Incorporating supervisory human inputs into autonomous robot navigation

by

Sai Hemachandra Vemprala

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2013 by the
Graduate Supervisory Committee:

Srikanth Saripalli, Chair
Georgios Fainekos
Pavan Turaga

ARIZONA STATE UNIVERSITY

December 2013

ABSTRACT

With robots being used extensively in various areas, a certain degree of robot autonomy has always been found desirable. In applications like planetary exploration, autonomous path planning and navigation are considered essential. But every now and then, a need to modify the robot's operation arises, a need for a human to provide it some supervisory parameters that modify the degree of autonomy or allocate extra tasks to the robot. In this regard, this thesis presents an approach to include a provision to accept and incorporate such human inputs and modify the navigation functions of the robot accordingly. Concepts such as applying kinematical constraints while planning paths, traversing of unknown areas with an intent of maximizing field of view, performing complex tasks on command etc. have been examined and implemented. The approaches have been tested in Robot Operating System (ROS), using robots such as the iRobot Create, Personal Robotics (PR2) etc. Simulations and experimental demonstrations have proved that this approach is feasible for solving some of the existing problems and that it certainly can pave way to further research for enhancing functionality.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Srikanth Saripalli for giving me the opportunity to work with him and his continuous support through my period of research and for his motivation. His timely help, excellent guidance and his provision of a great and flexible atmosphere for doing research has helped me very much during all this time and during the writing of this thesis.

Besides my advisor, I would like to thank Drs. Georgios Fainekos and Pavan Turaga for agreeing to be part of my thesis committee and thus sharing their knowledge and insights with me.

My sincere thanks also goes to Dr. Alberto Behar, for offering me exciting opportunities in his groups and leading me to work on diverse exciting projects, thus rekindling my interest in the field of scientific exploration.

Last but not the least, I would like to thank my family: my parents Mr. Sai Sivaramakrishna and Mrs. Tirumala Devi for their continued support and best wishes throughout my career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Planetary Robotics and the Need for Autonomy

One of the most important application of robots is in scientific exploration and most importantly, in extra-terrestrial planetary exploration. The presence of chaotic and unpredictable parameters in such environments is one of the major problems that has led to a wide amount of research in the field of navigation and motion planning. Because of this reason, it is usually desired of a robot to execute its tasks with a certain level of self-sufficiency rather than a continuous dependence on human inputs. In order to enable reliability of success of a robot unknown environment such as on a planetary surface, that robot must have adaptable software which can determine how to respond in unforeseen situations. Such adaptability is a form of intelligence involving reasoning, decision making and fault tolerance.

Autonomy is implemented using a mixture of various techniques from control theory, artificial intelligence, signal processing, cognitive science, and linguistics. A very common approach used to achieve autonomy on a robotic platform is what's referred to as the *sense-plan-act* model of decision-making. This model is typified by artificial intelligence techniques, such as logics and planning algorithms and usually also incorporates theoretic concepts from control, which have been used very successfully in various other fields such as aviation, aeronautics, missile control, and etc.

Path Planning

Path planning is considered to be one of the most significant problems in robotics research. Given a start estimate and a final goal, path planning is the procedure of coming up with navigation solutions in environments ranging from completely well known to moderately unpredictable. Path planning solutions usually take inputs from various sensory appliances that are used in conjunction with the robot itself, and try to generate the best path between two points in the forms of speed and rotation commands to the robot’s moving mechanisms. A basic motion planning problem is to produce a trajectory or a set of points that can facilitate a smooth motion that connects an initial configuration A and a goal configuration B, while avoiding collision with obstacles. The robot and obstacle geometry and the path are described in a 2D or 3D space.

Necessity of Human Supervision

Lunar and planetary surfaces are the one of the most hostile working environments into which any entity can be sent. Owing to this reason, intelligent robots are often sought after to replace or precede human presence in such environments. However, whenever the limits of autonomy are reached, a human will need to intervene, preferably by tele-supervising the robotic assets. Employing an effective tele-supervision architecture to augment the ingenuity and intuition of a human supervisor with state-of-the-art autonomous systems results in a manifold increase in the human’s performance and a significant improvement in safety. This completely changes the risk profile of a mission, and allows astronauts to perform substantial amounts of hazardous work from a well-supplied operations base.

The scales of autonomy can be of different types between a robot and a human. Some main types have been specified by Miller and Parasuraman (2007) as listed below:

1. Complete human control, no assistance from robot.
2. Human receives suggestions from robot and approves them.
3. Robot executes tasks automatically with feedback to human.
4. Robot takes decisions and acts autonomously, ignoring the human.

Variations of this scale have been developed and used by various authors. Importantly, Miller and Parasuraman (2007) have noted that such scales may not be applicable to an entire problem domain but are rather most useful when applied to each subtask within a problem domain. The authors further suggest that previous scales actually represent an average over all tasks.

Robots can extend operational capabilities and robustness in the face of harsh environmental conditions, whereas humans possess varied expertise and the ability to correctly interpret and react to novel situations difficult to achieve in robots. Both human safety and robot utility are increased when combining these advantages in a human supervised system. Employing multiple robots to perform dangerous and repetitive tasks with maximal autonomy preserves human attention resources and allows humans to focus on high-level guidance and assistance when situations are beyond the capabilities of the autonomy. Such a system thus effectively multiplies both human and robotic capabilities in times of need.

Problem Description and Statement

Although robots certainly will require their own intelligence to successfully complete assigned tasks in uncertain and volatile environments that exist on lunar and planetary surfaces, sometimes it's essential to receive and respond to human commands. Although this does not need to be a complete slide of autonomy towards manual tele-operation, with scientists in the background that possess higher knowledge and intuition, there is a need for a middle layer to exist between the human and the robot's low level autonomy. In addition to knowledge of terrain and task, human-interacting robots also need knowledge of human needs and the ability to respond to human commands, movements, and dynamics. The autonomy needs to adapt to changing courses of action according to human decisions and allow them to supersede the intelligence of the robot if and when needed. This does not mean complete transfer of control from the robot to the operator; since the intelligence of the robot can always help itself to plan ahead and overcome its physical or environmental constraints. The low level autonomy of the robot helps in predicting where it would be needed and how. By planning its own paths, traveling at its own rates, the robot can guide itself to achieving the human interests. This level of independence is also a key issue for human-robot interaction in a planetary exploration perspective. Time being precious and very limited, the more robots are capable of doing by themselves, the more helpful they will be in such scenarios.

Autonomy on the robot's end is needed to allow effective collaboration between the human and itself. For achieving this collaboration, the 'middle layer' needs to be capable in at least three areas: two-way communication with humans and the robot operations layers, self-monitoring and feedback. While aspects belonging to straight-forward autonomy can be accomplished with simple combinations of software and hardware, research has shown that careful analysis and implementation is required for a hybrid layer to exist that allows robots to respond to human commands and at the same time, maintain their own safety and prior task assignments: thus facilitating a hybrid autonomy (HA) architecture to exist.

The problem that is being tried to solve through this thesis is the formation of such a hybrid autonomy layer that integrates several key technologies: human robot interaction with variable autonomy; robot task allocation and cooperation with monitoring and feedback; and high fidelity tele-presence. Such a system forms a very flexible and efficient framework by allowing supervisory control on robot autonomy during execution, which can be applied with any types of robots in widely varying environments. Although, it needs to be noted that the navigation or human interaction related problems that are being handled do not exclusively belong to any of these areas; but rather, they are being brought together to make the hybrid autonomy feature more efficient.

