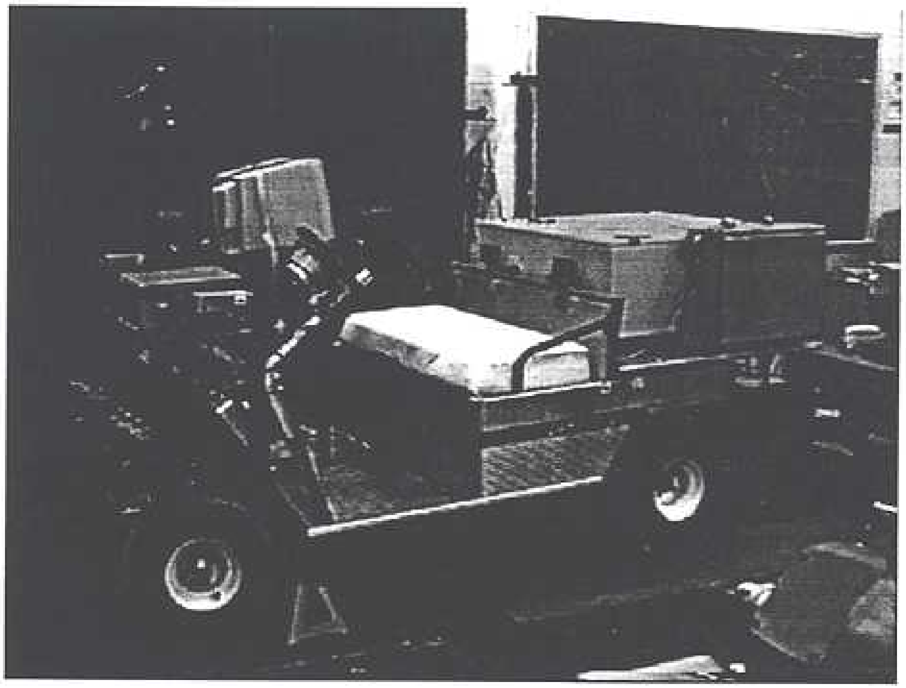
MOSFET

The **M**ichigan **O**ff-**R**oad **S**ensor **F**using **E**xperimental **T**estbed



Team Members: Mike Beauvais, Randy DeFauw, Chris Kreucher

Vehicle Electronics Laboratory, EC 1231
Electrical and Computer Engineering Department
University of Michigan - Dearborn
Dearborn, MI 48128-1491
Email: golf_cart@umich.edu

Faculty Advisor: Professor Sridhar Lakshmanan

Overview

The Michigan Off-road Sensor Fusing Experimental Testbed (*MOSFET*) is an autonomous vehicle designed to navigate an unknown terrain which includes obstacles, lanes, and other traditional roadside obstructions. In order to gather data about the world around it, *MOSFET* is equipped with several heterogeneous sensors. These sensors include one forward-looking camera used for obstacle detection and lane tracking, two side-looking cameras used for lane sensing, and a bank of sonar sensors used to detect physical obstacles.

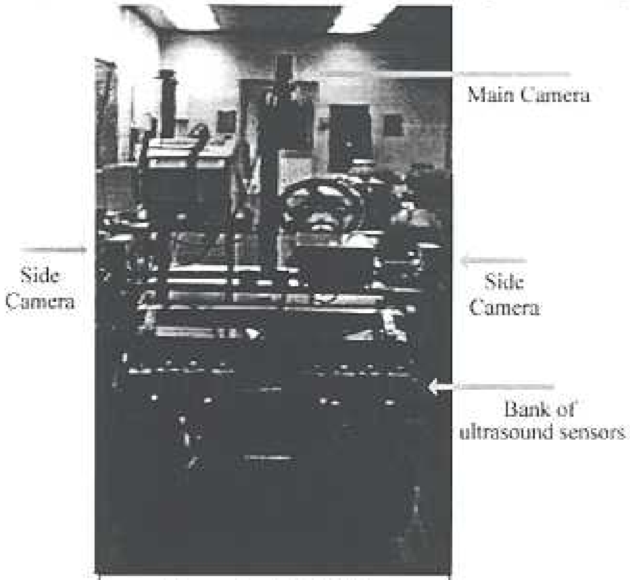


Figure 1: *MOSFET*

MOSFET's intelligence consists of three steps:

- First, it extracts information from each individual sensor.
- Next, it accomplishes a systematic fusion of the information extracted from the individual sensors. It augments this fused information by the use of temporal analysis, which keeps track of previous (obstacle and lane) information and updates that information in accordance with the movement of the vehicle.
- Finally, based on it's understanding of the immediate world, it actuates a motion that keeps it within lanes while simultaneously avoiding obstacles.

Shown below are some sample images showing how *MOSFET* navigates autonomously. The insert in the upper right shows what the forward-looking camera views.

