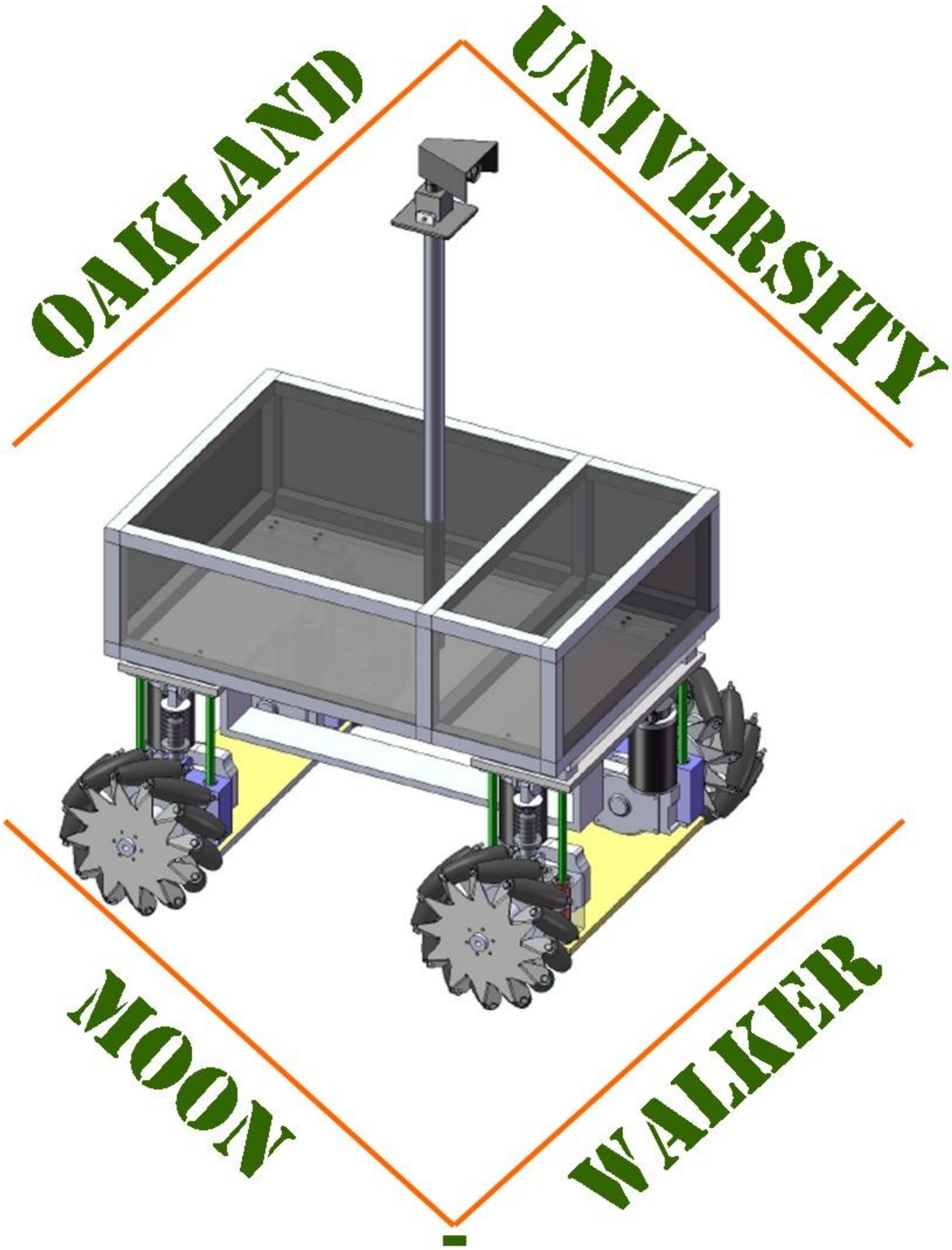


2009 Intelligent Ground Vehicle Competition
June 5 - 8, 2009

I certify that the engineering design present in this vehicle is significant and equivalent to work that would satisfy the requirements of a senior design or graduate project course. – **Dr. Ka C. Cheok**

1 Introduction

Oakland University introduces Moonwalker for the 2009 Intelligent Ground Vehicle Competition, an omnidirectional platform using Mecanum wheels. The name "Moonwalker" is in reference to the space exploring rovers envisioned during the design phase, as well as the interesting appearance of the vehicle as it moves. The motion of the vehicle does not always intuitively reflect the wheel rotations, and seems to "Moonwalk."

Moonwalker's platform has been completely designed from scratch, except for the Mecanum wheels. This document discusses the design of Moonwalker's systems, and illustrates the innovative hardware and software solutions developed to make a robust and intelligent vehicle for the 2009 competition.

2 Project Management

This section discusses the structure and organization of the team, as well as the design process followed to develop Moonwalker's systems.

2.1 Team Membership

A list of the team members and their person-hours of commitment is shown below. All of the students have graciously volunteered their time to make Moonwalker as complete and successful as possible.

Moonwalker's Design Team

Name	Academic Status	Department	Expended Hours
Pavan Vempaty	Graduate Student	Systems Engineering	400
Micho Radovnikovich	Graduate Student	Systems Engineering	400
Irfan Baftiu	Graduate Student	Electrical Engineering	400
Naveen Chilukoti	Graduate Student	Electrical Engineering	200
Feyzullah Koca	Graduate Student	Electrical Engineering	50
Alex Pawlowski	Undergraduate	Mechanical Engineering	400
Steve Grzebyk	Undergraduate	Electrical Engineering	100
George Mathew	Undergraduate	Computer Science	100
Steve Cowden	Undergraduate	Electrical Engineering	50

2.2 Team Organization

The design of Moonwalker is broken up into several subsystems, and the team members are assigned to work on one or more of the subsystems by the team leaders. One team member is made responsible for a particular subsystem, and schedules testing time and enlists help from the other members to accomplish specific goals and meet the deadlines set by the team leadership.

