System for Planning and Monitoring Driving Strategy

Maciej Gwiżdż and Wojciech Skarka(✉)

Silesian University of Technology, Konarskiego 18A, Gliwice, Poland
maciej.gwizdz1994@gmail.com, wojciech.skarka@polsl.pl

Abstract. The article presents a computer application for planning and monitoring the driving strategy of an energy-saving vehicles. The system is an element of the Monitoring Center of the energy-efficient vehicle and is used by the team participating in competitions of energy-efficient vehicles. The application has a number of functions that allow to define any track profiles and visualize pre-set control data enabling easy tracking of the driver's driving strategy in relation to the planned one. The description of the whole system and its operation is presented in the context of the entire telematics system of the vehicle. The system is used in real time to supervise the implementation of the previously calculated strategy, but it can be used with a strategy that is calculated in real time while driving. In addition, the application is also used to analyze various aspects of already completed racing and test rides, thanks to the visualization functions of any tracks and recorded telemetry data from races, it allows easy analysis of track data after the race and their comparison with strategies obtained in various rides and data from computer simulations. The results of these analyzes are used to correct planned driving strategies and to adjust vehicle design and settings.

Keywords: Driving strategy · Monitoring · Computer application · Energy efficient vehicle

1 Introduction

The reduction of energy consumption used to power a moving vehicle is done by optimizing the design of the vehicle during the design and adjusting the vehicle settings as well as by choosing the appropriate driving strategy. Especially the last issue brings quite big effects when the vehicle route is known. In Shell Eco-marathon competitions, energy-efficient vehicles drive along a well-known route and our special attention is paid to choosing the right driving strategy and research related to this issue becomes the main research tasks carried out by the teams involved. Simulation models or specially dedicated software for simulation purposes are used for design purposes as well as for vehicle development [1, 4, 8, 13]. The selection of our case strategy takes place through time-consuming simulation calculations [13], [18, 19] before the races for the conditions of the given route and the current weather conditions. They are implemented using the vehicle simulation model and route as well as race conditions. It is true that there are real-time driving optimization methods, but they do not give such

J. Mikulski (Ed.): TST 2019, CCIS 1049, pp. 216–229, 2019.
https://doi.org/10.1007/978-3-030-27547-1_17

good results as the aforementioned offline optimization calculations in this case. The Smart Power team [12] since their first competitions in Shell Eco-marathon [17] (SEM) races optimizes the racing vehicle driving strategy [13, 19] especially for the last vehicle HydroGENIUS (Fig. 1) which is UrbanConcept Category vehicle with Hydrogen Fuel Cell Stack used as promising electric energy source [20]. During the race it is necessary to oversee this driving strategy and verify its implementation. It is accomplished by the center supervising the work of the race driver and driving parameters in relation to the calculated strategy. So far monitoring has been implemented through:

– familiarizing the driver with the strategy and earlier training of the driver's implementation of this strategy,
– supervising driving parameters by the monitoring center, identifying deviations from this strategy and instructing the driver to alter the driving corrections,
– monitoring driving parameters through the use of appropriate software to identify deviations from the strategy in a quantitative manner [2]
– recording driving parameters and analyzing these results by a team of strategists on a regular basis during the race and between subsequent races

Fig. 1. A vehicle of the UrbanConcept category of the Smart Power team taking part in the Shell Eco-marathon competition [own study]

In particular, the latter method brought a lot of benefits [1, 4, 13] and had a decisive impact on the quality of strategy implementation. Using the experience of previous years of starts, it was decided to develop new modular and developmental software that will allow to easily and quickly supervise driving during a race and implement driving strategies. The software should enable easy development and in particular integration with the dynamic driving strategy implementation system, which is planned to be developed in future years. In addition to work related to vehicle optimization and driving strategies, Smart Power team conducts a number of works related to, among others, building support and driver systems and driving automation [6, 7], using generative modeling to automate the design process.

