Simulation and Test of Communication in Multi-Robot Systems using Co-Simulation

Thiago J. S. Oliveira
Center of Informatics - CI
Federal University of Paraiba - UFPB
João Pessoa, PB, Brazil
thiago.oliveira@dce.ufpb.br

Alisson V. Brito
Center of Informatics - CI
Federal University of Paraiba - UFPB
João Pessoa, PB, Brazil
alissonbrito@ci.ufpb.br

Abstract*— Multi-Robot System consisting of multiple interacting robots, each running a specific control strategy, which is not driven centrally. The fact that robots go to work cooperatively with the goal to solve a problem, caused such a system would increase its complexity. Since the multi-robot systems are becoming increasingly complex, it became difficult to design and simulate. Its development process involves important design decisions, communication between teams from different areas, integration with hardware, software, sensors and other things. All this, when together, should be very well modeled and simulated. However, make tools, methodologies and teams from different fields can work together is not an easy task to accomplish. Academic and industrial communities have developed standards and tools for this purpose. High Level Architecture (HLA) was developed to provide a simpler way to connect different simulators. HLA provide a specification of a common architecture to several simulation tools in Department of Defense of United States. It is a pattern described in IEEE 1516 series and has been developed to provide an architecture for distributed model and simulation. In this work, HLA is fundamental to integrate all the simulators involved. This paper presents a project for simulation of communication in Multi-Robot Systems (ROS Stage) using co-simulation. To this end, the standard used is the High Level Architecture (HLA), that provide a more flexible way to interconnect different simulators. The simulation uses the Omnet++ for communication, a simulator done in C ++, originally built for network simulations. This simulator is responsible for communication between the Multi-Robot Systems, so that it is possible to exchange information. Using the Omnet++, delays, packet losses and other things may be added. That way, the simulation will be more realistic.

Keywords*— *Multi-Robot Systems, Co-simulation, Omnet++, High-Level Architecture

I. Introduction

Multi-robot systems (MRS) consist of multiple interacting robots, each executing an application-specific control strategy, which is not centrally steered (CALISKANELLI; BROECKER; TUYLS, 2014). In a distributed MRS there is no centralized control mechanism—instead, each robot operates independently under local sensing and control, with coordinated system-level behavior arising from local interactions among the robots and between the robots and the task environment (LERMAN et al., 2006). The development of robot software is a demanding discipline. Technical challenges arise from the need to develop complex, software-intensive products that take the constraints of the physical world into account (BROENINK; NI; GROOTHUIS, 2010).

In 2007, a group of scientists, industry and engineers create an open-source robotic framework called Robot Operating System (ROS) (Quigley et al., 2009). It is a flexible framework for designing robots, providing a collection of tools, libraries, and conventions aiming to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

To Nicolescu et al. (2007), the co-simulation represents one of the most popular techniques of validation in heterogeneous systems. According Souza et al. (2003), the fundamental principle of co-simulation is to provide support to execute different simulators in a cooperative way. It allows the union of heterogeneous simulations with different models of execution.

High Level Architecture (HLA) was developed in order to provide a simpler way to interconnect simulators. It provides a specification of a common architecture to several simulations tools in Department of Defense of United States (DAHMANN; FUJIMOTO; WEAKTHERLY, 1997). It’s a pattern described in IEEE 1516-series and has been developed to provide a common architecture to distributed model and simulation (SYMINGTON; MORSE; PETTY, 2001).

OMNeT++ is a C++-based discrete event simulator for modeling communication networks, multiprocessors and other distributed or parallel systems (VARGA et al., 2001). Using the OMNeT++, delays, packet losses and other things may be added. That way, the idea of using OMNeT++ is to make the simulations more realistic as possible, once it brings factors that exist and affect the communication among devices connected in any network.

In this work, we use these three concepts to provide a heterogeneous architecture aiming simulate and test the main impacts of utilization of a network simulator in Multi-Robot Systems simulations, through High Level Architecture.

