Real-time simulator of collaborative autonomous vehicles.

Farid Bounini
LIV – Université de Sherbrooke
Sherbrooke, Canada
farid.bounini@usherbrooke.ca

Vincent Lapointe
Opal-RT Technologies Inc,
Montreal, Canada
vincent.lapointe@opal-rt.com

Denis Gingras
LIV – Université de Sherbrooke
Sherbrooke, Canada
denis.gingras@usherbrooke.ca

Dominique Gruyer
LIVIC - IFSTTAR
Versailles, France,
dominique.gruyer@ifsttar.fr

***Abstract*— Collaborative autonomous vehicles will appear in the near future and will transform deeply road transportation systems, addressing in part many issues such as safety, traffic efficiency, etc. Validation and testing of complex scenarios involving sets of autonomous collaborative vehicles are becoming an important challenge. Each vehicle in the set is autonomous and acts asynchronously, receiving and processing huge amount of data in real time, coming from the environment and other vehicles. Simulation of such scenarios in real time require huge computing resources. This paper presents a simulation platform combining the real-time OPAL-RT Technologies for processing and parallel computing, and the Pro-SiVIC vehicular simulator from Civitec for realistic simulation of vehicles dynamic, road/environment, and sensors behaviors. The two platforms are complementary and their combining allow us to propose a real time simulator of collaborative autonomous systems.**

***Keywords*— Real time simulation, parallel computing, sensors, data fusion, embedded systems, collaborative autonomous vehicles.**

I. INTRODUCTION

The development of embedded sensors and communication technologies, during the last few decades, has affected significantly the automotive sector, helping to overcome traffic problems and to achieve greater safety on roads and highways.

Nowadays, on board active safety systems are essentially designed to avoid collisions rather than only reducing their impact on passengers as with passive systems. Furthermore, augmented night vision, automated park assistance and navigation systems are functionalities that are already implemented on high-end vehicles [1]. Although current ADAS (Advanced Driver Assistance Systems) are built as isolated subsystems that are typically not sharing information within or between vehicles, future generations of ADAS will feature a much higher level of integration as well as extended communication capabilities allowing to exchange information between vehicles. This will lead to fully automated vehicles capable of sharing information and interacting with dynamic environment. Some research efforts are focusing on the cooperative and optimized behavior of nodes, and Multi-Hop Networks [2], some specific aspects are introduced as friends/enemies, dominance, etc. in [3] to optimize the behavior of multi-hop network. On the other side, some basic scenarios on collaborative autonomous vehicles have been experimented with three collaborative autonomous vehicles at the Griffith University's Intelligent Control Systems Laboratory (ICSL) with its partners [4]. In these experiments, simple scenarios were considered such as overtaking, transversal intersection and lane platooning.

This paper presents a real-time simulator of collaborative autonomous vehicles on parallel computing tools, and deals mainly with five points: Section II is dedicated to embedded sensors technology and ADAS systems. In section III we propose a strategy to build a real time simulator architecture of collaborative autonomous vehicles. Section IV presents the OPAL-RT platforms (software: RT-LAB and hardware: e.g. OP5600) for real time simulation and parallel computing in high data throughput situations. These platforms allow us to simulate the embedded information processing systems and the "brain" behind each individual autonomous vehicle, as well as the VANET exchange of information between each vehicle in collaborative scenarios. In section V, we provide a brief presentation of the Pro-SiVIC simulator and explain our choice due to its high capability to simulate the other essential components of multi-vehicular collaborative scenarios such as the various sensors, the road environment and the dynamic of each vehicle in real-time. In section VI, we propose to interconnect RT-LAB and Pro-SiVIC platforms to complete the required architecture to simulate in real-time scenarios of collaborative autonomous vehicles. In section VII, we give an example and present some results using a real time ACC (Adaptive Cruise Control) scenario with inter-vehicular communication capabilities. As the main embedded processing/control algorithms, we consider a PI controller, where the details are given in section VIII. In section XI, we analyze and comment the data obtained from the simulation, followed by a conclusion.