2 Overview of Vehicle Monitoring Systems

Systems for monitoring vehicle motion parameters and driving strategies used for sports purposes, especially those highly specialized, are developed or adjusted only for the objectives and characteristics of the team or vehicle and are not widely available. However, it is widely common to use systems that monitor vehicle operational parameters for commercial purposes, they are primarily used to manage a fleet of vehicles.

The SAT-DOG [3] system is an example of a vehicle monitoring system. It is a system that collects data on company servers, which are secured through client authentication based on a special encryption. The SAT-DOG device enables monitoring the current location of the vehicle based on the GPS system and much technical information using sensors, circuit breakers, measuring probes and alarm systems. The Map Center produced by the Polish company E-Map is used to visualize the location of vehicles.

The system also allows fuel measurement. Additionally, it allows to control the amount of fuel in the tank by measuring the fuel from the float of the measuring system in the vehicle's tank. A very accurate measurement is possible if an additional fuel probe in the tank is installed. Thanks to this, it is possible to analyse the average fuel consumption.

Another example of vehicle monitoring systems is the Globtrak system [9] with monitoring and system functions.

Monitoring functions: Indications available from the vehicle's CAN, Fuel level, Opening/closing the door, Driver identification, Private ride, Using air conditioning.

System functions: Summary of fuel consumption in any time interval, Archiving routes and data from monitored functions, Implementation of the route plan, Reporting to a spreadsheet or text file.

The presentation of data read from vehicles also applies to applications for mobile devices. An example of such an application is DashCommand. It reads the data through the vehicle diagnostic connector. It allows tracking selected vehicle parameters. The system allows the analysis of vehicle diagnostic data in real time, thanks to the connection to the OBD II module (ELM 327) via Wi-Fi or Bluetooth.

The examples of parameters that can be observed include:

Speed, Engine speed, Fuel consumption data, Acceleration, Torque, Display of temperature sensors, Display of pressure sensors, Location, Number of stops, Driving time.

Most of the available systems for monitoring vehicles are based on the GPS (Global Positioning System) system - satellite navigation system for detecting position on the map.

In addition to the GPS system, various types of sensors installed in vehicles and CAN (Controller Area Network) information technology are also used. It is a two-wire bus, where the transmission takes place in real time at speeds up to 1 Mb/s. It enables communication between electronic modules.

3 Supporting Vehicle Driving Strategies

So far, the strategy has been calculated on the basis of complex calculations of optimization simulation carried out with MBDO (Model Based Design and Optimization) techniques [13, 18], using a simulation model built in the Matlab/Simulink software. The simulation model created for this purpose included:

Simulation model of the vehicle itself reflecting the equations of moving vehicle motion, aerodynamic characteristics, characteristics of subsystems affecting driving and energy consumption, i.e. engine, drive system, drive transmission system, track simulation model, its geometrical configuration and type of surface, external conditions, weather and weather conditions.

Simulation calculations led to obtaining a driving strategy which included mainly the speed profile and corresponding moments of driving the drive system as a function of the position on the track. In addition, the strategy was directly verified during test drives on the track and tuned directly before the race.

Implementation of the strategy is a task carried out by the driver and the control center and is based on experience, learning the driver's strategy and its supervision and correction by the persons supervising the ride. It is extremely difficult because the rides are held in racing conditions and the main factors disrupting the strategy implementation process are other racing vehicles moving on the track. In the peak moments there is quite a lot of traffic on the track and the attention of the driver must focus on avoiding collisions and at the same time on implementation of the strategy. In addition, there are problems in updating the current position and there is a delay in responding to the supervision center. Based on the experience of the supervision center and drivers and the application of the previous software and used good practices, assumptions and concepts of the currently implemented application supporting the work of the strategists' center were developed. So far, during the competition, the crew set some checkpoints to optimize the travel time on which the team members were. They communicated using walkie-talkies with a strategist, announcing that the vehicle is at a given control point. The strategist checked the current time on the stopwatch and wrote it down on a piece of paper, and then analyzed it. The developed application was to include a system in which this process would be automated.

