Terrain Exploration, Planning and Autonomous Navigation with MRPTA Rover

V. Polotski[*,***], F.J. Ballotta[**] and J. James[***]

* *Ecole de Technologie Superieure*
e-mail: vladimir.polotski@etsmtl.com

** *Independent software contractor*
e-mail: franco.ballotta@gmail.com

*** *Cohort Systems Inc*
e-mail: jjames@cohortsys.com

Abstract

Micro-Rover Platform with Tooling Arm (MRPTA) project has been initiated and financed by the Canadian Space Agency. It targeted the development of a robotic system with remote control and autonomous navigation capabilities for testing various analogous planetary missions. An autonomous navigation system was developed that is capable of movingthe platform to predefined position(s)/orientation(s), assessing the terrain traversability and choosing the most appropriate path to follow. It uses simple sensor configuration and extensive fusion algorithms for pose estimation, and a nodding laser scanner for building a world model of the surrounding environments. A continuously maintained terrain map is used for planning the path, that is executed by the platform controller. Among the system achievements are very good position accuracy (especially for a reconfigurable skid steer platform), modular structure and a high level of software abstraction resulting in easy adaptation to newly defined tasks.System development testing,discovered drawbacks and lessons learned are also described.

1 Introduction

Planetary exploration in general, and rover navigation in particular, have been a subject of continuous interest to the mobile robotic community over the last two decades. Examples of early works worth mentioning are [1,2]. More recent results are often related to the development and analysis of the Mars missions [3,4]. The Space Exploration Group at the Canadian Space Agency (CSA) has also been very active investigating long term autonomous navigation activities [5].

Starting in 2009 CSA initiated and financed the series of projects devoted to various aspects of planetary exploration activities. This paper describes an onboard navigation system developed to guide the Micro-Rover Platform with Tooling Arm (MRPTA) designed for CSA in 2010-2012. MRPTA rover's main goal is to help astronauts remotely explore unknown environments within short to medium range proximity (approximately 100m) from their location. The typical mission consists of "scouting" - moving autonomously to specified locations, taking video recordings, making scientific measurements or collecting samples, then returning back to the starting point. The work described in this paper has been accomplished by Cohort Systems Inc. under a subcontract to Engineering Services Inc (ESI) according to CSA contract F028-090480. Electromechanical design and low–level software design was the responsibility of the ESI engineering team and are not addressed in this paper. We focus on the high level architecture for planning, control and sensor fusion, as well as on high level software design. We also address the system experimental validation and lessons learned along the development, field testing and delivery of the rover.

2 System Architecture

The rover navigation software is designed to support autonomous and tele-operated operation modes. At a high level it is partitioned into 3 layers: (1) a deliberative or mission execution layer (DL), (2) a reflexive or navigation layer (RL), and (3) a platform control or low level layer(PL). In order to provide the human operator a way to interact with the rover in autonomous mode, and to execute the tasks requiring tele-operation, an Operator Control Unit (OCU) was developed. The low-level platform control is tightly connected to the hardware development and is not addressed here; we limit our discussion to the deliberative and reflexive layers, and shortly address the OCU structure.

The deliberative layer continuously evaluates a best sequenceof actions that will achieve mission goals and elaborates these action sequences into an ordered list of commands that are performed by thereflexive layer. Individual commands are presented to the RL for execution and continuously monitored. The DL decides on appropriate parallel or subsequent actions based on the execution state reported by the reflexive software (*in process, success, failure, aborted,* or *cancelled*). Development of the deliberative layer follows the methodology of "robotic autonomy" proposed in [6]. This layer also contains the Deliberative-Reflexive Interface - a communication portal between two layers used to transfer commands and status messages. The interface module performs protocol conversion between the layers: it translates message contents to a form required by the receiving component and dispatches the translated content to the intended receiver using an appropriate communication protocol. The DL, in particular, initiates and monitors the execution of complementary tasks such as scooping, scientific measurements, or taking panoramic views according to the goals set in the mission plan. Task planning essentially follows the methodology described in [7].

