

# UNIVERSITY OF WATERLOO

FACULTY OF ENGINEERING

---

## INDRIK : Design Report 2011 Intelligent Ground Vehicle Competition

---

*Authors:*

Arun DAS, Kent STOLTZ, Peiyi  
CHEN, Prasenjit MUKHERJEE,  
Siddhant AHUJA, Yan MA

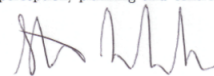
*Supervisors:*

Dr. Steven WASLANDER,  
Dr. William MELEK

---

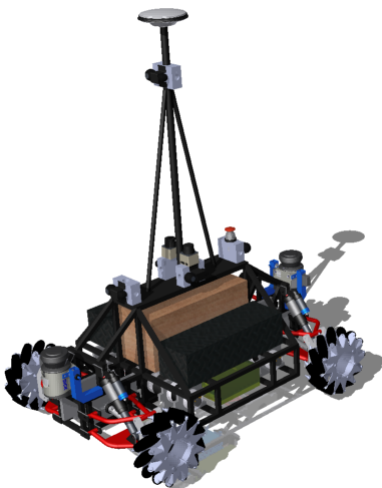
To Whom It May Concern:

I hereby certify that the design and engineering of the vehicle by the current student team from the University of Waterloo for entry into the 2011 IGVC has been significant and equivalent to what might be awarded credit in a senior design course. The team has made significant upgrades to the overall design including a new independent suspension chassis design fabricated from the ground up, electric drive propulsion, advanced sensing and control, and significant upgrades to perception, planning and control software.



Dr. Steven Waslander

May 9, 2011



Waterloo Autonomous Vehicles  
Laboratory



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Design Innovations . . . . .	2
1.2	Design Process . . . . .	2
1.3	Design Team and Organization . . . . .	3
<b>2</b>	<b>Hardware Design</b>	<b>3</b>
2.1	Mechanical Design . . . . .	3
2.2	Electrical and Computer Systems Design . . . . .	5
2.2.1	Safety . . . . .	6
2.3	Sensor Suite . . . . .	6
<b>3</b>	<b>Perception</b>	<b>7</b>
3.1	Mapping and Obstacle Detection . . . . .	7
3.2	Vision systems and Lane Detection . . . . .	8
<b>4</b>	<b>Planning and Obstacle Avoidance</b>	<b>9</b>
4.1	Modified PRM Algorithm for Unknown Environments . . . . .	9
4.2	Sampling Strategy and PRM Regeneration . . . . .	10
4.3	Heuristic Algorithm for Improving Solution Path . . . . .	10
4.4	Kinematic Motion Primitives . . . . .	10
4.5	Decoupled Yaw with Omni Wheels . . . . .	10
4.6	Path Planning Simulation Results . . . . .	11
<b>5</b>	<b>Estimation and Control</b>	<b>11</b>
5.1	Modelling . . . . .	12
5.2	Estimation . . . . .	12
5.3	Control . . . . .	13
<b>6</b>	<b>Summary</b>	<b>14</b>

# 1 Introduction

The 2011 Intelligent Ground Vehicle Competition provides the robotics community at the University of Waterloo with a unique opportunity to learn, practice and explore the design and implementation of advanced unmanned systems. As the proliferation of unmanned systems becomes a requirement in several different industries, competitions such as IGVC provide the testing grounds required for the team to prepare for participation in the unmanned systems industry. With this in mind the University of Waterloo team set out to design and build a versatile, dependable and novel solution to the autonomous systems problem posed by IGVC. The result, *Indrik* (named after a majestic beast in Russian folklore) is presented in this report.

## 1.1 Design Innovations

The team has implemented some distinctive innovations for the problem presented in the IGVC competition. These include:

- **Chassis:** A new, custom-designed chassis was fabricated with manoeuvrability, versatility and dependability as the design focus. This has led to a double-wishbone independent suspension design (shown in Figure 1). The vehicle can carry payloads of upto 300lbs for long distances, over challenging terrain be it dirt, mud, water, gravel, rocks or snow. More details of this design can be found in Section 2.1.
- **Omni-Directional Movement:** Mecanum wheels are employed in this year's design due to their ability to slip at certain angles which allows for lateral and diagonal movement of the robot. The slip created by the rollers on these wheels allows for a movement in any direction, thereby minimizing yaw rotations and enhancing the ability to elegantly traverse a terrain and move through a set of obstacles. We can also switch between a highly manoeuvrable omni-wheel design and a traditional skid-steer design on the go. More details of this design can be found in Section 2.1.
- **Tilting 3D LIDAR system:** In order to sense the 3D terrain and avoid obstacles placed in the robot's path, we have employed a tilting mechanism for the LIDAR which provides us with a 3D point cloud of the scene in front of the robot (shown in Figure 4). For more details on the sensor see Section 2.3, details on the mapping software can be found in Section 3.1
- **Planning:** The team has also developed and extended a novel planning algorithm based on probabilistic road maps. This allows the platform to plan robust 3D trajectories taking full advantage of the omni-directional movement. The planning software has been extended for use in platforms that employ omni-wheel and/or skid-steer, as described in Section 4.

## 1.2 Design Process

After a comprehensive assessment of the competition requirements the team decided to focus their efforts on six points of design and implementation;

- Mechanical Chassis Design
- 3D Mapping
- Vision software
- Path planning
- Estimation and Control
- Systems Integration

Wherever possible the team has attempted to leverage off of dependable existing solutions for parts of the problem. This includes work in sensor driver development, high level software management (operating systems, process management etc.) and low-level motor control. This has allowed the team to focus the limited human resources onto the six aspects of the design shown above. ROS (robot operating system) is a popular open-source meta operating system platform that was leveraged by the team for sensor drive development, high level software

management and basic 3D occupancy grid software. The low-level motor control was abstracted by a pair of Roboteq HDC2450 motor controllers. The remaining aspects of the platform have been comprehensively designed by the team over a period of four months in an iterative fashion with frequent design overview meetings to ensure the design stays cohesive and meets all the requirements.

