architecture can work in both series mode and parallel mode. The series-parallel configuration enjoys the advantage of having the engine decoupled from the vehicle, thus making it possible for the vehicle to be powered using the electric motors only (Liu, Peng and Filipi, 2005). Series-parallel architecture requires the internal combustion engine and the electric motors to be bonded together by the power-split device, which limits the flexibility of layout. Compared to parallel hybrid architectures, this architecture is generally more expensive; torque constraints can limit towing capacity and acceleration (Wu, Zhang and Dong, 2015).

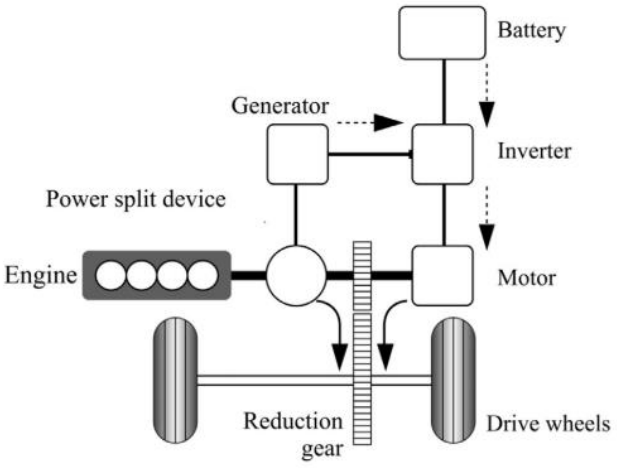


Fig. 1-6 Configuration of series-parallel hybrid vehicle

There are three fundamental topologies of series-parallel hybrid vehicles, including input-split, output-split, and compound-split, as shown in Fig. 1-7. More complicated

power-split can be generated from the three fundamental architectures (Wu, Zhang and Dong, 2015). In all topologies, the internal combustion engine (ICE) connects with a generator via a planetary gear set (PGS). For input-split architecture, as shown in Fig. 1-7 (a), the output shaft of the planetary gear-set and the electric motor (EM) are connected at the input shaft of the differential. Output-split architecture requires ICE, one EM and output shaft are connected to three ports of PGS and the second EM is linked to ICE fixedly (Miller, 2006), as shown in Fig. 1-7 (b). Compound-split architecture, as shown in Fig. 1-7 (c), is more complex since the compound power-split device contains two interconnected planetary gear sets, which are bonded by two compound branches inside the power-split device.

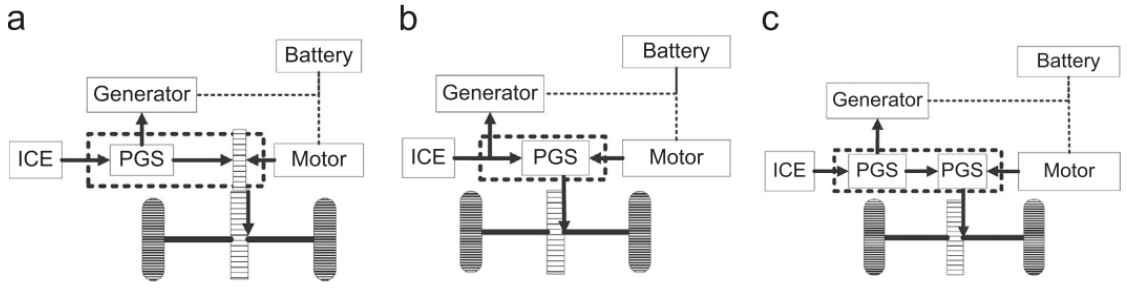


Fig. 1-7 Fundamental series-parallel architects: (a) input-split; (b) output-split; and (c) compound-split (Wu, Zhang and Dong, 2015)

d) Axle-split hybrid electric vehicles

Axle-split hybrid architecture allocates the engine/generator and the traction motor separately in different axles. In 2008, Ohio State University developed an axle-split

hybrid car for Challenge-X competition sponsored by General Motors and the US Department of Energy (Koprubasi, 2008). University of Victoria (Canada) has also developed this kind of architecture for Eco car 2 competition (Wu, Zhang and Dong, 2015). The most famous axle-split hybrid electric vehicle is the BMW i8. Normally, the front axle applies a parallel hybrid topology, where the front wheels are connected physically with the engine and the motor/generator. The rear axle is driven purely by an electric motor, and there is no mechanical connection between the front axle and the rear axle, as in Fig. 1-8. The axle-split architecture is a variant of series-parallel architecture without the planetary gear set so that the total cost can be reduced. Without the auto torque compiling function enabled by the planetary gear set, the axle-split architecture requires advanced control algorithm to allocate power for the front and rear axles.

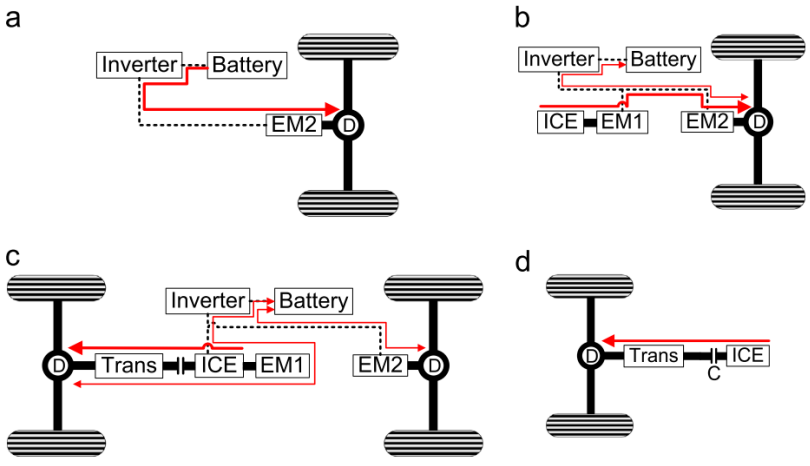


Fig. 1-8 The operation model of axle split hybrid vehicle: (a) pure electric drive mode, (b) series mode, (c) parallel mode, and (d) engine driving mode (Wu, Zhang and Dong, 2015)

The axle-split hybrid vehicle can work in four different modes as shown in Fig. 1-8, including pure electric drive, series mode, parallel mode, and engine driving mode. The vehicle is purely driven by the rear axle in a purely electric drive model, and it is front axle driven by the internal combustion engine in engine driving mode. In series mode and parallel mode, the energy flow can be controlled as a series hybrid or parallel hybrid architecture.

1.1.3 The hybrid powertrain for heavy-duty applications

The electrification of heavy-duty vehicles, including buses, trucks, tractors, special utility vehicles and military vehicles, have also been researched for many years. Most of the above hybrid architectures have been applied in heavy-duty vehicles. Series and parallel architectures are usually used for hybrid buses and light trucks (Wang, Zeng and Wang, 2003; Chomchai, Sonjaipanich and Cheewaisrakul, 2011; Liu *et al.*, 2017), and the series architecture is very commonly used for heavy trucks and tractors (Okui and Kobayashi, 2015; Zou *et al.*, 2016). Power-split architecture is rare to be found in heavy-duty application because it is hard to develop a cost-efficient and reliable power-split device for heavy-duty application. Series topology is adopted by most powertrains of heavy-duty vehicles because it allows more flexible allocation of traction motors and it takes full advantage of electric drive in high-torque and low-speed domains. For example, the researchers from the Beijing

Institute of Technology have developed a tracked vehicle using series architecture, as shown in Fig. 1-9, which is equipped a diesel engine-generator with a maximum power of 300kW and a 50Ah 470V battery package to power two 150 kW traction motors for each track (Liu *et al.*, 2015b).

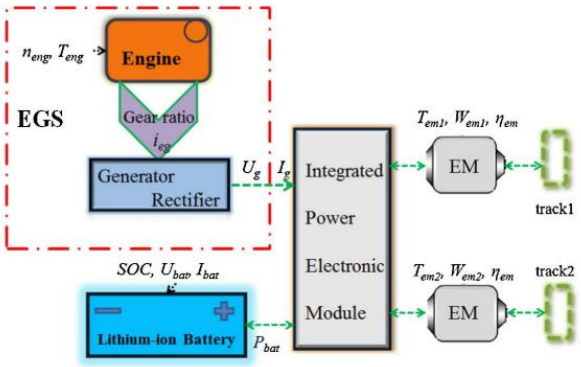


Fig. 1-9 The architecture of a hybrid tracked vehicle (Liu *et al.*, 2015b)

For special utility vehicles, the power for the operation parts (e.g. the swing system of an excavator) would also require large power while vehicle in operation. Komatsu has developed a series of hybrid crawler excavator using series hybrid architecture, which includes a deiseal engine-generator, electric swing motor, power inverter and a capacitor package. The swing motor can capture and regenerate electric power from the slow-down of the upper part of the excavator. The regenerated energy is stored within the capacitor and will help to swing or help the engine to accelerate. The system with a 202kW engine-generator has the capability to provide power for up to 37tone operating weight and save up to 30% energy comparing with the conventional hydraulic swing system (*Komatsu Hybrid Technology*, 2019).

Fig. 1-10 Hybrid swing system of Komatsu's excavator (*Komatsu Hybrid Technology*, 2019)

1.2 Motivations

This PhD research is partially supported by an Innovate UK project (No.102253) which aims to develop a brand-new product of hybrid electric aircraft-towing tractor. This is a highly industrial-oriented project which motivates the innovation and research in new artificial intelligence methodologies for hybrid vehicle design and energy management control. This research is based on the following four observations:

1) Legislations for electrified off-highway vehicles have forced the OEMs seeking for new design methodology for hybridization and electrification of special utility vehicles.

2) Challenge of modern design process involving multiple variables and multiple objectives requires reliable and rapid offline optimisation technology to

replace the complex iteration of trial-and-error.

3) Advanced model-based predictive energy management control is still under development which is seeking for effective real-time nonlinear model-based predictive control.

4) Vehicle system would have the capability of continues self-optimisation to fulfil the requirement of future RDE (real-world driving emissions) test procedure which involves more uncertainties that are hard to be modelled in the R&D process.

1.3 Objectives

This PhD research aims to develop an artificial intelligence methodology for hybrid powertrain design optimisation and real-time energy management control, which will help the OEMs to develop their new vehicle prototype meeting design targets and achieving maximum energy efficiency in daily operations. In addition, this research will also demonstrate how artificial intelligence assists the electrification of off-highway vehicles and improve the competitively of vehicle products. The specific objectives are:

1) To build a fully real-time system model of the electrified aircraft-towing tractor with scalable powertrain components, for design optimisation and real-time

energy management control.

2) To develop a particle swarm intelligence algorithm for off-line multiple objective design optimisation with multiple variables, including variables for sizing and rule-based energy management control.

3) To develop online optimisation method for real-time optimal energy management control with a nonlinear predictive model of vehicle energy-flow.

4) To develop life-long continuous vehicle performance optimisation method to maximise the vehicle efficiency in real-world operations based on reinforcement learning.

5) To evaluate the vehicle performance in each stage and demonstrate the advancement of the proposed artificial intelligent methodology for design optimisation and energy management control of the hybrid vehicle system.

1.4 Thesis Outline

The rest of this thesis is organised into six chapters, including one for literature review, one for methodology introduction, three main chapters presenting the original contributions, and the last one for the conclusions.

Chapter Two reviews current state-of-the-art technologies relevant to modelling

method for vehicle electrification research, offline and online optimisation methods for hybrid powertrain developments. This chapter also highlights the future trend for modelling method, offline design optimisation of hybrid vehicle and its real-time energy management control.

Chapter Three introduces the methodology and facilities used for this study. Firstly, the vehicle development process is divided into four main tasks, including modelling, Hardware-in-the-Loop (HiL) testing, design optimisation and optimal energy management control. The methodologies for research on design optimisation and optimal energy management control are illustrated, followed by the introduction of the driving cycles for aircraft-towing tractor development. The HiL testing system and research facilities are introduced in this chapter, followed by the development of real-time models for HiL testing.

Chapter Four carries out the research into offline design optimisation. A new design optimisation method using chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm is proposed. The outcomes of this chapter have been published with *Applied Energy* (Zhou *et al.*, 2017). The design optimisation (including component sizing and control parameter calibration) is first formulated as a multi-objective categorical optimisation problem by mathematical modelling. The optimisation is firstly carried out using conventional accelerated particle swarm optimisation (APSO) algorithm, and then a new CAPSO algorithm is proposed to

improve the optimisation performance. The design optimisation result is secured via a two-stage optimisation based on CAPSO algorithm.

Chapter Five studies the model-based predictive control method for real-time energy management. This chapter proposes a new cyber-physical control for energy management using online swarm intelligent programming (OSIP), which has been published with *IEEE Transactions on Industrial Informatics* (Zhou *et al.*, 2018). The optimisation problem for energy management control is firstly formulated, followed by control-oriented modelling for model-based predictive control. The OSIP for solving the online optimisation problem is developed based on the CAPSO algorithm proposed in Chapter Five. The advantages of model-based predictive energy management control are validated by HiL testing.

Chapter Six researches a new concept of 'model-free' predictive energy management method with multi-step reinforcement learning to maximise vehicle efficiency in daily operations. The original contribution of this chapter has filed a UK patent (Zhou and Xu, 2018), and a research paper has been published with *Applied Energy* (Zhou *et al.*, 2019). Firstly, the energy management of the vehicle is formulated as a Markov decision problem. The model-free predictive energy management with three algorithms for multi-step reinforcement learning is then studied through the investigation of their learning performance in different optimisation scenarios. The real-time control feasibility and performance of the

proposed energy management method are examined and evaluated by hardware-in-the-loop testing.

Chapter Seven summaries the outcomes and impacts of the research conducted in this thesis. A recommendation for future work is also given in this chapter following a critical analysis of this research.

Chapter Two

LITERATURE REVIEW

Modern vehicle engineering incorporates many optimisation problems. Model-based development, which uses mathematical models to predict the vehicle's performance with different design and control parameters for optimisation, is widely used for offline and online optimisation of vehicle systems. This chapter presents a comprehensive literature review about 1) modelling method for research of electrified vehicles; 2) offline optimisation methods for hybrid electric vehicles; and 3) real-time control method for energy management of hybrid electric vehicles. The literature study provides the foundation of the development of the methods used in this thesis.

2.1 Modelling Methods for Hybrid Vehicle Research

Modelling is necessary for the research and development of hybrid vehicles because accurate vehicle system models can help evaluate the performances of the vehicles with the minimised cost. It allows simulation of a vehicle's performance (e.g. fuel economy, cost, emission) with different design parameters and control strategies before a physical prototype is built. This section introduces four mainstream modelling methods for hybrid vehicles: the kinematic approach, the quasi-static

approach, the dynamic modelling approach, and the semi-hardware-in-the-loop modelling approach.

2.1.1 The kinematic approach

The kinematic modelling approach, as shown in Fig. 2-1, is a backward calculating method, which firstly determines the wheel speed and wheel torque from the velocity and acceleration of the given driving cycles. Using the speed and torque of the wheel, a powertrain model is used to calculates the energy consumption (including fuel consumption and battery usage) and engine emission through a series of look-up-tables (Genta, 1997). A well-known hybrid vehicle simulation software ADVISOR is developed based on the kinematic modelling method (Gao, Mi and Emadi, 2007).

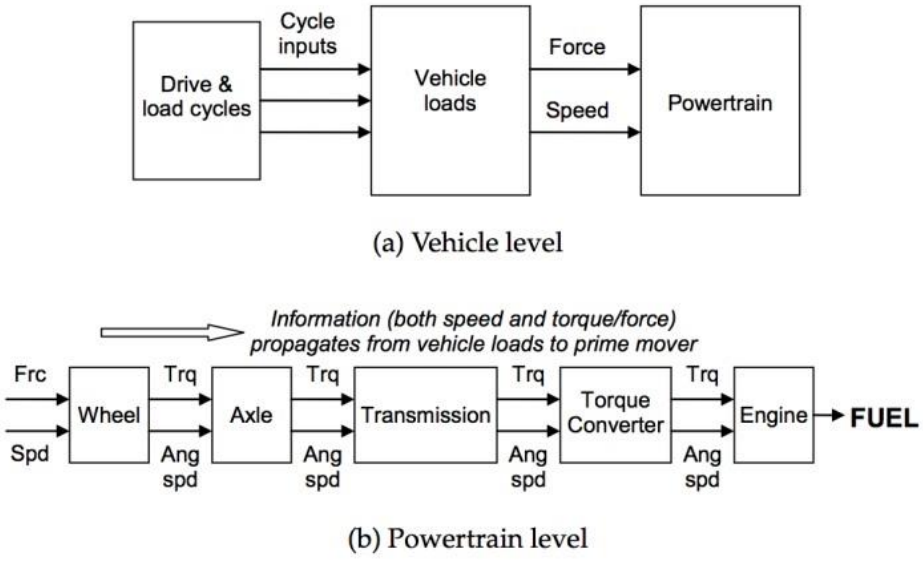


Fig. 2-1 kinematic model of HEV: (a) vehicle level, and (b) powertrain level (Guzzella and Sciarretta, 2007)

For hybrid electric vehicles, an energy management system would be modelled before the transmission model in Fig. 2-1 (b), which is used to allocate the required torque and speed to relevant engine and electric motors. Fuel consumption and exhaust emission of the internal combustion engine will be obtained through a 2D look-up-table (BMEP and engine speed as inputs, fuel rate/emission rate as output) as shown in Fig. 2-2. Electric motors will use another 2D look-up table to determine its electric-to-mechanical efficiency at current speed and torque point so that the total electric energy consumption can be obtained. The electric energy will then be transferred into voltage and current so that the battery state-of-the-charge can be obtained via the battery model. The energy management system model would also be used to determine when to use the engine to charge the battery.

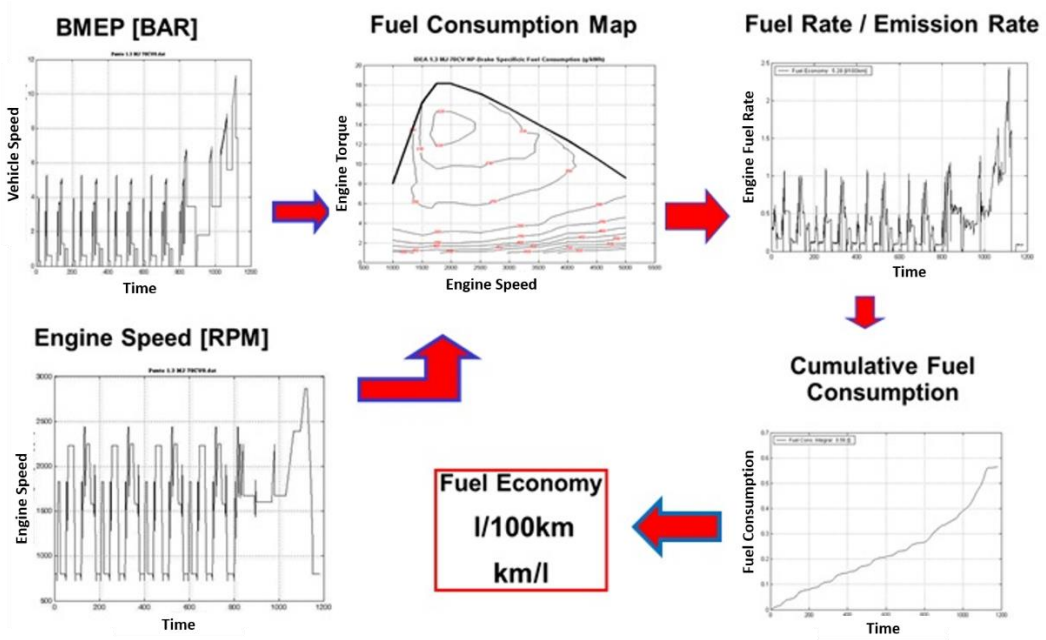


Fig. 2-2 Engine fuel consumption and emission simulation using 2D Look-up tables in a kinematic model (Millo, Rolando and Andreata, 2011)

The kinematic approach assumes that driving speed profile will be exactly followed and the vehicle will meet the dynamic target performance so that the vehicle speed should be determined in advance; thus, this method enjoys the advantage of simplicity and low computational cost (Millo, Rolando and Andreata, 2011). However, there exist no guarantees that a given vehicle will actually be able to meet the desired speed trace, since the power request is directly computed from the speed and not checked against the actual powertrain capabilities (Enang and Bannister, 2017b).

2.1.2 The quasi-static approach

The quasi-static approach introduces a drive model to a frontward-facing calculation process to control the vehicle system following the target speed profile to simulate the dynamic performance of the vehicle in the scenario of tracking a given driving cycle, as shown in Fig. 2-3. The driver model in quasi-static approach usually applies a PID control (Millo, Rolando and Andreata, 2011) or fuzzy logic control (Zhou *et al.*, 2017, 2018; Cash *et al.*, 2018a) to control the vehicle with the signals (of vehicle gas pedal and brake pedal) which are calculated with the speed error between the actual speed and desired speed. The energy management system allocates the torque demands of engine and motor according to the driver command and sends the signals to the relevant downstream controllers (e.g. engine controller, motor controller). In the quasi-static method, engine and motor are modelled by static

performance maps (e.g. torque, fuel consumption, efficiency, emission, etc. al.) or using 0-D dynamic models (Millo, Rolando and Andreata, 2011). The model uses control signal and current speed or velocity as input to calculate their available torque and transfer the toque to its downstream models. The quasi-static method is well-adopted by many commercial vehicle simulation software, e.g. AVL CRUISE, PSAT, IPG CARMAKER, CARSIM, etc. al. (Gao, Mi and Emadi, 2007).

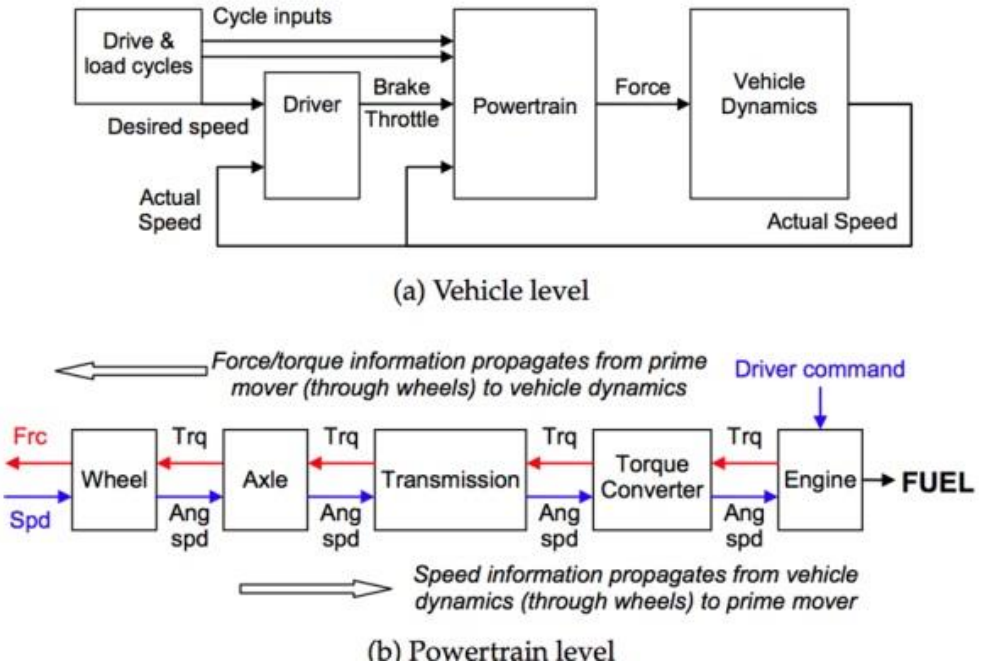


Fig. 2-3 Quasi-static powertrain model (Guzzella and Sciarretta, 2007)

Bond graph modelling is one of the most effective modelling tools for quasi-static modelling (Xia, Linkens and Bennett, 1993; Hubbard and Youcef-Toumi, 1997), which is a graphical tool to describe subsystem interactions in the form of power exchange as in Fig. 2-4. Three elements, including resistance (R), capacitances (C), and inertias (I) are used to model a second-order dynamic system of power

interchange. Although these names initially suggest a direct application in electrical systems, they are used in any other domains as well, e.g., friction as mechanical resistance, a compressible fluid as capacitance, and a flywheel as an inertial element (Gao, Mi and Emadi, 2007). The power exchange of each element is based on the port(s) of each element (normally one or two ports) using the product of two variables including effort (e) and flow (f). Taking a mechanical system as an example, torque is an effort and speed is a flow, and the power is torque times speed. Detailed hybrid vehicle modelling procedure using bond graph can be found in (Xia, Linkens and Bennett, 1993; Hubbard and Youcef-Toumi, 1997).

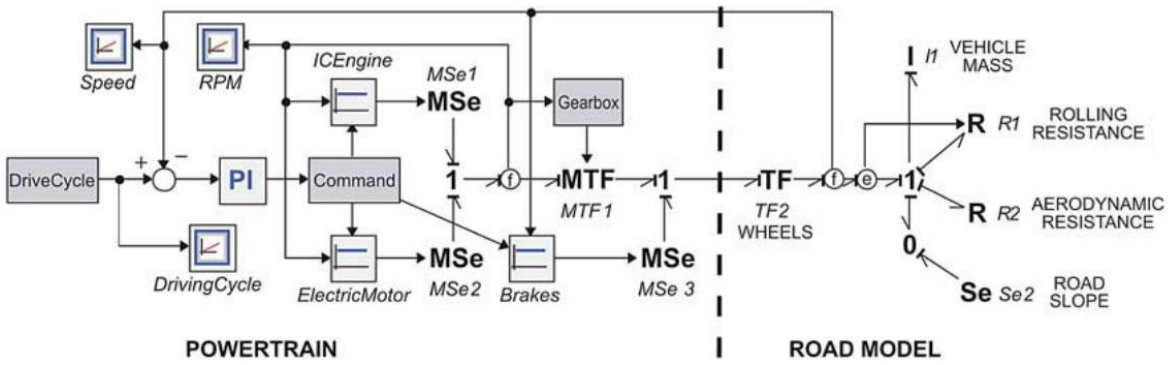


Fig. 2-4 Bond graph model of an HEV (Gao, Mi and Emadi, 2007)

The quasi-static modelling approach provides reasonable accuracy when it comes to the evaluation of the fuel consumption and NOx of a vehicle equipped with conventional powertrain (Enang and Bannister, 2017b). For pollutants like soot, the transient operations and related “turbo-lag” phenomena significantly contribute to the cumulative emissions of the cycle; thus a more detailed model is needed to properly

capture the engine transient behaviours (Guzzella and Sciarretta, 2007).

2.1.3 The full dynamic modelling approach

Full dynamic modelling approach would consider the dynamics of internal combustion engines and batteries, in which, the fluid mechanics and chemical reactions will be modelled in details. For an internal combustion engine, the intake and exhaust systems can be represented as a network of ducts connected by junctions that represent either physical joints between the ducts, such as area changes or volumes, or subsystems such as the engine cylinders (Millo, Rolando and Andreata, 2011). The finite element analysis technique is widely used for solving the governing equations of the conservation of mass, momentum and energy flow (Enang and Bannister, 2017b). The full dynamic modelling method allows reasonable accurate simulation of highly dynamic events such as abrupt vehicle accelerations. However, the implementation of full dynamic models requires huge computational effort, and it is hard to be applied in modelling for vehicle control level.

2.1.4 The hardware-in-the-loop modelling approach

The hardware-in-the-loop approach is developed for modelling of a hybrid vehicle system by involving one or more real hardware to obtain the real-time performance of the device(s) under research. The models for hardware-in-the-loop approach

should have the capability of real-time simulation; therefore, the model is always built by the kinematic method or the quasi-static method (Gao, Mi and Emadi, 2007). Hardware-in-the-loop test is widely used in controller development and concept validation, which is usually developed with controller-in-the-loop, powertrain-in-the-loop, or driver-in-the-loop for hybrid electric vehicle development.

Controller-in-the-loop system is the most compact version for hardware-in-the-loop testing, as shown in Fig. 2-5, which includes a physical controller and a real-time computer connecting via inputs/outputs interface or CAN interface for vehicle controller development. With the models running in a real-time computer, the performance of the control algorithms can be tested and evaluated by the controller-in-the-loop test. The controller-in-the-loop system requires reliable real-time computers to provide necessary signals generated from the vehicle model. dSPACE, ETAS, and National Instruments (NI) are the world-leading suppliers for real-time testing. dSPACE DS2202 Mid-size plant was used by the Ohio State University for development of the supervisory controller for a hybrid vehicle (Mura, Utkin and Onori, 2015). The University of Birmingham has developed a controller-in-the-loop system with ETAS DESK-LABCAR real-time computer for cyber-physical control of a hybrid vehicle (Zhou *et al.*, 2018). NI PXI-8135 was used to develop a controller-the-loop system for control of an electric vehicle (Ciornei *et al.*, 2018).

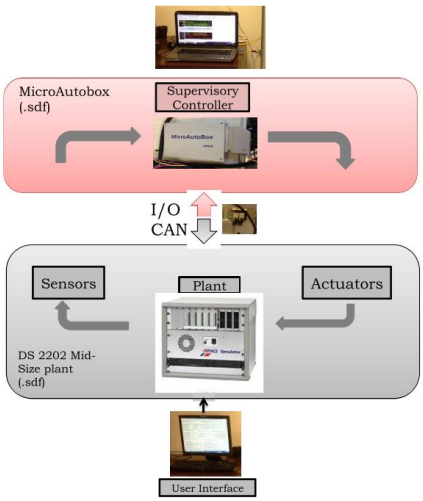


Fig. 2-5 Controller-in-the-loop testing system (Mura, Utkin and Onori, 2015)

It is necessary to involve some real powertrain subsystems or the whole powertrain in the real-time simulation loop to research the dynamic performance of the hybrid vehicle, especially for the subsystem that is hard to be accurately modelled, e.g. engine, battery, and ultra-capacitor. An engine-in-the-loop system is developed for research of soot emission of a diesel hybrid vehicle, as shown in Fig. 2-6, in which a real diesel engine is connected within the simulation loop via EMCON 400 flexible testbed (Kim *et al.*, 2015). The soot emission from the real engine with the proposed power management strategy (PMS) was measured with DMS500 manufactured by CAMBUSTION Ltd. A transmission-in-the-loop system is developed for a series-parallel hybrid vehicle using PROCYON and UNICODRIVE (Zhang, Zhang and Yin, 2016), in which, the real-time vehicle model is running in PROCYON and generate a synchronised signal to the UNICODRIVE to control an electric motor. A fuel-cell/battery-in-the-loop system is developed with CHROMA DC load and a real-time

PC to research the power management system of a hybrid energy storage system (Gauchia and Sanz, 2010).

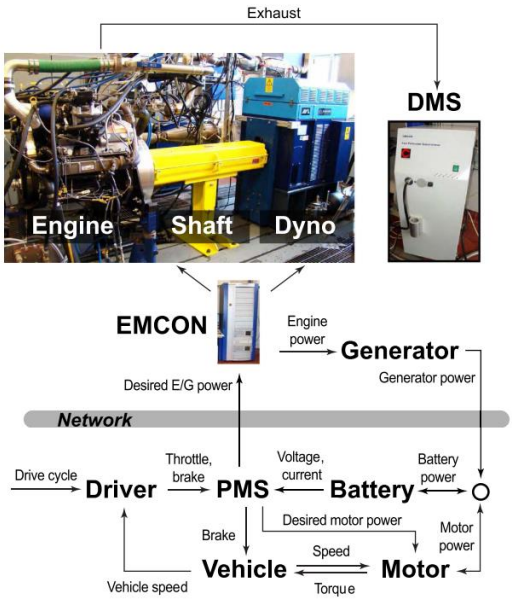


Fig. 2-6 Engine-in-the-loop testing system (Kim *et al.*, 2015)

Driver behaviour is an immediate response to the traffic conditions, which affects the energy use of the vehicle system significantly (Zhang *et al.*, 2017). Recent proposal on real-world driving emissions (RDE) testing procedure has drawn lots of attention on drivers' behaviours for vehicle powertrain development (Martínez and Cao, 2019b), and the dynamics involved by human driver is hard to be fully modelled via simple PID or fuzzy logic control as it is in a quasi-static model (Zhou, Ravey and Péra, 2019). Therefore, it is necessary to involve a real human driver into the simulation loop for the research on driver behaviour modelling and advanced vehicle control algorithms. A driver-in-the-loop testing system is built by the Ohio State

University for the development of driver models, as shown in Fig. 2-7 (Zeng and Wang, 2017). A six-degree-of-freedom driving simulator is developed for the research of control strategy for a hybrid truck (Martínez and Cao, 2019c).

Fig. 2-7 Driver-in-the-loop testing system (Schnelle *et al.*, 2017)

2.1.5 The outlook of modelling methods for hybrid vehicles

With the rapid development in computer science and information technology, future hybrid vehicle products are expected to be more '*information integrated*' and '*intelligent*'. The individual vehicles and the OEMs would be parts of the Internet of Vehicles (IoV) and will be benefit from the sharing of information and connected computing sources.

In the process of future vehicle research and development, and integrated information system including physical coupling (of physical hardware) and data coupling (of cybernetic hardware and software) will connect all the testing facilities and Cloud computing resource via the Internet for global optimisation of vehicle

products as in Fig. 2-8 (Yi Zhang *et al.*, 2018). The concept of supper-hardware-in-the-loop (or X-in-the-loop) is under development in academia and industry, which is expect to be integrated with artificial intelligence algorithms for real engineering development (*XiL Approach: An ETAS Solution*, 2019). The University of Birmingham together with Textron Ltd. and Hyper-drive Ltd. have demonstrated the development of a hybrid aircraft-towing tractor (including sizing and energy management) based on information integration and artificial intelligent algorithms (Rachel Cooper, 2017). These make it possible for future vehicle development with advanced optimisation algorithms for design optimisation and controller calibration with multi-variable and multi-objective (Ma *et al.*, 2017; Zhou *et al.*, 2017; Cash *et al.*, 2018a; Yunfan Zhang *et al.*, 2018).

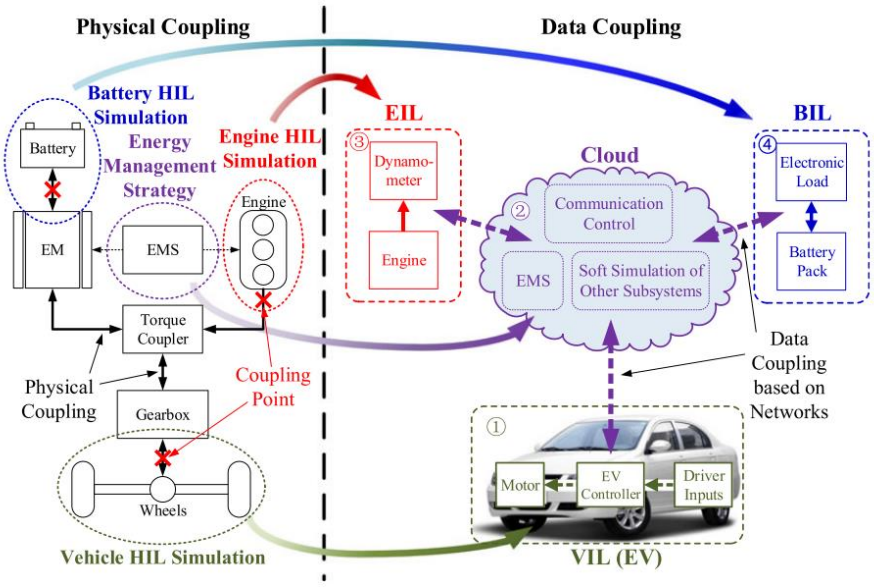


Fig. 2-8 Information integration for future electrified vehicle development (Yi Zhang *et al.*, 2018)

Future hybrid vehicles will also benefit from information fusion of on-board sensor signals and V2X (vehicle-to-everything) network data, in terms of operation safety and further energy efficiency improvement (Martinez *et al.*, 2017). An iHorizon (intelligent horizon) system is proposed as a concept prototype for future hybrid vehicle, which integrates the functions of driving style recognition (DSR), short-term speed prediction (with the help of on-board sensors), and long-term speed prediction (with the help of V2X, ITS), as shown in Fig. 2-9 (Martínez and Cao, 2019a). This framework allows the hybrid vehicle to be personised via online control strategy optimisation (Orecchini *et al.*, 2018; Qiu *et al.*, 2019) and control parameter calibration (Ma *et al.*, 2014, 2017). The short-term prediction and long-term prediction with vehicle platooning would also be developed based on online optimisation algorithms, which can help the vehicle operate in a more safe and efficient way (Alam *et al.*, 2015; Jia *et al.*, 2016).

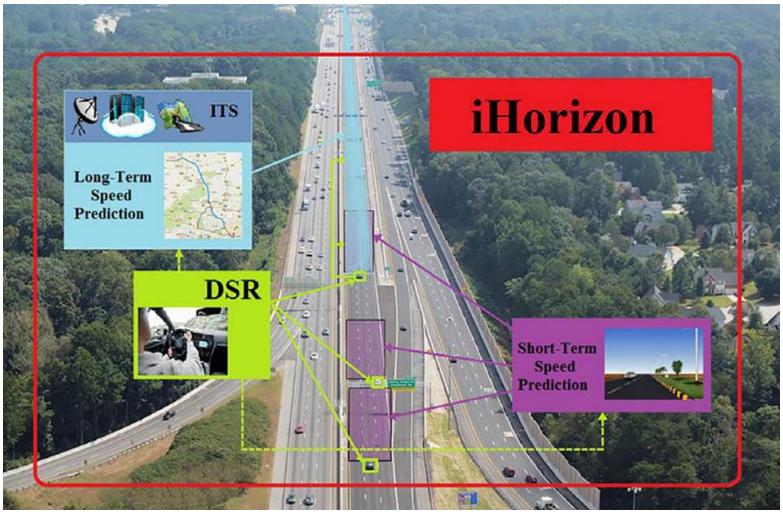


Fig. 2-9 System framework of the iHorizon (Martínez and Cao, 2019a)

2.2 Offline Optimisation of Hybrid Vehicles

Offline optimisation deals with the problems that the start point and target point are available, and the decision should be made from a lot of candidate routes from the start to the target. Offline optimisation of hybrid vehicle determines the optimum components' size and control parameters based on a given driving cycle (e.g. WLTP, NEDC). The optimisation results will allow the vehicle achieving its best performance (e.g. fuel economy, emission, power). This section provides a review of existing research on components sizing and controller calibration for electrified vehicles.