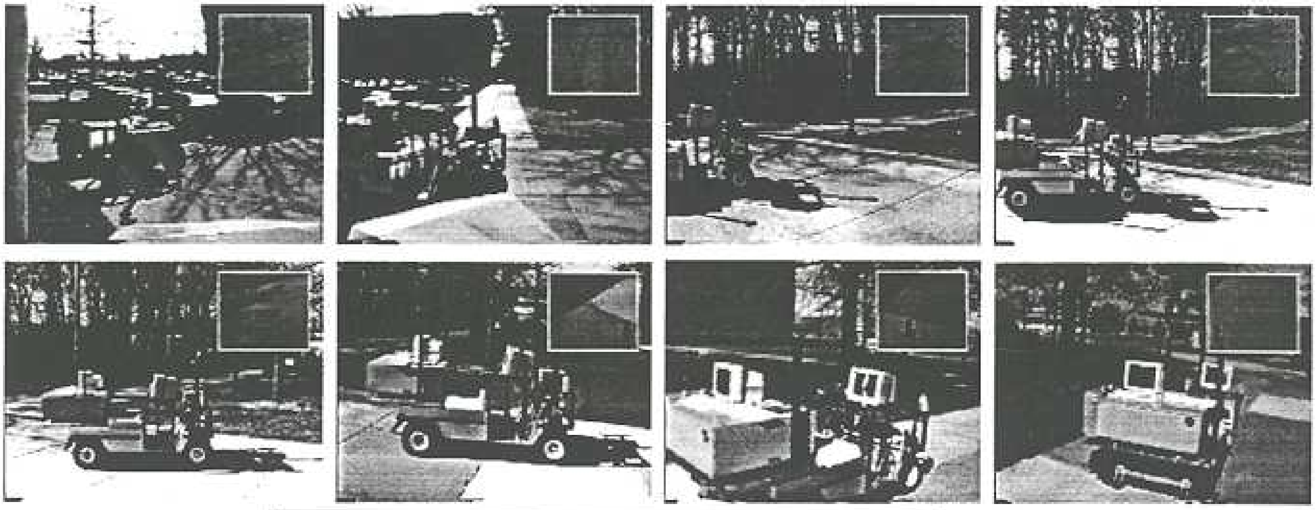


Figure 2: A Sequence of *MOSFET* navigating a curve.

MOSFET's hardware is a novel adaptation of commercial off-the-shelf parts and computers, with much of the innovation involving systems integration. The more significant (and truly original) innovations that went into the design of *MOSFET* are at the algorithmic level, and one of the principal purposes of this report is to shed some light on those enabling algorithms.

Innovative Aspects of Design

Before describing the details of the innovative aspects of *MOSFET*'s design, it is beneficial to lay a proper foundation for their need. Recall that the ultimate design goal is to produce a vehicle that will autonomously navigate a previously unknown course (specified by lane markings). The course is expected to contain a variety of surface types, lane markings, and obstacles. Successful navigation includes avoiding obstacles while staying within the lane markings. To accomplish this goal, we designed *MOSFET*.

MOSFET's base vehicle is a *Club Car Carryall* golfcart. It is heavy enough to provide adequate traction on a variety of surfaces, but at the same time it is small enough to maneuver around obstacles while still staying inside the course. This golfcart was originally modified in 1993, and it entered the competition that year under the name *MAVERIC*. The salient features of *MAVERIC* were:

- It relied on gathering information about the world from just a forward-looking camera.
- The computations were all carried out aboard a Sun IPX workstation and a 286-based PC.
- All the motor controls were accomplished mechanically, using pulleys, chains, and links.

This resulted in a vehicle that was reasonably suited for the purpose at hand but was observed to be quite sluggish and unreliable. *MOSFET* is a completely overhauled version of *MAVERIC*:

- The onboard computers were replaced with a Pentium PC (the master) and a 486 PC (the slave).
- The velocity controller was modified so that the main motor is controlled electronically.
- A gear system was designed to accomplish quick and accurate steering.
- Two side-looking cameras were added to augment the (lane) information from the main camera.
- A bank of 8 ultrasonic sensors was added to aid in detection of physical obstacles in front of the vehicle.
- A completely new set of algorithms was designed to enable *MOSFET*'s navigation capabilities.
- A cooling system was designed to aid the transfer of heat generated by the onboard electronics to the outside.
- The E-STOP system was enhanced. It now disables the motor completely until a reset button is pushed.
- All these different elements/systems were integrated to result in a reliable and responsive autonomous vehicle whose performance meets the goal at hand.

In the following subsections we provide more details on the design innovations that helped accomplish the above.

A. Electronics

As mentioned previously, *MOSFET*'s base vehicle is an electric powered golf cart. The battery pack consists of six series-connected six-volt batteries, providing a total of thirty-six volts DC. The batteries provide all power to the vehicle during normal operation. The electric propulsion system is a series-wound DC motor. Velocity control was initially (in *MAVERIC*) provided by switching between six resistive loads to change the amount of current provided to the motor. This system, however, did not provide adequate velocity control. It was replaced by a PMC transistorized motor controller, providing a continuous range of velocities. It is controlled during manual operation by a potentiometer connected to the accelerator pedal. The direction of the vehicle is controlled by reversing the polarity of the voltage to the motor. This is done via a series of four solenoids, which switch the power lines leading to the motor field winding. In manual mode, the solenoids are controlled with a manual lever in the passenger area. To obtain onboard AC and standard DC power from the batteries, power converters are used to convert the 36-volt battery line into 5, 12, 24, and 120 VAC lines.