3.1 The Impact of Driving Strategies on Fuel Consumption

During the design process, the built simulation model was used to optimize the design features [13]. Under operating conditions, the model was used to determine the settings of the vehicle, e.g. the transmission of the main transmission and the development of the driving strategy [18]. Doubts as to the rationale of determining the driving strategy and its impact on the result of the race were examined during the first simulation calculations where a comparison between the strategy calculation results obtained from the simulation model and the strategy selected based on experience and principles of energy-efficient driving was made [19]. Additionally, the same verification was carried out once more while racing. Because the number of racing trials is significantly limited, in practice, verification has been made only once. The results obtained by simulation calculations were significantly better than those obtained during rides and simulations based on the driver's experience and skills. In extreme cases, the differences between the optimal strategy results from the simulation model were twice as good. Typically, the results of optimization gave a few-dozen percent improvement in results obtained in relation to the strategy invented by the driver [19].

3.2 Assumption and Concept of the Application

The main assumptions for the application resulted from the needs of the driving strategy verification center. The assumptions adopted for creating the application included:

- Application of the software during the race to the current analysis and monitoring and then after the race to analyze the saved data and compare it with simulation data
- Simple interface for working during the race
- Complex comparative analyzes conducted after the race
- Visualization of driving during the race in graphic mode on the track model based on data transmitted online from the vehicle via the telematics system
- Defining checkpoints allowing verification of positions based on real observations
- Bidirectional data transmission
- Access to detailed data in various comparison modes, e.g. current deviation from the planned strategy
- Monitoring technical data and parameters of the operation of the sub-assemblies
- Implementation of corrections and recommendations for the driver directly by the strategy based on the observation of results
- Possibility of extending the application in particular by integrating the module to automatically propose corrective actions for the current driving mode with the possibility of automatic feeding of vehicle control signals

Based on these general assumptions, the following detailed technical requirements were defined and application concepts were proposed.

Data to be available in period of time: Speed, Power (current, voltage), Detailed assumptions:

- Frequency of data downloading - 1 s
- The method of representation of data collection - a graph
- Type of data downloaded - floating point numbers (float/double)
- Type of data project - natural numbers (int)
- Move data on the chart for available reading it in real time
- Possible changes in the size of the chart
- Offering the current state of the chart, as image and csv file

Assumptions about the location of the vehicle on the map:

- The position of the vehicle on the track by coordinates x, y
- Representation method - map
- Map refresh rate - 1 s

In the assumptions of the race, it is required to drive a certain number of laps in a given time. Therefore, it is necessary to select a few checkpoints on the map and define for them the desired average speed with which the car should move. If the actual average speed is too low, we should get a message to make the driver speed up, whereas if the speed is too high, let the driver release it.

An additional facilitation will be the timing difference from the set average value at a given control point. In the transmission, when the driver, exceeding the control point, is late by 5 s, we will receive a message of +5 s, while if it is 5 s late for the given average speed we will receive a message of −5 s.

Current state of vehicle sensors allows to measure the following data:

Velocity, Current, Voltage, Vehicle position.

In the future it is planned to have the following measuring parameters:

Hydrogen consumption measurement, Cell temperature, Fan speed.

The speed measurement is carried out by means of a magnetic encoder mounted on the wheel. The present measurement is carried out thanks to a special measuring system with the use of current measurement sensors connected directly to the electrical system. The voltage measurement was realized on the principle of a simple voltage divider.

At present, in each car the GPS module is assembled by the organizers of the competition. Nevertheless, the plans are to implement our own module enabling tracking of the current position of the vehicle.

Hydrogen consumption measurement will be possible due to the use of a flow sensor. This sensor is now available and requires integration with the current system.

There are plans to add the ability to measure the temperature of the hydrogen cell and control the speed of the fans and to supply air to the cell in order to reduce energy consumption and at the same time prevent the cell from overheating. The temperature will be read by the Arduino UNO using the NTC 110 thermistor, and the fans will be controlled by the PWM signal.

The programming language that was used to create the application was C++.

