Toward Autonomy for Deep-Space Infrastructure

C. Langley*, A. Allen*, F. Kabanza**, P. Bellefeuille**

*MacDonald, Dettwiler, and Associates, Canada
e-mail: {chris.langley, andrew.allen}@mdacorporation.com

**Menya Solutions, Canada
e-mail: {kabanza, philipe.bellefeuille}@menyasolutions.com

Abstract

Future international deep-space missions, such as a Gateway outpost at the Earth-Moon Libration L2 point (EML2), will require advanced robotic servicing with a very high level of autonomy. Realization of this autonomy should also increase the efficiency of ground operations and analyses by making it easier to plan and monitor operations under various mission constraints. While a great deal of research has gone into studying the autonomy problem for planetary rovers and science spacecraft, the domain of robotic servicing beyond low Earth orbit is relatively immature. This paper reports the progress of an ongoing study aimed at developing and maturing enabling autonomous robotic servicing technologies for deep-space exploration. A software breadboard is being used to evaluate candidate technologies for the architecture and provide proof-of-concept to a Technology Readiness Level of TRL 3.

1 Introduction

1.1 The Study and its Goals

Future missions to establish infrastructure in deep space, such as an international outpost at EML2, will need to be autonomously serviced in order to enable long-duration maintenance in an unmanned state and to mitigate the risks caused by increased communication latency, dropouts, and bandwidth restrictions. At the same time, the human cost of operating this advanced autonomy must be minimized in order to meet increasingly demanding budget constraints. Under the guidance of the Canadian Space Agency (CSA), MDA and Menya Solutions are currently performing a technology development study to develop and mature the enabling technologies that will reduce the operational cost of future space infrastructure by significantly reducing the effort required by operations analysts to plan servicing tasks. Considerable cost savings can be realized over the lifetime of a flight system by making it easier to execute and monitor operations and by reducing the execution time and resource expenditure of the flight system.

Automated monitoring *in situ* will be necessary to provide the safety and reliability checks for a human spaceflight program, and will also enable the infrastructure to be maintained in an unmanned state. Applying autonomous control strategies to a large-scale, long-duration robotic servicing system that has comparable complexity to the Mobile Servicing System (MSS) on the International Space Station (ISS) will not be trivial. Canada's extensive ground control experience in robotic servicing of manned and unmanned space assets will provide a solid foundation for this effort.

Two of the main technical challenges in maturing the autonomous robotic servicing technologies are the choice of the appropriate mixture of human vs. autonomous control and the modeling of domain knowledge that is needed by planning and execution monitoring algorithms. Increasing the scope of autonomous operations leads to an increase in the number of system interactions (commands and telemetry) needed, with a commensurate complexity increase in the specification of the planning domain, and hence more sophisticated search-control knowledge is needed to maintain the efficiency of plan generation. Developing the trust of the operators and analysts in the autonomous system's capabilities is another key issue. A gradual increase in the operational level of autonomy is a key strategy to foster this trust.

At the start of the study, the concept of advanced autonomy for space robotic servicing was generally considered TRL 2 (technology concept and application formulated). The current study aims at maturing the concept to TRL 3 (analytical and experimental critical function and characteristic proof-of-concept) by retiring a number of the fundamental risks associated with automated planning and execution monitoring in this domain. Demonstrating that the core technologies can be used to solve a number of "straw man" test cases – defined using a simplified planning domain and executed and monitored using a primitive model of the plant – is an essential first step to building acceptance of advanced autonomy.

The remainder of this section discusses the EML2 mission concept and its requirements for robotics. Section 2 provides a brief overview of related work in autonomy for space exploration. In Section 3 the challenges of autonomous robotic servicing are discussed in terms of both the development of the technology and its adoption by the space community. Preliminary results from a proof-of-concept breadboard are discussed in Section 4. Finally, Section 5 discusses the near-term future work of the study and a potential roadmap for further maturation of the technology toward a flight demonstration mission.

