Journal of Mechatronics and Robotics

Original Research Paper

Development of a Fuzzy Logic Controller for Real-time Energy Optimization of a Hybrid vehicle

***Sk. Khairul Hasan and Anoop Kumar Dhingra**

Department of Mechanical Engineering, University of Wisconsin Milwaukee, WI 53211, USA

Article history
Received: 16-08-2020
Revised: 19-10-2020
Accepted: 21-10-2020

Corresponding Author:
Sk. Khairul Hasan
Department of Mechanical
Engineering University of
Wisconsin Milwaukee, WI
53211, USA
E-mail: skkhasan@uwm.edu

Abstract: The ability to combine positive features of an internal combustion engine with those of an electric motor has been fundamental to the advancement of the high-performance energy-optimized hybrid vehicles. However, due to a lack of reliable and realistic hybrid vehicle models, much of the hybrid vehicle controller research has been limited to computer simulations. To overcome this shortcoming, this paper utilizes a highly reliable vehicle model (Autonomie) for simulation. A state-of-the-art fuzzy logic controller was developed that considers the battery state of charge, wheel torque demand and vehicle speed as the input variables. An ARM Cortex M3 microcontroller-based control hardware prototype was developed and the processor in loop simulation was performed to verify the feasibility of the developed controller in an embedded real-time application. The results of this study indicate that the developed fuzzy logic controller significantly improved the performance (up to 48%) of the hybrid vehicle in a real-time application compared to Autonomie's built-in controller. The processor in loop test results provide evidence of the effectiveness of the developed control algorithm in the embedded real-time form.

Keywords: Fuzzy Logic Control, Plug-in Hybrid Vehicle, Autonomie, Fuel Economy, Processor in Loop Simulation (PIL)

Introduction

An ever-increasing demand for energy combined with a limited supply of sources has led to an increased awareness of the efficient uses of energy. Based on the US Energy Information Administration (U.S. EIA) annual report of 2018 (EIA, 2018), the amount of energy consumed by the transportation sector is often more or close to the residential, commercial, or industrial power sector. Ground vehicles consume the maximum amount of energy in the transportation sector. Consequently, a significant amount of attention is being given to the field of efficient energy management in-ground vehicles.

Plug-in Hybrid Electric Vehicle (PHEVs) is the latest generation in the hybrid electric vehicle family. Plugin hybrid electric vehicles are popular because of their higher fuel efficiency and low emissions. A majority of the plug-in hybrid electric vehicles involve a parallel architecture. In the case of the parallel hybrid vehicle, both the Internal Combustion Engine (ICE) and the electric motor work in parallel to fulfill the driver's demand torque. For the parallel hybrid vehicle, the distribution of demand power can be related to the driver's demand torque, State Of Charge (SOC) of the battery, vehicle speed, etc. During running conditions, the batteries recharge by regenerative braking system/engine. The performance of the plug-in hybrid electric vehicle depends on the distribution of the demand power between the engine and the electric motor. Because of the enormous scope in energy optimization of plug-in hybrid electric vehicles, significant research has been done to improve the efficiency of plug-in hybrid electric vehicles.

The plug-in hybrid electric vehicle consists of a complex power system with multiple sources and sinks. A sophisticated energy management system is required to maximize the efficiency of the plug-in hybrid electric vehicle. Summarized next are works by several researchers who have modeled the plug-in hybrid electric vehicle energy optimization problem in different ways.

Improving the efficiency and performance requires a realistic model of the vehicle. In the majority of the cases, the researchers developed their own vehicle models and the used models were often purely theoretical in nature (Naderi *et al.*, 2009; Bahar *et al.*, 2009; Kim *et al.*, 2010).

Science Publications

A significant number of studies have used the Advanced Vehicle Simulator (ADVISOR) vehicle simulator for performance evaluation of energy optimization algorithms (Kim *et al*., 2010; Meng and Langlois, 2010; Wipke and Cuddy, 1996; Slezak, 2004). ADVISOR was developed based on the simple laws of physics (Kilic *et al*., 2007). Most of the time, the real vehicle behavior deviates significantly from idealized vehicle models. A realistic vehicle model is essential for reliable performance evaluation.

Boyali and Guvenc (2010) used the Ford Otosan Hybrid Electric Vehicle (FOHEV) for performance evaluation of their developed algorithms (Kilic *et al*., 2007). FOHEV is a prototype vehicle (Majdi *et al*., 2009) that was developed through a collaboration between Ford Otosan, İTÜ and TÜBİTAK. It is not available to other researchers. As the vehicle model is not available to others, it cannot be used as a standard for performance evaluation.

Many researchers have attempted to solve the energy optimization problem by using artificial intelligence techniques such as fuzzy logic, neural networks and genetic algorithms (Naderi *et al*., 2009; Kim *et al*., 2010; Wipke and Cuddy, 1996; Zhu and Yang, 2012; Xu *et al*., 2010). Each of the three methods has its pros and cons. A fuzzy logic-based approach is attractive because of its simplicity and computational power requirements. It is based on capturing an expert's knowledge, often a small amount of data or no data required for development and implementation. The fuzzy logic is the best way to make a connection between a linguistic variable and a numerical variable. The neural network is another way to provide artificial intelligence to the system. This approach creates a mathematical model using approximate functions. The performance of the neural network depends on the training phase, which requires a considerable amount of data and computation. Collecting meaningful data for network training is challenging. As multiple layers and multiple neurons construct the neural network, it is difficult to explain how individual inputs influence the output. The genetic algorithm is another way to impart artificial intelligence to a machine. It requires significant computational times and may not reach the global optimum solution. For the reasons mentioned above, the fuzzy logic or modified fuzzy logic-based algorithms appear promising for solving the PHEV energy optimization problem and are used herein.

Boyali and Guvenc (2010) used neurodynamic programming to solve the plug-in hybrid electric vehicle optimization problem. A use of dynamic programming is difficult for real-time applications because it needs a priori information about the environment; further, the method requires significant computational times. For this reason, the authors developed an artificial neural network and utilized dynamic programming based optimized solutions for its training. Boyali and Guvenc reported a significant improvement in the fuel economy.

Bin *et al*. (2009) applied spatial domain dynamic programming to get the optimum solution for a given drive cycle. A precise vehicle model is needed for dynamic programming; further, the method works efficiently only on the predefined drive cycles. Kim *et al*. (2010) proposed a real-time optimal control strategy for the power split hybrid electric vehicle based on Pontryagin's minimum principle. The rate of fuel consumption and SOC of the battery were considered as the cost function.

Meng and Langlois (2010) used an optimized fuzzy logic controller to minimize fuel consumption and emissions. For training the fuzzy rules, an adaptive neuro-fuzzy approach was used. Xu *et al*. (2010) proposed a control strategy based on fuzzy logic for controlling a parallel hybrid electric vehicle. Driver torque demand and battery SOC were considered as the input to the fuzzy logic controller with engine torque and motor torque considered as the controller outputs.

Used an Adaptive Neuro-Fuzzy Inference System (ANFIS) for implementing the parallel hybrid electric controller. The ANFIS system utilizes optimal data sets for developing fuzzy sets and fuzzy rule base. Compared to the fuzzy logic based system, the performance of ANFIS is more dependent on training data. High-quality data needed for training the system is not readily available. The authors considered battery SOC and desired torque as the input of the fuzzy system and the output variable was the engine throttle control.

Developed a fuzzy logic-based global power management system for a permanent magnet electric vehicle transmission. To maximize the performance of the vehicle, they developed three individual fuzzy logic controllers which (i) split the power between the engine and the electric motor, (ii) maximized the energy capture during the braking process and (iii) sustained the SOC of the battery. The battery SOC, vehicle’s velocity, traction torque and vehicle’s requested power were considered as the input variables. The authors used the ADVISOR vehicle model for simulation and performance evaluation of the developed controller.

Developed a fuzzy logic controller for parallel hybrid vehicles considering the driver power demand, SOC of the battery and electric motor speed as the inputs for the fuzzy logic controller. They used the Powertrain System Analysis Toolkit (PSAT) for simulation and performance evaluation of the developed controller. The performance of the developed fuzzy logic controller was compared with the controllers that came with the PSAT package.

Developed a fuzzy logic controller for maximizing the fuel economy of the vehicle and minimizing the emissions. They considered the desired driving torque, SOC of the battery as the inputs for the fuzzy logic controller and throttle angle as the output of the fuzzy logic controller. For the performance evaluation ADVISOR vehicle modeling and simulation were used.

Developed a fuzzy logic controller whose structure is similar to the fuzzy logic controller mentioned in reference. They also used the ADVISOR software for the simulation and performance evaluation of the developed controller.

A review of the literature mentioned above reveals that most of the previous efforts are based largely on analytical models of the vehicle. Algorithms and controllers developed based solely on analytical models are of limited use on real vehicles. Some researchers used the dynamic programming method to solve the optimization problem in real-time, but the method requires prior knowledge about the trip making the solution specific to a particular route. Further, the method requires large computational power and significant computational times to arrive at a solution, which renders a real-time implementation quite challenging. Efforts have also been made that combine the geographical information and global positioning system data with dynamic programming. However, geographic information data is not available in all areas. Some researchers have used SOC of the battery and driver torque demand as the input variable, whereas others have used vehicle speed and SOC as input variables; often, two variables among three quantities are not enough to describe the state of the vehicle. The above literature also reveals that most of the previous works have been limited to simulations alone without any consideration of hardware implementation.

To overcome these shortcomings, this paper addresses the modeling problem by using a reliable vehicle model based on the Autonomie vehicle simulation software developed by Argonne National Lab. Autonomie utilizes real data collected from an actual vehicle to construct a vehicle model. Until now, none of the research efforts have used this type of highly reliable vehicle model for controller development and performance evaluation. Three input variables are considered for the developed fuzzy logic controller: The state of charge of the battery, the vehicle speed and the driver's demand torque. Most of the previously developed fuzzy logic controllers used only two variables as the input. Vehicle speed can play an important role in optimum decision making. An expert's knowledge was gathered and transferred in the form of a total of 75 fuzzy rules and fuzzy set theories were used to help improve the efficiency of a fuzzy logic-based controller. Comparisons were also performed for evaluating vehicle performance using Autonomie's built-in controller as well as the fuzzy logic controller developed herein. A processor in loop simulation was also performed to verify the controller feasibility in an embedded real-time application. Finally, an Arm Cortex M3 microcontroller-based controller prototype was built for the considered application.

