

CONTROL OF INTELLIGENT ROBOTS IN SPACE*

E. Freund, Ch. Bühler

Institute of Robotics Research
University of Dortmund
4600 Dortmund 50, West Germany

Abstract

In view of space activities like International Space Station, Man-Tended-Free-Flyer (MTFF) and free flying platforms, the development of intelligent robotic systems is gaining increasing importance. The range of applications that have to be performed by robotic systems in space includes e. g. the execution of experiments in space laboratories, the service and maintenance of satellites and flying platforms, the support of automatic production processes or the assembly of large network structures. Some of these tasks will require the development of bi-armed or of multiple robotic systems including functional redundancy.

For the development of robotic systems which are able to perform this variety of tasks a hierarchically structured modular concept of automation is required. This concept is characterized by high flexibility as well as by automatic specialization to the particular sequence of tasks that have to be performed. On the other hand it has to be designed such that the human operator can influence or guide the system on different levels of control supervision, and decision. This leads to requirements for the hardware and software concept which permit a range of application of the robotic systems from telemanipulation to autonomous operation. The realization of this goal requires strong efforts in the development of new methods, software and hardware concepts, and the integration into an automation concept.

1. Introduction

With respect to increasing space activities, e. g. ISS, free-flying platforms or planetary operations, it is necessary to reduce the operational costs for space systems. One major key for future operational systems will therefore be the application of robotic systems in space /1/. It is planned to use different kinds of manipulators and robots in space to support and execute several tasks inside space-modules or in free space, especially for

- Docking/Berthing,
- Repair and module exchange,
- Service and maintenance of free-flying platforms,
- ORU (Orbit Replaceable Unit) - Exchange,
- Assembly of large structures,
- Experiment execution and production tasks.

* This work was supported by a grant of the "Bundesminister für Forschung und Technologie (BMFT)" of Federal Republic Germany.

The execution of this variety of tasks for manipulators and robots in space requires a hierarchically structured modular concept of automation including as well telemanipulation as autonomous operation. This design should cover the range of possible applications and has to provide interfaces for human interaction on different levels of control, supervision, and decision. To reach this goal, intensified efforts in the development of future-oriented robotic systems are necessary, where the integration into an overall concept of automation should be included from the very beginning.

In this paper structures for the control of multi-robot systems for space applications are considered. In chapter II the general structure for an autonomous multi-robot system in space is introduced. One of the key issues of the concept is automatic task management which is discussed in section III. The structuring of the levels of coordinated operation and collision avoidance and a mathematical formulation of the substructures is presented in chapter IV. Finally a test facility for the proposed intelligent control structure and for specific control features, which was built up at IRF Laboratory in connection with the national project CIROS (Control of Intelligent Robots in Space), is described in section V.

II. Overall System

In present space technology astronauts have to leave the space vehicle for almost every execution of activities outside this vehicle. Robots and manipulators could be used in space to reduce the risk and cost of such "Extra Vehicular Activities", where robots and telemanipulators will evolve step by step from simple telemanipulation robots to autonomous robotic systems /2/.

Teleoperation is the first step in this development. Here a human operator controls the manipulator by means of a model or from a control panel. The structure of a manipulator system for teleoperation is shown in fig. 1. The first level of the system representation is the mechanical construction of the manipulator, which includes internal sensors for the measurement of positions, velocities, forces, and torques.

The dynamic model of the robot can be described by highly nonlinear differential equations

$$\underline{x}(t) = \underline{A}(\underline{x}) + \underline{B}(\underline{x}) u(t), \quad \underline{y}(t) = \underline{C}(\underline{x}) \quad (1)$$

with nonlinear couplings between the variables of motion. In eq. (1), $\underline{x}(t)$ is the n dimensional state vector, $\underline{u}(t)$, $\underline{y}(t)$ are the m dimensional input- and output-vectors, respectively, and $\underline{A}(\underline{x})$, $\underline{B}(\underline{x})$, $\underline{C}(\underline{x})$ are matrices of compatible order, which describe the dynamics of the system. By use of the nonlinear control concept

$$\underline{u}(t) = \underline{F}(\underline{x}) + \underline{G}(\underline{x}) \underline{w}(t) \quad (2)$$

with

$$\underline{F}(\underline{x}) = - \underline{D}^{*-1}(\underline{x}) \left[\underline{C}^*(\underline{x}) + \underline{M}^*(\underline{x}) \right], \quad \underline{G}(\underline{x}) = - \underline{D}^{*-1}(\underline{x}) \underline{\lambda},$$

