



# VT-LEONARDO



## Team Members

Adam Shoemaker, ME

Chuong Nguyen, ME

Labiba Quaiyum, ME

Navneet Nagi, ME

## Faculty Advisor Statement

I hereby certify that the engineering design on VT-Leonardo was done by the current student team and has been significant and equivalent to what might be awarded credit in a senior design course.

A handwritten signature in red ink, reading "Alexander Leonessa", is written over the text.

**Signature**

**Professor Alexander Leonessa**

**Dept of Mechanical Engineering**

## 1. Introduction

The VT-Leonardo team competing this year focused on the design of a robust electrical platform to be used for autonomous 3D mapping project. Even though IGVC was not our main focus, we were able to incorporate several of the design requirements in the platform design. As compared to the earlier team who competed in the IGVC 3 years ago, we have completely redesigned and manufactured our chassis to add a much needed robust suspension system and designed a completely new electrical and software platform to meet the needs of IGVC and 3D mapping project. VT-Leonardo is a differentially steered robot, which uses a NovAtel SPAN-CPT GNSS/INS system for position and attitude feedback. Additionally, the vehicle incorporates an HOKUYO UTM-30LX as well as an Ubiquiti Networks Rocket M-5 unit for long range communication. All processing is done using an onboard Cappuccino PC running a full Windows 7 operating system.

## 2. Team Organization

Our team consists of 4 graduate students. In the table below, we show how our team has been divided into 3 different divisions and the responsibilities were assigned as shown below.

<b>Division</b>	<b>Objective</b>
<b>Chassis and Suspension</b>	Manufacture the vehicle
<b>Electrical Component Design</b>	Design of circuit design, sensors and designing a power supply system
<b>Software and Control Algorithm</b>	Software design, algorithm design and testing

## 3. Design Process

A variety of design methods and tools were employed to facilitate the platform development process. Figure 3.1 shows a general breakdown of the tasks required. We started out by scoping the requirements of our 3D mapping project and IGVC, we then went on to develop our vehicle concept which we tried to make as flexible as possible so it could be used for the IGVC and multiple other research projects. Drawing on ideas from the concept generation, a high-level system design was formalized. Next the detail design was done and the VT- Leonardo was designed to be a four-wheeled vehicle, utilizing two direct-drive electric motors on the front wheels, and two casters in the rear. The new suspension was composed of high strength independent parallel link suspension in the rear, and an air-spring trailing arm configuration in the front. Besides the chassis and suspension system, the battery system was designed. We then proceeded on the systems integration and fabrication phase where we were able to integrate the sensing and autonomy capabilities of VT-Leonardo. From there we started testing and evaluating while redoing numerous electrical and software design to complete the vehicle.

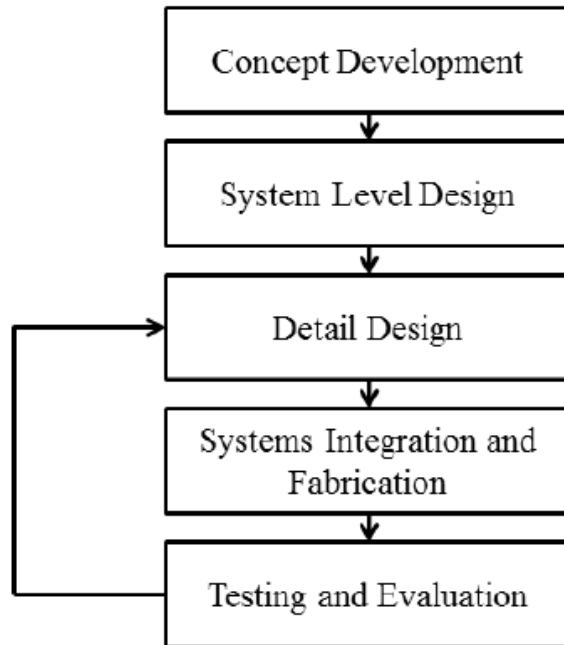


Figure 3.1: Overview of the VT-Leonardo platform design process

#### 4. Innovations

Below is a list of the main innovative aspects of VT-Leonardo's design. They are summarized here, and discussed in more detail in their respective sections.

- Mechanical
  - Robust suspension system
- Electrical
  - Incorporation of relays and switch
  - Solution of floating ground
- Software
  - Predator-prey controller
  - Potential field obstacle avoidance technique

#### 5. Mechanical Design

The VT-Leonardo vehicle was designed to meet specific operational criteria, namely, to: carry a large payload, reach speeds in excess of 10 mph, have an 8 hours endurance under normal operation, have zero turn radius, and be affordable enough to duplicate the design.

**5.1 Vehicle chassis:** The conceptual design of the vehicle specified a two wheel drive, differentially steered arrangement with dual rear casters. The chassis model was then subjected to finite element analysis to determine its performance subject to loading it would undergo during operation. Systems constraints and loads were defined. The constraints as defined by the design are the four wheel

support locations. Therefore, the chassis was constrained in the vertical direction at the four wheel mount locations: two casters near the rear of the frame and two drive wheels at the front of the frame. Loads placed on the chassis would be both static and dynamic, but due to the difficulty in accurately characterizing the dynamic loads resulting from vehicle operation this early in the design process, initial analysis was restricted to static loading. Several load configurations were defined, including those resulting from the heavy and dense battery pack, projected payload layouts, and various drivetrain designs.

