Simulation Environments for Space Robot Design and Verification

J.-C. Piedbœuf† M. Doyon† R. L'Archevêque† E. Martin†
M. Lambert‡

†Canadian Space Agency, 6767 Airport Road, St-Hubert, Quebec, Canada, J3Y 8Y9, jean-claude.piedboeuf@space.gc.ca

‡Opal-RT, 1751 Richardson, bureau 2525, Montreal, Quebec, Canada, H3K 1G6, michel.lambert@opal-rt.com

1 Introduction

An important characteristic of space robots is their inability to support their own weight. Design, testing and training for these space robots represent a challenge since we cannot use these robots in one-g environment. Therefore, simulation tools coupled with hardware components are essential in design, testing and training. In this paper, we will cover two specific applications: the SPDM Task Verification Facility (STVF) and the generic training of astronauts on robotics system using real-time simulation. These two applications rely on SYMOFROS, a tool developed at the Canadian Space Agency for modeling, simulation (in non real-time (NRT) and in real-time (RT)) and control of robots with flexible links and elastic joints.

2 Task Verification: STVF

Canada's contribution to the International Space Station (ISS) is the Mobile Servicing System (MSS), which is composed of the Mobile Transporter, the Space Station Remote Manipulator (SSRMS) and the Special Purpose Dexterous Manipulator (SPDM). The SPDM will be used to manipulate Orbital Replacement Units (ORUs) or scientific payloads. An important aspect of a typical SPDM task is the insertion/extraction of payloads.

Due to the complexity of an SPDM task, a verification of the operation must be performed on the ground for each ORU manipulation. The main difficulty in this validation is verifying the part of the task for which the SPDM end-effector or payload undergoes contact with the environment. This part is verified using a hardware-in-the-loop simulation (HLS) to generate the real contact force using mockup of the payload that needs to be manipulated.

The STVF Manipulator Testbed (SMT) [1] is used to perform the HLS. As illustrated in Figure 1, the output of a real-time simulator representing a space robot is used as the input to the ground robot controller. The real contact forces are measured and fed back to the simulator. This approach is very flexible since we can represent not only SPDM but also other space manipulators.

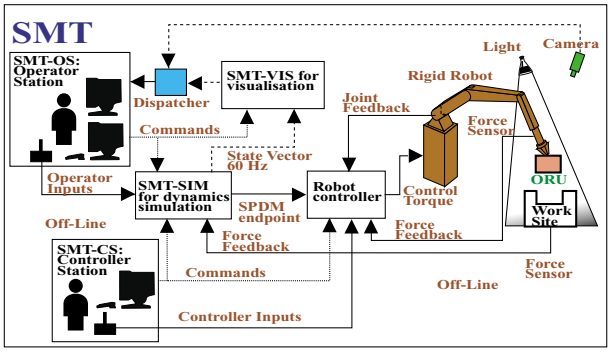

Figure 1: Concept of the HLS

Figure 2 shows the hardware architecture required for the test-bed. The real-time simulation is performed using the MSS Operation and Training Simulator (MOTS). The simulator includes the dynamics of the mobile base and the one of SSRMS in addition to the two arms of the SPDM. The full model has more than 50 degrees of freedom. The dynamic engine (SMT-SIM) is running at 1 kHz on an Origin 200 machine with four processors. The visualisation is running at 25 Hz on a four processors ONYX machine. The real-time control of the robot is achieved using a cluster of Pentium processors running QNX, and using Simulink Real-Time Workshop with Opal-RT RT-LAB for the code generation and multi CPU management. The graphical user interface on the SMT-CS is developed using Labview. The communication between Labview and the real-time system is managed by RT-LAB.

The models required for the controller and for the simulation on the cluster are generated using SYMOFROS. The robot controller in the HLS mode is based on a cartesian feedback linearisation [2]. For the design and the tune-up phases, we developed an equivalent model of the robot using SYMOFROS. This model reproduces exactly the same interfaces as the robot ones. Therefore, we can choose between

