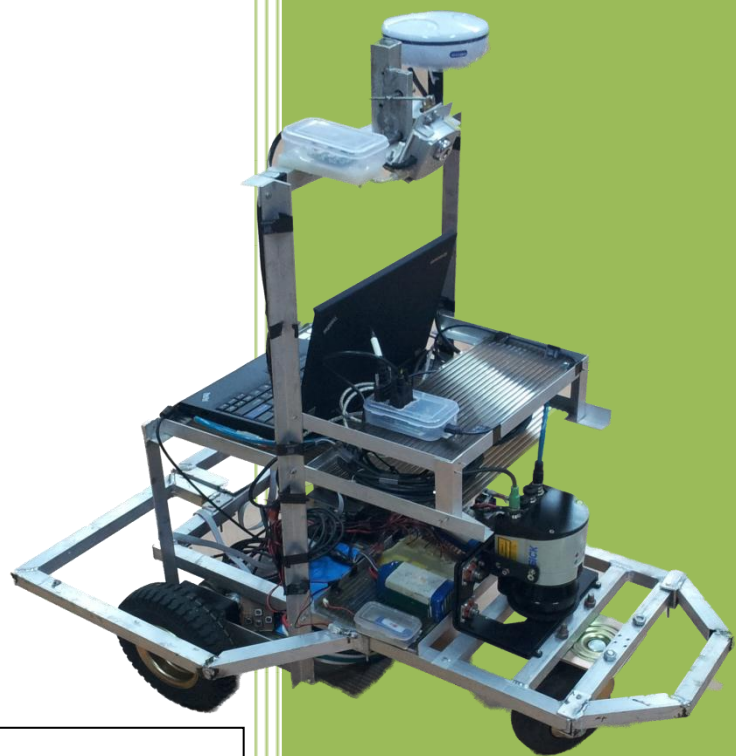


# 2012

## PUSHPAK IIT BOMBAY



### Faculty Advisor Statement:

I hereby certify that the development of the vehicle described in this report, Pushpak, has been significant and is consistent with the effort required in a Senior Design Course.

Prof. C. Amarnath

Prof. S. N. Merchant

Team IGVC

IIT Bombay

3/5/2012

## CONTENTS

|     |                                |    |
|-----|--------------------------------|----|
| 1   | Team Overview .....            | 2  |
| 2   | Design Process .....           | 3  |
| 3   | Sponsorship And Funding .....  | 3  |
| 4   | Hardware .....                 | 4  |
| 4.1 | Hardware Innovation .....      | 5  |
| 4.2 | Fabrication and Assembly ..... | 6  |
| 4.3 | Mechanical Design .....        | 6  |
| 4.4 | Electrical Design .....        | 7  |
| 4.5 | Electronics and computing..... | 8  |
| 4.6 | Sensors .....                  | 9  |
| 4.7 | Safety .....                   | 10 |
| 5   | Software .....                 | 11 |
| 5.1 | Software Innovation.....       | 11 |
| 5.2 | Platform .....                 | 12 |
| 5.3 | Obstacle Detection.....        | 13 |
| 5.4 | Lane Detection .....           | 14 |
| 5.5 | Navigation .....               | 14 |
| 5.6 | Mapping .....                  | 15 |
| 5.7 | Path Following.....            | 15 |
| 6   | Conclusion.....                | 15 |



## 1 TEAM OVERVIEW

The Intelligent Ground Vehicle Challenge provides an excellent opportunity for the student fraternity of the Indian Institute of Technology Bombay (India) to explore the design and implementation of advanced unmanned systems. The IIT Bombay team consists of students of various engineering disciplines, many of whom have worked on the Autonomous Robot Design Challenge organized by the Defense Research & Development Organization (Ministry of Defense, India) and bring in experience in mechanical design as well as autonomous navigation with obstacle detection.

The team has 12 student members from various engineering disciplines. Considering the small size of the team, we opted for a flat team structure. Following the “Guru-Shishya” system of ancient India, each group has one senior experienced student with one or two understudies.

