Smart driving assistance system using Raspberry Pi and actuator networks

Saiful Anwar Che Ghani¹, Maurice Kettner², Ferhat Aslan², Fatimah Dzharuddin³, Norazlianie Sazali³

¹Automotive Engineering Centre, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
²Institut für Kälte-, Klima- und Umwelttechnik, Karlsruhe University of Applied Sciences, Germany
³Department of Mechanical, College of Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia

*Email: anwarcg@ump.edu.my

Abstract. Testing and certification of vehicles in succession to their development are usually achieved on chassis dynamometer testbeds. This paper proposes the implementation of a smart real-time driving assistance system using cloud-computing, embedded tools and motorised actuator for vehicles with manual transmission powertrain under testing. For clutch actuation of the vehicle, a robotic leg actuator connected to a high torque stepper motor is controlled by Python code programmed on Raspberry Pi 4, linked through a smart network for remote controlling of the system via a user interface dashboard and digitalized working system so as to employ data connectivity through cloud-computing. By applying this network structure, the implementation functionality of smart driving assistance system using free cloud infrastructure is investigated, when the hardware is setup in Malaysia, whilst the control is from Germany. The framework that governs this prototype involves different domains of engineering such as data processing, manufacturing, mechanics, networking, embedded systems and actuators is able to be utilised in smart driving assistance system for car tested on a chassis dynamometer with latency of less than 1 s.

1. Introduction

Normally, chassis dynamometers are employed to test and certify of newly developed vehicles when it’s completely assembled, usually for assessing the different aspects of the engine and powertrain performance [1], [2]. Noise emission, exhaust emission, engine performance, fuel consumption and electromagnetic compatibility are among the common parameters investigated in the test trials [3]. Recently, chassis dynamometer testbeds are more preferred than the test runs on the road due to economical factor, ease of operation, reliability, repeatability and the most tempting to the research and industry with the advancement of technology is the possibility for fully automation [4]. Furthermore, chassis dynamometers also allow for transient trials under varied and pre-set environmental conditions when the tests are conducted inside a climate or high-altitude chamber [5]. In addition, the integration of multiples devices such as emission measurement devices, load control and data acquisition is much easier with chassis dynamometer as compared to road tests [6].

Test trials are normally automated by employing a robotic device, which replaces the functions of a driver [4], [7]. The tasks of the robotic device are commonly for operating the brake, controlling the force given to the accelerator and the engagement of the clutch pedal, and to the extent of manipulating the gear lever to switch gears. Normally when in operation, a pre-defined velocity profile
and an associated gear shifting strategy must be pursued for automated test trials on the chassis dynamometer testbed, in which the challenges for control engineering lie in the consideration of actuators as manipulated variables in a closed loop of the vehicle and a driver controller. The driver controller has to be developed to handle different tasks that constitute of several overlying controllers dealing with routines such as drive away, acceleration, gear shifts, deceleration and stopping. The control of the entire remote routine can be addressed by implementing the Internet-of-Things (IoT) approach. This may be beneficial for the development team to access the test facility without the need to be physically present at the testbed. This is important to allow for cross-country collaboration which is crucial to technological innovation which is the main drivers for change in this sector.

This study proposes a conceptual framework to implement a smart real-time driving assistance system using cloud-computing, transport mechanism, embedded tool and motorised actuator for vehicles with manual transmission powertrain under test. The embedded tool is used as client to acquire signals from cloud broker and manipulate the signals to control the movement and speed of the robotic leg to engage and disengage clutch pedal. The test trials on hardware setup in Asia are conducted by exchanging data with data IoT agent in Europe.

2. Methodology

2.1. Message Queuing Telemetry Transport (MQTT) Messaging Protocol

The transport mechanism that is implemented in this work uses the Message Queuing Telemetry Transport (MQTT) messaging protocol which allows the system to reduce the total amount of data sent using a well-known paradigm called publish-subscribe. For a system with restricted equipment, low bandwidth, high latency and reliable networks such as chassis dynamometer in an automotive lab, MQTT protocol is suitable since it is a very simple, lightweight information transmission mechanism [8]. Since MQTT is designed to minimize network bandwidth and equipment resource requirements, while ensuring reliability and a certain degree of delivery guarantee at the same time, the use of MQTT with the driving assistance system that is delay-sensitive and uses little battery power is considered ideal.