2.2.1 Components sizing

The components sizing is an essential procedure in vehicle development, which was based on the experience of human engineers in the early stage by considering the target vehicle specification, e.g. packaging, components availability, etc. Normally, the sizing results can be obtained using the OEM's database with the help of some computer aid design and engineering software (CAD, CAE) such as ADVISOR (Wipke, Cuddy and Burch, 1999) and AVL CRUISE (Zamora *et al.*, 2013). Recently, optimisation methods have been developed for component sizing based on the 'model-in-the-loop' framework, as shown in Fig. 2-10.

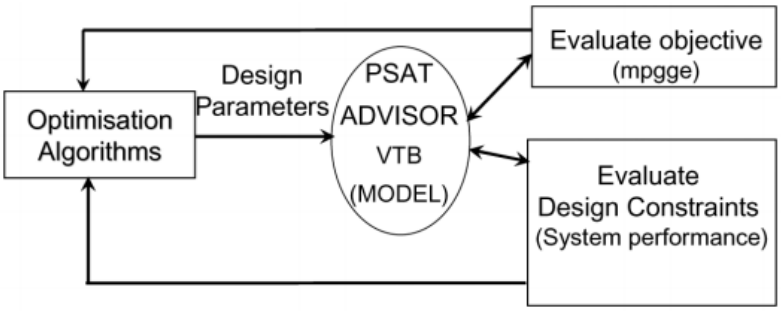


Fig. 2-10 Model-in-the-loop design optimisation process (Gao and Mi, 2007)

Main optimisation objectives of components sizing include fuel economy, energy efficiency, cost and size of components, etc. The most concerned objective is to minimise the fuel consumption with the constraint of the battery SoC level at the end of the given driving cycle (Yang *et al.*, 2019). Normally, component sizing is formulated as a multi-objective optimisation process with more than two objectives, for example, fuel economy and components' price are considered (Ebbesen, Dönitz and Guzzella, 2012; Murgovski *et al.*, 2012b); energy efficiency and component size are considered in (Zhou *et al.*, 2017). Recently, component sizing with 'many-objective' (more than three) has been researched. Fuel economy and emissions including NOx, CO, HC are considered as a weighted four-objective optimisation in the component sizing of a parallel hybrid vehicle in (Xiaolan Wu *et al.*, 2008). A seven-objective optimisation has been carried out for hybrid vehicle optimisation (Cheng *et al.*, 2017), in which three evolutionary algorithms have been researched to identify the decision-maker's preference.

In terms of the driving cycles used for component sizing, there is a trend of

incremental transformation from the conventional quasi-static driving cycle (e.g. NEDC), towards driving cycles with more transient operations (e.g. WLTP), and finally, real-world driving data (e.g. RDE) will be used. The operation points of different driving cycles including conventional NEDC (New European Driving Cycle), ongoing WLTP/WLTC (Worldwide Harmonised Light Vehicle Test Procedure/Cycle), future RDE (Real-word Driving Emission) are compared in Fig. 2-11. There is also a trend to optimising the components' size considering their life-cycle performance, e.g. life-cycle CO2 emission, life-cycle cost, etc.

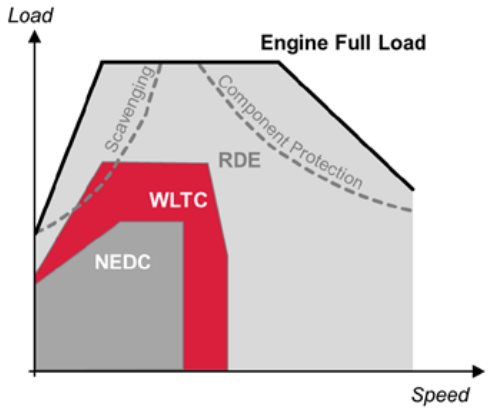


Fig. 2-11 Cycle operating ranges comparison of NEDC, WLTC, and RDE

(Sanguinetti, 2018)

The state-of-the-art optimisation algorithms have been implemented for component sizing. Dynamic programming has been developed for component sizing of a hybrid vehicle, the advantage of which is the capability of global optimisation with nonlinear, non-convex objective functions of the components consisting of continuous and mixed integer variables (Zoelch and Schroeder, 1998). Convex optimisation was

used for component sizing of a plug-in hybrid powertrain in (Murgovski *et al.*, 2012b), which determines the size of the engine and electric machine simultaneously using thermal models of components. The paper suggests that component sizing using convex optimisation enables two or more variables optimisation with low computation time (Murgovski *et al.*, 2012b). A multi-objective optimisation problem of sizing drive-train components is solved by particle swarm optimisation in (Ebbesen, Dönitz and Guzzella, 2012) which indicates the proposed algorithm performs significantly better than the baseline methods (i.e. Nelder-Mead simplex algorithm and exhaustive search algorithm). A chaos-enhanced and accelerated particle swarm optimisation algorithm for component sizing is developed by the author (Zhou *et al.*, 2017). According to the research, the proposed chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm can accelerate the convergence and prevent the diversity of optimisation results so that the proposed algorithm can help to obtain more reliable sizing results (Zhou *et al.*, 2017).

2.2.2 Controller calibration

Hybrid vehicles can operate in many different modes, as discussed in Section 1.1.2. In the design process, the control parameters should be carefully tuned to ensure the basic functionalities and guarantee the vehicle products satisfying the regulations and legislations. The process of tuning control parameters is called ‘calibration’. The

most commonly used method for calibration in the industry is the design of experiments (DoE). Although the design of experiment can be assisted with software (e.g. AVL CAMEO, ETAS INCA), the calibration process still needs human's experience to select proper set points and control parameters initially (Ma *et al.*, 2014).

Obviously, human's experience cannot always guarantee the calibration results are the global optima, and it is hard for human engineers to deal with multi-objective optimisation simultaneously (Tayarani, Yao and Xu, 2015). 'Model-in-the-loop' methods, as discussed in Section 2.2.1, can also be used for the optimisation of control parameters. Recently, a new 'Hardware-in-the-loop' optimisation scheme or so-called 'non-model-based optimisation' has been developed for controller calibration which can prevent the negative influence (caused by inaccurate models) to the optimisation results (Ma *et al.*, 2017). Taking the engine calibration process as an example, a diagram illustrating the difference between model-based calibration and non-model-based calibration is shown in Fig. 2-12.

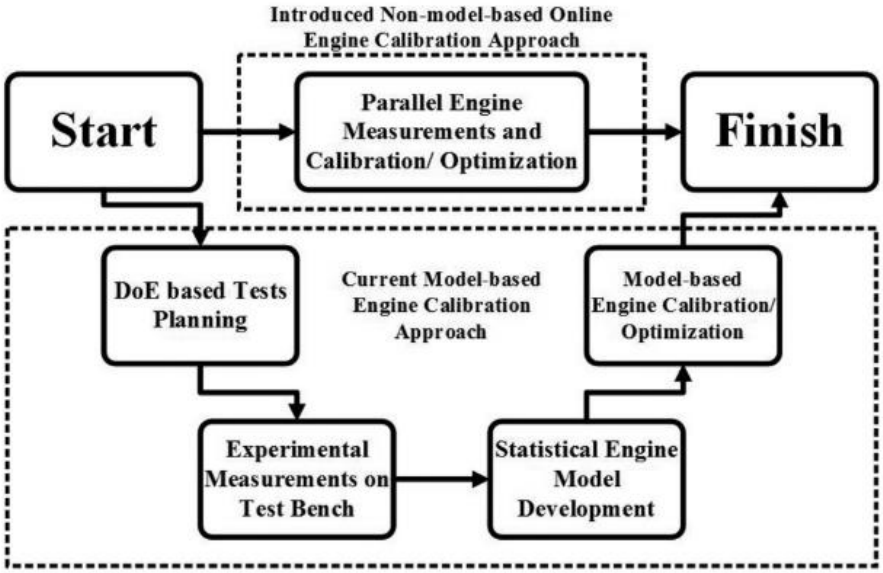


Fig. 2-12 Model-based and non-model-based method for engine calibration (Ma *et al.*, 2017)

Emerging technologies of artificial intelligence have been researched and developed in the field of control parameters calibration for both supervisory controllers and subsystem controllers of electrified vehicles (Huang *et al.*, 2018). Hadj-Said et al. developed a convex optimisation method for energy management control parameters of parallel hybrid electric vehicles (Hadj-Said *et al.*, 2016). A rapid dynamic programming approach was developed for power split control of a hybrid vehicle, which can save 6.56% and 3.15% fuel under FTP72 and HWFET cycles respectively (K. T. Chau and Wong, 2002). Particle swarm optimisation was used to calibrate the energy management controller (Shen *et al.*, 2017). Model-based calibration with SPEA-II algorithm was developed for control parameter optimisation of HCCI engine and GDI engine in (Ma *et al.*, 2014) and (Ma *et al.*, 2018) respectively.

Calibration results determine the performance of the vehicle system and subsystems; this will affect the optimal size of the hybrid vehicle (Sundstrom, Guzzella and Soltic, 2010). Therefore, the research on simultaneous optimisation for optimal sizing and control calibration emerges. The multi-objective genetic algorithm was used for simultaneous optimisation of hybrid vehicle parameters (Fang *et al.*, 2011). An optimisation considering both control parameters and component size is a performance with a two-stage algorithm, which dynamic programming was used for control optimisation while genetic algorithm was used for component sizing (Chen, Lin and Ren, 2018).

2.2.3 Outlook for offline optimisation of hybrid vehicle

There is a trend for the future hybrid vehicle to involve more energy storage systems (Cheng *et al.*, 2017) as well as more sensors and information fusion controllers to improve its energy efficiency, which will increase the complexity in the design process (Huang *et al.*, 2018). It can be predicted that future design optimisation will include: 1) multiple-objective or even many-objective optimisation tasks considering both system-level global performance as well as subsystem level local performance (Cheng *et al.*, 2017); 2) unified optimisation combining component sizing and controller calibration (Xu *et al.*, 2015); 3) more engagement of artificial intelligence algorithms, cybernetic information, and Cloud computing (Pothirasan, Rajasekaran

and Muneeswaran, 2018); 4) consideration of the differentiation of driver behaviour and traffic dynamics in real-world driving (Pourabdollah *et al.*, 2017).

This thesis will research into a simultaneous offline optimisation for intelligent components sizing and control parameters tuning of an aircraft towing tractor in its most common driving condition. A Particle Swarm Optimisation algorithm is used for the offline optimisation as the baseline method for the following reasons: 1) It is a derivative-free optimisation algorithm which does not require detailed mathematical expressions for the nonlinear systems; 2) It is a global search algorithm which can deal with mixed discrete and continuous variables, and multiple objectives that cannot be solved by convex optimisation; 3) Its algorithm has fewer parameters to be tuned compared with the genetic algorithm, the bee algorithm, the ant colony algorithm, and the simulated annealing algorithm, so that it is easier to implement for mechanical engineers who have limited knowledge of computer science; 4) It is a computational more efficient algorithm compared with dynamic programming, so that it can be implemented for real-time optimisation.

A new variant of the PSO, Chaos-enhanced Accelerated Particle Swarm Optimisation (CAPSO) algorithm (Gandomi, Yun, X.-S. Yang, et al., 2013), will be explored as an improved method in this thesis. Different chaotic mapping strategies will be researched for the development of the best CAPSO algorithm for the optimisation of the hybrid tractor powertrain. The new algorithm will accelerate the

convergence of the ‘particles’ with faster speed than the baseline method. The consistency of the results will be improved to make the optimisation more reliable for real engineering application.

2.3 Energy Management of Hybrid Vehicles

For any hybrid electric powertrains, there are normally two energy-flow paths including the electric energy flow path (through batteries, ultra-capacitors, electric motors, etc.) and mechanical energy path (through the engine, transmission, clutch, driveshaft, differentials, etc.). Energy Management System (EMS) determines the optimal distribution of energy flows (including the electric path and mechanical path) in HEVs to satisfy the driver’s demand and achieve maximum energy efficiency. Generally, energy management strategies can be classified into two main categories including rule-based strategies and predictive strategies.

2.3.1 Rule-based methods for energy management

Energy management strategies with explicit control rules include the thermostat strategy, power following strategy and state machine strategy. Controlled by the thermostat (on/off) control strategy, internal combustion engine operates at its highest efficiency point once it turns on, while the battery’s state-of-charge (SoC) is always maintained between its pre-set upper and lower bounds by switching the

internal combustion engine ‘on’ and ‘off’ (Wang *et al.*, 2017). Power following strategy uses the battery as an energy buffer to smooth the power demand so that the engine can work in a relevant high efficient condition to follow a smoothed power demand, which has been successfully applied in mass-produced HEVs such as Honda Insight and Toyota Prius (K. . Chau and Wong, 2002). The state machine-based approach determines the transition between operating modes (as described in section 1.1.2) by a state machine that is based on vehicle operating conditions and driver’s command (Tie and Tan, 2013).

Fuzzy logic control enables energy management with fuzzy rules which have the capability to handle numerical data and linguistic knowledge. Two separated fuzzy logic controllers for Mode Decision and Parallel-driving Energy Management are proposed in (Xiong, Zhang and Yin, 2009), which can help the hybrid vehicle reduce 30.3% energy consumption comparing with the one using explicit rule-based strategies. The robustness of the energy management system to variations of driving cycles can be improved by applying an adaptive fuzzy inference system (Mohebbi, Charkhgard and Farrokhi, 2005).

The energy management strategy with fixed control rules lacks the ability to deal with the uncertainties brought by dynamics in real-world driving (Salmasi, 2007). The control rules, including explicit rules and fuzzy rules, can be optimised as an offline optimisation or calibration in the R&D process to ensure the hybrid vehicle product

passing the regulations as discussed in Section 2.3.2. Online optimisation technologies for control rules of energy management system are emerging, which can adjust the control rules during real vehicle operation to make the control strategy more adaptive to the real-world driving comparing with the energy management control using fixed control rules (Martinez *et al.*, 2017; Wang *et al.*, 2017; Huang *et al.*, 2018). Operation-mode prediction is used for optimisation-based rule-correction of the energy management control, which results in at least 9.6% fuel economy improvement in selected driving cycles (Liu *et al.*, 2018). The threshold of the energy management rules of a series hybrid vehicle was optimised online based on Pontryagin Minimum Principle, which can reduce 6.83% fuel consumptions (Shabbir and Evangelou, 2019).

2.3.2 Predictive methods for energy management

Trip information is critical to the energy management of hybrid electric vehicles (Martinez *et al.*, 2017). The energy management methods based on the prediction of future information of vehicle speed and power demand are being researched as a global online optimisation method for hybrid vehicles (Huang, Wang, *et al.*, 2017). Model predictive control (MPC) is currently considered as one of the most effective approaches for online systems optimisation with multiple variables and objectives. As shown in Fig. 2-13, predictive model and online optimizer are the two key

elements for MPC.

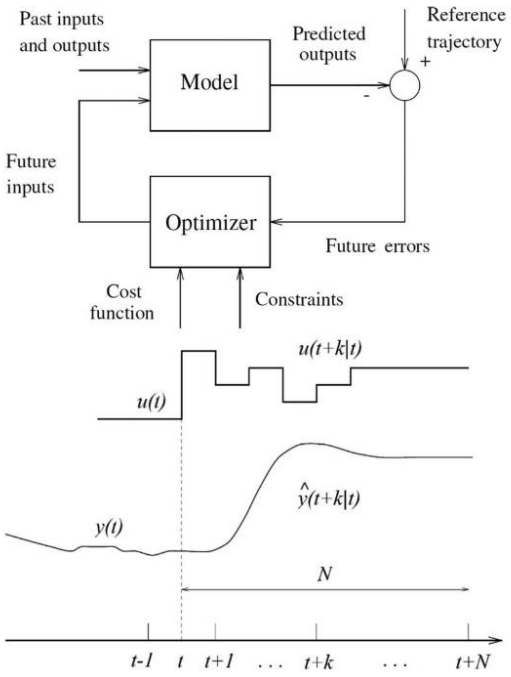


Fig. 2-13 Problem-solving structure of MPC (top) and inputs/outputs signal of a single iteration in the MPC algorithm (bottom) (Martinez *et al.*, 2017)

The operation of MPC comprises four main steps: 1) prediction over a fixed horizon with length N, which depends on the historical data recorded and system model; 2) control policy calculation from t to t+N based on the previous prediction; 3) application of the control policy calculated for the current instant t, discarding the rest; 4) update with real measurements at t, and return to Step 1. Using fast control algorithms in step 2 is particularly important due to the requirement of real-time computations (Borhan et al., 2009).

The essential procedure of MPC is the prediction of future vehicle speed or power

demand over a predictive horizon. The simplest method to predict the future vehicle speed information is the exponential varying expression method, which assumes the future driver's torque demand is exponentially decreasing over the prediction horizon (Chao Sun *et al.*, 2015). A fuzzy predictor was developed to determine future states (i.e. vehicle speed and torque demand) according to the historical data through a look-ahead window (Tie and Tan, 2013). Stochastic Markov chain modelling, which predicts the future events using the transition possibility matrix, is a promising and important method utilized in modelling driver behaviour or predicting the vehicle velocity and power demands (Moura *et al.*, 2011; Shen *et al.*, 2018). A fuzzy encoder is integrated with the transition possibility matrix for identification of vehicle states, which can improve the accuracy of Markov chain model for vehicle power demand prediction in various of driving conditions (Liu *et al.*, 2017). A new deep fuzzy predictor is developed by the author's team to achieve 19% more accurate vehicle speed prediction compared to a discrete Markov chain model (J. Li *et al.*, 2019). Artificial neural networks have also been used for vehicle state prediction. Recurrent neural network (RNN) has been used for energy management of a mild hybrid vehicle in (Feldkamp, Nasr and Kolmanovsky, 2009). Long-term-short-term-memory (LSTM) deep network is developed for electric load forecasting (Bouktif *et al.*, 2018).

An appropriate solver is necessary for the optimisation at each time interval for MPC. MATLAB command 'quadprog' (Sampathnarayanan *et al.*, 2010), 'CVXGEN' (Romijn

et al., 2015) and ‘qpOASES’ (Huang, Khajepour, *et al.*, 2017) can be used for quadratic programming (QP) of linear constrained MPCs. The nonlinear MPC can be solved by sequential quadratic programming (SQP) (Hu, Wang and Tang, 2017); the mixed-integer optimisation problems in hybrid MPC can be solved by the hybrid optimisation toolbox in MATLAB (Ripaccioli *et al.*, 2009). However, only linear MPCs have been realized the feature of the real-time implementation, running nonlinear MPC (NMPC) with QP or SQP for HEV energy management in real-time controllers has yet to be demonstrated (Huang, Wang, *et al.*, 2017). Particle swarm optimisation (PSO) algorithm is a potential candidate for real-time NMPC solving (Xu *et al.*, 2016) since it works with fewer tuning parameters and less computational effort. PSO also has the capability of dealing with integer variables (Ostadi and Kazerani, 2015; Pourabdollah *et al.*, 2015). Chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm for real-time model-based predictive control was developed by the author (Zhou *et al.*, 2018).

Currently, researchers are exploring and investigating more advanced methods for online optimisation problems in real-time with the help of Cloud computing and Artificial Intelligence (AI) for HEV energy management (Hu, Wang and Tang, 2017; Zhou *et al.*, 2018). Energy management of HEV with reinforcement learning algorithm is an emerging research topic, and there are only a few publications can be found in this field. Zou and Liu have proposed a series of predictive energy

management strategies using conventional Q-learning (Liu *et al.*, 2015a, 2015b, 2017; Zou *et al.*, 2016), the methods can significantly improve the vehicle performance compared with conventional rule-based strategy. The authors have proposed a multi-step reinforcement algorithm for model-free energy management control of a hybrid vehicle, which can save at 7.8% energy (Zhou *et al.*, 2019).

2.3.3 Outlook for hybrid vehicle energy management

The performance of existing MPC-based energy management is affected by three main aspects including 1) accuracy of predictive models (Murphey *et al.*, 2012; C Sun *et al.*, 2015; Soriano, Moreno-Eguilaz and Álvarez-Flórez, 2015; Sun, He and Sun, 2015; Sun, Sun and He, 2017); 2) length of predictive horizon (Nuijten, Koot and Kessels, 2003; Huang, Khajepour and Wang, 2016) and 3) optimisation ability of the control algorithm (Ripaccioli *et al.*, 2009; Sampathnarayanan *et al.*, 2010; Romijn *et al.*, 2015; Hu, Wang and Tang, 2017; Huang, Khajepour, *et al.*, 2017).

It can be predicted from the existing researches that future efforts would be made in 1) more accurate vehicle information prediction using information fusion of both near-field sensors and global information from V2X network (Martínez *et al.*, 2019); 2) more powerful computing technology for real-time optimisation of vehicle control system (Huang, Wang, *et al.*, 2017; Zhou *et al.*, 2018); 3) collaborative control with model-based predictive control framework and emerging ‘model-free’ control with the

self-learning capability of control strategies through reinforcement learning (Liu *et al.*, 2017; Bouktif *et al.*, 2018; Xiong, Cao and Yu, 2018).

This thesis will research into predictive methods for energy management of the aircraft-towing tractor. This will follow the most commonly used model-based predictive method. New nonlinear model-based predictive methods will be explored by the implementation of the online particle swarm optimisation algorithm. The CAPSO algorithm will be used for solving the nonlinear programming problem of energy management in real-time. Reinforcement learning, which is a powerful online optimisation method, will be explored as an advanced predictive method for energy management. New multi-step reinforcement learning will be researched to enable the 'model-free' predictive control of the energy flow. It will allow a parallel system (to the conventional rule-based/model-based control system) which can continuously optimise the vehicles' energy efficiency in real-world driving.

2.4 Summary

This chapter provides a comprehensive review of 1) modelling methods for hybrid vehicle development, 2) offline design optimisation technologies of hybrid vehicles and 3) energy management method for hybrid vehicles. From the existing literature, the future trends of the above areas can be summarised as:

- The modelling method for hybrid vehicles is transiting from the kinematic approach to the methods that can present the real dynamic of the HEV system with the capability of real-time computing. Emerging IoT and Hardware-in-the-Loop technologies will allow the simulation and validation of the vehicle system in a more realistic environment.

- It is the tendency to consider both component size and control parameters simultaneously during the design optimisation process, and advanced artificial intelligence is expected to help boost the design optimisation process with reliable and robust results. PSO algorithm and its variants are suitable candidates for solving the simultaneous optimisation problem.

- Model-based predictive control (MPC) is a robust optimal control method; however, advanced technologies for online nonlinear optimisation is still needed to be developed for energy management of the hybrid electric vehicle.

- Artificial intelligence with the capability of reinforcement learning is expected to enable the adaptive optimisation of control policy with model-free predictive control, which will sit along MPC for further control performance optimisation.

Chapter Three

RESEARCH METHODOLOGY, FACILITIES, AND REAL-TIME MODELS

This PhD thesis focuses on the development of an electrified aircraft-towing tractor, including design optimisation and optimal energy management control. This chapter firstly introduces the research target and technical route for technology development, followed by the description of the research methodologies for design optimisation and optimal energy management. This chapter illustrates the research facilities as well as the development of a demonstrator and its testing system. The vehicle system and subsystems are modelled in this chapter for real-time simulation and the hardware-in-the-loop (HiL) test.

3.1 Research Target and Methods

The vehicle system under research is an aircraft-towing tractor, as shown in Fig. 3-1. Different from passenger cars, the aircrafts towed by the tractor can significantly affect the fuel consumption and emission of the vehicle. The vehicle is initially equipped with a solo diesel engine, and it will be transient into a series hybrid vehicle in this study. The main target of this research is to develop a new electrified aircraft-

towing tractor prototype with minimum components size and maximum energy efficiency in use. Both design optimisation (offline) and control optimisation (online) will be carried out for vehicle development, and advanced optimisation methodologies will be researched.

Fig. 3-1 Aircraft-towing tractor system

The development of the hybrid aircraft-towing tractor consists of four main tasks, as shown in Fig 3-2. Task 1 develops models of the electrified aircraft-towing tractor for 1) offline optimisation, 2) HiL testing, and 3) model-based predictive control. The work in Task 1 will be introduced in this chapter. A Hardware-in-the-Loop Testing system will be built in Task 2 (as described in this chapter), and it will be used for evaluation and validation of the control algorithms. Task 3 carries out the design optimisation and develops the optimisation algorithms for offline optimisation and model-based predictive control. Chapter Four gives the details on the complement

of Task 3. Task 4 develops online optimisation methods for optimal energy management control. Model-based predictive control method and model-free predictive control method for energy management are developed in Chapter Five and Six respectively.

There will be five deliverables from this research: 1) a real-time model of the hybrid aircraft-towing tractor, 2) a HiL testing system for control algorithm evaluation and validation, 3) the components' size and the parameters for rule-based energy management control to achieve the maximum energy efficiency with the minimum components' dimension, 4) MPC-based energy management method, and 5) Model-free predictive method for energy management.

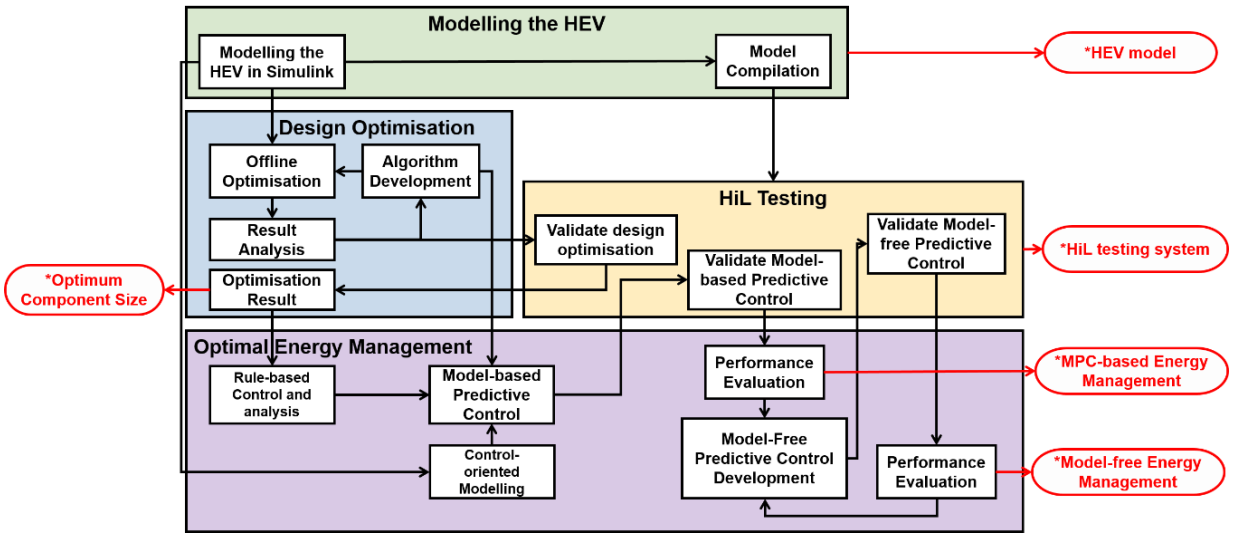


Fig. 3-2 The technical route for the development of the hybrid off-highway vehicle system

3.1.2 Research method for the design optimisation

The procedures of the research on design optimisation method are illustrated in Fig. 3-3. The offline optimisation for obtaining the optimum component size and control parameters will be carried out using the vehicle models developed in Task 1.

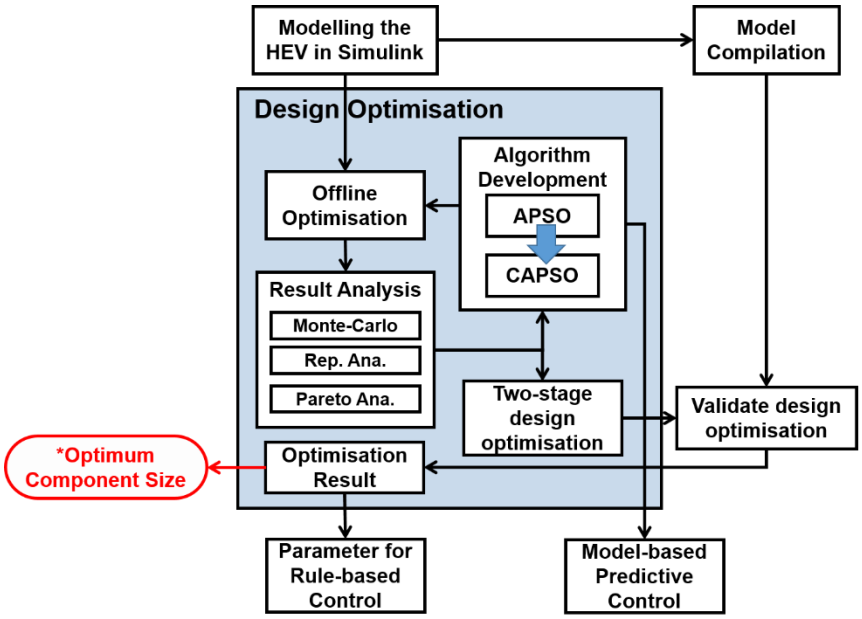


Fig. 3-3 The research on design optimisation

Initially, conventional Accelerated Particle Swarm Optimisation (APSO) algorithm will be used to carry out the optimisation. The drawbacks and limitation of the APSO will then be analysed. The algorithm development for both offline optimisation and online model-based predictive control will be progressed from APSO to Chaos-enhanced Accelerated Particle Swarm Optimisation (CAPSO) algorithm. Different chaotic mapping strategies will be investigated to improve the performance of CAPSO. Monte-Carlo analysis, reputation analysis and Pareto Analysis will be carried out for

a comparison study on the performance of design optimisation using APSO and CAPSO. A two-stage optimisation using CAPSO will be carried out to secure the final optimisation result, and the result will be validated by HiL testing. The deliverable of this task will be the optimum powertrain component size of hybrid tractor. The optimum control parameters for rule-based energy management control and the proposed CAPSO algorithm will be used for the investigation on online optimum energy management control in Task 4.

3.1.3 Research method for energy management control

The research program for energy management control is shown in Fig. 3-4. The research will be firstly carried out an evaluation of rule-based energy management method, and the necessity of the development of predictive control will be discussed. Following the modelling work in Task 1, the control-oriented models will be built for model-based predictive energy management control. A new online optimisation scheme named ‘Online Swarm Intelligent Optimisation’ will be investigated based on the CAPSO algorithm developed in Task 3. The model-based predictive energy management control will be evaluated via HiL testing. To further improve the performance of a predictive energy management method, a model-free predictive control method will be developed based on multi-step reinforcement learning. The multi-step reinforcement learning strategies will be investigated. The model-free

predictive energy management control will also be evaluated via HiL testing.

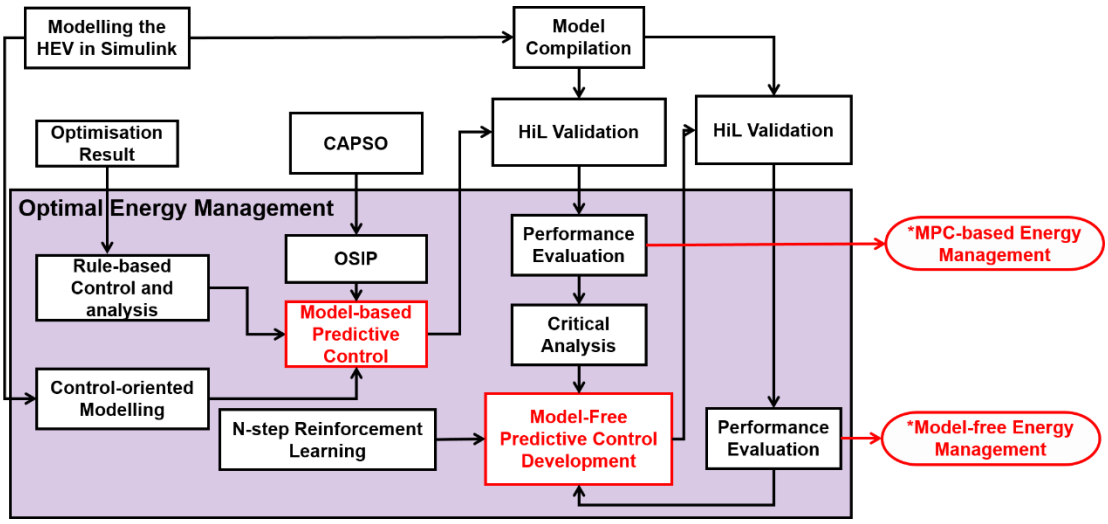
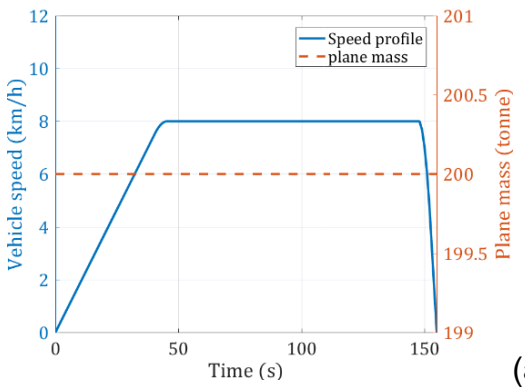


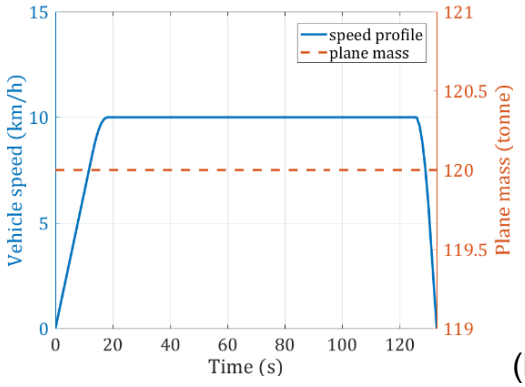
Fig. 3-4 The research on real-time energy management

3.2 Driving Cycles

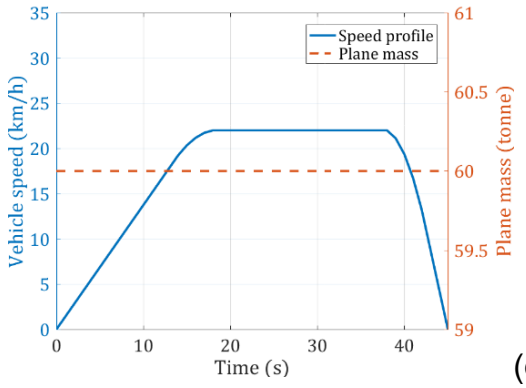
Speed and push-back load of the aircraft-towing tractor varies in real practice. This research uses the Push Back Driving Cycle (PBDC) provided by the OEM based on statistical data collected at London Heathrow airport. The PBDC is made up of four typical driving cycle components, namely, heavy pushback, medium pushback, light pushback and solo run, as shown in Fig. 3-5.



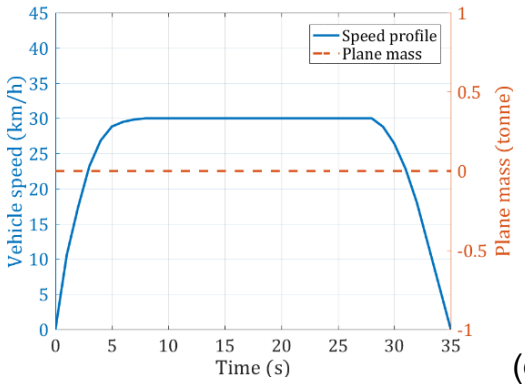
(a)



(b)



(c)



(d)

Fig. 3-5 Speed and plane mass profile of driving cycle components: (a) heavy pushback, (b) medium pushback, (c) light pushback, (d) solo run

Each driving cycle component includes the profile of vehicle speed and the push-back load (plane mass), as shown in Table.3-1.

Table.3-1 Cycle Component Profile

Cycle component	Max. speed	Plane Mass	Acc. time	length
Heavy pushback (H)	8 km/h	200tonne	43s	155s
Medium pushback (M)	10 km/h	120tonne	16s	133s
Light pushback (L)	22 km/h	60tonne	17s	45s
Solo run (S)	30 km/h	0tonne	7s	35s

PBDC-I gives a comprehensive driving cycle, including all the possible scenarios, as shown in Fig. 3-6, i.e. the tractor may pushback a heavy aircraft to the track and then come back solely without the aircraft, or pushback a medium aircraft and then come back solely, or pushback a light aircraft and then come back solely.

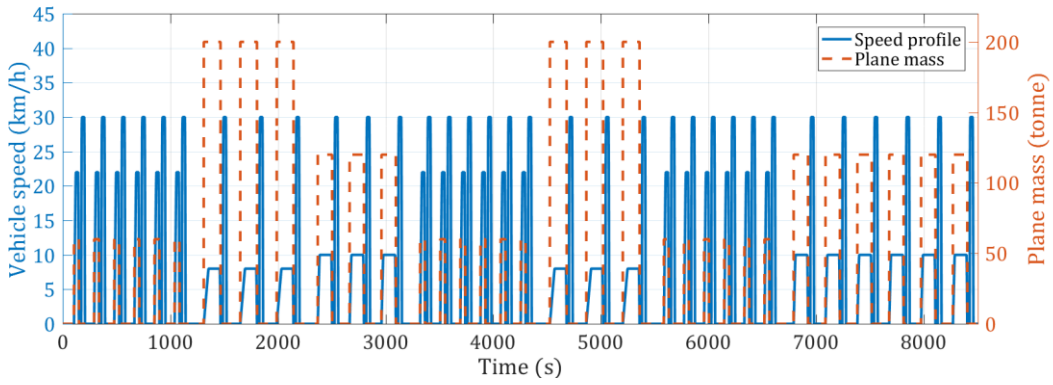


Fig. 3-6 Profile of a push-back duty cycle (BPDC-I)

In an airport, some tractor may also serve a terminal in which the size of aircraft is similar. Therefore, PBDC-II to PBDC-IV is created for those tractors which only serve

the specified terminals. The detailed driving cycle profiles are shown in Table.3-2.