The team leaders are responsible for coordinating the efforts of the team and making strategic decisions based on the recommendations of the other team members. An organization chart of the team illustrating the responsibilities of the team members is shown in Figure 1.

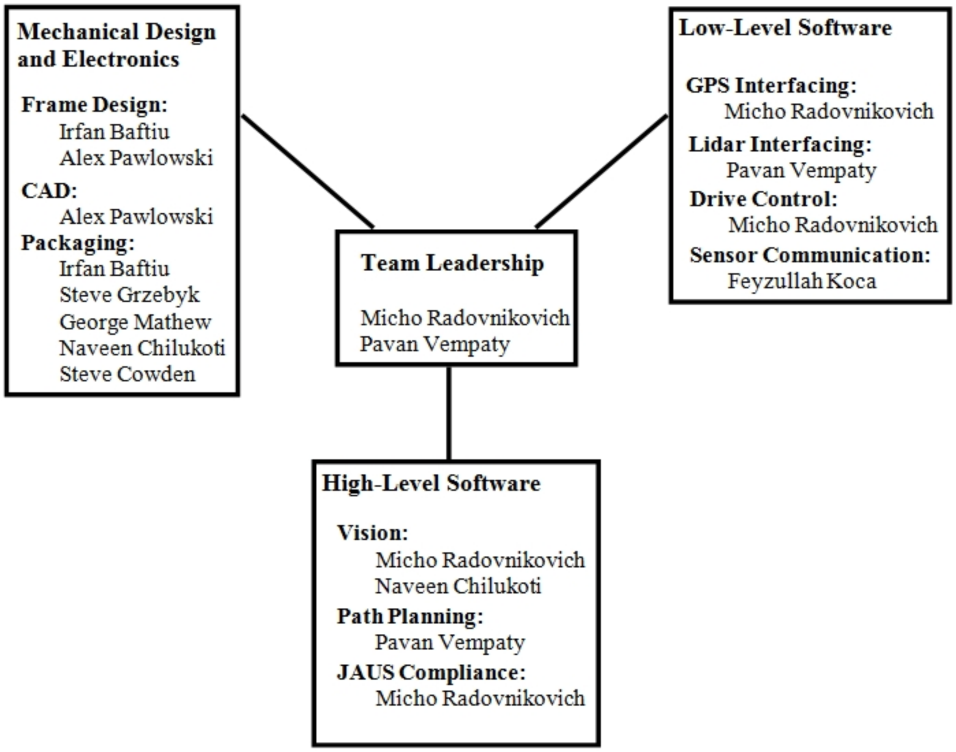


Figure 1: Organization of the Moonwalker design team.

2.3 Design Process

Moonwalker's design process followed the diagram shown in Figure 2. The first step of the design process was to brainstorm features that would help achieve good performance in the competition. Using knowledge and experience gained from last year's competition, these potential features were narrowed down and prioritized into an initial design plan. The Mecanum ODV design was chosen because the introduction of an extra degree of freedom greatly improves the vehicle's mobility. The rest of the features were all designed to accommodate the omnidirectionality of the vehicle in the best possible way.

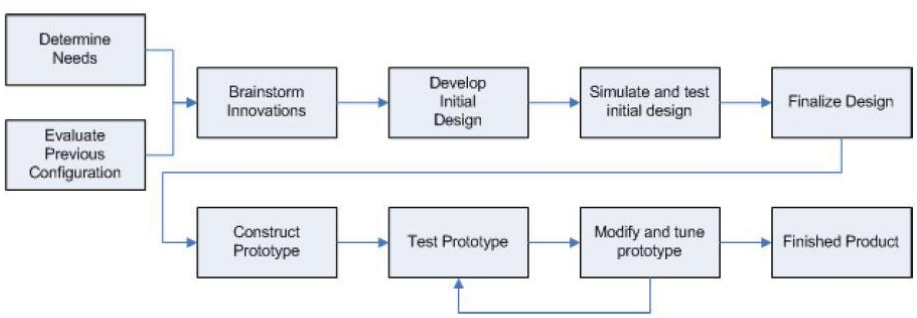


Figure 2: Design process followed to develop Moonwalker.

After developing the platform and designing the basic software architecture, testing of the first versions of the algorithms was performed. Based on observations of the results of the test, appropriate changes were made until the testing was deemed successful and repeatable in as many situations as possible. Every system was tested in a similar fashion.

3 Innovations

This year's entry is a completely new vehicle built from scratch. Some of the interesting things implemented for this year's competition are listed below.

Chassis Innovations

- **Mecanum Wheels** – Moonwalker's drive system is based on Mecanum wheels, allowing completely omnidirectional motion. By separately controlling the four wheels, movement in the forward, sideways and rotational dimensions can be controlled independently. This type of drive system is not typically used in an off-road environment, but Moonwalker is designed to handle potentially rough terrain.
- **Four-Wheel Independent Suspension** – Each of the four motor assemblies is mounted to the body of the vehicle with a shock absorber, and is allowed to travel up and down independent of the body. Since the Mecanum drive system requires good wheel contact in order to work well, the suspension system helps to ensure that the wheels always maintain contact with the ground.
- **Modular Components** – Moonwalker is completely modular, with all components on easily removable drawers, shelves or mounting assemblies. It can be configured differently depending on what is required of it. For example, when testing the drive control system or taxiing the vehicle to a different location, the whole upper part of the frame can be removed, leaving only the necessary components for manual control.

