REAL-TIME SIMULATOR FOR DYNAMIC SYSTEMS USING A FPGA PLATFORM

Sérgio N. Silva*, Marcelo A. C. Fernandes†

*Department of Computer Engineering and Automation
Federal University of Rio Grande do Norte, UFRN

Emails: sergionatan@dca.ufrn.br, mfernandes@dca.ufrn.br

Abstract— In most cases, the development of new embedded algorithms for the automation and control of industrial plants requires real-time testing. However, the use of real plants can often involve high costs, which can prohibit the testing of new algorithms. The objective of this project was therefore to develop an embedded field-programmable gate array (FPGA) platform for the real-time simulation of dynamic systems. FPGAs are flexible reconfigurable hardware platforms that may be altered or programmed in run time. Different to commonly used processors, this reconfigurable computer enables adjustment and optimization of the hardware structures for executing specific algorithms, aiming to improve execution performance. The platform proposed here, called the Real-Time Simulator for Dynamic Systems (RTSDS-FPGA), have potential in both academic and industrial environments. In industry, the RTSDS-FPGA can be used in the optimization and tuning of embedded control and automation algorithms. In the academic field, it can be used to support research into new solutions in the area of automation and control, and as a tool to assist teaching at undergraduate and postgraduate levels.

Keywords— Real time, Simulation, Dynamic systems, FPGA.

1 Introduction

There are high costs associated with the implementation of new control algorithms in industrial plants. This is due to the need of real smaller-scale models or programmed halts in operation of the plant. There is therefore a need for the use of real-time simulators (RTSs) able to represent the dynamic functioning of the plant and its real-time restrictions. For example, under real conditions, the calibration of a control algorithm in automotive vehicle injection systems requires tests that require at least two drivers, a test track, and a vehicle.

In recent years, increasing interest in platforms for simulation in real time has been largely due to advances in hardware technologies. Various studies have been reported using different hardware platforms associated with microprocessors (uP), microcontrollers (uC), digital signal processors (DSP - *Digital Signal Processors*), and *Field Programmable Gate Array* (FPGAs) (Monga et al., 2012; Champagne et al., 2003; Keller and Kolb, 2009; Matar and Iravani, 2010). (de Souza et al., 2014) presented a low-cost RTS proposal that could be easily implemented using uPs and uCs. However, an important drawback of the technique is the processing speed, with simulation of low complexity systems having time restrictions in the region of seconds.

Different to proposals described in the literature, the aim of this work is to present a proposal for a real-time simulation platform for reconfigurable hardware. This platform, called the RTSDS-FPGA, follows the same structure described by (de Souza et al., 2014), but uses an FPGA as the main control module, enabling the simulation of more complex systems with time restrictions on the order of microseconds.

2 Physical Architecture and Logic of the RTSDS-FPGA

Figure 1 shows a block diagram of the architecture of the RTSDS. The main control module (MCM) is coupled to a set of M data acquisition modules (DAMs) and N signal generator modules (SGMs). The function of the MCM is to control, manage, and configure the real-time simulation platform, and it is responsible for the input and output of data by means of a data communication protocol. The MCM is implemented using an FPGA, and is directly responsible for configuring and executing the simulation of a previously selected dynamic system. It collects and processes the data, which is then sent to the output. A MCM is associated with M DAMs and N SGMs.

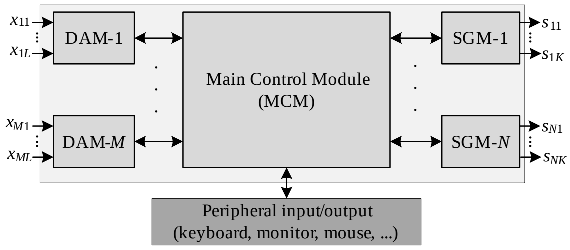

Figure 1: Physical architecture of the RTSDS platform.

