The Challenge of Planning and Execution for Spacecraft Mobile Robots

Gregory A. Dorais and Yuri Gawdiak

NASA Ames Research Center
MS 269-2
Moffett Field, California 94035
gdorais@arc.nasa.gov, ygawdiak@arc.nsa.gov

Abstract

The need for spacecraft mobile robots continues to grow. These robots offer the potential to increase the capability, productivity, and duration of space missions while decreasing mission risk and cost. Spacecraft Mobile Robots (SMRs) can serve a number of functions inside and outside of spacecraft from simpler tasks, such as performing visual diagnostics and crew support, to more complex tasks, such as performing maintenance and in-situ construction. One of the predominant challenges to deploying SMRs is to reduce the need for direct operator interaction. Teleoperation is often not practical due to the communication latencies incurred because of the distances involved and in many cases a crewmember would directly perform a task rather than teleoperate a robot to do it. By integrating a mixed-initiative constraint-based planner with an executive that supports adjustably autonomous control, we intend to demonstrate the feasibility of autonomous SMRs by deploying one inside the International Space Station (ISS) and demonstrate in simulation one that operates outside of the ISS. This paper discusses the progress made at NASA towards this end, the challenges ahead, and concludes with an invitation to the research community to participate.

Introduction

In order to robustly achieve increasingly ambitious mission goals for longer periods with less ground support than traditionally required, we expect future space flight projects to increasingly require advanced onboard autonomy to support both manned and unmanned missions. Moreover, autonomously-controlled mobile sensors and manipulators (that can be encapsulated in a SMR) can provide additional capabilities and productivity that would otherwise require greater mission cost or risk.

Sensing Tasks

Generally, sensing tasks are viewed as more readily achievable than tasks that require sensing and manipulation. As such, the systems that we are initially developing are spacecraft robots restricted to mobile sensing and this paper is restricted to discussing planning and execution of such robots. Consider a SMR (a mobile robot with a variety of sensors) that can operate within a spacecraft such as the International Space Station (ISS).

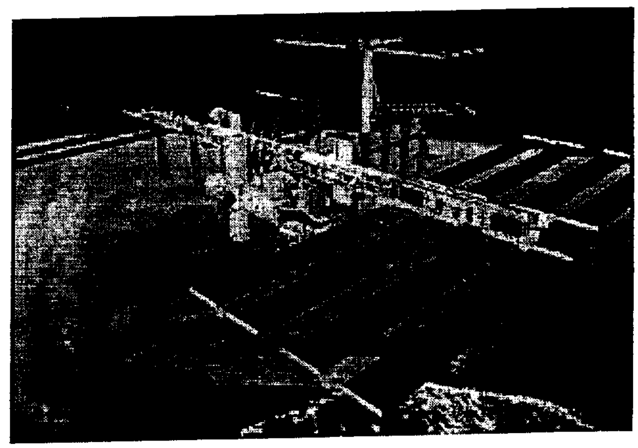

Figure 1: International Space Station Illustration

Such a robot could potentially perform a number of tasks such as:

- Measuring and localizing toxic gases. In former Russian MIR space station, there was concern that batteries might leak sulfur dioxide. During a fire on MIR, toxic gases were released. In both cases it would have been helpful to have a SMR measure and, if necessary, localize the source of such gases.
- Measuring changes in pressure and ratios of nominal gases, e.g., oxygen and carbon dioxide. The first crew of the Salyut 1 station all tragically died by suffocation when a valve failed on its return vehicle. Fixed sensors can also fail or not be available during a crisis such as happened on MIR when a collision caused a loss of cabin pressure down to ~600mb, far below the safety level. A SMR can provide early warning of anomalies and be a redundant, portable system during a crisis.
- Validate fixed environmental sensing systems. In event that an anomaly is detected by the ISS life support system, there exists the possibility that the problem is a fixed sensor and not the environment. A SMR can be autonomously deployed or controlled by Mission Control to validate if the sensor is defective or not. If the sensor is defective, the SMR can act as a virtual sensor until the

fixed sensor is replaced. If not, the SMR help isolate the source of the anomaly.

