The Avatar Project

Remote Robotic Operations Conducted from the International Space Station

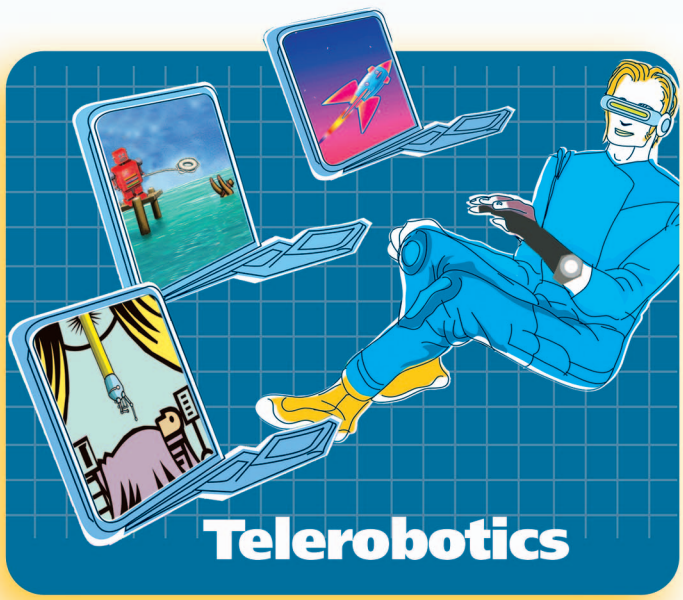
BY ERIC MARTIN, RÉGENT L'ARCHEVÊQUE, SÉBASTIEN GEMME, IOANNIS REKLEITIS, AND ERICK DUPUIS

As the International Space Station (ISS) flies over the Canadian Space Agency (CSA) headquarters located near Montreal, Canada, a Russian cosmonaut is carefully peering at the screen of his laptop controlling a robotic test bed located in CSA laboratories, 350 km below. He must hurry because his communication window will last at most 10 min until the radio ground station disappears below the horizon. His next opportunity to interact with the robot will come approximately 90 min later.

This cosmonaut is running a series of teleoperation experiments collectively named as Avatar, whereby different command and control schemes are evaluated in view of future planetary exploration missions. These experiments are invaluable in view of the future exploration initiatives currently being defined by most space agencies. Indeed, these plans place a strong emphasis on robotic missions, either in support of human missions or as a precursor to astronaut flights. Most of these robotic missions will involve remote operation of robotic assets. Examples include controlling orbiting robots or planetary rovers from Earth-based stations or from manned orbital stations. For better efficiency, the operation of these robotic systems shall be performed with various level of autonomy.

During the past few years, CSA has been developing the autonomous robotics and ground operations (ARGO) software suite [1]. ARGO provides a framework for the integration of the space operations process from planning to postflight analysis. The objective of ARGO is to reduce the operational costs and increase the efficiency by providing operator aids and permitting the implementation of a level of autonomy appropriate to the application. The target applications of the ARGO framework cover the full spectrum of autonomy from supervisory control such as might be expected for the ISS robotics to more autonomous operations such as would be encountered in planetary exploration missions. It also covers the full range of space robotic applications from orbital manipulators to planetary exploration rovers. One of the important features of the ARGO framework is that it does not provide a universal architecture for ground control and autonomy. Instead, ARGO provides a set of toolboxes that can be assembled in a variety of manner depending on the application and its requirements. To facilitate the reuse of software, the design is modular and portable to the maximum extent possible.

Within the ARGO framework, in 2006–2007, the robotics group of the Space Technologies division of CSA demonstrated many times the autonomous capture of a tumbling satellite. CSA's Automation and Robotics Test Bed (CART) was operated from different remote locations using low bandwidth Internet links. As shown in Figure 1, CART is a dual-manipulator test bed where each arm has 7 degrees of freedom (DoF). One arm emulates



© CREATAS, ARTVILLE, & DIGITAL VISION

Digital Object Identifier 10.1109/MRA.2008.929926

the free-flyer dynamics, and the second is the chaser robot equipped with a self-adapting robotic auxiliary hand (SARAH) [2]. The laser camera system (LCS) [3] from Neptec is used to guide the chaser robot. A hierarchical finite state machine (FSM) engine based on Cortex, one of the components of ARGO, is used to coordinate the autonomous capture task and to provide feedback to the remote operator. The role of the operator is very simple: initiate the capture by submitting a high-level command and monitor the chaser robot while performing the task. The sequencing of events (approach, fly together, and capture) is fully autonomous.

In the first planned Avatar mission, Avatar RESCUE, the project, described in the previous paragraph, is brought one step further by initiating the commands to operate the system from the ISS. An amateur radio already available on the Russian segment of the ISS is used to communicate with a ground radio counterpart located at CSA facility in St-Hubert. The concept of Avatar RESCUE flight demonstration is summarized in Figure 2. This mission is to be executed during the fall of 2008.

The Avatar RESCUE experiment is implemented using a reverse ground-control configuration. In a real satellite servicing mission, the robot would be in the Earth orbit, and the operator would be located at a ground-control station somewhere on Earth. For Avatar RESCUE, the robot is located in a laboratory on Earth, and the operator is orbiting the Earth onboard the ISS. This seemingly reversed configuration faces exactly the same challenges as the real implementation but allows the costs to be dramatically reduced by keeping the robotic infrastructure on the ground. The only component that needs to be sent to space is the operator station.