### 1.3 Design Team and Organization

The team consisted of six core undergraduate and graduate team members with various other members of the University of Waterloo Robotics team aiding wherever possible. Overall, over 1000 person-hours has been spent on both hardware and software for this year’s entry. The breakdown of the design tasks is shown in Table 1

Table 1: Design Task Breakdown

Team Member	Design Task
Yan Ma	Vision Systems
Arun Das	Team Lead, 3D Mapping and System Integration
Peiyi Chen	Path Planning
Siddhant Ahuja	Mechanical Design, Vision Systems and System Integration
Kent Stoltz	Mechanical Design and Fabrication
Prasenjit Mukherjee	Estimation, Control and System Integration

The team would also like to acknowledge Kevin Ling for his contributions in making the systems JAUS compliant. Design advice and direction was provided by the team advisor Prof. Steven Waslander.

## 2 Hardware Design

In terms of mechanical design, the team decided to focus on reducing the effect of vibrations on the structural components and also the sensors themselves. An independent double-wishbone suspension with adjustable rising rate spring set-up and variable damping rates was selected as the preferred, off-road capable design. A detailed design of the robot’s suspension, drive-train and chassis are given in Section 2.1. A complete electrical system is discussed in Section 2.2, followed by a description of the sensor suite in 2.3. An overall list of components used on *Indrik* and their respective costs are shown in Table 2.

### 2.1 Mechanical Design

*Indrik* was fully designed using *Autodesk Inventor* CAD software before any parts were fabricated. Wherever feasible, Finite-Element Analysis (FEA) was conducted to ensure a minimum safety factor of 2. The chassis is made of 5/8” 4130 Chrome Moly square tubing which has a higher strength to weight ratio. *Indrik* measures 34” wide, 41” long, and 70” tall; and weighs approximately 300 pounds excluding the payload. A CAD visualization of the robot featuring an independent double un-parallel A-arm suspension is shown in Figure 1.

#### Power-train

Based on the power requirements, custom made brushed DC motors with rear shaft mounted quadrature encoders were purchased from Midwest Motion Products. To determine the final gearbox ratio and wheel diameter, an analysis of the motor curve was performed. It was assumed that there exists an equal weight distribution on each wheel (about 75 pounds), the gearbox efficiency is 75% and the standard backlash is 33 arc min. Therefore, in order to produce a continuous torque of 75 pounds-inch, a 14:1 gear ratio was required. To maintain a speed of 10 mph in competition with the selected gear ratio, 10” diameter wheels each driven by their own motors with 1:1 chain drives were selected. This configuration provides for quick acceleration/deceleration and enough safety

Table 2: List of Costs for Building *Indrik*. All zero-cost items were donated.

Item	Actual Cost	Team Cost
NovAtel OEMV-3 GPS	\$10,1000.00	\$0.00
2x SICK LMS-111 LIDAR	\$6,000.00	\$3,000.00
2x Hokuyo LIDAR	\$4,500.00	\$0.00
4x Logitech QuickCam Cameras and 1x PointGrey Camera	\$900.00	\$200.00
MicroStrain IMU	\$2,000.00	\$0.00
Computers	\$3,800.00	\$0.00
Motors and Motor Controllers	\$3,294.00	\$3,294.00
Chassis and Suspension Components	\$5,000.00	\$5,000.00
Mecanum Wheels	\$850.00	\$850.00
Misc. Mechanical Components	\$1,000.00	\$1,000.00
Batteries and Battery Chargers	\$1,650.00	\$1,650.00
Misc. Electrical Components	\$500.00	\$500.00
Ethernet Router and Switch	\$150.00	\$0.00
<b>TOTAL</b>	<b>\$39,644.00</b>	<b>\$15,494.00</b>

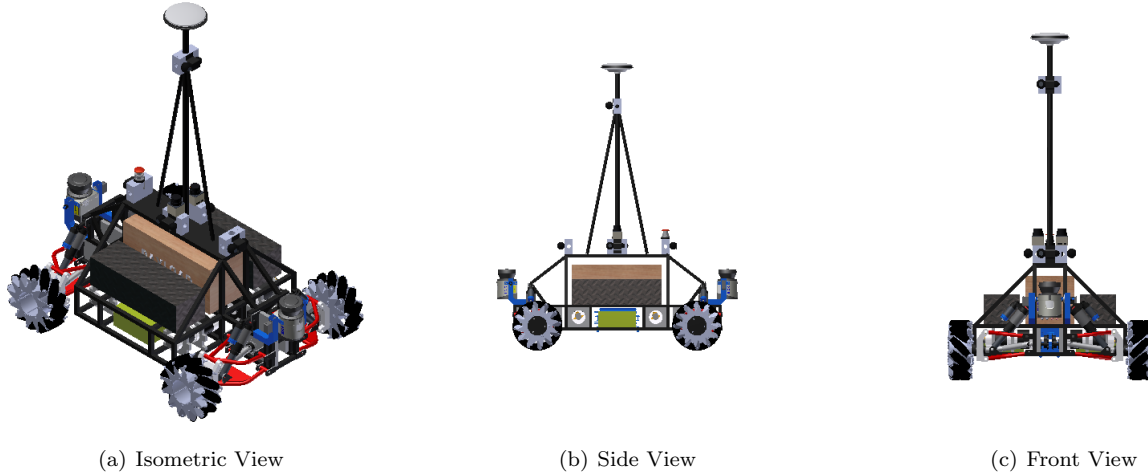


Figure 1: *Indrik*: 3D CAD Visualization

margin for torque loading. For robustness and easy maintenance, a chain-sprocket drive connected to telescoping U-Joints was selected.

### Suspension

One of the main objectives of a suspension system is to ensure that the wheels maintain a close contact with the ground with a maximum tire contact patch in any driving situation. After considering many options such as MacPherson struts, trailing links, swing axle designs etc., a double un-parallel A-arm suspension design was selected with a rake of 12 degrees from the horizontal. This provides better transmission of shock forces when the vehicle lands after jumps and when approaching steep inclines. The shock absorption is supplied through four independent Fox Air DHX 5.0 shocks with 3.0 inches of travel. Air shocks provide significant advantages over coil-over systems in terms of weight reduction, increased travel and adjustable spring and damping rates. The A-arms were fabricated with 5/8" 4130 chromoly tubing, which is a strong, lightweight, and workable material. This adds durability to the suspension, thus improving the reliability and safety of the vehicle. A finite element analysis was conducted in AutoCAD Inventor in order to ensure a safety factor well above 2 for both front and rear. The stress plot of the shock loading analysis can be seen in Figure 2.

