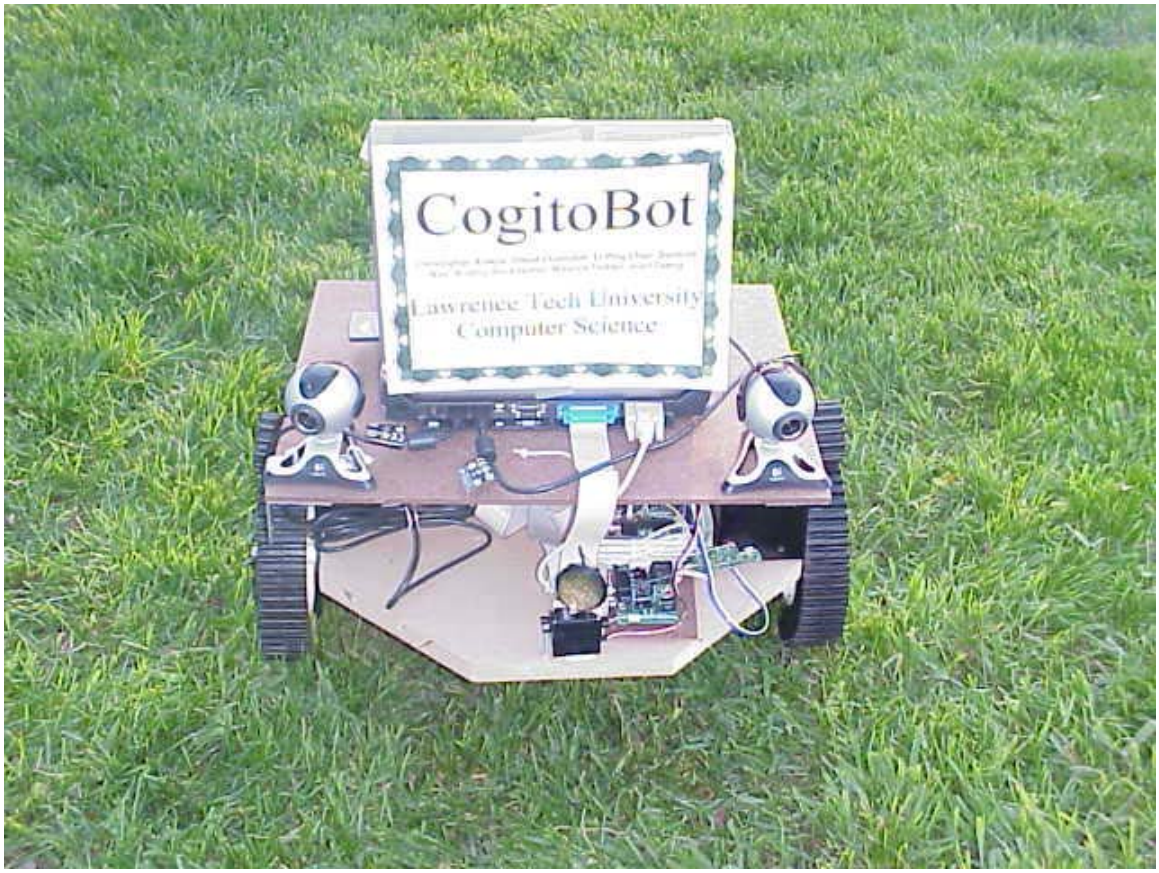
CogitoBot
IGVC Autonomous Vehicle
From Lawrence Technological University
May-June 2003



Team Name: LTU Thinkers
By: Andrey Shvartsman, Santosh B. Nair, Li Ping Chen
Date: 05/15/2003

Table of Contents

1. Introduction

Autonomous robotics projects encompass the rich nature of integrated systems that includes mechanical, electrical, and computational components. The availability of smaller and cheaper hardware components have helped make possible a new dimension in the operational autonomy. A key challenge in mobile robotics is designing intelligent and adaptable software components that allow robots to function autonomously in unstructured, dynamic, partially observable, and uncertain environments. With Hardware becoming more powerful, smaller, and cheaper, the biggest challenge is to develop adaptable software for unknown environments
So our team tries to introduce a fuzzy logic controller to solve this problem.
This concept allows us to develop robust and adaptive robots.