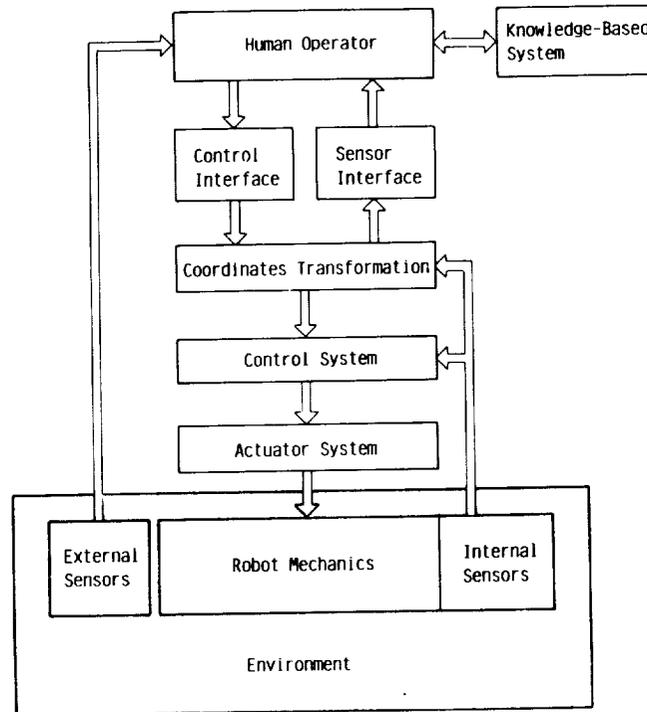


Figure 1. Control structure of a manipulator system for teleoperation

where the matrices \underline{D}^* , \underline{C}^* and $\underline{\lambda}$ are given from the nonlinear decoupling and control law /3/, one obtains a linear, decoupled behaviour for each axis:

$$\ddot{y}_i(t) + a_{i11} \dot{y}_i(t) + a_{i10} y_i(t) = \lambda_i \cdot w(t), \quad (i = 1, \dots, m) \quad (3)$$

The level of controllers computes signals for actuation which are transformed by the actuator system into forces and torques acting on the mechanical construction /3/.

At this stage man is still in the loop, responsible for motion planning, coordination, collision avoidance and supervision. The operator may get help information from a knowledge-based system, where system data can be stored and later on recalled for similar tasks. During teleoperation from ground time delays of up to 5 sec decrease the performance. Therefore it is necessary to provide the manipulator or robotic system even at this stage of development with a certain level of autonomy.

Further steps towards autonomous robotic systems are telesupervision and teleautomation, where simple tasks are accomplished automatically, while the operator controls the system as telemanipulator for specific tasks. So the operator is not required to be permanently in the loop, which eases the job of the crew considerably. Due to the variety of the tasks, possible long duration missions and from aspects of reliability of the system the highest degree of automation is desirable. The last stage in this development is a fully autonomous operating multi-robot system, whose structure is given in fig. 2. The system consists of several robots with one or more arms each. Based on the structure of a manipulator system in fig. 1, the evolution to autonomous multi-robot systems is characterized by the

integration of additional hierarchical levels. Strategies for automatic task management, coordinated operation and collision avoidance became integral parts of the structure.