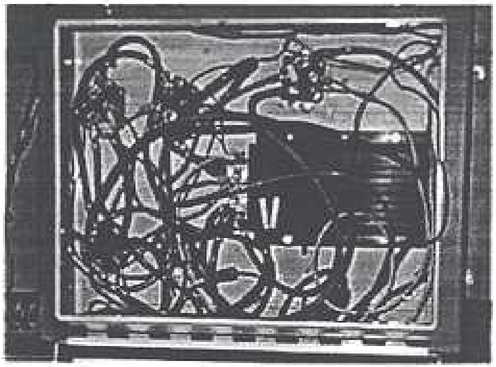


Figure 3: PMC Motor Controller

B. Actuators

The navigation decisions made are actuated by two Omnitech Robotics MC1000 motion controller cards — one for steering and another for velocity. These cards receive control commands from the slave 486 PC and drive the rest of the actuation system. The steering motor signal is amplified via a PWM amplifier. This PWM amplifier drives a 24-volt servo motor, which then turns the gears. Part of the effort involved in actuating the steering system was the design and implementation of the gearing system. The velocity signal is an analog 0—5 volt signal, which is used to control a high-current power amplifier. The power amplifier provides voltage to the motor. The velocity card issues a separate sign output indicating forward or reverse. This logic level signal is amplified to switch the solenoids that control the drive lines to the motor. This was added so that the motor would have reverse drive capability.

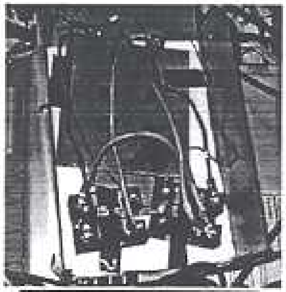


Figure 4: Electronic Switch

C. Software

There are several elements that go into the overall software[1] responsible for the autonomous navigation capability of *MOSFET*. Shown in Figure 5 is an organization chart of the various software elements in terms of the various roles they play:

---

[1] The higher-level code was written in Borland C++ 5.01, and the lower-level code was written in Borland Turbo C.

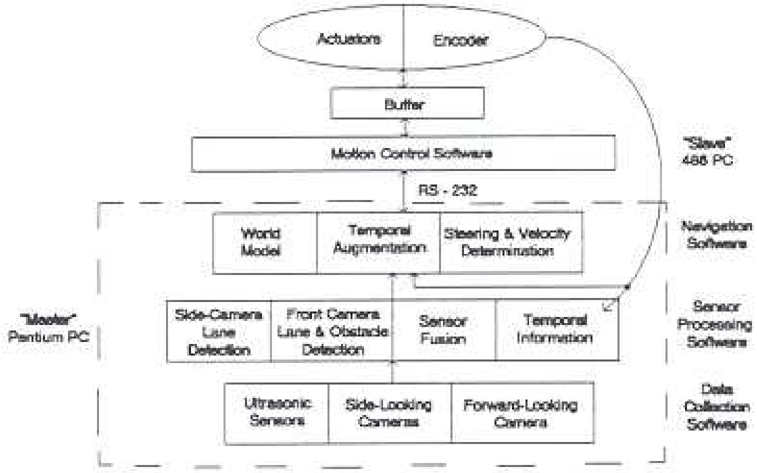


Figure 5: Software Organization

D. Sensors

MOSFET relies on three sensor systems. Data is collected from each system, processed, and fused together to form a coherent map of the surrounding area. The three sensor systems are a forward-looking camera, two side-looking cameras, and a bank of ultrasonic sensors:

- The main (forward-looking) camera is a standard color video camcorder. Its role is to provide information about lanes and obstacles in front of *MOSFET*. It is positioned approximately 1.8 meters off the ground and angled in such a way that the field of view is from 1.5 meters to 6 meters in front of the vehicle. These parameters were chosen after consideration of the tradeoffs between minimizing the 'dead zone' near the vehicle while maximizing viewing distance, while simultaneously operating within the constraints of the camcorder's imaging system.

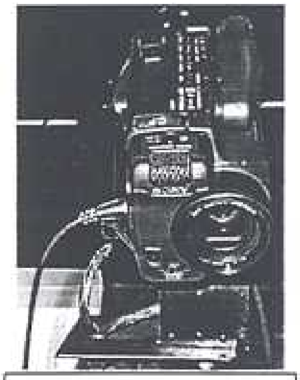


Figure 6: Main Camera

- The two side looking cameras are standard color cameras. They are used to augment the information obtained from the main camera. The side cameras are used primarily to locate lane markers immediately to either side of the vehicle. They are mounted one meter off the ground, and view an area that extends from the immediate edge of the vehicle to approximately 0.75 meters away from the side. This viewing area compensates for the dead zone of the main camera close to the vehicle. Some filters were added to the side cameras to alleviate the problem of glare experienced on bright days. The main and side camera outputs are multiplexed into a video digitizer, that grabs a set of three (still) images from them every 0.6 seconds.

