Design, build, and test drive a FSAE electric vehicle

Kala Meah¹, Donald Hake II¹, Stephen Wilkerson²

¹Department of Electrical and Computer Engineering and Computer Science, York College of Pennsylvania, York, Pennsylvania, USA
²Department of Civil and Mechanical Engineering, York College of Pennsylvania, York, Pennsylvania, USA

Abstract: This study presents a multidisciplinary end-to-end design, build, and test drive experience of a Formula Society of Automatic Engineers (FSAE) electric vehicle. The design team members' background include computer engineering, electrical engineering, and mechanical engineering expertise. Each team member worked on a sub-team namely mechanical, drivetrain, supervisory control and data acquisition (SCADA), and battery management system (BMS). The mechanical team modified/ redesigned a gasoline FSAE car frame to accommodate the battery pack, cooling systems, electric drive train, and SCADA hardware. The drivetrain team designed a combinational logic-based tractive system control board to enhance the safety and reliability of the drivetrain. The SCADA team used a model-view-controller software architecture and agile development practices to design and implement a custom electric vehicle operating system and the main brain hardware. The BMS team designed the sensing network and communication protocol to provide an update on the state of the health of the energy source. A thorough description of each sub-system is provided herein. System integration and field test results are also included.

1 Introduction

Electric vehicles (EVs) have been experiencing an extraordinary growth in the recent years around the globe. In 2018, the global electric car production added 2.1 million vehicles to bring the total number to more than 5.1 million. During the first half of 2019, EV sales increased by 22% in the USA, 66% in China, and 35% in Europe compared to the first half of 2018. The U.S. EV sales represented ∼13% of global EV sales for the first half of 2019 [1, 2]. EVs have some key advantages over internal combustion engine vehicles: low emissions, lower or zero fuel costs, fewer moving parts, low maintenance, and high acceleration capability. Disadvantages of EVs include but are not limited to higher cost, driving range, and lack of charging infrastructure. China, the USA, and Europe are leading in EV market shares. EV market can be predicted by income and availability of recharging infrastructures in the country as noted by Mersky et al. [3]. The good news is that key factors such as battery technologies, infrastructure development, and private sector interest are working in favour of EV production and marketing.

Academic research on a complete end-to-end design, build, and test of an EV is limited. To address this issue as well as to equip future engineers equipped with the proper technical knowledge and skills, this team engaged in a multidisciplinary, complex, and real-world EV project. The team decided to build the EV based on the Society of Automatic Engineers (SAE) hybrid EV guidelines. The project was divided into the following sub-teams:

- mechanical design
- drivetrain
- supervisory control and data acquisition (SCADA)
- battery development and battery management system (BMS)

The formula SAE (FSAE) EV design team utilised the gasoline-powered formula (FSAE) frame from the previous year. The team had to modify the frame to accommodate the electric motor, motor controller, differential, battery back, cooling systems, and the drivetrain electrical systems. The front of the vehicle was modified to accept the SCADA system, dashboard, brake, and the electric acceleration systems. Teams will typically assign tasks in accordance with their own strengths. This team management used the proven techniques of project-based learning [4–7]. The project was given some basic engineering constraints on cost and milestone deadlines, and had to also conform to the FSAE regulations for EV. Other than that, the design teams were free to formulate their own solutions. The rest of this paper provides descriptions of the design and implementation of the mechanical frame, drivetrain, battery pack, SCADA system, and the BMS, as well as system integration and laboratory and field testing.

2 Mechanical design

The Emrax 208 motor was a good fit for the project due to its high power-to-weight ratio and shaft torque [8]. The Unitek Bamocar D3 motor controller was selected for its versatility and proven track-record with the Emrax 208 motor [9]. The motor and controller combination can provide up to 80 Nm of continuous torque with a 75 kW power at 6000 RPM at the drive shaft. The frame starting point is given in Fig. 1. The task was to redesign the rear half of the chassis to accommodate the battery box and electrical drivetrain. The team researched possible solutions and then justified their decisions. The tools at their disposal included computer aided design (CAD) software (SolidWorks) and finite-element analysis (FEA). The existing frame, motor, and motor controller, along with the battery box and the need for a cooling system was the starting point for the team's effort.

