University of Birmingham

School of Engineering

Department of Mechanical Engineering

Individual Engineering Project

BEng

Surname	Vigneswaran
First Name(s)	Vinush
ID number	1690302
Supervisor's Name	Professor Hongming Xu & Dr Quan Zhou
Project Title	Intelligent energy management strategy for plug-in hybrid electric vehicle based on Fuzzy Logic and Particle Swarm Optimisation

Abstract

Environmental as well as economic issues provide a compelling impetus to develop clean, efficient, and sustainable vehicles. The promise for highly efficient, and low emission individual transportation is represented by Plug-In Hybrid-Electric Vehicles (PHEV), by shifting the demand from crude oil to electricity. This paper presents an intelligent energy management strategy for a series PHEVs by developing a Fuzzy Logic Controller (FLC). A detailed description of developing a FLC, based on vehicle dynamics and internal power distribution, is provided. Fuzzy logic provides a suitable method for realising an optimal trade-off between the efficiencies of all components of the PHEV, however, this strategy provides limited sub-optimal solutions. Therefore, the FL control actions and rules, are developed using accelerated Particle Swarm Optimisation (PSO), to achieve global optimisation. The objective of the PSO algorithm is to minimise the energy cost during vehicle operation whilst maintaining the battery's State-of-Charge (SoC), therefore increasing fuel economy and system efficiency. The FLC is optimised offline using real vehicle data (BMW i3), to train the model. In order to validate the optimisation strategy, the PSO FL will be compared with other prominent optimisation strategies; Genetic Algorithm (GA), Simulated Annealing (SA), PSO and Pattern Search (PS) hybrid. An analysis on the fuel economy and system efficiency indicates PSO and PSO-PS outperforms GA and SA. PSO FL based strategy improves the range of the vehicle for the NEDC drive cycle by 3.49 % and the system efficiency increases by 8.9 %.

Keywords:

Intelligent energy management systems; Particle swarm optimisation; Fuzzy logic controller; Plug-in hybrid electric vehicle

Nomenclature

Acronym	Definition
PHEV	Plug-in hybrid electric vehicle
ICE	Internal combustion engine
FLC	Fuzzy logic controller
FIS	Fuzzy logic inference
MF	Membership functions
PSO	Particle swarm optimisation
GA	Genetic algorithm
PS	Pattern search
SA	Simulated annealing
SoC	State-of-charge
APU	Auxiliary power unit
EMS	Energy management system
EGU	Engine generator unit

Symbols

P	power (kW)
η	efficiency
i	controller input
c	controller output
ω	angular speed
M	torque
SoC	state of charge
T	temperature
t	time
R	resistance
I	current
N	number of cells
Cap	nominal capacitance

Subscript

demand	demand
PMM	permanent magnet motor
aux	auxiliary
battp	battery pack
PC	power convertor
egu	engine-generator unit
battp	battery pack
grid	grid
f	fuel
mech	mechanical
regen	regenerative brakes
loss	loss
max	maximum
int	internal
p	parallel
s	series

1. Introduction

1.1. Project Background

The development of PHEVs plays a crucial role in the UK government's Road to Zero strategy [1], which aims for at least 50-70% of new car sales to be ultra-low emission by 2030 [1]. A PHEV consists of an electric motor and an internal combustion engine (ICE) for propulsion, with energy supplied from fuel and the national grid (plug-in). The combination of an ICE and electric motor provides a compromise between minimum fuel consumption and maximum driving range [2]. A range-extended PHEV with a series hybrid topology is discussed in this study, whereby, the electric motor is solely responsible for propulsion [2], and the battery is charged via three mediums: the national grid, small engine generator (using fuel) and regenerative braking.

The major challenge in plug-in hybrid vehicle design is the management of the energy flow from/to different powertrain components dynamically during real vehicle operation to achieve the maximum energy efficiency, while maintaining the battery's state-of-charge. In order to achieve the best fuel economy, it necessary to develop an efficient power management system, by considering power demands based on driving behaviour and the battery's State of Charge (SoC).

1.2. Project Aims

The predominant aim of this project is to investigate the implementation of the fuzzy logic controller to develop an optimal energy management strategy for PHEVs, therefore increase fuel economy. The fuzzy logic controller will be developed using accelerated Particle Swarm Optimisation (PSO), and validated against other optimisation algorithms such as; genetic algorithm, simulated annealing and pattern search (hybridised with PSO).

1.3. Project Objectives

The objective of this study can be broken down into the following:

1) Research into fuzzy logic and optimisation theory, and its current applications in control engineering using credible literature.
2) Model hybrid vehicle powertrain components and energy management system, using MATLAB/Simulink
3) Develop an offline optimisation of the FLC using accelerated PSO, PSO-PS, GA and SA.
4) Analyse the performance of the FLC in real-time simulation.
5) Discuss the results comparing with the original model (based on BMW i3), custom fuzzy-rule EMS, and the other optimisation strategies (i.e. PSO-PS, GA and SA)

1.4. Literature Review

The energy management system (EMS) is responsible for managing the energy flow from the two sources of energy to the motor and auxiliary power use [3] [4]. The optimal control can be achieved by analysing the control inputs and providing a command [5], to vary engine-generator power output. Therefore, the EMS design can be simplified to defining a functional relationship between the inputs and the outputs, as described by Wang et al. [6]. This functional relationship can be obtained using a deterministic or fuzzy rule-based strategy. The EMS design is a complex, non-linear, and uncertain dynamical problem. Therefore it would be difficult to develop a mathematical model with rigorous formulas from every real-time input. Deterministic rule-based strategies use threshold input values to decide the output command. The parameters' adjustment is reliant on human experience or "trial and error" methods [7]. Using this strategy, the output power distribution of the vehicle is categorised into various priority levels, selected based on discrete threshold values for the SoC and power required by the motor. This method is advantageous due to its simplicity and easy implementation in real-time control; however, this provides sub-optimal results.

In contrast, fuzzy rule-based strategies are favoured by automotive manufacturers, for their effectiveness and real-time application [8]. The common types of fuzzy reasoning are: Mamdani-type and Sugeno-type, both of which work with crisp data inputs [9]. The Mamdani-type fuzzy logic system is composed of three processes; fuzzification, rule evaluation and defuzzification. Fuzzification converts the 'crisp' inputs into fuzzy sets. The rule evaluation process takes qualitative descriptions of the system, as if-then statements (rules), and evaluates the input data against the rules [6]. An input value belongs to a fuzzy set to a certain degree, represented by the degree of membership [10]. The output of the controller is obtained by evaluating the degrees of membership of the if-parts of all rules, and the then-parts of all rules are averaged, weighted by these degrees of membership [10]. A fuzzy output is then converted into a crisp output. The fundamental difference between Sugeno and Mandami-type FLCs arise from the method of generating the crisp outputs [11]. Unlike Mandami, Sugeno uses weighted average to compute a crisp output [11] [9]. Therefore Mamdani consists of output membership functions. Mamdani-type is widely accepted for capturing expert knowledge [11]. It allows describing the expertise in an intuitive manner; however, it entails a substantial computational burden [11]. In this paper a Mamdani-type, fuzzy logic strategy will be investigated, and the membership functions optimised to provide the best fuel economy.

Energy management systems using rule-base and fuzzy logic control (FLC) strategies provide limited sub-optimal solutions [12]. The major issue with the design of fuzzy logic control systems, is the lack of a systematic approach to defining the membership functions and the linguistic control rules that governs the fuzzy sets' mapping strategy. Hence, the rules are independent of the membership functions, therefore a good performance of the system is not guaranteed [6]. The system performance can be improved by adjusting the membership functions. Thus, the fuzzy system can be formulated as a search problem in high dimensional space, where each point represent by a rule set and membership function [13] [14] [6]. Developing the optimal fuzzy system design is equivalent to finding the optimal location of a hyper-surface, produced by the performance of the system given some performance criteria [6] [13] [14] [15]. These characteristics make Swarm Intelligence (SI) better candidates for searching the space [6] [16]. Therefore, this paper presents a strategy for optimising the membership functions and the fuzzy rule base, using Particle Swarm Optimisation (PSO), and evaluate its performance against other optimisation strategies such as genetic algorithm and simulated annealing.

Genetic Algorithm (GA) is a probabilistic technique, developed by John Holland [17], effective in optimising fuzzy logic controllers [7][18][19]. GA starts with a population of randomly generated solutions, which represents a vector in hyperspace. The population advances towards better solutions by applying “genetic” operators [18]. The generated new population is based on natural selection, whereby a relatively good solution will produce offspring that replace the bad solutions [19], determined by a fitness function [18]. The advantage of GA for optimization is that it is highly-explorative, capable of parallel processing and does not depend on gradients. However, GA has limited accuracy of the final solution and requires a high number of function evaluation to obtain the global solution [20]. Furthermore, GA has a tendency to converge at a local optima, this occurrence can be minimised by starting with a large population, however, this will significantly reduce the speed of the optimisation [21].

Simulated Annealing (SA) is based on metal annealing process [17]. At each iteration, a new point is generated a certain distance from the current solution, based on a probability distribution [17] [22]. The major disadvantage of simulated annealing is that exploitation is relatively weak, as the acceptance is carried out by a probability condition. Liu et al. [22] presented a study investigating the application of SA in learning and tuning membership functions of fuzzy inference system, and concluded that SA algorithm is very effectual. Compared to genetic algorithms, the main strength of simulated annealing is its wide applicability [23], and low computational effort required.