978-1-4799-3080-7/14/$31.00 ©2014 IEEE

II. Proposed simulation strategy

As the system complexity increases, the design, prototyping and validation of autonomous vehicles in multi-vehicular scenarios will become a major challenge. The computer power, memory space and data communication requirements for real-time simulation of these complex scenarios will need dedicated and special hardware/software architectures. The goal of this paper is to describe some solution elements which we propose based on the RT-LAB platform produced by OPAL-RT and the advanced vehicle simulator called Pro-SiVIC provided by Civitec. We consider several scenarios involving groups of cooperative vehicles equipped with given sets of sensors, and on board processing capabilities, able to share some of their data dynamically (e.g: position, speed, acceleration, etc). Whereas the Pro-SiVIC simulator allows to generate detailed simulation of the vehicles, their dynamic, their embedded sensors, the road and its environment. The RT-LAB platform allows to simulate the on-board real-time processing, to estimate the various parameters needed for generating the various control signals, and to be used by the Pro-SiVIC simulator, in order to control the dynamic of each vehicle involved in the scenario under study. The on-board processing of each vehicle may be quite different and rely on different computing resources. Typically, the set algorithms involved in each vehicle include some multi-sensor data fusion, detection and estimation processes as well as recognition, tracking and risk assessment analysis. Each vehicle has its own set of algorithms that may differ substantially from one vehicle to another, bringing further complexity to the multi-vehicular scenarios, which lead to intractable NP (Non-deterministic Polynomial) problems.

In order to assess the behavior of those complex systems, advanced simulation tools are required, which allow to randomize the various parameters in play and to sample the multi-dimensional operating space of the systems under study, either for design, testing or validation. Therefore, our goal is to develop a real-time simulator for collaborative autonomous vehicles, which will allow to assess and improve the algorithms behavior used to perform pre/post processing of heterogeneous multi-sources data before moving to physical prototyping. Thanks to real time simulation, we can minimize the cost in the design phase and have a better assessment of the systems while interacting in multi-vehicular scenarios, given the complexity of new technologies embedded in vehicles, and the requirement of compliance with safety standards from transportation agencies. This new concept allows in virtual mode to deal with specific, unlikely, dangerous cases must be identified, before real prototyping, in order to detect potential failures or defects before deployment on the market. Indeed, whenever a defect is detected a priori, OEMs (Original Equipment Manufacturers) can win hundreds of times to recall and repair cost as often seen in the automotive industry. Hence, extensive tests must be performed before large scale manufacturing of technologies used to build autonomous and connected vehicles.

III. Real-time simulator architecture for multi-vehicular scenarios.

To evaluate the performance of algorithms for processing and decision making in autonomous multi-vehicular scenarios, we need to use a set of simulation tools that can operate in real time and emulate real conditions as closely as possible. In this paper, we focus on the development of parallel architectures for the real-time simulation of data fusion algorithms, and real time processing of multi-sensors data. We deal in particular with cooperative localization and perception issues, in the context of active safety for autonomous vehicles.

Simulating these scenarios involves the detailed mathematical modeling of the vehicles dynamic, the road environment, the on-board sensors, and the real-time processing, communication and control systems embedded in each vehicle. Each vehicles processes its own data and transmits some of them to others. They independently operate and make their own decisions in real time. To simulate the different tasks performed in each vehicle, we use a set of processing cores operating in parallel on the OPAL-RT RT-LAB platform. Once the parallel processing of all vehicles (cores) is completed, vehicles states will be communicated to the others cores, in order to make further decision or accomplish other actions, such as improving its own ego (vehicle) state estimate.

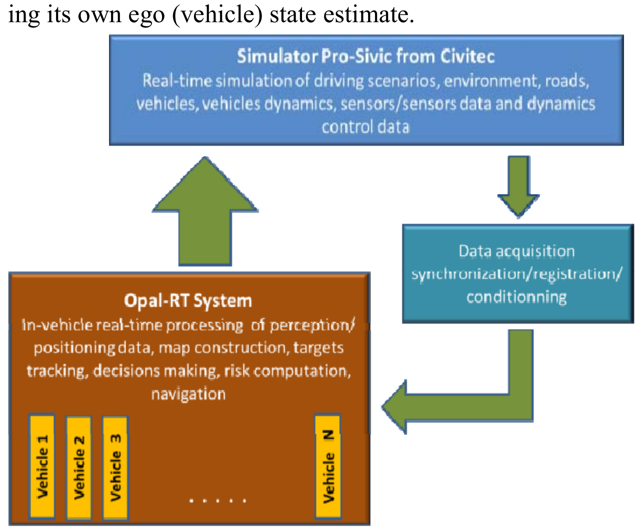

figure 1: OPAL-RT platform and cores attribution for parallel processing.

