The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Using Prediction to Enhance Remote Robot Supervision across Time Delay

Robert R. Burridge and Kimberly A. Hambuchen

Abstract— **The Predictive Interactive Graphical Interface (PIGI) is a suite of tools developed at NASA's Johnson Space Center (JSC) for supervising robots across expected Earth-moon time delays (5-10 second round trip). These tools improve interaction between a human supervisor and a remote robot by mitigating the effects of the time delay. Using a combination of robot behavior prediction and task queuing, PIGI enables the supervisor to reduce robot idle time, which leads to more efficient completion of the tasks. PIGI was used in 2007 and 2008 to remotely command five different NASA robots in Arizona, California, Texas, and Washington, all from a single location at JSC in Houston.**

I. INTRODUCTION

NASA'S current plans for space exploration call for humans to return to the moon by 2020 and establish a permanent base for further exploration of the solar system [1]. These plans rely heavily on robotic systems for reconnoitering landing areas and preparing and building the lunar base. Robotic tasks will be required both prior to and after human arrival. Although many of the robotic tasks will be automated, it is assumed by the space community that they will need human supervision, and that many tasks will require active operator participation.

The current plan is to land many assets on the moon prior to human arrival. These assets will include robots capable of exploration, payload transportation, site preparation, and maintenance. Many robots will perform tasks involving unstructured or under-constrained activity beyond the current capabilities of autonomous systems. Thus, it will be necessary for humans on Earth to supervise the activity closely. Later, when humans are present, it will still be beneficial to have Earth-based humans supervising robotic assets for routine operations so that lunar-based humans can perform more complex tasks with the assistance of these robots.

This paper presents a novel approach for overcoming time delays expected between Earth and the moon. This involves keeping the human in the loop as much as possible while taking advantage of short-term robot autonomy. The approach includes robot behavioral models that enhance a supervisor's situational awareness, and queuing capabilities that allow the supervisor to execute multiple tasks, enabling the robot to always have a task "on deck".

II. BACKGROUND

Many approaches exist for controlling robots in the presence of time delay. The magnitude of the time delay is the largest factor in how these control strategies are applied to remote robots. Short time delays (< 2 sec.) enable manual teleoperation techniques, including bilateral control methods to stabilize motion [2]. Long time delays (> 10 sec.) usually require that robots have some autonomous abilities. The Mars rovers, Spirit and Opportunity, perform their tasks autonomously, since the 24-90 minute round trip communication latency is too great for real-time human-in-the-loop interaction [3][4].

Intermediate time delays (2-10 sec.) provide a unique opportunity. The communication latency is short enough that real-time human interaction can occur, yet it is long enough to require that the remote robot must have some autonomous capabilities. While some have taken the bilateral control approach for time delays in this range [5], a supervisory control strategy may be more suitable in this situation [6]. Under supervisory control, a human operator normally sends symbolic commands to the remote robot, but may intervene with manual commands if desired [7]. These analogic commands can be achieved by using one of the many bilateral control methods or by using the "bump and wait" approach. When good models of the remote robot's behavior are available, prediction methods can also be used in supervisory control to give the human operator a better understanding of what may be occurring during latency periods [8].

Since the robots involved in this work each have differing degrees of autonomy, a supervisory approach to control over time delay was chosen. The majority of control strategies for operating remote robots seek to stabilize the robot's behavior, while the main goal of the approach presented here is to mitigate the time delay to reduce robot idle time, increasing the utility of the robot. Thus, a prediction scheme approach inspired by approaches used in [9] for manipulators, in [10] for submersibles, and in [11] for mobile robots is employed to lessen the time needed for a robot to complete its mission. However, it should be noted that the approach presented in this paper differs from the above references in that it uses prediction for supervisory

Manuscript received February 27, 2009. This work was supported by NASA's Exploration Technologies Development Program, Human-Robot Systems Project.

