REGENERATIVE BRAKING SYSTEM FOR THROUGH-THE ROAD PARALLEL HYDRAULIC HYBRID VEHICLE

By

DENESH KUMAR A/L SIVA KUMAR

(Matrix no: 137804)

Supervisor:

Dr. Muhammad Iftishah Bin Ramdan

12th July 2021

This dissertation is submitted to.

Universiti Sains Malaysia

As partial fulfillment of the requirement to graduate with an honors degree in

BACHELOR OF ENGINEERING (MECHANICAL ENGINEERING)

School of Mechanical Engineering

Engineering Campus

Universiti Sains Malaysia

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed……………………………………………… (Denesh Kumar A/L Siva Kumar)

Date………………………………………………………………. (12/07/2021)

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated.

Other sources are acknowledged by giving explicit references.

Bibliography/references are appended.

Signed……………………………………………… (Denesh Kumar A/L Siva Kumar)

Date………………………………………………………………. (12/07/2021)

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for interlibrary loan, and for the title and summary to be made available outside organizations.

Signed……………………………………………… (Denesh Kumar A/L Siva Kumar)

Date………………………………………………………………. (12/07/2021)

ACKNOWLEDGEMENT

First, thanks to God almighty for the blessings and courage gave to complete this research successfully despite going through this tough pandemic period. I acknowledge the assistance of my supervisor, Dr. Muhammad Iftishah Bin Ramdan, with great gratitude. His consistent encouragement and comments have aided me in completing this project to the best of my abilities. Next, I would like to appreciate the School of Mechanical Engineering at Universiti Sains Malaysia for providing me with the necessary equipment and supplies to finish my Final Year Project (FYP). Grateful for the chance to complete this four-year study in Mechanical Engineering, and I am confident that all the knowledge and technical abilities I have gained will be extremely useful in my future work as an engineer. The next gratitude goes to En. Mohd Zalmi Yop was always willing to help in drilling holes and lending a hand to fix the experiment equipment's on the vehicle. Besides that, my acknowledgment is to Mr. Tan Pe Hao postgraduate student who was always there experimenting together with me throughout the semester, without his guidance, the experiment would not have gone this far.

Furthermore, my greatest gratitude goes to my family, friends, and people who were indirectly involved in this research, they have kept me going on this quest. They have stood with me through the highs and lows of this undertaking, as well as life in general. . Without their support I would have not been the person I am today, thank you all for the support.

Table of Contents

LIST OF FIGURES:

LIST OF TABLES:

LIST OF ABBREVIATIONS:

DP	Dynamic programming
HEV	Hybrid electric vehicle
HHV	Hydraulic Hybrid vehicle
HPM	High-pressure motor
ICE	Internal combustion engine
SOC	State of charge
TTR	Through-the road
VSS	Vehicle speed sensor
LCD	Liquid crystal Display

ABSTRAK

Seni bina selari hibrid hidraulik boleh menjadi kunci untuk mengatasi kadar pencapaian kenderaan hibrid yang cekap. Reka bentuknya adalah konfigurasi melalui jalan(through-the road) dengan seni bina selari kenderaan Hidraulik Hidraulik. Sistem strategi kawalan berasaskan peraturan Arduino untuk sistem brek regeneratif hibrid hidraulik melalui jalan pada kenderaan prototaip dikembangkan untuk meminimumkan penggunaan bahan bakar kenderaan. Pengaturcaraan dinamis digunakan untuk mengekstrak grafik dan data dari siklus pemanduan kenderaan sebagai rujukan untuk membangun strategi pengendalian berdasarkan peraturan. Berdasarkan hasil yang diperoleh strategi pengendalian berasaskan peraturan dikembangkan menggunakan mikrokontroler Arduino. Keadaan terkawal membolehkan pam sistem brek regeneratif berfungsi mengikut kemampuannya dan juga memaksimumkan penggunaan brek regeneratif sepanjang kitaran pemacu.

ABSTRACT

The hydraulic hybrid parallel architecture could be the key to tackle the low attainability rate of efficient hybrid vehicles. The design is a through-the-road configuration under the parallel architecture of a Hydraulic Hybrid vehicle. An Arduino rule-based control strategy system for the through-the-road hydraulic hybrid regenerative braking system on the prototype vehicle was developed to minimize vehicle fuel consumption. The dynamic programming was used to extract graphs and data from the vehicle's drive cycle as the reference to build the rule-based control strategy. Based on the results obtained the rule-based control strategy was developed using the Arduino microcontroller. The controlled conditions allow the pump of the regenerative braking system to function within its capabilities while also maximizing the usage of regenerative braking throughout the whole drive cycle.

CHAPTER 1: INTRODUCTION

1.1 Introduction to Hybrid hydraulic vehicle

The development of the automotive vehicle industry has focused on hybrid vehicles for the betterment of the environment. Due to concerns about excessive fossil fuel consumption and the impact of carbon emissions on the environment, hybrid vehicle as proven to reduce energy consumption and carbon emission, (Z. Wang, Jiao, Pu, & Han, 2018). By alternating the power train with the help of electrical or hydraulics energy is classified as a hybrid system. In addition, this hybrid technology allows the vehicle to more fuel-saving and environment friendly than conventional cars, many of which, such as the Toyota Prius, are almost silent during full-electric drivetrain mode. Engine-idle technology, energy regeneration, and other technologies make hybrid vehicles more fuel-efficient, with mileages of twenty to thirty percent greater than conventional automobiles. (Adithya Jayakumar, Rizzoni, & Yingbin Liang Abhishek Gupta Tunc Aldemir, 2018) thereby allowing us to save more of those resources.

Improving the fuel efficiency of vehicles is a promising way of addressing the shortage of energy generation problems of the environment. Electric and hybrid-electric drive systems are extensively employed in passenger automobiles and medium-duty trucks. (Pugi, Pagliai, Nocentini, Lutzemberger, & Pretto, 2017). The power requirements are matched with the electric motor and larger battery, which considerably increases the vehicle's cost and heft. With its great power density and cheap cost, the hydraulic hybrid drive system has gotten a lot of attention in recent decades.

The hydraulic hybrid parallel architecture could be the key to tackle the low attainability rate of an efficient hybrid vehicle. (Bravo, De Negri, & Oliveira, 2018). This parallel hydraulic hybrid through-the-road system can be retrofitted from conventional vehicles. This build architecture has a regenerative braking system fix into existing vehicles. The hydraulic system is very durable and reliable, due to its simplicity, it is sturdy and can be maintained easily by a specialized mechanic.