Particle Swarm Optimisation (PSO), developed by Kennedy and Eberhart in 1995 [17], is a metaheuristic algorithm proposed for global, multi-objective optimisation of non-linear problems [17]. PSO has become one of the most widely used swarm intelligence-based algorithms in engineering [17]. In the PSO algorithm, the system is initialised with a population of random solutions, called particles, with an associated random velocity. The movement of the particle consists of a stochastic and deterministic component [17]. Each particle records its coordinates, ***pbest***, associated with its local best solution, ***lbest,*** based on the fitness function, as it “searches” through the problem space. The local best solution is attained within a local topological neighbourhood of particles. Similarly, the global best, ***gbest***, the overall best solution by any particle in the population, and its location, is also recorded. At each time step, the velocity of the particles are adjusted towards the coordinates of the local best and the populations’ global best solution in history, while it also has a tendency to move randomly. A simplified version of PSO, accelerated PSO, was proposed by Xin-She Yang in 2008 [24], whereby it is suggested that the diversity and quality of the solutions can be maintained by omitting the local best solution of the original PSO, and instead simulating the diversity using a random variable, r, from N(0,1) [17] [24]. The advantage of this simplification is the significant decrease in computational effort due to reduced number of calculations per iterations [24].

1.5. Contribution of this project

Although fuzzy logic controllers have many advantages in increasing the performance of energy management systems, its performance is heavily dependent on the parameters of the membership functions and rule bases. This paper provides a strategy for optimal energy management through a metaheuristic algorithm to solve, a complex multi-objective, non-linear problem. The PSO algorithm is further validated by conducting a critical performance comparison with genetic algorithm, simulated annealing and PSO-PS.

1.6. Organisation of this paper

In Section 2, the methodology is divided into three parts. Firstly, the problem formulation and the control system model of the PHEV is described. Secondly, a fuzzy energy management system for a PHEV in real-time is proposed, based on the SoC and the power demand of the motor. Thirdly, the methodology of the FLC optimisation, using PSO, is described. Section 3 presents the results, analysis and discussion. In this section, a comparison of the effect of the different optimisation algorithms (PSO, GA, SA and PSO-PS) in tuning the FLC, is presented. Therefore, the effect of the engine-generator power supply, fuel consumption, SoC and the battery power is analysed, and optimal power management strategy evaluated. Finally, a conclusion is provided.

2. Methodology

2.1. Modelling a cyber-physical power management system for Plug-in HEV

A PHEV can run as a conventional battery electric vehicle (BEV), operating in **charge depleting (CD)** mode, or it can run as a hybrid electric vehicle (HEV) and maintain the average SoC, in **charge sustaining (CS)** mode. A series PHEV configuration is shown in figure1

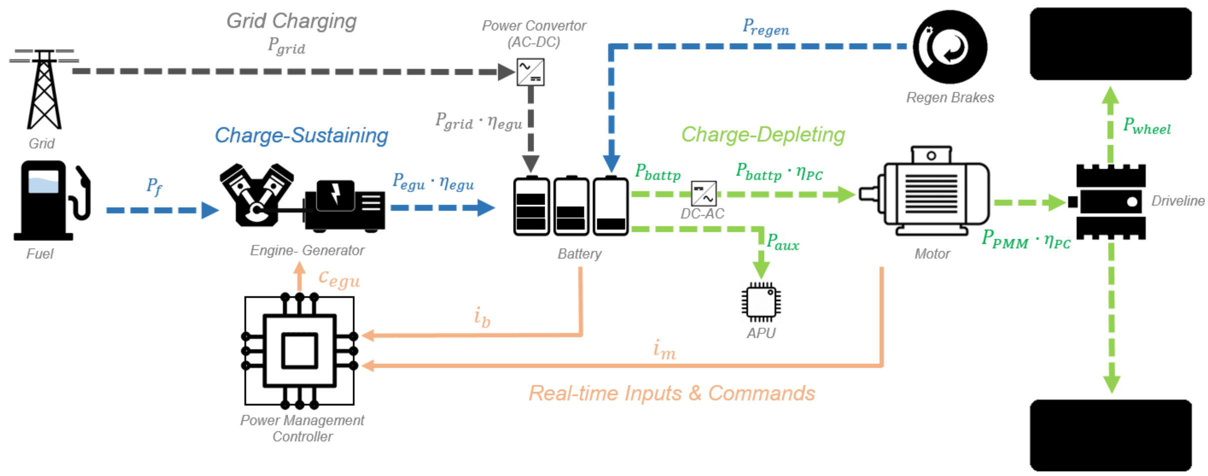

Figure 1: *Power flow & control model*

A model of the PHEV is required to analyse and specify the optimal energy management control strategy. In this paper, a quasi-static forward modelling approach is suggested, by using the conservation law. The power demand as a function of time is shown in equation 1.

$$P_{demand}(t) = \frac{P_{PMM}(t)}{\eta_{PMM}} + P_{aux}(t) = P_{battp}(t) \cdot \eta_{PC} + P_{egu}(t) \cdot \eta_{egu} \qquad \text{(Equation 1)}$$

where $P_{PMM}(t)$ is the power requirement by the permanent magnet motor, and $P_{aux}(t)$ denotes the auxiliary power requirement by the vehicle; $P_{battp}(t)$ is the power provided by the battery package, $P_{egu}(t)$ is the power provided by the engine-generator unit. η_{PC}, η_{egu} and η_{PMM} denote the efficiency of the power converter, engine-generator unit and motor, respectively. Figure 1 shows a schematic of the power flow model. Note that in the following calculations the efficiency variable will be omitted, to describe the control parameters and modelling process, however the efficiencies were implemented respectively in the computer simulations. Additionally, the energy from the grid is stored in the battery before the vehicle switches on, therefore the power from the grid can be implemented by setting the initial State-of-Charge as 100%.

The driver model, simulated on MATLAB/Simulink, controls the speed of the vehicle via an accelerator pedal input (%) to follow the drive cycle's objective velocity, and implementing a feedback of the actual velocity. The battery, motor and the vehicle body was modelled on MATLAB/Simulink using the powertrain blocksets. The vehicle's body was modelled as a single degree of freedom, rigid two-axle vehicle body with constant mass undergoing longitudinal motion. The block set for the vehicle body accounts for aerodynamic drag, road incline and force distribution between axles during acceleration. The parameters of the vehicle modelling for this case study was based on the BMW i3-REX, shown in table 1.

The motor was modelled using a mapped motor powertrain block set parameterised by a torque-speed envelope, shown in figure 2. Hence the mechanical power produced by the motor is shown in equation 2. The required electrical power is shown in equation 3.

$$P_{mech} = (\omega_{PMM} \cdot M_e) \qquad \text{(Equation 2)}$$

$$P_{PMM} = P_{mech} + P_{loss} - P_{regen} \qquad \text{(Equation 3)}$$

where, $P_{mech}(t)$ denotes the mechanical power transferred from the motor, ω_{PMM} denotes the motor shaft speed (rad/s) and T_e denotes the motor output shaft torque (N.m). In equation 3, P_{PMM} denotes the electrical power required from the battery, and P_{loss} is the power loss. P_{regen} denoted the power regenerated by the brakes.

The regenerative braking system can be modelled as reducing the power requirement of the motor, given certain conditions. In equation 1, the motor power is the net power after the reduction of the regenerative braking power. In this study the regenerative brakes only functions when the speed of the vehicle is greater than 5 mph. The regenerative brakes are calculated based on the speed of the vehicle and the brake torque applied during deceleration.

Table 1: *Basic Parameters of the case study vehicle*

Parameter	Value	Unit
Vehicle mass	1315	kg
Radius of the wheel	350	mm
Frontal Area	2.82	m^2
Motor power rating	125	kW
Engine Power (@ 5000 rpm)	28	kW
Stated Capacity of Battery	18.8	kWh
Lithium-ion Nominal Voltage	355.2	V

The Datasheet Battery block implements a lithium-ion battery which was parameterised using the data from Argonne National Library [25], based on the 2014 BMW i3-REX. The battery output voltage is determined using lookup tables for the battery's internal resistance and open circuit voltage data. The lookup tables are functions of the state-of-charge (SoC) and battery temperature, characterising the battery performance at various operating points [26], as shown in equation 4 and 5.

$$E_m = f(SoC) \qquad \text{(Equation 4)}$$

$$R_{int} = g(T, SoC) \qquad \text{(Equation 5)}$$

Where, E_m denotes the battery open-circuit voltage, SoC denotes the State-of-Charge, R_{int} is the battery internal resistance, and T is the battery temperature. The state-of-charge was calculated as shown in equation 7, using equation 6.

$$I_{batt} = \frac{I_{in}}{N_p} \qquad \text{(Equation 6)}$$

$$S = S(t_0) - \frac{1}{Cap_{batt}} \int_0^t I_{batt} \; dt \qquad \text{(Equation 7)}$$

Where, I_{battp} is the battery current per module, I_{in} is the combined current flowing from the battery network (to meet the current requirement of the motor), and N_p denotes the number of cells in parallel. In equation 7, the nominal battery capacity is Cap_{batt} and t denotes time. Equation 8 can be used to calculate the unfiltered output battery voltage, and the power transferred from the battery pack was calculated using equation 9.

$$V_{battp} = N_s \cdot (E_m - I_{batt} \cdot R_{int}) \qquad \text{(Equation 8)}$$

$$P_{battp} = V_{batt} \cdot I_{batt} \qquad \text{(Equation 9)}$$

Where, V_{battp} signifies the combined voltage of the battery and N_s denotes the number of cells in series. In equation 9, P_{battp} denotes the battery power.