RELATED WORK

As introduced earlier, the main concept that is being considered here revolves around studying multiple aspects such as path planning approaches, human-robot collaboration, inclusion of various types of constraints in motion planning, task allocation and how they can be integrated successfully with the ability to receive human commands and with the main perspective being applications in planetary robotics. Accordingly, a brief review of past research in these relevant fields is provided in this section.

Path Planning Approaches

There has not been a lot of research that *directly* addresses the problems that are being discussed in this thesis. Most of the research has been in nearly related areas and the trend has been to consider very specific sub problems and present solutions to it. From the days of human tele-operation to fully autonomous navigation, there have been several methods tried and tested. From a planetary robotics perspective, the most important approaches are navigating through unknown areas, implementing various constraints to path planning etc.

Roy et al. (1999) look at including the ability to localize the robot, using natural landmarks, when planning paths. This produces paths which remain close to walls. While their work was not trying to explore new regions it is considering two criteria, shortest distance and localization ability, when planning paths. Further, the ability of a robot to localize itself has been considered as an information source while exploring. Estlin et al. (2000) present a planning system for science exploration with multiple robots. The system clusters spectrometer readings to determine rock types. Goal points are set at the two mutually most distant points in physical space for each rock type seen so far. The planner is thus biased to explore towards the edges of its known world and thus stray into unknown areas.

There have been several approaches that were proposed to discuss applying constraints to a robot's inherent navigation capability at the lowest level: the level where a planner is constructed. Tetty et al (2008) discuss an approach where the constraints such as kinematics and the uncertainty of area traversal of a robot are treated as state variables along with the others and are used in trajectory planning in a multidimensional space. Kuwata et al (2008) examine the importance of constraints in a planetary environment. With the hypothesis that constraints such as computational constraints, energy constraints, hardware etc. pose, the paper goes on to provide a framework for constrained navigation for spatial and temporal planning in dynamic environments and for trajectory optimization.

Concepts dealing with maximizing field of view in an unknown area have been dealt with by various authors as well. Gerkey et al (2004) present an approach to allow robots traverse unknown areas with maximum visibility by implementing a concept called a pi searcher, which takes into account a parameter known as the visibility polygon. This visibility polygon is directly related to the angle and range of the visualization mechanism on the robot, and based on the coverage at every step, comes up with a shortest path to cover the entire area. Another approach with similarity is presented by Lee et al (2002) where a concept of k-searcher is implemented, with k being a number of infinitely thin rays stretching out from the visualization mechanism.

Supervised Autonomy and Human – Robot Collaboration

Human interaction in robot navigation is a very diverse topic that has been worked upon extensively, but most of the research till now has been extremely specific. Some of the most recent advances include work by Akce et al (2012) who propose a brain-machine interface for navigating robots through visually complex environments. Shah et al (2013) consider and discuss an approach where maps are treated as uncertain human input and waypoint extraction and obstacle avoidance as treated as probability problems by the robot. Another end of this research deals with how to present human inputs to a robot: gesture recognition, inputs that resemble natural language etc. Although human robot interaction is a concept that is being used in this thesis, as in human inputs being presented to a robot, the way in which this is done is not being considered as a major concern.

More relevant research; such as the one by Sanjiv Singh et al (2006) discusses a concept that serves the purpose of existence of sliding autonomy, where the autonomous functions of the robot are completely replaced by human tele-operation at times of need. They discuss a modular framework for a human to control a robot based on whether the robot is capable of performing the task by itself or whether it needs human commanding. But this is rather a complete switch from automatic to manual when required; than a hybrid mix of both. . Podnar et al (2008) describe a Multilevel-Autonomy Robot Telesupervision Architecture (MARTA) that acts as an interface for human – robot collaboration in planetary exploration. This system assumes a human supervisor for task planning and allocation, as well as monitoring, coordination and tele-operation if needed.

Although this approach is a step ahead of the former, it still requires a lot of effort from the human, i.e., such as specifying waypoints one by one. Dorais et al. (1998) provide several examples of how Sliding Autonomy will be essential for space operations where demands on the operator must be focused and minimized. Although Dorais et al use the term “Adjustable Autonomy”, it is in essence the same concept as a complete slide because the level of autonomy is set prior to execution. Zheng et al [2013]’s discussion relates to the area of supervisory control of robots in social environments, but this control is not directly on the navigation functions themselves but to make the robots feel more natural while navigating in a social environment

Some more noteworthy research in areas revolving around human supervisory control of robotic systems has been performed in laboratory and/or field environments on Earth for a number of applications, which include test cases belonging to construction, search and rescue, detection of hazardous material, navigation, and exploration. Fong et al. (2006) present a system called ‘Human-Robot Interaction Operating System’ which is explicitly designed to enable humans and robots to engage in natural dialogue that is task-oriented and which helps in problem-solving. Simmons et al. (2007) emphasize concepts of high-granularity sliding autonomy and how that can be applicable to heterogeneous robot teams. Crandall and Goodrich (2005) make use of terms such as ‘neglect time’ (the amount of time a robot by itself is effective between necessary human interventions) and interaction time (the time a human needs to restore a robot to effectiveness from a non-effective state) to predict how many robots a human can supervise and how. Wang et al. (2009) point out the deficiency of adopting such measures for strongly cooperative tasks and instead, propose having highly differentiated human roles and discuss how this can help handle increased control difficulty which is unavoidable as the size of the robot team increases. Trouvain et al. (2006) use multi-modal visual and auditory feedback to improve tele-supervisory control. Arkin (2000) and Murphy (1999) were both proponents of the hybrid deliberative-reasoning/reactive-execution architecture as the best general architecture for fully realizing robot potential.

Apart from a few exceptions, most of the work that has been mentioned above tends to handle very specific problem formulations and seek to provide appropriate solutions. Unlike such an approach, this thesis tries to incorporate multiple problems dealing with supervisory control of robot navigation in a single framework that enables a hybrid combination of combine supervisory human input and appropriate response from the robot which still maintains its level of autonomy, thereby producing an effective combinations of the capabilities of a human mind and those of an autonomous robot.

SOLUTION METHODOLOGY

In accordance with what has been briefly discussed before, the thesis tries to develop solutions for multiple problem formulations that arise through the same concept: how to provide supervisory inputs to a robot and modify its navigation function with multiple concepts being taken into account while writing this thesis, the questions being dealt with can be divided into sub problems such as the follows:

A. How to enable an extra level between the high level human inputs and low level robot autonomy.
B. How to command and perform traversal of unknown areas in a map. Traversal can be of different types: a swift flyby while collecting samples, or a drive through of a slow pattern that maximizes field of view while the area is scouted and mapped.
C. How to change various constraints relating to navigation/path planning when robot is already in motion and navigating.
D. How to traverse areas and still provide control on various other tasks such as varying the ways of traversal, performing operations with the other parts such as arms, collecting samples etc.

To effectively achieve this, a representation that is both powerful enough to subscribe to various pieces of the robot navigation stack and still simple enough to use by a human needs to be implemented. The solution, hence, follows a hierarchical structure, with the existing autonomous functions of the robot such as global/local path planners, localization etc. at the lowest level and the human inputs and commands related functions at the highest. The solution methodology is in the form of an infrastructure enabling hybrid autonomy acts as an interface between the above two layers in the middle. From now on, the solution node will be referred to as the Hybrid Autonomy (HA) layer or node.