IV. The RT-LAB platform.

RT-LAB (Real-Time Laboratory) is a powerful real-time simulation platform, which allows to implement complex mathematical dynamic model that we design primarily on Simulink/MATLAB, for real-time integration with external equipments (HIL: Hardware In the Loop or RPC-Rapid Prototyping Controller modes) forming closed-loop systems. It offers the possibility to separate complex model into several simple sub models (subsystems), and processes them in parallel architectures to perform complex tasks [5-12]. Its Input/Output (I/O) modules allow user to communicate with real environment during simulation, which provide the possibility to test a multitude of products in real life.

The XHP (eXtra High Performance) mode allows very fast computation of real-time models on the target system, while using I/O modules at high frequency [10]. Its great computing speed allows designer to work on all types of real time distributed complex systems such as the one considered here.

IV.1. Implementation on OPAL-RT software (RT-LAB).

The RT-LAB builder process allows generating several C++ executables from Simulink model by distributing the multi-vehicular scenario into several subsystem (vehicle) elements. The code is optimized for real-time execution on LINUX-Redhat and QNX operating systems. Depending on available resources, OPAL-RT platforms cores are either automatically assigned to each subsystem built in Simulink, or manually trough assignment with graphical user interface.

Numerous capabilities of the platform allow on-line parameter adjustment, signal acquisition, data logging, and other commonly used features making model development and testing easier.

The platform includes data acquisition modules and signal processing software that is optimized for visualisation and analysis of real-time results. It may be used to simultaneously load, view, and process data from applications such as MATLAB and Com-trade. It also possesses advanced mathematical post processing capabilities, and is able to save and load templates associated to customized sessions of similar simulations.

RT-LAB allows several commercial softwares to quickly interface with other simulators such as Pro-SiVIC from Civitec including modelling tools: Plecs, Matrixx/System Build, Statemate, C/C++, FORTRAN, etc.

IV.2. OP5600 Simulator.

To implement our multi-vehicular scenarios, we use the OPAL-RT RT-LAB OP5600 simulator based on either Spartan 3 or Virtex 6 FPGA platforms and CPU [11], that makes it an ultrafast simulator, since it uses multiple cores for parallel computing.

This simulator can be used as a desktop or traditional rack mount for high precision and high speed in real time simulation. New improvements on I/O circuits allow the avoidance of input/output adaptors like standard connectors (DB37, RJ45, etc) and make monitoring easier.

The OP5600 simulator has the following features [11].

- Great computation power: Powerful real-time target (12 CPU cores 3.46 GHz), real-time OS (Linux Redhat), and distributed parallel computation.
- Important Inputs/Outputs capabilities: Up to 128 analog or 256 digital inputs I/O or mix of both, rear D-sub 37 connector for external devices, many chassis can be connected together to form larger I/O lines amount, etc.
- Connectivity: Embedded hard driver for real-time data logging, and various communication protocols are available and offered by RT-LAB to communicate between the subsystems (Dolphin, Ethernet, Firewire), and many others dedicated to automobile buses like, CAN, LIN, FLEXRAY, AFDX, ARINC 429.

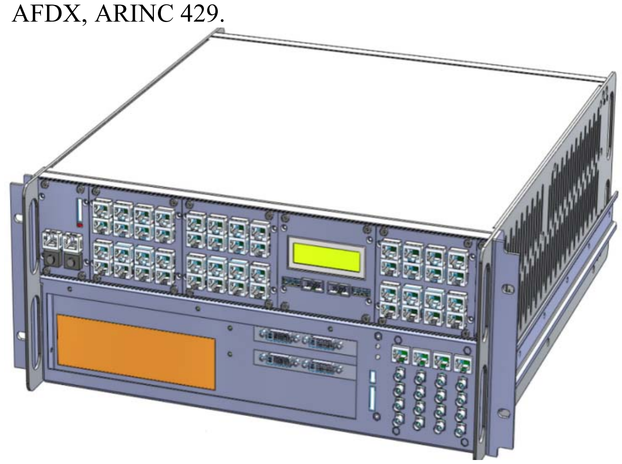

Figure 2: OP5600 simulator.

V. Pro-SiVIC simulator.

Nowadays, most of developments aim to improve the safety of road through driving assistance systems. These studies generally take into account vehicle perception and maneuvers (e.g. head angle changing, braking and accelerating). Generally, ego perception is no longer sufficient. Additional information is required to eliminate or at least minimize accidents on roads and make them safer by collaborative making decisions for driving assistance. This additional information requires additional resources which are both pricey and time-consuming for data processing. Therefore, it becomes essential to have real-time tools allowing to prototype and to assess, at each stage, the performance and improvement of collaborative and autonomous vehicles before design. Real time simulation platforms have to integrate physical models of vehicle environment, vehicles embedded sensors (proprioceptive, exteroceptive), and communicating devices. Other mathematical models should be added like vehicle dynamics coupled with actuators (steering wheel angle, torques on each wheel). Pro-SiVIC is a real time simulation tool and offers necessary requirements for developing and prototyping collaborative and autonomous driving systems with cooperative and environment perception.

