

Development of the KIIRA EV SMACK Master Controller Logic

Pauline Korukundo^{1*}, Sandy Stevens Tickodri-Togboa², Paul Isaac Musasizi³, Dennis Kibalama⁴, Richard Madanda⁵ and Victor Tumwine⁶

¹Department of Electrical and Electronics Engineering, Kiira Motors Project, Kampala, Uganda

²Department of Digital Communications, Kiira Motors Project, Kampala, Uganda

³Department of Mechanical Engineering, Kiira Motors Project, Kampala, Uganda

⁴Department of Electrical Engineering, Kiira Motors Project, Kampala, Uganda

⁵Department of Embedded Systems Engineering, Kiira Motors Project, Kampala, Uganda

⁶Department of Operations Management, Kiira Motors Project, Kampala, Uganda

Abstract

The growth in number and complexity of controlled electrical sub systems in the modern car has created a need for centralized information interchange point. The master controller provides this centralized unit. It interacts with all other networked control electronics to control the dynamic driving demands. In a hybrid vehicle, the master controller not only provides a communication interface, but also an efficient energy management, coordination and performance during the various vehicle-specific drive cycles. The control complexity at the master control unit is greater in the hybrid car because of the extra energy resources integrated into the vehicle.

This work presents the design, implementation and testing of the Kiira EV SMACK vehicle master control unit. In the Kiira EV SMACK, the master controller oversees the human machine interfaces, low voltage electronics, the motor and generator controllers, battery management and thermal management systems. A model based design approach was followed to implement the controller logic for initiation of startup of the low level controllers and implementation of a power source switching strategy. The switching strategy is based on speed demands and available energy resources (battery state of charge and fuel capacity).

Keywords: ECU; Kiira EV; Kiira EV SMACK; Vehicle electronic control units; Master control units; Hybrid

Introduction

The Kiira EV SMACK is a hybrid electric vehicle developed by the Kiira Motors Project to expand the range of her first prototype - the fully electric Kiira EV- as well as support her vision of being at the forefront of the emerging automotive industry in Uganda [1].

Hybrid vehicle powertrains are typically designed with an engine and at least one electric machine in a series, parallel or series - parallel configuration [2]. The SMACK is based on a series hybrid configuration with 64 lithium ion cells and a 4-cylinder engine-generator set connected via a converter to an electric motor. The electric motor is coupled directly to the front axle using a single gear ratio for propulsion. The engine extends both power assist and onboard charging roles.

The introduction of alternative fuel solutions into the vehicle has led to an increase in the number of vehicle subsystems especially in the powertrain, with each having a control unit that requires centralized control. The master control unit (MCU) provides this top level control logic. The extra energy resources in a hybrid vehicle create a need for the MCU to implement an efficient energy management module alongside its other roles.

This paper discusses major steps taken to implement the MCU logic for the Kiira EV SMACK using a model based design approach. A detailed account of the switching strategy employed for energy management is also given.

Design and Implementation

The roles of the MCU in the Kiira EV SMACK include initiating vehicle startup, monitoring vehicle operation, implementing a shutdown sequence and providing energy-, thermal- and fault management strategies. To achieve this, it communicates with the control units using a two-level architecture illustrated in Figure 1. The

architecture was realized using a use case analysis for each MCU signal as defined in Table 1.

System architecture and requirements

All the CAN-enabled control units were networked on a two-level architecture using the CAN2.0B version of the Bosch vehicle bus system. The components connected to the vehicle bus include a battery

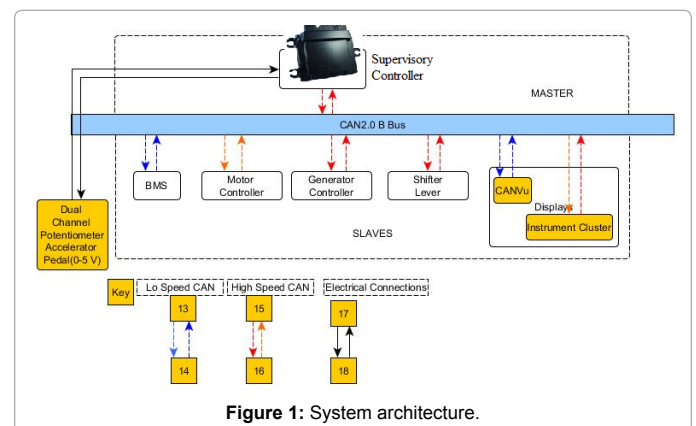


Figure 1: System architecture.