- Visually validate regions of the spacecraft. Multi-spectral cameras on a SMR can provide crew members, Mission Control, and scientists a visual record of anything from a piece of equipment to a crew activity, to a science experiment, without tying up a crew member to perform the task.
- Perform time-consuming special monitoring tasks. Specially-equipped SMRs can be deployed to specific tasks such as measure or localize certain sounds. Detecting unusual sounds is a method often used by people to diagnose a failing piece of equipment. Also, small leaks can be detected by the sound they emit. An autonomous SMR can isolate and localize particular sounds that human ears cannot detect.

An example of a task for a SMR operating outside a spacecraft is:

- Detecting external spacecraft damage. Astronaut EVAs are risky and time consuming. As a result, monitoring tasks such as checking the Shuttle for tile damage prior to reentry and looking for micrometeorite damage on ISS are not routinely performed. Once extended to remote spacecraft, failure assessment alone is of enormous value. Extraordinary effort is made to determine failure causes often with low confidence due to lack of data.

SMR and Terrestrial Mobile Robot Comparison

Although there are many similarities between SMRs that operate in engineered dynamic environments that may include people, and mobile robots that operate in natural terrains other than Earth, there are also striking differences that present challenges for SMRs including:

- Operates in close range in complex, dynamic, structured environment in 3 dimensions.
- Recognizes, and in some cases manipulates, many engineered objects
- Observes nominal and diagnoses off-nominal situations
- Interacts with people in a number of ways:
  - People are commanders (at various levels of authority to command at various levels of autonomy)
  - People are agents instructed by robot to achieve goal
  - People are dynamic obstacles to avoid
  - People are dynamic objects to track
  - People are peers to collaborate on achieving joint goals

These tasks and the operational environment levy a number of requirements on the planner(s) used to achieve such tasks over an extended period:

- Mixed-initiative task planner/scheduler
- Mixed-initiative path planner
- Local obstacle avoidance path planning
- Resource management
  - Power & Energy
  - Momentum
  - Thermal power management
  - Battery-life
- Multi-agent state estimation and control (people, SMRs, in-situ systems)
- Reactive planning and adjustably autonomous control
- Real-time planning and execution

Spacecraft Mobile Robots at NASA

NASA has begun to address the need for SMRs and the above challenges. Currently, two spacecraft mobile robots in particular are under development at NASA, the Personal Satellite Assistant (PSA), and the Sprint AERCam. This paper will focus on the PSA all many of the issues and technologies are relevant to both.

Personal Satellite Assistant (PSA)