Table.3-2 Push-back Driving Cycle Profile

Cycle name	Cycle components arrangement
BPDC-I	6 L&S+3 H&S+3 M&S+6 L&S+3 H&S+6L&S+6 M&S
BPDC-II	6 L&S+ 6 L&S+ 6 L&S+ 6 L&S+ 6 L&S
BPDC-III	3 M&S+ 3 M&S+ 3 M&S+ 3 M&S
BPDC-IV	2 H&S+2 H&S+ 2 H&S+ 2 H&S+ 2 H&S+ 2 H&S

3.3 Research Facilities

This research is carried out at the Advanced Engine and Vehicle Technology Research Centre at the University of Birmingham. Facilities for HiL testing supplied by ETAS (including both hardware and software) will be used for demonstration and testing validation.

3.3.1 Hardware-in-the-loop testing systems

Hardware-in-the-Loop (HiL) testing is a widely accepted technology in the industry for development and test of complex embedded system, in which, real-time computing and signal emulating technologies are used to enable the functionality

test of control systems. ETAS is a supplier providing world-leading HiL testing facilities for the automotive industry. This research builds a HiL testing system using the hardware provided by EATS including a prototype controller (ES910) and a real-time computer (LABCAR) as shown in Fig. 3-7. The energy management strategies are implemented in ES910 to control the LABCAR for functionality validation. LABCAR emulates the signals as in real vehicles using real-time models and communicates with the ES910 via CAN bus. ETAS software (including ITECRIO, INCA, IP and EE) is used for the real-time implementation of control algorithms and HEV model.

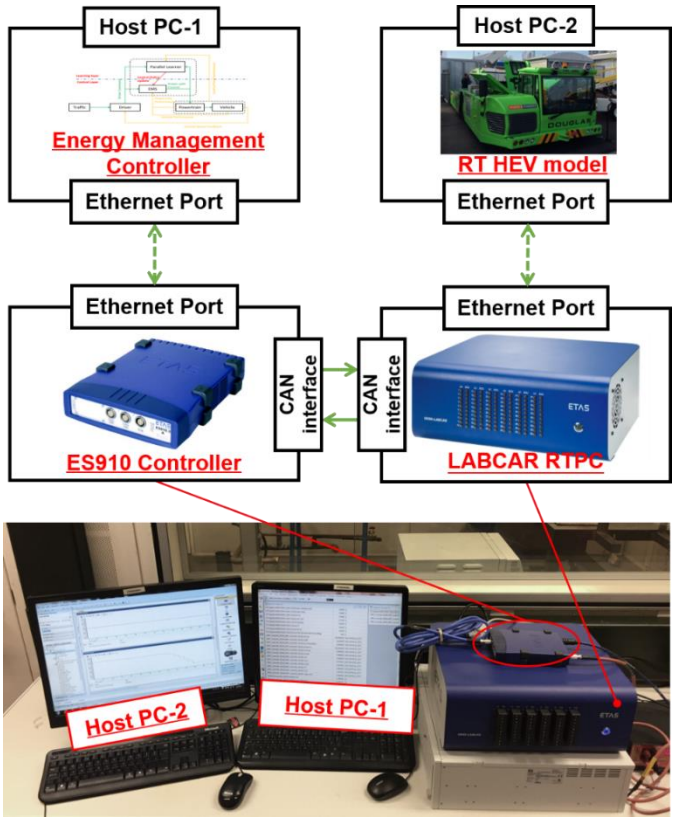


Fig. 3-7 Hardware-in-the-Loop Testing System

3.3.2 ETAS Desk LABCAR

The LABCAR real-time system used for this research consists of three main sub-systems, including ES5100 simulation target real-time PC, ES5340 HEV simulator, and IXXAT PC/CAN interface as shown in Fig. 3-8.

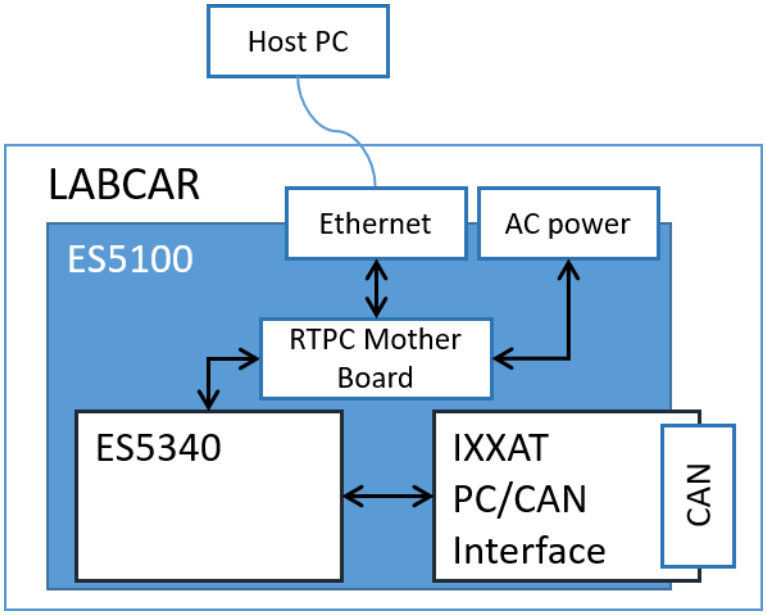


Fig. 3-8 System diagram of the ETAS LABAR system

a) ES5100 simulation target real-time PC

ES5100 is a compact real-time PC for HiL testing which can be located on the desktop, as shown in Fig. 3-9. There is an embedded system with *Linux* pre-installed in ES5100, and real-time models can be download from the host computer via Ethernet. ES5100 is configured with an Intel Core i7-4700 @3.1GHz process, 8GB RAM and 500GB hard disk. There are five external PCIe slots available for add-on modules such as ES5400 HEV simulator and CAN card module. ES5100 is equipped

with a breakout-box (BoB), which can enable the signal by-pass for controller testing. ES5100 sits in the centre of the HiL system which enables the communication with host-PC and external ECU.

(a) (b)

Fig. 3-9 ES5100 real-time PC: (a) front view; (b) rear view

b) ES5340 hybrid vehicle simulation board

ES5340 hybrid vehicle simulation board is used for ECU testing in signal level, as shown in Fig. 3-10, which can emulate signals communications as in the real vehicle with the inputs/outputs of real-time models. ES5340 communicates with ES5100 via PCIe interface. There is a configurable FPGA-based inverter/PMSM model embedded in the board for simulating the motor performance with ultra-high-speed. The FPGA model generates all electrical and mechanical values for the inverter and the electric motor and takes into account all important physical effects, such as saturation and temperature effects. Analogue signals, digital signals and PWM signals can be generated by ES5340. It also allows analogue, digital and PWM inputs from the external controller. In this study, ES5340 is used for real-time

simulation of the HEV plant model.

Fig. 3-10 ES5340 hybrid vehicle simulation board

c) IXXAT iPC-I XC16/PCIe PC/CAN interface

The IXXAT iPC-I XC16/PCIe PC-CAN interface, as shown in Fig. 3-11, is a powerful electronic component which enables the communication between LABCAR and external ECU via CAN bus. It is installed on the ES5100 with PCIe interface. Configured with a 16-bit microprocessor with 40MHz clock, 512Kb RAM, 128Kb flash, and two independent CAN lines, the PC-CAN interface can enable CAN bus connection in accordance with ISO 11898-2 (High-speed), as well as the connection in accordance with ISO 11898-3 (Low-speed). In this study, IXXAT PC/CAN interface is used to enable the CAN bus communication between the LABCAR and the prototype controller.

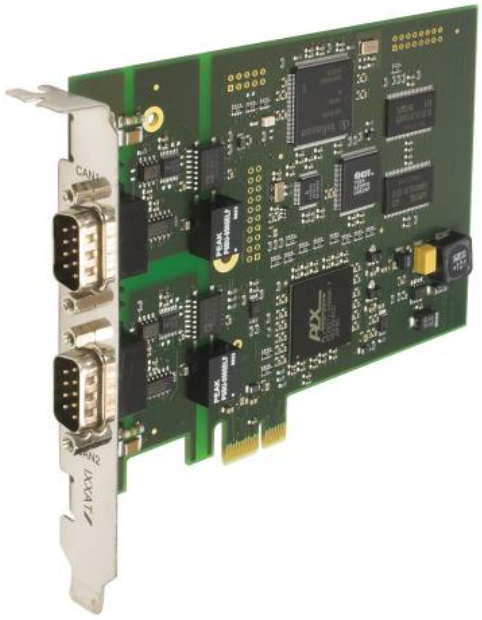


Fig. 3-11 IXXAT PCIe PC/CAN interface

3.3.3 ETAS ES910

ETAS ES910 is a controller prototype for HiL testing, whose core components includes a 1.5 GHz microprocessor, 4Gb RAM, 1Gb/s Ethernet communication and communication interfaces. The communication interfaces including CAN, LIN and ETK enables the control communication with down-stream ECUs. Software and control functionalities can be implemented in ES910 and validated by HiL testing network, as shown in Fig. 3-12. CAN and LIN bus can be used to control downstream controllers using ES910 as a supervisory controller. ES910 can also be used to bypass a commercial ECU for specified functions development using ETK. In this study, ES910 is used as a supervisory HEV controller for functional validation of energy management strategies.

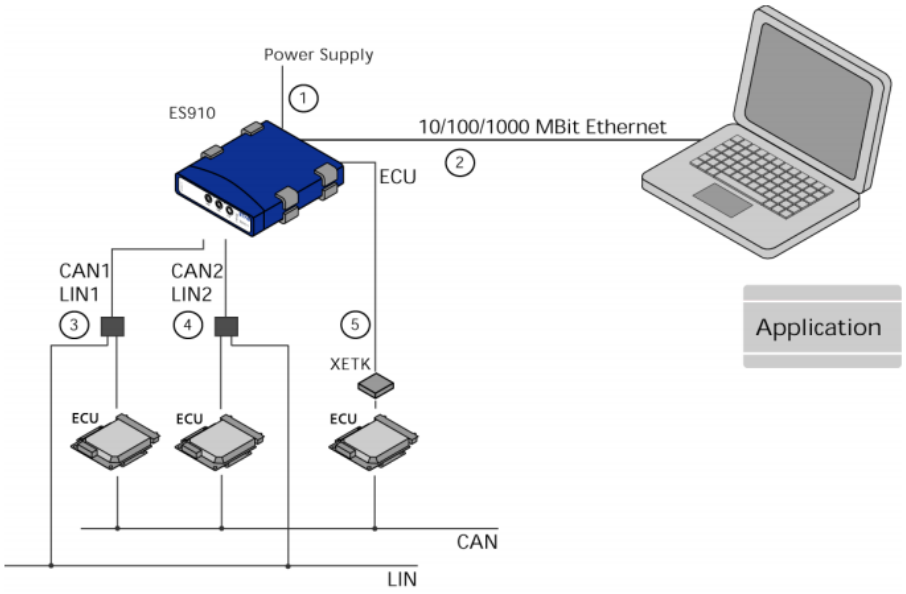


Fig. 3-12 The testing network of ES910

3.3.4 Software for the HiL testing

The software for the HiL testing is used mainly for modelling, compiling and implementation, as shown in Fig. 3-13. The software is installed in host-PCs which communicate with ETAS hardware via Ethernet. The real-time models for the hybrid vehicle and energy management controller are built by MATLAB/Simulink. The real-time models connected with the hardware interface (e.g. CAN) and compiled into C-code for real-time computing in the compiling procedure. ETAS EE and INTECRIO are the model compiling software used for LABCAR and ES910 respectively. Complied models of vehicle and controllers are implemented in the LABCAR and ES910 through ETAS Experiment Environment (EE) and INCA respectively. The model performance in LABCAR and ES910 can be monitored in the host-PCs

through EE and INCA.

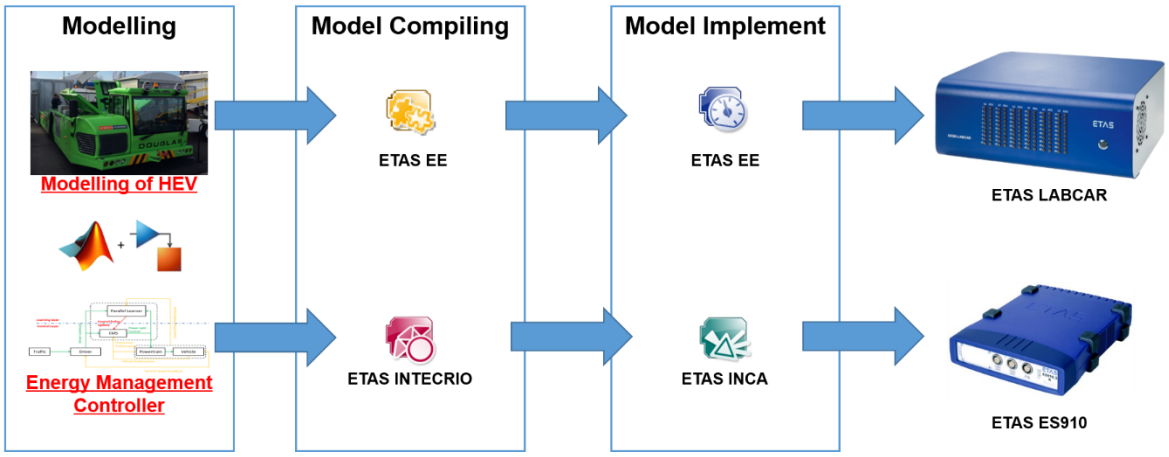


Fig. 3-13 Software for HiL testing

3.4 Real-time Modelling for the HiL Testing

Computing models are critical keys for today's vehicle development, especially for model-based design optimisation, model-based predictive control and real-time simulation for control functionality testing and validation. This chapter will develop the full real-time models of the hybrid powertrain system (including vehicle dynamics and power-flow, engine generator unit, electric motor, and lithium-ion battery package) mainly for Hardware-in-the-Loop (HiL) testing purpose. These models will also be used for design optimisation in Chapter 5, and its simplified version will be used for model-predictive control in Chapter 6.

3.4.1 Vehicle system dynamics and power-flow

The vehicle system is frontward modelled as in Fig. 3-14, which is built-in MATLAB/Simulink. The driving cycle used as the system input is obtained with the help of Douglas Equipment Ltd. from a real aircraft towing tractor working in London Heathrow. The driver model is a fuzzy-logic controller using the driving cycle speed as the target speed and generates the control signals to traction motor and brake with the feedback of real vehicle speed.

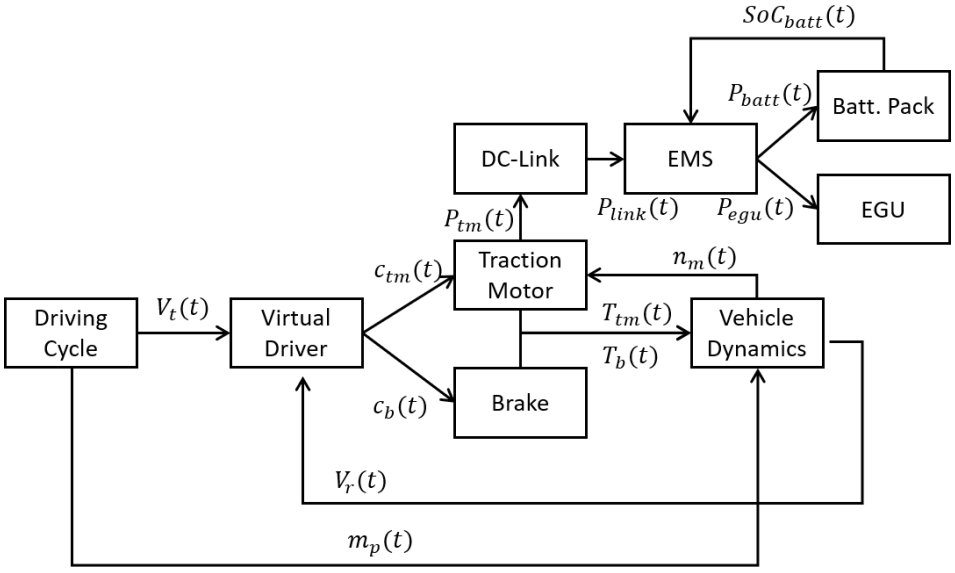


Fig. 3-14 Information-flow of the electrified aircraft-towing tractor model

The vehicle dynamics are modelled with newton's theory:

$$\left(m_{veh} + m_{plane}\right) \cdot \frac{dv_r}{dt} = \frac{(T_t - T_b)}{r_{whl}} - F_f - F_a - F_g \tag{3-1}$$

where, m_{veh} and m_{plane} are the vehicle mass and aeroplane mass respectively and v_r is the real vehicle speed; T_t and T_b are the torque generated by traction

motor and friction brake; F_f, F_a and F_g are the friction, air drag and resistance of gradient respectively, which obey:

$$\begin{cases} F_f = \left(m_{veh} + m_{plane}\right) \cdot f_f \cdot \cos\left(\alpha\right) \\ F_a = \frac{1}{2} \cdot \rho_{air} \cdot C_d \cdot A_f \cdot v_r^2 \\ F_g = \left(m_{veh} + m_{plane}\right) \cdot g \cdot \sin\left(\alpha\right) \end{cases} \tag{3-2}$$

The values of vehicle parameters used for modelling are presented in Table.3-3.

Table.3-3 Vehicle parameters

Parameter	Description	Value
m_{veh}	The mass of the aircraft-towing tractor	16t
r_{whl}	The radius of the wheels	0.75m
f_f	The friction coefficient	0.02
ρ_{air}	The density of the air	1.2258
C_d	Aerodynamic drag coefficient	0.8
A_f	Effective front area	6.8m^2
α	Gradient angle	0

The brake model is a 1-D lookup table using the data from AVL Cruise. The traction motor is a 3-phase permanent magnet synchronous motor, which is modelled in Section 3.4.4.

3.4.2 Driver model for speed control

A fuzzy logic controller is used to regulate the vehicle speed subjective to the desired

vehicle speed. The inputs of the driver model are the desired vehicle speed and the feedback of actual vehicle speed from the vehicle model. The outputs of the driver model are the torque demand to the electric motor and brake demand. These controllers use the linguistic terms Positive Large (PL), Positive Medium (PM), Positive Small (PS), Zero (Z), Negative Small (NS), Negative Medium (NM) and Negative Large (NL).

The membership functions for the FL speed controller are given in Fig. 3-15 and the rule base is given in Table.3-4. This controller uses the vehicle's velocity error ΔV_x at time t as the first input and its time derivative as the second input to generate a suitable pedal activation level. Positive controller outputs represent gas pedal activation levels, and negative outputs represent brake pedal activation levels.

$$\Delta V_x(t) = V_x^*(t) - V_x(t) \tag{3-3}$$

Table.3-4 Fuzzy logic speed controller rule base

		ΔV_x						
		NL	NM	NS	Z	PS	PM	PL
$\frac{\Delta V_x}{dt}$	NL	NS	Z	Z	PS	PM	PM	PL
	NM	NM	NS	Z	Z	PS	PM	PL
	NS	NM	NM	NS	Z	PS	PM	PL
	Z	NL	NM	NS	Z	PS	PS	PM
	PS	NL	NM	NS	Z	PS	PS	PM
	PM	NL	NM	NS	Z	Z	PS	PM
	PL	NL	NM	NM	NS	Z	Z	PS

An input range of $\pm$5kph in Fig. 3-15(a) is used for the controller's first input ΔV_x because this offers the best trade-off between a low steady-state velocity error and control realism. Reducing this input range improves the steady-state error of the vehicle, but it also resulted in erratic pedal control. The second input range $\pm$20kph/s in Fig. 3-15(b) for the time derivative of ΔV_x is used. Most typical drivers would accept $\pm$20kph/s to be a *large* acceleration value. A controller output with a magnitude 1 shows that the pedal is fully activated.

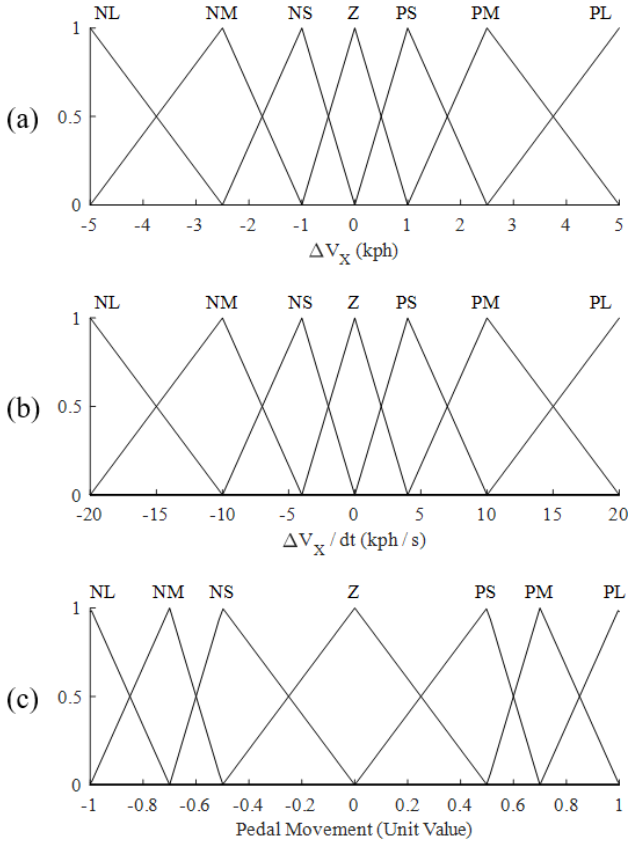


Fig. 3-15 Fuzzy Logic speed controller membership functions: (a) Input 1 - Velocity error, (b) Input 2 – Velocity error time derivative, (c) Output- Acceleration pedal movement

According to Table.3-4, some examples of the rule base for the fuzzy logic speed controller are explained in Fig. 3-16.

- If ΔV_x is (Z) and $\frac{\Delta V_x}{dt}$ is (Z) then **Pedal Movement** is (Z). If the vehicle is travelling at its target velocity and is not accelerating or decelerating, then no accelerator or brake pedal is required.
- If ΔV_x is (NL) and $\frac{\Delta V_x}{dt}$ is (NS) then **Pedal Movement** is (NM). Even though the vehicle is travelling much faster that its target velocity, the brake pedal does not need to be fully applied because the vehicle is already decelerating towards the target velocity.
- If ΔV_x is (PL) and $\frac{\Delta V_x}{dt}$ is (NS) then **Pedal Movement** is (PL). If the vehicle is travelling much slower than the target velocity and decelerating, the accelerator pedal is fully applied to bring the vehicle back up to the target velocity.

Fig. 3-16 Rule base for the fuzzy logic speed control

3.4.3 Engine generator unit

The engine-generator unit is modelled with the capability of flexible scaling to any size, which is centred around a Williams approximation (as shown in Fig. 3-17.) with a baseline internal combustion engine (Luján *et al.*, 2016).

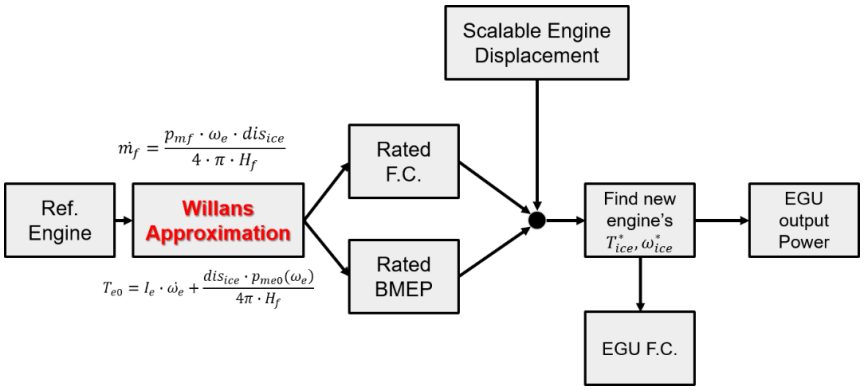


Fig. 3-17 Diagram of the engine-generator unit model

The mean effective cylinder pressure is approximated by an affine function of mean effective fuel pressure p_{mf},

$$p_{mf} = e(n_{ice}) \cdot p_{mf} - p_{me0}(n_{ice}) \tag{3-4}$$

where, $e(n_{ice})$ is the indicated efficiency and $p_{me0}(n_{ice})$ is the mean effective pressure loss due to friction, gas exchange, and auxiliary devices; the total torque generated by the engine is calculated as:

$$T_{e0} = I_{ice} \cdot \dot{n_{ice}} + \frac{dis_{ice} \cdot p_{me0}(n_{ice})}{4\pi \cdot H_f} \tag{3-5}$$

where, I_{ice} is the inertia of the engine; dis_{ice} is the displacement of the engine; n_e is the rotation speed of the crankshaft; H_f is the heat value of the fuel. The fuel consumption $\dot{m}_f$ is calculated by:

$$\dot{m}_f = \frac{p_{mf} \cdot \omega_e \cdot dis_{ice}}{4 \cdot \pi \cdot H_f} \tag{3-6}$$

Using Equation 4-6 and 4-7 the engine torque and fuel consumption could be scaled by its displacement. And for the generator, this report assumes that it works on a steady-state and the power generating power could be calculated by:

$$P_{egu} = \frac{T_{ice}^* \cdot n_{ice}^*}{9.55} \tag{3-7}$$

where, T_{ice}^* and n_{ice}^* are the most efficient torque and speed of the internal combustion engine, respectively. The parameters of the baseline engine-generator used for engine generator unit modelling are listed in Table.3-5.

Table.3-5 Parameters of the baseline engine

Parameter	Description	Value
L_{ice}	Engine displacement (L)	6.8
P_{ice_max}	Rated power of baseline engine (kW)	157
n_{ice_max}	Rated speed of baseline engine (rpm)	2750
Num_{ice_cl}	Number of cylinders	6
$\dot{m}_f$	Engine fuel mass-flow rate (kg/s)	Map data[1]
I_{ice}	The inertia of baseline engine (kg*m^2)	1

3.4.4 Electric motor

A permanent magnet synchronous motor (PMSM) is used to propel the vehicle. The inputs of the PMSM model is the torque demand from the drive T_{tm_req} and equivalent motor speed ω_m in rad/s. The output of the PMSM model is the required electric power from the battery. The PMSM is modelled as two systems, including mechanical system and electric system.

The mechanical system of the electric motor is used to calculate the motor speed ω_m:

$$\omega_m = \int \frac{1}{I_m}\left(T_e - T_f - f_m \cdot \omega_m - T_m\right) \tag{3-8}$$

where, I_m is combined inertia of rotor and load; T_f is the rotor shaft static friction torque; f_m is combined viscous friction of rotor and load; T_m is the output torque of

[1] Map data using engine speed and engine torque as input to obtain the fuel rate which is provided by the engine supplier

the motor; T_e is electromagnetic torque, which can be calculated using the electric system as (Cash *et al.*, 2018b):

$$T_e = 1.5 \cdot n_{po} \cdot \left[\lambda_{pm} \cdot i_q + (L_d - L_q) \cdot i_d \cdot i_q\right] \tag{3-9}$$

where, n_{po} is the number of pole pairs; λ_{pm} is permanent magnet flux linkage; L_d is direct axis inductances; L_q is quadrature axis inductances; i_d and i_q are direct axis current and quadrature axis current respectively, which obeys (Cash *et al.*, 2018b):

$$\begin{cases} \frac{d}{dt} i_d = \frac{V_d}{L_d} - \frac{R_{tm\_s}}{L_d} i_d + \frac{L_q}{L_d} \cdot n_{po} \cdot \omega_m \cdot i_q \\ \frac{d}{dt} i_q = \frac{V_q}{L_q} - \frac{R_{tm\_s}}{L_q} i_q - \frac{L_q}{L_d} \cdot n_{po} \cdot \omega_m \cdot i_q - \frac{\lambda_{pm} \cdot n_{po} \cdot \omega_m}{L_q} \end{cases} \tag{3-10}$$

where, R_{tm_s} is the resistance of the stator windings; ω_m is the rotation speed of the motor rotor.

The torque generated by the motor is controlled by a proportional-integral (PI) controller with relevant i_d and i_q, which are calculated based on the driver's torque requirement T_{tm_req} and the feedback of actual electromagnetic torque T_e. Motor parameters provided by the industrial partner for modelling are list in the table below:

Table.3-6 Motor parameters

Parameter	Description	Value
n_{po}	Number of poles	8
P_{tm_max}	Rated power of traction motor (kW)	245
n_{tm_max}	Rated speed of traction motor (rpm)	3375
V_{tm_s}	Rated voltage of traction motor (V)	155
I_{tm_s}	Rated current of traction motor (A)	450
R_{tm_s}	Rated stator resistance of motor (omega)	0.0083
λ_{pm}	Permanent magnet flux linkage (Wb)	0.071
L_d	d-axis inductance (mH)	0.174
L_q	q-axis inductance (mH)	0.293
I_{tm_m}	Rotor inertia of traction motor (kg*m^2)	0.5

3.4.5 Lithium-ion battery package

The inputs of the battery package model are: the number of battery cells n_{bc} and the required power P_{bp} from the DC link. Battery package model outputs the battery state of charge to the hybrid system for energy management control. Battery cell current and voltage are used for the iterative calculation to simulate the dynamics of battery cells. From the beginning of each iteration, the battery cell current I_{bc} is firstly calculated by:

$$I_{bc} = \frac{P_{bp}}{n_{bc} \cdot V_{bc}} \tag{3-11}$$

where, V_{bc} is the available voltage which can be provided by the battery cell. A two

R-C equivalent battery model is used to model the current-voltage dynamics of a lithium-ion battery cell, as shown in Fig. 3-18.

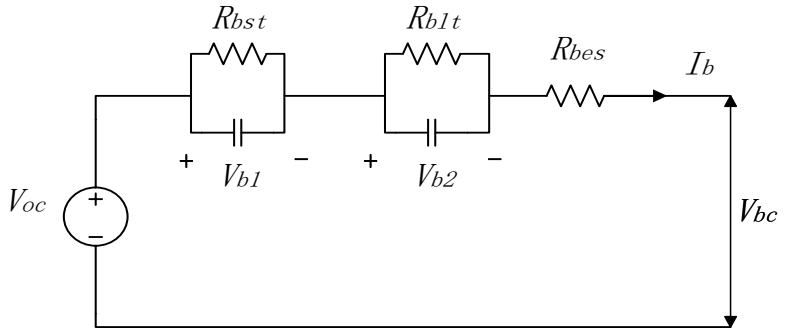


Fig. 3-18 Electric model of the battery cell

The battery's voltage dynamics obey (Hu *et al.*, 2014):

$$\begin{cases} V_{bc} = V_{oc}(SOC) - V_{b1} - V_{b2} - R_{bes}(SOC)I_b \\ C_{bst}(SOC) \cdot \frac{dV_{b1}}{dt} = I_b - \frac{V_{b1}}{R_{bst}(SOC)} \\ C_{blt}(SOC) \cdot \frac{dV_{b2}}{dt} = I_b - \frac{V_{b2}}{R_{blt}(SOC)} \end{cases} \tag{3-12}$$

where, R_{bes}, R_{bst}, and R_{blt} are the effective series resistance; the short transient resistance and long transient resistance respectively and all of them are a function of the battery cell's state of charge (SoC); C_{bst} and C_{blt} are the short transient capacity and long transient capacity, respectively, which are functions of SoC; The SoC of the battery cell is calculated by:

$$SOC = SOC_0 - \int_0^t \frac{I_b}{Q_b} dt \tag{3-13}$$

The battery cell data and calibrated model parameters are provided in Table.3-7.

Table.3-7 Battery cell parameters

Parameter	Description	Value
V_{bc_max}	Rated battery cell voltage (V)	3.6
I_{bc_cmax}	Rated battery cell charge current (A)	2.25
I_{bc_dmax}	Rated battery cell discharge current (A)	11.25
Q_{bc_max}	Rated battery cell capacity (mAh)	2900
V_{oc}	$1.031 \cdot e^{-35 \cdot SOC} + 3.685 + 0.2156 \cdot SOC - 0.1178 \cdot SOC^2 + 0.3201 \cdot SOC^3$	
R_{bes}	$0.1562 \cdot e^{-24.37 \cdot SOC} + 0.07446$	
R_{bst}	$0.3208 \cdot e^{-29.14 \cdot SOC} + 0.04669$	
R_{blt}	$6.603 \cdot e^{-155.2 \cdot SOC} + 0.04984$	
C_{bst}	$-752.9 \cdot e^{-13.51 \cdot SOC} + 703.6$	
C_{blt}	$-6056 \cdot e^{-27.12 \cdot SOC} + 4475$	

3.5 Summary

This chapter introduces the methodology and facilities for the research. The main work which has been carried out in relevant to this chapter are:

- The technical route for the overall research has been developed and introduced in this chapter. The overall design objective is specified into four sub-tasks: real-time modelling, the hardware-in-the-loop test, the design optimisation, and the real-time energy management control.

- This chapter discussed the methodology for design optimisation, which stresses the main challenges and the specified procedures for offline optimisation methodology development.

- The real-time energy management of hybrid electric vehicle is defined as an online optimisation process in this chapter. Model-based predictive control method will be investigated, and the research will focus on advanced online programming method for online optimisation. Reinforcement learning method will be then researched to enable further optimisation of the energy management control.

- This chapter described the work of HiL testing setup. The main facilities used for the research are also introduced.

- Real-time models have been developed for the design optimisation, the optimal control, and the HiL test.

Chapter Four

OFFLINE OPTIMISATION OF COMPONENTS' SIZE AND CONTROL PARAMETERS

The development of the electrified aircraft towing tractor starts from offline optimisation of component size and energy management control parameters. The offline optimisation is formulated firstly as a multi-objective integer optimisation via mathematical modelling of the vehicle powertrain system. A new algorithm is developed to enhance the performance by upgrading the conventional APSO algorithm to chaos-enhanced accelerated particle swarm Optimisation (CAPSO) algorithm (Zhou *et al.*, 2017). Monte Carlo analysis and reputation evaluation are carried out to investigate the optimisation performance. Pareto analysis is also carried out to explore the 'trade-off' phenomena using different weighting factor. Finally, the optimum design parameters are secured by a two-stage optimisation.

4.1 Optimisation Objectives and Constrains

4.1.1 Objective functions

The offline optimisation aims to determine the optimal combination of design

parameters (including components' size and the energy management control parameters), to achieve the maximum energy efficiency over a selected driving cycle, with the minimum component size. This will determine the optimum size of the engine-generator (represented by displacement of the engine) and battery package (represented by the number of battery cells). The control parameter to be optimised is a coefficient in the function for the battery's SoC-dependent power distribution control.

The mathematical expression of the offline optimisation can be formulated as:

$$[L_{ice}^{*} \quad n_{bc}^{*} \quad c_{ems}^{*}] = \arg\min(J_{loss} \quad J_{size})$$

$$\text{s.}t.\begin{cases} L_{ice}^{-} \le L_{ice} \le L_{ice}^{+} \\ n_{bc}^{-} \le n_{bc} \le n_{bc}^{+} \\ c_{ems}^{-} \le c_{ems} \le c_{ems}^{+} \\ n_{bc} \ge \max\left[ceil\left(\frac{P_{max}-L_{ice}\cdot P_{ice}\cdot \eta_{egu}}{P_{bcn}\cdot \eta_{bpn}}\right), ceil\left(\frac{E_{min}}{P_{bcn}\cdot \eta_{bpn}}\right)\right] \end{cases} \tag{4-1}$$

where, L_{ice} is the displacement of the engine for the engine generator and L_{ice}^{*} is the optimal engine displacement; n_{bc} is the number of battery cells for the battery pack and n_{bc}^{*} is the optimal size of battery package; c_{ems} is the coefficient for the battery's SoC-dependent power distribution function, and c_{ems}^{*} is the optimal coefficient value; L_{ice}^{-}, n_{bc}^{-}, and c_{ems}^{-} represent the lower boundary of the design parameters, whereas L_{ice}^{+}, n_{bc}^{+}, and c_{ems}^{+} represent the higher boundary of the design parameters; P is the power requirement while E is the energy requirement; P and E are calculated using a vehicle model; J_{loss} is the total energy loss of the

HEV system over a selected driving cycle and J_{size} is the total dimension of the powertrain components.

The first optimisation objective J_{loss} is to minimise the total energy loss over a selected driving cycle. This loss can be calculated using the measurement of the fuel consumption of the engine generator, and the measurements of the voltage, the current and the SoC of the battery package as in the following Equation (Murgovski *et al.*, 2012a):

$$J_{loss} = \int_{t_0}^{t_p} P_{fuel}(t) \cdot dt + \int_{t_0}^{t_p} (V_{est}\big(SoC(t)\big) - V_r(t)) \cdot I_r(t) \cdot dt \qquad (4\text{-}2)$$

where $\mathrm{SoC}(t)$, $V_r(t)$, $I_r(t)$ are the real-time SoC (State-of-Charge), voltage and current of the battery package; V_{est} is the estimated battery source voltage, which is a function of the battery's SoC; $P_{fuel}(t)$ is the equivalent power for engine-generator fuel usage.

As the overall size of the vehicle is limited, the second optimisation objective considers the size of the overall components, which is formulated as:

$$J_{size} = vol_{egu}(L_{ice}^{*}) + n_{bc} \cdot vol_{bc} \qquad (4\text{-}3)$$

where, G_{gpu} indicates the size gain value, which represents the linear relationship between the engine displacement and the engine generator's size; vol indicates the volume of each battery cell. The influence of the components size and control parameter on the design optimisation objectives will be discussed in the following

sections for component scaling (4.1.2) and power distribution function (4.1.3).

4.1.2 Component scaling

The components required to be sized in this research includes an engine-generator set and battery package. The engine-generator set will use the data provided by JCB, and the battery pack will be made up of several Litmus-ion cells from Panasonic.

a) Scaling of the engine-generator

JCB is a supplier providing hundreds of types of engine-generator sets from small scale to large scale. It is necessary to model the relationship between the overall size of the engine generator and its engine displacement. A diagram of an engine generator and the definition of its dimensional parameters are shown in Fig. 4-1. The dimensional parameters of five different types of engine-generators used for modelling are listed in Table.4-1.