According to the results of the finite element analysis, the static stresses experienced by the frame are minimal compared to the yield strengths of commonly available metals like steel and aluminum. Additionally, the maximum stress is concentrated in a small, localized area. This means the worst case scenario is localized yielding, complete failure is unlikely under normal operation.

**5.2 Robust suspension system:** The vehicle previously did not have a robust suspension which caused excessive vibration even on relatively smooth grassy surfaces that eventually lead to a minor structural failure due to material fatigue. When the vehicle was remanufactured, we designed a suspension system to solve this problem. The vehicle was redesigned with a high strength independent parallel link suspension in the rear, and an air-spring trailing arm configuration in the front. The diameter of the drive wheels and the under-chassis ground clearance was increased to improve off road performance, and the overall wheelbase was narrowed to improve maneuvering ability in close quarters. A CAD rendering of the vehicle's design can be seen in Figure 5.1.

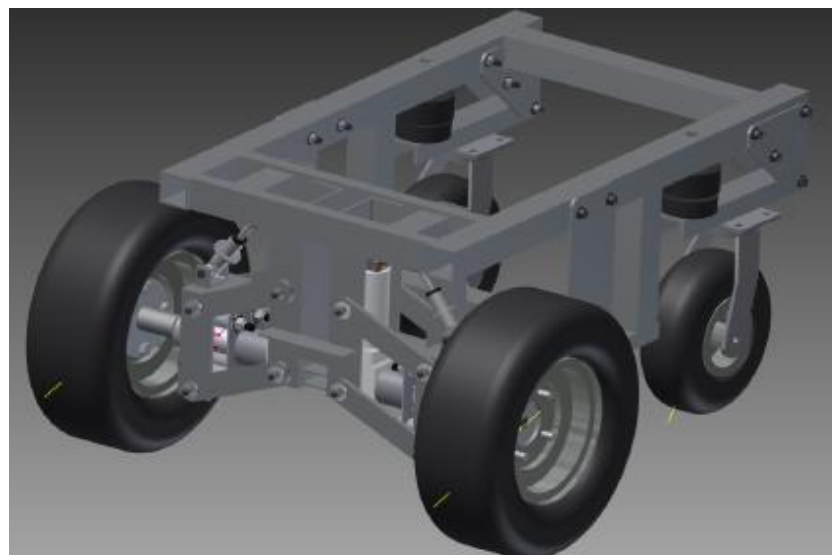


Figure 5.2: CAD model of the VT-Leonardo design, highlighting the parallel link rear suspension.



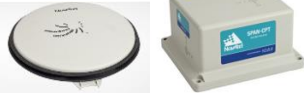
## 6. Electrical Design

The main controller of the vehicle is an onboard Cappuccino PC running a full Microsoft 7 operating system. All the sensors used on the vehicle to obtain the information needed for autonomy

are transmitted via USB cables and/or Ethernet connection. The Ubiquiti Networks Rocket M-5 unit is used for long range communication.

### 6.1 Sensors

In the table below, the sensors installed on VT-Leonardo is listed. The specifications of the sensors and the reason of their respective use is also listed

Sensor	Detection	Specification
 <p>LRF</p>	Obstacle Flag Pole	<p>HOKUYO UTM-30LX</p> <ul style="list-style-type: none"> <li>● Long Detection range: 30m</li> <li>● Wide Angle: 270 degrees</li> <li>● Angular resolution:0/25 degrees</li> </ul>
 <p>Camera</p>	Lane Flag Obstacles	<p>Basler scA1300-32gc</p> <ul style="list-style-type: none"> <li>● Resolutions:1296 x 966</li> <li>● Pixel size(micro-m):3.75 x 3.75</li> </ul>
 <p>GPS,IMU,INS</p>	Location Speed Angle	<p>SPAN-CPT Single Enclosure GNSS/INS Receiver</p> <ul style="list-style-type: none"> <li>● Horizontal accuracy:0.15m</li> <li>● Gyro range(<math>\pm</math>deg/s):375</li> <li>● Accelerometer range(<math>\pm</math>g):10</li> </ul>

### 6.2 Circuit design and system integration

Due to the integration of numerous devices on VT-Leonardo, it was critical we designed a circuit system which would be able to integrate all the needed components effectively. After numerous issues with floating grounds and different components not working, we had to incorporate numerous relays, fuses, inductors and switches in our circuit design.

The whole integrated circuit can be seen in 3 different divisions for simplicity. The first division includes the computer, GPS system, LFR, camera and the ethernet connection box is powered off a 12V bus from the main power supply unit. All these components were kept in one division for safety and because of the common need of a 12V supply. The second division includes the battery charging, battery management system and finally an individual 12V power supply for the battery management system. It includes the circuitry to charge the battery and to manage the 8 cell battery balanced discharging and charging. The last division is responsible for powering the two motors. Certain new things need to be incorporated to the circuit design to enable the functioning and safety of the all the components.

**Safety features:** A relay was incorporated between the powering the main power supply unit (PSU) and the battery in order to secure the division 1 components from any current surges from the battery. This relay is controlled by an e-stop switch and power from the battery management system. Another relay was incorporated between the motors and the battery, which is installed to ensure that the motors do not get power if the energy management system is not power on. Two additional e-stop switches are also incorporated in the design. One is used to control the power to the motors and the second is used to control the power to the energy management system.

**Solution of floating ground:** The motors are grounded physically on to the chassis ground as well as to the physical battery ground. The setup was causing issues with floating grounds and certain sensors not working, as the PWM control from the motor controller was at 20 Hz while the edge frequency of the system was 5 MHz. This major issue was fixed by adding inductors between the motor controller and motors. The inductors allowed us to decrease the edge frequency down and solve the floating ground problem.

