Towards Supervising Remote Dexterous Robots Across Time Delay*

Kimberly Hambuchen†, William Bluethmann, Michael Goza, Robert Ambrose

Automation, Robotics and Simulation Division
NASA Johnson Space Center
Houston, TX 77058

{hambuchen, bluethmann, sgoza, ambrose }@jsc.nasa.gov

Kevin Wheeler

Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA 94035

kwheeler@mail.arc.nasa.gov

Ken Rabe

Planning Software Systems Group
Jet Propulsion Laboratory
Pasadena, CA 91109

kjrabe@jpl.nasa.gov

The President's Vision for Space Exploration, laid out in 2004, relies heavily upon robotic exploration of the lunar surface in early phases of the program. Prior to the arrival of astronauts on the lunar surface, these robots will be required to be controlled across space and time, posing a considerable challenge for traditional telepresence techniques. Because time delays will be measured in seconds, not minutes as is the case for Mars Exploration, uploading the plan for a day seems excessive. An approach for controlling dexterous robots under intermediate time delay is presented, in which software running within a ground control cockpit predicts the intention of an immersed robot supervisor, then the remote robot autonomously executes the supervisor's intended tasks. Initial results are presented.

Index Terms – control over time delay, space exploration, remote robots, task assistant, sensory egosphere, autonomous tool use.

I. Introduction

In January 2004, the President's Vision For Space Exploration laid out the strategic plan for exploration of the solar system [1]. Early stages of the plan call for robotic missions to the Earth's moon to demonstrate new technologies and to initiate work on operations prior to the arrival of human astronauts. Functions to be completed during these precursor missions will likely include mapping the lunar surface, precision landing, environmental monitoring, communications network setup, infrastructure build-up and in-situ resource utilization [2]. The machines and robots to complete these tasks will inevitably be varied in shape and form. Within this portfolio of machines, there will be robots that have manipulative capabilities. These dexterous robots will likely have some degrees of autonomy, but the need to control these robots from the ground will also exist.

Based on the speed of light, the round trip delay between issuing a command from Earth to the moon and seeing any result from that command is on the order of 1.5 seconds. A round trip time delay closer to 10 seconds is possible with data being routed through various satellites. Even under a 1.5 second time delay, bilateral control without compensation causes instabilities in a robot; bilateral control under delays up to 10 seconds will be very challenging.

Research on controlling remote robots over time delay has been occurring for many years, with solutions to the problems generally falling into one of four methods. From a control systems perspective, the simplest method is the "bump and wait" technique, *i.e.* a teleoperator inputs small commands then waits for the motion to settle. The "bump and wait" solution can be effective. With this solution, though, a teleoperator wastes a large amount of time by sitting idle. Astronauts on the Space Shuttle and International Space Station (ISS) employ the "bump and wait" technique when berthing large payloads with remote manipulator systems, although no time delay occurs between command and feedback.

Predictive display methods immerse the teleoperator in an environment with solid or wire-frame virtual models of the remote location overlaid onto live video. The teleoperator can view past, present and future states of the remote robot. Past views are represented by delayed video. Present views are found in predictions of the current state of the robot based on past commands from the previous time delay period and a model of the remote environment. The future is represented by the commands currently leaving the ground. This method augmented with intelligence on-board the remote robot was used for ground control on the ROTEX experiment that flew aboard STS-55 in 1993 [3].

A significant amount of effort has been put forth in stabilizing bilateral control of manipulators across time delay. The seminal works in this area were published by Anderson and Spong (scattering theory) and extended by Niemeyer and Slotine (wave variables) [4,5]. In 1999, bilateral control of sliding and peg-in-hole tasks were successfully completed across a 7 second time delay during

* NASA and the Mobile Autonomous Robot Software program in the DARPA Information Processing Technology Office (IPTO) sponsored this work

† Research performed as NRC Research Associate

the Engineering Test Satellite 7 (ETS-VII) experiment [6]. This controller employed a modified PD controller for the bilateral control.

