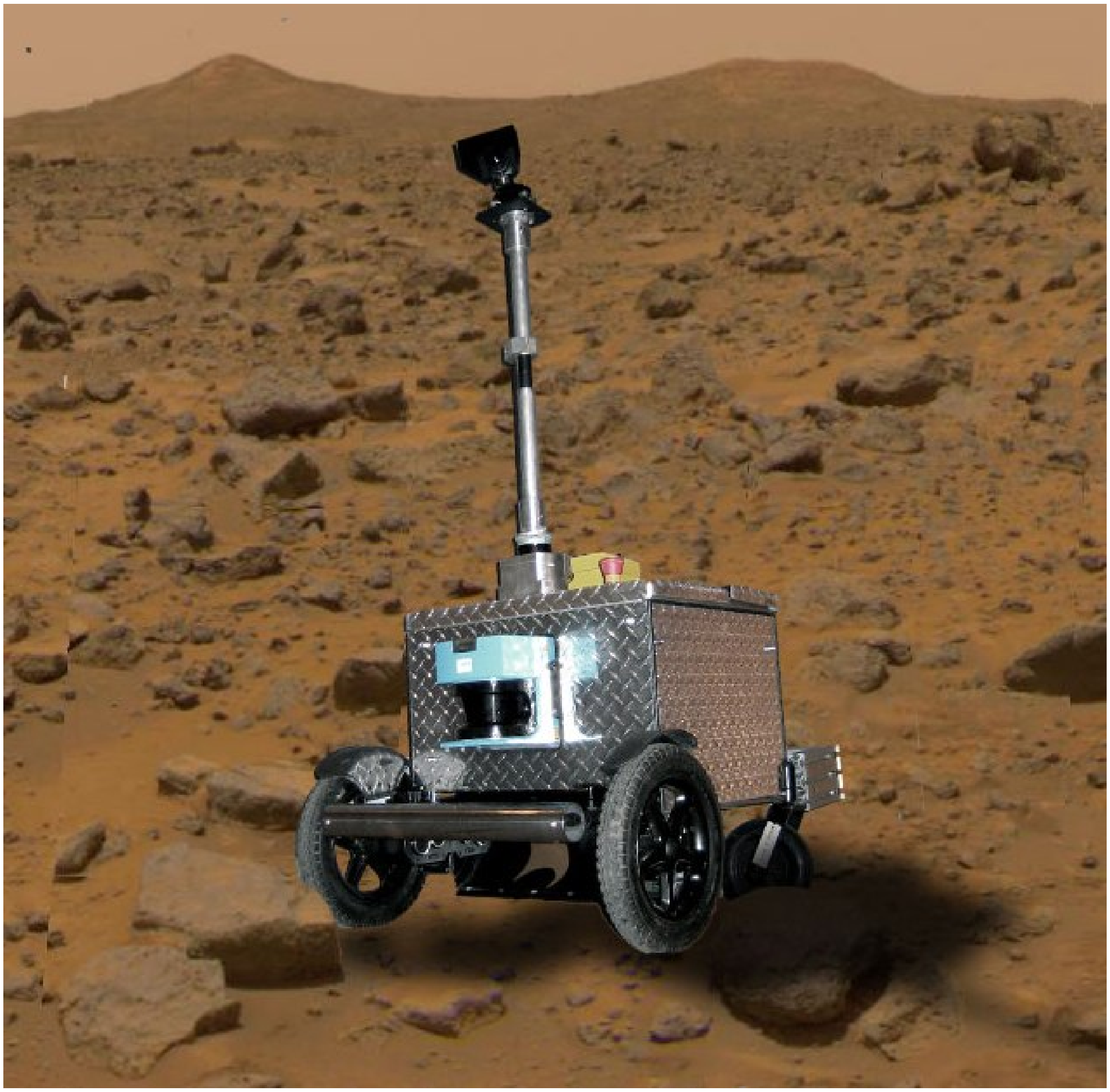
Oakland University Presents:



X-Man

2008 Intelligent Ground Vehicle Competition
May 30 - June 2, 2008

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 X-Man for the 2008 Intelligent Ground Vehicle Competition. The title "X-Man" denotes "unmanned," and of course is similar to the name of the popular comic strip and movie trilogy. X-Man was developed from an electronic wheelchair, but all that remains of the original wheelchair are the motors and wheels. Everything else has been redesigned to meet the requirements decided upon during the design phase.

This document discusses the design of X-Man's systems, and illustrates the innovative hardware and software solutions developed to make a robust and intelligent vehicle for the 2008 competition.

2 Project Management

This section discusses the structure and organization of the Oakland University team, as well as the design process followed to develop X-Man's systems.

2.1 Team Membership

This year, the Oakland University team has recruited several new members. Each of X-Man's control systems has a single team member in charge of designing it and enlisting the help of the other members to debug and test it. This is a great improvement over last year, where there were not enough people to sufficiently work on and test all the necessary control systems and algorithms. A list of the team members and their amount of contribution is shown below.