### 6.3 Power distribution

The center VT-Leonardo's power system is the two power supplies. One is responsible for powering the computer and all the sensors while the other is responsible for powering the energy management system.

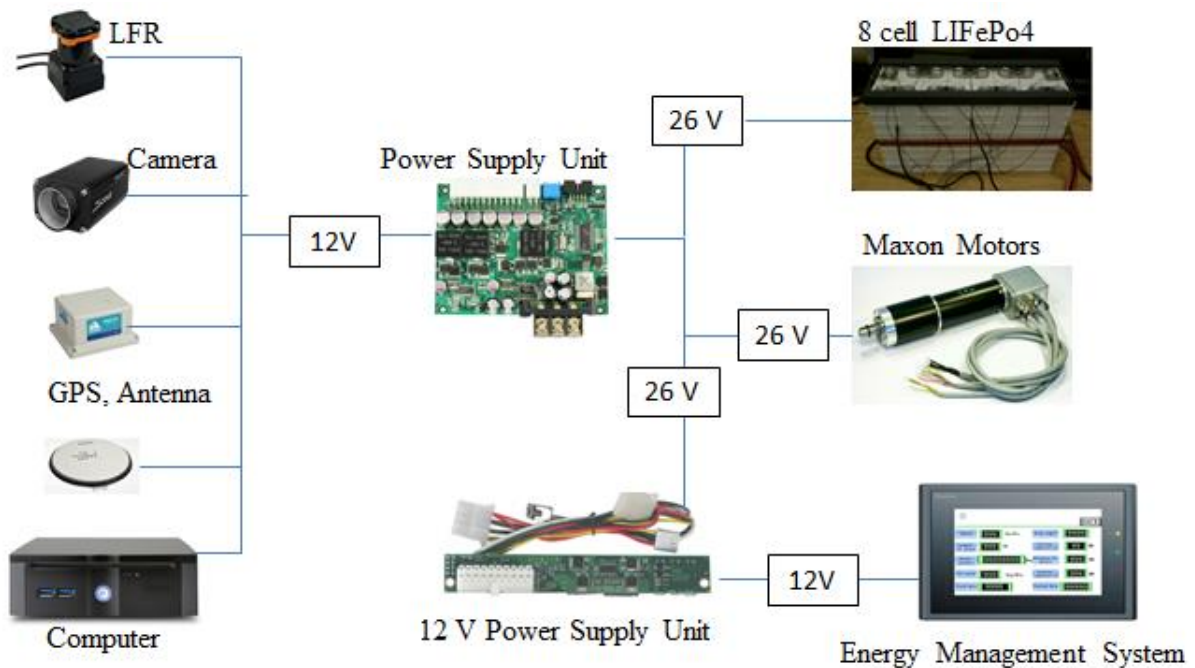


Figure 6.3: Power distribution for VT-Leonardo

#### 6.4: Signal processing distribution

The central point of integration is a Cappuccino PC with a Gigabit Switch. All the communication is done via USB or RJ 45. Cappuccino. The LFR, motor controller and GPS uses USB while the camera, base station uses RJ 45.

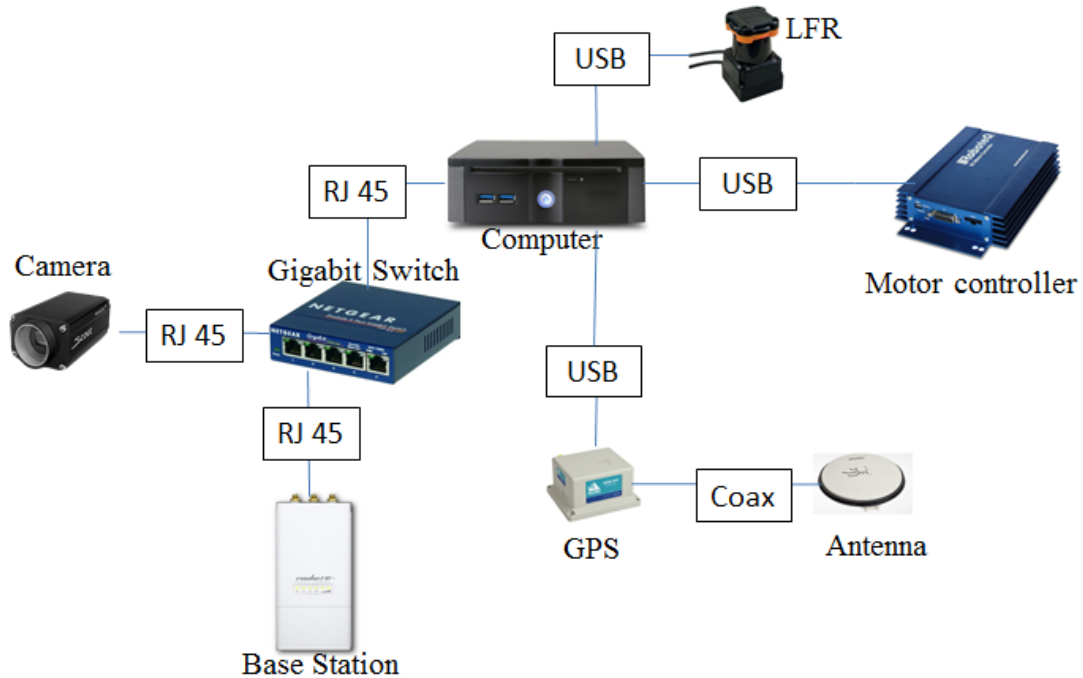


Figure 6.4: Sensor integration for VT-Leonardo

#### 6.5 Energy management system

In order to determine the battery capacity requirement, a test scenario representative of heavy vehicle operation during an 8 hour mission was defined. A dynamic scenario was created in which during an 8 hour span, the vehicle spends 1 hour sitting still, 1 hour running 5 degrees uphill on grass at 10mph, 1 hour running at 5 degree uphill on concrete at 10 mph, 2 hours running on grass surface at 10mph and 3 hours running on at concrete surface at 10 mph. The total energy required for this dynamic scenario was approximately 1800 Wh.