2. Design Process

2.1 Design Philosophies

An on-board laptop computer was selected as the main control system. It is interfaced to a microcontroller based control system, which serves as a link between the main control module, sensor modules, and motor control hardware. We created a modular mobile robot platform for computer science and engineering education with the following considerations in mind. The system is low in cost, using off the shelf components, suitable for a mass-production manufacturing model and is able to serve as an educational mobile robot platform.
A Fuzzy Logic Controller was introduced as a unique adaptive way of solving the line following and obstacle avoidance problems.

2.2 Team Organization

Team Member	Duties	Class	No. of Hours
Li Ping Chen	Software Design, Fuzzy Logic Controller, Navigation	Advanced Topics in Intelligent Systems (MS Computer Science)	300
Santosh Nair	Software Design, Implementation of Image Processing and Obstacle Avoidance Algorithms	Advanced Topics in Intelligent Systems (MS Computer Science)	225
Andrey Shvartsman	Hardware Design and Low-Level Software Drivers	Current Developments in Intelligent Systems (BS Computer Science)	275

2.3 Development Methodologies

Since this is our first time participating in IGVC competition and we did not have a robot base when we started the project in the Fall of 2002 as a part of AI class, we had to use the concept of incremental spiral model. The development process is summarized in a figure 1.

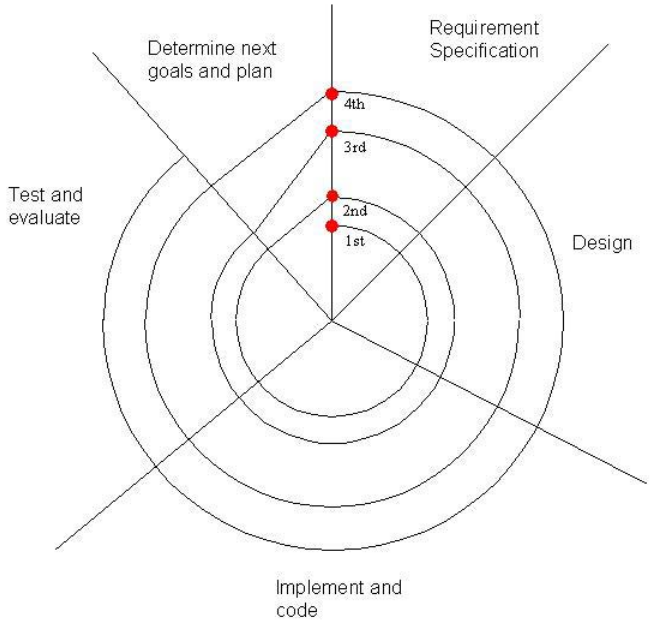


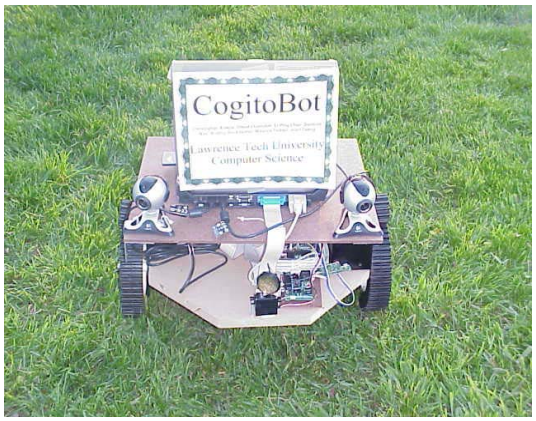
Figure 1

Development Model

Our first goal was to develop a prototype robot using a laptop as main control interface with a camera used to follow a line drawn on a blue tarp. Our second objective was to build a prototype robot with a camera to follow a curved dashed while line and returning back to base. Our third goal was to develop another prototype which detects and avoids obstacles on the course by introducing sensors such as distance and sonar. At the third iteration, we introduced the PWM motor control. Now we are in the fourth stage - optimizing the prototype toward the final version for the competition.

