DLR's Experiments on the ETS VII Space Robot Mission

K. Landzettel, B. Brunner, K. Deutrich, G. Hirzinger, G. Schreiber, B.M. Steinmetz

DLR Oberpfaffenhofen
German Aerospace Center
Institute of Robotics and System Dynamics
D-82234 Wessling
Klaus.Landzettel@dlr.de

Abstract

This paper describes the *GETEX* (German ETS-VII EXperiments) joint robotics experiments, which has been conducted in April '99 at the first free-floating space robot on NASDA's ETS-VII satellite[1]. All the experiments have been performed and controlled from DLR's ground control system for space robotics applications which was connected to NASDA's ground station in Tsukuba. The telerobotic system combines sensor-based task-level *teleprogramming* (as the basis for autonomy) with the features of *teleoperation* and shared autonomy. The hierarchical system structure is shown as well as the flexibility in programming and controlling each kind of space robotics application. This approach has led to a modular task-directed programming scheme, called *M*odular *A*&*R Co*ntroller (MARCO), which provides a very flexible architecture to adapt the application-specific requirements to a given controlling scheme. A peg-in-hole experiment, using VR methods and the „*vision&force*“ control scheme, by closing sensor control loops directly on-board (force) and via the ground control system (vision) is explained. During GETEX we conducted experiments with relevance to the behavior of ETS-VII in *free motion mode* in order to verify the existing dynamic models.

1 Introduction and Overview

After the success of ROTEX, the first remotely controlled robot in space, DLR has focused its work in telerobotics on the design of a high-level task-oriented robot programming system, which is characterized as *learning by showing in a virtual environment*. 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 already tested in ROTEX[2], but in addition provides the feasibility to program a robot system at a task-directed level, including a high degree of on-board autonomy.

This means that a non-specialist user like a payload expert will be able to control a remote robot system e.g. for internal servicing within a space station, i.e. in a well-known environment. This requires a sophisticated man-machine-interface, which hides the robot control details and provides an *intuitive programming interface*. The user interacts via the virtual view with the real environment, as (s)he has only to define, what (s)he wants to do, not how it has to be done. Supported operations are e.g. open/close a door/drawer, pick&place an orbital replaceable unit etc.
However, for external servicing with free-flying robots, e.g. the repair of a defect satellite, high interactivity between man and machine is required, because the remote environment will be mainly unknown. All the well-known problems w.r.t. teleoperation under long time delays can only be solved by the *predictive graphics* approach. One of the main requirements is the feasibility to update the simulated world according to the real world as well as to provide *local autonomy based on intelligent sensor data processing* without large a priori knowledge.

To fulfill the requirements of both application fields, we have developed a 2in2-layer model[3], which represents the programming and control structure from the executive to the planning level in a hierarchical way. According to the application requirements the user can use the necessary and sufficient level of commanding and programming or switch between the *different layers* especially in case of failure detection and recovery.
The ground control facilities of our MARCO (Modular A&R COntroller) system were used in April '99 to remotely control the Japanese ETS-7 robot, the first robot in free space. The main goals of DLR's contribution within the GETEX project were the utilization of the world model update concept using the real video images, to verify our task-level programming approach including on-board autonomy via selected image features and force-torque information as well as the verification of the dynamic simulation of the interactions between the robot and the carrier.
Our cooperation with NASDA w.r.t. to the dynamics verification was one important step towards *a free-flying service satellite*. For more details of our work on repair satellites see [4].

2 The MARCO system

The goal for the development of our high-level programming system was to design a *unified concept* for a flexible, highly interactive, on-line programmable teleoperation station as well as an off-line programming tool, which includes all the sensor-based control features as tested already in ROTEX, but in addition provides the possibility to program a robot system on an implicit, task-directed level.
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 an machine is demanded. For that reason the design of our programming system is based on a 2in2-layer-concept, which represents the *hierarchical control* structure from the planning to the executive layer:

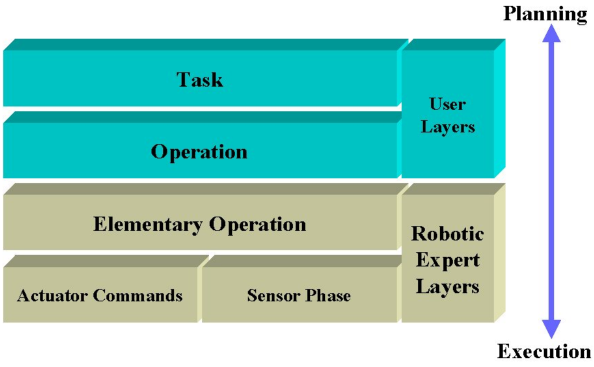


Figure 1 2in2-layer-model

On the user layer the instruction set is reduced to what has to be done. 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 robotic expert layers.