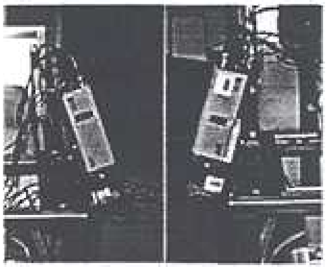


Figure 7: Side cameras

- The ultrasonic sensor system consists of a bank of 8 individual ultrasonic sensors. This system is used to detect physical objects at distances of 0.5 to 10 meters from the vehicle. The sensors are arranged to maximize overlap and detection area. The system is controlled by an HC11 microprocessor, which initiates the firing of each sensor and collects and transmits the sensor data to the onboard Pentium PC. The eight sensors fire sequentially, with each waiting for the previous sensor to finish before firing. This eliminates the problem of crosstalk, where the signal from one sensor interferes with another sensor. The system is set to fire all 8 sensors and transmit data every 0.5 seconds, making transferring data to the onboard computer easy.

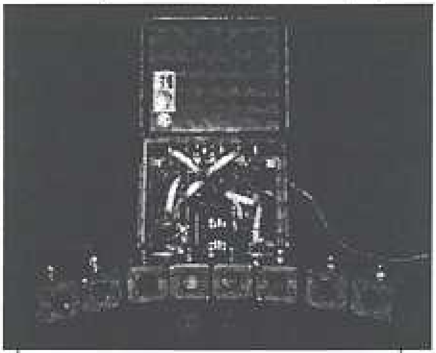


Figure 8: Ultrasonic units

E. Computers

The master onboard computer is a standard off-the-shelf Pentium 133MHz. A PC was chosen as the main processor due to availability of low-cost software and hardware, especially for image processing. This computer acquires data, from the various sensors, processes the data and makes navigation decisions. A *Matrox Meteor* image-processing card was added to this PC to aid in the data acquisition.

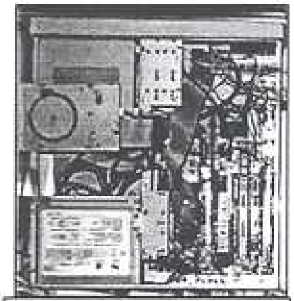


Figure 9: P133 Master Computer

A standard off-the-shelf 486 66MHz slave PC is connected via RS-232 serial communications to the master. This slave is responsible for controlling the actuation of the steering and velocity motors in order to realize the navigation decisions made by the master. It is also responsible for the vehicle's speed control system, which is designed to maintain constant speed regardless of the inclination.

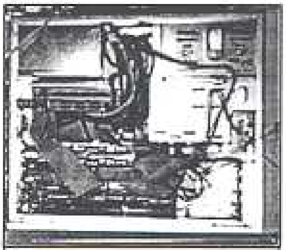


Figure 10: 486 Slave Computer

F. System Integration

Since much of the onboard electronics is off-the-shelf, except for the forward/reverse switching mechanism, systems integration is a crucial part of *MOSFET*'s design. In broad terms:

- Each of the components of the overall systems were designed and tested individually.
- Once the performance of the individual components was deemed "acceptable", they were integrated and tested for overall functionality.

More specifically, first, once the initial design of electronics and actuators was completed, the design was tested by attempting to manually navigate the vehicle. Next, the vision sensors were tested as a group using the image processing card, serial communications, and software written explicitly for the testing procedure. (The ultrasonic sensors from the *MAVERIC* vehicle were already nominally functional.) Third, a computer was purchased to meet the specific requirements of the imaging system. Upon accomplishing the above tasks, a collection of software algorithms was designed for the various categories outlined above in Figure 5. Each of these algorithms was written and tested individually and redesigned until acceptable performance was achieved. Once this was accomplished, the entire collection of software was integrated along with the hardware into a single functional package. The integration was done in a step-by-step procedure. First, the main camera routines alone were used along with the navigation routines to attempt to autonomously navigate *MOSFET* around lanes and obstacles easily distinguishable by color. Second, the sonar information was incorporated to allow more reliable obstacle detection. Third, the lane detection information from the side cameras was added to aid in better lane detection. Finally, vehicle motion information from the control card encoders was feedback to provide temporal correlation and sense of "memory" for *MOSFET*.

Ready Made Components

As mentioned previously, many of the electronic components were purchased ready-made to accommodate the particular needs of *MOSFET*. Several components associated with the vision system came complete. The various color cameras, the image-processing card to transfer the images to the computers, and the video brick that functions to synchronize the various cameras were all off-the-shelf components. The ultrasonic units were provided as functioning units, although it was necessary to design the software that interpreted the output of the units to provide range data. The amplifiers used to drive both the steering and velocities motors were used without modification. The processing power aboard the vehicle, namely the master and slave PCs, was also off-the-shelf, as were the motion control cards. The steering gears were purchased from just a standard catalog, as well as the cooling fans. Finally, the base vehicle itself is a commercially available golf cart.

The Design Process

Perhaps the most innovative component of the design effort put forth in designing *MOSFET* involves software development, which includes the writing, testing, and (re-)evaluation of crucial algorithms that enable the vehicle to navigate as intended. We mean the algorithms responsible for sensor data acquisition, sensor data processing, sensor fusion, and navigation decisions.