3. Hardware Design

3.1 Body



The robot body was designed and built from off-the-shelf components. The majority of the parts were purchased at a local hardware store.
The robot is 3 feet long and 16 inches wide in the middle. The main body was constructed from particle board 0.75 inches in thickness. The upper body was constructed from particle board 0.25 inches in thickness.

3.2 Drive Train

The drive train was designed to be front-drive to allow the robot to climb the ramp easily.
The wheel axle was constructed out of a hollow steel rod 22 inches in length. The axle was placed slightly ahead of the center of gravity, and a caster wheel was added in the back for additional support and balancing. The main wheels used on the robot are 8 inch lawn mower wheels placed directly on the wheel axle. Each wheel has 51 teeth. This serves a dual purpose. First, it provides better traction on different kinds of surfaces, and second, it allows direct coupling to a gear mounted on the motor shaft, as a more viable alternative to creating an intricate system of gears and pulleys for the drive system. The axle itself is stationary, while the wheels are allowed to freely rotate on the axle.

3.3 Gearing and Motors

We used windshield wiper motors as our drive motors. Each motor allows individual speed and direction control of each side. As a result, the robot is able to execute a zero-point turn. These are 12V DC wormgear high-torque motors. These motors have a maximum current draw of 4.5-5 Amps. The motor shaft rotates at 120 RPM. We added additional external gearing to the shafts of the motors for greater torque and reduced speed. The 13-tooth gear is coupled directly to the motor shaft, and to the wheel on the axle. The resulting 4:1 gear ratio allows for approximately 30 RPM for our drive train. Thus the maximum speed of our robot is around 1 mph.

3.4 Motor Control

The motors are controlled using a Vantec CDFR-21 dual-channel driver board. The motor board can handle up to 14 Amps of current per channel, which makes it an ideal component for this particular robot. This driver board provides PWM speed and directional control. The board interfaces to a PC through the parallel port. Commands are sent through the parallel port to the motor driver board, which in turn interprets these commands, and generates a PWM signal that is sent to the motors.
We wrote low-level drivers in C++ for accessing the parallel port and sending commands to the motor driver board. These drivers present the programmer with a high-level functional interface

to the board. The programmer makes a call to functions such as forward () or reverse (), and the driver parses these commands and sets the desired bits on the parallel port, which in turn forces the motor driver board to send the desired PWM signal out to the motors.

4. Electrical Systems

4.1 Sensors

Our design innovation for obstacle detection consists of using two different types of sensors as a means of detecting obstacles and rotating the sonar sensor on a servo. Our primary method for obstacle detection is provided by two infrared distance (IR) sensors and one rotating sonar sensor. We chose these sensors over camera based obstacle detection because they are a proven technology and have high reliability. Our previous experience with the Sony GD2D02 IR sensors and Polaroid sonar module comes from our work with small-scale Lego and Handy board robot projects. The IR sensors are very robust at detecting objects less than two feet away and remain in a fixed mounted position on the front of the robot. There is one sensor for the left side and one for the right side. This arrangement provides continuous feedback for the left and right edge zones of the robot to identify an immediate turn vector necessary to avoid the obstacle. The IR sensors are primarily a backup detection method for the Polaroid sonar sensor module. The sonar provides long-range directional obstacle detection up to 35 feet (8 feet reliably) for a 180-degree scan zone in a 15-degree cone. Rotating sonar gives the robot range and angular position data of the obstacle detected. Since our robot only has one sonar sensor we use a servomotor to rotate it and scan at 15-degree angle increments.