MRPTA inter-component communication and the full reflexive layer implementation follow the guidelines of the Joint Architecture for Unmanned Systems (JAUS) and the OpenJAUS standards in particular. JAUS has evolved from an experimental development initiative to a current set of SAE –compliant standards maintained by the JAUS Working Group. JAUS is a message-based architecture organized in a three-level hierarchal network (subsystem, node and component) with *component* being a main functional element. All modules in the reflexive layer and the OCU are implemented as JAUS components. Components collaborate by exchanging well-defined messages. Each JAUS component is a self-contained entity that provides services to other components. Each service is implemented as a messaging interface that uses a standard message protocol to interact with other components. Each component is defined as a state machine with 6 standard JAUS states: Initialization, Standby, Ready, Emergency, Failure and Shutdown. The state machine within each component operates at a specified frequency (the default is 10 Hz). State transitions are initiated by internal logic or by external messages.

3 System Functionality

Autonomous navigation is a complex task requiring pose estimation, assessment of theenvironment and motion execution. Below we briefly address the sensing equipment then describe the functionality of the most important components involved in autonomous navigation (there are ~30 components in total, and their full description is impossible due to space and scope limitations).

3.1 Sensing Equipment

MRPTA pose is reconstructed from the data provided by an absolute inclinometer, an azimuth gyro, and an *odometer* composed of encoders attached to the left and right driving motors. Since the rover may be used in several driving configurations, an effective wheel radius for each configurations is stored onboard, and the configuration is a part of mission definition. An AMD900-TW (Applied Geomechanics) absolute inclinometer provides pitch and roll measurements and a KVH DSP-3000 fiber-optic gyro provides azimuth angle increments. These sensors were chosen for their accuracy and robustness. For operations on high slopes (beyond the range of the AMD900-TW) a small IMU is used, which also serves as a backup sensor. A laser range finder manufactured by Hokuyo (UT-30LX) is used to collect environment surface data. This sensor is mounted on a servo controlled tilt unit in order to increase the sensing area. Each sensor has a dedicated software component - a *server* responsible for providing sensor data to client components.

3.2 Pose Estimation

The pose estimation algorithm (PE) first takes data from an azimuth gyro and an inclinometer to estimate platform rotational pose (yaw/pitch/roll) and then combines the results with incremental displacements obtained from an odometer. The PE developed for the MRPTA rover has two additional features. First, it performs *zero-update*, namely it keeps the rover orientation constant and estimates the gyro bias whenever the platform is consistently not moving for a certain period of time. Second, it performs best possible estimates of wheel slippage. Because the rover is skid-steered in both wheeled and tracked configurations slippage-induced errors can significantly affect pose estimation. The ability to eliminate or reduce slippage-induced error has proven to be very important [3,4]. Slippage correction in our system relies on redundant platform rotation estimates: those obtained from differential odometry and those from the azimuth gyro. Careful monitoring and correction of rotational error allows the PE to provide consistent fusion-based displacement and rotation estimates. In the absence of independent (visual-based) pose estimates, this capability minimizes longitudinal odometry errors, that otherwise will grow rapidly.

3.3 Terrain Evaluation

Based on the data collected by the LIDAR the terrain evaluator (TE) determines traversability values for the area around the rover's location and stores this data in a centered-on-platform toroidal grid (T-grid). The T-grid is continuously updated with the new LIDAR data and updated T-grid data is regularly sent to the Map Manager where the world model is developed and maintained. If there are not enough data-points in the area in front of the rover the TE issues commands to the LIDAR Tilt Controller to achieve a minimum point density over contiguous grid cells immediately in front of the vehicle. The TE attempts to provide cell coverage within a ~6 meters horizon in front of the rover and to guarantee a minimal point density over a shorter, configurable distance. Currently the TE assigns point data to a (121x121) T-grid with cell size of (0.15x0.15) meters. Traversability assessment uses an approach proposed in [8] and is based on the combination of three characteristics: (1) slope estimate *S*, (2) roughness estimate *R*, and (3) neighbourhood estimate *N*. Traversability values are scaled between 1 and 14 - higher values mean better traversability, 7 is neutral and 2 or less is not traversable. The LIDAR pose needed for computations is obtained from the Tilt controller. The slope estimates uses the best fitting plane through the data points in each cell.The roughness estimate is represented by the variance of the elevation of the data points within a cell. The neighborhood estimate is based on the assessment of the travel from a cell to the center of the grid (note that T-grids are centered-on-platform). The rationale is given below.