To modify the path planning approach of the robot based on received human input, the solution node publishes/subscribes to various nodes that are already running on the robot while acting as a pseudo planner/task allocator. This enables delegation of tasks accordingly to the actual low level localization/navigation algorithms on the robot. As a result, for instance, the solution node cannot be responsible for deriving specific motion sequences, but to force the robot's inherent navigation capability to generates *in situ* trajectories in accordance with the goal that the HA layer is trying to achieve.

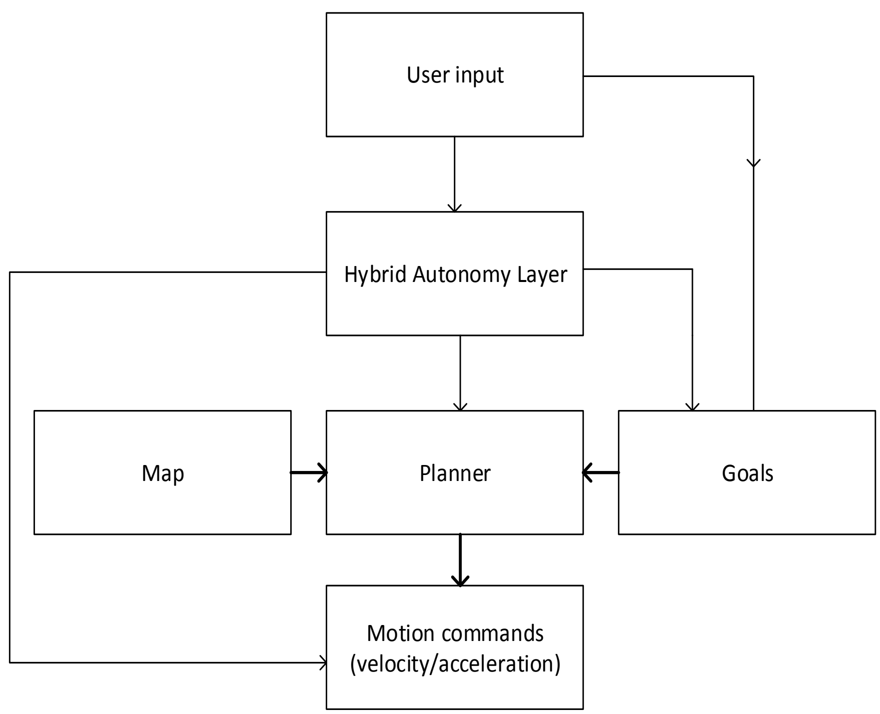


Figure 1. Hierarchy of layers in the software

Traversal of Unknown Areas

Through the architecture developed by this approach; given a sufficiently detailed map so as to enable basic navigational features, a robot can be commanded on the fly to deviate from its original plan and pass through user specified points or areas. As is mostly the case with planetary robotics, the map used to start with is a sufficiently high resolution picture obtained from satellites orbiting high above the body in question. But these maps lack the finer details and often as the feedback is being received from the robot, scientists may require it pass through areas which are either just found to be of interest or those that deviate from an original plan, if one exists. In such times of need, this architecture provides an approach to command the robot to guide itself through human specified points or areas by enabling a deviation from the original plan.

Given a space C, and assuming that the user defined point is not contained inside or on a potential obstacle, the solution node receives the coordinates of the point relative to the current position of the robot and modifies the existing global plan so as to include these coordinates in the existing set of points that determine the original plan. The advantage of such an approach is that the properties and parameters of the original planning algorithm are retained during this new planning process and the actual trajectory planning is offloaded to the robot's existing capability. The modified plan then contains the original goal of the robot reached through a different path that also includes the user-specified area and/or orientation.

This concept of passing through previously unknown/not considered points can be expanded into polygonal areas as well. The HA layer is capable of constructing its own area around a given point or take the boundaries of an area on the map as input and develop the best path such that the robot passes through the boundaries of the area. Depending on the properties of the area, the HA layer has the capability to compute the best way to traverse it: it could be a straight line through the middle of the sector, suitable for narrow passages, or it could be the longest diagonal through the area, suitable for large asymmetrical polygonal areas. Depending on this decision, the HA layer sends a set of waypoints to the navigation stack which are then followed as the robot navigates itself through the area. As mentioned before, this aspect of the motion planning adheres to the original navigation algorithm that the robot works with, so the existing capabilities such as obstacle avoidance are used directly from the framework. As the system is constantly checking the status of a particular goal, once a partial goal is reached and left behind, the robot aims for the next one from its current position. In some cases, the HA layer may change the set of goals if necessary. So the final behavior might not be exactly what was intended theoretically, but a best case scenario that combines both the user's intentions and the actual environmental conditions where the robot is navigating. Another useful way of traversing such unknown areas is by maximizing the covered area, and this is discussed in the next section.

Maximum Coverage in Unknown Areas

One of the very useful approaches to traversing unknown areas is to make sure that the robot follows such a path that the area visualized by its camera or scanner or any other mechanism as maximized.

The hybrid autonomy architecture that is developed here provides an efficient way of solving this problem. Usually, low level control on the robot navigation takes map, scanner data and other similar variables as inputs and develops trajectories, whereas the higher level processing involves deliberative patterns, usually followed/performed by a commanding entity. The layered architecture of the hybrid autonomy system provides a bottom – up way for this: the HA architecture receives a command that specifies the supervisor's interest in covering an area so that field of view is maximized, which it translates into a specific set of control commands for the lower layer. And then, the responses from the low level layer are received by the HA architecture and thereby a global solution is determined in the form of an 'adaptive grid search' scheme. Such a structure is particularly interesting for dynamic and semi-explored environments, because the already existing low level capabilities of the system can react efficiently to unexpected obstacles or other variables and offloading those tasks to the robot's software provides a chance for very flexible implementations of the HA system on any robot.

A brief overview of the steps in the algorithm that is used to construct the set of waypoints is given below.

Building the point set.

Given a configuration space C, and a bounding box B, the HA system initially decides which corner of the box B is closest to the current position of the robot in the world (P_C).

1. Compute the closest corner of the bounding box B. Assume field of view of the scanner is φ degree (or radian), with the consideration that $\varphi \epsilon (0, 2\pi]$ and the range of the scanner to be *r* meter.

2. If $\varphi < \pi$, the first waypoint is placed at a distance x away from the closest corner, where x is calculated as

$$x = (r * \sin\varphi)/sin(\pi - \varphi) \quad (1)$$

Else, if $\varphi > \pi$, $x = r$ (2)

3. Divide the box into multiple 'slices' S_1 to S_n, with the width of each slice equal to the range of the field of view. $\forall\, i < n\, ; S_i \in B$

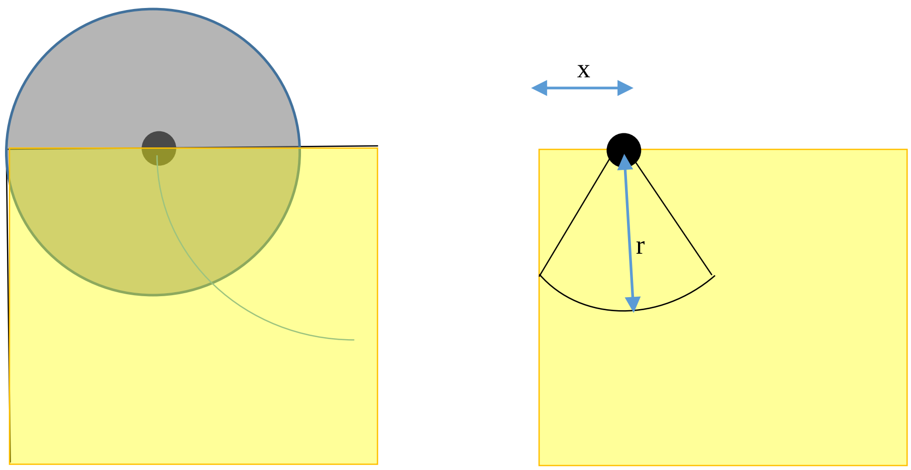


Figure 2. A robot entering an unknown area. Left: 360 degree field of view. Right: Field of view less than 180 degree.