X-Man's Design Team

Name	**Academic Status**	**Department**	**Expended Hours**
Pavan Vempaty	Graduate Student	Systems Engineering	300
Micho Radovnikovich	Graduate Student	Systems Engineering	300
Ravi Anand	Graduate Student	Electrical Engineering	100
Pavan Raja	Graduate Student	Systems Engineering	50
Naveen Chilukoti	Undergraduate	Electrical Engineering	200
Sriharish Govindarajulu	Undergraduate	Electrical Engineering	100
Nathan Jones	Undergraduate	Electrical Engineering	500
Phil Stene	Undergraduate	Computer Science	50
Alex Pawlowski	Undergraduate	Mechanical Engineering	60

All of the students have graciously volunteered their time to make X-Man as complete and successful as possible.

2.2 Team Organization

Each of X-Man's systems has a team member in charge of designing it. The team leaders are also responsible for coordinating the efforts of the team and making strategic decisions based on the recommendations of the subsystem project leaders.

Each team member is responsible for scheduling testing time for his system and enlisting the help of the other members to meet the deadlines set by the team leaders. An organization chart of the team is shown in Figure 1.

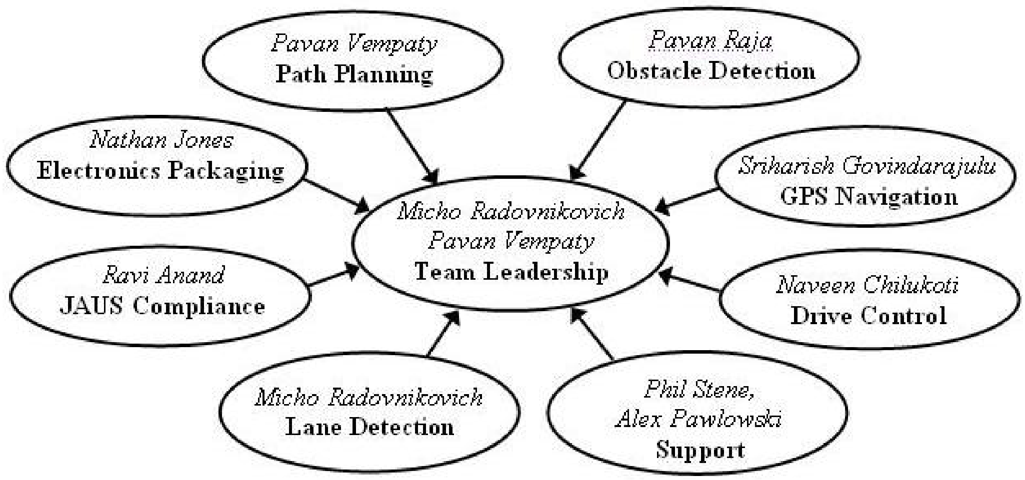


Figure 1: Organization of the X-Man design team

2.3 Design Process

X-Man's design process followed the diagram shown in Figure 2.

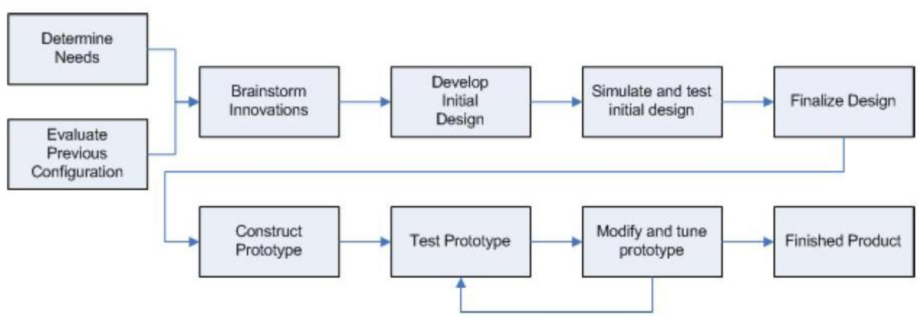


Figure 2: Design process followed to develop X-Man

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.

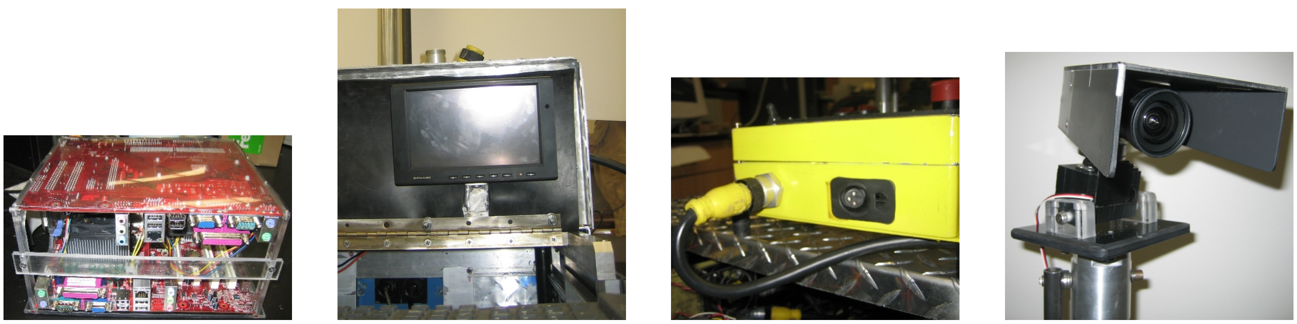
After developing the platform and designing the basic software architecture, testing of the first versions of the algorithms was performed. Based on observations of the results of the test, appropriate changes were made until the testing was deemed successful and repeatable in as many situations as possible.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

3 Innovations

This year's robot is completely different from any previous IGVC entry from Oakland University. With new members came new ideas and better organization, and fostered the development of better, more reliable and more permanent designs. Some of the interesting things implemented for this year's competition are listed below.