In order to write a window application monitoring the vehicle's driving, it was necessary to use a library that made it possible. For this task a set of portable Qt libraries was chosen. It is a programming tool dedicated to other languages for C++, which consists of classes for building GUI (Graphical User Interface). It allows for multi-threaded programming and file operations. It also enables network communication - sockets and HTTP, FTP and SSL protocols.

One of the ways to visualize data were graphs. They have been used both to display the received data in the time domain and to create a graphical representation of the race route [5]. For this purpose, it was necessary to use a library to create charts, It was decided to use the QCustomPlot library [14]. QCustomPlot is a widget dedicated to Qt [15, 16], written in C++ for the drawing and visualization of data. It is very well documented, allowing to create nice looking graphs and visualize data in real time.

4 Description of the Application

The main tab (Fig. 2) contains the most important data. It consists of the following elements: Graph monitoring, Track monitoring, Time monitoring, Laps time viewer, Checkpoints monitoring.

Fig. 2. Main window tab of the application [own study]

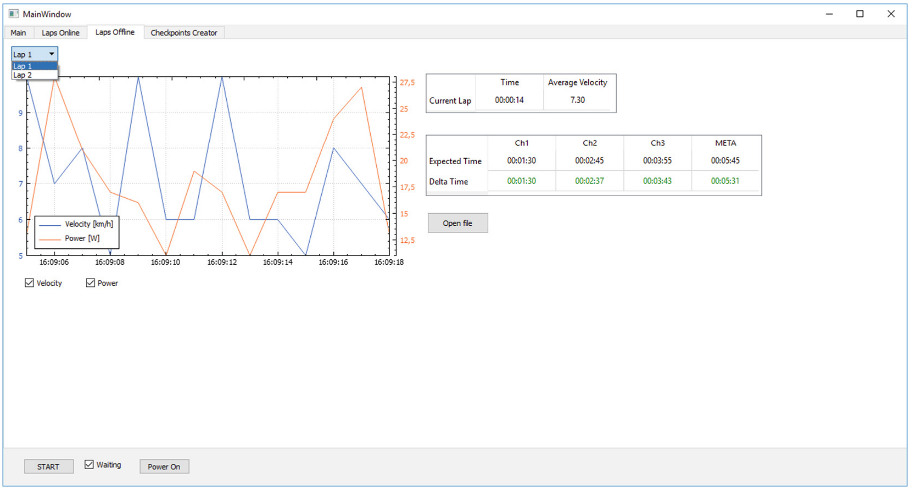

Fig. 3. Laps Offline tab of the application [own study]

The Laps Online and Laps Offline tabs (Fig. 3) are very similar to each other. The difference is that the Laps Online tab allows you to analyze data from the current race, while the Laps Offline tab allows you to analyze data from a race that has already ended. Laps Online includes a 'Save As' button that allows you to save the current data to a file, while Laps Offline has an 'Open File' button that allows you to read the data you want to analyze from a file.

This is a tab (Fig. 4) that allows you to create checkpoints on the trajectory that supervise the optimal vehicle speed.

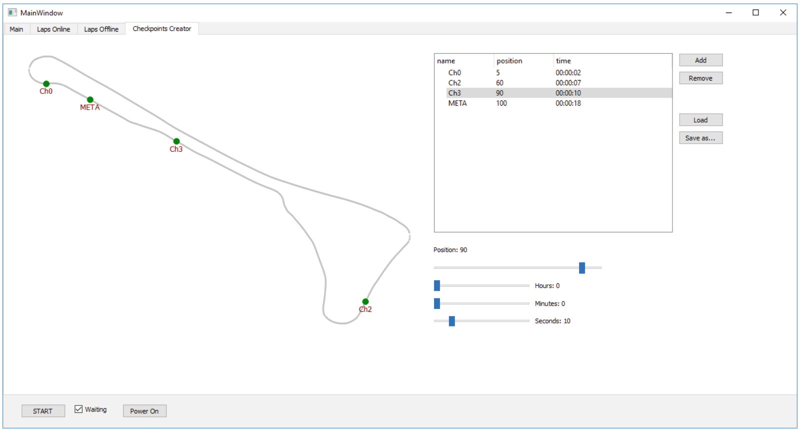

Fig. 4. Checkpoints creator tab of the application [own study]

