

## Research Article

# Development of an Android OS Based Controller of a Double Motor Propulsion System for Connected Electric Vehicles and Communication Delays Analysis

Pedro Daniel Urbina Coronado,<sup>1</sup> Horacio Ahuett-Garza,<sup>2</sup> Vishnu-Baba Sundaresan,<sup>3</sup> and Ruben Morales-Menendez<sup>4</sup>

<sup>1</sup>Center for Innovation, Design and Technology, Tecnológico de Monterrey, E. Garza Sada 2501, 64849 Monterrey, NL, Mexico

<sup>2</sup>Department of Mechanical Engineering, Tecnológico de Monterrey, E. Garza Sada 2501, 64849 Monterrey, NL, Mexico

<sup>3</sup>Department of Mechanical & Aerospace Engineering, The Ohio State University, 201 W 19th Avenue, Columbus, OH 43210, USA

<sup>4</sup>Research and Graduate Program, Tecnológico de Monterrey, E. Garza Sada 2501, 64849 Monterrey, NL, Mexico

Correspondence should be addressed to Pedro Daniel Urbina Coronado; [pd.urbina.phd.mty@itesm.mx](mailto:pd.urbina.phd.mty@itesm.mx)

Received 6 June 2014; Revised 15 October 2014; Accepted 15 October 2014

Academic Editor: Kang Li

Copyright © 2015 Pedro Daniel Urbina Coronado et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Developments of technologies that facilitate vehicle connectivity represent a market demand. In particular, *mobile device* (MD) technology provides advanced user interface, customization, and upgradability characteristics that can facilitate connectivity and possibly aid in the goal of autonomous driving. This work explores the use of a MD in the control system of a conceptual *electric vehicle* (EV). While the use of MD for real-time control and monitoring has been reported, proper consideration has not been given to delays in data flow and their effects on system performance. The motor of a novel propulsion system for an EV was conditioned to be controlled in a wireless local area network by an ecosystem that includes a MD and an electronic board. An intended accelerator signal is predefined and sent to the motor and rotational speed values produced in the motor are sent back to the MD. Sample periods in which the communication really occurs are registered. Delays in the sample periods and produced errors in the accelerator and rotational speed signals are presented and analyzed. Maximum delays found in communications were of 0.2 s, while the maximum error produced in the accelerator signal was of 3.54%. Delays are also simulated, with a response that is similar to the behavior observed in the experiments.

## 1. Introduction

Transportation systems of the future will likely rely on some form of autonomous driving capabilities from vehicles [1]. Potential benefits of autonomous driving technology include the reduction or elimination of accidents, improved traffic flow, and increased energy efficiency [2, 3]. Vehicle connectivity is considered to be a key technological feature of reliable and universally autonomous driving. A “connected vehicle” is capable of communicating and interacting with other devices, vehicles, infrastructure, and remote servers [4, 5]. The recent passing of *vehicle to vehicle* (V2V) connectivity laws in Europe and the USA [6] is only the latest of a series of events that highlight the importance that connectivity has for the transportation industry.

*Mobile devices* (MD), such as smartphones and tablets, use *operating systems* (OS) that represent a potential enabling technology for connected vehicles. MD possess features and characteristics that may be extremely useful for such applications. For example, MD offer a wide catalogue of connectivity options, such as Wi-Fi, Bluetooth, NFC, and 4G LTE [7], provide location awareness as a key feature [8, 9], and provide built-in useful sensors like accelerometers and video cameras [10–12]. Features that are inherent to MD, such as customizable and upgradable user interfaces, have become highly desirable for vehicle operation simply because people have become familiarized with them [13–15].

There are many examples of the use of MD technology in automotive applications. For example, the use of smartphone OS in infotainment and navigation subsystems has been

reported [16]. Also, recent products such as Android Auto and CarPlay allow a safe use of the phone features inside the car. Android Auto apps must be synchronized with installed phone apps and interact mainly with navigation and infotainment services. However, the potential of the use of MD in the vehicle environment extends beyond infotainments. Advantages of including a smartphone in the energy management systems of electric vehicle [17] and hybrid electric vehicles [18] have been presented. There are also examples of the use of smartphones in critical applications. For example, smartphones have been connected to the *on-board diagnosis* (OBD) network to extract information of potential risks on the road [19]. Other studies have explored the inclusion of a smartphone in a motorcycle helmet to obtain information and produce actuation with the objective of enhancing safety [20].

In spite of all their capabilities and the enthusiasm with which they have been used in transportation applications, there are important technological issues that need to be addressed before mobile devices can be introduced as a reliable enabler of connected vehicle capabilities. In particular, delays in communication and the impossibility to rigorously meet deadlines in scheduled tasks are factors that affect system performance. The studies discussed before have not addressed these issues. In particular, data processing for decision making and control of autonomous vehicles in future transportation systems will require real-time capabilities.

Current MD OS are optimized for telecommunications and energy savings but not for critical real-time tasks [21]. The use of unmodified smartphone OS to control mechatronic systems in real-time depends on the nature of the application. Common sense dictates that critical functions, such as the operation of antilock braking system in a vehicle whose faulty operation can result in serious damage, are beyond the capabilities of these systems. However, there are other possible applications in which potential damage can be prevented and for which the risks associated with the use of MD OS may be offset by their advantages. For example, applications in EVs have been reported for personal transportation [22], wheelchairs [23], and small unmanned electric vehicles [24]. Although research is currently underway to modify Android OS to produce a real-time operating system [25–27], there is still an open case about which subsystems can use and exploit the advantages of current OS [28] in real-time applications, particularly those in a vehicle. Of particular interest are the effects of the delays on the system being controlled, an issue that is critical for establishing the feasibility of using a MD OS for a given purpose.

With the ultimate goal of facilitating vehicle connectivity, this work explores the use of an *Android based control ecosystem* (ABCe) to manage critical functions within the propulsion system of an experimental, task-oriented electric vehicle of modular design that is currently under development, called EvTEC. This work presents the basic architecture of the ABCe, a modular controller that incorporates a MD with unmodified Android OS that communicates with an embedded system. Special attention is paid to the measurement of the delays occurring in the control system. For this reason, the ABCe is evaluated through the measurement of the signal

delays and their effect on performance of the propulsion system of the vehicle. The effects of the delays on the electric motor acceleration signal are quantified, from which the implications of the use of the MDOS are discussed.

The paper is arranged as follows. Details of the vehicle and propulsion system are provided in Section 2. Section 3 describes the Android based control ecosystem and the experimental setup. Section 4 presents the results of the experiments. Section 5 provides a discussion and remarks obtained, while Section 6 presents a simulation of the delays and a comparison with experimental results. Future work and conclusions of this study are shown in Sections 7 and 8, respectively.

## 2. The Electric Powertrain of EvTEC

The modular controller presented in this paper was developed to manage the powertrain of EvTEC, an electric vehicle that is currently under development. The goal of the EvTEC project is to design and build a utilitarian electric vehicle whose design is characterized by modularity, energy efficiency, and connectivity. The basic concept is shown in Figure 1. At the current stage, a prototype vehicle is under construction to serve as a test bed for the development of technology that will be required to achieve the intended design characteristics. The open architecture Android based controller is an example of such technological developments. The propulsion system of this vehicle is composed of a  $\text{LiFeMnPO}_4$  battery pack with a 4.2 kWh storage capacity and a powertrain composed of two electric motors connected to a gear train.

As mentioned before, energy efficiency in an electric vehicle can be improved by optimizing energy management inside the vehicle according to the foresighted driving conditions or as a byproduct of reduced vehicle weight, which in turn is a consequence of improved safety capabilities [2, 3]. The design goals for the ABCe are thus to provide connectivity with the user and the infrastructure and to manage power flow within the system in such a way that energy use is optimized.