***Corresponding author:** Pauline Korukundo, Department of Electrical and Electronics Engineering, Kiira Motors Project, Kampala, Uganda, Tel: +256414545029; E-mail: pktpaulie@gmail.com

Received August 16, 2015; Accepted December 29, 2015; Published December 31, 2015

Citation: Korukundo P, Tickodri-Togboa SS, Musasizi PI, Kibalama D, Madanda R, et al. (2016) Development of the KIIRA EV SMACK Master Controller Logic. Adv Automob Eng 4: 127 doi:10.4172/2167-7670.1000127

Copyright: © 2016 Korukundo P, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Use Case	Use Case Description	Actors
M.C.U UC01: Initialize and Enable	Send 12 V signal Check fault status Check operational status Receive CAN messages Initiate Motor pre-charge sequence Check Battery State of Charge (SOC)	BMS Motor Control Unit
M.C.U UC02: Request for Torque	Receive Accelerator Pedal Position Determine Torque Requests Send Torque Requests to Motor control unit Calibration of Accelerator Pedal position to Torque Request	Accelerator Pedal Motor Control Unit
M.C.U UC03: Engine Startup	Send torque requests to engine-generator set Check Battery SOC Send Charge Signal to BMS	BMS Motor Control Unit

Table 1: Sample use case analysis for the master controller.

Control Unit	Specifications
Supervisory Controller: Woodward: ECM-5554-112-0904-C00-M	112 Pin Platform, Operating DC Voltage: 12 V Calibratable Memory: 512 K CAN 2.0B Channels: 3
Motor Controller: TM4 MΦTIVE: Series C060	Operating DC Voltage: 220-400 V Minimum Operating DC Voltage: 180 V Other Features: Four Quadrant Operation
Battery Management System: Orion BMS L4275D05	BMS supporting 72 Cells, Integrated low loss Passive Cell Balancing, Integrated error Detection and fault handling, Support OBD2 automotive Protocol Communication: CAN 2.0
Generator Control Unit: Rhinehart Motion Systems RMS-PM150DZ	Peak Power: 100 kW Continuous Power: 70 kW Continuous Output Current: 300 A Peak Output Current: 500 A
Shift Lever	Operating DC Voltage: 12 V Communication: CAN
CANvu Display	Communication: CAN 2.0 Active fault alarms and display
Accelerator Pedal	Operating DC Voltage: 12 V Sensor: Programmable Hall Effect Sensor Type: Dual Potentiometer

Table 2: Summary of component specifications.

management system (BMS), two motor control units, a shift selector and display units (instrument cluster and CANvu). A summary of the specifications for the control units connected to the master controller are given in Table 2.

The MCU logic was designed to receive battery parameters on the low speed CAN channel at speeds of 250 kbps to communicate with the motor controller, generator controller, instrument cluster and at 500 kbps with the electronic shift selector on the high speed channel. The MCU also controls various relays and contactors for the pre-charge sequence of the electric motors (drive motor and generator) and enables the high voltage connection to the DC-DC Converter. It also interacts with other components using analogue connections such as the accelerator pedal, body electronics and relay switches connecting the thermal management system.

Implementation

The design was implemented using the MotoHawk development tool built by Woodward to work with the MATLAB Simulink Environment. MotoHawk allows developers to write code that runs on MotoHawk-enabled electronic control units. It plays a key role in model based design, i.e., it enables code and model re-use through user defined libraries, facilitates rapid prototyping and testing on production grade hardware, as well as design verification through simulation [3].