Electronic Hardware Innovations

- **PIC Processors** – Motor control and sensor interfacing are performed on small, low power dsPIC processors. These are perfect for small, low level applications because they take up very little space and can be selected almost exactly to the need, since there are several different models to choose from.
- **On-Board Computer** – Moonwalker is equipped with a built-in computer that performs all the high-level decision making and path planning. The computer is constructed from laptop components, thereby providing plenty of computational power with relatively low power consumption.

Computing and Control Innovations

- **Omnidirectional Vehicle Control** – The navigation and path planning systems fully exploit the extra degree of freedom provided by the Mecanum ODV to maneuver around obstacles in the most efficient way. Being able to control the sideways motion of the vehicle in addition to the forward and turning motion creates a redundant robotic system, where different optimization constraints can be imposed on the trajectory planning system.
- **Rapid Prototyping on Embedded Hardware** – All motor control algorithms and low-level sensor interfacing is performed on PIC processors, which are programmed through the Real-Time Workshop component of Matlab/Simulink. The algorithm is implemented using Simulink block diagrams and Embedded Matlab Functions, and Real-Time Workshop generates C code that can then be compiled and downloaded to the PIC. Rapid prototyping provides great flexibility and simplicity in low level software development.

- **Parallel Processing** – In order to save processing time on the main computer, multiple embedded processors are used in parallel. These embedded processors take care of the low level functions such as drive control and GPS filtering, while the computer is left to handle the high level functions such as image processing and artificial intelligence.

4 Vehicle Design

This section discusses the physical structure of X-Man. This includes the mechanical design, the sensors used, how all the electronic hardware is connected, and how power is distributed to the various components.

4.1 Chassis

Moonwalker's chassis was designed and built from scratch, and utilizes Mecanum wheels to achieve omnidirectional motion. Since Mecanum wheels are not designed to operate on rough terrain, the chassis must be designed to avoid some of the problems associated with off-road travel.

An independent suspension system was implemented to ensure all four wheels maintain contact with the ground. As shown in Figure 3c, a two rod and shock absorber system was chosen to only allow up and down movement. The weight of the robot introduces a moment on the rods and linear bearings. Therefore, the linear bearings were chosen to support a weight estimate of 200 pounds. The linear bearings are shown in Figure 3a and are highlighted in red.

The shock absorber springs were chosen to have a stiffness of 70 pounds per inch, again based on weight estimates. The specific shock absorbers used in the design enable the springs to be pre-loaded, allowing for adjustments in the field based on experimental or environmental conditions.

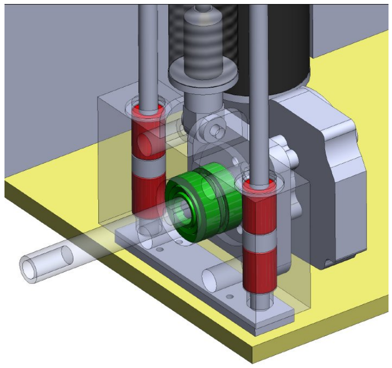
The wheel sub assembly supported by the suspension system is shown in Figure 3b. This assembly consists of a wheel hub, a Mecanum wheel, a shaft, a bearing block containing the linear and tapered bearings, a motor plate, a motor, and the encoder. This assembly is supported by two rods and the shock absorber. The hub, which has two keyed shafts, is attached to the shaft, which then travels to the bearing block. Inside the bearing block are two tapered bearings. Tapered bearings were chosen to handle both the radial and axial loads introduced by the Mecanum wheel dynamics. The bearings are sealed to prevent damage from dirt. The tapered bearings are highlighted in green in Figure 3a.

4.2 Sensor Array

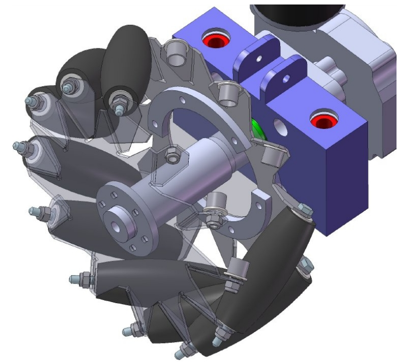
A list of the various sensors used to detect the environment and provide dead reckoning data is shown below.

- **Optical Wheel Encoder** – A U.S. Digital E3 Kit Encoder is used to measure the rotational speed of each wheel. The encoders themselves provide 500 counts per revolution, but they are mounted on the shaft before the gear reduction. The gear ratio was measured to be about 31:1, so the encoders give approximately $500 \times 31 = 15,500$ counts per wheel revolution. This high resolution allows for accurate measurements of the wheel speed, and makes the PI controller for the wheels very robust.

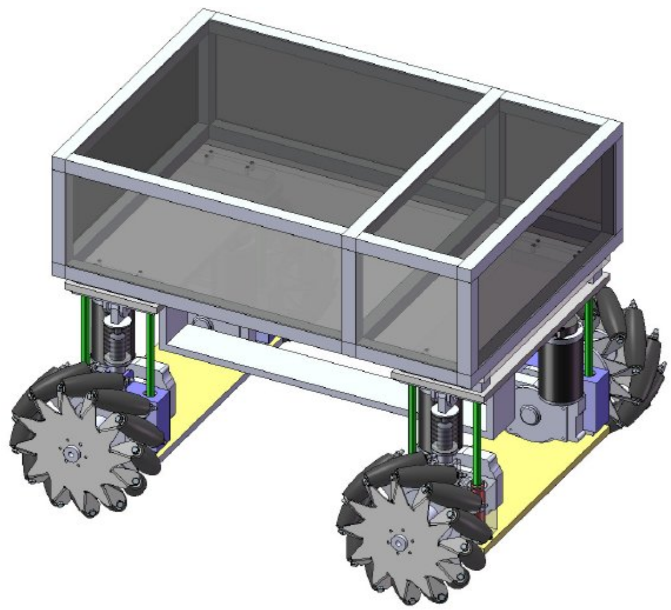
- **Lidar** – A Hokuyo URG-04LX Lidar is used to detect physical obstacles near the robot. This sensor has a 240 degree field of view, and has a range of 4 meters. Being able to sense obstacles that are slightly behind



(a) Bearing Structure



(b) Wheel Assembly



(c) Base Platform

Figure 3: CAD Drawings of Moonwalker's Chassis.

the vehicle is very handy for an omnidirectional platform. The vehicle can move sideways or diagonally without risking sideswiping an obstacle in a blind spot.