The team conducted extensive analysis and modelling using SolidWorks and FEA. Fig. 2 shows the FEA analysis for the left motor mount with a minimum factor of safety 1.9. After comprehensive analysis and trade-off studies were conducted, the team determined that the front portion of the frame, from the roll-hoop forward, could be reused, as long as the rear of the frame was extended to accommodate the additional electrical components. These modifications also needed to adhere to FSAE rules and size constraints. The modified frame with battery pack, electromechanical drivetrain, and control circuits is shown in Fig. 3.

A cooling system was designed and implemented using water to keep the motor, motor controller, and battery cells cool. The cooling system pump has a flow rate of 12–16 litres per minute (LPM) and the pump was powered by a separate 24 VDC battery. The pump sends flow through the radiator and then splits it into the motor/controller and the internal cooling plates. The motor
controller and motor specify optimal/maximum pressures and flow rates while keeping the flow split in parallel maintains the flow within its limits. A quick connect is situated at each six-way splitter such that the battery cooling plates can be disconnected from the circuit to allow the box to drop out without taking the entire cooling loop with it. The simulation analysis for the cooling plates was conducted using the Autodesk Computational Fluid Dynamics tool. The simulation used a flow rate of 1.2 LPM. A slight temperature gradient was used to simulate the battery heat generation. The batteries used have a tendency to heat more at the top, so a top down gradient was used. The overall outcome was such that the cooling plates individually will support 1 kW of heat transfer. The cooling system was constructed using aluminium and copper plates. The laboratory test showed the cooling systems can support 1 kW heat transfer per plate with 1.2 LPM. Fig. 4 shows the cooling loop layout and a cooling plate.

A chain drive was used to transmit the torque from the motor shaft to the rear wheels. The team was provided with a driving sprocket and a driven sprocket. These sprockets gave the car a 2.7 gear ratio which multiplies the torque at the rear wheels. It was determined that a limited slip differential would allow the wheels to spin at different rates when entering a turn and provide an advantage. A limited slip differential has the ability to still transmit the torque to both wheels when there is a sudden change in torque.

3 Drivetrain

The drivetrain motor draws power from the high-voltage DC batteries, via the three-phase AC Unitek Bamocar D3 motor controller. The speed command input, which notifies the motor controller how much power to apply to the motor, is controlled from the SCADA system and communicates with the motor controller via a controller area network (CAN). The specifications used to design the system were provided by the SAE in the formula hybrid SAE rules [10]. These specifications were followed by the team while designing and building the vehicle. The drivetrain sub-team designed and fabricated a tractive system control board (TSCB) to meet the above requirements. Fig. 5 shows the overall operation and communication of the drivetrain subsystem. The heart of the drivetrain subsystem is the custom designed TSCB [11]. The SAE requires that the high-voltage system be controlled by non-programmable logic in order to increase the safety of the high-voltage system. The logic circuit for the TSCB, along with SCADA and other sub-system interfaces, can be seen in Fig. 6. The TSCB uses digital logic to engage and disengage the voltage system and the motor controller inputs. Non-programmable
combinational logic circuits are widely used for the vehicle tractive systems (TSs) for their reliability in detecting failures to ensure safe operations [12–17].

To enable the TS, the TSCB receives digital logic signals from SCADA. These signals can be seen in Fig. 6. These connections are: SCADA nominal, SCADA pre-charge, and SCADA drive (all of which are outputs from SCADA and inputs to the TSCB); 90% charge (which is an output from the TSCB and an input to SCADA). The signals that were used between the TSCB and the motor controller are: digital output 1 (DOUT1), DOUT2 (which are outputs from the motor controller); rotating field enable (RFE), and RUN (which are inputs to the motor controller).

