MSS Ground Control Demo with MARCO

Klaus Landzettel¹, Bernhard Brunner², Günter Schreiber¹, Bernhard-Michael Steinmetz¹
Érick Dupuis²

¹Institute of Robotics and Mechatronics, DLR Oberpfaffenhofen, D-82234 Weßling, Germany
²Canadian Space Agency, 6767 route de l’aéroport, St-Hubert (Qc) J3J 8Y9, Canada
Klaus.Landzettel@dlr.de

Keywords teleoperation, ground control, high-level robot programming, autonomous robot, user interface, ISS, MSS, MBS, SSRMS, SPDM, MOTS

Abstract

A ground control programming and teleoperation system is described, which uses the Modular Architecture for Robot Control (MARCO) framework to teleoperate the Mobile Servicing System (MSS). A first implementation is built around the MSS Operations and Training Simulator (MOTS), to show the feasibility of controlling a space robot system on ISS from the ground.

1 Introduction

Canada’s contribution [1] to the International Space Station (ISS) is the Mobile Servicing System (MSS), which is composed of the Mobile Remote Servicer Base System (MBS), the Space Station Remote Manipulator System (SSRMS) and the Special Purpose Dexterous Manipulator (SPDM).

Until now the planned mode of operations for SSRMS and SPDM is teleoperation by an astronaut at the robotics workstation inside the ISS. But it is predicted that this way to operate the MSS will consume a lot of crew time because of the low velocities at which the SSRMS and SPDM will be operated and because of the potentially large displacements to be performed by these manipulators. As an alternative to reduce the load imposed on the astronauts a portion of the MSS operations could be conducted from a ground station. SSRMS and SPDM provide control modes that would be suitable for ground operations. However, ground control is hampered by communication link limitations such as time delays and reduced bandwidth and by the lack of good situational awareness of the operator. Situational awareness is impeded in any remote operation where the operator is limited only to equipment-mounted camera views with which to perceive the work site. To overcome these limitations a virtual reality approach is suitable. This provides to the operator the realistic feeling having the robotic system always fully under control. This requires a predictive graphical simulation of the entire robotic system on the ground control system, to compensate the relatively high data round trip time (more than 5-6 sec) as well as to provide an easy-to-use user interface for programming, controlling, and supervising the remote robot system. To ensure that MSS operations could be safely carried-out from the ground, it is necessary to demonstrate the proposed ground control technologies within a realistic environment.

2 Ground-Control Test-Bed

To validate the ground control concept for MSS in a representative environment, a test-bed has been developed on which the MARCO system can be integrated and tested. The objective is to faithfully reproduce the interfaces and dynamics of MSS as well as the communication limitations. One of the main components of the test-bed is the MSS Operation and Training Simulator (MOTS): a real-time dynamics simulator currently used for MSS operator training and for operation planning. It provides a high fidelity simulation of MSS operations accurately emulating the rigid and flexible body dynamics of MSS, its control software including the relevant control modes and features as well as all relevant environmental effects. To simulate MSS ground control from MARCO, an interface was added to MOTS to simulate the transmission of commands and telemetry in the same
fashion as the MSS will, through the ISS command and telemetry servers.

3 MARCO

The Modular Architecture for Robot Control (MARCO) developed by DLR is a spin-off of the ROTEX flight experiment [2] conducted by DLR in 1993. Subsequent to this experiment, the ground segment has been further developed to add more and more capabilities to the system. Over the last years, DLR has focussed its work in space robotics on the design and implementation of a high-level task-directed robot programming and control system. The goal was to develop a unified concept for a flexible, highly interactive, on-line programmable teleoperation ground station as well as an off-line programming system, which includes all the sensor-based control features partly tested in the ROTEX scenario. But in addition to the former ROTEX ground station it should have the capability to program a robot system at an implicit, task-directed level, including a high degree of on-board autonomy.

The current system provides a very flexible architecture, which can easily be adapted to application specific requirements. To increase autonomy, the programming and control methodology is based on an extensive usage of various sensors, such as cameras, laser range finders, and force-torque sensors. It combines sensor-based teleprogramming (as the basis for on-board autonomy) with the features of telemanipulation under time delays (shared control via operator intervention). Robot operations in a well-known environment, e.g. to support or even replace an astronaut in intra-vehicular activities, can be fully pre-programmed and verified on-ground – including the sensory feedback loops – for further sensor-based execution autonomously on-board. A payload user, who has normally no expertise in robotics, can easily compose the desired tasks in a virtual world. An intuitive man machine interface is provided to the operator. This interface includes a sophisticated VR-environment with DataGlove and high-performance graphics.