The DAMs and SGMs are auxiliary hardware components whose function is to receive and generate signals related to the dynamic system to be simulated by the MCM. In practice, they can be implemented by various hardware platforms, such as DSPs, uCs, and other devices that possess digital and/or analog inputs and outputs. As shown in Figure 1, each m-th DAM has L digital and/or analog inputs, and each n-th SGM has K digital and/or analog outputs. Each RTSDS-FPGA must possess at least one DAM and one SGM.

Figure 2 illustrates the logical architecture of the RTSDS-FPGA, where one uC embedded in the MCM, here called the principal application (PA), enables circuits that are also embedded (in the FPGA), called the object dynamic system (ODS). A ODS represents the dynamic system to be simulated in real time, and is composed, amongst others, of nine main attributes, expressed by $\{P, H, f(t), \mathcal{M}, \mathcal{C}, \mathcal{D}, MethodODE, t_a, L.b\}$ where,

- P: Represents the number of inputs of the dynamic system.
- H: Represents the number of outputs of the dynamic system.
- $f(t)$: Defines the system of ordinary differential equations (ODEs) of the dynamic system to be simulated in real time.
- $\mathcal{M}$: Defines a set of adjustable parameters associated with the dynamic system.
- $\mathcal{C}$: Characterizes the set of codes, the ADAMs, that will be on board the DAMs associated with the inputs of the dynamic system.
- $\mathcal{D}$: Characterizes the set of codes, the ASGMs, that will be on board the SGMs associated with the outputs of the dynamic system.
- *MethodODE*: Attribute that defines which numerical ODE resolution method will be used in the real-time simulation. The RTSDS-FPGA can implement various resolution methods, such as the Euler, Runge-Kutta, and others (Butcher, 2008).
- t_a: Represents the sampling time to be used in the real-time simulation process.
- $L.b$: Represents the fixed point resolution used by the ODS. L is the total number of bits, and b is the number of bits associated with the fractional part of the total.

One MCM possesses various embedded ODS circuits that can be easily enabled by means of the PA.

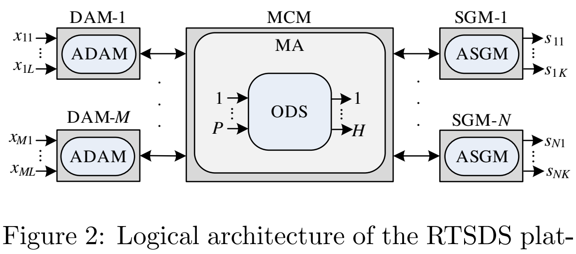

Figure 2: Logical architecture of the RTSDS platform.

Different to the proposal of (de Souza et al., 2014), where the ODSs were implemented using *threads*, the RTSDS-FPGA implements each ODS using a dedicated hardware circuit, so that various ODSs can be executed in parallel, in a much shorter sampling time. Following from the construction of a ODS, the sampling time, t_a, can be expressed by:

$$t_a \geq t_{DAM} + t_1 + t_{DAT} + t_s + t_2 + t_{SGM}, \quad (1)$$

where t_{DAM} is the greatest processing time associated with all the DAMs utilized by the MCM, t_1 is the time related to the lowest rate of transfer (in bps) between the MCM and all the DAMs utilized, t_{DAT} is the processing time corresponding to the time taken to read the information received from all the DAMs, $t_s$is the time taken to execute the ODE solution, t_2 is the time for the lowest rate of transfer (in bps) between the MCM and the SGMs, and t_{SGM} is the highest processing time associated with all the SGMs utilized by the MCM.

3 Description of the Prototype

Validation of the RTSDS-FPGA platform involved the development of a prototype using a Xilinx ML-605 kit. This kit includes a Virtex-6 Model xc6vcx240t-1ff1156 FPGA, together with other functionalities such as external DDR3 memory, Gigabit Ethernet connections, PCI Express, USB, and USART, amongst other peripherals. The MCM was implemented in the Virtex-6 FPGA with a ODS to simulate a longitudinal automotive vehicle model, Figure 3. The ODS was implemented at a fixed point using $L = 32$ bits, of which $b = 14$ bits were used for the fractional part. It is important to emphasize that one MCM (FPGA) could implement various ODSs in parallel, performing other simulations. The entire development of the prototype was achieved using a rapid prototyping platform called the System Generator (Xilinx, 2008).