Details of the powertrain architecture are shown in Figure 2. The shaft of the motor/generator A (M/G A) is connected to one of the lateral gears on differential 1 (D1). The shaft of the motor/generator B (M/G B) is connected to the other lateral gear on D1. The rotation of the lateral gears in D1 move the planetary gears inside D1 and the carrier. The carrier of D1 is meshed with a gear reduction (R). R is meshed with the carrier of differential 2 (D2) which is connected to the shaft of the wheels and provides traction. Lock 1 (L1) and lock 2 (L2) can fix the shaft of M/G A and M/G B, respectively.

In operation, the control will coordinate the motors response to obtain the highest combined efficiency according to the torque and speed requirement of the powertrain. The idea being to make the best use of the energy stored in the battery pack to extend the driving range of the vehicle.

The ABCe is used to control and manage the electric powertrain, as illustrated in Figure 3.

As a key element of the control system, the MD is a useful input/output user interface(UI), which enables the

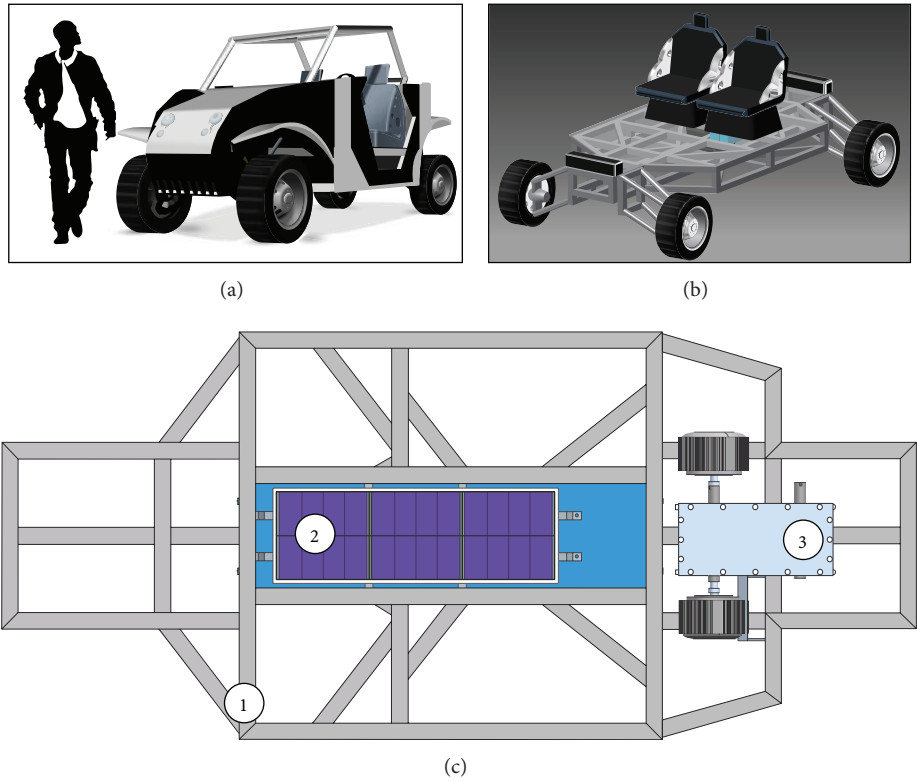


FIGURE 1: EvTEC concept. (a) Relative size with human being. (b) Platform structure with location of tires and seats. (c) Platform structure (1), battery pack (2), and propulsion module (3).

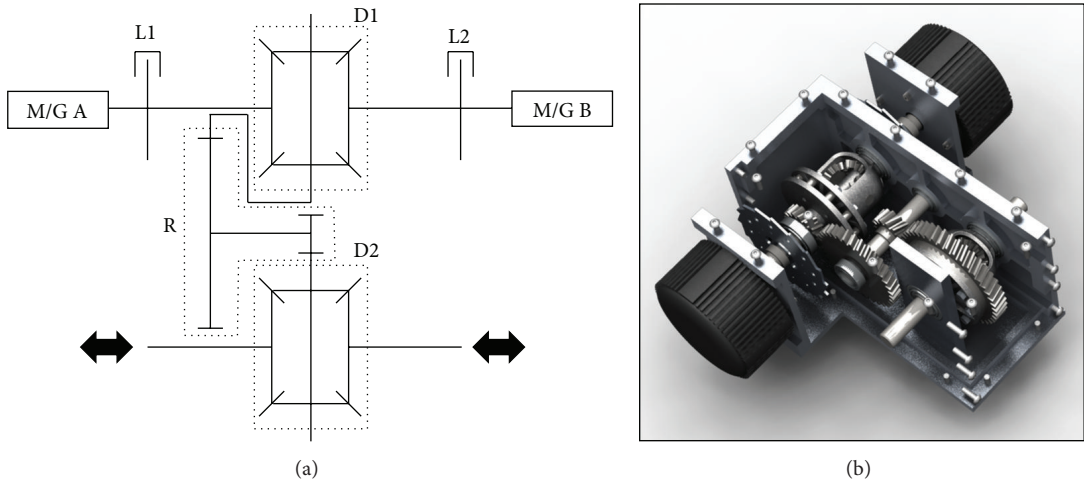


FIGURE 2: (a) Schematic of the gears, shaft, and actuators in the proposed powertrain and (b) propulsion system.

connection of the powertrain to the external world. The MD OS makes for a modular controller in that it is portable, customizable, and upgradable. This work uses these characteristics to develop an app that can be used to control the powertrain from a tablet or a phone.

### 3. The Android Based Control Ecosystem (ABCe)

The primary goal of the ABCe is to control the vehicle's forward movement. To enable this, the system was designed

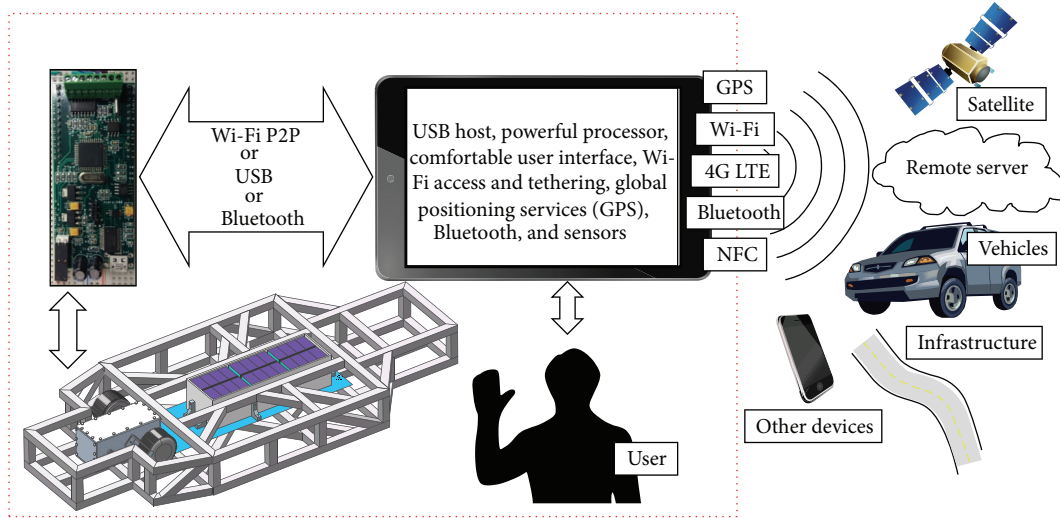


FIGURE 3: Android based control ecosystem (ABCe) applied to the connected electric vehicle.

to include an electronic board that can communicate with a MD using Bluetooth, USB cable, or *wireless local area network* (WLAN) connection. The particular connection through WLAN enables a *drive over wireless* (DoW) application, which eliminates the need for physical cable and provides superior transfer rates in comparison with a Bluetooth connection. This idea is in tune with the modular and customizable concept because it enables the control of the vehicle from anywhere, inside or outside of the passenger compartment.