Methodology

The solution approached adopted herein is briefly described next. This is followed by a detailed explanation of what actions were performed in each step.

For energy optimization of the hybrid vehicle, a fuzzy logic controller was developed. The controller development involves defining fuzzy sets for the three input variables and two output variables. This was followed by a development of the fuzzy rule base. Next, the fuzzification and defuzzification approaches were established. The fuzzification method converts a crisp input to a fuzzy input and the defuzzification method converts a fuzzy value to a crisp output. It is important to mention that fuzzy sets, membership functions, fuzzy rule base and fuzzification and defuzzification methods completely depend on the designer’s/expert’s knowledge. A more detailed explanation of each of these steps is given in section 2.3.

To verify the performance of the developed controller, a highly reliable vehicle model is used. The vehicle model is derived from the Autonomie software package developed by Argonne national lab. A detailed description of the vehicle model used for simulation is given in section 2.1. The developed fuzzy logic controller replaced the default controller in Autonomie. The performance of the designed controller was compared with the available controller in the Autonomie package.

The simulation results are presented in section 3. Simulink was used for automatic code generation and hardware implementation. An Arm Cortex M3 microcontroller was used as the prototype controller. To evaluate the performance of the developed controller, a processor in loop simulation was performed where the output of the simulation was compared with the output of the developed prototype hardware. The simulation results show that there is no difference between the simulation output and the actual controller’s output, demonstrating that the developed controller is performing as intended (section 4).

Vehicle Simulation

Autonomie is a Matlab/Simulink® based vehicle modeling and simulation software. It provides a realistic estimation of wheel torque demand, fuel economy and SOC of the battery. Additionally, it provides engine, electric motor efficiency curves and emissions information. The software includes a driver block to simulate engine start/stop, throttle control and braking action. The vehicle can also be run through different standard driving cycles.

Autonomie also provides flexibility to change any default parameters such as the mass of the vehicle, environmental parameters such as surrounding temperature and pressure, etc. It allows engineers system-level development and performance evaluation that includes replacing an existing propulsion controller, an engine or battery with a newly modeled one, etc.

Figure 1 shows the internal power flow diagram of a hybrid vehicle. The engine and electric battery together supply the power to run the wheels. The regenerative braking system recharges the battery during the braking action. The whole power system runs based on the vehicle propulsion control algorithms. We used fuzzy logic algorithms to provide artificial intelligence to the vehicle propulsion controller. Each component in the model in Fig. 1 is based on a lookup table with table entries provided in the Autonomie software.

Different dynamometer drive schedules are used for determining the fuel economy and vehicle emissions under various driving conditions. For instance, the Environmental Protection Agency (EPA) highway fuel economy testing drive schedule is used for measuring highway gas mileage and emissions. The Urban Dynamometer Driving Schedule (UDDS) cycle is used herein for the simulation and performance evaluation of the vehicle propulsion controller.

The urban dynamometer driving schedule simulates the traffic and environmental effects on busy American city roads and highways. The UDDS cycle has the following specifications: total driving duration of 8219 sec, traveled a distance of 44.27 miles with a maximum speed of 56.7 mph and an average speed of 19.58 mph. Figure 2 shows the vehicle speed during the UDDS cycle.

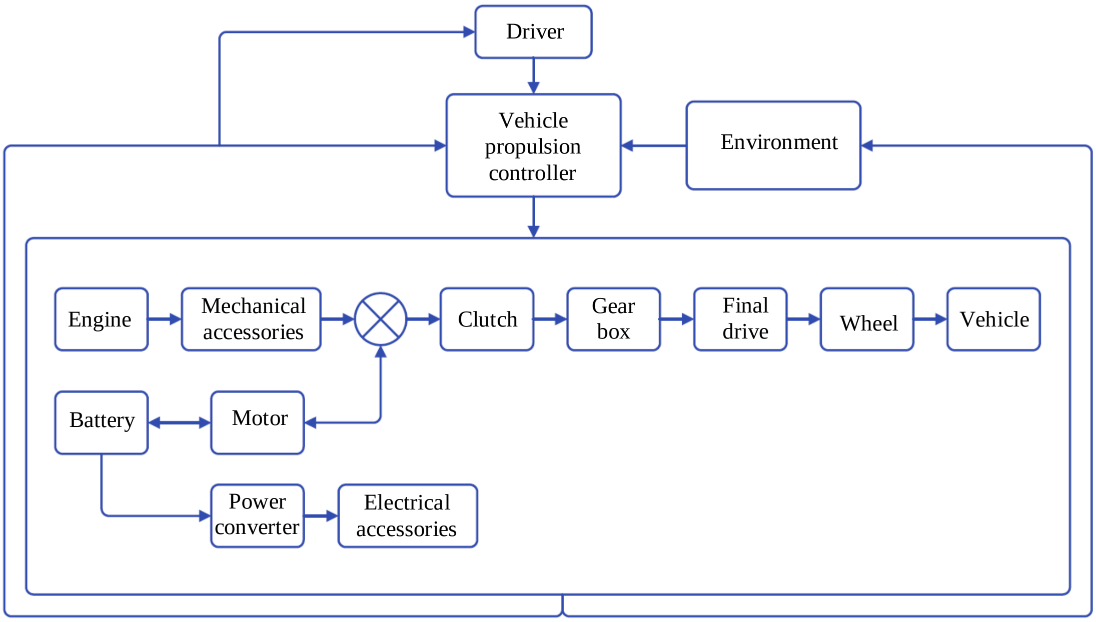


Fig. 1: Hybrid vehicle power flow diagram

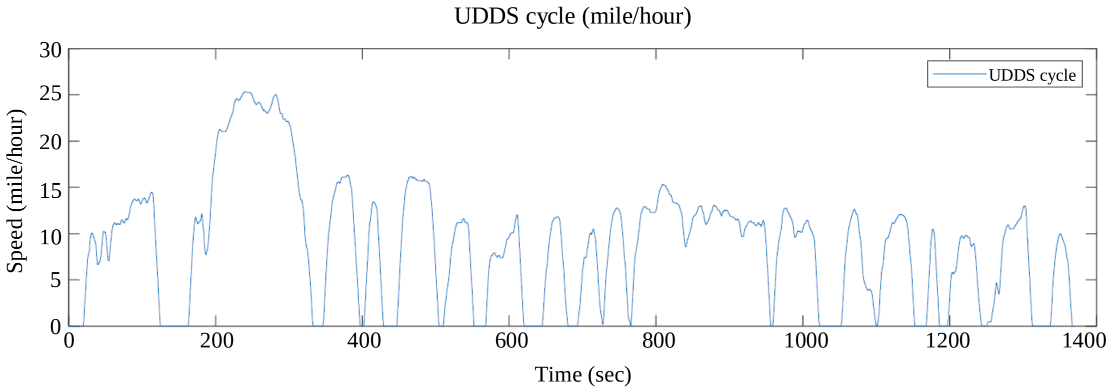


Fig. 2: Vehicle speed along the UDDS cycle

Table 1: Vehicle simulation parameters

Component	Specifications
Engine	GM ECOTECH, four cylinders, cylinder volume 2.2-liter, power output 110 kilowatt
Electric motor	PM DC motor, continuous power 7 kW, maximum torque 140 Nm, the coefficient of regeneration 1
Transmission	5 speed manual transmission (ratios 2.563, 1.552, 1.022, 0.727, 0.52)
Final drive	Gear ratio 4.438
Wheel	Radius 0.3175 m
Battery	Li-ion battery 555 amp-hrs.
Coefficient of drag	0.30
Vehicle mass	1630 kg
Ambient pressure	1 bar
Ambient temperature	20°C

While the vehicle executes the UDDS cycle, the driver block takes necessary actions to follow the prescribed trajectory. Autonomie calculates the performance of individual components, fuel consumption and SOC of the battery. At the end of the simulation, it generates a detailed report on the performance and state of the components. Table 1 shows the vehicle and environmental parameters used for the simulation. The developed fuzzy logic controller was simulated using the vehicle given in Table 1, but it is not necessary to use the same model or the exact specifications. The controller can be used for other configurations such as an automatic transmission, diesel engine, six-cylinder engine, different final drive gear ratios, etc.

Autonomie Vehicle Model Propulsion Controller

Autonomie vehicle model comes with a built-in vehicle propulsion controller. Figure 3 presents the state flow diagram of the default vehicle propulsion controller. It considers multiple factors while executing control actions such as the vehicle torque demand, the state of the battery, the state of the engine and the state of the gearbox. Based on the wheel torque demand and engine speed, it turns on/off the internal combustion engine. Depending on the engine speed and wheel torque demand, it switches system state between stop/braking and propulsion mode. Once the system is in propulsion state, it checks the state of the gearbox and the state of charge of the battery. Based on these two pieces of information, the controller switches between power-assist mode and battery charging mode. Finally, the system state switches between propulsion mode and performance mode. In the performance mode, if the battery state of the charge is above a certain threshold, it draws the maximum possible amount of torque from the motor, which leads to maximum energy efficiency and minimum pollution. Figure 4 presents the state flow diagram of the performance mode.

Autonomie default controller is based on the state flow diagram where state switching occurs based on different conditions. Simply stated, it is based on a lookup table. A lookup table-based decision-making system is a commonly used approach. The main advantage of lookup table based algorithms is that they are easy to implement and require smaller computational times and processing power. The main drawback of this approach is that it does not ensure an optimum solution. To overcome this shortcoming, we proposed a fuzzy logic controller based approach to find an optimum solution.

Development of a Fuzzy Logic-Based Controller

Fuzzy logic is a well-established method for providing artificial intelligence to a machine. A set of well-defined mathematical procedures are used to convert approximate human reasoning capabilities to a knowledge-based system. The fuzzy logic-based control system gives precise output to a high bandwidth system. It does not require high computational power like a neural network or a genetic algorithm. Because of its ease of adaptability and high performance, nowadays, fuzzy logic is popular for prediction systems, modeling and control applications. Fuzzy logic-based systems are suitable where expert's well-documented knowledge is available. It shows excellent performance for systems with complex/unknown plant models, nonlinear and noisy sensor outputs (De Silva, 2018; Passino and Yurkovich, 1997).