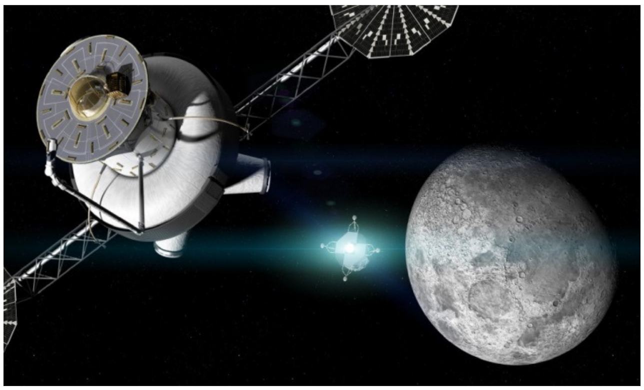

Figure 1: EML2 Gateway concept, from [2].

1.2 Mission Concept: EML2 Servicing Robotics

The primary mission concept considered for this study is a deep-space infrastructure at the second Earth-Moon Libration point, referred to as the "EML2 Gateway" (see Figure 1). The concept is described in a number of sources, from an earlier study [1] to more recent discussions [2][3][4] in the context of the Orion Multi-Purpose Crew Vehicle (MPCV), Space Launch System (SLS) development, and the potential for reuse of ISS componentry or technologies.

Essentially, the Gateway station will be assembled incrementally, in much the same way as the ISS was constructed, in a halo orbit around EML2. It will have a 15-year design lifetime, and must be maintainable in both a crewed and uncrewed state. As noted by Bobskill and Lupisella [2], "It is not always required that human crew be present for these activities. Robotic systems could operate autonomously, to some degree, or operations could be controlled from a remote site." The mission concept assumes both extravehicular (EVR) and intravehicular (IVR) robotics for a variety of tasks, including: inspection, orbital replacement unit (ORU) removal and replacement, construction (module and truss/superstructure manipulation), and likely for free-flyer capture and berthing and crew assistance during extravehicular activities (EVA).

2 Related Work

Autonomous space robotic servicing concepts have been experimented on in various unmanned and human-supported missions [5]. For the most part, these systems have been built around automated planning and verification capabilities. Automated planning algorithms provide the capability to autonomously generate robotic servicing plans (direct robot-command scripts or actions, or low-level operator procedures) governed by both resource and time constraints. Meanwhile, verification algorithms provide the capability to monitor and diagnose system faults and mission contingencies. Planning-and-execution architectures autonomously coordinate the planning, plan execution, execution monitoring, and fault diagnosis and recovery processes.

NASA Remote Agent Experiment (RAX) deployed on the Deep Space 1 spacecraft launched in 1998 successfully demonstrated the capability to autonomously plan onboard activities and correctly diagnose and respond to simulated faults in the spacecraft components [6]. More recently, NASA's Mars exploration rover 'Curiosity' has been made capable of autonomously generating and executing plans based on mission goals and can respond to contingencies by repairing the current plan [7].

Until now, autonomous agents in space have been mostly focused on deep space probes [8], Mars rovers [7][9][10], and Earth observation satellites [11][12][13]. Notably, autonomous robot servicing is completely absent from the MSS on the ISS. Robotic operations aboard the ISS are primarily conducted by ground operators, with some tasks still performed manually by the onboard crew [14]. The stakes with the ISS infrastructure are far higher than in the previously mentioned domains, not just because of the cost of the infrastructure itself, but also because of the astronauts onboard. The perceived safety concerns surrounding autonomous technology have outweighed economic and efficiency issues, so that rather than injecting autonomy, considerable manpower (with all its attendant opportunity costs) has been allocated to achieve the mission operations.

The long communications latency (3 to 4 seconds based on line-of-sight distance, and likely incrementally longer with signal routing) provides a greater push for on-board autonomy than in the case of robotics for low Earth orbit (LEO). The EML2 infrastructure will have to respond to unexpected conditions or system failures without input from terrestrial control centres. In this context, a lack of autonomy would actually incur safety risks to missions [3]. However, autonomous operations for this level of complexity and with limited crew size have not yet been demonstrated nor are they fully understood.

3 The Challenges of Robotic Servicing

There are many challenges underlying the development of an autonomous robot servicing concept for deep-space infrastructures such as EML2, particularly the domain complexity, the technology development, and the flight deployment.

