Interactive Autonomy and Robotic Skills

A. Kellner, B. Mädiger
Deutsche Aerospace AG
Raumfahrt-Infrastruktur
Huenefeldstr. 1-5
28199 Bremen, Germany
Tel 421 539 4987 Fax 421 539 5726
e-mail: bernd.maediger@erno.de

KEY WORDS AND PHRASES

Autonomy, Neural Nets, Robotic Skills, Space Robotics

INTRODUCTION

Current concepts of robot-supported operations for Space Laboratories (payload servicing, inspection, repair and ORU exchange) are mainly based on the concept of "interactive autonomy" which implies autonomous behaviour of the robot according to predefined timelines, predefined sequences of elementary robot operations and within predefined world models supplying geometrical and other information for parameter instantiation on the one hand, and the ability to override and change the predefined course of activities by human intervention on the other hand.

Although in principle a very powerful and useful concept, in practice the confinement of the robot to the abstract world models and predefined activities appears to reduce the robot's stability within real-world uncertainties and its applicability to non-predefined parts of the world, calling for frequent corrective interaction by the operator, which in itself may be tedious and time-consuming.

In this paper methods are presented to improve this situation by incorporating "robotic skills" into the concept of interactive autonomy.

CONTROL FUNCTIONS AND INFORMATION BASES FOR INTERACTIVE AUTONOMY

The control and information architecture associated with the concept of interactive autonomy can be conceived as a three-layered structure, where the top-layer (the system layer) reads in the timeline of robot, payload and subsystem tasks driving the whole system, checks the tasks for consistency and delegates them to the different recipients (robot, payloads, subsystems), the middle layer (subsystem layer) breaks down the tasks into robot- and payload-specific action sequences, instantiates their parameters and delegates them to the bottom layer (equipment layer) where the final control execution is performed.

Associated with each control layer is a database of predefined operational knowledge (timelines, action sequences, control strategies, as well as failure handling methods) and a database containing predefined environment representations (e.g. geometrical world-model for the robot) updated according to predefined transitions after action execution.

To support interaction with the real world, predefined expected sensor values (e.g. forces and torques) may be supplied with the predefined actions.

Moreover, associated with each control layer there is an MMI which allows operator interaction on the respective layer at any time during the autonomous execution of the timelines, thus providing for interactive autonomy.

NEED FOR OPERATIONAL ENHANCEMENTS

First analyses and practical experience with prototypes realizing the a.m. control and information architecture show both the power of this concept of interactive autonomy and its shortcomings.

The power of the concept is particularly apparent on system level in the case of payload servicing operations. By a suitable MMI, the coordinated, interactive robot-payload operations can easily be monitored, and whenever a change in robot-payload interaction is necessary, this can easily be achieved by changing the task sequences accordingly.

However, on subsystem-level problems can occur when there is a mismatch between predefined world-model and real-world data, e.g. due to erroneous input or update, deformation in the environment, or miscalibration of the robot, or when objects need to be handled which have not been foreseen in the world-model or which are not amenable to modelling, e.g. hoses and cables.

Operator intervention on subsystem-level in this case implies selection of robot action sequences and action parameter tuning, which can be extremely tedious and time-consuming.

Of course, operator intervention on equipment level, i.e. by telemanipulation (joystick control) seems more appro-priate in these cases.

However, if the control is performed from the ground, the command-feedback round-trip time of several seconds again leads to tedious and time-consuming operations, not to speak of the problems inherent per se in fine-manipulation using video feedback.

The same applies to problems which may occur on equipment-level during control execution, such as jamming in insert/extract operations.

Obviously, some type of sensor-based control algorithms would be required to eliminate these problems.

However, in general these cannot only be of the type providing closed-loop sense-act cycles (e.g. for force/torque-based compliant motion) but need to provide strategies based on general knowledge, e.g. how to grasp objects which are not amenable to modelling in a world-model, such as hoses or cables. This leads to the concept of "robotic skills" as an additional, essential ingredient of the concept of interactive autonomy.

ROBOTIC SKILLS

As examples, in the following two skills are presented: the "grasping skill" and the "insert/extract-skill".

In the first case, the robot is provided with the ability to grasp an a priori unknown object indicated by placing the cursor on its 3D-video image generated by a pair of gripper cameras - certainly an enhancement of the a.m. concept of interactive autonomy, which would otherwise require action sequence selection and parametrization "by hand", or telemanipulation as explained above.

In the second case, the skill provides for a general jamming-free insertion/extraction capability.

Grasping Skill

This skill comprises an image preprocessing function which segments out the object indicated by the cursor, and a "sensomotory mapping" which incorporates generic knowledge for mapping object images onto robot commands such that the gripper can grasp the objects. In the following, only these sensomotory mappings are discussed further:

Since they represent generalized "grasping knowledge" which is not easily amenable to explicit (algorithmic) coding, the approach taken was to encode them in Neural Nets trained on a set of samples and to investigate the generalization capability of these mappings.

In the first, straightforward analysis a 3-layered backpropagation net was trained on a large number of objects, each in various orientations, together with the corresponding correct grasping poses of the robot, thus providing mappings from object shape and orientation to robot commands. Essentially these commands are joint angle increments which improve the gripper pose relative to the "graspable" area of the object. After each increment execution, the sensomotory mapping is performed again, thus providing a "servoing" on the object's shape. However, training times appear to be quite prohibitive and, in particular, the generalization capabilities to non-trained shapes is not satisfactory.

In a second approach the image of the indicated object is scanned for grasping areas by means of a filter realized by a 3-layered backpropagation net which has learned the human (!) assessment of a large number of object-partitions which can be grasped and partitions which cannot be grasped by the robot. This method produces excellent results in acceptable computation times.

Surprisingly, a third method also proved very promising: in this case both architecture and synaptic weights of a Neural Net were designed "by hand" such that as soon as an area fitting between the gripper fingers is detected by the first layer of neurons as the robot slowly rotates (by default) the gripper cameras over the object, the shape of the area generates robot commands such that the area's line of gravity is aligned with the symmetry line between the gripper fingers. Grasping is performed when the width of the aligned are is identified by the net as large enough for the robot's gripper. However, this method only applies for objects with not too complex structures of the grasp surfaces.

Of these three approaches, the first was analyzed by simulation only. In the latter two cases both simulation and subsequent testing on a 6 DOF commercial robot with gripper cameras were performed.

Insert/extract-Skill

In this case the "sensomotory mapping" is given by the mapping of force/torque-histories typical for imminent jamming (measured by suitable sensors in the robot's wrist) onto appropriate corrective robot commands to avoid the jamming situation in insert or extract operations.

Input signals are the 6 components of the force/torque signals and the current position of the robot. In order to incorporate the temporal evolution of the input signals, backpropagation nets with tapped delays are used. The difficulty lies in the training procedure: the only possibility is to record a large number of examples of a human operator performing jamming-free inserts/extracts or remedies in case jamming is imminent, and to train the net on this human behaviour.

First tests already showed promising results. However, further investigation is necessary to provide a truly general insert/extract-skill module.

CONCLUSIONS

The current concept of interactive autonomy for robot operations in Space Laboratories can be enhanced by robotic skills. Since these imply complex sensomotory mappings not easily amenable to explicit coding, training these mappings by Neural Nets seems to be an appropriate approach.

First tests with such Neural-Net-based skills for grasping and insert/extract operations provided promising results and appear to undergird the feasibility of the method of neural control.