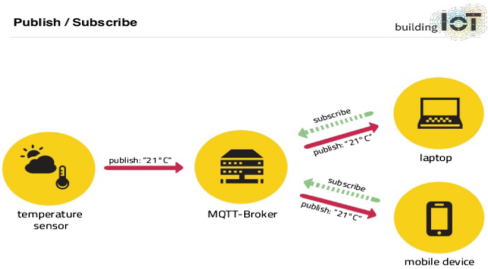

Fig. 5. Idea of operation in publisher/subscriber mode [own study]

For wireless data transfer, the MQTT protocol was selected [11]. MQTT (MQ Telemetry Transport) - is designed for devices that do not require high bandwidth. It ensures high reliability and energy saving, which in our case is a very big advantage. It was created by Andy Stan-Ford-Clark from IBM, and by Arlene Nipper of Arcom (now Eurotech) in 1999. MQTT uses the Facebook and Messenger protocol for data transmission.

MQTT works on the basis of the publisher/subscriber pattern (Fig. 5). It is based on the fact that each client connects to the broker. Clients can subscribe to a given topic and publish data on a given topic. When the client publishes information on the given topic, each of the clients who subscribes to this topic downloads this data.

5 Application Tests and Verification

In order to guarantee the correct operation of the application, a detailed test plan was prepared. It happens quite frequently, for this type of application, that tests are skipped or neglected, which contributes to the poor reception of the application. During the detailed tests described below, not only the obvious errors were corrected but the functionality of the application itself was also improved, adding better detailed solutions and the user-friendliness and transparency of the interface. The tests were carried out in an iterative manner for particular revised and improved software versions. The previously planned and systematic testing procedure enabled efficient and rapid software prototyping.

The test of the application has been divided into three parts:

– Manual test
– Data transmission test
– Test on actual race data

Configs.txt file (Fig. 6), was created for testing which contains the following data:

– state - this is the mode of operation of the program, three basic modes can be distinguished- Offline, Online and OnlineRandom mode, where Offline mode means offline simulation when generating random data, OnlineRandom is a test using random data that is sent over the network using the MQTT library whereas online mode is the intended operation of the application during the race, if the application connects to the broker and waits for the data provided to it
– simulatingDt - means the speed at which data should be generated during simulation in Offline or OnlineRandom mode
– brokerName - name of the broker
– topicName - the name of the topic

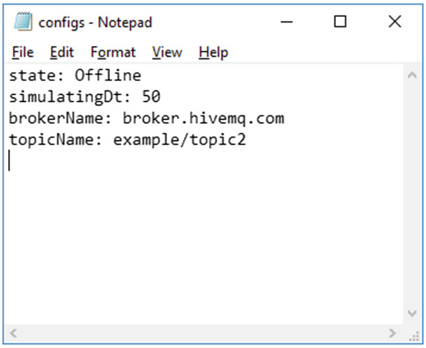

Fig. 6. Exemplary configs.txt file for the offline testing of the application [own study]

In order to carry out the tests, the Received Data tab (Fig. 7) has also been added, which displays the received data. This is very important for the Online test to see if we receive data and if received data is correct.

The manual test consisted in checking the correct operation of available program functions. The Offline mode was set in the configs.txt file during this test.