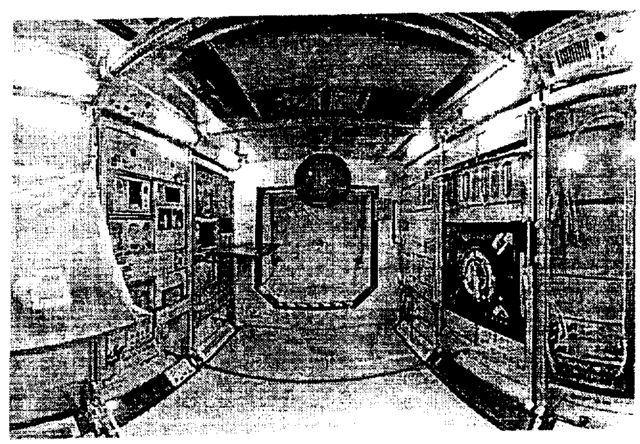

Figure 3: PSA Prototype depicted in ISS Node Mockup

The PSA is being designed as a softball-sized flying robot that operates autonomously onboard manned and unmanned spacecraft in micro-gravity, pressurized environments, and in particular onboard ISS. PSA's hardware architecture is being designed to accommodate a wide range of components that enable a broad set of mission support scenarios. Environmental sensors for gas, temperature, and pressure provide the ability for the PSA to monitor spacecraft for abnormal conditions, e.g., overheating equipment, payload and crew conditions. Video and audio interfaces will provide support for navigation, remote monitoring and video-conferencing. A radio frequency identification tag reader/writer and/or bar-code reader on the PSA will enable it to recognize specific objects and update their location in an inventory control system. Ducted fans/blowers will provide propulsion and batteries will provide portable power. An auto-docking locker will enable the PSA to autonomously recharge its batteries and provide a secure storage location when not in flight. The PSA will be connected by a wireless network to a laptop computer that will provide a user interface with the crew and to a server for additional information processing capacity, primarily for PSA planning A speech interface and dialogue management system for the PSA will permit spoken language commanding and data queries of the PSA and databases that the PSA has access to via its wireless network. A long-range goal for the PSA is to connect it via the wireless network to the spacecraft's avionics data, payload networks, and uplink/downlink communications.

The main benefit PSA is expected to provide is for it to act as a crew work-force multiplier by performing intra-vehicular activities on behalf of the crew. Current spacecraft are constrained in terms of crew size, power, volume, and computing resources. Crew time on the International Space Station is one of the most constrained resources and is projected to cost hundreds of dollars per minute per astronaut. The crew will have to maintain complex critical ISS systems, perform dozens of major simultaneous payload experiments, and perform general housekeeping. Enhancing the crew's ability to perform their duties is critical for successful, productive, and safe space-based operations. Moreover, PSA can enhance crew safety by performing monitoring tasks that might endanger a crewmember or not otherwise be performed.

The PSA's autonomy capabilities are expected to significantly improve productivity by directly supporting flight crews, ground controllers, and the principle investigators of science experiments. The biggest benefits to those users will come from its ability to monitor the environment, e.g., detect abnormal concentrations of CO^2, act as a mobile camera/camcorder/data terminal, and track inventory using advanced inventory micro-tags. For example, when the PSA detects a sharp pressure drop while performing an inventory audit, it would then notify the crew of the abnormal condition and attempt to localize it. If however, a fixed sensor on ISS detected a pressure drop, the PSA could be used to validate the reading. If the sensor is diagnosed as defective, the PSA could act as a temporary replacement sensor. We expect this entire activity could be conducted without the need for human intervention or be initiated by the ground operators, onboard crew, or the spacecraft itself.

The PSA will provide an additional side-benefit by acting as an autonomy and mobile robot testbed for researching intra-vehicular robots that eventually will be used for long-term missions, e.g., operating onboard a crew return vehicle orbiting Mars for two years while the crew explores the surface.

PSA Operational Requirements

In order to support the development of suitable autonomous control system for the PSA, the following subset of operational requirements were defined:

1. Achieve set of 10 commands in an optimal sequence where each command is to take a picture and environmental sensor reading at a global <x, y, z, yaw, pitch, roll> specified immediately prior to execution. Perform in each of the following ISS Node environments:
Environment A: uncluttered, static
Environment B: known clutter, static
Environment C: unknown clutter, static
Environment D: unknown clutter, dynamic

2. Validate two environment fixed-sensors. For example, go to the location of a fixed sensor indicating high temperature and measure environment. If the fixed sensor is accurate, localize the source of the heat. If the fixed sensor is not accurate, station-keep at the fixed sensor location transmitting temperature readings until the fixed sensor readings are accurate then return to base locker.