In this article, the various components of the Avatar RESCUE mission will be described in detail in the following sections. An overview of the next phase, the Avatar EXPLORE mission, is presented in the "Avatar EXPLORE" section.

Concept of Operation

The concept of operation of the Avatar experiments is dictated primarily by the constraints imposed by the communications between the operator and the remote robot.

The first factor to consider is the duration of the windows of communications: interactions between the ISS operator and the ground robot can only occur while direct radio contact is established. As mentioned previously the ISS orbits around the Earth at an altitude of approximately 350 km and with a period of approximately 90 min. As a result, there are typically four or five windows of communication per day with the ISS for a fixed ground-based radio station. These communication windows last for a maximum of 10 min and are approximately 90 min apart. It is, therefore, important to design the operations such that they can be safely executed within this window.

The second factor to consider is the limited bandwidth imposed by the usage of an amateur radio link for command and telemetry transfer. The current setup is limited to approximately 600 b/s. It is, therefore, impossible to transfer a video stream of the remote scene to the operator, thus limiting his situational awareness. To help the operator, a three-dimensional (3-D) model of the chaser manipulator and the target satellite is rendered. This model requires only the transfer of 13 numbers corresponding to the seven joint angles of the manipulator and the six numbers required to represent the pose of the satellite with respect to the base of the manipulator. This 3-D model is presented in Figure 3 as part of the operator station. The scarcity of information returned to the operator limits his decision-making capability and forces much of the decision-making capability to reside on the robot. It is unrealistic to assume that the operator could guide a robot manually using a set of joysticks to capture a tumbling satellite under these conditions.

Given these constraints, the operator is, therefore, conducting operations in a supervisory mode. Different levels of autonomy

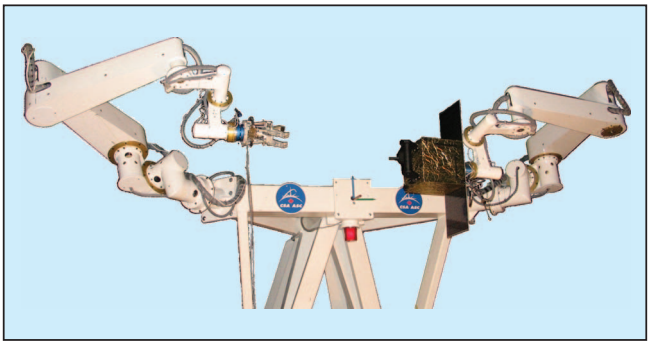


Figure 1. *A dual manipulator system that simulates the tracking and capture scenario. The manipulator on the left is equipped with a hand, and the manipulator on the right is carrying a satellite mock-up.*

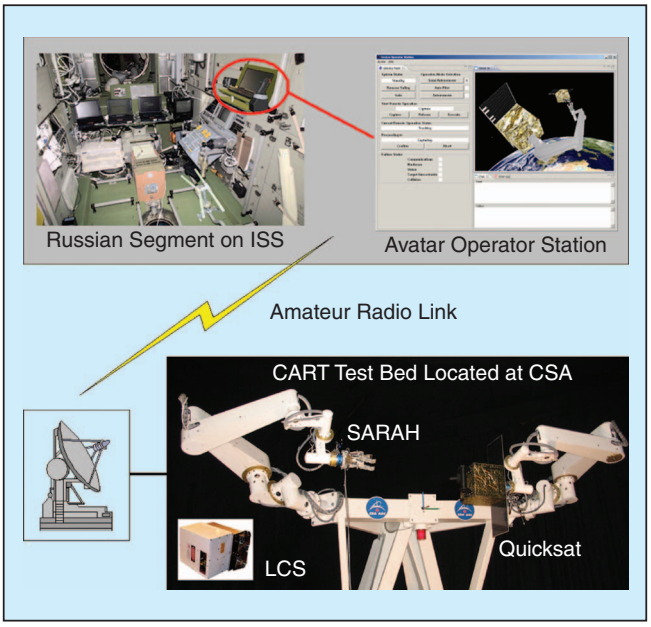


Figure 2. *Concept of the flight demonstration.*

are investigated through three different modes of operation: autonomous, semiautonomous, and autopilot operations, later referenced as operation modes.

In autonomous mode, the transitions between the various phases in the operations are commanded by the autonomy software residing on the robot. The sequencing of commands is determined by the autonomy engine that will take the necessary actions to capture the satellite or abort the mission in case of malfunctions. The only intervention possible by the operator is the abort command, which aborts the mission. At the limit, if the communication link is broken after the capture command is received on the ground, the operation will continue. Once the communication is reacquired, the operator will get feedback on the state of the system and see if the operation was successful.