4.2 Handy Board Microcontroller and Sensor Data Processing

As part of our design requirement, we did not want the central laptop CPU involved in any low-level sensor processing because it would require extra CPU processing time that we needed for vision processing. The IR and Polaroid sonar sensors do not provide data output directly but must instead be driven by a separate microcontroller for this function. The microcontroller handles the lower level sensor processing functions without requiring any processing power from the central laptop computer CPU. In addition, the central laptop computer is not equipped with sensor drivers and digital/analog I/O circuitry necessary for processing IR and Sonar sensor data. The Handy Board is a good choice for this application because we have previous experience using the Handy Board with these sensors in previous small-scale robot projects. Other features of the Handy Board include a 2 MHz Motorola 68HC11 microprocessor, 32K static RAM, servomotor controls, digital and analog I/O, and RS232 serial port for communicating with the central laptop computer. The central laptop computer receives IR and sonar sensor data from the Handy Board via our custom ASCII format protocol data packets transmitted though the RS232 serial port

connection. This ASCII serial port format has the advantage of computer hardware and software independence because as long as the laptop has a serial port and the programming language supports serial port I/O then any arrangement of computer (PC or Mac), operating system, or language (Java or C++) can use these sensors for obstacle detection. The Handy Board can also support up to four IR sensors, four servomotors, and additional I/O as our future needs expand.

4.3 Vision Sensors

To reduce the complexity and cost of providing computer vision hardware for our robot design we selected camera equipment already available in our lab and free software available from Intel to interface with the camera and process image data. Detection of the autonomous course lines is provided by a 2 Logitech camera (640 X 280 video resolution VGA CCD with 30 fps) using a USB interface. No other hardware is needed except for a laptop with a USB port. Capturing and processing image data is accomplished using the Intel Open Computer Vision Library for C++, which is provided as a free download from Intel from the Internet. This library provides over 300 functions for handling camera data and provides simple software interface functions for capturing, processing, and displaying camera data. The main functions we use are the camera device driver function and the image frames capture functions, which give the programmer access to image pixel data. There are also line, edge, and contour detecting functions available.

4.4 Power System

One 12-volt 7amp hour battery provides electrical power to the drive motors, Vantec motor driver board, and Handy Board external power. Based on experimental test data we found that our battery can last for 1 hour before being replaced by a newly charged battery. The GPS unit and central laptop computer both have their own rechargeable batteries separate from the main robot power system. Although this design requires the maintenance of separate batteries, it isolates and protects the laptop and GPS from the unlikely event of a dangerous electrical surge in the robot power system. A ten-amp fuse protects the entire robot electrical system and Vantec motor board from over currents. An additional 500mAmp fuse protects the Handy Board microcontroller from over currents. For safety reasons and as part of the IGVC qualification requirements, manual and remote electronic emergency stop switches are wired in series with the battery connection. Thus, both switches must be closed for electrical power to the main system power bus. Any one of these switches can be opened to cut off all electrical power and stop the robot in the event of a dangerous loss of control. The remote e-stop consists of an automotive keyless entry remote switch unit with a range of 50 feet and a 30-amp relay in series with the battery connection to the main power bus. One design innovation that we implemented is a logical electrical system layout that is made self-explanatory by proper labeling and aids in on the spot troubleshooting. These

safeguards designed into our electrical system also protects against electrical failures due to human error. Figure 2 shows the robot's electrical diagram.