4. As the robot enters the box, determine the orientation of the robot with respect to the box. This helps compute the best way to enter the area.
5. The robot starts navigating into the area by constructing a spline that turns the robot essentially 90 degrees away from its original heading. This is referred to as the entry mechanism.
   a. At every step, from present point P_i, compute next point P_{i+1} well as the rotation quaternion based on the heading between P_i and P_{i-1}.

$$Q(P_i,\ P_{i-1}) = \arcsin\left(\frac{y_i - y_{i-1}}{d(P_i, P_{i-1})}\right) \tag{3}$$

6. Determine how far the robot is from the opposite edge of the box. If that distance is more than twice the range of the field of view, start driving towards it until that's true. If not, skip to step 7.
7. Construct another spline that turns the robot 90 degrees away from the current orientation.
8. Consider the current slice of the box as covered. If the box is bigger than the currently covered area, store current orientation and repeat from step 4.

This entire operation can be visualized using the pictures in the next page. All the points obtained by the HA node from the above described algorithm become partial goals or waypoints to the robot. As seen in step 5, the rotation quaternions at every point are formed in such a way that there is minimum curvature change between successive waypoints so as to avoid slippage. It is again emphasized that the HA node does not force the robot to follow a fixed trajectory because of the possibilities of unexpected obstacles and other factors; but rather lets the robot move flexibly from one waypoint to another, thus lowering the probability of an entire global plan being rendered infeasible. This figure below illustrates how the HA planner tends to maximize the field of view. For simplicity's sake, a scanner with a field of view of 2π radian has been presented.

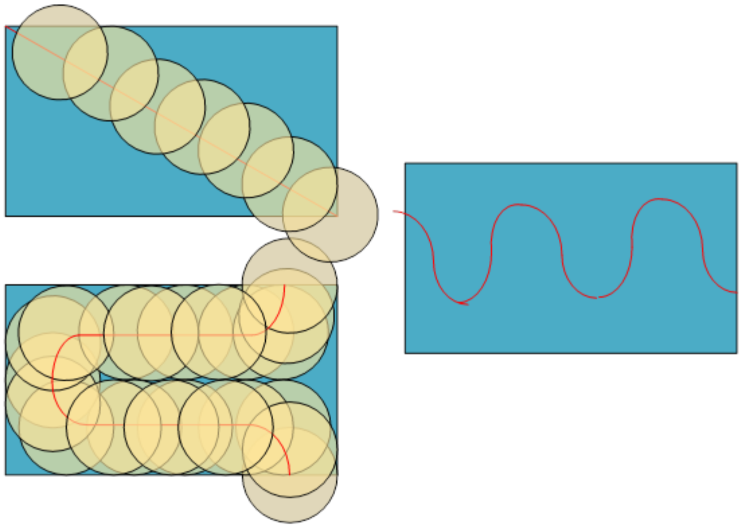


Figure 3. Maximizing field of view in an area. Clockwise from top left

a. *Non-optimal straight line path*
b. *One way of navigating the area as specified by the algorithm*
c. *Second way of maximizing the field of view that the algorithm can result*

Either pattern A or C is constructed through a set of appropriately placed waypoints. The reason for considering this pattern is that this easily results in almost complete coverage while traversing a box than any other random approach. The choice is made based on the length of traversal (shortest/fastest route preferred) and maximum area coverage but this can be modified as per need.

Dynamic Constraint Modification

Robot navigation by itself, depends on several constraints. Kinematical constraints such as velocity and acceleration limits are directly related to high level considerations such as time and energy consumption. In a planetary exploration perspective, proper handling of these constraints is considered to be highly essential.

Much research has already been performed over how to make navigation functions relative to application of several constraints.

Considering that navigation algorithms inbuilt to the robot already possess a certain amount of capability to constrain path planning functions according to different parameters, the supervised autonomy architecture developed here provides a way to dynamically modify the parameters themselves rather than force the planner to follow new constraint mechanisms. Every algorithm running on the robot such as localization, mapping and navigation can be abstracted out into a layer below the solution node, thus exposing the subset containing the parameters that belong to the algorithms. Hence, parameters can be modified during runtime and this can be used for essentially any type of parameter that happens to be present on the robot's software platform. More information has been provided in the coming sections, where details relating to how this is applied in software have been discussed. Some of the number of parameters that can be changed through this approach are:

- Translational/rotational maximum/minimum velocity
- Acceleration limits
- Trajectories for specific components such as arms or drillers etc.
- Goal or obstacle tolerances and bias

Responsive Navigation and Emergent Behavior

The HA planner is also responsible for receiving feedback from the actual navigation stack and reacting appropriately. By maintaining constant communication with the navigation nodes, the HA planner receives information if any of the user specified/HA waypoints cannot be reached owing to presence of any other external factors. Whenever a list of partial goals in available, the HA layer commands the robot to reach the closest one first in a reactive way. However, if obstacles are encountered at any of the goals, it tries to compute a new direction by communicating with the lower level layers and finding how the obstacles populate the immediate surroundings. If it does not help, the HA planner can then return the status to the user pending further instructions.

Multiple Task Allocation

The hybrid autonomy infrastructure performs most of its communication with the other levels through the concept of callbacks. A callback is a name given to a certain executable code that can be passed to another already running piece of code, with an expectation that this argument is executed and the sender is ‘called back’ when that happens. This callback can be of two types.

- Synchronous callback: Immediate response. Example: Querying the robot’s position
- Asynchronous callback: Happens at a later time. Example: Passing a constraint that is supposed to be applied at a certain future instant

Due to the presence of this mechanism, the HA layer can handle multiple threads at once and ‘spin’ them individually, i.e., monitor progress and make necessary modifications when the lower layer calls back with status updates. This enables a very flexible architecture for multiple task allocation to the robot through this layer. An example is as follows:

A planetary rover is moving from point A to B, when a scientist notices an area of interest in the distance. He then proceeds to input the coordinates of that area, thus letting the HA layer activate and adjust the ongoing path planning process to traverse that area. As the robot reaches near the area of interest, he then commands it to reduce its velocity by half and request another piece of hardware on the robot, such as a robotic arm to start taking samples every few seconds/minutes.

In this case, the human does not perform any sort of tele-operation on the robot, but instead specifies a few high level descriptors of the tasks that it is supposed to undertake. The HA layer receives and understands the input and commands the other nodes to operate consequently.

Application in Software

The solution node has been coded in C++ and applied to robots running through the Robot Operating System (ROS). ROS contains several open source packages and tools and provides a single platform for executing all of them. Any package related to the robot runs as a 'node' in ROS. ROS provides several types of localization, mapping and navigation approaches for robots.