The remaining of this paper is organized as follow. Section 2 presents some related works. The architecture is proposed in Section 3. The initial results are presented in Section 4. Conclusions, discussions and main difficulties can be found in Section 5. Finally, Section 6 is about the future works.

II. Related Works

An operational semantics of co-simulation allows the discrete and continuous models to run on their respective simulators and managed by a coordinating co-simulation engine (FITZGERALD et al., 2013). Some works in literature use the concept of co-simulation to provide a solution in heterogeneous and distributed simulations.

BRITO et al. (2013) proposed the development and evaluation of a solution to model and simulate heterogeneous Models of Computation (MoCs) in a distributed way integrating Ptolemy II and High Level Architecture (HLA), creating an environment to run heterogeneous models of large scale in high-performance.

SOUZA et al. (2003) presented an environment that aims integrate virtual components on models of distributed co-simulations. The co-simulation is based in a modified version of High Level Architecture, called Distributed Co-Simulation Backbone (DCB). It imposes proprietary standards for data exchange and requires explicit calls to functions presents in Run-Time Infrastructure (RTI).

SUNG et al. (2009) integrates the Matlab and DEVSim++ in a unique and distributed simulation environment through High Level Architecture in way to obtain good results with regard to integration of two Models of Computation (MoCs) with hybrid simulators.

PINCIROLI et al. (2012) presented a Multi-System Robot simulator named ARGoS. It was projected to simulate complex experiments involving large swarms of different robots. According the authors, the simulator is, in same time, efficient and flexible, allowing high level of customization.

STRAßBURGER et al. (1998) proposed a distributed simulation of traffic using High Level Architecture, where a set of interoperable federations cooperates and communicate through the RTI's HLA.

ROTH et al. (2014) developed a framework for rapid integration of different simulators. A car-to-car communication application was presented, where SystemC is used to model the electronic controller of the car, OMNeT++ is responsible simulation of communication and Sumo simulates car traffic.

III. The Architecture

Next, is presented the environment to simulate Multi-Robot Systems through co-simulation, objective of this paper. The architecture can be seen in figure below.

The proposed environment have two parts. The first part composes all the ROS environment, the robots, its interface (interface ROS) and core, responsible for coordinate all the ROS environment in simulation. The second part is the HLA environment, with RTI, the ambassador, responsible for communication with the ROS interface and the OMNeT++ ambassador that communicates with RTI and forward information from OMNeT++ simulator.

The intersection point between the environments is called bridge. It implements a ROS node Interface and RTI ambassador to aim messages from RTI to ROS and vice versa. The bridge checks the ROS variables of all robots (such, speed, position) and send this information to HLA environment. This way, is possible to exist several robots sharing their data with any other simulator, such OMNeT++ or another HLA simulator. The co-simulation environment's flow is present in following figure. The co-simulation environment can be divided in two parts. The first, called Multi-Robot System Environment, and second, Communication.