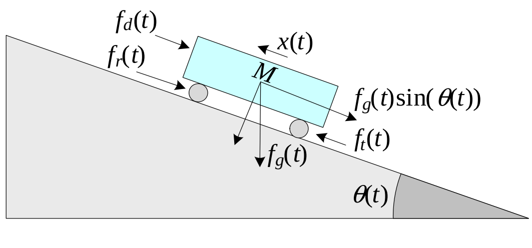

Figure 3: Schematic diagram of the longitudinal vehicle model.

The dynamics of the vehicle can be expressed by:

$$M\frac{dx(t)}{dt} = f_t(t) - f_a(t), \quad (2)$$

where M is the mass of the vehicle (in kg), $x(t)$ is the linear velocity of the vehicle (in m/s), $f_t(t)$ is the traction force exerted by the vehicle (in N), and $f_a(t)$ is the friction force (in N). The torque can be expressed by:

$$f_t(t) = \frac{\tau_m(t)}{r} \quad (3)$$

where r is the radius of the wheels of the vehicle (in m) and $\tau_m(t)$ is the torque (in N.m) generated by the motor, which can be expressed by:

$$\tau_m(t) = V_e(t)\tau_m^{max}, \quad (4)$$

where τ_m^{max} is the maximum torque (in N.m) and $V_e(t)$ is one of the input signals of the simulation.

The friction force, $f_a(t)$, can be described by:

$$f_a(t) = f_d(t) + f_r(t) + f_g(t)\sin(\theta(t)), \quad (5)$$

where $f_d(t)$ is the aerodynamic friction force (in N), $f_r(t)$ is the rolling resistance force (in N), $f_g(t)$ is the force due to gravity (in N), and $\theta(t)$ is the inclination angle of the plane on which the vehicle is located. The variable $\theta(t)$ can be expressed as:

$$\theta(t) = V_o(t)\theta^{max}, \quad (6)$$

where θ^{max} is the maximum inclination angle, and $V_o(t)$ is the second input signal of the simulation.

The aerodynamic friction is given by:

$$f_d(t) = \frac{1}{2}\rho C_d A_{fr} x^2(t), \quad (7)$$

where ρ is the density of air, C_d is the aerodynamic drag coefficient, and A_{fr} is the frontal area of the vehicle (in m^2). The rolling resistance force is given by:

$$f_r(t) = Mg\left(C_0 + C_1 x^2(t)\right), \quad (8)$$

where C_0 and C_1 are the rolling coefficients, and g is the acceleration due to gravity (in m/s^2). Finally, the gravitational force is given by:

$$f_g(t) = Mg. \quad (9)$$

The dynamic system was simulated in real time, using the Euler method to solve the ODEs. In this method, Eq. 2 can be rewritten as:

$$x(t) = \begin{cases} u(t) \text{ for } u(t) < x_{max} \\ x_{max} \text{ for } u(t) \geq x_{max} \end{cases}, \quad (10)$$

where x_{max} is the maximum velocity of the vehicle, and

$$u(t) = x_0 + \int_{t_0}^{t} h_{vs}\left(s, x(s)\right) ds,\ t > t_0 \text{ e } x_0 \geq 0, \quad (11)$$

where x_0 is the initial velocity of the vehicle, and $h_{vs}\left(s, x(s)\right)$ can be expressed as:

$$\begin{aligned} h_{vs}\left(s, x(s)\right) &= \frac{1}{M}\Bigg(V_e(s)\frac{\tau_m^{max}}{r} - \\ &\quad \frac{1}{2}\rho C_d A_{fr} x^2(s) - \\ &\quad Mg\left(C_0 + C_1 x^2(s)\right) - \\ &\quad Mg\sin(V_o(s)\theta^{max})\Bigg). \quad (12) \end{aligned}$$