The core of the electronic board used in this research is a P8X32A microcontroller. This microcontroller is made up of 8 independent cores of 32 bits each that share clock and I/O resources. The electronic board has 16 general use I/O digital channels and 8 analog input channels.

Figure 4 shows a schematic of the connections of the electronic board. The board controls the flow of accelerator signals (D I/O 0 and D I/O 4), regenerative braking signals (D I/O 1 and D I/O 5), and the signals to reverse the rotational speed (D I/O 2 and D I/O 6) of the motors. The electronic board also receives the signals from sensors and can measure the speed of the motors (D I/O 3 and D I/O 7), the current and voltage drawn from the battery (AI 0 and AI 1), the accelerator pedal signal (AI 6), and the braking pedal signal (AI 7).

The electronic board can be connected to the MD to WLAN, Bluetooth, or USB through the specialized transmitter and receiver pins (Tx and Rx in Figure 4). In this case, WLAN connection is used with an additional adapter, a RN XV Wi-Fly module. The MD used in this work is a first generation Nexus Tablet running Android 4.4.

The next section presents the details of the signal conditioning implemented for the control of the motor accelerator signal and the motor rotational speed sensing, as well as the details of the communication between the MD and the electronic board. Effects of the generated delays in communication over the speed of one of the motors already connected to the gear box are also discussed.

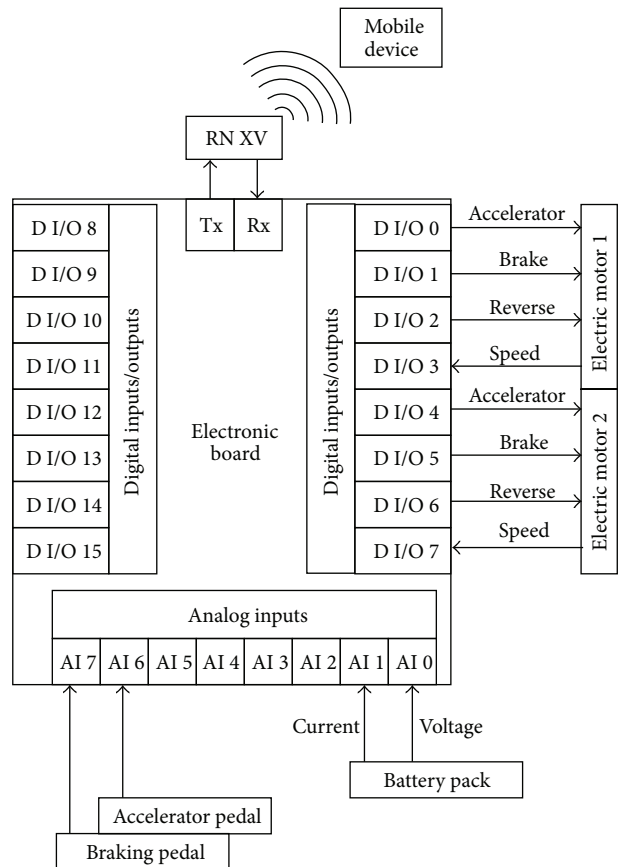


FIGURE 4: Input and output signals managed by the electronic board.

**3.1. Control of the Accelerator Signal.** In this control, the MD generates a command for the accelerator, which can be either produced by the user by an autonomous driving algorithm

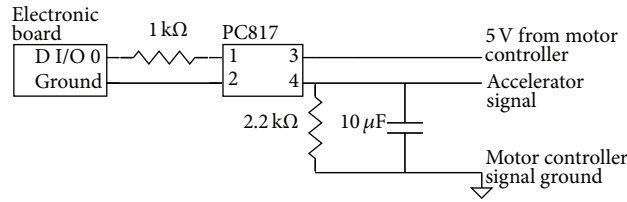
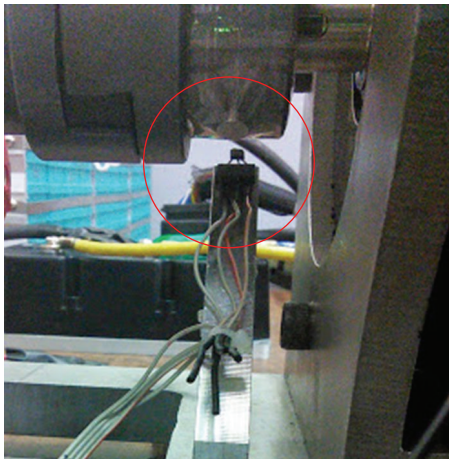
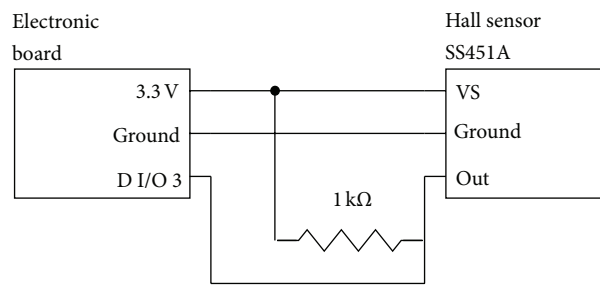


FIGURE 5: Accelerator signal circuit generated from a PWM signal coming from PIN D I/O 0 in the electronic board.



(a)



(b)

FIGURE 6: (a) Installation of hall sensor and magnet in the shaft of the electric motor and (b) circuit to measure the rotational speed.

or received remotely by the connectivity options that are free. The command for the accelerator is transmitted to the electronic board at a programmed writing rate. The command for the accelerator is received by the electronic board Rx pin, at the programmed writing rate  $\pm$  delays, which occur with a random nature. The command for the accelerator is translated into a *pulse width modulated* (PWM) signal which is conditioned to produce the accelerator signal: a voltage from 0 to 5 V that is received by the electric motor controller. The circuit schematic showing the implementation of the signal conditioning is presented in Figure 5.

**3.2. Sensing of the Electric Motor Speed.** The rotational speed of the electric motor is sensed by attaching a magnet to the electric motor shaft. A Hall Effect sensor is located in the near zone. When the sensor detects the magnetic field of the magnet, it acts like a switch, producing digital pulses. Figure 6 shows the implementation of the sensor and the schematic of the circuit to measure the rotational speed. The electronic board translates the signal received from the sensor to a rotational speed. The information of the rotational speed is transmitted through the Tx pin to the MD at a programmed Tx rate. The MD reads this information at the Tx rate  $\pm$  delays, which are inherent to the MD in the control and monitoring system.

**3.3. Experimental Setup.** Delay data was obtained from direct measurements of the electric motor performance. The electric

motor controller transmits the energy stored in the battery pack to the electric motor. The motor controller receives the acceleration signal from the electronic board as explained before. Figure 7 shows the experimental setup.

The RN XV Wi-Fly module in the electronic board is working as the host of the WLAN connection. Communication is performed under IEEE 802.11 g standard, working in the 2.4 GHz band. The baud rate was set at 11520. The operation range is 38 m. For tests purposes, authentication and encryption were left open.

#### 4. Measurement Results

An accelerator signal is generated in the MD ( $\alpha_W$ ) as function of time in seconds ( $t$ ). The function is of the type  $A \cdot \sin(B \cdot x + C) + D$ , a sine function where the amplitude is  $A$ , the period is  $2 \cdot \pi / B$ ,  $C$  is the phase shift, and  $D$  is the vertical shift. In this particular case, a sine function as shown in (1) was implemented. Consider

$$\alpha_W(t) = 20 \cdot \sin(1.25663706 \cdot t + \pi) + 118. \quad (1)$$

The electronic board receives the acceleration signal filtered by a floor function ( $\alpha_{Rx}$ ), as shown in (2). Consider

$$\alpha_{Rx}(t) = \lfloor 20 \cdot \sin(1.25663706 \cdot t + \pi) + 118 \rfloor. \quad (2)$$

TABLE 1: Definition of delays in each of the sampling loops in MD and electronic board.