- **GPS Receiver** – Moonwalker utilizes a uBlox^{TM} AEK-4P GPS module to receive position fixes from the GPS satellites. The unit is quite small and compact, and provides approximately 2 meter accuracy. Since the accuracy is not good enough to reliably navigate to some of the waypoints in the competition, a Kalman filter is used to fuse the readings from the GPS with the information from the wheel encoders.

- **Camera** – An IDS μEyeLE camera is used for lane detection. The camera was designed to alleviate the problems with using webcams for machine vision applications without sacrificing the ease of use. The lens of the camera is a Tamron 13VM286. It has manual controls to adjust the polarization, optical zoom and focal length. The camera is connected to the on-board computer through USB, and is easily interfaced with Matlab/Simulink.

4.3 Hardware Architecture

A high level diagram of how the various devices connect and communicate with each other is shown in Figure 4.

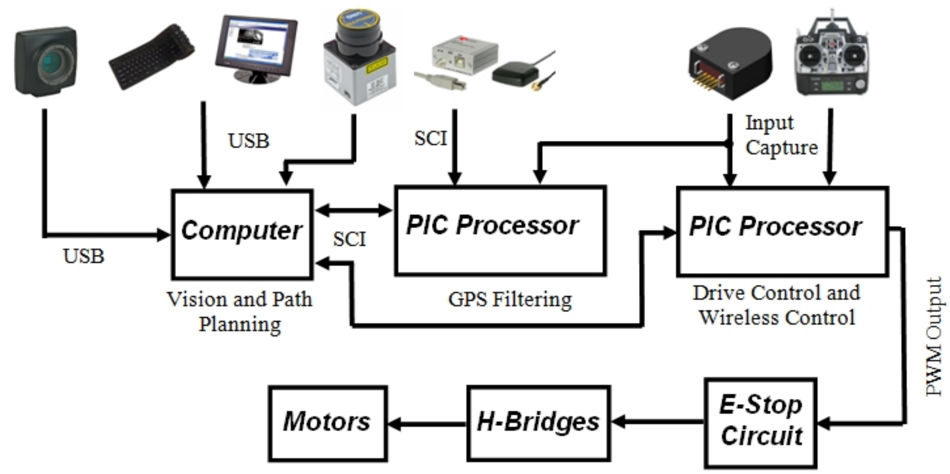


Figure 4: Moonwalker's hardware architecture.

Moonwalker uses three processors in parallel to perform the necessary computing to guide it through the obstacle course and to the GPS waypoints. The cameras and Lidar are connected to an on-board computer, constructed from regular laptop components. This computer runs Matlab on a standard Windows operating system and executes the vision and path planning algorithms, making decisions based on the sensor inputs.

Two Microchip^{TM} dsPIC 30F4011 processors are used to handle the low level functions of drive control, GPS data filtering and sensor interfacing. The processors have PWM, SCI and Input Capture peripheral modules to interface with the various devices on the vehicle.

4.4 Electronic Power Distribution

Moonwalker is powered from a 24 volt battery system. This battery system consists of four 12 volt, 16 AH sealed lead acid batteries wired in two sets of parallel batteries in series. The voltage from the batteries is regulated to produce the other voltages required by the various devices on Moonwalker.

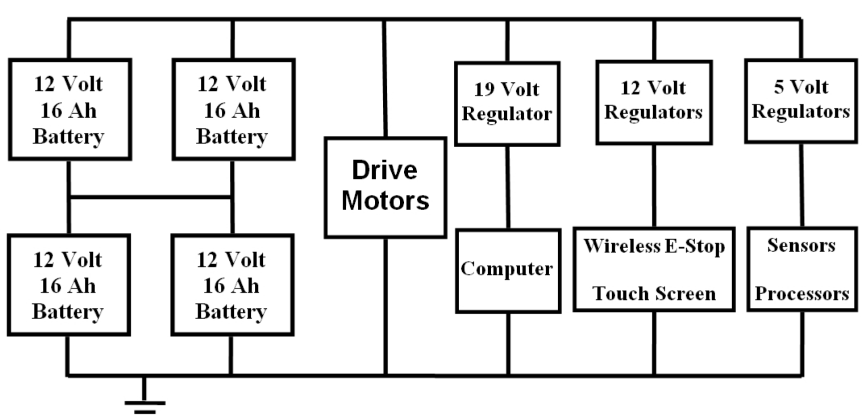


Figure 5: Power distribution block diagram.

5 Control System Design

This section discusses the software and control algorithms used on Moonwalker. All of the software systems are developed in Matlab and Simulink, which allows for easy integration. A high level block diagram of the software systems is shown in Figure 6.

