

# Supervising Remote Humanoids Across Intermediate 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

Kenneth Rabe

*Planning Software Systems Group  
Jet Propulsion Laboratory  
Pasadena, CA 91109*

kjrabe@jpl.nasa.gov

Mark Allan

*Intelligent Robotics Group  
NASA Ames Research Center  
Moffett Field, CA 94035*

mallan@email.arc.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 humanoids under intermediate time delay is presented. This approach uses software running within a ground control cockpit to predict an immersed robot supervisor's motions which the remote humanoid autonomously executes. 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 humanoids that have manipulative capabilities. These dexterous robots will likely have some degrees of autonomy but may need to be supervised by humans. Due to the limited number of astronauts and their full schedules while in space or on the lunar surface, ground personnel will likely need to remotely supervise these robots.

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 would be very challenging.

Solutions for controlling remote robots over time delay usually fall into one of four methods, or a combination of these methods: 1) "move-and-wait", 2) bilateral control stabilization, 3) predictive displays and 4) supervisory control.

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, though the 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.

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) [3, 4] and Bejczy and Kim (shared compliance control) [5]. In 1999, bilateral control of sliding and peg-in-hole tasks were successfully completed across a 7 second time delay during the Engineering Test Satellite 7 (ETS-VII) experiment [6]. This controller employed a modified PD controller for the bilateral control.

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

---

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

used for ground control on the ROTEX experiment that flew aboard STS-55 in 1993 [7]. Predictive displays were also used for telemanipulation over time delay by Bejczy and Kim [6]. Whereas the shared compliance control was used for direct manipulation tasks, predictive displays were used to enhance the teleoperator's performance while the robot was in free motion.

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 [7] also tested a form of supervisory control, identified as "tele-sensor programming". Kheddar *et al.* created a supervisory control system in which the teleoperator commands the robot from a virtual environment [8]. The virtual tasks are decomposed into low-level motions and are mapped onto the robot's state space rather than using autonomous motions. With this system, the teleoperator must wait for each task to be completed before moving onto another activity.

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 to enable control of remote humanoids across time delays of up to 10 seconds has been developed. The robot teleoperator becomes a supervisor operating within a "smart cockpit". The supervisor guides the remote robot's operations while working at his own pace in an immersive virtual world representing the robot's environment. The Task Level Assistant advises the supervisor on task sequences and tracks which tasks have been completed successfully or not by both the supervisor and the robot. The Supervisor Intent Prediction software monitors the supervisor's motions and transforms predicted tasks into symbolic commands. The remote robot autonomously executes these symbolic commands as they are predicted rather than as they are performed. This shortens the time needed for the humanoid to complete tasks. Since this system allows a teleoperator to work at his own pace, teleoperator fatigue is much less likely to set in allowing the teleoperator to perform more tasks in less time. Paul *et al.* [9] describe their system which uses teleprogramming to control a remote robot over time delay. They highlight the notion that their work is not supervisory control, but it does perform functions similar to the work described in this paper. Their system situates a teleoperator in a graphical model of the remote site and converts the teleoperator's motion commands into symbolic commands transmitted to the robot. This approach does not allow the robot to "get ahead" of the teleoperation, however, like the "smart cockpit" approach.

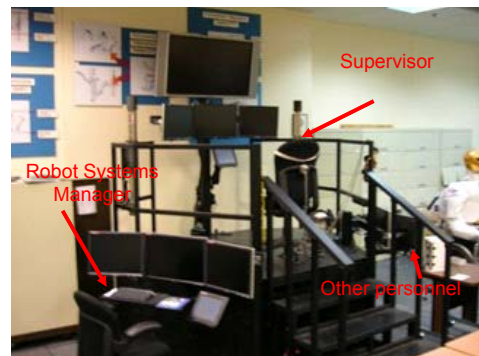
## II. SYSTEM DESCRIPTION

The system for supervising remote operation of a humanoid across time delay consists of the smart cockpit and the robot, each on separate sides of the time delay. This

section describes the hardware and software of these components.

### A. Cockpit Hardware

The cockpit design provides component housing, multiple personnel seating, and an array of display options for viewing the multiple video sources from the robot's remote environment and from computer displays. The design also houses all electronics (computers, video switchers, *etc.*) required to supervise the robot. Three workstations exist in the cockpit, each of which contain displays to provide video and computer imagery and a wireless mouse and keyboard. The cockpit can seat the supervisor, a robot systems manager and a third person as needed (safety manager, mission specialist, *etc.*). The monitors and keyboards can be switched to control any cockpit computer using touch screens located on bendable booms within the workstation. The supervisor workstation on the cockpit platform 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. Each tracking system monitors the teleoperator's body motion and converts the motions into robot commands, although only one tracking system is used at a time. These tracked commands are sent directly to the robot or converted into symbolic commands by cockpit software. Figure 1 shows the cockpit in the DRL.