The Euler method was applied to the integral shown in Eq. 11, using the area of a rectangle whose base has length Δt (Butcher, 2008) (Griffiths and Higham, 2010). In other words,

$$\begin{aligned} u(t+\Delta t) &= x_0 + \int_{t_0}^{t} h_{vs}\left(s, x(s)\right) ds + \\ &\quad \int_{t}^{\Delta t} h_{vs}\left(s, x(s)\right) ds \\ &\approx u(t) + h_{vs}\left(t, x(t)\right) \times \Delta t \quad (13) \end{aligned}$$

where Δt refers to the step of the Euler method (Butcher, 2008) (Griffiths and Higham, 2010). For the RTSDS, Δt is the execution time of the stretch of code responsible for the simulation (main loop), so $t_s = \Delta t$. The implemented solution method can then be expressed by:

$$x(t + t_s) = \begin{cases} v(t) \text{ for } c(t) < x_{max} \\ \\ x_{max} \text{ for } c(t) \geq x_{max} \end{cases} \quad (14)$$

where,

$$\begin{aligned} c(t) &= x(t) + \Bigg(\frac{1}{M}\Bigg(V_e(t)\frac{\tau_m^{max}}{r} - \\ &\quad \frac{1}{2}\rho C_d A_{fr} x^2(t) - Mg\left(C_0 + C_1 x^2(t)\right) - \\ &\quad Mg\sin(V_o(t)\theta^{max})\Bigg)\Bigg) \times t_s. \quad (15) \end{aligned}$$

Hence, the ODS Figure 4 possesses two inputs $P = 2$, namely torque, $V_e(t)$, and inclination angle, $V_o(t)$, and one output $H = 1$, which is the velocity of the vehicle in a longitudinal direction, $x(t)$.

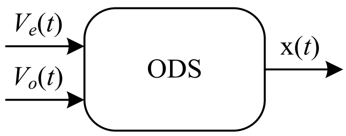

Figure 4: ODS composed of two inputs, $P = 2$ (torque, $V_e(t)$, and inclination angle, $V_o(t)$), and one output, $H = 1$ (velocity of the vehicle in a longitudinal direction,$x(t)$).

The prototype was developed with two DAMs (DAM-1 and DAM-2) and one SGM (SGM-1) that functioned as transducers of signals from the real world to the platform, and vice versa. The components DAM-1 and DAM-2 shown in Figure 5 are A/D converters responsible for acquiring the signals for the torque, $V_e(t)$, and the inclination angle, $V_o(t)$, respectively. SGM-1, also shown in Figure 5, is a D/A converter responsible for the output signal of the system, $x(t)$. The physical structure of the prototype is illustrated in Figure 5.

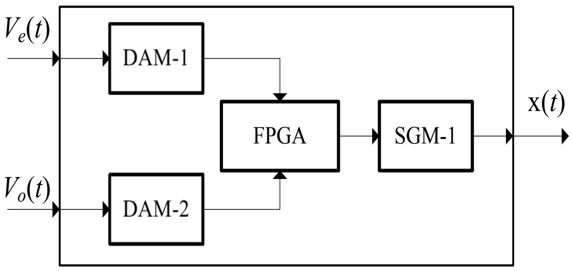

Figure 5: Implemented prototype of the ODS

Based on Equation 14, the ODS was implemented in the FPGA by means of the circuit illustrated in Figure 6. This circuit is composed of smaller sub-circuits, corresponding to parts of Equation 15. The ADF sub-circuit, illustrated in Figure 7, implements the aerodynamic drag force, described by Equation 7. The RRF sub-circuit, illustrated in Figure 8, implements the rolling resistance, described by Equation 8. The final sub-circuit, illustrated in Figure 9, implements the Euler method.