The devices that are used to send data can publish them on a specific topic whereas devices or services that are mutually interested in the information can subscribe to the topics of interest. In between these two entities there is another entity called the broker which is responsible for delivering message updates to the subscriber. This framework as depicted in Figure 1, constitutes of components, the GUI Browser Cayenne, as the publisher, local supervisor and Raspberry Pi as the subscribers and the HiveMQ in the cloud as the broker. The communication protocol for the Raspberry Pi that acts as the MQTT-client is written in Python programming language. The Python MQTT client class was implemented which provides the required functions to publish messages and subscribe to topics. Methods from the paho mqtt client class such as connect, disconnect, subscribe, unsubscribe and publish were used within the code.

The Raspberry Pi 4 used in this project needs to subscribe to the topics that are created in the communication protocol to be able to send and receive data, namely target speed, clutch position and send back the On-board Diagnostics (OBD) data such as the actual speed and Revolutions per Minute (RPM). The MQTT signals manipulated by Raspberry Pi will be used to control a stepper motor, which is connected to the clutch pedal, using a circuit shown in Figure 2.

2.2. Robotic Leg

Robotic leg mechanism is driven by a high torque stepper motor. Stepper motor model Nema 34 is used in this project to overcome the torque, \( \tau \) required in pushing the clutch pedal to the maximum point. The measurement and calculation of the torques when the clutch pedal is pushed at different angles are listed in Table 1. \( F_m \) is the measured forces perpendicular to the clutch pedal acquired when the pedal clutch is pushed to the maximum, whereas \( F_x \) and \( F_y \) are two component vectors of forces resolved into two components. Lever angle and pushrod length are the geometrical parameters of the linkages for the kinematics to transfer the force from stepper motor to the clutch pedal imitating the gait of a human driver.
Figure 1. Communication protocol framework.

Figure 2. Schematic diagram of stepper motor circuit controlled by Raspberry Pi 4.

Table 1. Measurement and calculation of the torque, \( \tau \) when the clutch pedal is pushed at different angles.

| Pedal angle (°) | \( F_x \) (N) | \( F_y \) (N) | Lever angle (°) | Pushrod length (mm) | \( F_m \) (N) | \( \tau \) (Nm) |
|-----------------|---------------|---------------|-----------------|---------------------|---------------|---------------|
| 30              | 129.90        | 75.00         | 60              | 175                 | 173           | 30.28         |
| 40              | 114.90        | 96.42         | 50              | 136                 | 175           | 26.52         |
| 45              | 106.06        | 106.06        | 45              | 124                 | 212           | 26.29         |
| 60              | 75.00         | 129.90        | 30              | 101                 | 300           | 30.30         |
2.3. Real-time monitoring

Real monitoring of the developed driving assistance system is realised with the development of web-based graphical user interface (GUI) Cayenne. The dashboard that is produced by drag-and-drop function contains slider and meter for clutch position control and vehicle speed, respectively as shown in Figure 3.

![Figure 3. Cayenne Dashboard.](image)

In order to equip the chassis dynamometer with the driving assistance system, a platform as shown in Figure 4 should be fabricated to accommodate all sub-systems, namely electronic enclosure containing Raspberry Pi 4 and electronic components, motor mechanism (stepper motor powered by 110 VDC with gearbox of 10:1 ratio), and linkages (push rod and pedal clamp). The hardware platform is assembled to provide retrofit capabilities of the system in the manual transmission powertrain vehicle without requirement for major modification that can damage the physical of the car.

