ABSTRACT

In this paper the TAS-I (Thales Alenia Space Italy) Test Bench for Robotics and Autonomy (TBRA) is presented. It is based on a flexible and modular software architecture (Framework Engine) in which each functional module (representing the GNC subsystems) implements a key functionality of the GNC (Guidance Navigation and Control). Modules communicate by means of standardised interfaces designed for exchange of necessary information among the modules composing the entire system. This approach permits the interchange-ability of each subsystem without affecting the overall functionalities of the GNC system. In this paper the TBRA system, together with the implemented functional modules will be described. Tests results will be reported and future development will be discussed.

1. TBRA ARCHITECTURE

The Autonomous Navigation system of a Rover is made of a set of functional inter-communicating modules, scheduled according to the selected navigation cycle (stop and go, perception during motion).

The interfaces between these modules can be standardized, in order to allow:
- Validation and trade-off of different algorithms for the single functional module, based on
  - open loop testing of the single module
  - closed loop testing of the whole navigation cycle exchanging only the single module
- Validation and trade-off of the navigation cycle based on closed loop testing
- Parallel implementation of the functional modules as “black boxes”
- Interchange-ability of each subsystem without affecting the overall functionalities of the Guidance Navigation and Control (GNC) system

Starting from the above assumptions, TAS-I has developed a flexible Test Bench for Robotics and Autonomy (TBRA) for developing and testing innovative GNC systems, with the focus on locomotion, localisation and navigation subsystems. Fig.1 shows the TBRA architecture. Flexibility is achieved by the definition of modular software architecture (Framework Engine) in which each functional module (representing

Figure 1 - TBRA Architecture
the GNC subsystems) implements a key functionality of the GNC. The GNC system is also characterised by an adequate level of autonomy, which will lead, in further studies, to the development of a complete goal oriented mission planning. This deliberative layer, which replaces the executive and planner layers of the classical three layer architecture, is implemented as a single module. Modules communicate by means of standardised interfaces designed for exchange of necessary information among the modules composing the entire system. This approach permits the interchange-ability of each subsystem without affecting the overall functionalities of the GNC system, allowing an easy reconfiguration. Each module can be configured to be local (running in the same processing unit of the framework) or remote (connected to the framework through a TCP/IP connection) by changing just one line of code. This feature is useful to distribute the

2. ROBOTIC PLATFORM

The robotic platform is based on a MobileRobot’s Powerbot commercial base on which the GNC sensors have been integrated. Fig. 2 shows the complete setup of the TBRA Robotic Platform. The PowerBot base comes with 28 sonars, 2 wheels encoders (internally integrated to perform wheels odometry localization), 12 bumpers and 2 commercial Desktop PCs (Pentium Dual Core at 1.66GHz, 2GB DDR2, 80GB HDD) with Ethernet connection. This base hardware has been integrated with the following GNC sensors:

- Localization Sensors:
  - Systron Donner MotionPak II Inertial Measurement unit (IMU),
  - 2 SPI-TRONIC Pro3600 Inclinometers,
  - PointGrey Research BumbleBee2 Stereo Camera (2-eyes), for Visual Odometry (VO)
- Perception Sensors:
  - PointGrey Research BumbleBeeXB3 Stereo Camera (3-eyes),
  - Fraunhofer Institute rotating 3D Laser Scanner (based on sick LMS200)

The 3-eyes stereo camera in mounted on a Directed Perception D46-70 Pan-Tilt Unit position. Moreover, to enhance networking capabilities a ROUTER has been installed ensuring that the antennas are visible in any pose the robot could assume. The robot is powered by two lead batteries (24V 88Ah)

3. FUNCTIONAL MODULES

In the current TBRA architecture the following modules have been implemented:

Figure 2 - TBRA Robotic Platform
3.1. Sensors Modules

The sensors modules are deputed to the acquisition and the conditioning of the raw measures which are read from the GNC sensors embedded on the robotic platform. These modules exploit the Hardware Abstraction Layer (HAL) of the framework to interact transparently with the underlying hardware.

