

MSS Ground Control Demo with MARCO

Klaus Landzettel*, Bernhard Brunner*, Günter Schreiber*, Bernhard-Michael Steinmetz*
Érick Dupuis**

*Institute of Robotics and Mechatronics, DLR Oberpfaffenhofen,
D-82234 Wessling, Germany

**Canadian Space Agency, 6767 route de l'aéroport, St-Hubert (Qc) J3Y 8Y9, Canada
Klaus.Landzettel@dlr.de

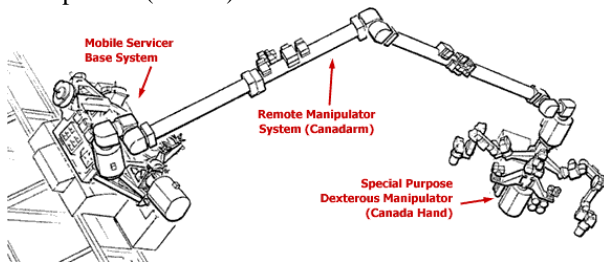
Keywords teleoperation, ground control, high-level robot programming, autonomous robot, user interface, ISS, MSS, MBS, SSRMS, SPDM, MOTS

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 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 part of the MSS, operations on the MSS 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. In this context, *situational awareness* refers to the operator's knowledge of the spatial relationships amongst the work site equipment, features and obstacles. 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 mandatory, which gives the operator the immersive feeling to have 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, CSA has added an interface to MOTS that will allow it to receive

commands and transmit 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 get the robots more and more intelligent, 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. As man machine interface, a sophisticated VR-environment with DataGlove and high-performance graphics is provided. 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

A non-specialist user – e.g. a payload expert – should be able to remotely control the robot system in case of internal servicing in a space station (i.e. in a well-known environment). However, for external servicing (e.g. the repair of a defect satellite) high interactivity between man and machine is mandatory. To fulfil these requirements, the design of the programming system is based on a 2in2-layer concept (See Figure 1), which represents the hierarchical control structure from the planning to the executive layer:

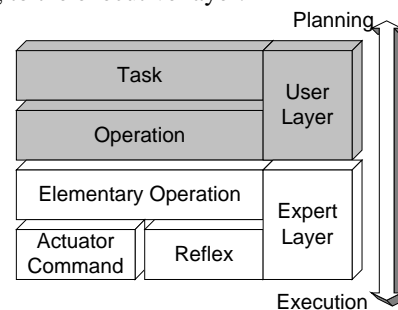


Figure 1 - MARCO Control Methodology

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 lowbrow user, which is not so 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 don't worry about the robot, what it is exactly doing, 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

To establish the command and telemetry data transmission links, the DLR's telerobotic system has been interfaced to MOTS via a client-server communication. To do the communication transparent to the origin MARCO system, an interface-computer was installed, which acts as a data transformation station between MARCO and MOTS. That means that the MARCO interface structures has been adapted to the existing MOTS interfaces and their timing. This approach has been proven very well during the ETS-VII space robot mission [4].

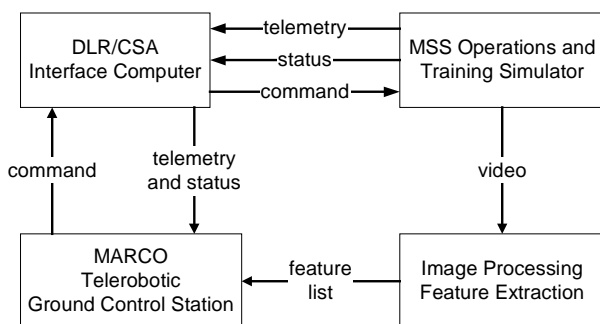


Figure 2 Interface structure of the MARCO/MOTS demonstrator

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.