The SCADA nominal signal notifies the TSCB once the grounded low voltage (GLV) system is up and operational. Whenever the signal is low, the high-voltage system is shut down. This signal is used not only to verify that GLV is operating nominally, but also to indicate that the high-voltage system needs to power-down and for the GLV to shut down or go into standby mode. The SCADA pre-charge signal initiates pre-charge by closing the negative path on the accumulator isolation relays (AIRs) and closing the positive resistive path on the AIRs. Once the DOUT1 signal from the motor controller has been received by the TSCB, the signal is converted into a +5 V digital logic signal which indicates 90% charge. The 90% charge signal is then used to enable RFE, engage the positive non-resistive path on the AIRs, and disengage the positive resistive path on the AIRs. The signal is also sent back to SCADA for notification that 90% charge has been completed. Once the driver is ready to put the car into drive, the driver pushes the ‘ready to drive’ (RTD) on the dashboard and SCADA sends the SCADA drive signal to the TSCB. Once the TCSB receives the SCADA drive signal, the TSCB enables the RUN signal for the motor controller. Once the motor controller receives the RUN signal, the motor controller is ready to accept speed command values from SCADA via CAN. The signal DOUT2 is not used in the current implementation, but has been included for future use. The shutdown signal is converted +5 V and is also used to determine if the shutdown circuit is open or closed. If the circuit is broken at any point, then the high-voltage system shuts down due to the AIRs losing power.

The design team conducted extensive laboratory tests to finalise the design and to fabricate the printed circuit board (PCB). During testing, the team resolved issues with the motor controller RUN signal always being in high state, and with chattering of the AIRs relays. The final PCB of the TCSB along with the safety circuit and motor resolver is shown in Fig. 7.

4 Supervisory control and data acquisition

The SCADA system processes information and makes decisions based on an algorithm developed by the team to control the actions of the vehicle. The SCADA system mainly interacts with the TSCB and the BMS. Decisions made by the SCADA system directly control the motor, high-voltage and low-voltage systems, and static data storage system. The SCADA also controls the decisions pertaining to various rules, communication with subsystems, input from the driver, and the presentation of relevant vehicle status information to the driver. The communication protocol and system design were guided by the FSAE formula hybrid EV rules book [10]. Significant consideration was given to the motor controller while selecting the protocol and interface of the SCADA system. After a comparison of the various communication protocols compatible with the motor controller, CAN bus (a two-wire differential voltage communication protocol) was selected [18]. This is one of the five protocols supported by on-board vehicle diagnostics standard. This protocol is so robust that communication is still possible even with the two wires shorted together.

The heart of the SCADA hardware system is a Teensy 3.6 microcontroller which has a microSD slot, real-time clock, and floating point co-processor [19]. The microcontroller tasks include logging of data collected from various inputs, controlling the systems over the CAN bus, handling the drive-by-wire input system, and providing feedback to the driver. Fig. 8 shows the logical connections of the hardware to the Teensy 3.6. The CAN bus is connected to the Teensy through MCP2562 transceiver module [20]. The SCADA hardware includes the following sub-systems:

- A brake over-travel switch;
- TS master switch (TSMS);
- Grounded low-voltage master switch;
- Two side-mounted big red buttons (BRBs);
- Insulation monitoring device (IMD) interlock;
- Cockpit-mounted shutdown button;
- Accumulator management system interlock;
- Two AIRs as the primary means of engaging and disengaging the TS voltage.

A search of the published literature revealed only a partial simulation and/or implementation of a SCADA system for an EV [18, 21–25]. A complete software and hardware system implementation is absent from the current research. This paper contributes to this emerging field by designing and implementing the software and hardware system for a formula SAE EV. Safety is paramount in FSAE EV operation and the SCADA system monitors all safety protocols. To maintain the safety, the car shutdown circuit must not be reestable by the driver from anywhere within the cockpit; the non-programmable control circuit TSCB must control IMD and BMS interlocks; the shutdown circuit
must use normally open relays; two safety system OK (SSOK) lights must be mounted on the sides of the car and must be powered directly by the shutdown circuit; SSOK lights must serve as the logical AND of all shutdown circuit elements, except for the TSMS and dashboard BRB. The low-voltage rail was equipped with appropriate filters to avoid noise issues during high-current switching. The driver communicates with SCADA system using the dashboard interface that includes a graphical liquid crystal display (GLCD), system indicator LEDs, and various control buttons. The driver can monitor vehicle status such as IMD, BMS, high- and low-voltage systems, ready-to-drive, and so on. The drive also has the control to shut down the vehicle, initiate the pre-charge, and to put the vehicle in standby. The GLCD provides information on vehicle speed, battery charge level, motor temperature, and so on. To finalise the hardware system of the SCADA, the team went through comprehensive simulation testing and breadboard hardware testing with all peripheral sub-systems. After satisfactory test results the final PCB of the SCADA system was built as shown in Fig. 9. This board shows all the inputs and outputs signals, as well as the power connections.