The fourth and final technique, supervisory control, attempts to circumvent the time delay problem by breaking the direct link between the teleoperator and the remote robot. Commands flowing from the supervisor are primarily symbolic. This form of control requires autonomous capabilities on the remote robot to execute the symbolic commands. As the remote robot performs its work, the supervisor also serves as a monitor for the robot. The ROTEX experiment [3] also tested a form of supervisory control, identified as "tele-sensor programming". Sheridan provides an in-depth review of all previously mentioned methods for controlling remote robots across time delay [7].

This paper describes new work in the area of supervisory control being performed in the Dexterous Robotics Laboratory at NASA's Johnson Space Center. A system is being developed to control remote dexterous robots across time delays of up to 10 seconds. The robot supervisor operates within a "smart cockpit". The supervisor guides the remote robot's operations while working at a comfortable pace in an immersive virtual world. As the supervisor functions in this immersive world, a Task Level Assistant advises the supervisor on task sequences. The Supervisor Intent Prediction software monitors the supervisor's motions and interprets them as symbolic commands. The remote robot can autonomously execute these symbolic commands as they are predicted, thereby shortening the time needed to complete tasks. Results from initial experiments using the NASA Robonaut system to manipulate Extravehicular Activity (EVA) handrails under time delay are presented.

II. System Description

The system for supervising remote operation of robots across time delay consists of the smart cockpit and the robot, each existing on separate sides of the time delay. The cockpit has both hardware and software components, as does the robot.

A. Cockpit Hardware and Design

The cockpit was designed to meet several criteria concerning the structural integrity and the human-factors goals of the cockpit. Structurally, the cockpit houses all electronics, monitors, and workstations for the personnel required to manage the remote robot. The personnel consist of a supervisor, a robot systems manager, and a safety manager. As previously mentioned, the supervisor is responsible for the commands sent to the robot. The robot systems manager oversees the health of the robot at a core-level and assists the supervisor as needed. The safety manager monitors all robotic and supervisory activities to limit and/or prevent harm to the robot, supervisor and all other cockpit components. The robot systems and safety managers should be able to see the supervisor at all times for communication and safety purposes. Fig. 1 shows the current cockpit as it exists at JSC. The supervisor position in the cockpit is located on the raised platform, allowing other operators to view all supervisory activity. The workstations for each of the three personnel in the cockpit contain a center monitor that can display video or computer images in either quad view or single view mode. Two smaller monitors flank the center display. All monitors can display scenes from any computer system in the cockpit and an array of video channels. The workstation also contains a wireless mouse and keyboard. The monitors and keyboards are controlled using touch screens located on bendable booms within the workstation.

Fig. 1 Cockpit Structure

The supervisor workstation has additional hardware for virtual reality (VR) immersion. The equipment consists of a Kaiser Pro-View helmet-mounted display, two Immersion Corporation CyberGloves®, a Polhemus magnetic tracking system and a Phoenix Technologies, Inc. optical tracking system. While both tracking systems exist, only one is selected and used during remote operations. Each tracking system monitors the teleoperator's body motion and converts the motions into robot commands. These commands are either sent directly to the robot or converted into symbolic commands.

All electronics required to run the cockpit are housed under the supervisor's platform. The electronics consist of eight desktop computer systems, virtual reality equipment controllers, and a half high rack of video switchers, VGA switchers, and keyboard/mouse switchers. All cables run in trays located throughout the structure. A final criterion needing to be met is portability - the entire structure can be broken down, moved/shipped to another location and re-assembled. Disassembly/reassembly of the structure can be accomplished in about 8 hours with a team of 3-5 people.

B. Cockpit Software

The cockpit software consists of an assistant that monitors task lists for both supervisor and robot, a supervisor intent predictor and an immersive environment generator. Fig. 2 shows the software architecture on the cockpit side of the time delay.