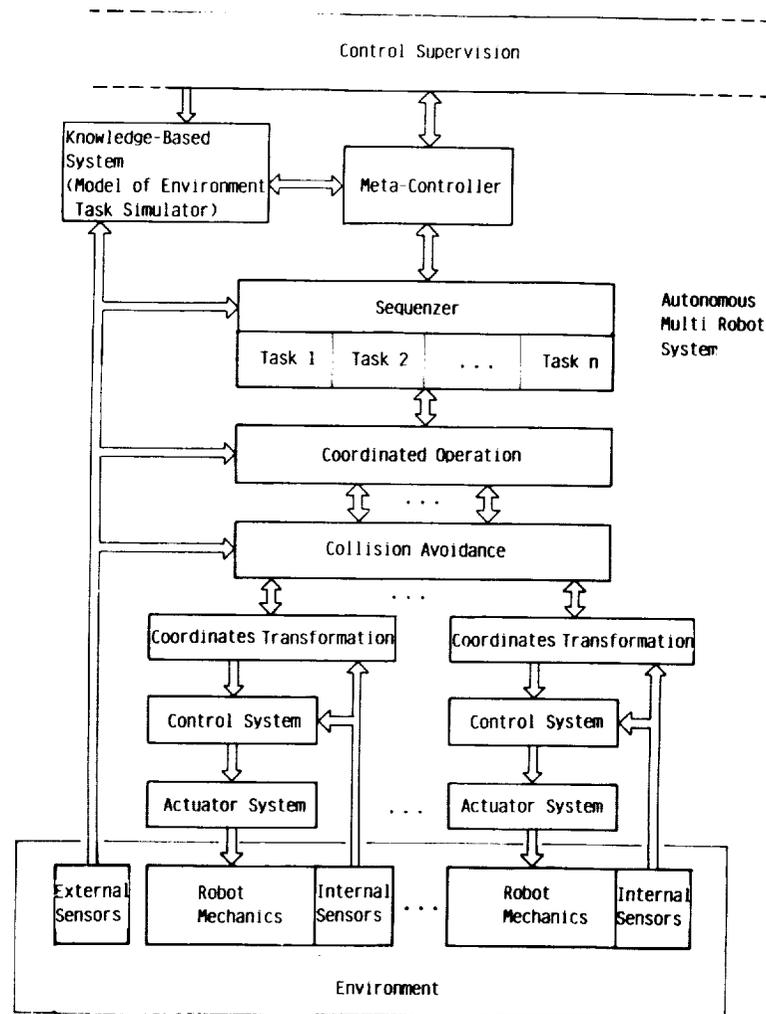


Figure 2. Structure of an autonomous multi-robot system

The levels for collision avoidance and coordinated operation are hierarchically placed above the level of coordinate transformation, which includes the various kinematics of the robots involved. The level of coordinated operation is responsible for the generation of reference values, which enable a coordinated task execution. To avoid collisions of the robots with themselves as well as with obstacles, in the layer of collision avoidance appropriate strategies are implemented. The formulation of these strategies is based on a systematic design procedure for multi-robot systems /4/, which has to be applied on a group and a system level.

The superimposed task management executive is responsible for automatic task activation choice and reservation of appropriate robots, execution control and performance control. Also included are safety and emergency reactions, which are initiated in case of failure and contingency. The operation is assisted by a knowledge-based system, which runs a model of environment as well as a task simulator, whereas the model of environment will be continuously updated by evaluation of relevant sensor data and status reports. The task simulator checks out descriptions of new tasks, which are passed through the control supervision.

After a successful testing out, the resulting executable tasks will be transferred by the Meta-Controller to the sequencer for storage in a dedicated task memory, from which the program can be executed on demand.

In this system, the task of the human operator reduces to initiation, supervision and acknowledgement of completed robot tasks by means of the control supervision unit. Nevertheless, the operator is always able to take control over the system, especially in case of failure or emergency situations.

III. Automatic Task Management

In a large robot system, e. g. in a scenario of space station including robots with various capabilities, e. g. mobility, multi-armed systems, different working modes and on the other hand a broad range of very different tasks automatic task management is one of the key issues. Due to the complexity of the system itself and of the range of applications the problem must be solved on an abstract level far beyond the level of move commands of commercial robot languages. Situated between the Meta-Controller as an interface to man and the system coordinator (CoS) as the intelligent interface to the single robots the automatic task management has to provide a lot of functions performing system control and the break down from the highest level of abstraction to a middle level at the robot side. At the input level complex tasks are described, which include implicitly the use of a group of appropriate robots for execution. The whole work of each task can be subdivided in parts, which can be performed sequentially or in parallel according to the special needs of the task. The system coordinator accepts coordination primitives, which address groups of robots on a multi-robot movement level. The break down to collision free move commands for single robots is performed at the levels of coordinated operation and collision avoidance. The structure of the levels mentioned is shown in principle in fig. 3 from the task input to the output to the robots.

The tasks are transferred for execution from the Meta-Controller with a task specific global priority. The task management activates the task with the highest priority, if the capabilities of the multi-robot system match the needs of the task. At this level groups of robots according to the task are defined but the robots are not yet booked. Also the group coordinators (CoG) are configured and the group collision avoidances (CAG) are initialized.