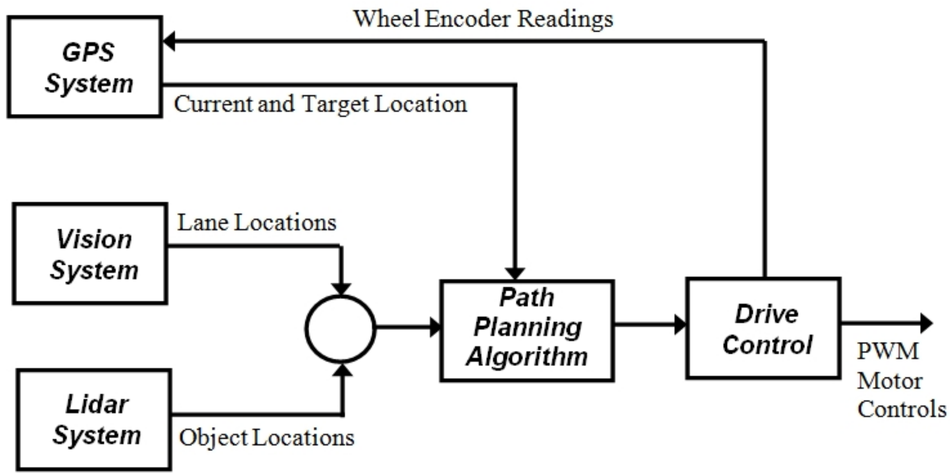


Figure 6: High level block diagram for Moonwalker's software architecture.

5.1 Drive Control System

Moonwalker's drive system uses Mecanum wheels, which are among the most widely used omnidirectional platforms. The Mecanum system consists of four fixed, independently controlled wheels. The contact surface of each wheel is a series of rollers, each oriented at 45 degrees to the wheel's axis of rotation. When the wheel is rotated, the rollers spin freely while applying force along their axis of rotation. By controlling the speed and direction of the four wheels separately, these diagonal forces can be combined to allow the vehicle to move in any direction while at the same time controlling its rotational speed.

Figure 7 shows an illustration of a Mecanum ODV defining the nomenclature of the variables of motion and the geometry of the kinematics model. The rotational speeds of the wheels are given by ω_1 through ω_4, and the positive direction of rotation corresponds to the dashed arrows in Figure 7. The solid arrows represent the force applied by the given wheel, and v_x, v_y and ω_z are the velocities in each of the three degrees of freedom of the ODV.

The kinematic relationship between the wheel rotation speeds and the vehicle's motion can be defined using a single matrix transformation:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix}. \qquad (1)$$

The purpose of the drive control system is to take vehicle speed commands from the path planning algorithm, convert them into wheel speed commands using Equation (1), and then apply PI control loops on each motor using the wheel encoders for speed feedback. This is all performed on a PIC processor, where the speed commands are

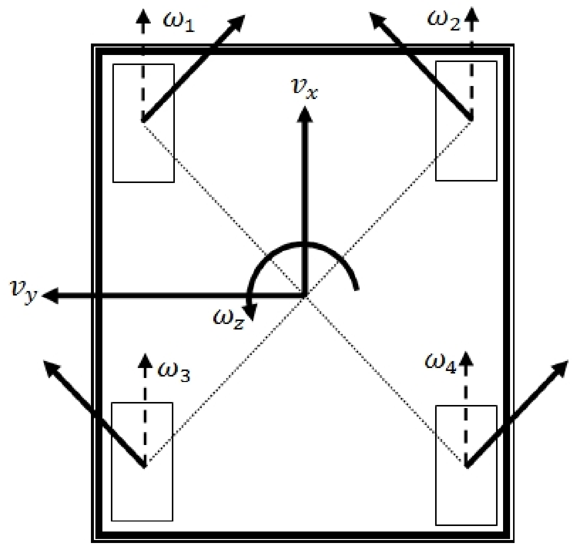


Figure 7: Illustration of the Mecanum ODV platform.

transmitted to it via RS232.

5.2 Vision System

The purpose of the vision system is to reliably detect lane lines, and to estimate their distance and orientation relative to the robot. First, the input image from the camera is conditioned to improve the quality of the image for detection purposes, and the Hough Transform algorithm is applied to detect lane lines. The pixel locations of these detected lines are then transformed into the vehicle's coordinate system using a calibrated kinematics transformation. A block diagram of the vision system is shown in Figure 8.

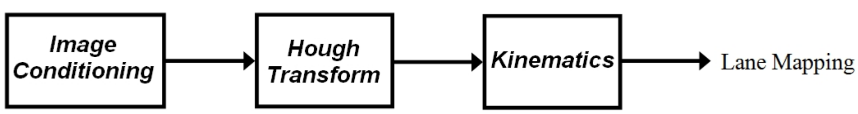


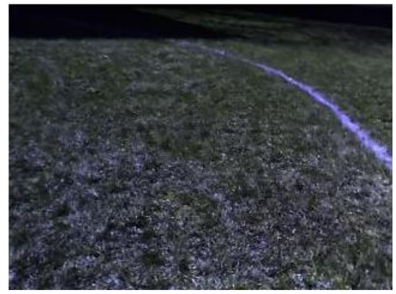
Figure 8: Block diagram of the operation of the vision system.

5.2.1 Image Conditioning

First, the RGB image from the camera is transformed to the XYZ colorspace. This provides much greater contrast between the green grass and white lines because the colorspace is based on luminance and chromaticity instead of just color presence. White distinctly stands out in this colorspace because of this.

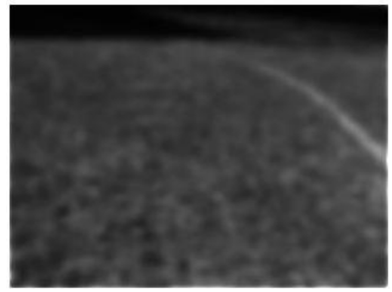
The XYZ image is then transformed into a gray level image, and is blurred by applying a Gaussian pyramid of level 2, and then applying a reverse Gaussian pyramid also of level 2. This practically eliminates the noise introduced by the sharp edges of the grass blades, but retains enough of an edge on the lane lines to allow reliable edge detection. Figure 9 shows the stages of this processing.