2.1 Reflex (Sensor Control Phase)

At the lowest level of the MARCO system the sensor control mechanism is active. These so-called reflexes guarantee the *local autonomy* at the remote robot‘s site via using sensory data processing algorithms in an extensive way. The 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 we store the nominal sensory patterns and generate appropriate reactions (of robot movements) on deviations in the sensor space. In GETEX we used vision and force for implementing the reflex layer (see section 3.1).

2.2 Elemental Operations

The robot expert layer is completed by the Elemental Operation (*ElemOp*) level. It integrates the sensor control facilities with position and endeffector control. According to the constraint frame concept, the non-sensor-controlled degrees of freedom (dof) of the cartesian space will be position controlled

- in case of *teleoperation* directly with a telecommand device like the SpaceMouse.
- in case of *off-line programming* by deriving the position commands from the selected task. Each object, which can be handled, includes a relative approach position, determined off-line by moving the virtual end-effector in the simulation into the desired pose w.r.t. the respective object and storing the geometrical relationship between the object’s reference frame and the tool center point (TCP).

A model-based on-line collision detection supervises all the robot activities. For global transfer motions a computational very fast *path planning* algorithm avoids collisions and singularities in the robot’s joint space.

2.3 Operations

Whereas the Reflex and ElemOp levels require the robotics expert, the task-directed level provides a powerful man-machine-interface for the non-specialist user. We divide the task-directed layer into the Operation and the Task level.
An Operation is characterized by a sequence of ElemOps, which hides the robot-dependent actions. Only for the specification of an Operation the robot expert is necessary, because (s)he is able to build the ElemOp sequence. For the user of an Operation the manipulator is fully *transparent*, i.e. not visible.

We categorize the Operation level into two classes:

- An Object-Operation is a sequence of ElemOps, which is related to a class of objects available within the workcell, e.g. GET <object>, OPEN <door>.
- A Place-Operation is related to an object, which has the function of a fixture for a handled object, e.g. INSERT <object> INTO <place>. Before a Place-Operation can be activated, the corresponding Object-Operation has to be executed. <object> is the object, known from the predecessor Object-Operation, <place> the current fixture, to which the object is related.

2.4 Tasks

Whereas the Operation level represents the subtask layer, the possibility to specify complete robot tasks must be available in a task-directed programming system. A Task is described by a consistent sequence of Operations, which are instantiated with concrete object instances (see Figure 2). To generate a Task, we use the VR-environment. All the Operations, activated by selecting the desired objects or places, are recorded with the respective object or place description. An expressive example for applying this *pick&place* scheme on a peg-in-hole task on the TaskBoard of the ETS-VII satellite will be given in section 3.1.

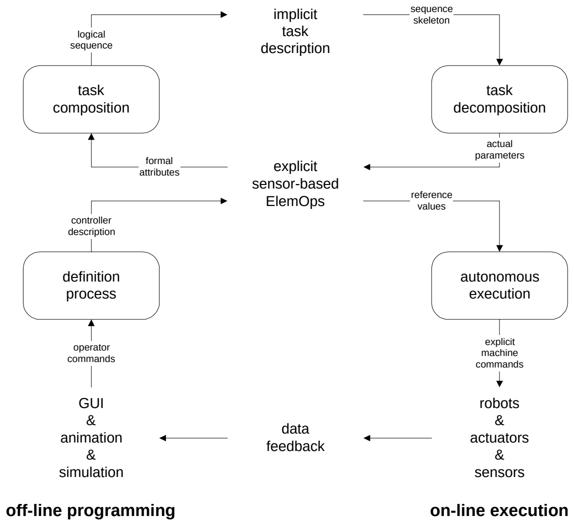


Figure 2 task-directed sensor-based programming

2.5 Different graphical user interfaces

Our task-directed programming system with its VR-environment provides a man-machine-interface at a very high level, i.e. without any detailed system knowledge.
To edit all four levels as well as to apply the Reflex and ElemOp level for teleoperation, a sophisticated graphical user interface based on the OSF/Motif standard has been developed (see Figure 3, screen down on the left). This GUI makes it possible to switch between the different execution levels in an easy way.