3. Demonstrate mixed-initiative planning for both path and deliberative planning. This shall include:
a. Adding temporary constraints to change an existing plan
b. Adding goals to an existing plan
c. Rejecting goals in an existing plan
d. Rejecting goals from a plan that fails to converge

4. Demonstrate mixed-initiative execution. Includes allowing human interrupts and command additions, retractions, & modifications as well as asking humans or other agents for assistance during execution. Levels of autonomy to be demonstrated:
a. High-level teleoperation
b. Guarded & guided teleoperation
c. Dynamic commanding of PSA by human
d. Dynamic commanding of PSA by another agent
e. Dynamic commanding of human by PSA
f. Dynamic querying and modification of plan currently being executed
g. Executing and modifying generated plan due to environment uncertainty

5. Demonstrate teleconferencing. Includes face-tracking.

6. Demonstrate crew following. Includes body-tracking.

7. Demonstrate energy resource management including dynamic auto-recharging.

8. Demonstrate leak isolation using acoustics and a leak isolation expert agent.

9. Demonstrate spoken language commanding and status reporting.

10. Demonstrate inventory sensing and location tracking.

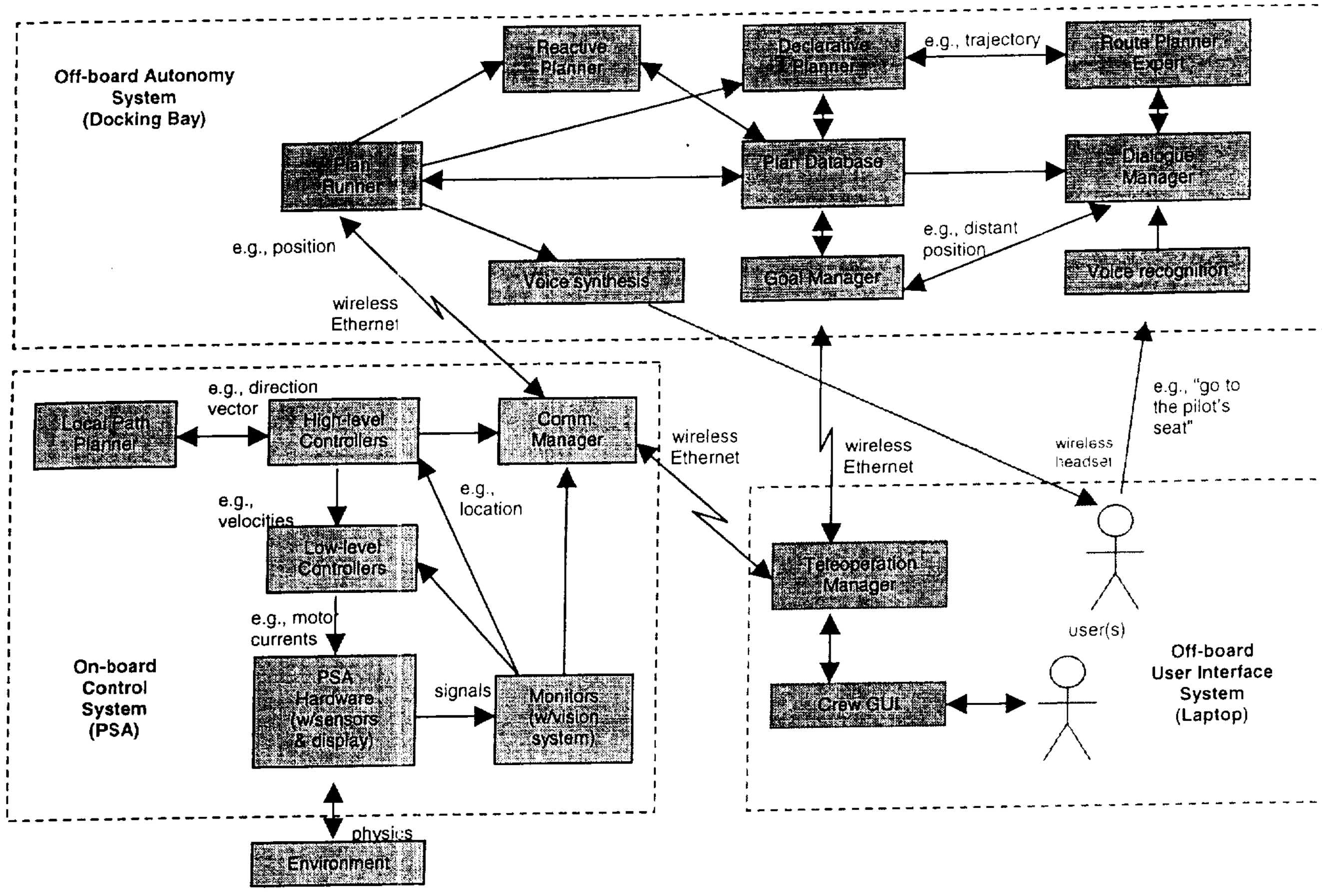

Figure 4: PSA Top-level Autonomy Architecture

PSA Autonomy Control Architecture

A prototype autonomy control architecture, illustrated in figure 4, has been developed to address the operational requirements. The architecture implementation was distributed over three processors as depicted by the dashed boxes:

- Onboard flight processor for sensing and real-time control. Software for localization to a global map, object recognition, and obstacle avoidance using stereo vision and other proximity and inertia sensors is executed here.
- User-interface laptop for commanding and displaying information. This includes interfaces for interactively creating and modifying the plan and teleoperation. Our intent is for this interface to support operation at various autonomy levels that can be dynamically changed and range from teleoperation to high-level autonomous control.
- Off-board docking bay processor for high-level autonomous control including planning, scheduling, command sequencing, and human and other agent communication and coordination.

The high-level autonomous control system, depicted by the top dashed box in figure 4, is a planning and execution system in its own right based on the unified agent framework described in [Muscettola et al. 2000]. This agent is composed of the following subsystems:

- **Plan Database**

This is a temporal, constraint-based network of tokens that defines the past, the present, and flexibly-defined future states and actions of the system. Each token represents the "state" of a state variable for a period of time. The token data structure is a tuple that specifies the state variable, the procedure and its arguments that is invocated when the token is "executed," and the token start and end time bounds. The plan database supports multiple timelines with constraints on and between tokens. If none of the constraints are violated for a given instantiation of the plan database, the database is defined to be consistent. The current implementation uses a next-generation plan database of the Remote Agent plan database described in [Jonsson et al. 2000], which was part of the Remote Agent control system demonstrated on the Deep Space One spacecraft in 1999 [Bernard et al. 1998].

