UMD-SMART: Un-Manned Differentially Steered Multi-purpose Autonomous Robust Transporter

And

GCAT: GPS–enabled Conventional-steered Autonomous Transporter

V. Varghese, S. Makam, M. Cinpinski, E.Mordovanaki, A. Mordovanaki,

B.Y. Sun, D. Wrosch, O. Adabonyan

UMD-SMART

GCAT

Department of Electrical and Computer Engineering

University of Michigan-Dearborn

May 16, 2003

***UMD-SMART*: Un-Manned Differentially Steered Multi-purpose Autonomous Robust Transporter**

***GCAT*: GPS–enabled Conventional-steered Autonomous Transporter**

Introduction

UMD-SMART (Un-Manned Differentially Steered Multi-purpose Autonomous Robust Transporter) and *GCAT* (GPS-enabled Conventional-steered Autonomous Transporter) are autonomous vehicles that were designed to participate in the 11th Annual IGVC of 2003 in the following events: the *Autonomous Challenge*, the *Navigation Challenge* and *Follow-The-Leader*. In the *Autonomous Challenge*, the vehicle has to complete a course (marked by two lane markers) and avoid all obstacles such as construction barrels and potholes (simulated using white paint). In this part of the competition, the vehicle must be able to travel on different types of terrain, such as grass, sand and wooden ramps. In *the Navigation Challenge* the vehicle is expected to autonomously visit waypoints specified by their latitude and longitude and avoid any obstacles along the way. In *Follow-The-Leader*, it has to follow a lead vehicle while maintaining a constant 3-meter headway.

The design team consists of students from the department of Electrical and Computer Engineering at the University of Michigan-Dearborn. Most of the members are receiving design credit as part of their senior design course.

The requirement given to the team was *to design two fundamentally different vehicles, (*one using a differential torque to steer the vehicle and the other using a conventional-steering mechanism*) using common design procedures and to use interchangeable modules* as much as possible. Hence, this report covers *both* vehicles. The team was formed in January of 2003 and as time was a critical factor, the team decided to purchase the platforms and modify them to as needed. The academic advisor also assisted the team by providing a C++ API for performing real time video processing using a standard analog camera and a capture device such as the dazzle DV80. As part of the API, the team was also provided with a module that would determine the location of the lanes and provide a parametric representation of them. The team decided to use the lane detection module as a black box. This report describes the steps taken, the testing performed and the capabilities of the resulting systems.

Team Organization

Each member of this design team was assigned both a primary and secondary task to complete on the vehicle. However, the team frequently worked together when serious problems were encountered. As a result, the entire team, rather than just a certain individual made all critical design decisions. The table that follows indicates each member's primary and secondary responsibilities, their academic department, and their class level. The total number of hours invested into this project is indicated below:

Name	**Responsibilities**	**Class, Department**
M. Cinpinski	Image Processing *GCAT* Steering Control	Senior, ECE
S. Makam	*UMD-SMART* Steering Control GPS Software	Senior, ECE
B.Y. Sun	Ultra sound Sensors	Senior, ECE
V. Varghese	*UMD-SMART* Steering Control Electronics GPS System Integration Software Sensor design and integration	Senior, ECE
E. Mordovanaki	Mechanical construction	Senior, ECE
A. Mordovanaki	Image processing	Senior, ECE
O. Adabonyan	Harness design and construction HC11 programming *GCAT* Steering Control Wireless switch interface	Senior, ECE
D. Wrosch	*UMD-SMART* Steering control Magnetic compass	Graduate student, ECE

Total number of hours: 2000 hours

Team leaders shown in bold face.

Design Process

The design and construction of the vehicles took place as an incremental process. After each step, the design was tested to ensure that the system performed as required. Often the team had to develop new techniques, such as an interface to a joystick, Internet based control etc., to test each module. The following describes the steps taken to complete the design:

1. **Starting point for the projects**

Chassis: For *UMD-SMART*, an Invacare *Storm 2000* electric wheel chair with differential steering capability was acquired. For *GCAT*, a Manco 285C go-kart chassis with a conventional steering mechanism was acquired.