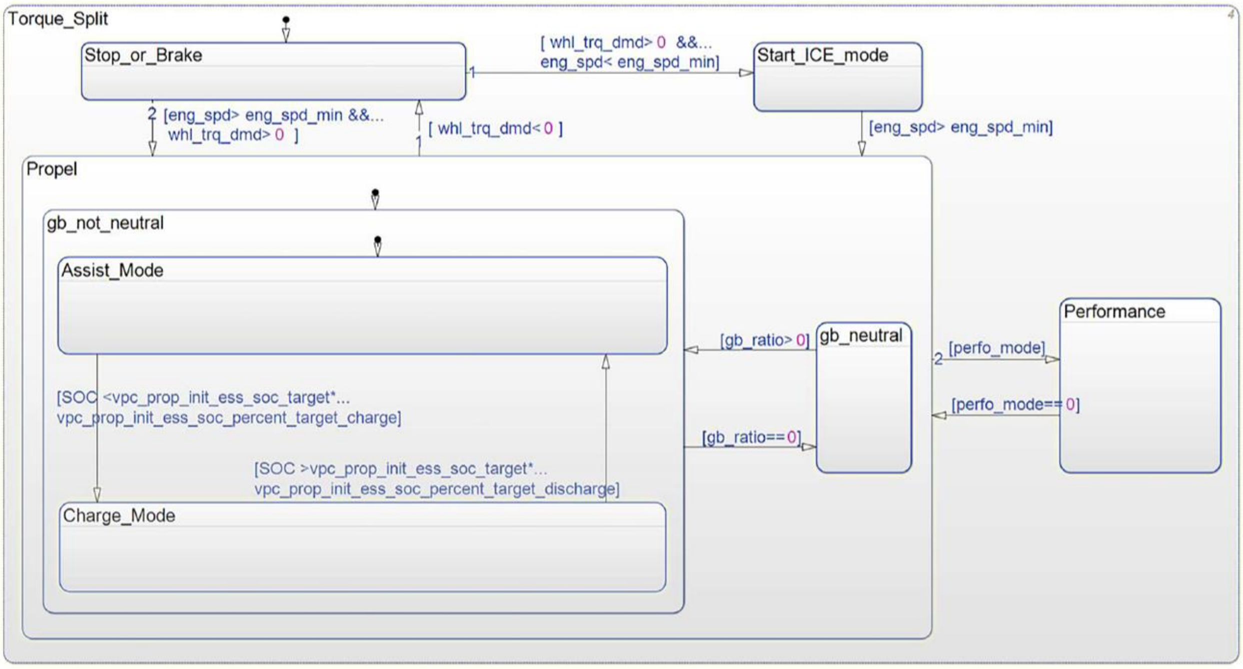
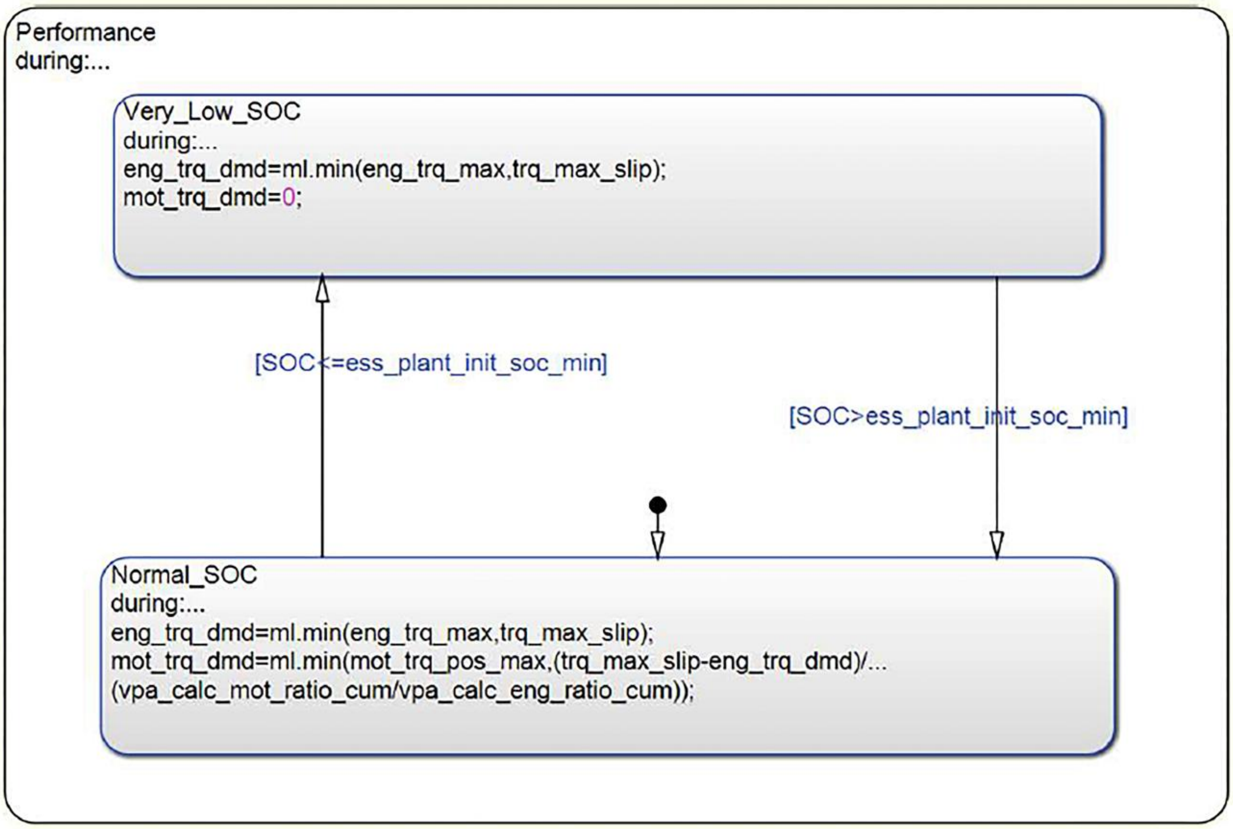


Fig. 3: Autonomie default hybrid controller



```
Very_Low_SOC
during:...
eng_trq_dmd=ml.min(eng_trq_max,trq_max_slip);
mot_trq_dmd=0;
```

```
Normal_SOC
during:...
eng_trq_dmd=ml.min(eng_trq_max,trq_max_slip);
mot_trq_dmd=ml.min(mot_trq_pos_max,(trq_max_slip-eng_trq_dmd)/...
(vpa_calc_mot_ratio_cum/vpa_calc_eng_ratio_cum));
```

Fig. 4: Vehicle propulsion controller performance mode

The structure of the fuzzy logic-based control/decision-making system is shown in Figure 5. A fuzzy logic controller has the following major components: fuzzy knowledge base, fuzzifier, fuzzy rule base, inference engine and defuzzifier. The fuzzy knowledgebase captures the expert's knowledge required for fuzzification, defuzzification and the fuzzy inference engine. The fuzzifier converts crisp input to fuzzy values and contains fuzzy sets for every input. Each fuzzy set is described by fuzzy membership functions. For instance, Figure 6 shows the fuzzy set corresponding to the battery state of charge. It consists of three membership functions: Low SOC (SOC<40%), medium SOC (10% <SOC<90%) and high SOC (SOC>60%). Expert(s) specify the number, shape and size of the membership functions. These specifications are completely dependent on the expert’s knowledge. The fuzzy inference system enables approximate human reasoning capabilities to the fuzzy logic system. It consists of a series of *if-then* rules. The fuzzy inference system is also developed based on the expert’s knowledge. Two different experts may define the inference system differently. It is very difficult to say which rule sets will deliver better performance. The defuzzifier converts fuzzy values obtained from the fuzzy inference system into crisp values. Fuzzy sets and fuzzy membership functions used for defuzzification also come from the expert's knowledge. In a fuzzy logic-based system, there are no fixed rules for defining the fuzzy sets and membership functions and for the fuzzification and defuzzification methods.

A fuzzy logic controller was developed for the energy optimization of plug-in hybrid electric vehicles. The developed controller has three input variables: vehicle speed, wheel torque demand and state of charge of the battery and two output variables: Motor torque demand and engine torque demand. The controller was developed by considering that the battery can be recharged directly from the electrical power grid as well from the engine. Input-output fuzzy sets and fuzzy membership functions were built based on the expert's knowledge. Figure 6 to 8 depicts the fuzzy sets and membership functions for the three input variables.

The battery state of charge is represented using three membership functions, namely, low, medium and high (Fig. 6). If the state of charge is less than 0.4, it is considered as low. If it lies between 0.10 to 0.90, then it is considered as a medium and if the state of charge is more than 0.6, then it is considered as high value. Similarly, fuzzy sets and membership functions for the wheel demand torque (Fig. 7) and vehicle speed (Fig. 8) were defined.

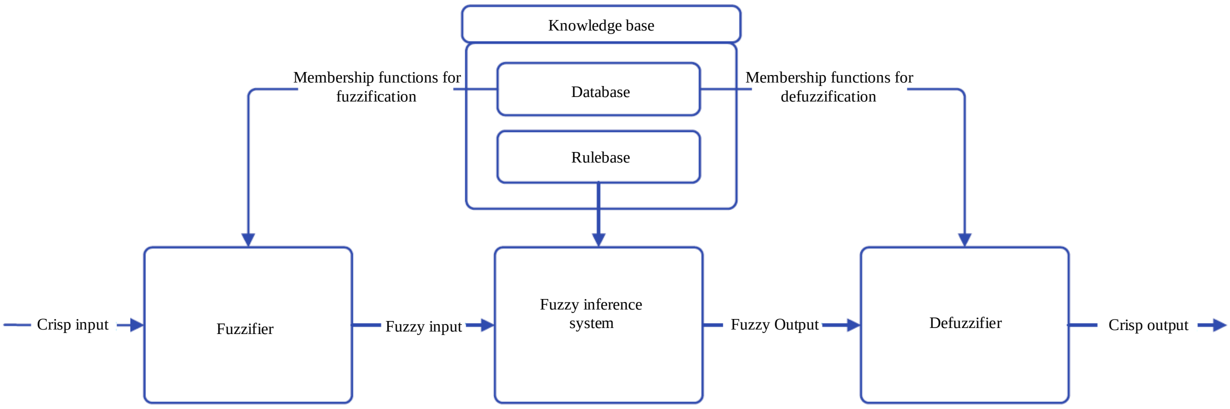


Fig. 5: Fuzzy logic system architecture

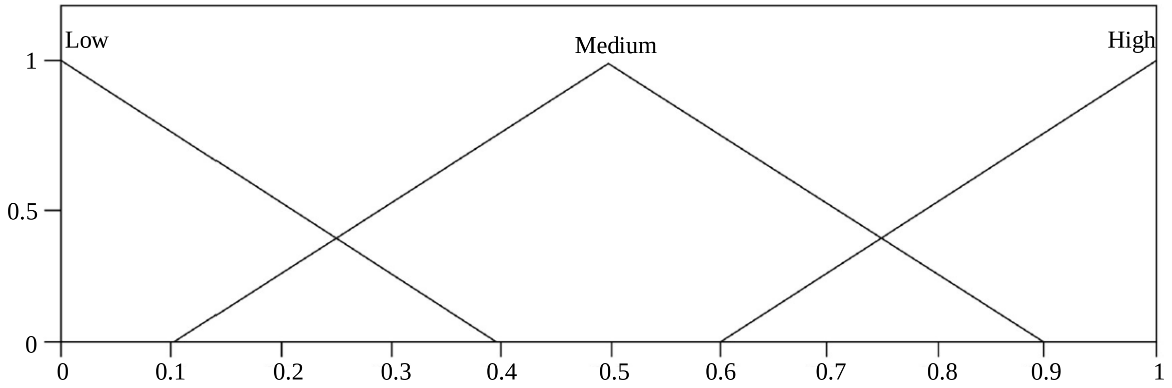


Fig. 6: Membership function of battery state of charge (SOC)