R. R. Burridge is with TRACLabs, Inc., Houston, TX 77058 USA (phone 281-461-7884; e-mail: Robert.r.burridge@nasa.gov).

K. A. Hambuchen is with the National Aeronautics and Space Administration, Houston, TX 77058 USA. (e-mail: Kimberly.a.hambuchen@nasa.gov).

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

control based on robot behavior models and task queueing to accomplish the mitigation of the time delay.

III. ROBOTS

This section briefly describes the robots that have been controlled using PIGI.

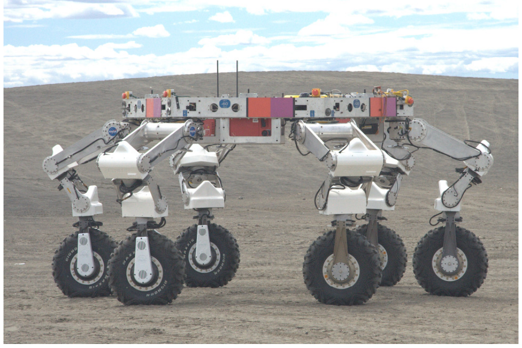

Figure 1. ATHLETE

A. *ATHLETE*

The All-Terrain Hex-Legged Extra-Terrestrial Explorer (ATHLETE, see figure 1) developed by the Jet Propulsion Laboratory (JPL) has six legs, each with seven degrees of freedom and a wheel at the end. It is capable of rolling over relatively flat terrain and stepping through rough or steep terrain [12].

Figure 2. Centaur

B. *Centaur*

Centaur (see figure 2) developed by the Johnson Space Center (JSC) combines Robonaut's upper body with a four-wheeled mobile base [13]. Robonaut is a humanoid robot with roughly the same size and dexterity as a suited astronaut. Centaur is an experimental platform for studying both teleoperation and autonomy for mobile dexterous manipulation.

Figure 3. Chariot

C. *Chariot*

Chariot (see figure 3) was developed by JSC as part of the Exploration Technology Development Program (ETDP), Human-Robotic Systems (HRS) project as a "lunar truck" prototype vehicle. Chariot has six mobility modules, each with independent steering, active suspension, hi/low-speed transmission, and a pair of wheels. In the unpressurized option shown above, it can support two space-suited astronaut drivers and two more passengers in an emergency, or it can be operated with a pressurized crew module on the vehicle. In addition, with the right attachments, the Chariot will be capable of serving a large number of functions on the lunar surface, including serving as a regolith mover and as a carrier to deploy power systems and habitats to the desired location on the lunar base. The "crab drive" mobility system allows the direction the chassis is facing to be completely decoupled from the direction of vehicle motion. The suspension can lower the chassis to the ground and raise it to about 35 cm clearance [14].

Figure 4. K-10

D. *K-10*

K-10 (see figure 4) was designed by the Ames Research Center (ARC) as an astronaut assistant for site survey and inspection operations. It features a 4-wheel steer, 4-wheel drive rocker chassis and a footprint of roughly one meter squared. Top driving speed for K-10 is approximately 3.2 km/hr (roughly walking speed). The avionics box and mast can hold payloads of up to 30 kg, and the frame has been designed to accommodate a wide variety of science instruments [15].

Figure 5. SCOUT

E. SCOUT

The Science Crew Operations and Utility Testbed (SCOUT) was developed by JSC and is shown in figure 5. The SCOUT project focused on the development and testing of advanced rover technologies and operation concepts related to transportation of suited astronauts. SCOUT can carry two suited crew members, traverse 15 degree slopes, and drive at speeds up to 15 km/hr [16].