Table.4-1 Dimensional parameters of engine-generator sets

Engine Size (L)	W (mm)	H (mm)	D (mm)
2.2	1948	1423	835
3.4	2265	1567	950
4.4	2850	1850	1140
4.8	3334	1912	1200
6.8	3800	2033	1270

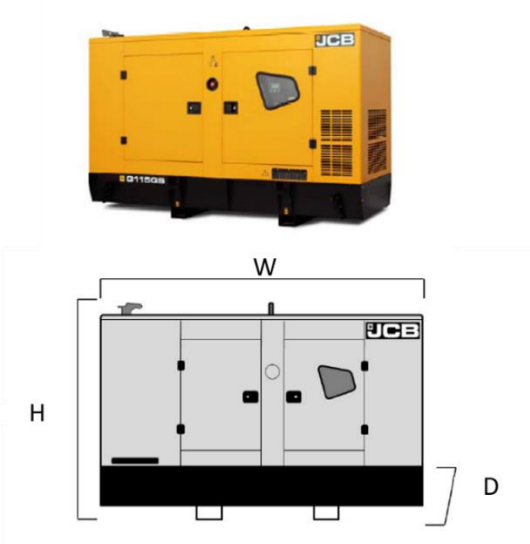


Fig. 4-1 The engine-generator set and its geographic parameters

In this research, a logistic function is used to map the nonlinear relationship between engine displacement and the volume of an engine generator as:

$$vol_{egu}(L_{ice}) = \frac{1}{c_{e1}+e^{-(c_{e2}\cdot L_{ice}+c_{e3})}} \tag{4-4}$$

where, vol_{egu} is the total volume of the engine generator; L_{ice} is the displacement of the engine; c_{e1}, c_{e2}, c_{e3} are the tuning parameters of the logistic function. The tuning parameters are determined using the MATLAB data fitting toolbox with the original data in Table.4-1, the optimal value of which are listed in Table.4-2.

Table.4-2 Parameters for the function of engine-generator size

Parameter	Value
c_{e1}	9.5e-5
c_{e2}	0.9006
c_{e3}	5.685

Two curve-fitting methods, including the widely used linear method and the proposed logistic method, are compared with the original data in Fig. 4-2.

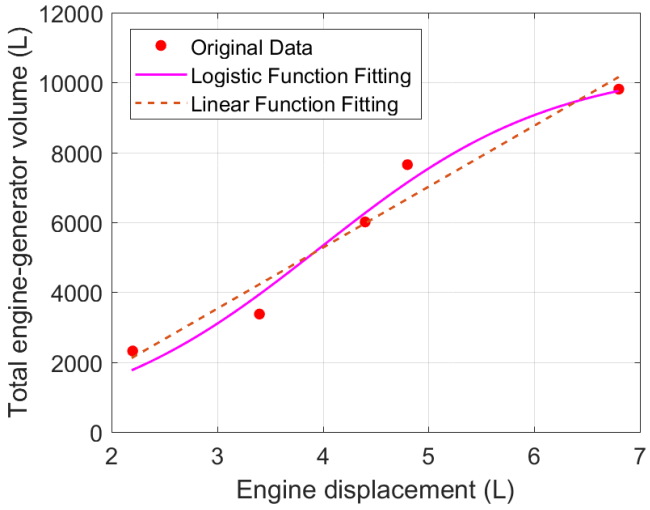


Fig. 4-2 Fitting results comparison of engine-generator size

According to the goodness of fit of both fitting functions as shown in Table.4-3, the logistic function has a better fitting performance than the linear function because it has a higher R-square value, and a lower value of root mean square error (RMSE) and squared errors of expectation (SEE).

Table.4-3 The goodness of fit for logistic function and a linear function

Indicates	Logistic	Linear
R-square	0.975	0.951
RMSE	484.4	784.6
SSE	0.938e6	1.847e6

The fuel consumption $fuel_{egu}$ of the selected engine-generators at different power

rates r_{egu} are illustrated in Table.4-4. Five engine generator sets are selected, and their fuel consumptions at five different power rates from 0~100% are measured by the supplier. These data are used to model the equivalent power of the fuel consumption of different selected engine generators for scaling.

Table.4-4 Fuel consumption of the selected engine-generator

	Fuel consumption at different power rates u_{egu} (L/h)			
Max. Power (kW)	25%	50%	75%	100%
15	2.60	4.40	6.30	8.60
32	3.30	5.70	8.10	11.10
86	6.90	13.00	18.60	25.80
100	7.80	14.50	21.10	29.00
120	9.80	17.80	25.40	34.80

In this research, the fuel consumption is modelled as a first-order function of power rate u_{egu} and a second-order function of the maximum output power of the selected engine-generator unit P_{egu_max} (Murgovski *et al.*, 2012a):

$$fuel_{egu}(r_{egu}, P_{egu\_max}) = c_{f00} + c_{f10} \cdot u_{egu} + c_{f01} \cdot P_{egu\_max} + c_{f11} \cdot u_{egu} \cdot P_{egu\_max} + c_{f02} \cdot {P_{egu\_max}}^2 \quad (4\text{-}5)$$

where, c_{f00}, c_{f10}, c_{f01}, c_{f11}, and c_{f02} are the model parameters, the values of which are calibrated with the MATLAB curve fitting toolbox as in Table.4-5.

Table.4-5 Parameters of the scaled fuel consumption model

Parameter	Value
c_{f00}	9.032
c_{f10}	38.71
c_{f01}	-0.352
c_{f11}	2.566
c_{f02}	2.822e-3

The fuel consumptions of different engine generators at different power rates compared with the original testing data are shown in Fig. 4-3.

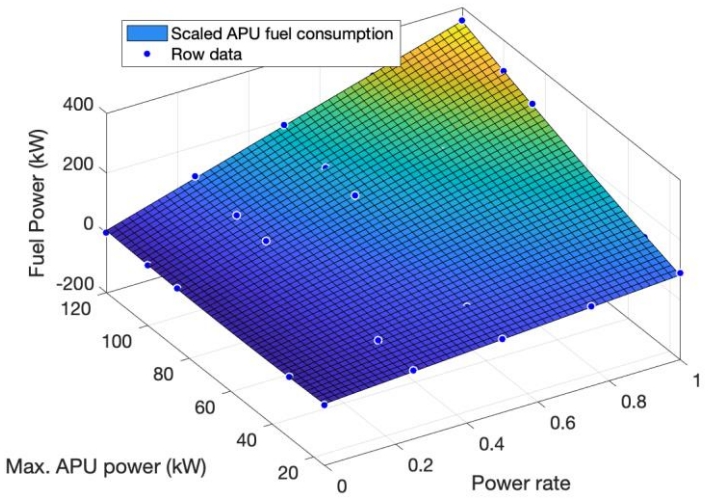


Fig. 4-3 Fuel consumption model vs the testing data

According to the goodness of fit assignment result as shown in Table.4-6, the scaled fuel-consumption model is acceptable to represent the equivalent power of fuel consumption at different power rates for different engine generators.

Table.4-6 The goodness of fit for the scaled engine-generator model

Indicates	Value
R-square	0.9978
RMSE	5.25
SSE	551.2

b) Scaling of the battery package

The Panasonic NCR-18650 series battery cell, as shown in Fig. 4-4 is used to build the battery pack. The V-I dynamics of the battery package have been discussed in Chapter 3.4.5. The size of the battery package vol_{batt} is scaled by the number of Lithium-ion battery cells as:

$$vol_{batt}(n_{bc}) = n_{bc} \cdot 4 \cdot {r_{cell}}^2 \cdot h_{cell} \tag{4-6}$$

where, r_{cell} is the radius of the battery cell and h_{cell} is the height of the battery cell.

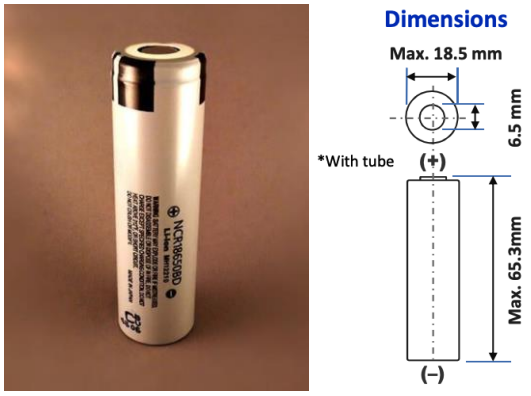


Fig. 4-4 Panasonic NCR-18650 series battery cell

4.1.3 Power distribution function

The power supplied by the engine generator and battery is calculated as:

$$\begin{cases} P_{egu} = u_{egu}(SoC) \cdot P_{egu\_max} \\ P_{batt} = P_{req} - P_{egu} \end{cases} \quad (4\text{-}7)$$

where, P_{batt} is the power supplied by the battery pack and P_{egu} is the power supplied by the engine generator; P_{egu_max} is the maximum power that can be provided by the engine generator; P_{req} is the required power for driving the traction motor and powering the onboard auxiliary devices.

The power distribution between the battery P_{batt} and engine P_{egu}, is controlled by the engine generator using a power distribution function $u_{egu}(SoC)$ of the battery's SoC level (Huang, Wang, *et al.*, 2017).

$$u_{egu}(SoC) = \begin{cases} 0 & SoC\epsilon(0.8\ 1] \\ round(20 \cdot e^{\left(-\frac{(SoC-SoC^{-})^2}{2 \cdot c_{ems}{}^2}\right)})/20 & SoC\epsilon[0.2\ 0.8) \\ 1 & SoC\epsilon[0\ 0.2) \end{cases} \quad (4\text{-}8)$$

Where, the core of the power distribution function is a rounded exploration function, which rounds the control command to a number with a resolution of 0.05; SoC is the current battery's SoC; SoC^{-} is the lower battery SoC boundary, normally $SoC^{-} = 0.2$ is chosen to ensure battery safety; c_{ems} is the control parameter to be optimised for power distribution control, power distribution functions with different control

parameter values are shown in Fig. 4-5.

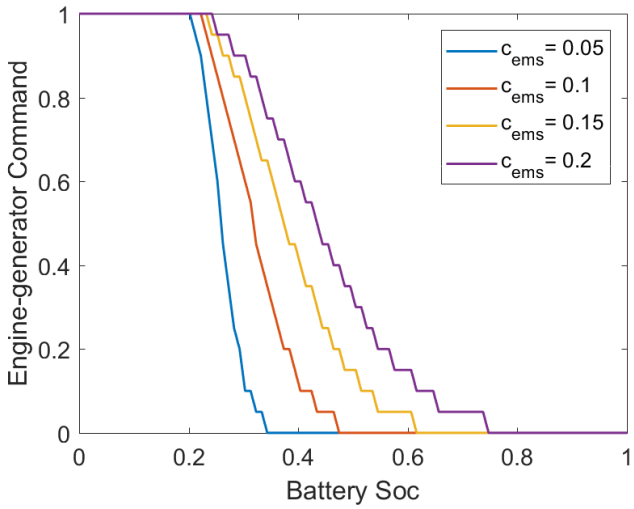


Fig. 4-5 Power distribution function with different control parameter value

4.2 Optimisation with the conventional APSO

4.2.1 Optimisation algorithm

Particle Swarm Optimisation algorithms need fewer tuning parameters and less computational effort. They also have the capability of dealing with integer variables. Furthermore, the convergence speed of PSO can be improved by the accelerated particle swarm optimisation (APSO) algorithm. A standard CAPSO is used as the base-line method for design optimisation. The process of design optimisation using the APSO is illustrated in Table.4-6, which consists of three main procedures: 1) initialization, 2) main iteration, and 3) co-simulation with MATLAB/Simulink. The

procedure of initialization defines the position of the particles as well as the constraints of their search space. The optimisation algorithm iterates with the vehicle model for several rounds in the main iteration procedure to determine the optimisation results with the system performance feedbacks from the co-simulation with the MATLAB/Simulink model. Details of each procedure are described as follows.

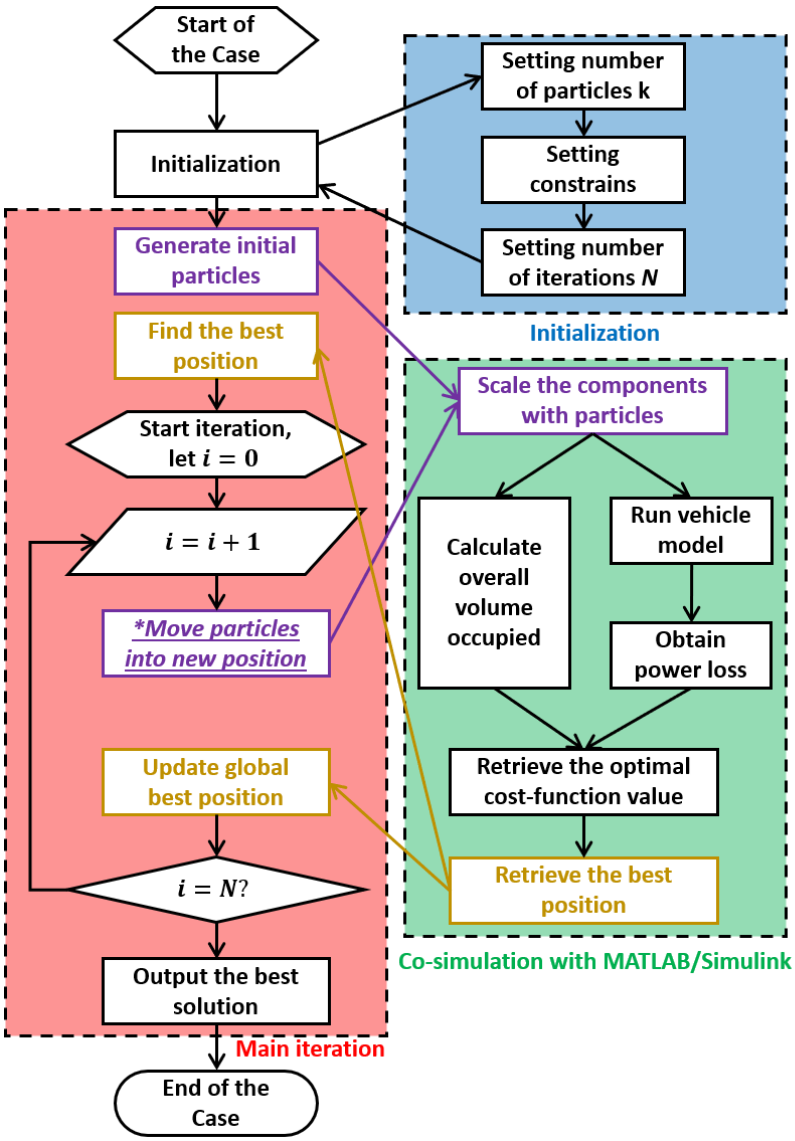


Fig. 4-6 Flow-chart of design optimisation with APSO

a) Initialization

In the initialization procedure, the particles (computing agent) are defined using a Euclidean coordination system as:

$$x_{i,j} = [L_{ice} \quad n_{bc} \quad c_{ems}]^T$$
$$i = 0,1,2,\dots,N;\ j = 1,2,..,p \tag{4-8}$$

where, $x_{i,j}$ is the position of the j^{th} particle at i^{th} iteration; the coordinate of the particle position is represented by the engine displacement, number of battery cells and energy management control parameter respectively; N is the maximum number of iterations, which is used to terminate the iterations; p is the population of the particles, which defines the capability of the global search in each iteration via computing agents.

b) Main iteration

The first step in the main iteration is to generate initial particles randomly within the search area and use the particles to determine their objective function values at their current positions. The objective function value is the call-back from the co-simulation with the vehicle model in MATLAB/Simulink. The best position in the initial trial is then obtained by:

$$x_{0,*} = \arg min. J(x_{0,:}) \tag{4-9}$$

where, $x_{0,*}$ is the best position in the initial trial; $J(x_{0,:})$ is the cost function, which is

modelled as a weight sum function:

$$J(x_{0,:}) = \left(w \cdot \frac{J_{loss}(x_{0,:})}{J^*_{loss}} + (1-w) \cdot \frac{J_{size}(x_{0,:})}{J^*_{size}}\right) \tag{4-10}$$

where w is the weight factor of each objective; J^*_{loss} and J^*_{size} are the scale factor in ensuring the values of the objective functions are on the same scale (normally between 0 and 1); $J_{loss}(x_{0,:})$ and $J_{size}(x_{0,:})$ are the objective function values calculated with all the particles in the initial trial $x_{0,:} = [x_{0,1}, x_{0,2}, \dots, x_{0,p}]$.

After the initial trial, two dummy sets are defined as:

$$\begin{cases} \boldsymbol{X} = [x_{0,*}, x_{0,*}, \dots, x_{0,*}] \\ \boldsymbol{Y} = [J(x_{0,*}), J(x_{0,*}), \dots, J(x_{0,*})] \end{cases} \tag{4-11}$$

where $\boldsymbol{X}: \mathcal{R}^{3\times N}$ is the local best variable set, which will be updated at the end of each iteration; $\boldsymbol{Y}: \mathcal{R}^{3\times N} \to \mathcal{R}^{1\times N}$ is a local best cost-function value set to allocate the cost-function values with the local best of each iteration. With the local best variable set and local best objective value set, the global best position can be calculated as:

$$g_{bst} = \arg\min F(X) \in X \tag{4-12}$$

The following iteration starts at a random move of the particles. The APSO algorithm updates the position of the particles using:

$$\begin{gathered} x_{i+1,j} = x_{i,j} + \beta \cdot (g_{bst} - x_{i,j}) + \alpha^{i+1} \cdot r_{i,j} \\ i = 0,1,2, \dots, N;\ j = 1,2,..,p \end{gathered} \tag{4-13}$$

where, $x_{i+1,j}$ and $x_{i,j}$ are the position of the j^{th} particles at $i+1^{th}$ and i^{th}

iteration respectively; g_{bst} is the global best position; $r_{i,j}$ is a unique random number that is different for each particle in each iteration; α^{i+1} is a decreasing factor in reducing the influence of random move; β is the attraction factor which controls how the global best position will attract the moving of each individual particle. For the APSO algorithm, the value of the attraction factor is fixed, and $\beta = 0.5$ is usually used (Chopard and Tomassini, 2018).

A series of new cost function $J(x_{i+1,:})$ values are then determined with the new particle positions $x_{i+1,:}$ and the local optimal solution at $i+1^{th}$ iteration is then obtained by:

$$x_{i+1,*} = \arg min. J(x_{i+1,:}) \tag{4-14}$$

At the end of each iteration, the local sets of variables and objective function values will be updated as:

$$\begin{cases} \boldsymbol{X}(i+1) = x_{i+1,*} \\ \boldsymbol{Y}(i+1) = J(x_{i+1,*}) \end{cases} \tag{4-15}$$

where $\boldsymbol{X}(i+1)$ and $\boldsymbol{Y}(i+1)$ denote the $i+1^{th}$ element of the vector $\boldsymbol{X}$ and $\boldsymbol{F}$, which are updated with the value of $x_{i+1,*}$ and $J(x_{i+1,*})$. Therefore, the global best position value g_{bst} will also be updated with the new set of variables and objective function values, which will guide the particles moving in the next iteration.

c) Co-simulation with powertrain model in MATLAB/Simulink

A co-simulation platform, including the optimisation algorithm and vehicle simulation models, is established to perform the design optimisation, as shown in Fig. 4-7.

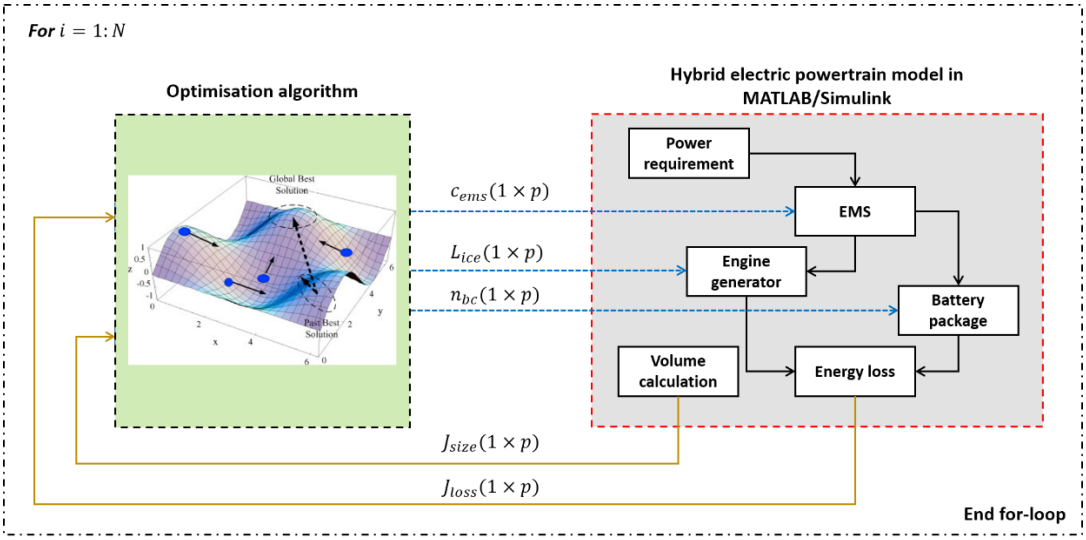


Fig. 4-7 The interface of the co-simulation platform for design optimisation

In each iteration, the inputs of the vehicle model are the control parameter for energy management strategy $\boldsymbol{c}_{ems} \in \mathcal{R}^p$, the number of battery cells $\boldsymbol{n}_{bc} \in \mathcal{R}^p$, and the displacement of the internal combustion engine $\boldsymbol{L}_{ice} \in \mathcal{R}^p$. The vehicle model in Simulink runs the simulation of p cases in parallel at each iteration and outputs the total power loss $\boldsymbol{J}_{loss} \in \mathcal{R}^p$ and total components' size $\boldsymbol{J}_{size} \in \mathcal{R}^p$. The power requirement profile is obtained separately by a forward-looking vehicle model (developed in Chapter 4) using the PBDC-I (as discussed in Chapter 3). The outputs are also a p-dimension vector, which is used to 1) retrieve the local best particle position, 2) update the global best position; 3) move all the particles to new locations

for the next round iteration. The iteration ends when the pre-set condition ($i = N$) is met, and the algorithm will output the optimal sizing information in the last iteration. The pseudo-code for the design optimisation is provided in Fig. 4-8.

Intelligent design optimisation using APSO

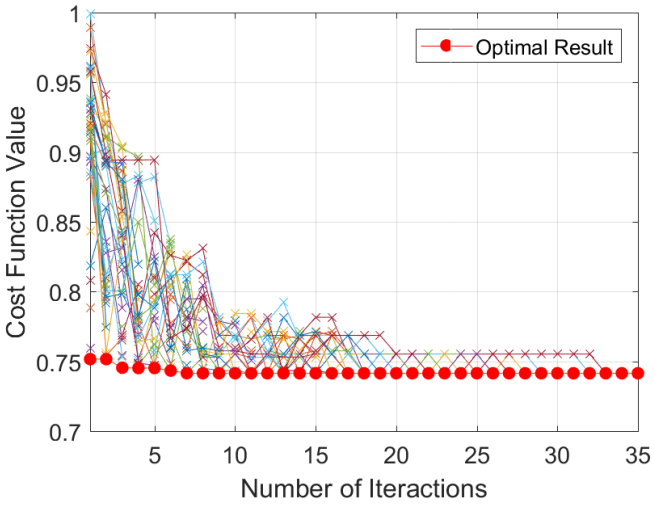
- *Load vehicle basic parameters and driving cycle;*
- *Setting APSO parameters* β*=0.5,* $\alpha = 0.98$**;**
- *Initialization position* $x_{0,j}$**;**
- *Run Simulink mode of HEV;*
- *Find optimal position in the initial iteration* $x_{0,*}$
- *Define best sets* $\boldsymbol{X} = [x_{0,*}, x_{0,*}, \dots, x_{0,*}]$, $\boldsymbol{F} = [J(x_{0,*}), J(x_{0,*}), \dots, J(x_{0,*})]$;
- *Obtain global best position:* $g_{bst} = \arg\min F(X) \in X$
- *For* $i = 1{:}N$
  - *Update particle position with eq.5-13;*
  - *Run Simulink mode of HEV;*
  - *Calculate cost function value J*
  - *Find optimal position in the initial iteration* $x_{i,*}$ $= \arg min. J(x_{i+1,:})$;
  - *Update best sets* $\boldsymbol{X}(i) = x_{i,*}$, $\boldsymbol{F}(i) = J(x_{i,*})$;
  - *Update global best position:* $g_{bst} = \arg\min F(X) \in X$;
- *End for*
- *Output the final optimisation results*

Fig. 4-8 Pseudo-code for design optimisation using APSO

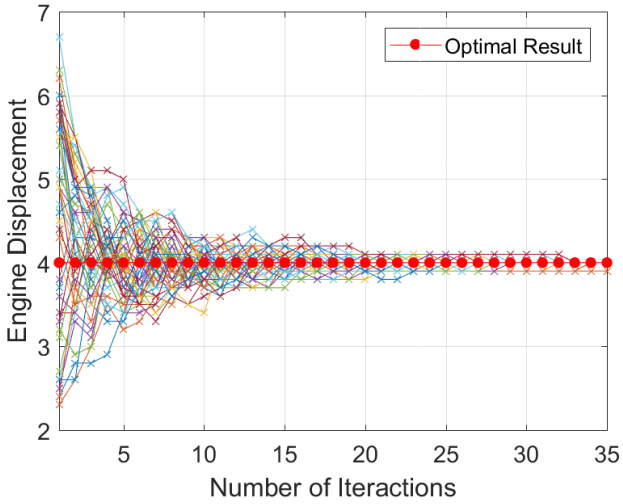
4.2.2 Optimisation results with the APSO

With the APSO algorithm using 50 particles and terminating at the 35th iteration, the evolution of the optimisation process of a single trial is present in Fig. 4-9. The optimisation is performed in MATLAB version 2017a using a desktop computer (with i5 processor and 8G RAM), which requires a total computing time of 282 seconds (around 5 minutes) to obtain an acceptable design optimisation result. In each subplot of Fig. 4-9, the red round line shows trajectories of the optimal value and the

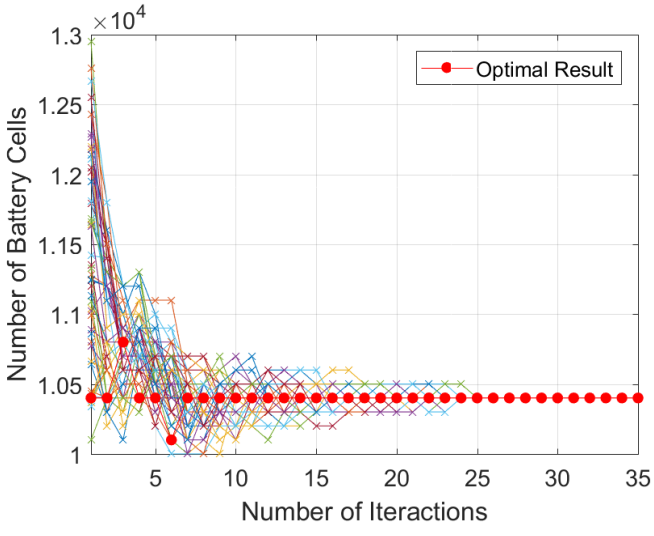
respective optimal component size in each iteration while the other lines are the trajectories of other particles. The convergence of the cost function values is shown in Fig. 4-9 (a). Fig. 4-9 (b), (c) and (d) presents the convergence of the three design variables which represent the positions of the particles. Using the APSO algorithm, the optimisation can converge to an optimal result within 35 iterations.



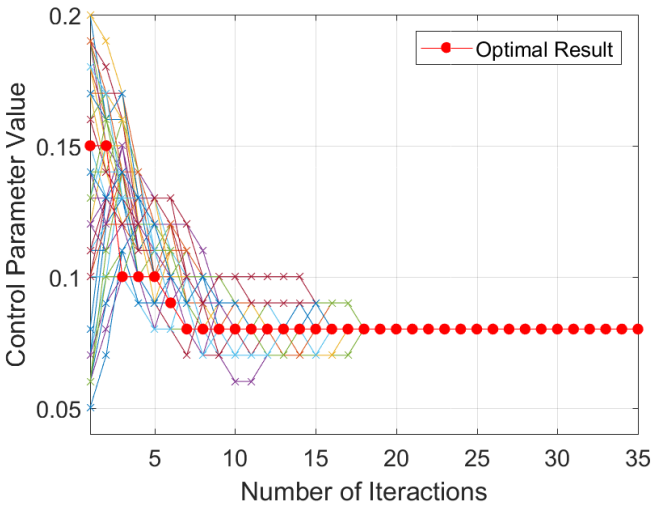
(a)



(b)



(c)



(d)

Fig. 4-9 Evolution of the optimisation results with APSO

4.2.3 Diversity of the optimisation results by the APSO

To evaluate the consistency of the optimisation results, 20 repeated experiments have been carried out, and the optimisation results are listed in Table.4-7. According to the tests, the cost function value of 0.7414 and 0.73942 are achieved for 15 times and 2 times respectively, whereas the global optimal value should be 0.7374 which was only obtained for 3 times. The diversity of the cost-function value obtained in each trial is caused by the particles converged to a local position.

Table.4-7 Diversity of optimisation results by APSO

No. of Test	Cost function Value	Engine disp.	No. of cells	Ctrl. parameter
1	0.741410742	4	10400	0.08
2	0.741410742	4	10400	0.08
3	0.741410742	4	10400	0.08
4	0.741410742	4	10400	0.08
5	0.741410742	4	10400	0.08
6	0.741410742	4	10400	0.08
7	0.739426271	4	10900	0.07
8	0.741410742	4	10400	0.08
9	0.741410742	4	10400	0.08
10	0.741410742	4	10400	0.08
11	0.741410742	4	10400	0.08
12	0.739426271	4	10900	0.07
13	0.737496805	4	12100	0.06
14	0.741410742	4	10400	0.08
15	0.741410742	4	10400	0.08
16	0.741410742	4	10400	0.08
17	0.741410742	4	10400	0.08
18	0.737496805	4	12100	0.06
19	0.737496805	4	12100	0.06
20	0.741410742	4	10400	0.08

For the APSO algorithm, the convergence of particles is controlled by two vectors including a random vector $\overrightarrow{V_a}$ and a vector towards the global best position $\overrightarrow{V_b}$, as shown in Fig. 4-10. The magnitude of the random vector $\overrightarrow{V_a}$ decreases as the

progression of the iteration and the vector towards the global best position $\overrightarrow{V_b}$ is a fixed proportion of the distance between the current position and the global best position.

At the initial iterations, the distance between the particles' current position and the global best position may be very long, and the vector $\overrightarrow{V_b}$ will significantly affect the direction of the particles' move. As the iteration progresses, although the magnitude of $\overrightarrow{V_b}$ decreases, the magnitude of $\overrightarrow{V_a}$ is reduces more significantly as an exponential function. Therefore, the vector towards the global best position $\overrightarrow{V_b}$ is always more powerful than the random vector $\overrightarrow{V_a}$ and will convergence the particles to the global best position g_{bst}.

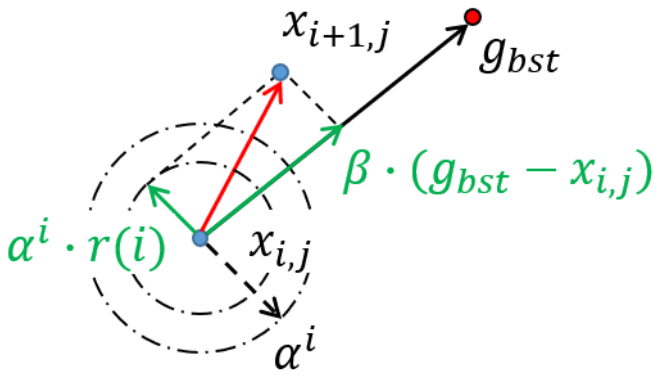


Fig. 4-10 Two vectors for particle convergence control

For example, if the particles located in $x_{i,*} = [4, 12100, 0.06]$ was regarded as the 'global' best position g_{bst}, all the other particles will move towards this particle until convergence. The particles may sometimes jump out from the local optima only if a better position is located on the route from $x_{i,j} \rightarrow x_{i,*}$, however, the probability is very low. Therefore, a more effective search method should be developed to avoid the local falling of particles so that the consistency of the optimisation results would be improved.

4.3 Optimisation with CAPSO

The basic idea of the chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm is introducing an iteration-variant attraction factor value $\beta(i)$ to reduce the power of $\overrightarrow{V_b}$ at initial iterations to allow enough local search to avoid the drop of the local optima, and further enhanced the power of $\overrightarrow{V_b}$ at later interactions to further accelerate the speed of convergence. The core algorithm for the particle position update of the CAPSO is developed as:

$$x_{i+1,j} = x_{i,j} + \beta(i) \cdot (g_{bst} - x_{i,j}) + \alpha^{i+1} \cdot r_{i,j}$$
$$i = 0,1,2,\dots,N;\ j = 1,2,..,p \tag{4-16}$$

where the most significant development rooted from Equation 4-13 is an iteration-variant attraction factor $\beta(i)$. In this study, the attraction factor at the initial iteration is 0.3. Iteration-variant functions will be investigated to generate a series of different $\{\beta(1), \beta(2), \dots, \beta(N)\}$, and converge its value to a higher value (between 0.5 and 1) as the iteration progresses.

Chaotic maps are used to build the iteration-variant attraction factor $\beta(i)$. Four typical mapping strategies are studied in this research: Gauss/mouse map, singer map, sinusoidal map and logistic map, which have been evaluated as the best four out of twelve candidates for solving the standard algorithm testing functions (i.e. Griewank function, Ackley function, Sphere function) (Gandomi *et al.*, 2013).

a) Gauss map

A Gauss/mouse map is defined using a mod function as (Di He *et al.*, 2001):

$$\beta(i+1) = \begin{cases} 1 & if\ \beta(i) = 1 \\ 1 - \frac{1}{1-\beta(i)} mod(1) & if\ \beta(i) \neq 1 \end{cases} \quad (4\text{-}17)$$

where, $\beta(i)$ is the attraction factor at i^{th} iteration; $\beta(i+1)$ is the attraction factor at $(i+1)^{th}$ iteration; $\frac{1}{1-\beta(i)} mod(1)$ is the remainder of the division of $\frac{1}{\beta(i)}$ by 1, for example $\frac{1}{0.8} mod(1) = 0.25$. The CAPSO algorithm using the Gauss map is named CAPSO-I in the rest of this thesis.

b) Singer map

A Singer map is a quartic polynomial as (Fister *et al.*, 2015a):

$$\beta(i+1) = 1 - \mu_1 \cdot \sum_{k=1}^{4} c_k (1 - \beta(i)^k) \quad k = 1,2,3,4 \quad (4\text{-}18)$$

where the value of μ is 0.95; the value of c_1 is 7.86; the value of c_2 is -23.31; the value of c_3 is 28.75; the value of c_4 is -13.30 (Fister *et al.*, 2015b). The CAPSO algorithm using the singer map is named CAPSO-II in the rest of this thesis.

c) Sinusoidal map

A sinusoidal map is defined as a sinusoidal function as (Li, Deng and Xiao, 2011):

$$\beta(i+1) = 1 - \sin(\pi \cdot (1 - \beta(i))) \quad (4\text{-}19)$$

The CAPSO algorithm using the sinusoidal map is named CAPSO-III in the rest of this thesis.

d) Logistic map

A Logistic map uses an Equation which appears as the nonlinear dynamics of biological population evidencing chaotic behaviour as (Gandomi *et al.*, 2013):

$$\beta(i+1) = \mu_2 \cdot \beta(i) \cdot (1 - \beta(i)) \tag{4-20}$$

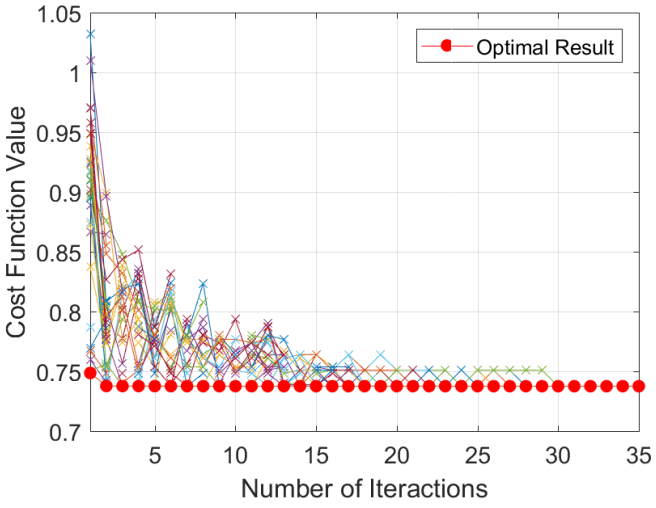
where, $\mu_2 = 4$ is chosen for this research (Gandomi *et al.*, 2013). The CAPSO algorithm using the logistic map is named CAPSO-IV in the rest of this thesis.

4.4 Results and Discussion

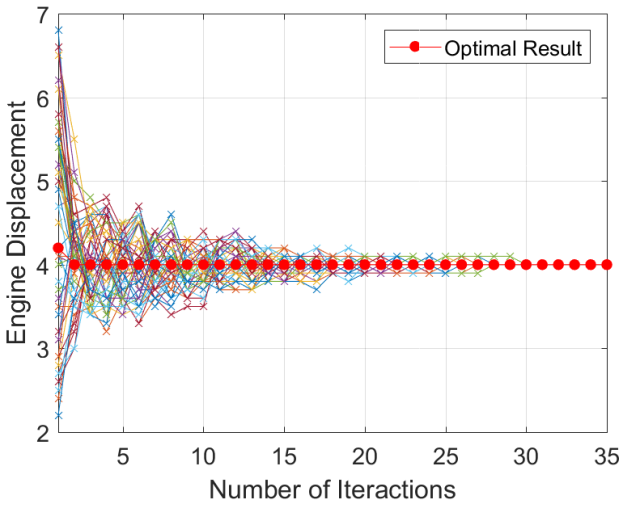
4.4.1 Optimisation results with CAPSO

The process of design optimisation in a single trial using the CAPSO algorithm is presented in Fig. 4-11. The logistic map is used for mapping the attraction factor value in this case. The red dots in Fig. 4-11 (a) tracks the evolution trajectory of the best cost-function value of the design optimisation while the red dot in Fig. 4-11 (b), (c), and (d) indicate the moving of the best particle positions. The swarm can search for the optimal position with a wide range at the initial iterations, and convergence at around 30 iterations (faster than the APSO which requires 33 iterations to converge

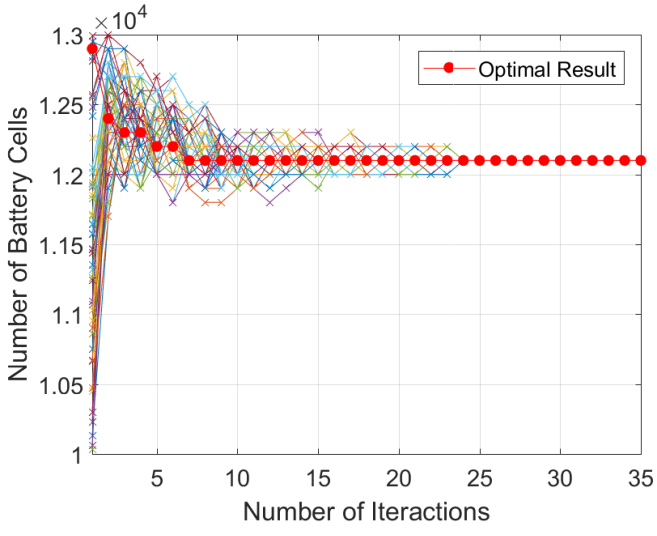
the swarm).