Electronic Hardware Innovations

- **On-Board Computers** – The parts for two complete dual-core processor computers were separately purchased and packaged in a custom box to maximize space utilization. One computer is used for obstacle detection, GPS and JAUS, and the other is used to handle the vision and path planning algorithms. Obstacle data from the first computer is transmitted to the path planning computer using Ethernet and UDP. These two computers allow X-Man to have parallel processing capability.
- **Touch Screen Control** – A seven inch touch screen monitor is used to control the two on-board computers. It also controls the modes of operation of all the software systems on X-Man through a Matlab based GUI.
- **Easily Accessible Electronics** – All of X-Man's electronics, including H-bridges, microcontroller, on-board computers and voltage regulation board, are all packaged in a drawer that slides out for easy access. All the electronic modules are secured and well packaged.
- **Battery Charging Circuit** – An on-board circuit is used to charge X-Man's batteries without disconnecting or removing them. The charger is simply plugged into the E-stop box on the top of the vehicle, and the batteries start charging when a switch is thrown. The power systems of the robot are isolated when the batteries are charging.



(a) Computer box | (b) Touch screen | (c) Battery charging socket | (d) Rotating camera head

Figure 3: Hardware innovations on X-Man

Computing and Control Innovations

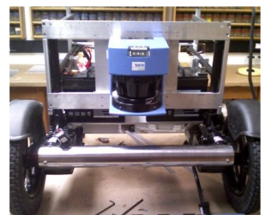
- **Rapid Prototyping on HCS12** – X-Man's drivetrain system was implemented on a Motorola HCS12 microcontroller using real-time embedded C code generated from Simulink's Real-Time Workshop. This provides greater flexibility and reliability than writing the drive control loops by hand in C.
- **Servo Controlled Panning of Vision Camera** – In order to have a more reliable lane detection system, a servo controlled panning system was developed to allow the vehicle to intelligently turn its head to look for lines when it needs to.
- **Motorized Telescoping of Camera Shaft** – In addition to the automatic panning system, a wormgear DC motor is used to lengthen and shorten the camera shaft. Currently this is manually controlled, but in the future it will be intelligently controlled by the robot.

- **Embedded Wireless Manual Control** – A purely microcontroller based system for controlling the robot with a radio model aircraft controller was implemented. This way, the robot can be manually controlled without having to boot up a computer.

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



X-Man's chassis is built on the base of a motorized wheelchair, but only the motors and core frame structure remain of the original design. All structural modifications were custom made to accommodate the requirements of X-Man's hardware placement.

The interior of the chassis contains three layers: a battery tray on the bottom, a sliding drawer in the middle, and a hinged lid on the top. The sliding drawer holds all of the on-board electronics, as well as a flexible, waterproof keyboard. On the underside of the hinged lid is mounted a touch screen monitor. When it is necessary to control X-Man, the lid is simply flipped up and the keyboard on the drawer layer is accessible along with the monitor.

The interior is completely enclosed, thereby making X-Man "rainproof." In the case of a drizzle, X-Man will be able to run without risking damage to the electronic equipment. On the top of the robot are the camera shaft and the control box. The payload for the competition has a dedicated rack on the rear of the vehicle.

The frame is very compact and uses most of the available space. This allows for good maneuverability without having to sacrifice packaging ability or restricting the position of the payload.

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.

- **SICK Lidar** – A SICK LMS-200 Lidar is used to detect physical obstacles in front of the robot. The Lidar sweeps at 20 Hz with one degree resolution. A vector of 181 range points is gathered from the RS-232 interface.

- **Digital Compass** – A Honeywell HMR3200 digital compass is used to measure the magnetic heading of the robot to provide more dead reckoning information than can be extracted from the wheel encoders alone.
- **GPS Receiver** – X-Man utilizes a Ublox 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 compass and wheel encoders. The on-board dead reckoning sensors provide cross-checking for the noisy GPS position fix.
- **Machine Vision Camera** – An IDS μEyeLE camera is used for lane and pothole detection. The camera was designed to alleviate the problems with using webcams for machine vision applications without sacrificing the ease of use. The camera connects to one of the on-board computers through USB, and is easily interfaced with Matlab. The lens of the camera is a Tamron 13VM286. It has manual controls to adjust the polarization, optical zoom and focal length.