Let us consider a cell at the row *i* and column *j* from the T-grid center. The cell height $h(i,j)$ is computed first by averaging elevations of all points in a cell. Then the weight factors c1, c2, and c3 are defined as follows:

$$c1 = \frac{|i|}{C}; c2 = \frac{|j|}{C}; c3 = 1 - c1 - c2;$$

$$hereC = \left(|i| + |j| + \frac{|i| + |j|}{\sqrt{2}}\right)$$

Using the heights of the cell and of its neighbors on each side, and the weight factors the neighborhood estimate $N(i,j)$ is computed as follows:

$$N(i,j) = h(i,j) - c1 * h(i-1,j) - c2 * h(i-1,j-1) - c3 * h(i,j-1)$$

This formula is for a cell in the first quadrants ($i, j>0$), other cases are similar. The final traversability estimate $T(i,j)$ is then calculated as: $T = \min(S + R/2\,, N)$.

3.4 Map Building and Maintenance

Map Manager (MM) maintains a *world model* containing all spatial information used by the navigation system. At start-up it reads in map data from external sources if available or initializes its *world model* as a flat terrain. During operations it updates the *world model* using position/orientation data received from PE and terrain information obtained from TE and provides map data to other navigation components. The MM maintains a traversability map and an elevation map where *elevation* is identical to *Z(X,Y)* in JAUS world coordinates. Information from the TE is received in a form of a T-grid. With information from the PE, MM registers the central cell of the T-grid with the rover's pose, it monitors the rovers displacement and adjusts the T-grid to ensure T-grid shifts (cell entering and falling off the grid). The obtained terrain traversability map differs from the elevation map and is used only by path planning components.

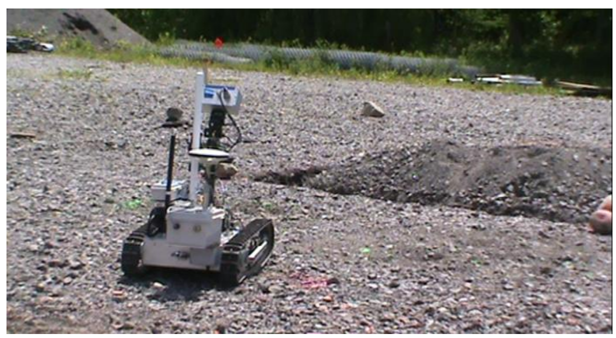
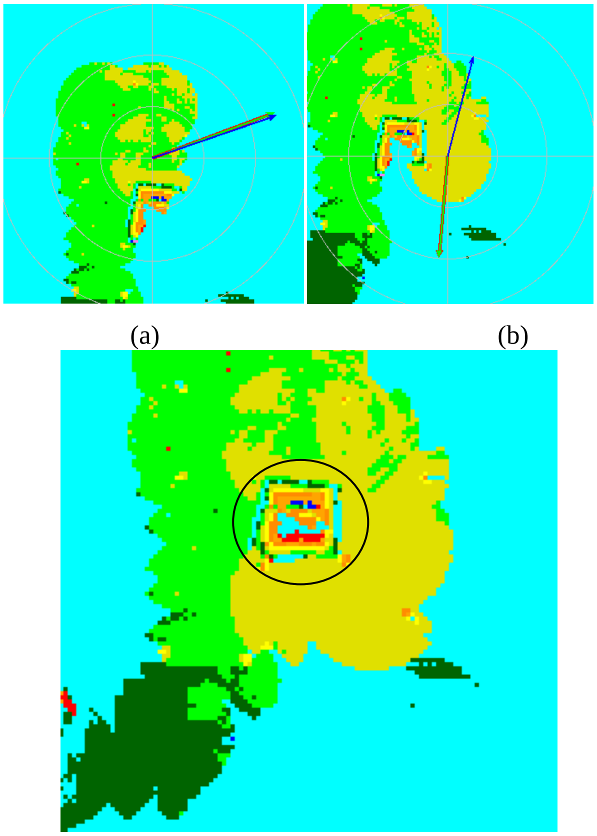


Figure 1.MRPTA on the Cohort testing site

Figure1 shows the rover on the Cohort testing site approaching an artificially built highly uneven area - a 'castle-type' structure, containing an elevated area surrounded with a trench. It is built for testing the rover's ability to estimate positive and negative heights and is shown in Figure 2.