Pro-SiVIC platform includes several categories of sensors and communication tools [13,15-17]. The exteroceptive sensors are cameras, laser scanner, RADAR and LIDAR. The proprioceptive sensors involve odometers and INS (Inertial Navigation Systems).

Communication protocols for collaborative and autonomous systems include both 802.11p communication media and beacon. Pro-SiVIC provides the possibility to tune and fix stamp time in real time, for each sensor, communication media, intrinsic and extrinsic parameters. The user could activate several operation modes, which can be modified during the simulation: 'Off' and 'On' to switch on or switch off a sensor. "Record" in order to collect data in a file, 'RTMaps', and 'MATLAB' to send sensor data for external applications. Figure 3 shows some scenarios with operating embedded sensors.

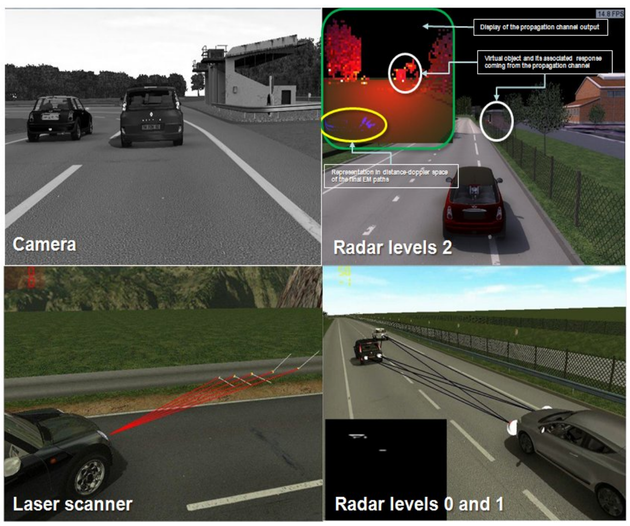

Figure 3: Some exteroceptive type of sensors in Pro-SiVIC.

VI. RT-LAB/Pro-SiVIC: Interconnected platforms for real time simulation of collaborative and autonomous vehicles.

Pro-SiVIC simulates the virtual prototypes of vehicles with their proprioceptive and exteroceptive embedded sensors. OPAL-RT RT-LAB platform coupled to the Pro-SiVIC simulator allows to simulate in real-time the embedded data processing and decision making processes for each vehicle involved in the scenario. Pro-SiVIC provides all the required models to simulate accurately the vehicles, the road environment, and the sensors embedded in each vehicles. Therefore Pro-SiVIC allows to replace real-life data by simulating the dynamic of multiple vehicles, allowing RT-LAB to operate in SIL mode (Software In the Loop). Both interconnected platforms thus provide us with full platform, for advanced real time data processing, inter-vehicle communication, prototyping, validation of the control and perception algorithms for real time simulation of collaborative autonomous vehicles.

RT-LAB processes at high speed the virtual data coming from SiVIC's virtual vehicles and their respective virtual embedded sensors. The RT-LAB output results are used as input control signals back to the Pro-SiVIC simulator in order to control the vehicle dynamics. Hence, once virtual processing and decision making are completed on the RT-LAB platform, control signals are sent in real time from the RT-LAB platform to Pro-SiVIC simulator to control the set of virtual autonomous vehicles. Thanks to RT-LAB/Pro-SiVIC platform all data processing and control algorithms, once tested and validated using extensive simulations, can be directly transferred and used in real hardware vehicles.

VII. Example: ACC (Adaptive Cruise Control) design.

As a simple example to illustrate our simulator, we consider an adaptive cruise controller in 2D (2 Dimensions) involving two vehicles maintaining a safe distance between vehicles, which is based on multi-sensor data and inter-vehicular communication exchanging data about their respective dynamic state.

After exchanging information about their dynamic state according to their respective spatial coordinates, the distance between the two vehicles is determined as follows:

$$D_{veh_1 veh_2} = \sqrt{(x_{veh_1} - x_{veh_2})^2 + (y_{veh_1} - y_{veh_2})^2} \qquad (1)$$

To improve the second vehicle's behavior, instructions must be taken, by considering the status (position, speed, acceleration, head angle, etc.) of the first one at any time, and maintain a safe distance in spite of various external perturbations and sensors noise. The second vehicle must be able to respond to tracking constraints, which justifies the choice of PI (Proportional Integrator) controller, so the position set reference of the second vehicles is determined as follows.