![Figure 4. The smart driving assistance system in (a) CAD exploded view and (b) actual device.](image)
Table 2 provides the specifications of the test vehicle which is a Perodua Myvi 1.3 L manual transmission car. The car has a reference mass of 955 kg and a maximum mass of 1,015 kg. With total displacement of 1298 cc the rated maximum power and torque outputs were 69 kW @ 6000 rpm and 121 Nm @ 3200 rpm, respectively.

| Vehicle model                  | Perodua Myvi, 5-speed manual gearbox, front wheel drive |
|--------------------------------|--------------------------------------------------------|
| Manufacturer year              | 2017 (standard gasoline 95 RON)                        |
| Reference/ maximum mass        | 955/ 1,015 kg                                          |
| Engine type                    | 1.3-liter 1NR-VE                                       |
| Rated maximum power            | 69 kW (93 hp)                                          |
| Rated maximum torque           | 121 N·m                                                |
| Number of cylinders            | In-line 4 cylinders                                    |

2.4. Case study
The experiments were conducted in the UMP Automotive Excellence Centre (AEC) in Pekan, Pahang, Malaysia where the chassis dynamometer was certified to European standards. The test vehicle was tied on a 17.8-inch-roller Mustang Dynamometer. In this conceptual study, the vehicle was driven under no driving cycle conditions, since the aim of the study is merely to investigate the functionality of driving assistance system fitted in the vehicle under test on chassis dynamometer. Figure 5 illustrates experimental setup on Mustang chassis dynamometer testbed. Licensed driver is hired to perform switching and acceleration in the experimental trials. Investigation of latency in communication protocol between MQTT clients and MQTT-broker during the experiment is recorded.

3. Results and discussion
The experimental results are presented in two subsections, namely (1) the functionality of communication protocol framework and (2) the reliability of the driving assistance system in conducting the driving cycle test on chassis dynamometer tested.

The performance of the MQTT has been analysed through the latency performance of the subscriber node in broker end for MQTT to highlight efficiency of the communication protocol proposed. The result is shown in Table 3.
Table 3. Experimental trials for the entities in cloud-based communication protocol.

| Entity   | Device     | Location          |
|----------|------------|-------------------|
| Publisher| Cayenne    | Karlsruhe, Germany|
| Subscriber| Raspberry Pi 4 | Pekan, Malaysia |
| Broker   | HiveMQ     | BW Cloud          |
| Latency  |            | 765 ms            |

The experimental result shows that MQTT communication protocols written in Python language can perform the gear pedal pushing routine with latency of less than 1 s, which is similar to published results by Mukherjee, Dey and De [9] that obtained 30% improvement of average message delivery performance in case of enhanced MQTT-SN (sensor network) and 17% of improvement for enhanced MQTT with latency of 15.5 ms. The data displayed on Cayenne web-server dashboard also shows that reliable information pertaining the speed profile, speed, throttle pedal, clutch position can be obtained accurately in real time.

Test trials to retrofit the driving assistance system inside the tested car resulted the position and reached distance of the push rod as shown in Figure 6. The fixation of the hardware platform is designed based on finite element analysis to study the deformation of the platform when a force is applied axially at the end of the push rod (which is attached to the foot pedal) and the hardware is fixed at different locations along the hardware frame that are aligned with the existing fixations of driver seat. Experiment shows that if at standby condition the position of push rod is 0°, so at midway is 6° and at the end point is 52°. Whereas the reach distance for the push rod in the pushing process is 0 mm, 105.2 mm and 210.4 mm at the standby, mid-point and end-point, respectively.

![Figure 6](image)

Figure 6. Retrofitting of driving assistance system (a) for clutch control with (b) initial position and (c) end position when the clutch is fully engaged.

4. Conclusion
In this paper a conceptual framework to implement a smart real-time driving assistance system for clutch pedal control using MQTT communication protocol, python, Raspberry Pi 4 and push rod linked to geared stepper motor for vehicles with manual transmission powertrain under chassis dynamometer test. The results show that MQTT communication protocol for MQTT-clients in different continents communication, namely Asia and Europe can be realised with high latency (< 1 s). The retrofitting of the system for use in a manual transmission power train vehicle for emission study or engine performance assessment is effortlessly being done using the existing fixation for the driver seat in the vehicle. In the future, motorised system for accelerator can be utilized to achieve fully IoT driving assistance system for vehicle under test on chassis dynamometer. In addition, real-time implementation of MQTT for hardware, software and On-board diagnostic (OBD) can be done to justify the system’s efficiency.
Acknowledgements