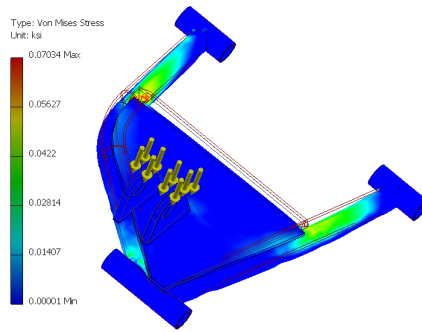


Figure 2: Finite Element Analysis for Bottom A-Arm under 20psi shock load.

In order to reduce the amount of body roll during cornering, the distance between the center of mass (COM) of the vehicle and the roll center was reduced, which enables the use of softer shock absorbers. The four bar linkage analysis of the system and location of instantaneous centers along with the COM and Roll Center is presented in Figure 3.

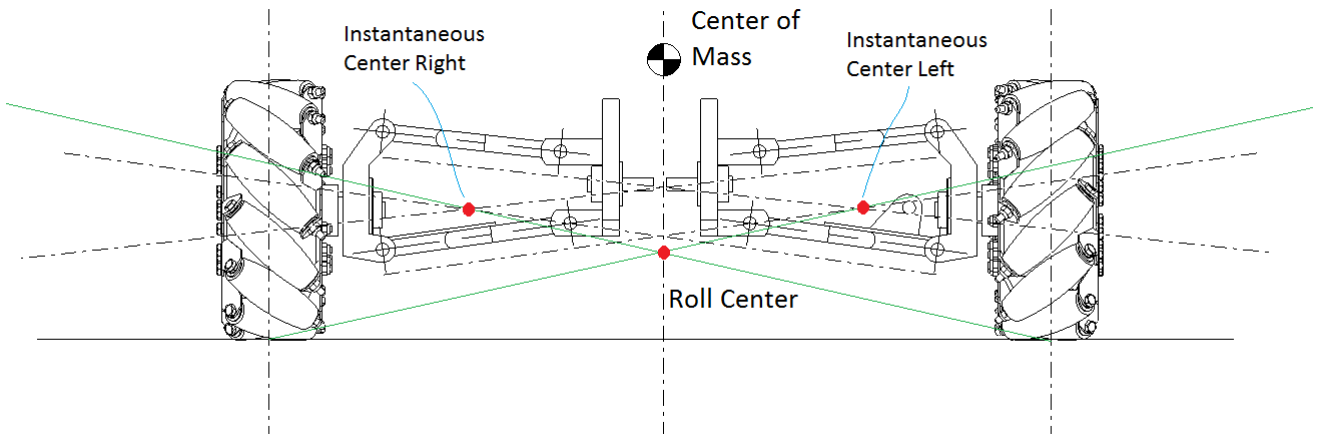


Figure 3: Locating the roll center.

## 2.2 Electrical and Computer Systems Design

In order to guarantee safe operation of the motors and related circuitry during high-current demands, Roboteq’s HDC2450 motor controllers were selected which can provide up to 150 amps on each channel, capable of driving two motors simultaneously. It has a built-in closed loop speed control which integrates well with our algorithms. All of the sensitive onboard electronics are shock-isolated with rubber padding and passive air-cooling is employed for heat dissipation. The platform is powered by two 24V, 30A-h custom built LiPo packs capable of discharging at a continuous rate of 45C. A DC-DC Converter is used to supply low-level controllers and some sensors with a regulated 12V and 5V supply.

Due to the near real-time performance requirements for the software algorithms and computationally intense calculations, *Indrik* requires one embedded computer dedicated to sensor data processing and analysis, one computer for vision algorithms, and one computer for supervisory control, navigation, and interfacing with other low-level controllers. Two Toshiba Tecra laptops boasting a dual core Intel i7 processor (2.66 GHz Processor), 4 GB of RAM and a 128GB solid state hard drive are utilized. The solid state hard drive was selected to eliminate hard drive failure due to vibration and shock, which is common among standard hard drives. The embedded computational platform is an ADLink embedded Intel i7 dual core industrial computer, with 2GB of RAM and

a 30 GB solid state hard drive. The system also includes a gigabit ethernet switch and 802.11n wireless access point in order to enable high-speed data transfer between the sensor nodes and computers. This feature is useful for outdoor testing and for satisfying the JAUS challenge requirements.

### 2.2.1 Safety

*Indrik* can be stopped on command by manually activating the hardware emergency stop (“E-Stop”) system which brings its motors to a complete halt. It is bright red in color and located on the center rear of the vehicle at a distance of about 2.5 feet from the ground. It can also be stopped by a wireless E-stop with an effective range of atleast 100 feet. In addition a safety light has been included on top which flashes when the robot is in autonomous mode and becomes solid when its not. All sensors, motor controllers and critical electronic equipments have been protected by circuit breakers and fuses ensuring safe operation.

## 2.3 Sensor Suite

For this competition the selected sensors facilitate the vehicle’s ability to perceive its environment, and localize itself within that environment. The sensor suite includes a vision system, a state measurement system and LIDAR scanning system. The vision system consists of a cluster of two Logitech Quickcam 9000 Pro cameras and one PointGrey Firefly camera. These are primarily used for white line and flag detection, with the Firefly pointing forward for high speed imaging of the most interesting elements of the environment, and the Quickcams focused on line detection and lateral obstacle detection. The state measurement system consists of motor shaft encoders, which provide shaft rotation rate, a Microstrain 3DM-GX3 AHRs sensor, which provides accurate attitude estimates of the vehicle, and the NovAtel OEMV-3G GPS sensor, which provides information on the vehicle’s spatial location and ground speed.

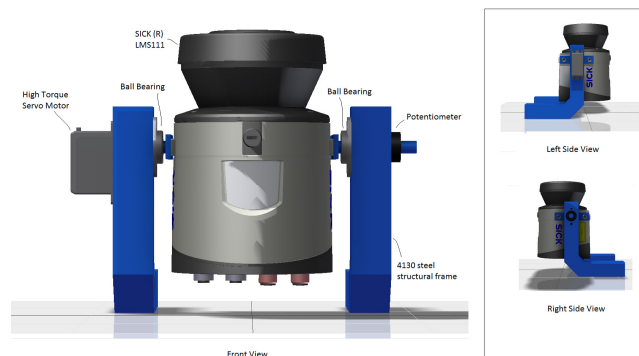


Figure 4: SICK LMS111 laser scanner mounted on tilting platform.