It also features an option for auto generation of code documentation.

The selection of the Motohawk development tool was based on the benefits of model based design for a startup as discussed by Sameer and Jonathon [4] such as reduced production time and costs. The major steps followed in developing the software using MotoHawk include:

1. Model definition; splitting the complex control system into smaller components called models which can be designed and tested independently. The models could represent an entire subsystem control unit or may be functional models such as the Kiira EV SMACK energy management module. This simplifies development and fosters developer collaboration.

2. Implementation involves writing the control logic using Simulink blocks and Stateflow. MotoHawk features blocks that target particular Woodward hardware platforms. It also has vehicle specific blocks such as specific engine control, analog and digital I/O, CAN, calibration, fault management and diagnostics, and automated documentation.

3. Code generation is done using a compatible compiler such as Greenhills, CodeWarrior or GCC. This creates a file which can be flashed directly onto the hardware.

4. Calibration involves modifying block data at runtime using the MotoTune tool. Calibratable blocks can have default values which are edited as the generated code runs directly on the hardware, hence providing flexibility for values that are only determined at runtime. This step is not applicable on production hardware modules.

5. Flashing the target hardware with the generated code is the final step which is achieved with a Boot Key.

The implemented logic is as follows:

1. Drive Inputs: At startup, the MCU responds to drive commands from the drive inputs; shift selector, start button, accelerator and brake pedals. Based on the shift selector position and pedal position, it determines the vehicle state as idle, operational or shutdown. It also computes torque commands. A dual channel accelerator pedal provides a voltage of 0-5 V which is mapped to a position percentage, 0-100%. The MCU uses a pedal position index array to determine dynamic torque command values ($-Torque_{min}$ to $Torque_{max}$) using a torque-pedal map lookup table. The position can be mapped against the motor torque using a linear interpolation as illustrated in Figure 2 which is based on an electrical machine with a maximum rated torque of 157 Nm. Research also shows that accelerator pedals can be calibrated with braking and regenerative braking regions Boekel, Besseling, Nijmeijer [5] based on the vehicle performance target [6].

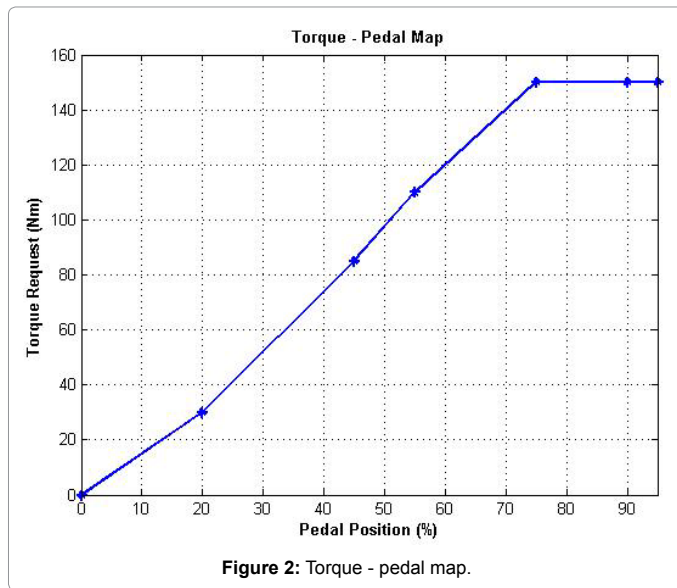


Figure 2: Torque - pedal map.

2. The drive selector communicates position messages over the high speed channel to the MCU which interprets these messages and sends them to the motor control unit. A torque request and position messages enable the motor control unit to respond with torque commands to the drive motor.

3. Battery Management System: The BMS monitors the battery cells for state of charge, battery temperature, battery voltage, V_{batt} , current, battery pack health and fault status. The MCU uses these parameters in motor operation, switching drive regimes, thermal management, fault mitigation, vehicle startup and shutdown.