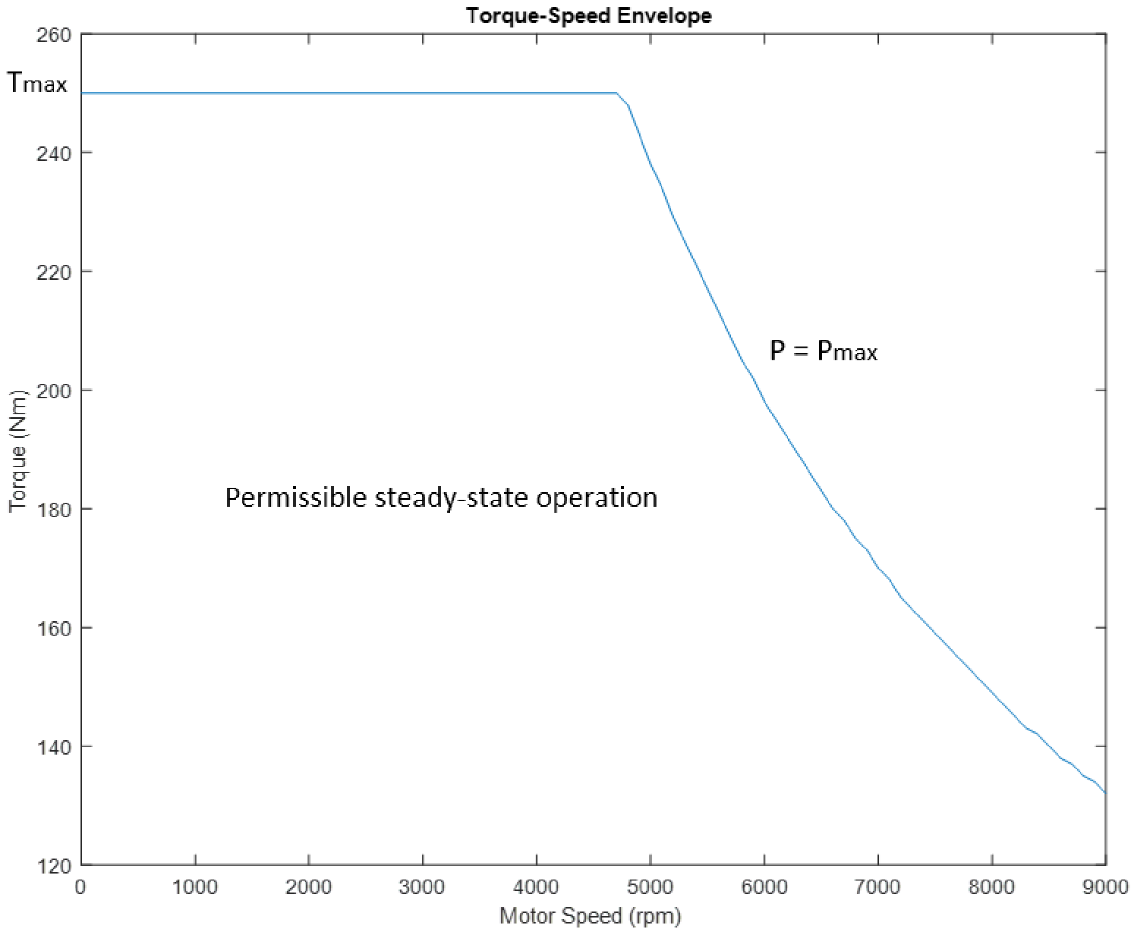

Figure 2: *(a) Torque-Speed envelope data for 2014 BMW i3 motor*

A modelled engine-generator incorporates a command signal, as shown in figure 1, sent to the engine-generator from the controller. The command signals are based on the real-time SoC of the model. The SoC threshold values were separated into 4 profiles, corresponding to 4 discrete engine power commands mimicking the BMW i3 control system. The engine-generator is modelled as a simple, dual mode switch case [27], as shown in equation 10 and 11. Where, m denotes the minimum SoC threshold to determine the operation mode.

Charge Depleting mode condition: $SoC > m$

$$\begin{cases} P_{egu} = 0 \\ P_{battp} = P_{demand} \end{cases} \qquad \text{(Equation 10)}$$

Charge Sustaining mode condition: $SoC \leq m$

$$\begin{cases} P_{egu} = \max\{0,\ P_{demand}\} & \boldsymbol{if} \quad f > 0 \\ P_{egu} = 0 & \boldsymbol{if} \quad f = 0 \\ P_{battp} = P_{demand} - P_{egu} \end{cases} \qquad \text{(Equation 11)}$$

Where f denotes the fuel level in the tank. In this study, three typical drive cycles, NEDC, US06, and WLTC are analysed using the MATLAB/Simulink model shown in figure 3. A high-fidelity model was used for the fuel flow, whereby the volumetric fuel flow was calculated based on the engine speed, fuel lower heating value and fuel mass flow. The PHEV model considers the power losses within the driveline, by calculating the mechanical power transferred via the driveshaft and the rear axle. The longitudinal behaviour of an ideal wheel was implemented to consider the braking force, therefore the power transferred, which can be calculated by summing the tractive power, power from external torque (applied by the axle to the wheel) and the power from the vertical force applied to the wheel by the vehicle or suspension.

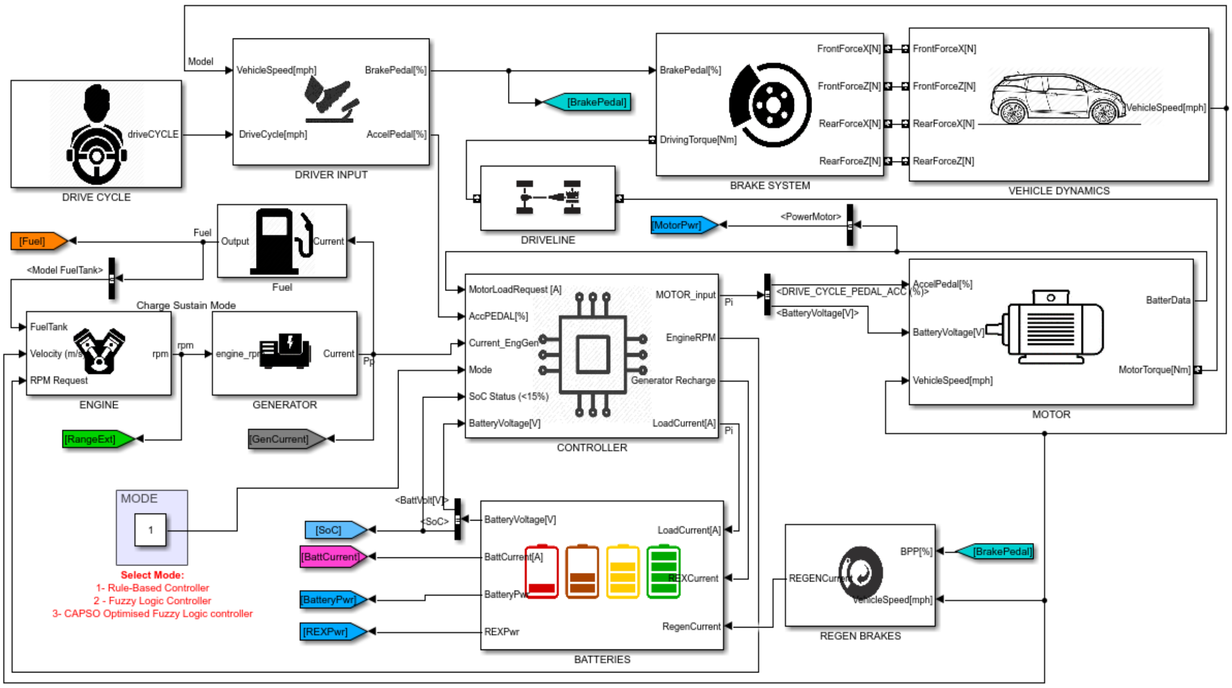

Figure 3: *Top level simulation schematic of PHEV (MATLAB/Simulink)*

2.2. Formulation of a Fuzzy Logic Controller for power management

2.2.1. Fuzzy Logic Inputs and Output

The objective of this control problem is to minimise the energy losses during operation. It is important to note in a series hybrid vehicle, the engine-generator only powers the battery, as shown in Figure 1, therefore the vehicle power requirement does not directly correlate with the load on the engine-generator. The fuzzy logic requires to make a "decision" on the engine power based on some parameters which describe the State-of-Charge of the battery pack and the requirement by the motor and the auxiliary devices. Therefore during the Charge Sustain mode, the power of the engine-generator unit can be described as the output of the FLC, as shown in equation 12.