Recall that the left and right cameras are used to determine the offset and orientation of the lane markings with respect to the vehicle. This is accomplished by using a deformable template procedure. In broad terms, a template of the ideal lane marker is constructed, which is then deformed to match the observed (line) intensity profile. The matching criteria involve a novel modification of the customary signal-to-noise ratio (SNR) based matched filtering criteria. A Karhunen-Loeve type color transformation is used for transforming the RGB color channels of the given image on to a composite color channel, in order to eliminate some of the image imperfections/noise such as shadows, specular reflections, etc. Separate transformations are used for different scenarios, so that the algorithm will work on either grass or pavement, with yellow or white lanes. One line of the image is processed to find the location of the lane marker in image coordinates. Processing only one line saves both time and complexity, as the problem is reduced from two dimensions to one dimension, and the number of pixels to be processed is reduced by a factor of 240. The standard perspective transformation is used for transforming the offset information from image coordinates onto ground coordinates. The distance from the edge of the vehicle to the lane marker is reported in meters to the main navigation algorithm.[2] In some cases, the lane marker is found at three places in the image by processing three lines. Determining distance at multiple points allows for lane orientation as well as distance to be determined. This redundancy also improves the robustness of the lane-finding algorithm. Sample output of the lane-finding algorithm is shown above in Figure 11.

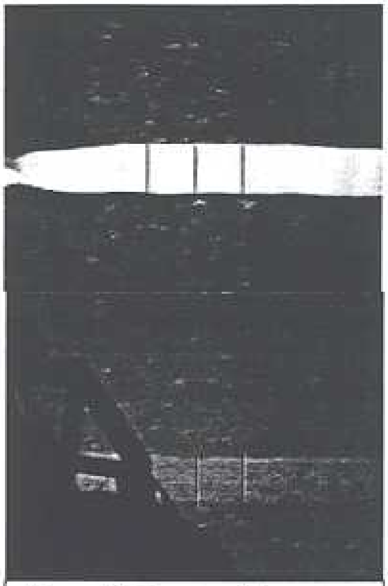


Figure 11: Lane tracking output

Recall that the sonar sensors are used to detect physical obstacles in front of the vehicle. The sonar sensor algorithm uses what's called the trivial sonar model. Under this model each sensor's output is interpreted as a return coming from an obstacle at a proportionate range (indicating the distance from the sensor to the obstacle), whose azimuthal (angular) position is at the center of the beam cone. The algorithm processes the output of all eight sonar units and results in a range-azimuth map of all physical obstacle locations with respect to the vehicle.

Recall that the forward-looking camera is what is called the main sensor, since it contains the most comprehensive information regarding the location of lanes and obstacles relative to the vehicle. Every 0.6 seconds, an image is grabbed from the main camera and segmented by color into two categories: lane/obstacle and non-lane/obstacle. The color segmentation algorithm proceeds in two phases: Phase 1 is the training phase — from a "typical" image, patches are first manually identified corresponding to potential obstacles, the lanes, and background. The Red (R), Green (G), and Blue (B) channels of the typical image are then linearly combined so that in the resulting composite color image the "difference", in a (Fisher-type) information discriminant sense between the lanes/obstacles and the background is maximized. The maximization problem is solved using standard results from linear algebra. Phase 2 is the actual segmentation phase — the color transform that results from phase 1 used to combine the R, G and B channels for all subsequent images from main camera and the segmentation of

---

[2] Details of the lane-finding algorithm for the side camera can be found in the publication, "STARLITE: A Steering Autonomous Robot's Lane Investigation and Tracking Element," *Mobile Robots XI and Automated Vehicle Control Systems*, Proc. SPIE 2903, 1996.

various color is achieved via a simple histogramming procedure in the composite color domain. Using this methodology, several color transformations have been developed to isolate particular colors of interest[3]:

- A white/green transform, designed to highlight white lanes while ignoring green, with the additional feature that makes it possible for it to ignore other common path colors such as black, brown, and gray.

Figure 12: White / Green Transform

- A red/green transform, used to find exclusively red obstacles again while ignoring green, as it was anticipated that the obstacles in the AUVS ground robotics competition would all be red. This same algorithm could be easily adapted for any other specific color as well.

Figure 13: Red / Green Transform

- A yellow/green transform, designed to highlight yellow lanes while ignoring green.

Figure 14: Yellow / Green Transform

The fusion algorithm combines the color-based lane/obstacle segmentation results from the main camera algorithm and lane offset and orientation results from the side camera algorithm. This combination is accomplished by perspectively projecting those results onto the ground plane and using yet another deformable template procedure. This time the template

---

[3] See "OLSON: A Multi-Color and Multi-Modal Data Fusion Algorithm", under review, 4th *IEEE International Conference on Image Processing*, Santa Barbara, 1997.

corresponds to ideal lane shapes on the ground plane, which are then deformed to match the detected lanes by using a Hough transform-like criterion. Physical obstacles detected by the ultrasound sensor algorithm are overlaid on the same grid, and a correlation measure is computed between the main camera and the ultrasound obstacle detection results. Only those obstacles that have a high degree of correlation between the two modalities (according to this measure) are identified as "real" obstacles.