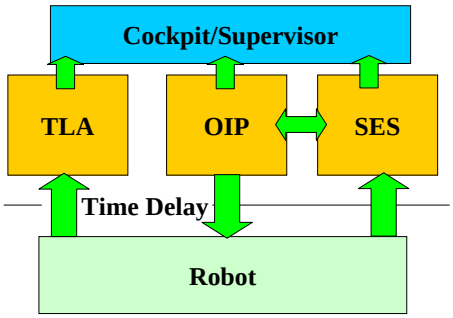

Fig. 2 Architecture of the Cockpit Side of Time Delay

1) Task Level Assistant

The Task-Level Assistant (TLA) monitors tasks performed by both the supervisor and the robot. The TLA generates a task plan using a set of *a priori* goals. The plan consists of high level tasks (*e.g.* replace damaged handrails, set up EVA platform) that can be decomposed into activities (*e.g.* grasp handrail, move to box) performed by the supervisor in an immersive world and by the robot in its real-world environment. As the supervisor follows the task plan, the TLA tracks each activity and indicates when the supervisor completes an activity. The TLA then updates the plan and displays the next activity to cockpit personnel. The same process occurs for the remote robot. Cockpit personnel can at any time monitor activities that have yet to be performed, activities that have been completed by the supervisor but not the robot, and activities that have been completed by both the supervisor and the robot.

Ideally, the TLA operates as an automated checklist that continually illustrates to the supervisor what needs to be done as well as what has been done. In the real world, deviations from the plan will occur on both sides of the time delay. A deviation will require re-planning of the current set of tasks to generate a new plan that allows the supervisor to accomplish the complete set of tasks.

Deviations from expected behavior fall into four categories: 1) intentional supervisor deviation, 2) unintentional supervisor deviation, 3) loss of resource and 4) missing resource. To handle supervisor deviations, the TLA queries the supervisor to establish whether the deviation was intentional. For intentional deviations, the re-planned tasks should not attempt to undo the supervisor's action. When an unintentional deviation occurs, re-planed tasks should attempt to immediately undo the supervisor's actions. Loss of resource is indicative of a failure on the robot. In this situation, the TLA re-planning process should take into account tasks that need a lost resource and alternatives to using that resource. For example if the robot's right arm fails, new tasks should attempt to use the left arm instead. Missing resources indicate that environmental expectations were not met, *i.e.* objects were not where they were expected to be. This type of deviation will typically require significant supervisor interaction to get back to a valid task plan as it is significantly outside of normal operations.

2) Supervisor Intent Predictor

The purpose of the Supervisor Intent Prediction (SIP) software is to predict the supervisor's intended actions from motion commands. If the SIP correctly predicts the supervisor's intent, the prediction becomes a symbolic command that triggers the predicted activity as an autonomous action on the robot side. For the SIP to be worthwhile, the prediction must trigger an activity on the robot side before the robot receives teleoperated-commands. Therefore, predictions must occur before the supervisor begins to execute the task. The supervisor commands are measured in terms of commanded end effector position (x, y, z, roll, pitch, and yaw) as sensed by the tracker and hand shape (angles of the fingers) as measured by the CyberGloves®.

The SIP software uses a state machine embedded with Hidden Markov Models (HMMs). Each HMM is responsible for a particular type of movement such as reaching for a horizontal handrail or grasping a particularly shaped object [8]. As the supervisor begins to move, the probability of each HMM associated with a particular gesture with respect to the task plan is computed. The most likely model prediction and the confidence in that model are transmitted to the other cockpit software modules. Currently the models used recognize reaching for horizontal and vertical handrails, hand openings, and hand closings. The SIP continuously monitors the supervisor's movements and provides predictions at a 100Hz frequency.

3) Immersive Environment

In an immersive environment, the supervisor can perform desired tasks at a faster rate than the robot and without waiting for feedback from the robot. The Sensory Ego-Sphere (SES) enables an immersive environment by representing known information about the robot and its environment in an egocentric manner. The SES is a short-term memory for Robonaut that exists as a virtual geodesic dome interface linked to a database [9]. Vertices on the dome link to records in the database creating nodes. Data sensed in the environment are stored at nodes closest to their direction of origination. The SES is centered at Robonaut's chest and the vertices correspond to (azimuth, elevation) angle pairs. The geodesic nature of the SES allows for quick and efficient search of the sensory space.