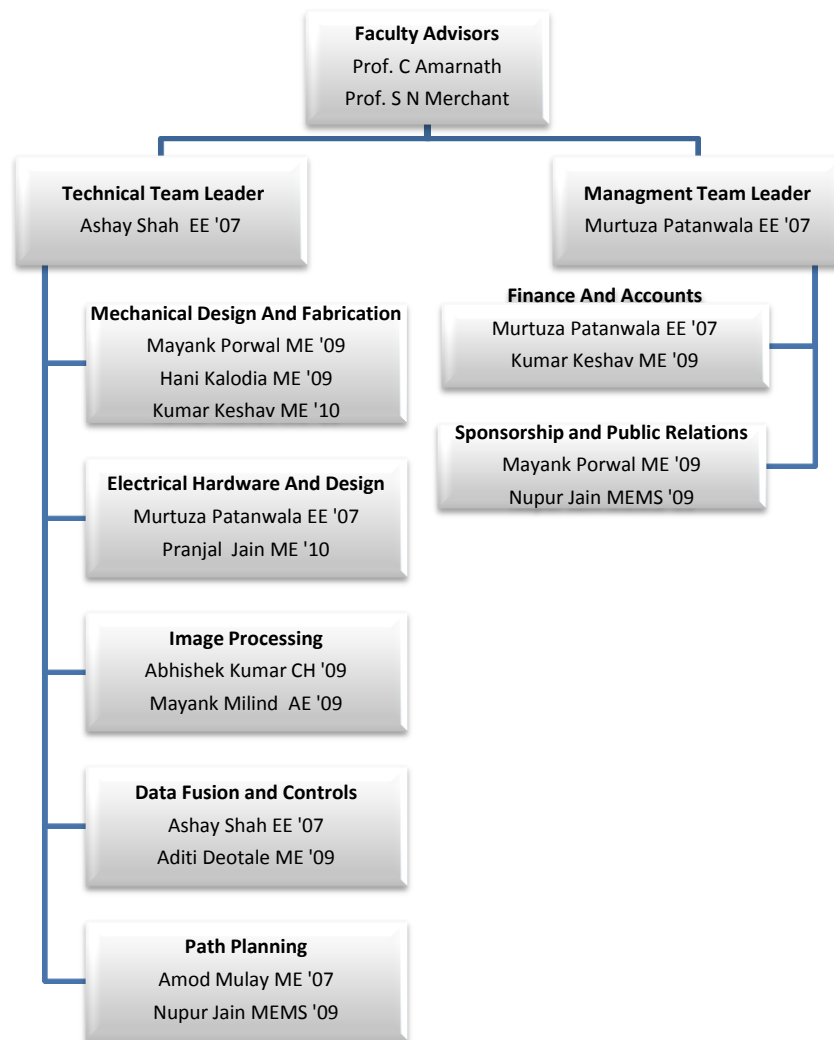


Figure 1 Team Structure



## 2 DESIGN PROCESS

The design process is described in figure 2. In our experience with practical research vehicles, we found that the “Serial Prototyping” method works best for these types of systems. The designing is done incrementally starting with Proto 0, a LEGO design of the mechanical system.

While the prototype is tested for reliability of installed systems, additional systems are designed and simulated in parallel. During the creation of next prototype, all successfully simulated systems are built and implemented.

With this method, the cost of building the machine is reduced considerably as the probability of a complete redesign is very low. Since, at any given time, one version of the machine is always functional, the development is always “forward” with very little downtime.

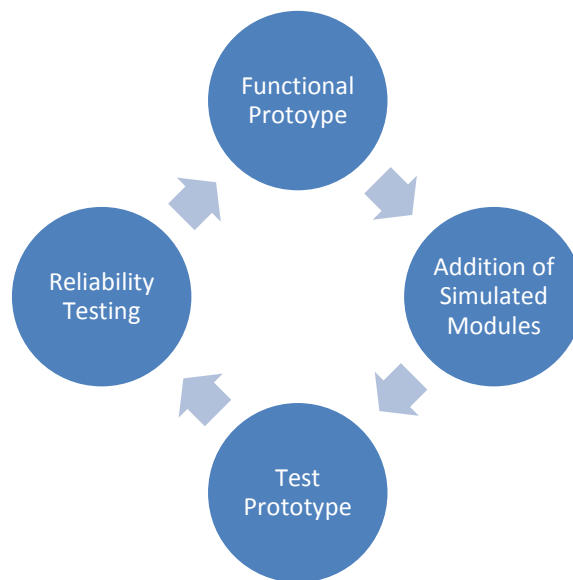


Figure 2 Design Process

## 3 SPONSORSHIP AND FUNDING

The team aims to form a research group for autonomous vehicles at IIT Bombay and implement the technologies developed during IGVC in the real world. IIT Bombay is our chief sponsor and provided the funding for machine construction, design and transport. For our sensor requirements, we approached some leading corporations and obtained complete or partial sponsorship.