Vision system: A SONY video camera was interfaced to a PC using a dazzle DV 80 to capture and process images. Lane detection software from FAST (UMD's vehicle, which participated in the 10th annual IGVC) was used to detect lanes in the obstacle course. The software provides a C++ interface, which permits real time processing of the video images.

Processors: The team chose the Motorola 68HC11 microprocessor to control the vehicle, based on the team members' familiarity with it. The BS2x (stamp) processor was chosen as a secondary processor to interface with the range sensors.

2. **Understanding the vehicle**

UMD-SMART: To understand the power requirements of the vehicle and to determine the strategy to control the vehicle, the joystick that came with the wheel chair was replaced with electronic circuits that would simulate the movement of the joystick. An interface to the 68HC11 was used to control the electronics. The team also developed interfaces to an electronic compass. As an initial test, the team developed control algorithms that would turn the wheelchair and move it along a specified heading. During this test, the current drawn by the vehicle was measured and the vehicles ability to climb the incline that would be required in the competition was verified. At this stage, analog circuits performed all the control.

GCAT: The bare chassis was loaded with about 40 Kg, and was pulled up a ramp using a spring balance. The team found that a maximum of 50 lbf (225 Newtons) was needed. Assuming a maximum speed of 4 Km/h (~1 meter/second) on a ramp, it was estimated that 225 watts of power would be required. A ¾HP DC motor was chosen for this task and eventually the team purchased a 1 HP motor to aid in future expansion. The Go Kart uses a 420 size sprocket with 55 teeth. Based on the speed requirements, a motor with a nominal speed of 1800 RPM and 5 teeth on the driving sprocket was needed. A sprocket with 9 teeth was used to give additional head room at high speeds. A windshield wiper motor that was coupled to the steering rod by a chain drive replaced the steering wheel on the Go Kart.

3. Special Sensors and electronics

UMD-SMART did not require any specialized sensors. For *GCAT*, a 5K potentiometer was used to determine the angular position of the steering rod. The team designed a circuit that can convert the resistance to a voltage within the range accepted by the 68HC11. In addition, limit switches were added to prevent the motor from trying to turn the wheels beyond their physical limits.

4. Designing the control circuit

UMD-SMART: The team first designed and implemented a power electronics circuit that enables the microprocessor to control the voltage applied to each wheel. As can be seen from the accompanying figure, the circuit has an optical isolator to protect the control processor from high power motor drive circuits as well as comparators to increase

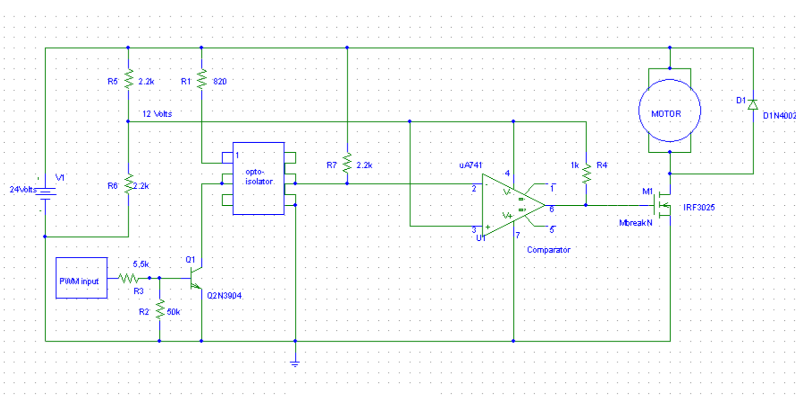

Figure 1: Power electronics circuit to interface to a micro

the noise immunity. Components were selected to handle up to 30 amps, even though the wheel chair motors typically draw 3 to 5 amps under normal circumstances and about 10 amps when climbing a ramp. This provided a sufficient safety margin.

