

FORMING HUMAN-ROBOT TEAMS ACROSS TIME AND SPACE

Kimberly A. Hambuchen*

NASA Johnson Space Center, U.S.A., kimberly.a.hambuchen@nasa.gov

Robert R. Burrige**, robert.r.burrige@nasa.gov, Robert O. Ambrose*, robert.o.ambrose@nasa.gov,
William J. Bluethmann*, william.j.bluethmann@nasa.gov, Myron A. Diftler*, myron.a.diftler@nasa.gov,
Nicolaus A. Radford*, nicolaus.a.radford@nasa.gov

NASA pushes telerobotics to distances that span the Solar System. At this scale, time of flight for communication is limited by the speed of light, inducing long time delays, narrow bandwidth and the real risk of data disruption. NASA also supports missions where humans are in direct contact with robots during extravehicular activity (EVA), giving a range of zero to hundreds of millions of miles for NASA's definition of "tele". Another temporal variable is mission phasing. NASA missions are now being considered that combine early robotic phases with later human arrival, then transition back to robot only operations. Robots can preposition, scout, sample or construct in advance of human teammates, transition to assistant roles when the crew are present, and then become caretakers when the crew returns to Earth.

This paper will describe advances in robot safety and command interaction approaches developed to form effective human-robot teams, overcoming challenges of time delay and adapting as the team transitions from robot only to robots and crew. The work is predicated on the idea that when robots are alone in space, they are still part of a human-robot team acting as surrogates for people back on Earth or in other distant locations. Software, interaction modes and control methods will be described that can operate robots in all these conditions. A novel control mode for operating robots across time delay was developed using a graphical simulation on the human side of the communication, allowing a remote supervisor to drive and command a robot in simulation with no time delay, then monitor progress of the actual robot as data returns from the round trip to and from the robot. Since the robot must be responsible for safety out to at least the round trip time period, the authors developed a multi layer safety system able to detect and protect the robot and people in its workspace. This safety system is also running when humans are in direct contact with the robot, so it involves both internal fault detection as well as force sensing for unintended external contacts.

The designs for the supervisory command mode and the redundant safety system will be described. Specific implementations were developed and test results will be reported. Experiments were conducted using terrestrial analogs for deep space missions, where time delays were artificially added to emulate the longer distances found in space.

I. INTRODUCTION

Telerobotics at NASA will eventually encompass distances that span the Solar System. At this scale, time of flight for communication is limited by the speed of light, inducing long time delays, narrow bandwidth and the real risk of data disruption. NASA also supports missions where humans are in direct contact with robots during extravehicular activity (EVA), giving a range of zero to hundreds of millions of miles for NASA's definition of "tele". NASA missions are now being considered that combine pre-cursor robotic phases with later human arrival, which then transition back to robot only operations. Robots can preposition, scout, sample and

construct in advance of human teammates, transition to assistant roles when the crew are present and become caretakers when the crew returns to Earth.

This paper will describe advances in robot safety and command interaction approaches developed to form effective human-robot teams and overcoming challenges of time delay. 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".

* NASA Johnson Space Center, U.S.A.

** TRACLabs, Inc., U.S.A.

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 [1]. Long time delays (> 10 sec.) usually require that robots have full autonomy[2].

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. Initial activities using Robonaut 1 were implemented over short time delays. Because of the dexterous nature of this robot, a unique approach was taken for handling the delay under supervisory control. The approach centered on operator intent prediction [3]. Teleoperation under telepresence was a major mode of control for Robonaut 1. This led to virtual teleoperation of Robonaut 1. The intent of the teleoperator was then devised from a combination of arm motion, hand position and object proximity. This approach would not translate into control of a mobile platform, however.

While some have taken the bilateral control approach for intermediate time delays [4], the team decided that a supervisory control strategy is preferable for human-assistive telerobotics [5]. Under supervisory control, a human operator normally sends symbolic commands to the remote robot, but may intervene with manual commands if desired [6, 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 those referenced in that it uses prediction for supervisory control based on robot behavior models and task queuing to accomplish the mitigation of the time delay.