4. Motor Control Unit: The motor controller and MCU follow a strict handshaking protocol to transition between motor states. There are seven motor states; Initialization, Standby, Startup, Operational, Tear Down, Close Down and *ReadyToShutdown*. In the Initialization state the MCU sends an *Enable* signal to the motor control unit and initiates the motor pre-charge sequence by closing the HV (high voltage) contactors. The motor system moves into the Standby state during which the motor control unit sends a *StandbyReady* signal to the MCU. The MCU ensures that the shift selector is in the Park Position and sends 0 Nm Torque Request. In the Startup state, it ensures that the high voltage at the motor contactors is equal to the battery voltage, V_{batt} reported by the BMS. If $V_{batt} \geq 180$ V, the minimum operating voltage of the motor, and the pre-charge sequence is complete it sends a *Startup Ready* signal to the motor control unit. When the motor goes into the Operational state, it readily responds to torque requests and drive selector changes between Reverse, Drive and Neutral. The MCU initiates the shutdown sequence by sending a shutdown request and deactivating the Enable signal. The above control logic was implemented using State flow.

5. Generator Control Unit: There are three use cases implemented by the MCU for this module; Initialization, Throttle Request and Monitoring. The Initialization use case is characterized by the MCU sending an Enable signal to initialize the engine-generator set start up, a Crank Engine signal and disabling torque requests. It then computes and sends throttle requests as summarized in Figure 3. When running, the MCU logs status information from the generator control unit.

6. The CANvu display indicates drive state information, motor

and battery parameter information as well as fault information from the MCU. This unit has the capability to provide fault management i.e., display information on how to correct and clear faults.

Energy Management and Drive Regimes

The Kiira EV SMACK powertrain drive mode selection is based on three drive regimes: Purely electric, Engine only and Hybrid. Energy resources are dynamically drained and replenished during vehicle operation, the MCU decides whether to implement a switching strategy or to maintain the current drive regime. In the EV mode, only the battery powers the electrical machine while in engine only mode, the engine powers it. The engine-generator configuration is designed to charge the battery pack in this mode. When the switching strategy is well-designed, the purely electric mode operates at low start/stop speeds, the engine is only run at maximum efficiency, and the hybrid mode provides range extension benefits.

- **Purely Electric:** In this regime the final drive is powered by batteries only. No engine charging is possible. This drive regime is operable when the battery SOC is above 80% and vehicle speed is below 50 km/hr. At such low speeds the power requirements can be sustained by the battery system. 50 km/hr is chosen because it is the speed limit for the Kampala city drive cycle, therefore operating the drive train at zero tail pipe emissions is desirable. When the fuel level is below 8%, the engine cannot operate on an empty tank; therefore the only possible drive regime is purely electric. Figure 4 shows the rules which govern vehicle operation in this regime.

- **Engine Only:** In this mode, the engine powers the vehicle; the battery system is disabled by opening the battery contactors. This mode is engaged at vehicle speeds above 50 km/hr and fuel capacity above 8% to ensure engine is not started with an empty tank. At such speeds, the power requirements are high and therefore, the engine can be operated in the efficient operating regions. If the battery SOC is below 20%, the SC enables battery charging to avoid draining the battery beyond this limit below which it has a degraded performance.

- **Hybrid Regime:** In this mode, the engine and battery systems provide power either simultaneously or interchangeably. This drive regime can sustain the vehicle operation at very high speeds (above 120 km/hr) and hill climbing.

The detailed description of energy management and drive control regimes is summarized in Table 3.