Regenerative braking is an important hybrid vehicle technology that has sparked a lot of research attention. (Liu, Zheng, Su, & Zhao, 2013). The development of a control strategy is one of the most important subjects in regenerative braking to construct the parallel HHV through-the-road vehicle(TTR) in the most efficient process, and it may be divided into two groups based on the research goal. The main goal is to increase regeneration efficiency and energy resource conservation. Another is to improve braking performance and comfort while driving. Furthermore, the hydraulic hybrid propulsion system is the optimum option for regenerative braking, particularly for operations with a high stop-and-go frequency. (Wu, Hu, Yuan, & Di, 2016).

This study contributes to optimizing the through-the-road parallel HHV's regenerative braking. An Arduino rule-based control strategy is developed to increase the conservation of energy using the hydraulic pump. Hence, regenerative braking of the system would be more efficient in charging the accumulator.

1.2 Objective

To develop an Arduino rule-based control strategy system for a through-the-road hydraulic hybrid regenerative braking system on the prototype vehicle and explore the effectiveness of the regenerative braking on a parallel hybrid hydraulic architecture. The architecture used in the system is parallel HHV on through-the-road (TTR) vehicles. To obtain an efficient method to convert the kinetic energy from the brakes to hydraulic potential energy and reuse the energy for the acceleration of the vehicle.

1.3 Project background

Developing countries like Malaysia has comparably low sales for hybrid electric vehicles. This is due to the low-income consumers limiting the ability for maintenance and even to purchase because most of the hybrid versions are above the range of RM80,000. ("Know Before You Buy," n.d.). This hybrid vehicle is not be fully dependent on one source power supply to drive the vehicle in case of emergency. The main concept of the hybrid vehicle using regenerative braking is to recover the kinetic energy from the braking, which can be reused later to propel the vehicle. A hybrid hydraulic Vehicle (HHV) offers high power-density components for high-frequency regenerative braking, allowing the vehicle to recover large amounts of energy in a short period.

The vehicle during the braking process, the hydraulic pump is powered by momentum to charge the hydraulic accumulator where the high-pressure fluid is stored as potential energy the accumulated energy during acceleration is used to drive a hydraulic motor by releasing the high-pressure fluid in the accumulator. The parallel architecture for the HHV is the simplest because it is like a conventional drive train together equipped with the hydraulic hybrid component that allows regenerative braking. The parallel architecture of the through-the-road discussed in this project further simplifies the parallel architecture since it does not conflict with the traditional front-wheel drive train. The through-road parallel HHV components are installed in the Perodua Myvi passenger vehicle. The braking system when the brake is engaged and the pump is engaged to obtain the highest efficiency for regenerative braking system will be conducted. (Ramdan et al., 2020) The vehicle could be powered individually by the engine power train or in combination by the engine and the high-pressure motor(HPM). When the HPM operates, the valve is opened and the HPM is engaged. During vehicle launch, the HPM operates, and fluid flows from the high-pressure accumulator to the low-pressure accumulator. The high-pressure accumulator pressure lowers when hydraulic energy help in the acceleration of the vehicle. At the HPM, the low-pressure accumulator is compressed into the high-pressure accumulator as a pump for regenerative braking fluid. With the rise in accumulator pressure, the braking energy is recovered and deposited in the high-pressure accumulator. The current system can run this hydraulic system manually with the control of a switch to activate the pump for the regenerative braking to occur.

1.3.1 Problem Statement

The HHV architectures can be classified into parallel, series, and power split. The parallel architecture is the simplest of all three because it is like that of the conventional drive train but is equipped with additional hydraulic hybrid components that allow regenerative braking. The main idea of the project is to incorporate the dynamic programming results into a rule-based control strategy. To develop an

Arduino-controlled rule-based system to activate the hydraulic system. When the right conditions are achieved, the HHV system will function at the highest efficiency.

1.3.2 Scope of work

The scope of work is to convert the Perodua Myvi conventional vehicle in the lab into parallel hybrid hydraulic on the rear wheel with the regenerative braking system and accumulator attached on it, the main scope of work is to develop a rule-based control strategy on Arduino microcontroller, to control the solenoid valves of the pump to compress the hydraulic fluid during regenerative braking. Depending on the vehicle speed and the state of charge (SOC) in the accumulator the pump would function at the highest efficiency. The system would be able to convert the kinetic energy into potential energy to be stored and used again when needed in acceleration.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction to Hydraulic hybrid vehicle(HHV) system

The basic concept of the HHV is it utilizes a pressurized fluid power source, combined with a traditional internal combustion engine (ICE), to obtain improved fuel efficiency and reductions in hazardous emissions. Hydraulic pumps and motors make up the hydraulic hybrid architecture, the recovered energy from the braking event is stored in an accumulator as pressurized fluid. This energy is then utilized to accelerate the vehicle by driving hydraulic motors (Barbosa, da Silva, Pujatti, & Gutiérrez, 2020). The hybrid architecture can be classified into parallel, series, and power split. In the parallel architecture, the hydraulic pump/motors are connected in between the transmission and engine to the drive shaft. While the power-split architecture, combines both series and parallel architectures, as with the parallel design, the mechanical drive system is conventional, and series hydraulic pumps and motors are also connected to the drive wheel shafts. (Pakdelbonab, Kazerooni, Payganeh, & Esfahanian, 2021). The parallel architecture is the most straightforward of the three because it is identical to a normal drive train but has additional hydraulic hybrid components that allow for regenerative braking. These hydraulic systems are used in heavy vehicles such as refuse trucks and transportation sector vehicles like bus and delivery vehicles, but for through the road vehicles, only a few research have proposed hybrid hydraulic architectures. The TTR parallel architecture allows for the installing of regenerative braking components into an existing conventional car, a through-the-road hydraulic hybrid parallel design could be the answer to the low attainability rate of efficient hybrid vehicles. (Ramdan et al., 2020).

2.2 Implement the regenerative braking system on HHV.

The basic concept of the hybrid vehicle is the regenerative braking system that recovers kinetic energy during braking, which can then be used to move the vehicle. The control strategy for regenerative braking is important to improve braking efficiency and to increase regeneration efficiency while conserving resources. (Liu et al., 2013). The regenerative braking system in the HHV consist of the low-pressure reservoir, high-pressure accumulator and hydraulic pump, and the motor unit, it converts the kinetic energy from the wheel during the braking of the vehicle to compress the hydraulic fluid to be stored as high-pressure fluid in the accumulator. This stored pressurized fluid is then used by the hydraulic pump and the motor to generate torque during the following vehicle launch and acceleration.