Figure 2. Artificial structure (a castle) for testing MRPTA mapping capabilities



(a) (b)

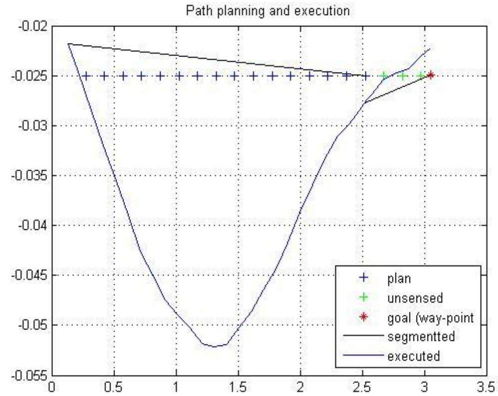
(c)

Figure 3 Maps dynamically build by MRPTA while travelling around the castle: (a) and (b) partial, gradually growing maps, (c) final map.

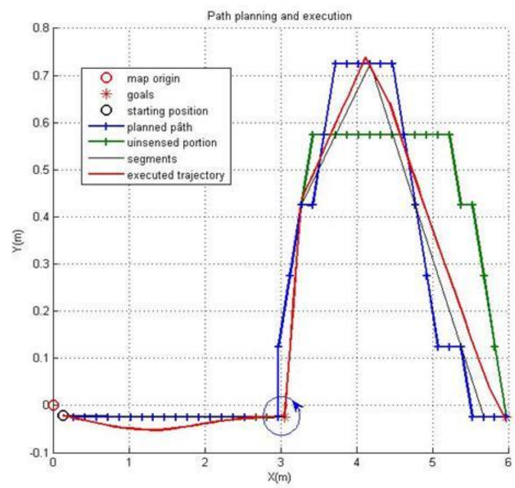
Figure 3 illustrates the process of dynamically building an elevation map along an experimental run. The height is coded with the colors:blue corresponds to zero-level and is also used for map initialization, so the large peripheral area of the map is blue. Red corresponds to >40cm height, yellow to >15cm height and green to <8cm height. Red arrows in Figures 3(a) and 3(b) define the direction of motion. In Figure 3(a) the castle structure starts to appear along with the portion of surrounding trench (green as compared to yellow corresponds to ~ 10 cm of relative depth). In Figure 3(b) the structure continues to build up. One can see that portions of the inside of the castle are still occluded – that is shown in blue (un-sensed). Also portions of the trench are too deep to be sensed from the close locations and appear as blue. In Figure 3(c) the castle-structure appears in full (encircled) with the elevated area at the center and some occluded regions aside of it. The big region around the castle has a height of ~15 cm, with the trench partially perceived with the height of ~8cm (green) and partially un-sensed (blue). These results are in good agreement with the actual trench depth of ~10cm.

3.5 Path Planning and Execution

Path planning is based on the terrain traversability map constructed by TE and MM components. The planning algorithm computes the path to the goal considering the area that has not been sensed yet as being neutral (flat), but keeping them tagged as un-sensed.Description of the state of the art can be found in [8]. The planner in our system is based on D*-search and is relatively simple due to a short sensing range resulting in the low computational load.



(a) Paths to the first via-point



(b) Paths from the first to the second via-point

Figure 4 Planning, Segmentation and Execution

The planned path is represented by the sequence of cells to be visited. This information is passed to the

Segment Driver (SD). The SD is responsible for defining the portion of the path containing only cells that have been sensed and for dividing this portion into a sequence of straight line segments. SD also verifies that the directions of consecutive segments are sufficiently close (a 10 degree threshold is currently applied), and (alternatively) inserts the *in-situ* rotation between neighboring segments. A final set of segments (and *in-situ* rotations) is presented to the Motion Controller for execution. Motion control along the straight lines and the execution of *in-situ* rotation are routine procedures and their description is omitted.