We are looking forward to partner with an Indian conglomerate and develop an autonomous transport vehicle.



## 4 HARDWARE

Pushpak measures 3 feet by 2 feet. Its height is variable to allow for increase in camera vision range. The unladen weight is less than 20kg. Pushpak uses a three wheeled platform, which is stable and ensures full contact even on uneven terrain. It uses a differential drive platform using two driven motors and one freewheeling castor. This allows for in place turns for full maneuverability while allowing for simple control algorithm. Pushpak has 10 inch pneumatic wheels for shock absorption and better grip. The ground clearance is 5inches.



Figure 3 CAD Rendering of Pushpak

The entire machine was designed using Solidworks, a CAD modeling software of Dassault Systems. This reduced the prototyping time as many problems could be foreseen and avoided before actual fabrication started. The net weight, the center of mass and the moment of inertia of the machine were known beforehand.

| Item  | Actual Cost (USD) | Team Cost (USD) |
|---|-------------------|-----------------|
| Mech-tex DC Geared Motors (DC52)            | 100               | 100             |
| Wheels & Castor                             | 60                | 60              |
| Roboteq HDC2450 Motor Controller            | 645               | 645             |
| Li Polymer Batteries                        | 200               | 200             |
| SICK LMS 111 Laser Range Finder             | 4300              | 4300            |
| Hemisphere A100 Crescent GPS Antenna        | 1500              | 0               |
| PNI TCM3 Digital Compass                    | 1096              | 0               |
| Unibrain Fire-i Digital Camera              | 260               | 260             |
| IFM –RA6012 Quadrature Encoders             | 450               | 0               |
| FTDI-2232H USB-Serial Modules               | 35                | 35              |
| Belkin F5U505 Firewire Express card         | 50                | 50              |
| ThinkPad X220 laptop with Slice Battery     | 1925              | 1925            |
| Hobbyking 2.4Ghz Wireless Remote + Receiver | 25                | 25              |
| Raw Material for Chassis                    | 500               | 500             |
| Miscellaneous Hardware                      | 300               | 300             |
| <b>Total</b>                                | <b>11446</b>      | <b>8400</b>     |

Table 1 Cost Analysis



## 4.1 Hardware Innovation

The principle of design of Pushpak has been efficiency. The weight without the payload is 20kg. The main chassis weighs only 3kg. The dimensions of the machine were kept to the minimum as stated by the competition rules. The frugal weight of the machine meant that power required to drive the machine would be much lower than the standard in this competition. Our drive system uses only about 40W during normal operation. This has a positive feedback on efficiency as low power drive also implies a smaller battery. By using high density lithium polymer batteries, we are able to crank out long drive times out of a battery pack weighing only 1.5 kg.

Another concept possibly unique among the teams is the use of “crumple zones”. Inspired by their namesake in cars, these bumpers are riveted to the main chassis. In case of a collision, these will bear the brunt of the impulse while the main chassis is unaffected. Thus, we were able to reduce the strength specification of the chassis. Further, the bumpers can easily be replaced with spares within minutes in case of a collision.

In keeping with the green nature of our machine, we have installed a solar panel on our machine. The solar panel has a power output of 45W. Although this power output is not enough to run the entire machine, it is capable of completely running either the drive or the sensors on its own. Thus, using the panel further increases the drive time of Pushpak in between recharges.

Pushpak is to be transported over 13000Km from India to the US. The small size and low weight of the machine coupled with our revolutionary packing system means that Pushpak can be packed into two standard size suitcases, each weighing less than 15kg. Further, it can be deployed and fully operational within 30 minutes of unpacking. This reduces its environmental footprint.