The choice and requisition as well as the derequisition of the robots with the appropriate performance capabilities is executed on the subtask level, where the subtasks provide a list of performance attributes. The subtasks are initiated according to a set of rules, which consider priorities, the logic flow of execution, time critical paths and the availability of appropriate robot systems. The actual priority of a subtask is computed based on the global task priority, the attributes of the subtask and the current system status. For these reasons the system is event driven and the complete execution sequence is in general not known in advance. Each subtask contains a number of coordination primitives, which are transferred to the system coordinator sequentially. These sequences are only broken in case of failure or emergency, while in ordinary operation the prescribed sequence of coordination primitives of the subtask is executed.

IV. Coordinated Operation and Collision Avoidance

Due to the complexity of a space robotic system containing e.g. free-flying servicers, OMV's, RMS's, SMS's, etc. robot coordination and collision avoidance have to be considered on two different levels. First on a global system level, which takes a global but rough

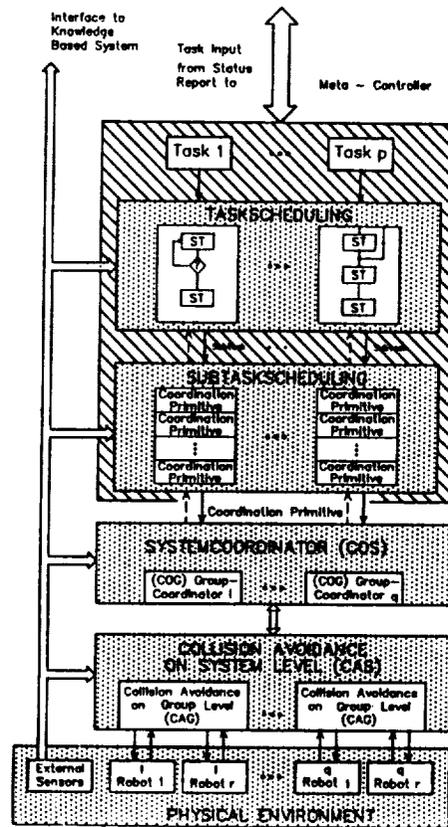


Figure 3. Substructure automatic task management, coordinated operation and collision avoidance

view over all robot systems involved. Second on a group level, which takes a close look at a number of robots working at the same task or subtask. In order to bring out the highest degree of flexibility strong real-time capabilities are required on the group level.

Considering a robot group consisting of r robots, each described by eq. (1) and controlled according to eq. (2), one obtains r sets of equations of form (3) for the closed loop robotic systems. As the robots work in coordinated operation, the reference inputs $\underline{w}_1(t), \dots, \underline{w}_r(t)$ have to be coordinated by a hierarchical coordinator of the type

$$\begin{bmatrix} \underline{w}_1(t) \\ \vdots \\ \underline{w}_r(t) \end{bmatrix} = \tilde{\underline{H}}_G(\underline{x}_1, \dots, \underline{x}_r; \underline{v}_1, \dots, \underline{v}_r), \quad (4)$$

where $\underline{v}_1(t), \dots, \underline{v}_r(t)$ symbolize the move commands in coordination space for the nonlinear controlled robots.

Applying the nonlinear control scheme eq. (2) to the individual robot arms it is /5/

$$\begin{bmatrix} \dot{\underline{x}}_1(t) \\ \vdots \\ \dot{\underline{x}}_r(t) \end{bmatrix} = \begin{bmatrix} \underline{A}_1^* & \cdot & \cdot & \cdot & \underline{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \underline{0} & \cdot & \cdot & \cdot & \underline{A}_r^* \end{bmatrix} \begin{bmatrix} \underline{x}_1(t) \\ \vdots \\ \underline{x}_r(t) \end{bmatrix} + \underline{B}^* \cdot \tilde{\underline{H}}_G(\underline{x}_1, \dots, \underline{x}_r; \underline{v}_1, \dots, \underline{v}_r). \quad (5)$$

This formulation contains the dynamics \underline{A}_i^* ($i = 1, \dots, r$) of the axes of motion of all r robots and defines a second high level control loop by means of the hierarchical group coordinator $\tilde{\underline{H}}_G(\underline{x}_1, \dots, \underline{x}_r; \underline{v}_1, \dots, \underline{v}_r)$.