	Schedule in Test 1 (s)	Schedule in Test 2 (s)	Actual sample period (s)	Delay (s)	
Writing loop in MD	$\tau_{Ws} = .02$	$\tau_{Ws} = .01$	$\tau_{Wa}(n)$	$\delta_W(n) = \tau_{Wa}(n) - \tau_{Ws}$	(*)
Receiving loop in electronic board (Rx)	$\tau_{Ws} = .02$	$\tau_{Ws} = .01$	$\tau_{Rxa}(n)$	$\delta_{Rx}(n) = \tau_{Rxa}(n) - \tau_{Ws}$	(**)
Transmitting loop in electronic board (Tx)	$\tau_{Txs} = .025$	$\tau_{Txs} = .01$	$\tau_{Txa}(n)$	$\delta_{Tx}(n) = \tau_{Txa}(n) - \tau_{Txs}$	(***)
Reading loop in MD	$\tau_{Txs} = .025$	$\tau_{Txs} = .01$	$\tau_{Ra}(n)$	$\delta_R(n) = \tau_{Ra}(n) - \tau_{Txs}$	(****)

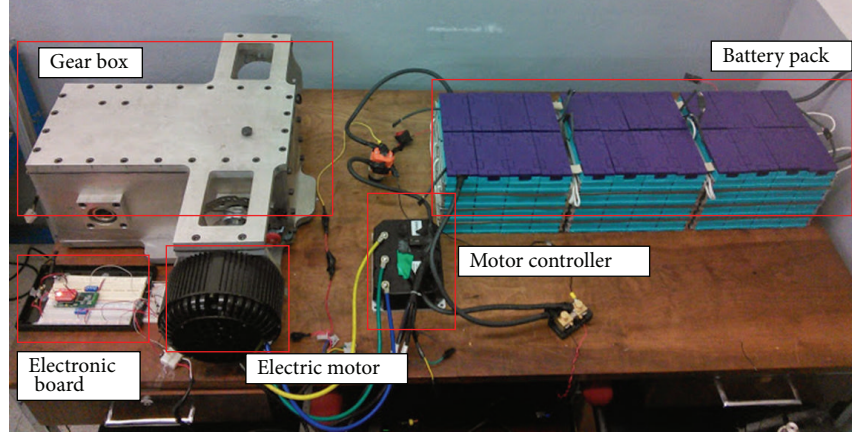


FIGURE 7: Components of the experimental setup.

Finally, the acceleration signal translated to a PWM percentage ( $\alpha_{PWM}$ ) is given by (3). Consider

$$\alpha_{PWM}(t) = \frac{\alpha_{Rx}(t) \cdot 25}{100}. \quad (3)$$

Experiments were performed at two different sample rates. In the first set of experiments (Test 1), a writing rate of 20 ms was programmed in Android to send the command for the accelerator signal. In return, a Tx rate of 25 ms was scheduled in the electronic board to send the rotational speed of the electric motor to the MD. In a second set of experiments (Test 2), the sample rates for writing in Android and the Tx in the electronic board were scheduled at 10 ms both. Each test was run 3 times. The mobile OS can schedule operations; however, it cannot guarantee the rigorous accomplishment of deadlines. The delays ( $\delta$ ) incurred in the sending and receiving periods ( $\tau$ ) for each sample ( $n$ ) are calculated using the equations shown in Table 1.

Figure 8 shows a typical plot of the delays in the sampling period for the experiments run here. The plot of Figure 8 corresponds to the writing loop delays ( $\delta_W$ ) in the MD for the run 1 of the Test 1.

The magnitude and variation of the delays are represented graphically in boxplots as shown in Figures 9(a) and 9(b), where data from the sampling loops for Test 1 and Test 2 are shown. The left whisker represents the 2-percentile mark while the right whisker represents the 98-percentile mark. The left and right extremes of the box represent the 25- and 75-percentile marks, respectively. The bar inside the box

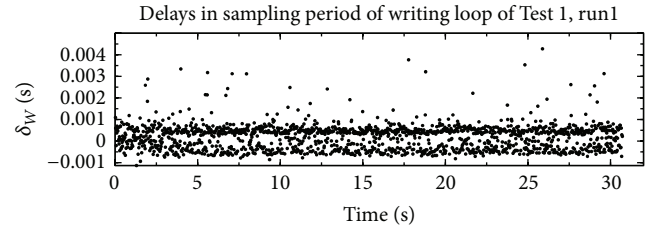


FIGURE 8: Plot with the delays measured in the sample period for the writing loop of Test 1.

represents the median. The star represents the mean and the circles represent outliers. There were 1500 samples in each run, so the boxplots, which shows data of the three runs, contain 4500 samples each. The boxplots reveal that, in average, the delays are close to zero; however the distribution, mainly of the reading and receiving loops, presents large spreads.

The delays introduced by the communication with the MD have an effect in the transmitted data. For example, the effect of the delays in the accelerator signal is displayed in Figure 10. The plots show the accelerator signal generated in the MD as a thin line and the accelerator signal received by the electronic board in markers. The three runs of the tests are displayed. Percentage of errors ( $e_W$  and  $e_{Rx}$ ) between the ideal accelerator signal ( $\alpha_W$  and  $\alpha_{Rx}$ ) with respect to the real values sent and received ( $\alpha_{Wa}$  and  $\alpha_{Rxa}$ ) are calculated with (4) where  $t_n$  is the time in which sample  $n$  happened.

TABLE 2: Average, maximum, and minimum percentage of error between the ideal accelerator signal and the transmitted and received accelerator signal.

	Run 1		Run 2		Run 3	
	Writing ( $e_W$ )	Reception ( $e_{Rx}$ )	Writing ( $e_W$ )	Reception ( $e_{Rx}$ )	Writing ( $e_W$ )	Reception ( $e_{Rx}$ )
	Test 1					
Average % of error ( $e$ )	-0.00050703	0.001664	0.000115	-0.00307177	-0.00065	-0.00437
Maximum % of error ( $e$ )	0.028597682	2.290076	0.01548	1.941747573	0.043425	1.941748
Minimum % of error ( $e$ )	-0.02858194	-1.5873	-0.01425	-1.57480315	-0.04398	-3.53982
	Test 2					
Average % of error ( $e$ )	-0.00147714	-0.00929	-0.00058	-0.00599	-0.0003124	0.001745285
Maximum % of error ( $e$ )	0.039644728	1.010101	0.02188	1.73913	0.02372237	2.272727273
Minimum % of error ( $e$ )	-0.03899027	-2.63158	-0.0224	-1.02041	-0.0230413	-1.88679245