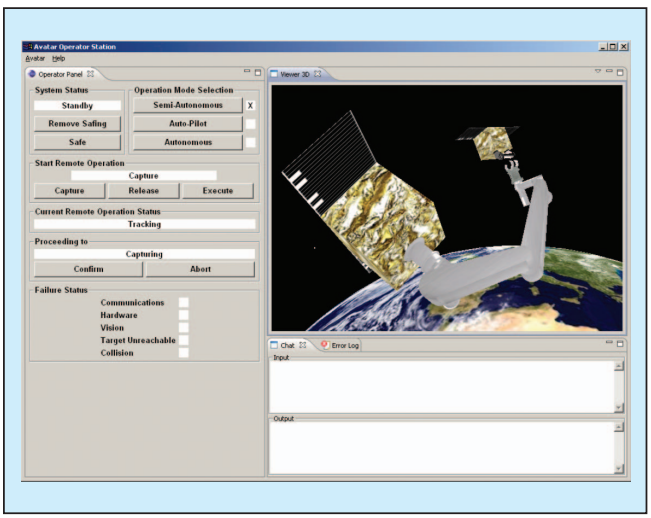


Figure 3. *GUI of the Avatar Operator Station.*

In semiautonomous mode, the autonomy software verifies conditions required for transitions between states. However, operator confirmation is required to trigger transitions between critical phases of the operation. If the communication link is broken in this mode, then the mission will abort and the chaser manipulator will return to a safe state.

The autopilot mode is halfway in between the autonomous mode and the semiautonomous mode. The operations can be conducted from end to end without operator intervention, but the transitions between critical phases are driven by autonomy software running on the operator station. In autopilot mode, the software located on the robot segment is identical to the case of semiautonomous operations. If the communication link is lost in autopilot mode, the mission will abort, and the chaser manipulator will return to a safe state. This control mode was implemented to reflect the situation that, for space robotics applications, it is often easier to change software on the operator station than control software on the robot itself.

In all operation modes, the control system on the robot is responsible for dynamic control actions such as tracking the tumbling satellite based on artificial vision data. The differences between the three modes are implemented in a decision layer that controls transitions between control modes throughout the operation.

A typical satellite capture operation is implemented as illustrated in Figure 4. This figure depicts an FSM implementation of the different control modes (e.g., initial approach, align,

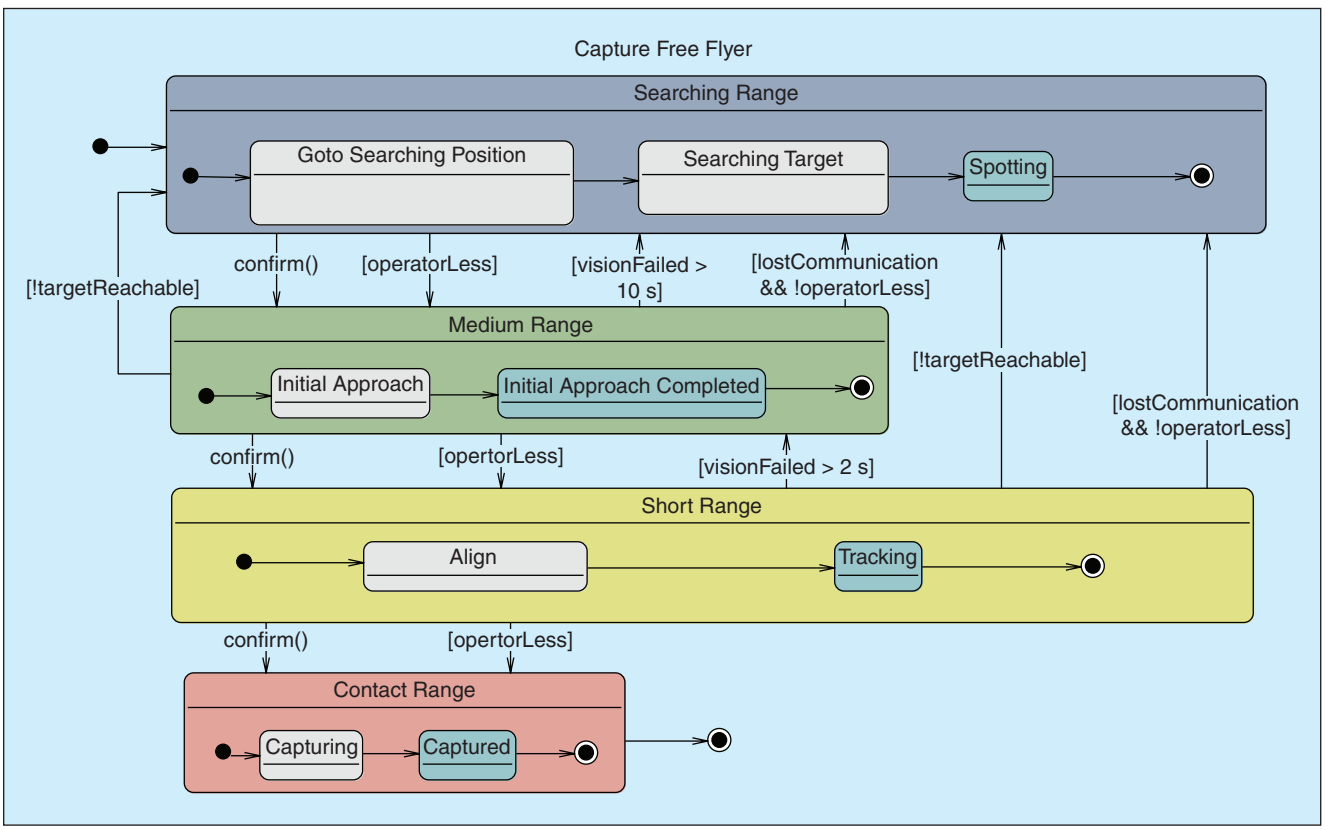


Figure 4. *Avatar capture state machine.*

tracking, capture) involved in the capture process and their sequencing. The operation is decomposed into four major phases: the searching-range phase, the medium-range phase, the short-range phase, and the contact-range phase.