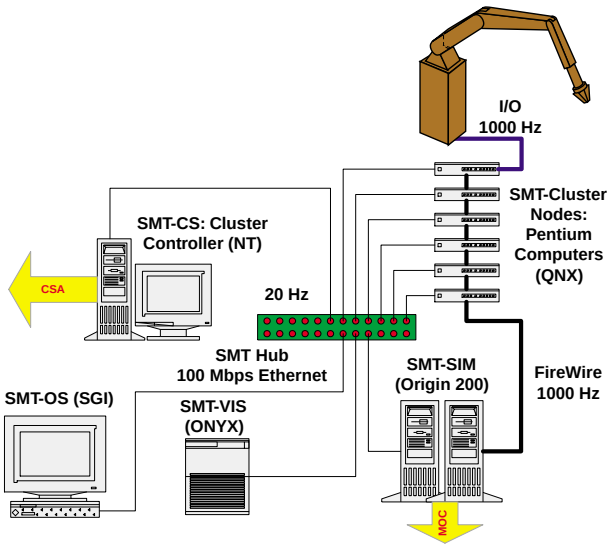

Figure 2: Computer Architecture for the HLS

the real robot and the simulated robot simply by clicking a switch. For the same reason, a simplified model with a reduced number of degrees of freedom has been developed for the space robot using SYMFROS. This model uses exactly the same interface as SMT-SIM. This flexibility is critical for the development since this allows the engineers to develop the overall software architecture in their offices and then download the code on the real-time system. There is no re-coding between the pure simulation phase and the HLS phase.

3 Generic Robotic Simulator

The assembly and maintenance of the International Space Station require the usage of complex robot systems like the Mobile Servicing System (MSS). The prospect astronauts background is normally science and research rather than engineering or military. In general, these new astronauts are not familiar with robots operation and they require robotics-training courses. For the above reason, the space agencies involved in the Space Station have the mandate to provide a Generic Robotic Training course. The main objectives of this course are to develop situation awareness (environment knowledge and safety issues), to provide the understandings of kinematics and dynamics of robotic systems, to increase the dexterity of the operator and finally to provide basic knowledge of robotics systems.

To support those objectives, CSA has developed a course based on traditional tutorial followed by training exercises and test operations performed on

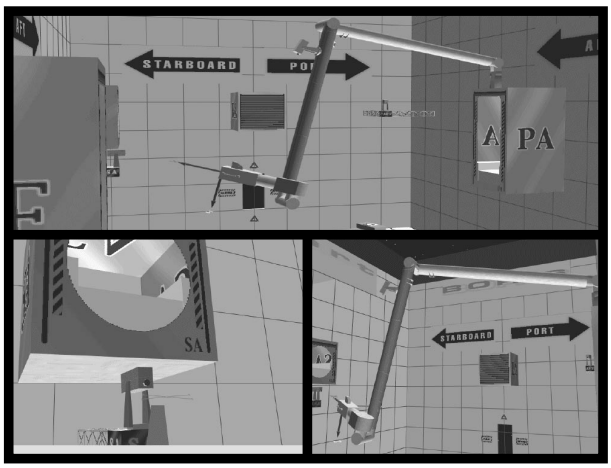

Figure 3: BORIS Visual Rendering Snapshots

a new CSA simulator called BORIS[1]. A 6 degrees-of-freedom generic robot located within a virtual spaceport may handle payloads attached to standard palettes and move them into docking modules. The robot may have rigid or flexible links and/or elastic joints according to the lesson level and these features can be selected in-line by the instructor. A fixed camera is attached at the robot end effector and a pan-tilt camera is located close to the robot elbow. The environment includes four additional pan-tilt cameras located at different strategic points of the spaceport.

Figure 3 presents the views provided by two spaceport cameras (top and right view) and one tip camera (left view). The operator may adjust the pan, tilt, zoom and light of those cameras using the Camera Control Panel Window. The robot may be controlled in cartesian, joint or trajectory mode by using a pair of hand-controllers (linear and rotational). To achieve palette capture, the operator may select a command frame, a display frame, a frame of resolution. Once these frames are configured, a palette target is used to properly to position and orient the robot end effector for the capture. A capture envelope allows the operator to accomplish the capturing task by using the Automatic Grappling Sequence based on the palette target. A safety module provides critical information related to singularities, self-collisions and joint position limits approaching. To increase the efficiency of the simulator usage, a snapshot module has been developed to store and retrieve particular robot and environment setup (robot position, camera parameters etc).