The electronic circuits were tested manually by entering the desired turning value in the microprocessor and verifying that the vehicle turned as expected. Manually entering the desired turning angle in the PC and then transmitting this information to the microprocessor using the parallel port tested communication between the main PC and the microprocessor. Once it was determined that the electronics and the control algorithms functioned correctly, a joystick that could be connected to an off-vehicle computer was acquired. Software was designed to enable the computer to read the joystick and transmit the information (using the Internet) to the computer on the vehicle. At this stage, one could remotely control the direction of the vehicle and the team was able to calibrate the steering system of the vehicle and test the ability to control the vehicle's speed.

GCAT: For speed control, the team chose a CURTIS 24 V motor controller. This controller was available from the department and as a result allowed the team to avoid the high cost of such controllers. A resistor was used to control the speed, and other circuits were added to limit the speed of the vehicle. The team added a safety (dead-man) switch that has to be continuously pressed in order to move the vehicle. Provisions were made to disable this feature during competition.

5. Emergency Stop and safety circuits

To ensure safety, a manual emergency stop button was added to stop the vehicle in an emergency. Additional fly-back diodes were added to all the motor circuits to avoid voltage spikes when power is interrupted. Adequate fuses were added in series to all power circuits.

6. Speed Control

When it became possible to run the vehicle at a constant speed and control the turning angle, additional features were added to the control algorithms and the electronics. First, a sensor circuit was designed to monitor the on/off switch to the motor. The control algorithm was modified so that the microprocessor would wait for the switch to be turned on and then gradually increase the speed to the desired value, thus eliminating sudden fast movements when starting. The team also developed a speed sensor using an RC car that would send two signals, one to increase the speed, and the other to decrease the speed. The control algorithm was modified to respond to these signals. The GPS unit that provides more accurate speed information would replace this speed sensor. However, the team plans to retain the sensor as a backup.

7. Obstacle Detection

Once it became possible to run the car with the speed sensor, an ultrasonic sensor (SRF04) was added to detect obstacles. The sensor was interfaced to the BS2x (stamp) microprocessor, which would send two signals to 68HC11, one when obstacles are too close (currently 60 centimeters), and one when they are farther away (currently 150 centimeters). The control algorithm was modified to stop the vehicle if the obstacle was too close, or else take evasive action for obstacles that were farther away. Once testing was done using one sensor, two more sensors were added to increase the field of view to 180 degrees. A circuit that would turn on a red warning light whenever an obstacle was detected was also added. The brightness of the light would determine how close the obstacle was to the vehicle.

8. Lane following

A camera was interfaced to the computer using the dazzle DV80. A wide-angle lens was needed to detect the lane markers that were far away from the vehicle. The vision system was calibrated so that the vehicle could be controlled to go to the center of the lane at a specified distance in front of it. In the accompanying figure, the center dotted line shows the center of the lane and the horizontal line is at 6 meters in front of the vehicle. The control algorithm was calibrated to determine the turning angle required to make the vehicle go to the point where the center of the lane crosses this line. The ability to track straight lanes was tested extensively with random initial orientation and it was determined that the

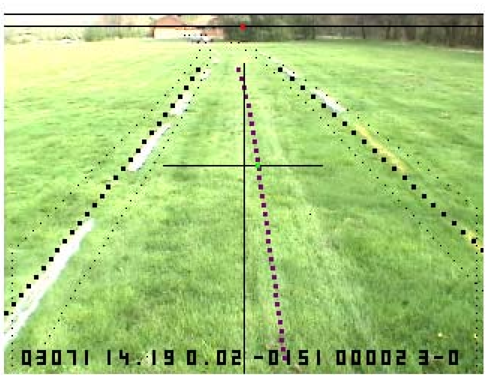

Figure 2: Lane detection

vehicle would be partitioned to be less than 50 centimeters from the center. It was also verified that the lane-tracking program could track both solid and dashed lines, as well as track white and yellow lanes.