We have created a very simple but effective method of isolating the sensor circuit from the high power drive noise. Thanks to the low weight of the batteries, we have been able to use different power sources for the motor drive and the sensors. Since these power supplies are completely isolated, the mutual noise coupling is very low.



## 4.2 Fabrication and Assembly

As mentioned earlier, Pushpak was completely designed and simulated before the actual manufacturing began. It is manufactured using aluminum alloy extrusions. These provide high strength to weight ratio for very low cost.

The main chassis is welded. This part supports the payload and has the motors attached. Most of the forces acting on the machine are handled by this part. Other add-ons for the sensors are riveted and can be easily adjusted.

A tower has been built on the chassis. This tower houses all the sensors, providing them with unobstructed view of the sky and the surroundings. The freewheeling castor is attached to a cantilever to increase the wheelbase and thus the stability. This part folds inside the sensor tower when packed, allowing for small travel size.

Polycarbonate sheets have been installed over the aluminum sections throughout. These sheets not only improve the aesthetics of the machine, but also provide protection from natural elements. In some cases, the sheet has also been used as an insulated platform for the electronics

## 4.3 Mechanical Design

The maximum load on the machine is during a 15° incline. Assuming a total weight of 30 kg, and a wheel radius of 11 cm, the load torque on each motor is

$$\zeta = \frac{30 \times \sin 15 \times g \times 0.11}{2} = 4.2 \text{ N} - \text{m}$$

The machine has to achieve an average speed of 0.44m/s. We designed the machine to have a max speed of four times this value. Hence, the speed of the motor at rated load is

$$\omega = \frac{0.44 \times 4}{0.11} \times \frac{60}{2 \times \pi} = 152 \text{ rpm}$$

We used a locally manufactured DC geared motor to fulfill these requirements. The motor specs are

|                     |                 |
|---------------------|-----------------|
| <b>Rated Speed</b>  | <b>5000 rpm</b> |
| <b>Stall Torque</b> | <b>0.58N-m</b>  |

Adding at 30:1 reduction gearhead ensures matching of our requirements.



## 4.4 Electrical Design

Power requirements for all the electrical components used is analyzed below

| Average Power Drawn (W)           | Peak (W)   | Idle (W)   |
|-----------------------------------|------------|------------|
| Driving Straight (60% Duty Cycle) | 43.2       | 0          |
| Integrated Turning                | 13.2       | 0          |
| Climbing (15° Incline)            | 43.2       | 0          |
| Drivetrain Weighted Average       | 34.8       | 0          |
| Hemisphere A100 GPS               | 2          | 2          |
| PNI TCM3 Magnetometer             | 0.1        | 0.1        |
| SICK LMS111 Lidar                 | 25         | 25         |
| ThinkPad X220 Laptop              | 60         | 60         |
| RA6011 Encoders (2 No's)          | 3.6        | 3.6        |
| Unibrain Fire-i-Digital Camera    | 1          | 1          |
| Miscellaneous Electronics         | 10         | 10         |
| <b>Total</b>                      | <b>137</b> | <b>102</b> |

Table 2 Power Requirements

While creating the power distribution network, we considered the following major issues

1. Battery life
2. Voltage input limits of different sensors
3. Noise coupling between signals
4. Safety (Domino effect) and Reliability
5. Battery weight
6. Hot swapping of batteries

With these requirements in mind, we came up with a network with three different parts – sensors, drive and computation. The specs of each system are given in figures 4, 5 and 6.