The LIDAR scanning system is comprised of two SICK LMS111 LIDARs capable of generating 3D scans of the environment with a range of 18 meters at 50 Hz placed front and back and two Hokuyo URG-04LX LIDAR with 6 meter range at 10 Hz placed on the sides. In order to generate 3D environmental data, the LIDARs are mounted on respective nodding gimbals, as seen in Figure 4. The tilt angle of the laser is measured by a high resolution potentiometer mounted on the frame. The analog voltage is interpreted and relayed to the high level computer via an ATmega328 microcontroller with a serial/USB interface. This microcontroller also controls a servo motor responsible for rotating the laser scanner in a nodding fashion.

### 3 Perception

#### 3.1 Mapping and Obstacle Detection

The vehicle mapping algorithm relies on an open source implementation of 3D octree mapping called OctoMap [14]. OctoMap provides a method for creating 3D occupancy grid maps from point-cloud data. The main advantages of the OctoMap library is that it is real-time updatable, flexible and compact. The updates are performed in a *probabilistic* fashion, thus helping to reduce the effect of sensor noise on the generated map. The implementation is also flexible, as scans can be added to the map at any time, and no prior knowledge about the boundaries of the environment are required. Finally, since the data is stored in an octree data structure with real-time pruning, the map is stored efficiently in memory.

The map is updated using a probabilistic update rule. This implies that each occupancy cell in the environment is assigned a probability based on the incoming point cloud. The probability that a cell  $n$  is occupied given an incoming point cloud  $z_{1:t}$  is given by

$$P(n|z_{1:t}) = \left[ 1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1 - P(n)} \right]^{-1} \quad (1)$$

The term  $P(n|z_t)$  is known as the *inverse measurement model* and determines the probability that a cell is occupied given a certain sensor measurement. Using the log odds notation,  $L(n) = \left( \frac{p}{1-p} \right)$ , the map update rule Equation. (1) (assuming an initial belief of  $P(n) = 0.5$ ) can be expressed as

$$L(n|z_{1:t}) = L(n|z_{1:t-1}) + L(n|z_t) \quad (2)$$

Equation 2 illustrates that occupancy probability in a cell at time  $t$  is the previous belief that the cell was occupied at time  $t - 1$ , plus the information given by the inverse measurement model at time  $t$ . Through this update, if the probability of a cell is less than  $l_{min}$ , then it is classified as free (or unoccupied). If the probability of a cell becomes greater than  $l_{max}$ , then it is classified as occupied. For probabilities between  $l_{min}$  and  $l_{max}$ , the cell has an unknown state. This probabilistic update demonstrates how the mapping technique reduces the effect of sensor noise, as multiple consistent sensor readings to a cell are required for the occupancy probability of that cell to exceed the threshold of  $l_{min}$  or  $l_{max}$ .

The OctoMap is built using 3D pointclouds generated by the vehicle’s LIDARs and point-clouds corresponding to white lines given by the vision system. An example of a point cloud and its resulting OctoMap is given in Figure 5. In order to determine drivable terrain in 2D, a local OctoMap is created from the occupied cells within a bounding box of the current vehicle location. The cell probabilities are then summed and normalized along the columns of the local OctoMap. Columns with the highest probabilities correspond to non-drivable terrain (obstacles), and are thresholded between  $p_{min}$  and  $p_{max}$ , where a probability greater than  $p_{max}$  corresponds to an obstacle occupied in the column. Occupied columns are then simply projected onto a plane, yielding a local 2D map of drivable terrain. In this way, the 3D map is flattened to 2D using a “bird’s eye view” approach.

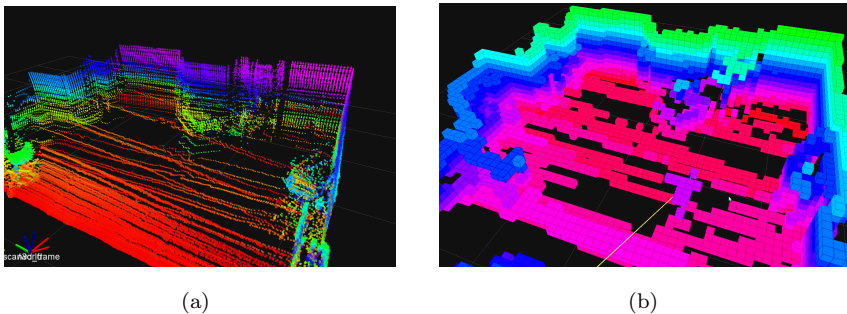


Figure 5: Mapping of an indoor environment. (a) Point cloud generated by the tilting LIDAR. (b) Resulting OctoMap with 10cm resolution



## 3.2 Vision systems and Lane Detection

The lane detection algorithm is outlined by the flow chart in Figure. 6. Each image frame is processed to locate pixel candidates for white lane marking. A multiband automatic thresholding method [12] is used to filter saturation and value channels of an HSV image to derive two binary images. It iteratively computes thresholds to maximize the separation of means for two clusters, providing robustness to changes in ambient illumination from frame to frame. These two images are subsequently combined to derive a binary image that indicates white pixel locations.

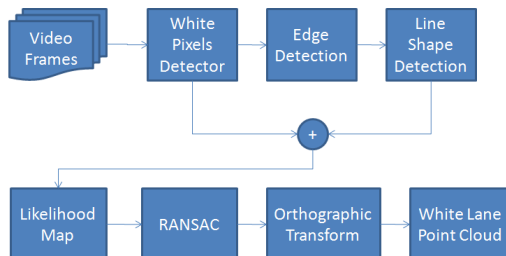


Figure 6: Lane detection algorithm flow chart

After white pixel thresholding, additional filtering is performed to remove pixels associated with objects. Each column of the HSV image is scanned from the top to find the number of nongreen colored pixels. If the number is over some threshold (tunable), white pixels in this columns would be ignored. This minimizes the error in lane marking candidate identifications. As shown in Figure. 7, white stripes in the orange pylons in the original image Figure 7(a) are successfully rejected in Figure 7(b) as a result of this heuristic.