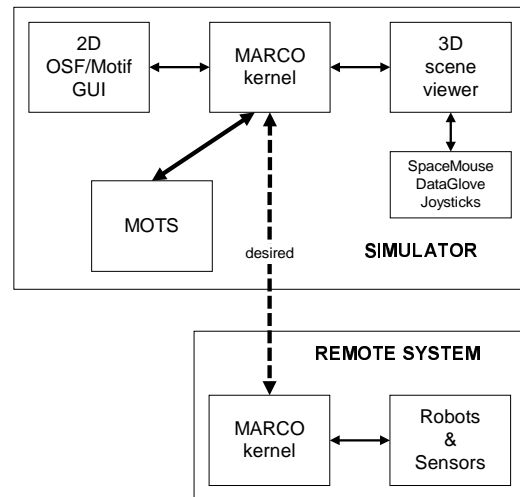


Figure 3 - Architecture: MARCO Demonstration

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.

4.2 Sensor-based local autonomy

As demonstrated a few years ago in the ROTEX experiment [2], the predictive graphics approach, which also includes the simulation of all the available sensors (force-torque, distance, vision, or s.th. else), is a very helpful tool to test and verify the local control loops, based on sensory data, even on-ground. In spite of the lack, that MARCO cannot install its own controllers on-board the ISS, it would be possible, to do autonomous task with the MBS, e.g. using the vision systems, mounted on the SSRMS. All the objects, which can be grasped by the end-effector (EE) of the SSRMS, are equipped with an Power and Data Grabble Fixture (PDGF). Also the SSRMS itself is attached to the ISS via such a PDGF: PDGFs are round, antenna-like devices, designed for mechanical actuation and

electrical, power, data transfer to, and from, a variety of devices and payloads through two pairs of umbilical connectors. Power and data connections will be supplied to, and from, either end of the SSRMS as well as to and from any payload equipped with a PDGF. The PDGF also services at the base for the SSRMS and SPDM. Many of them will be located around the Station's external structure, allowing the arm to literally walk hand-over-hand. A PDGF has a target marker to support the astronauts during telemanipulation of the SSRMS (see the target pin with the two crossed lines in the framed region of Figure 4).

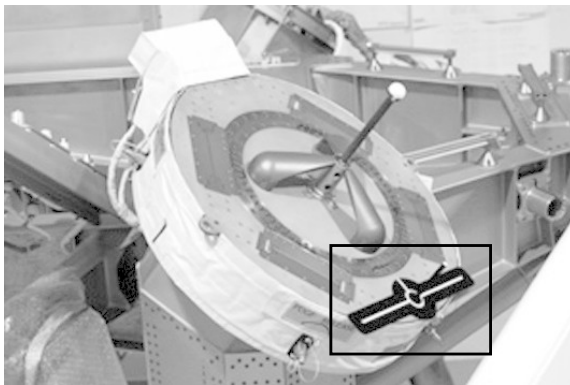


Figure 4 The Power and Data Grabble Fixture (PDGF)

Similar to the approach as verified in the ETS-VII mission [6], this “operational” marker can be used to automatically bring the EE into a position, from where the PDGF can easily be attached. 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*. Assume, a correct model of the PDGF markers and the cameras is available, MARCO provides the simulation environment, to prepare, test, and verify the entire visual servoing task fully on-ground without connecting the real space system.

Currently we have to implement the autonomous behaviors, i.e. MARCO's reflex layer, on-ground with all the limitations concerning the up- and down-link communications. Similar to the work on ETS-VII, we apply a move-and-wait strategy, getting the actual video images, extracting the PDGF markers, and generating appropriate motion commands, which will be sent to the on-board robot (see Figure 2). Actually, 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. For the future, it would be nice to have the possibility to integrate the reflex layer into the SSRMS on-board controller.

Another application for vision-based path refinement could be an active vibration damping, using the the same visual servoing approach as described above. Due to the inherent oscillations on the Tool Center Point (TCP) of the SSRMS – the length of the the SSRMS is about 17 meters – it would be very helpful, especially time-saving for the operator, if a desired motion stops more quickly as done without active damping.