III. PIGI

The advanced command interaction approach developed for remotely supervising robots is the Predictive Interactive Graphical Interface (PIGI). At PIGI’s core is its ability to manage and utilize command queues. The PIGI paradigm also provides quick supervisor interaction and able prediction of supervised robots’ behavior. Figure 1 shows a block diagram of the connections between major components of PIGI.

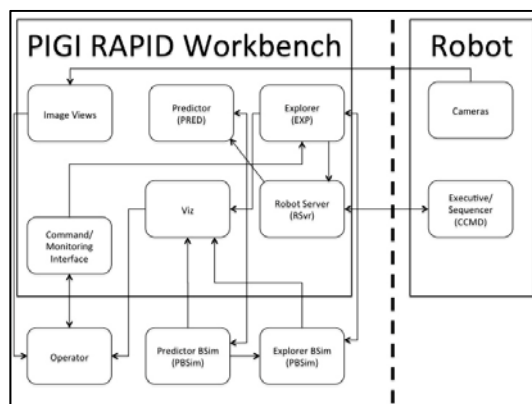


Fig. 1. PIGI Components. The dashed line separates the robot and the supervisor on either side of the time delay.

A command interface provides the supervisor telemetry displays and a door into PIGI. The “Explorer” (EXP) receives commands from the user via the command/telemetry interface. 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 (Viz).

Command Queues

A unique object that contains a command's information represents each robot command issued from PIGI. This differs from traditional teleoperation, where a continuous stream of velocities or positions is sent to the robot. The commands are queued first-in-first-out (FIFO), meaning that each command is added to the tail of the queue and removed from the head after execution.

Commands are associated with three different queues on-board the robot (PENDING, ACTIVE, and COMPLETED), depending on their status. When the robot receives a new command, it places the command at the tail of PENDING. As soon as resources become available, the command at the head of PENDING shifts to ACTIVE, and the robot performs the command. When finished, the task shifts to COMPLETED (for success or failure). Robot telemetry contains the queue status and result of all commands, along with the current state of the robot. For some command, the state of the robot when the command was initiated must be reported with the telemetry in order for the prediction to work correctly. An ACTIVE command of "Move ten meters ahead" would be impossible to model without knowing where the motion started.

Robot Server

The component of PIGI that manages the flow of messages between the supervisor and the robot is the 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 that have not yet shown up in robot telemetry. All outbound commands stay in SENT for at least the round-trip time delay before showing up in telemetry.

RSvr sends the current robot state to the display, and combines the current robot state with all commands currently on SENT,

PENDING, and ACTIVE into a message that goes to the Predictor

An additional message that includes the COMPLETED queue was used in experiments with decision-support software tools assisting the supervisor with longer-term mission planning [12].

Predictor

The Predictor (PRED) informs 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, where it is used as an initial condition for modeling the robot's response to additional commands. Both path and final state are sent to the display.

PRED makes use of a "Robot Behavior Simulation" (BSim), which takes an initial state and models the response of the robot to a sequence of commands. The BSim is a symbolic state machine, predicting the outcome much faster than real-time every time new telemetry arrives from the robot. Currently, BSim only models drive commands on mobile robots. It derives the expected paths for the initial experiments 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 the same BSim application. These tools will be referred to as PBSim and EBSim, when needed.

Explorer

The Explorer (EXP) enables the supervisor to observe the expected outcomes of 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. 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 go to the display. 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. Otherwise, the command is retracted and the target icon is repositioned.

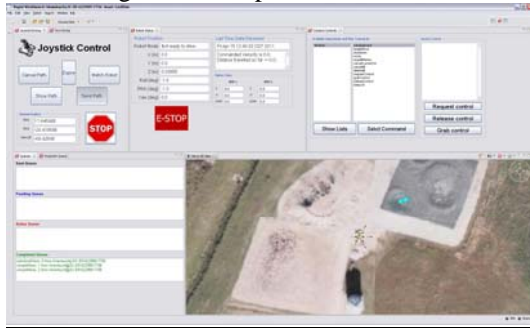


Fig 2. RAPID Workbench showing PIGI and Viz.