The software system of the SCADA includes an EV operating system (EVOS) which was developed using model-view-controller (MVC) architecture [26, 27] to work with the SCADA hardware described above. The EVOS is broken into stages: BootTest, standby, pre-charge, energised, driving, and shutdown. The algorithm ensures that the code would not execute out of order, e.g. the car would not move if the driver presses the RTD button while in standby mode where the car has not yet executed the pre-charge and energised stages. The StageManager class is the core area where devices are “processed”. It contains functions for each device where controllers are accessed in order to perform a given task for that device, such as handling CAN messages when the appropriate event and task flags have been set. The main module contains the all necessary information to allow EVOS to function as a real-time operating system. It implements a simple super loop architecture. The super loop continuously checks for updated event flags and task flags and instructs the StageManager to call the appropriate processing functions. With this MVC architecture, the logical organisation of classes was firmly established. Some of the devices controlled by EVOS require many varying tasks to execute within their processing function, including all of the buttons on the dashboard. In order to differentiate the subset of actions that would have to be serviced for one device, TaskFlags were implemented for complex devices. This allowed for EVOS to capture both the device that triggered, and the specific task that needs to be handled in the processing function for that device. For example, when the pre-charge button is pressed on the dashboard, the Dash EventFlag and pre-charge TaskFlag are set. This results in the appropriate processing function to be called and the correct task for that device to be executed. The finite state machine of the vehicle operation is shown in Fig. 10.

During software development and integration, the team followed agile development practices [28, 29]. In order to verify the development team stayed updated on what was happening with EVOS, the team organised Scrum meetings. These occurred on a semi-daily basis where each member gave a short 2–3 min status update about what they did, what they planned to do, and what was blocking them. This also became effective in determining who on the team was putting the effort forth towards the issues assigned to them. Each SW issue represented features/bugs of EVOS, with each one assigned a priority level (critical, high, medium, low) and a duration estimate [small (1–2 days), medium (3–5 days), large (1 week)]. Issues would then be distributed to people on the development team based on how many issues they completed the previous sprint. The development process throughout the semester was broken up into week-long sprints. Each sprint required a sprint planning meeting, where everyone gave their input about what went well, and what issues should be completed during the upcoming sprint.

5 Battery management system

The FSAE electric hybrid vehicle guidelines were followed to design and build the BMS [10]. The battery had to be ≤300 V with a max of 120 V per segment and the max energy storage for each segment had to be <6 MJ. The BMS must constantly monitor all cell voltages, must monitor 30% of cell temperatures, must be able to shut the entire car down, and can only be restable manually during fault conditions. In addition, all voltage sense wires must be protected by fuses. To satisfy those requirements, each of the segment packs for the battery is made of 30 series-connected A123 AHP14 prismatic pouch cells. The A123 AHP14 batteries are 3.3 V each with a 14 Ah capacity. Each of the segment packs is nominally 100 V when charged. The entire battery is made of three segment packs connected in series to total 300 V. The maximum energy storage was 5.44 MJ.

The battery pack is constructed with heat sink plates lined with sheets of compressible silicone foam interleaved between each cell. That filler material is required by the FSAE hybrid rules. Each battery module is held together by end plates and four aluminium rods, one in each corner of the plate, pulling the cells together, putting pressure on all cells in the module. The requirement for pressure is to allow 8–12% expansion of the cells before the
error line which then sets off the safety circuit. When the safety for errors in the battery pack. If an error is detected, it triggers an limits on current for both input (charging) and output (powering the done by this BMS are the individual cell voltages, the total pack, temperature; fourth byte – average temperature; fifth and sixth bytes – maximum open cell voltage; and seventh and eighth bytes – low charge; second byte – state of health; third byte – highest temperature; fourth byte – average temperature; fifth and sixth bytes – max open-cell voltage; and seventh and eighth bytes – low open-cell voltage. For ID 0 × 421: first and second bytes – pack discharge current limit of the pack; third and fourth bytes – pack summed voltage; fifth and sixth bytes – pack current; and seventh and eighth bytes – average open cell voltage. Table 1 shows the visual of CAN message configuration.