The performance of the potential battery pack designs were calculated of three battery types and compared, LiFePO<sub>4</sub> was proved to be the most effective and hence chosen. Bus bars connected 8 individual LiFePO<sub>4</sub> cells in series for a nominal 25.6 V battery pack. Each cell is also wired separately to an energy management system (EMS) so that the cell voltage can be monitored during both the charging and discharging cycles. The energy management system (EMS) was sourced from Ligoo/Wicom, and includes several components for safe use of the battery pack. The EMS is structured with a Master-Slave(s) architecture, with the master module CCM managing one or multiple slave module DCMs. Each DCM is capable of monitoring the voltage of up to 12 individual cells, as well as two temperature measurements. An open loop hall current sensor is connected to the CCM. It monitors the current drain of the battery up to 1000 A. Using this current and the cell

voltage data, the CCM estimates the real time state of charge with the built-- with the EMS for easy monitoring and parameter configuration.

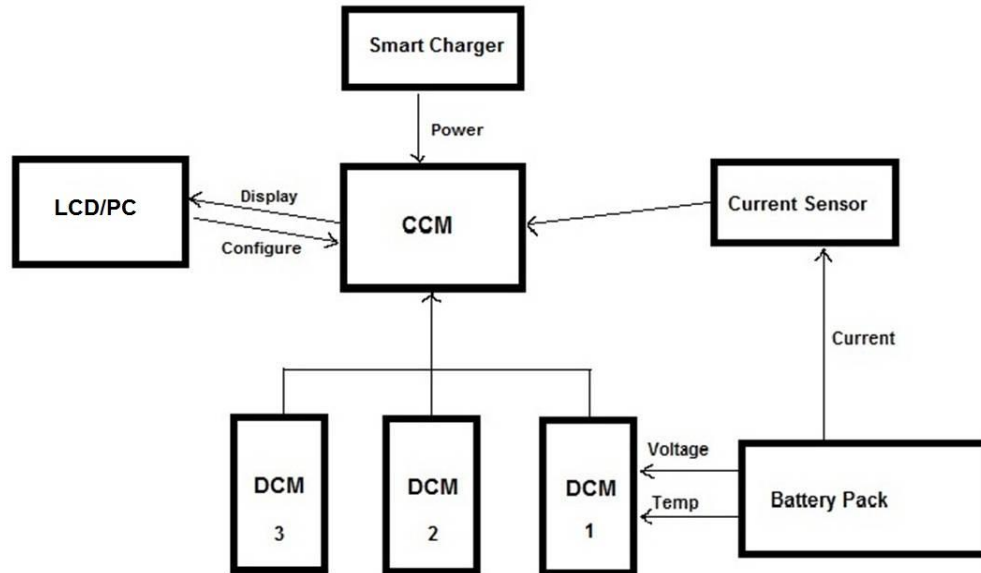


Figure 6.5: System architecture of the energy management system (EMS).

## 7. Software strategy

VT- Leonardo's software system was developed using a combination of National Instrument LabVIEW, Robot Operating System(ROS) and Open Source Computer Vision(OpenCV). LabVIEW is being used for the implementation of most of our controller strategies while ROS and OpenCV is being used for the sensor data manipulation such as LFR and camera. ROS runs on Ubuntu Linux and is an open source collection of software frame for robot software development. It has several drivers such as LIDARs, cameras and GPS in its library. This allowed for faster implementation of our control strategy. OpenCV is a library of programming function which is aimed at real-time computer vision which again allowed us faster implementation for data manipulation from the camera.

## 8. Lane Detection

Lane detection algorithm begins with the use of a simple thresholding technique to identify a grass background. From the resulting binary image, a Laplacian of Gaussian (LoG) edge detection technique is used. After additional processing, the Hough transform is applied to segments of the image. The transform creates a pseudo-spline that identifies the lane within the given image frame. The step by step process for the algorithm is shown below via images.





Figure 8.1: Grass Sample. The above figure demonstrates the area of sampling that is deemed grass. By assumption of the robot maintaining a certain distance from the line, this portion is sampled to determine the true grass properties in the current lighting.

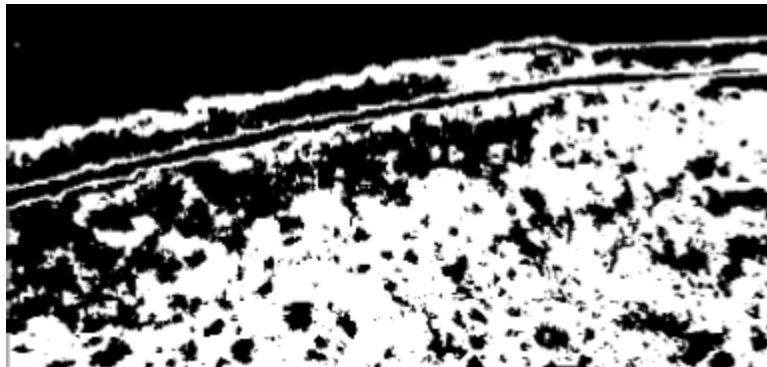


Figure 8.2: Binary Image Result from Thresholding. The above image demonstrates the resultant binary image which is achieved by thresholding based on the values obtained from the sample window in Figure 8.1. The white pixels represent the believed grassy areas.