4.3 Hardware Architecture

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

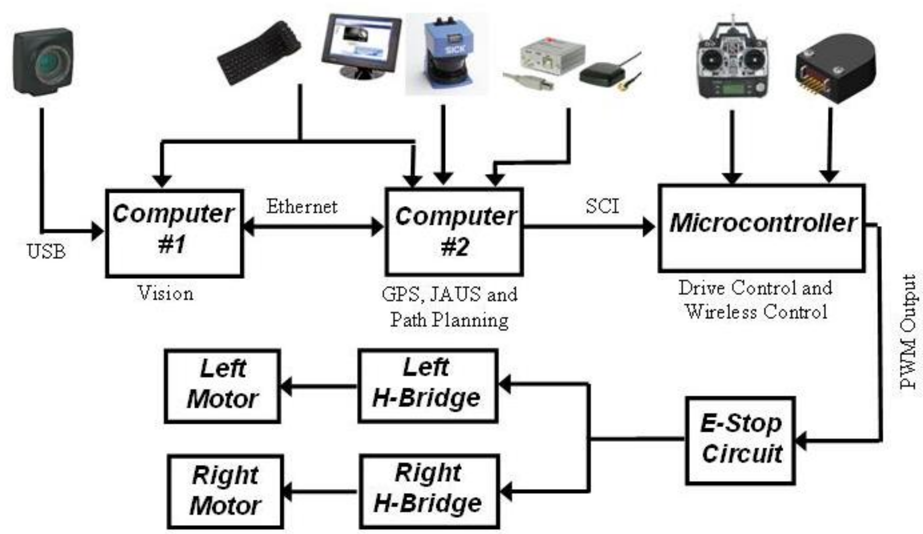


Figure 4: X-Man's hardware architecture

X-Man has two on-board computers to process input data from the sensors and to intelligently control the robot. One computer is used solely for vision since it is the most computationally intensive. The outputs from the vision system are transmitted to the second computer via Ethernet, where the data is integrated with obstacle data from the SICK to perform path planning.

In addition to obstacle detection and path planning, the second computer is also used to process data coming from GPS and to interpret JAUS commands. For the Navigation Challenge, the vision computer is not used, and a different path planning system is run on the second computer to avoid obstacles and approach the GPS target points.

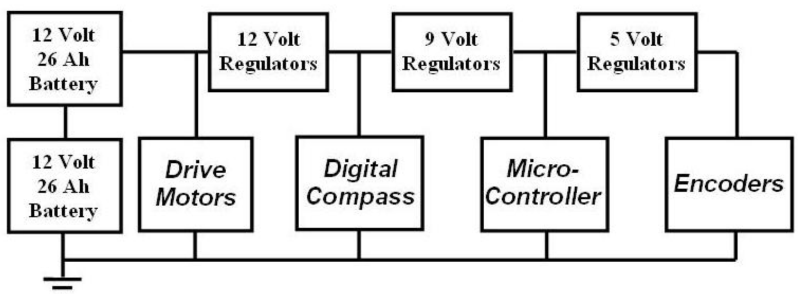
The path planning computer sends forward and steering speed commands to the HCS12. This controls the robot's wheels with PI controllers, using feedback from the encoders. PWM signals to the motors are routed through the E-stop circuit to the H-bridges.

4.4 Electronic Power Distribution

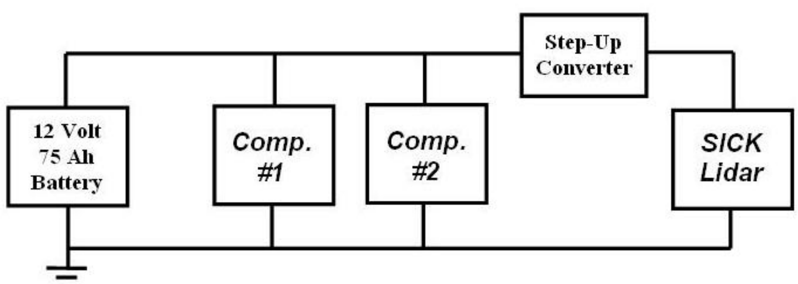
The power distribution system for X-Man uses components and equipment designed for use in the Automation industry. These components were chosen for the ease of packaging and maintaining. The circuit designs for X-Man can also be reused for future generations of robots.

The design of the power distribution circuits are based on the use of two power sources, a 24 volt battery system and 12 volt battery. The 24 volt system was developed using two 26 amp-hour, 12 volt batteries wired in series. The 24 volt circuit is the primary source that powers the robot's drive motors and all of the low-power devices. The 24 volts is regulated to 12 volts, 9 volts, and 5 volts to provide power to these low power devices.

Another 75 amp-hour, 12 volt battery is used to power the on-board computers and the SICK Lidar, which are the only heavily power hungry devices on X-Man. A DC-DC step up converter is used to convert the 12 volts of the battery into the 24 volts that is required by the SICK. A block diagram of the power distribution system is shown in Figure 5.



(a) Primary power system for motors and low power devices



(b) Secondary power system for continuous high power devices

Figure 5: Primary and secondary power systems on X-Man

A very challenging portion of the design of X-Man's electrical system was the placement and packaging of the power circuits and electronics. This was performed by housing all circuits in a 16 x 18 inch packaging drawer. The electronics packaging drawer gives easy access to the electronic circuits, and quick-disconnecting connectors make removing the drawer to service any of the electronic components possible. Cooling systems are also installed inside the drawer to help dissipate heat from various components.

5 Control System Design

This section discusses the software and control algorithms used on X-Man. All of the software systems were developed using Matlab and Simulink, which allowed for easy integration. A high level block diagram of the software systems is shown in Figure 6.