The position of this system plays an important role in determining efficiency, by fixing the regenerative braking force on the rear wheel axle of the vehicle, the rear wheels should be locked after the front wheels to ensure vehicle stability while braking. (Liu et al., 2013). Fitting the system to the rear axle would be the most efficient method, the design of the hydraulic accumulator plays an important role in the preservation and conservation of the energy, which is very critical for the HHV system. (Podrug, 2019). The hydraulic regenerative braking system's braking force should be consistent and effective. The exchange of high and low pressures occurs in hydraulic regenerative braking mode. The hydraulic motor's speed and the hydraulic system efficiency are inextricably linked. (Wu, Liu, Zhou, Hu, & Yuan, 2019).

Comparing the HHV system to the hybrid electric vehicle (HEHV) has higher regenerative braking efficiency due to higher energy density. (Yang, Luo, & Li, 2017). The hydraulic system is used to absorb the high-power braking energy and release the stored energy during start-up or acceleration. Therefore, the energy conservation system in HHV is more efficient than HEV in a parallel hybrid structure because able to release high power of torque to the motor during acceleration in a short time (Chen, 2015). The hydraulic hybrid propulsion system's high power density is an appealing feature for regenerative braking in a vehicle that stops frequently. (Wu et al., 2016). High power density, long cycle life, and frequent charging and discharging capabilities are all advantages of hydraulic energy storage devices. A good control approach can also be utilized to retrieve the braking energy that has been generated. (Z. Wang et al., 2018).

The HHV can conserve about 80% of the vehicle's kinetic energy and convert it into productive work for the same amount of energy. The reason for the difference between HHV and conventional vehicle inefficiency is due to regenerative braking. (Podrug, 2019). In a hybrid car, how soon the system can conserve energy is critical, at HHV the regenerative charging is much faster. Depending on the pressure differences, the hydraulic regenerative might have a very wide working range but not enough for the necessary brake torque. (B. Wang, Huang, Wang, Guo, & Zhu, 2015).

2.3 Usage of dynamic programming

The rule-based energy management approach is used for real-time control of a new hydraulic hybrid vehicle. As an optimization condition, the dynamic programming optimization examines the system state not only at the previous time but also at the upcoming moment. This dynamic programming approach is a good way to develop the optimal control parameters for a rule-based energy management strategy. (Zhou, Xu, Liu, Liu, & Zhang, 2018). The result from the dynamic programming will be used to build the rule-based strategy and develop an efficient HHV system is obtained. The dynamic programming method, which is based on the drive cycle concept is capable of solving this global optimal control problem effectively. The objective function and the constraint conditions of brake energy recovery are important aspects of the dynamic programming algorithm. (Zhang & Chen, 2014). Dynamic programming is a global best solution that may be used to evaluate the effectiveness of other energy management solutions. It is worthwhile to look at the world's best solution. (Zhou et al., 2018).

CHAPTER 3: METHODOLOGY

3.1 Introduction to experiment

The parallel Hydraulic Hybrid Vehicle(HHV) system for this experiment consists of the high-pressure accumulator, low-pressure hydraulic tank, motor to deliver the compressed hydraulic fluid, the pump which would transfer the brake kinetic energy to the compressed hydraulic potential energy, 3/2-way throttle solenoid valve, solenoid check valve, pressure sensor, and pressure gauge Table 3.1 shows the function of these components. Using this equipment, a hydraulic hybrid system was designed as in Figure 3.1, in the engine lab attached to an axle which is mounted on the table.

Figure 3.1 Initial setup of the hydraulic system

Table 3. 1: List of components

List of components	Function
High-pressure accumulator	To store the pressurized hydraulic fluid from the pump and supply high-pressure fluid to the motor
Low-pressure tank	Store the hydraulic fluid and supply it to the system through the pump.
Hydraulic Motor	To accelerate the wheel using the high-pressure fluid from the accumulator
Hydraulic Pump	To pump the fluid from the low-pressure tank when the brake is pressed.
3/2 throttle solenoid valve	To control the flow of the fluid from the pump, to send it forward to the pressure sensor, or to send it back to the low-pressure tank
Pressure sensor and pressure gauge	To observe and monitor the pressure going into the high-pressure accumulator
Solenoid check valve	To control the hydraulic fluid going through the pressure gauge

This system was then fitted onto the through-the-road vehicle Perodua Myvi. An Arduino microcontroller was added into the hydraulic system to control the flow of the hydraulic fluid through the motor and pump by controlling the solenoids that control the opening and closing of the motor and pump valves. The drive cycle was gained through the experiment and then the dynamic programming MATLAB was used to gain operating torque, gear ratio, state of charge(SOC), and the fuel efficiency map from the developed fuel consumption scenario. The process flow for this project was obtaining the SOC charge/discharge situation based on the vehicle speed, the pump/motor displacement, and the engine torque and speed resulting from the MATLAB dynamic programming, then applying the rule-based control strategy into the Arduino microcontroller then finally connecting the Arduino to the hybrid hydraulic system.

3.2 Dynamic programming

Dynamic programming was used to find the optimal trajectory of the control input like activation of pump and activation motor depending on SOC, vehicle speed, and brake command from driver to construct a control strategy for through the road parallel HHV. One of the numerical optimization strategies is dynamic programming, it is used to simulate the SOC, pump and motor displacement, and vehicle fuel consumption from the drive cycle. These algorithms are applied for the optimization of the energy conservation process of the HHV system. To construct the dynamic programming (DP) there a flow is created as in Figure 3.2.