RAPID Workbench

PIGI is currently embedded in the RAPID workbench, which was developed to provide a common set of tools that can interact with any RAPID-enabled NASA robot. RAPID is the Robot Application Programming Interface Delegate, developed at NASA Johnson Space Center (JSC), NASA's Ames Research Center (ARC) and the NASA Jet Propulsion Laboratory (JPL)[13]. It is a common API that allows NASA robots and user interface tools to be used together. The visualization used in the workbench is Viz, a RAPID-enabled 3D visualization developed at the ARC. The display visualizes the *a priori* map of a robot's location, the robot's current state, token markers used to "fly" to planned waypoints, trail markers that denote predicted paths both before and after drive commands are sent to a robot and the robot's position history. The workbench also allows for camera views from any RAPID-enabled camera on a robot. These two tools combined with the PIGI GUIs in the workbench provide a full set of features needed to remotely supervise robots over a short to intermediate time delay.

Previous to 2010, PIGI was a standalone GUI that used the JSC's visualization and simulation tool Enigma [14]. Enigma was replaced by Viz when PIGI was ported into the RAPID workbench.

IV. ROBOTS

This section briefly describes the robots at JSC used in testing described in section 5 of this paper. Two other robots from ARC and JPL are also mentioned.



Fig. 2. Centaur 1 robot. Consists of a Robonaut 1 upper body with a four-wheeled lower body.

Centaur 1

Centaur 1 (C1) is an experimental platform combining a humanoid torso (Robonaut 1) with a four-wheeled mobile base [15]. C1 was used to study both teleoperation and autonomy for mobile dexterous manipulation. C1 has cameras and lasers to detect obstacles in its planned paths.



Fig. 3. Space Exploration Vehicle (SEV). Previously known as Chariot or the Lunar Electric Rover (LER).

Space Exploration Vehicle

The Space Exploration Vehicle (SEV) is a rover developed at JSC for both crewed driving and for remote teleoperation. It has six mobility modules, each with independent steering, active suspension, two-speed transmission, and a pair of wheels. It can be operated with or without a pressurized crew module on the vehicle. With certain attachments, the SEV is capable of moving regolith and deploying power systems and habitats [16].

In the spectrum of active suspensions, the SEV system falls into the category of a series active or low bandwidth suspension. The passive suspension elements absorb the high frequency content of driving over rugged terrain and the active element sets the height of the suspension

allowing the vehicle to conform to the terrain. This suspension system is capable of raising and lowering the vehicle, adjusting roll and pitch attitude for docking operations, leveling the chassis against gravity, and balancing the force across the six wheels during low speed operations [17]. Sensors that have been or are currently on an SEV include multiple cameras, SICK lasers, an IBEO laser, and Effector lasers. The active suspension system acts as a safety mechanism to maneuver the rover over rough terrain that is either undetectable by its sensors or may be seen as “easy” terrain through its cameras.

Centaur 2 Base

The Centaur 2 Base (C2 base) was developed to serve as a mobile platform for Robonaut 2, and to hold payloads, such as a dozer implement or an ISRU system. The C2 base has 4 independent “legs”, each which have independent steering and drive motors. Thus, the C2 base can also drive in any direction. It also has a removable pitch joint to allow for interchanging Robonaut 2 with other payloads.



Fig. 4. Centaur 2 base.

Robonaut 2

With 42 independent degrees-of-freedom (DOF's) and over 350 sensors, Robonaut 2 (R2) is an impressive example of mechatronic integration [18]. Encompassing two 7-

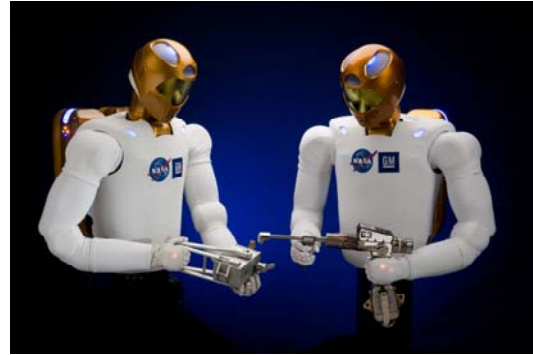


Fig. 5. Robonaut 2. Both versions A and B are seen here. B arrived on aboard the ISS in February 2011.