The ODS was implemented at a fixed point ($L = 32$ e $b = 14$ bits). In this model, the torque at the rear wheels, $\tau_m(t)$, and the longitudinal inclination angle of the track, $\theta(t)$, are the ODS input parameters, and the output parameter is the velocity of the vehicle, $x(t)$.

4 Results

After development of the prototype, two real-time simulation scenarios were tested. In the first scenario, the variables composing the input of the system were kept constant, the torque, $\tau_m(t)$, was fixed at 2000 N.m and the value of $\theta(t)$ was fixed at 0°. In the second scenario, the value of $\theta(t)$ was varied over time, simulating an ascent of 5° over 50 seconds and a descent of −5° over 150 seconds. In the third scenario, the torque, $\tau_m(t)$, was initiated at 2000 N.m and was then linearly decreased to 1000 N.m over 25 seconds, followed by a step decrease to 500 N.m at 50 seconds, and then step increases at 100 seconds and 150 seconds, reaching torque of 1500 N.m. A final linear increase was initiated at 200 seconds and ended at 250 seconds, with final torque of 2000 N.m. The value of $\theta(t)$ was varied over time, with an ascent of 5° at 50 seconds and a descent of −5° at 150 seconds. In order to realize the tests and the results of comparisons , the Matlab/Simulink was configured with the Euler method (integration method of the dynamic system) using a fixed step of 1ms.

The results (Figures 10, 11, and 12) were compared with simulations in non-real-time performed using Matlab/Simulink. The mean error ME and the variance VAR for the difference between the responses obtained by the RTSDS-FPGA and Matlab/Simulink simulations for the two scenarios are shown in Table 1. The value of ME was determined using:

$$ME = \frac{1}{N}\sum_{i=0}^{N-1}|X(i) - Y(i)| \tag{16}$$

where $X(i)$ is the value of the i-th sample provided by the RTSDS-FPGA, and $Y(i)$ is the value given by the Matlab/Simulink simulation at the same i-th instant. The variable N represents the number of samples collected. Both procedures considered the value for the velocity of the vehicle, $x(t)$. The sampling time used, t_a, was 1 ms.

The variance can be described by:

$$VAR = \frac{1}{N-1}\sum_{i=0}^{N-1}(X(i) - \bar{X}(i))^2 \tag{17}$$

where $X(i)$ is the i-th value of the set of samples, and $\bar{X}(i)$ is the expected value of the variable in question.

Table 1: Measures of similarity between the different simulation techniques for scenarios 1, 2, and 3

Scenarios	ME	Variance
Scenario 1	5, 2661	$6,9029e-05$
Scenario 2	1, 5858	$3.8962e-05$
Scenario 3	2, 1617	$2,2621e-05$

The results shown in Figures 10, 11, and 12 demonstrate that the RTSDS-FPGA method provided excellent performance in both scenarios. The values obtained were very close to those of the non-real-time simulations using Matlab/Simulink, as indicated by the mean errors Table 1.

Table 2 presents the results obtained after synthesis of the RTSDS-FPGA using the Virtex 6 FPGA. It can be seen that the spatial occupation was small, at around 1% of the total, showing that the RTSDS-FPGA could house various embedded ODSs. Another important point is that the minimum sampling time that the RTSDS-FPGA can work with is 50 ns. With this time analysis, it was observed the system simulation viability in real time with more complex models. Moreover, be observed that the STRSD - FPGA can also be used as a tool to accelerate systems with slow dynamics of simulations.