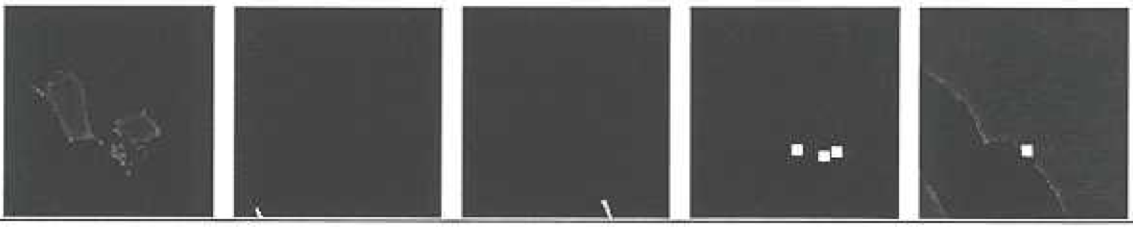


Figure 15: From L to R, The main camera data, left camera, right camera, sonar data, and the final fused image.

MOSFET uses an accumulated grid model in developing the navigation map. The lanes and the obstacles detected are all laid out in a grid of *MOSFET"s* immediate world by the map-building algorithm. Each grid element represents a certain distance and angular position from *MOSFET*. Associated with each element is a value proportional to its "relative obstacleness." The relative obstacleness is measured by a weighted combination of the lane, ultrasound, and color algorithm outputs. Grid elements that are outside of the lanes are deemed "severe" obstacles. After the information extracted from the various heterogeneous sensors is correlated to form a map of the world in front of *MOSFET*, this map is augmented by use of temporal information. This temporal information essentially consists of similar world maps retained from previous iterations. In order to do this, accurate information regarding *MOSFET*"s motion (the distance that the vehicle has moved over the last time increment and the steering angle of the vehicle over that time increment) must be collected. This information is obtained from *MOSFET*'s motion controller. Using those measurements, and assuming those values stayed the same over that small time increment, three motion parameters are calculated. First, the change in the heading angle $\Delta\phi$ is calculated. Once the change in heading angle has been calculated, the translational movement Δx and Δy of the vehicle can be calculated. These three motion parameters allow the world grid map to be updated. First, the translational movement of the vehicle is used to shift each grid in the world map, so that the position of the vehicle is again the center-front of the grid. Next, the world grid map is rotated to take into account the change in the

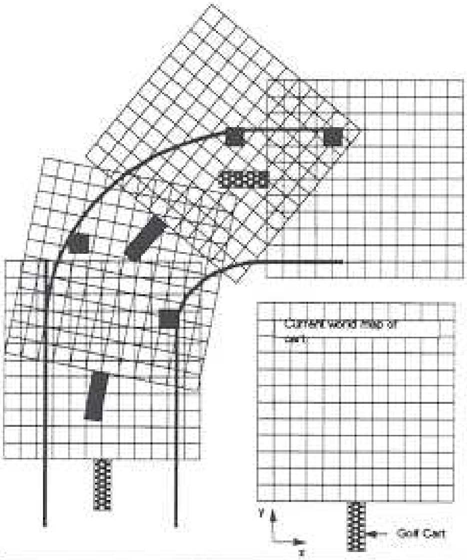


Figure 16: The Accumulated Grid Model

heading of the vehicle. Each grid is rotated using a lookup table, which contains rotation information in 0.25-degree increments. The lookup table is created by considering the polar coordinates of each grid and subtracting the vehicle heading change from angular position of the grid. A decayed sum of all the previous "relative obstacleness" values plus the new one constitutes the accumulated obstacleness for each square grid. A cleaning procedure to get rid of stray/isolated single square obstacles is then performed, and the resulting grid is passed to the navigational algorithm. The use of temporal information gives *MOSFET* a sense of memory, without which it has trouble with lanes/obstacles that are entirely contained in the so-called dead-zones of its FOV, as illustrated in Figure 17.

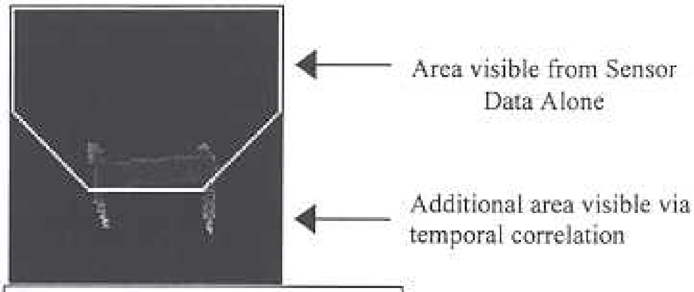


Figure 17: Field of view example

The navigation grid is then used by the navigation algorithm to generate a suitable steering angle for the vehicle. The first step in this procedure is to sweep through all possible future steering angles in two degree increments and determine the amount of obstacles present at each potential future heading. Next, the plot of total angle obstacleness versus steering angle is considered. A sample navigation grid and a sample obstacleness versus steering angle plot are shown in Figure 18.