- **Plan Runner** (command sequencer)

The plan runner is a process responsible for "executing" tokens in the plan database at the appropriate time.

Executing a token involves calling the procedure with its arguments defined by the token, updating the plan database with the token return values when the procedure terminates, constraining the plan database so that planners only have limited ability to change the past, and calling planners, described below, as needed to update the plan database. The plan runner implemented is described in more depth in [Muscettola et al. 2000].

- **Planners**

This architecture support the integrated use of a number of planners so that planners can be specialized for various functions depending on the domain requirements. For the purposes of this paper, with the exception of the plan runner, a planner is any process that modifies the plan database or provide information to be added to the plan database at the request of a planner. The planners in this implementation include:

1. Declarative Planner

The declarative planner is based on the Remote Agent Planner/Scheduler described in [Jonsson et al. 2000]. It is responsible for generating a consistent, flexible plan in the plan database given a start and end horizon time bound, an initial state of the timelines at the start time, and a set of goals. A flexible plan is loosely defined as a set of timelines, each consisting of tokens on each timeline, token order constraints that prevent overlapping tokens on the same timeline, and token procedure variable constraints. Plan flexibility is characterized by the set of decisions yet to be made in a plan database that is consistent. The declarative planner is called to initialize the plan database and also is called during plan execution as specified by the plan being executed. It is typically called to plan for a period of significant duration sufficiently in the future such that the deliberative planner will complete prior to the start time of this period, but not so far in the future that the initial state at the future start horizon is not known with high confidence.

2. Reactive Planner

The reactive planner is also based on the Remote Agent Planner/Scheduler described in [Jonsson et al. 2000], but typically uses different heuristics. It is regularly called by the plan runner to insure that the plan database is consistent after token return values are posted to the database (repairing the plan as necessary), to insure the database contains a token on each timeline being executed or to immediately start executing, and to remove any ambiguity in whether a token is ready to execute and what its procedural arguments are.