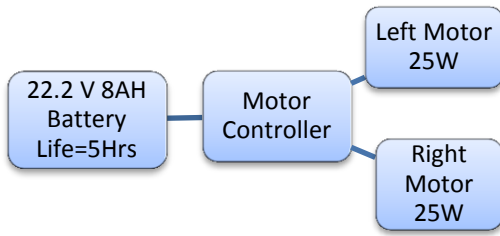


Figure 4 Computation Power Distribution Network

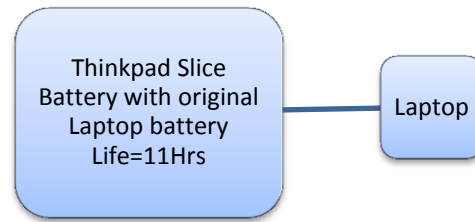


Figure 5 Drive Power Distribution Network

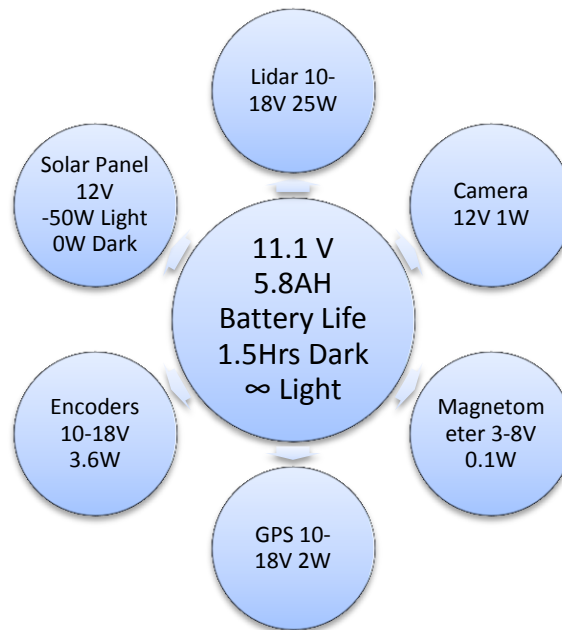


Figure 6 Sensor Power Distribution Network

### 4.5 Electronics and computing

A ThinkPad X220 Laptop is used to communicate with all the sensors, process all the data and control the motors. The laptop was custom configured with an Intel Core i-7 2.8 GHz processor, 6 GB RAM and a 160GB Solid State Drive according to the computational intensity of the algorithms used. A USB to high speed serial development module (FT2232) from FTDI is used to add two serial ports to the laptop. The GPS module and the magnetometer module are connected to the serial ports.

The 54mm express card slot in the laptop is used to connect the F5U505 Belkin Firewire Express card which adds two 400Mbps firewire slots to the laptop. A Unibrain Digital Camera is connected to the firewire port.



The SICK LMS111 Lidar has a serial as well as an Ethernet interface. Considering the high data output of the LIDAR, the Ethernet interface is preferred. This also eliminates the need for another USB to serial convertor.

The Roboteq HDC2450 motor controller has an inbuilt 32-bit microcomputer which is used to connect to the computer via the USB or serial interface. The controller also has Digital/Analog input/output pins which are used to connect the emergency stop button, flash light, wireless receiver channels and the quadrature encoder outputs. An internal script is loaded in the controller which processes the data from the various input/output pins and either transfers it to the computer or directly commands the motors. The controller is connected to the laptop via the USB port and a Labview program is used to transfer the motor PWM values to the internal script. The script measures the internal motor and battery current and voltages and transfers the value to the Labview program via the USB port.