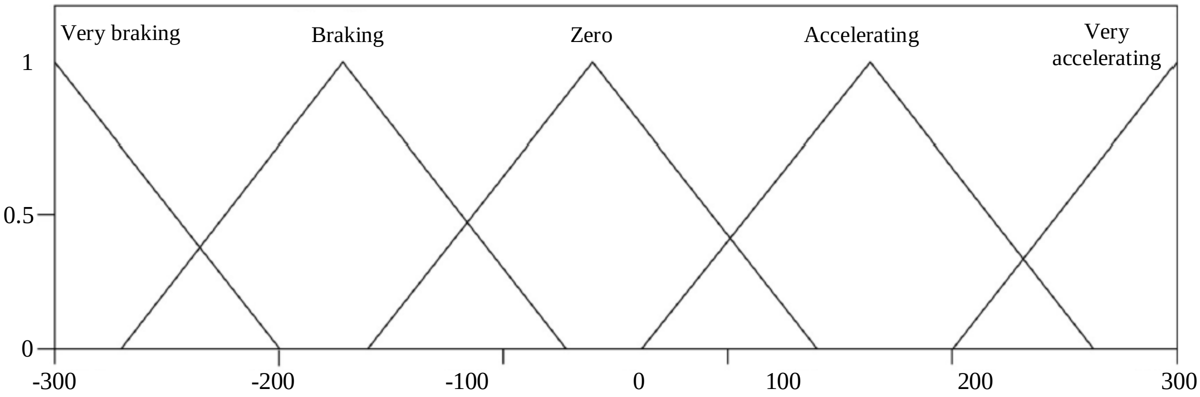


Fig. 7: Membership functions of wheel torque demand

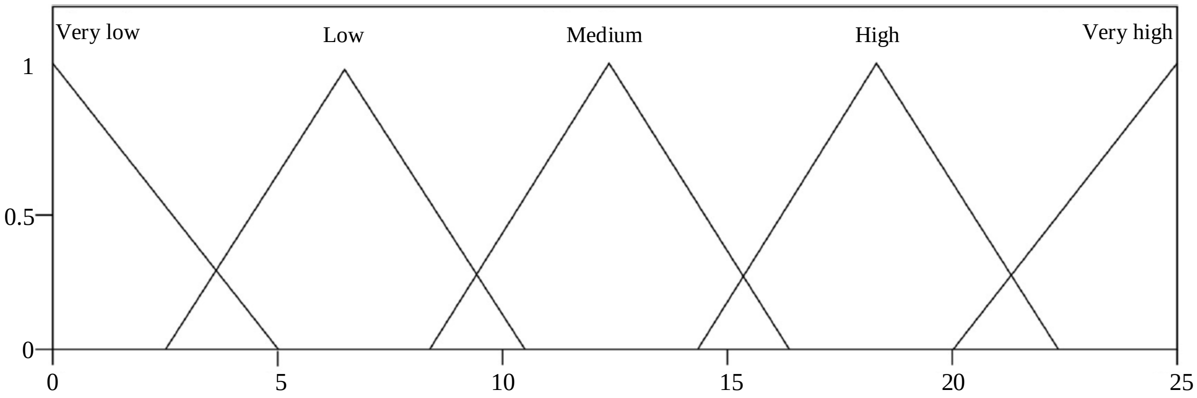


Fig. 8: Membership functions of vehicle speed

Table 2: Fuzzy input variables and membership function

Input variables	Membership function	
State of charge of the battery 0.33	$\mu_{\text{low batt}}$	= 0.20
	$\mu_{\text{medium batt}}$	= 0.56
Vehicle speed 10 m/s	$\mu_{\text{low speed}}$	= 0.20
	$\mu_{\text{medium speed}}$	= 0.45
Driver torque demand -195 Nm	μ_{braking}	= 0.50
	$\mu_{\text{very braking}}$	= 0.22

The fuzzy inference system maps fuzzy input to fuzzy output. Two fuzzy output sets were defined to convert the fuzzy output to crisp output. Figures 9 and 10 show the fuzzy sets and membership functions of the two output variables.

Fuzzification is the process of mapping the crisp input variables on the fuzzy set to determine the membership values of the fuzzy membership functions. For example, at some instant (Fig. 11), the battery SOC of 0.33 covers two membership functions – low SOC and medium SOC. The membership in low SOC is 0.20, whereas the membership in medium SOC is 0.56. Similarly, based on the crisp values of the other two individual inputs, the relevant fuzzy set's membership function's membership values were calculated. Figures 12 and 13 show this calculation for wheel torque demand and vehicle speed, respectively.

After the fuzzification process, every input comes with specific membership value for a membership function. Suppose the value of the state of charge of the battery is 0.33, the driver torque demand is -195 Nm and the vehicle speed is 10 m/s, then for these input values, the corresponding memberships are given in Table 2.

After fuzzification, the next step in the fuzzy logic system is the fuzzy inference system. Fuzzy inference rule base consists of a set of antecedent-consequent linguistic rules relating system inputs and outputs using fuzzy sets. The fuzzy inference system is a logical presentation of an expert's knowledge. For example, a rule may be stated as.

Rule 1

if **STATE OF CHARGE** is **LOW, WHEEL TORQUE DEMAND** is **BRAKING** and **VEHICLE SPEED** is **LOW** then **MOTOR TORQUE** is **GENERATOR** and **ENGINE TORQUE** is **MEDIUM.**