Charge Sustaining mode (FLC switched on): $SoC \leq m_{flc}$

$$\begin{cases} P_{flc} = FLC(SoC,\ P_{demand}) & \\ P_{egu} = \max\{0,\ P_{flc}\} & \textbf{\textit{if}} \quad f > 0 \\ P_{egu} = 0 & \textbf{\textit{if}} \quad f = 0 \\ P_{battp} = P_{demand} - P_{egu} & \end{cases} \qquad \text{(Equation 12)}$$

where m_{flc} is not a constant, and is determined by the FLC rules, and is dynamically implemented based on the inputs. In this paper the fuzzy logic will be used to provide an engine power command to the engine-generator based on two inputs; the SoC of the battery and the vehicle power requirement (electric motor power demand), as shown in figure 4.

The Mamdani fuzzy inference system's inputs are crisp (non-fuzzy) values, these inputs are evaluated in parallel using fuzzy reasoning. The output of each rule is a fuzzy set derived from the output membership function and the implication method of the FIS. In this study, the output fuzzy sets are combined into a single fuzzy set using the maximum aggregation method, for each output variable [28]. The combined output fuzzy set is defuzzified using the centroid defuzzification method [28], which computes the crisp value by considering the fuzzy set as an area with uniform thickness and density, therefore the crisp single output can be obtained by calculating the centre of gravity of the fuzzy set along the horizontal axis. A summary of the FLC schematic can be seen in figure 4 and 5(a).

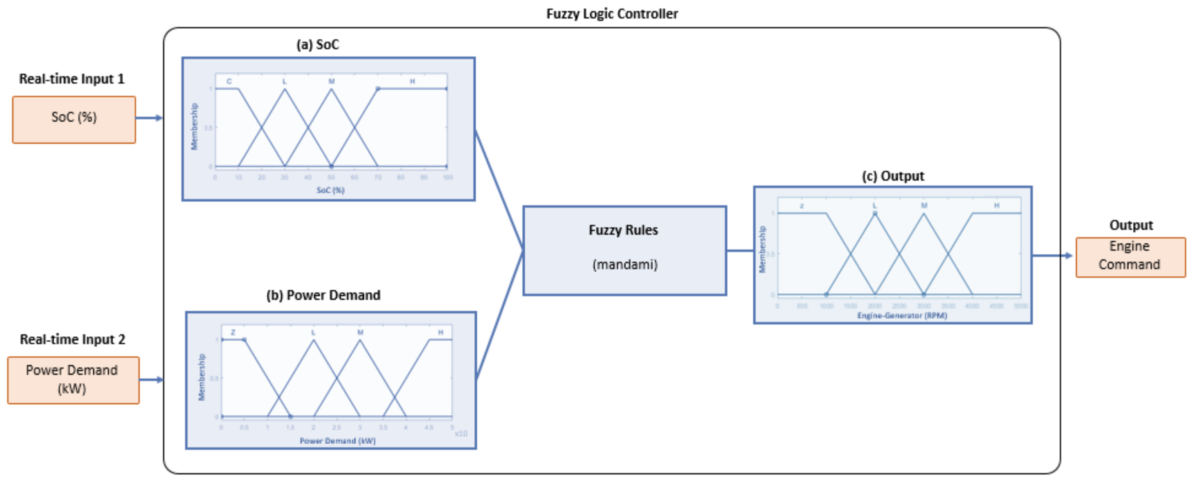

Figure 4: *Fuzzy logic membership function plot (a) SoC input (b) Power demand input (c) Engine speed output*

2.2.2. Membership Functions & Fuzzy Rules

The membership functions relate the crisp input to a fuzzy set. Trapezoidal memberships are parameterised with the four vertices of the trapezium, $[x_1\ x_2\ x_3\ x_4]$. Triangular membership functions can be categorised as a particular case of trapezoidal membership function, where the vertex $x_2 = x_3$. Barua et al. have described an interval-based theoretical explanation for the preference in the use trapezoidal membership function [29].

The input SoC consists of four balanced membership functions, figure 4(a), which can be described using the following linguistic states; High (H), Medium (M), Low (L) and Critical (C). The input electric motor power demand also consists of four membership functions (figure 4b) High (H), Medium (M), Low (L) and Minimal (Z) when the vehicle is idling or the brakes are applied. The engine power output, can be classified into the same membership functions, figure 4(c), as the motor requirement. As there are four membership functions for each input, the rules can be summarised by a 4 by 4 matrix, with 16 potential fuzzy set outputs, shown in table 2. The fuzzy rules were determined based on the current BMW i3 EMS, as well as expert experience and intuition.

Table 2: *Custom Fuzzy Rules for FLC*

State of Charge \ *Motor Power Demand*	**H**	**M**	**L**	**Z**
H	Z	Z	Z	Z
M	L	L	Z	Z
L	H	M	M	L
C	H	H	H	M

2.3. Accelerated particle swarm optimisation algorithm

2.3.1. Cost function

The cost function is essential for evaluating the fitness of each particle. For the tuning of the fuzzy logic controller, the Mean Root Squared Error (MRSE) cost function was used as shown in equation 13. The benchmark data for the inputs of MRSE, were obtained from Argonne Laboratory [25].

$$MRSE = E(k) = \frac{1}{N}\sum_{i-1}^{N}\sqrt{\left(e_1^2(i) + e_2^2(i)\right)} \qquad \text{(Equation 13)}$$

where $e_1(i)$ denotes the trajectory of the error of the ith sample for the SoC input, $e_2(i)$ denotes the trajectory error of the ith sample for the motor requirement input, N is the number of sample and k is the iteration number.

2.3.2. Mechanism for accelerated Particle Swarm Optimisation algorithm (PSO)

The accelerated PSO was implemented by modifying the 'particleswarm' function from the Global Toolbox in MATLAB. In the simplified accelerated PSO version, the particles' trajectory move towards the global best, with a tendency to move randomly. The accelerated PSO does not considers the particle's best position, only the global best. The position vector of the particle can be described as follows:

$$x_i^{t+1} = (1-\beta)x_i^t + \beta(gbest) + \alpha r \qquad \text{(Equation 14)}$$

where x_i denotes the position vector, r is a random variable drawn from a Gaussian probability distribution, α and β are the learning parameters. The diversity provided by including the particle's individual best in PSO, is substituted by the αr term to induce randomness. According to Yang et al, α needs to be consistent with the scale of the problem, and therefore it is a function of the scale of each variable (L), typically α = 0.1 L to 0.5 L. β is associated with the "attraction" parameter of the particle, that characterises the attraction towards the global best β=0.5 is suitable for this case-study [30].

2.4. Tuning Fuzzy Inference System using Optimisation Algorithms

The membership functions and the rule bases of the fuzzy inference system were tuned using accelerated particle swarm optimisation. The FIS system was restricted to 4 membership function for the input and the output, and 16 possible output membership functions. In order to tune the FIS rules using the Global Optimisation Tool on MATLAB, the input and the output membership function parameters were **initially** kept constant, and the FIS was allowed to learn the rules using training and validation (testing) data obtained from Argonne Laboratory based on the BMW i3 [25]. The data, obtained from over 12 different drive cycles (e.g. NEDC, WLTC, US06, UDDS, etc.), was partitioned into odd and even-indexed data for testing and training. Once the rules have been learnt, then the parameters of the membership functions and the rule bases be optimised using accelerated PSO. Parallel computing, as well as the limited FIS logic rule parameters, allows the optimisation algorithm to converge relatively quicker during the rule training. The training error can be minimised by increasing the number of iterations, however; this may cause an increase in validation error due to over-tuned parameters. The PSO optimisation process of the fuzzy logic is shown in figure 5, similarly this process was repeated with Genetic Algorithm, Simulated Annealing and finally PSO-PS, whereby PSO is used for rule training and then pattern search (PS), a local optimisation algorithm, for membership function tuning. The fuzzy logic system was then exported and interfaced to the EMS module in the Simulink/MATLAB environment.