3. Goal Manager

The goal manager essentially acts as a meta-planner for the declarative planner. As stated above, the declarative planner requires a start and end horizon time bounds, an initial state of the timelines at the start time, and a set of goals. The goal manager interacts with the user to determine this information. This may include negotiation of goals when all goals are not achievable or supporting mixed-initiative planning for hypothetical situations.

4. Route Planner Expert

The route planner expert is called by any one of the above planners to determine the time, route, and energy required to move between two points in the environment or to cover a certain space. It has access to a global map that can be updated with sensed obstacles. A route plan request is typically made by the deliberative planner as part of developing the initial plan, but may also be called by the reactive planner to develop an alternate route if necessary, e.g., the route is blocked or there is insufficient energy to complete the current plan. In addition, a user may initiate a request to answer a hypothetical question about a particular goal.

- **Spoken Language Interaction**

A simplified abstraction of the spoken language interaction system can be viewed as consisting of the following three subsystems:

1. Dialogue Manager

The dialogue manager is responsible for acting as an intelligent interface between a person speaking a restricted natural language and the planner modules along with the plan database. New goals can be inserted or removed in the plan database, and queries can be made by spoken commands.

2. Voice Recognition

The voice recognition subsystem essentially converts an audio signal into a parsed text stream. In the past, we have used commercial products to accomplish this. We anticipate that we can continue to use such products, upgrading them as improvements are made. However, it may be necessary to filter the audio signal for noise.

3. Voice synthesis

Conversely, the voice synthesis subsystem essentially converts text to speech. Similarly, we use a commercial product for this purpose.

Current State of the PSA Project

The PSA project began in 1998 and according to the current project schedule, the PSA begins flight operation in 2006. At this time, an oversized version of the flight model has been developed and is being tested on a granite table and is supported by a test stand with a compressor that enables the prototype to float on a thin cushion of air. On this test facility, we have demonstrated visual-servoing to various locations as well as vision-based localization to a global map. A 3D test facility that will house a full-size station node mockup is nearing completion. With the aid of a crane-like support mechanism and gimble, the PSA prototype will be able to move in 6 degrees-of-freedom (DOF), i.e., (X, Y, Z, yaw, pitch, roll) as if it were in a micro-gravity environment. The facility will also enable crewmembers to interact with the PSA in this environment while being suspended by a sling. A next-generation version of the prototype is also under development and is scheduled for testing in 2003.

In addition to the physical hardware for testing, a simulator has been developed. The simulator primarily reads the force commands generated by the controller and moves the PSA in an ISS module accordingly. It also provides simulated PSA sensors signals, e.g., vision, temperature, at various fidelities depending of the required tests. Although, the simulator is typically operated in force mode, it can also be operated in velocity or position modes when it is desirable to interact directly with high-level control systems. The PSA motion along with dynamic obstacles and in-situ crewmembers are rendered in 3D. The simulator also supports multiple PSAs. In addition, the simulator supports scripted environmental events, such as a fire.

An initial version of the spoken language interaction system has been developed and tested with a simplified PSA simulation. The system has also been integrated with the plan database such that the database can be queried and modified in simple ways in response to spoken commands.

An initial version of the autonomous control system has also been developed and deployed, although certain modules, namely the goal manager and the route planner expert have been stubbed at this time. Although currently the reactive planner has been integrated and used by the system to accomplish simple scenarios, scenarios involving plan repair are not scheduled until later this year.

Sprint AERCam

In contrast to the PSA, the AERCam is being designed to operate in unpressurized regions, essentially outside spacecraft, primarily the ISS. However, in many other respects the planning and execution challenges are similar to those faced by the PSA.