3.2. Actuators Modules

The actuators modules are deputed to send the commands (control references) to the underlying actuators. In case of the PTU module it performs the closed loop control of the actuator positioning. The platform locomotion actuator module receives the speed and jog references generated by the higher level control module and calculates the corresponding wheels velocities on the basis of the robotic platform characteristics and the given constraints. The communication is performed exploiting the abstraction provided by the HAL.

3.3. Stereo Vision Perception Module

The Stereo Vision Perception module is in charge of the generation of a Digital Elevation Map (DEM) of the surrounding environment. This module receives a “generate DEM” command from the Perception Data Fusion with a specified Field of View (typically 180deg or 360deg) and a number of perceptions to be performed. Then the module computes the needed camera poses (in terms of pan and tilt values for the scan) and commands the PTU to move the 3-eyes stereo camera. Each perception is cut to keep only its major confidence region (1-5m, 60deg FOV) and then is added to a vectorial DEM with a 0.05m x-y discretization. A median filter is then applied to the vectorial DEM in order to generate the local DEM rejecting the acquisition noise. Fig.3 shows an example of a DEM generated by the Stereo Vision Perception module. The output DEM is a 400x400 matrix where each pixel is a 0.05x0.05m region of the environment. X-Y discretization is the result of a trade-off between the sensors accuracies, the total region coverage needing and the amount of data to be exchanged between the modules (and therefore the memory allocation size).

3.4. 3D Laser Scanner Perception Module

The Laser Scanner Perception module is in charge of...
the generation of a Digital Elevation Map (DEM) of the surrounding environment. This module receives a “generate DEM” command from the Perception Data Fusion module and starts collecting data from the rotating 3DLS. The output of such acquisition is a DEM which is very detailed in the first 2-3m but becomes sparse at greater distances. This is due to the angular resolution which results from the 3DLS rotation speed. In order to have a total scan time of maximum 40s the 3DLS angular resolution is limited to 0.25deg and the speed is 10deg/s. Vertical scanning is performed at 0.5deg resolution over 200 points from +40deg to -60deg in 26.5ms. Fig.4 shows an example of DEM generated by the 3DLS Perception module.

3.5. Perception Data Fusion Module
The Perception Data Fusion (PDF) module absolves two main tasks: the merging of DEMs generated from each Perception Module and the noise reduction of the resulting DEM. The PDF module can be configured to rely on only one sensor (no merging needed) or to merge the DEMs of the Stereo Vision Perception module and the 3DLS Perception module giving higher priority to the former or the latter as preferred by the user. The merging process takes the higher priority DEM (assume that is the 3DLS one) as base DEM then fills the unknown regions with the lower priority DEM (Vision in this case) in order to reduce the unknown points as much as possible. Despite the former filtering that is performed by each perception module, the experimental results shown that this merged DEM is still affected by a noise similar to “salt-and-pepper” noise. This kind of noise needs a specific filtering which has been designed taking into account the environment where the rover has to operate. The basic assumption is that the main obstacles in a planetary environment (e.g. Mars) are constituted by rocks and boulders. The filter has been tailored considering the obstacles dimensions and proportions, resulting in the deletion of thin and tall objects (e.g. thin columns) from the map. Fig.5 shows an example of output DEM after the filtering process. The output DEM is a 800x800 matrix where each pixel is a 0.05x0.05m region of the environment.