The HA architecture is coded in C++ and acts as a separate node in ROS. This node is figuratively between the user level and the robot level. It has the capability to publish and subscribe to either the upper and lower layer. Communication between different nodes is achieved through the 'ActionLib' messaging protocol. ActionLib is a ROS stack which expands upon the ROS Service model in order to provide both dynamic feedback and the possibility of graceful goal preemption. The ROS Wiki (2013) defines ActionLib as

"The ActionLib stack provides a standardized interface for interfacing with pre-emptable tasks. The ActionLib package provides tools to create servers that execute long-running goals that can be preempted. It also provides a client interface in order to send requests to the server."

ActionLib can be extremely useful for complex robotic systems with large amounts of decoupling between components. The action client and server communicate with each other using a predefined action protocol. This action protocol relies on ROS topics in a specified ROS namespace in order to transport messages. It follows a threaded approach to combine various sets of publishers and subscribers and achieve all the goals. Note that this goal does not refer to a goal in the robot navigation space; but rather, the task allocated to an existing action server is referred to as a goal.

Action Server: Any node running on the solution platform which is capable of receiving goals is called an action server. The lower level robot functions play the roles of action servers, with their capabilities being receiving commands from the HA architecture.

Action Client: The HA architecture acts as an action client and communicates with the nodes that act as action servers accordingly.

Goal: To accomplish tasks using actions, a notion of a goal that can be sent to an action server by an action client is utilized. In the case of moving the robot, the goal would be a message that contains information about where the robot should move to in the world. For controlling a laser scanner, the goal would contain the scan parameters (min angle, max angle, speed etc.).

Feedback: Feedback provides server implementers a way to tell an action client about the incremental progress of a goal. For moving the base, this might be the robot's current pose along the path. For controlling a laser scanner, this might be the time left until the scan completes.

Result: A result is sent from the action server to the action client upon completion of the goal. This is different than feedback, since it is sent exactly once. This is extremely useful when the purpose of the action is to provide some sort of information. For movement of the robot, this might contain the success/failure parameter along with the final pose of the robot.

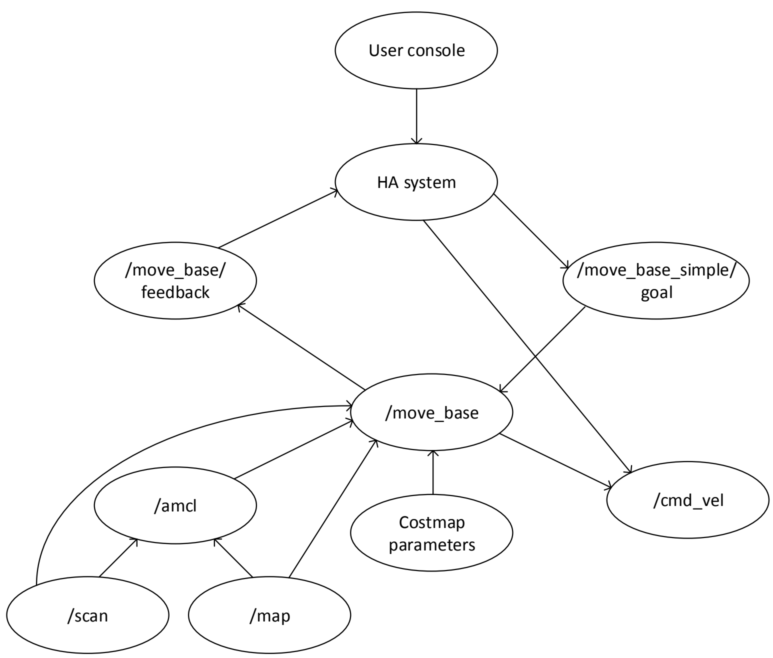


Figure 4. ROS nodes and their interconnections when the hybrid autonomy layer is running on an iRobot Create

The above figure shows how the connection of several topics would look like with an iRobot Create running on ROS with this hybrid autonomy infrastructure.

Table 1

Description of ROS topics while running the HA architecture on an iRobot Create

Name of topic	Description
/move_base	Responsible for the planning operations and navigation velocity command generation
/scan	Laser scan data from scanner
/map	2d PGM/YAML topological map data
/amcl	ROS default Adaptive Monte Carlo Localization node
/cmd_vel	Velocity command data for Create wheels

RESULTS AND DISCUSSION

Based on the methodology described in the previous section, the concept of the HA architecture has been applied through ROS to robots such as the iRobot Create and the PR2 that were simulated through Gazebo and visualized through rViz. The mapping was performed with an iRobot Create with a Hokuyo 2TMLX LIDAR. This section introduces the various tests performed and discusses the validity and importance of the obtained results.

Path Planning Through User Specified Points

This is the most straightforward implementation of the HA architecture. The HA layer examines the current path being followed by the robot and introduces the point specified as a supervisory input into the path, thus creating a new plan which ensures that this point is met but still stays true to its original goal. Because the iRobot Create can only move in the x direction at any given time, it can only be commanded to move at a certain velocity or to face a certain direction only once it has stopped, and not in mid-motion.

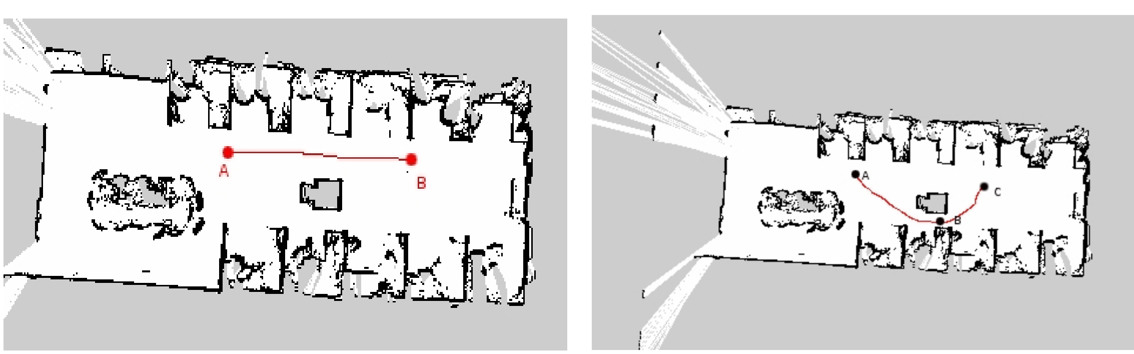
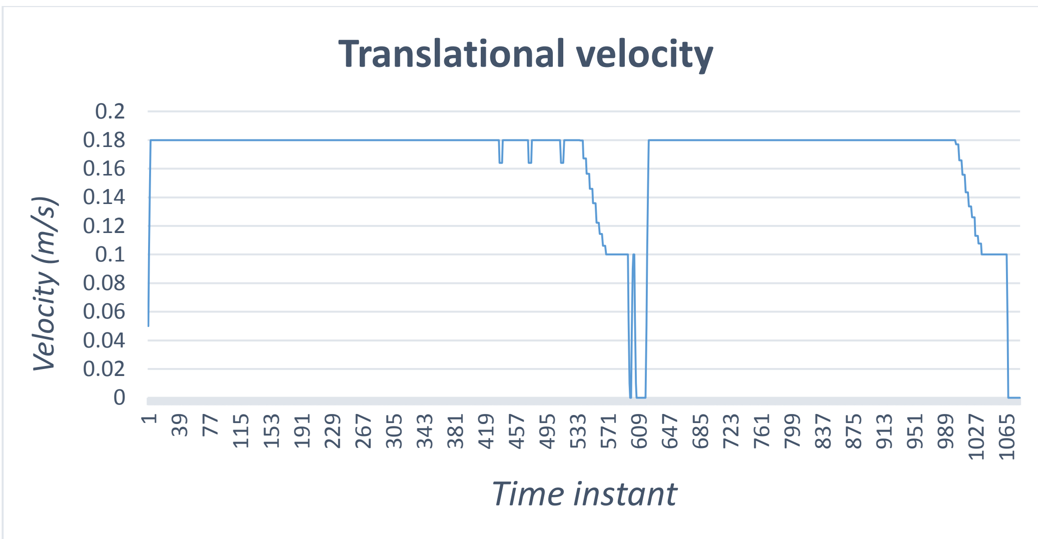


Figure 5. How the dynamic modification of an iRobot Create's planner looks in ROS.