Figure 2: Sprint AERCam during 1999 Flight Test

The Sprint AERCam is a teleoperated, free-flying spherical robot. It weighed about 35lbs and was 14" in diameter. It had 12 nitrogen-gas thrusters, each producing about 0.08lbs of thrust, for propulsion and attitude control. It was designed to operate for about 7 hours outside of and near spacecraft at low velocities relative to the spacecraft, less than 30cm/s. Its primary mission sensors are two color video cameras. Its primary function is to provide video supporting a crew extra-vehicular activity (EVA) or perform reconnaissance in lieu of an EVA. The Sprint was successfully flight-tested for about 30 minutes on the STS87 space shuttle flight in 1999.

Two limitations of AERCam are its size and the teleoperation requirement. In order to address these limitations, a mini AERCam is being developed and efforts have begun to develop an autonomous control system that will enable it to be autonomously controlled at levels varying from entirely teleoperated to entirely autonomously controlled.

The PSA and AERCam projects are coordinated so that they can leverage each others technologies, but it remains to be seen the extent that the autonomy architectures will be similar due to different operational requirements.

Challenge: Spacecraft Mobile Robot Scenarios

In order to measure the system capabilities with reference to the operation requirements and to identify the challenging problems, several scenarios have been developed. These scenarios were designed to be executed both in simulation as well as with the prototype hardware in the test facilities. The current scenarios that the system is being designed to address are:

Scenario A: Robust generation of an ISS node environment map

Description:
PSA will create an environment map of the ISS node by traversing the space in a serpentine path recording the environment sensor readings along the way. During this activity, its path will be blocked by static obstacles (some of which are known of ahead of time) and moving obstacles. At one point the PSA will be interrupted to be teleoperated and then perform a station-keeping task at a location specified by an ISS Rack Locker name, after which it will complete its original environment-mapping task.

Purpose:
- Demonstrate navigation to several waypoints in an environment that has static and dynamic obstacles.
- Demonstrate mixed-initiative execution including autonomous task interruption and resumption, guarded teleoperation, and visual servoing by command.
- Demonstrate generation of a near-optimal 6-DOF route plans
- Demonstrate obstacle detection and avoidance
- Demonstrate stereo vision-based 6-DOF localization and map registration

Scenario B: Participate in the diagnosis and recovery of an ISS node fault

Description:

A fixed sensor in the ISS node signals a high temperature to the Environmental Control Life Support System (ECLSS). However, it is not known whether the sensor is defective or the source or the heat. PSA is given a command by ECLSS to go the fixed sensor location and verify the temperature at that location. If PSA confirms the fixed sensor is correct, PSA is to locate the heat source and signal the source to ECLSS, will then power down the locker at that location. Once PSA verifies that the temperature has returned to normal, it returns to its docking bay. If the fixed sensor is not correct, PSA is to stay at that location until the fixed sensor is made operational. Once PSA verifies the sensor, PSA returns to its docking bay.

Variation Summary:

1. Perform with faulty fixed sensor
2. Perform with overheating locker

Purpose:

- Demonstrate IVHM
- Demonstrate cooperative multi-agent planning and execution
- Demonstrate generation of a near-optimal 6-DOF route plans
- Demonstrate stereo vision-based 6-DOF localization and map registration

Scenario C: Fault Detection and Cooperative diagnosis of an ISS node atmosphere leak

Description:

PSA is commanded to perform a routine task to monitor an ISS locker. While en route, PSA detects a drop in pressure in the node. It interrupts its current task and performs a set of directional microphone sensor readings to determine the cause is a leak to space and then PSA isolates the general location of the leak. PSA reports this information to ECLSS, which then dispatches and external SMR (AERCam) to the general location outside station where it images the region of the leak to get visual confirmation.