### B. Cockpit Software

The cockpit is responsible for directing tasks and converting the supervisor's motions into symbolic commands for the remote robot.

#### 1) Task Level Assistant

The objective of the Task Level Assistant (TLA) is to monitor task completion for both the supervisor and the robot and to re-plan tasks accordingly. The TLA generates an initial 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). The TLA tracks each activity performed by the supervisor and indicates when an activity is complete, after which it updates the plan and displays the next activity to cockpit personnel. The same process occurs for the remote robot. The TLA lists all activities: those that have yet

to be performed, those that have been completed by the supervisor but not the robot, and those 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 ultimate goals. 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-planned 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 Supervisor Intent Prediction (SIP) software predicts the supervisor's intended actions from his motion commands during teleoperation. The objective of this software module is to observe the supervisor's actions and make educated guesses as to what the supervisor's *next* action will be. When the SIP predicts a supervisor's intended action, the prediction is converted into a symbolic command that triggers a related autonomous action in the robot. By predicting the supervisor's actions with a high level of accuracy several seconds before the actual action is performed, the robot is given the opportunity to work ahead of teleoperation commands and mitigate the time delay. Two versions of this software were developed and tested. It is currently assumed that all of the supervisor's tasks center around manipulation of and motion about objects.

The first generation SIP software monitors the supervisor's commands using a state machine embedded with Hidden Markov Models (HMMs) while also observing the TLA output. The supervisor commands are measured in terms of commanded end effector Point of Resolution (POR -  $x, y, z$ , roll, pitch, and yaw) as sensed by the Polhemus tracker and hand shape (angles of the fingers) as measured by the CyberGloves®. A separate HMM exists for each activity (*e.g.* reaching, grasping). As the supervisor initiates a movement, the probability of each particular HMM with respect to the TLA-generated task plan is computed. The SIP outputs the most likely model prediction and the confidence in that model. This prediction and its confidence are pre-cursors to symbolic commands for the robot. The SIP continuously monitors the

supervisor's movements and provides predictions at a frequency of 100Hz. In depth discussion of the SIP can be found in [10].

The second generation SIP module attempts to model manipulator approach trajectories in a manner which is invariant to supervisor speed as well as to object position and orientation. Whereas the former approach models supervisor motion directly, this method takes an object-centric view of the approaching robot arm by converting its hand into the object's local coordinate system. Central to this approach is the idea that there are a limited number of ways the humanoid's arm can approach and interact with an object. As the arm moves towards a target object, the trajectory will eventually fall into an approach manifold. As the distance between the robot's hand and the object decreases, the range of possible POR and finger joint angle values collapses. The approach manifolds are object dependent and further constraints can be added depending on scene construction.

### 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 spatio-temporal memory for Robonaut that exists as a virtual geodesic dome interface linked to a database [11]. 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 geodesic nature of the SES allows for quick and efficient search of the sensory space when retrieving object information.

The SES communicates with the visualization software Enigma [12] to display SES data in the environment of a simulated version of Robonaut, which is commanded by the supervisor. In combining the SES with RoboDisplay, the supervisor, wearing the virtual reality HMD, sees a virtual robot and a virtual world populated by representations of detected real-world data. Essentially, the SES populates a virtual world in which the supervisor can execute a set of tasks.

Fig. 2 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. The HMD displays the left and right eye views to the supervisor.

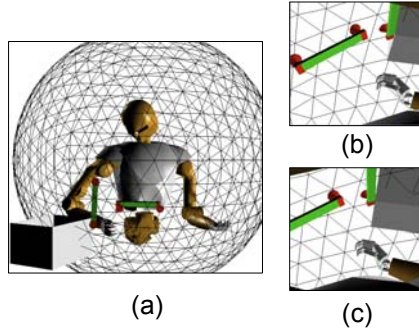


Fig. 2 Simulated robot in SES (a), supervisor’s left eye view (b) and supervisor’s right eye view (c)

#### 4) Cockpit software communication

The SIP continuously communicates with the TLA for determining task-status and the SES for retrieving object information. The SES retrieves confidence information from the SIP for use in the immersive environment. The confidence with which the SIP predicts motions and the targets of those motions can be conveyed to the supervisor visually by associating the SIP confidence with the color or transparency of targets. This feedback lets the supervisor see if actions performed are being correctly predicted. The visual feedback also lets the supervisor move to the next activity without completing 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. The SES can provide the knowledge of nearby objects and whether they can be manipulated in the desired fashion. The hexagonal neighborhoods on the virtual geodesic surface of the SES afford a quick and efficient search for objects in desired locations to complete virtual grasps and releases. Fig. 3 presents the communication flow between software modules in the cockpit and the remote robot.