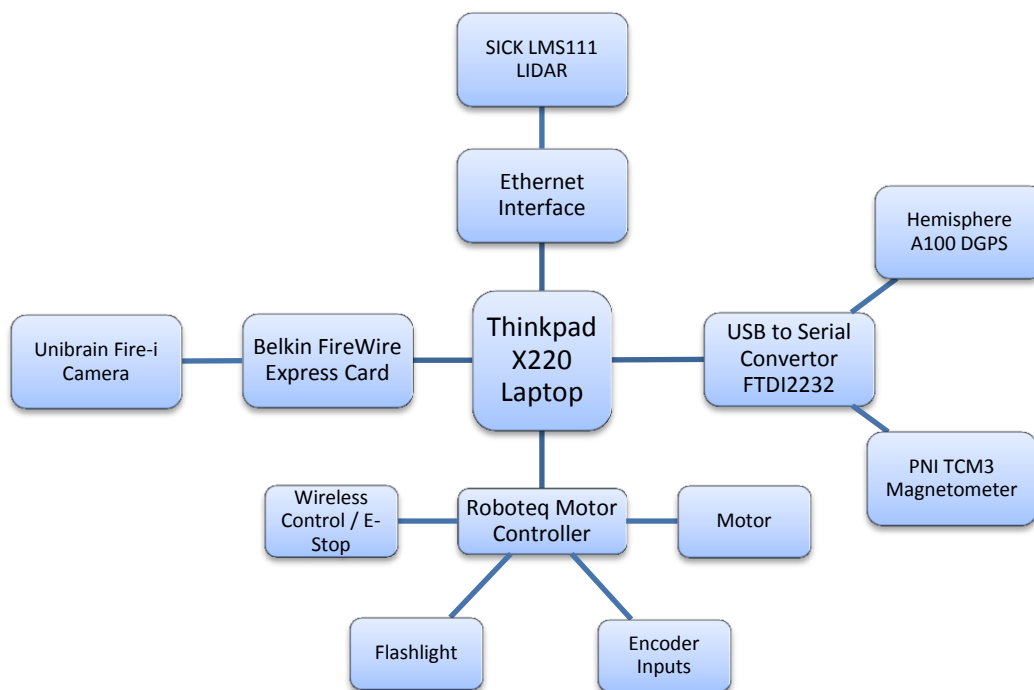


Figure 7 Electrical Hardware Network

#### 4.6 Sensors

A Unibrain Fire-i Digital Camera has been used for lane detection and is mounted near the top of the sensor tower. The camera mounting has a mechanism which allows us to vary the pitch and hence



the field of view. Although the minimum detection distance is 1m, the camera is mounted in such a way that this undetectable zone is contained entirely within the robots physical envelope.

A Hemisphere A100 GPS module, located at the top of the machine, provides global position data in form of NMEA sentences at a maximum frequency of 20Hz with an accuracy of 60 cm via the RS232 serial connection.

A PNI TCM3 3-axis compass module provides tilt compensated high resolution digital compass heading with 0.5° accuracy.

For precise local motion estimates the machine uses two IFM RA6011 quadrature encoders, each mounted at the back of the motor gearbox. The encoders connect to the input/output pins on the Roboteq controller and have a resolution of 200 counts per revolution.

#### 4.7 Safety

The machine is equipped with a red colored hardware emergency stop switch which is mounted at the rear of the machine. When activated the Estop disables the input signal to the Roboteq controller, thus causing the machine to come to an immediate stop.

The wireless Estop is implemented using a 6-channel Hobbyking 2.4Ghz wireless remote which can even navigate the machine remotely. The 2.4 GHz receiver connects directly to the Roboteq controller and the wireless Estop operates in parallel with the emergency manual Estop.



Figure 8 Layers of Protection



## 5 SOFTWARE

### 5.1 Software Innovation

One of the major steps on the software side was the use of a full 3D physics simulation for testing and development of control as well as navigation algorithms. Generally, the problems running a physical machine do not surface until the machine is actually built and tested. However, since the building and integration of both mechanical and electronics can take considerable time, software development times are strongly affected. To overcome this, we used The Open-Source Racing Car Simulator (TORCS) for testing of the code. Thus both the hardware and the software were developed in parallel.



Figure 9 TORCS Racing Screen

Since we are using a single computation system, we had to make sure that we are very efficient in using computation time. To avoid slow response of the system, we came up with a novel system. In this system, the control algorithm and the sensor data accumulation system run independent of each other. To get max efficiency, the sensor data accumulation is run on a timed loop and is unaffected by control computation intensity. The timer value is varied depending on the sensor to get maximum efficiency. For example, the encoder values need to be serviced a lot faster than the GPS value.

To implement the above system, a shared variable library is used. Each sensor is run by a virtual instrument which periodically stores the relevant value in the library. The control algorithm accesses the values directly from the library. Thus, delay in any sensor system will not affect the control algorithm. Similarly, control algorithm delays do not affect data acquisition.

A unique feature of our algorithm is the masking of camera values based on obstacle input. The algorithm masks areas known to be obstacles from the camera frame. Thus, white colored patches on the obstacles are masked beforehand. Line detection is now much simpler as it involves only the detection of white colored pixels.