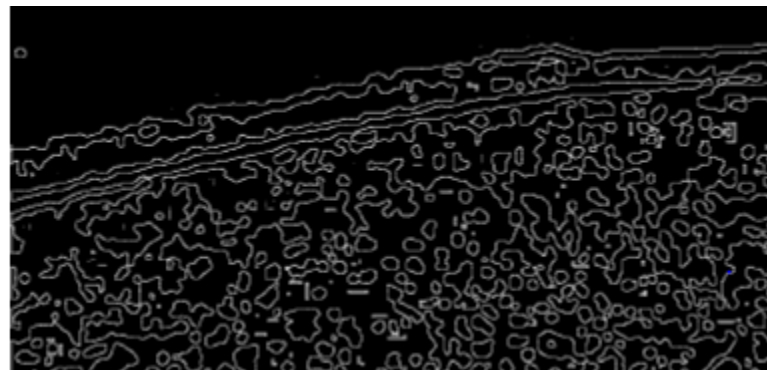


Figure 8.3: Edge Detection of Binary Image. The above image demonstrates the resultant edge detected image using the LoG approach. This image was created using the data from Figure 8.2.

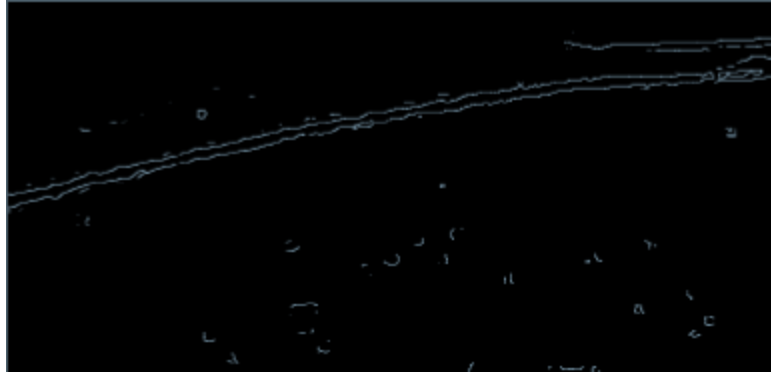


Figure 8.4: Edge Detection with White Properties Incorporated. Using the image given in Figure 8.3, this image is created by incorporating color information from Figure 8.1 to determinate vicinity to white pixels.



Figure 8.5: Lane Detection Result for Test Image. The above image demonstrates the results of the Hough transform conducted on the final edge image given by Figure 8.4. As expected, it detects the lane and accounts for changing geometries.

## 9. Color Detection Method

In addition to a lane detection algorithm, we have been working on a color detection algorithm that we would use for the colored flags. The example showed below is using the color white. In order to detect color, the image is first converted into Lab or HSV color space. Then each pixel is compared with a predefined target color by using the L2 distance (Lab) or block-distance (HSV). Any pixel that has distance less than a threshold value is considered as the object. This algorithm will also be used to detect flags.

In order to detect the road, the white color (line) and green color (Grass) are first detected. The threshold color image will have 3 values: 1 for white pixels, 0.5 for green pixels, and 0 for other pixels that are not white and green. Then, at each white pixel, a small window is formed and its histogram is computed. The histogram has 3 pins corresponding to value [0, 0.5, 1] and is represented by 3D vector. This color histogram is compared with a predefined template histogram by

L1 or L2 norm. Then, a road pixel is detected if the distance between 2 histograms is less than a threshold value.

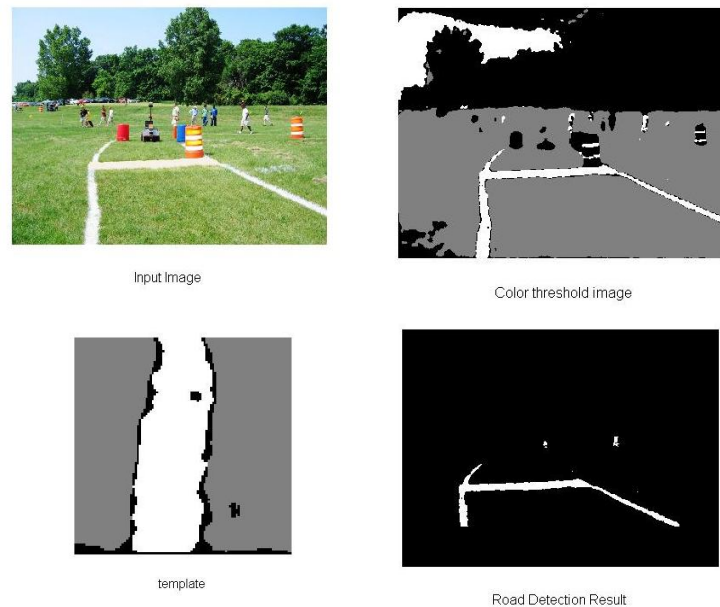


Figure 9.1: Color white and green detection algorithm

In order to map the position of the road to the global coordinate, the output image from road detection step will be divided into many grid cells. Position of each cell is mapped to the robot coordinate by using a calibrated look-up table, and then its location will be mapped to the global coordinate by multiplying with a rotation transform matrix.