3.1 Goals and Domain Complexity

The first step toward developing an autonomy architecture for deep-space robotics is gaining an understanding of the goals of the system and the domain in which it will operate. This analysis was performed by decomposing the robotic servicing needs of the EML2 mission into candidate autonomy tasks and functions as follows:

- ***EVR/IVR Inspection:*** Performing a static inspection or survey using the manipulator camera(s).
- ***Self-Reconfiguration:*** Relocation of any of the robotics elements (for example, by hand-over-hand operations or a rail-like base system), and mating / demating a dexterous small arm to a large manipulator.
- ***Construction:*** Berthing of visiting vehicles and installing or configuring infrastructure modules using a large arm.
- ***EVR/IVR Dexterous Manipulation:*** ORU removal and replacement, as well as manipulating any robotic-compatible hardware (e.g., torquing bolts, flipping switches).
- ***Free Flyer Capture/Release:*** Acquisition and tracking of an incoming visiting vehicle into the capture box, and servoing to capture; also performing the same tasks in the opposite order to release a departing vehicle.
- ***EVA Crew Support:*** Acting as a work platform or repositioning device for crew during EVA tasks.

In order to accomplish these mission goals, the robotic servicing system will need a number of capabilities which are beyond the current state-of-the-art in manned spaceflight, such as:

- Algorithms for initial acquisition of range/bearing and 6-DOF relative pose of a visiting vehicle.
- Algorithms for visual servo to alignment targets and berthing interfaces.
- Constrained path planning algorithms which plan a collision-free path from a starting pose to a goal pose, given full knowledge of the geometric configuration of the Gateway infrastructure, robotic system, and payload.
- Functions for powering / depowering and establishing data communications with a payload, as well as commanding the Gateway infrastructure to power / depower and mate / demate a module or ORU.
- Knowledge of the real-time positions and tasking of all crew members.

In addition, complexity factors for the deep-space robotics domain stem from an environment which is:

- ***Dynamic:*** Robotic operations involve dynamic movement of structures, data, and people, which can change the overall environment state.
- ***Partially observable:*** Many states in the system are unmeasured, or not telemetered, and/or communications are frequently disrupted.
- ***Stochastic:*** The interdependent nature of the systems and the sparseness of measurements results in a high degree of uncertainty in assessing the success or duration of a particular operation.
- ***Multi-agent:*** Space robotics systems are typically comprised of multiple robotic agents who will need to interact and cooperate to achieve complex goals.

The fact that any deep-space robotic servicing system will be embedded in an intricate arrangement of man-made systems is another source of complexity not seen in other autonomous space robotic systems, such as rovers and probes. The robotic system will have to be rigorously managed with strict rules for interaction in order to preserve safety.

3.2 Technology Development

As with other planning-based autonomous robot servicing systems, deliberative functions such as planning and verification require domain models of the agent and its environment. Such models include abstract (symbolic) representations of states, goals, and actions, as well as geometric representations and operations constraints. In particular, the models for an EML2 robotic system include the pose and operational status of every module, on-orbit replaceable unit (ORU), berthing mechanism, articulated structure (e.g., solar arrays and antennae), and EVA/IVA crew members, if present.

One of the ongoing challenges towards a complete definition of the EML2 planning models is to elicit underlying search-control knowledge, that is, expert knowledge related to how operations are conducted and which can be exploited by the search engine of a planning algorithm to cope with the state explosion typically associated with complex domains [15].

Planning at different levels of abstractions is another issue. In particular, at the mission level, a task can be achieved by a plan that schedules high-level procedures such as grasping a payload, moving a payload from one position to another, and releasing a payload. At the execution level, the arm grasp, release and motion operations are achieved by script-level programs which might be produced in part or in full through another planning process, including a motion-planning process. The current approach for handling such hybrid planning, in particular integrated task-planning and path planning, is a refinement of each action in a plan generated by the task planner (e.g., TLPlan [15]) by local plans generated by a lower planner (e.g., a path planner [16]). The long-term objective, however, is to also experiment with unified hybrid planning approaches, in particular those permitting integrated task-planning and motion planning [17].

Related to planning is the verification of safety and progress conditions during the execution of a plan, in particular the verification of action preconditions and postconditions. The robot's model of the mission (i.e., self, environment, mission goals, current internal state, current external state) must be comprehensive enough to reliably verify properties of the execution state, such that if a problem arises, it will necessarily be identified.