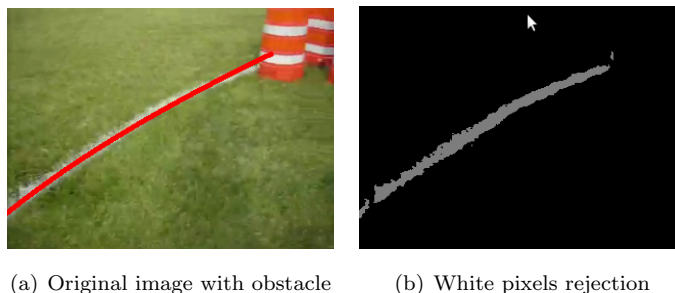


Figure 7: Example of rejection of white stripes on pylon obstacles.

A rectangular pulse kernel in various orientations is subsequently convolved with the gray scale image to detect line shape responses. This operation filters out white blobs that are not elongated in shape, to differentiate white lanes from environments such as sandpits and sky. The size of the kernel determines the width of line to be detected, and is a tuning parameter depending on the line width of the lane marking expected to be seen by the camera. Figure. 8(b) demonstrates the image filtered by the line detection kernel.

The line image and white binary image are summed to produce a heat map, assigning each pixel a likelihood as a part of the lane marking. Figure 8(c) contains a sample heat map from a test trial. A Random Sample Consensus (RANSAC) algorithm [7] then attempts to fit parabolic models on continuous segments of likely pixels in the heat map, as first proposed by formulated by the 2008 Princeton IGVC team [1]. The ability of RANSAC to reject outlying white pixels offers robustness over a straightforward regression fit. Figure 8(d) shows the complete lane detection results.

After finding white lanes, the inertial frame map is updated with the lane locations to aid in path planning. To achieve this, it is assumed that the ground plane is flat and every pixel seen in an image is on the ground, this is a valid assumption since the camera will be oriented downwards in order to reject pixels that are not part of the ground plane. An Orthographic projection [4] is used to translate the lane markings from the perspective image plane onto the ground plane. Figure 3.2 demonstrates the effect of orthographic projection after calibration.

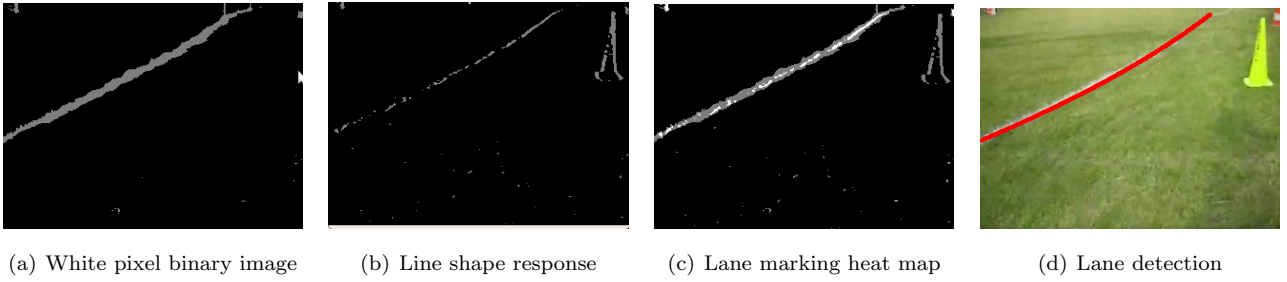


Figure 8: Lane Detection Algorithm steps.

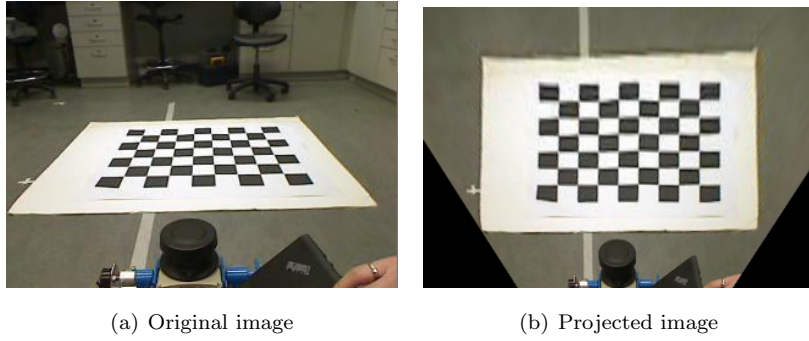


Figure 9: Rectification of an image

## 4 Planning and Obstacle Avoidance

The path planning algorithm assumes that the planning and estimation problems are decoupled (i.e. estimations of the vehicle state and environment map are available and accurate). The algorithm first uses a modified probabilistic roadmap (PRM) planner to find a collision-free path, and then divides the path into kinematic motion primitives by exploiting the vehicle's ability to turn on the spot.

### 4.1 Modified PRM Algorithm for Unknown Environments

PRM planners [10] can efficiently capture the configuration space,  $\mathcal{C}$ , of large and complex environments. In this problem,  $\mathcal{C} \in \mathbb{R}^3$  corresponds to the position,  $x$  and  $y$ , and yaw,  $\psi$ , of the vehicle. In unknown environments, the PRM planner assumes unexplored areas are obstacle free and updates the solution path as new obstacles are detected. This algorithm, inspired by the D\* algorithm [13] and Lazy Collision Checking [2], is summarized in Algorithm 1, where  $\mathcal{C}_{free}$  is the obstacle-free subspace of  $\mathcal{C}$ ,  $(x_d, y_d)$  denote the desired final position, and the function  $\text{CheckCollisionPath}(p)$  returns true if path  $p \not\subseteq \mathcal{C}_{free}$ . Note that each PRM node has a variable yaw, selected at runtime, to align the vehicle to each adjoining edge during collision checking. This ensures that the vehicle is travelling forwards along each segment of the solution path.