Figure 3 GUI of the universal programming and control station (MARCO)

Based on the ROTEX experience we have implemented a prototypic teleoperation station, to remotely control space robotics applications by predictive graphics. Figure 3 shows different views of the simulated environment (far, near, camera view), the Motif-GUI, and the real video feedback image, superimposed with a wire-frame model of the predicted state (up on the right). All the screens can be viewed in stereo mode for full immersion into the workcell environment.

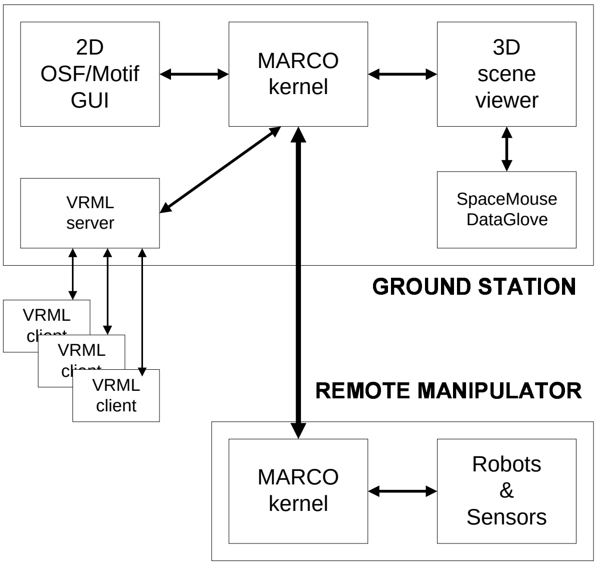


Figure 4 User Interface Structure

3 The GETEX experiments on ETS-VII

From April 19-21, 1999 the DLR's MARCO telerobotic and programming system was used to control the robot arm on the Japanese ETS-VII satellite. The main goals of the GETEX experiments on (ETS VII) were

- to verify a MARCO-based telerobotic *ground control* station for remote control of a free-floating robot, in particular
- to perform a peg-in-hole experiment, using VR methods and the „*vision&force*" control scheme, by closing sensor control loops directly on-board (force) and via the ground track (vision), thus proving MARCO's sensor-based autonomy features,
- to conduct experiments with relevance to the behavior of ETS-VII in *free motion mode* and thus to verify the existing 6 dof dynamic models for the interaction between a robot and its free-flying carrier satellite.

All experiments could be performed very successfully. To implement the User Interface Structure as depicted in Figure 4, we had to add some modules for communication with NASDA's ground control system, but not to change the overall MARCO ground control structure.

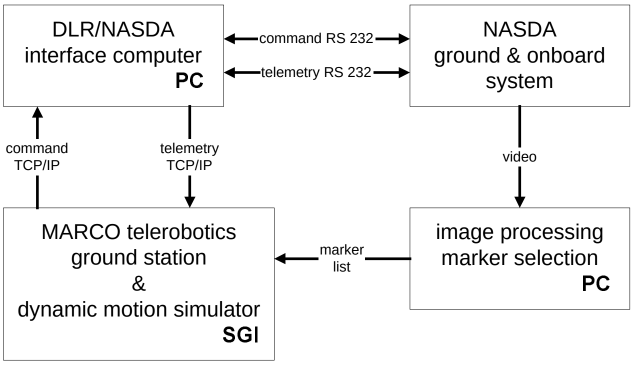


Figure 5 GETEX ground control configuration

To check and test our interfaces as well as our MARCO control station within the ETS-VII scenario, an on-line simulator has been developed, which emulates the remotely operated robot, its command interfaces and its environment. The simulator is able to emulate all different operational modes, timing, the environmental interactions, and the prediction of satellite attitude while moving the robot arm. This kind of simulation has turned out to be very useful for software development and test without having access to the NASDA system.