(a) Raw Image



(b) XYZ Image

(c) Grayscale Image

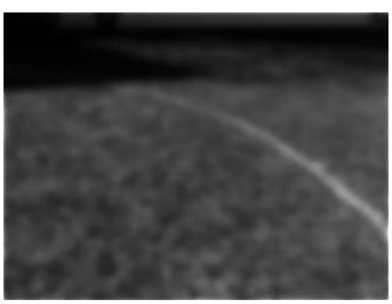


(d) Blurred Image

Figure 9: Stages of the image conditioning process.

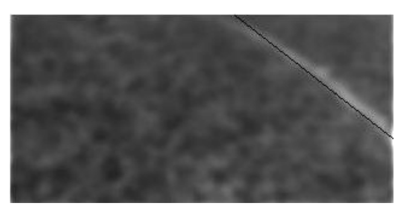
5.2.2 Hough Transform

Sobel edge detection is applied to the blurred image from the image conditioner to generate a binary image with ones at the detected edges. The Hough Transform operates on this binary image and detects one line in each half of the image. To save computation time and to comply with the kinematics algorithm, the input image is cropped such that only the image up to about 10 feet in front of the robot is considered. Figure 10 shows the stages of this process. The endpoints of the superimposed line in Figure 10c are outputted to the kinematics algorithm for projection onto the vehicle's coordinate system.



(a) Blurred Input Image

(b) Edge Detection Image



(c) Output Image

Figure 10: Stages of the Hough Transform process.

5.2.3 Kinematics Transformation

Once the lane lines are successfully detected in the image, their position relative to the vehicle must be estimated using a kinematics algorithm. In order to do this, the algorithm needs to be calibrated with the camera configuration.

The vertical image positions of specific distances are required, as well as the number of horizontal pixels that span one foot at each of these distances. With these measurements, the kinematics algorithm is capable of making quite accurate distance estimations that were found to have errors no more than three inches.

A calibration routine was devised to perform the described measurements using sticky notes to mark the points of interest. After carefully placing the sticky notes at the proper locations, a snapshot was taken and the calibration measurements were made manually. A sample calibration image is shown in Figure 11.

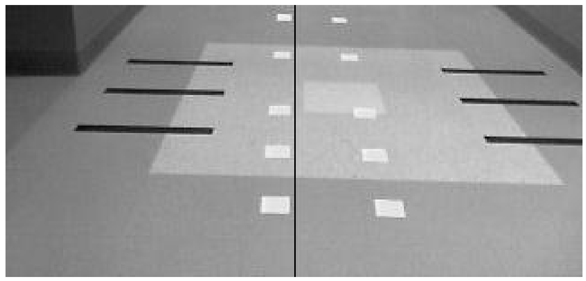


Figure 11: Sticky notes are placed to make appropriate calibration measurements.

The kinematics algorithm first defines the equation for the detected Hough line based on its endpoints. Then the calibration measurements are used with this line equation to compute the coordinates of several points on the line. Using these coordinate values, the distance and orientation of the detected line with respect to the vehicle is found. This is done for both the right and left line, and is outputted to the path planning algorithm.

5.3 Lidar System

A Hokuyo URG-04LX Lidar is used to provide two dimensional obstacle location information. The Hokuyo Lidar has a 240 degree field of view and a range of 4 meters. Before sending information to the path planning system, the incoming obstacle location data is processed. After processing, the coordinates of key locations and a suggestion of where to go are given to the path planning system.

The Lidar system always recommends moving forward until an obstacle comes within a certain radius. This reactionary zone acts as a “shield,” which the robot uses to deflect the obstacles that it gets too close to. Figure 12 illustrates this concept, where the red dashed line represents the shield.

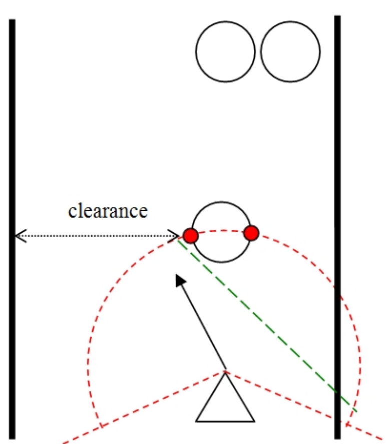


Figure 12: Moonwalker uses a shield based obstacle avoidance system

When an obstacle comes within the radius of the shield, the endpoints of the obstacle are recorded and sent to the path planning algorithm. These correspond to the red circles in Figure 12.

In addition to generating the coordinates for the endpoints of the obstacle, the Lidar system also makes a recommendation on which one of the endpoints to avoid. To make a recommendation, the algorithm considers all the obstacle data and determines the direction of the most open space.

For example, in Figure 12 the robot will see all three barrels on the right and a large open space to the left. It will suggest avoiding the left endpoint, since it is more open on that side. It will send the coordinates of the two red dots and the suggestion to the path planning system.

5.4 GPS System

The purpose of the GPS interface is to provide the path planning system with the distance offsets to the GPS target in the local x and y coordinates, as well as the current heading of the vehicle. A uBlox$^{\text{TM}}$ AEK-4P GPS kit is used to provide position readings, and is configured to give standard GPGGA position correction messages thorough an RS-232 interface.

To compute the distance offset from the GPS target, the current coordinates are first subtracted from the target coordinates to get error values in latitudinal and longitudinal degrees. The Northing and Easting errors are then converted to distance in feet by the following conversion factor, exploiting the knowledge of the radius of the Earth:

$$\frac{3963.1676 \quad \text{mi.}}{\text{rad}} \times \frac{\pi \quad \text{rad}}{180 \quad \text{deg}} \times \frac{5280 \quad \text{ft.}}{\text{mi.}} = 365,219 \frac{\text{ft.}}{\text{deg}}$$

To provide a heading estimate, consecutive GPS readings are compared and the direction of travel is interpreted as the heading. This is unreliable when the vehicle is stopped, but it is assumed the vehicle is continuously moving during the trial run.