In the searching-range phase, the autonomy software commands the LCS to detect and identify the target satellite to be captured. This phase ends after the vision system has positively identified the satellite and returns the pose of the satellite in a data stream.

The next phase in the operation is the medium-range phase. The chaser arm is commanded to approach the satellite grapple fixture within a given standoff distance but without lining up the end effector with the grapple fixture. This phase is completed when the chaser manipulator attains the initial approach location.

The next phase in the operation sequence is the short-range phase, where the chaser manipulator is commanded to align the end effector with the grapple fixture and move to the final approach location. The visual-servoing controller then maintains the end effector aligned with the grapple fixture of the tumbling satellite at a distance of a few centimeters, ready for capture.

The final phase is termed the contact-range phase. The manipulator is then commanded to engage the satellite and to complete the capture of the grapple fixture using the end effector, thus completing the operation.

Off-nominal conditions can occur at any time throughout this sequence. For example, the LCS could lose track of the target satellite, communications could be lost, or the target satellite could move out of capture range. The local autonomy software is in charge of commanding appropriate maneuvers to keep the system safe in the presence of anomalies. These safety-critical maneuvers are triggered autonomously on the robot regardless of the operation mode selected by the operator. The selection of operation mode affects mainly the nominal transitions between the four main phases in the operations sequence.

Autonomy Software

The heart of the Avatar RESCUE experiment is the autonomy software that is used to assist the operator in successfully conducting robotic operations. The software is based on two closely coupled layers. The first layer is the control layer that is in charge of guiding the robotic chaser toward the tumbling satellite. The second layer is a decision layer that handles state changes throughout the capture operation and that handles safety critical maneuvers in case of anomalies. The decision layer is based on CSA's Cortex Autonomy Toolbox, which is one of the toolboxes developed under the ARGO framework.

Cortex Autonomy Toolbox

Cortex provides a set of tools to implement onboard autonomy software based on the concept of hierarchical FSMs. Cortex allows an operator to graphically generate the behaviors to be implemented on the remote system. It automatically generates the code to be uploaded, and it can be used to debug and monitor the execution of the autonomy software online and off-line.

Cortex has been developed in light of the fact that the development of such behavior sets rapidly becomes labor intensive even for relatively simple systems when using low-level programming languages, thus making reusability very difficult, if not impossible. Cortex is based on the FSM formalism, which provides a higher-level way of creating, modifying, debugging, and monitoring such reactive autonomy engines. Some advantages of this representation are its intuitiveness and ease with which it can be graphically constructed and monitored by human operators. The concept of hierarchical FSM allows a high-level FSM to invoke a lower-level FSM. This provides the capability to implement hierarchical task decomposition from a high-level task into a sequence of lower-level tasks. If the FSM is implemented in a modular fashion, it allows the implementation of the concept of libraries that provide the operator with the reuse of FSMs from one application to another.

Capture Autonomy Scenario

Using Cortex, the Avatar capture state machine presented in Figure 4 has been implemented. Based on the filtered poses of the capture frame $\mathcal{F}_C$ attached to the capture handle, and the tool frame $\mathcal{F}_T$ attached to the grasping device (see Figure 5), a Cartesian velocity command is generated in the tool frame $\mathcal{F}_T$ of the manipulator to progressively reduce the distance between them. Generating the command in the tool frame provided the opportunity to independently activate and assign each DoF to a different task. In the experiments, the activation of the tasks is triggered by Cortex based on the distance between the grasping device and the target satellite. Each of these tasks generates a part of the desired end-effector twist as

$$\mathbf{t}_{\text{desired}} = \begin{bmatrix} \dot{\mathbf{r}}_{\text{desired}} \\ \boldsymbol{\omega}_{\text{desired}} \end{bmatrix} = [\, v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z \,]^T. \quad (1)$$

Referring to Figure 4, once the autonomy software is activated, it automatically enters in the searching-range phase where the chaser manipulator goes in its safe or searching position. Once the vision system identifies the target satellite, the autonomy software goes in the medium-range phase. At that point, the initial approach is started, and the manipulator end effector orients itself toward the target by adjusting its pan-tilt motion using only 2 rotational DoF as

$$\omega_y = -k_{R_y} \tan^{-1}(r_z/r_x), \quad (2)$$

$$\omega_z = -k_{R_z} \tan^{-1}(r_y/r_x), \quad (3)$$

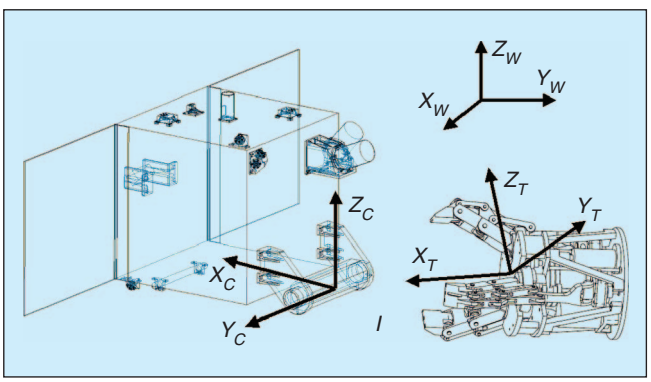


Figure 5. *Computer-aided design (CAD) models of Quicksat and SARAH showing reference frames.*