The SES communicates with the visualization software RoboDisplay [10] to display data stored in the SES in the environment of a simulated version of Robonaut, which is commanded by the supervisor. The merging of the SES with RoboDisplay lets the supervisor see the virtual world populated by representations of detected real-world data and the virtual robot using the HMD. Essentially, the SES creates a virtual world in which the supervisor can execute a set of tasks.

The SES can use information from the SIP to alter the supervisor's environment. The confidence with which the SIP predicts activities and the targets of those activities can be conveyed to the supervisor visually by associating the confidence with the color or transparency of targets. This feedback lets the supervisor know if actions performed are being correctly predicted. The visual feedback also lets the

supervisor move to the next activity without finishing the predicted activity. This in turn may alter the supervisor's actions which can affect the SIP prediction. The SES also enables the supervisor to manipulate and move objects in the environment. To mimic the robots abilities, the supervisor must be able to grasp, move, release and interact with objects. This requires linking the virtual objects to the supervisor's virtual simulation detection of grasps, releases or other activities and knowledge of any objects in position for the desired interaction. The SES can provide the knowledge of nearby objects and whether they can be manipulated in the desired fashion. The SES can also alter the linkage between the virtual robot and virtual objects so that they appear in the correct spot for the supervisor. The hexagonal neighborhoods on the virtual geodesic surface of the SES afford a quick and efficient search for objects in desired locations to perform this linkage. Fig. 3 shows (a) the simulated robot in a virtual SES and objects in its environment, (b) the left eye view and (c) right eye view as seen by the supervisor.

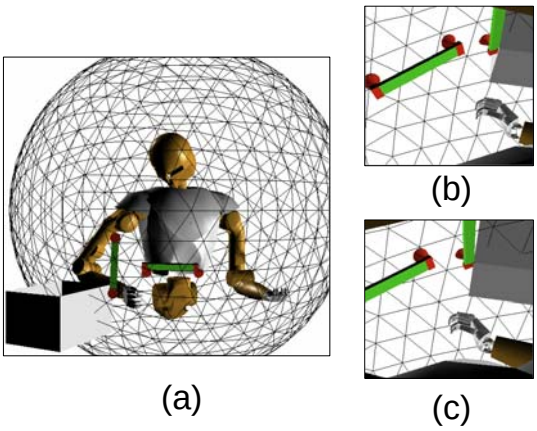

Fig. 3 Simulated robot in SES (a), supervisor's left eye view (b) and supervisor's right eye view (c)

C. *Robonaut Hardware*

The Robonaut systems are anthropomorphic humanoid robots specifically designed for space. The robots have over 40-DOFs each with two 7-DOF arms, each ending with a five-fingered hand. Both Robonaut Unit A and Unit B integrate technology advances in dexterous hands, modular manipulators, and lightweight materials. The Robonaut systems have articulated waists that, combined with the anthropomorphic arms, allow for large workspace areas. Both systems have heads that house pan/tilt stereo vision cameras which provide visual information for both teleoperators and vision processing. The Robonaut systems possess the correct anatomy to function with existing EVA tools and hardware. While Unit A is stationary, Unit B may operate either on a mobile platform for traveling the surface of planets or using a single leg designed to attach to ISS worksites used by astronauts. Robonaut Unit A is currently being used for the work described in this paper. Fig. 4 shows Robonaut Unit A.

Fig. 4. Robonaut, the NASA/DARPA humanoid robot

D. *Robonaut Software*

The proposed technique of predicting a supervisor's intent and executing that intent on a robot is predicated on the remote robot having autonomous capabilities. Robonaut's autonomous capabilities include a number of primitive behaviors (*e.g.* move to touch, grasp to position or force, track object,) and a few task-oriented combinations of these primitives.