The authors are grateful to the Baden Württemberg Stipendium Plus and Universiti Malaysia Pahang, Malaysia (www.ump.edu.my) for financial supports given under RDU192702, UIC191513, PGRS190305 and iTOPS grant (https://www.bw-stipendium.de/en/the-program/news/current-article/german-malaysian-teamwork/). The authors are indebted to Danial Haris Limi Hawari, Abdul Fattah Mohamad, Nur Hanna Khairul Anuar, Arzielah Ashiqin Alwi, Johannes Neckel, Jessica Schmidt, Carsten Pokluda and Jannik Lassen for the system fabrication and constructive discussion. Furthermore, the authors would like to express the gratitude to the experts in Faculty of Manufacturing and Mechatronics Engineering Technology (FTKPM), Faculty of Mechanical and Automotive Engineering Technology (FTKMA) and Automotive Engineering Centre (AEC) for the technical supports in this project.

References

[1] L. Chen, Z. Wang, S. Liu, and L. Qu, “Using a chassis dynamometer to determine the influencing factors for the emissions of Euro VI vehicles,” Transp. Res. Part D Transp. Environ., vol. 65, pp. 564–573, 2018, doi: https://doi.org/10.1016/j.trd.2018.09.022.
[2] C. Huang and Zhou L, “Evaluation of emission factors for light-duty gasoline vehicles based on chassis dynamometer and tunnel studies in Shanghai, China,” Atmos. Environ., vol. 169, pp. 193–203, 2017, doi: https://doi.org/10.1016/j.atmosenv.2017.09.020.
[3] G. Triantafyllopoulos, A. Dimaratos, L. Ntziachristos, Y. Bernard, J. Dornoff, and Z. Samaras, “A study on the CO2 and NOx emissions performance of Euro 6 diesel vehicles under various chassis dynamometer and on-road conditions including latest regulatory provisions,” Sci. Total Environ., vol. 666, pp. 337–346, 2019, doi: https://doi.org/10.1016/j.scitotenv.2019.02.144.
[4] N. Euler-Rolle, C. H. Mayr, I. Škrjanc, S. Jakubek, and G. Karer, “Automated vehicle driveaway with a manual dry clutch on chassis dynamometers: Efficient identification and decoupling control,” ISA Trans., vol. 98, pp. 237–250, 2020, doi: https://doi.org/10.1016/j.isatra.2019.08.021.
[5] K. Shen, long Chang, H. Chen, Z. Zhang, B. Wang, and Y. Wang, “Experimental study on the effects of exhaust heat recovery system (EHRS) on vehicle fuel economy and emissions under cold start new European driving cycle (NEDC),” Energy Convers. Manag., vol. 197, p. 111893, 2019, doi: https://doi.org/10.1016/j.enconman.2019.111893.
[6] E. S. Mohamed, “Development and performance analysis of a TEG system using exhaust recovery for a light diesel vehicle with assessment of fuel economy and emissions,” Appl. Therm. Eng., vol. 147, pp. 661–674, 2019, doi: https://doi.org/10.1016/j.applthermaleng.2018.10.100.
[7] C.-W. Hong and T.-W. Shio, “Fuzzy control strategy design for an autopilot on automobile chassis dynamometer test stands,” Mechatronics, vol. 6, no. 5, pp. 537–555, 1996, doi: https://doi.org/10.1016/0957-4158(96)00010-4.
[8] X. Liu, T. Zhang, N. Hu, P. Zhang, and Y. Zhang, “The method of Internet of Things access and network communication based on MQTT,” Comput. Commun., vol. 153, pp. 169–176, 2020, doi: https://doi.org/10.1016/j.comcom.2020.01.044.
[9] A. Mukherjee, N. Dey, and D. De, “EdgeDrone: QoS aware MQTT middleware for mobile edge computing in opportunistic Internet of Drone Things,” Comput. Commun., vol. 152, pp. 93–108, 2020, doi: https://doi.org/10.1016/j.comcom.2020.01.039.