Autonomy scenario is coded using the CSA's Cortex autonomy toolbox.

where k_{R_y} and k_{R_z} are control gains for their respective axes, and r_x, r_y, and r_z are the components of $[\mathbf{r}_{T/C}]_T$, namely, the position of the capture frame $\mathcal{F}_C$ with respect to the tool frame $\mathcal{F}_T$. It can be calculated as

$$[\mathbf{r}_{T/C}]_T = \mathbf{R}_T^T([\mathbf{r}_C]_w - [\mathbf{r}_T]_w), \tag{4}$$

where the position of the capture frame $\mathbf{r}_C$ is the output of a Kalman filter operating on LCS data, and $\mathbf{r}_T$ and $\mathbf{R}_T^T$ are calculated from the kinematics model of the chaser manipulator. These quantities are all expressed in the base frame $\mathcal{F}_w$. At this point, the desired distance between the grasping device and the target is controlled by the translational DoF that move the end effector forward and backward. The command is computed from

$$\dot{\mathbf{r}} = \mathbf{K}\mathbf{R}_T^T\mathbf{R}_C([\mathbf{r}_T]_C - [\mathbf{r}_{\text{des}}]_C), \tag{5}$$

where $\mathbf{R}_T$ and $\mathbf{R}_C$ are, respectively, the rotation matrices representing the orientation of the tool frame $\mathcal{F}_T$ and the capture frame $\mathcal{F}_C$ with respect to the base frame $\mathcal{F}_w$; $\mathbf{K}$ is the control gain matrix defined as $\mathbf{K} = \text{diag}(k_x, k_y, k_z)$; $[\mathbf{r}_T]_C$ is the position of the tool frame expressed in the capture frame; and, finally, $[\mathbf{r}_{\text{des}}]_C$ is the desired tool frame position relative to the capture frame. In (5), the gains k_x, k_y, and k_z are not used simultaneously. At first, during the initial approach, only the approach gain (k_x) is activated.

To avoid undesirably large end-effector velocities, the grasping device is aligned perpendicular to the target only when they are close to each other. Therefore, only when $\dot{\mathbf{r}}(x) = 0$, the state of the system becomes initial approach completed, and the chaser manipulator is ready to proceed to the short-range phase. On entering the short-range phase, the translational alignment is initiated by activating the alignment gains (k_y and k_z) to use the remaining 2 translational DoF of (5). When $[\mathbf{r}_T]_C = [\mathbf{r}_{\text{des}}]_C$, the grasping device is positioned directly in front of the target, and the final roll alignment is initiated to align the hand so that the handle fits between the fingers of the hand. To that end, the remaining rotational DoF is used:

$$\omega_x = -k_{R_x}\theta_{x_{T/C}}, \tag{6}$$

where k_{R_x} is the control gain, and $\theta_{x_{T/C}}$ is the orientation of the capture frame $\mathcal{F}_C$ about the x-axis of the tool frame $\mathcal{F}_T$.

At the completion of these tasks, with all the corresponding controllers active, the chaser manipulator is positioned very close to the target and is tracking its motion. The final capture command is issued by entering the contact-range phase. The capture is executed by adjusting the desired relative pose, $[\mathbf{r}_{\text{des}}]_C$, to have the handle in the middle of the hand and then closing the SARAH hand fingers.

Malfunctions

In all three operation modes, five different malfunctions have been considered, and the autonomy scenario has been coded accordingly. These malfunctions can be generated artificially by the controller of the mission on the ground to check the correctness of the autonomy scenario. They can also be detected automatically during the mission if an anomaly would actually occur. In both cases, the feedback to the operator will be the same and presented in the operator panel. As one can observe in Figure 3, the possible malfunctions are as follows:

- loss of communication link
- hardware problem
- loss of vision system
- target unreachable
- risk of collision.

Some malfunctions may have a different effect depending on the mission stage. For example, as illustrated in the capture state machine of Figure 4, if the vision system failed during the medium-range phase where the chaser manipulator is a few meters away from the satellite, the chaser arm would continue to approach the satellite for 10 s while the autonomy engine tries to restart the vision system. If the restarting commands are not successful, then the arm would return to a safe state. On the other hand, if the failure would occur during the final approach, in the short-range phase, the approach could still continue but only for a shorter period of time of 2 s, while trying to restart the vision system. If the vision signal is not reacquired, then the autonomy engine would bring back the chaser arm at the end of the initial approach phase for a period of 8 s, again trying to solve the problem with the vision system. Finally, if the failure would occur during the contact-range phase, the capture of the satellite will proceed by using the prediction of the pose of the satellite generated by the extended Kalman filter.