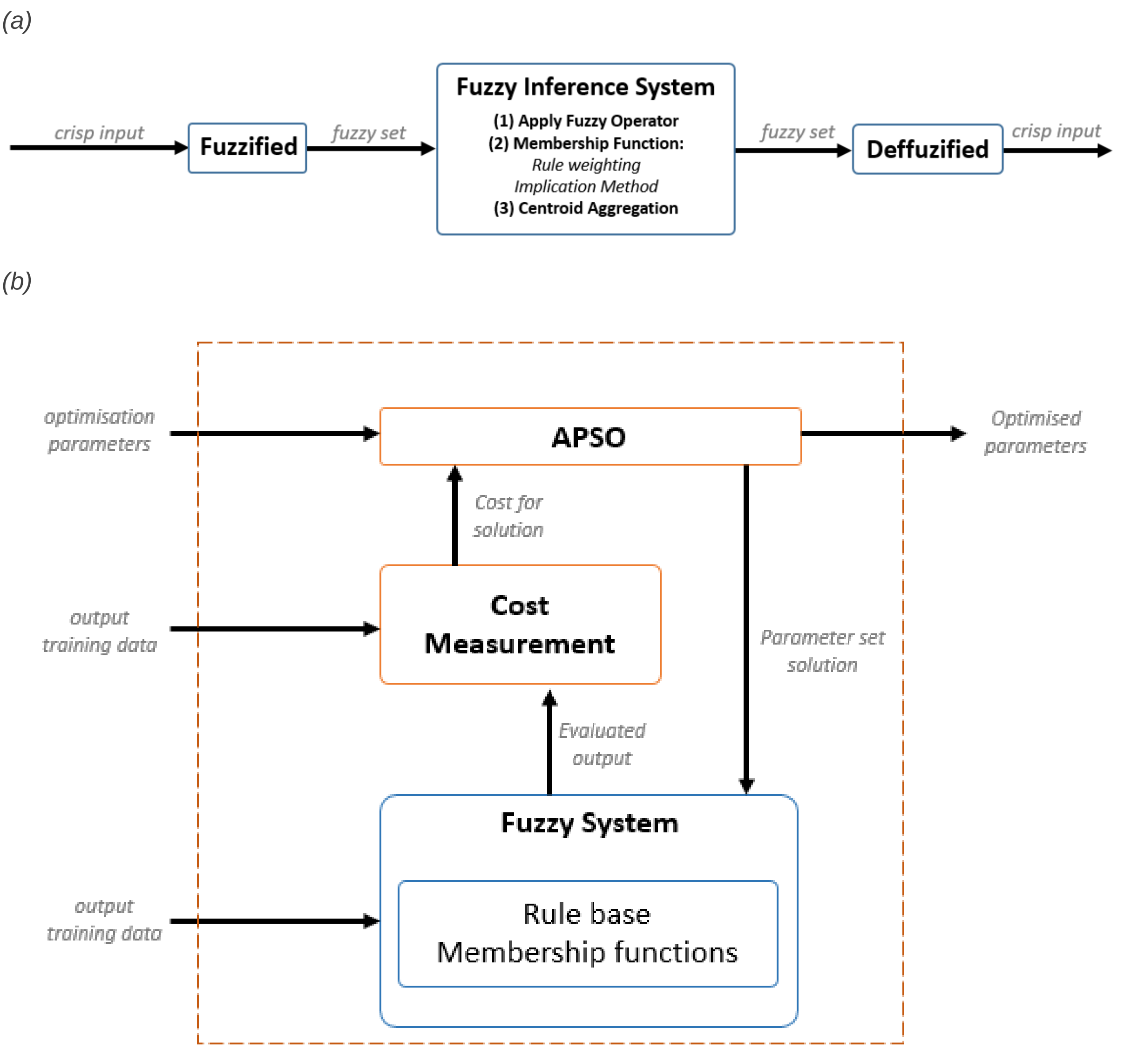

Figure 5: *(a) Fuzzy Logic Controller (FLC) Schematic (b) Optimisation of Fuzzy Inference System*

3. Results and discussion

3.1. *Performance validation of optimisation algorithm*

The results of the cost function can be compared, by analysing the RMSE of the membership functions (MF) with constant parameters and the RMSE cost function with optimised parameters, to validate the performance of the optimised fuzzy inference system, this is shown in table 1.

Table 3: *Results of the cost function for the different optimisation algorithm implemented on FLC.*

Fuzzy Logic MF Criteria	PSO	PSO-PS	GA
Number of Rule Bases	13	13	12
RMSE for strategy with non-optimised MF	2.54	2.54	2.55
RMSE for strategy with optimised MF	2.36	2.45	2.43
Reduction (%)	7%	4%	5%

The table shows the PSO has the best performance, with the highest reduction in RMSE value. Simulated Annealing utilises a probability of acceptance as the objective function therefore cannot be compared with PSO and GA. In order to limit the number of iterations PSO, PSO-PS and GA were limited to 100 iterations, and SA was limited to 200 iterations. These iteration values were obtained through trial and error. It was found that PSO and GA tend to converge before the maximum iteration limit was reached.

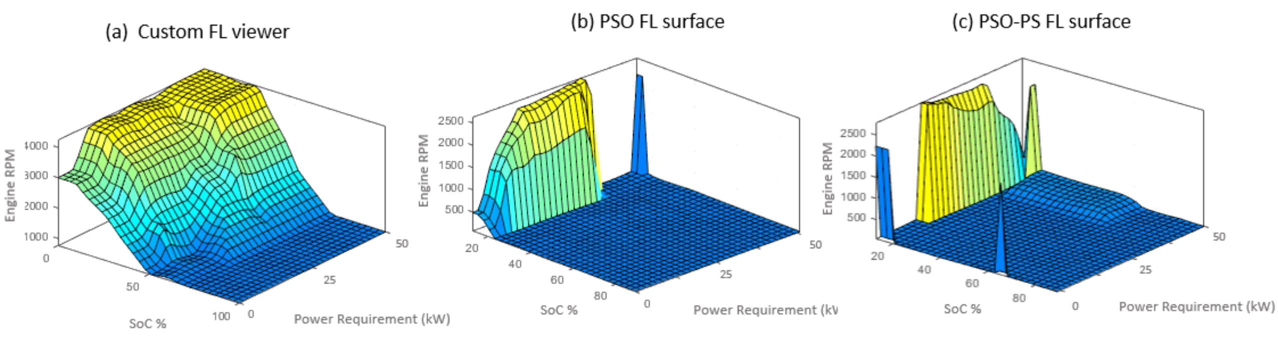

Figure 6: *Surfaces of (a) custom FIS (b) PSO FIS and the PSO-PS FIS*

Figure 6 shows a graphical relationship between the SoC, engine RPM and the output power requirement. It is evident from figure 6(a) that the custom FL's engine is switched on earlier than the PSO-FL and PSO-PS FL relationship. Additionally, the optimal FLCs show that the engine should be working at a lower RPM, when the SoC is around 20%, demonstrated by the sharp rise in the surface. Another interesting observation is that the engine RPM is relative low, when the power requirement is between 25 and 50 kW, suggesting that during this section the EGU is less efficient. Finally a narrow peak can be observed in both figure 6(b) and 6(c), when the SoC is low and the power requirement is high, this is to ensure power demand is met at low SoC, preventing the vehicle from suddenly dropping in SoC, incapable of maintain a steady SoC.

3.2. Drive Cycle

A comparison of the different energy management strategies are compared when run over three different drive cycles: NEDC, UDDS and WLTC, as shown in figure 7. The simulation model was left to run multiple cycles, until the SoC reaches 5 %. As expected, the general trend of the SoC shows a linear decrease, until the SoC threshold. Therefore the SoC is maintained within the range of 10% to 25% when the engine-generator unit is recharging the batteries (charge-sustain mode). Due to the different power requirement by the different drive cycles, in general the vehicle can complete a certain amount of drive cycles; NEDC x 22, UDDS x 25 and WLTC x 8.

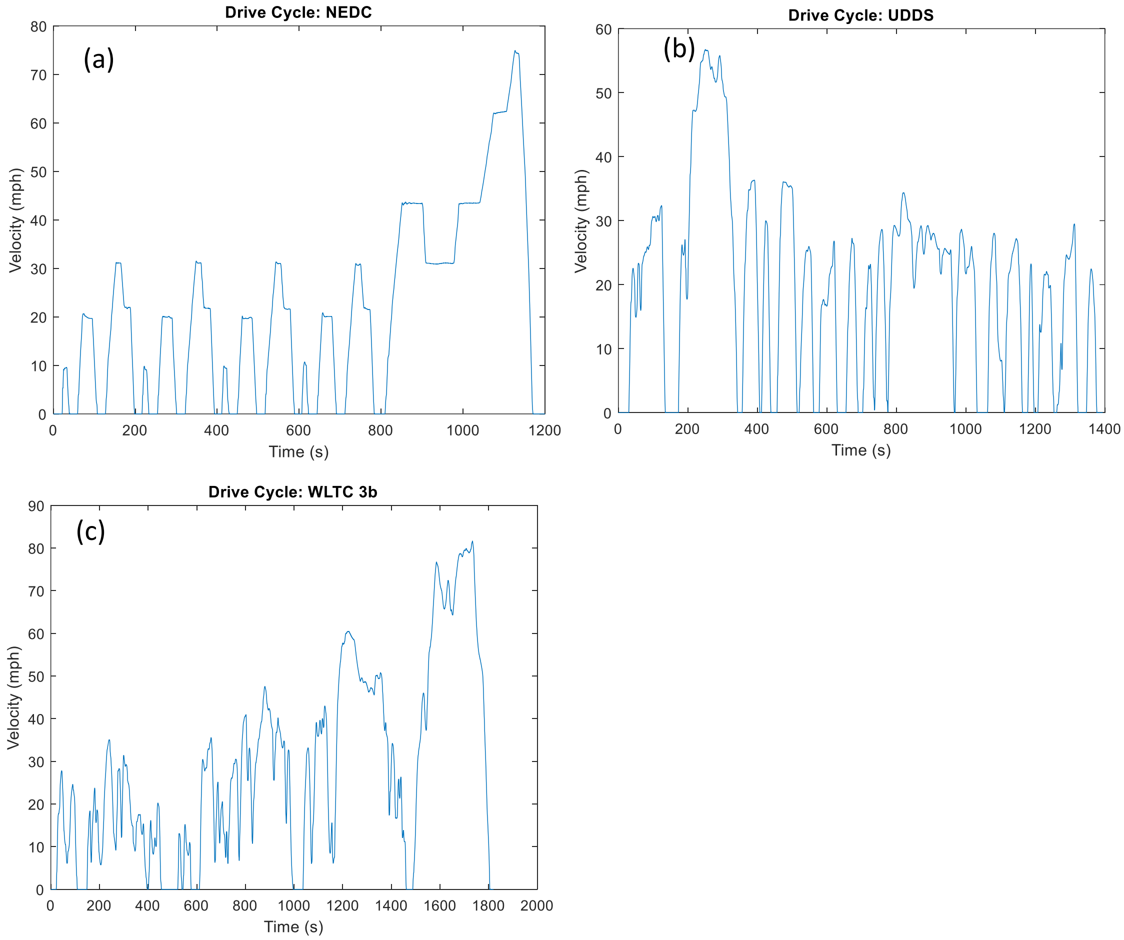

Figure 7: *Drive cycles (a) NEDC (b) UDDS (c) WLTC*

The simulation was allowed to run until the SoC reaches 5%, therefore the fuel is expected to reach 0% at the end of the simulation. The SoC and fuel consumption of the different energy management strategies were recorded for the different drive cycles as shown in figure 7. In figure 8(a), (b) and (c) it is evident that the engine-generator unit does not provide any power until the SoC reaches approximately 50%. Therefore at this point, the different energy management strategies start to deviate from one another. Whereby, custom FL refers to the fuzzy logic system with intuitive rules specified in table 2 and the membership functions shown in figure 4.