The original MARCO kernel couldn't be implemented on-board the ETS-VII, because only the ElemOp-Layer was available on-board. All the other layers were implemented as add-ons on-ground, but this was no limitation to the verification of our task-level programming methods, because the downlink feedback data were rich enough to parametrize the next ElemOp according to the current execution state.

All the experiments were available live at Internet via Video transmission and a VRML simulation which showed the current robot and satellite status in an expressive way[5]. In principal it would have been possible to control the task execution via our Java/VRML interface (see Figure 4) from any point all over the world without restrictions e.g. concerning time delay.
It should be mentioned, that the know-how, gained during the phase of adapting the MARCO system to the ETS-VII constraints, will be very useful for further space robot missions.

3.1 The peg-in-hole experiment, using VR methods and the „vision&force“ control scheme

The MARCO system worked very well, that we decided, together with the Japanese partners, to execute the whole peg-in-hole experiment with the TBTL (TaskBoard TooL) in the automatic mode: after teach-in of the desired task sequence (pick TBTL, see Figure 6, and place it into HOLE A, see Figure 7) in the VR environment, the execution was started and performed fully automatically. No voice confirmation between each ElemOp was further needed, as it had to be done during the test runs.

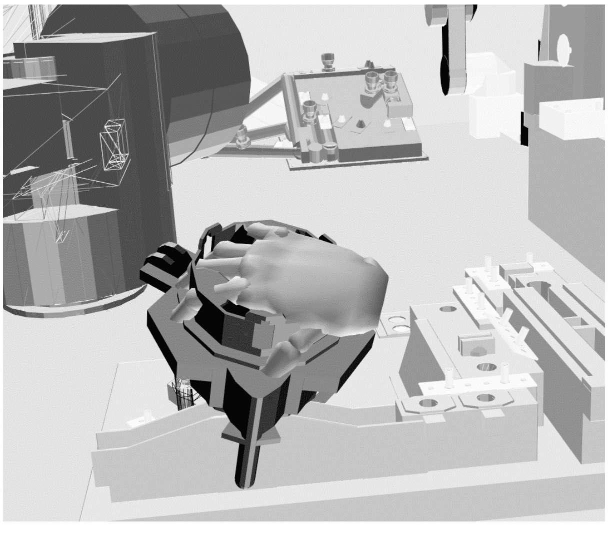


Figure 6 Pick TBTL by VR-hand

To get the TBTL, we first carried out a visual servoing task (at the reflex layer). For that job we used the markers mounted all over the TaskBoard. Originally, these markers should help the operator to teleoperate the TCP into the accurate grasping position over the TBTL or another part mounted on the TaskBoard: for each camera lens of the stereo pair, three corresponding markers have to be aligned as well as the middle have to be centered within the outer ring (see Figure 8).
Now, we used these operator-markers for controlling the robot autonomously over the TBTL grasping position.
To extract the markers from the video image we used a blob-finding algorithm, described in the following: due to the extreme contrast inside a marker (bright spots on a black background, see Figure 8) as well as strong fluctuations of illumination a crude grayscale image of the task board is reduced to a standardized ternary image (black/gray/white).

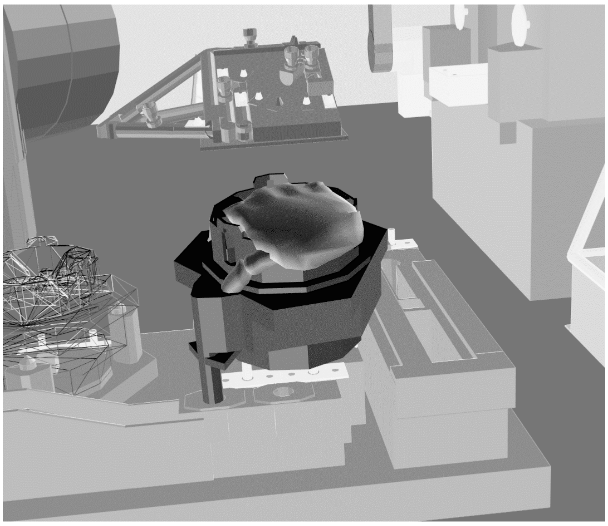


Figure 7 Place TBTL into Hole A
The real robot and object status (here the TBTL), fed back in the telemetry channel, is shown wire-framed.