4.3 Path planning

In the current schedule for the SSRMS activities performed via telemanipulation by the astronauts, a lot of tasks will be transfer motions of mounting parts from the shuttle's cargo bay to a destination point on the ISS. Anybody could imagine, that these tasks are very time-consuming as well as difficult to do, because the SSRMS has to be reconfigured many times during the transfer motion. That means, that only 6 joints of the SSRMS can be used during an arm motion, because the 1st or 2nd joint is always locked. Therefore we propose 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, supervised by the ground control station, sure. 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 ommits a complete representation of the high-dimensional search space. Opposite to 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 Redundant kinematics

Additionally to the path planning approach, which uses all the 7 DOFs of the SSRMS, a redundant mode could be very useful during cartesian movements. Imagine, the operator would like to move the TCP a little bit further, but one of the long links of the SSRMS would collide with another part of the ISS. Then the operator has to switch into an joint level mode, to reconfigure the SSRMS into a position, from where the motion wouldn't collide. But every joint level mode will change the current TCP, which could be undesirable, because the SSRMS should hold the current TCP position. Otherwise collisions between the carried object, attached to the TCP and another part could occur. Similar to the human arm, a redundant motion [9] holds the TCP and moves the arm links into a configuration which is suitable for proceeding the current motion task.

4.5 Telepresence application

For service tasks on ISS, e.g. maintenance of the resp. 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 such jobs, the possibility to immediately interact with the remote environment via haptic input devices and direct vision feedback should be given. The respective requirements for telepresence applications can easily be fulfilled by the MARCO control and programming features. But this requires another kind of communication infrastructure aboard ISS than the existing one. Currently, we study the feasibility of establishing exactly this communication structures on the European part of ISS, to provide direct communication links (data and video) with significantly reduced time delays (from several seconds to approximately a few hundreds of milliseconds).

5 Future Work

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

[1] J.-C. Piedbœuf, É. Dupuis, S. Kalayciogly, D. Bourque, L. Gregoris, C. Sallaberger, D. King, “Canada’s Technology Development in Space Robotics”, ASTRA Workshop, ESTEC, Noordwijk, The Netherlands, 5-7 December, 2000.

[2] G. Hirzinger, B. Brunner, J. Dietrich, J. Heindl, “Sensor-Based Space Robotics – ROTEX and its Telerobotic Features”, IEEE Transactions on Robotics and Automation, Vol.9, No.5, pp. 649-663, Oct. 1993.

[3] K. Landzettel, B. Brunner, G. Hirzinger, “The Telerobotic Concepts for ESS”, IARP Workshop on Space Robotics, Montreal, Canada, July 6-8, 1994.

[4] B. Brunner, K. Landzettel, G. Schreiber, B.M. Steinmetz, G. Hirzinger, “A Universal Task-Level Ground Control and Programming System for

Space Robot Applications - The MARCO Concept and its Application to the ETS-VII Project”, 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), ESTEC, Noordwijk, The Netherlands, 1-3 June, 1999.

[5] A. Kheddar, C. Tzafestas, P. Coiffet, “The Hidden Robot Concept - High Level Abstraction Telerobotics”, Proc. IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, 7.-11. Sep., Grenoble, France, pp. 1818-1824, 1997.

[6] K. Landzettel, B. Brunner, K. Deutrich, G. Hirzinger, G. Schreiber, B.M. Steinmetz, “DLR’s Experiments on the ETS-VII Space Robot Mission”, 9th Int. Conf. on Advanced Robotics (ICAR '99), Oct. 25-27, Tokyo, Japan, 1999.

[7] B. Baginski, “Motion Planning for Manipulators with Many Degrees of Freedom - The BB-Method”, PhD Thesis, Technical University of Munich, Germany, 1999.

[8] E. Ralli, G. Hirzinger, “A global and resolution complete path planner for up to 6DOF robot manipulators”, Proc. IEEE Conference on Robotics and Automation, pp. 3295-3302, Minneapolis, Minnesota, April 1996.

[9] G. Schreiber, M. Otter, G. Hirzinger, “Solving the Singularity Problem of non-redundant Manipulators by Constraint Optimization”, Proc. IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, Kyongju, Korea, pp. 1482-1488, 1999.