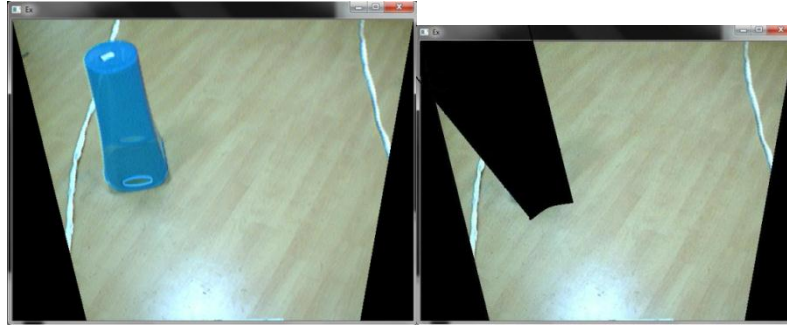


Figure 10 Original Image Figure 11 Masked Image

We have created a separate debug mode for the entire system. In this mode, while the path planning and data acquisition systems run normally, the control algorithm accepts inputs from a remote control instead of the path planning algorithm. This allows us to put the system in various situations manually and check the response of the algorithm. The same mode also doubles up as the wireless E-stop.

## 5.2 Platform

NI Labview runs the main path planning and data acquisition programs. This forms the backbone of the autonomous navigation. Labview has a very easy to use GUI and allows for combining different codes very easily using its nested virtual instruments structure. NI's visa system allows for creation of interface between the hardware and software.

Labview can also import C codes. This is done by creating dynamic linked libraries (DLL's) and importing them into Labview. The advantage is that any standard code can be inserted into the environment and used without complications. Since most of the team had experience with C code, a large part of the algorithm was implemented via this route.

Image processing is done efficiently using Intel's OpenCV library. This is an established image processing platform and includes most of the structures and algorithms associated with image processing. OpenCV library is used in conjunction with C++ coding to create the entire lane detection algorithm.

C++ coding aforementioned was done using the Dev-C++ IDE.



The Open Source Racing Car Simulator (TORCS) is a 3D physics simulation engine intended for race cars. We modified the track to match that of the IGVC. TORCS provides the image of the front camera, the position of the machine and the obstacle map to the “driver”, which in this case is the algorithm. These values simulate data obtained from the camera, GPS/Magnetometer and LIDAR respectively. Hence, TORCS is an exceptional platform for autonomous vehicle testing. It not only simulates the path finding algorithm but also allows for testing of both the data acquisition and the control code.

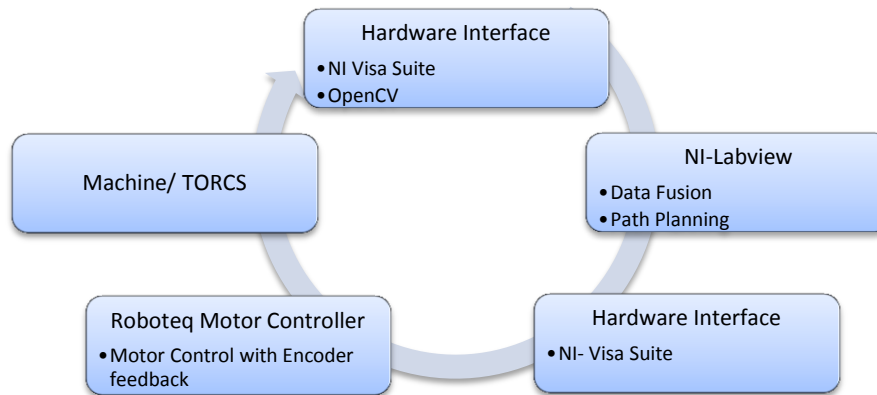


Figure 12 Platform Structure

### 5.3 Obstacle Detection

A SICK LMS111 LIDAR sensor serves the duties of obstacle sensing aboard Pushpak. The LIDAR is extremely accurate and reliable, and is used extensively for industrial and automotive applications. The scanned data from the sensor is available as distance values at every  $0.5^\circ$ . This data can be refreshed at rates as high as 50 Hz.

The amount of data involved in this mapping is very large. To reduce the data influx, two methods are employed in parallel. The first involves reducing the refresh rate of the LIDAR. While the hardware rate cannot be reduced below 25Hz, the electronics just buffers the last available data map. This buffer is accessed at 10Hz and then computed upon. The second method involves the grouping of the data into zones of  $5^\circ$  each. All further computations are carried out using “zone” values only, thus saving a lot of computation time.