Figure 4 illustrates the described algorithm with experimental data.In Figure 4(a) the planned path(to the first via point) is marked with crosses (the sensed portion with blue crosses, the un-sensed portion - with green crosses, the via-point - with a red cross). Black solid line corresponds to a computed segment. An offset between starting points of the planned path and the segment is due to the fact that planned points are the centers of the cells, but the segment starts at the rover's location. The solid blue line corresponds to the executed trajectory (PE estimates). After execution of the segment the path is re-planned. Previously un-sensed points can be now used and a new segment starting close to the end of the previous one is defined and then executed. Note the very different scale factors along X-axis (0.5 m) and Y-axis (5 mm). In Figure 4(b) it is shown how the path to the next via-point is planned, segmented and executed. The initially sensed portion of the planned path leads to one segment. In-situ rotation (marked as a circular arrow)needs to be inserted due to a large angular offset between consecutive segments. By the time these segments are executed, the path to the second via-point is re-planned and is fully sensed, soit is segmented and executed.

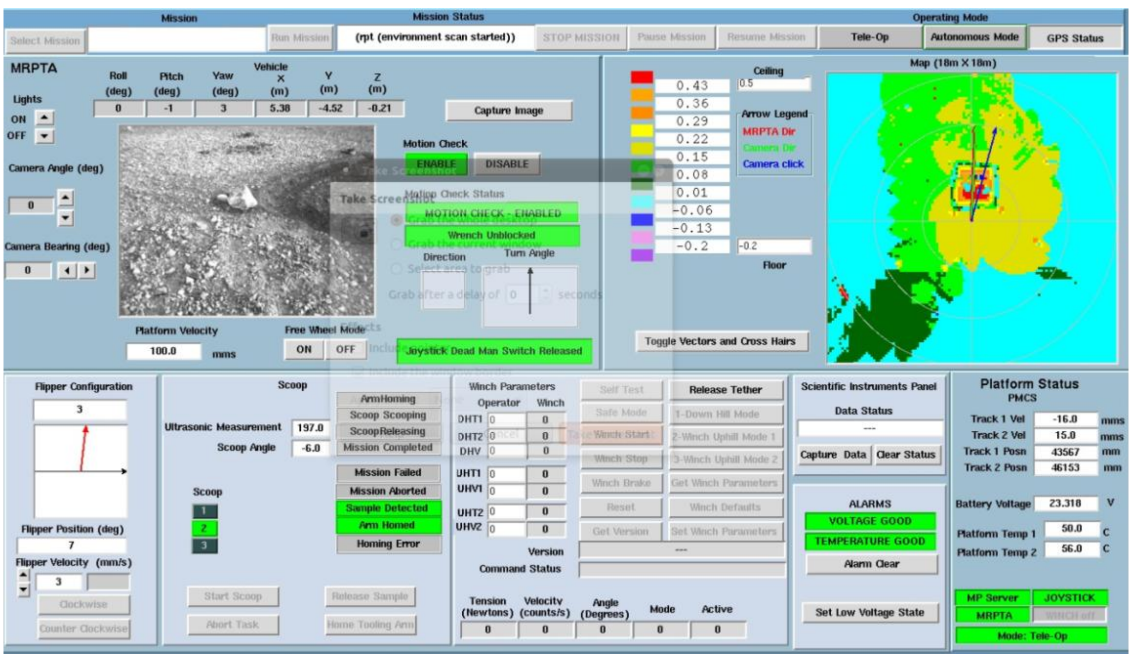


Figure 5 – OCU snapshot along an experimental run on the Cohort testing site

3.1 Interfacing with the Operator

The OCU is a graphical interface providing a human operator with the possibility to control the system. The OCU continuously processes the events triggered by interface signals and incoming JAUS messages. It also manages various logging activities and stores data for future analysis. Figure 5 illustrates the OCU showing (1) constructed map in the upper right corner with the elevations coded by colors (as shown by a panel on the left of the map), (2) camera view ensuring operator situational awareness – in the upper left corner, (3) platform status parameters in bottom right corner. The OCU is used in both autonomous and tele-operation modes. The interface includes a POP-UP menu for mission definition (not shown). In tele-operation mode the operator uses a hand held controller and may switch from platform driving to controlling various pieces of equipment, including a *flipper* – a variable length arm used to change the track shape (flipper position is shown in the bottom left corner). The flipper arm is visible in Figure 6 where the rover is in short track configuration (the triangular shape) while the long track configuration used for long range missions is presented in Figure 8.