9. <u>Follow the leader</u>

The color information in the image was used for the *follow the leader* competition. The program looks for a specific color. During testing, the team chose red as the training color, but this can be easily changed. The choice of the appropriate color to use will be based on the circumstances on the day of the competition. The program determines the range of rows and columns occupied by the object that is trying to be tracked. Visually, this is displayed on the screen as follows: a blue band on the left indicates the rows and a yellow band at the bottom indicates the columns that lack the particular color that the camera is looking for. Based on this, a target point is determined and shown as a red square on the screen. Figure 3 illustrates a typical example obtained during testing. The turning angle and the change in the speed of the vehicle are determined based on the location of the target. In addition to the vision system, ultrasound range sensors will be utilized to maintain a constant distance from the lead vehicle.

Figure 3: Tracking a red object for follow the leader

10. <u>Obstacle detection and avoidance using vision system</u>

Obstacle detection and avoidance uses the same strategy as follow the leader, except a particular color is not specified and the control is programmed to move away from the obstacle instead of towards it. Instead of looking for a specific color, the program selects a region of interest in front of the vehicle and determines the

Figure 4: Obstacle shown in black color

statistical distribution of the colors (red, green and blue) in the region. Any point in the image whose color is large (two standard deviations above the mean) is considered a potential obstacle. If the region in front of the vehicle is obstacle free, it is easy to compute the number of potential obstacle points seen by the camera. If the camera detects more than the expected value (at 90 percent significance) than an obstacle has been detected. This strategy is still evolving.

11. GPS

The team initially decided to compare a Win-Oncore GPS evaluation kit and a Garmin-Legend GPS unit before deciding on which one to use. The choice was narrowed down to these two units as they were available to the team and also it was found that these two units were comparable to other relatively inexpensive units that are available. The Garmin GPS unit was eventually chosen because it could more reliably track satellites. The GPS receiver was interfaced to a serial port (RS232) of the computer. A serial interface program, using the C++ language, was developed to receive and process the data coming from the unit. An approximation to the Kalman filter was developed to increase the accuracy of the data. Also, a history of past velocities was used to accurately determine the heading of the vehicle.

Initially, one point was chosen as a target. The vehicle was tested to see if it would repeatedly come back to the same point. During this test, it was noticed that the update rate from the GPS unit (once a second) was too slow when the vehicle is close to the target. As a result it was decided to decrease the speed of the vehicle when it got closer to the target. By slowing down the vehicle and filtering the data, the vehicle was able to come within 20 centimeters of the target with very high probability (less than two failures in twenty-five attempts). Making four attempts at a target by moving away, turning around and making another pass increased the probability of success. Theoretically, the failure was estimated to be less than 10^{-3}.

The ability of the vehicles to go from one target to the next and eventually return to the starting point was also estimated. For this, three points were selected with one of them being the starting point and the other two acting as waypoints.

System Description

	UMD-SMART	*G-CAT*
Chassis	Modified Invacare wheel chair	Modified Manco Go-Kart
Drive	Original Equipment	DAYTON 6ML06, 1 HP 12 volt PM DC motor (chain drive)
Steering	Differential torque	Steering wheel coupled to a 1/3 HP PM DC motor (chain drive)
Electrical control: Steering	Specialized circuit with 30 Amps per wheel, controlled by a PWM signal	50 Amp H-bridge
Electrical control: Drive	Same as steering	Curtis 1205 motor speed controller (24 V/ 400 A)
Vision	Sony handycam + Dazzle DV 80	Same as *UMD-SMART*
GPS	Garmin Etrex	Same as *UMD-SMART*
Range Sensor	Three SRF04 and one SRF08 ultrasound range sensor	Same as *UMD-SMART*
Digital Compass	Vector 2X (1 degree resolution, 2 degree accuracy)	Same as *UMD-SMART*
Main Processor	DELL Pentium 4, 2.66 GHz processor speed with 512MB of memory	HP Pentium 4, 2.3 GHz processor speed with 512MB of memory
Control processor	Motorola 68HC11	Same as *UMD-SMART*
Peripheral processor	BS2X basic stamp to interface to range sensors	Same as *UMD-SMART*
Drive Batteries	Two 32 AH lead acid batteries	Same as *UMD-SMART*
Secondary Battery	One 32 AH lead acid battery to power the computers. Provides 2 hours of running time (measured and verified)	One 32 AH lead acid battery to power the computers and the steering motor. Provides 2 hours of running time (estimated and not verified)