Experimental Test Bed

The experimental test bed used for the experiment is based on CSA's CART. It is composed of three main hardware components, namely, the target satellite held by one robotic arm, a vision system composed of a scanning laser range sensor with pose determination software, and a chaser manipulator equipped with a dexterous end effector: the SARAH hand. Another important element of the Avatar project is the amateur radio communication link between the ISS and CSA ground station. A short description of each element is presented later; see [4] for an in-depth description.

Target Satellite

The target satellite is a two-third scale mock-up of a Canadian microsatellite called QuickSat. Its trajectory is generated offline, and it mimics the dynamics of the tumbling motion of a satellite in free fall. The ensued motion of the target satellite can be characterized as a major rotation about a spin axis with small precession and nutation [5]. This is similar to the motion described in [6].

Vision System

The LCS, shown in Figure 6(a), and the Collision Avoidance and Pose Estimation (CAPE) software, both from Neptec [3],

are used to generate the pose (position and orientation) of the target satellite to guide the Chaser manipulator throughout the capture operation.

The LCS sensor is particularly suited for space application as it is immune to harsh and/or changing lighting conditions [3]. In addition, the LCS is capable of handling solar interference. It has been successfully flown on the space shuttle Discovery (STS-105) in 2001 and is now used routinely to inspect the thermal protection tiles on every space shuttle flight. The range data from the LCS sensor is processed using Neptec's proprietary software CAPE, and the pose of the satellite is obtained. The pose estimation method is model based, using a CAD model of the target satellite and a modified version of the iterative closest point algorithm [3].

The pose is calculated at about 2 Hz, with a delay of 0.5 s on the pose of the object at a given instant. Careful calibration of the LCS positioning is required to transfer the pose data in the proper coordinate frame to guide the capture.

Chaser Manipulator

The chaser manipulator is symmetric to the manipulator carrying the satellite mock-up. It is equipped with an underactuated dexterous robotic hand developed by Université Laval [2]. The SARAH hand, has three reconfigurable fingers mounted on a common structure. SARAH has 10 DoF but can be actuated with only two drive systems. One drive controls the opening and closing of the fingers, whereas the other drive controls the orientation of the fingers to reconfigure the grasp. Each finger of SARAH has three phalanges. The self-adaptability of the hand is obtained using underactuation. Note that, although the hand passively adapts to any geometrical shape, it is not back-drivable and, therefore, provides a firm grip. The hand is shown in Figure 6(b).

The chaser manipulator is guided strictly based on target pose data obtained from the vision system. An extended Kalman filter is used to filter the LCS raw measurements and to provide a smoothed pose of the target satellite every millisecond. The Kalman filter is fully adaptive and does not need any a priori knowledge of the inertia properties of the satellite or the noise properties of the vision sensor [7]. After a short observation period, the filter converges to the actual motion of the satellite and can be used to accurately predict the pose of the satellite even when the vision system becomes occluded. The system has been demonstrated to successfully perform the capture of the satellite after the vision system had been blinded for periods of over 20 s [8].

Communication Link

The communication infrastructure between the ISS and the ground station is based on an amateur radio link using the AX.25 protocol to transfer data. AX.25 is the equivalent of the IP part in the transmission control protocol (TCP)/IP protocol. To guarantee reception of the AX.25 packets, an extra layer, the delay tolerant protocol (DTP), was developed internally at CSA. DTP is the equivalent of the TCP part of the TCP/IP protocol.

Table 1 compares many candidate communication protocols to run over AX.25. The comparison includes the space communication protocol standard (SCPS), Licklider transmission protocol (LTP), user datagram protocol (UDP), TCP, AX.25, and DTP. Each of these protocols was evaluated for its suitability for the Avatar mission using the following criteria:

- data integrity (assurance of data transmission)
- low bandwidth usage (the effective bandwidth of an amateur radio link is about 600 b/s)
- adaptability to amateur radio
- tolerance to high communication delays (in the order of minutes).

From Table 1, three choices appear suitable for the proposed application, namely, SCPS, LTP, and DTP. The SCPS protocol was not considered because of its complexity of implementation and adaptation for the amateur radio. The LTP protocol could have been appropriate. However, the LTP protocol was not available at the time this work was started: it has been developed during the same period as the proposed DTP protocol.

The DTP protocol is purely a transport layer protocol that can be overlaid onto any network level protocol such as, in our case, AX.25. DTP would also be suitable for IP-based communication, simply by integrating it with the IP protocol as its network layer.

The DTP implementation is in the form of a Java input/output stream, which means that communicating by DTP is identical to writing to any other stream. For instance, combining the DTP output stream with an object output stream allows us to send Java objects through ham radio, the same way someone would do over a TCP/IP network.

Current Status and Future Plans

Avatar RESCUE

At the time this article is submitted, the first test runs of the Avatar RESCUE experiment are about to be executed. The

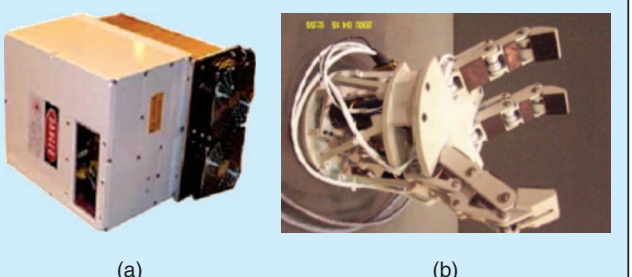


Figure 6. *(a) LCS. (b) Underactuated SARAH hand.*

Table 1. Comparison of data communication protocols.