Optimal Result
Cost Function Value
Number of Iteractions
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
5
10
15
20
25
30
35

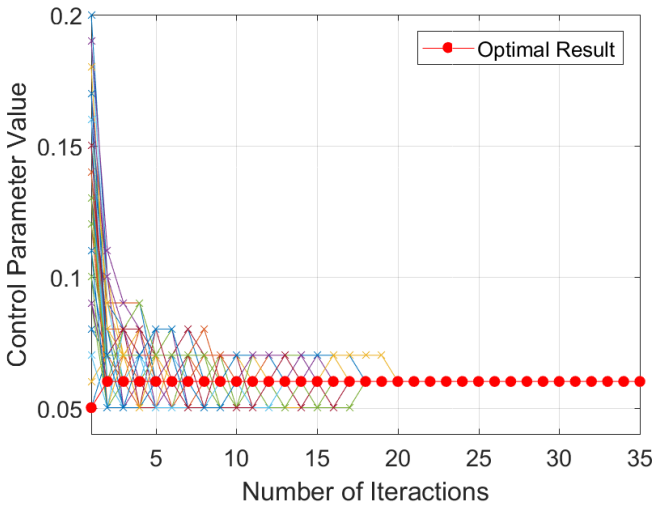
(a)


Optimal Result
Engine Displacement
Number of Iteractions
7
6
5
4
3
2
5
10
15
20
25
30
35

(b)



(c)



(d)

Fig. 4-11 Evolution of the optimisation results with CAPSO (logistic map)

The optimisation results (including the values of the design parameters and the corresponding cost function values) of 20 different trials using the CAPSO (with logistic map) are listed in Table.4-8. The CAPSO can achieve the minimum cost-function value (0.737497) for 9 times (the APSO only achieves 3 times). The CAPSO algorithm significantly reduces the chance to local optimum result (0.741411) from 11 times to 2 times.

Table.4-8 Diversity of optimisation results by CAPSO (logistic map)

No. of Test	Cost function Value	Engine disp.	No. of cells	Ctrl. parameter
1	0.739426	4	10900	0.07
2	0.739426	4	10900	0.07
3	0.739426	4	10900	0.07
4	0.741411	4	10400	0.08
5	0.737497	4	12100	0.06
6	0.737497	4	12100	0.06
7	0.737497	4	12100	0.06
8	0.737497	4	12100	0.06
9	0.741411	4	10400	0.08
10	0.739426	4	10900	0.07
11	0.737497	4	12100	0.06
12	0.737497	4	12100	0.06
13	0.737497	4	12100	0.06
14	0.737497	4	12100	0.06
15	0.739426	4	10900	0.07
16	0.737497	4	12100	0.06
17	0.739426	4	10900	0.07
18	0.739426	4	10900	0.07
19	0.739426	4	10900	0.07
20	0.739426	4	10900	0.07

4.4.2 Monte Carlo analysis

A Monte Carlo analysis is carried out to evaluate the performance of the algorithms,

including the standard APSO and CAPSOs with the proposed chaotic maps from the perspective of statistics. With running each set-off 20 times with uniformly distributed random initial values, the statistic results of the cost-function values obtained by the five optimisation algorithms in the 20 trials are listed in Table.4-9.

Table.4-9 Statistic property of the 20 individual optimisation tests

Algorithm	Best value	Mean value	σ
APSO	0.7375	0.7405	2.18e-6
CAPSO-I	0.7375	0.7393	2.62e-6
CAPSO-II	0.7375	0.7394	2.01e-6
CAPSO-III	0.7375	0.7394	3.73e-6
CAPSO-IV	0.7375	0.7338	2.04e-6

All of the APSO and CAPSO algorithms have the capability to converge the swarm to the same best value of 0.7375, which can be regarded as the global optimisation result in this study. The mean values of the cost-function values obtained by the CAPSO algorithms are smaller than those obtained by the APSO, which shows the advancement of the CAPSO algorithms. The CAPSO-IV algorithm with a logistic map achieves the minimum average cost function value of 0.7338, resulting in a 0.91% improvement compared with conventional APSO. In terms of the standard derivation of the 20 trials which indicates the diversity of the optimisation results, the CAPSO-II with a singer map enjoys the lowest standard derivation value; while the CAPSO-IV with a logistic map has the second-lowest standard derivation value.

4.4.3 Reputation evaluation

In real practice, engineers always concern more about the probability of how an optimisation algorithm achieves the real global best results, which is a more strict and observable evaluation compared with the Monte Carlo analysis. The reputation evaluation of the optimisation algorithm is performed in this research, and the reputation index is defined as:

$$R_i = \frac{Num_{opt.}}{Num_{trial}} \tag{4-21}$$

where, R_i is the reputation index; $Num_{opt.}$ is the number of achieving the global best result; Num_{trial} is the total number of the trails in the random repeat test. The reputation index values of all the APSO and CAPSOs are listed in Table.4-10.

Table.4-10 The reputation index value for different algorithm

Algorithm	Total Trials	No. of Best	R_i
APSO	20	3	0.15
CAPSO-I	20	7	0.35
CAPSO-II	20	5	0.25
CAPSO-III	20	4	0.20
CAPSO-IV	20	9	0.45

The reputation index value of the conventional APSO is 0.15, which indicates the APSO algorithm can only access to the global best from 3 times out of 20 trials. All CAPSOs have a higher reputation index value than the conventional APSO. The

CAPSO-IV algorithm with a logistic map has the highest reputation value of 0.45 which is twice as higher as that of APSO.

A comprehensive reputation evaluation is performed to choose the most suitable swarm intelligent algorithm for design optimisation considering the average cost-function value, the standard derivation of the cost-function value, and the reputation index value. A scoring system is established based on the ranking of each algorithm in terms of its reputation index value as well as the mean value and standard derivation of the cost-function values. The higher the ranking of the algorithm is the higher score it will win. For example, a 5 score is given to the algorithm with the 1st ranking, and 1 score is given to the algorithm with the 5th ranking. The scoring of the conventional APSO and CAPSOs are shown in a spider chart in Fig. 4-12. According to the comprehensive evaluation, the CAPSO-IV with a logistic map is considered as the most effective optimisation algorithm for design optimisation.

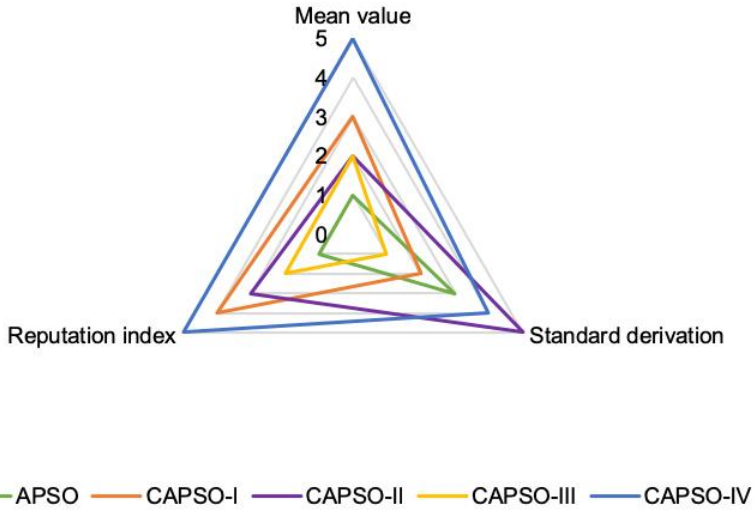


Fig. 4-12 Spider-chart of scoring for APSO and CAPSOs

4.4.4 Pareto optimal frontier

The Pareto frontier of the design optimisation is obtained using CAPSO-IV with the weight factor changing from 0 to 1, as shown in Fig. 4-13. The weight factor of 1 means the optimisation only consider the minimisation of energy loss whereas the weight factor of 0 means only the minimisation of the component size is considered. The Pareto optimal frontier is a set of non-dominated results considering both optimisation objectives with different weight values and shows the trade-off between the two objectives.

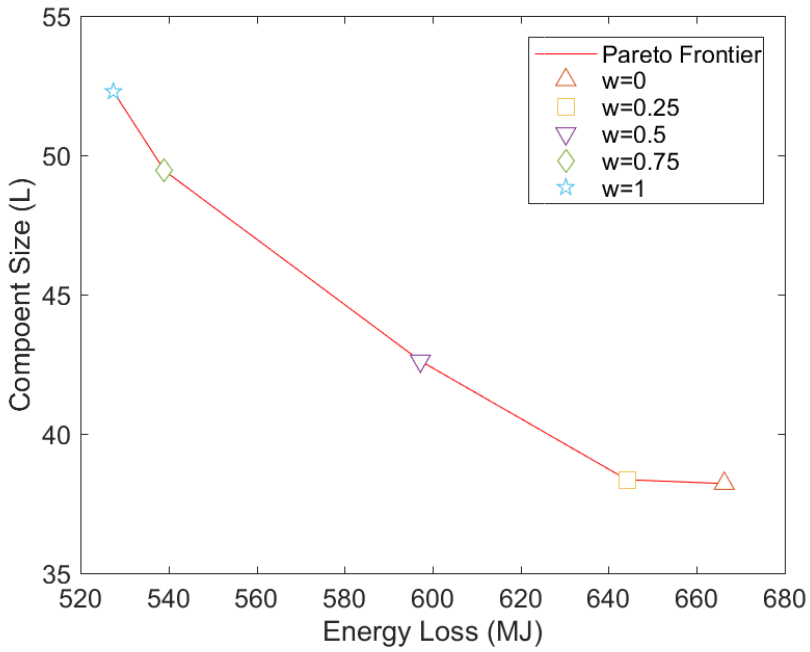


Fig. 4-13 The Pareto frontier of the multi-objective design optimisation

The analytical result showing how power-loss can be reduced by the increase of the component size is presented in Fig. 4-14. The results are calculated with the Pareto

frontier set, which indicates a 3.3% energy loss can be reduced by an increase of 0.36% in components size when the weight value changes from 0 to 0.25; an 11.52% component size increase can result in a 10.36% energy loss reduction when the weight value changes from 0 to 0.5; a 19% energy loss reductions requires at least a 29% component size increase when weight value changes from 0 to 0.75; a 37% increase of the component sizing can only achieve a 20% energy loss reduction when the weight value changes from 0 to 1.

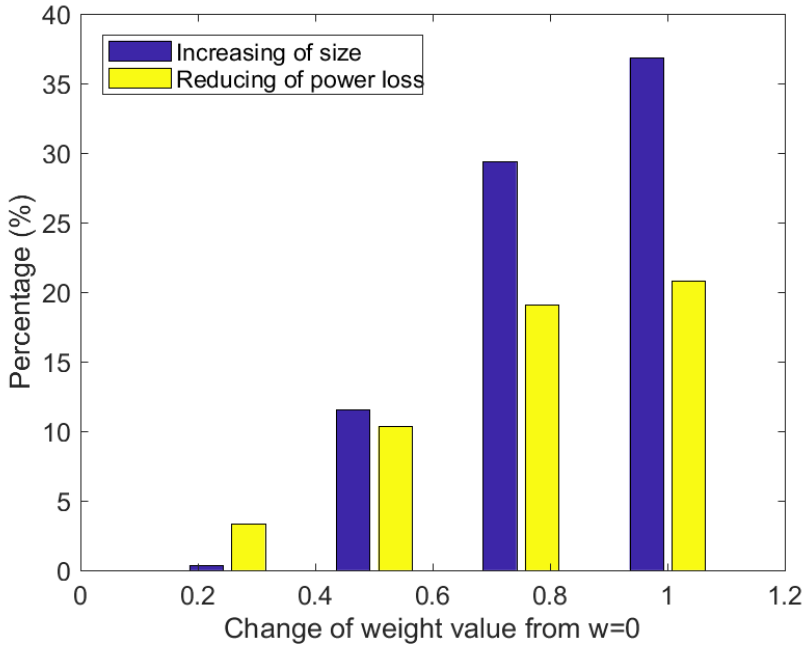


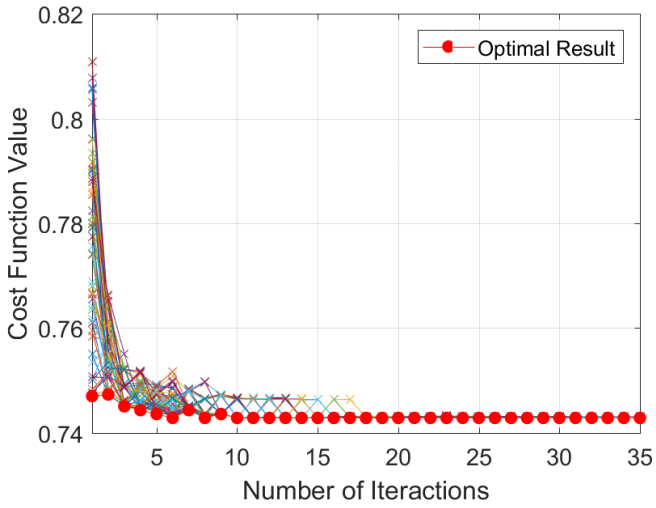
Fig. 4-14 The trade-off between two optimisation objectives

4.4.5 Final optimisation results confirmation

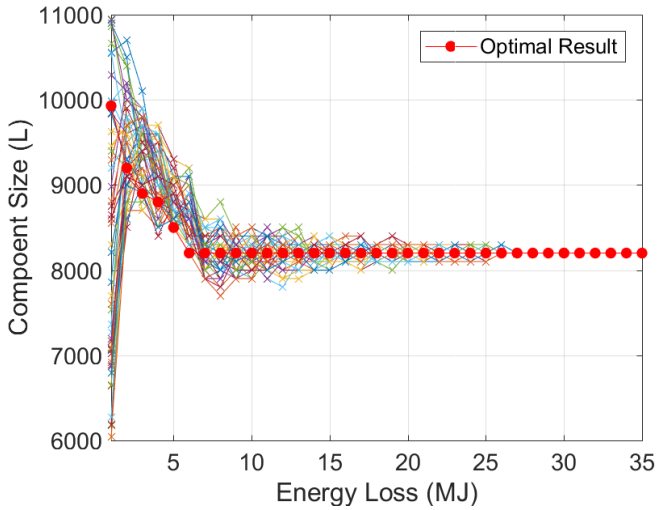
According to the requirement of the OEM, the global optimum result of the design optimisation using weight factor value of 0.5 is chosen which determines the optimal

engine size is 4.0L, the optimum number of battery cells is 12100, and the optimum control parameter value is 0.06. The engine-generator set is the most expensive subsystem for the hybrid powertrain. It is more cost-efficient to choose an existing product from the product library of JCB. Therefore, a JCB engine-generator unit with a 4.4L diesel engine is chosen for this study and second-round design optimisation is operated to determine the optimal battery cell number and the control parameter.

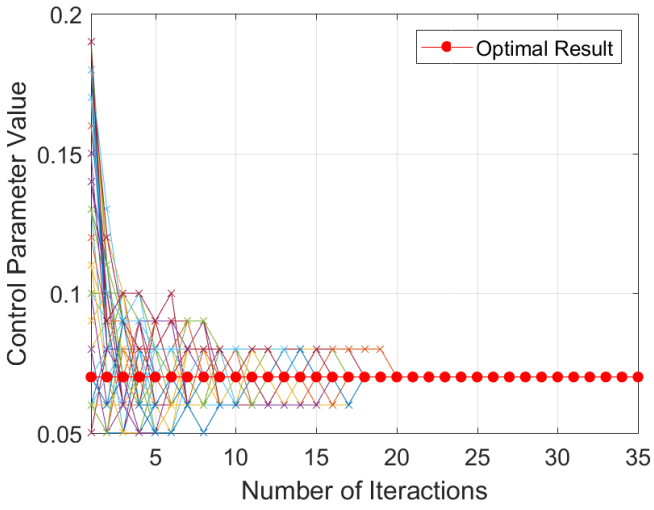
The process of the second-round optimisation is shown in Fig. 4-15.



(a)



(b)



(c)

Fig. 4-15 Process of second-round optimisation

According to another 20 trials for the second round design optimisation, the optimum battery cell number and control parameter are 8200 and 0.07 respectively. The second-round optimisation result is adopted by the OEM, and it helped them to confirm the size and control parameter for their prototype finally. The optimisation result will also be used for the research of new energy management methodologies.

4.5 Summary

The design optimisation for component sizing and control parameters calibration of the electrified vehicle is performed in this chapter. Well-adapted accelerated particle swarm optimisation (APSO) is used as the base-line technology for this research. A new design optimisation methodology is developed based on a chaos-enhanced accelerated particle swarm optimisation (CAPSO) algorithm. Four different chaotic-

mapping strategies are investigated and evaluated using Monte-Carol analysis and reputation evaluation. The Pareto frontier of the design optimisation has been obtained, and the principle of the weight-sum function design has also been discussed. The conclusions drawn from the investigation are as follows:

- Both APSO and CAPSO have the capability of global optimum search and the electrified vehicle in this research is expected to achieve the global optimal design target with an engine size of 4.0L, a battery package of 12100, and power distribution function parameter value of 0.06.

- CAPSOs algorithms with all four proposed chaotic mapping strategies have the capability of optimisation result improvement over the conventional APSO according to the Monte-Carlo analysis. A significant improvement of 200% has been achieved by a CAPSO with a logistic mapping strategy in terms of reputation index. This contribution is extremely important, which can improve the reliability of global search with the proposed algorithm in real engineering practice.

- According to a comprehensive evolution in terms of the Monte-Carlo analysis and reputation evolution, the CAPSO with a logistic map is selected as the most effective algorithm for design optimisation of electrified vehicles.

- The Pareto optimal frontier is obtained to illustrate the trade-off of the design

objectives by different chosen weigh values. When the weight value changes from 0 to 0.5, a higher energy loss reduction rate can be achieved with less increase in the size of the components.

Further research and investigation of online optimal energy management control will be progressed with the results from the second-round optimisation (i.e. engine size of 4.4L, the battery cell number of 8200, and control parameter of 0.07); which have been confirmed with an industrial partner. In addition, the proposed method has the capability of exploring the optimum combination in a ‘free-selection’ world when using different settings for the optimisation constraints. Therefore, it can break the restrictions from the available on-the-shelf components and provide recommendations to the component suppliers.

Chapter Five

ONLINE PARTICLE SWARM INTELLIGENCE OPTIMISATION FOR MODEL-BASED PREDICTIVE ENERGY MANAGEMENT CONTROL

This chapter develops a new model-based predictive energy management control strategy, which enables the capability of online control optimisation of the electrified aircraft-towing tractor. The proposed energy management strategy is centred on the Online Swarm Intelligent Programming (OSIP), which implements a new particle swarm optimisation scheme for online nonlinear optimisation (Zhou *et al.*, 2018). The online optimisation is firstly analysed via control-oriented modelling followed by the development of the OSIP. After the implementation of the proposed control strategy with a distributed control system, the advantages of the proposed method are demonstrated and evaluated.

5.1 Optimisation Problem associated with the Energy Management

The optimisation problem associated with the energy management of the electrified vehicle is to retrieve the optimal control signals of the battery package u_{batt} and the

engine generator u_{egu} in real-time to minimise energy loss (namely the useless energy dissipated by heat) over any given prediction horizons so that the vehicle efficiency can be improved in real-world operations. If the control signal of the engine generator is determined, the control signal of the battery pack can be calculated with:

$$u_{batt}(t) = \frac{P_r(t) - u_{egu}(t) \cdot P_{egu\_max}}{P_{batt\_max}} \tag{5-1}$$

where, P_r is the power demand for driving the vehicle which can be calculated using the predictive model (will be discussed in section 5.2.1); P_{egu_max} and P_{batt_max} are the maximum power of the engine generator and the battery package, respectively.

Therefore, the optimisation problem in each time interval is to obtain the optimal control signal of the engine generator:

$$s.t. \begin{cases} u^*_{egu}(t) = \arg\min(\sum_{i=t}^{t+p} x_{egu}(i) \cdot \Delta t + \sum_{i=t}^{t+p} x_{bp}(i) \cdot \Delta t) \\ u^-_{egu} \le u_{egu} \le u^+_{egu} \\ s_{egu}(t) = u_{apu}(t) \cdot P_{apu\_max}(1 - \eta_{f2e}(u_{apu}(t) \cdot P_{apu\_max})) \\ s_{batt}(t) = (\frac{1}{2}(V_{oc}(SoC) - \sqrt{V_{oc}(SoC)^2 - \frac{4R_{loss}(\cdot) \cdot u_{batt}(t) \cdot P_{bp\_max}}{n_{bc}}}))^2 \cdot \frac{n_{bc}}{R_{loss}(\cdot)} \\ u_{batt}(t) = \frac{P_r(t) - u_{apu}(t) \cdot P_{apu\_max}}{P_{apu\_max}} \\ \mathrm{SoC}^- \le \mathrm{SoC} \le \mathrm{SoC}^+ \end{cases} \tag{5-2}$$

where, $u^*_{egu}(t)$ is the optimal control command for the engine generator at t^{th} time interval; s_{egu} and s_{bp} are the transient states of the engine generator and the battery pack respectively which will be predicted in model-based predictive control; $u^-_{egu} = 0$ and $u^+_{egu} = 1$ are the lower boundary and higher boundary for the command signal; $\mathrm{SoC}^- = 0.2$ and $\mathrm{SoC}^+ = 0.8$ are the lower and higher boundaries

of the battery's SoC; η_{f2e} is the power conversion efficiency of 'fuel to electricity'; V_{oc} is the average open-circuit voltage of the battery cell; SoC is the State of Charge of the battery; R_{loss} is the internal resistance of the battery; n_{bc} is the number of battery cells within the battery pack.

According to Equation 5-2, the optimisation is an online optimisation with nonlinear constraints:

$$\begin{cases} s_{apu}(t) = u_{apu}(t) \cdot P_{apu\_max} \cdot (1 - \eta_{f2e}(\cdot)) \\ s_{batt}(t) = (\frac{1}{2}\left(V_{oc}(SoC) - \sqrt{V_{oc}(SoC)^2 - \frac{4R_{loss}(\cdot)u_{batt}(t) \cdot P_{bp\_max}}{Num_{bc}}}\right))^2 \cdot \frac{Num_{bc}}{R_{loss}(\cdot)} \end{cases} \quad (5\text{-}3)$$

In this case, the control signal of the engine generator has a resolution of 0.05. Therefore, the optimisation problem associated with energy management control is an integer nonlinear optimisation problem.

5.2 Model-based Predictive Energy Management Control

To solve the integer nonlinear optimisation problem in real-time, this Chapter proposed a new model-based predictive control methodology. Predictive models are firstly developed for control purpose based on a necessary simplification of the real-time models in Chapter 3. The online swarm intelligent programming is then proposed as a new method for online nonlinear optimisation.

5.2.1 Control-oriented predictive modelling

The control-oriented models are necessary for prediction of the system performance feedback with different trial control signal inputs so that the optimal control signals can be obtained by online optimisation. The control-oriented models should have the capability of computing with ultra-high-speed comparing with the real-time system. This section carries out control-oriented modelling based on the reasonable simplification of the real-time models developed in Chapter 3.

a) Power-flow of hybrid electric vehicle

The power-flow among the traction motor, the engine-generator unit, and the battery package are studied considering energy conversion and transfer. The battery pack (BP) and the engine generator unit (EGU) are connected within the DC-link. The traction motor could be powered by either the BP only or BP and EGU together. The EGU can also be used to charge the battery pack to maintain its SoC within the proper level. The power flow of the vehicle system obeys:

$$P_{link}(t) = P_{apu}(t) + P_{bp\_d}(t) - P_c(t) \tag{5-4}$$

where, $P_{egu}(t)$ is the output power provided by the EGU, $P_{bp\_d}(t)$ is the discharge power of the BP, $P_c(t)$ is the charge power for the battery pack, and $P_{link}(t)$ is the power of the DC-link, and it obeys:

$$P_{tm}(t) = P_{link}(t) - P_{loss,tm}(t) \tag{5-5}$$

where, $P_{loss,tm}(t)$ is the power loss in the traction motor, and $P_{tm}(t)$ is the power for driving the vehicle, and it obeys:

$$\frac{P_{tm}(t)\cdot\eta_t}{V_{dem}(t)} = \left(m_{veh} + m_{tow}(t)\right)\cdot\frac{dv_{dem}(t)}{dt} + \left(m_{veh} + m_{tow}(t)\right)\cdot g\cdot f_{veh\_r} + \frac{C_d\cdot A\cdot v_{dem}(t)^2}{21.15} \quad (5\text{-}6)$$

where, η_t is the transmission efficiency (in this work a fixed transmission efficiency is chosen as 80% as an approximation); $v_{dem}(t)$ is the vehicle driving speed; m_{veh} is the vehicle mass; $m_{tow}(t)$ is the towing mass and it varies with time while towing different aircraft; g is the gravity acceleration which equals to 9.81m/s^2; f_{veh_r} is the rolling resistance coefficient; C_d is the drag coefficient, and A is the vehicle front area.

As the push-back tractor is mainly working in the plain ground, only the acceleration resistance, rolling resistance and drag resistance need to be considered when modelling the vehicle dynamics. The major vehicle information for vehicle dynamics calculation are listed in Table.3-3.

b) Prediction of future power demand

Future power demands of the hybrid off-highway vehicle are obtained by the exponential varying expression method, in which torque demands of the traction motor $\boldsymbol{T}_{em} \in R^{1\times h}$ over the prediction horizon are predicted to be exponentially decreasing as(Li *et al.*, 2011; Borhan *et al.*, 2012):

$$T_{tm}(k+i-1|k) = T_{tm}(k-1|k) \cdot e^{\left(-\frac{i\tau_s}{\tau_d}\right)} \quad (i = 1, 2 \dots h) \tag{5-7}$$

where, $T_{tm}(k-1|k)$ is the known motor torque value measured at the end of the last time interval; $T_{tm}(k|k), T_{tm}(k+1|k), T_{tm}(k+2|k) \dots T_{tm}(k+h-1|k)$ are the predicted motor torque over the prediction horizon h; $\tau_s = 1s$ is the sampling time and τ_d is the decay rate, which was a constant for passenger cars.

In this work, the plane mass is much larger than the vehicle mass, and the torque demand varies significantly when pushing back different aeroplanes, therefore, a mass-varying parameter $\tau_d(mass_{plane})$ is proposed as a function of plane mass in kilogram in this research as:

$$\tau_d(mass_{plane}) = \tau_0 \cdot mass_{plane} \tag{5-8}$$

where, $\tau_0 = 2.36 \times 10^{-3}$ is the unit decay rate which is tuned when assuming the pushing back mass is 1 tonne. The vehicle speed could be predicted by solving the vehicle dynamics Equation as:

$$v_{veh}(k+i-1|k) = v_{veh}(k+i-2|k) + \frac{(T_{em}(k+i-1|k) \cdot i_{fr} \cdot r_{tire} - F_r - F_d)}{m_{veh} + m_{plane}} \cdot \tau_s \quad (i = 1, 2 \dots p) \tag{5-9}$$

where, $v_{veh}(k-1|k)$ is the known vehicle speed at the end of the last time interval, $v_{veh}(k|k), v_{veh}(k+1|k), v_{veh}(k+2|k) \dots v_{veh}(k+h-1|k)$ are the predicted vehicle speed over the prediction horizon p, $T_{tm}(k+i-1|k)$ is the predicted motor torque calculated by Equation (24); i_{fr} is the final drive gear ratio, r_{tire} is the radius of the

wheel; F_r and F_d are the vehicle's rolling resistance and drag resistance that could be calculated by Equation (3-2).

Power demands for the electric motor is predicted by:

$$P_r(k+i-1|k) = \frac{T_{tm}(k+i-1|k)\cdot v_{veh}(k+i-1|k)\cdot i_{fr}\cdot r_{tire}}{\eta_{tm}(T_{tm}(k+i-1|k),n_{tm}(k+i-1|k))} \quad (i = 1,2 \dots h) \quad (5\text{-}10)$$

where, $P_r(k|k), P_r(k+1|k), P_r(k+2|k) \dots P_r(k+h-1|k)$ are the predicted power demand over the prediction horizon h; n_{em} is the motor speed; η_{tm} is the motor efficiency.

c) Traction motor

The selected traction motor in this study is a heavy-duty electric motor (type: LSM280A HV-2700) provided by TM4 electrodynamic system Ltd. The motor specification is listed in Table.3-6. As for a heavy-duty application under investigation in this work, the vehicle usually operates in low speed and high load duty cycle, so it is not cost-effective to apply a regenerative braking system. Therefore, the traction motor is only working in traction mode.

In the controller, the mechanical power generated by the motor is modelled as:

$$P_{tm}(t) = \frac{T_{tm}(t)\cdot n_{tm}(t)}{9550} \quad (5\text{-}11)$$

And the motor loss is characterized by a quadratic function of the motor torque(Hu

et al., 2013):

$$P_{loss,tm} = a_2(n_{tm}) \cdot {T_{tm}}^2 + a_1(n_{tm}) \cdot T_{tm} + a_0(n_{tm}) \qquad (5\text{-}12)$$

where, a_i (i=0,1,2) are motor-speed-dependent coefficients, which are modelled as cubic functions of the motor speed:

$$a_i(n_{tm}) = a_{i,3} \cdot {n_{tm}}^3 + a_{i,2} \cdot {n_{tm}}^2 + a_{i,1} \cdot n_{tm} + a_{i,0} \quad (i = 0,1,2) \qquad (5\text{-}13)$$

In Equations (5-11) and (5-12), the coefficient values are determined by curve fitting the testing data, and the motor efficiency properties calculated by:

$$\eta_{tm} = \frac{P_{tm}}{P_{tm}+P_{loss,tm}} \qquad (5\text{-}14)$$

Performance of the traction motor model is validated with the original data provided by the motor supplier in Fig. 5-1.

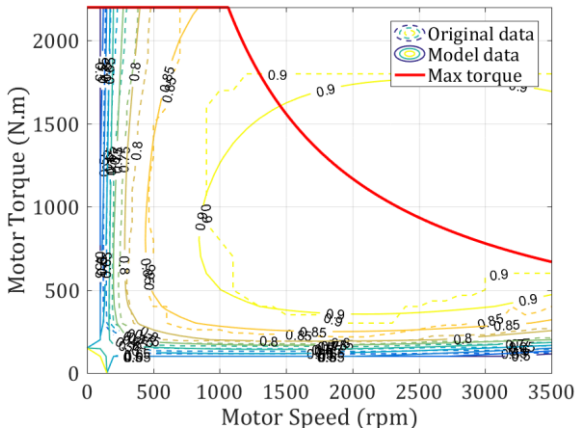


Fig. 5-1 Validated control-oriented model of traction motor efficiency

d) Engine generator unit

In this powertrain system, an 86.2 kW engine-generator unit (EGU) produced by JCB

(*JCB Generator Technical Specifications*, 2015) is selected. The EGU is powered by a 4.4L diesel engine and generates electric power with a 3-phase AC generator, the EGU itself has a 120Ah battery, and therefore, this work assumes no energy required from the vehicle battery pack when the engine is starting. Main technical parameters of the selected EGU are listed in Table.5-1

Table.5-1 Specification of the engine generator

EGU specification	Value
Max. primer power	86.20 kW
Frequency	50 Hz
Voltage	230V~
Phases	3-AC
Fuel type	diesel
50% load fuel rate	13.00 L/h
75% load fuel rate	18.60 L/h
100% load fuel rate	24.10 L/h
Fuel tank capacity	285 L

The on-board model for EGU fuel-electricity conversion uses a quadratic function of the EGU's power output P_{apu}, thereby yielding diesel fuel power (Hu *et al.*, 2013):

$$P_{fuel} = b_2 \cdot {P_{apu}}^2 + b_1 \cdot P_{apu} + b_0 \tag{5-15}$$

where, P_{fuel} is the equivalent power of fuel consumed, which could be calculated as:

$$P_{fuel} = \frac{\dot{v}_f \cdot \rho_f \cdot H_f}{3600} \tag{5-16}$$

where, $\dot{v}_f$ is fuel consumption rate in L/h; ρ_f is the density of the fuel, for the diesel fuel, $\rho_f = 0.87$ kg/L; H_f is the heat value of the fuel, for the diesel fuel, $H_f = 44 \times 10^6$ J/kg(Isermann, 2014).

Then, the coefficient b_1 and b_2 are determined using curve fitting of the fuel rate data in Table.5-1. The fuel-electricity conversion efficiency η_{f2e} is obtained by:

$$\eta_{f2e}(P_{apu}) = \frac{P_{apu}}{P_{fuel}} = \frac{P_{apu}}{b_2 \cdot {P_{apu}}^2 + b_1 \cdot P_{apu} + b_0} \tag{5-17}$$

Performance of the traction motor model is validated with the original data provided by the motor supplier in Fig. 5-2.

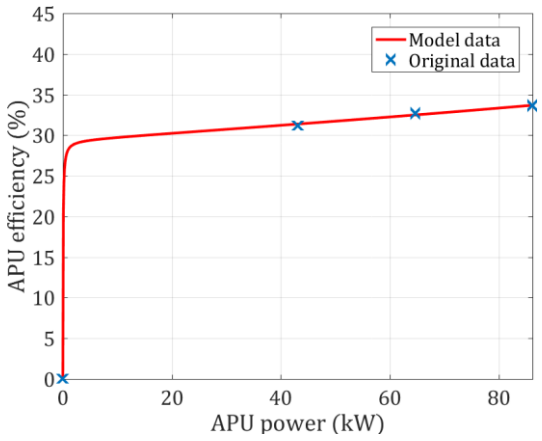


Fig. 5-2 Validated control-oriented model of engine-generator efficiency

e) Battery pack

The battery pack (BP) is made up of the battery cell type NCR-18650 series provided by Panasonic Automotive & Industrial System Ltd. The basic parameters of each individual cell could be found at (*Lithium Ion Battery-Cylindrical, Type: UR-18650*,

2017). The voltage of battery cells ranges from 2.5 V to 4.2 V, and the nominal battery voltage is 3.7V. The battery cell's rated capacity is 2450mAh. The battery pack parameters are obtained by arranging the battery cells in parallel and series. For this study, the battery pack is made up of 8200 battery cells. For the control-oriented battery model, a simple resistive circuit is chosen (Murgovski *et al.*, 2012b).

For the BP, the output power $P_{bp}(t)$ is simplified as the difference of discharge power $P_{bp\_d}(t)$ and charge power $P_c(t)$:

$$P_{bp}(t)=P_{bp\_d}(t) - P_c(t) \tag{5-18}$$

where the power output of the battery package is:

$$P_{bp}(t) = Num_{bc} \cdot V_{oc}(SoC) \cdot I_{bc}(t) - Num_{bc} \cdot R_{loss}(SoC) \cdot I_{bc}(t)^2 \tag{5-19}$$

where, Num_{bc} is the total number of battery cells in the BP, $V_{oc}(SoC)$ is the open-circuit voltage of a single battery cell, and $R_{loss}(SoC)$ is the internal resistance in the equivalent battery circuit. Both V_{oc} and R_{loss} are SoC dependent functions. In Equation (10) $I_{bc}(t)$ is the battery cell current, and can be expressed as:

$$I_{bc}(t) = \frac{1}{2R_{loss}}\left(V_{oc}(SoC) - \sqrt{V_{oc}(SoC)^2 - \frac{4R_{loss}P_{bp}(t)}{Num_{bc}}}\right) \in [I_{bc\_min}, I_{bc\_max}] \tag{5-20}$$

where $[I_{bp\_min}, I_{bp\_max}]$ is the battery cell current limits. The open-circuit voltage V_{oc} and the battery's internal resistance R_{loss} are modelled as SoC dependent exponential functions using the original data from the battery manufacturer. As discussed in (Chen and Rincón-Mora, 2006), the model for the open-circuit voltage

V_{oc} and the battery internal resistance R_{loss} are:

$$\begin{cases} V_{oc}(SoC) = c_4 \cdot e^{c_5 \cdot SoC} + c_3 \cdot SoC^3 + c_2 \cdot SoC^2 + c_1 \cdot SoC + c_0 \\ R_{loss}\left(SoC,\ I_{bc}\right) = \frac{SoC}{c_5(I_{bc}) + c_6(I_{bc}) \cdot SoC} \end{cases} \tag{5-21}$$

where, c_i $(i = 0,2,3 \ldots 7)$ are the model parameters, in which, c_1 to c_4 are constant and c_5 and c_6 are I_{bp} dependent polynomial functions, all of which are determined by curve fitting using the test data. And the battery cell's SoC is calculated by:

$$SoC = SoC_0 - \int \frac{I_{bc}}{Q_{bc}} dt \tag{5-22}$$

where, SoC_0 is the battery's initial SoC value, Q_{bc} is the battery cell's rated capacity. The battery V-I dynamics is validated with the original data provided by the battery supplier in Fig. 5-3.