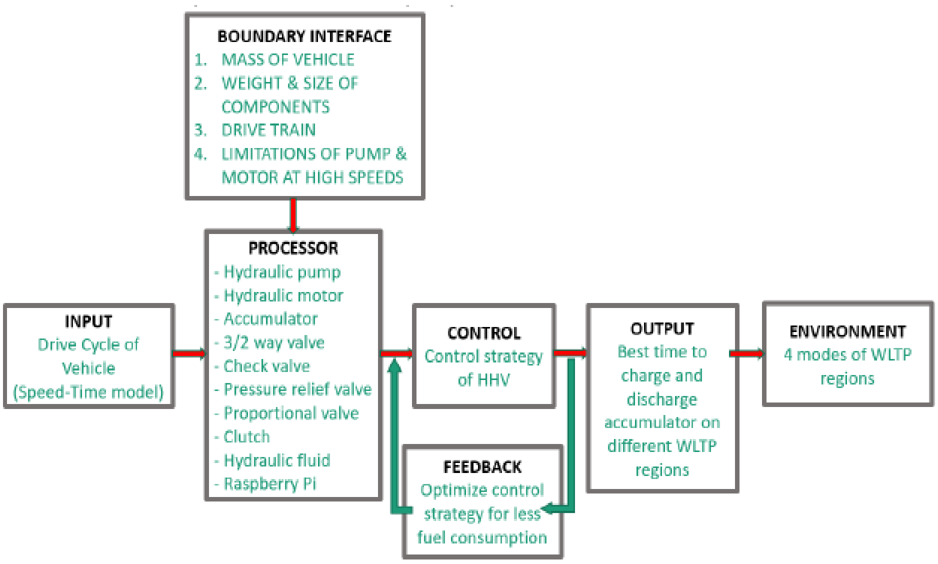

Figure 3.2: Process chart for the dynamic programming

Figure 3.2 was obtained from the capstone project group MIR1, it explains the flow chart of the control strategy of the dynamic programming where the drive cycle is the input. The data and codes were obtained based on previous research with slight editing from the supervisor of this project. Figure 3.3 shows the environment of MATLAB used to extract the data together with the codes and data variable set on the right obtained from Dr. Muhammad Iftishah Bin Ramdan. The result from the programming is the vehicle speed, state of charge (SOC), displacement of motor and pump, engine torque against time, engine speed against time, engine efficiency against time, the gear index, the wheel torque, hybrid on/off time, the vehicle engine fuel consumption map and the engine efficiency map graphs. Based on this optimal solution from the dynamic programming, the rule-based control is constructed.

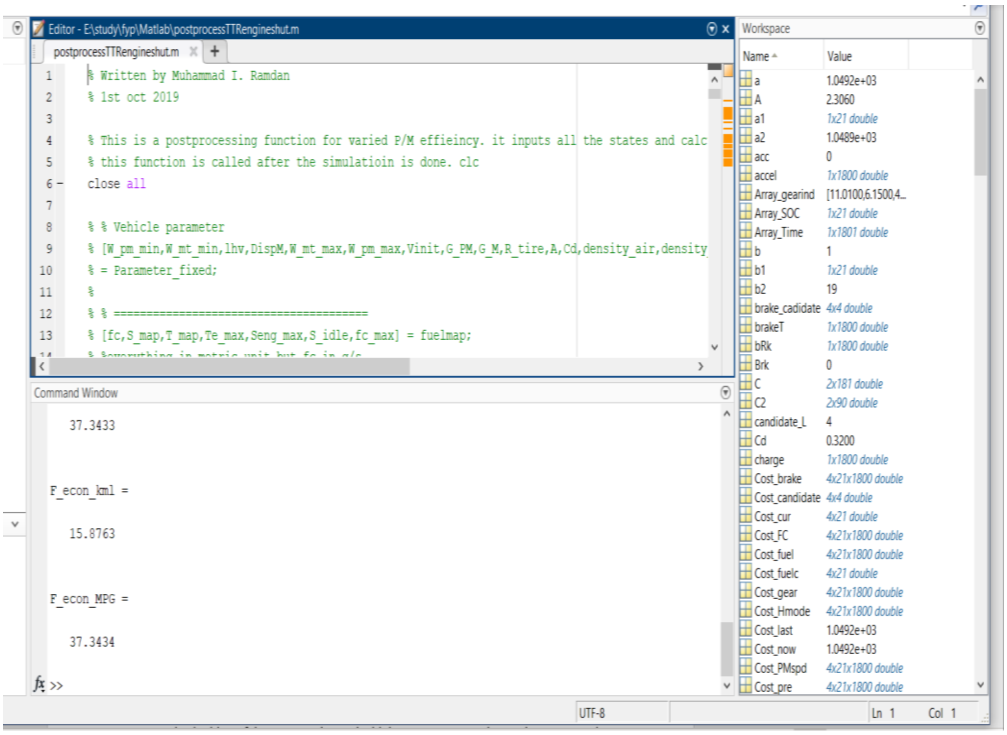

Figure 3.3: MATLAB environment

3.3 Arduino control system

The microcontroller used in this experiment was Arduino to control the pump valve for the regenerative braking of the HHV to charge the high-pressure accumulator. There are certain conditions for the Arduino to control the regenerative braking system, the inputs that are needed for the system are the brake pedal pressed, pressure signal input, and the vehicle speed sensor(VSS). Whenever the brake pedal is pressed and the other two input vehicle speeds and SOC are matched, the system pump would pressurize the hydraulic fluid from the low-pressure tank to the accumulator. Based on the dynamic programming of the results obtained, the pump should be activated when the vehicle speed is 72 km/h and above, the regenerative braking on the pump would not work therefore no charging will occur, only the mechanical brakes would work. From the result of MATLAB, the regenerative braking system would also not work when the SOC in the high-pressure accumulator is at 100% full charge. The pressure sensor will be used to obtain the pressure reading an analog signal to be read by the Arduino. The circuit diagram is shown in Figure 3.4, the connection of the Arduino is based on this diagram. The pressure sensor, push button, push switch, the car brake light, the vehicle speed sensor, and the LCD screen are set up as below. The table explains the list and functions of the sensors and electronic components involved.