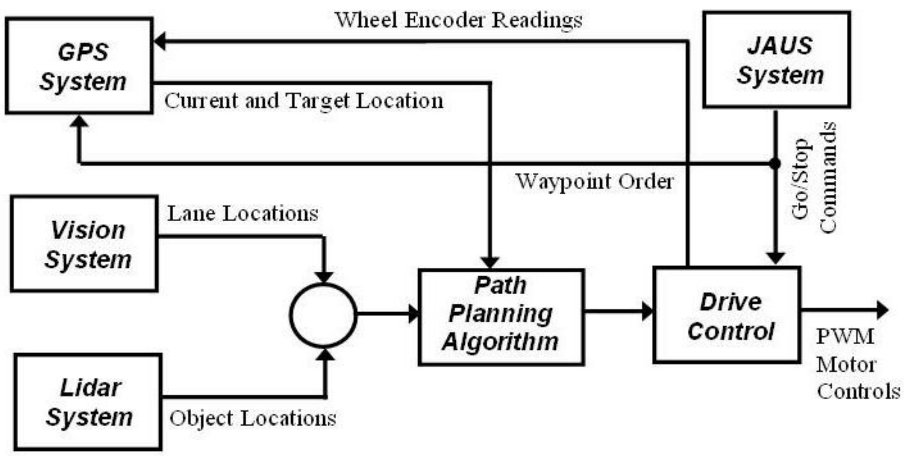


Figure 6: High level block diagram for X-Man's software architecture

5.1 Drive Control System

The drive control system is designed to take in a forward and steering speed command, convert them into individual wheel speed commands, and then control the wheels using PI controllers with encoder feedback. The system was developed using Simulink, and real-time C code was generated using Real-Time Workshop in Matlab. This code was then embedded on the HCS12 microcontroller. A block diagram of the drive control system is shown in Figure 7.

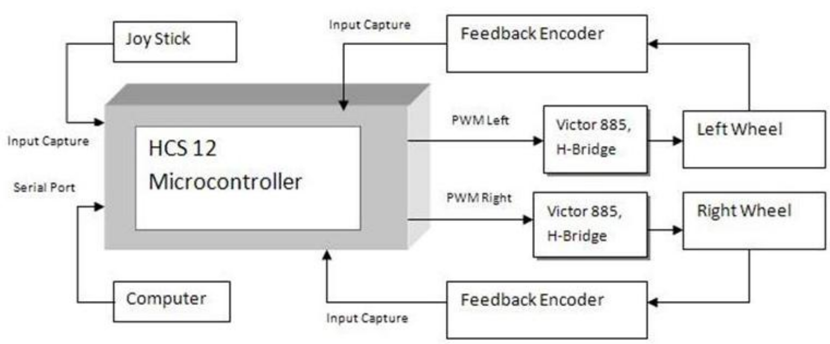


Figure 7: Top level structure of X-Man's drive control system

A differential drive system is used to control X-Man. With this control scheme, the vehicle is steered by adjusting the ratio between the speeds of the two wheels. For differential drive, the individual wheel speeds can be related to the forward and steering speeds:

$$V_r = \frac{2V_f - WV_s}{2} \qquad V_l = \frac{2V_f + WV_s}{2}$$

where V_f is the forward speed of the robot, V_s is the steering speed of the robot, and W is the width of the robot.

The forward and steering control signals are input from the joystick or the computer. When the computer is not running, the drive control system automatically changes to use the joystick as its input source. When the autonomous control algorithms are running on the computers, the manual control is overridden by the autonomous commands. However, the computer can give control back to the joystick at any time through the GUI on the touch screen.

In order to smooth out sharp acceleration impulses and maintain consistent speed, a PI controller for each wheel is used. The PI controller control compares the speed of the wheel as measured by its encoder to the reference speed from the command signal and tries to maintain zero error between them. The gains of the controller were determined experimentally, since the motors could not be modeled accurately due to the lack of the motor parameters.

5.2 Vision System

The purpose of the vision system is to reliably detect lane lines and potholes, 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. Then, pattern matching is applied to detect potholes and the Hough Transform algorithm is applied to detect lane lines. Finally, the pixel locations of these detected features 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.

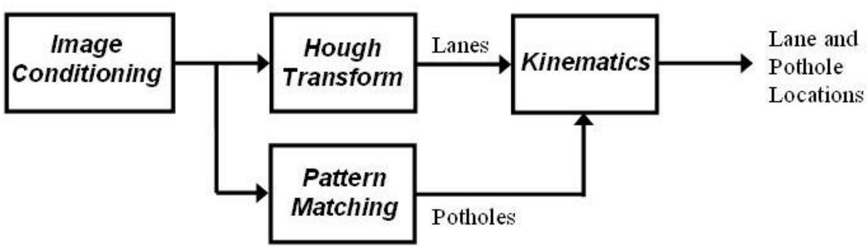
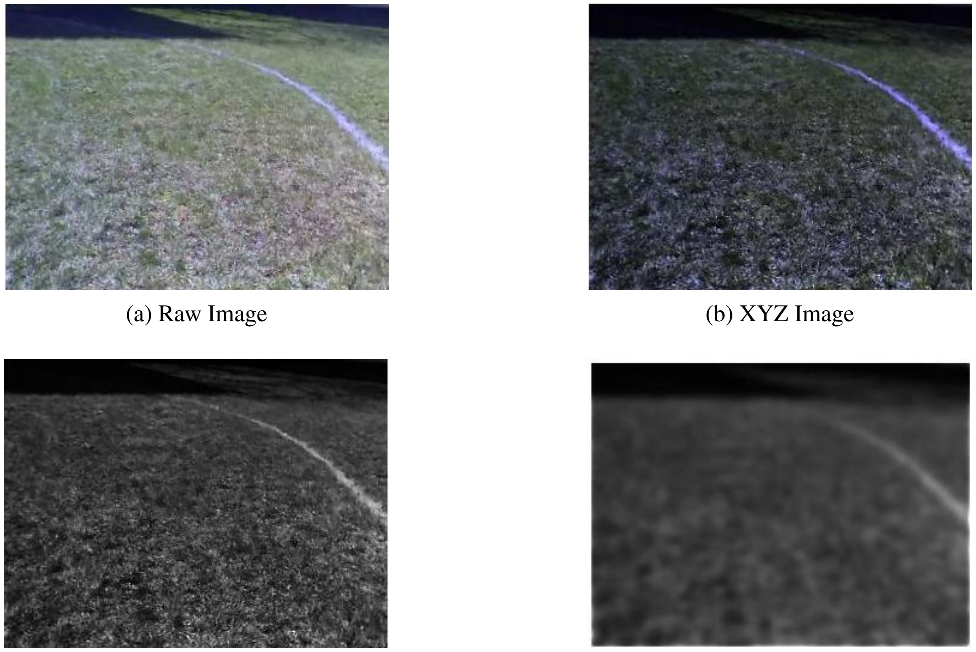