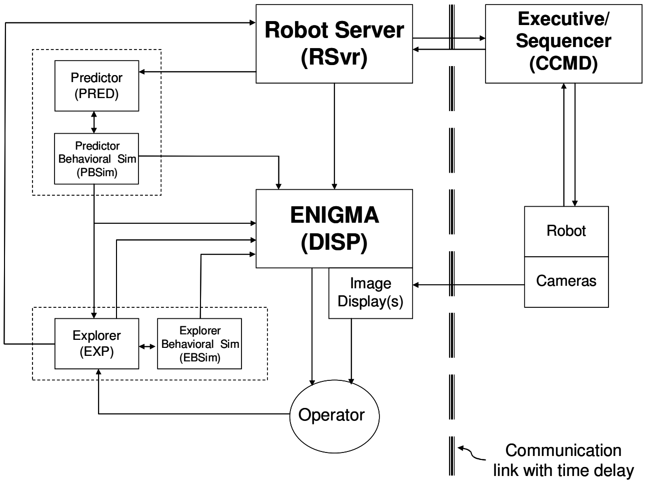

Figure 6. Components of PIGI. The dashed line represents the separation (with time delay) between the Cockpit (on the left) and the Robot (on the right).

IV. PIGI

The foundation of the Predictive Interactive Graphical Interface (PIGI) is its ability to manage and utilize task queues. Figure 6 provides a block diagram of the connections between the major components of PIGI. The "Explorer" (EXP) is the command interface for the human supervisor. The supervisor uses EXP to send commands to the robot. An application called the "Robot Server" (RSvr) keeps a record of each command sent to the robot, combining it with the status of previous commands reported by telemetry from the robot. This information goes to the "Predictor" (PRED), which uses a low-fidelity "robot behavior simulation" (PBSim) to model the response of the robot to all uncompleted commands. The predicted end state returns to EXP, which uses another robot behavior simulation (EBSim) to allow the supervisor to investigate additional commands. The latest robot state reported by telemetry, along with results from PBSim, EBSim, and EXP are all displayed in a 3D graphical display (DISP).

Task queues and the applications that use them are described in the following sections.

A. Task Queues

Each robot command issued from PIGI is considered to be a discrete task object containing a unique ID, command type, and parameters. This differs from traditional teleoperation, where a continuous stream of velocities or positions is sent to the robot. Command streams do, of course, at some level comprise atomic commands, but the difference in paradigm is similar to that between streaming video and sending individual images. The queues are first-in-first-out (FIFO), meaning that tasks are added to the tail of the queue and removed from the head after they are executed.

Tasks are associated with three different queues on-board the robot (PENDING, ACTIVE, and COMPLETED), depending on their completion status. When the robot receives a task, it is placed at the tail of PENDING. As soon as resources become available, the task at the head of PENDING shifts to ACTIVE, and the robot starts to perform the task. When it finishes, the task shifts to COMPLETED (for success or failure). Robot telemetry contains the queue status and result of all tasks, along with the current state of the robot. Note that for some tasks, especially relative motions, the state of the robot when the task was initiated must be reported with the telemetry in order for the prediction to work correctly. An ACTIVE task of "Move ten meters ahead" would be impossible to model without knowing where the motion started.

B. Robot Server

The component of PIGI that manages the flow of messages between the supervisor and the robot is the "Robot Server" (RSvr). This application keeps its own version of PENDING, ACTIVE, and COMPLETED. Although these reflect the most recent telemetry, they necessarily lag behind those on-board the robot by the communication delay of the system. In addition, RSvr maintains a fourth queue, SENT, containing a record of all outbound tasks received from the Explorer (EXP, Section IV.D) that have not yet been identified with an on-board queue in robot telemetry. All outbound tasks will stay in SENT for at least the round-trip time delay before being updated to another queue by telemetry.

RSvr sends the current robot state to the Display (DISP, Section IV.E), and combines the current robot state with all tasks currently on SENT, PENDING, and ACTIVE into a message that goes to the Predictor (PRED, Section IV.C).

RSvr may also be configured to produce a status message with the robot state and all four queues, including information about the outcome for each COMPLETED task. This message type was used in experiments with decisionsupport software tools assisting the supervisor with longer-term mission planning [17].