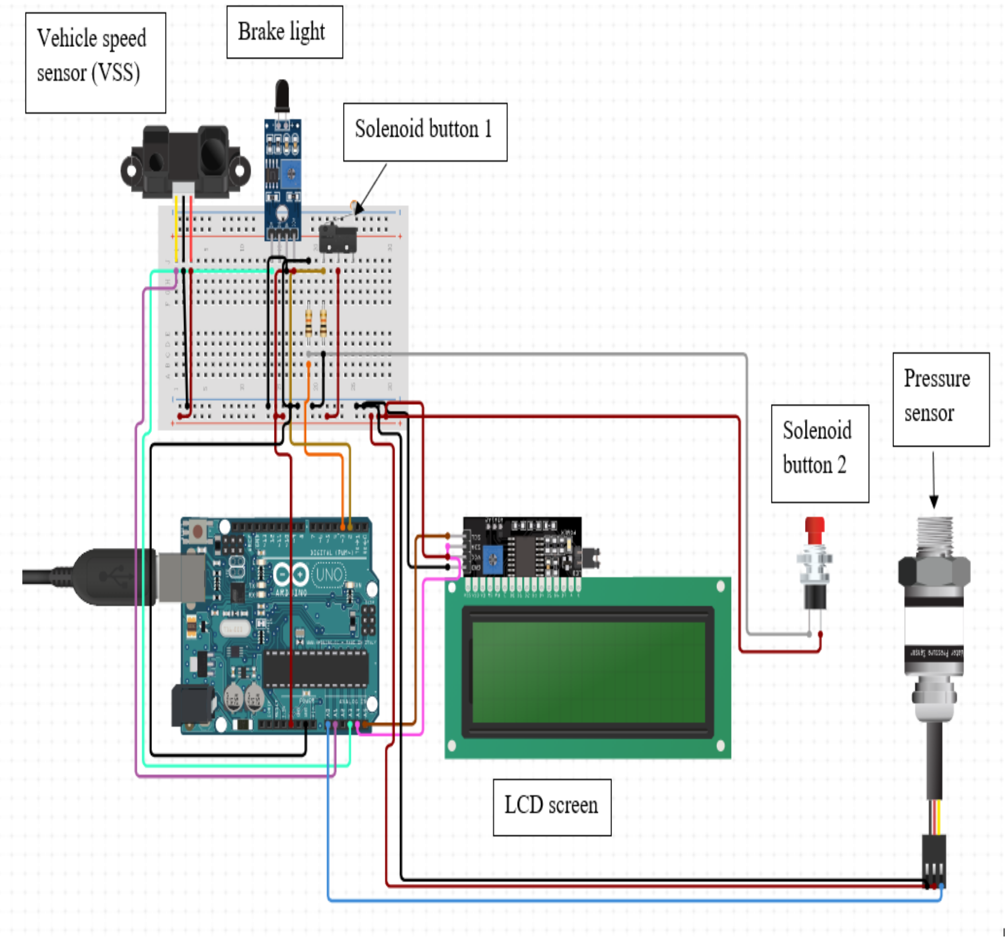

Figure 3.4: Circuit diagram of the connection in Arduino

Table 3. 2: List of electronic components

	Electronic components	Function
1	Vehicle speed sensor (VSS)	Detects the vehicle speed from the speedometer
2	Brake light	To detect the brake pedal signal
3	Solenoid button 1	To control the solenoid valve 1
4	LCD screen	To display the pressure value and SOC
5	Solenoid button 2	To control the solenoid valve 2
6	Pressure sensor	To detect the pressure in the system
7	Arduino UNO	Controls the input and output of the circuit based on the conditions

Therefore, the Arduino would be connecting with inputs from the vehicle speed sensor (VSS), the pressure sensor, and the brake pedal signal extract from the brake light. The output for the Arduino would be the control of the solenoid valve to allow the hydraulic flow from the pump to the high-pressure accumulator and hold the charge from releasing to the motor from the high-pressure accumulator. Hence the coding is divided into two parts, the pressure sensor code, and the conditions of when the pump should charge the accumulator.

3.3.1 Pressure sensor

The pressure sensor is connected to the controller that is connected to the LCD screen to display the pressure of the hydraulic system. The sensor used in this project is EDYBT-01 Oil Pressure Transducer, which can measure up to 100bar with an output signal voltage of 0V – 5V. Therefore, the code would read the sensor's input pressure in Bar and then convert it into kilopascal (kPa). The pressure sensor is tested with a digital multi-meter, the output from the signal is at below 5V. Hence the pressure sensor signal wire is connected directly to the Arduino pin A1. Next, the maximum and minimum pressure signal value is calculated using the range of the sensor. At the 100bar the sensor gives out 5V to the Arduino which has an analog value of 1024. The minimum would be 0 bar at 0.5V which gives an analog value of 102.4. The code starts with declaring the pressure maximum and lowest pressure, from this detail the input signal from the pressure sensor coming into the Arduino will be used to calculate the final pressure value in kilopascal (kPa). The pressure value from the sensor signal is calculated by:

$$\mathrm{pressureValue} = ((\mathrm{pressureValue} - \mathrm{pressureZero}) * (\frac{\mathrm{pressuretransducermaxBAR}}{\mathrm{pressureMax} - \mathrm{pressureZero}}) \qquad (1)$$

Where the pressureZero is the minimum pressure, pressuretransducermaxBAR is the highest pressure that can be measured by the pressure sensor which is 100bar that would be declared in the code, pressureMax is maximum pressure declared in the code.

Based on the pressure value and the maximum pressure, the state of charge (SOC) of the accumulator is calculated based on the equation. Pressure limit is the maximum pressure in kPa. The SOC is calculated using:

$$Soc = \left(\left(\frac{pressurekpa - pressurekpamin}{pressurekpamax - pressurekpamin}\right) * 100\right) \qquad (2)$$

Where the pressurekpa is the pressure value converted from bar to kilopascal (kPa), pressurekpamin is the pressure minimum in kilopascal (kPa) and the pressuremaxkpa is the maximum pressure declared in kilopascal (kPa). Figure 3.5 shows the flow of the pressure sensor coding.