The variance in the GPS estimate from the uBlox was measured to be around 8 feet. In order to improve the accuracy, a discrete Kalman Filter is used to fuse the noisy GPS readings with the wheel encoder readings. The filter is modeled with second order Brownian motion dynamics, and is tuned to yield the best performance. The discrete time state space of the model is given by Equation (2), where T is the sample rate of the filter and a_1 through c_4 are the parameters of the slip matrix in Equation (1).

5.5 Artificial Intelligence and Path Planning System

The Lidar system sends the two edges of an obstacle that is within the "shield" along with its recommendation of which way to go. The main task of the path planning algorithm is to decide whether to accept the recommendation of the obstacle detection algorithm, or to override the recommendation while still using it to make a decision of its own. This is done differently for each challenge.

After deciding upon a travel direction, speed commands are generated that will move the vehicle in the given direction. The speed is controlled proportionally to the distance to point of contact from the nearest obstacle or lane line.

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix}_{k} + \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & \frac{1}{2}T^2 \\ 0 & 0 & T \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$z_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & a_1 & 0 & b_1 & 0 & c_1 \\ 0 & a_2 & 0 & b_2 & 0 & c_2 \\ 0 & a_3 & 0 & b_3 & 0 & c_3 \\ 0 & a_4 & 0 & b_4 & 0 & c_4 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix}_{k} \tag{2}$$

5.5.1 Autonomous Challenge

In the Autonomous Challenge, the lane locations from the vision system are used to cross-check the Lidar recommendation. A flowchart of the decision-making process is shown in Figure 13.

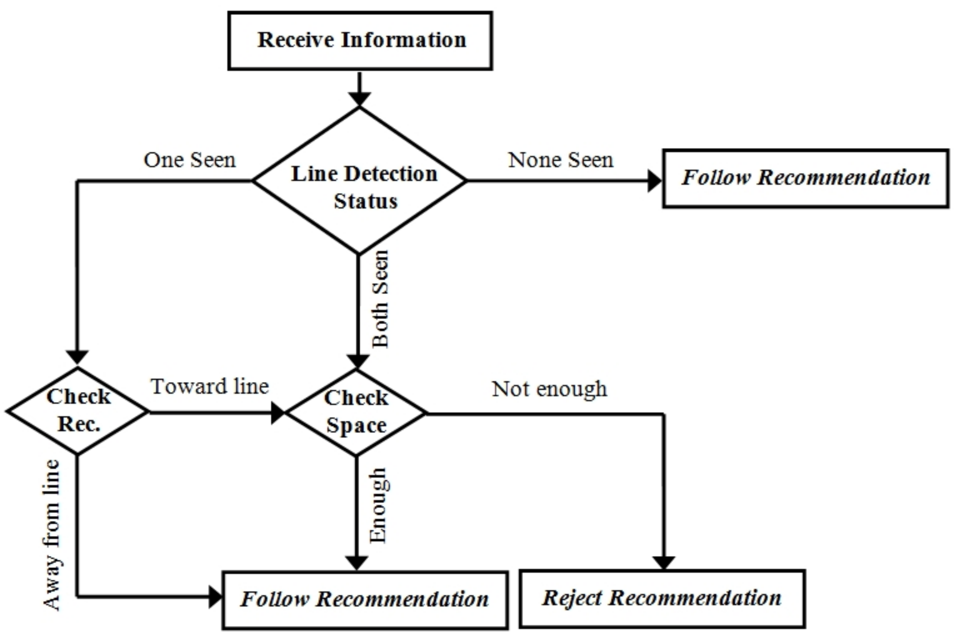


Figure 13: Basic flowchart of the artificial intelligence algorithm

Depending on which lines the vision system is currently detecting, the algorithm uses a different procedure to determine whether to trust the obstacle detection system or not. If both lines are detected, then the distance from the recommended side to the detected line is estimated. If the gap is wide enough, then the algorithm accepts the recommendation. Otherwise, it rejects it and turns the opposite way.

If only one line is detected, the algorithm first checks to see if the Lidar system is asking it to turn toward the line or not. If so, it checks the distance from the obstacle to the detected line and performs the same analysis as described above. However, if the obstacle detection system asks it to turn away from the line, the algorithm readily accepts.

If no lines are detected, the recommendation from the Lidar is followed unconditionally.

After determining which way to go, the algorithm draws a line from the closest point to a detected lane line to the avoidance point from the Lidar system. If the decision is to turn left, then this is done with the right line, and vice versa. This artificial line corresponds to the green dashed line in Figure 12. The distance and angle of the line is computed and the robot is controlled appropriately to avoid it.

5.5.2 GPS Navigation Challenge

In the GPS Navigation Challenge, the location of the target is used to cross-check the Lidar recommendation. When there are no detected obstacles between the vehicle and the target, a straight line path is drawn between them and the robot will travel directly toward the target. However, when an obstacle is detected within the shield range, two segment paths are generated to the target that avoid the obstacle.

If the Lidar suggestion puts the robot on what the artificial intelligence determines to be a round-about way to the target, the recommendation will be overridden. However, if there is little difference in distance to the target between the paths, the artificial intelligence trusts the Lidar.

The cost function for finding a good path is also dependent on whether the robot already passed the same area the path is leading to. This helps the robot avoid going in circles.

6 JAUS Compliance

A system was developed to communicate with the COP as prescribed in the rules of the competition. The entire JAUS compliance system was implemented in Matlab/Simulink, and makes use of the Stateflow toolbox to graphically design the state transitions and actions.