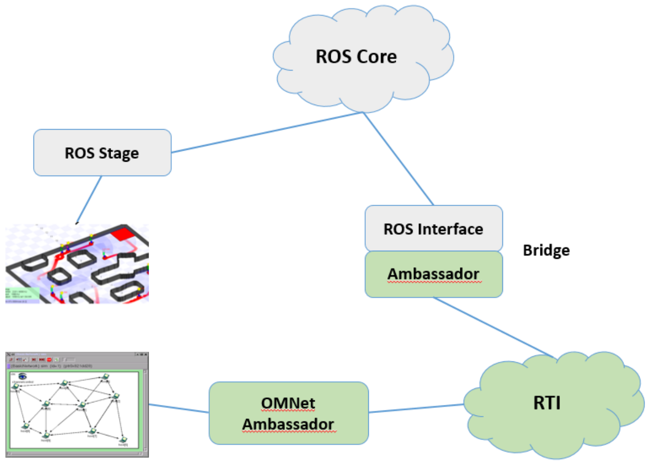

Figure 1 - Proposed Architecture

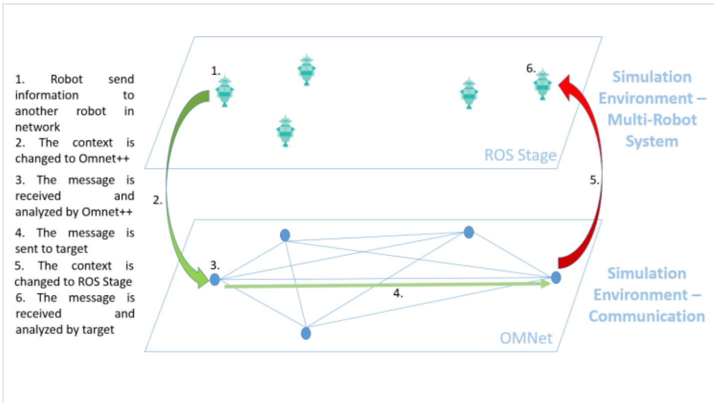

Figure 2 - Co-Simulation Environment

The Multi-Robot System Environment is responsible to simulate several robots in the same environment and use ROS, as stated earlier. It initiates the robot nodes that will transmit data to second layer of co-simulation environment, the Communication layer.

The Communication layer uses OMNeT++, responsible for exchange data with ROS Stage. The way is possible to simulate robots interacting in the network and test various situations presents in this type of communication (network protocols, packet loss, and interferences), making simulations even closer to reality.

A. *Simulations*

To assess the impact caused by addition of OMNeT++, two types of simulations are created. The first uses ROS Stage and HLA and a federate (called Federate01) exists to send information to ROS and be possible to move the robots according to received messages.

This kind of simulation executes in "ideal environment", once that do not exist any interference in communication among robots existing in ROS Stage. The simulation will be the bases to compare that will be made using the second type of simulation. The following figure shows the ROS Stage, with two robots moving according the received message from HLA.

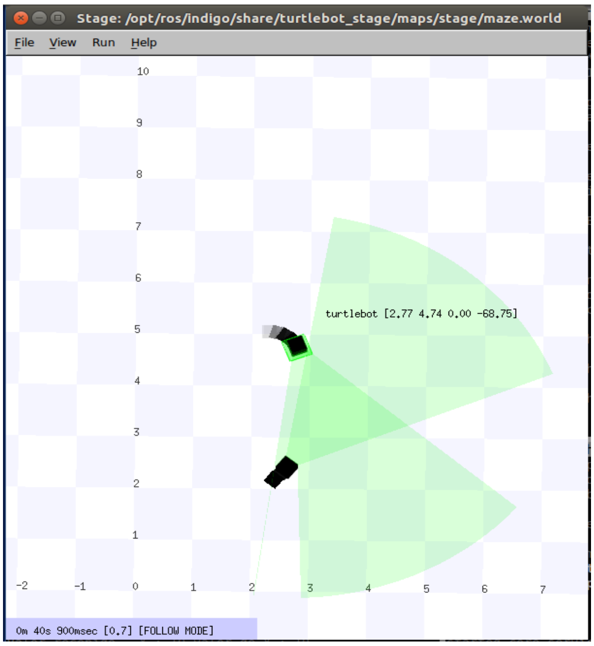

Figure 3 - Stage ROS in Execution

In the second type of simulation, the OMNeT++ simulator (figure 6) is responsible for exchange messages among the robots present in ROS Stage. The utilization of OMNeT++ allows delays, packet loss and other things, in way to let the simulation even closer to reality. That way, we can compare the executions of the two types of simulation and observe the impacts that simulated network brings to communication among Stage robots.