6 System integration

System integration started with the mechanical structures that include the modified frame to accommodate the electric drivetrain, SCADA system, battery pack and BMS, GLV system, and cooling system. During testing and integration, constant communication between sub-teams was very crucial. The final assembly of putting all of the systems together was the most difficult due to the space constraints. The team had to insure reliable operation of the vehicle without sacrificing the integrity of the design or the safety of the system. The mechanical team installed the differential, motor, drive-chain, brakes, and chain guard in the rear. The mechanical team also made the housing for the battery and its cooling system, as well as installing the pump, manifold, and then running the cooling line. The drivetrain team started working on the DC high-voltage wiring to the motor controller and discharge box, AIRs connections, and the three-phase high-voltage wiring. The TSCB was the most important integration element of the drivetrain because it enables the high voltage, and engages and disengages the inputs to the motor controller under all circumstances. The TSCB was mounted in the low-voltage section of the battery box. The motor controller was mounted onto the frame such that there was enough room below it to account for the routing of the high-voltage wiring. All design and assembly information, including the wiring diagrams for the drivetrain system were stored on a common share drive.

The SCADA system integration process for EVOS involved incremental steps with hardware. These steps included working with the main brain PCB, laying out the dashboard, and building the GLV power distribution board. The main brain was tested with one sub-system at a time to verify each sub-systems functionality and compatibility. The dashboard was custom built to provide the driver with the vehicle status. The GLV system was designed and assembled to supply power to the BMS, cooling pump, fan, main brain, TSCB, and motor controller. With the intensive...
communication between sub-teams and the meticulous assembly and testing, the vehicle came together as shown in Fig. 3.

7 Laboratory and field test

The team designed and built a table to support the vehicle for operational testing in the laboratory. The table gave the rear wheels the ability to freely rotate during testing. The team developed a safety protocol manual for starting, running, and stopping the vehicle and all team members received training on that manual. The manual includes instructions for battery charging, GLV operation, high-voltage operation, safety circuit descriptions, engaging and disengaging drivetrain, and SCADA system monitoring. The step-by-step procedures helped the team test the vehicle several times in the laboratory. During testing, the SCADA team found an issue with an interrupt triggered by crosstalk noise. Appropriate filtering was added to resolve that issue. To identify and troubleshoot bugs in the SCADA software, a logging system was designed and implemented to record critical events and data as the EVOS ran. The logs were CSV-based to allow for ease of importing into programs such as NeoOffice, Google Sheets, and Excel. During testing, this log aided the team in determining what was occurring (and in what order) and to verify that the system was performing correctly. Table 2 shows a log output of dry run testing. The y-axis represents analogue-to-digital converter values that represent the acceleration pedal and the vehicle speed in kilometres/hour (km/h).

The drivetrain team noticed that the relays (AIRs) that were connected in parallel to 1000 μF capacitors were not engaging correctly. They would ‘chatter’ – rapidly engage and disengage while the capacitor charged. Only after the capacitor had completely charged did the AIRs relay properly engage. This behaviour was due to the capacitors drawing excess charging current, causing a drop in relay voltage. To fix this issue, the TSCB was modified in three locations to add diodes and resistors. The TSCB also needed to be modified to include 100 kΩ resistors across the two timing circuit capacitors. These were required to quickly discharge the capacitors so that a timer could be used relatively quickly if the car went to standby and then was pre-charged soon after the AIRs opened.

After the design team was satisfied with the laboratory test results, they took their car out for its first test drive. The team took the vehicle to an empty parking on campus, set up appropriate barriers and safety measures (students manning fire extinguishers at multiple locations). Team members took turns running laps of the parking lot, reaching sustained speeds of 40 km/h, which was software limited due to the physical constraints of the parking lot. The car continued to run with only minor issues that were addressed in the field for ~8 h across two separate testing sessions. The temperatures of the various parts in the cooling system (motor, controller, battery pack) remained well within tolerance the entire time. Fig. 14 shows the vehicle during a field test driving.