As shown in figure 4, the simulator is composed of four subsystems. The BORIS GUIs have been de-

[1]BORIS stands for Basic Operational Robotic Instructional Simulator

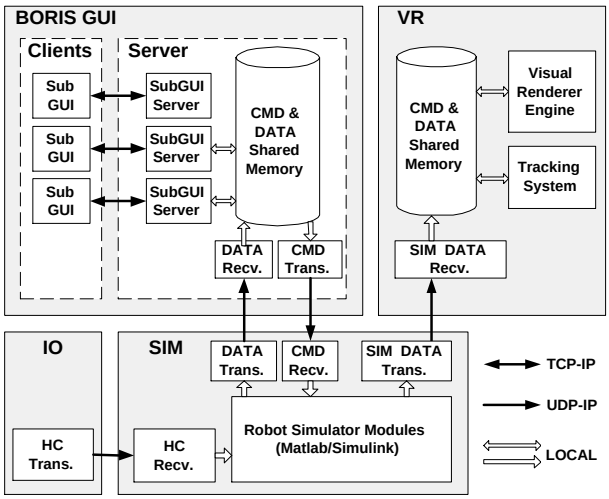

Figure 4: BORIS Software Architecture

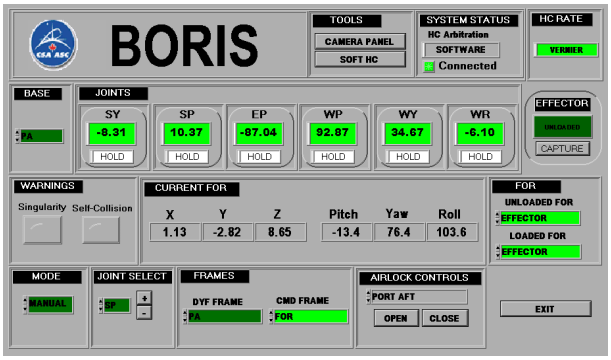

Figure 5: BORIS Main Operator GUI

veloped using Labview 5.1 and runs on Windows operating systems. The sub GUIs may be distributed among a list of PCs and communicate with a GUI server using TCP-IP protocol. Figure 5 presents the Operator Main Window. This GUI allows the operator to monitor data, e.g. the robot joints, the safety status. Similarly the GUI allows the operator to setup the command frame, the display frame and other robot parameters. Furthermore, BORIS provides Software Hand-Controller Window and Simulator Parameters Window sub GUIs.

The GUI server communicates at a 20 Hz rate with the robot simulator using UDP-IP protocol. The robot simulator has been developed using SYMOFROS, Matlab, Simulink and Real-Time Workshop. All dynamics parameters related to the robot and the payload attached are considered in the simulation. Windows, Linux and Solaris executables have been generated for implementation into the CSA real-time simulation environment. Figure 6 presents the top view of the robot simulator. Pseudo real-time Simulink blocks are present in order to simulate a hard real-time system.

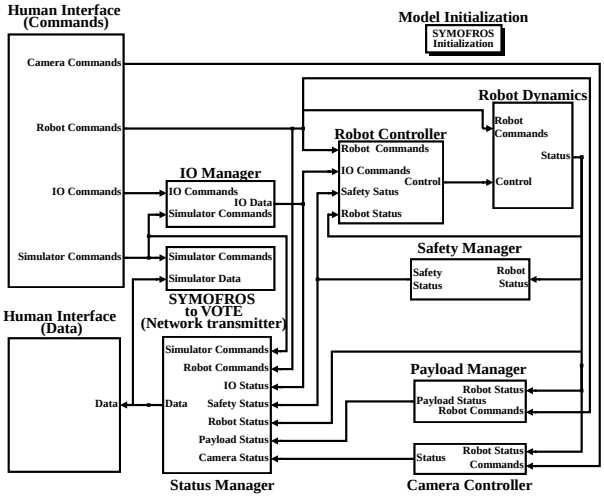

Figure 6: BORIS Robot Simulator Top View

To provide visual rendering, the robot simulator sends its status to a virtual reality engine at a 30 Hz rate. The virtual reality hardware includes a tracking system, two stereoscopic views provided a 3D helmet and three monitors. The operator may wear the helmet and look into the spaceport by a virtual window or look at the monitors to see the cameras view. The engine may also generate virtual tools like virtual ruler, virtual floor or virtual axis to better support the trainee during the first lessons. The full virtual reality engine runs on a SGI ONYX 4 CPUs with two graphics pipes. But a new visual renderer engine has been developed and is running on a single processor Linux system.