The components of the robotic servicing architecture responsible for execution must be implemented in such a way that there are no silent faults and that all important state changes are reported. This level of detail of state reporting is novel since traditionally the ground segment team monitors the execution progress using basic telemetry: success/failure codes, system states and modes, and sensor feedback. In order for the autonomous system to continue to work during loss of signal, it is necessary that one or more components of the system have sufficient rules and/or perceptual algorithms to convert the engineering telemetry from the mission system into the symbolic states and events which are relevant to the planning domain.

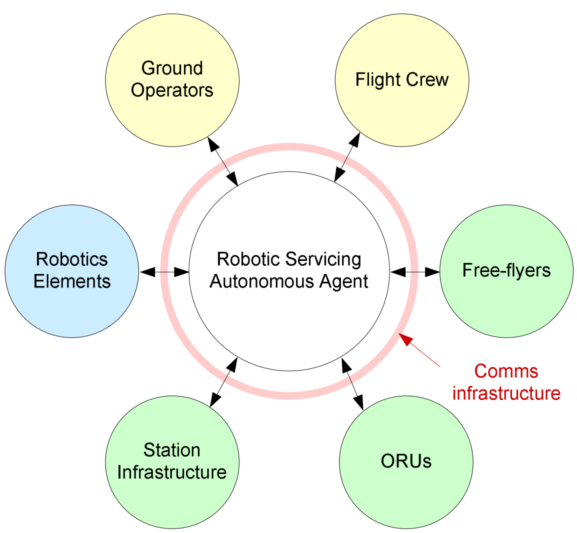

Figure 2: The robotic servicing agent will need to interface with many elements over a limited communications infrastructure.

The complexity of the data interfaces between the autonomous agent, the control software for the robotics, and the software controlling the rest of the deep space infrastructure is another major technical challenge (Figure 2). This is particularly true when, as on the ISS, multiple planners are attempting to fulfil their mission objectives under coupled constraints, for example, coordinating solar array articulation, attitude control burns, dexterous robotic operations, and availability of video monitoring on the ground [14]. To adequately manage these competing objectives, a "master" planner could consider all of these subsystems simultaneously; or, the robotics planner could take as input a timeline of tasks planned for external agents that have dependencies with the robotics. This timeline will need the expected start and end times of tasks as well as an evaluation of the likelihood and length of potential delays.

Perhaps the most critical aspect of the interfaces between the autonomous robotics and the rest of the station systems is the communications. The infrastructure and protocols must be designed for planned and unplanned loss of signal, and safety and contingency rules for the autonomous operations must explicitly address communications dropouts. If the robotics must have a reliable communications link to an external agent, the ability to synchronize, mutually report status messages, and allow mutual triggering of events will be mandated. The clearest example of this need would be during autonomous free-flyer capture. First, the station must provide the robotic system with an initial pose so that the latter can acquire and track the flyer using its end-effector camera. Next, the free-flyer must be given an Authorization to Proceed into the capture box of the manipulator. Confirmation of stationkeeping in the capture box triggers the free-flyer to disable its attitude control system, which in turn triggers the manipulator to servo to and grapple the spacecraft. If the autonomous agent detects a failure of the robotics at any time, it must be able to trigger the free-flyer to execute an abort manoeuvre.

3.3 Technology Deployment

The realization of autonomy for robotic servicing is not solely dependent on solving the technical challenges; indeed, gaining acceptance of autonomy by the community of stakeholders will likely require equally as much effort. The consequences of losing some or all of an international deep-space infrastructure, even in an uncrewed state, are significantly higher than the loss of any given rover, orbiter, or deep space probe. As such, any autonomous agent will be called upon to "prove itself" rigorously before it will be trusted to operate in flight. Some of the methods identified for winning this approval are outlined below.

The first step will be to ensure that autonomy is not "forced upon" the community right from the start. This means designing the mission systems with the automatic features which would be reasonably expected from the current state-of-the-art, without making them dependent on the implementation of an autonomous control system. Colloquially, this is "inserting" the agent between the human operators and the servicing system, such that the flight hardware elements need not be aware of whether they are under human control or autonomous control (or a mixture of the two). This assumption implies that the inputs to the autonomy software would be the same type and quality of telemetry which would be reported to a human-in-the-loop monitoring station.