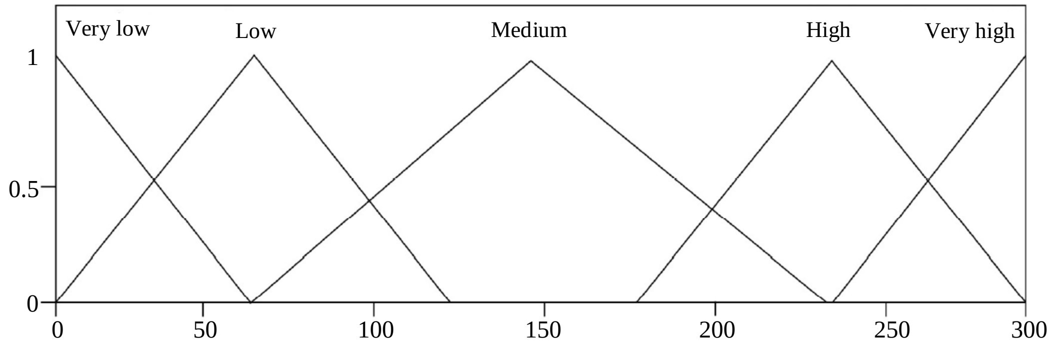


Fig. 9: Membership functions of engine torque demand

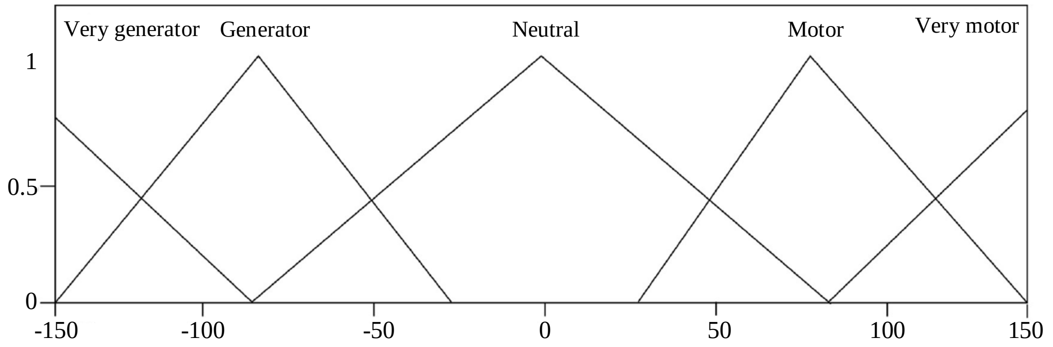


Fig. 10: Membership functions of motor torque demand

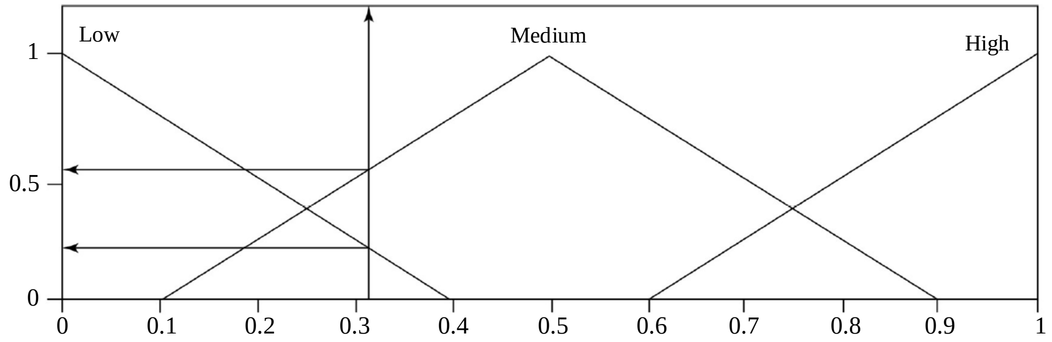


Fig. 11: Fuzzification of battery state of charge

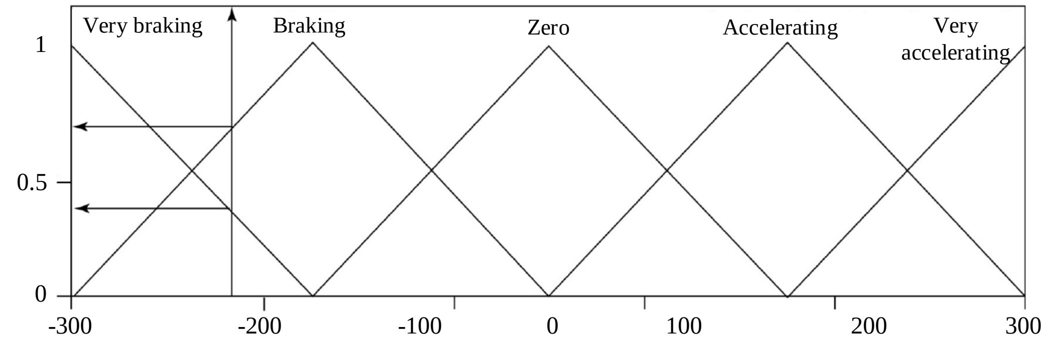


Fig. 12: Fuzzification of equivalent wheel torque demand

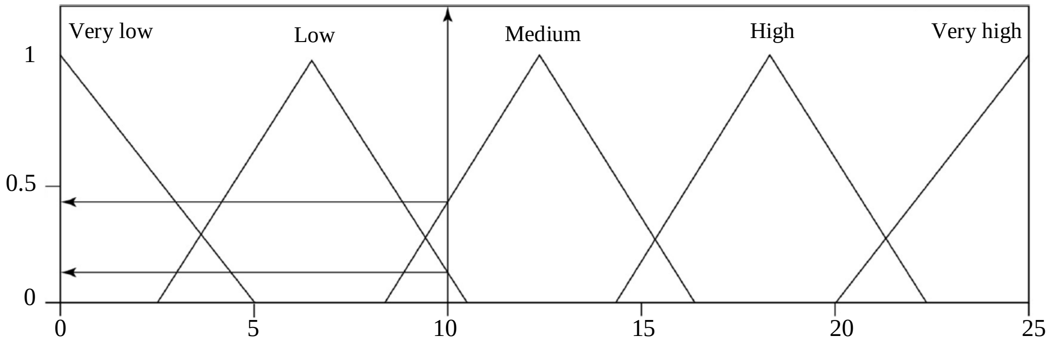


Fig. 13: Fuzzification of equivalent vehicle speed

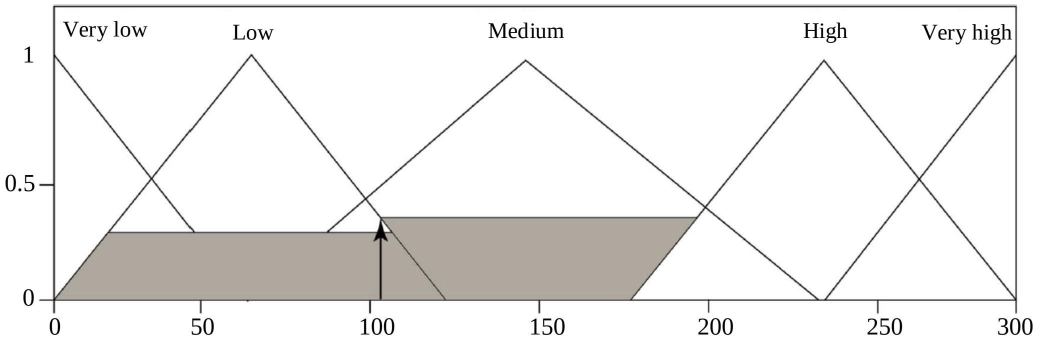


Fig. 14: Defuzzification process for engine output torque

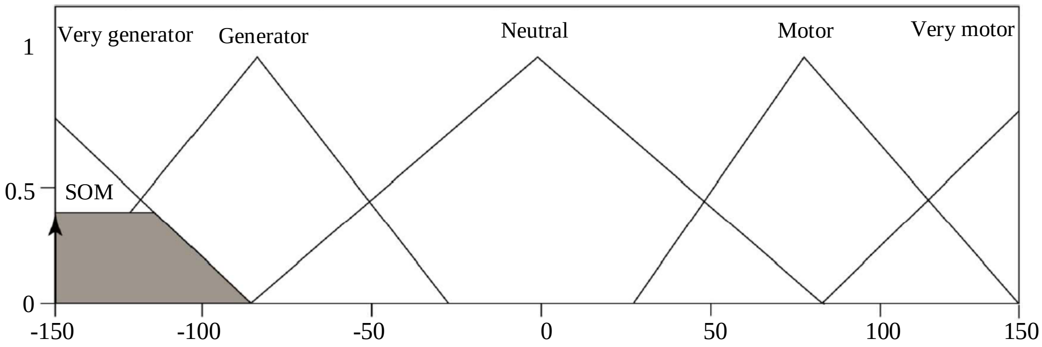


Fig. 15: Defuzzification process for motor output torque

Rule 2

if **STATE OF CHARGE** is **MEDIUM** and **WHEEL TORQUE DEMAND** is **VERY BRAKING** and **VEHICLE SPEED** is **MEDIUM** then **MOTOR TORQUE** is **VERY GENERATOR** and **ENGINE TORQUE** is **LOW**.

A total of 75 such rules were developed for the proposed fuzzy logic controller.

Determining the Firing Strength

After fuzzification and fuzzy inferencing, firing strength is used for determining the membership of the output fuzzy sets. As the fuzzy sets are continuous with the "and" operation being equivalent to the minimum of the contributing membership function's membership, the firing strengths of rules 1 and 2 are determined as follows.

If state of charge is low, wheel torque demand is braking and vehicle speed is low then motor torque is generator and engine torque is medium.

Since the "and" operation is equivalent to the minimum of all memberships, so:

$$\mu_{\text{low batt.}} \text{ and } \mu_{\text{low speed}} \text{ and } \mu_{\text{braking.}}$$
$$= \min(\mu_{\text{low batt.}}, \mu_{\text{low speed}}, \mu_{\text{braking.}})$$
$$= \min(0.20, 0.20, 0.50)$$

So, the firing strength of the first rule is 0.20.

Similarly,

If state of charge is medium and wheel torque demand is very braking and vehicle speed is medium then motor torque is very generator and engine torque is low:

$\mu_{\text{medium batt.}}$ and $\mu_{\text{medium speed}}$ and $\mu_{\text{very braking.}}$
$= \min(\mu_{\text{medium batt.}}, \mu_{\text{medium speed}}, \mu_{\text{very braking.}})$
$= \min(0.56, 0.45, 0.22)$

So, the firing strength of the second rule is 0.22.

Defuzzification is the process of getting the fuzzy output membership function to a crisp output value. The smallest of the maxima defuzzification method is used.

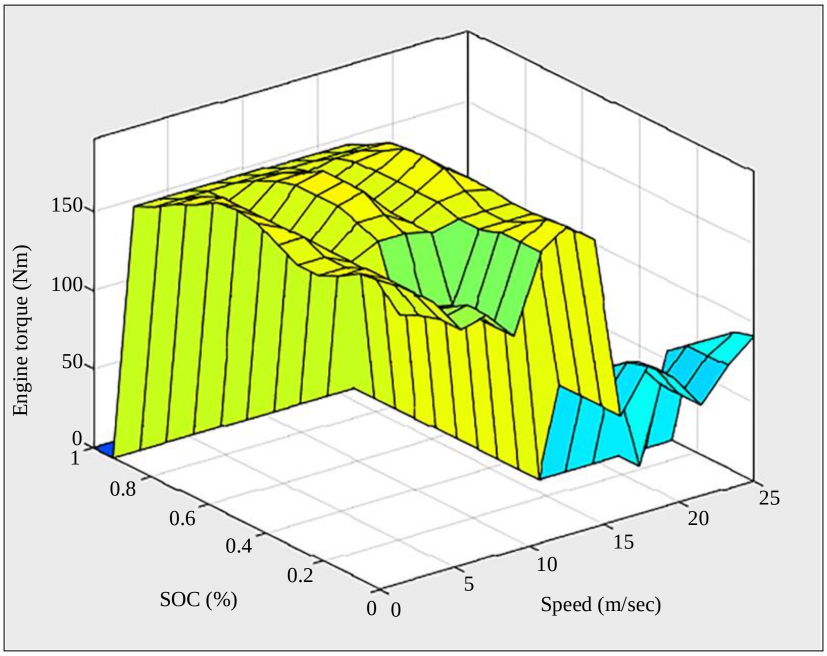


Fig. 16: Output engine torque variation with respect to speed and driver torque demand

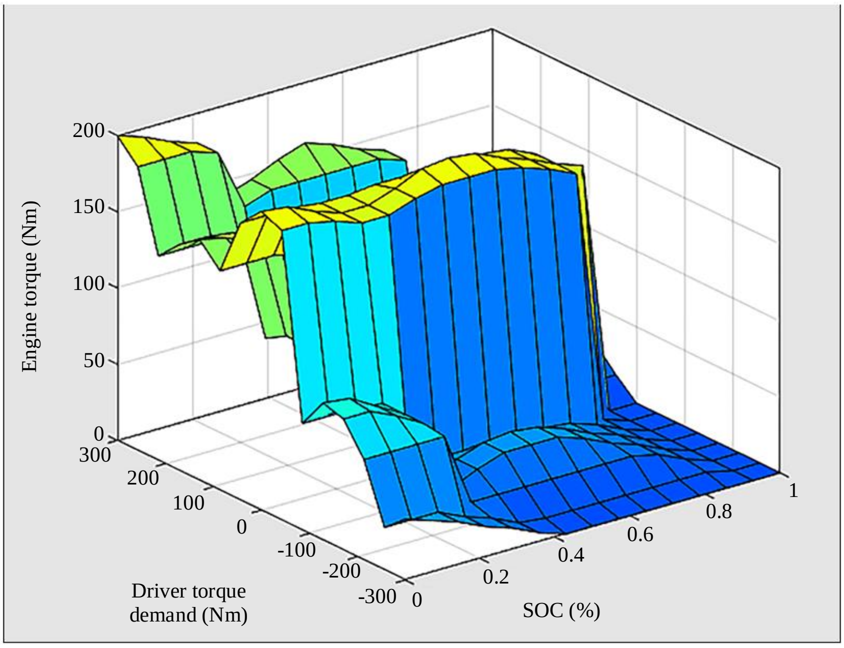


Fig. 17: Engine output torque variation with respect to the State Of Charge (SOC) of the battery and driver torque demand