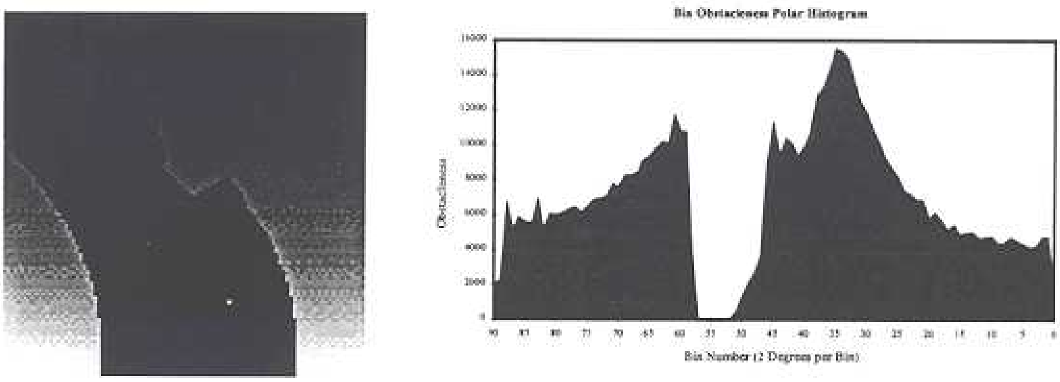


Figure 18: Sample navigation grid and obstacleness plot

Minima on the obstacleness curve that are located in the middle of a valley (i.e. minima surrounded by a series of angles with obstacleness below a dynamically calculated threshold) are located. A valley is necessary to accommodate the size of the vehicle. The minimum width of the valley allowable as well as the minimum and maximum steering angles were designed by studying the ability of the steering and velocity systems. From the set of minima, each of which represents a possible future steering angle, the optimal angle is

Figure 19: Steering valley and angle chosen

chosen. Factors including distance from previous heading, distance from straight, and obstacleness of the angle are all used in determining this optimal angle. The image in Figure 19 shows the chosen angle graphically imposed on the worldmap.

Once the optimal steering angle has been chosen, the desired angle is sent to the slave computer via serial communications. Software on the slave computer receives the desired steering angle, decodes the angle into a signal suitable for transmission to the motion controller cards, and transmits the desired angle. The slave computer performs another critical task. First, it is responsible for attempting to maintain a constant velocity regardless of terrain. A software controlled feedback system computes the velocity of the vehicle by looking at a one-second position delta. If this speed exceeds a certain critical velocity, the commanded velocity is decremented. Likewise, if the velocity is less than the lowest allowed velocity, the commanded velocity is incremented. The timing involved in calculating the velocity and the frequency at which the commanded velocity is incremented and decremented were designed to maximize the stability of the system, while minimizing oscillations inherent in this feedback configuration.

Safety, Reliability, and Durability

The vehicle chosen to be the testbed for this project was the *ClubCar Carryall* golfcart. This vehicle, by its size, carrying capacity, etc., is meant to simulate an actual road vehicle — carrying up to two passengers comfortably, while navigating autonomously. It can operate both on- and off-road, and can navigate relatively steep slopes without any problems. The design overhaul of the original *MAVERIC* vehicle that resulted in our current vehicle *MOSFET* was undertaken with specific view toward issues of safety, reliability, and durability.

To ensure safe operation of the vehicle's power system at all times, a number of switches are used to control the availability of battery power. Separate custom switches are used to enable AC power, DC power, and power to the motor. The standard golf cart key switch will also disable the motor. As a final measure of safety, the emergency stop system will disable the motor and apply the brake. The emergency stop is activated when either an on-board button is pressed or a remote transmitter is used. It remains in effect until an on-board reset button is pressed, to ensure that the vehicle remains in a safe state until any problem can be corrected.

Reliability is achieved via modular design. Our design and systems integration strategy provides a firm handle when it comes to assessing and improving the reliability of *MOSFET*. We state an example here to underscore this point: When we first designed the color-based lane/obstacle segmentation algorithm, we had implemented a fixed-threshold binarization scheme. This threshold was picked based on a collection of sample images of outdoor scenes that we had collected. When *MOSFET* tried to autonomously navigate based on grid maps obtained by using this fixed threshold algorithm, its performance was very unreliable. At that point we were faced with the task of trying to find the exact cause of this lack of reliability. A systematic examination of *MOSFET*'s grid maps immediately narrowed the problem down to the color segmentation algorithm — it turned out that when it was overcast or if for some other reason the lighting was poor, the entire world in front of *MOSFET* was segmented as "lane/obstacle." To fix this (reliability) problem, we designed an adaptive thresholding scheme for binarization. The resulting lane/obstacle segmentation algorithm performed admirably under a wide variety of lighting conditions.

Durability is perhaps the most serious flaw in *MOSFET*. Many of the off-the-shelf components are not designed to handle a rough testing environment. Problems such as bouncing, extreme temperatures, and outdoor environments are stressful on electronic components. In particular, the computer systems and the optical encoders are vulnerable to a poor operating environment. However, *MOSFET* is a test-bed and not a commercial vehicle, so some extra maintenance is acceptable.

Problems Encountered

A. Hardware

Many hardware difficulties were experienced and overcome in the process of designing *MOSFET*. For instance, devising the subsystem necessary to drive the motor in autonomous mode presented a challenge. Similarly, the electronics and gearing sub-systems that are utilized to actuate the steering wheel from the slave computer presented a formidable task in terms of hardware construction and interface with the computer. The forward/reverse switching mechanism outlined in the Electronics section, necessary to allow the vehicle to move in both directions as well as to allow the vehicle to slow down quickly, was another challenge. Finally, the enhancements made to the emergency stop sub-system design involved a significant amount of effort.