System Integration

The figure below shows the integration of the various subsystems into the whole system.

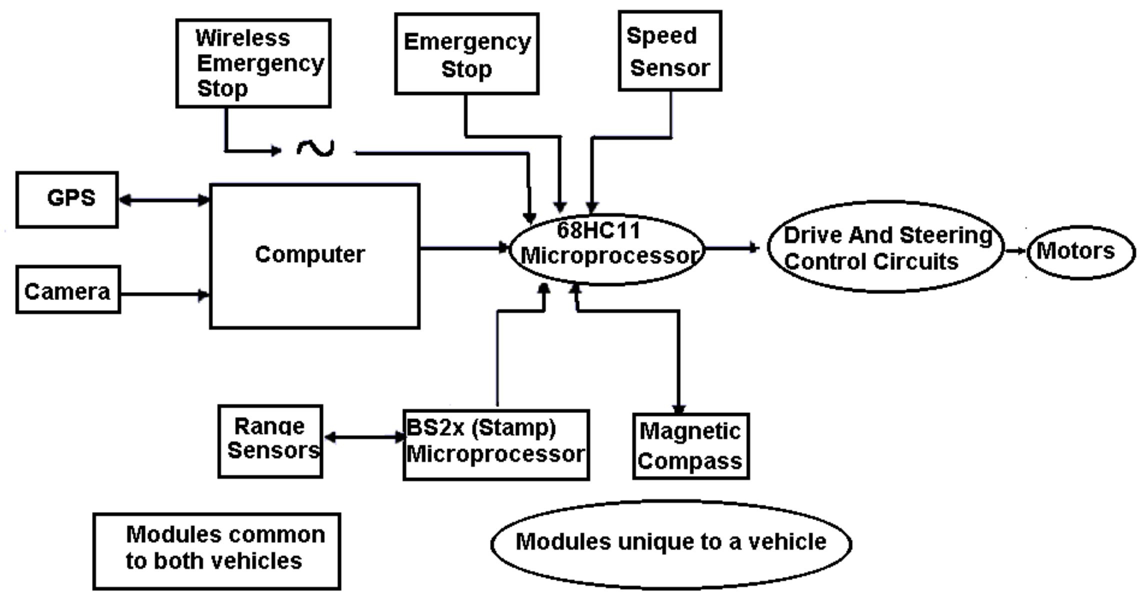

Figure 5: System block diagram

Software strategy

Autonomous challenge:

The PC control software consists of the C++ API for lane tracking, an obstacle detection and avoidance system, and steering control. The obstacle detection and avoidance system consists of the image processing software and the BS2x sensor decoder software. The main microprocessor has the software for the motor control.

Follow the leader:

The PC control software has a C++ API particularly designed to follow the lead vehicle. The software sends out the steering control signals (through the parallel port to the main microprocessor) depending on the motion of the lead vehicle. The BS2x has the software for processing the signals of the range sensors. If the range sensors detect the lead vehicle within a 3-meter range, the near flag will be asserted to instruct the main processor to decrease the speed of the vehicle. On the other hand, if the vehicle lags behind, another flag is asserted to increase the speed of the vehicle.