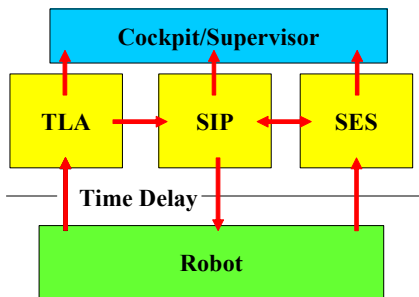


Fig. 3 Cockpit software communication flow

#### C. Robonaut Hardware

Robonaut is an anthropomorphic humanoid robot specifically designed for space. Robonaut has over 40-DOFs with two 7-DOF arms, each ending with a five-fingered hand. Both Robonaut systems, 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 was used for the work described in this paper.

#### D. Robonaut Software

The objective of the software on Robonaut is to provide partial autonomy that is guided by the remote cockpit. 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. Some of the behaviors use the stereo vision system to locate target points while others use the tactile hand sensors to indicate behavior completion [13].

An arbitration module serves as the primary interface between the robot and the cockpit. This module determines which mode of operation takes control of the robot: teleoperation or supervisory guidance. Teleoperation mode is required when attempting to complete a task beyond the autonomous capabilities of the remote robot or for error recovery. The arbitrator receives input from the SIP and the TLA. When a symbolic command from the SIP is received, the arbitrator uses the prediction’s confidence value to determine the quality of the prediction generated. If the arbitrator deems the prediction good (above a predetermined threshold), 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 tasks received. The arbitrator queues the predicted tasks and moves them from the queue when the robot becomes idle. Fig. 6 shows the software communication on the robot side of the time delay.

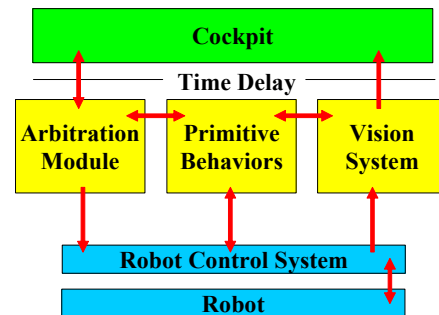


Fig. 6 Robot software communication flow

### III. INITIAL EXPERIMENTS AND RESULTS

Three different experiments were constructed to determine baselines for the robot and teleoperator, to collect data for the SIP and to integrate the robot and the cockpit. A

remote environment setup was designed to mimic possible panel configurations existing on the outside of the ISS or on a lunar surface module. The cockpit environment was a virtual reproduction of all the robot can sense in the remote environment.

#### A. Experimental Setup and Protocol

The taskboard contains two separate EVA handrails in a perpendicular fashion with a push-button below one of the handrails. 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 which object is a task or activity target. Another requirement for this setup was that it be realistic, *i.e.* a similar setup could be found on the ISS or lunar surface, so that activities and tasks tested were applicable to NASA operations. The taskboard and an EVA box for dropping handrails form the experiment hardware setup. The tasks that comprise the experimental protocol are listed below. They can be performed in any order.

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

Figure 7 shows the remote environment setup.



Fig. 7 Remote environment setup

#### B. Delayed 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 both with and without a two-second time delay. In both cases, the teleoperator relied solely on visual feedback to perform the task. Results from these two teleoperation trials and from robot autonomy trials 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

The results of the delayed task reflect the teleoperator’s use of the move-and-wait method, which was completed in almost twice the time of the non-delayed task. The lack of feedback in teleoperation tasks, though, led to similar maximum force values applied by the robot 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 between the robot’s hand and the handrail. In comparison to a teleoperator, the robot’s autonomous grasping takes longer to complete but significantly reduces peak contact loads and the measure of “wear and tear”. These results show that it is preferable to have the robot run autonomously when possible.

#### C. Immersive Environment Grasping

Grasping trials were also run in the immersive environment. The objective of these trials was to collect data to train the SIP models and to test if these models could predict the teleoperator’s intention. Trials were run over three months using the experimental protocol described in A. Multiple supervisors performed the experiments in the fully immersive virtual environment, although the SIP was trained on only one of the supervisors for the initial tests of the system. Natural starting position and the positions of handrails were varied over all trials. During the first sets of trials, data was collected to develop models necessary for the SIP. Once configured, the SIP was incorporated into the system and more trials were run to test the accuracy of the SIP. During these trials, the supervisor received feedback about predictions via variations in colors of the virtual handrails. Data collected on these trials were used to adjust the SIP models for generalization over space and time. Several more trials were run to determine the average time to prediction. Using this technique, grasping motions were predicted more than 5 seconds prior to the hand close event with zero false alarms and no missed detections. However, the system was sensitive to changes in experimental setup and did not generalize well to new situations.