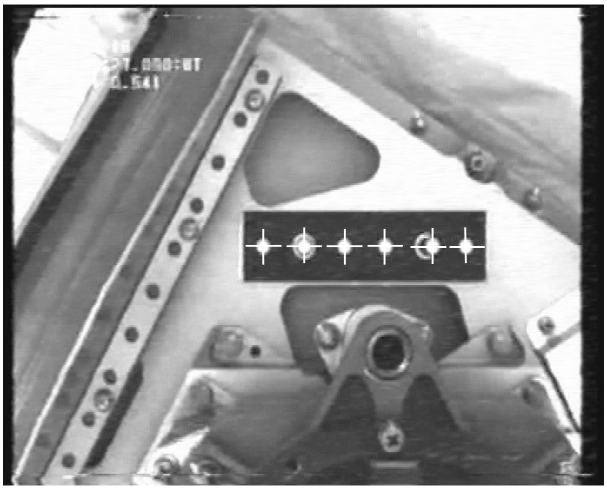


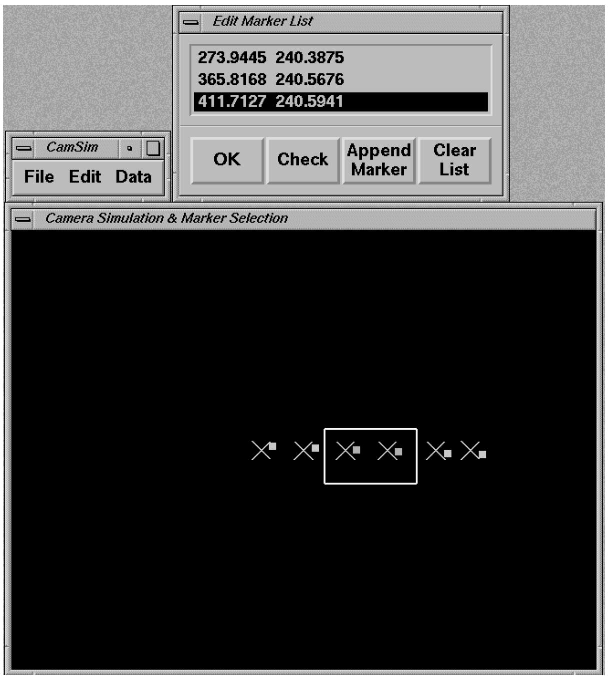
Figure 8 View out of the hand camera, showing the tracking markers for visual servoing

This conversion is based on an illumination estimation derived from intensity histograms of a limited image history and the last set of verified markers, if there are any. Furthermore the identification of new marker locations is performed in tree stages:

1. Noise and drop outs of adjustable size will be eliminated by adaptive standard filters (e.g. convolution methods).
2. Each spot of appropriate form, size, and structure, which is fully included, will be extracted out of ternary image.
3. Both measured features of each spot and a scalable marker model are used for a statistical conformity analysis of measured and predicted background intensity distribution.

The necessary reference features as well as the marker models for the outlined selection are estimated starting from the last valid marker set (if any), the elapsed time, and the intensity distribution of the current grayscale image. The detected markers are tagged in the grayscale image. Their center of gravity coordinates are available on request.

Because this algorithm delivered more „markers“ as desired, e.g. due to bad lighting conditions, we selected the markers interactively and checked the resulting control command before sending it to the real robot. Figure 9 shows the simulated (■) and the real (X) markers, with the interactive selection frame. The differences between the ■ and X markers result from a different TCP pose, showing the two representations. If real and simulated TCP are the in the same pose, the real and simulated markers have nearly the same 2D-coordinates. All the control algorithms we applied at the real environment were parametrized and tested on our MARCO ground station, which provides the necessary sensor simulation as well as sophisticated graphical tools for VR visualisation.



273.9445 240.3875
365.8168 240.5676
411.7127 240.5941

Figure 9 marker features, got from image processing (X) and vision simulation (■)

As mentioned above we used the marker features in the video image to control the TCP of the robot autonomously into the desired sensor-related pose. For that reason we have developed an approach, which doesn't need any calibration. The control law may be written as

$$v_c = \alpha\, C\, (s - s^*)$$

where (s-s*) is the vector-valued deviation between the current and the nominal sensory pattern indicating the displacement of the current robot pose x from the nominal pose x*. v_c is the velocity command, α represents a scalar dynamic expression, at least a real constant, determining the closed control loop behavior, and C represents a projection operator used for mapping the sensor space onto the (cartesian) control space. C is determined by neural network learning or using analytical methods.