C. Prediction

The purpose of the Predictor (PRED) is to inform the supervisor of the anticipated activity of the robot from its most recently reported state to the completion of the final command on the SENT queue. Using messages from the RSvr, the Predictor produces the robot's expected path, represented by a series of tightly spaced points, and its expected final state. The final state is sent to the Explorer (EXP, Section IV.D), where it is used as an initial condition for modeling the robot's response to additional task commands. Both path and final state are sent to the Display (DISP, Section IV.E).

PRED makes use of a "Robot Behavior Simulation" (BSim), which models the response of the robot to a sequence of commands, given an initial state. The BSim is essentially a state machine, predicting the outcome much faster than real-time, and re-computing the results every time new telemetry arrives from the robot. In particular, the expected paths for the driving tasks of the initial experiments are derived analytically, without the need for integration. In the case of Chariot, BSim uses the same code to produce the path as the on-board navigation system. PRED and EXP use separate instances of BSim, but the underlying application is the same. These tools will be referred to as PBSim and EBSim, when the distinction is needed. Currently, BSim only models drive commands.

D. Exploration

The Explorer (EXP) enables the supervisor to observe the expected outcomes of various possible new commands before selecting one to send to the robot. For drive commands, the supervisor manipulates a "destination" target icon in the Display, using either a joystick or numeric values. The target icon represents the desired destination for the next drive command. When the supervisor has selected a target location, a drive command is sent to EBSim, along with the final state from PRED (to provide the initial state for the new drive). EBSim predicts the likely path and final state, which are sent to DISP. If the supervisor is satisfied with the result, the command is accepted and sent to RSvr, which places it on SENT and forwards it to the robot. If the supervisor is not satisfied, the command is retracted and the target icon is repositioned.

E. Visualization

The primary display for visualization (DISP) in PIGI uses Enigma [18], a 3D display environment used at JSC for simulation and animation. The display shows several things: (1) The surface upon which the robots drive, (2) A model of the robot state according to the latest telemetry from RSvr, (3) A trail of bread crumbs showing the predicted path for committed commands, from PRED, (4) A token representing the final state of the robot after the completion of current commands, from PRED, (5) A trail of bread crumbs showing the predicted path for potential commands, from EXP, and (6) A token representing the desired target location, from EXP.

Figure 7. PIGI display for Centaur showing paths paths from the Predictor (shaded) and the Explorer (white). The small ball at the end of the path from PRED marks the location of the final expected state for all committed drive tasks. EXP paths start from there.

Figure 7 shows a snapshot from DISP for a Centaur drive (on a featureless plane). Since Centaur has Ackermann steering, the path it will select is not always obvious. In the figure, Centaur is allowed to use reverse driving to make the drives more efficient. As a result, the path for the first drive command (from PRED) has two segments: arcing forward slightly to the right (lighter), then backward to the left (darker). At that location, the final state from PRED is displayed as a small ball. The second drive (from EXP) has three segments, arriving finally at the middle of the image, facing toward the top. Additional monitors display imagery from cameras in the field and on the robots.

V. EXPERIMENTS

PIGI was used in JSC's Automation, Robotics, and Simulations Division (ARSD) Cockpit to control the five robots described in Section III across time delay. The JSC ARSD Cockpit, shown in figure 8, was initially developed to support supervisory control of dexterous robots over intermediate time delay [19]. The Cockpit has been extended to supervise control of mobile robots.

Figure 8. ARSD Cockpit at NASA JSC

A. Commands

The primary task command types used for these experiments were relative and absolute "drive-to" commands. These contain a goal location of (X, Y, Angle), either in the robot-centric frame (relative) or a local site coordinate frame (absolute). Other types of command were sent from PIGI during these experiments but, since DISP mainly shows changes in location, driving is currently the only task modeled in BSim. For instance, no observable change results from a "Capture an Image" command. Nonetheless, these other tasks go on the queues in the same manner as drives, and the functionality of RSvr allows the supervisor to mitigate the time delay.

B. Central Command and RAPID Sequencer

Since most of these robots do not manage task queues in the manner needed by PIGI, an additional task-sequencing executive was created for some of them.