3.6. Traversability Module
The Traversability Module is in charge of generating the Navigation Map (NavMap) from the current perceived local DEM and the stored NavMap. The resulting NavMap is a 1024x1024 matrix where each pixel is a 0.05x0.05m region of the environment. Fig.6 shows the Traversability Module flow. The flow is split in two independent processes: discontinuities detection and slopes detection. The first aims to generate a NavMap which costs are related to the soil discontinuities (steps). Whenever a discontinuity is greater than the maximum negotiable discontinuity of the rover the corresponding map region is marked as obstacle. Then all the obstacles are grown by the half size of the rover (0.45m) to ensure that it can be regarded as a point in the subsequent path planning process. The second process is the slopes detection. Slopes detection is performed by evaluating terrain steepness every 30deg around each DEM point,

Figure 5 – 3DLS Perception Module Filtered DEM

Figure 6 – Traversability Module Flow
hypothetically placing the rover in such point and orienting it in the various directions. The costs are calculated on the basis of the highest relative steepness of opposite areas. Anytime the steepness is greater than the rover maximum negotiable slope the point is marked as obstacle. The two partial NavMaps are then merged to obtain the local NavMap related to the current perception. Then the older NavMap, containing the “history” of the rover exploration is aligned to the current rover position and merged with the current NavMap. Assuming that the last perception is the most reliable, the overlapped points are replaced with the newest values. Current NavMap unknown regions are filled with the old NavMap known regions. Finally a cost function is applied around all the obstacles ensuring that the rover keeps itself as far as possible from hazardous regions. The whole traversability process takes maximum 350ms to be completed.

3.7. Path Planner Module

The Path Planner Module is in charge of generating a path (sequence of waypoints) from the rover current position to a given goal across the full NavMap (current plus old). The path planning process is divided in two main steps: “coarse path finding” and “fine path finding”. In the first step the NavMap is subdivided in quadrants and stored in a QuadTree data structure which intrinsic characteristics are exploited to minimize memory consumption and maximize path planning algorithm performances. After this first subdivision the NavMap is composed by big-tiles (from 0.40m to 0.80m by default) each one containing the mean of the costs if the region is completely navigable. If unknown points or obstacle points falls inside a region, such region is entirely regarded as an unknown or obstacle region. The coarse path is then calculated running an ad-hoc algorithm which derives from A*. If a coarse path from the rover current position to the goal exists, a subgoal is then identified among its waypoints list. Subgoal selection is prone to the maximum distance which the rover can cover between two successive perceptions (new DEM generation), which is the result of a trade-off between the perception sensors, localization sensors and locomotion actuator performances. The maximum distance at current stage is 3m. After subgoal identification the NavMap is windowed around the rover current position and the identified subgoal, than a complete QuadTree (maximum depth, 0.05m granularity) is grown on this sub-NavMap and the fine path is calculated using the same ad-hoc path planning algorithm. The whole path planning process takes a maximum of 500ms to be performed. Fig.7 shows a the path followed by the rover during an outdoor test performed in TAS-I over the merged NavMap.

3.8. Locomotion Module

The Locomotion Module is in charge of generating the speed and jog reference values to follow the path provided by the Path Planner module. It performs a close loop control reading the feedback from the Localization Data Fusion (LDF) module which provides the rover pose estimation (x,y,z,roll,pitch,yaw).

\[ \text{speed} = KX e_x \]  
\[ \text{jog} = (KY e_y)^2 e_y + KTH e_\theta + KITH i_\theta \]

Eq.1 and Eq.2 shows the linear control law used to calculate speed and jog references.

The displacement and heading errors are calculated as stated in Eq.3, Eq.4 and Eq.5:

\[ e_x = (x_c - x_e) \cos \theta_e + (y_c - y_e) \sin \theta_e \]

Eq.8 shows the control law being applied to the rover. The feedback is the actual path followed by the rover.

\[ u = \text{speed} \cos \theta + \text{jog} \sin \theta \]

The control law is applied to the rover to follow the path generated by the Path Planner Module.
While in movement the locomotion module reads the rover pose and tries to correct the trajectory in order to keep the robot within a boundary of 0.20m around the nominal path. If the rover position exceeds the maximum allowed error (e.g. due to slipping) the Locomotion module stops the rover movement and reports an error to the Navigator which performs the proper recovery action. The Locomotion module performs also Hazard Detection in terms of slopes and unexpected obstacles. In both cases an error is reported to the Navigator module which provides the proper recovery action. Fig.8 shows the concept of the path following constraints.