The autonomy design will have to be simple enough that technical experts without training in intelligent systems can understand how it works. In particular, the design of the human-machine interface (HMI) must make it easy for the user to specify goals and constraints without knowing the formal language of the planner. As far as possible, the planner should communicate plans and intent to the user, for example by displaying the planned path before and during execution. Likewise, real-time, high fidelity, high frame rate visualization of both the flight environment and the internal execution state of the autonomous agent will need to be provided for monitoring. Finally, the agent should provide rationale for its choices, so that users do not see it as a "magical black box". Users, particularly those involved in monitoring, should be adequately trained (e.g., through virtual environments) with the automation in the loop, in order to build familiarity with both expected and off-nominal behaviours.

Forethought must be given to the gradual build-up of autonomy, both in the dimensions of operational risk as well as the level of autonomy. To address operational risk, the first tasks to be tested under autonomous control should be those with the least safety criticality. As trust builds, incrementally riskier operations can be performed. For example, in the EML2 scenario, the agent may be first tasked with executing a (non-contact) inspection survey. Next, the dexterous manipulation of ORUs for which there are both redundant units and redundant berths could be demonstrated. With greater trust the automated manipulation of unpressurized superstructure, and eventually pressurized modules, could be adopted. The riskier operations of free-flyer capture and EVA crew assistance would come only after trust in the autonomous system has been very well established. The second dimension is that of the level of autonomy: essentially how much of the execution authority is delegated from the human to the agent. The architecture should be designed to support mixed initiative operations, namely the ability to configure the operator-agent interaction to one of the levels described in Table 1. Again, as trust increases, the level of autonomy can be gradually increased.

For quasi-stationary tasks (i.e., where response time is not critical), low-level functions should be tested on the ground first, before implementing in the flight segment. Doing so should minimize the software development cost, particularly for validation of flight-worthiness. An example is servoing to a grapple fixture on a module already berthed to the station. While there are some oscillations in the manipulator end-effector pose due to dynamics, thermal distortions, and external perturbations, for the most part the situation is quasi-static. It should therefore be possible to run visual servoing in a look-and-move paradigm on a ground terminal. Here the visual tracking function would give the human-in-the-loop operator a set point to move to, based on its pose estimation function. Motion commands would start as fully manual, but as trust increases, these could eventually become semi-automated with an Authorization To Proceed (ATP). This allows ground to build confidence in the visual target tracking system before allowing it to close the feedback loop on the flight system.

Deliberation need not be tested based upon the real-time state of the deep-space mission. It could first be tested using logged telemetry from prior flight anomalies, such as "near miss" situations, or under conditions which the ground crew found particularly challenging from an operational or scheduling standpoint.

Table 1: Levels of automation in a mixed initiative system

Automation Level	Description
Manual	The function is performed entirely by the operator.
Validation	The function is performed by the operator but is monitored by an automated system which warns the operator of potential errors or dangers.
Collaboration	The function is performed through collaboration between the operator and the automated system.
Recommendation	Automated tools recommend a solution which must be explicitly accepted or corrected by the operator.
Operator Supervision	The function is automated but is monitored by the operator who has the ability to reestablish manual control at any time. The automated system pauses at appropriate times for a certain duration to allow the operator enough time to react.
Automated	The function is completely automated, but the operator still has the ability to pause or stop the execution of the function. This level can be thought of as a particular case of the operator supervision level with zero pause duration.