B. Software

One of the major software challenges involved the image processing aspect of the obstacle and lane detection algorithms. Although it was evident from the beginning that color and intensity provided the strongest visual cue that could be exploited by our image processing algorithms, it was found that there was a significant variation in the background shade and colors as well as a large diversity in obstacle type, shade, shape, and color. The apparent color of the grass, for example, varied widely at different times of the year, different times of the day, and when under different lighting conditions. Therefore, the transformation, histogramming, and segmentation procedures described earlier were all designed to minimize the effect of these variations as much as possible. Furthermore, on particularly bright and sunny days, glare from the background often resulted in erroneous readings from the side-looking cameras. This problem was alleviated through image processing techniques as well as through the addition of anti-glare filters to the cameras. Shadows cast onto the image also presented difficulties in segmenting images based on color. Depending on the orientation of the sun with respect to the vehicle, shadows can exist in the main camera plane or either of the side camera's field of view. Once again, this problem was remedied via software routines. The main camera color transforms, for example, were written so that black (shadows) would transform to a path. This was allowable for two reasons. First, problematic shadows are typically those that present themselves immediately in front of the vehicle, which is always the path. Second, those shadows that cover lane markings will only cover those markings that are very near the vehicle. The lane markings near the vehicle are precisely those that are located using the side looking cameras, so no information is lost.

Other software problems presented themselves in the navigation subsystem. Several steps were taken after the initial testing stages to minimize some of the problems experienced. For example, obstacles far away from the vehicle are weighted less heavily in the navigation decision process to eliminate distant obstacles from unfairly biasing the vehicle. Large deviations from the current steering angle are also not allowed, which helps to overcome the problem of anomalous or erroneous sensor readings that are sometimes experienced. The accuracy and field of view of the various sensors is taken into account so that the routine does not steer toward certain regions that are clear merely due to the lack of sensor data.

Predicted Performance

The *MOSFET*'s performance is predicted below, based on extensive tests performed during the design process:

- On paths easily separable by color from the off-path, such as traversing a sidewalk surrounded by grass, *MOSFET* will very reliably maintain its course.
- Under such separability conditions, *MOSFET* will stay on the path even in the presence of path discolorations, moisture on the ground, and path inclinations.
- In the absence of sharp curvatures the vehicle can be expected to locate and navigate successfully between the lanes. *MOSFET* will have problems, however, over sharply curved portions of the course.
- *MOSFET* will have problems with back-to-back curves, such an S-curves, even if the component curves are not very sharp.
- *MOSFET* suffers from a "ringing" (damping) problem whenever it is forced to make a sharp turn. After making a sharp turn, such as to avoid an obstacle or navigate around a sharp curve, when it tries to steer back to normalcy it tends to overshoot/overestimate the desired steering angle, and as a result it will exhibit an oscillatory behavior.
- When physical obstacles are present on the course, *MOSFET* will be able to detect their location and steer away while maintaining the lane.

Cost Analysis

The following tables represent a breakdown of the material costs associated with the construction of *MOSFET*, and a time estimate of the work necessary to complete the design.

Component	*Cost*
Golf Cart	$2,500
Electronics	
Transistorized Motor Controller	$1,200
PWM Servo Amplifiers	$200
Power Converters (DC/DC)	$600
Inverter	$1,800
Computers	
Pentium PC (with accessories)	$2,000
486-66 Slave Computer	$750
Imaging Hardware (Digitizer)	$1,000
Motion Controller Cards	$150
Hardware	
Steering Motor & Optical Encoder	$600
3 Color Cameras	$1,200
Sonar System (with microcontroller)	$1,000
Vehicle Isolators	$375
Miscellaneous	$1,000
Total	**$14,375**

Category	*Man-Hours*
Software Design	
Initial SW feasibility study	440
Obstacle Detection	700
Lane Finding	625
Sensor Fusion	900
Navigation	1265
Hardware Design	
Initial HW work	325
Modifications, Redesign & Maintenance	240
Testing	475
Total	4,970

Team Organization

The *MOSFET* team was organized as follows:

- Mike Beauvais: Graduate Student, University of Michigan-Dearborn (UM-D), Electrical and Computer Engineering (ECE) Department (*Team Leader*). Responsibilities included: Sensor fusion, temporal analysis, gear design, electronics, motion control interface, and systems integration.
- Randy DeFauw: Senior, UM-D, ECE Department. Responsibilities included: Lane sensing, gear design, electronics, motion control interface, cooling system, systems integration, and marketing.
- Chris Kreucher: Graduate Student, UM-D, ECE Department. Responsibilities included: Color segmentation, navigation sub-system, user interface design, and systems integration.
- Professor Sridhar Lakshmanan: UM-D, ECE Department (*Faculty Advisor*).
- Other student contributors: Elaine Linas and Kevin Hersh, Seniors, UM-AA, EECS Department (color segmentation and navigation sub-system); Craig Cornwall, Senior, UM-D, ECE Department (navigation sub-system and computer communications); and Mike Moore, Junior, UM-D, ECE Department (wiring).
- Additional Help From: Professors Nattu Narasimhamurthi, John Miller, and Van Godbold, UM-D, ECE Department (color transformation, image processing board, sensor sub-system, image filtering, electronics). Professor Yi Zhang, UM-D, ME Department (gear design).