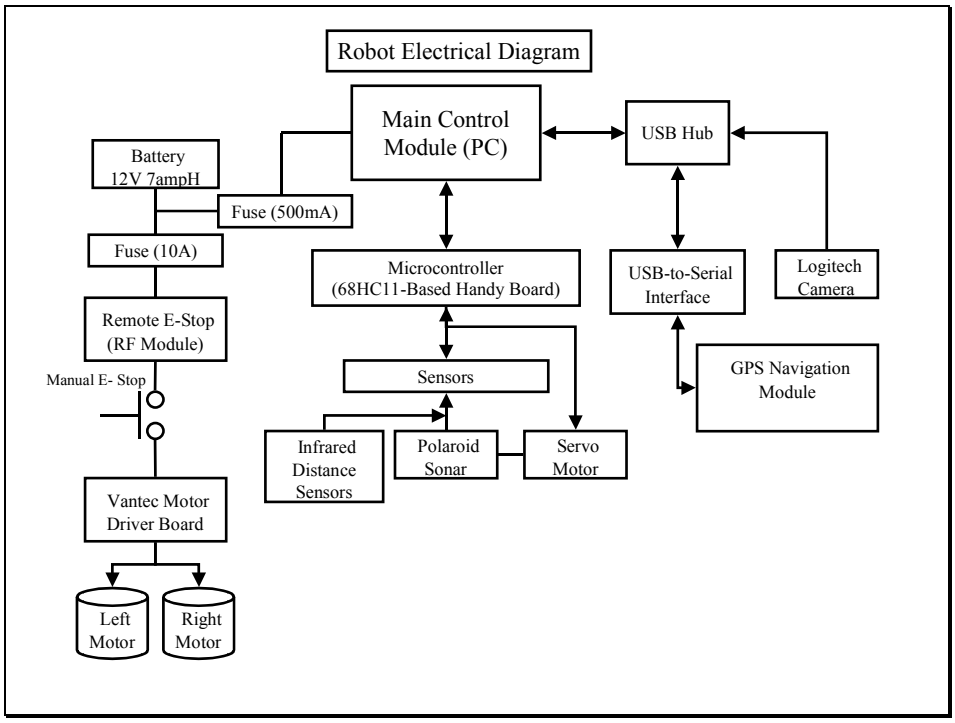


Figure 2
Electrical Diagram

4.5 GPS Navigation

To effectively compete in the navigation challenge we decided to use GPS navigation to navigate the navigation course. The Garmin LSV-25 OEM GPS unit was selected for this purpose because it provides the necessary range of features at a reasonable price. The built in features of the GPS unit provides all the functions necessary to navigate the robot autonomously. It allows us to set the destination coordinates and continuously provides feedback indicating the direction and distance to the destination waypoint. This information is used in a feedback loop with the robot motion control system to guide the robot through the navigational challenge course. All that is needed to use the navigational features of the GPS unit is to interface using the RS232 serial port and the NMEA (National Marine Electronics Association) protocol for sending and receiving data from the GPS unit. If an obstacle is detected in the path of the robot then the obstacle avoidance routine assumes priority over GPS control of the robot and uses a reactive scheme to steer the robot away from obstacles. Once the obstacles are avoided, GPS navigation control of the robot resumes.

4.6 Computational Hardware

The onboard computer used for all vehicle control, sensor interfacing, and communications is a laptop computer that uses a Pentium III 600 MHz processor running a Windows 2000 operating system. The laptop has a single USB port which is extended by a 4-port USB hub for connecting the camera and other peripheral devices, a parallel port for connecting the motor driver board, and a serial port for connecting to the microcontroller, which in turn is connected to all of the on-board sensors. The laptop is connected to a virtual private network through a wireless LAN system. This allows for remote debugging and observation by a team of programmers.

5. Software Design

5.1 Overview

CogitoBot software consists of several key components. Vision processing is used as a primary sensory input for line and obstacle detection. IR and sonar sensor processing is utilized as a backup sensory input system to avoid crashing into obstacles in case a vision processing failure occurs. Decision processing uses a Fuzzy Inference System to determine the rules for following the line and avoiding obstacles. And motor processing is used to send commands to the motors and control the motion of the robot.

5.2 Architecture

Figure 3 shows the key elements involved in the processing for the Autonomous challenge.