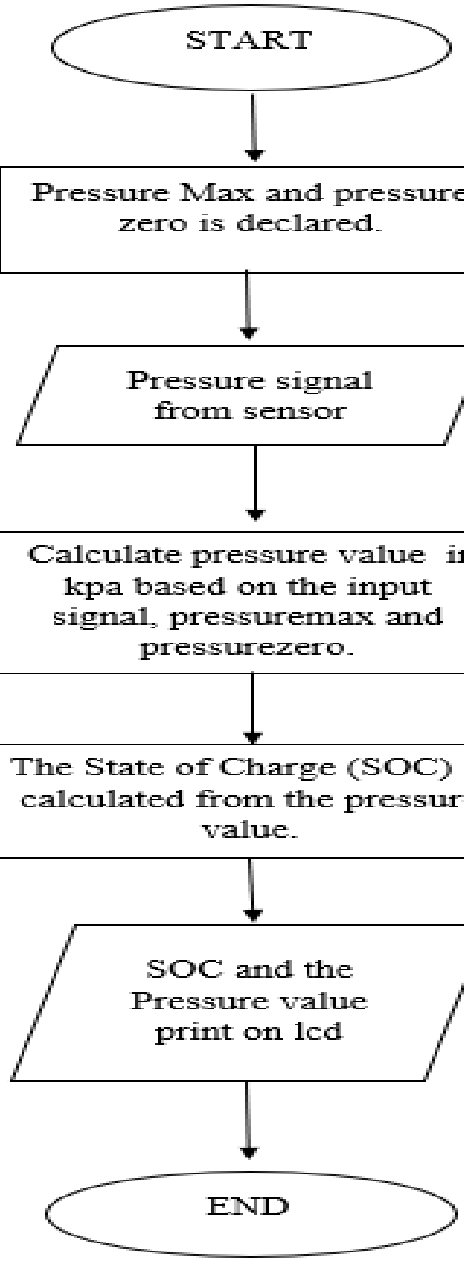

Figure 3.5: Flow chart of the pressure sensor coding

3.3.2 Compiled code with conditions

The code starts with declaring the input variables which are the pressure sensor's maximum and minimum pressure to calculate the SOC, the vehicle speed sensor (VSS) wave frequency with its calibration variables, and the brake light. The codes include the acceleration part of the HHV system will focus on the regenerative braking part of the system for this paper. The code for extracting the brake pedal signal is from the brake light from the rear of the vehicle. The analog signal from the brake light is at 12V when the brake is pedal was pressed, the Arduino microcontroller would only read a signal maximum of 5V analog at the reading of 1023, therefore a breadboard with two resistors of 1.8k Ω and 1.3k Ω are connected to lower the voltage to 5V so the voltage value can be read by the Arduino. The brake idles off condition the reading of analog will be at 0 and when the pedal is pressed the reading would be 1023. Therefore, the brake pedal conditions would be reading the analog pin and have two conditions if the analog has read more than 1000 when the brake pedal is pressed, and the analog read below 1000 when no brake is applied. The 1000 value is used instead of 1023 because when testing out the signal from the Arduino, the average reading of the signal during the brake pedal pressed was at 1000 this might be due to the voltage drop by using the resistors.

The vehicle speed sensor (VSS) from the car speedometer gives out pulse wave, the pulse is high and low, the frequency of the pulse is calculated:

$$frequency = \frac{1000000}{pulseTotal} \tag{3}$$

$$Vehiclespeed\left(\frac{km}{h}\right) = frequency * 0.75 \tag{4}$$

The pulse high and pulse low are added to obtain the pulse total in microseconds. The frequency is obtained by dividing the 10^{-6}s by the total pulse. Then the vehicle speed is obtained from the frequency, the vehicle speed output from the Arduino is compared and calibrated with the vehicle's speedometer, the constant c is added to calibrate the result to match the speedometer's km/h:

$$c = w * r \tag{5}$$

Where the c is the constant for calibration, r would be the radius of the wheel in (inch) and w is the reading from the frequency of the vehicle speed sensor(VSS). The value of c obtained was 0.75 which was then multiplied with the frequency to obtain the vehicle speed.

The following flow chart in Figure 3.6 shows that the input would be the VSS, SOC, and brake light. The code would then read the brake pedal input signal first, if no brake pedal is detected the solenoid button 1 and 2 solenoid would let the hydraulic fluid flow from the motor valve to the low-pressure tank as there is no regenerative charging occur, if the brake light is detected the code would continue to read the SOC charge value when the SOC is not at 100%, the code would read the speed of the vehicle from VSS if the speed is below 72 km/h, the system would activate the solenoid 1 which opens the solenoid valve of the pump to compress the hydraulic fluid and for the charge to hold in the accumulator and do not flow out to the motor another solenoid 2 valve is activated.

When the condition is not met, solenoid 1 will not be activated resulting in the solenoid valve will direct the hydraulic fluid from the motor to the low-pressure tank reservoir. The SOC is printed on the LCD screen which is attached to the Arduino microcontroller.