Protocol	Data Integrity	Bandwidth Usage	Adaptability	Tolerance to Delays
SCPS	Yes	Average	Yes	Yes
LTP	Yes	Low	Yes	Yes
UDP	No	Low	No	Yes
TCP	No	High	Yes	No
AX.25	No	Low	Yes	Yes
DTP	Yes	Low	Yes	Yes

Robotic operations are performed over low-bandwidth communication links.

operator station software has been shipped to the ISS via a Russian Soyuz flight on 8 April 2008. A first communication experiment was performed in October 2008, and other communication tests are planned for November 2008. The teleoperation experiments should be performed by ISS Expedition-18 before the end of December 2008.

Avatar EXPLORE

The next planned experiment in the Avatar series is the Avatar EXPLORE experiment whereby a mobile robotic test bed located in the Mars-emulation terrain (MET) located at CSA headquarters in St-Hubert will be autonomously operated from the ISS. This Mars-like terrain, shown in Figure 7, simulates the topography found in typical Mars landscapes.

The Avatar EXPLORE experiment will build on the building blocks tested under Avatar RESCUE, such as the transfer of data at low bandwidth using an amateur radio link to control a robotic experiment from space. It will also build on CSA-developed autonomous planetary exploration software, based on the ARGO framework [9].

In the exploration of the MET, the mission is to search and identify a target thermally different from the rest of the environment. The exploration is conducted by a rover operated by an astronaut onboard the ISS. The rover is equipped with a 3-D laser range scanning sensor, a thermal imager, a flash lidar for obstacle detection, and a six-axis inertial measurement unit.

When the mission starts, the operator has a coarse global map of the MET loaded on his operator station. The operator prepares a list of commands for the rover to travel to selected destination points and to acquire laser scan and thermal images. The list of commands is displayed in the operator station, and once the operator is satisfied, he can download the commands before the loss of contact with the ground.

Figure 7. *A view of the MET located at CSA Headquarters in St-Hubert, Quebec, Canada.*

The rover travels autonomously to the destination points considering exclusion zones provided by the operator. Using a flash lidar, the rover can detect unexpected obstacles. In this situation, the rover autonomously stops, takes a new scan of its environment, and plans a new path considering again the exclusion zones specified by the operator.

The data resulting from each scan are decimated by an application that generates a representation of the terrain as a mesh of a minimum set of triangles. The thermal images are processed and compressed on a hybrid processing board on the rover in a way to provide useful information to the operator and being transferable using the ham radio link. The resulting temperature images are relayed to the operator station along with the terrain scan data. The topographical data are displayed to the user superimposed onto the global terrain map. The operator has the possibility to superimpose all previous scans on the global map. He also has the possibility to select any thermal images taken since the beginning of the mission and visualize it on a two-dimensional (2-D) view panel of the operator station. The operator is provided with a set of tools to analyze the content of the 2-D thermal images and to triangulate in the 3-D terrain model the location of thermally interesting features.

Given the short duration of the communication windows, the large amounts of data to be transferred, and the slow speed of planetary rover operations, the rover will operate mostly when the operator is out of radio contact. The contact period will be used to send telemetry to the operator, to plan the next sequence, and to send commands to the rover. The process continues until the thermally distinct target is identified and the rover reaches it to take a final scan of the target's surroundings. The Avatar EXPLORE mission is currently planned for the summer of 2009.

Conclusions

CSA is currently developing and conducting a series of experiments dubbed Avatar to investigate different command and control schemes allowing operators to interact with robots in space or on other planets. The objective of the Avatar experiments is to develop and test concepts in support of future space exploration missions.

Although some of the concepts can be (and have been!) tested on Earth by simulating space-relevant communication links, the benefits of conducting them from the ISS are numerous. First, the usage of an intermittent amateur radio link has raised several issues regarding the robustness of the software: it is impossible to cheat when the communication link really goes down. It has also allowed the team to develop unique operational expertise. One final advantage not to be neglected is the fact that these experiments will have provided flight heritage to the command and control concepts described in this article and to the software that was used to implement them. Such heritage is precious in the traditionally conservative space community.

By the time this article goes into press, preliminary communication tests will have already been conducted, and the first set of trials of Avatar RESCUE should have been initiated. The next Avatar experiment, Avatar EXPLORE, is planned for the summer of 2009. In Avatar EXPLORE, an astronaut

located on the ISS will control an autonomous rover operating in a planetary analog terrain located at CSA headquarters.

Keywords

Space robotics, teleoperation, autonomy, on-orbit capture of satellites, amateur radio, low-bandwidth communication link.

References

[1] E. Dupuis, R. L'Archevêque, P. Allard, I. Rekleitis, and E. Martin, "A framework for autonomous space robotics operations," in *Intelligent Space Robotics*, A. M. Howard and E. W. Tunstel, Eds. Albuquerque: TSI Press, 2006, pp. 217–234.