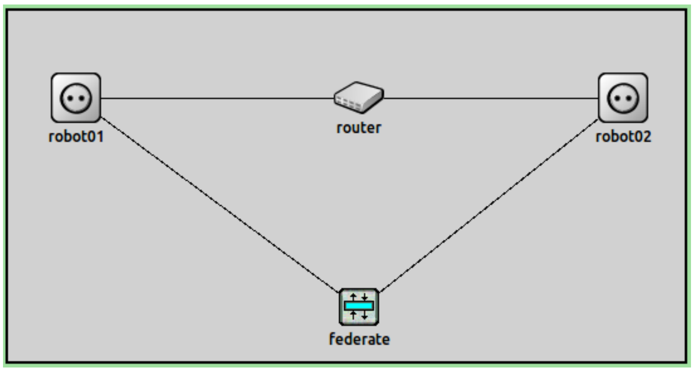

Figure 4 - OMNet++ Simulation

In order to make OMNeT++ capable to exchange data with HLA is necessary that exists a node called Federate. His function is exchange information between the network simulator and HLA without interfering network communication, or in other words, not interfering the communication inside OMNeT++, i.e. does not have latency of communication, packet loss or any other aspect related to network. In this way, the message flow is showed in figure 2.

The network used in experiments have two robots, a router and a federate. The Robots represents the ROS's robots and all the information destiny to them should be pass through OMNeT's robots. Communication among OMNeT's robots and the router have 59 milliseconds of delay and five different percentages for packet loss (1, 2, 3, 4 e 5). Once the robot sends a message to another, a counter is initiated and 1200 milliseconds after, if the confirmation message don't has arrived, the same message is sent again. Next, will see some results about the simulations and discusses futures researches.

IV. Partial Results

The tests in two types of simulations were performed using 50, 100, 200 and 400 as simulation time (in HLA) to all simulators. In total, 20 simulations were performed for each type of simulation. In this paper, two aspects were observed when we compare the two types of simulations: Execution Time and Data Exchanged. In terms of OMNeT++, there are other aspects to observe, such Packet losses, internal messages and how these two aspects contribute to Execution Time of simulations. The following table shows a big picture of data collected in simulations.

Table 1 - Data Simulations Overall

Simulation Time	Execution Time		Bytes Sent		Bytes Received	
50	50590	64098	10713	32111	25614	87179
100	100379	127960	20436	62671	48943	169036
200	200199	255498	40117	123158	95796	331845
400	399876	511146	79775	245153	189816	659524

ROS

ROS + OMNeT++

The main conclusion about table 01 is that the three aspects have a considerable increase after the inclusion of OMNeT++ but we will discuss these aspects separately and in more detail.

About the Execution Time (see Table 02) in OMNeT++, the time is calculated taking into account the Simulation Time, the delay's communication and the time generated for packet losses. That way, the Execution Time tends to increase considerably in cases of high values for packet loss percentages and how higher is the simulation time, once the delay affects every communication between the robots.

The Table 2 shows the packet losses in the simulations performed during the tests using the average of lost packets in simulations. Taking into account that every packet loss increases the Execution Time in 1790 milliseconds (counter + delay), the 21 packets losses in simulations with simulation's time equals to 400, will increase the Execution Time in 37590 milliseconds. The other impact of packet losses is the increase of data exchanged. In this simulations with time equals 400, in average, 400 messages are sent, 354856 bytes are received and 6.6 messages are lost per execution. This way, every message has 887 bytes, what results in 5854 bytes of missing data per execution. The following chart shows the amount of missing data in this simulations. In 10 simulations, 66 packets were lost resulting in 385430 bytes of data missing.

The second aspect observed was data exchanged between the simulators. The following table shows the total of data exchanged in two types of simulations. After the inclusion of OMNeT++, the amount of exchanged data was multiplied by 3,33 (average). This happens because the quantity of data exchanged is proportional to number of federates existing in HLA environment plus the messages generated by own OMNeT++, once that every message in HLA is received by every Federate, even if this federate will not catch the message. Beyond that, all massages destiny to a robot in ROS must pass through the correspondent robot in OMNeT++. In simulations presents in this paper, the ROS ignore the messages that not come from OMNeT++, not communicating with other federates of HLA. Following are presented some considerations and future works of this research.