The hierarchical group coordinator (CoG) eq. (5) is structured as follows eq. (6):

$$\tilde{\underline{H}}_G(\underline{x}_1, \dots, \underline{x}_r; \underline{v}_1, \dots, \underline{v}_r) = \tilde{\underline{H}}_G^a \begin{bmatrix} \underline{x}_1(t) \\ \vdots \\ \underline{x}_r(t) \end{bmatrix} + \tilde{\underline{H}}_G^b(\underline{x}_1, \dots, \underline{x}_r) + \tilde{\underline{E}}_G \begin{bmatrix} \underline{v}_1(t) \\ \vdots \\ \underline{v}_r(t) \end{bmatrix} \quad (6)$$

Equation (6) is a formulation in joint space of the robots, which shows the applicability of this method. The dynamics of the links of the robots can be arbitrarily chosen by appropriate selection of the matrix \underline{H}^a . Useful couplings between links of different robot can be introduced by \underline{H}^b . The matrix \underline{E} contains gains for input vectors in coordination space /4/. The method of nonlinear decoupling offers a useful opportunity to choose the dynamics of the robotic system not only in joint space but in other e.g. task or group oriented coordinate systems as well.

With an appropriate definition of the task variables the different modes of coordinated operation e.g. synchronisation, docking and cooperated payload handling can be implemented, while the structure of the coordinator remains the same /7/. Each group coordinator independently works on one coordination primitive and generates the input commands for the robots involved. The hierarchical system coordinator initializes the group coordinators allocates the coordination primitives and supervises the execution on a middle level, reporting the system state to the units above. In case of a collision danger between single robots of different groups the system coordinator works close together with the system collision avoidance searching collision free but still group coordinated paths. This can be accomplished by priority considerations or a reconfiguration of the robot groups.

The level of collision avoidance is responsible for a collision free operation of the whole system. This includes at first the realtime detection of collision danger between robots working in the very same group as well as between robots of different groups. Only in case of collision danger this module intervenes whereas in case of no danger of collision the original reference inputs are applied to the robots. The level of collision avoidance is split in two parts: collision avoidance on group level (CAG) i.e. between robots of the very same group and on system level (CAS) i.e. between robots of different groups. The group collision avoidance has to distinguish three major operation modes: independent action, synchronised actions and fully cooperated motion of the robots involved. It is obvious that the avoidance strategy is closely dependent on the mode of operation, but in each case the strategy can be

described by a similar structure as the group coordinator eq. (6):

$$\underline{H}_G = \underline{H}_G^a (\underline{H}_G^b) \cdot \underline{x}_I + \underline{H}_G^b (\underline{x}_I, \underline{v}_I) + \underline{E}_G (\underline{H}^b) \cdot \underline{v}_I \quad (7)$$

In this formulation the states and inputs of all robots have been noted in a condensed form

$$\underline{x}_I^T = (\underline{x}_1, \dots, \underline{x}_{rI}) , \quad \underline{v}_I^T = (\underline{v}_1, \dots, \underline{v}_{rI}) \quad (8)$$

For the collision avoidance between robots of the very same group $\underline{H}_G^b (\underline{x}_I)$ in eq. (7) is the essential part of the structure. The detection of danger of collision as well as the collision avoidance strategy are based on the calculation of the collision avoidance trajectories \underline{f}_j , ($j = 1, \dots, r$). These trajectories are determined by on-line prediction of the robot movement, regarding the current robot states the preprogrammed paths and the task oriented right-of-way priorities of the robots /6/. These trajectories are described in the elements of the matrix \underline{H}_G^b including the information if currently collision danger was detected. The matrices \underline{H}_G^a permits the change of the control dynamics in dependence on the level of danger of collision and the original inputs \underline{v}_I are cancelled by means of \underline{E}_G in case of a predicted collision.