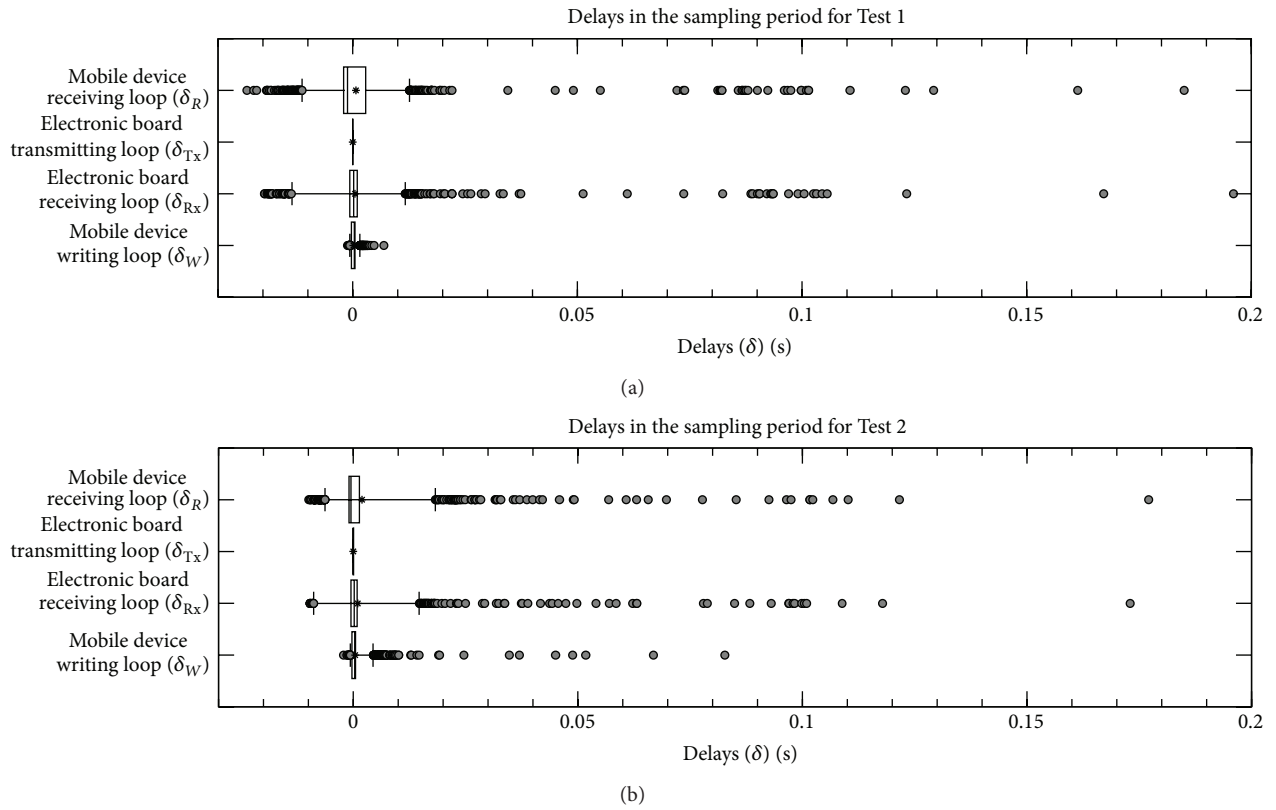


FIGURE 9: Boxplots for distribution of delays in (a) Test 1 and (b) Test 2.

Table 2 summarizes the maximum, minimum, and average percentage of error obtained in each run. Consider

$$\begin{aligned}
 e_W(n) &= \frac{[\alpha_{Wa}(n) - \alpha_W(t_n)] \cdot 100\%}{\alpha_{Wa}(n)}, \\
 e_{Rx}(n) &= \frac{[\alpha_{Rxa}(n) - \alpha_{Rx}(t_n)] \cdot 100\%}{\alpha_{Rxa}(n)}.
 \end{aligned} \tag{4}$$

The effects of the delays on the speed of the motors can be appreciated in Figure 11. In Figure 11(a), the results for speed for the 3 runs of Test 1 are presented. In Figure 11(b), the results for speed for the 3 runs of Test 2 are presented. The speed transmitted by the electronic board is displayed like

a thin line, while the speed read by the MD is presented in markers. The internal algorithm in the electronic board for the measurement of speed cannot yet effectively distinguish speeds below 2 revolutions per second. As a consequence, it takes some time to register the speed of zero. However, the data presented is still valid for the differences between what is sent and received between the electronic board and the MD. It is also accurate for speeds above 2 revolutions per second.

## 5. Discussion

The generation of a control signal in the MD is an important function for the purpose of controlling the electric motor

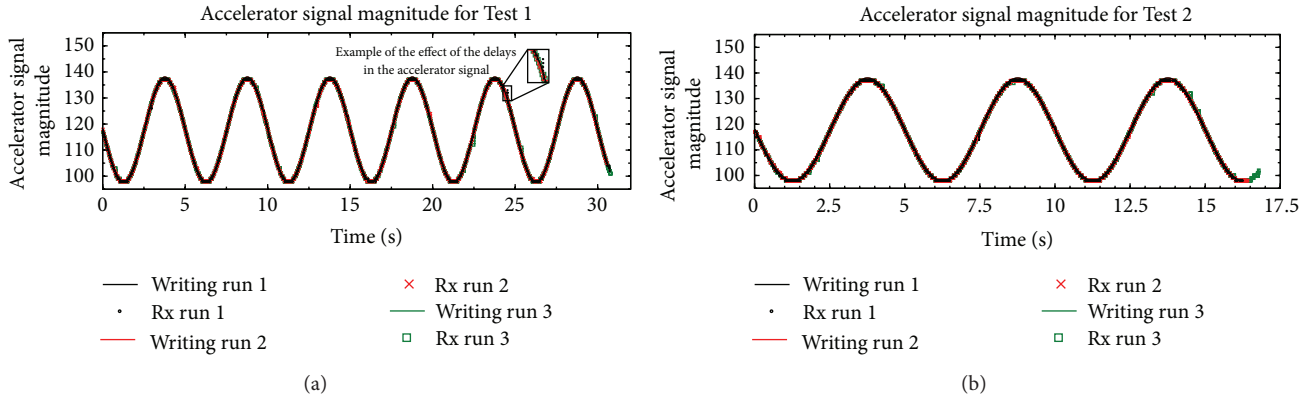


FIGURE 10: Plots for the accelerator signal. (a) Test 1, scheduled sampling rate of 20 ms. (b) Test 2, scheduled sampling rate of 10 ms.

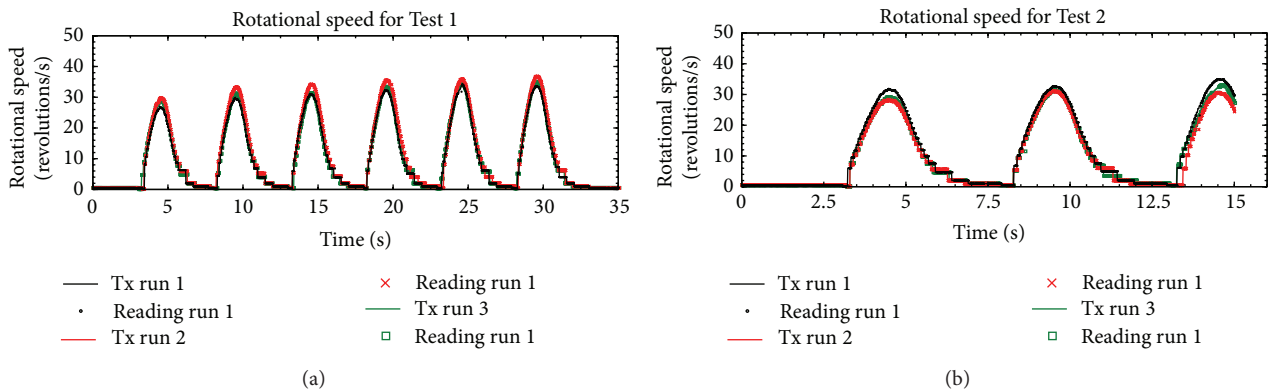


FIGURE 11: Rotational speed measured. (a) Test 1, sending rate scheduled in 25 ms. (b) Test 2, sending rate scheduled in 10 ms.

of the vehicle powertrain. The purpose of the choice of a sine function introduced in (1) is not to reproduce a realistic accelerator signal function but to have control of what is sent to the electronic board. More work is needed to conclude whether the generation of the accelerator signal with the MD is the best choice, when, alternatively, an intended speed profile can be sent and, after that, control the necessary accelerator signal to produce that speed directly in the real-time embedded system of the electronic board.