At JSC, an on-board executive called "Central Commander" (CCMD) has been used since 2005 to manage communication between off-board supervisors and the robot's control system. This was used for Robonaut in the Peer-to-Peer experiments [20], for Centaur in the Coordinated Field Experiment [13], and most recently for Chariot at the Moses Lake demonstration [21].

Among other duties, CCMD manages the three on-board task queues, interacts with various on-board subsystems to execute the ACTIVE tasks, and reports robot subsystem and queue status back to the supervisor.

Three of the robots used in these experiments (ATHLETE, K-10, and SCOUT) do not use CCMD as part of their robot control architecture. To overcome this for the PIGI experiments in 2007, it was necessary to develop a CCMD for each of them.

In 2008, a standard interface protocol was developed called "RAPID" [22]. A "RAPID bridge" was developed for each robot that translated the native API of the robot into RAPID. For the PIGI experiments of 2008, a "RAPID Sequencer" was developed that provided the queuing functionality of CCMD and communicated with cockpit and robot using RAPID. In the Cockpit, another "RAPID bridge" converted RAPID into the standard Cockpit interface protocol.

C. Time Delay

The intention of this project is not to develop new transport layers for communication in the presence of time delay. Thus, it was desired to use the simplest possible method that would cause the system to behave as though it had a ten-second round trip time delay, without diving into the realm of delay on actual Ethernet transport layers. Since RSvr is the single source for outbound traffic from the Cockpit to the robots, outbound commands were held for the full ten seconds and telemetry and images were sent back from the robot without delay.

In 2007, a separate application was used to intercept command messages, hold them for a set amount of time, and then send them on to the robot. The surrogate CCMD applications for K-10 and SCOUT ran in the Cockpit, but on the remote side of the time delay. Thus, they behaved as part of the robot, even though they ran on Cockpit computers. The CCMD applications communicated with those robots using their robot-native CORBA-based protocols. (This would not have worked well with an intrinsic delay on the link, because parts of CORBA are TCP-based, which is sensitive to latency on the network.)

In 2008, RSvr was enhanced to produce the effect of time delay by holding outbound messages internally. The RAPID Sequencer and bridges ran on the remote side of the time delay with K-10 and ATHLETE. Once again, since RAPID is CORBA-based, this would not have been practical if the delay had been on the main link.

In all testing described below, a 10 second round-trip delay was inserted between the remote and cockpit sites. This duration was chosen as a worst-case scenario for the communication latency between ground and lunar operations.

D. FY 2007 Testing

In 2007, the components of PIGI described in Section IV were developed as part of an inter-center project to investigate the issues of time delay and to develop a common workbench for robot control. Centaur was chosen as the platform for initial development of PIGI.

Initially, PIGI was designed to send relative drive commands to Centaur. Drives succeed when the robot is within some deadband tolerance of the goal in position and orientation, leading to uncertainty in the precise final state. Unfortunately, this final state becomes the initial state for BSim to predict the results of the next drive. The uncertainty compounds rapidly, because small adjustments in the initial heading produce large changes in the resulting path, and it was found that multiple commands could not be queued due to extreme uncertainty in final state. For Centaur, this problem was solved by switching to absolute drive commands. Unfortunately, K-10 only accepted relative commands for this project, and SCOUT's navigation system did not try to achieve the commanded heading at all – just the position.

1) Centaur

Numerous development runs were conducted using Centaur during the spring and summer of 2007, culminating in a demonstration for JSC and Constellation Program management in September 2007. All runs were on-site at JSC, either indoors or outside on paved surfaces. Plan authoring and monitoring tools developed to assist the robot supervisor in keeping track of more complex missions were also tested [17]. The robot was rarely idle between segments of the drive.