DOF arms, two 12-DOF hands, a 3- DOF neck and a single DOF waist, the system includes 50 actuators with collocated, low-level joint controllers embedded throughout. The system also integrates built-in computing and power conversion inside its backpack and torso.

Launched on Space Shuttle Atlantis in February 2011 and now onboard the ISS, R2 is designed to fulfill astronaut-assistant duties on board the ISS, while remaining a safe robot for use within working range of the astronauts. To prevent damage to ISS systems and prevent injury to ISS crewmembers, NASA Safety Panels levied a seldom-used set of requirements against R2, requiring that the R2 return to a pre-determined safe state in the event of any off-nominal events.

To prove the safety of R2, the team defined two distinct Fault Containment Regions (FCRs) within the components of R2. These regions consist of sensors, processors, and effectors that are able to measure and react to forces placed on or by the robot joints. Any forces exceeding predetermined limits trigger a failsafe action, namely removing motor power from the joints to prevent any further motion. The first FCR uses force sensors located in the robot's shoulders and forearms to detect the amount of force imparted by the joints. Central processors in the robot's main computer chassis compare these force readings against set limits and can disable motor power in the event of an excessive force. The second FCR is focused in the individual joints of the robot and uses absolute position sensors (APSs) to measure the torque on a spring. Torque above a set limit opens relays in the joint driver to disable motor power and set a fault flag. That fault flag is monitored by the central processors that can trigger a robot-wide motion stop. [19]

Centaur 2

Centaur 2 (C2) is the combination of the C2 base and R2.



Fig. 4. Centaur 2.

SCOUT

The Science Crew Operations and Utility Testbed (SCOUT) was a test-bed rover that preceded the SEV. The main focus of this rover was to develop and test advanced rover technologies and operation concepts, mostly related to transportation of suited astronauts [20].



Fig. 6. SCOUT.

Other robots

The robots presented above are NASA JSC robots that have been supervised remotely. However, the remote supervision paradigm has been used on other NASA centers' robots. They are:

1. **ATHLETE:** The All-Terrain Hex-Legged Extra-Terrestrial Explorer (ATHLETE), developed by 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 [21].

2. **K-10:** K-10 was designed by

ARC as an astronaut assistant for site survey and inspection operations. It features a 4-wheel steer, 4-wheel drive rocker chassis, and the frame can accommodate a wide variety of science instruments [22].

V. TESTING AND RESULTS

PIGI was used to control the robots described across different time delays. Results were taken from multiple analog field tests, which provided “real-world” wireless networking between the robots and the supervisor. In all of these instances, the supervisor was located at JSC, while the robots were several states away.

Commands

The primary task commands used for experiments involving mobile platforms 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). For the manipulation platform, commands used contained relative end-effector positions and opening/closing of hands. Other types of commands were sent from PIGI during these experiments but, since the visualization currently shows changes in location, driving is currently the only task modeled in behavioral simulations. Nonetheless, these other tasks go on the queues in the same manner as drives, and the functionality of the robot server allows the supervisor to mitigate the time delay.

Central Commander and RAPID Sequencer

To accommodate robots that did not already have a functional task queuing service, an additional task-sequencing executive was created for 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. The CCMD exists on C1, SEV and the C2 base to manage on-board robot operations. CCMD manages the three 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 PIGI experiments (ATHLETE, K-10, and SCOUT) do not use CCMD as part of their robot control architecture. Hence, it was

necessary to develop a CCMD for each of them.

The RAPID protocol was developed in 2008. A RAPID bridge was used for each robot that translated the native API of the robot into RAPID. After 2008 a RAPID Sequencer was developed that provided queuing and communicated with both supervisor and robot using RAPID.

Because of the nature of R2 operations on ISS, a CCMD does not currently exist for R2. It uses the R2 Command and Control Interface, which allows for development of quick and easy scripted behaviors. The process of modifying the functionality of the executive commander to work with PIGI is under development. However, because of time constraints, no CCMD or Sequencer exists to function with R2 during PIGI operations, nor does a behavioral simulation exist for manipulation commanding.