[2] T. Laliberté, L. Birglen, and C. M. Gosselin, "Underactuation in robotic grasping hands," *Jpn. J. Mach. Intell. Robot. Contr. (Special Issue on Underactuated Robots)*, vol. 4, no. 3, pp. 77–87, 2002.

[3] S. Ruel, C. English, M. Anctil, and P. Church, "3DLASSO: Real-time pose estimation from 3D data for autonomous satellite servicing," in *Proc. 8th Int. Symp. Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Munich, Germany, Sept. 5–8, 2005. [CD-Rom].

[4] I. Rekleitis, E. Martin, G. Rouleau, R. L'Archevêque, K. Parsa, and E. Dupuis, "Autonomous capture of a tumbling satellite," *J. Field Robot. (Special Issue on Space Robotics)*, vol. 24, no. 4, pp. 275–296, 2007.

[5] H. Goldstein, *Classical Mechanics*, 2nd ed. Reading, MA: Addison-Wesley, 1980.

[6] H. Nagamatsu, T. Kubota, and I. Nakatani, "Capture strategy for retrieval of a tumbling satellite by a space robotic manipulator," in *Proc. IEEE Int. Conf. Robotics and Automation*, Minneapolis, MN, 1996, pp. 70–75.

[7] F. Aghili and K. Parsa, "Configuration control and recalibration of a new reconfigurable robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, Roma, Italy, May 2007, pp. 4077–4083.

[8] F. Aghili, K. Parsa, and E. Martin, "Robotic docking of a free-falling space object with occluded visual condition," in *Proc. 9th Int. Symp. Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Los Angeles, CA, Feb. 26–29, 2008. [CD-Rom].

[9] I. Rekleitis, J.-L. Bedwani, and E. Dupuis, "Over-the-horizon, autonomous navigation for planetary exploration," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, CA, Oct. 2007, pp. 2248–2255.

Eric Martin received his Ph.D. degree at McGill University on the dynamic interactions of a space manipulator with its base attitude controller. He joined the Robotics Group at the CSA in 1999, where he was one of the main developers of the Special Purpose Dexterous Manipulator (SPDM) Task Verification Facility used to emulate on ground the Dextre robotic system on the ISS. He was involved in the development of numerous robotics simulators. Since 2005, he has been leading the R&D activities in on-orbit servicing and assembly at the CSA, where the main focus is on the autonomous operation of robotic systems. He is the project manager for the Avatar missions presented in this article.

Régent L'Archevêque received his bachelor's and master's degrees in software engineering from École Polytechnique de Montréal and specialized in virtual reality and computer graphics. He has been a software engineer at the CSA since 1998. He started his career in robotics in the development of multibody systems modeling tools and astronaut training simulators. He has been mainly involved in the development of software to support teleoperation of autonomous systems. He recently designed and implemented the ground data system of the Meteorological station of the Phoenix Mars polar lander. He is leading the software development under JC2Sat that consists of a two satellite constellation with free formation flying capability.

Sébastien Gemme received his bachelor's degree in computer science from Université du Québec à Montréal and a master's degree in computer engineering from École Polytechnique de Montréal, specializing in computer vision. He has been a software developer and a UNIX systems administrator at the CSA for the past eight years. His work has been focused on automatic 3-D registration and amateur radio data communication for space applications.

Ioannis Rekleitis received his B.Sc. degree from the Department of Informatics, University of Athens, Greece. He obtained his Ph.D. and M.Sc. degrees from the School of Computer Science, McGill University. During 2002–2003, he was a postdoctoral fellow in Carnegie Mellon University and worked on multirobot coverage and single robot exploration. He is currently a research scientist and adjunct professor at the School of Computer Science, McGill University. His work focuses on human–robot interfaces for underwater vehicles, multirobot algorithms, and sensor networks. He also continues his work on autonomous planetary exploration and on-orbit servicing of satellites that he started during his work at the CSA in 2004–2007. His research has focused on mobile, space, and underwater robotics and, in particular, cooperating intelligent agents with application to multirobot cooperative localization, mapping, exploration, and coverage. His interests extend to computer vision, human–robot interfaces, and sensor networks. He has authored or coauthored more than 40 journal and conference papers in the aforementioned areas.

Erick Dupuis received his B.Sc.A. degree from the University of Ottawa, master's degree from the Massachusetts Institute of Technology, and Ph.D. degree from McGill University, all in mechanical engineering, with specialization in robotics. He joined the CSA in 1992 and is currently the manager of the robotics group in the Space Technologies Branch. He has been a research engineer in robotics and the lead systems engineer for the SPDM Task Verification Facility and for several projects in planetary exploration. He was also a member of the Jet Propulsion Lab's Mars Program Systems Engineering Team. He is currently a member of the space robotics working group of the European Technology Platform. His research interests have concentrated on the remote operation of space robots, starting with ground control of Canada's Mobile Servicing System on the ISS. He is now directing R&D on autonomous rover navigation for planetary exploration missions.

Address for Correspondence: Eric Martin, Space Technologies, Canadian Space Agency, 6767 route de l'Aeroport, Saint-Hubert, QC J3Y 8Y9, Canada. E-mail: Eric.Martin@space.gc.ca.