The management of the drive-train power sources is also based on two (2) operating modes;

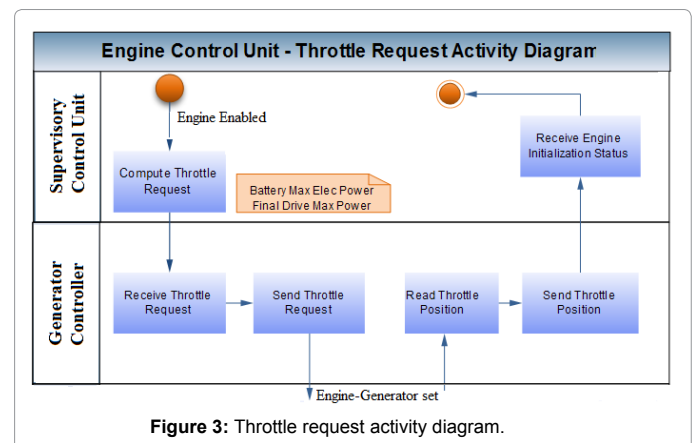


Figure 3: Throttle request activity diagram.

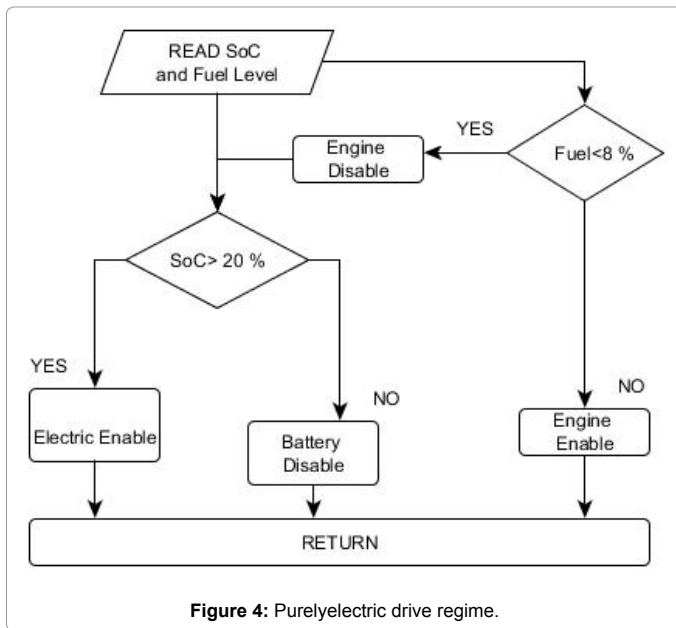


Figure 4: Purelyelectric drive regime.

Speed	SOC	Fuel Level	Mode
<50 or 50<s<120 or >120	<20%	<8%	OFF
<50	SOC>80	<8%	ELECTRIC
50<s<120	SOC>80	<8%	ELECTRIC
>120	SOC>80	<8%	ELECTRIC
<50	<20%	>8%	ENGINE-GENERATOR
50<s<120	<20%	>8%	ENGINE-GENERATOR
>120	<20%	>8%	ENGINE-GENERATOR
50<s<120	SOC>80	>8%	ENGINE-GENERATOR
>120	SOC>80	>8%	HYBRID
<50	SOC<80	>8%	ELECTRIC

Table 3: Drive regimes based on current powertrain parameters.

1. Automatic Control Mode: In this mode, the selection of the active optimal drive regime is based on rule-based deterministic control taking current battery SOC, engine fuel capacity and vehicle speed metrics upon which the decisions on electric only, engine only or hybrid drive regimes are made.

2. Manual Control Mode: In this mode, control is switched from the SC to the driver. The driver can then select the drive regime as desired.

Thermal Management

During operation, the heat generated by the machines in the engine bay needs to be extinguished. A thermal management module employing pumps, fans and radiators is used. The mechanical system is connected using relay switches to the MCU. The MCU determines when to turn on/off these auxiliary units based on a predefined threshold value. This maintains an optimal temperature for the machines in the engine bay and cells in the battery pack. Two independent systems are used; one for the batteries and another for the engine and motors.