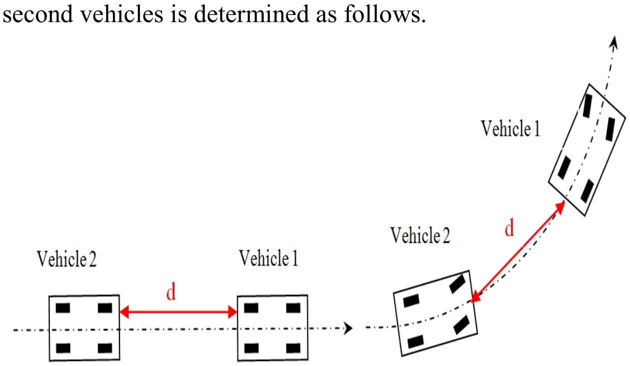

Figure 4 : Pattern of two cars for maintaining safety distance.

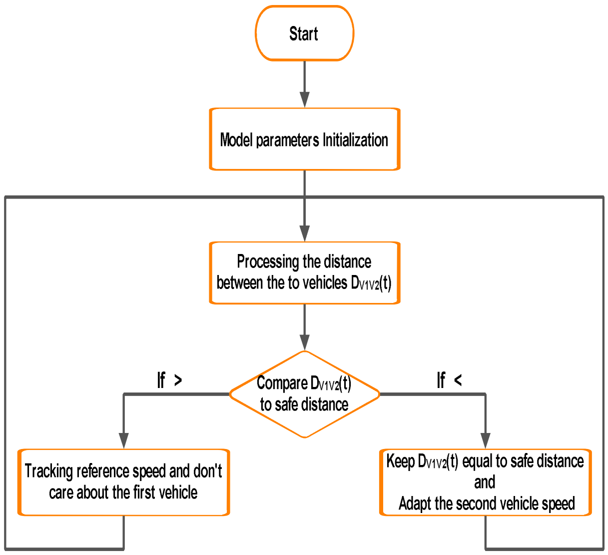

Figure 5: Adaptive Cruise Control flow chart.

$$\begin{cases} D_{V_1V_2}(t) = V_{V_2}(t) * t_d + D_{Safety} \\ x_{refV_2}(t) = x_{veh_1}(t) - D_{V_1V_2}(t)\cos\left(\tan^{-1}\left(\dfrac{y_{veh_1}(t) - y_{veh_2}(t)}{x_{veh_1}(t) - x_{veh_2}(t)}\right)\right) \\ y_{refV_2}(t) = y_{veh_1}(t) - D_{V_1V_2}(t)\sin\left(\tan^{-1}\left(\dfrac{y_{veh_1}(t) - y_{veh_2}(t)}{x_{veh_1}(t) - x_{veh_2}(t)}\right)\right) \end{cases} \quad (2)$$

Where:

D_{Safety}: Safe distance, t_d: driver's reaction time, $V_{V_2}(t)$: second vehicle current speed, $D_{V_1V_2}(t)$: inter-vehicular distance, $x_{veh_2}(t)$ and $y_{veh_2}(t)$ are the coordinates of the first and the second vehicles on "x" and "y" axis respectively.

VIII. PI CONTROLLER FOR LINEAR SYSTEMS.

Due to the PI controller, static errors are eliminated and external perturbations are largely rejected [18-23]. To construct the PI controller, additional state variables are required for the open loop linear systems, like tracking error integral, and the following augmented linear system is obtained:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \\ \dot{z}(t) = e(t) = y_{ref}(t) - y(t) \end{cases} \quad (3)$$

The extended model is written as follows:

$$\begin{cases} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & 0_{n\times q} \\ -C & 0_{q\times q} \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0_{q\times p} \end{bmatrix} u(t) + \begin{bmatrix} 0_{n\times p} \\ I_{q\times p} \end{bmatrix} y_{ref}(t) \\ y(t) = \begin{bmatrix} C & 0_{q\times q} \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \end{cases} \quad (4)$$

We then place the poles of the extended system by choosing adequate feedback gain matrices such that:

$$u(t) = -\begin{bmatrix} K_x & K_z \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \quad (5)$$

The following figure summarizes both controller and observer in state space closed loop:

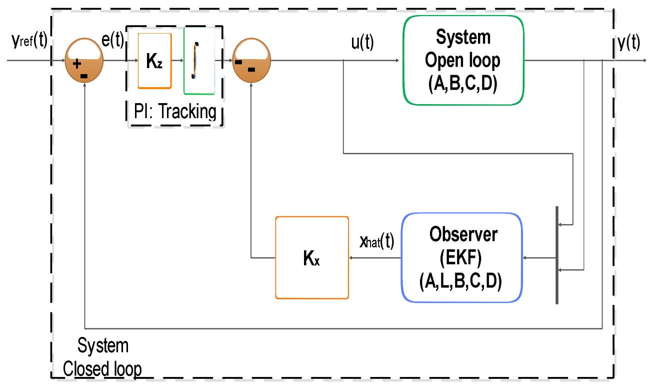

Figure 6: Closed loop scheme with PI and state feedback controllers.

IX. SIMULATION OF ADAPTIVE CRUISE CONTROL.

To simulate the two vehicles for Adaptive Cruise Control, the vehicles dynamic, data processing and control algorithms have been implemented on Simulink. Each vehicle was implemented by a subsystem, according to the RT-LAB protocol. Figure 7 shows the closed loop configuration.

We can see that the inter-vehicle distance D_{V1V2}(t) in figure 8 converges to the safe distance set to 30 meters, where t_d depends on driver type/nature (human driver $t_d=1.5s$, autonomous vehicle $t_d=10^{-3}s$, etc). The different spikes are caused by sensors noise which is filtered by the extended Kalman filter EKF algorithm on each vehicle during the simulation. The closed loop with PI controller offers the system robustness, tracking and precision, which rejected exogenous perturbations. The second vehicle is programmed to maintain inter-vehicle distance at least equal to a safe distance (set here to 30 meters).

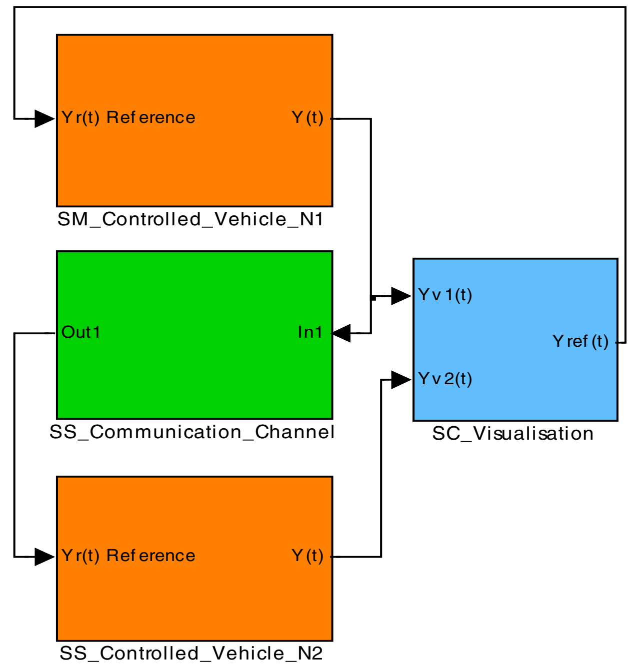

Figure 7: Simulink model of Adaptive Cruise control with RT-LAB standard configurations.

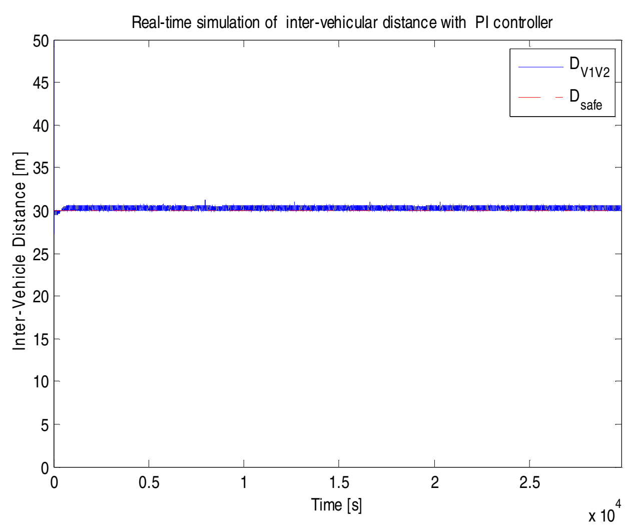

Figure 8: Real-time simulation of inter-vehicular distance with PI controller.