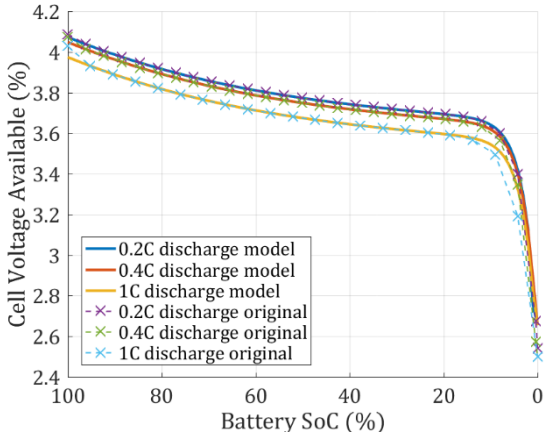


Fig. 5-3 Validated control-oriented model of battery cell V-I dynamics

5.2.2 Online swarm intelligent programming with CAPSO

Fig. 5-4 shows the core algorithm developed for OSIP in a single time instant. The algorithm is developed based on CAPSO, which has three main procedures, namely, initialization, main iteration, and optimal position retrieving. The details and principle of the CAPSO algorithm working procedure are discussed in Chapter 4.3. To solve the optimisation problem in Equation (5-2) online, the algorithm is customised and modified in the following aspects:

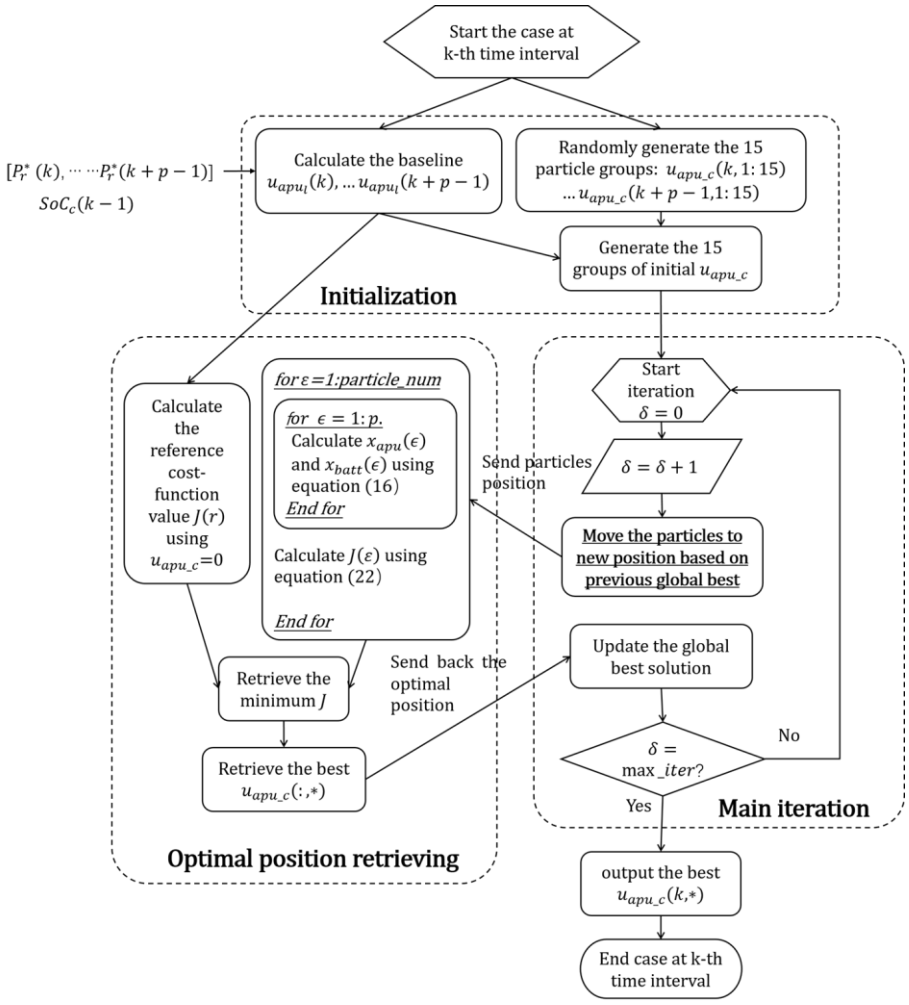


Fig. 5-4 Flow-chat of CAPSO algorithm for online optimisation

a) The definition of particle position

In the initialization procedure, the position of particles is defined as:

$$x_{\delta,\varepsilon} = \boldsymbol{u}_{apu_t} \quad \delta \in [1, \max\ \_iter], \varepsilon \in [1, particle\_num] \tag{5-23}$$

where, $\boldsymbol{u}_{apu\_l} = [u_{apu\_l}(k|k), u_{apu\_l}(k+1|k), \ \ldots, u_{apu\_l}(k+h-1|k)]$ is the EGU command modification vector over the predictive horizon h; δ is the index for the number of iterations; ε is the index for each particle. To obtain sufficient adequate accuracy with the least computing effort, the value of maximum iteration $max\ \_iter$ is 30 and the number of particles $particle\_num$ is 15 as (Xu *et al.*, 2016).

b) Random number generation with a specific resolution

In the main interaction procedure, the key step is moving the particles to the new position, and the random number with the resolution is required for moving particles. The random number generation with a resolution of 0.05 is modified from the standard Linear Congruential Generator (LCG):

$$\begin{cases} R_\epsilon = (a \cdot R_{\epsilon-1} + c)\ mod\ M \\ rnd(k+\epsilon-1) = round(20 \cdot \frac{R_\epsilon}{M})/20 \end{cases} \tag{5-24}$$

where, multiplier a , additive constant c, and modulus M are integers; Equation (5-24) defines a series of random number with the initial seed R_0. The vector $\{rnd_{i,\epsilon}\ (\epsilon = 1,2, \ldots h)\}$ is a random number sequence from 0 to 1, with a resolution of 0.05. To maximise the pseudo-random number performance, the parameters of the LCG are (Xu *et al.*, 2016): R_0 =9, $a = 27$, $c = 0$, and $M = 2^{20}$.

c) Updating of particles' position

In the main interaction procedure, the position of particles is updated as:

$$x_{\delta+1,\varepsilon} = res \cdot round\{(1-\beta)\cdot\frac{x_{\delta,\varepsilon}}{res} + \beta\cdot\frac{g_{\delta,*}}{res} + \gamma^{\delta}\cdot\zeta\cdot[\mathrm{rand}(0,1)-0.5]\} \quad (5\text{-}25)$$

where, $p_{\delta+1,\varepsilon}$ is the updated position; $p_{\delta,\varepsilon}$ is the particle's position at the current iteration; $g_{\delta,*}$ is the best position of the present iteration; δ is the iteration generation; ε is the particle's individual index; res = 0.05 is the variable resolution; $\gamma = 0.85$ is the convergence parameters of CAPSO; $\zeta = 80$ is the search area factor, and β is the attraction parameters of CAPSO.

The study in Chapter 4 suggested that the CAPSO with the logistic chaotic-map is the best for integer optimisation (Zhou *et al.*, 2017). The attraction parameters β is mapped in logistic map as:

$$\beta_{\delta+1} = \alpha\cdot\beta_{\delta}\cdot(1-\beta_{\delta}) \quad (5\text{-}26)$$

where, $\beta_1 = 0.7$ and $\alpha = 4$ are used for the logistic chaotic-map (Zhou *et al.*, 2017).

d) Final outputs

When convergence has been achieved, the algorithm ends the main iteration and outputs the best position at the end iteration $g_{\max\_iter,*}$ as the global optimal solution. Then the first element of the control sequence $u(k) = (u_{apu\_c}(k|k))$ is the final output of the OSIP controller.

5.3 Implementation with a distributed control system

With the proposed cyber-physical system, tractors will be connected to the remote server in the airport control via the roadside units (RSUs) near the tractors' working area. The remote server will enable the real-time optimisation based on cloud computing via advanced online programming algorithms, Through the V2I communication, the online swarm-intelligent programming will be available working on the remote server located in the airport control and will send the optimal control command to the local vehicle controller. Subsequently, the V2V communication between the tractor and aircraft will enable the basic control function of the tractor controller.

The real-time optimal control system includes the local energy-flow control and the Cloud-based Online Swam Intelligent Programming (OSIP) in Fig. 5-5. The local control performs on the on-boarded vehicle controller, and the OSIP operates on the connected server. As the energy management system mainly considers the energy split and management, one second is chosen according to (Zhang, Xiong and Sun, 2015) as the sampling time, which is approved to be able to track the system dynamics while reserving enough time slot for algorithm computing. The mechanism of local energy-flow control and OSIP is as follows:

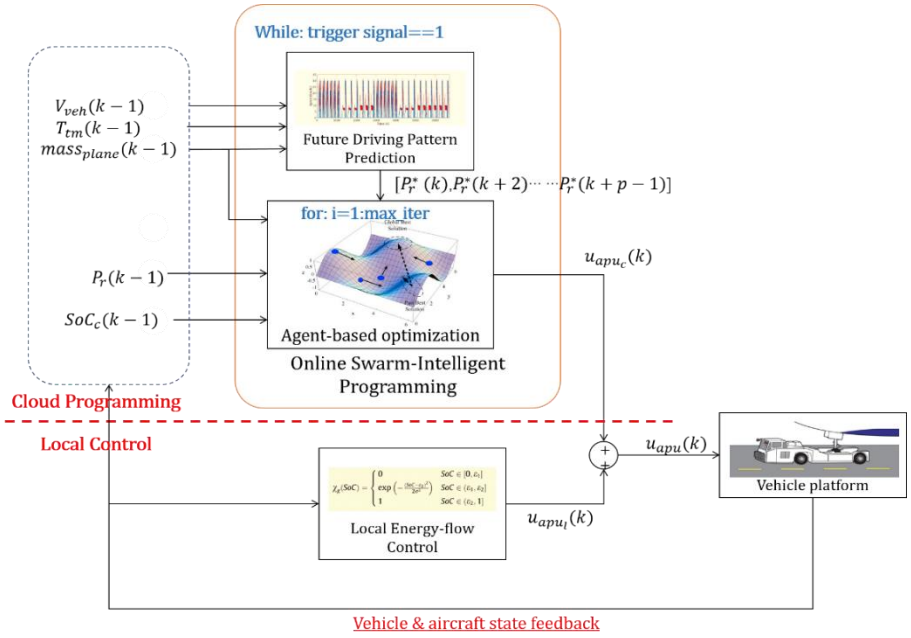


Fig. 5-5 Real-time optimal energy flow control based on OSIP

5.4 Performance and Discussions

5.4.1 Optimisation performance

The performance of the proposed optimisation method is researched via a comparison study with a Genetic Algorithm (GA). Both algorithms are written in MATLAB 2017b. The GA uses the default settings in MATLAB's GA toolbox (a population of 30 individuals with maximum interaction of 50). The OSIP uses CAPSO algorithm with 30 particles for 15 iterations. Table.5-2 shows that the OSIP is able to find the global optimal solution identical to that obtained by the GA but in a much shorter time. The results were obtained using a desktop computer configured with an i5 processor and 8GB RAM. The optimisation process in four random selected

time instants (500s, 1800s, 5050s and 6800s) is repeated using the two methods respectively for 20 times, and the optimal cost in the 20 trials is considered as the 'global optima'. Although the average cost function value is slightly lower with GA (<4%), the average computing time using GA for each time instant is more than 20s, whereas it is only less than 1s using CAPSO. Therefore, the advantage of CAPSO is outstanding because this fast response is extremely important for real-time control.

Table.5-2 Optimisation performance in single time instant

Time instant	Algorithm	Optimal Cost	Average Cost	Ave. time
500s	GA	180678.53	194348.41	23.36s
	CAPSO	180678.53	200038.25	0.72s
1800s	GA	4787.43	5245.43	25.10s
	CAPSO	4787.43	5377.13	0.81s
5050s	GA	236331.20	244481.97	24.58s
	CAPSO	236331.20	252185.55	0.76s
6800s	GA	1653523.00	1662715.87	26.20s
	CAPSO	1653523.00	1687830.51	0.83s

5.4.2 Computational effort

The computational cost is a natural concern for real-time implementation, and the prediction horizon size is the most sensitive factor which affects the computational cost (Hu, Wang and Tang, 2017). The computational cost of the proposed method with the respective size of the predictive horizon p is hereby investigated. The

optimisation problem is solved by the ETAS ES910 real-time controller. The average computational cost per time step including the data communication, is shown in Fig. 5-6. It indicates that while the augmented prediction horizon size p leads to increased computational load, prediction horizon size p being less than 36s can make the controller implementable in real-time, as the computing time is less than the sampling time of 1 second.

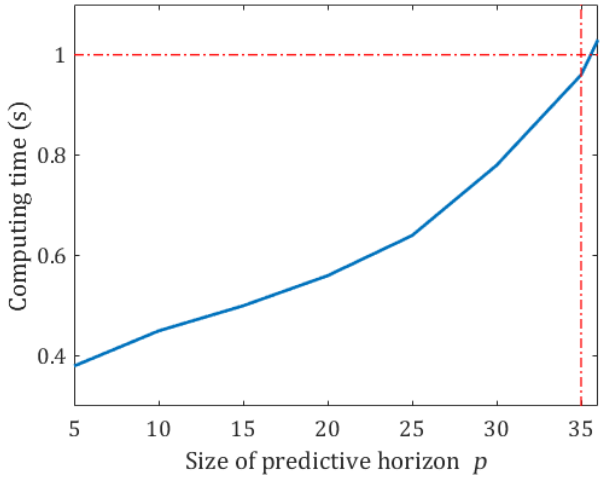
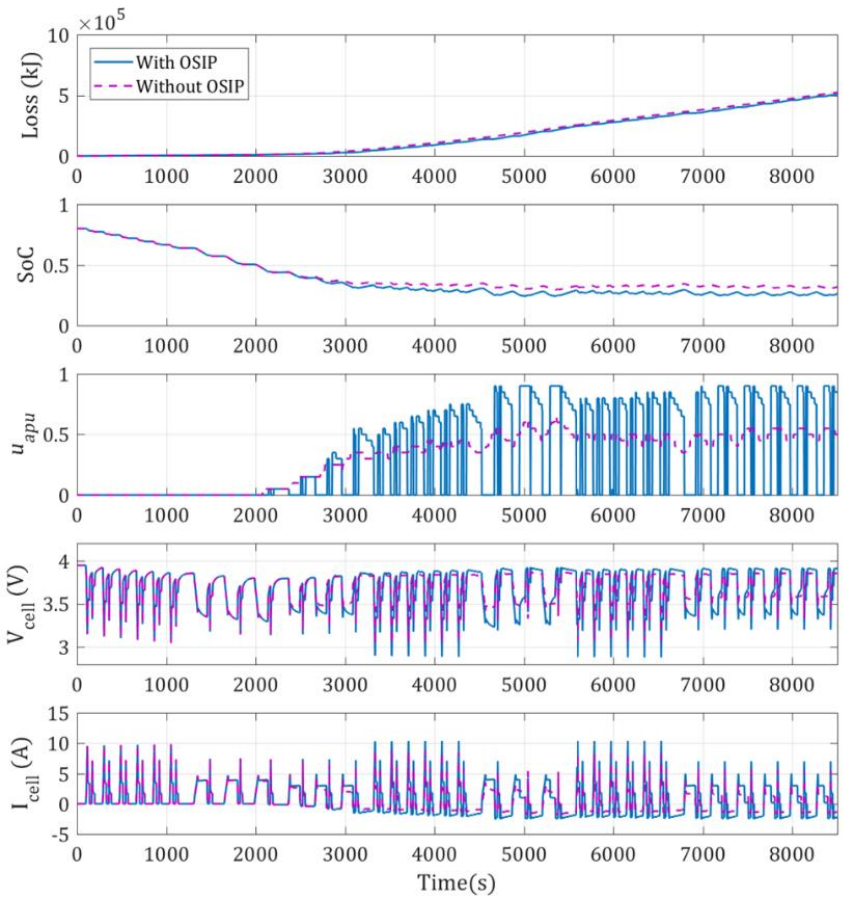


Fig. 5-6 Average computation time per step

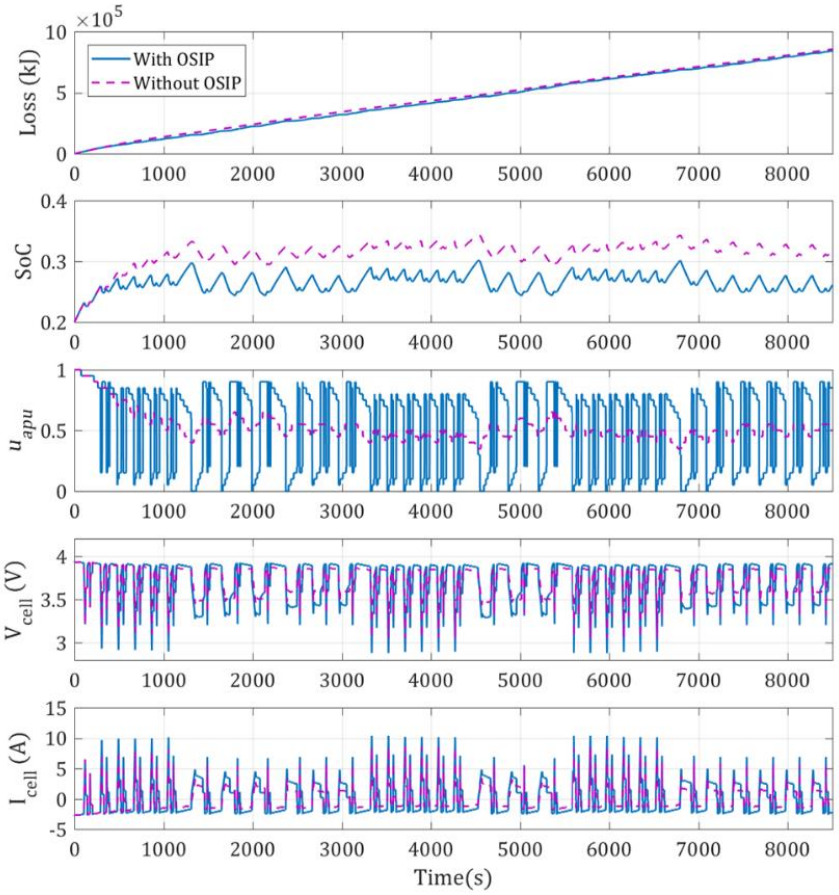
5.4.3 Vehicle system performance in real-time

The real-time performance of the connected system is evaluated and compared with the system using local control only. Different battery initially SoC values of 80% and 20% are investigated respectively. The proposed control method can maintain the HEV's components working within the proper range in real-time. Fig. 5-7 (a) shows the HIL test result in PBDC-I assuming the battery is initially in full charge. In this

condition, the connected system can save more energy than the one with local control only. Fig. 5-7 (b) shows the HIL test result in PBDC-I assuming the initial battery SoC is low due to some unknown error. The connected system can work properly and also outperform the one without OISP.



(a)



(b)

Fig. 5-7 Real-time performance: a) 80% initial SoC, b) 20% initial SoC

5.4.4 Robustness and repeatability

Working conditions vary among different scenarios in real-world driving; therefore, the test of robustness and repeatability is needed. The HEV systems in four PBDCs with different initial SoC values (80%, 50% and 20%) are evaluated. The results, as shown in Table.5-3, indicate that in all the scenarios under investigation, the proposed method can save energy from online control optimisation. The proposed method can reduce up to 13.06% of total energy loss. The highest energy saving rate are obtained over PBDC-III with the initial SoC of 80%.

Table.5-3 Vehicle performance over different Scenarios

Driving cyc.	Initial SoC	Control Method	Power loss(MJ)	Savings
BPDC-I	80%	CD/CS	526	-
	80%	OSIP	509	3.24%
	50%	CD/CS	695	-
	50%	OSIP	682	1.87%
	20%	CD/CS	856	-
	20%	OSIP	843	1.55%
BPDC-II	80%	CD/CS	139	-
	80%	OSIP	132	5.09%
	50%	CD/CS	310	-
	50%	OSIP	307	0.66%
	20%	CD/CS	474	-
	20%	OSIP	467	1.47%
BPDC-III	80%	CD/CS	95	-
	80%	OSIP	82	13.06%
	50%	CD/CS	261	-
	50%	OSIP	242	7.59%
	20%	CD/CS	424	-
	20%	OSIP	403	4.89%
BPDC-IV	80%	CD/CS	198	-
	80%	OSIP	185	6.70%
	50%	CD/CS	306	-
	50%	OSIP	295	3.67%
	20%	CD/CS	411	-
	20%	OSIP	399	2.77%

5.5 Summary

Model-based predictive energy management method has been researched in this chapter. A new online optimisation method named online swarm intelligent programming (OSIP) is proposed to fill the technical gap of solving nonlinear optimisation problems in real-time. Another contribution of this chapter is providing a demonstration of vehicle energy management control use a distributed system with V2X connectivity; this is extremely important for the future development of advanced vehicle control and optimisation technology taking full advantages of internet of the things (IoT). The conclusions drawn from the investigation of this chapter are:

- The global optimum control outputs can be determined using the proposed OSIP with much faster computing speed comparing with the bench-mark GA.

- OSIP can optimise the vehicle performance in real-time with a maximum prediction horizon size of 35 s, and the optimal control signal can be obtained and sent to relevant controllers within 1 s.

- The vehicle with OSIP outperforms the system without it in energy saving at all initial battery SoC level, and it has more potential in fuel-saving when initial battery SoC is high.

- The proposed energy management method is robust and reliable for energy

saving in all push back driving cycles, and up to 13% total energy loss can be saved via the proposed cyber-physical control.

This research has provided an efficient and rapid online optimisation method for solving nonlinear optimisation problems which are not limited in energy management of the hybrid off-highway vehicle but also for a wide range of engineering application. Some relevant research has implemented the proposed method in 1) the model-based predictive control of diesel engine air-path (Zhang, 2018); 2) online optimisation of gasoline direction injection engines ; 3) online optimisation of driver behaviour classification and forecasting model.

Chapter Six

MULTIPLE-STEP REINFORCEMENT LEARNING FOR 'MODEL-FREE' PREDICTIVE ENERGY MANAGEMENT CONTROL

This chapter proposes new 'multiple-step' reinforcement learning algorithms for 'model-free' predictive energy management, which enable all-life-long adaptive control optimisation without human knowledge of predictive modelling (Zhou *et al.*, 2019). Firstly, the energy management of the electrified vehicle is formulated as a Markov decision problem. The model-free predictive energy management strategy with three multiple-step reinforcement learning algorithms is then studied through the investigation of their learning performance in different optimisation scenarios. The real-time control feasibility and performance of the proposed energy management method are evaluated with the hardware-in-the-loop test.

6.1 Markov Decision Process of Energy Management

Energy management of hybrid vehicles can be formulated as a Markov decision process (MDP), which includes interactions between the energy management system (EMS) and its external environment (e.g. driver's power requirement and the

vehicle system)(Liu *et al.*, 2015a; Zou *et al.*, 2017; Hu *et al.*, 2018). The variables in the Markov decision process are the states (driver's power requirement, battery SoC), the actions (control commands of power units), and the rewards (fuel consumption and battery's SoC remaining). The main uncertainty involved in this Markov decision process is the driver's power demand in real-world driving condition, which varies according to different driver and road condition and it directly affects the power demand from the hybrid system. The interaction between the energy management system (EMS) and its external environment is a periodic process, as shown in Fig. 6-1, which includes three main steps repeating in each time interval: 1) a power demand from the driver is sent to the HEV and the EMS; 2) the EMS makes the decision of power distribution by observing the state variables; 3) on receiving the control command from the EMS, the vehicle outputs the reward variables.

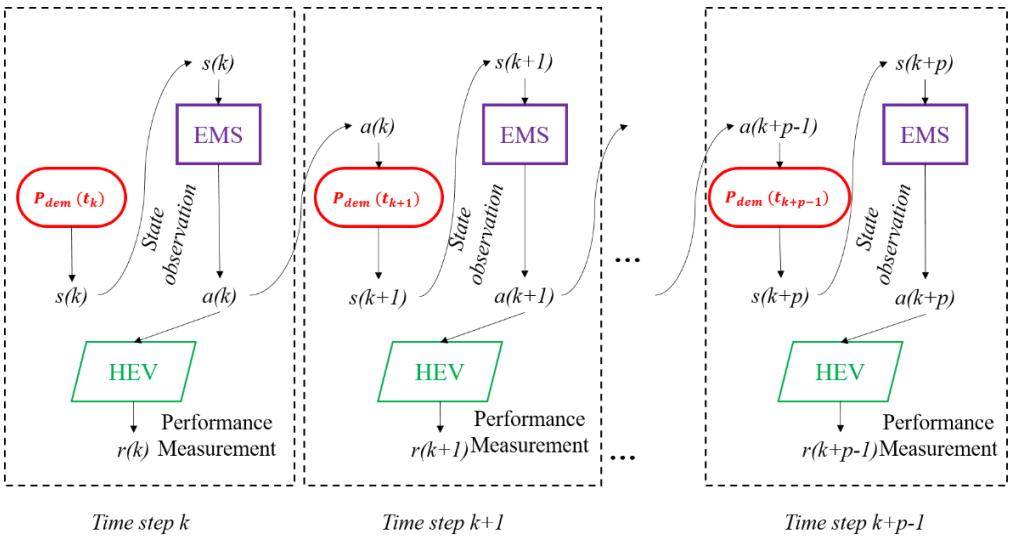


Fig. 6-1 Interaction between the EMS and the environment

6.2 Model-free Predictive Control System

A new ‘model-free’ predictive control system is researched to optimise the Markov decision process of energy management, which is a layered and distributed system as in Fig. 6-2. The system includes two main layers connected via a V2X network: a control layer located in the vehicle controller, and a learning layer in the server computer.

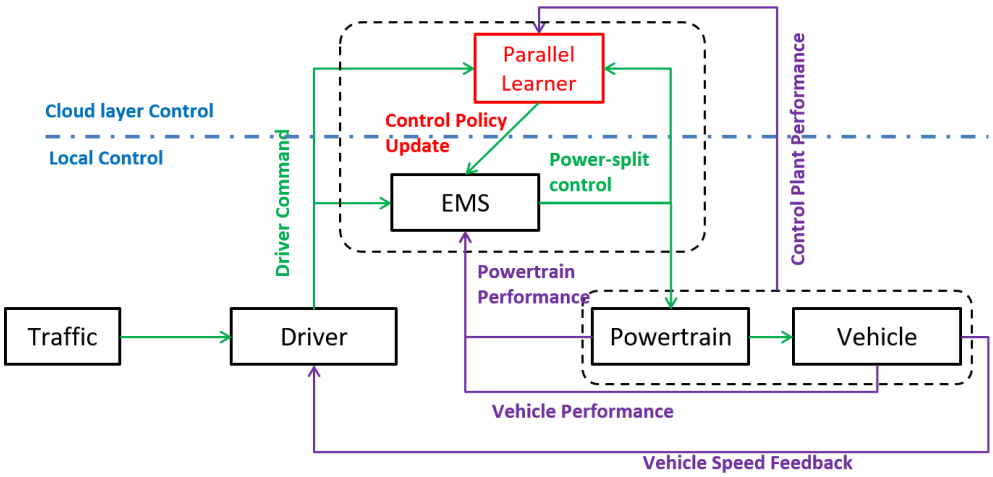


Fig. 6-2 Layered control system for model-free predictive control

The control layer with the EMS control policy allocates the power-flow based on the driver’s power demand and the observation of current vehicle states. A parallel learner in the learning layer performs a multiple-step reinforcement learning algorithm to update the control policy regularly. Thus, the control policy of the EMS can be adaptive to real-world driving as the result of online multi-objective optimisation over a predictive horizon.

The working process of the parallel learner is shown in Fig. 6-3, which introduces two additional elements to the MDP system, namely 'policy' and 'value function'. The 'policy' defines the control policy, which represents the learner's way of behaving at a given time. The 'value function' specifies the performance of the vehicle system with an immediate sense of 'reward' (i.e. a cost function of multi-objective) (Zou *et al.*, 2016).

The predictive models for the MDP are no longer needed in the proposed 'model-free' control method. With the experience from the past 'state', 'action' and 'reward' within the same step length as the predictive horizon, the reinforcement learning algorithm will learn how the action in the current step will affect the vehicle performance over the predictive horizon. The optimised control policy learnt through real-world interactions will ensure that each action will gradually lead to the optimal vehicle performance.

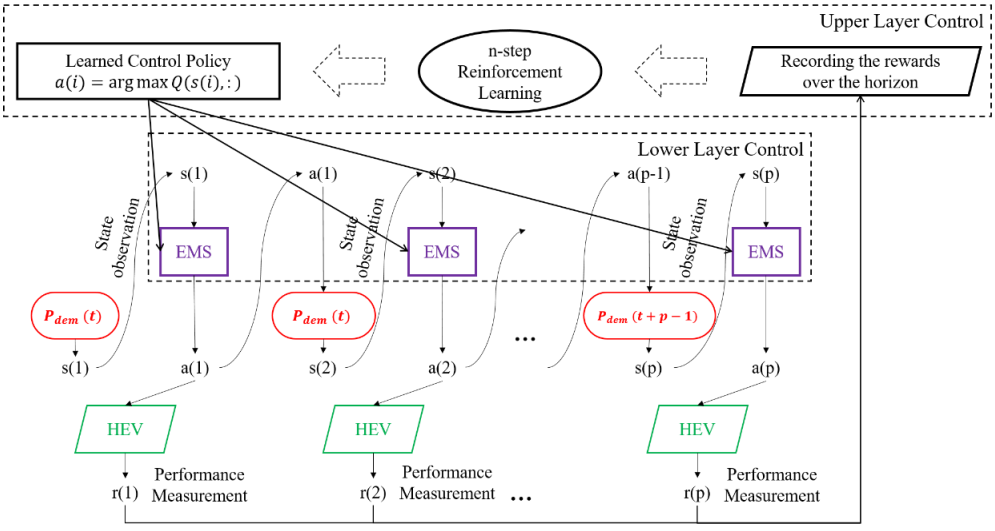


Fig. 6-3 Working process of the parallel learner in each time step

The rotational working process of parallel learner includes three main steps in each round of learning. Firstly, the learner receives the measured data from the vehicle including power demand, EMS command, energy consumption and battery SoC; and the data is recorded in the learning layer as a history set labelled by time. Secondly, the learning layer will optimise the control policy using a 'multi-step' reinforcement learning algorithm with the recorded data. Thirdly, the updated control policy will be sent back to the control layer before a new round of the learning process starts.

At t[he] time step, the state history set $\boldsymbol{S_h(t)}$, the action history set $\boldsymbol{A_h(t)}$, and reward history set $\boldsymbol{R_h(t)}$ used for reinforcement learning are:

$$\begin{cases} \boldsymbol{S_h(t)} = \{s(t-p) \quad s(t-p+1) \quad s(t-p+2) \quad \dots \quad s(t)\} \\ \boldsymbol{A_h(t)} = \{a(t-p) \quad a(t-p+1) \quad a(t-p+2) \quad \dots \quad a(t)\} \\ \boldsymbol{R_h(t)} = \{r(t-p) \quad r(t-p+1) \quad r(t-p+2) \quad \dots \quad r(t)\} \end{cases} \tag{6-1}$$

At $(t+1)^{th}$ the time interval, the state history set $\boldsymbol{S_h(t+1)}$, the action history set $\boldsymbol{A_h(t+1)}$, and reward history set $\boldsymbol{R_h(t+1)}$ used for reinforcement learning are rotationally updated as:

$$\begin{cases} \boldsymbol{S_h(t+1)} = \{s(t-p+1) \quad s(t-p+2) \quad s(t-p+3) \quad \dots \quad s(t+1)\} \\ \boldsymbol{A_h(t+1)} = \{a(t-p+1) \quad a(t-p+2) \quad a(t-p+3) \quad \dots \quad a(t+1)\} \\ \boldsymbol{R_h(t+1)} = \{r(t-p+1) \quad r(t-p+2) \quad r(t-p+3) \quad \dots \quad r(t+1)\} \end{cases} \tag{6-2}$$

where, p is the control horizon; $s(i)$ is the state value at *i-th* time interval ($i = t-p, \dots, t, t+1$); and $r(i)$ is the reward value at *i*-th time interval ($i = t-p, \dots, t, t+1$).

In this case, the state variables recorded in each time interval are defined as:

$$s(t) = [P_{dem}(t) \quad SoC(t)] \tag{6-3}$$

where, P_{dem} is the real-time power requirement from the driver and SoC is the battery's state of charge. To minimize the vehicle power loss and maintain the battery SoC level at the same time, a value function of the system reward is defined as a multi-objective function of overall vehicle power loss P_{loss} and battery's SoC. Here, the power loss P_{loss} and absolute SoC value lower than the reference SoC, $|SoC_{ref} - SoC(t)|$ are added as a penalty to the initial constant reward r_{ini} so that the learning system can remember which actions have been attempted and the rewards received after these actions:

$$r(t) = \begin{cases} r_{ini} - P_{loss}(t) & SoC(t) \geq SoC_{ref} \\ r_{ini} - P_{loss}(t) - \alpha\left|SoC_{ref} - SoC(t)\right| & SoC(t) < SoC_{ref} \end{cases} \tag{6-4}$$

where, SoC_{ref} is the reference battery SoC value for maintaining the battery's SoC within an acceptable SoC level, for the best performance and health condition of the battery, SoC_{ref} is chosen as 30%; α is a scale factor used to balance the consideration of the SoC level and power efficiency; $P_{loss}(t) = Loss_{eng}(t) + Loss_{batt}(t)$ is the total power loss of engine and battery; the power loss of the engine generator $Loss_{eng}(t)$ and power loss of the battery $Loss_{batt}(t)$ can be calculated by measuring the fuel consumption rate $\dot{m}_f$, engine torque T_{eng}, engine speed n_{eng} and battery current I_{batt} as:

$$\begin{cases} Loss_{eng}(t) = \dot{m_f}(t) \cdot H_f - \frac{T_{eng}(t) \cdot n_{eng}(t)}{9550} \\ Loss_{batt}(t) = R_{loss}(SoC) \cdot I_{batt}(t)^2 \end{cases} \quad (6\text{-}5)$$

where, H_f is the heat value of fuel (for diesel, $H_f = 44 \times 10^6$ J/kg), R_{loss} is equivalent internal resistant of the battery; and R_{loss} is a function of the battery's SoC.

The control policy for the model-free predictive control is to determine the optimal control action based on the current state observation, state history, and reward history as:

$$a(t) = \Pi(s(t) \quad Q(\boldsymbol{S_h}, \boldsymbol{A_h}, \boldsymbol{R_h})) \quad (6\text{-}6)$$

where, $a(t)$ is the current control action/command; $s(t)$ is the current vehicle state observed, $\boldsymbol{S_h}$, $\boldsymbol{A_h}$, and $\boldsymbol{R_h}$ are the state history set, action history set, and reward history set respectively; Q is the Q-table storing the relationship among state, action and reward. The reinforcement learning aims to optimize the control policy Π, and the core of the optimisation is to update Q-table with reinforcement learning.

6.3 Multiple-step Reinforcement Learning Algorithm

6.3.1 Fundamental of reinforcement learning

Reinforcement learning is an episode-based repeatable process. As shown in Fig.

6-4, each episode has two main procedures: the action-taken procedure and the policy-learning procedure.

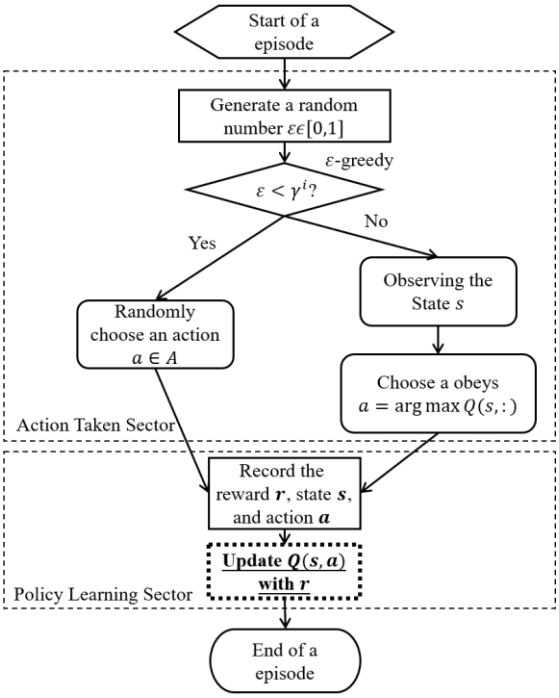


Fig. 6-4 Flowchart of Q-learning procedure in one episode

In the action-taken procedure, the action signal of the power distribution command is taken with the most widely used ε-greedy policy: a random number $\varepsilon\epsilon[0,1]$ is generated and compared with a decreasing number γ^i (where $\gamma\epsilon[0,1]$ and *i* is the generation of episodes). When ε is greater than γ^i, action will be taken by data-sampling from the action set; otherwise, the action will be taken by observing the action with the maximum Q value in the current state as:

$$a(t) = \arg\max Q(s(t),:) \quad a(t) \in \boldsymbol{A} \tag{6-7}$$

where, $a(t)$ is the action at t-th time interval and $\boldsymbol{A}$ is the action set. In this study,

the action is the control command for EGU $u_{egu} \in \boldsymbol{A} = \{0: 0.05: 1\}$ as in (Zhou *et al.*, 2018). Q is a table storing the system performance when taking different actions 'a' at different state 's'. The Q table will be optimised and updated in the learning sector.

The learning procedure records the current state $s(t)$, current action $a(t)$ and the reward value $r(t)$ feedback from the HEV system and updates the Q table $Q(s(t), a(t))$ with the learning strategy based on the reward r. For one-step reinforcement learning, the Q value $Q(s(t), a(t))$ for the current state $s(t)$ and the current action $a(t)$ updates directly from current system reward $r(t)$ as:

$$Q(s(t), a(t)) \leftarrow Q(s(t), a(t)) + \alpha[r(t) + maxQ(s(t+1), :) - Q(s(t), a(t))] \quad (6\text{-}8)$$

where, α is the learning rate, $maxQ(s(t+1), :)$ is the estimated Q value for the next step, which is obtained by looking up the old Q table. One-step Q learning is based on the Bellman's 'principle of optimality'(Bellman, 2010) which indicates that for an optimal Markov decision chain, the Q value of each time interval can be linked directly with the reward as (Liu *et al.*, 2015b; Sutton and Barto, 2017):

$$Q^*(s(t), a(t)) = r(t) + Q^*(s(t+1), a(t+1)) \quad (6\text{-}9)$$

where, $Q^*(s(t), a(t))$ and $Q^*(s(t+1), a(t+1))$ are the optimal Q value for t-th time step and $(t+1)$-th time step. Theoretically, when the optimal Q table is obtained, the Q value will not be updated anymore. Therefore, the learning strategy illustrated in Equation (7-8) can secure that the control policy learned from

reinforcement learning converges to the optimal control policy.