## 10. Predator-Prey Trajectory Tracking

Instead of using a generic trajectory tracking that mimics the path of the target, we decided to explore a more intuitive behavior that of a generic predator-prey interaction. A cheetah chasing its prey, for example, moves to the shortest trajectory rather than mimicking the path of its target. This behavior results in a relatively smooth path for the cheetah, while the target may be moving somewhat chaotically. The smooth path, in turn, allows the cheetah to maintain its high speed and still meet the desired goal. To provide a more realistic behavior, we introduce constraints on angular rate and velocity commands, which are based on Ackermann steering platform kinematics.

We assume that the reference system can be controlled to follow a desired trajectory. By ensuring the actual system converges to a small neighborhood of the reference system, we also ensure that the system generally follows the desired trajectory as well. At the same time, a sufficiently large separation between reference system and vehicle allows a buffer for the vehicle to track the reference, without violating its nonholonomic constraints. In order to properly govern the following distance, it is necessary to define the error between the vehicle and reference system. We start by

assuming that the origins of the body fixed coordinate frame,  $B$ , and of the global inertial coordinate frame,  $U$ , coincide with the center of mass of the vehicle in the horizontal plane. An orthonormal transformation from  $B$  to  $U$  is then defined.

$$R(\theta(t)) \triangleq \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix} \quad (1)$$

Using this transformation along with unicycle system model the position error can be expressed in the bod coordinate frame as,

$$e(t) \triangleq R^T(\theta(t))(p_r(t) - p(t)) \quad (2)$$

where  $p(t)$  is the vehicle's position,  $p_r(t)$  is the reference system's position. By guaranteeing that the tracking error  $e(t)$  converges to the distance vector,  $d(t)$ , the longitudinal error will converge to  $d(t)$  while the lateral error goes to zero, where  $d(t) > 0$  is the commanded following distanced designed such that  $d(t) \rightarrow d^*(t)$  as  $t \rightarrow \infty$ . To provide this behavior, we introduce the following control law shown below.

$$\begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \Delta^{-1}(t)(K \tanh(e(t) - \delta(t)) + R^T(\theta(t))\dot{p}_r(t) - \dot{\delta}(t)) \quad (3)$$

where

$$\Delta(t) \triangleq \begin{bmatrix} 1 & 0 \\ 0 & d(t) \end{bmatrix}, K \triangleq \begin{bmatrix} k_v & 0 \\ 0 & k_w \end{bmatrix} \quad (4)$$

and  $k_v > 0$ ,  $k_w > 0$ , are scalar tuning constants. Ultimately, the algorithm is shown to satisfy the Lyapunov stability criteria, while satisfying all of these constraints. Finally, the proposed algorithm was implemented on VT-Leonardo and the results of which are shown below.

The low speed test results are given in Figure 10.1 below. As expected, the vehicle (red) tracks the reference trajectory (black) tightly. This is the typical behavior at low speeds, as the vehicle lags the reference system by a relatively small amount. While there are some slight deviations at the bottom and top of the graph, the overall performance is favorable in light of experimental results.

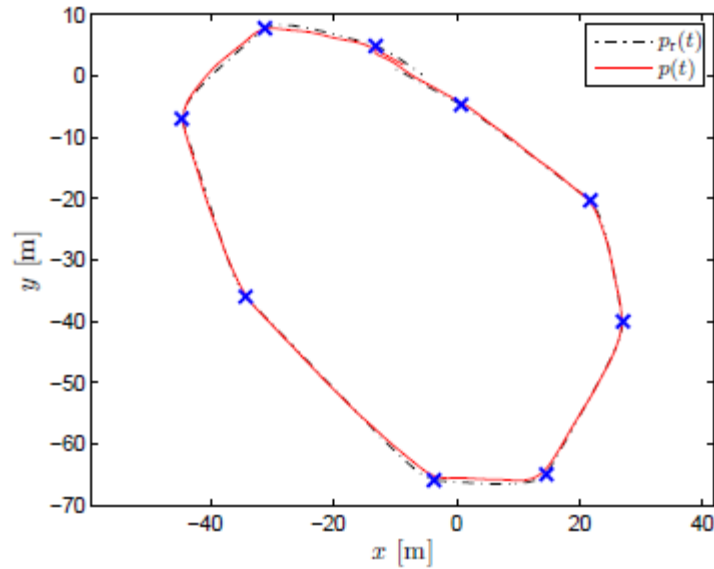


Figure 10.1: Tracking performance at 2 m/s. The blue x's are desired waypoints.

Next we examine the tracking performance at high speeds, the results of which are given in Figure 10.2. This test appears as an extreme case of the medium speed result. In fact, the vehicle's path remains well within the bound laid out by the reference system aside from the initial straightaway and the last turn.