The spline creation/trajectory rollout is performed by the robot itself and not the HA architecture, so the finer advantages of the navigation algorithm existing on the platform can be utilized to the full extent. The planner returns a smooth navigation function, which can be seen through the graph of the translational velocity of the Create. The break indicates when the robot reaches the point of interest.



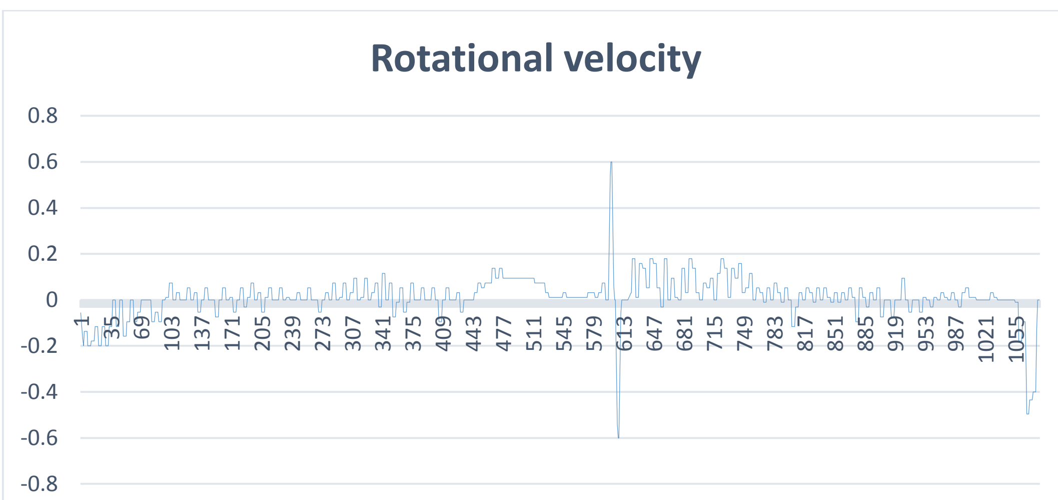


Figure 6. Translational and rotational velocities of the Create through the entirety of the planning process

Traversal of Unknown Areas

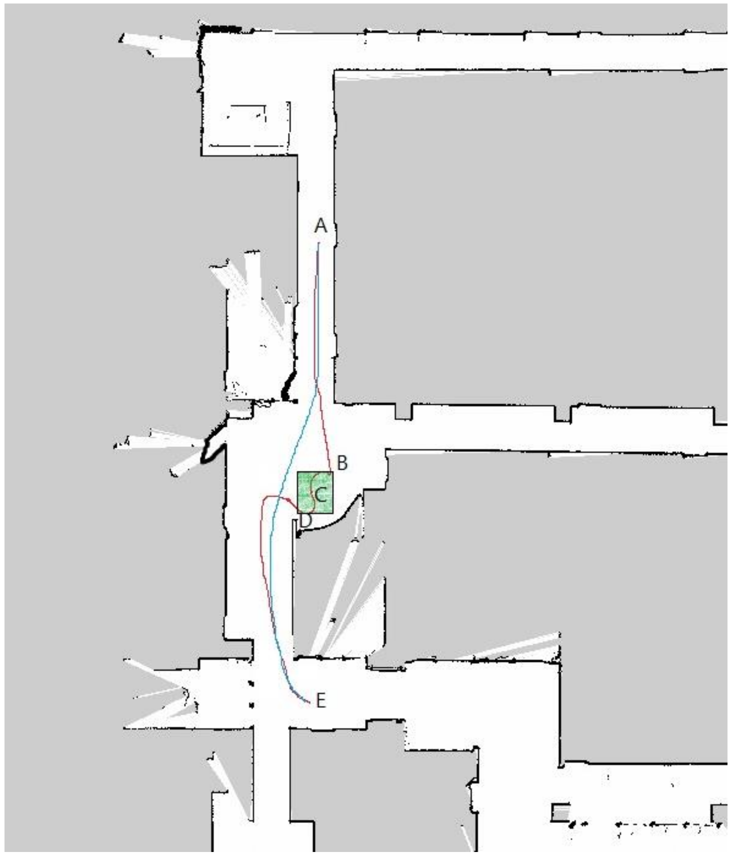


Figure 7. Another scenario where an unknown area (in green) is investigated after a command to. The original path is shown in blue whereas the modified path is in red.

In this case, the ability to form and pass through previously unknown areas has been examined. The area can either be a highly specific polygon; or the robot can be told to examine a certain amount of distance around a point of interest. In the above figure, the blue path shows the original plan that the robot intends to follow, whereas the green area is that of interest. When coordinates of point C are passed to the HA layer, it forms of field of interest around it and then calculates which corner of the bounded box is closest to the current position of the robot in the map. Just for testing purposes, the robot was instructed to still face its original path now and then even when navigating through the area, so at points B and D, the robot faces the blue path at a 180 degrees orientation (which explains the curvature when moving between B and C). As long as the current position of the robot is not equal to any one of the partial or final goals, the HA layer is active along with the other navigation related functions. Another test on the same map drives a simulated PR2 robot through multiple areas of interest. A test on the multithreaded callback capability proves an ability to provide supervisory inputs to the PR2's orientation or arms as well while the mobile base in in motion and being monitored.

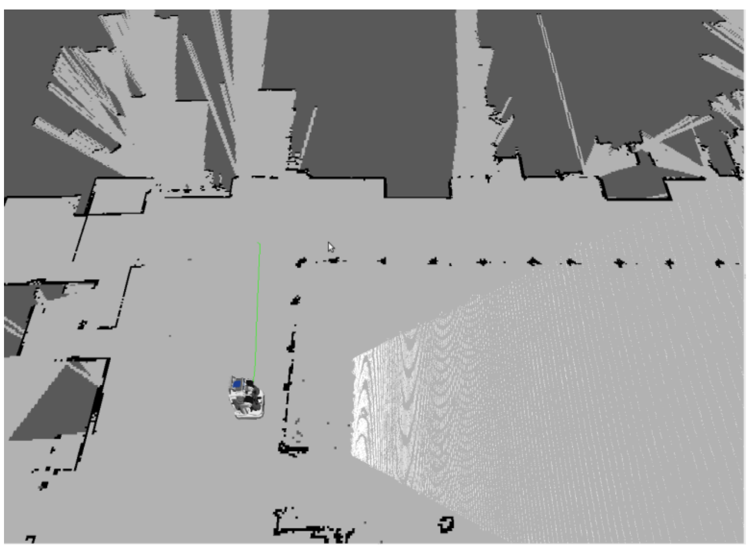


Figure 8. A PR2 examining a wall as it moves towards its final goal

A similar path has been examined for dynamic constraint modification. Once the robot reaches the border of the unknown area, it has been commanded to reduce the maximum velocity limit from 0.4 m/s to 0.2 m/s through the dynamic reconfigure capability. The change is reflected in the graphs of velocity.