3.3. SoC and fuel consumption over multiple cycles

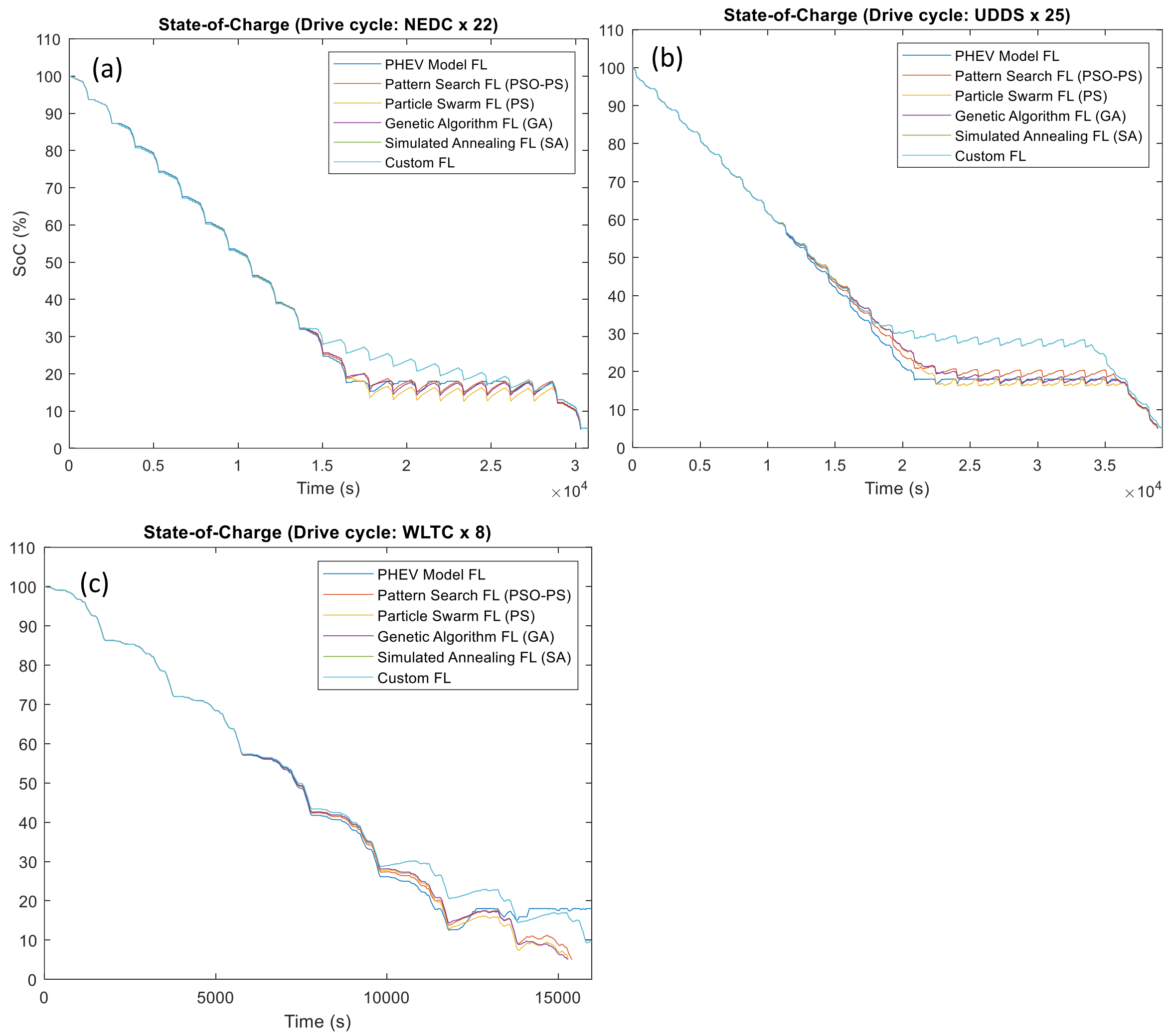

Figure 8: *(a) SoC for NEDC, (b) SoC for UDDS (c) SoC for WLTC*

In figure 8(a) and 8(b), the custom FL EMS maintains the SoC at a higher threshold compared to the other systems. In figure 8(b), an observation can be made that the charging and discharging rate around the threshold SoC varies between the different strategies. WLTC drive cycle caused the PSO, PSO-PS, GA and the SA FL EMS to end before the fuel reached 0%, as shown in figure 9, this suggests that the power demand of the motor could not be met by the engine-generator's power output, resulting in the inability to maintain the battery's SoC. This may be due to the rapid changes in the velocity of the WLTC drive cycle, and a high maximum velocity compared to UDDS and NEDC, therefore requiring higher peaks of power. Figure 9, the fuel consumption for the NEDC and UDDS cycle, show that the drive cycle is a significant factor contributing to the fuel efficiency, therefore introducing drive cycle prediction into the EMS may lead to a more efficient system.

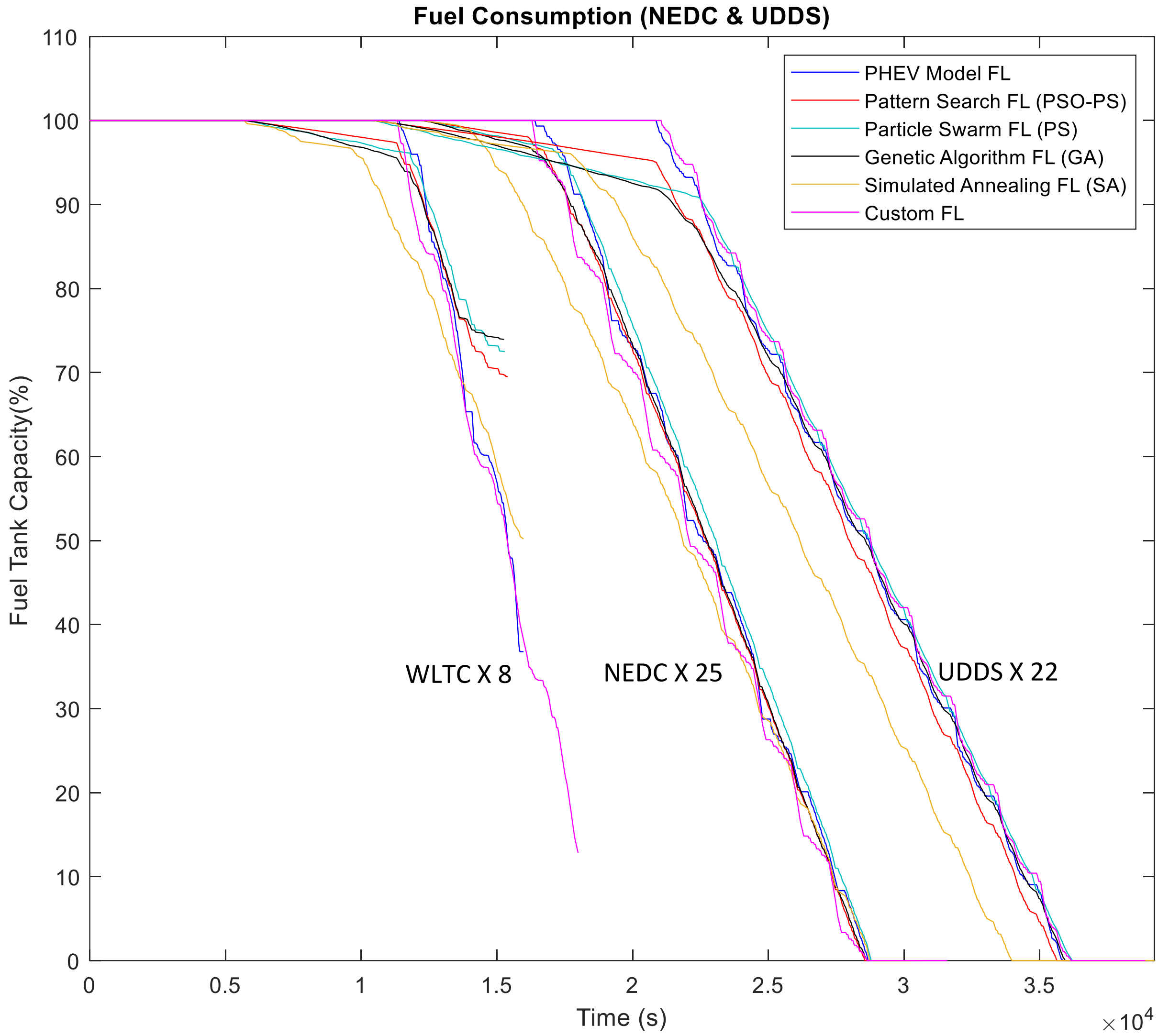

Figure 9: *Fuel consumption of multiple cycles of WLTC NEDC and UDDS*

3.4. Range and system efficiency

The distance travelled with a full fuel tank, the fuel economy and the system efficiency is summarised in table 4. The fuel economy was calculated as the litres of fuel per 100 km, therefore can be compared with the different energy management strategies. The system efficiency is the ratio of the energy output (motor) over the total energy (regenerative brakes, engine-generator, motor and battery).