Navigation challenge:

The PC control software has a C++ API to decode the GPS signal information, and the obstacle detection and avoidance system. The obstacle detection and avoidance system consists of the image processing software and the BS2x sensor decoder software. The data obtained from the GPS unit is used to send steering control signals to the main microprocessor to navigate the vehicle from one waypoint to other. When an obstacle is detected, the obstacle avoidance system will take control of the vehicle until the obstacle is avoided.

Performance (most of the testing was performed on *UMD-SMART*)

Tasks	Performance
Lane following	Tracks lanes with an accuracy of 0.5 meters (verified)
Obstacle detection	Detects obstacles within the range of 10cm to 5 meters
Pothole detection	Testing in process
Dead end detection	Testing in process
Braking distance	15 cm on grass and 30 cm on cement (*UMD-SMART*)
Maximum speed	3 km/hr
Turning radius	25 inches (*UMD-SMART*)
Ramp climbing ability	Can climb inclines of 15° at 2 km/hr
Waypoint detection	Less than 1 meter using Kalman filter
Manual emergency stop	Instantaneous (*UMD-SMART*)
Wireless emergency stop	100ft (not verified)
Battery life	Auxiliary battery - 2 hours of continuous operation 8 hours of drive

Safety, Reliability and Durability

The team wanted to ensure the safety, durability and reliability of the vehicle and as a result, the following methods were implemented into the design:

Dead-man switch

GCAT has a limit switch which acts as a dead-man switch. The normally closed connection is across the resistor, which controls the speed. If the switch is not pressed, the resistor is shorted out and as a result the vehicle is stopped. On the other hand, the switch must be continuously pressed (there is an override switch for autonomous mobility) to run the vehicle.

Manual emergency stop

The manual emergency stop consists of a red push button to stop the vehicle immediately. Pushing the stop button cuts off the power to the motor and in the case of *UMD-SMART* locks the wheels by turning on the electronic brakes; this in turn brings the vehicle to an immediate stop

Wireless emergency stop

A wireless doorbell with a range of 100ft (30 meters) was modified to stop the vehicle remotely. When the remote button is pressed, it cuts off the power to the motor, which stops the motor immediately.

Durability, and Reliability

The vehicles were tested for endurance by running them continuously. They both ran continuously for two hours. All the circuits are color-coded to reduce the chance of accidental misconnections. Optical isolators are used to isolate low power components from high power circuits. Comparators are used to increase noise immunity. A light turns on when the vehicle detects an obstacle and if the obstacle is too close, the vehicle is stopped. LED's were connected to every circuit to verify that all the circuits were functioning properly.

Redundancy

The design of vehicles incorporates various redundancies to ensure vehicle operation in case of a particular system failure. The redundancies are as follows:

1. An obstacle detection and avoidance system was implemented using range sensors and a vision system. In case the vision system fails, range sensors will take control of the system.

2. To keep track of the vehicle's position GPS and a magnetic compass were used. If GPS loses the signal from the satellites, the magnetic compass will be used to keep track of the direction that the vehicle is moving in.

Cost

Equipment	***UMD-SMART***	***GCAT***
Chassis	$950	$600
GPS	$150	$150
Magnetic compass	$50	$50
Sensors (SRF08, SRF04)	$200	$200
Batteries	$50	$50
Components	$200	$200
Computers (PC, 68HC11, Stamp)	$1270	$1270
Video camera + Dazzle	$350	$350
Miscellaneous	$200	$200
Total	**$3420**	**$3070**

Acknowledgments

The team would like to thank Motorola for their gift of the GPS unit, and Daimler Chrysler for their gift of the window regulator units, even though they were not utilized. The team would also like to thank Dean Sengupta for his financial support, Andy Woodrich for helping us with the mechanical construction, Hassan Aliahmad, Nick Sidiropolis, Joe Musallam, Tat Lohachitkul, Walid Aldeeb, Elizabeth Tharakan, Rania Saman, and Brian Awood for all their support and assistance with this project.

Faculty Advisor Certification

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

____

N. Natarajan
Associate Professor,
Department of Electrical and Computer Engineering
University of Michigan-Dearborn