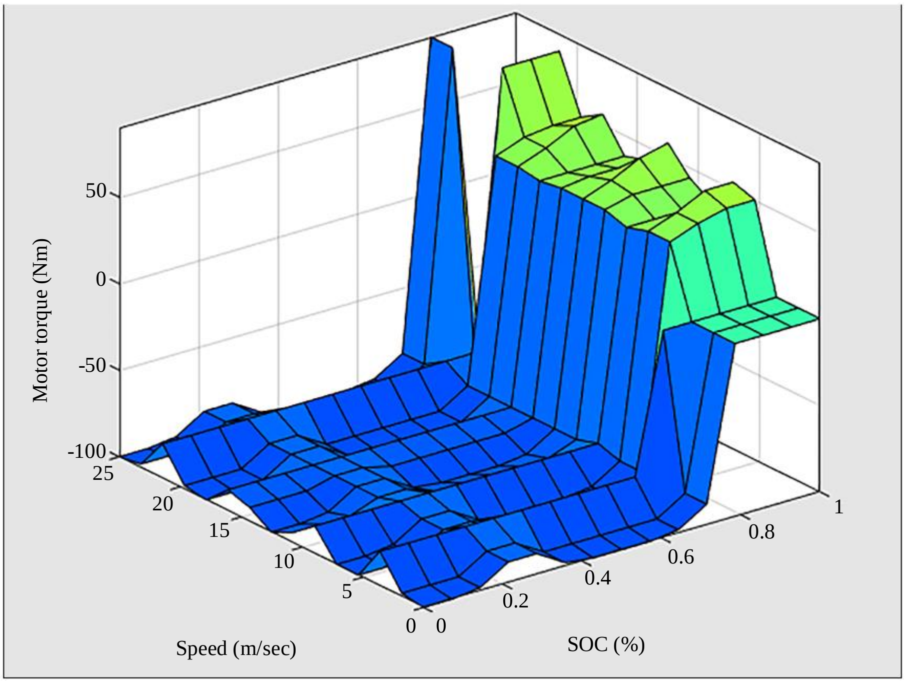


Fig. 18: Motor output torque variation state of charge (SOC) of the battery and vehicle speed

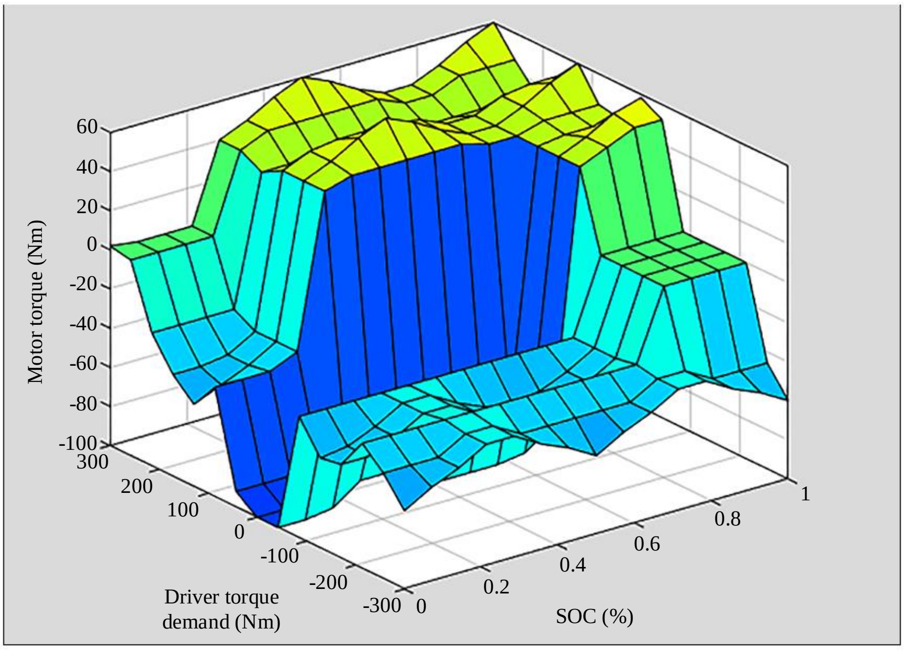


Fig. 19: Motor output torque variation with respect to the State Of Charge (SOC) of the battery and driver torque demand

From the fuzzy rule base, two membership functions from output variable Engine Torque contributed. Based on the smallest of the maxima defuzzification process, the crisp output is 101 *Nm*. Similarly, for the Motor Torque, the crisp output is -150 *Nm*. Figures 14 and 15 show the defuzzification process.

The fuzzy logic controller has a total of 75 rules with three input variables: State of charge of the battery, vehicle speed, wheel torque demand and two output variables, namely engine output torque and motor output torque. The variation of outputs torques with respect to changes in inputs are shown in Fig. 16 to 20.

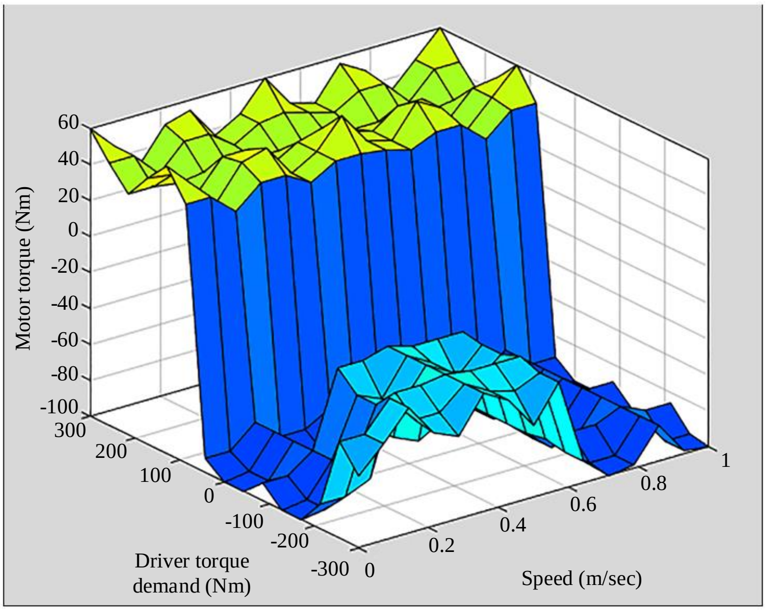


Fig. 20: Motor output torque variation with respect to the speed and driver torque demand

Simulation and Result Analysis

The UDDS cycle simulates the traffic and environmental effects on busy American city roads and highways. We used the UDDS cycle to measure the fuel economy. Figure 21 shows the vehicle's tracking performance for the UDDS cycle. Since the tracking error is small, it is difficult to differentiate between the input trajectory and the vehicle output trajectory. Figure 22 shows the engine on/off states while completing the UDDS cycle. At the end of the simulation, Autonomie provides the performance characteristic curves of all the components, the fuel consumption and the battery state of charge.

Table 3 and Fig. 23 show the variation of the SOC of the battery and fuel economy versus the total distance driven using the developed fuzzy logic controller.

Figure 23 shows the variations of vehicle mileage to the state of charge of the battery. The user will get 42 or more miles per gallon for the first 150-miles. Figure 24 shows the performance comparison between the developed fuzzy logic controller and the default controller that comes with the Autonomie package. At fully charged condition, the vehicle with fuzzy logic controller gave 16 more miles per gallon of fuel than the default Autonomie controller; this is equivalent to 48% additional mileage than the default mileage. Based on a 2014 American survey of driving habits, Americans, on average, drive 32.60 miles per day (Triplett *et al*., 2015). From Fig. 24, it can be easily concluded that the average driver will get a mileage upwards of 50 miles per gallon.

Figure 23 shows the variation of the fuel economy with the change of the SOC of the battery. The developed fuzzy logic controller gives very high mileage for the first 175 miles (35 miles/gallon or higher). Due to the high SOC of the battery, based on the developed fuzzy logic controller, the electric motor contributes more driving torque to fulfill the driver’s demand, resulting in a higher fuel economy. If the battery is not recharged, the vehicle constantly delivers 34.65 miles/gallon. The vehicle performance in terms of fuel economy also decreases with the SOC of the battery.

Table 3: total distance traveled, SOC of the battery and Fuel economy

Cycle #	Distance per cycle	Total distance	Initial SOC (%)	Final SOC (%)	Δ SOC (%)	Equivalent fuel economy (mile/gal.)
0	44.72	0.00	100.00	100.00	0.00	52.81
1	44.72	44.72	100.00	73.38	26.62	45.19
2	44.72	89.44	73.38	51.61	21.77	46.68
3	44.72	134.16	51.61	38.15	13.47	41.61
4	44.72	178.88	38.15	38.15	1.94E-05	34.71
5	44.72	223.6	38.15	38.15	-5.76e-7	34.67
6	44.72	268.32	38.15	38.15	-1.52e-5	34.65
7	44.72	313.04	38.15	38.15	1.49e-5	34.67
8	44.72	357.76	38.15	38.15	-3.69e-7	34.64