Figure 8: Block diagram of the operation of the vision system

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 Pattern Matching

Potholes are detected using pattern matching. For this, a template image of a pothole is stored in memory. Spatial cross-correlation is performed with the template and the incoming camera image. The result is thresholded at a level low enough such that small differences between the template and real potholes still result in a detection, but high enough as to limit false detections. At the time of this writing, a template for a pothole is not available, so no experimental images can be shown.

However, the detection process will be similar to what is shown in Figure 10, where the template of a construction barrel is correlated with a live image. In the final application, the template of the barrel will be replaced by a pothole template. After detecting potholes, their image coordinates are projected onto the vehicle's coordinate system by the kinematics algorithm, thereby estimating their location for the path planning algorithm to use.

5.2.3 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 11 shows the stages of this process.

The endpoints of the superimposed line in Figure 11.c are outputted to the kinematics algorithm for projection onto the vehicle's coordinate system.

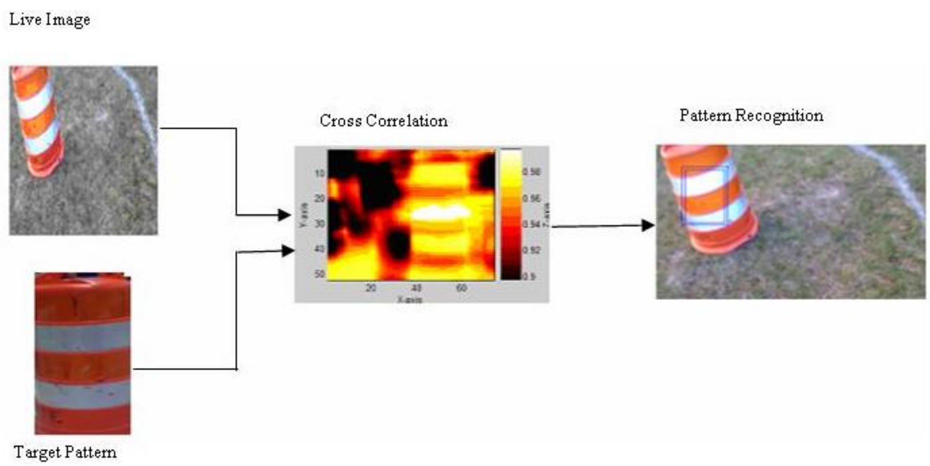


Figure 10: Spatial domain cross-correlation for pattern matching

(a) Blurred Input Image (b) Edge Detection Image (c) Output Image

Figure 11: Stages of the Hough Transform process

5.2.4 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 12.

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.

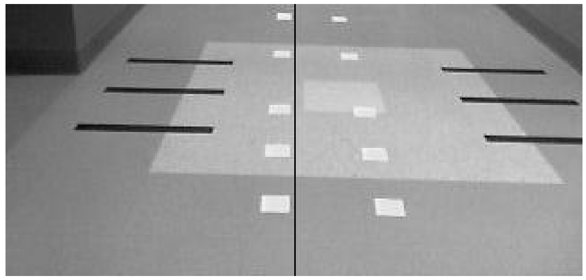


Figure 12: Sticky notes are placed to make appropriate calibration measurements

5.3 Obstacle Detection System

The obstacle detection system is responsible for detecting objects in front of the vehicle and communicating the locations of the obstacles to the path planning algorithm. The obstacle detection algorithm observes obstacles out to about 12 feet, but it only reacts to obstacles within a specified safe distance, and within a certain arc. This reactionary zone acts as a "shield," which the robot uses to deflect the obstacles that it gets too close to. Figure 13 illustrates this concept, where the red dashed line represents the shield.

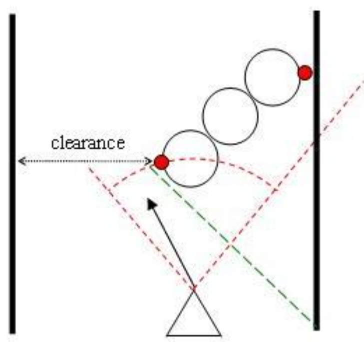


Figure 13: X-Man uses a shield based obstacle avoidance system

When an obstacle comes within the radius of the shield, the endpoints of the obstacle are both recorded and sent to the path planning algorithm to decide which endpoint to steer away from. These correspond to the red circles in Figure 13.