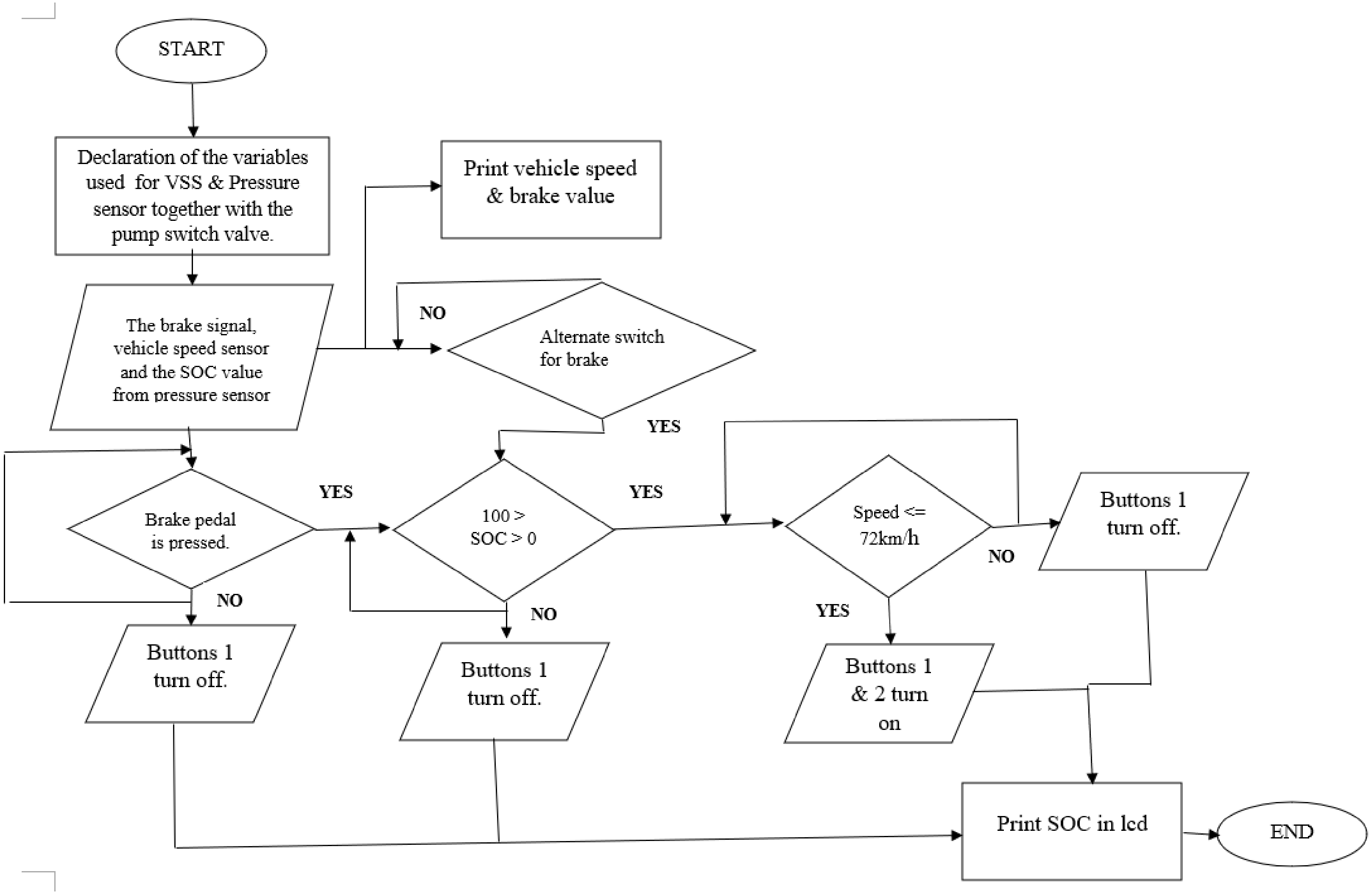

Figure 3.6: Flowchart of the overall code

3.4 Project setup

The hydraulic hybrid system consists of the low-pressure tank, pump, pressure gauge, 3/2 solenoid valve, solenoid check valve, high-pressure accumulator, and a motor with hydraulics hose connecting the system like in Figure 3.7. The hose connection is hose 1 from the low-pressure tank to the pump, hose 2 from the pump connects the pump to the 3/2 solenoid valve, which has hose 3 connecting from the valve to the low-pressure tank. Hose 4 connects the 3/2 solenoid valve to the pressure gauge, hose 5 connects the pressure gauge to the high-pressure accumulator, hose 6 connects the accumulator to another one-way valve that valve then is connected to hose 7 to the motor, and the final hose 8 connecting the motor to the 3/2 solenoid valve to complete the cycle. As referred to in the schematic diagram in Figure 3.8 with the hose numbers labeled.

Figure 3. 7: Connection of the hydraulic hose

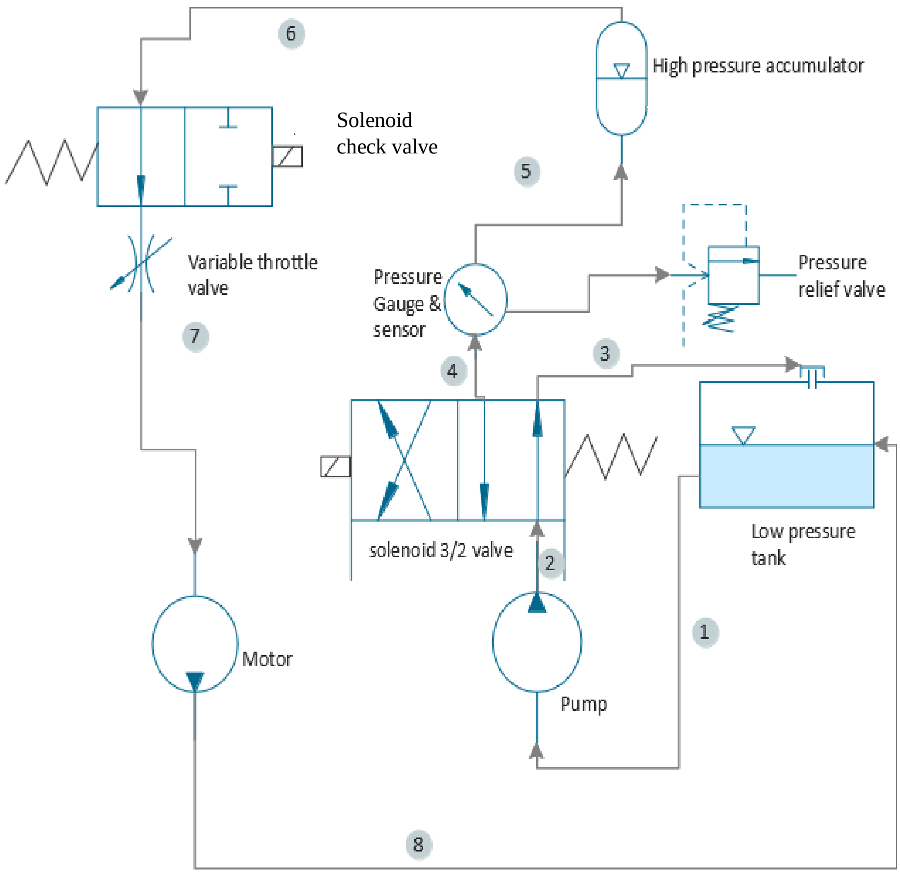

Figure 3. 8: Schematic diagram of the setup

This setup is then fixed on a through-the-road vehicle, Perodua Myvi 1.3 liter engine capacity with a manual transmission shown in Figure 3.9. This would be the prototype vehicle that the hydraulic system will be tested on with the tanks, motor, and pump fitted on the vehicle.

Figure 3. 9: The prototype vehicle

The hybrid hydraulic system with the pump and motor is fixed to the rear axle of the vehicle. The vehicle's axle with the brake system is still maintained, the hydraulic system will be alternate to the brake system. The brake drum conditions of the vehicle on the left and right of the vehicle are checked to ensure they are in good condition.

Figure 3. 10: The system set up in the car.

The rear passenger seat of the vehicle was removed for the setup to be fitted onto the vehicle as shown in Figure 3.10. The tank is then welded on the vehicle body using a metal plate to make sure the tank is stable during driving. Holes were drilled on the car body to enable the hose from the motor and pump on the axle to connect with the tank and accumulator system. The Arduino microcontroller was fitted inside the car with a case to protect the connecting the wires from the inputs and an LCD screen was added to the Arduino system to display out the SOC value without the use of a laptop or pc, for the driver to constantly monitor the SOC value.

CHAPTER 4: RESULT & DISCUSSION

4.1 Results from the dynamic programming

The results obtained from the MATLAB dynamic programming would be essential to create a rule-based control strategy to continue further with the Arduino coding. Based on the obtained drive cycle from the speed vs time data, the vehicle speed is obtained from that the vehicle's speeds in m/s flow are obtained. The state of charge (SOC) in the accumulator graph is recorded with the highest state of 100% and the lowest is 35%. As in Figure 4.1, the motor and pump displacement which is an on/off function. When the pump displacement is high, the system is using the pump to charge the accumulator to deaccelerate the vehicle, causing the SOC to increase at that moment. When the motor displacement is high the system is discharging the accumulator to release the charge to the motor causing the SOC to decrease.