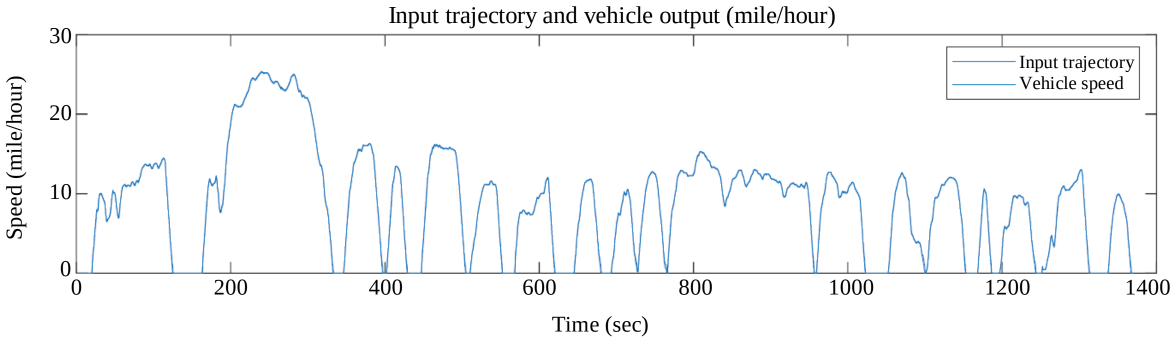


Fig. 21: UDDS cycle and vehicle output speed along the UDDS cycle

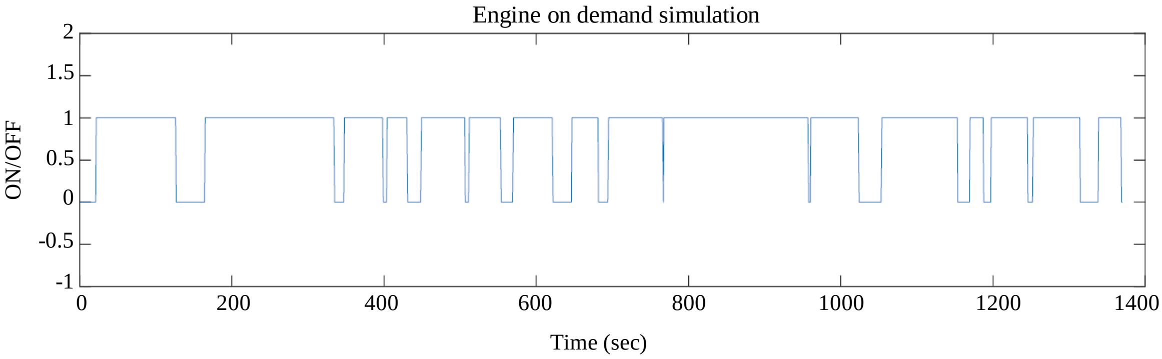


Fig. 22: Vehicle engine On/Off simulation

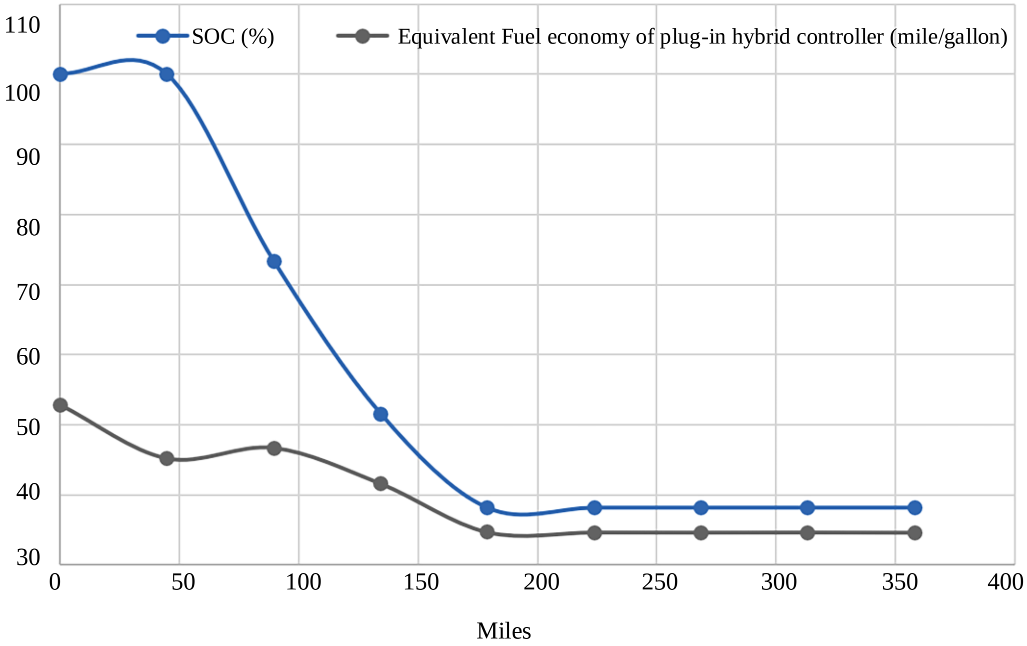


Fig. 23: State Of Charge (SOC) of the battery and fuel economy using the developed controller

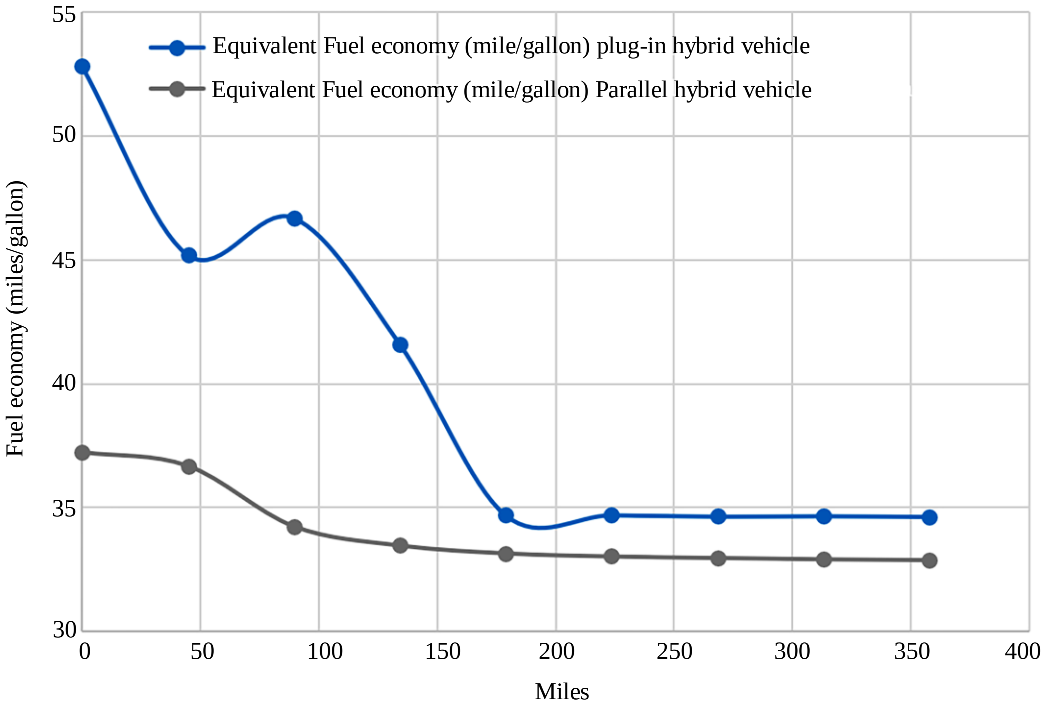


Fig. 24: Comparison of fuel economy between developed and Autonomie's default controller

Processor in Loop Simulation

The processor in the loop is a well-established technology for early problem detection and numerical performance evaluation of control algorithms in an embedded environment (MathWorks, 2019). The processor in the loop plays a crucial role in model-based controller development. It helps to reduce the dependency on physical hardware. Figure 25 shows the PIL simulation architecture.

From Figure 25, we can see that for PIL simulation, the plant model and the control algorithms run on the host computer and another copy of the control algorithm runs on the external microcontroller or DSP chip. The same input is provided to both controllers and then their outputs are compared. We used Matlab-Simulink for code generation and PIL simulation. An ARM Cortex-M3 microcontroller was used as the target processor. The host computer (Intel Core i7 processor-based Windows PC) communicates with the target processor via a high-speed serial bus. The developed fuzzy logic controller runs in parallel on the host computer and on the Arm Cortex M3 microcontroller. At every iteration, the output from the Arm controller is sent back to the host computer and compared with the output of the control algorithm running on the host computer. Figure 26 shows the difference between the simulated controller's output (running on the host computer) and the target ARM controller's output. As there is no difference (curve is zero throughout) between the simulation output and the controller's output (Figure 26), this guarantees that the developed controller is behaving as expected in the embedded real-time application.

Controller Prototype Development

An ARM Cortex M3 microcontroller was used for developing a control prototype (Figure 25). ARM Cortex M3 Microcontroller has twelve analog inputs and two analog outputs. The battery state of charge, vehicle speed and driver torque demand were considered as the controller inputs. As the ARM Cortex M3 microcontroller accepts a maximum input value of 3.3 volts, all input sensor values were mapped between 0 to 3.3 volts. Three analog input channels (A0, A1, A2) and two built-in analog output channels (DAC0, DAC1) were utilized for interacting with the sensors and delivering the output signals. The analog output voltage represents the engine torque demand and motor torque demand command. The microcontroller output voltage varies between 0.55 volt to 2.75 volts. Therefore, the engine torque demand and motor torque demand were expressed by 0.55 to 2.75 volts. The controller prototype is shown in Figure 27. For testing purposes, instead of using data from real sensors, three potentiometer based simulated sensors were used.