8 Conclusion

This paper presented a successful, end-to-end multidisciplinary project where the team designed, built, and test drove a FSAE EV. The team designed the following: a custom interactive system control board based on combinational logic to improve the safety of the drive train; a complete cooling system for the motor, controller, and battery pack; a complete hardware and software solution for the SCADA using the MVC architecture and agile development practices and scrum meetings; a BMS that accepted a sensor network for cell temperature, voltage, and current. System integration and laboratory testing were conducted. The FSAE EV design was tested through a series of test driving. The vehicle accelerated to and maintained 40 km/h without any significant issues.

Table 2  Log output of a dry run testing

| A | B         | C                      | D           |
|---|-----------|------------------------|-------------|
| 1 | Millis    | Calling Class          | Message     | Log Level   |
| 2 | 529       | MAIN                   | Bootup begin at 400 ms | LOG         |
| 3 | 2660      | MAIN                   | Bootup complete at 2660 ms, took 2260 ms | LOG         |
| 4 | 2682      | STAGE_MGR              | Started configuring STAGE_STANDBY | LOG         |
| 5 | 2683      | STAGE_MGR              | Stage: Standby | LOG         |
| 6 | 10,456    | STAGE_MGR              | Dash – TF_DASH_ENCODER | DEBUG       |
| 7 | 10,457    | GLCD_C                 | GLCD display mode advanced | LOG         |
| 8 | 17,017    | DASH                   | Dash - TF_DASH_PRECHARGE | DEBUG       |
| 9 | 17,018    | STAGE_MGR              | Started configuring STAGE_PRECHARGE | LOG         |
| 10| 17,024    | STAGE_MGR              | Stage: Precharge | LOG         |
| 11| 17,692    | STAGE_CON              | Full Numeric battery voltage 6423 | LOG         |
| 12| 17,698    | STAGE_CON              | 75 percent numeric battery voltage 4817 | LOG         |
| 13| 17,711    | STAGE_CON              | Current HV-bus numeric voltage 158 | LOG         |
| 14| 17,717    | UNITEK                 | Expected HV-bus numeric voltage: 4817, actual is: 158 | ERR         |
| 15| 17,723    | STAGE_CON              | Stage: shutdown | LOG         |