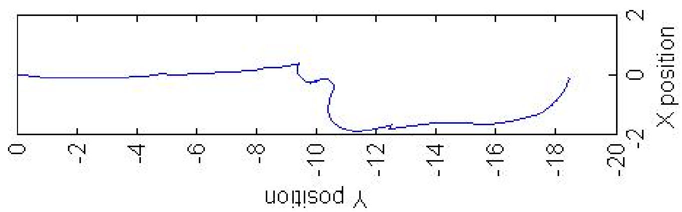


Figure 9. A path navigated by a Create through a previously unknown area plotted in Cartesian space

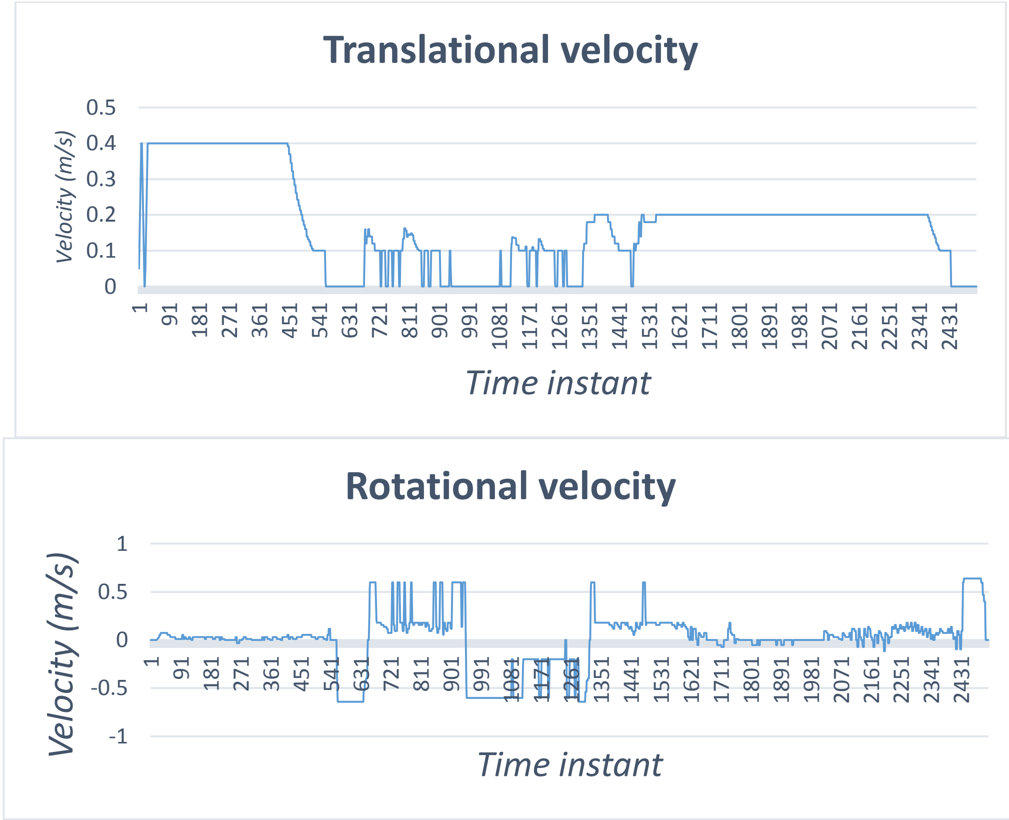


Figure 10. Graphs showing how translational (top) and rotational (bottom) velocity constraints are adapted whenever commanded to

The ability of navigating through areas has been tested in a larger setting where the robot was initially moving from a point A to B and then would be commanded to navigate through multiple areas away from the path that it had initially planned to follow.

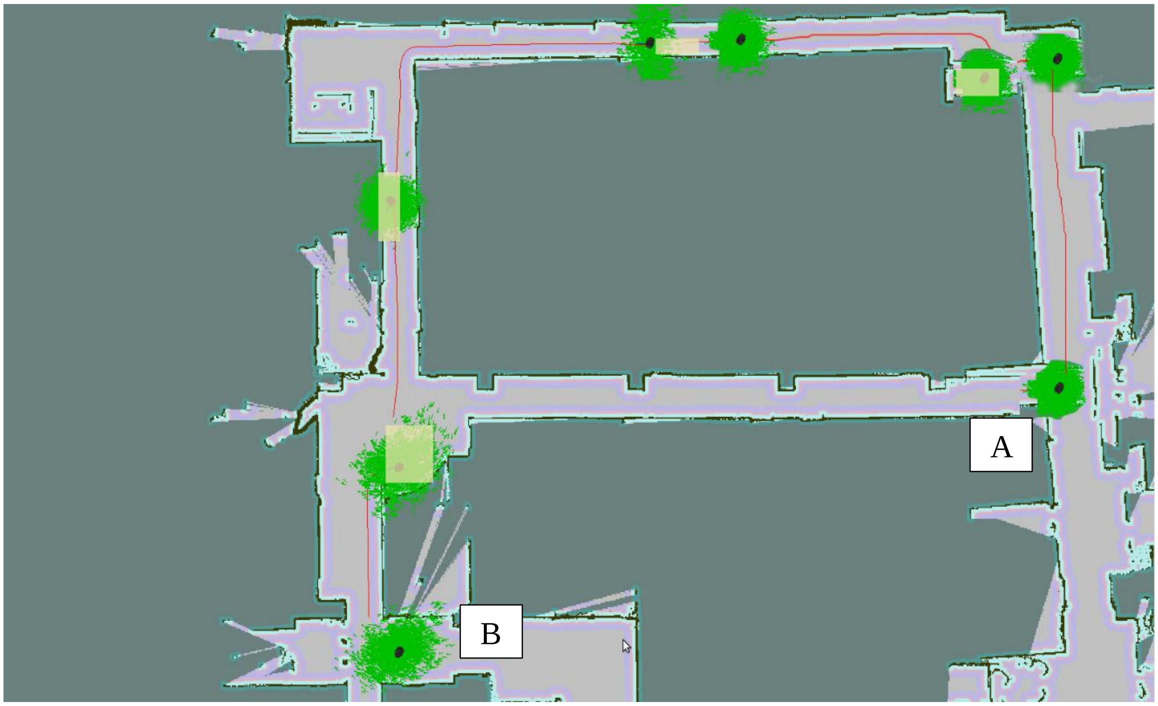


Figure 11. Time lapse of a Create navigating itself through various unknown areas instead of taking the shortest path from A to B. The yellow shaded regions are the areas of interest

The illustration above shows a time lapse of an iRobot Create navigating itself through various places of interest; instead of what would've been a shorter path from its initial position to its goal.

Maximizing Coverage through Visualization Mechanism

Although the Hokuyo LIDAR has a considerably high range, for the purpose of proving this concept, the range of the scanner has been considered to be relatively small. A value of 1 m has been used along with the true value for field of view, which is 270 degrees. For this test, the right side of the map below has been considered as the unknown area and which is desired to be mapped. Passing the coordinates of the box to the HA layer resulted in a path that looks as shown in the picture below.

Figure 12. Path through an area that's supposed to maximize area covered. Robot position and the field of view circle are approximate.