The findings demonstrated that there was substantial remaining hardware capacity, allowing the possibility of executing various ODSs at

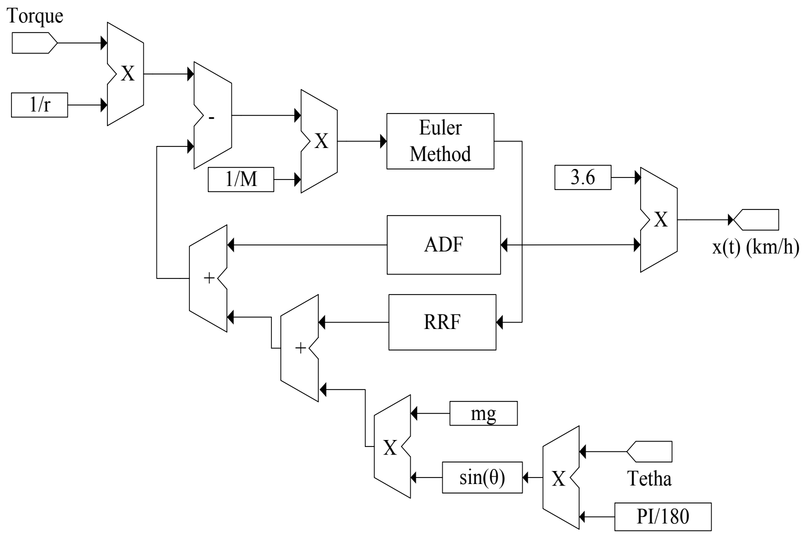

Figure 6: Circuit for the ODS of the prototype

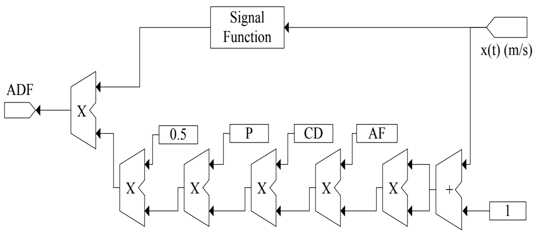

Figure 7: Detail of the ADF sub-circuit (aerodynamic drag force).

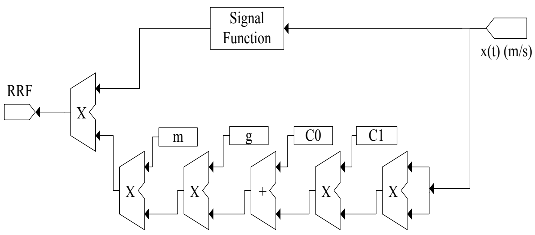

Figure 8: Detail of the RRF sub-circuit (rolling resistance force).

the same time or a ODS that requires a simulator with low time granularity.

5 Conclusion

This work proposes an embedded platform, called the RTSDS-FPGA, for real-time simulation of dynamic systems. The platform is composed of a central hardware module (MCM) and a set of auxiliary hardware modules (DAMs and SGMs). Experimental tests were performed using a prototype developed with a Virtex-6 FPGA as the MCM. The results obtained confirmed the satisfactory performance of the RTSDS-FPGA platform, with small errors when compared to non-real-time simulations. The temporal analysis on the netlist file shows that it is possible to simulate in real time with more complex models. The RTSDS - FPGA could also be used to optimize and synchronize embedded control algorithms employed in industrial plants. It could also be employed as a tool to support graduate and post-graduate teaching.

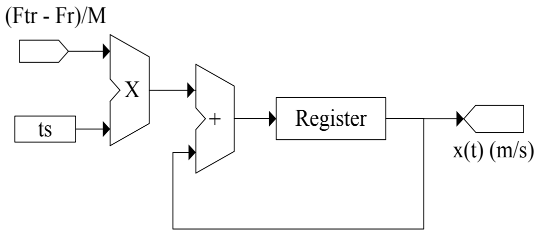

Figure 9: Detail of the Euler sub-circuit.

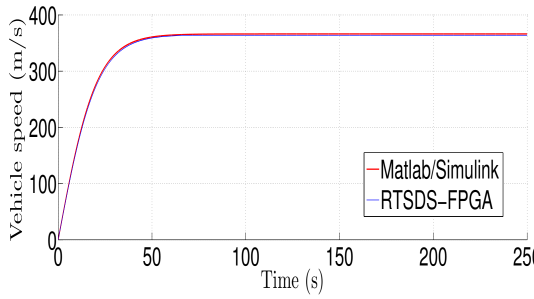

Figure 10: Scenario 1 - Torque, $\tau_m(t)$, fixed at 2000 N.m and $\theta(t)$ fixed at 0°.