Fig. 13  Laboratory speed testing results
Fig. 14 Test driving

9 References

[1] Global EV outlook: Available at https://www.iea.org/publications/reports/globalevoutlook2019/. 2019
[2] Electric Vehicle Sales: ‘Facts and figures, Edison electric institute’, Washington, DC, USA, 2019
[3] Mokhtar, W., Duesing, P., Hildebrand, R.: ‘Integration of project-based learning (PBL) into mechanical engineering programs’, Int. J. Learn., 2008, 15, p. 8
[4] McInerny, S., Dai, Y.: ‘Basic vibration signal processing for bearing fault diagnosis’. 2016, 46, pp. 56–68
[5] Mokhtar, W., Duesing, P., Hildebrand, R.: ‘Integration of project-based learning (PBL) into mechanical engineering programs’, Int. J. Learn., 2008, 15, p. 8
[6] Wilkerson, S.A., Forsyth, J., Korpela, C.M.: ‘Project based learning using the robotic operating system (ROS) for undergraduate research applications’. ASEE Annual Conf. & Exposition, Columbus, Ohio, 2017, https://peer.asee.org/28768
[7] Wilkerson, S.A., Gadsden, S.A., Lee, A., et al.: ‘Board 64: ROS as an undergraduate project-based learning enabler’. ASEE Annual Conf. & Exposition, Salt Lake City, UT, USA., 2018
[8] Wilkerson, S.A., Gadsden, S.A., Cerreta, J., et al.: ‘Drone-based technologies used to assess modern farming practices in undergraduate research with project-based learning’. ASEE Conf., Buffalo NY, December 2018
[9] Unitek.: ‘Unitek Drives,’ 01 January 2019. Available at https://www.unitek-industrie-elektronik.de/images/pdf/BAMOCAR%20Digital/BAMOCAR-PG-D3_EN.pdf. Accessed 15 August 2019
[10] Wilkerson, S.A., Gadsden, S.A., Lee, A., et al.: ‘Board 64: ROS as an undergraduate project-based learning enabler’. ASEE Annual Conf. & Exposition, Salt Lake City, UT, USA., 2018
[11] Wilkerson, S.A., Gadsden, S.A., Lee, A., et al.: ‘Board 64: ROS as an undergraduate project-based learning enabler’. ASEE Annual Conf. & Exposition, Salt Lake City, UT, USA., 2018
[12] EMRAX.: 02 August 2018. Available at: http://emrax.com/wp-content/uploads/2017/10/user_manual_for_emrax_motors.pdf. Accessed 10 August 2019
[13] Unitek.: ‘Unitek Drives,’ 01 January 2019. Available at https://www.unitek-industrie-elektronik.de/images/pdf/BAMOCAR%20Digital/BAMOCAR-PG-D3_EN.pdf. Accessed 15 August 2019
[14] Bocci, D., Fraser, D., Gee, T., et al.: ‘2018 formula hybrid rules’. SAE Int. J. Altern. Powertr., 2019, 2, pp. 115–136
[15] Sossa, J.L., Magallán, G.A., Angelo, C.H.D., et al.: ‘Implementation of a supervisory control system for an electric vehicle’. 9th IEEE/IAS Int. Conf. on Industry Applications, Sao Paulo, Brazil, 2010
[16] Wilkerson, S.A., Gadsden, S.A., Lee, A., et al.: ‘Board 64: ROS as an undergraduate project-based learning enabler’. ASEE Annual Conf. & Exposition, Salt Lake City, UT, USA., 2018
[17] Wilkerson, S.A., Gadsden, S.A., Cerreta, J., et al.: ‘Drone-based technologies used to assess modern farming practices in undergraduate research with project-based learning’. ASEE Conf., Buffalo NY, December 2018
[18] Pedret, P.: ‘Control systems for high performance electric cars’. World Electric Vehicle Symposium and Exhibition (EVS32), Barcelona, Spain, 2013
[19] Jayasurya, D. C., Rankin, M., Jones, T., et al.: ‘Modeling and validation of an unbalanced LV network using smart meter and SCADA inputs’. IEEE 2013 Tencon - Spring, Sydney, NSW, Australia, 2013
[20] Lee, T.-K., Kim, Y., Stefanopoulou, A., et al.: ‘Hybrid electric vehicle supervisory control design reflecting estimated lithium-Ion battery electrochemical dynamics’. American Control Conf., San Francisco, CA, USA, 2011
[21] Leite, V.M., Palma, J.G., de Oliveira, F.H.: ‘Definition of a computing independent model and rules for transformation focused on the model-view-controller architecture’, World Acad. Sci. Eng. Technol., Int. J. Comput. Elect. Autom. Control Inf. Eng., 2017, 11, (2), pp. 244–251
[22] Zissis, D., Xidias, E.K., Lakkas, D.: ‘A cloud based architecture capable of perceiving and predicting multiple vessel behaviour’, Appl. Soft Comput., 2015, 35, pp. 652–661
[23] Lindsjom, Y., Sjoberg, D., Dingdyar, A., et al.: ‘Teamwork quality and project success in software development: A survey of agile development teams’, J. Syst. Softw., 2016, 122, pp. 274–286
[24] Rolland, K., Fitzgerald, B., Dingsoyr, T., et al.: ‘Problematizing Agile in the large: alternative assumptions for large-scale Agile development’. 37th Int. Conf. on Information Systems (ICIS 2016), Dublin, 11–14 December 2016
[25] Lee, T.-K., Kim, Y., Stefanopoulou, A., et al.: ‘Hybrid electric vehicle supervisory control design reflecting estimated lithium-Ion battery electrochemical dynamics’. American Control Conf., San Francisco, CA, USA, 2011
[26] Bossa, J.L., Magallán, G.A., Angelo, C.H.D., et al.: ‘Implementation of a supervisory control system for an electric vehicle’. 9th IEEE/IAS Int. Conf. on Industry Applications, Sao Paulo, Brazil, 2010
[27] Jayasurya, D. C., Rankin, M., Jones, T., et al.: ‘Modeling and validation of an unbalanced LV network using smart meter and SCADA inputs’. IEEE 2013 Tencon - Spring, Sydney, NSW, Australia, 2013
[28] Lee, T.-K., Kim, Y., Stefanopoulou, A., et al.: ‘Hybrid electric vehicle supervisory control design reflecting estimated lithium-Ion battery electrochemical dynamics’. American Control Conf., San Francisco, CA, USA, 2011