By avoiding abnormally large changes in curvature, it can be ensured that the blind spot of the laser scanner does not leave major areas uncovered.

A similar approach has been tested in another area of the map that has a considerably big obstacle in it, which resulted in a path as shown below. Even though the field of view maximization part does not possess obstacle avoidance mechanisms by itself, it can guide itself around obstacles by utilizing the navigation stack.

Figure 13. Path through an area which demonstrates how the obstacle avoidance of the lower level is utilized by the HA layer

CONCLUSIONS

This thesis document has presented an approach that helps introduce the capability of incorporating supervisory inputs that change and make autonomous navigation functions of any robot much more flexible. The test script written for the robots has been programmed to receive various types of inputs which range from unknown areas for mapping on a given map to applying dynamic kinematical constraints on a mobile platform as per need. This level of hybrid autonomy on a robot can be very useful in certain use cases as opposed to what usually limits the capabilities of a robot to what were programmed prior to execution.

A careful perusal of existing research and capacities of various robotic platform proves that this is certainly a step towards much better technologies in robotics; concentrating on mainly planetary applications but which could also be extended to various other areas. The presence of highly qualified scientists in the background and highly capable autonomous platforms in the foreground provides an opportunity for an excellent mix of both types of knowledge in the process of pursuing and meeting scientific goals. As planetary robotics is usually quite a risky field involving often small and precise windows of time, tolerance for human or machine errors is extremely low. Also, use cases are mostly worked out based on intuition and existing maps; but might be subject to change based on requirements. Considering the efficient structure of the hybrid autonomy layer and its considerably simple implementation on any platform, much more enhanced capabilities of the complete system can be experienced. For instance, a human might suddenly require his robot companion to follow him/her with its stereoscopic camera, but just because the robot has to deviate from its usual tasks does not necessitate a complete shift of control from itself to the astronaut. Implementations such as this will enable the robot visualize the human and navigate on its own, adapting to terrain changes and letting the astronaut concentrate on more important tasks. With this approach, it is definitely possible to include more advanced dynamic monitoring and modifications to robotic functions, thus paving way for the betterment technologies to serve in scientific exploration.

REFERENCES

Akce, A., Johnson M., Dantsker O., Bretl T., 2013. A Brain–Machine Interface to Navigate a Mobile Robot in a Planar Workspace: Enabling Humans to Fly Simulated Aircraft with EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* Volume 21, Issue 2, pp. 306 – 318.

Arkin, R.C., Behavior-Based Robotics, *MIT Press, Cambridge, Massachusetts, USA*, 1998.

Avorkor K.G., Stentz A., Dias M.B., 2008. Continuous-field path planning with constrained path-dependent state variables. *ICRA 2008 Workshop on Path Planning and Costmaps*. Retrieved from http://www.ri.cmu.edu/publication_view.html?pub_id=6100

Crandall, J.W., Goodrich, M.A., Olsen, D.R. Jr., Nielsen, C.W., 2005. Validating Human-Robot Interaction Schemes in Multitasking Environments. *IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 35, Issue 4, pp. 438-449.*

Dorais G., Banasso R., Kortenkamp D., Pell P., Schreckenghost D., 1998. Adjustable Autonomy for Human-Centered Autonomous Systems on Mars. *Proceedings of the Mars Society Conference, 1998.* Retrieved from www.barneypell.com/papers/mars_adj_auton98.pdf

Elfes, A., Dolan, J., Podnar, G., Mau, S., Bergerman, M., 2006. Safe and Efficient Robotic Space Exploration with Tele-Supervised Autonomous Robots. *Proceedings of the AAAI Spring Symposium 2006, Stanford, CA, USA, March 27-29,* Retrieved from http://repository.cmu.edu/cgi/viewcontent.cgi?article=1192&context=robotics

Fong, T. et al., 2006. A Preliminary Study of Peer-to-Peer Human-Robot Interaction. *IEEE International Conference on Systems, Man and Cybernetics, pp. 3198-3203.*

Gerkey B.P., Thrun S., Gordon G., 2006. Visibility-based pursuit-evasion with limited field of view. *The International Journal of Robotics Research,* Volume 25, No. 4, pp. 299-315.

Heger F. and Singh S., 2006. Sliding Autonomy for Complex Coordinated Multi-Robot Tasks: Analysis & Experiments. *Proceedings of Robotics: Systems and Science, August 2006.* Retrieved from www.ri.cmu.edu/pub_files/pub4/heger.../heger_frederik_2006_1.pdf

http://www.ros.org/wiki/

Kuwata Y., Elfes A., Maimone M., Howard A., Pivtoraiko M., Howard M.T., Stoica A., 2008. Path Planning Challenges for Planetary Robots. *IROS 2008 2nd Workshop: Planning, Perception and Navigation for Intelligent Vehicles.* Retrieved from www.isr.uc.pt/~urbano/WorkIROS08/.../P21-Kuwata-2PPNIV-FI.pdf

Murphy, R.R., Introduction to AI Robotics, *a Bradford Book, MIT Press, Cambridge, Massachusetts, USA, 2000*.

Nikolaidis, S., Lasota P., Rossano G., Martinez C., Fuhlbrigge T., and Shah J., 2013. Human-Robot Collaboration in Manufacturing: Quantitative Evaluation of Predictable, Convergent Joint Action. *Proceedings of the 44th International Symposium on Robotics.* Retrieved from http://interactive.mit.edu/sites/default/files/documents/snikol_ISR2013.pdf

Podnar G., Dolan, J., Elfes, A., Bergerman, M., 2008. Multi-Level Autonomy Robot Telesupervision. *Robotics Institute, Paper 181*. Retrieved from http://repository.cmu.edu/robotics/181

Roy, N., Burgard, W., Fox, D., Thrun, S. 1999. Coastal Navigation - Mobile Robot Navigation with Uncertainty in Dynamic Environments. *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*. DOI: 10.1109/ROBOT.1999.769927

Simmons, R., Apfelbaum D., Burgard W., Fox D., Moors M., Thrun S., Younes H., 2000. Coordination for Multi-Robot Exploration and Mapping. *Proceedings of the National Conf. on Artificial Intelligence.* Retrieved from robots.stanford.edu/papers/Simmons00a.html

Simmons, R., Singh, S., Heger, F., Hiatt, L., Koterba, S., Melchior, N., Sellner, B., 2007. Human-Robot Teams for Large-Scale Assembly. *Proceedings of the NASA Science and Technology Conference 2007 (NSTC-2007).* Retrieved from http://www.ri.cmu.edu/pub_files/pub4/simmons_reid_2007_1/simmons_reid_2007_1.pdf

Trouvain, B.A., Schlick, C.M., 2006. Binaural Auditory Information Fusion for Effective Supervisory Control of Mobile Multirobot Systems. *Proceedings of the IEEE International Conference on Multi-sensor Fusion and Integration for Intelligent Systems, pp. 20-24.*

Wang, H., Chien, S.Y., Lewis, M., Velagapudi, P., Scerri, P., Sycara, K., 2009. Human Teams for Large Scale Multirobot Control. *IEEE International Conference on Systems, Man and Cybernetics, pp. 1269-1274, San Antonio, Texas, October 2009*.

Zheng K., Glas D.F., Kanda T., Ishiguro H., Hagita N., 2013. Supervisory control of multiple social robots for navigation. *Proceedings of the 8th ACM/IEEE International conferences on Human Robot Interaction (HRI '13), pp. 17-24*