In addition to generating the coordinates for the endpoints of the obstacle, the detection system also makes a recommendation on which one of the endpoints to avoid. To make a recommendation, the algorithm looks on either side of the obstacle to see how much space is available, and recommends the direction that is more open.

For example, in Figure 13 the robot will see the two recessed barrels on the right and a large open space to the left. Therefore, it will suggest avoiding the left endpoint. This decision is reviewed by the path planning algorithm where it is cross-checked with the vision data.

5.4 Artificial Intelligence and Path Planning System

The obstacle detection system sends the two edges of an obstacle that is within the "shield" along with its recommendation. The main task of the artificial intelligence algorithm is to decide whether to accept the recommendation of the obstacle detection algorithm, or to override the recommendation. This decision is made using the lane location data from the vision system. A flowchart of the artificial intelligence algorithm is shown in Figure 14.

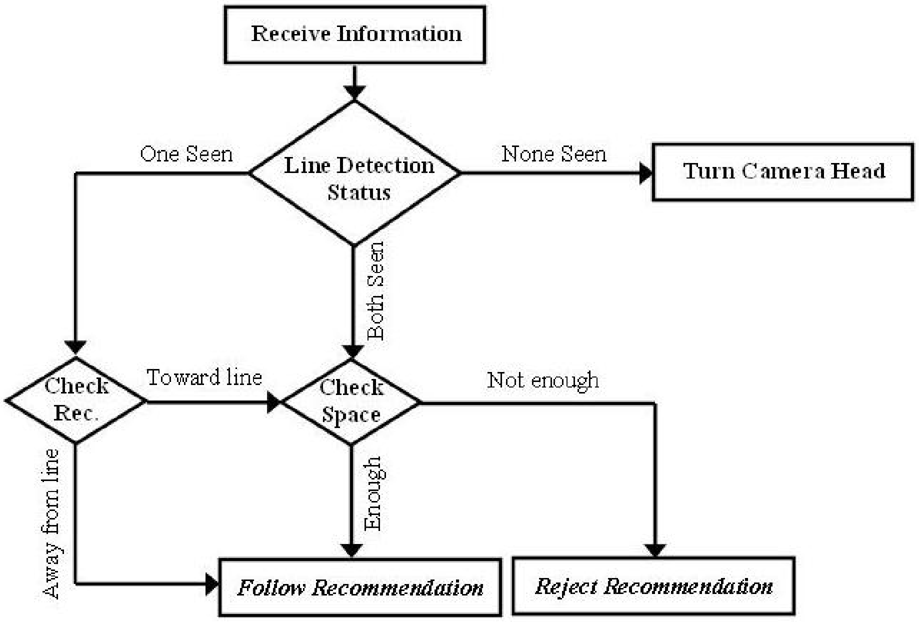


Figure 14: Basic flowchart of the artificial intelligence algorithm

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 obstacle detection 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 robot slows down or stops, then turns its camera head until it does see a line, and then it reverts to one of the above cases. If it doesn't end up seeing a line, it simply follows the previous heading.

After determining which way to turn, the algorithm draws a line from the closest point to a detected lane line to the avoidance point from the obstacle detection 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 13. The distance and heading of the line is computed and the robot is controlled appropriately to avoid it.

5.5 GPS Waypoint Navigation System

A Matlab program was developed to receive data from the GPS receiver via a serial port. Received data is parsed to get the latitude and longitude position of the robot, which is validated using Google Earth. There was a marginal difference between the actual location and GPS location data.

The Geodetic coordinates are converted into Cartesian coordinates using the below equations and are input to an Extended Kalman filter. The filter estimates the x - y position of the robot and determines the heading (radians) of the robot. Then the x, y, heading and successive waypoint of the robot are input to the path planning algorithm. The successive waypoints are retrieved from the JAUS interpreter. A diagram of the control structure is shown in Figure 15.

$$v = \frac{a}{\sqrt{1 - e^2 \sin^2(lat)}} \qquad h = \text{height above geoid}$$

$$x = (v + h)\cos(lat)\cos(lon) \qquad y = (v + h)\cos(lat)\sin(lon)$$

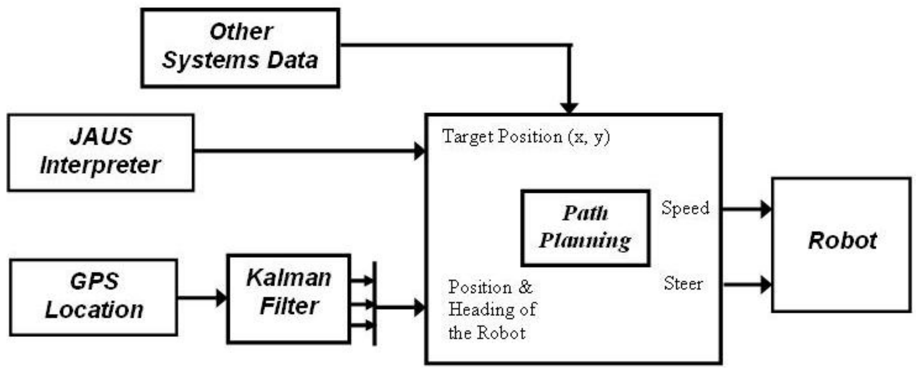


Figure 15: Control structure for GPS waypoint navigation

5.6 JAUS Interpretation System

The JAUS system was implemented using both Microsoft C# .NET and Matlab version R2007a. After some analysis, the Matlab system was chosen because it was much easier to interface with the rest of X-Man's systems. A diagram showing the implemented architecture is shown in Figure 16.