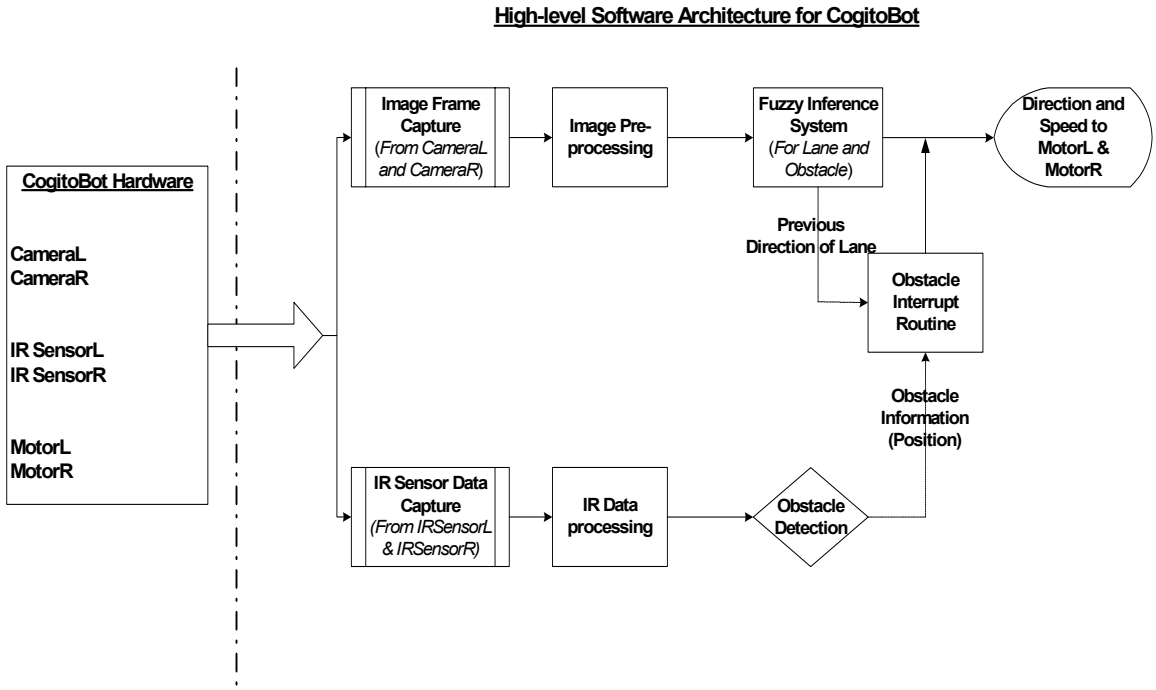


Figure 3

Software Architecture for CogitoBot

5.3 Design Details

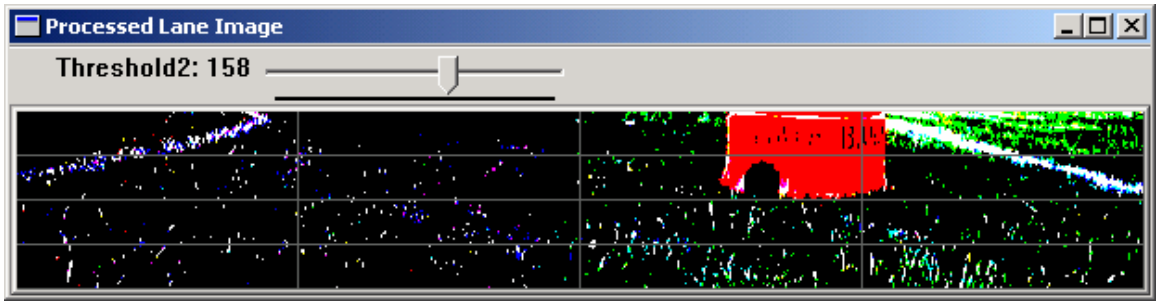
The software for IGVC robot is implemented in Visual C++ 6.0 using MFC. The application is Menu driven for convenience, with options to initialize, execute different competitions and exit. Middle layer software is used to communicate with the Hardware. The camera interface is implemented using the Intel Image Processing Libraries and Intel OpenCV libraries. The IR interface is implemented through the HandyBoard microcontroller. Data is communicated via the Serial Port of the Laptop from the Handy-board. The Motor Interface is implemented by a motion control driver that accesses a system driver which accesses the parallel port directly. Data is sent between the motor driver board and the program via the parallel port of the laptop.