Official training of astronauts using BORIS will start at the end of November 2000. Additional research and development will be performed in order to implement free-flying objects, collision detection and contact dynamics. The open architecture of the system and the quality of the tools used allow a fast and efficient development.

4 SYMOFROS

SYMOFROS (Figure 7) is the acronym for Symbolic Modeling of Flexible Robots plus Simulation. This package has been developed since 1994 at the Canadian Space Agency, intially for research purposes and then for operational projects. It is based on Maple for modeling and code generation and, for now, on Matlab/Simulink for the simulation (NRT and RT). SYMOFROS is used to model mechanisms in tree topology with closed kinematics loops and non-holonomic constraints. It can handle rigid, flexible and elastic components for each part of the mod-

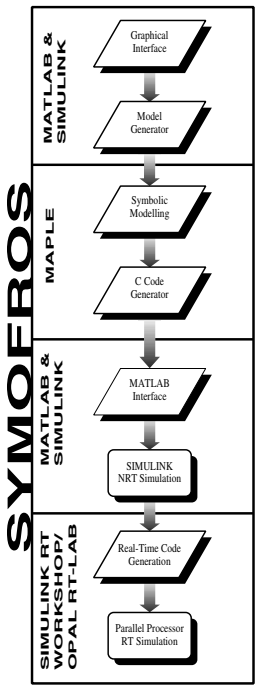

Figure 7: SYMOFROS: Functional Description

eled structure. The dynamic equations are built using Jourdain's principle. The kinematics and the dynamics are developed recursively to optimise the model. Maple's recursive procedures are used to obtain an efficient model generation tool (entirely symbolic).

A SYMOFROS model is first built using Simulink as a graphical user interface with a library of primitives. Once complete, it is exported to an ascii file. This file is then read and interpreted by Maple to generate C code for a set of functions related to the model. These functions are then made available in the Simulink environment for simulation.

By accessing the SYMOFROS model's functions with Simulink, it is possible to create a complete simulation within a short development time, e.g. BORIS simulator. In addition, the SYMOFROS components are compatible with Real Time Workshop (RTW) (companion product to Simulink). SYMOFROS can be targeted to RT-LAB software from Opal-RT to run the code on a cluster of PCs.

SYMOFROS is now used for various projects in robotics both at CSA and outside. The strong point of SYMOFROS is the capacity to take a graphical model and generate all the functions required for NRT simulation and more importantly the functionalities required for real-time control and simulation. This allows users with little knowledge in dynamics to generate complex model-based controllers, to test them in NRT simulation and to download the code on the real system. In the real-time environment the user can select the update rates of the different functions to improve the efficiency. This fast prototyping approach has been proven very efficient in various projects.

For the future release of SYMOFROS, we are working on different improvements. The first one is to increase the capabilities for closed-kinematic loops, to handle parallel manipulators (e.g. Stewart platforms). We are also working in a collaborative project to add a contact dynamics toolkit to SYMOFROS to allow the simulation of complex multi-points contact scenarios. A satellite docking simulator has already been developed using a third party contact dynamics toolkit. We will also increase the capabilities of SYMOFROS to handle more degrees of freedom.

5 Conclusion

Due to the complexity of the space robots and their inability to support their weight on earth, simulation tools are needed for the design and the verification of these manipulators. The Canadian Space Agency has developed SYMOFROS, a package that allow a user to develop quickly a model that can be used in NRT and RT simulation. Coupled with RT-LAB, SYMOFROS can generate a simulation or a controller that will run on a parallel plateform of computers. In this paper, two utilisations of SYMOFROS are demonstrated. The first one is the SPDM Task Verification Facility where SYMOFROS is used for the development and the implementation of a complex model-based control. The second application is BORIS, a simulator to be used as a generic tool for astronaut training. The main advantage of SYMOFROS over other systems available at CSA is its simplicity of use and the possibility of running on simple PC machines.

References

[1] F. Aghili, E. Dupuis, J.-C. Piedbœuf, and J. de Carufel, "Hardware-in-the-loop simulations of robots performing contact tasks," in *Fifth Int. Symp. on AI, R&A in Space* (M. Perry, ed.), (Noordwijk, The Netherland), pp. 583–588, ESA Publication Division, 1999.

[2] J. de Carufel, E. Martin, and J.-C. Piedbœuf, "Control strategies for hardware-in-the-loop simulation of flexible space robots," *To appear in the IEE Proceedings - Control Theory and Applications*, 2000.