The second generation SIP was then trained and tested using the same setup and supervisor as described above. This approach resulted in slightly better prediction times than the embedded HMMs, is computationally inexpensive, and is significantly more robust to changes in experimental setup. While the system was trained on one operator, multiple operators were able to perform tasks in the virtual environment while having their motions successfully predicted using the second generation SIP. This method performed better than the first generation SIP in all instances.

#### D. Integration Testing

The robot and cockpit components were integrated to test communication and to determine the latencies of the system under various time delays. Three trials per time delay were conducted with round trip time delays of 2, 5, 7 and 10 seconds. The SIP accurately predicted all of the supervisor’s motions, as described in C. above. The SIP predictions were successfully and immediately sent across the time delay as symbolic commands and executed by the robot. For

integration tests, all systems were operated nominally, using an experienced supervisor, handrails in nominal positions, and no deviations from the plan, either intentional or unintentional.

The results of the integration 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. Also as expected, 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.

The metric selected to analyze the performance of the system is task completion time per time delay (unidirectional),  $T_c/T_d$ . For the most complex aspect of the task (reach and grasp handrail), this ranged between 5 and 25, for a 5 second and 1 second unidirectional delay respectively. These high values for the  $T_c/T_d$  metric illustrate that the intermediate time delay does not affect the performance of the robot in task completion. This highlights the utility of this approach to controlling robots across time delay, *i.e.* the time delay does not hamper the robot's ability to complete tasks in a timely manner. Another benefit of this approach is that the supervisor can work at a natural human pace without affecting the robot's task completion. 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

This paper presents an initial effort for supervising remote dexterous humanoids over intermediate time delays. The need for remote supervision arises from the limited number of astronauts available to teleoperate robots in space and from the limited autonomy currently available on dexterous humanoids. Remote operation of a robot either on the ISS or on the lunar surface will produce anywhere from a 2 to 10 second time delay. The intermediate time delay combined with the dexterity of humanoids like Robonaut create a unique situation in teleoperation. This solution involved the combination of robot autonomy with human supervision.

A smart cockpit was developed at JSC to house the essential personnel and equipment to supervise the robot. The cockpit 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 motions so that tasks may be guided rather than directly commanded 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 VR equipment in the cockpit. Robonaut, the dexterous robot, uses autonomous behaviors 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.

The work presented in this paper shows the initial results of the system using a simple experimental setup that does not allow for deviations from the task plan. Future work will involve creating an experimental setup that is more complex, requires more autonomous actions, and has the possibility for multiple deviations from the task plan. The robustness of the system will be measured by using different supervisors as opposed to testing the system on a single supervisor. Since the supervisor could only function in an immersive state, the next experiments will allow the supervisor to function without the immersive gear. The supervisor will be given a display-based interface for interaction with the robot. Finally, the approach will be tested controlling a system consisting of Robonaut Unit B mounted on a 4-wheeled mobile system.

#### REFERENCES

- [1] "The Vision for Space Exploration," National Aeronautics and Space Administration, [www.nasa.gov/pdf/55583main\\_vision\\_space\\_exploration2.pdf](http://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](http://exploration.nasa.gov/documents/ESMD-RQ-0014_Rev_A_RLEP.pdf), February 2005.
- [3] Anderson, R. , M. Spong, "Bilateral control of tele-operators with time delay," *IEEE Transactions on Automatic Control*, 34(5), pp 494-501, 1989.
- [4] Niemeyer, G. , Slotine, J. , "Stable adaptive teleoperation," *IEEE Journal of Oceanic Engineering*, 16(1), pp. 152-162, 1991.
- [5] Bejczy, A.K., W.S. Kim, "Predictive Displays and Shared Compliance Control for Time-Delayed Telemanipulation," in *Proceedings of the 1990 IEEE IROS*, pp. 407-12.
- [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] 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.
- [8] Kheddar, A., Tzafestas, C., P. Coiffet, T. Kotoku, S. Kawabata, K. Iwamoto, K. Tanie, I. Mazon, C. Laugier, R. Chellali, "Parallel Multi-Robots Long Distance Teleoperation," in *Proceedings of the International Conference on Advanced Robotics 1997*, Monterey, CA, July 7-9, 1997.
- [9] Paul, R., T. Lindsay, C. Sayers, "Time Delay Insensitive Teleoperation," in *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC July 7-10, 1992.
- [10] 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.
- [11] 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)*, Tokyo, Japan, November 22-24, 2001.
- [12] 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.