The objective of Test 1 and Test 2, which differ in the communication sample rate, is to find a difference in the generation of delays in dependence of the scheduled sample rate, which was already found for the MD in [29]. Figure 9 not only expands that information for the electronic board but also illustrates the magnitude and number of outliers (data outside 2 and 98 percentiles). It was found that, indeed, the spread in the delays is different for each test, being larger for Test 1. Test 2 is performed with a sample rate of 0.01 s, which is the sample rate at which standard *Controller Area Network (CAN)* systems manage information in automotive applications [30]. While the median and mean in the delays measured are close to zero, the outliers above 98 percentiles show big delays. For Test 1 and Test 2, four and two values, respectively, of the sample period delays were between 0.15 and 0.2 s. Those are relatively big delays compared to the

delays presented in the scheduled transmitting loop in the electronic board, which are almost zero. The electronic board, which is a real time embedded system, can schedule tasks with very good compliance with deadlines. For the current EvTEC propulsion system design, at its maximum speed of 40 km/h, a 0.2 s span represents about 2.2 m. Studies have shown that driver reaction times lie within 1.2 sec on average, and the NHTSA uses 1.5 sec as a standard for calculations [31]. In the worst case, signal delays may contribute significantly to the reaction time of the driver. A suitable safety strategy would have to be devised to attenuate the effects of such a delay.

The errors between the information that is sent and received (accelerator signal and speed) between the electronic board and the MD are small in all runs. The writing loops in the MD present lower errors than the receiving loop in the electronic board. This can be explained by the fact that while the writing loop is running, the data to be sent are stored in a buffer still inside the Android OS, which introduces variability in the receiving loop of the electronic board. The absolute maximum percentage of error found in the data was 3.54%, which corresponds to the difference between the ideal accelerator signal and the signal received in the electronic board. For the rotational speed, the errors can be appreciated in the plots (Figure 11). The rotational speed read follows



TABLE 3: Parameters for the probability density function.

$f(x)$ for delays in	Parameters							
	$a_1$	$b_1$	$c_1$	$d_1$	$a_2$	$b_2$	$c_2$	$d_2$
Writing cycle	702.9	.0003881	.0003288	5.787	365.8	.0005971	.0007046	14.57
Receiving cycle	2294	$3.4e-5$	.001515	1.53	104.4	.007136	.01445	-2.968

closely the rotational speed transmitted in each run; the largest variations are observed between runs.

For the purpose of energy management, the control and monitoring system was able to pick up behavior from which power consumption data can be extracted. For example, Figure 11 clearly shows that the rotational speed data sent from the electronic board and the rotational speed data received in the MD have some variations. In fact, a pattern that indicates some type of transitory behavior between consecutive runs can be observed within the trials. Each acceleration cycle shown in the figure is produced by electric pulses ordered by the controller. In this particular case, all electric pulses were measured to be identical. Hence, these changes could be attributed to physical mechanisms taking place inside the power module, such as variations in oil viscosity caused by temperature fluctuations or the inertial effects of the moving masses of oil. Further work and instrumentation is necessary to find those factors and measure and control them or to adjust operating conditions to make better use of available energy (stored in the batteries).

The Android application is installed in an unmodified version of the OS, which means that processes that have a negative effect on predictability, like the *garbage collector* (GC), are still there. GC is responsible for memory management and periodically frees memory occupied by objects no longer used in the program [25, 27, 28]. The tests presented here are conducted in the presence of the actions of the GC.

## 6. Simulation of the MD OS Delays

Clearly, the delays produced by MD OS affect the response of the system. Models for these delays can help the designer predict the performance of a proposed control. To reproduce the effects of the delays in the proposed system with a simulation, an approach based on probability density functions is now introduced. The delays measured experimentally in Section 4 can be simulated using a mixture of skewed Gaussian distributions. The proposed probability density function is presented in (5) to (7), where  $a$  is the mixture weight,  $b$  represents the location,  $c$  represents the scale, and  $d$  represents the shape factor. Consider

$$f(x) = a_1 \cdot \phi\left(\frac{x-b_1}{c_1}\right) \cdot \Phi\left(d_1 \cdot \left(\frac{x-b_1}{c_1}\right)\right) + a_2 \cdot \phi\left(\frac{x-b_2}{c_2}\right) \cdot \Phi\left(d_2 \cdot \left(\frac{x-b_2}{c_2}\right)\right), \quad (5)$$

$$\phi\left(\frac{x-b}{c}\right) = e^{-((x-b)/c)^2}, \quad (6)$$

$$\Phi\left(d \cdot \left(\frac{x-b}{c}\right)\right) = \frac{1}{2} \cdot \left[1 + \operatorname{erf}\left(\frac{x-b}{c\sqrt{2}}\right)\right]. \quad (7)$$

The parameters of the distributions that provide a best fit for the experimental data are provided in Table 3. Random delays  $\delta_W$  and  $\delta_{Rx}$  can then be generated using  $f(x)$ . Figure 12 compares histograms of simulated and experimental delays obtained for the cycles of the MD and the electronic board.

The simulated delays can be plugged into equations (\*) and (\*\*) in Table 1 to predict the time at which each sampling cycle starts in the MD ( $t_W$ ) and the electronic board ( $t_{Rx}$ ). Equation (8) is obtained in this manner. Consider

$$t_W(n) = \tau_{Ws} + \delta_W(n) + t_W(n-1), \quad (8)$$

$$t_{Rx}(n) = \tau_{Rs} + \delta_{Rx}(n) + t_{Rx}(n-1).$$

The accelerator signals sent by the MD are calculated using (9), where  $t_{proc}$  is the processing time between the start of the sampling cycle  $n$  (when the ideal acceleration is computed) and the moment in which  $\alpha_{Wa}(n)$  is calculated and sent. Figure 13 illustrates  $t_{proc}$  measured experimentally and simulated with a uniform distribution with values between 1.4 ms and 3.5 ms. Consider

$$\alpha_{Wa}(n) = \alpha_W(t_W(n) + t_{proc}(n)). \quad (9)$$

The values of  $\alpha_{Wa}$  are transferred to the electronic board ( $\alpha_{Rxa} = \lfloor \alpha_{Wa} \rfloor$ ). The errors ( $e_W$  and  $e_{Rx}$ ) are calculated according to (4). Figure 14 shows the obtained errors and their comparison with experimental results.

As can be seen, the simulated results follow a pattern similar to that of the experimental data. Clearly, a limitation of the approach presented here for the simulation of the delays is that outliers cannot be represented properly. For example, in this test, the 2.57% of the most extreme values of the delays in the MD were not shown in the simulation. For the case of the electronic board, 0.8% of the most extreme values of the delays were not simulated. Nevertheless, most of the delays in the sampling cycles can be effectively covered by this approach. To prepare a full simulation of the effects of the delays on the speed measured in the powertrain, it is necessary to include the mathematical models of the operation of the electric motors and the mechanical resistance created by the powertrain, which is beyond the scope of this paper.

## 7. Future Work

Future work includes a redundant system to handle the communication and enables the implementation of a fault