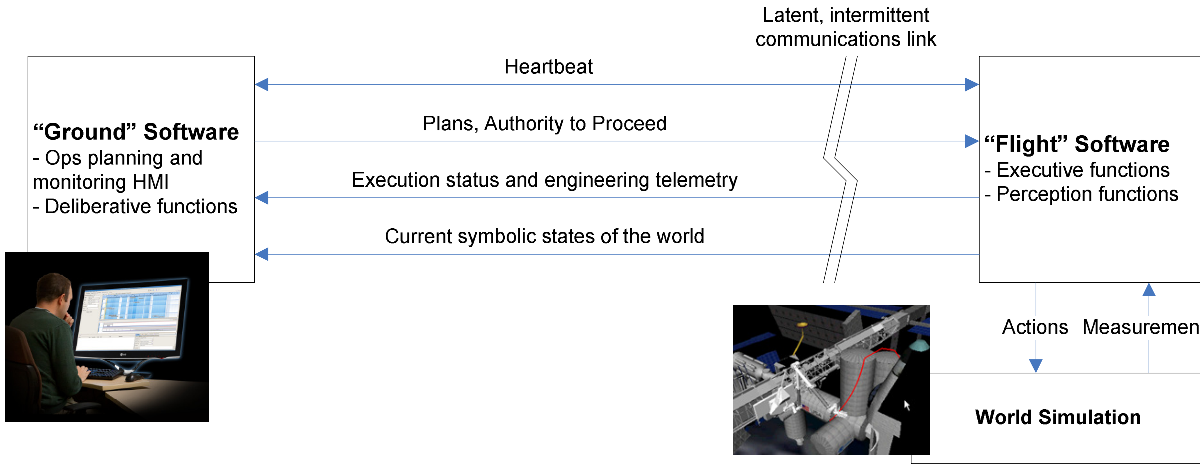

Figure 3: Proof-of-concept breadboard

4 Proof-of-Concept Breadboard

4.1 Concept

The study team is in the early stages of creating a proof-of-concept breadboard of an autonomous robotic servicing system to demonstrate some selected features of the architecture in a simulated environment (see Figure 3). Domain definition, mission specification, plan generation, execution monitoring, plan-repair functions, and latent/intermittent communications will be demonstrated.

This demonstration will help to identify the trade-offs that must be made between the expressiveness of a domain modelling language and the efficiency of a planning algorithm, the comparative efficiency of different planning algorithms in the EML2 domain, the strategies for planning at different levels of abstraction, and the conceptualization of mixed initiative planning and execution monitoring strategies. The challenges of converting a symbolic plan to an executable representation, and the related problem of mapping sensor telemetry to a symbolic perception of the world must also be addressed. Simulating delayed and intermittent links also drives out early requirements for the communications and contingency protocols to ensure safe and correct behaviours. An understanding of these numerous issues could not be developed solely based upon a literature review.

As a stand-in for the EML2 Gateway and its robotic servicing system, the breadboard will use an existing model of the ISS and the MSS (Figure 4). Unlike LEO, however, a four second (or higher) roundtrip delay for communications between ground and flight systems will be simulated, which is a close approximation of the best case scenario for an outpost at EML2.

The domain will contain a large robotic arm (based on the SSRMS), modules that can be grasped and moved by the arm and can be berthed to or un-berthed from the station, grapple fixtures that the arm's end effector can grasp, a mobile platform allowing the arm to move between locations on the station, and cameras on both the station and the arm.

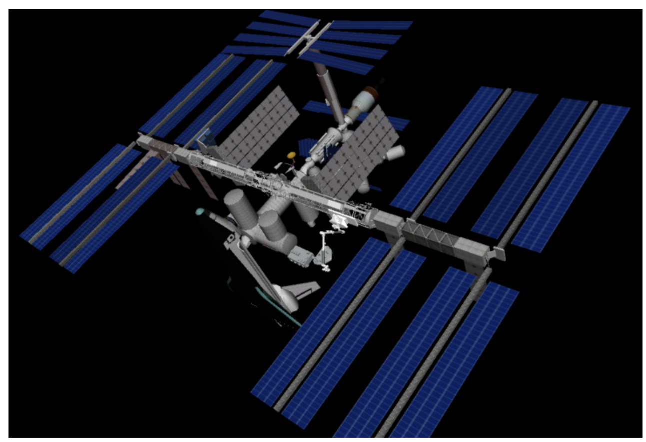

Figure 4: ISS environment simulated in the breadboard

The autonomous robotics will be able to execute the following functions: move to a given position (specified either in joint or Cartesian space, using the actual arm control logic of the SSRMS), grasp or release a grapple fixture, berth or un-berth a module to the station, power or un-power a module, as well as control the pan/tilt/zoom of cameras.

Powered by the TLPlan planning algorithm [15] integrated with a probabilistic roadmap path planning algorithm [16], the breadboard will support missions with multiple goals, temporal relationships between these goals (e.g., complete goal A at least 2 hours before goal B), constraints on goals and actions, and objective functions.

Constraints can be time related, communication related, resource related, robotics related, operational (e.g., visibility constraints for operators monitoring through outside camera views), or convey operator preferences. The executive will translate symbolic plans into executable data structures (e.g., using the open source PLan EXecution Interchange Language (PLEXIL) framework [18]), with the flexibility to support mixed initiative through configurable ATP points in the plan. It will monitor feedback from the simulation environment (where hardware faults and other off-nominal conditions can be injected), and perform at least one local fault detection and recovery function without the need to involve the deliberative functions on the ground.