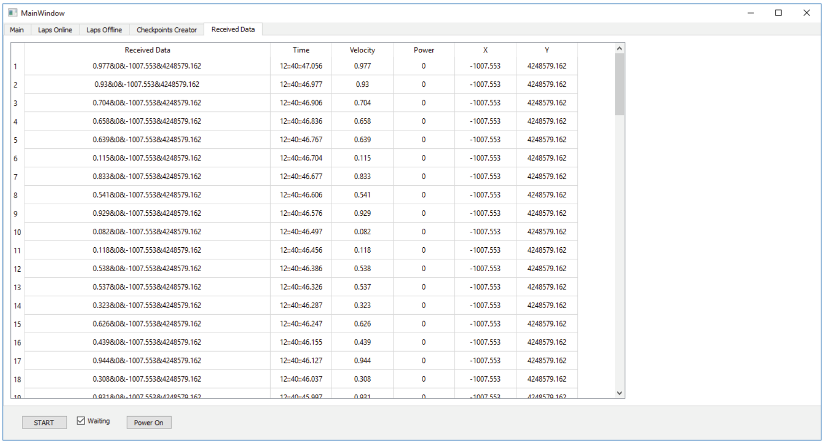

Fig. 7. Additional window tab – Received Data – displaying current received data for verification in test application process [own study]

The main tab test plan:

1. Pressing the START button - the graph should display a motion with a value between 0 and 1 and a constant power value of 0.
2. Pressing the Power On button - a vertical green line marking the beginning of a new lap should appear on the graph, speed and manual power take random values from the specified range.
3. In the Time Monitoring table, the current lap time and average speed can be observed.
4. On the route, a moving white and red point symbolizing the vehicle can be observed
5. We can also observe the Checkpoints Monitoring table and check if the time deviation appears in the table when the point symbolizing our vehicle crosses the green dot marking the control point in the table.
6. Vehicle reached finish line - the vehicle stops moving, a vertical red line should appear on the graph symbolizing the end of the lap, the counter calculating the current lap time is stopped, the total time and time of the last lap is added to the Laps Time Viewer table. The Waiting checkbox is cleared. Data from the last lap will be saved to the backup file.
7. To simulate the next round, select the Waiting check box and then press the Power On button again.

The plan to run the Laps Offline and Laps Online tabs test is as follow:

1. The vehicle shall be simulated for at least two laps.
2. In the Laps Online tab, check if there is a possibility of previewing the laps that have been passed.
3. Then save the file anywhere on the disk.
4. Next, go to the Laps Offline tab, load the previously saved file and check if it has been loaded correctly and whether we can observe the data.

Plan to test Checkpoints Creator tab is as follow:

1. Use the Add and Remove buttons to add a specific number of control points.
2. Then, using the slider responsible for the position of the point on the map, set the positions of all control points in such a way that the control point with the larger index is further away from the one with the lower index.
3. Using the sliders responsible for time, set the waiting times for all control points.
4. Then save our set checkpoints to give the disk.
5. We can also load any saved control points and modify them and save them again.
6. If we want to use other settings with the control points in the Main tab, we need to replace the file checkpoints.txt in the folder. Then turn the application off and on.

Conducting a data transmission test consists in setting in the file configs.txt the operating mode of the application on OnlineRandom and setting appropriate host and any topic. For this purpose, the host 'test.mosquitto.org' [10] first served (Fig. 8). The test was similar to the previous one, with the difference that the generated data was placed in the character string according to the following system: time & velocity & power & x & y, where: time - the current time taken from the computer, velocity, power, x - vehicle position from the GPS module at position x, y - vehicle position from the GPS module at position y.

Then, the data was sent thanks to the use of the MQTT protocol.

Unfortunately, during the test lap, sometimes there was a temporary loss of connection (Fig. 8). At the moment when communication was recovered, everything returned to normal and this did not cause any critical errors for the operation of the entire application.

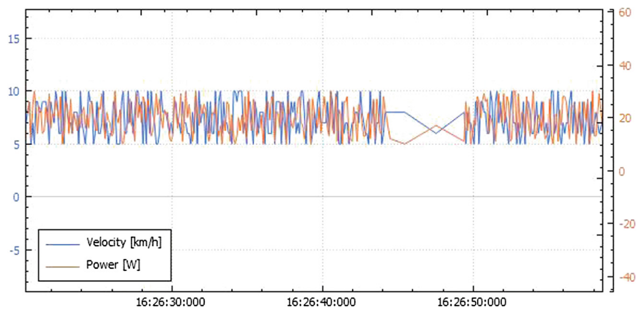

Fig. 8. Test result of the downloaded data using the 'test.mosquitto.org' host. Observed problem resulting from temporary loss of connection [own study]