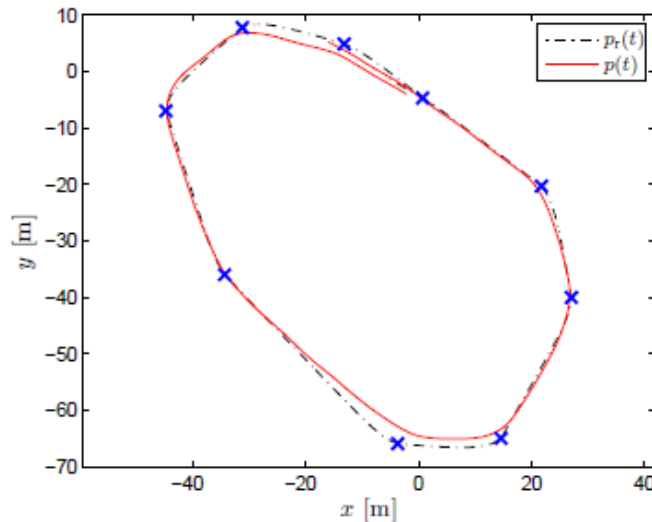


Figure 10.2: Tracking performance at 2 m/s. The blue x's are desired waypoints.

## 11. Obstacle Detection and Potential Field Obstacle Avoidance

VT-Leonardo's obstacle avoidance algorithm operates when the vehicle is within 2 meters of an obstacle. The LRF sensor provides angular position and distance information that enable the obstacle avoidance algorithm.

The vehicle uses a virtual force field method along with the controller presented above to avoid obstacles. As proposed by Borenstein and Koren in 1989, the method allows for fast, continuous and smooth motion of the controlled vehicle among unexpected obstacles. The method uses a 2D Cartesian grid. Each cell in the grid holds a certain value (CV) which represents the confidence of the algorithm in the existence of an obstacle at that location. The CV is updated with reading from a sensor indicating the presence of an object at that cell. At the same time, the potential grid concept is applied to the 2D grid. As VT-Leonardo moves a window of  $w \times w$  cells (active region) accompanies it, overlying a square region of  $C$ . The cells inside the active region are called active cells. Each active cell exerts a virtual repulsive force  $F_{ij}$  toward the robot. The magnitude of this force is proportional to value of CV of that particular cell and inversely proportional to  $d_n$ , where  $d_n$  is the distance between the cell and the center of the vehicle. All this virtual repulsive forces added up to yield the resultant repulsive force  $F_r$ . On the other hand, a virtual attraction  $F_t$  of constant magnitude is applied to VT-Leonardo pulling it towards the desired trajectory. In the implementation of our algorithm, the attraction and repulsive force are defined as below:

$$F_a(t) \triangleq \frac{F_{ac}}{d_{des}(t)} \begin{bmatrix} x_{des,i} - x_r(t) \\ y_{des,i} - y_r(t) \end{bmatrix} \quad (5)$$

$$F_r(t) \triangleq \sum_{i=1}^{N_{obs}} -F_{rc} \left( \frac{W}{d_{obs,i}(t)} \right)^n \begin{bmatrix} x_{obs,i} - x_r(t) \\ y_{obs,i} - y_r(t) \end{bmatrix} \quad (6)$$

Where  $F_{ac} > 0$ ,  $F_{rc} > 0$ ,  $W > 0$ , and  $n > 0$  are tuning parameters.  $d_{obs,i}(t)$  is the distance between the reference system and surrounding obstacle waypoint.  $d_{des}(t)$  is the distance between the reference system and destination.  $x_{des,i}$  and  $y_{des,i}$  is the x and y coordinate of the current destination waypoint.  $x_{obs,i}$  and  $y_{obs,i}$  are the x and y coordinates of a particular obstacle waypoint.

With the use of the force vectors presented in (5) and (6), the reference system is incorporated into the dynamic system described as

$$m\ddot{p}r(t) + c\dot{p}r(t) = Fa(t) + Fr(t) \quad (7)$$

where  $m > 0$ , and  $c > 0$  are tuning constants for inertia and damping of the system, respectively. To define these constants, we start by considering the maximum desired velocity in the presence of only attractive forces. When maximum velocity of the reference system is achieved, the reference acceleration reduces to  $\ddot{p}r(t) = 0$ , which yields,

$$c\dot{p}r(t) = Fa(t) \quad (8)$$

By taking the norm of both sides and considering the maximum reference velocity case we can determine the damping as,

$$c = \frac{Fac}{vrmax} \quad (9)$$

In regard to determining the inertia, we assume the reference system persists at maximum velocity throughout the trajectory. If the kinetic energy is maintained along the path, then the path maintains the same characteristic shape, regardless of velocity. As such, the inertial term is defined as,

$$m = \frac{2E}{v_{rmax}^2} \quad (10)$$

where E is the user defined kinetic energy term of the reference system.

Using a combination of the trajectory tracking controller and obstacle avoidance, we examine the performance of the given approach in the hopes of finding more intuitive behavior. Figures 11.1 and 11.2 examine the case of obstacle avoidance at low speed. In Figures 11.1, the black triangles and red circles denote the location of the reference system and vehicle, respectively, at certain instances in time. Looking at Figure 11.1, the vehicle tightly tracks the reference system despite a sharp, awkward corner around the obstacle. Since the speed is small overall, seen in Figure 11.2, this behavior is acceptable. Of particular interest, is the fact that the velocity drops to only 1 m/s as the system avoids the obstacle. This is a natural result of the potential field system, which we utilize to obtain an intuitive response from the platform. Consequently, by examining the time stamps of Figure 11.1, we see that the distance between the reference and vehicle reduces during the cornering, allowing the vehicle to better track the turn.