An air cooled system is used to keep the batteries at optimum operating temperature. The system utilizes cooling ducts that conduct cool air to the battery bank. The flow of the air is accentuated by a blower/fan controlled by the MCU. The development of the thermal control strategy was targeted towards a tropical climate where

temperatures range from 15-35°C. As such, cooling the batteries to a desired temperature rather than battery heating was the core focus of this module. The control strategy for the cooling of the battery pack is an ON/OFF strategy where the fan is switched on once the battery temperatures reported by the BMS exceeds a set threshold, $T_{batt,max}$. The MCU controls the Fan Enable Signal of the BMS by switching ON/OFF of a relay.

The fans continue running until the temperature of the battery pack goes below the $T_{batt,max}$ temperature threshold. The fans run at a single speed and do not support PWM functionality. Possible avenues to improve the functionality of this control strategy of thermal management include:

1. Thermal management for both low and high temperature thresholds hence battery warming and cooling so that the vehicle can perform optimally in diverse weather conditions.

2. PWM Support for the Fan operation. This improves functionality by providing different fan speeds based on the temperature of the battery pack. A large difference between battery pack temperature and predefined threshold would imply a high fan speed whereas a small difference between battery pack temperature and predefined threshold would imply a low fan speed.

System Tests

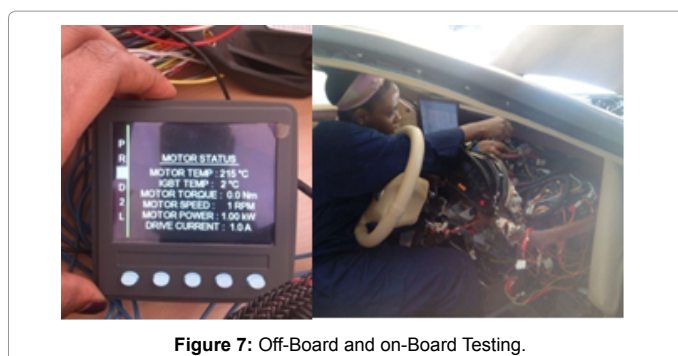
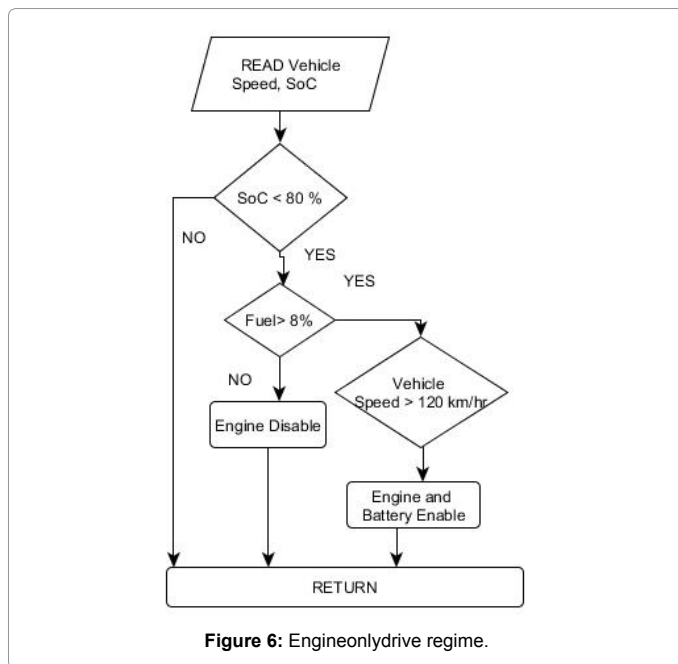
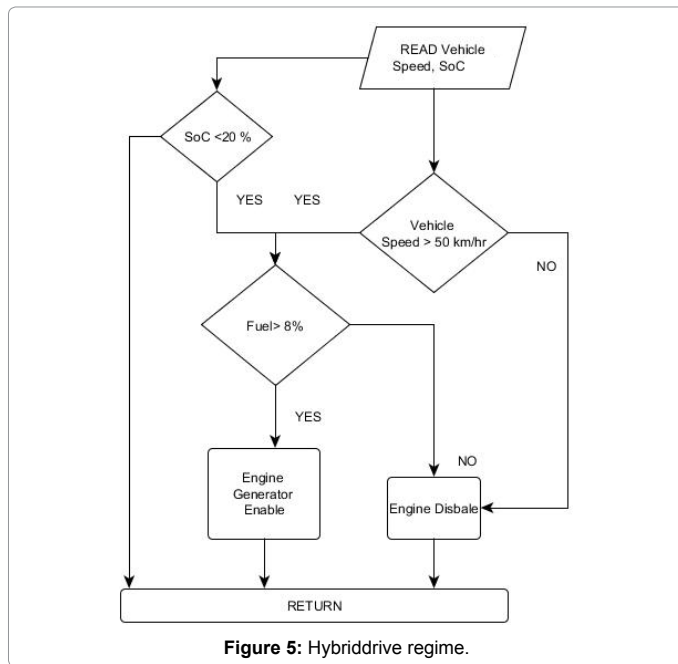
Thorough V&V (Validation and Verification) tests were carried out to ensure conformity with design requirements and component manufacturer recommendations minimize costly re-design/repairs and eliminate unexpected system behavior. These tests were carried out both off-board and on-board on each subsystem connected to the MCU. The off-board tests were made on both isolated and integrated sub systems and in some cases were characterized by dummy input data to simulate various operational conditions. This set of tests was fundamental in both design validation and verification while the on-board tests mainly targeted verification concerns. The on-board tests were based on real load data with all system components integrated into the vehicle. Table 4 shows a test specification used to carry out functional tests on the MCU (Figures 5-7).