5.4 Vision Processing

Image frames from two cameras (CameraL and CameraR) are captured using Intel OpenCV library functions. These two images are then concatenated to form a single image, which now has a "wide view" of the scene in front. This enables easy view of both the lanes, rather than just working with a single lane.

The image is then pre-processed to identify and filter various colors, primarily, white, orange/red and beige colors. Depending on the light conditions, the threshold intensity of the image can be adjusted dynamically at run-time using the "slider" on the image window.

This pre-processed image is then split into a grid of 32 cells (4X8).



Information from these cells are interpreted (based on color intensities) and stored into an array of same size (4X8). Each array element will contain amount of "white, orange, beige, others" color information (represented by Pixel Count)

Based on white pixel count in each cell, the lane center position is calculated. If one of the lanes is not visible, then the lane center is estimated based on the visible lane. The obstacle edge position is calculated by counting orange and beige pixels in each cell. The lane center position and obstacle edge position serve as inputs to the Fuzzy Inference System in the decision processing module.

5.5 Sensor Data Processing

Data from two IR sensors (IR SensorL and IR SensorR) and a sonar sensor is received using the sensor interface software module. IR Sensors are used as backup sensors to avoid crashing in case an obstacle is undetected by the Vision Processing or the sonar sensor. Based on which IR Sensor (Left or Right) detected the obstacle and based on previous direction of the Robot (as stored in memory) appropriate actions will be taken by the Decision processing module.

5.6 Decision Processing

This consists of two main parts:

- Fuzzy Inference System (FIS) – The primary decision making for CogitoBot is done using a Fuzzy Inference System.

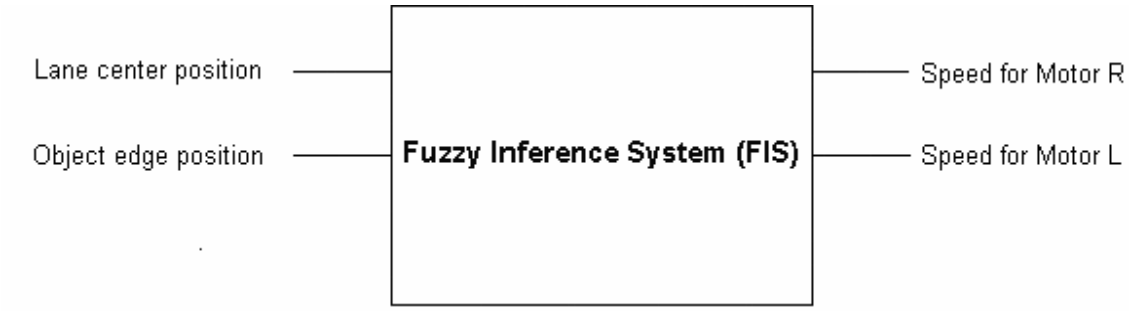


Figure 4

High Level FIS

(NOTE: The output of the FIS is actually speed and direction. Since the CogitoBot uses "differential steering" mechanism, the direction of CogitoBot is controlled by motor speeds)

This Fuzzy Controller uses the Sugeno model for the Fuzzification and De-fuzzification of data and "Triangular membership function" to define the Data sets.

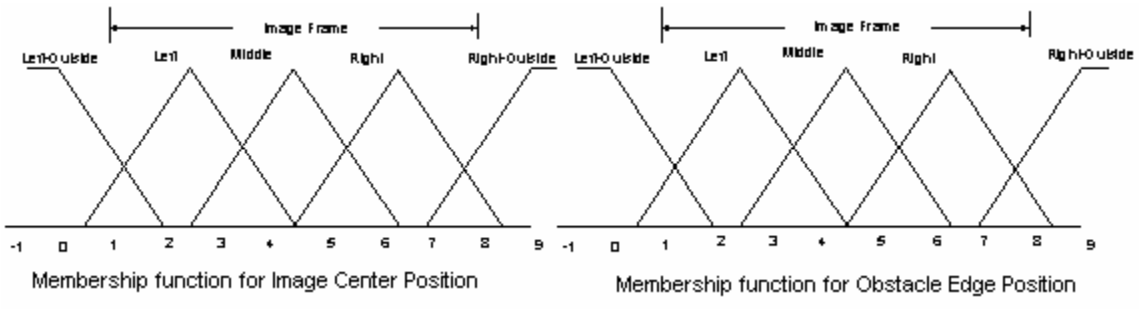


Figure 5

FIS Input Member Function

Rules for the FIS:

The following table lists all the rules for the FIS. The header column data is the position of the calculated lane center, and the header row is the position of the obstacle edge. The data within the table represents the result produced by the FIS rules, and corresponds to the robots direction and speed.

Object Edge / Line Center	No Obstacle	Left	Middle	Right
Far Left	Hard Left	Left	Left	Slight Left
Left	Left	Left	Left	Slight Left
Middle	Straight	Slight Right	Random (Left or Right)	Slight Left
Right	Right	Right	Hard Right	Right
Far Right	Hard Right	Slight Right	Right	Slight Right

- Obstacle Interrupt Routine – If an IR Sensor detects an obstacle, it means the obstacle went undetected by the Vision Processing. To correct this, the program enters this Obstacle Interrupt routine. This routine checks the current direction the CogitoBot was moving (as stored in memory, based on the FIS outputs), and depending on which IR Sensor(s) detected the obstacle, the vehicle moves in the reverse direction for a short distance and moves ahead with reduced speed, hoping the Vision processing will recognize the obstacle and the action will be processed by the FIS. A counter of how many times the Vision failed to detect is stored. If the Vision fails to detect the obstacle after two attempts of correction, the interrupt routine will try to by-pass the obstacle using simple logic of turning CogitoBot to certain angle, moving

forward for a certain distance and then correcting the direction again. During this correction FIS will not be allowed to decide the action for the CogitoBot.

5.7 Motor Processing

The direction and speed of the CogitoBot is controlled by direction of each motor and its speed. The Motor interface software functions take the motor name and speed as the input. Turning of the motor is achieved by reversing each motor depending on the direction desired. For example, Left motor is forward and Right motor is reverse with a high speed value would give a sharp right turn.

6. Vehicle Cost Summary

The robot was designed to be low-cost and modular. Thus our expenses were kept as small as possible. The table below summarizes our expenses for this vehicle.

Component	Purpose	Cost per Unit	Total Cost
Vantec CDFR-21	Motor Driver Board; provides PWM and directional control	$250.00	$250.00
Handy Board	Microcontroller; provides interface between sensors and laptop	$250.00	$250.00
2 – 12 VDC Motors	Integral component of the drive train	$20.00	$40.00
GP2D02 IR Sensors	Distance Ranging Sensors	$15.00	$30.00
Polaroid 6500 Sonar Sensor	Distance Ranging Sensor	$48.00	$48.00
0.75“ Particle Board	Lower Portion of Robot Base	$3.00	$3.00
0.25” Particle Board	Top Portion of Robot Base	$2.00	$2.00
8” Lawn Mower Wheels	Integral Component of the Drive Train	$8.00	$16.00
Camera	Provides visual information about the environment	$50.00	$100.00

GPS	Navigational Info	$150.00	$150.00
Miscellaneous	Connectors, batteries…	-	$100.00
Total			$989.00

7. Conclusion

This is the first year that Lawrence Technological University is participating in the IGVC competition. Our team has overcome multiple challenges in both hardware and software design to get to this point. The integration of several software techniques and application of a fuzzy logic controller to a real world problem produced a unique vehicle, the CogitoBot, which is the cumulative result of more than a year's work by many different people.

Faculty Advisor Statement

I, Dr. Chan-Jin Chung of the Department of Math and Computer Science at Lawrence Technological University, do hereby certify that the design of the vehicle, CogitoBot, by the LTU Thinkers team, has been significant and each team member has received credit from this University for the work completed.

Signed,

Dr. Chan-Jin Chung
(248) 204-3504