References

Butcher, J. (2008). *Numerical Methods for Ordinary Differential Equations*, 1st edn, Wiley.

Champagne, R., Dessaint, L.-A. and Fortin-Blanchette, H. (2003). Real-time simulation of electric drives, *Mathematics and Computers in Simulation* **63**(3-5): 173 – 181.

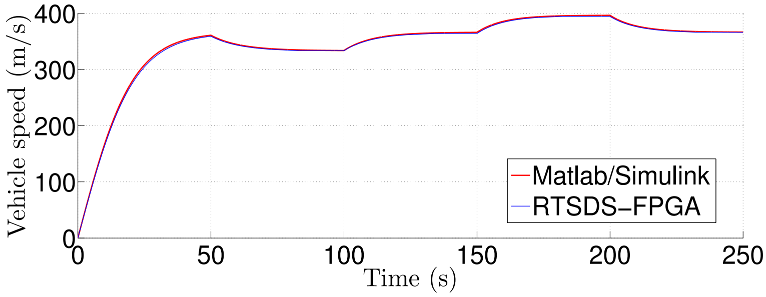

Figure 11: Scenario 2 - Torque, $\tau_m(t)$, fixed at 2000 N.m and $\theta(t)$ varied over time, with an ascent of 5° at 50 seconds and a descent of -5^o at 150 seconds.

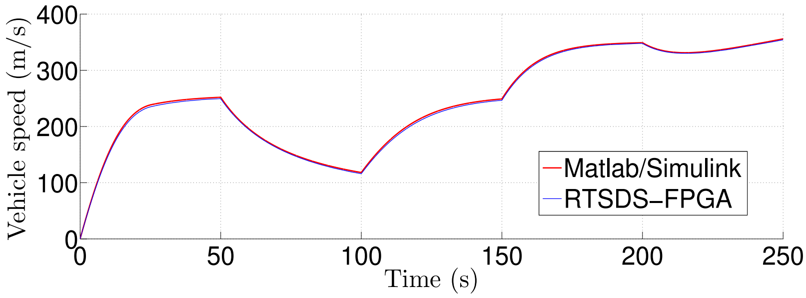

Figure 12: Scenario 3 - Values of both input variables vary over time in accordance with the descriptions given previously.

Table 2: Netlist analysis of the platform.

Item	Used	Percentage of use
Registers	188	1%
LUTS	3316	2%
Multipliers	12	1%
Sample time, t_a	50 ns	–

de Souza, I. D. T., Silva, S. N., Teles, R. and Fernandes, M. A. C. (2014). Platform for real-time simulation of dynamic systems and hardware-in-the-loop for control algorithms, *Sensors* **14**(10): 19176–19199.

Griffiths, D. and Higham, D. (2010). *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*, Springer undergraduate mathematics series, Springer.

Keller, M. and Kolb, A. (2009). Real-time simulation of time-of-flight sensors, *Simulation Modelling Practice and Theory* **17**(5): 967 – 978.

Matar, M. and Iravani, R. (2010). Fpga implementation of the power electronic converter model for real-time simulation of electromagnetic transients, *Power Delivery, IEEE Transactions on* **25**(2): 852–860.

Monga, M., Karkee, M., Sun, S., KiranTondehal, L., Steward, B., AtulKelkar and Zambreno, J. (2012). Real-time simulation of dynamic vehicle models using a high-performance reconfigurable platform, *Procedia Computer Science* **9**(0): 338 – 347.

Xilinx (2008). System Generator for DSP, `http://www.xilinx.com/products/design-tools/vivado/integration/sysgen.html`. [Online; accessed 31-05-2015].