2) K-10

PIGI was used to command K-10 in the Mars Yard at ARC in June 2007. A pre-defined set of points was shown in DISP, and the supervisor needed to navigate the robot to each point in turn while making sure the robot did not enter any forbidden zones. The mission was completed successfully. However, because relative drive commands were used for K-10, the supervisor found it necessary to limit the queue to one command and drop back to "bump-and-wait". Thus, the robot was often idle for the full 10 seconds of time delay between segments of the drive.

3) SCOUT

In September 2007, SCOUT participated in field tests at Cinder Lake, AZ, as part of the Desert Research and Technology Studies (D-RATS) with JSC's Advanced Spacesuit group. The experimental scenario was for the robot to drive to nine pre-defined waypoints (given by their GPS coordinates) and conduct a battery of observations at each point, including communications quality and capturing a panoramic image. This scenario was conducted by on-board drivers in spacesuits, off-board teleoperators without time delay, and from the ARSD Cockpit at JSC with time delay.

This scenario was well-suited to the strengths of PIGI, and the supervisor was able to keep tasks on PENDING at all times while still making near-term driving decisions based on the terrain visible in the camera images. When driving between waypoints, the supervisor was able to comfortably see and command drives more than 10 seconds ahead and queue the science activities after the final drive command was sent – the one that reached the observation point. During the science operations (which lasted more than 10 seconds), the supervisor was able to queue up the first leg of the drive to the next waypoint based on images captured at the waypoint. Thus the robot had no idle time due to time delay during these runs and was able to drive without stopping to the observation points.

E. FY 2008 Testing

In 2008, work on PIGI continued as part of the ETDP HRS lunar remote robotic operations task. The core components of PIGI continued to be refined, the communications protocol inside the Cockpit was updated, and a BSim and CCMD were developed for Chariot. The year culminated in a two-week field test in Moses Lake, Washington, that included Chariot, ATHLETE, and K-10. During this field test, PIGI was used extensively to drive Chariot, and was demonstrated with ATHLETE and K-10. In addition to the PIGI runs, the Moses Lake field tests included several other robots and many experiments and demonstrations that did not involve remote operations.

1) Single-Robot Operations

During the spring of 2008, PIGI was used to drive the Chariot in JSC's rock yard, ATHLETE in JPL's mars yard, and K-10 in ARC's mars yard. In addition, short driving excursions were conducted with ATHLETE and K-10 when the robots were all at Moses Lake. Chariot was driven extensively using PIGI at Moses Lake, both for long periods of time and distance and for short traverses interspersed with other activities. In particular, a 1.6km traverse was undertaken to determine how well PIGI could mitigate the time delay. Operating under a 10 second delay, PIGI was used in its full capacity to drive Chariot for the first 0.8km. The time to complete this part of the traverse was 45 minutes. The next 0.8km were driven using a bump-and-wait method, largely due to inconsistencies between the map used in PIGI and the actual terrain at Moses Lake. The time to complete this part of the traverse was approximately 75 minutes. This was our initial indicator that PIGI can drastically reduce the time needed to supervise a robot over time delay.

2) Dual-Robot Operations

In Moses Lake, two command and control paradigms for operating Chariot and K-10 simultaneously were tested.

In the first experiment, PIGI was used to control both K-10 and Chariot. K-10 was mounted on Chariot, representing an experimental sensor package. In the Cockpit, two complete instances of PIGI ran simultaneously – one for Chariot and one for K-10. The task was to drive to a pre-defined location and capture a high-resolution LIDAR panorama. The supervisor sent only drive commands to Chariot, queuing them as usual to get to the destination. During the high-resolution panorama, Chariot was required to take twelve small rotational steps about the center of K-10, allowing K-10 to capture an image at each orientation. Thus, alternate commands were sent to the two robots. When each command showed up in the COMPLETED queue for that robot's RSvr, the next command was given to the other robot, and so on.

Because there was no coordination of the robots on the far side of the time delay, PIGI's queuing capability could not be used for this activity. When one robot completed its current task, that information had to reach the supervisor, and the supervisor's next command had to reach the other robot before the next action could occur. This led to an unavoidable ten-second idle period between each command.