By the way, MARCO can also be used as a telepresence system without need of most of the graphical VR-environment. For service tasks in an unknown or only partly known environment, e.g. catching and repairing a failed satellite [3] or assembling and maintenance of ISS modules, a high amount of flexibility in programming and controlling is required. Additionally the operator must have the impression to directly manipulate the objects in the environment with the robotic system as a “prolonged arm” into the space. For that task, the possibility to immediately interact with the remote environment via haptic input devices and vision feedback must be given.

In 1999, MARCO was used to teleoperate the robot manipulator on the Japanese ETS-VII satellite [4]. The improved MARCO system will be used now to demonstrate that MSS operations could be safely carried-out from the ground.

3.1 Programming and Control Methodology

The MARCO system is based on a 2in2-layer concept (See Figure 1), which represents the hierarchical control structure from the planning to the executive layer:

On the user layer the instruction set is reduced to “what” has to be done (planning level). No specific robot actions will be considered at this task-oriented level. On the other hand the robot system has to know, “how” the task can be successfully executed, which is described in the expert layer (execution level). For details see [4].

3.2 Expert Layer

At the lowest system level, the sensor control mechanism is active. In analogy to the human, we named it Reflex, which means that all the actions initiated and performed at this level, will be executed fully automatically. A teaching by showing paradigm is used at this layer to show the reference situation, which the robot should reach, from the sensor’s view: in the virtual environment the nominal sensory patterns are stored and appropriate reactions (of robot movements) on deviations in the sensor space are generated. The expert programming layer is completed by the Elemental Operation (ElemOp) level. It integrates the sensor control facilities with position and end-effector control. In telemanipulation mode, the user generates position commands and selects the appropriate sensor control strategies for path refinement (shared control).

3.3 User Layer

The task-directed level provides a powerful man-machine-interface for a novice user, which is not familiar with robotics. An Operation is characterised by a sequence of ElemOps, which hides the robot-dependent actions. For the user of an Operation the manipulator is fully transparent. This means, that the user need not worry about the details of the robot, i.e. the robot action
is apparently a “hidden” one [5]. To apply the Operation level, the user has to select the object/place he wants to handle, and to start the Object-Place-Operation: via a 3D-interface (DataGlove or SpaceMouse) an object can be grasped and moved to an appropriate place. After the user has moved all the objects to their target locations, the execution of the generated Task can be started. The system provides status information and comprehensive quick look displays for task execution monitoring purposes.

4 Control of MSS

To validate the concept of operating MSS from ground, a demonstration scenario has been designed, where the MARCO software is used to drive the MSS Operations and Training Simulator (MOTS). The MARCO system has been adapted to the command and telemetry interfaces of MOTS via an interface task at the MARCO control station.

4.1 System interfaces

MOTS is used as a dynamic engine to close control loops on the ground simulation with the same behavior as expected on the real SSRMS in space. The current input devices, as SpaceMouse or DataGlove will be complemented by two joysticks, one for position and the other one for orientation control of the manipulator, to build a high fidelity replica of the on-board user interface.

![Architecture: MARCO Demonstration](image)

Figure 2 - Architecture: MARCO Demonstration

This configuration is suitable to provide all the functionalities to telemanipulate the SSRMS as the astronauts will do it. In addition to that we can demonstrate MARCO’s control features such as task decomposition, path planning, collision avoidance, redundant kinematics, sensor based control, and shared control. The MARCO architecture for controlling robots in space includes all the features to program, control, and supervise the MSS on the ISS. Various teleoperation modes are available from direct telemanipulation to task-oriented programming and execution.

To establish the command and telemetry data transmission links, the DLR’s telerobotic system has been interfaced to MOTS using a client-server architecture. An interface-computer is used to translate data between MARCO and MOTS. This approach to adapt the MARCO interface structures to the existing interfaces has been proven very well during the ETS-VII space robot mission [4].

![Interface structure of the MARCO/MOTS demonstrator](image)

Figure 3 Interface structure of the MARCO/MOTS demonstrator

4.2 Sensor-based local autonomy

As demonstrated with the ROTEX experiment [2], the predictive graphics approach, which also includes the simulation of all the available sensors (force-torque, distance, vision, etc), is a very helpful tool to predict and verify the progress of an operation using local control loops, based on sensory data. With appropriate modelling, this can even be performed on-ground.