Purpose:

- Demonstrate autonomous IVHM
- Demonstrate dynamically changing plan to respond to fault detected in the environment
- Demonstrate multi-agent cooperative diagnosis

Scenario D: Cooperative Data Collection and Crew Instruction for Performing Interactive Mission Science Experiments

Description:

Crewmember commands PSA to follow the crewmember to an ISS rack where the crewmember will perform an experiment. When the crewmember arrives, he/she commands PSA to point at the locker where the crewmember will work. The crewmember commands PSA to start recording the video and audio. The crewmember then commands PSA to brief him/her on experiment X then instruct him/her on the first step of the experiment. Once the crewmember completes that step, he/she requests the next step and so on until all steps of the experiment are completed. The crewmember then commands the PSA to visually servo to his/her face to record a summary of the experiment while the crewmember is moving. The crewmember then instructs PSA to stop recording and return to its docking bay, which it does.

Purpose:

- Demonstrate automated data collection
- Demonstrate human – autonomous system collaboration
- Demonstrate autonomous teleconferencing with face-tracking
- Demonstrate person following
- Demonstrate automated task instruction
- Demonstrate spoken language commanding and reporting

Scenario E: Long-term mixed-initiative planning and optimization including inventory tracking

Description:

PSA is given a list of visual servoing goals with time constraints and is requested to generate a near-optimal plan to achieve the goals. The goals will be such that it will be necessary to schedule multiple battery recharges in order to achieve them. The operator will dynamically change the plan prior to its execution. During the execution, PSA will monitor the location of inventory items it senses as it passes by. PSA will encounter static and dynamic obstacles in the environment. Due to an inaccurate battery model, PSA will have to replan to prevent running out of power prior to recharging at the docking bay. Once PSA has completed the goal list, it given a list of inventory items to locate, some of which it passed by. PSA responds with the locations of the items it senses and then generates a plan to explore the areas of the ISS node it did not previously explore in order to locate the other items.

Purpose:

- Demonstrate near-optimal path plan generation
- Demonstrate resource planning
- Demonstrate static and dynamic obstacle avoidance
- Demonstrate mixed-initiative plan generation
- Demonstrate spoken language commanding and reporting
- Demonstrate inventory item sensing and location tracking

Invitation to the Research Community

As previously discussed, as part of the development process for the PSA, a simulation has been developed that supports operating multiple PSAs in the ISS and interacting with in-situ crew members and dynamic obstacles in 3D. If there is sufficient interest by the research community in exploring this domain, a version of this simulation, including the simulated hardware and environment but without the autonomy, control and spoken language software, may be made available for distribution to the research community in order to encourage research in this area. Please email gdorais@arc.nasa.gov if you would be interested in such a simulation and to signify support for its release.

Summary

Spacecraft Mobile Robots, such as PSA and AERCam provide a challenging domain for a number of planning and execution problems. By developing a modular software architecture and realistic simulator, a wide number of planning and execution approaches can be analyzed. Moreover, the overall system can be incrementally improved as new planning technologies are developed. Making the simulator, scenario definitions, and operation requirements available to the research community is viewed as one way to encourage the development of such technologies that operate in a real-world environment.

Acknowledgements

The authors acknowledge the outstanding contributions of the entire PSA team and others from the NASA Ames Autonomy and Robotics Area and the Systems Engineering Division. We also acknowledge the financial support for this project by the NASA Cross-Enterprise Technology Development program and the NASA Engineering Complex Systems program. Images are courtesy of NASA JSC.

References

D. E. Bernard, G. A. Dorais, C. Fry, E. B. Gamble Jr. B. Kanefsky, J. Kurien, W. Millar, N. Muscettola, P. P. Nayak, B. Pell, K. Rajan, N. Rouquette, B. Smith, B. C. Williams, 1998. "Design of the remote agent experiment for spacecraft autonomy." In Proc. of the IEEE Aerospace Conference.

A.K. Jonsson, P. Morris, N. Muscettola, K. Rajan, B. Smith 2000. "Planning in interplanetary space: theory and practice", in Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems (AIPS '00), Breckenridge, Colorado.

N. Muscettola, G. Dorais, C. Fry, R. Levinson, C. Plaunt, 2000. "A unified approach to model-based planning and execution," in Proceedings of the Sixth International Conference on Intelligent Autonomous Systems, Venice, Italy.