An arbitration module serves as the primary interface between the robot and the cockpit. The arbitrator judges the quality of the prediction generated by the SIP using the prediction's probability. If the arbitrator deems the prediction good, it initiates the behavior controller to begin execution. Because the supervisor works ahead of the robot, multiple tasks can be predicted while the robot performs other tasks. The arbitrator also queues the predicted tasks and moves these tasks from the queue when the robot becomes idle.

Lastly, the arbitrator allows the supervisor to switch control of the robot between supervisor-guidance and teleoperation. Teleoperation mode is required when attempting to complete a task beyond the autonomous capabilities of the remote robot or for error recovery. Fig. 5 illustrates the software architecture on the robot side of the time delay.

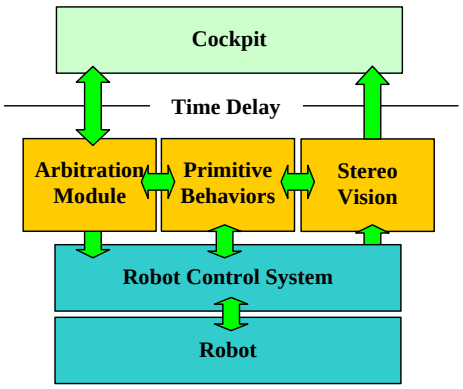

Fig. 5 Architecture of the Robot Side of the time delay

III. Initial Experiments and Results

A. *Experimental Design*

The experimental setup was designed to mimic a possible panel configuration existing on the outside of the ISS. A taskboard used to represent the ISS contains two separate EVA handrails in a perpendicular fashion. A requirement of the SIP was to see at least two objects within the robot's FOV. Without this requirement, the SIP would simply need a gaze vector from the robot to determine what object is a task or activity target. Another requirement of this setup was that it be realistic, *i.e.* a similar setup could be found on the ISS, so that activities and tasks tested were practical. The taskboard and an EVA box for dropping handrails form the experimental setup. The tasks that comprise the experimental protocol are listed below.

1. Supervisor grasps vertical/horizontal handrail.
2. Supervisor moves that handrail to an EVA box.
3. Supervisor releases handrail over EVA box.
4. Steps 1-3 are repeated for the horizontal/vertical handrail.

This protocol simulates actions that might be taken by Robonaut on the outside of the ISS (*e.g.* removing burred handrails, disassembling EVA setup for astronauts).

B. *Grasping Experiments*

To establish a baseline for the robot's autonomous grasping of handrails, trials were conducted in which the supervisor directly teleoperated Robonaut to grasp a rigidly mounted handrail. Results from teleoperation trials run without a time delay and with a two second round trip delay are shown in Table 1.

TABLE 1
RESULTS OF HANDRAIL GRASPING

Grasp Method	Completion Time, s	Time In Contact, s	Maximum Force, N	Integral of Force, N-s
No delay	25	10	117	542
2 second delay	48	23	137	1308
Automated	35	25	28	330

For both sets of trials, the teleoperator relied only on visual feedback. The results show that the teleoperator was more capable of performing the task without delay, but the operator was able to complete the task using a "bump and wait" technique. The delayed task took nearly twice as long to complete because of this technique. The lack of feedback, though, led to similar maximum force values for the delayed as well as the non-delayed trials. The integral of the contact force, which is a measure of wear and tear, is much greater under time delay. This is primarily driven by the longer time in contact.

The grasping algorithm is a state machine guided by Robonaut's stereo vision system, as shown in Fig. 6. For more information on the stereo vision system, see [12]. During the initial traverse to the handrail approach point, the grasping algorithm disables Cartesian control of the end effector pitch. At the end of the approach, a grasping decision is made using a model of the hand and handrail that maximizes the approach corridor to the handrail and the likelihood of a successful grasp. During the grasp sequence, the manipulator uses a damping control law with relatively high gain damping gains (3500 N/(m/s) force-130 Nm/(rad/s)-moment)b/in/s), low thresholds (11 N/0.5 Nm) and low saturation (22 N/2.25 Nm). The nonlinear effects of the thresholds and saturation allow for a responsive system that does not engage inadvertently due to modeling errors in the arm distal to the load cell. These effects also inhibit the arm from becoming unstable during contact with the environment.