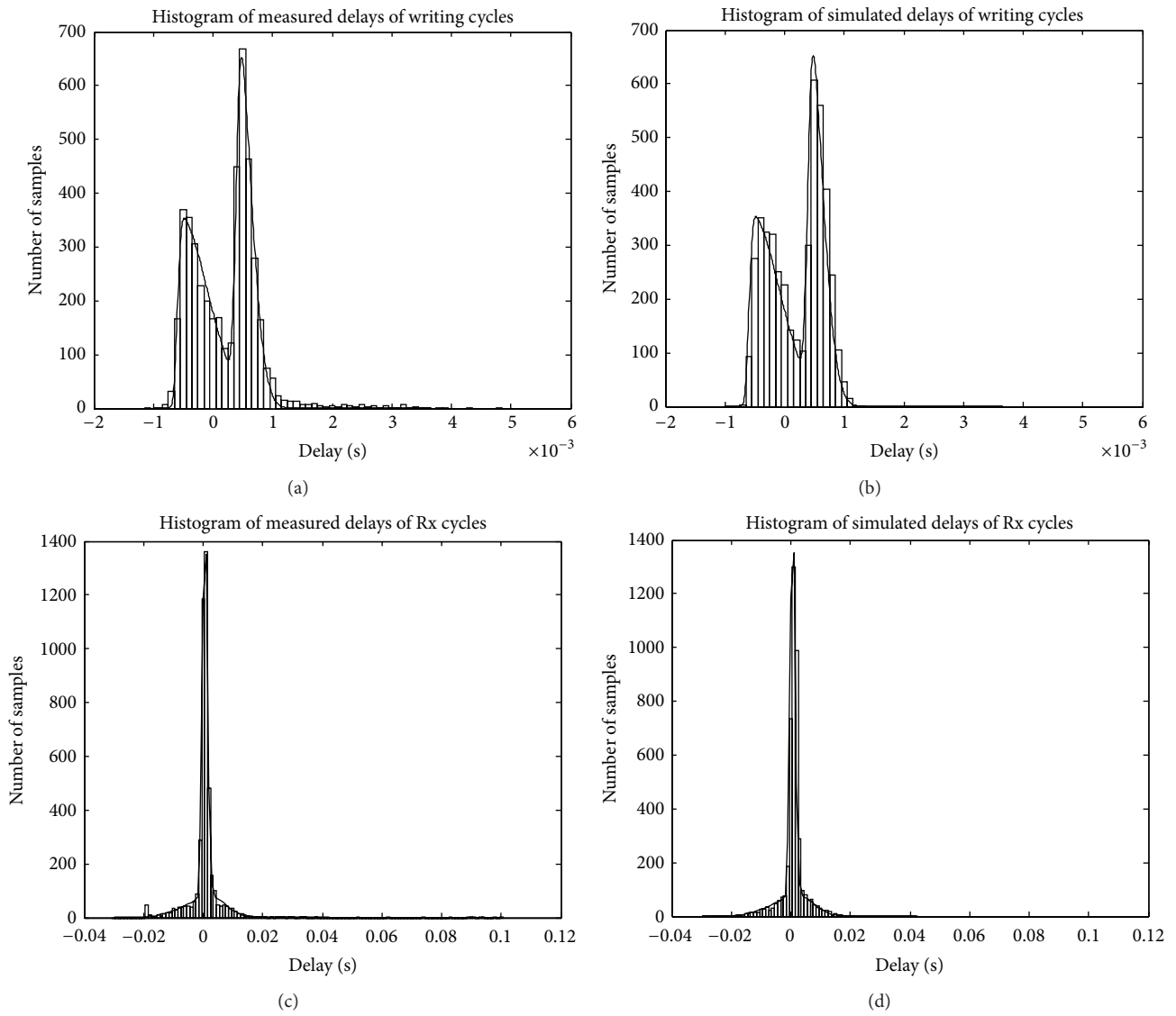


FIGURE 12: Histograms with the distribution of delays obtained in experimentation and by simulations with probability density function of (5).

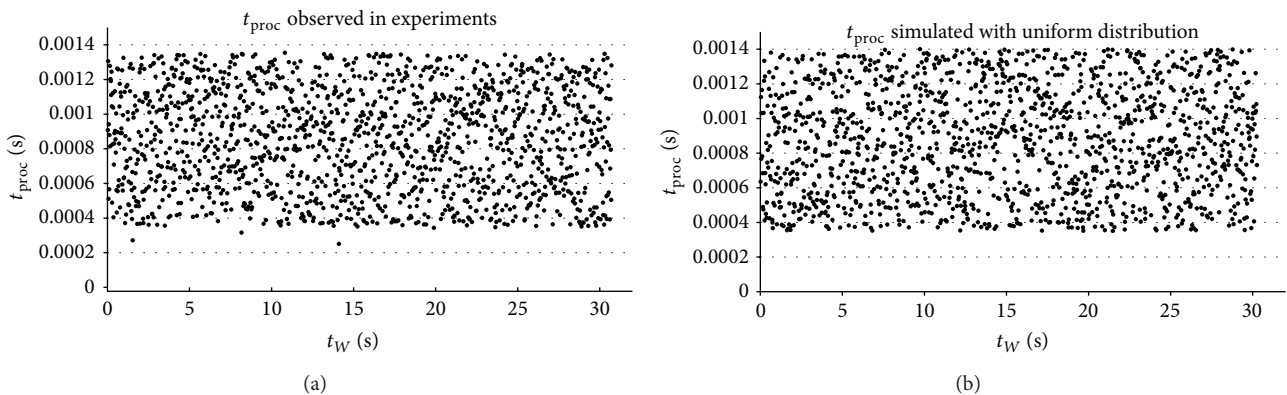


FIGURE 13:  $t_{proc}$  obtained experimentally (a) and generated with a uniform distribution (b).

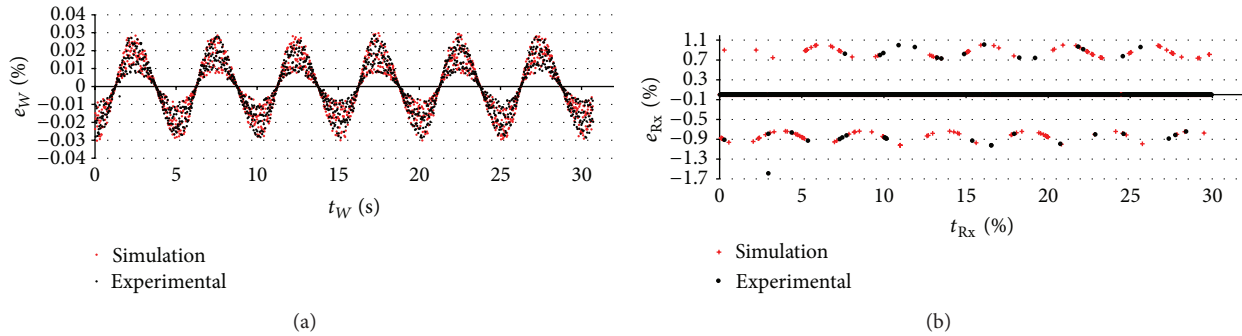


FIGURE 14: Errors produced in the accelerator signal as they were sent by the MD (a) and received by the electronic board (b).

tolerant control system. This would imply that the performance of the mechanical system will not be affected by the delays introduced by the communication with the MD. Other aspects to explore are the integration of the ABCe with a CAN bus and the possibility of complying with international communication standards. Finally, effects of electric and magnetic fields in the proposed control system must be further studied and evaluated. WLAN is susceptible to interferences from other networks working at the 2.4 GHz band [32], depending on factors such as the location inside the car. For that reason, an evaluation of possible interference sources given the distance between the WLAN host and the MD (which is expected to be short most of the time) must be addressed.

## 8. Conclusions

This work presented the basic architecture of an Android OS based controller used to control a critical system of the EvTEC concept. In particular, control of the accelerator signal was used. To achieve this, one of the electric motors that compose the propulsion system was conditioned and instrumented to interact with the Android based control ecosystem (ABCe). The integration of the MD is intended to provide the propulsion system of EVTEC with a modular and customizable controller that potentially can facilitate connectivity and driving autonomy.

Two stages were identified in the communication process between the electronic board and the MD: when the MD generates and writes commands to be received in the electronic board and when the electronic board transmits information to be read in the MD. Sample rates were programmed for these two stages.

Experiments were performed in which an intended acceleration signal was established and programmed in the MD to be sent to the electronic board. The electronic board transformed the received accelerator signal into a PWM signal to control the accelerator of the electric motor. Rotational speed of the shaft was measured and the information was transmitted back to the MD. During two tests, composed of three runs each, the sample periods in which the communication was produced between the MD and the electronic board were registered.

On average, the sample rates are accomplished as scheduled; however, maximum delays in communication occasioned by the introduction of the MD in the system can be as big as 0.2 s. The maximum absolute percentage of error in the accelerator signal caused by the delays in the scheduled rates, compared with the intended accelerator signal, was found to be only 3.54%.

A simulation for the delays in the sampling cycles is presented. A mixture of Gaussian distributions was fitted to the experimental data and was used to reproduce randomly the delays. With the simulated delays in the writing and Rx cycles, the errors in the transmission of the accelerator signal were simulated.