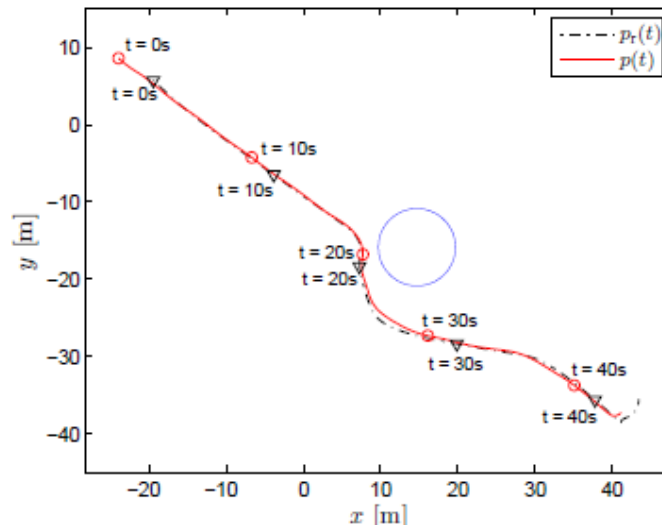


Figure 11.1: Obstacle avoidance at low speeds. Black triangles and red circles denote the location of the reference system and vehicle, respectively, at certain instances in time.

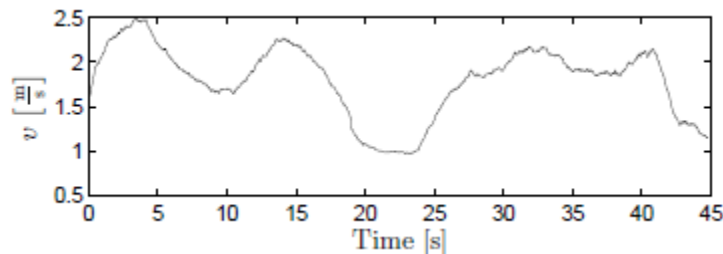


Figure 11.2: Velocity of obstacle avoidance at low speeds

## 12. Performance Analysis

**Vehicle Speed:** VT-Leonardo's motor has a nominal load of 7970 RPM at 24 V, the theoretical maximum speed is 10m/s and we have so far tested the vehicle at a maximum of 6m/s which it was able to handle efficiently.

**Ramp climbing ability:** Testing has shown that VT-Leonardo can climb and handle efficiently inclination to 35 degrees.

**Battery life:** VT-Leonardo can operate for a maximum of 8 hours before the battery needs to be recharged. As there is only one main battery that powers everything, we do not have to worry about any issues cause the battery life to be compromised.

**Distance at which obstacles are detected:** According to the data sheet for HOKOYU, objects can be detected at distance up to 30 meters away, and testing has not disproved that theory thus far.

**Accuracy of arrival at navigation waypoints:** The accuracy of the GPS on VT –Leonardo is +/- is 0.15 meters. The software team has programmed the robot to navigate to within 1 meter of a waypoint. With these two factors, the expected accuracy of arrival at navigation has been close to the desired value.

## 13. Conclusion

VT-Leonardo is a reliable integrated platform designed to meet the requirement of IGVC and our 3D mapping project. Through extensive circuit design, testing, and analysis, the vehicle includes new electronic and software implementation system designed to perform efficiently and robustly. More importantly, this innovation design incorporates a robust suspension for the chassis, addition of relays, switches, and inductors for a circuit and sensors integration and robust performance, predator-prey controller for trajectory following and potential obstacle avoidance technique.



#### 14. Vehicle Cost Summary (Labiba)

Item	Part Number	Cost(Each)	Quantity	Cost	Cost past year
1x2 6061 Box Tube 1/8" Wall 6' Long	6546K393	\$32.05	3	\$96.15	\$0
2.25x4 6061 Channel 0.19" Base 0.29" Leg 5' Long	1630T18	\$55.38	1	\$55.38	\$0
2x2 6061 Angle 1/8" Thick 8' Long	8982K25	\$32.27	1	\$32.27	\$0
24x24 6061 Sheet 1/8" Thick	89015K48	\$71.54	2	\$143.08	\$0
Lithium Iron Phosphate Battery	Customized	\$2,232.00	1	\$2,232.00	\$0
Protection Circuit Board	LFP25.6V100A	\$225.95	1	\$225.95	\$0
Maxon Motor	EC 45 136210	\$673.30	2	\$1,346.60	\$0
Gear Head	GP62 110502	\$603.70	2	\$1,207.40	\$0
Machine Shop – Welding		\$300	1	\$300.00	\$0
Scout Machine Camera SCA640	780884-01	\$818.09	1	\$818.09	\$0
Novatel GNS+INS with Antenna	SPAN-CPT	\$23,000.00	1	\$23,000.00	\$0
Hokuyo	UTM-30LX	\$5,590.00	1	\$5,590.00	\$0
Toradex compact computer	Robin Z530 V2.0	\$517.00	1	\$517.00	\$0
Roboteq Motor Controller with Encoder support	E6	\$330.00	2	\$660.00	\$0
US Digital E6 Encoders		\$95.00	2	\$190.00	\$0
Cappuccino PC	MP67-Di	\$615.00	1	\$615.00	\$615
Netgear Prosafe 5 port Gigabit Switch	GS105	\$35.00	1	\$35.00	\$35
1-Port USB-to-USB Isolator, 2 kV	SP386A	\$94.95	2	\$189.90	\$190
Intelligent Automotive DC-DC Car PC Power Supply	M4-ATX	\$89.50	1	\$89.50	\$90
12V Micro Power Supply	PW-200-V	\$49.95	1	\$49.95	\$50
Durakool Relay	DG85C	\$6.04	1	\$6.04	\$6.04
				<b>\$37,399.31</b>	<b>\$979</b>