Table 2 - Missing Data in Simulations with Time Equals 400

Bytes Received	Average Data per Message	Packet Losses	Missing Data
354489	886,2	9	7976
354481	886,2	3	2659
354688	886,7	4	3547
354575	886,4	5	4432
355128	887,8	7	6215
355120	887,8	7	6215
355223	888,1	3	2664
355062	887,7	7	6214
354876	887,2	10	8872
354915	887,3	11	9760

V. Final Considerations

Since the proposed architecture was executed with success, is necessary concentrate the efforts to analyze the impacts of various factors that influence the execution of simulation and robot's behavior in ROS Stage. In OMNeT++, for example, the latency and packet loss directly influence the time of simulation as a whole (managed by HLA). One possible approach is make the OMNeT++ responsible to manage the time of Simulation Time. That way, will not exist difference between the Time Execution. In ROS Stage, some configurations influencing the robot's behavior, such as Rate, that defines the frequency of robot's cycles, especially in cases of packet losses. The aspects observed in ROS Stage will be researched in more detail to know how exactly is the impact of them in the simulations and the behavior of the robots.

References

[1] Brito, Alisson V., et al. "Development and evaluation of distributed simulation of embedded systems using ptolemy and hla." Distributed Simulation and Real Time Applications (DS-RT), 2013 IEEE/ACM 17th International Symposium on. IEEE, 2013.

[2] Broenink, Jan F., Yunyun Ni, and Marcel A. Groothuis. "On model-driven design of robot software using co-simulation." (2010).

[3] Caliskanelli, Ipek, Bastian Broecker, and Karl Tuyls. "Multi-robot coverage: A bee pheromone signalling approach." Proceedings of the International Conference on Artificial Life and Intelligent Agents (ALIA 14). 2014.

[4] Dahmann, Judith S., Richard M. Fujimoto, and Richard M. Weatherly. "The department of defense high level architecture." Proceedings of the 29th conference on Winter simulation. IEEE Computer Society, 1997.

[5] Fitzgerald, John S., et al. "A formal approach to collaborative modelling and co-simulation for embedded systems." Mathematical Structures in Computer Science, Apr 2013, pp. 726-750.

[6] Lerman, Kristina, et al. "Analysis of dynamic task allocation in multi-robot systems." The International Journal of Robotics Research, Mar 2006, pp. 225-241.

[7] Nicolescu, Gabriela, et al. "Methodology for efficient design of continuous/discrete-events co-simulation tools." High Level Simulation Languages and Applications-HLSLA. SCS, San Diego, CA, pp. 172-179, 2007.

[8] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System."ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

[9] Pinciroli, Carlo, et al. "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems." Swarm intelligence 6.4, pp. 271-295, 2012.

[10] Roth, Christoph, et al. "A Simulation Tool Chain for Investigating Future V2X-based Automotive E/E Architectures." Embedded Real-time Software and Systems (ERTSS). 2014.

[11] de Mello, Bráulio Adriano, et al. "Tangram: Virtual integration of IP components in a distributed cosimulation environment." IEEE Design and Test of Computers, May 2005, pp. 462.

[12] Straßburger, Steffen, et al. "Internet-based Simulation using off-the-shelf Simulation Tools and HLA." Proceedings of the 30th conference on Winter simulation. IEEE Computer Society Press, 1998.

[13] Sung, Chang Ho, Jeong Hee Hong, and Tag Gon Kim. "Interoperation of DEVS models and differential equation models using HLA/RTI: hybrid simulation of engineering and engagement level models." Proceedings of the 2009 Spring Simulation Multiconference. Society for Computer Simulation International, 2009.

[14] Symington, Susan, Katherine L. Morse, and K. Petty. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification (IEEE Std 1516-2000)." (2001).

[15] Varga, András. "The OMNeT++ discrete event simulation system."Proceedings of the European simulation multiconference (ESM'2001). Vol. 9. No. S 185. sn, 2001.

[16] Onoe, Yuko, et al. "Cooperation of multiple simulators for Mobile IP networks." Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on. IEEE, 2003.