---

#### Algorithm 1 Modified PRM for Unknown Environments

---

```

Select  $p = \text{PRM solution path assuming unexplored areas} \in \mathcal{C}_{free}$ 
while  $x \neq x_d$  and  $y \neq y_d$  do
  if  $\text{CheckCollisionPath}(p)$  then
    Remove all edges  $\not\subseteq \mathcal{C}_{free}$  from the PRM graph
    while  $(x, y)$  and  $(x_d, y_d)$  are not connected do
      Add additional samples to the PRM graph
    end while
     $p = \text{new solution path returned by A* search on the updated PRM graph}$ 
  end if
end while

```

---

## 4.2 Sampling Strategy and PRM Regeneration

A hybrid sampling strategy that combines uniform [10], bridge [9], and exploration samples allows the PRM planner to effectively capture the connectivity of  $\mathcal{C}_{free}$ . Uniform and bridge sampling are well established techniques in PRM literature for sample generation in expansive and narrow regions of  $\mathcal{C}_{free}$  respectively. Exploration sampling, a technique inspired by Gaussian sampling [3], improves the connectivity between  $\mathcal{C}_{explored}$  and  $\mathcal{C}_{unexplored}$  by generating sample pairs along the boundary of the explored map. The exploration sampling algorithm is given in Algorithm 2, where  $s \sim \mathcal{P}$  denotes a node  $s$  sampled from a probability distribution  $\mathcal{P}$ ,  $\mathcal{U}[S]$  is a uniform distribution spanning space  $S$ ,  $\mathcal{N}[S]$  is zero mean Gaussian distribution spanning  $S$  with some user selected variance,  $E_{s_1, s_2}$  is an edge connecting nodes  $s_1$  and  $s_2$ , and  $\text{CheckCollision}(E)$  returns true if edge  $E \not\subseteq \mathcal{C}_{free}$ .

---

### Algorithm 2 Exploration Sampling

---

```

while valid exploration sample pair not found do
  Select  $s_1 \sim \mathcal{U}[\mathcal{C}]$ 
  Select  $s_2 \sim \mathcal{U}[\{s \in \mathcal{C} : \|s - s_1\| = d, d \sim \mathcal{N}[\mathbb{R}_+]\}]$ 
  if ( $s_1 \in \mathcal{C}_{explored}$  and  $s_2 \in \mathcal{C}_{unexplored}$ ) or ( $s_1 \in \mathcal{C}_{unexplored}$  and  $s_2 \in \mathcal{C}_{explored}$ ) then
    if not  $\text{CheckCollision}(E_{s_1, s_2})$  then
      Accept  $s_1$  and  $s_2$  as a valid exploration sample pair
    end if
  end if
end while

```

---

To preserve computational efficiency, the PRM graph is regenerated when it grows beyond an environment dependent threshold. This is done by deleting all nodes and edges except for those that have already been traversed. It is important to retain the traversed nodes and edges to ensure that the vehicle can return to a previous area if needed which helps in situations such as dead ends or center islands.

## 4.3 Heuristic Algorithm for Improving Solution Path

PRM planners do not maximize the clearance around the solution path nor do they remove redundant segments. The heuristic algorithm described in [5] is used to mitigate these shortcomings. The algorithm attempts to replace each node in the path with a sample (generated within a neighborhood of that node) that has greater obstacle clearance such that the path remains collision-free. Redundant segments are eliminated, where possible, by connecting node  $i$  to  $i + 2$  in the path. This process is iterated greedily to maximize the average obstacle clearance around each node in the path.

## 4.4 Kinematic Motion Primitives

Traversing a piecewise linear path is inefficient since it requires the vehicle to momentarily stop and turn at the end of each segment. Thus, the path will be divided into kinematic motion primitives using the method described in [5]. Note that the worst case for this method will return the same piecewise linear path requiring the vehicle to stop and turn. The method divides the path into straight and corner motions based on kinematic constraints of the vehicle (which will be experimentally determined). The vehicle will not exceed its kinodynamic capabilities while traversing the path as long as it doesn't exceed the velocity bound on each motion primitive.

## 4.5 Decoupled Yaw with Omni Wheels

With omni wheels, the vehicle becomes holonomic and gains the ability to yaw independently while traversing a path. This allows the vehicle to direct its sensors at objects of interest without having to stop or re-route. Given a max yaw rate and tangential velocity, it is possible to calculate the positions at which the vehicle should start and finish its yaw transitions to reach the target yaws. The modified path sections should be collision

checked with the new yaw values to ensure it is still collision-free. If the commanded yaws cause a collision, then depending on the command priority, the vehicle can either ignore the command or find a new solution path.

## 4.6 Path Planning Simulation Results

The simulation uses the ROS RVIZ stack for visualization and OctoMap [14] for mapping. The starting point of the path is kept constant in these simulations to better illustrate the effects of the algorithm. In practice, the vehicle’s latest configuration will be used as the starting point each time the algorithm needs to modify the path.

The simulation results for a skid-steer vehicle is shown in Figure 10. The solution path is indicated in yellow, the PRM graph in purple, and the obstacles in green. There are initially no obstacles, so the path starts as a straight line (Figure 10(a)). As obstacles are detected, the path is modified to remain collision-free (Figure 10(b)). If needed, additional samples are added to the PRM graph until a valid solution path exists (Figure 10(c)). Figure 10(d) shows the final path through the fully mapped environment.

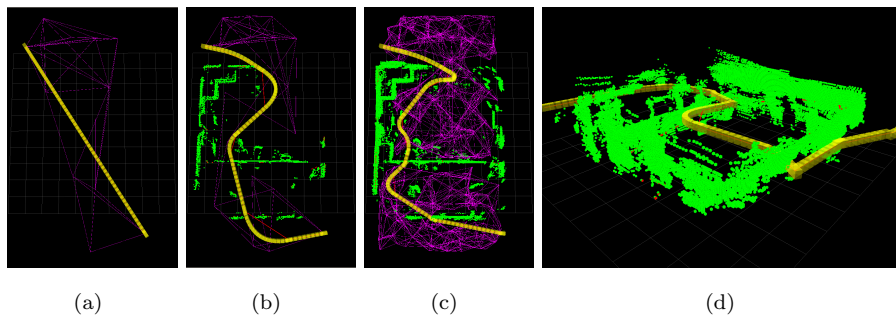


Figure 10: Simulation of the modified PRM algorithm in an unknown environment.