This problem drew our attention to a certain situation. At the moment when the vehicle is just before the checkpoint and the connection is broken, then after a few seconds, when the car has passed the control point and will be behind its radius, the detection checkpoint will not be passed and we will not receive information about the time deviation for given control point. To avoid such a situation, it is necessary to introduce a security system that will cope with such an event. Another idea was to test another broker. A host named ‘broker.hivemq.com’ [11] has been subjected to the tests. The test result for this broker was very satisfactory.

As you can see in the figure above (Fig. 9), no disturbances occurred during the test. However, there is such a probability, so the safety button has been added to the application.

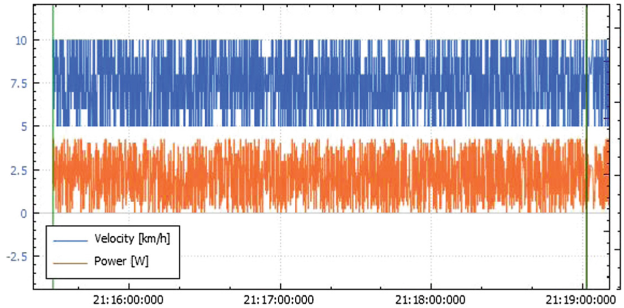

Fig. 9. Test result of the downloaded data using the ‘broker.hivemq.com’ host [own study]

In the Main tab, the REACH CHECKPOINT button has been added. It allows to manually pass the checkpoint. At the moment when there are disturbances, as a result of which the vehicle will pass through the checkpoint, when it is not noted, then the person supervising the crossing may manually mark the checkpoint.

6 Conclusion

The purpose of the work was to develop a usable version of the *Hydro*GENIUS vehicle monitoring system. This system is to be used as a driving control center during the route during the Shell Eco-marathon competition. It should enhance the construction of the most energy-efficient vehicle.

Its main task was to be real-time visualization of downloaded data and their interpretation by a strategist who communicates with the driver during the race. This would help a driver to be able to focus more on driving. In addition, it will allow the analysis of data that the driver would not be able to observe and analyze while driving.

The main assumptions have been fulfilled. Conducted detailed and systematic tests confirmed that the basic functions of the application work correctly and also enabled the improvement of the application. While testing the application, one of the brokers has a temporary loss of signal, which may cause the lack of calculated time deviation at the control point. For this purpose, a button has been added that allows to pass the checkpoint manually.

Additionally, the problem of different data format received by the GPS module and the points that form the route was encountered. Therefore, it is necessary to harmonize them.

In the future, this system will be capable of extending which should allow vehicle diagnostics in real time. The list of planned works includes wider collection of information on the technical parameters of the vehicle subsystems and the extension of the monitoring system on the vehicle. In addition, it is planned to implement a system that, instead of manually, by changing the points.txt file, will allow you to load the route from the application level automatically. It is also possible to change the current way of presenting the route by using Google Maps, which will allow for the correct operation of the application anywhere without having to know the GPS coordinates of the entire route. It is also intended to add functionality that allows to change the application mode, data download frequency during simulation, broker and topic settings. In the future, the system will be expanded with a real-time strategy system that, based on the vehicle speed profile, will be able to determine the optimal speed of the car between the control points. It is also prearranged to add a tab with a chart, to which you can download any diagnostic data. From the user's point of view, it is also worth thinking about designing the graphic design of the application.

It should be noted that although the application works correctly, in the next seasons of the race it is planned to systematically develop the application including, on the one hand, increase of diagnostic functionality through the ability to display subsequent data and, on the other hand, the introduction of intelligent diagnostic and strategic functions. Particularly these latter tasks, however, require advanced research in this area, which is currently being carried out by the team members and supervisors. From the beginning, such systematic and planned development of the application will facilitate future tasks. The already developed working application is an excellent help for the strategy supervision center and is a base for integration of new intelligent functions.