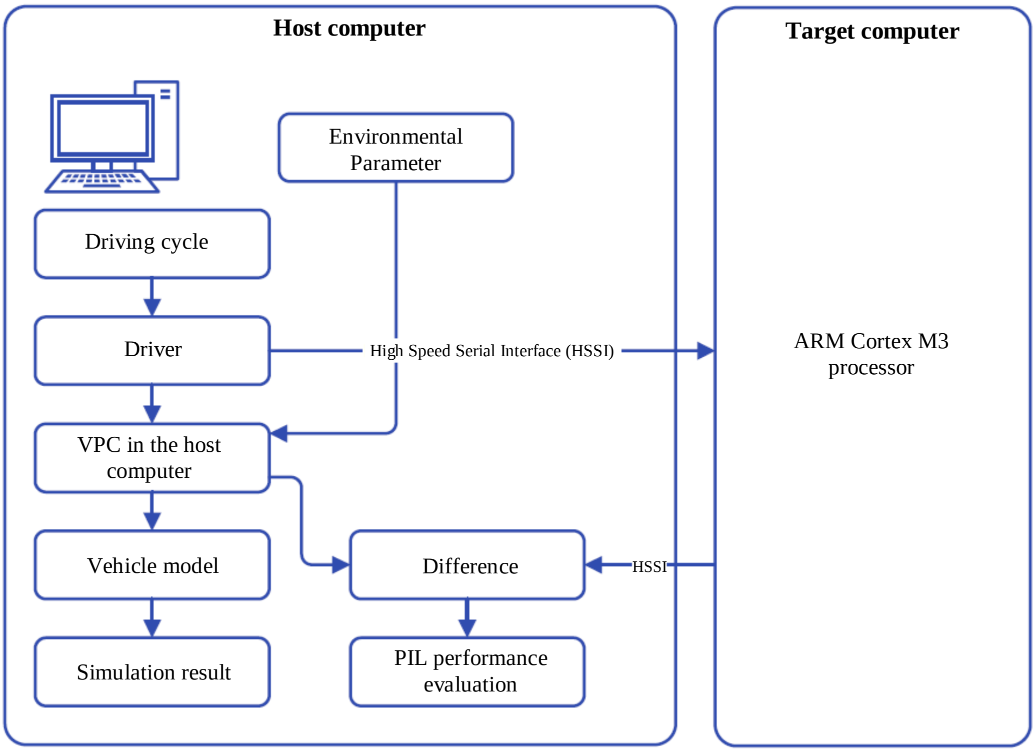


Fig. 25: Processor In Loop (PIL) simulation architecture

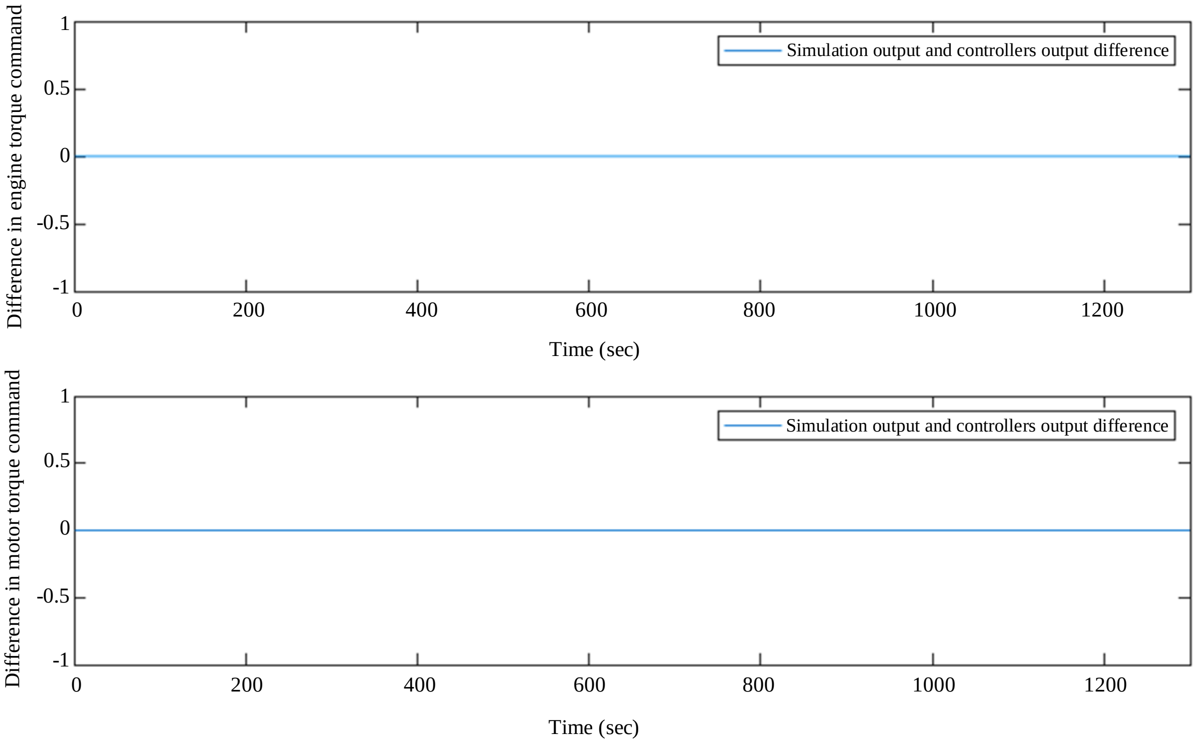


Fig. 26: Performance verification of processor in the loop simulation

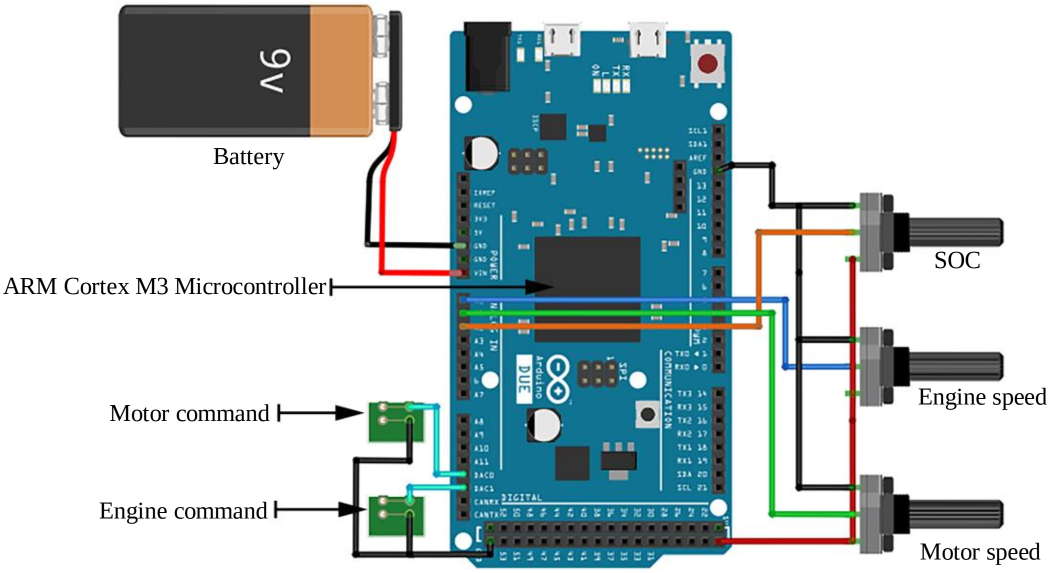


Fig. 27: Control hardware prototype

Conclusion

This paper presented the development of a fuzzy logic controller for a plug-in hybrid vehicle, which improved the fuel efficiency of the vehicle by up to 48 percent compared to the default controller available in the Autonomie vehicle simulation software. The battery SOC, wheel torque demand, vehicle speed are considered as the input of the fuzzy logic controller. The expert's knowledge was captured to develop the fuzzy rule base (75 rules) and fuzzy sets for the controller. For vehicle modeling and performance evaluation of the designed controller, a highly reliable and real data based vehicle model from the Argonne National Lab was used. For hardware realization and performance verification of the developed fuzzy logic control algorithms, an Arm Cortex M3 microcontroller based prototype was developed. The processor in loop simulation was performed to evaluate the performance of the controller in an embedded real-time application. While we have done our best in capturing the expert's knowledge in the form of 75 rules, it may be noted that a more precise and informative capturing of the expert's knowledge can yield further improvements in the controller performance.

Acknowledgement

Sk. Khairul Hasan was supported by an assistantship from the Mechanical Engineering Department, University of Wisconsin-Milwaukee. This support is gratefully acknowledged.

Author's Contributions

Sk. Khairul Hasan: Literature search, Modeling and Simulation, paper organization and writing.

Anoop Dhingra: Paper organization, proof-reading and editing the manuscript.

Ethics

This article is original and contains unpublished material. Authors declare that are no ethical issues and no conflict of interest may arise after the publication of this manuscript.

References

Bahar, D. M., Çimen, M. A., & Tuncay, R. N. (2009, November). Development of control strategy based on fuzzy logic control for a parallel hybrid vehicle. In 2009 International Conference on Electrical and Electronics Engineering-ELECO 2009 (pp. II-342). IEEE.

Bin, Y., Li, Y., Gong, Q., & Peng, Z. R. (2009, June). Multi-information integrated trip specific optimal power management for plug-in hybrid electric vehicles. In 2009 American Control Conference (pp. 4607-4612). IEEE.

Boyali, A., & Güvenç, L. (2010, October). Real-time controller design for a parallel hybrid electric vehicle using neuro-dynamic programming method. In 2010 IEEE International Conference on Systems, Man and Cybernetics (pp. 4318-4324). IEEE.

De Silva, C. W. (2018). Intelligent control: Fuzzy logic applications. CRC press.

EIA, (2018). Administration USEI (2018) Monthly Energy Review. URL: https://www.eia.gov/totalenergy/data/monthly/

Kilic, V., Ozatay, E., Guvenc, L., Göktan, A., & Yılıdırım, M. (2007, May). Ford Otosan Hybrid Electric Vehicle – FOHEV. In 2nd European Ele-Drive Transportation Conference EET-2007At: Brussels, BELGIUM

Kim, N., Cha, S., & Peng, H. (2010). Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle. IEEE Transactions on control systems technology, 19(5), 1279-1287.

Majdi, L., Ghaffari, A., & Fatehi, N. (2009, December). Control strategy in hybrid electric vehicle using fuzzy logic controller. In 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 842-847). IEEE.

Mathworks. (2019). SIL and PIL Simulations. Mathworks. https://www.mathworks.com/help/ecoder/ug/about-sil-and-pil-simulations.html

Meng, X., & Langlois, N. (2010, August). Optimized fuzzy logic control strategy of hybrid vehicles using ADVISOR. In 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering (Vol. 4, pp. 444-447). IEEE.

Naderi, P., Farhadi, A., & Bathaee, S. M. T. (2009). Forward Simulation of a Parallel Hybrid Vehicle and Fuzzy Controller Design for Driving/Regenerative Propose. International Journal of Electrical and Computer Engineering, 4(5), 332-339.

Passino, K. M., & Yurkovich, S. (1997). Fuzzy Control Addison-Wesley. Reading, PA, USA.

Slezak, L. (2004). Success story: ADVISOR Simulation Tool for Vehicle Evaluation and Testing. U.S. Department of Energy FreedomCAR and Vehicle Technologies Program. EERE Information Center. URL: https://www1.eere.energy.gov/vehiclesandfuels/pdfs/success/advisor_simulation_tool.pdf

Triplett, T., Santos, R., & Rosenbloom, S. (2015). American driving survey: Methodology and year one results, May 2013–May 2014.

Wipke, K. B., & Cuddy, M. R. (1996). Using an advanced vehicle simulator (ADVISOR) to guide hybrid vehicle propulsion system development (No. NREL/TP-425-21615; CONF-9609227-2). National Renewable Energy Lab., Golden, CO (United States).

Xu, B., Li, M., Yang, S., Guo, B., & Cui, H. (2010, October). Design and simulation of fuzzy control strategy for parallel hybrid electric vehicle. In 2010 International Conference on Intelligent System Design and Engineering Application (Vol. 1, pp. 539-543). IEEE.

Zhu, C., & Yang, B. (2012, March). Energy management in parallel hybrid vehicle using fuzzy control. In 2012 IEEE International Conference on Information Science and Technology (pp. 62-67). IEEE.