Time Delay

The intention of PIGI is not to develop new transport layers for communication in the presence of time delay. During the first few analog tests, no delay capability existed to allow delay of all network traffic. Thus, since RSvr is the single source for outbound traffic from the supervisor to the robots, outbound commands were held for the full delay and telemetry and images were sent back without delay. During the last two analog tests, a network delay was developed for the system such that all information relayed back to remote supervisory stations was equally delayed.

Each test section below describes the delay periods used. The durations chosen were assumed to be the worst-case scenario for communication latency between ground and lunar operations.

2007 Testing

In 2007, the components of PIGI described in Section 4 were developed to investigate the issues of time delay. 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 dead-band 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.

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.

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.

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 tele-operators without time delay, and from the 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 that enabled the rover to reach 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.

2008 Testing

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.

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, with some runs lasting for over an hour and covering over a kilometer. During these drives, there was only idle time when the supervisor was making higher-level decisions about where to go next.

One of the most challenging tasks was "docking", in which the Chariot had to approach an inductive charging station and dock in such a manner that the charging paddle (off-board) was seated into the docking receptacle (on-board). The constrained nature of this task caused the supervisor to adopt "bump-and-wait" with small motions in order to avoid undesired contacts between the robot and charging station.

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.

2010 Testing

In 2010, the Desert RATS field test site was at the Black Point Lava Flow near Flagstaff, AZ. The objective for PIGI testing was to drive the SEV long distances (~10km) and to control two SEVs at the same time through PIGI, using two different supervisors.

The total SEV driving done remotely using PIGI during these tests is 21.8 km. However,

only 8.7 km of that driving was uninterrupted, barring stops for decision-making by the remote operator. The rest of the driving data was marred by weather-related stops, GPS drop-outs or other interruptions. The 8.7 km of driving occurred over 8.65 hours. This translates to an average velocity of 0.28 m/s. The average waypoint distance was roughly 50 meters. Using a “bump-and-wait” philosophy, the time to completion of all driving will be the combination of driving, waiting for return data and the time to plan a path in between waypoints. Assuming an average velocity of 0.28 m/s, a round-trip delay of 10s and an average path-plan time for the operator of 60s, the result of using a bump-and-approach would extend completion of the day’s activities from 8.65 hours to 12 hours. This testing demonstrated that PIGI does reduce the time of operation for remote supervision of robots, even in a 10s round-trip delay.

The dual-SEV operations allowed each supervisor of an SEV to see the predicted paths of the other SEV in their own visualization. This allowed for a total of 0.65km of convoy driving of both SEVs.

2011 Testing

The Black Point Lava Flow was again the analog site for testing in FY 2011. The objectives for this test were to operate the C2 base remotely as a geologist’s tool and to operate C2 (both R2 and C2 base combined) as a human-assistant robot. In both of these cases, a 100 second round-trip communication delay was inserted into the network stream. This was the largest time delay that was used in all of the analog testing.

PIGI was used with a team of geologists to direct the desired locations of C2 and to capture imagery from C2’s camera for further study. The queuing of commands aided in shortening the time it took to get to specific locations, but it was discovered that camera views could not be queued, due to the time delay. Because of the exploration nature of the geologists work, knowing the exact location to direct the camera to was impossible. Therefore, a bump-and-wait approach had to be undertaken to get the camera into a desired position for image sampling. The delay became very noticeable during this mode of operation.

PIGI was also used to direct the C2 base during C2 operations, which involved C2 collecting geology samples. Because R2 had not been outfitted with a sequencer, each command sent to the R2 body was a single command. However,

R2’s fault containment system was used to keep the arms safe during pick-up maneuvers. Because the pick-up procedure involved detection of forces, it was possible to send the arm into the ground without any safety mechanisms to stop motion when force values go out of range. The fault containment feature was used during multiple instances when the geology sample was not accurately found.