Table 4: *Summary of total distance travelled, fuel economy and system efficiency for all three drive cycles*

Energy Management Strategy	Distance Travelled (km)			Fuel Economy (L/100 km)			Ave System Efficiency (%)		
	NEDC	***UDDS***	***WLTC***	***NEDC***	***UDDS***	***WLTC***	***NEDC***	***UDDS***	***WLTC***
PHEV model	241.02	297.38	172.37	21.69	26.76	15.51	60.77	61.74	50.98
Custom FL	241.24	298.01	174.80	21.71	26.82	15.73	61.94	62.48	50.77
PSO-PS FL	242.44	**300.05**	184.74	21.82	**27.00**	16.63	63.30	**66.04**	53.77
PSO FL	**249.43**	297.48	**207.63**	**22.45**	26.77	**18.69**	**66.18**	63.13	**55.92**
GA FL	241.68	297.75	184.73	21.75	26.80	16.63	61.03	62.47	54.28
SA FL	241.44	297.84	171.50	21.73	26.81	15.43	59.49	61.42	51.69

Table 5: *Change in distance and system efficiency for NEDC drive cycle only*

Energy Management Strategy	**Change in Fuel Economy (%)** *compared to PHEV model*	**Change in ave. system efficiency (%)** *compared to PHEV model*
Custom FL	**-0.09%**	1.91%
PSO-PS FL	-0.59%	4.16%
PSO FL	**-3.49%**	**8.90%**
GA FL	-0.27%	0.43%
SA FL	-0.17%	**-2.11%**

For the NEDC drive cycle, the PSO FL energy management strategy has the highest range, shown in table 4, for the same amount of fuel. Table 5 shows that the PSO-FL energy management strategy compared to the original energy management system model, will reduce the distance by 4.88%, and increase the system efficiency by 8.9%, the highest increase in system efficiency. However, in table 6, PSO FL yielded the lowest increase in range, compared to the original EMS, of -0.03%. Furthermore, it is also evident in table 4 that for the UDDS drive cycle the PSO-PS FLC, has the greatest distance travelled and fuel economy. Table 6 shows that the UDDS increases the system efficiency by 6.97%, which is significantly higher than the PSO FLC. This analysis suggests that using pattern search for local optimisation of membership functions, may outperform PSO, in certain drive conditions. Therefore, to further increase efficiency, the EMS may require to switch between different FLCs in real-time based on power requirement prediction.

Table 6: *Change in distance and system efficiency for UDDS drive cycle only*

Energy Management Strategy	Change in Fuel Economy (%) *compared to PHEV model*	Change in ave. system efficiency (%) *compared to PHEV model*
Custom FL	-0.21%	1.21%
PSO-PS FL	**-0.90%**	**6.97%**
PSO FL	**-0.03%**	2.25%
GA FL	-0.13%	1.18%
SA FL	-0.16%	**-0.51%**

GA and SA performed relatively poor, with SA achieving a lower system efficiency than the original EMS, this is expected as SA is a better candidate for local searches due to its dependence on a probability function [17]. The custom FLC's performance was reasonable, yielding a higher system efficiency than the original EMS, suggesting that the intuitive rules shown in table 2 are valid in achieving a higher fuel efficiency, as well as validating the use the of fuzzy logic compared to the modelled EMS. It is important to note that tables 4, 5 and 6 show the average system efficiency, which does not indicate the system's performance for a particular instance. An improvement of the study, would be to categorise different drive cycle local behaviours based on different levels of power consumptions and evaluate the best FL strategy for the specific cases.

3.5. Power analysis

Figure 10 shows the simulated power demand, battery and engine power when following the NEDC drive cycle. The graphs compare the original EMS with the PSO-FL EMS. The power demand, shown in figure 10(a) and 10(b), is approximately the same, with the average value deviating by 3.9%. The individual peaks represent the higher power required at higher velocity, hence the number of peaks represent the cycle number. The negative battery power indicates the battery recharging via regenerative braking and the engine-generator unit. In figure 10(c) and 10(d), the engine-generator unit (EGU) recharging can be seen clearly, with the mean battery power shifting towards the negative quadrant of the graph. There is a 6% decrease in the average battery power of the PSO FL strategy. As excepted, this change occurs when the engine generator unit switches on, as shown in figure 10(e) and 10(f). Both the original EMS and PSO-FL EMS start the EGU at the same time, however the peak power and intervals vary significantly. The PSO-FL strategy indicates a decrease in the average engine power, however the engine-generator runs for a longer time. Additionally, it is evident that the EGU works at higher peak power, but at shorter intervals, for the PSO FL strategy, compared to the varied dual threshold engine power strategy used by the original EMS model. This change in EGU power provides a higher fuel economy and higher system efficiency as depicted in table 5.

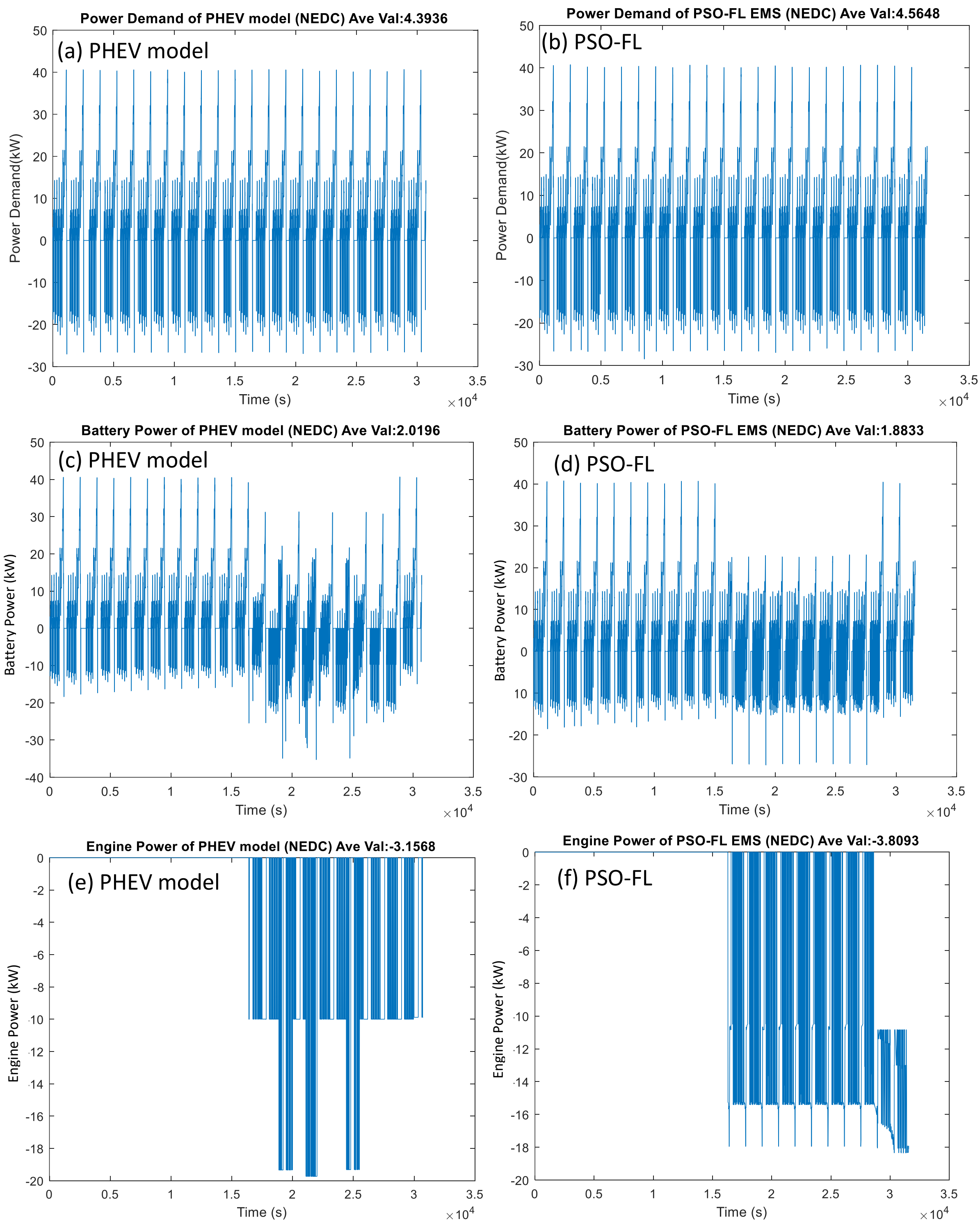

Figure 10: *(a) Original model power demand (b) PSO FL power demand (c) original model battery power (d) PSO FL battery power (e) original model engine power (d) PSO FL engine power*

3.6. Limitations

The training data for the optimisation algorithms, from Argonne Laboratory, were chosen based on 10 drive cycles, and the corresponding vehicle parameters (SoC, engine RPM, battery data, etc.), therefore the data may have been skewed towards those specific data, hence may not result in the optimal membership functions and fuzzy logic controller. Additionally no engine torque data were provided by Argonne Laboratory, which could have provided information on the EGU power efficiency and fuel economy. Additionally, the data obtained for this case study were limited to a single drive cycle, and therefore it is difficult to predict the BMW's engine, battery and motor performances over multiple drive cycles. Due to the restricted computation power, no measures were employed to systematically obtain the number of iterations for the PSO algorithm, and therefore the algorithm may be partially over-tuned to the training data.