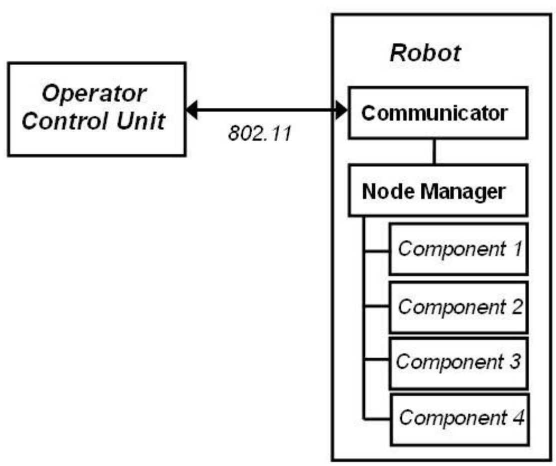


Figure 16: Implemented JAUS architecture

Using the reference architecture documentation from the JAUS Working Group, the Communicator block in Figure 16 was developed to receive the JAUS commands. After recording the commands, it has to act upon them. For the JAUS challenge, it was only necessary to handle a subset of the full JAUS command set.

The focus of the system is on starting and stopping the vehicle in autonomous mode, enabling a warning device, and querying GPS waypoints. The Communicator block performs all of these functions, and when JAUS is enabled, it relays the commands to X-Man's control systems such that it responds properly. Since all systems are developed in Matlab and Simulink, this communication is very simple.

6 Practical Considerations

6.1 Safety, Reliability and Durability

Throughout the design process, safety, reliability and durability have always been a priority. X-Man's design fulfills these objectives through the redundant E-stop switches, completely insulated wiring, compact packaging, and material and motor strength. The payload rack can carry much more than the 20 pound box required for the competition, as seen in the picture to the right.

The artificial intelligence performance and status is always tracked to observe the behavior of the decision making algorithms. If the decision making process is chaotic, the AI safety algorithm will terminate the entire software process to avoid un-wanted and dangerous events.

6.2 Scalability

There are several features that make X-Man very scalable and adaptable to new designs. Some of these are listed below:

- X-Man's two dual-core computers gives it more computing power not only for the current data and algorithms but also for the data and algorithms that would be developed in the future.
- The sensors that are embedded in X-Man serve to give complete information of its surroundings. These sensors are affixed in such a way that they can be used and controlled with any control scheme or AI upgrade.
- X-Man is robust in its structural design, which allows it to easily integrate sensor, structural or AI upgrades.
- With its immense battery power, redundant power distribution and redundant motor controllers make X-Man's electrical system address any upgrade in terms of sensor fusion or control schemes.
- With two powerful high torque DC motors, X-Man could easily carry more load and adapt to any enhancement in its drivetrain control system.
- X-Man's drive control system is designed to interface with any SCI or JAUS based communication. This gives flexibility to any third party control that intends to drive X-Man.

6.3 Budget and Cost Analysis

The table on the next page shows a cost breakdown of X-Man's equipment. The largest economic innovation on X-Man is the on-board computers. The two computers were built for only about $845, which is less than what even a single retail laptop would have cost. In return however, twice the computational power is achieved, and the custom box makes the computers take up very little space.

Equipment Cost Breakdown

	Quantity	Price	Ext. Price	Cost to Team
Sensors				
Optical Wheel Encoder	2	\$52	\$104	\$104
SICK Lidar	1	\$4000	\$4000	\$0
Digital Compass	1	\$175	\$175	\$175
Ublox GPS Unit	1	\$198	\$198	\$198
Machine Vision Camera	1	\$380	\$380	\$380
Camera Lens	1	\$75	\$75	\$75
On-Board Computers				
2.4 GHz Dual Core Processor	2	\$70	\$140	\$140
MicroATX Motherboard	2	\$50	\$100	\$100
Hard Drive	2	\$50	\$100	\$0
Memory	2	\$45	\$90	\$0
DC Power Supply	2	\$50	\$100	\$100
Touch Screen Monitor	1	\$300	\$300	\$250
Flexible Keyboard	1	\$15	\$15	\$15
Miscellaneous				
Electric Wheelchair	1	\$1,100	\$1,100	\$1,100
12 Volt, 75 Ah Battery	1	\$230	\$230	\$230
HCS12 Microcontroller	1	\$85	\$85	\$85
Motor Controller	2	\$115	\$230	\$230
Wire, Cabling and Connectors	N/A	N/A	\$200	\$200
Circuit Components and ICs	N/A	N/A	\$100	\$100
Total			**\$7,722**	**\$3,482**

7 Conclusion

This year, Oakland University has made great progress in developing a long-term vehicle platform for future competitions. The efficient mechanical design and electronics packaging, combined with the robust control algorithms provide a very strong base on which to expand.

Acknowledgements

The team would like to thank Peter Taylor, the Oakland University machinist, for his help with the mechanical construction of X-Man. Also, the team is thankful to Oakland graduate student Aravinda Nanduri for her contribution. The team would also like to thank the Center for Student Activities and the School of Engineering and Computer Science for their generous funding and support that made this project possible.