VI. CONCLUSIONS AND FUTURE WORK

This paper has presented advances in robot safety and command interaction approaches developed to form effective human-robot teams in the presence of time delay. PIGI, a novel method of interacting remotely with robots, has shown that it allows a human to supervise multiple types of robots. 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 and expedite the coordination of tasks between robots.

These results demonstrated the advantage of PIGI’s queuing process and prediction aspects. Specifically, it was learned that interleaving non-driving and driving tasks reduced the idle time of the robot and thus improved the efficiency of the overall mission. Also, visualizing when the predicted behavior of the robot changed due to interaction with the environment aided all operations, especially the dual-SEV operations. These changes showed up immediately in the display and the supervisor was able to modify the commands in the queue.

Safety and robustness of remote operations were shown to be enhanced by advanced features developed on some of NASA’s robots. The Space Exploration Vehicle’s active suspension response was detailed, as was Robonaut 2’s fault tolerance system for operating in the presence of ISS astronauts.

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

It should be noted that R2's system will continue to be adapted for use with PIGI which will enable the option of a supervisor in mission control using the predictive interfaces to assist astronauts on ISS in controlling R2, freeing up the astronauts' time for other duties. This will require complex behavioral simulations for manipulation and the addition of a sequencing mechanism that interacts with the current R2 command and control application.

REFERENCES

- [1] Hokayem, P.F., Spong, M.W.. Bilateral teleoperation: An historical survey, *Automatica* Volume 42, Issue 12, December 2006, Pages 2035-2057.
- [2] Hambuchen, K., W. Bluethmann, S.M. Goza, R. Ambrose, K. Rabe, M. Allan, "Supervising Remote Humanoids Across Intermediate Time Delay," *Proceedings of the 2006 IEEE-RAS International Conference on Humanoid Robots*, Genoa, Italy, December 4-6, 2006.
- [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] 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.
- [5] Ferrell, W.R., T.B. Sheridan. Supervisory control of remote manipulation. *IEEE Spectrum* 4, no. 10, October 1967. pp 81-88.
- [6] 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.
- [7] Sheridan, T. B. *Telerobotics, automation, and human supervisory control*. MIT Press, Cambridge, MA. 1992.
- [8] Frohlich, J., R. Dillman. Interactive Robot Control System for Teleoperation. In *Proceedings of 24th International Symposium on Industrial Robotics*, Tokyo, Japan 1993.
- [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. 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] Schreckenghost, D., T. Ngo, R. Burrige, L. Wang, M. Izygon. Remote Task-Level Commanding of Centaur over Time Delay. *Proceedings, Space Technologies and Applications International Forum*, Albuquerque, NM, Feb. 2008.
- [13] Torres, R. J., Allan, M., Hirsh, R., Wallick, M.N. RAPID: Collaboration Results from Three NASA Centers in Commanding/Monitoring Lunar Assets. In *Proceedings of IEEE Aerospace Conference*, Big Sky, Montana. March, 2009.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] Bluethmann, B., E. Herrera, A. Hulse, J. Figuered, L. Junkin, M. Markee, R. O. Ambrose. An Active Suspension System for Lunar Crew Mobility. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana. March, 2010.
- [18] M.A. Diftler, Mehling, J.S, Abdallah, M.E., Radford, N.A., Bridgwater, L.B., Sanders, A.M., Askew, R.S., Linn, D.M., Yamokoski, J.D., Permenter, F.A., Hargrave, B.K., Platt, R., Savely, R.T., Ambrose, R.O. Robonaut 2 – The First Humanoid Robot in Space. In *Proceedings of 2011 International Conference on Robotics and Automation*. May, 2011.
- [19] M. A. Diftler, Ahlstrom, T. D., Ambrose, R. O., Radford, N. A., Joyce, C.A., De La Pena, N., Parsons, A.H., Noblitt, A.L. Robonaut 2 – Initial Activities On-Board the ISS. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana. March, 2012.
- [20] Rochlis, J., F. Delgado, J. Graham. Science crew operations and utility testbed. *Industrial Robot: An International Journal*, 33(6), pp 443-450, 2006.
- [21] 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.
- [22] 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.