In the second experiment, K-10 was mounted on Chariot, riding along while suited astronauts drove Chariot to a site of geological interest. Once at that location, the astronauts handed off control of Chariot to the Cockpit at JSC. Via PIGI, a sequence of commands was sent to Chariot to lower the suspension and deploy the ramps for K-10. When that sequence was complete, control authority was handed off to the K-10 science team, who then drove K-10 off Chariot and preceded with independent science exploration using K-10. Control of Chariot was passed back to the astronauts, who drove Chariot to their next site.

VI. CONCLUSIONS AND FUTURE WORK

This paper has presented PIGI – a control method used for supervision of mobile robots over intermediate time delays. PIGI has shown that it allows a human to supervise multiple types of robots, each running in its native software architecture. The queuing functionality of PIGI provides the robot supervisor an enhanced situational awareness over other approaches, whether predictive models of the robots exist or not. During driving maneuvers, PIGI can mitigate communication latencies for the robot under supervision. Multiple-robot testing has shown that PIGI can increase the coordination of tasks between robots.

Testing done with SCOUT during the FY 2007 field test demonstrated the advantage of the queuing process. Interleaving non-driving and driving tasks reduced the idle time of the robot and thus improved the efficiency of the overall mission. Testing done with Chariot during the FY 2008 field tests illustrated the advantage of the prediction aspect of PIGI. When the predicted path of the robot changed due to interaction with the environment, this showed up immediately in the display and the supervisor was able to modify the tasks in the queue.

For relative-drive robots, the task-sequencing executive must store commands as absolute, converting them to relative commands at the time they become ACTIVE.

For future versions, PIGI should allow sequencing of task queues to go through a single RSvr for all robots working together. It was also realized through this testing that sequencing of commands must occur on both the robot side of the time delay and on the supervisor's side.

Many options exist for extending the current system. These include enhancing the Display, enhancing the supervisor's ability to command the robot, and adding more applications to assist the supervisor and improve situational awareness in various ways.

ACKNOWLEDGMENT

The authors would also like to thank David Mittman and Jay Torres from NASA JPL, Mark Allan and Hans Utz from NASA ARC and Ryan Necessary from JSC.

REFERENCES

[1] The Vision for Space Exploration," National Aeronautics and Space Administration, http://www.nasa.gov/pdf/60736main_M2M_report_small.pdf, 2003.

[2] Hokayem, P.F., Spong, M.W.. Bilateral teleoperation: An historical survey, AutomaticaVolume 42, Issue 12, December 2006, Pages 2035-2057.

[3] Maimone, M., A. Johnson, Y. Cheng, R. Willson, and L. Matthies, Autonomous navigation results from the Mars Exploration Rover (MER) mission, *Proceedings, International Symposium on Experimental Robotics (ISER)*, Singapore. June 2004.

[4] Biesiadecki, J.J., P.C. Leger, M.W. Maimone. Tradeoffs between Directed and Autonomous Driving on the Mars Exploration Rovers. International Journal of Robotics Research, Vol. 26, Issue 1, January 2007, pp 91-104.

[5] Imaida, T., Y. Yokokohji, T. Doi, M. Oda, T. Yoshikawa. 2004. Ground-Space Bilateral Teleoperation of ETS-VII Robot Arm by Direct Bilateral Coupling Under 7-s Time Delay Condition. IEEE Transactions on Robotics and Automation. Vol 20, No. 3, June 2004.

[6] Ferrell, W.R., T.B. Sheridan. Supervisory control of remote manipulation. IEEE Spectrum 4, no. 10, October 1967. pp 81-88.

[7] Sheridan, T.B. Human supervisory control of robot systems. In Proceedings of IEEE Conference on Robotics and Automation, Vol. 3, pp 808-812, April 1986.

[8] Sheridan, T. B. Telerobotics, automation, and human supervisory control. MIT Press, Cambridge, MA. 1992.