Conclusion

The Master Controller is a module central to interfacing and controlling different vehicle control units for synchronous operation. This paper has illustrated the major considerations taken while implementing the firmware on this module for the Kiira EV SMACK hybrid vehicle. The design utilizes the typical two-level control architecture for the CAN based control units and a model based software design approach making it highly scalable. Future prospects include having body electronics controlled by the controller and addition of smart features such as park assist and collision avoidance to this module.

Test Case	Test Procedure	Expected Outcome	Results
Acceleration	<ul style="list-style-type: none"> Step on the Brake Pedal, Place Shift Lever in R. Depress the Accelerator Pedal and Observe Motor Speed and Rotation Direction Depress the Accelerator Pedal with Shift Selector in N position. Observe Motor Speed and Rotation Direction 	<ul style="list-style-type: none"> Motor Speed is varies between- 20000-20000 rpm Motor Speed is observed to vary between- 20000-20000 rpm 	Test Passed

Table 4: Summary of test specification for acceleration test case.



Acknowledgements

The Kiira Motors Project is funded by the government of the Republic of Uganda. The authors are thankful for all the funding that has supported the work leading to the realization of the Kiira EV SMACK vehicle. The entire participating research and support team on this project is appreciated for their contribution.

References

1. Musasizi PI, Togboa TSS, Matovu F (2014) Design and Implementation of Uganda's First Hybrid Car Concept. European Electric Vehicle Congress, Belgium.
2. Larminie J, Lowry J (2003) Electric Vehicle Technology Explained. West Sussex, John Wiley & Sons Ltd., England.
3. http://mcs.woodward.com/support/wiki/index.php?title=Main_Page
4. Prabhu SM, Friedman J, Smith P (2001) Best Practices for Establishing a Model-Based Design Culture. The MathWorks Inc.
5. van Boekel JJP, Besselink IJM, Nijmeijer H (2015) Design and realization of a One-Pedal-Driving algorithm for the TU/e Lupo E. EVS28 International Electric Vehicle Symposium and Exhibition, Kintex.
6. Kim D, Peng H, Bai S, Joel M (2007) Maguire Control of Integrated Powertrain with Electronic Throttle and Automatic Transmission. IEEE Transactions on Control Systems Technology 3: 474-482.