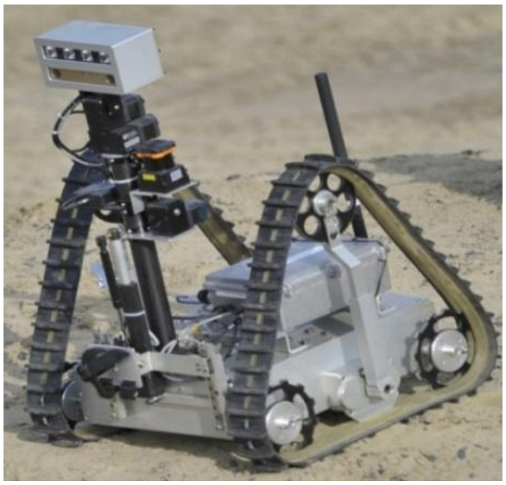


Figure 6 MRPTA - short track configuration

4 Integration and Testing

MRPTA project required integration of several pieces of sophisticated equipment. In Figure 6 one can see the Hokuyo LIDAR mounted on the tilt unit, a stereo camera mounted on pan-tilt unit and a scientific instrument attached to the tooling arm.

Scoop opening

Scoop lowering

Cutting the soil (ground entering)

Scooping (getting the soil sample)

Figure 7 Autonomous scooping executionat the Cohort testing site

The sequence of pictures in Figure 7 illustrates the autonomous execution of ‘scooping’ – taking soil samples using a small scoop specifically designed for the MRPTA and attached to the tooling arm. The platform is in “wheeled” configuration (as compared to Figure 6 where it is on “short tracks”). One can observe the scoop gradually opening, reaching the ground and finally collecting the soil. One observed drawbacks consisted of improper monitoring of the force along the penetration (soil cutting) phase, resulting sometimes in mission failure.

Figure 8 shows the MRPTA in long track configuration during acceptance testing at the CSA Mars Yard. Hundreds of hours have been devoted to verification and integration tests and kilometers of experimental runs have been executed. For scouting type missions the via-points were usually defined 3-5 m apart and the accuracy of achieving those points was growing in average as 2.5% distance traveled. The accuracy of MRPTA observed at the return to the starting point was ~2% of the travelled distance for the missions of ~100 meters, executed at an average speed of 0.1 m/sec. Obtained accuracy is comparable to that of other systems relying on dead-reckoning and inertial sensors and can be qualified as a very good because it is achieved on the skid-steer platform on uneven terrain with the use of relatively low-cost sensors. Better performance - 1% of the distance traveled – was required in another CSA founded project (Lunar Analogue Rover) and reported in [10], but the developed system contained a high-grade 6 DOF IMU and two separate visual-based localization modules [10, 11].

Figure 8 – MRPTA moving across the CSA Mars Yard along the delivery demonstrations

Mapping capabilities provided by MRPTA, although not as accurate as its positioning, are on a very acceptable level. These support on-board planning and serve for collecting data useful for off-board analysis of the visited areas and planning of subsequent missions. The traversability estimates are quite sensitive to terrain characteristics and required significant tuning; different parameter sets are required for significantly different terrains. The JAUS messaging system proved to be stable and facilitated modular development and testing. We noted that sometimes the JAUS framework would fail to start cleanly, but attributed this to a flaw in our process supervisor implementation and not an issue inherent to JAUS.

5 Conclusions

The autonomous navigation system developed for the MRPTA rover has been described including software architecture, sensing equipment and navigation algorithms on the level of details allowed by the limited length of the paper. Although designed primarily for planetary exploration, our system, due to its modularity and high level of abstraction from the actual hardware, is easily adaptable to the tasks that may arise invarious terrestrial applications. Our system does not require GPS positioning and therefore can be used in GPS-obstructed zones as often met in construction sites, or within totally GPS-unreachable sites as *e.g.* in underground mines.

As compared to the system developed specifically for mining applications[12], MRPTA uses more advanced positioning and planning algorithms and has more mapping capabilities.