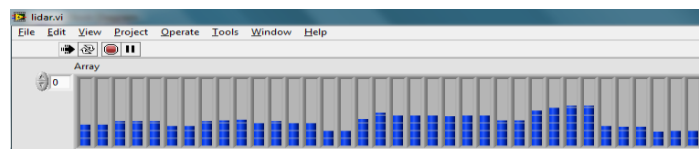


Figure 13 Zonal View LIDAR



While squeezing into extremely tight spaces, the 270° cone of the LIDAR may not be enough for maneuvering. This is especially true in cases where the machine has to retrace its path or has to make zero point turns. For such conditions, Pushpak packs in 3 ultrasound sensors. These ensure full awareness of the surroundings.

## 5.4 Lane Detection

Pushpak has a firewire camera for vision based sensing. The camera allows adjustment of aperture size and shutter speed to capture the best possible image, even in full sunlight.

The image received is first corrected for perspective. This turns the image to a bird's eye view of the area. This makes it much easier to fuse the data with that gathered from the other sensors. Two constants control the nature of the transform and thus any change in either the pitch or yaw can be corrected for promptly.

As mentioned earlier, this image is masked using the position of the obstacles detected by the LIDAR. Since obstacles are masked in the image beforehand, all white colored pixels can be assumed to be part of the line. White colored pixels are marked in this masked image using a linear combination of R, B and G components of the image. This eliminates the need of threshold based detection and ensures detection independent of light conditions.

The image is then divided into left, right and center zones. This adds further reliability by ensuring that a false positive is not detected. A Hough line transform is used to plot lines in the regions and the average line is used as reference for the lane following algorithm. Thus gaps in between lines do not affect the algorithm as long as at least one segment of the line is within the field of view.

## 5.5 Navigation

Position and orientation detection at low frequencies is accomplished by a combination of Hemisphere A100 differential GPS and the PNI TCM3 magnetometer. The differential GPS allows for error as low as 0.6m. The software ensures that loss of GPS signal is detected and appropriately handled. Thus GPS dropout problems are effectively negotiated. The magnetometer module also includes 3-axis accelerometer. Using the extra data, pitch, roll and true north compensation is achieved.

Although the data from the aforementioned systems is accurate over time, its response is very slow. Their limit is easily reached during in-place turns or sudden change in speed. To overcome this problem,



this data is fused with data from the encoders, which have fast response times. The incremental error associated with encoders is then corrected by the values available from the GPS/Magnetometer.

## 5.6 Mapping

The “Map” in the path following algorithm is an array that describes the environment around the machine. The environment is split into a grid of tiles. The map treats both obstacles and lines as inaccessible tiles. Hence, both lane following and obstacle avoidance are part of the same algorithm.

While creating a map of the entire arena would enable extraction of the most efficient route, it is very challenging to implement. A small error in calculating the position and orientation of the machine with respect to the arena would lead to large errors in positioning of the detected inaccessible tiles. Thus, a much more accurate positioning system such as an IMU will have to be used to create a global map. We plan to implement this algorithm for next year’s competition.

Currently, a local map with respect to the machine is created. It holds the location of all inaccessible tiles detected in a 10m zone around the machine at any particular time. This map is refreshed by the path following algorithm as per requirement.

## 5.7 Path Following

The pseudo waypoint is created in the direction of the nearest target just beyond the edge of the local map. The Dijkstra algorithm is used to plot the fastest path to the pseudo waypoint on the local map. This ensures that any false path within the local map is detected and avoided. In case a false path ends beyond the local map, the machine may follow the false path until the dead end or trap enters the local map.

Once a particular path is planned, the system uses a control system using fusion of encoder and GPS/Magnetometer data to control the machine so as to followed the calculate path.

## 6 CONCLUSION

The design and implementation of Pushpak represents a robust and reliablerobotic system that will be competitive at the 2012 Intelligent Ground Vehicle Competition. We believe that this paper demonstrates the innovative problem solving undertaken by this team. Thanks to the IGVC we have gained greater understanding of the engineeringprocess and look forward to competing this year and in years to come.