[9] Xiong, Y., S. Li, M. Xie. 2006. Predictive display and interaction of telerobots based on augmented reality. Robotica, Vol. 24, Issue 4, pp 447-453, July 2006.

[10] Cheng, C.C. Predictor displays: theory, development and application to towed submersibles. Sc.D. thesis, MIT, Cambridge, MA. June 1991.

[11] Cobzas, D., M. Jagersand, H. Zhang. 2005. A Panoramic Model for Remote Robot Environment Mapping and Predictive Display. International Journal of Robotics and Automation, Vol. 20, No. 1, pp 25-34.

[12] Wilcox, B. H., Litwin, T., Biesiadecki, J., Matthews, J., Heverly, M., Morrison, J., Townsend, J., Ahmad, N., Sirota, A., and Cooper, B. 2007. ATHLETE: A Cargo Handling and Manipulation Robot for the Moon. *Journal of Field Robotics* [online journal], Vol. 24, No. 5, URL: http://www3.interscience.wiley.com/journal/114211098/issue, Wiley Periodicals, May 2007, pp. 421–434.

[13] Mehling, J.S., P. Strawser, L. Bridgwater, W.K. Verdeyen, R. Rovekamp. Centaur: NASA's Mobile Humanoid Designed for Field Work. In Proceedings of 2007 IEEE International Conference on Robotics and Automation, pp 2928-2933, Rome, Italy, April 2007.

[14] Harrison, D., Ambrose, R., Bluethmann, W., Junkin, L. 2008. Next Generation Rover for Lunar Exploration. *Proceedings, IEEE Aerospace Conference*, Big Sky, MT, Mar. 2008, pp. 1-14.

[15] Fong, T. W., Allan, M. B., Bouyssounouse, X., Bualat, M. G., Deans, M. C., Edwards, L., Flückiger, L., Keely, L., Lee, S. Y., Lees, D., To, V., and Utz, H. 2008. Robotic Site Survey at Haughton Crater, *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Jet Propulsion Laboratory, Pasadena, CA, Feb. 2008.

[16] Rochlis, J., F. Delgado, J. Graham. Science crew operations and utility testbed. Industrial Robot: An International Journal, 33(6), pp 443-450, 2006.

[17] Debra Schreckenghost, Tam Ngo, Robert Burridge, Lui Wang, Michel Izygon, 2008. Remote Task-Level Commanding of Centaur over Time Delay. *Proceedings, Space Technologies and Applications International Forum*, Albuquerque, NM, Feb. 2008.

[18] Enigma, Software Package, Version 13.1, NASA/Johnson Space Center, Integrated Graphics Operations and Analysis Laboratory, Contact: Sharon.P.Goza@nasa.gov, (281) 483-8451 or David.L.Shores@nasa.gov, (281) 483-1586.

[19] Hambuchen, K.A., W. Bluethmann, S.M. Goza, R. Ambrose, K. Rabe, M. Allan. Supervising Remote Humanoids Across Intermediate Time Delay. Proceedings of IEEE-RAS International Conference on Humanoid Robotics, December 2006, pp 245-251.

[20] T.W. Fong, J. Sholtz, J. Shah, L. Flueckiger, C. Kunz, D. Lees, J. Schreiner, M. Siegel, L. Hiatt, I. Nourbakhsh, R. Simmons, R. Ambrose, R. Burridge, B. Antonishek, M. Bugajska, A. Shultz, and J.G. Trafton. 2006. A Preliminary Study of Peer-to-Peer Human-Robot Interaction. In *Proceedings, IEEE International Conference on Systems, Man, and Cybernetics*, October 2006.

[21] Chariot: http://chariot.jsc.nasa.gov

[22] D. Mittman, J. Norris, R. Torres, K. Hambuchen, R. Hirsh, M. Allen, H. Utz, R. Burridge. 2008. The Exploration Technology Development Program Multi-Center Cockpit. In *Proceedings, AIAA SPACE 2008 Conference & Exposition,* San Diego, CA, Sep. 2008.