Ideally, autonomous behaviors, i.e. MARCO’s reflex layer, should be running on-board the ISS. Unfortunately, it is not possible to install the MARCO local controller to control directly the motion of the SSRMS. Autonomous behaviors will therefore have to be implemented on-ground with all the limitations concerning the up- and down-link communications. In a similar manner as what was done for ETS-VII, a move-and-wait strategy will then be applied.

One example of such an autonomous behavior would be vision guidance of the SSRMS. All objects to be manipulated by SSRMS are equipped with grapple fixtures. An example of such a fixture, a Power Data Grapple Fixture (PDGF), is shown in Figure 4. PDGF’s are used to provide a mechanical interface to the SSRMS Latching End-Effector, to pass electrical power as well as video and data signals. Many of these fixtures will be located around the Station’s external structure, allowing the SSRMS and SPDM to be positioned in different locations. Of particular interest
for autonomous operations, each PDGF is equipped with a visual marker to guide the astronauts during telemanipulation of the SSRMS (see the target pin with the two crossed lines in the framed region of Figure 4).

Figure 4 Power and Data Grapple Fixture (PDGF)

Similar to the approach used for the ETS-VII mission [6], this visual marker can be used to automatically bring the end-effector (EE) into a position, from where the PDGF can easily be captured. Image analysis generates the required commands to align the EE with the PDGF. Expressed in the MARCO terminology, this visual servoing task can be considered as a reflex. Assuming a correct model of the PDGF markers and the cameras is available, MARCO can use the simulation environment, to prepare, test, and verify the entire visual servoing task fully on-ground without connecting the real space system.

Since it will be impossible to load a MARCO local controller on-board the ISS, the video images will be downlinked to the ground station, where a vision system will extract the PDGF marker location, and generate appropriate motion commands. These commands will then be sent to the on-board robot (see Fehler! Verweisquelle konnte nicht gefunden werden.). In the MARCO philosophy, the reflex layer is located at the lowest execution level (see Figure 1), i.e. “near” the robot controller, which really should integrate all the local control loops, such as autonomous visual servoing. Ideally, the reflex layer of MARCO should be integrated into the SSRMS on-board controller.

4.3 Path planning

A sizeable of the tasks performed by SSRMS will consist of loading and unloading of payloads from the shuttle’s cargo bay. These tasks will involve very large displacements, will very likely be time-consuming, and difficult to perform because the SSRMS has to be reconfigured many times during the transfer motion. MARCO could be used to plan the whole path on-ground, using the geometric model of the shuttle, the SSRMS and the ISS using all the 7 joints of the SSRMS, and to execute it autonomously on-board, under the supervision of the ground control station. The path-planning component, integrated in MARCO, uses a fast method [7] with linear complexity in the number of degrees of freedom (DOF). It proved to be very efficient as it omits a complete representation of the high-dimensional search space. Unlike most of the known path planning approaches e.g. [8], whose complexities reach from a quadratic to a exponential one, it can be applied also for robots with any number of DOFs.

4.4 Telepresence application

MARCO has the capability to provide more direct interaction with the remote manipulator and give the operator the impression to directly manipulate the objects in the environment with the robotic system as a “prolonged arm” into the space. For such jobs, the possibility to immediately interact with the remote environment via haptic input devices and direct vision feedback is required. This is not supported by the current communication infrastructure aboard ISS. Currently, the feasibility of establishing a communication infrastructure that would support these capabilities on the European part of ISS is being studied. This would provide direct communication links (data and video) with significantly reduced time delays (from several seconds to approximately a few hundreds of milliseconds).

Conclusion

A ground control system, based on DLR’s MARCO architecture, was proposed to program and teleoperate the Mobile Servicing System (MSS) aboard the ISS. In a first prototypic implementation the feasibility of controlling a space robot system on ISS fully from ground has been shown, using the MOTS at CSA as the target system, which emulates all the behavior and communication limitations as expected with the real system. The MARCO ground control station includes all the features to program, control, and supervise the MSS on the ISS. Various teleoperation modes are available from direct telemanipulation to task-oriented programming and execution. To ensure that MSS operations could be safely carried-out from the ground, the proposed ground control technologies have been demonstrated using the MOTS simulator.

References