4. Conclusion

An intelligent energy management strategy for plug-in hybrid electric vehicle has been proposed, with a focus on developing optimal power management using a fuzzy logic controller, and evaluating accelerated particle swarm optimisation for tuning the fuzzy logic threshold functions and rule bases. The following conclusions can be drawn from this study:

- Incorporating fuzzy logic strategy into the EMS improves the system efficiency by 1.21% for UDDS drive cycles, by 1.91% for NEDC drive cycles, and by 9.7% for WLTC.
- PSO FL strategy improves the range of the vehicle for the NEDC drive cycle by 3.49 % and the system efficiency by 8.9 %.
- PSO for the fuzzy rule optimisation performed better than the intuitive non-optimised fuzzy rules, as well as, outperforming the GA tuned rule base and SA tuned rule base.
- The offline optimisation using the training data can be improved in the real world by replacing the training data with samples of data obtained by "common journeys" performed by the vehicle, therefore improving the fuel efficiency over time.
- The system efficiency varies between different drive cycles, therefore the power management strategy can be improved by developing an approach for predicting the drive cycle power requirement.
- The study can be further improved by conducting an experiment with the BMW i3 to examine its performance over multiple cycles, therefore achieving a refined and more precise model.

Acknowledgements

Throughout the writing of this paper I have received a great deal of support and assistance. I would like to thank my project supervisor Professor Hongming Xu, for his support and technical advice on PHEVs. I would also like to thank Dr Quan Zhou, whose expertise was invaluable in the formulation of the research topic, as well as, his technical advice, encouragement and provision of data. In addition, I would like to thank my tutor, Dr Carl Anthony, for his valuable guidance throughout the year, and providing the resources to complete my dissertation successfully.

I extend my thanks to all my friends for their support and continued encouragement. I express my extreme gratitude to my friend Pablo, for keeping me company during the long hours in the library and the labs. Finally, I would like to thank my parents, brother and sister for always being there, providing wise counsel and keeping me sane.

References

[1] HM Government (2018). The Road to Zero. London: Dept for Transport, pp.2-7,34-42.

[2] Panday, A. and Bansal, H.O., 2014. A review of optimal energy management strategies for hybrid electric vehicle. International Journal of Vehicular Technology, 2014.

[3] Jalil, N. and Kheir, N., 1998. Energy management studies for a new generation of vehicles (Milestone No. 6, fuzzy logic for the parallel hybrid). Technical Report, Department of Electrical and Systems Engineering, Oakland University, Rochester, MI, USA.

[4] Li, S.G., Sharkh, S.M., Walsh, F.C. and Zhang, C.N., 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.

[5] Geering, H. (2007). Optimal control with engineering applications. Berlin: Springer, pp.8-9, 15 – 36, 39-46.

[6] Wang, A. and Yang, W., 2006, June. Design of energy management strategy in hybrid vehicles by evolutionary fuzzy system part I: Fuzzy logic controller development. In 2006 6th world congress on intelligent control and automation (Vol. 2, pp. 8324-8328). IEEE.

[7] Lihao, Y., Youjun, W. and Congmin, Z., 2016. Study on fuzzy energy management strategy of parallel hybrid vehicle based on quantum PSO algorithm. International Journal of Multimedia and Ubiquitous Engineering, 11(5), pp.147-158.

[8] Arcos-Aviles, D., Pascual, J., Marroyo, L., Sanchis, P. and Guinjoan, F., 2016. Fuzzy logic-based energy management system design for residential grid-connected microgrids. IEEE Transactions on Smart Grid, 9(2), pp.530-543.

[9] Dadios E, editor. Fuzzy Logic: Controls, Concepts, Theories and Applications. BoD–Books on Demand; 2012 Mar 28.

[10] Schouten, N.J., Salman, M.A. and Kheir, N.A., 2003. Energy management strategies for parallel hybrid vehicles using fuzzy logic. Control engineering practice, 11(2), pp.171-177.

[11] Shleeg AA, Ellabib IM. Comparison of mamdani and sugeno fuzzy interference systems for the breast cancer risk. International Journal of Computer, Information Science and Engineering. 2013 Sep 3;7(10):387-91.

[12] Chen, Z., Xiong, R. and Cao, J., 2016. Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions. Energy, 96, pp.197-208.

[13] Esmin, A.A.A., Aoki, A.R. and Lambert-Torres, G., 2002, October. Particle swarm optimization for fuzzy membership functions optimization. In IEEE International Conference on Systems, Man and Cybernetics (Vol. 3, pp. 6-pp). IEEE.

[14] Bingül, Z. and Karahan, O., 2011. A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control. Expert Systems with Applications, 38(1), pp.1017-1031.

[15] Arslan, A. and Kaya, M., 2001. Determination of fuzzy logic membership functions using genetic algorithms. Fuzzy sets and systems, 118(2), pp.297-306.

[16] Park, D., Kandel, A. and Langholz, G., 1994. Genetic-based new fuzzy reasoning models with application to fuzzy control. IEEE Transactions on Systems, Man, and Cybernetics, 24(1), pp.39-47.

[17] Yang, X., 2014. Nature-Inspired Optimization Algorithms. Elsevier.

[18] Herrera F, Lozano M, Verdegay JL. Tuning fuzzy logic controllers by genetic algorithms. International Journal of Approximate Reasoning. 1995 Apr 1;12(3-4):299-315.

[19] Wang A, Yang W. Design of energy management strategy in hybrid electric vehicles by evolutionary fuzzy system part II: tuning fuzzy controller by genetic algorithms. In2006 6th World Congress on Intelligent Control and Automation 2006 Jun 21 (Vol. 2, pp. 8329-8333). IEEE.

[20] Laribi MA, Mlika A, Romdhane L, Zeghloul S. A combined genetic algorithm–fuzzy logic method (GA–FL) in mechanisms synthesis. Mechanism and machine theory. 2004 Jul 1;39(7):717-35.

[21] Osman MS, Abo-Sinna MA, Mousa AA. A combined genetic algorithm-fuzzy logic controller (GA–FLC) in nonlinear programming. Applied Mathematics and computation. 2005 Nov 15;170(2):821-40.

[22] Almaraashi M, John R, Coupland S, Hopgood A. Time series forecasting using a TSK fuzzy system tuned with simulated annealing. InInternational Conference on Fuzzy Systems 2010 Jul 18 (pp. 1-6). IEEE.

[23] Liu G, Yang W. Learning and tuning of fuzzy membership functions by simulated annealing algorithm. InIEEE APCCAS 2000. 2000 IEEE Asia-Pacific Conference on Circuits and Systems. Electronic Communication Systems.(Cat. No. 00EX394) 2000 Dec 4 (pp. 367-370). IEEE.

[24] Gandomi AH, Yun GJ, Yang XS, Talatahari S. Chaos-enhanced accelerated particle swarm optimization. Communications in Nonlinear Science and Numerical Simulation. 2013 Feb 1;18(2):327-40.

[25] Anl.gov. 2020. Energy Systems D3 2014 BMW I3-REX. [online] Available at: <https://www.anl.gov/es/energy-systems-d3-2014-bmw-i3rex> [Accessed 3 October 2019].

[26] 2020. Powertrain Blockset™ User'S Guide. 1st ed. [ebook] Natick, MA: The MathWorks, Inc., pp.3-2 to 3-28. Available at: <https://www.mathworks.com/help/pdf_doc/autoblks/autoblks_ug.pdf> [Accessed 11 March 2020].

[27] Chen, Z., Xiong, R. and Cao, J., 2016. Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions. Energy, 96, pp.197-208.

[28] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975.

[29] Barua A, Mudunuri LS, Kosheleva O. Why trapezoidal and triangular membership functions work so well: Towards a theoretical explanation. Journal of Uncertain Systems. 2014;8(3).

[30] Yang XS, Deb S, Fong S. Accelerated particle swarm optimization and support vector machine for business optimization and applications. Ininternational conference on networked digital technologies 2011 Jul 11 (pp. 53-66). Springer, Berlin, Heidelberg.