3.9. Odometry Modules

The WO module is in charge of providing the differential rover displacement and heading \((x, y, z, \text{roll}, \text{pitch}, \text{yaw})\) between the current rover position and its last position. The IO module is in charge of providing the differential rover attitude (roll, pitch and yaw) between the current rover attitude and its last attitude. The VO module is in charge of estimating the complete rover differential pose \((x, y, z, \text{roll}, \text{pitch}, \text{yaw})\). Each module output is provided to the LDF module which is in charge of estimating the global rover pose.

3.10. Localization Data Fusion Module

The LDF module is in charge of estimating the 6-dimensions rover pose \((x, y, z, \text{roll}, \text{pitch}, \text{yaw})\) by merging the data coming from the three odometry sensors. The data fusion is performed using an Extended Kalman Filter which takes as input the differential estimations of each localization module together with their confidence parameter (related to each sensor error covariance).

3.11. Navigator Module

The Navigator module is in charge of triggering and monitoring the execution of all the lower level modules (Perception, Traversability, Path Planning, Locomotion and Localization). It executes the medium-level commands received from the Autonomy module and performs a first low-level Failure Detection, Isolation and Recovery (FDIR) of the Path Planning and Locomotion modules. Path Planner errors result in a new DEM generation request to the PDF. If the Path Planner generates three consecutive errors the failure is reported to the Autonomy Module. The Locomotion Module can return two types of errors which need to be recovered: “Rover out of path” error and “Hazard Detection” error. The former is recovered by stopping the rover motion, generating a new perception while biasing the localization sensors. The latter is handled with a reactive navigation action which moves the rover away from the unexpected obstacle and then performing a new DEM generation. All the other errors reported to the lower level modules are reported directly to the Autonomy Module for higher level handling.

3.12. Autonomy Module

The Autonomy module is in charge of translating the high-level commands (e.g. Explore a specified area) into a sequence of medium level commands that are forwarded to the Navigator. It also performs FDIR for all the unhandled errors reported by the Navigator module. In its current state the Autonomy module generates a spiral path inside the area of interest, define by the explore command parameters (central point and radius of a circular area to be explored). The spiral path has been designed in order to cover the whole area of interest. If one point of the spiral is not reachable (e.g. is inside a non-navigable area) the point is discarded and the next point is taken into account. Fig.9 shows how the modules interact while executing the “Explore Area command”

4. TESTS RESULTS

Thanks to the modularity of the underlying framework, each module has been tested as a separate entity (standalone tests). Then each standalone module has been ported in the corresponding TBRA module and the closed loop tests have been performed. These tests leded to further algorithms optimization to improve performances in time and accuracy of each subsystem.
The early performance tests have been done on paths with length <20m. Further tests are ongoing and will result in precise characterization of the system performances.

5. CONCLUSIONS

The 2010 output of the TBRA R&D project is a rover capable of both indoor and outdoor autonomous navigation relying only on its onboard GNC sensors (no a-priori knowledge of the environment is needed).

6. FUTURE WORK

This study will provide an implementation of the test bench including the complete framework and a set of new functional modules within 2011. All these subsystems will be updated with new more performing features in further developments foreseen in the next years.

The output of the 2011 study will lead to the following branch of research / products:
- **Functional Verificator** of GNC algorithms for performance evaluation, optimisation and open/closed loop testing, necessary during proposal and A & B project phases;
- **Test Bench** for development of new GNC algorithms based on new GNC sensors, suitable for space applications;
- **Test Bench** composed by 2 cooperating robots for robot swarms studies.

The activities foreseen for the 2012 R&D will be related to the coordination of multirobot systems (robot swarms), using the two platform developed during 2011. The "swarm" approach is perfect for space exploration, as a crowd of smaller, simpler devices can cover far more ground than even the most complex single unit.

7. REFERENCES