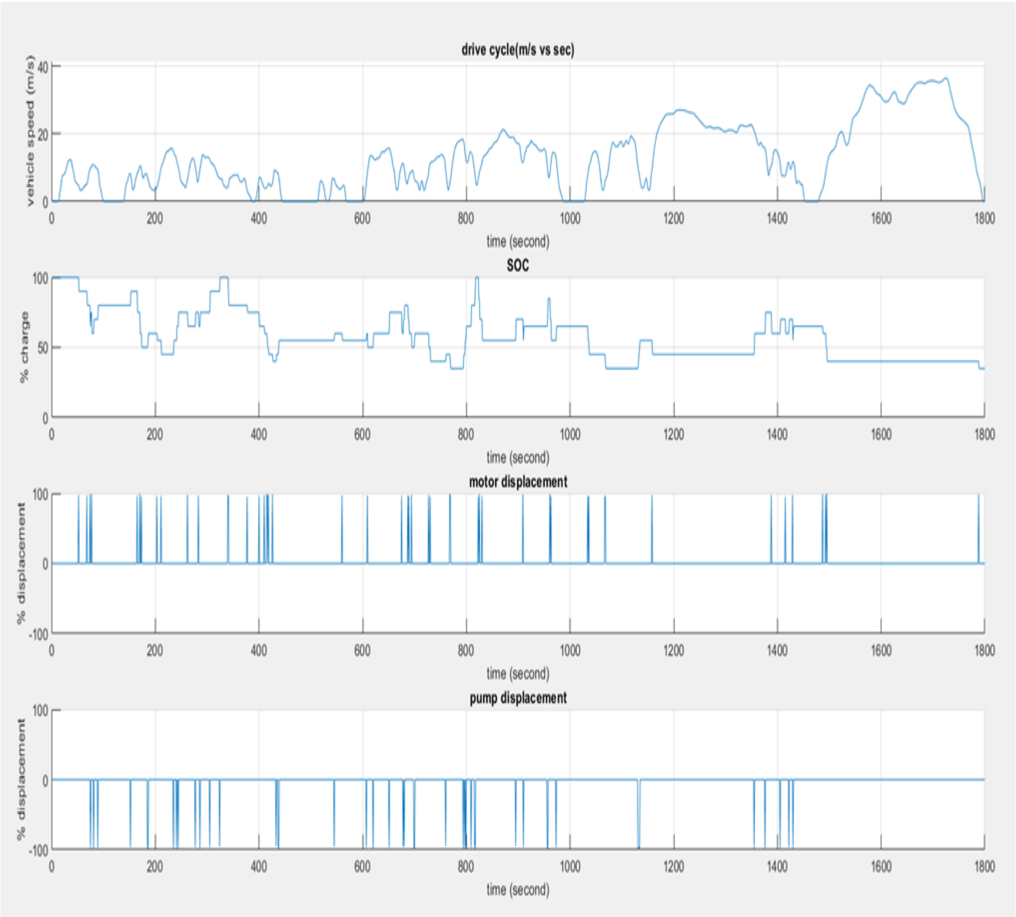

Figure 4. 1: Results from dynamic programming

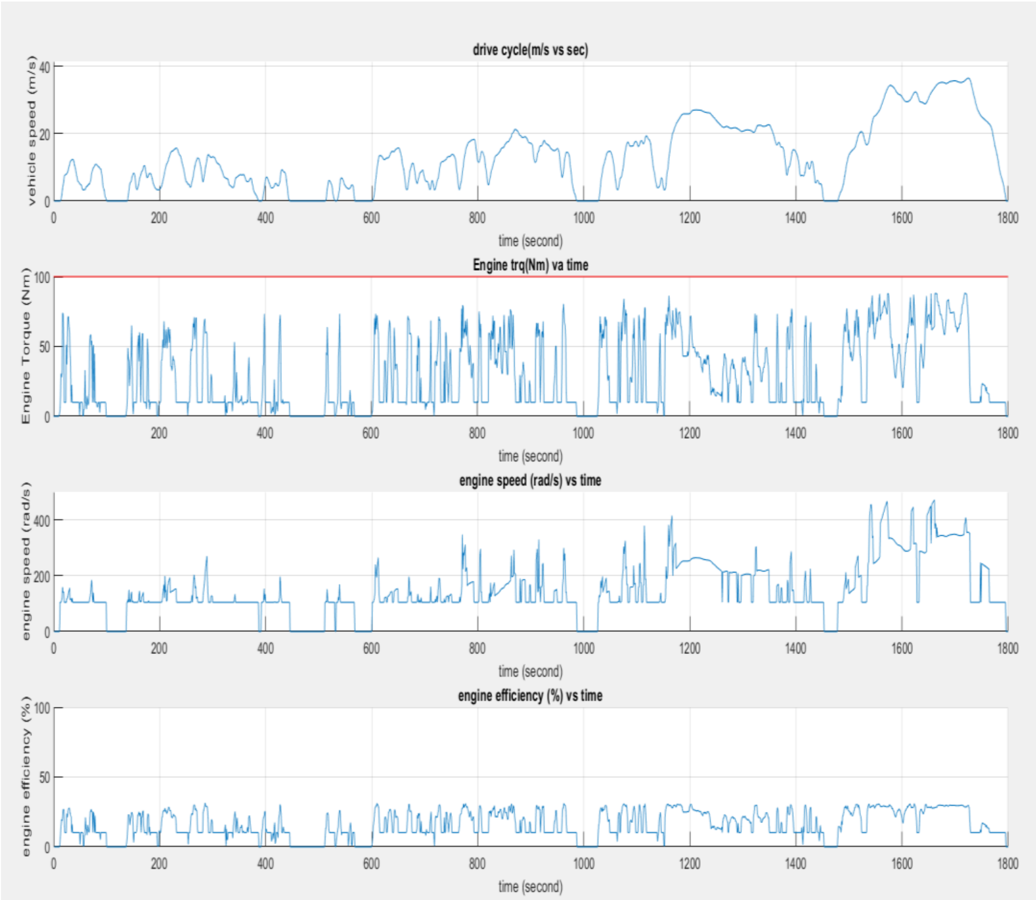

Figure 4. 2: Engine efficiency based on the drive cycle.

The engine speed and torque against time are obtained, the engine torque in the drive cycle has the highest 88Nm of torque and the engine speed maximum was 471.1 rad/s from Figure 4.2. The speed and torque are high during high vehicle speed where the regenerative braking system of the system stops charging the SOC after 70km/h based on figure 4.1. The engine efficiency was obtained based on the speed and torque plot, and it shows an average efficiency of 30% based on the programming.