When a JAUS message is received, the system first detects the type of message it is and determines if the identifier is correct. If so, the state machine is triggered to decide how to respond, and the state machine transfers the appropriate command to the vehicle's systems and initiates transmission of the correct response message. A block diagram of the JAUS compliance system for IGVC is shown in Figure 14.

Matlab/Simulink is a very convenient tool for JAUS compliance development, especially the graphical state machine design provided in the Stateflow toolbox. It would be relatively simple to expand this system into a general-purpose JAUS compliance module.

7 Practical Considerations

7.1 Vehicle Speed

The 24 volt DC gear motors used on Moonwalker have a maximum speed of 170 RPM at the applied load, which works out to a maximum traveling speed for the robot to be

$$170\frac{\text{rev}}{\text{min}} \times \frac{60 \quad \text{min}}{\text{hr.}} \times \frac{10\pi \quad \text{in.}}{\text{rev}} \times \frac{1 \quad \text{mi.}}{63,360 \quad \text{in.}} = 5.05\frac{\text{mi.}}{\text{hr.}},$$

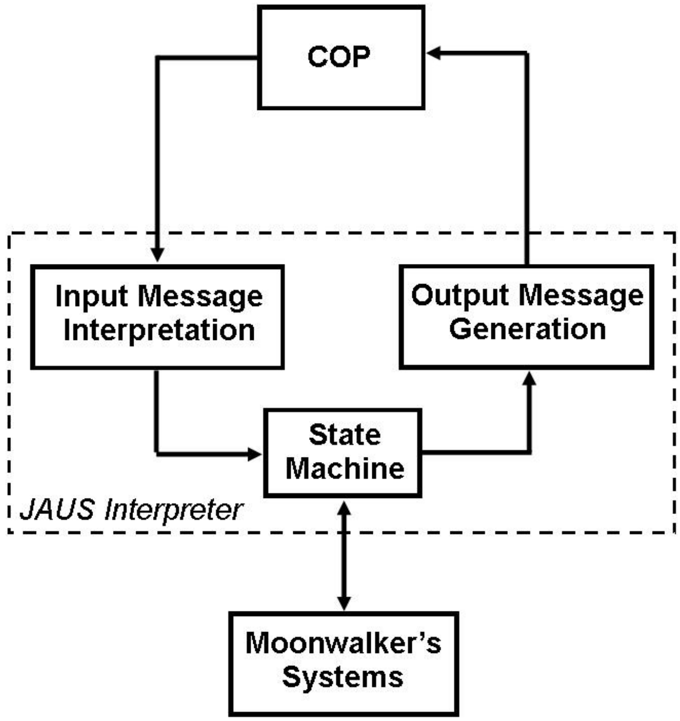


Figure 14: JAUS compliance system block diagram.

using the fact that the wheels are 10 inches in diameter.

7.2 Ramp Climbing Ability

Moonwalker weighs around 200 pounds, and the motors can apply an approximate maximum translational force of 80 pounds at nominal torque. Assuming the force required to propel the robot up a ramp is approximately equal to the weight multiplied by the sine of the slope angle, the maximum slope angle can be roughly computed to be

$$\theta = \sin^{-1}\left(\frac{F}{W}\right) \approx 23.6^\circ.$$

In the field, Moonwalker has been found to be able to climb small ramps with ease, more than capable to negotiate the ramps that are found on the IGVC course.

7.3 Battery Life

The continuous current draw from the on-board components such as the computer, Lidar, PIC processors and all the sensors is approximately 6 amps. Since there are about 32 Ah of charge, Moonwalker can last about 5.33 hours when just idling.

At full power under nominal load, the drive motors take about 12 amps, giving a total of 48 amps when moving at full speed with all wheels. If the vehicle is continually driving with full speed, the batteries would last approximately 35 minutes. In the field, it was found that the batteries last about an hour when doing normal testing.

7.4 Budget and Cost Analysis

Equipment Cost Breakdown

	Quantity	Price	Ext. Price	Cost to Team
Sensors				
Optical Wheel Encoder	4	$52	$208	$208
Hokuyo Lidar	1	$2,375	$2,375	$2,375
Ublox GPS Unit	1	$198	$198	$198
Machine Vision Camera	1	$380	$380	$380
Camera Lens	1	$75	$75	$75
On-Board Computer				
Core2Duo 2.0 GHz Processor	1	$250	$250	$250
MiniATX Motherboard	1	$300	$300	$300
Hard Drive	1	$50	$50	$0
Memory	1	$45	$45	$0
DC Power Supply	1	$85	$85	$85
Touch Screen Monitor	1	$300	$300	$250
Miscellaneous				
Frame Materials	N/A	$1,200	$1,200	$1,200
Batteries	4	$57	$228	$228
Mecanum Wheel Set	1	$710	$710	$710
PIC Processor	2	$15	$30	$0
H-Bridge	4	$125	$500	$0
Wire, Cabling and Connectors	N/A	N/A	$200	$200
Circuit Components and ICs	N/A	N/A	$100	$100
Total			**$7,234**	**$6,559**

8 Conclusion

This year, Oakland University embarked on an experiment to see if it is possible to make a reliable Mecanum ODV platform that can drive on grass. It is believed that off-road omnidirectional vehicles would be very useful, provided enough research effort is devoted to their development.

Acknowledgements

The Moonwalker team is grateful to Peter Taylor, for machining the vehicle. The team would also like to thank the School of Engineering and Computer Science, as well as the Center for Student Activities, for the generous funding and support provided to the efforts of this project. Special thanks also goes to our advisor, Professor Ka C. Cheok, for his guidance in learning the skills necessary for IGVC.