Here we have applied the analytical method for determination of *C*, which is represented by the pseudoinverse of the Jacobian matrix of the *m* deviations in the sensor space w.r.t. the *n* deviations in the control space. For that we moved the robot's TCP a little bit around in all $n=6$ degree of freedoms, recorded the corresponding sensor values and generated the Jacobian from the resulting difference quotients.

$$J_{i,j} = \left.\frac{\partial y_i}{\partial x_j}\right|_{x^*} \quad i = 1..m,\ j = 1..n$$

We performed the experimental determination of *C* in our simulation environment as well as in the real one. The result was nearly the same, due to the accuracy of our camera calibration, that we applied in the simulation. The intrinsic camera parameters have been estimated using an in-house developed camera calibration tool, the external could be easily extracted from the available CAD data.

To explain the accuracy, the respective simulated and measured pixel coordinates in the case that the (simulated resp. real) camera is looking straight onto the TBTL are shown in Table 1. They differ only in the subpixel domain.

	X1	Y1	X3	Y3	X4	Y4
real	273.80	240.17	365.70	240.35	411.60	240.37
sim	273.94	240.38	365.81	240.56	411.71	240.59

Table 1 measured and simulated pixel coordinates

The goal for the MARCO system was to find the markers in the life video image and to generate the appropriate straight path command to move the robot into the desired (sensor-defined) target pose. To verify the vision-based sensor control loop, we moved the TCP intentionally into a position different from the target pose (a few centimeters in all translational directions and about 20 degrees in z-rotation).

After 3 cycles (with $\alpha = 1$), the target pose was reached. To fulfil the *vision&force* control sceme, the force control loop, implemented on-board, was activated and supervised.

3.2 The Dynamic Motion Experiments

A major part of the GETEX experiment time was allocated to the so-called Dynamic Motion experiments, which consisted of a series of maneuvers carried out by the manipulator while the attitude control system of ETS-VII was switched off.

If a robot which is mounted on a spacecraft moves, it generates linear and angular momentum. In the case of an attitude and position controlled spacecraft, the attitude control system will permanently produce forces and torques compensating for the arm motion. The spacecraft may then be considered as inertial in the co-ordinates of an orbit-fixed system, and the problem of robot motion planning can be solved using the same methods as for terrestrial manipulators. While for the control of the spacecraft attitude electrically powered momentum wheels can be used as well as thrusters, for the control of the spacecraft translation fuel consuming thrusters are the only actuators currently in use. For this reason and because the position errors are generally negligible, most satellites are only attitude controlled. Due to the linear momentum conservation, which states that the centre of mass of the system comprising the robot and the satellite is constant, the motion of a manipulator mounted on the satellite will lead to a compensating motion of the satellite. The amount of satellite translation produced, depends on the masses of the bodies constituting the system.

For space robotic systems which are neither position nor attitude controlled the angular momentum conservation law leads further to a rotation of the spacecraft, by an amount which results from the mass and inertia properties of the manipulator links and the spacecraft. It is generally assumed that no external forces act on such *free-floating robots*[6,7]. The free-floating mode of operation is of interest for space robots not only for the reason that attitude control fuel may be saved what augments the robot life-span, it will also be of importance during repair missions, when the servicing satellite is very close or in contact to the target satellite: any action of the attitude control system of either of the two satellites during this phase would lead to a collision and thus to potential damage on the two spacecraft.

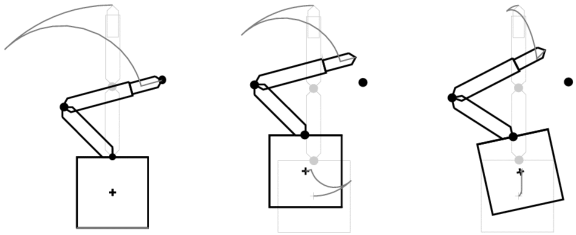
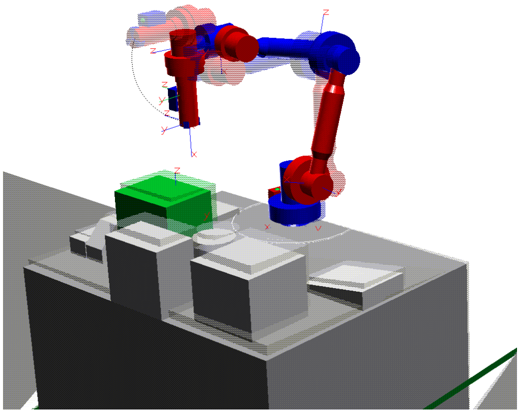