6.3.2 Multi-step learning strategy

The multi-step learning strategy is developed based on the one-step learning strategy in Equation (7-8), and in each time step it enables the capability of multi-step prediction by learning from the history sets with the same length as the predictive horizon including state set $\boldsymbol{S}_h$, action set $\boldsymbol{A}_h$ and reward set $\boldsymbol{R}_h$. The Q value of each time step within the history set is updated as:

$$Q(s(i), a(i)) \leftarrow \Phi(\boldsymbol{Q}, \boldsymbol{S}_h(t), \boldsymbol{A}_h(t), \boldsymbol{R}_h(t)] \tag{6-10}$$

where $s(i) \in \boldsymbol{S}_h(t)$ and $a(i) \in \boldsymbol{A}_h(t)$ are the state and action value at the i-th time step ($i = 1,2,3 \dots p$) of the history set collected t-th time interval; $\boldsymbol{S}_h(t)$, $\boldsymbol{A}_h(t)$, and $\boldsymbol{R}_h(t)$ are the history set of state, action and reward collected at t-th time interval; $\boldsymbol{Q}$ is the Q-table before updating; Φ is the multi-step learning strategy and three multi-step learning strategies are introduced as follows. To clearly illustrate how the multi-step learning strategies work, some concepts are defined:

Definition 1. 'Data-set package' with notation $\boldsymbol{D}(t)$ defines the history sets of state, action and reward used for multi-step learning at t-th time interval as $\boldsymbol{D}(t) = \{\boldsymbol{S}_h(t), \boldsymbol{A}_h(t), \boldsymbol{R}_h(t)\}$.

Definition 2. 'Elements' with notation $d(i)$ defines the components within the data

set package; where, $i \in [1, p]$ is the index of elements which will be defined in Definition 3. Each element includes the value of the state, action and rewards within the data-set package as $d(i) = \{s(i), a(i), r(i)\}$.

Definition 3. 'Order Index' with notation i defines the order of each element in the data-set package used for reinforcement learning at each time interval. For example, the order index of element $d = \{s(t-p+1), a(t-p+1), r(t-p+1)\}$ in the data-set package $\boldsymbol{D}(t)$ is 3, but in data-set package $\boldsymbol{D}(t+1)$, its order index will be 2.

Definition 4. 'Table-Lookup Index' with notation τ defines the position of each state and action in the set of state and action, which are used to find the respective Q value of each state and action from the Q table. For example, as the action set is $\mathrm{A} = \{0, 0.05, 0.10, 0.15 \ldots .0.95, 1\}$, the table lookup index of action variable $a = 0.05$ is $\tau(a = 0.05) = 2$; a similar principle can be applied to state variables.

a) Sum-to-Terminal Strategy (S2T)

The first multi-step learning strategy named 'Sum-to-Terminal (S2T)' is a straightforward strategy which connects the current Q value at each step of the predictive horizon $Q(s(i), a(i))$ to the terminal Q value $Q(s(p), a(p))$ directly with the sum of the reward during the time interval $\sum_{j=1}^{p} r(j)$ as:

$$Q(s(1),a(1)) \leftarrow Q(s(1),a(1)) + \alpha[\sum_{j=1}^{p} r(j) + maxQ(s(p),:) - Q(s(1),a(1))]$$

(6-11)

where α is the learning rate; $maxQ(s(p),:)$ is the estimated terminal Q value which is obtained by looking up the old Q table.

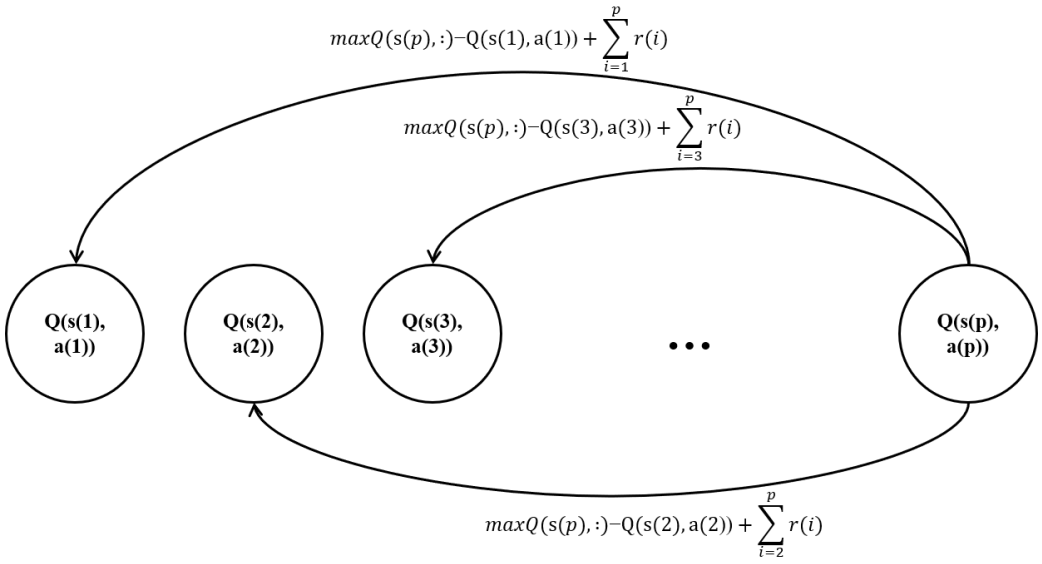


Fig. 6-5 Sum-to-Terminal (S2T) strategy for Q table updating

The relationship between the Q values of each time step to the terminal Q value is established as the extension of Equation (6-10) as shown in Fig. 6-5. In general, the S2T strategy could be summarised as:

$$Q(s(i),a(i)) \leftarrow Q(s(i),a(i)) + \alpha[\sum_{j=i}^{p} r(i) + maxQ(s(p),:) - Q(s(i),a(i))] \quad (6\text{-}12)$$

where $i = 1,2 \ldots p$ is the order index of each element within the data-set package. The pseudo-code for the S2T strategy is provided in Fig. 6-6.

Multiple-step Reinforcement Learning with S2T Strategy

- *Load Timer signal*
- *Load* $\boldsymbol{Q(s,a)}$*, S, A, R*
- *Choose action* $a \in A$ *based on* $\varepsilon - greedy$ *policy*
- *Recording s, a, r within predictive horizon p in* $\boldsymbol{S_h}$, $\boldsymbol{A_h}$, $\boldsymbol{R_h}$

% start of the learning with S2T %

- *For* $k = 1:p$
  - *Let the order index* $i = k$
  - *Find the table-lookup index of* $s(i)$, $a(i)$ *in S and A based on 'Nearest-Neighborhood policy'*
  - *Find the table-lookup index of* $s(p)$ *in S based on 'Nearest-Neighborhood policy'*
  - *Update* $Q(s(i), a(i))$ *using Equation (12) with max* $Q(s(p))$ *and* $\sum_{j=i}^{p} r(j)$
- *End for*

% end of the learning with S2T %

Fig. 6-6 Pseudo-code for multi-step reinforcement learning with S2T strategy

b) Average-to-Neighbour Strategy (A2N)

The 'Average-to-Neighbour (A2N)' is proposed by building the relationship of each step by updating the Q value with the average reward of the predictive horizon $\frac{1}{p}\sum_{i=0}^{p} r(t+i)$ as shown in Fig. 6-7. The A2N strategy connects the action taken in each time step with the global performance over the predictive horizon, by replacing the reward in each time step for one step reinforcement learning, with the average reward of the predictive horizon as:

$$Q(s(i),a(i)) \leftarrow Q(s(i),a(i)) + \alpha[\frac{\sum_{j=1}^{p} r(j)}{p} + maxQ(s(i+1),:) - Q(s(i),a(i))] \quad (6\text{-}13)$$

where $i = 1,2 \ldots p$ is the order index of each element within the data-set package.

The pseudo-code for the A2N strategy is provided in Fig. 6-8.

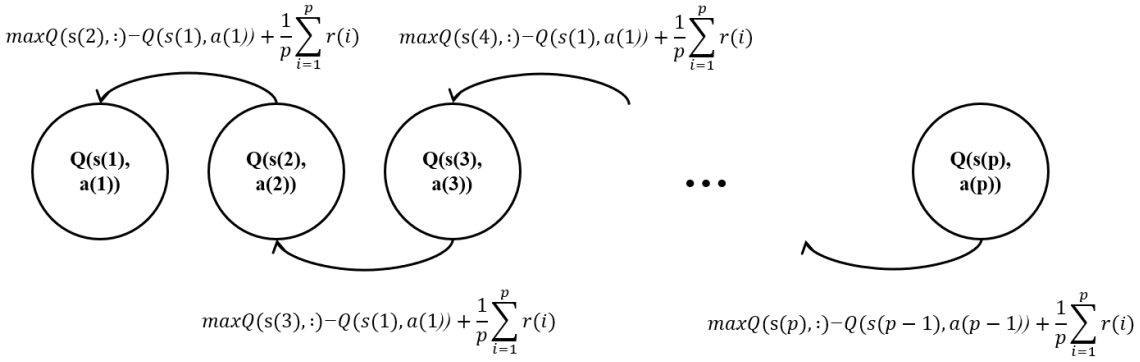


Fig. 6-7 Average-to-Neighbour (A2N) strategy for Q table updating

Multiple-step Reinforcement Learning with A2N Strategy

- *Load Timer signal*
- *Load* $\boldsymbol{Q(s,a)}$***, S, A, R,*** *p*
- *Choose action* $a \in A$ *based on* $\varepsilon - greedy$ *policy*
- *Recording s, a, r within predictive horizon p in* $\boldsymbol{S_h}$, $\boldsymbol{A_h}$, $\boldsymbol{R_h}$

% start of the learning with A2N

- *For* $k = 1:p$
  - *Let the order index*
  - *Find the table-lookup index of* $s(i)$, $a(i)$ *in* ***S*** *and* ***A*** *based on 'Nearest-Neighbourhood policy'*
  - *Find the table-lookup index of* $s(p)$ *in* ***S*** *based on 'Nearest-Neighbourhood policy'*
  - *Update* $Q(s(i),a(i))$ *using Equation (13) with max* $Q(s(i+1))$ *and* $\left(\sum_{i=1}^{p} r(i)\right)/p$
- *End for*

% end of the learning with A2N

Fig. 6-8 Pseudo-code for multi-step reinforcement learning with A2N strategy

c) Recurrent-to-Terminal Strategy (R2T)

As shown in Fig. 6-9, the 'Recurrent-to-Terminal (R2T)' strategy is developed as an extension of the S2T based on the Q(λ) learning algorithm (Peng and Williams, 1996). This updates the Q value recurrently from the current step to the terminal step, with consideration of the gap from the current time step i $(i = 1,2 \ldots p)$ to the terminal time step p by introducing the discount factor λ to scale the reward value in the different time step as:

$$Q(s(i), a(i)) \leftarrow Q(s(i), a(i)) + \alpha \cdot \sum_{j=i}^{p} V(j) \quad (6\text{-}14)$$

where $i = 1,2 \ldots p$ is the order index of each element within the data-set package; $V(j)$ is the value function for R2T for the element with order index j, which is defined as:

$$V(j) = \lambda^{j} r(j) + \lambda^{j+1} \mathrm{maxQ}\big(s(j+1)\big) - Q(s(j), a(j)) \quad (6\text{-}15)$$

where λ is the discount factor, $r(j)$ is the reward at (j)-th time interval. The pseudo-code for the R2T strategy is provided in Fig. 6-10.

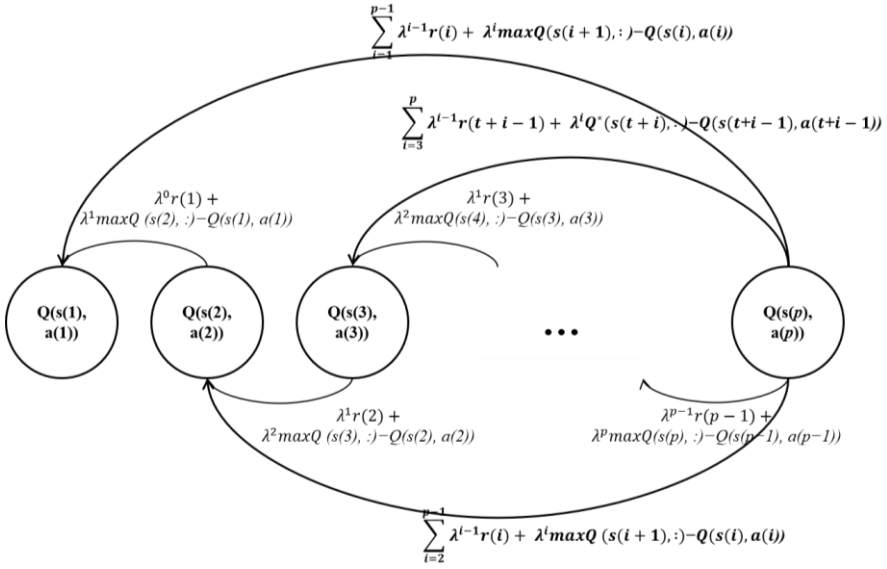


Fig. 6-9 Recurrent-to-Terminal (R2T) strategy for Q table updating

Multiple-step Reinforcement Learning with R2T Strategy

- *Load Timer signal t*
- *Load* $\boldsymbol{Q}(\boldsymbol{s}, \boldsymbol{a})$*, S, A, R, p*
- *Choose action* $a \in \boldsymbol{A}$ *based on* $\varepsilon - greedy$ *policy*
- *Recording s, a, r within predictive horizon p in* $\boldsymbol{S_h}$, $\boldsymbol{A_h}$, $\boldsymbol{R_h} \leftarrow$

% start of the learning with R2T %

- *For* $k = 1{:}\,p$
  - *Let the order index* $i = k$
  - *Find the table lookup index of* $s(i)$, $a(i)$ *in S and A based on 'Nearest-Neighborhood policy'*
  - *For* $m = 1{:}\,k$
    - *Let the order index* $j = m$
    - *Find the table-lookup index of* $s(j)$, $a(j)$ *in S and A based on 'Nearest-Neighborhood policy'*
    - *Find the table-lookup index of* $s(j+1)$ *in S based on 'Nearest-Neighborhood policy'*
    - *Calculate* $V(j)$ *using Equation (15) and reward value* $r(j)$ *with order index j*
  - *End for*
  - *Update* $Q(s(i), a(i))$ *using Equation (14)*
- *End for*

% end of the learning with R2T %

Fig. 6-10 Pseudo-code for multi-step reinforcement learning with R2T strategy

6.4 Performance and Discussions

6.4.1 Learning performance

According to the research in Chapter Five, the maximum predictive length of the model-based method using ES910 is 35 steps (Zh*ou et al.*, 2018). Therefore, the performance evaluation starts from the model-free predictive energy management with 35 steps. The system efficiency of powertrain energy conversion is calculated for every 4252s (total time of the given driving cycle) with the measured data of equivalent power loss for fuel consumption, battery power loss, and the power used by the traction motor, which is used to evaluate the learning performance. The process of 'learning from scratch' (initial Q table is a zero-set) is monitored, and the

improvement of the vehicle efficiency during the learning process is shown in Fig. 6-11. It indicates that all three proposed learning strategies in chapter 6.3.2 have the capability of continuous improvement with self-learning, and this demonstrates that the model-free predictive system works for the desired purpose. In Fig. 6-11, the process of 'learning from scratch' with each learning strategy can be roughly classified into two stages, i.e., a rapid improving stage (at the beginning) and a slowly improving stage (after sufficient experience for Q-table filling). To generate an exponential, $\varepsilon - \mathrm{greedy}$ is used for reducing function for controlling the probability for self-exploration, as in (Liu *et al.*, 2017), which can lead to a logarithmical improvement in system efficiency theoretically. The discrete sampling of the system efficiency, therefore, generates an inflect point at about 5 hours for the 35 step Q-learning. Among the three proposed learning strategies, the 'Recurrent-to-Terminal (R2T)' strategy is the most effective strategy for model-free predictive energy management with the prediction horizon of 35 steps.

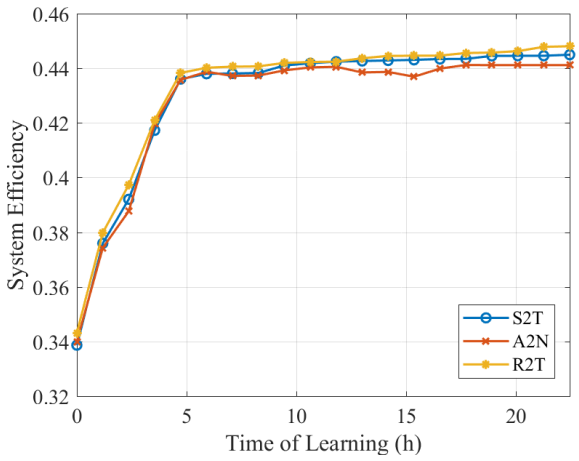
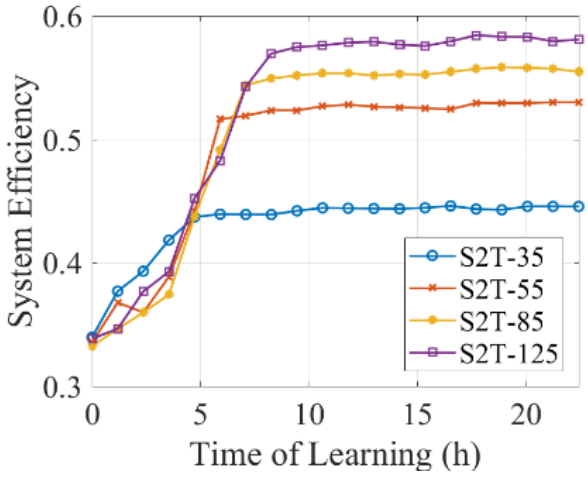
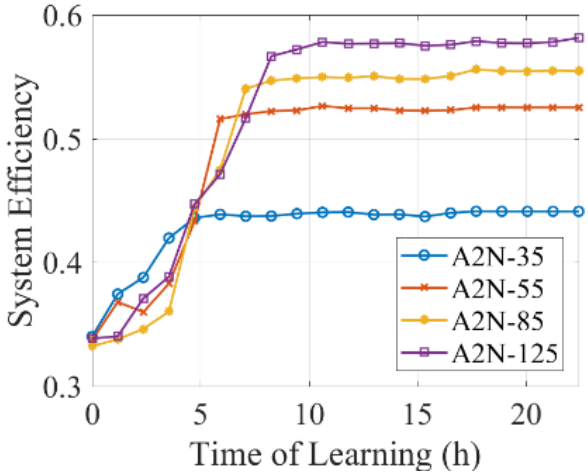


Fig. 6-11 Learning performance of different learning strategies (35-step)

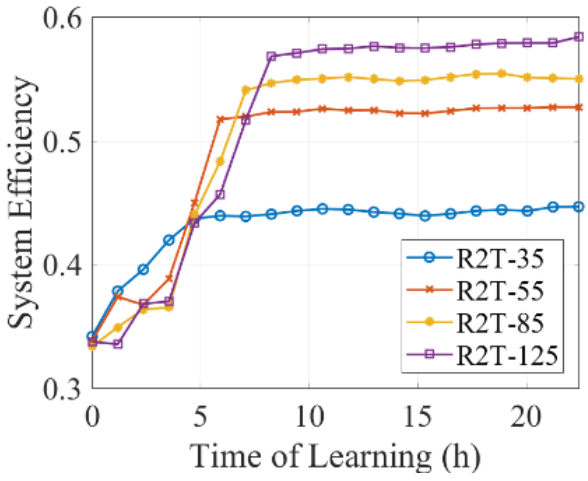
For HEV energy management, the longer the predictive horizon is, the better the system performance will be (Huang *et al*>, 2017). The performance of the model-free predictive energy management has been investigated by tracking the improvement of the system's efficiency with the prediction length increasing, and the results are presented in Fig. 6-12.



(a)



(b)



(c)

Fig. 6-12 Performance of different prediction length: (a)-S2T strategy; (b)-A2N strategy; (c)-R2T strategy

The system efficiencies after 24 hours reinforcement learning of Driving Cycle 1 with four different prediction step lengths are listed in Table.6-1.

Table.6-1 System energy efficiency of the proposed learning strategies with different prediction length

	Prediction step length			
	35	**55**	**85**	**125**
R2T	44.66%	53.00%	55.47%	58.38%
A2N	44.12%	52.53%	55.46%	58.14%
S2T	44.55%	52.70%	54.99%	58.08%

It is shown that all three proposed learning strategies have the capability of improving the system performance by increasing the prediction length. In each selected prediction size, the R2T strategy outperforms the other strategies by achieving

higher system efficiency at the end of the learning process.

6.4.2 Real-time Implementation Feasibility

The computational cost is a natural concern for real-time implementation, and the prediction length is the most concerning factor which affects the computational cost (Hu, Wang and Tang, 2017). The computational cost of the proposed method with respect to the size of the prediction horizon is hereby investigated using ES910. The average computational cost per time step including the data communication, is shown in Fig. 14. It indicates that while the augmented prediction size leads to increased computational load, a prediction step size of less than 65 steps can make the controller with the A2N and S2T strategy implementable in real-time; while 60 steps can make the model-free energy management with the R2T strategy implementable in real-time, as the computing time is less than the sampling time of 1 second.

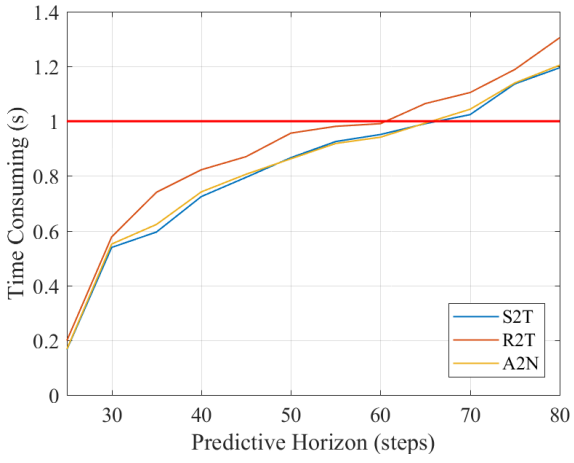


Fig. 6-13 Average computation time per step for different predictive horizons

The model-free predictive energy management with the R2T learning strategy outperforms other proposed strategies by achieving the best vehicle system efficiency with the same prediction length (as discussed in chapter 6.4.1). The advancement of the R2T strategy is that it includes more iteration loops (as shown in Fig. 6-10) to build up a more systematic learning system for storing more 'learning experience'. This additional complexity in contract costs more computational resources and shortens its maximum predictive step length in real-time. The full performance in real-time of the different learning strategies is investigated by monitoring the learning performance of each strategy with its maximal step length in real-time (e.g. 60 steps for R2T, 65 steps for A2N and S2T) and the results are shown Fig. 6-14. In real-time, although the maximum prediction step size of the R2T strategy is shorter than the other proposed strategies, it still outperforms others by achieving a better system efficiency at the end of the learning process and tends to be a further improvement.

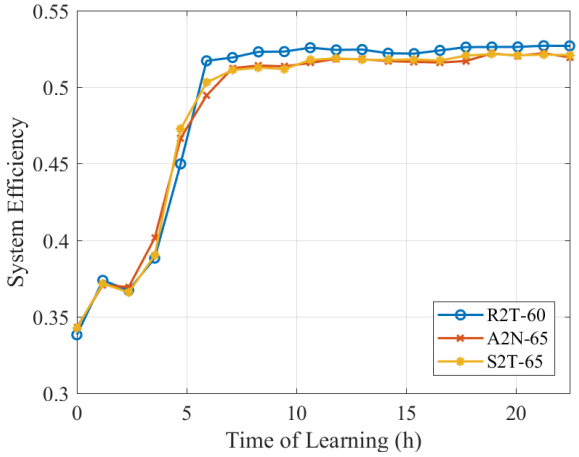
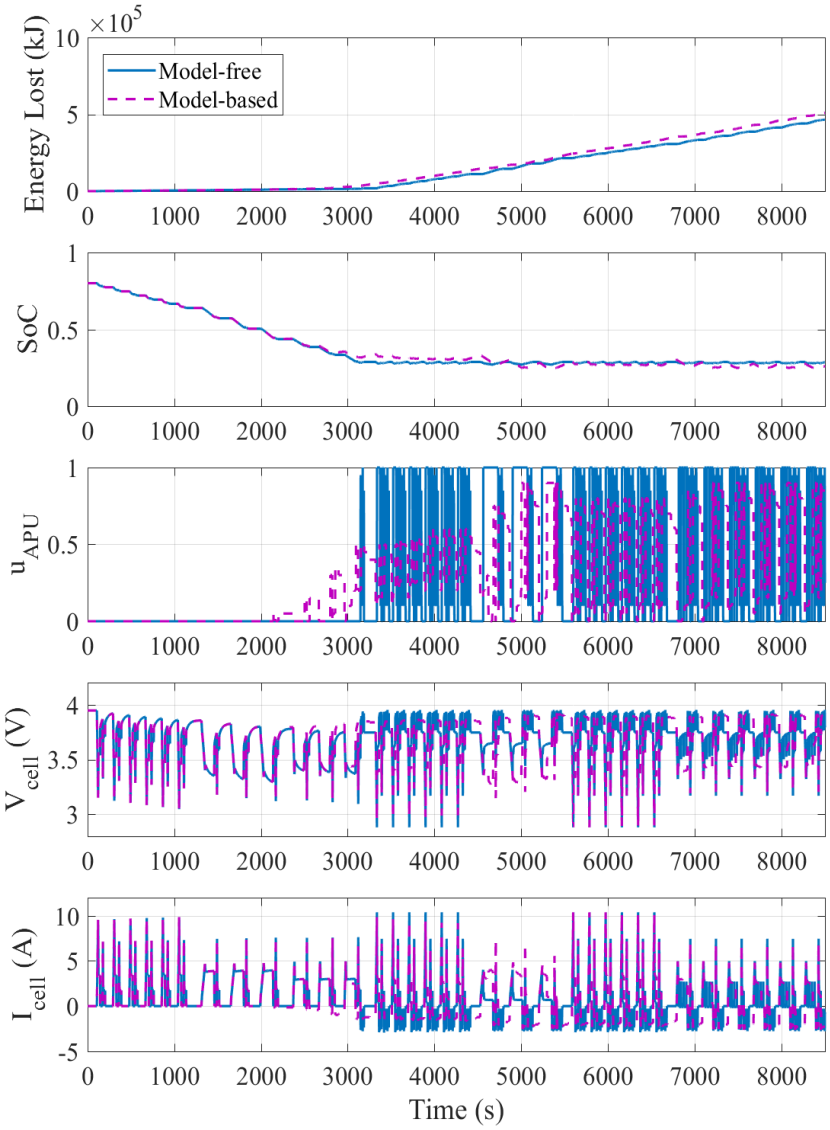


Fig. 6-14 Full performance in real-time of different learning strategy

6.4.3 Performance compared with the MPC

The real-time performance of model-free predictive energy management (with the R2T learning strategy) for the electrified aircraft-towing tractor is compared with the model-based method. The predictive horizon length for both the model-free method and the model-based method is 35 steps. Different battery initial SoC value of 80% and 20% are investigated under Real-World Cycle-1 in Fig. 6-15 (a) and (b) respectively.


Energy Lost (kJ)
×10^5
Model-free
Model-based
SoC
u_APU
V_cell (V)
I_cell (A)
Time (s)

(a)

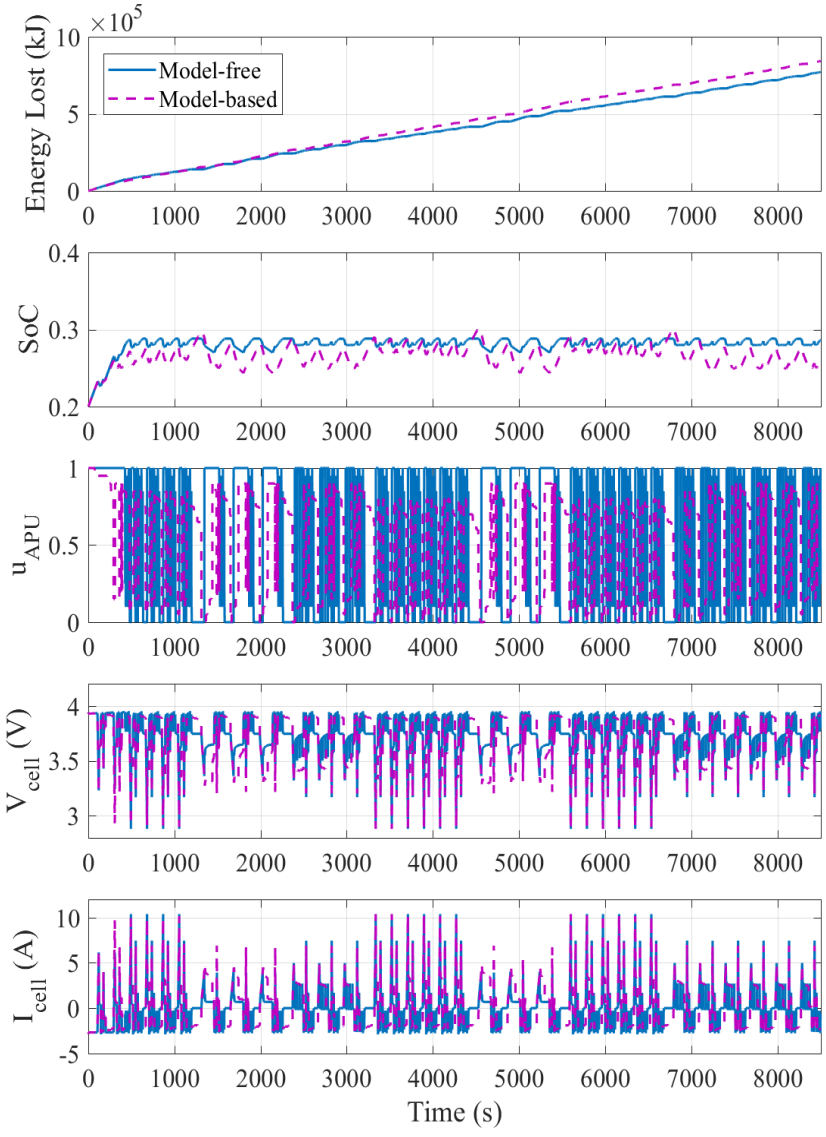


Fig. 6-15 HiL testing results: (a) Initial SoC=80%; (b) Initial SoC=20%

The HiL testing results for the model-free method are shown in solid blue lines, and the results of the model-based method are presented in magenta dashed lines. The time history of the energy loss, the battery's SoC, the engine command, and the battery cell's current/voltage are compared. The proposed control method can maintain the HEV's components working within the proper range in real-time. The model-free predictive energy management method outperforms the model-based

method, in terms of both energy consumption and the remaining battery's SoC.

The working condition varies among different scenarios; therefore, a test for robustness and repeatability is needed. The HEV systems in three driving cycles with different initial battery SoC values (80%, 50% and 20%) are evaluated. To make the comparison fair, the performance of the model-free method and the model-based method are firstly compared with the same prediction length of 35 steps (the maximal real-time step size for model-based method) in Table.6-2. Both the model-based and the model-free method can maintain the battery SoC at around 30% as predefined in the objective function. This will make full use of the electric energy in a plug-in hybrid tractor so that the vehicle users can save fuel. This will also help to reduce the emissions by reducing the operation time of the internal combustion engine.

The results indicate that in all the scenarios under investigation, the model-free method outperforms the model-based method in energy saving. With the same prediction length, the model-free method can save up to 10.54% energy compared with the model-based method. The full performance in real-time of the model-free method (with a prediction length of 60 steps) is then investigated and compared with the model-based method (with prediction length of 35 steps) in **Error! Reference source not found.**. It indicates that the upgrading of the model-free method to its full performance capacity leads to further performance improvement. Compared with the model-based method (maximum prediction step of 35 steps in real-time), the

model-free method (maximum prediction step of 60 steps in real-time) shows its advancement by at least 7.79% energy. The highest energy saving range is obtained over the Real-world Cycle-2 with the initial battery SoC of 50%.

Table.6-2 Performance of model-free method and model-based method (same prediction length)

Cycle	Initial SoC	Method	End SoC	Energy Usage (MJ)	Saving
Driving Cycle 2	80%	Model-based	29.74%	844.08	-
	80%	Model-free	28.68%	800.95	5.11%
	50%	Model-based	29.69%	1016.96	-
	50%	Model-free	28.66%	951.6	6.43%
	20%	Model-based	29.73%	1177.86	-
	20%	Model-free	28.68%	1105.35	6.16%
Driving Cycle 3	80%	Model-based	29.74%	293.08	-
	80%	Model-free	28.81%	262.17	10.54%
	50%	Model-based	29.74%	469.34	-
	50%	Model-free	28.81%	423.12	9.85%
	20%	Model-based	29.74%	628.75	-
	20%	Model-free	28.81%	579.37	7.85%
Driving Cycle 4	80%	Model-based	29.74%	234.9	-
	80%	Model-free	28.81%	213.5	9.11%
	50%	Model-based	29.68%	394.2	-
	50%	Model-free	28.81%	366.71	6.97%
	20%	Model-based	29.68%	555.34	-
	20%	Model-free	28.81%	520.45	6.28%

Table.6-3 Performance of model-free method and model-based method

(full performance - 60steps for model-free, 35steps for model-based)

Cycle	Initial SoC	Method	End SoC	Energy Usage (MJ)	Saving
Driving Cycle 2	80%	Model-based	29.74%	844.08	-
	80%	Model-free	28.68%	778.25	7.79%
	50%	Model-based	29.69%	1016.96	-
	50%	Model-free	28.66%	922.41	9.29%
	20%	Model-based	29.73%	1177.86	-
	20%	Model-free	28.68%	1071.34	9.04%
Driving Cycle 3	80%	Model-based	29.74%	293.08	-
	80%	Model-free	28.81%	254.80	13.06%
	50%	Model-based	29.74%	469.34	-
	50%	Model-free	28.81%	401.78	14.39%
	20%	Model-based	29.74%	628.75	-
	20%	Model-free	28.81%	552.29	12.16%
Driving Cycle 4	80%	Model-based	29.74%	234.9	-
	80%	Model-free	28.81%	208.34	11.31%
	50%	Model-based	29.68%	394.2	-
	50%	Model-free	28.81%	354.75	10.08%
	20%	Model-based	29.68%	555.34	-
	20%	Model-free	28.81%	505.15	9.04%

6.5 Summary

A new model-free predictive energy management method for a hybrid off-highway

vehicle has been studied and compared with the conventional model-based energy management method. Three different multi-step reinforcement learning strategies are proposed and investigated for model-free predictive control. The learning performance and real-time implementation feasibility of the model-free method are evaluated in a HiL testing system. The conclusions drawn from the investigation are as follows:

- The model-free predictive energy management method can improve the energy efficiency of the hybrid off-highway vehicle after a certain time length of vehicle operation, through the proposed learning strategies.

- The proposed learning strategies can optimise the control policy in real-time with a maximum prediction length of 65 steps. The optimal control policy can be obtained and implemented in an ES910 controller within 1 second;

- The proposed R2T learning strategy is the most effective multi-step reinforcement learning strategy for the model-free predictive energy management in the case study. It outperforms other proposed strategies in terms of same prediction step length and full performance in real-time;

- The proposed model-free predictive energy management method is robust for energy saving, and it outperforms the conventional model-based method by saving at least 7.8% energy.

Chapter Seven

CONCLUSION AND FUTURE WORK

This thesis demonstrates a methodology for solving challenging engineering problems with the help of cutting-edge technologies of artificial intelligence. This chapter draws the conclusions from this PhD research, summarises the research impact, and discusses possible future research activities.

7.1 Conclusions

Artificial intelligent (AI) methodologies have been developed for design optimisation and optimal energy management control of an electrified off-highway vehicle. ***AI*** technologies (including evolutionary computing and reinforcement learning) and ***cyber-physical control*** technologies have been used for this research. The conclusions of this thesis are drawn as follows, corresponding to the relevant chapters:

a) The proposed multi-objective optimisation using the CAPSO algorithm is an effective method for the design of an electrified off-highway vehicle.

According to the research carried out in Chapter 4, the proposed design optimisation

method with the CAPSO algorithm outperforms the conventional APSO algorithm by improving the consistency of the optimisation results and the probability of accessing the global optima. Specified conclusions drawn from this research are:

- The CAPSO algorithm can reduce the probability of local optima falling by introducing a dynamic attraction factor (which is a constant for the APSO) with chaotic maps.

- The logistic map is the most effective mapping strategy for the CAPSO, according to a comprehensive evaluation, including Monte Carlo analysis and reputation evaluation.

- When the weight value of the cost-function changes from 0 to 0.5, a higher energy loss reduction rate can be achieved with less increasing of components size.

b) The proposed online swarm intelligent programming enables real-time nonlinear model-based predictive control of the energy-flow for the hybrid off-highway vehicle.

Chapter 5 carries out the research concerning new nonlinear model-based predictive control; where online swarm intelligent programming (OSIP) is proposed to operate online nonlinear optimisation in real-time. The findings from this research are as

follows:

- The proposed OSIP using the CAPSO algorithm for online nonlinear optimisation can determine the optimal control outputs with a much faster computing speed compared to the bench-mark genetic algorithm (GA).

- The increasing of the prediction length increases the computational effort of the OSIP. The maximum prediction length can be 35 steps when using the ES910 rapid control prototype. The prediction length could be extended with the upgrading of vehicle controllers and computing devices in the future.

- Energy management with the proposed OSIP can save more energy when the initial battery SoC is relevant high; which can save up to 13% energy among the driving cycles in this research.

c) The proposed model-free predictive energy management method has the capability of online control optimisation without prediction models.

New ‘multiple-step’ reinforcement learning algorithms for ‘model-free’ predictive energy management have been researched in Chapter 6. These enable all-life-long adaptive control optimisation without human knowledge of predictive modelling. Conclusions drawn from the investigation are:

- Model-free predictive energy management with a longer prediction step

length provides better performance in energy saving. The maximum energy efficiency will be achieved when the prediction step length is equal to the time length of a driving cycle; however this is not realistic for real-world driving.