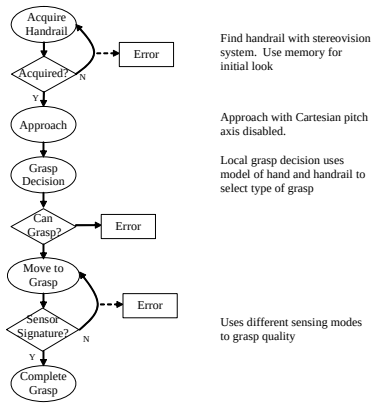

Fig. 6 Autonomous grasping algorithm

The results for the robot's autonomous reaching to and grasping of the handrail are shown in Table 1. In comparison to a teleoperator, autonomous grasping takes longer to complete but significantly reduces peak contact loads and the measure of "wear and tear" (the integral of the force).

C. *Immersive Environment Grasping*

Trials were run over three months using the experimental protocol described previously. A single supervisor performed the experiments in the fully immersive virtual environment. Over the trials, natural starting position and the positions of handrails were varied. Initially, data was collected to develop models necessary for the SIP. Once configured, the SIP was incorporate into the system and more trials were run. During these trials, the supervisor received feedback about predictions via variations in colors of handrails. Data collected on these trials were used to adjust the SIP models. Trials were run once more to determine the average time to prediction. The typical prediction time ahead of completion is approximately five seconds. This prediction time is more than adequate to compensate for a two second round-trip time delay under the robot's normal operating conditions.

D. *Integration Testing*

Complete integration testing was conducted with the cockpit system under a variety time delays. Three trials apiece were conducted with round trip time delays of 2, 5, 7 and 10 seconds. The SIP predicted the target handrail when the supervisor began his reach on all trials. The SIP predictions were successfully sent across the time delay as symbolic commands and executed by the robot. For integration tests, all systems operated nominally, using an experienced supervisor, with handrails in nominal positions, and no deviations from the plan, either intentional or un-intentional occurred.

The results of these tests show no correlation between the time delay and the supervisor's task-completion time in

the virtual world. This is expected since the supervisor operated in a virtual world and worked independently of the time delay. A positive correlation exists between the time delay and the robot's task-completion time. This correlation is attributed to the extra time needed for predicted events to reach the remote robot under longer delays. Because the supervisor did not have direct teleoperated control of the robot, no correlation exists between the ability of the robot to perform a predicted task and time delay.

For the most complex aspect of the task (reach and grasp handrail), the metric of task completion time divided by unidirectional time delay (T_c / T_d) ranges between 5 (for a 5 second unidirectional delay) and 25 (for 1 second unidirectional delay). Tasks with a T_c / T_d metric between 5 and 25 highlight the utility of this approach to controlling robots across time delay. The supervisor works at a natural human pace. However, the task-completion times are sufficiently large in comparison to the time delay so that the supervisor can intervene if the SIP predicts an incorrect task or the robot has trouble completing a task.

IV. Conclusions and Future Work

We have presented the initial effort for supervising remote dexterous robots over intermediate time delays. The need for remote supervision arises from the decreasing number of astronauts available to teleoperate robots in-orbit and remote supervision may also be useful for operating robots on surfaces such as the moon. Situations like these will produce anywhere from a 2 to 10 second time delay. The relatively small time delay combined with the dexterity of robots like Robonaut create a unique situation in teleoperation. Our solution involves the combination of robot autonomy with direct human teleoperation.