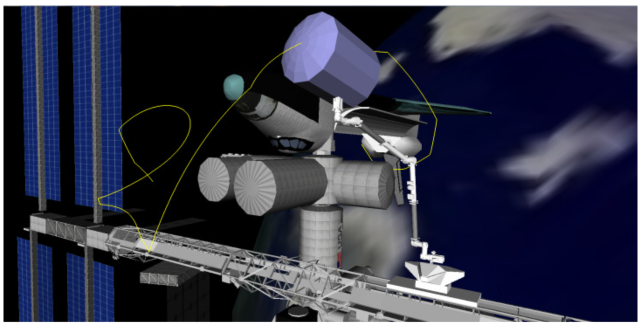

Figure 5: Close-up view of a planned path (yellow line)

4.2 Current Breadboard Development Status

The current implementation has a deliberative layer that integrates TLPlan with the path planner in order to demonstrate automated task planning and constrained path planning for goals, such as moving and berthing modules (Construction), inspecting elements of the station (EVR Inspection) and moving the robot arm (a subset of Self-Reconfiguration). High-level constraints such as precedence, visibility, communication, and collision and singularity avoidance are already modeled. Lower-level robotic constraints such as load and force limits are currently not modeled although mature technologies already exist to handle these functions.

The executive is currently able to run plans through the PLEXIL executive to issue commands and receive live telemetry from the simplified models of the robotics and station infrastructure. The 3D environment (see Figure 5) can be driven by the simulation states, and any collisions detected and reported. Off-nominal events (such as unexpected loss of communication or faults in the robotics or berthing mechanisms) will be injected during execution to exercise the breadboard's local fault recovery and re-planning capabilities.

The communications linking the deliberative and executive functions are under development at the time of writing, but are being designed to monitor the link status and automatically reconnect when signal is acquired. During loss of signal, the executive will modify its behaviour so that tasks which require monitoring or ATP are not initiated until the link with ground is re-established.

With the current models and implementation, the breadboard is able to demonstrate autonomous planning capabilities for a mission within the domain described in Section 4.1, for simultaneous goals, while considering temporal constraints on goals and restrictions on certain actions based on communication channels availability (i.e., certain actions require constant communication with ground control). The system can also consider a subset of the constraints from other systems in the form of additional time constraints.

One of the scenarios that the breadboard can demonstrate requires the robot arm to take a module from the cargo vehicle, berth the module to a specific location on the station, and inspect another location on the station. After these goals are achieved, the arm must return to a specified configuration. In addition, a loss of communications is expected for a particular time window during the mission. The system must complete this mission under several constraints, for example the module must be installed within an hour of being taken out of the cargo vehicle (a thermal constraint) and the module must be taken out of the vehicle before a specific time to allow enough time to prepare the vehicle for its upcoming departure (constraint from an external system).

The breadboard is able to find a solution to this scenario while considering a subset of the critical safety constraints specific to the robot servicing domain. It must also consider the communication delay between ground and flight when dealing with off nominal events.

The current breadboard is able to verify on the fly that the plan being executed satisfies safety and mission constraints. Apart from verifying that the current robot trajectory is free of collision and avoids singularities, the breadboard is able to ensure that it is within the best fields of view of the station's cameras available to the operator and that it avoids areas identified by safety review as presenting high risks of damage to the arm or the station components. The breadboard also verifies that the expected and actual states of the world are consistent. When the system detects an issue (a constraint is about to be violated, or the actual state does not correspond to the expected state), re-planning is triggered to find a new plan that satisfies the new situation. The verification of lower level robotics constraints, power usage, and lighting conditions are not currently supported.

The breadboard currently supports four levels of automation for planning, execution and verification:

manual, validation, recommendation and automated. For instance, in the case of task planning, a human operator is capable of providing his own plan (manual), which can optionally be verified using the verification tools and algorithms described earlier (validation). The system can also provide a plan which can then be accepted or modified by a human operator (recommendation). Alternatively, the plan can be executed immediately (automated).