Adapting the system to the mining or construction environments some of the assumptions made in the context of planetary exploration have to be revisited. For example, our system does not consider overhead obstacles (that might appear from suspended objects). Also MRPTA mapping is based on 2 D representation of the environment (with an added height component). Other applications may require more generic 3D representation as discussed in [13]. Modularity of our system greatly simplifies the development and integration of such additional functionalities. One particular useful system feature is the ability to reset the system position and/or orientation from an external authoritative source. In applications such as construction, mining or security it may allow integration with an existing RF-tag, visually augmented or other type of infrastructure-based positioning environment.

Possible system enhancements we would like to consider include the integration of stereo–camera-based scene reconstruction (visual odometry) with pose estimation in order to eliminate the effects of platform longitudinal slippage in long range missions. Further development of map representation that is more appropriate for the future use also seems necessary. A solution might be based on the concept of *atlas* [14], allowing the decomposition of the full map into the set of connected regional maps without losing representation accuracy and keeping the system computational load limited.

References

[1] L. Matthies, R. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin, *Mars microrover navigation: Performance evaluation and enhancement* Autonomous Robots Journal, Special Issue on Autonomous Vehicle for Planetary Exploration, 1995, Vol. 2 No. 4, pp 291-311.

[2] A. Howard and H. Seraji, *Real-Time Assessment of Terrain Traversability for Autonomous Rover Navigation*, Proceedings of the IEEE Int. Conference on Intelligent Robots and Systems, 2000.

[3] J. Biesiadecki and M. Maimone, *The Mars Exploration Rover Surface Mobility Flight Software: Driving Ambition*. Proceedings of the IEEE Aerospace Conference, 2006, March, Montana, USA.

[4] M. Maimone, Y. Cheng and L. Matthies, *Two years of Visual Odometry on the Mars Exploration Rovers*. Journal of Field Robotics , 2007, 24 (3), pp. 169–186.

[5] E. Dupuis, P. Allard, J. Bakambu, T. Lamarche, W.-H. Zhu and I. Rekleitis, *Towards Autonomous Long-Range Navigation*,“ISAIRAS-05” Conference, 2005, Munich, Germany.

[6] R. Alami, R. Chatila, S. Fleury, M.Ghallab and F. Ingrand, *An Architecture for Autonomy.* Int.Journal of Robotics Research (Special Issue on Integrated Architectures for Robot Control and Programming, 1998, Vol. 17, N° 4.

[7] F. Ingrand, S. Lacroix, S. Lemai-Chenevier and F. Py, *Decisional autonomy of planetary rovers*, Journal of Field Robotics , 2007, Vol. 24,No7, pp. 559 – 580.

[8] S. Solanki, *Development of Sensor Component for Terrain Evaluation and Obstacle Detection for an Unmanned Autonomous Vehicle* (Doctoral dissertation). University of Florida, 2007. (www.cimar.mae.ufl.edu/CIMAR/pages/thesis).

[9] J. Carsten, A. Rankin, D. Ferguson and A. Stentz, *Global planning on the Mars Exploration Rovers: Software integration and surface testing*, Journal of Field Robotics, 2009, Vol.26, No 4, pp. 337-357.

[10]M. Wagner, D. Wettergreen and P. Iles, *Visual odomtery for the Lunar Analogue Rover “Artemis”*,“ISAIRAS-12”, September, Turin, Italy.

[11] V. Simard Bilodeau, D. Beaudette, J.-F. Hamel, M. Alger, P. Iles and K. MacTavish, *Vision-based pose estimation for the Lunar Analogue Rover “Artemis”*,“ISAIRAS-12”, September, Turin, Italy.

[12] J. N. Bakambu and V. Polotski, *AutonomousSystem for Navigation and Surveying in Underground Mines*. Journal of Field Robotics, 2007, Vol. 24 (10), pp. 829-847.

[13] D. Huber and N. Vandapel, *Automatic Three-dimensional Underground Mine Mapping*,Int. Journal of Robotics Research,*2006* , Vol. 25, No 1, pp. 7-17.

[14] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis and H. Choset, *The Hierarchical Atlas*, IEEE Transaction on Robotics, 2005. Vol. 21, No. 3, pp. 473-481.