This work demonstrated that introducing a MD in the control and monitoring of physical systems will introduce delays in communications that must be taken into account in each application. While in this work the accelerator signal was generated and sent by the MD as a demonstration of capabilities and limitations, the MD can have numerous different uses in the context of connectivity, autonomous driving, and control of propulsion system, like speed limit detection, energy management, auxiliary in safety subsystems, or networks for cooperation between vehicles and infrastructure elements.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] T. Legget, *What If a Robot Drove Your Car*, 2013, <http://www.bbc.com/news/business-21804352>.
- [2] J. Levinson, J. Askeland, J. Becker et al., "Towards fully autonomous driving: systems and algorithms," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 163–168, Baden-Baden, Germany, June 2011.
- [3] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous ground vehicles-concepts and a path to the future," *Proceedings of the IEEE*, vol. 100, pp. 1831–1839, 2012.
- [4] D. Ni, J. Li, S. Andrews, and H. Wang, "A methodology to estimate capacity impact due to connected vehicle technology,"

- International Journal of Vehicular Technology*, vol. 2012, Article ID 502432, 10 pages, 2012.
- [5] A. Dardanelli, F. Maggi, M. Tanelli et al., "A security layer for smartphone-to-vehicle communication over bluetooth," *IEEE Embedded Systems Letters*, vol. 5, no. 3, pp. 34–37, 2013.
  - [6] N. Naylor, *U.S. Department of Transportation Announces Decision to Move Forward with Vehicle-to-Vehicle Communication Technology for Light Vehicles*, 2014, <http://www.nhtsa.gov/About+NHTSA/Press+Releases/2014/USDOT+to+Move+Forward+with+Vehicle-to-Vehicle+Communication+Technology+for+Light+Vehicles>.
  - [7] C. M. de Dominicis, D. Mazzotti, M. Piccinelli, S. Rinaldi, A. Vezzoli, and A. Depari, "Evaluation of bluetooth hands-free profile for sensors applications in smartphone platforms," in *Proceedings of the IEEE Sensors Applications Symposium (SAS '12)*, pp. 1–6, Brescia, Italy, February 2012.
  - [8] Y.-H. Weng, F.-S. Sun, and J. D. Grigsby, "GeoTools: an android phone application in geology," *Computers and Geosciences*, vol. 44, pp. 24–30, 2012.
  - [9] Y.-C. Lai, F. Han, C.-N. Lai, Y.-H. Yeh, and Y.-C. Szu, "A GPS navigation system with QR code decoding and friend positioning in smart phones," in *Proceedings of the 2nd International Conference on Education Technology and Computer (ICETC '10)*, pp. V566–V570, Shanghai, China, June 2010.
  - [10] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González, "Safe driving using mobile phones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1462–1468, 2012.
  - [11] Q. Wang, J. Wu, C. Long, and B. Li, "P-FAD: real-time face detection scheme on embedded smart cameras," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 210–222, 2013.
  - [12] E. Koukoumidis, M. Martonosi, and L.-S. Peh, "Leveraging smartphone cameras for collaborative road advisories," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 707–723, 2012.
  - [13] J. Ziomek, L. Tedesco, and T. Coughlin, "My car, my way: why not? I paid for it!," *IEEE Consumer Electronics Magazine*, vol. 2, no. 3, pp. 25–29, 2013.
  - [14] S. Durach, U. Higgen, and M. Huebler, "Smart automotive apps: an approach to context-driven applications," in *Proceedings of the FISITA 2012 World Automotive Congress*, vol. 200 of *Lecture Notes in Electrical Engineering*, pp. 187–195, Springer, Berlin, Germany, 2013.
  - [15] A. Schmidt and D. Bial, "Phones and MP3 players as the core component in future appliances," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 8–11, 2011.
  - [16] R. Bose, J. Brakensiek, K.-Y. Park, and J. Lester, "Morphing smartphones into automotive application platforms," *Computer*, vol. 44, no. 5, pp. 53–61, 2011.
  - [17] A. Dardanelli, M. Tanelli, B. Picasso, S. M. Savaresi, O. di Tanna, and M. D. Santucci, "A smartphone-in-the-loop active state-of-charge manager for electric vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 3, pp. 454–463, 2012.
  - [18] A. Cela, A. Hrazdira, A. Reama et al., "Real time energy management algorithm for hybrid electric vehicle," in *Proceedings of the 14th IEEE International Intelligent Transportation Systems Conference (ITSC '11)*, pp. 335–340, October 2011.
  - [19] J. Wideberg, P. Luque, and D. Mantaras, "A smartphone application to extract safety and environmental related information from the OBD-II interface of a car," *International Journal of Vehicle Systems Modelling and Testing*, vol. 7, no. 1, pp. 1–11, 2012.
  - [20] C. Spelta, V. Manzoni, A. Corti, A. Goggi, and S. M. Savaresi, "Smartphone-based vehicle-to-driver/environment interaction system for motorcycles," *IEEE Embedded Systems Letters*, vol. 2, no. 2, pp. 39–42, 2010.
  - [21] L. Perneel, H. Fayyad-Kazan, and M. Timmerman, "Can Android be used for real-time purposes?" in *Proceedings of the International Conference on Computer Systems and Industrial Informatics (ICCSII '12)*, 6, p. 1, December 2012.
  - [22] A. G. Bianchessi, C. Ongini, I. Boniolo et al., "A novel electric vehicle for smart indoor mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1429–1440, 2014.
  - [23] S. Park, T. T. T. Ha, J. Y. Shivajirao, J. Park, J. Kim, and M. Hahn, "Smart wheelchair control system using cloud-based mobile device," in *Proceedings of the 3rd International Conference on IT Convergence and Security (ICITCS '13)*, pp. 1–3, Macau, China, December 2013.
  - [24] H.-T. Lee, W.-C. Lin, C.-H. Huang, and Y.-J. Huang, "Wireless indoor surveillance robot," in *Proceedings of the 50th Annual Conference on Society of Instrument and Control Engineers (SICE '11)*, pp. 2164–2169, September 2011.
  - [25] I. Kalkov, D. Franke, J. F. Schommer, and S. Kowalewski, "A real-time extension to the android platform," in *Proceedings of the ACM 10th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES '12)*, pp. 105–114, October 2012.
  - [26] W. Mauerer, G. Hillier, J. Sawallisch, S. Hönick, and S. Oberthür, "Real-time android: deterministic ease of use," in *Proceedings of the Embedded Linux Conference Europe (ELCE '12)*, 2012.
  - [27] Y. Yan, S. H. Konduri, A. Kulkarni, V. Anand, S. Y. Ko, and L. Ziarek, "Rtdroid: a design for real-time android," in *Proceedings of the 11th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES '13)*, ACM International Conference Proceeding Series, pp. 98–107, 2013.
  - [28] K. Kolek, "Application of android OS as a real time control platform," *Automatica*, vol. 17, no. 2, pp. 197–206, 2013.
  - [29] P. D. U. Coronado, V.-B. Sundaresan, and H. Ahuett-Garza, "Design of an android based input device for electric vehicles," in *Proceedings of the 2nd IEEE International Conference on Connected Vehicles and Expo (ICCVE '13)*, pp. 736–740, December 2013.
  - [30] A. A. Ortiz Espinoza, *Diseño de una Red CAN para Implementar un Sistema de Control de una Suspensión Semiactiva de un Vehículo [M.S. thesis]*, Tecnológico de Monterrey, Monterrey, México, 2014.
  - [31] G. Taoka, "Break reaction times of unalerted drivers," *ITE Journal*, vol. 59, no. 3, pp. 19–21, 1989.
  - [32] F. Fang, J. Sun, Y. Han, and H. Zhu, "Performance analysis of ZigBee under WLAN and multiple bluetooth piconets interferences," in *Proceedings of the IET International Conference on Information Science and Control Engineering (ICISCE '12)*, pp. 1–7, Shenzhen, China, December 2012.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