5 Conclusions and Future Work

Autonomous robotic servicing in deep space is still an open problem in the literature. The present study is making progress toward understanding the unique considerations and constraints of this domain, as well as designing a preliminary architecture for its realization. While the study is ongoing at the time of writing, early results from the software proof-of-concept indicate that the technology is soundly at TRL 3.

There are several potential follow-on options for furthering deep-space servicing autonomy. Hardware-in-the-loop demonstrations can be performed using a ground testbed such as the Next Generation Canadarm (NGC). Further maturation of the software can be achieved by allowing the autonomous agent to command the System Avionics Integration Facility (SAIF) ground hardware emulation of the SSRMS. Future validation experiments could take place using the MSS in a TRL 7 demonstration test objective (DTO) mission.

Acknowlegdments

This work has been funded by the Canadian Space Agency's Space Technology Development Program. The authors wish to thank the CSA reviewers (Régent L'Archevêque, Kumudu Jinadasa, Yves Gonthier, Leslie Sponder, and Sarmad Aziz) for their guidance and insightful feedback, and acknowledge the efforts of the study teams at MDA (Justin Allport, Richard Rembala, Ben Chan, and Maggie Chen) and Menya Solutions (Simon Chamberland, Julien Filion, and Leo Hartman).

References

[1] J. Geffre. Lunar L1 gateway study. Technical Report EX15-01-001, National Aeronautics and Space Administration (NASA), 2001.

[2] M. R. Bobskill and M. L. Lupisella. The role of cis-lunar space in future global space exploration. Technical Report GLEX-2012.05.5.4x12270, National Aeronautics and Space Administration (NASA), 2012

[3] R. Ambrose, et al. Draft robotics, tele-robotics and autonomous systems roadmap. Technical report, National Aeronautics and Space Administration (NASA), 2010.

[4] M. Raftery and A. Derechin. Exploration platform in the earth-moon libration system based on ISS. In Proc. International Astronautical Congress, IAC-12-B3.1, 2012

[5] A. Jonsson, et al. Autonomy in Space Current Capabilities and Future Challenges. AI Magazine, 28(4):27-42, 2007.

[6] D. Bernard, et al. "Spacecraft autonomy flight experience: The DS1 Remote Agent experiment." In Proc. American Institute of Aeronautics and Astronautics (AIAA), 1999.

[7] T. Estlin, et al. "Increased Mars rover autonomy using AI planning, scheduling and execution." In Proc. IEEE International Conference on Robotics and Automation (ICRA), pages 4911–4918, 2007.

[8] A. Schwab, et al. "OBCPs-an integrated part of the BepiColombo autonomy and flexibility." In Proc. International Conference on Space Operations, 2012.

[9] T. Estlin, et al. Enabling autonomous rover science through dynamic planning and scheduling. In Proc. IEEE Aerospace Conference, pages 385–396. 2005.

[10] A. Ceballos, et al. A goal-oriented autonomous controller for space exploration. Astrophysics and Space Sciences Transactions (ASTRA), 11, 2011.

[11] G. Rabideau, et al. "Iterative repair planning for spacecraft operations using the ASPEN system." In Proc. Int. Symp. on AI, Robotics and Automation in Space (i-SAIRAS), volume 440, page 99, 1999.

[12] B. D. Smith, et al. "The RADARSAT-MAMM automated mission planner." In Proc. Innovative Applications of Artificial Intelligence (IAAI), 2001.

[13] S. Chien, et al. "The EO-1 autonomous science agent." In Proc. International Joint Conference on Autonomous Agents and Multiagent Systems, Volume 1, pp. 420-427, 2004.

[14] S. Aziz. Development and verification of ground-based tele-robotics operations concept for dextre. In Proc. International Astronautical Congress, pages IAC–10.B6.1.1, 2010.

[15] F. Bacchus and F. Kabanza. Using Temporal Logic to Express Search Control Knowledge for Planning. Artificial Intelligence , 116(1-2):123-191, 2000.

[16] K. Belghith, et al. Anytime Dynamic Path-Planning with Flexible Probabilistic Roadmaps. Proc. of IEEE International Conference on Robotics and Automation (ICRA), pages 2372-2377, 2006.

[17] L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. International Journal of Robotics Research (IJRR), 2012.

[18] Universities Space Research Association. PLEXIL. http://sourceforge.net/apps/mediawiki/plexil/, last accessed April 23, 2014.