Inter-vehicles safe distance and communication delay effects are visible on figure 9, where the second vehicle's coordinates on both axis (OX,OY) are delayed in comparison to the first vehicle's coordinates. Data exchanges are done trough communication channel which is modeled by a delay of 10 ms. Figure 10 shows the vehicles' trajectory on Satory test track, the precision of the second vehicle depends not only on communication delay, but on safe distance, radius and rate-of-turn too, therefore, if safe distance and rate-of-turn are important, radius of-turn is short then the precision becomes low, and vice versa. To offer more precision to the system, road maps, curvilinear coordinates, and path historic of vehicles must be shared, to determine the exact inter-vehicle distance as a function of radius and rate-of-turns.

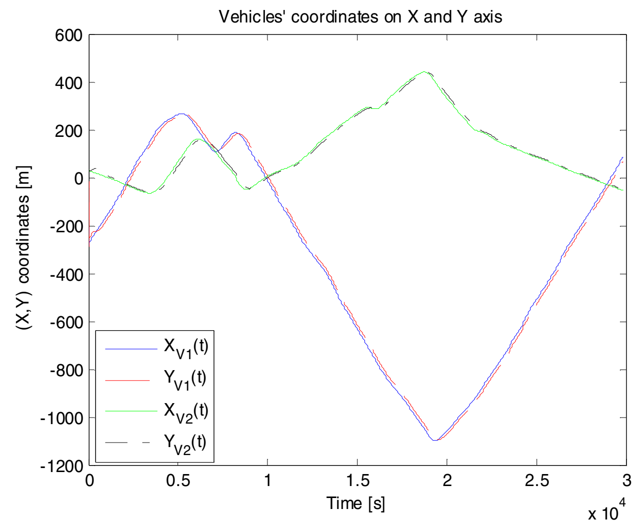

Figure 9: Real-time simulation of vehicles' coordinates evolution.

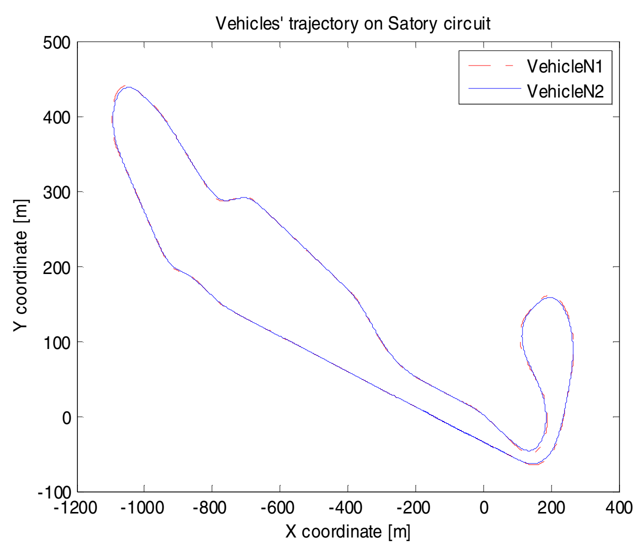

Figure 10:Vehicles' trajectories (Satory Circuit)

X. CONCLUSION:

This paper presents the first steps to design a real time simulator of collaborative and autonomous vehicles based on OPAL-RT Technologies, it offers several architectures for parallel computing, providing high speed processing and high precision, and Pro-SiVIC simulator from Civitec, we used to simulate the road environment and vehicles' dynamic as well as the embedded sensors. The main objective is to couple the two platforms (RT-LAB/Pro-SiVIC) for real time simulation of sets of autonomous vehicles, each one having its own sets of sensors and embedded processing/ communication capabilities. This paper has shown some preliminary results with a basic scenario with two autonomous vehicles corresponding to an adaptive cruise controller (ACC).

XI. ACKNOWLEDGEMENT

This work is part of COOPERCOM, a 3-year international research project (Canada-France). The authors would like to thank the National Science and Engineering Research Council (NSERC) of Canada and the Agence nationale de la recherche (ANR) in France for supporting the project.

REFERENCES

[1] Hong Cheng; "Autonomous Intelligent Vehicules, Theory, Algoritms, and Implementation"; Springer Advances in Computer Vision and Pattern Recognition; 2011.

[2] Zhefu Shi, Cory Beard, and Ken Mitchell, " Analytical models for understanding space, backoff, and flow correlation in CSMA wireless networks", Springer Science+Business Media, LLC 2012, Published online: 3 July 2012.

[3] Zhefu Shi, Cory Beard, and Ken Mitchell, "Competition, Cooperation, and Optimization in Multi-Hop CSMA Networks with Correlated Traffic", International Journal of Next-Generation Computing, Vol. 3, No. 3, November 2012.