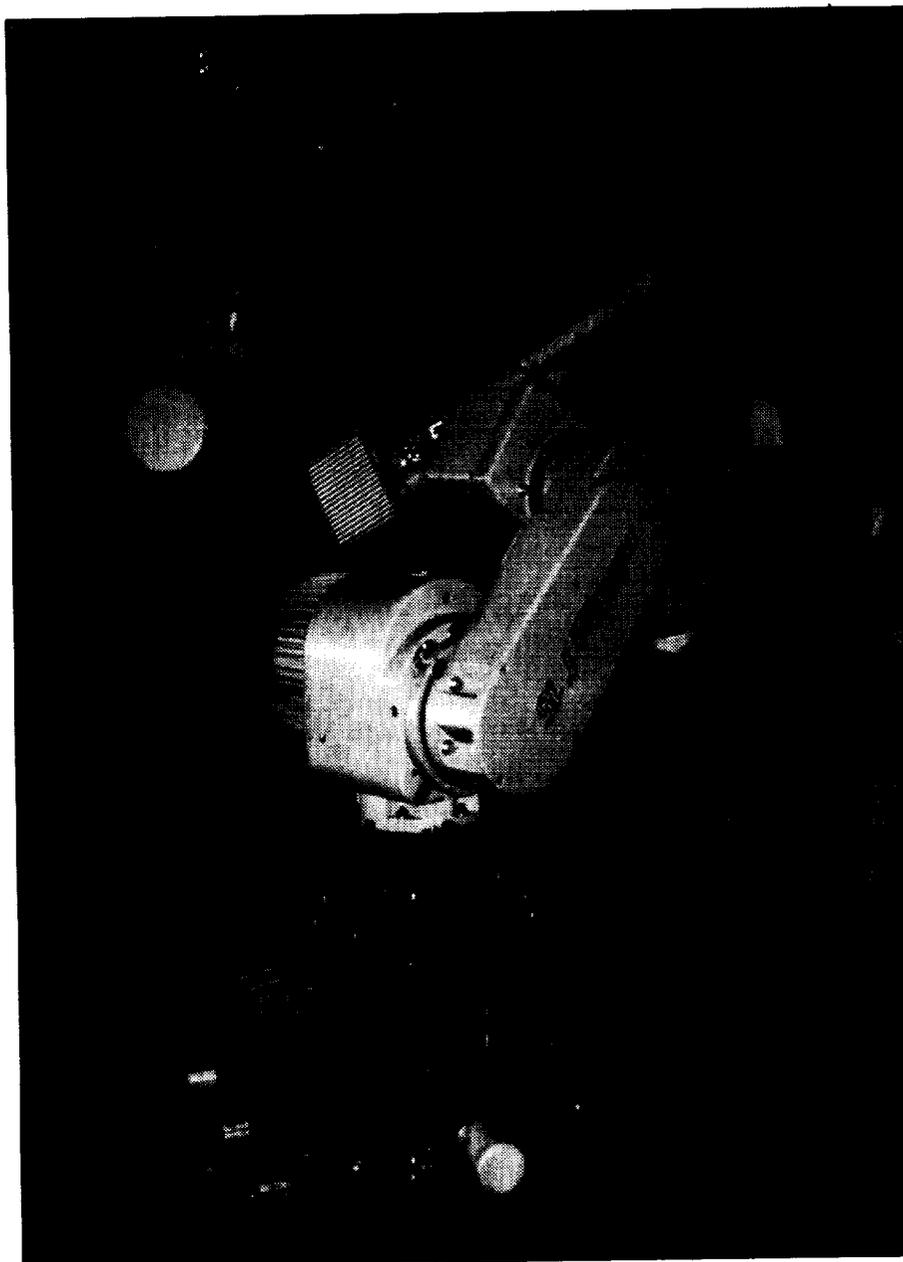
Considering now a multirobot system consisting of N groups of r_i ($i = 1, \dots, N$) robots each, where all robots are feedback controlled according to eq. (2) and each group equipped with a CAG-unit. This formulation leads to N robot groups, which are completely decoupled where the single robots of the very same group are in coordinated operation by means of CoG and under online collision avoidance by means of CAG. The case of collision avoidance between robots of different groups is implicitly included by the demand for collision free paths \underline{v}_i of robots of different groups, as inputs to CAG from the CAS unit.