- The computation effort increases as the prediction step length increases. The maximum number of prediction steps of the model-free control is 65 steps when using the ES910 prototype controller, which is longer than for the model-based control. This is because the model-free predictive control can save the computing resources which are used for prediction models and model-based optimisation in model-based predictive control.

- Multi-step reinforcement learning with R2T strategy is the most effective method for model-free predictive energy management. It can save at least 7% of energy compared with the model-based predictive control.

7.2 Innovation and Impact Summary

This four-year PhD research delivers three main novel features, which are:

- **Chaos-enhanced swarm intelligence for design optimisation of an electrified off-highway vehicle.** This work provides a new solution for design optimisation using artificial intelligence rooted from the well-adapted accelerated particle swarm optimisation (APSO) algorithm. This research investigates four

chaotic mapping strategies to improve the consistency of vehicle design optimisation for the first time. The probability of achieving a global optimum result has been improved by 200% over the conventional APSO. The research outcomes have been published with ***Applied Energy*** (Zhou *et al.*, 2017) which is a top journal (*IF=8.462, JCR Q1*) in the field of energy systems and their optimisation.

- **Online swarm intelligent programming for solving nonlinear model-based predictive control in real-time.** This research provides a new real-time nonlinear model-based predictive control method based on online swarm intelligent programming. The CAPSO algorithm is used in a real-time optimisation scenario for the first time, which shows 20 times faster than the genetic algorithm (GA). The computing effort of the OSIP is investigated for the first time to provide a guide for the design of model-based control for energy management. The journal ***IEEE transactions on Industrial Informatics*** (*IF=7.377, the top one journal in industrial engineering and a journal of top four in automation and control systems*) has published this research (Zhou *et al.*, 2018).

- **Multi-step reinforcement learning for model-free predictive energy management control.** A new energy management method is developed with the capability of both the ‘model-free’ and ‘predictive’ methods, based on a

layered control framework. Three strategies for multi-step reinforcement learning are proposed and investigated for the first time. The learning performances with different learning strategies and different amounts of prediction steps are researched. This research also tested the computing efforts of different learning strategy. The new energy management method has been filed with a ***UK patent*** (GB1810775.77). One research paper reporting this research has been published with ***Applied Energy*** (Zhou *et al.*, 2019) and another one in the relevant area is being prepared for submission to IEEE transactions.

The outcomes of this research, including 1) real-time models of the electrified off-highway vehicle system and subsystems; 2) an intelligent design optimisation software packages; 3) a model-based predictive energy management system, and 4) a model-free predictive energy management system, have provided impact on the following beneficiaries:

- **Government and policymakers** – The feedback from our recent contact with the EU Joint Research Centre (who make proposals to EU Vehicle Legislation and Testing Procedures) indicates that "*this research is very in line with the scope and objectives of our ongoing research concerning the impact of connected and intelligent vehicles*". The proposed methodology has the capability to determine the potential of future vehicle products in terms of energy

efficiency and emissions, which will assist the government and policymakers in the development of policies and regulations.

- **Industries** – Allan Cairns, CEO of HYPERDRIVE Innovation (our partner for the electrified vehicle development) said: “The project has been a fantastic collaboration bringing together innovation and technology from across the UK”. The senior programme manager of the automotive sector at Innovate UK also agreed the collaboration between the partners had delivered an “exciting world-leading prototype vehicle” (Rachel Cooper, 2017). These are examples of how the proposed methodology can assist industrial partners to develop and optimise their vehicle products.

- **Research communities** – This research has provided a demonstration of how interdisciplinary research can help improve vehicle performance and provided ideas that can inspire other researchers. A review paper in ‘Renewable and Sustainable Energy Reviews’ indicates the CAPSO algorithms developed by the author is a state-of-the-art technology for component sizing and energy management of electrified vehicles (Huang *et al.*, 2018). Another review paper in ‘Automotive Innovation’ cited the author’s work and indicates future engine and vehicle control development will 1) involve more AI technologies and 2) progress from model-based to model-free development (Shuai *et al.*, 2018).

- **Students** – The research outcomes have assisted a few PhD students at the University of Birmingham in their research. The model developed in Chapter 3 has been used for vehicle system development (Cash *et al.*, 2019) and component sizing (Cash *et al.*, 2018a). The code of the CAPSO has been used for air/fuel ratio control optimisation (Z. Li *et al.*, 2019), transient calibration of a diesel engine (Yunfan Zhang *et al.*, 2018) and driver-oriented energy management control (J. Li *et al.*, 2019). Several projects for undergraduate and master's students have been set up in relation to this research.

7.3 Future Work

This research opens a gate to enter the world of artificial intelligence and the Internet of the Things. A good demonstration prototype has been developed in the area of electrified off-highway vehicles. However, the optimisation and control of passenger cars and fleets on the highway and in urban driving are more complex and challenging. Future efforts are expected to extend this research in the following aspects:

- To consider the impact of different drivers which may result in different power demand inputs to the energy management system.

- The integration of model-based predictive control and model-free predictive

control to ensure functional safety as well as to improve the learning speed in different driving scenarios.

- The virtual generation of driving scenarios for ‘back play’ of reinforcement learning with limited training data. Advanced reinforcement learning and deep learning technologies, including generative adversarial network, deep Q network, and neuro-evolution are expected to be investigated for the next stage.

- To improve the prediction performance with data from the V2X and near-field sensors. The impacts of V2X and fleet management are expected to be considered in future research.

REFERENCES

Advanced Propulsion Centre (2017) *Passenger Car Roadmap 2017*. Available at: http://www.apcuk.co.uk/product-roadmaps-2/.

Alam, A. *et al.* (2015) 'Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency', *IEEE Control Systems*, 35(6), pp. 34–56. doi: 10.1109/MCS.2015.2471046.

Bellman, R. E. (2010) *Dynamic Programming*. 2010th edn. Princeton, New Jersey: Princeton University Press.

Borhan, H. *et al.* (2012) 'MPC-based energy management of a power-split hybrid electric vehicle', *IEEE Transactions on Control Systems Technology*, 20(3), pp. 593–603. doi: 10.1109/TCST.2011.2134852.

Bouktif, S. *et al.* (2018) 'Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches', *Energies*, 11(7). doi: 10.3390/en11071636.

Cash, S. *et al.* (2018a) 'A New Traction Motor Sizing Strategy for a HEV/EV based on an Overcurrent-tolerant Prediction Model', *IET Intelligent Transport Systems*. Institution of Engineering and Technology. doi: 10.1049/iet-its.2018.5016.

Cash, S. *et al.* (2018b) ‘A New Traction Motor Sizing Strategy for a HEV/EV based on an Overcurrent-tolerant Prediction Model’, *IET Intelligent Transport Systems*, (January 2019). doi: 10.1049/iet-its.2018.5016.

Cash, S. *et al.* (2019) ‘Development of a Series Hybrid Electric Aircraft Pushback Vehicle: A Case Study’, *Engineering*, 11(01), pp. 33–47. doi: 10.4236/eng.2019.111004.

Chan, C. C. (2007) ‘The state of the art of electric, hybrid, and fuel cell vehicles’, *Proceedings of the IEEE*, 95(4), pp. 704–718. doi: 10.1109/JPROC.2007.892489.

Chau, K. . and Wong, Y. . (2002) ‘Overview of power management in hybrid electric vehicles’, *Energy Conversion and Management*. Pergamon, 43(15), pp. 1953–1968. doi: 10.1016/S0196-8904(01)00148-0.

Chau, K. T. and Wong, Y. S. (2002) ‘Overview of power management in hybrid electric vehicles’, *Energy Conversion and Management*. Pergamon, 43(15), pp. 1953–1968. doi: 10.1016/S0196-8904(01)00148-0.

Chen, M. and Rincón-Mora, G. A. (2006) ‘Accurate electrical battery model capable of predicting runtime and I-V performance’, *IEEE Transactions on Energy Conversion*, 21(2), pp. 504–511. doi: 10.1109/TEC.2006.874229.

Chen, Q., Lin, T. and Ren, H. (2018) ‘Parameters optimization and control strategy

of power train systems in hybrid hydraulic excavators', *Mechatronics*, 56, pp. 16–25. doi: 10.1016/j.mechatronics.2018.10.003.

Cheng, R. *et al.* (2017) 'Evolutionary Many-Objective Optimization of Hybrid Electric Vehicle Control : From General Optimization to Preference Articulation', *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(2), pp. 97–111.

Chomchai, C., Sonjaipanich, S. and Cheewaisrakul, S. (2011) 'Powertrain System Optimisation for a Heavy-duty Hybrid Electric Bus', *International Journal of Automotive Technology*, 12(1), pp. 131–139. doi: 10.1007/s12239.

Chopard, B. and Tomassini, M. (2018) 'Particle swarm optimization', in *Natural Computing Series*. Elsevier, pp. 97–102. doi: 10.1007/978-3-319-93073-2_6.

Ciornei, S. *et al.* (2018) 'Real-Time Simulation of a Complete Electric Vehicle Based on NI VeriStand Integration Platform', in *Proceedings of 10th International Conference and Exposition on Electrical and Power Engineering*, pp. 107–112.

Continental Automotive GmbH (2017) *Worldwide Emission Standards and Related Regulations*.

Ebbesen, S., Dönitz, C. and Guzzella, L. (2012) 'Particle swarm optimisation for hybrid electric drive-train sizing', *International Journal of Vehicle Design*, 58(2/3/4), p. 181. doi: 10.1504/IJVD.2012.047382.

Enang, W. and Bannister, C. (2017a) ‘Modelling and control of hybrid electric vehicles (A comprehensive review)’, *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, 74(January), pp. 1210–1239. doi: 10.1016/j.rser.2017.01.075.

Enang, W. and Bannister, C. (2017b) ‘Modelling and control of hybrid electric vehicles (A comprehensive review)’, *Renewable and Sustainable Energy Reviews*, pp. 1210–1239. doi: 10.1016/j.rser.2017.01.075.

European Commission (2014) *Reducing CO2 emissions from passenger cars | Climate Action*.

European Commission (2017) *Proposal for post-2020 CO2 targets for cars and vans | Climate Action*.

Fang, L. *et al.* (2011) ‘Simultaneous optimization for hybrid electric vehicle parameters based on multi-objective genetic algorithms’, *Energies*, 4(3), pp. 532–544. doi: 10.3390/en4030532.

Feldkamp, L., Nasr, M. A. and Kolmanovsky, I. V. (2009) ‘Recurrent neural network training for energy management of a mild hybrid electric vehicle with an ultra-capacitor’, *2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, CIVVS 2009 - Proceedings*. IEEE, pp. 29–36. doi: 10.1109/CIVVS.2009.4938720.

Fister, I. *et al.* (2015a) ‘A review of chaos-based firefly algorithms: Perspectives and research challenges’, *Applied Mathematics and Computation*. Elsevier, 252, pp. 155–165. doi: 10.1016/J.AMC.2014.12.006.

Fister, I. *et al.* (2015b) ‘A review of chaos-based firefly algorithms: Perspectives and research challenges’, *Applied Mathematics and Computation*. Elsevier, 252, pp. 155–165. doi: 10.1016/J.AMC.2014.12.006.

Gandomi, A. H. *et al.* (2013) ‘Chaos-enhanced accelerated particle swarm optimization’, *Communications in Nonlinear Science and Numerical Simulation*. Elsevier, 18(2), pp. 327–340. doi: 10.1016/J.CNSNS.2012.07.017.

Gao, D. W., Mi, C. and Emadi, A. (2007) ‘Modeling and Simulation of Electric and Hybrid Vehicles’, *Proceedings of the IEEE*, 95(4), pp. 729–745.

Gao, W. and Mi, C. (2007) ‘Hybrid vehicle design using global optimisation algorithms’, *Int. J. Electric and Hybrid Vehicles*, 1(1), p. 57.

Gauchia, L. and Sanz, J. (2010) ‘A per-unit hardware-in-the-loop simulation of a fuel cell/battery hybrid energy system’, *IEEE Transactions on Industrial Electronics*. IEEE, 57(4), pp. 1186–1194. doi: 10.1109/TIE.2009.2036641.

Genta, G. (1997) *Motor Vehicle Dynamics: Modeling and Simulation*.

Global EV Outlook 2018 (2018). doi: 10.1787/9789264302365-en.

Gregor, E. (2019) ‘EU Legislation in Progress CO 2 emission standards for heavy-duty vehicles’, (December 2018).

Guzzella, L. and Sciarretta, A. (2007) ‘Vehicle Energy and Fuel Consumption-Basic Concepts’, in *Vehicle propulsion systems: introduction to modeling and optimization*, pp. 13–46.

Hadj-Said, S. *et al.* (2016) ‘Convex Optimization for Energy Management of Parallel Hybrid Electric Vehicles’, *IFAC-PapersOnLine*. Elsevier, 49(11), pp. 271–276. doi: 10.1016/J.IFACOL.2016.08.041.

Di He *et al.* (2001) ‘Chaotic characteristics of a one-dimensional iterative map with infinite collapses’, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(7), pp. 900–906. doi: 10.1109/81.933333.

Hu, J., Ni, J. and Peng, Z. (2018) ‘Transmission Architecture and Topology Design of EVs and HEVs’, in *Modeling, Dynamics, and Control of Electrified Vehicles*. Elsevier Inc., pp. 121–158. doi: 10.1016/B978-0-12-812786-5.00004-5.

Hu, X. *et al.* (2013) ‘Energy efficiency analysis of a series plug-in hybrid electric bus with different energy management strategies and battery sizes’, *Applied Energy*, 111, pp. 1001–1009. doi: 10.1016/j.apenergy.2013.06.056.

Hu, X. *et al.* (2014) ‘Comparison of three electrochemical energy buffers applied to a hybrid bus powertrain with simultaneous optimal sizing and energy management’, *IEEE Transactions on Intelligent Transportation Systems*, 15(3), pp. 1193–1205. doi: 10.1109/TITS.2013.2294675.

Hu, X., Wang, H. and Tang, X. (2017) ‘Cyber-Physical Control for Energy-Saving Vehicle Following with Connectivity’, *IEEE Transactions on Industrial Electronics*, 0046(c), pp. 1–1. doi: 10.1109/TIE.2017.2703673.

Hu, Y. *et al.* (2018) ‘Energy Management Strategy for a Hybrid Electric Vehicle Based on Deep Reinforcement Learning’, *Applied Sciences*, 8(2), p. 187. doi: 10.3390/app8020187.

Huang, Y., Wang, H., *et al.* (2017) ‘Model predictive control power management strategies for HEVs: A review’, *Journal of Power Sources*, 341, pp. 91–106. doi: 10.1016/j.jpowsour.2016.11.106.

Huang, Y., Khajepour, A., *et al.* (2017) ‘Modelling and optimal energy-saving control of automotive air-conditioning and refrigeration systems’, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. SAGE PublicationsSage UK: London, England, 231(3), pp. 291–309. doi: 10.1177/0954407016636978.

Huang, Y. *et al.* (2018) ‘A review of power management strategies and component sizing methods for hybrid vehicles’, *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, 96(April 2017), pp. 132–144. doi: 10.1016/j.rser.2018.07.020.

Huang, Y., Khajepour, A. and Wang, H. (2016) ‘A predictive power management controller for service vehicle anti-idling systems without a priori information’, *Applied Energy*, 182, pp. 548–557. doi: 10.1016/j.apenergy.2016.08.143.

Hubbard, G. A. and Youcef-Toumi, K. (1997) ‘Modeling and Simulation of a Hybrid-Electric Vehicle Drivetrain Massachusetts Institute of Technology’, in *Proceedings of American Control Conference*. Albuquerque, New Mexico, pp. 636–640.

International Energy Agency (2017) *CO2 Emissions from Fuel Combustion 2017 - Highlights*. doi: 10.1787/co2_fuel-2017-en.

Isermann, R. (2014) ‘Control of Diesel Engines’, in *Engine Modeling and Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 497–625. doi: 10.1007/978-3-642-39934-3_8.

JCB Generator Technical Specifications (2015) *JCB Power Products*. Available at: http://www.midas-uk.com/PDF/JCB/QS/G116QS_EN.pdf.

Jia, D. *et al.* (2016) ‘A survey on platoon-based vehicular cyber-physical systems’, *IEEE Communications Surveys and Tutorials*, 18(1), pp. 263–284. doi:

10.1109/COMST.2015.2410831.

Kim, Y. *et al.* (2015) ‘Reducing soot emissions in a diesel series hybrid electric vehicle using a power rate constraint map’, *IEEE Transactions on Vehicular Technology*. IEEE, 64(1), pp. 2–12. doi: 10.1109/TVT.2014.2321346.

Komatsu Hybrid Technology (2019) *KOMATSU INNOVATION*. doi: 10.1007/978-1-4614-0827-7_1.

Koprubasi, K. (2008) *Modeling and Control of a Hybrid-Electric Vehicle for Drivability and Fuel Economy Improvements*, *Thesis*. The Ohio State University.

Li, J. *et al.* (2019) ‘Dual-loop Online Intelligent Programming for Driver-oriented Predict Energy Management of Plug-in Hybrid Electric Vehicles’, *Applied Energy*, 253(2019).

Li, S. G. *et al.* (2011) ‘Energy and Battery Management of a Plug-In Series Hybrid Electric Vehicle Using Fuzzy Logic’, *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 60(8), pp. 3571–3585. doi: 10.1109/tvt.2011.2165571.

Li, Y., Deng, S. and Xiao, D. (2011) ‘A novel Hash algorithm construction based on chaotic neural network’, *Neural Computing and Applications*. Springer-Verlag, 20(1), pp. 133–141. doi: 10.1007/s00521-010-0432-2.

Li, Z. *et al.* (2019) ‘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*, p. 095440701986207. doi: 10.1177/0954407019862079.

Lithium Ion Battery-Cylindrical, Type: UR-18650 (2017) *Panasonic Automotive & Industrial Systems Europe GmbH*. Available at: https://eu.industrial.panasonic.com/products/batteries-energy-products/secondary-batteries-rechargeable-batteries/lithium-ion-batteries/series/cylindrical-type/ACI4002/model/UR-18650E.

Liu, J., Peng, H. and Filipi, Z. (2005) ‘Modeling and Analysis of the Toyota Hybrid System’, in *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Monterey, California, USA: IEEE/ASME, pp. 24–28.

Liu, T. *et al.* (2015a) ‘Reinforcement learning-based energy management strategy for a hybrid electric tracked vehicle’, *Energies*, 8(7), pp. 7243–7260. doi: 10.3390/en8077243.

Liu, T. *et al.* (2015b) ‘Reinforcement Learning of Adaptive Energy Management With Transition Probability for a Hybrid Electric Tracked Vehicle’, *IEEE Transactions on*

Industrial Electronics, 62(12), pp. 7837–7846.

Liu, T. *et al.* (2017) ‘Reinforcement Learning Optimized Look-Ahead Energy Management of a Parallel Hybrid Electric Vehicle’, *IEEE/ASME Transactions on Mechatronics*, 4435(c), pp. 1–1. doi: 10.1109/TMECH.2017.2707338.

Liu, Y. *et al.* (2018) ‘Rule-corrected energy management strategy for hybrid electric vehicles based on operation-mode prediction’, *Journal of Cleaner Production*. Elsevier, 188, pp. 796–806. doi: 10.1016/J.JCLEPRO.2018.04.024.

Luján, J. M. *et al.* (2016) ‘Cost of ownership-efficient hybrid electric vehicle powertrain sizing for multi-scenario driving cycles’, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 230(3), pp. 382–394. doi: 10.1177/0954407015586333.

Ma, H. *et al.* (2014) ‘Model-based Multi-objective Evolutionary Algorithm Optimization for HCCI Engines’, *IEEE Transactions on Vehicular Technology*, 9545(c), pp. 1–1. doi: 10.1109/TVT.2014.2362954.

Ma, H. *et al.* (2017) ‘Computational Intelligence Non-model-based Calibration Approach for Internal Combustion Engines’, *Journal of Dynamic Systems, Measurement, and Control*, 140(April), pp. 1–9. doi: 10.1115/1.4037835.

Ma, H. *et al.* (2018) ‘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*, pp. 1–12. doi: 10.1177/0954407018776743.

Martinez, C. M. *et al.* (2017) 'Energy Management in Plug-in Hybrid Electric Vehicles: Recent Progress and a Connected Vehicles Perspective', *IEEE Transactions on Vehicular Technology*, 66(6), pp. 4534–4549. doi: 10.1109/TVT.2016.2582721.

Martínez, C. M. *et al.* (2019) 'Conclusions, discussion and future direction for research', *Ihorizon-Enabled Energy Management for Electrified Vehicles*. Butterworth-Heinemann, pp. 335–374. doi: 10.1016/B978-0-12-815010-8.00008-9.

Martínez, C. M. and Cao, D. (2019a) *iHorizon driver energy management for PHEV real-time control*, *Ihorizon-Enabled Energy Management for Electrified Vehicles*. doi: 10.1016/B978-0-12-815010-8.00006-5.

Martínez, C. M. and Cao, D. (2019b) 'The driver in the loop', in *ihorizon-Enabled Energy Management for Electrified Vehicles*, pp. 77–131. doi: 10.1016/B978-0-12-815010-8.00003-X.

Martínez, C. M. and Cao, D. (2019c) 'The driver in the loop', in *ihorizon-Enabled Energy Management for Electrified Vehicles*, pp. 77–131. doi: 10.1016/B978-0-12-815010-8.00003-X.

Miller, J. M. (2006) ‘Hybrid Electric Vehicle Propulsion System Architectures of the e-CVT Type’, *IEEE Transactions on Power Electronics*, 21(3), pp. 756–767.

Millo, F., Rolando, L. and Andreata, M. (2011) ‘Numerical Simulation for Vehicle Powertrain Development’, in *Numerical Analysis – Theory and Application*, p. 64. doi: 10.5772/32009.

Mohebbi, M., Charkhgard, M. and Farrokhi, M. (2005) ‘Optimal Neuro-Fuzzy Control of Parallel Hybrid Electric Vehicles’, in *2005 IEEE Vehicle Power and Propulsion Conference*. Chicago: IEEE, pp. 252–256. doi: 10.1109/VPPC.2005.1554566.

Moura, S. J. *et al.* (2011) ‘A Stochastic Optimal Control Approach for Power Management in Plug-In Hybrid Electric Vehicles’, *IEEE Transactions on Control Systems Technology*, 19(3), pp. 545–555. doi: 10.1109/TCST.2010.2043736.

Mura, R., Utkin, V. and Onori, S. (2015) ‘Energy management design in hybrid electric vehicles: A novel optimality and stability framework’, *IEEE Transactions on Control Systems Technology*. IEEE, 23(4), pp. 1307–1322. doi: 10.1109/TCST.2014.2363617.

Murgovski, N. *et al.* (2012a) ‘Component sizing of a plug-in hybrid electric powertrain via convex optimization’, *Mechatronics*. Pergamon, 22(1), pp. 106–120. doi: 10.1016/J.MECHATRONICS.2011.12.001.

Murgovski, N. *et al.* (2012b) ‘Component Sizing of a Plug-in Hybrid Electric Powertrain via Convex Optimization’, *Mechatronics*, 22(1), pp. 106–120. doi: 10.1016/j.mechatronics.2011.12.001.

Murphey, Y. L. *et al.* (2012) ‘Intelligent Hybrid Vehicle Power Control - Part I: Machine Learning of Optimal Vehicle Power’, *IEEE Transactions on Vehicular Technology*, 61(8), pp. 3519–3530. doi: 10.1109/TVT.2012.2206064.

Nuijten, E. ., Koot, M. W. . and Kessels, J. . (2003) ‘Advanced Energy Management Strategies for Vehicle Power Nets’, in *EAEC 9th International Congress: European Automotive Industry Driving Global Changes*, pp. 1–9. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1383946.

Okui, N. and Kobayashi, M. (2015) ‘A Study on Hybrid Control Method for Improvement of Fuel Economy and Exhaust-gas Emission of Hybrid Trucks’, in *JSAE/SAE 2015 International Powertrains, Fuels & Lubricants Meeting*. Kyoto. doi: 10.4271/2015-01-1780.

Orecchini, F. *et al.* (2018) ‘Energy consumption of a last generation full hybrid vehicle compared with a conventional vehicle in real drive conditions’, *Energy Procedia*. Elsevier B.V., 148, pp. 289–296. doi: 10.1016/j.egypro.2018.08.080.

Ostadi, A. and Kazerani, M. (2015) ‘A Comparative Analysis of Optimal Sizing of

Battery-Only, Ultracapacitor-Only, and Battery-Ultracapacitor Hybrid Energy Storage Systems for a City Bus', *IEEE Transactions on Vehicular Technology*, 64(10), pp. 4449–4460. doi: 10.1109/TVT.2014.2371912.

Padeanu, A. (2016) *China-only BMW X1 plug-in hybrid debuts with 37-mile electric range*, *Motor 1.com*.

Peng, J. and Williams, R. J. (1996) 'Incremental multi-step Q-learning', *Machine Learning*, 22(1996), pp. 283–290. doi: 10.1007/BF00114731.

Pothirasan, N., Rajasekaran, M. P. and Muneeswaran, V. (2018) 'Real time reactive power compensation for battery/photovoltaic hybrid power source for internet of hybrid electric vehicle system', *Cognitive Systems Research*. Elsevier, 52, pp. 473–488. doi: 10.1016/J.COGSYS.2018.07.030.

Pourabdollah, M. *et al.* (2015) 'Optimal sizing of a series PHEV: Comparison between convex optimization and particle swarm optimization', in *IFAC-PapersOnLine*, pp. 16–22. doi: 10.1016/j.ifacol.2015.10.003.

Pourabdollah, M. *et al.* (2017) 'Effect of driving, charging, and pricing scenarios on optimal component sizing of a PHEV', *Control Engineering Practice*. Pergamon, 61, pp. 217–228. doi: 10.1016/J.CONENGPRAC.2016.02.005.

Qiu, S. *et al.* (2019) 'Hierarchical energy management control strategies for

connected hybrid electric vehicles considering efficiencies feedback', *Simulation Modelling Practice and Theory*, 90(April 2018), pp. 1–15. doi: 10.1016/j.simpat.2018.10.008.

Rachel Cooper (2017) *New hybrid aircraft push-back tractor on show at Inter Airport Europe Exhibition*, *Hyperdrive Innovation Limited*. Available at: https://hyperdriveinnovation.com/new-hybrid-aircraft-push-back-tractor-on-show-at-inter-airport-europe-exhibition/.

ReportLinker (2016) *Mild Hybrid 48V Vehicles 2016-2031*, *CISION PR Newswire*.

Ripaccioli, G. *et al.* (2009) 'Hybrid Modeling , Identification , and Predictive Control : An Application to Hybrid Electric Vehicle Energy Management', in *Hybrid Systems: Computation and Control*, pp. 321–335. Available at: http://cse.lab.imtlucca.it/~bemporad/publications/papers/hscc09-hev.pdf (Accessed: 7 September 2017).

Romijn, T. C. J. *et al.* (2015) 'Receding Horizon Control for Distributed Energy Management of a Hybrid Heavy-Duty Vehicle with Auxiliaries', *IFAC-PapersOnLine*, 48(15), pp. 203–208. doi: 10.1016/j.ifacol.2015.10.029.

Salmasi, F. R. (2007) 'Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends', *IEEE TRANSACTIONS ON*

VEHICULAR TECHNOLOGY, 56(5), pp. 2393–2404. doi: 10.1109/tvt.2007.899933.

Sampathnarayanan, B. *et al.* (2010) 'Model Predictive Control As an Energy Management Strategy for Hybrid Electric Vehicles', *Proceedings of the Asme Dynamic Systems and Control Conference 2009, Pts a and B*. ASME, pp. 1161–1168. doi: 10.1115/DSCC2009-2671.

Sanguinetti, T. (2018) *How to use Real Driving Emission cycles in Simcenter Amesim?*, *Siemens*.

Schnelle, S. *et al.* (2017) 'A Driver Steering Model with Personalized Desired Path Generation', *IEEE Transactions on Systems, Management, and Cybernetics: Systems*. IEEE, 47(1), pp. 111–120. doi: 10.1109/TSMC.2016.2529582.

Shabbir, W. and Evangelou, S. A. (2019) 'Threshold-changing control strategy for series hybrid electric vehicles', *Applied Energy*. Elsevier, 235, pp. 761–775. doi: 10.1016/J.APENERGY.2018.11.003.

Shen, P. *et al.* (2017) 'Particle swarm optimization of driving torque demand decision based on fuel economy for plug-in hybrid electric vehicle', *Energy*. Pergamon, 123, pp. 89–107. doi: 10.1016/J.ENERGY.2017.01.120.

Shen, P. *et al.* (2018) 'Optimal energy management strategy for a plug-in hybrid electric commercial vehicle based on velocity prediction', *Energy*. Pergamon, 155,

pp. 838–852. doi: 10.1016/J.ENERGY.2018.05.064.

Shuai, S. *et al.* (2018) ‘Recent Progress in Automotive Gasoline Direct Injection Engine Technology’, *Automotive Innovation*. Springer Singapore, 1(2), pp. 95–113. doi: 10.1007/s42154-018-0020-1.

Soriano, F., Moreno-Eguilaz, M. and Álvarez-Flórez, J. (2015) ‘Drive Cycle Identification and Energy Demand Estimation for Refuse-Collecting Vehicles’, *IEEE Transactions on Vehicular Technology*, 64(11), pp. 4965–4973. doi: 10.1109/TVT.2014.2382591.

Sun, Chao *et al.* (2015) ‘Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles’, *IEEE Transactions on Control Systems Technology*, 23(3), pp. 1197–1204. doi: 10.1109/TCST.2014.2359176.

Sun, C *et al.* (2015) ‘Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles’, *IEEE Transactions on Control Systems Technology*, 23(3), pp. 1197–1204. doi: 10.1109/TCST.2014.2359176.

Sun, C., He, H. and Sun, F. (2015) ‘The Role of Velocity Forecasting in Adaptive-ECMS for Hybrid Electric Vehicles’, *Energy Procedia*. Elsevier B.V., 75, pp. 1907–1912. doi: 10.1016/j.egypro.2015.07.181.

Sun, C., Sun, F. and He, H. (2017) ‘Investigating adaptive-ECMS with velocity

forecast ability for hybrid electric vehicles', *Applied Energy*. Elsevier Ltd, 185, pp. 1644–1653. doi: 10.1016/j.apenergy.2016.02.026.

Sundstrom, O., Guzzella, L. and Soltic, P. (2010) 'Torque-Assist Hybrid Electric Powertrain Sizing: From Optimal Control Towards a Sizing Law', *IEEE Transactions on Control Systems Technology*, 18(4), pp. 837–849. doi: 10.1109/TCST.2009.2030173.

Sutton, R. S. and Barto, A. G. (2017) *Reinforcement Learning: An Introduction*, *Reinforcement Learning: An Introduction*. The MIT Press. doi: 10.1016/S1364-6613(99)01331-5.

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

Tie, S. F. and Tan, C. W. (2013) 'A review of energy sources and energy management system in electric vehicles', *Renewable and Sustainable Energy Reviews*. Pergamon, 20, pp. 82–102. doi: 10.1016/J.RSER.2012.11.077.

Wang, H. *et al.* (2017) 'Energy Management of Hybrid Electric Vehicles', in *Modeling, Dynamics, and Control of Electrified Vehicles*, pp. 159–206. doi: 10.1016/B978-0-12-812786-5.00005-7.

Wang, W., Zeng, X. and Wang, Q. (2003) ‘Develop Hybrid Transit Buses for Chinese Cities’, in *2003 SAE World Congress*. Detroit, Michigan: SAE International. doi: 10.4271/2003-01-0087.

Wipke, K. B., Cuddy, M. R. and Burch, S. D. (1999) ‘ADVISOR 2.1: A User-Friendly Advanced Powertrain Simulation Using a Combined Backward/Forward Approach’, *IEEE Transactions on Vehicular Technology: Special Issues on Hybrid and Electric Vehicles*.

Wu, G., Zhang, X. and Dong, Z. (2015) ‘Powertrain architectures of electrified vehicles: Review, classification and comparison’, *Journal of the Franklin Institute*, 352(2), pp. 425–448. doi: 10.1016/j.jfranklin.2014.04.018.

X-engineer (2019) *Mild Hybrid Electric Vehicle (MHEV) – architectures*, *x-Engineer: Engineering Tutorials*. Available at: https://x-engineer.org/automotive-engineering/vehicle/hybrid/mild-hybrid-electric-vehicle-mhev-architectures/.

Xia, S., Linkens, D. and Bennett, S. (1993) ‘Automatic Modelling and Analysis of Dynamic Physical Systems using Qualitative Reasoning and Bond Graphs’, *Intelligent Systems Engineering*, 2(3), pp. 201–212. doi: 10.1049/ise.1993.0017.

Xiaolan Wu *et al.* (2008) ‘Application of Particle Swarm Optimization for component sizes in parallel Hybrid Electric Vehicles’, in *2008 IEEE Congress on Evolutionary*

Computation (IEEE World Congress on Computational Intelligence). IEEE, pp. 2874–2878. doi: 10.1109/CEC.2008.4631183.

XiL Approach: An ETAS Solution (2019) *ETAS Official Website: Applications*. Available at: https://www.etas.com/en/products/applications_virtualization_xil_approach.php.

Xiong, R., Cao, J. and Yu, Q. (2018) 'Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle', *Applied Energy*, 211, pp. 538–548. doi: 10.1016/j.apenergy.2017.11.072.

Xiong, W., Zhang, Y. and Yin, C. (2009) 'Optimal energy management for a series–parallel hybrid electric bus', *Energy Conversion and Management*. Pergamon, 50(7), pp. 1730–1738. doi: 10.1016/J.ENCONMAN.2009.03.015.

Xu, F. *et al.* (2016) 'Fast nonlinear model predictive control on FPGA using particle swarm optimization', *IEEE Transactions on Industrial Electronics*, 63(1), pp. 310–321. doi: 10.1109/TIE.2015.2464171.

Xu, L. *et al.* (2015) 'Multi-objective component sizing based on optimal energy management strategy of fuel cell electric vehicles', *Applied Energy*. Elsevier, 157, pp. 664–674. doi: 10.1016/J.APENERGY.2015.02.017.

Yang, Y. *et al.* (2019) 'Fuel economy optimization of power split hybrid vehicles: A

rapid dynamic programming approach', *Energy*. Pergamon, 166, pp. 929–938. doi: 10.1016/J.ENERGY.2018.10.149.

Zamora, D. R. *et al.* (2013) 'Development of an in-series hybrid urban bus model and its correlation with on-board testing results', *World Electric Vehicle Journal*, 6(2), pp. 405–415. doi: 10.1109/EVS.2013.6914852.

Zeng, X. and Wang, J. (2017) 'A Stochastic Driver Pedal Behavior Model Incorporating Road Information', *IEEE Transactions on Human-Machine System*, pp. 1–11.

Zhang, H., Zhang, Y. and Yin, C. (2016) 'Hardware-in-the-Loop Simulation of Robust Mode Transition Control for a Series-Parallel Hybrid Electric Vehicle', *IEEE Transactions on Vehicular Technology*. IEEE, 65(3), pp. 1059–1069. doi: 10.1109/TVT.2015.2486558.

Zhang, S., Xiong, R. and Sun, F. (2015) 'Model predictive control for power management in a plug-in hybrid electric vehicle with a hybrid energy storage system', *Applied Energy*, pp. 1654–1662. doi: 10.1016/j.apenergy.2015.12.035.

Zhang, Y. *et al.* (2017) 'Optimal energy management strategy for parallel plug-in hybrid electric vehicle based on driving behavior analysis and real time traffic information prediction', *Mechatronics*. Elsevier Ltd, 46, pp. 177–192. doi:

10.1016/j.mechatronics.2017.08.008.

Zhang, Yunfan *et al.* (2018) ‘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*. doi: 10.1177/0954407018776745.

Zhang, Yi *et al.* (2018) ‘Internet-Distributed Vehicle-in-the-Loop Simulation for HEVs’, *IEEE Transactions on Vehicular Technology*. IEEE, 67(5), pp. 3729–3739. doi: 10.1109/TVT.2018.2803088.

Zhang, Y. (2018) *Optimization and control of a dual-loop EGR system in a modern diesel engine*. The University of Birmingham.

Zhou, Q. *et al.* (2017) ‘Intelligent sizing of a series hybrid electric power-train system based on Chaos-enhanced accelerated particle swarm optimization’, *Applied Energy*, 189, pp. 588–601. doi: 10.1016/j.apenergy.2016.12.074.

Zhou, Q. *et al.* (2018) ‘Cyber-Physical Energy-Saving Control for Hybrid Aircraft-Towing Tractor based on Online Swarm Intelligent Programming’, *IEEE Transactions on Industrial Informatics*, 14(9), pp. 4149–4158. doi: 10.1109/TII.2017.2781230.

Zhou, Q. *et al.* (2019) ‘Multi-step Reinforcement Learning for Model-Free Predictive Energy Management of an Electrified Off-highway Vehicle’, *Applied Energy*,

255(2019), pp. 588–601.

Zhou, Q. and Xu, H. (2018) ‘Vehicle Power Management System and Method’. PCT/GB2019/051729.

Zhou, Y., Ravey, A. and Péra, M.-C. (2019) ‘A survey on driving prediction techniques for predictive energy management of plug-in hybrid electric vehicles’, *Journal of Power Sources*. Elsevier, 412(September 2018), pp. 480–495. doi: 10.1016/j.jpowsour.2018.11.085.

Zoelch, U. and Schroeder, D. (1998) ‘Dynamic optimization method for design and rating of the components of a hybrid vehicle’, *International Journal of Vehicle Design*, 19(1). doi: 10.1504/IJVD.1998.062090.

Zou, Y. *et al.* (2016) ‘Reinforcement learning-based real-time energy management for a hybrid tracked vehicle’, *Applied Energy*. Elsevier Ltd, 171, pp. 372–382. doi: 10.1016/j.apenergy.2016.03.082.

Zou, Y. *et al.* (2017) ‘A real-time Markov chain driver model for tracked vehicles and its validation: Its adaptability via stochastic dynamic programming’, *IEEE Transactions on Vehicular Technology*, 66(5), pp. 3571–3582. doi: 10.1109/TVT.2016.2605449.