References

1. Anh-Tuan, N., Reiter, S., Rigo, P.: A review on simulation-based optimization methods applied to building performance analysis. Appl. Energy **113**, 1043–1058 (2014)
2. Cichoński, K., Skarka, W.: Innovative control system for high efficiency electric urban vehicle. In: Mikulski, J. (ed.) TST 2015. CCIS, vol. 531, pp. 121–130. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24577-5_12
3. Description of the system. Sat-dog. http://www.sat-dog.pl/opis.php. Accessed 1 June 2018
4. Gao, D., Wenzhong, M.C., Emadi, A.: Modeling and simulation of electric and hybrid vehicles. Proc. IEEE **95**(4), 729–745 (2007)
5. Google Maps in LabVIEW. Forum National Instruments. https://forums.ni. Accessed 1 June 2018
6. Jezierska-Krupa, K., Skarka, W.: Design method of ADAS for urban electric vehicle based on virtual prototyping. J. Adv. Transp. Article Number: 5804536 (2018). https://doi.org/10.1155/2018/5804536
7. Jezierska-Krupa, K., Skarka, W.: Using simulation method for designing ADAS systems for electric vehicle. In: 23rd ISPE Inc. International Conference on Transdisciplinary Engineering, Federal University of Technology, Curitiba, Brazil, 03–07 October 2016. Transdisciplinary Engineering: Crossing Boundaries Book Series: Advances in Transdisciplinary Engineering, vol. 4, pp. 595–604 (2016)

8. Markel, T., et al.: ADVISOR: a systems analysis tool for advanced vehicle modeling. J. Power Sources **110**(2), 255–266 (2002). Article Number: PII S0378-7753(02)00189-1 AUG 22
9. Monitored functions. Globtrak. https://www.globtrak.pl/floty-handlowe/. Accessed 1 June 2018
10. Mosquitto. Mosquitto. https://test.mosquitto.org/. Accessed 1 June 2018
11. MQTT 101 – How to Get Started with the lightweight IoT Protocol. HiveMQ. https://www.hivemq.com/blog/how-to-get-started-with-mqtt. Accessed 1 June 2018
12. Smart Power. Smart Power Team. http://185.201.112.44/. Accessed 1 June 2018
13. Skarka, W.: Reducing the energy consumption of electric vehicles transdisciplinary lifecycle analysis of systems book series: advances in transdisciplinary. Engineering **2**, 500–509 (2015)
14. QCustomPlot. Introduction. http://www.qcustomplot.com/. Accessed 1 June 2018
15. Qt-based Clients for Google APIs. Ics. https://www.ics.com/technologies/qt/qt-based-clients-google-apis.com/t5/Example-Programs/Google-Maps-in-LabVIEW/ta-p/3512488. Accessed 1 June 2018
16. Qt Charts. Qt. https://doc.qt.io/qt-5/qtcharts-index.html. Accessed 1 June 2018
17. Shell eco-marathon. Shell. http://www.shell.com/energy-and-innovation/shell-ecomarathon.html. Accessed 1 June 2018
18. Targosz, M., Skarka, W., Przystałka, P.: Model-based optimization of velocity strategy for lightweight electric racing cars. J. Adv. Transp. Article Number: 3614025 (2018). https://doi.org/10.1155/2018/3614025
19. Targosz, M., Szumowski, M., Skarka, W., Przystałka, P.: Velocity planning of an electric vehicle using an evolutionary algorithm. In: Mikulski, J. (ed.) TST 2013. CCIS, vol. 395, pp. 171–177. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41647-7_22
20. Tyczka, M., Skarka, W.: Optimisation of operational parameters based on simulation numerical model of hydrogen fuel cell stack used for electric car drive. In: Conference: 23rd ISPE Inc. International Conference on Transdisciplinary Engineering, Federal University of Technology, Curitiba, Brazil, 03–07 October 2016. Transdisciplinary Engineering: Crossing Boundaries Book Series: Advances in Transdisciplinary Engineering, vol. 4, pp. 622–631 (2016)