The separation of collision avoidance on group and system level, respectively takes use of the fact, that mostly collision danger occurs inside a robot group, where the robots work close together. Here the group dedicated CAG provides a fast response in case of detected collision danger inside the group, taking the working mode of the robots into consideration. In case of collision danger between robots of different groups, which occurs less frequently e.g. with mobile robot systems but is of the same importance with respect to possible damage of the systems, the CAS intervenes. It generates a vector \underline{e} containing the draw back directions, which decrease the danger of collision the most. Based on this information the CoS and CoG react, where the possible constraints of coordinated operation are still kept. If conflicts or deadlock situations occur the strongest action is the reconfiguration of the groups and the CoG's taking the conflicting systems in the very same group. Otherwise evasive actions and priority increase of the conflicting robots inside the group can solve the problem.

V. CIROS test facility

To provide a realistic environment for development and test of the modules of the proposed hierarchical control structure for robots in space an appropriate facility was built up in the IRF laboratory, which is part of a national space project called CIROS (Control of Intelligent Robots in Space). Based on the modular concept it is possible to develop and

implement methods and strategies based on terrestrial robots, because for the transfer to a real space environment only the robots themselves and the low level control up to the coordinates transformation have to be changed. The upper levels of the control structure however remain the same with some minor adaptations. As test scenario an unmanned space laboratory (e. g. Spacelab, MTFF) was chosen, where typically experiment service repair tasks or experiment exchange are to be performed. The test facility is completed by a control and supervision board which could be integrated in a manned space station or in a ground based control center.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 4. Robots with common working space in the CIROS test facility

This environment provides the test facilities for all the upper levels of the control structure including different grades of automation. In order to study the problems in multi-robot systems for space applications two robots with widely overlapping working spaces are integrated. Both robots are equipped with tool exchange capabilities and additio-

nal sensors e.g. force–torque sensor, arm–mounted camera, proximity sensors ect. So each robot is able to perform every task in the system. Therefore it is possible to consider the automatic task management as well as coordinated operation and collision avoidance in a realistic environment. The hierarchical control is implemented on a real–time computer, which is interconnected with a knowledge based system and the control supervision, where time delay can be simulated between the different computers. The control board contains several input/output devices like alpha numeric terminals, graphics, video sensor ball, ect. From this board the system is supervised, a runtime documentation is done and interventions of the human operator are accepted. Additionally it is used as development facility and an off–line programming system is integrated as well as a cell simulation. A picture of the two used robots working on a rack is shown in fig. 4. In this cell, which was designed similar to a spacelab environment, the main functions of the hierarchical control structure can be implemented and demonstrated exemplarily. Especially the problems and capabilities of multi–robot systems for space applications can be studied.

VI. Conclusion

In this paper a hierarchical structure for the control of multi robot systems for space applications is presented. The break down from a high level of abstraction at task management level down to the single robot control is described step by step. The splitting in a consideration on system and on group level takes the distributed character of a large space system into account. As a possible space scenario for A&R an unmanned space station is focussed and introduced as development and test environment at the IRF–Laboratory. Based on this facility A&R with multi–robot systems can be studied at IRF in practical examples.

References

- /1/ Orientierungsrahmen Hochtechnologie Raumfahrt (OHR), Gesamtüberblick und Empfehlungen, Hrsg. DFVLR, Köln, 1987.
- /2/ Freund, E.: "Hierarchically Structured Control for Intelligent Robots in Space", Proceedings Intern. Symposium on Europe in Space – The Manned Space System –, Strasbourg – France, 1988
- /3/ Freund, E.: "Fast Nonlinear Control with Arbitrary Pole–Placement for Industrial Robots and Manipulators", International Journal of Robotics Research, Vol. 1, No. 1, 1982, pp. 65 – 78
- /4/ Freund, E.: "On the Design of Multi–Robot Systems", Proceedings IEEE International Conference on Robotics, Atlanta, Georgia, 1984
- /5/ Freund, E.; Hoyer, H.: "Collision Avoidance for Industrial Robots with Arbitrary Motions", Journal of Robotic Systems, Vol. 1, No. 4, 1984, pp. 317 – 329
- /6/ Freund, E.; Hoyer, H.: "Real–Time Pathfinding in Multi–Robot Systems Including Obstacle Avoidance", International Journal of Robotics Research, Vol. 7, No. 1, 1988, pp. 42 – 70
- /7/ Freund, E.; Kaefer, P.: "Autonomous Mobile Robots", IFAC–Symposium on Robot Control '88 SYROCO 88, Karlsruhe – FRG, 1988.