Figure 10 The influence of the satellite attitude control mode on the path described by the robot end-effector - the same joint motion is carried out by a robot with a fixed base (left), an attitude controlled robot (middle) and a free-floating robot (right).

As long as the tasks performed with the robot are described in robot-fixed coordinates, the fact that the satellite position remains uncontrolled has no influence. If, however, the task is described with respect to an orbit-fixed co-ordinate system, as it would be the case for example for the capturing of a defect satellite, the satellite motion has to be taken into account (see Figure 10). The equations relating the tool centre point motion to the manipulator joint motion, which for robots with an inertially fixed base are purely kinematical equations, become thus dependent on dynamic parameters in the case of free-floating space robots, due to the fact that the momentum equations are used to describe the satellite motion.

This influences the path planning methods which have to be applied. On one hand, singularities, that is joint configurations in which the robot is not controllable in Cartesian coordinates, are no more a function of the robot kinematics only, but become dependent on the dynamic properties of the robot, too. Therefore, iterative methods based on the direct kinematic equations have to be used instead of the inverse kinematics equations. Moreover, the angular momentum equation makes the system nonholonomic[8], which means that the satellite orientation is not a function of the current joint configuration only, but merely a function of the chosen path. Two different pathes starting at the same initial configuration of the robot, and leading to the same final configuration, will therefore result in different amounts of satellite rotation – and thus in different final inertial tool center point positions, too. As a consequence, nonholonomy offers the possibility to do a re-orientation of the satellite using manipulator motion only, by simply carrying out a closed-loop maneuver in joint space.



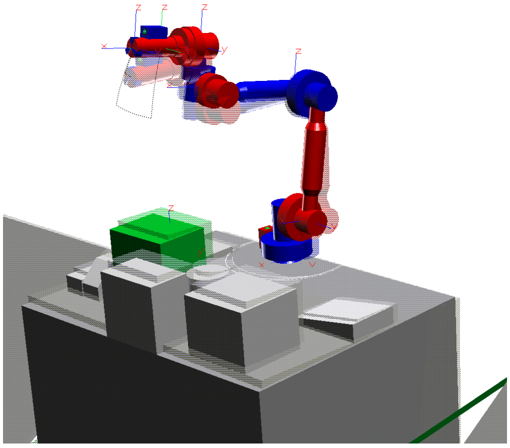


Figure 11 Examples of *Dynamic Motion* manoeuvres carried out during the GETEX mission: simple point-to-point manoeuvre (above) and re-orientation manoeuvre (down). The shaded robot indicates the reference position. The satellite reaction to the arm motion is scaled by a factor of 10 in this picture.

This kind of maneuver can be employed to significantly augment the workspace of the robot, since it allows to turn the satellite into any desired orientation, bringing back the manipulator into its reference configuration. The maximum workspace of a free-floating space robot is thus described by a hollow sphere of which the inner and outer radius are given by the minimum and maximum possible distance between the tool center point and the system center of mass. Another possibility resulting from nonholonomy is that any point which is inside the fixed-base workspace of the robot may be attained with zero satellite attitude error. In the simplest case, this may be done by planning and executing the maneuver as for a robot with a fixed base and adding a closed-loop re-orientation maneuver to compensate for the produced attitude error. Path planning for nonholonomic system has been investigated in the context of cars and wheel-driven robots[9]. While those systems may generally be considered as planar, the case of free-floating robots demands spatial methods.

Whatever path planning method is applied to free-floating robots is necessarily highly model-based. The parameters of the dynamic model have therefore to be known quite well. While this poses no problem for the geometric parameters and for the mass and inertia of the manipulator, the mass and the inertia of the spacecraft are subject to important changes during the lifetime of a servicing satellite. This is especially the case if the spacecraft is performing capturing or rendezvous/docking like operations.