The path planning simulation results for a vehicle with omni wheels is shown in Figure 11. The purple arrows indicate the yaw of the vehicle along the path, and the red sphere is a target object of interest. The aim is to traverse the path while facing the target object as often as possible. Figure 11(a) shows the initial solution path with the vehicle facing the target object at all times. In the presence of obstacles, the vehicle will no longer face the target object unless it has a clear line of sight to it (Figure 11(b)). Even with a clear line of sight, the vehicle will not face the target object unless it can transition to and from the required yaw (while staying on the the path) without collisions (Figure 11(c)).

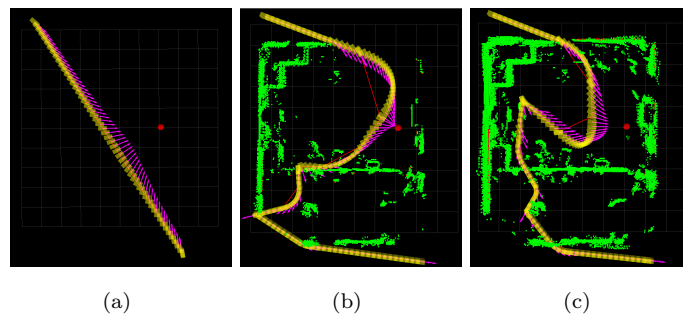


Figure 11: Simulation of omni wheeled skid-steer vehicle tracking a target object.

## 5 Estimation and Control

The planning and mapping algorithms both assume high quality information about the platform’s pose and precise control over the platform’s position are available. Robust and computationally efficient estimation and control systems are designed for both the skid-steer and omni-wheel versions of the vehicle to this end.

## 5.1 Modelling

A comprehensive estimation and control system needs a complete system model that relates the system inputs to the required states. Currently, only kinematic models are considered for the design to maintain computational efficiency.

**Skidsteer Model:** In most cases, the two wheeled robot model is used for skidsteer vehicles. The two wheeled robot model assumes no slippage at the wheel contacts, however, all skid steer vehicles operate on the condition of constant slippage. As such, a model that incorporates this constant slippage [11] is employed and presented here. The skidsteer vehicle has inputs,  $v_r$  and  $v_l$  which are the left and right wheel velocities, regulated by the HDC2450 motor controllers (based on encoder feedback). The wheel velocities can be translated to a velocity in the body x-direction,  $v_x$ , and an angular velocity about the body z-axis,  $\omega_z$  as,

$$u = \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{r(v_r+v_l)}{r} \\ \frac{r(v_r-v_l)}{2c} \end{bmatrix} \quad (3)$$

where  $r$  is the radius of the wheel, and  $c$  is the wheel base of the platform. Based on the system inputs, the position and heading of the system in the inertial frames can be derived as [11],

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & x_{ICR} \sin \psi \\ \sin \psi & -x_{ICR} \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} \quad (4)$$

where  $X, Y$  and  $\psi$  are the position and heading of the vehicle in the inertial frame and  $x_{ICR}$  is the location of the instantaneous center of rotation resolved to the body x-axis. Setting,  $x_{ICR} > 0$  assumes constant slippage for the model.

**Omnivheel Model:** The omnivheel system allows for decoupled motion in the body  $x, y$  directions and an angular velocity  $\omega_z$  about the body  $z$  axis. The omnivheel system can be considered to have four inputs  $u = [\omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4]^T$ . These can be related to motion in the body axes with the equation [6],

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\ -\frac{1}{4K} & -\frac{1}{4K} & -\frac{1}{4K} & -\frac{1}{4K} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} r \quad (5)$$

where  $K$  is the sum of the length  $l$  and wheelbase  $c$  of the platform,  $r$  is the radius of the wheels as before. While the equation above relates the motion inputs to its motion states for better estimation for control the inverse relationship is required. This can be realised using the pseudo-inverse approximation of the matrix in the equation above as derived in [6].

## 5.2 Estimation

In an effort to maintain robust and accurate pose estimates, a probabilistic estimator is implemented onboard the platform. This estimator combines known models of the plant, the system inputs and the available sensor information (IMU and GPS) to maintain a reliable estimate of the current pose. The state of the system as required by the planning and mapping software is chosen to be the 6-DOF pose of the robot,  $[X \ Y \ Z \ \phi \ \theta \ \psi]$ , 3D position and roll, pitch, yaw. Based on the models derived in previous sections two process models can be used. The skid-steer version of the vehicle has the process model as in Equation (6). Similarly, for omni wheel systems the motion model can be defined as in Equation (7).

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & x_{ICR} \sin \psi \\ \sin \psi & -x_{ICR} \cos \psi \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \quad (7)$$

where the inputs  $v_x, v_y$  and  $\omega_z$  are derived based on Equation (5). Note here that the 2D kinematic process models do not provide any information about the altitude, roll and pitch of the vehicle, as such these are assumed to have a random amount of error in the process model.

These two motion models are propagated whenever new control information is available. When new sensor information becomes available the following measurements are incorporated into the system,

$$y = \begin{bmatrix} X & Y & Z & v_x \cos \psi - v_y \sin \psi & v_x \sin \psi + v_y \cos \psi & \phi & \theta & \psi \end{bmatrix}^T \quad (8)$$

The measurements above include position (including altitude) and velocity measurements from the GPS and attitude measurements from the IMU. The above nonlinear motion and measurement models are processed through an Extended Kalman Filter framework to maintain accurate and computationally efficient estimates of the pose. Preliminary results onboard a skid steer vehicle are shown in Figure 12(c). This estimation system can now provide reliable estimates of the pose even with intermittent sensor measurements.

### 5.3 Control

As described in Section 4, the path of the system is discretized into a series of points that the system must now track. As before, two different control methods are derived for the omni-wheel and skid-steer versions of the platform.

**Skid-steer Control:** The skid-steer platform has a set of coupled kinematic relationships between position and heading and therefore a holistic control law that is able to operate with this coupling is required. The discrete set of points provide a path for the platform to follow. A cross-track error  $e_{ct}$  is calculated as the perpendicular distance between the platform and the nearest point on the required path. Similarly a yaw error  $e_{yaw}$  is defined to be the error in heading between the platform and the heading of the required path. Based on this parametrization of errors, a nonlinear tracking control law is suggested and proven to be asymptotically stable by Hoffmann [8] for kinematic vehicle plant models with coupling between the heading position (such as in Ackermann steering). This control law can be summarized as,

$$\omega_z = e_{yaw} + \tan^{-1} \left( \frac{ke_{ct}}{v_x} \right) \quad (9)$$

where  $\omega_z$  is the steering control input,  $v_x$  is the required forward velocity and  $k$  is a tunable controller gain. A simulation of this controller for a section of the path is shown in Figure 12(a).