A smart cockpit was developed at JSC to house the supervisor, robot systems manager and safety manager as well as all hardware needed to run both supervisory and robot systems. The cockpit (Earth) side of the time delay contains a Task Level Advisor to track tasks performed by both the supervisor and the robot. Supervisor Intent Predictor software on the cockpit side predicts the supervisor's intent so that tasks may be guided rather than teleoperated on the robot side. The supervisor operates in an immersive environment when not directly teleoperating the robot. This environment combines a Sensory Ego-Sphere with visualization software that allows the supervisor to perform tasks virtually using virtual reality equipment located in the cockpit. Robonaut, the dexterous robot, uses autonomous motions to follow the supervisor's guidance when possible. During unknown tasks or error situations, the supervisor is required to teleoperate Robonaut. Initial tests that integrated all software and hardware on both robot and cockpit sides show success under several different intermediate time delays.

For future work, we will begin by increasing the complexity of the experimental setup. More tools will be introduced to the robot's workspace both to increase population of the SES and to raise the difficulty level for the SIP. Different tools also entail different grasps or motions for Robonaut. Therefore, Robonaut's autonomous dexterous motions will be expanded to include grasping of other objects and pushing and pointing motions. The SIP will incorporate predictive models in a generative fashion to establish an automated method of generating low-level command sequences for sub-task level automation. This is challenging because the automation developed to date works to minimize grasping forces of which the teleoperator is typically unaware. Thus models built from teleoperator command sequences apply more force than desired.

Once experiments using the above revisions have been completed and proven successful, this work will be expanded to Robonaut Unit B, which is mobile. Mobility introduces a new factor into the research. The robot will need autonomous mobility motions while the SES will be expanded for mobile memory and the SIP will need to predict the mobile motions. With a mobile robot, experiments can be extended to include climbing tasks.

References

[1] "The Vision for Space Exploration," National Aeronautics and Space Administration, www.nasa. gov/pdf/55583main_vision_space_exploration2. pdf, February 2004.

[2] "Robotic Lunar Exploration Program Requirements Document, Revision A", National Aeronautics and Space Administration, exploration. nasa. gov/documents/ESMD-RQ-0014_Rev_A_RLEP. pdf, February 2005.

[3] Hirzinger, G. , Brunner, B. , Dietrich, J. , Heindl, J. , "Sensor-Based Space Robotics-ROTEX and its Telerobotic Features," *IEEE Transactions on Robotics and Automation*, 9(5), pp 649-663, 1993.

[4] Anderson, R. , M. Spong, "Bilateral control of tele-operators with time delay," *IEEE Transactions on Automatic* Control, 34(5), pp 494-501, 1989.

[5] Niemeyer, G. , Slotine, J. , "Stable adaptive teleoperation," *IEEE Journal of Oceanic Engineering*, 16(1), pp. 152-162, 1991.

[6] Imaida, T. , Yokokohji, Y. , Doi, T. , Oda, M. , Yoshikawa, T. , "Ground–Space Bilateral Teleoperation of ETS-VII Robot Arm by Direct Bilateral Coupling Under 7-s Time Delay Condition," IEEE Transaction on Robotics and Automation, 20(3), 2004.

[7] Sheridan, T. "Space Teleoperation Through Time Delay: Review and Prognosis,", IEEE Transactions on Robotics and Automation, 9(5), pp 592-606, 1993.

[8] Wheeler, K.R, R. Martin, V. SunSpiral, M. Allan, "Predictive Interfaces for Long-Distance Teleoperation," *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (ISAIRAS 2005)*, September, 2005.

[9] Peters, R. A. II, K. A. Hambuchen, K. Kawamura, D. M. Wilkes, "The Sensory Ego-Sphere as a Short-Term Memory for Humanoid Robots," *Proceedings of the 1st International Conference on Humanoid Robotics (Humanoids 2001)*,

[10] Goza, S. M., "RoboDisplay User's Manual," June, 2003.

[11] Diftler, M.A., R. O. Ambrose, S.M. Goza, K.S. Tyree, E.L. Huber, "Robonaut Mobile Autonomy: Initial Experiments," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[12] Huber, E. , Baker, K. , "Using a Hybrid of Silhouette and Range Templates for Real-time Pose Estimation", *Proceedings of IEEE International Conference on Robotics and Automation*, 2004.