One goal of the German ETS-VII Technology Experiments (GETEX) on the Japanese robotics satellite ETS-VII, has therefore been to identify the mass properties of the satellite after one year and a half of activity in orbit. Further objectives were the verification of the dynamic models and to obtain some insight into the nature and importance of the disturbances acting on a robotic satellite on low Earth orbit. Additionally, the mission aimed at gathering data for the future design of controllers which combine the manipulator motion control with the satellite attitude control. To meet all these objectives, a variety of different maneuvers were executed by the manipulator, while the attitude control system of ETS-VII was switched off.

These so-called *Dynamic Motion Experiments* include simple point-to-point operations and closed-loop re-orientation maneuvers (examples of which are given in Figure 11), sequences during which only one joint was active at a time as well as sequences during which all joints were moving simultaneously. The major constraints, due to mission security aspects, were the maximum satellite attitude error allowed by NASDA which was limited to $\pm 1.0^{\circ}$ around each axis and the fact that the maximum tool center point velocity was limited, too. Furthermore, the reaction wheels were turning at a very low but non-zero constant velocity during the experiments, which introduced undesired torques into the system. Their effects will have to be considered during the evaluation of the mission results.

In total, over 110 minutes of dynamic motion experiments have been carried out, of which 52 minutes have been spent in free motion mode. The remaining time was used to repeat the experiments in reaction wheel attitude control mode for verification purposes.

First evaluations of the measurement data confirm the need to account for external disturbance forces acting on the satellite, such as the gravity gradient torque and magnetic torque. The identification of models of these disturbances is actually done at our institute, it will be followed by the development of compensating path control strategies.

4 Conclusion

We have shown the universal capabilities of DLR‘s MARCO system for controlling any kind of robotics applications, especially for space. Recently (April ’99) we have performed the GETEX mission at ETS-VII with very successful results, i.e. in applying the vision&force control scheme to the peg-in-hole environment of the ETS-VII TaskBoard as well as in verifying our dynamic models of free-floating space robots. Now we believe that the extensive use of robotics, e.g. at the International Space Station or as a free-flying servicing satellite[10] including intelligent robot arms and grippers, must be pushed by all industrial and political partners.

5 Acknowledgements

We’d like to gratefully thank all the Japanese collegues at NASDA for their very helpful support during the preparation, tests and operational phases.

Many thanks to Roberto Lampariello for his external support during the mission concerning the modification of the dynamic motion pathes.

References

---

[1] M. Oda, K. Kibe, F. Yamagata, „ETS-VII – Space Robot In-Orbit Experiment Satellite“, IEEE Conf. on Robotics and Automation, Minneapolis, April 1996

[2] G. Hirzinger, B. Brunner, J. Dietrich, J. Heindl, „ROTEX – the First Remotely Controlled Robot in Space,“, IEEE Conf. on Robotics and Automation, San Diego, May 1994

[3] B. Brunner, K. Landzettel, B.M. Steinmetz, G. Hirzinger, „TeleSensorProgramming – a task-directed programming approach for sensor-based space robots“, Int. Conf. On Advanced Robotics (ICAR’95), Sant Feliu de Guixols, Spain, Sept. 1995

[4] K. Landzettel, B. Brunner, G. Hirzinger et. al., „DLR’s robotics lab – recent developments in space robotics“, i-SAIRAS 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1-3 June 1999, ESTEC, Noordwijk, The Netherlands

[5] http://getex.robotic.dlr.de

[6] Longman, R.W., Lindberg, R.E. and Zedd, M.F., “Satellite-mounted Robot Manipulators – New kinematic and Reaction Moment Compensation”, International Journal of Robotics Research, 3, 1987.

[7] Dubowsky, S. and Papadopoulos, E., “The Kinematics, Dynamics and Control of Free-Flying and Free-Floating Space Robotic Systems”, IEEE Transactions on Robotics and Automation, 5, 1993.

[8] Nakamura, Y. and Mukherjee, R., “Nonholonomic Path Planning of Space Robots via a Bidirectional Approach”, IEEE Transactions on Robotics and Automation, 4, 1991.

[9] Li, Z. and Canny, J.F., “Nonholonomic Motion Planning”, Kluwer Academic Publishers, 1993.

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