**Omnivheel Control:** The omni-wheel system has a decoupled set of dynamics where each individual state can be regulated separately. The controller is designed to regulate a positive velocity tangent to the path

prescribed by the path planner, a crosstrack error perpendicular to the path prescribed by the path planner, and a desired heading. These requirements can be summarized into a control law as follows,

$$u_{tan} = c \quad (10)$$

$$u_{per} = -k_{per}^p e_{ct} - k_{per}^d \dot{e}_{ct} \quad (11)$$

$$u_{\omega} = -k_{\omega}^p e_{yaw} - k_{\omega}^d \dot{e}_{yaw} \quad (12)$$

where,  $c$  is the required positive velocity,  $e_{ct}$  and  $e_{yaw}$  is the crosstrack and yaw error and  $k_{per}^p, k_{per}^d, k_{\omega}^p$  and  $k_{\omega}^d$  are tunable PD gains. The three control requirements  $u_{tan}, u_{per}$  and  $u_{\omega}$  can be translated to required body velocities using the inverse rotation matrix between the body frame and the path frame.

$$\begin{bmatrix} v_x^r \\ v_y^r \\ \omega_z^r \end{bmatrix} = \begin{bmatrix} \cos e_{yaw} & \sin e_{yaw} & 0 \\ -\sin e_{yaw} & \cos e_{yaw} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{tan} \\ u_{per} \\ u_{\omega} \end{bmatrix} \quad (13)$$

The above body velocities can be translated to required wheel velocities using the pseudo-inverse of the kinematic relationship in Equation (5). Again, a simulation of this control law on the described model is shown in Figure 12(b)

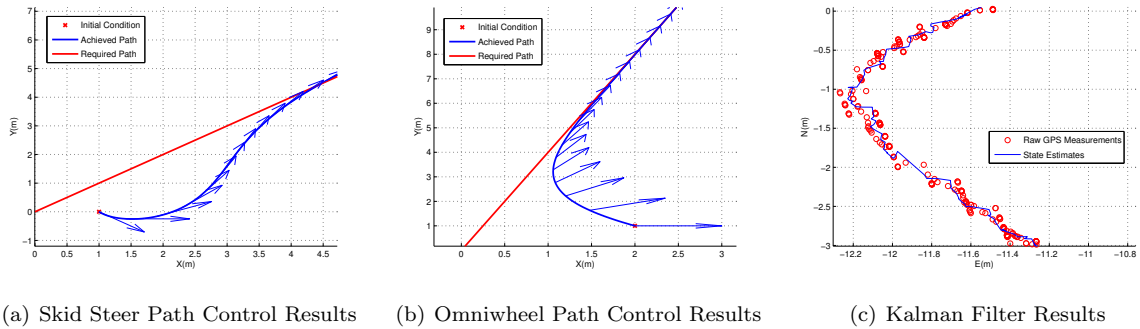


Figure 12: Path Control Results

## 6 Summary

*Indrik* unmanned ground vehicle is a rugged workhorse designed for use in a variety of environments. Featuring an independent suspension, all wheel electric propulsion, drive-by-wire capabilities, 6 hours of LiPo battery power, the vehicle can carry payloads of upto 300lbs for long distances, over challenging terrain be it dirt, mud, water, gravel, rocks or snow. It can climb ramps with inclines of upto 35 degrees. Taking advantage of the omni-wheel directional movement, the novel controller provides an extra degree of freedom for the robots movement. Fusing sensor data from multiple LIDARs and cameras placed around the robot, IMU, and wheel encoders, the robot is capable of detecting obstacles upto a range of 10m and reacting within 50ms of obstacle detection. It simultaneously localizes itself and maps its environment using advanced estimation and control algorithms. The novel PRM path planner is able to ensure optimized trajectory at a maximum speed of 10mph through randomly placed obstacles, ensuring an accuracy of +/- 60cm of arrival at navigation waypoints. The innovative design solutions presented in this paper ensure a competitive entry for the 2011 Intelligent Ground Vehicle Competition for the University of Waterloo team.

## References

- [1] C.A. Baldassano, G.H. Franken, J.R. Mayer, A.M. Saxe, and D.D. Yu. Kratos: Princeton University's Entry in the 2008 Intelligent Ground Vehicle Competition. In *Proceedings of IS&T/SPIE Electronic Imaging Conference*, volume 7252, 2009.
- [2] R. Bohlin and L.E. Kavraki. Path planning using lazy PRM. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 521–528. IEEE, 2000.
- [3] V. Boor, M.H. Overmars, and A.F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1018–1023. IEEE, 2002.
- [4] I. Carlbom and J. Paciorek. Planar geometric projections and viewing transformations. *ACM Computing Surveys (CSUR)*, 10(4):465–502, 1978.
- [5] Peiyi Chen and Steven L. Waslander. Kinodynamic Motion Planning for Holonomic UAVs in Complex 3D Environments. In *Guidance, Navigation and Control, 2010. Proceedings. 2010 AIAA International Conference on*, Toronto, ON, Canada, 2010. AIAA.
- [6] Ether. Kinematic analysis of a four wheeled mechnum vehicle. Technical report.
- [7] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [8] G.M. Hoffmann, C.J. Tomlin, M. Montemerlo, and S. Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *American Control Conference, 2007. ACC'07*, pages 2296–2301. IEEE, 2007.
- [9] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 4420–4426. IEEE, 2003.
- [10] L.E. Kavraki, P. Svestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [11] K. Kozłowski and D. Pazderski. Modeling and control of a 4-wheel skid-steering mobile robot. *International journal of applied mathematics and computer science*, 14(4):477–496, 2004.
- [12] TW Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE transactions on Systems, Man and Cybernetics*, 8(8):630–632, 1978.
- [13] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 3310–3317. IEEE, 1997.
- [14] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010.