[4] Jonathan Baber, Julian Kolodko, Tony Noël, Michael Parent; "Cooperative Autonomous Driving, Intelligent Vehicles Sharing City Roads Cooperative Autonomous Driving", IEEE Robotics & Automation Magazine, march 2005.

[5] John Chiasson, Leon Tolbert, Keith McKenzie and Zhong; "Real-Time Computer Control of a Multilevel Converter using the Mathematical Theory of Resultants"; Proceedings of the Electrimacs Conference, Montreal, Canada, August 18-21, 2002.

[6] M. A. Ouhrouche, N. Léchevin and S. Abourida; "RT-LAB Based Real-Time Simulation of a Direct Field-Oriented Controller for an Induction Motor"; Electricas 2002, august 18-21,2002.

[7] Christian Dufour, Jean Bélanger, Tetsuhiro Ishikawa, Kousuke Uemura; " Advances in Real-Time Simulation of Fuel Cell Hybrid Electric Vehicles "; Proceedings of the 21st Electric Vehicle Symposium (EVS-21), Monte Carlo, Monaco, April 2-6 2005.

[8] Simon Abourida, Christian Dufour, Jean Bélanger; " Real-Time and Hardware-In-The-Loop Simulation of Electric Drives and Power Electronics: Process, problems and solutions "; International Power Conference, Toki Messe, Niigata, Japan, 04 April - 08, 2005.

[9] Christian Dufour, Cacilda Andrade, Jean Bélanger; "Real-Time Simulation Technologies in Education: a Link to Modern Engineering Methods and Practices "; Proceedings of the 11th International Conference on Engineering and Technology Education ; INTERTECH 2010 Ilhéus, Bahia, Brazil, March 7-10, 2010

[10] Sergio Cavalcanti Marco Papini" Preliminary Model Matching of the EMBRAER 170 Jet "; Journal of Aircraft, Vol. 41, No. 4 (2004), pp. 703-710.

[11] RT-LAB User Guide Version 10.7 (RTLAB-UG-107-00)

[12] OP5600 HILBOX, USER GUIDE, Real-Time Simulator

[13] D. Gruyer, S. Glaser, B. Monnier, 'Simulation of vehicle automatic speed control by transponder equipped infrastructure', in proc. IEEE ITST2009, Lille (France), october 2009.

[14] B. Vanholme, D. Gruyer, S. Glaser, S. Mammar, 'Fast prototyping of highly autonomous cooperative driving system for public roads', in proc. IV2010, SanDiego, June 2010.

[15] D. Gruyer, S. Glaser, B. Monnier, 'SiVIC, a virtual platform for ADAS and PADAS prototyping, test and evaluation', in proc. FISITA2010, Budapest (Hungary) 30 may-4 June 2010.

[16] D. Gruyer, et al, 'A new generic virtual platform for cameras modeling', in proc. of International Conference VISION 2010, Montigny le Bretonneux (France), 6-7 october 2010.

[17] B. Steux, 'RTMaps, un environnement logiciel dédié à la conception d'applications embarquées temps-réel. Utilisation pour la détection automatique de véhicules par fusion radar/vision', PhD. thesis from Ecole des Mines de Paris, decenber 2001.

[18] C.S.Ravichandran; S. Subha Rani and; T. Manikandan. " Designing of PID Controller for Discrete Time Linear System Using Balanced Approach Reduced Order Model", American Journal of Applied Sciences, 2007.

[19]Y.P. Chang, et al, " Digital Modeling and PID Controller Design for MIMO Analog Systems with Multiple Delays in States, Inputs and Outputs", Circuits System Signal Process 2009, P: 111–145.

[20] Y. Zhang, et al," Digital controller design for analog MIMO systems with multiple I/O delays ", IEEE Conference Publications, Taiyuan, China, 23-25 May 2012 P: 39 - 44.

[21] Jennifer M. Madsen, Leang-San Shieh, and Shu-Mei Guo. " State-Pace Digital PID controller design for multivariable analog systems with multiple time delay", Asian Journal of Control, Vol. 8, No. 2, June 2006, pp. 161-173.

[22] Han-Qin Zhou ; Leang-san Shieh ; Qing-Guo Wang ; Liu, C.R." State-space digital PID controller design for linear stochastic multivariable systems with input delay, Control and Automation, 2005. ICCA '05. International Conference on, 2005.

[23] Baibanou Hassania, Pierre Sicard, and Abdellfattah Barazzouk; "Solutions to Typical Motor Load Emulation Control Problems"; Electricas 2002, august 18-21,2002.