Project Zenna

2004 IGVC entry from
Lawrence Technological University

Team Zenna:
Mike Nasers
Mohammad Jallad

Faculty Adviser Statement: I, Dr. Chan-Jin Chung of the Department of Math and Computer Science at Lawrence Technological University, do herby certify the members of Team Zenna produced this entry into IGVC with a high degree of effort for University Credit.

Dr. Chan-Jin Chung

1 Introduction

The goal of project LTU Zenna was to imbue a vehicle with the intelligence necessary to autonomously navigate an outdoor obstacle course and compete in the IGVC. Therefore, developing the vehicle's artificial intelligence became the primary concern of the project. Evolutionary artificial neural networks were used extensively for this purpose because of their high degree of adaptability. They were used both for image processing and vehicle navigation. The vehicle uses a laptop computer for both motor control and image processing. A Digital Video Camcorder was used to provide the sight for the vehicle. The Camcorder is interfaced to the laptop using a Firewire PCMCIA adapter. A GPS sensor was used for the navigation challenge and interfaces to the laptop via the RS-232 cable. The navigational hardware was designed to be as inexpensive as possible while still allowing the vehicle to easily navigate the IGVC course. This resulted in the extensive use of 'off the shelf' components. This kept the vehicle cost down and allowed more time for developing the vehicles intelligence.

2 Design Process

2.1 Overview

The Zenna project used an iterative design process. This process starts with simple tasks and gradually builds upon them in order to perform increasingly more complex tasks. The first task was for the vehicle to follow a single white line. Completing this task verified that a laptop could be used to control the vehicles motors and process frame data from a digital camera using firewire. Also, the basic hardware design was determined during this time.

The next task was to navigate an indoor obstacle course made up of orange barrels that were bounded by two white lines. The vehicle had to avoid the obstacles while remaining between the white lines. In order to complete this task, the motor control algorithm was modified to provide better performance, and an artificial neural network was implemented to improve the image processing. The artificial neural network provided better color recognition and was more tolerant to noise.

The first two tasks were performed using a simple stimulus-response algorithm to control the vehicle's navigation. The input into this algorithm was an information vector containing information about the current camera frame. This input vector was compared with user-defined IF-THEN rules. If the input vector matched a predefined antecedent (IF portion of a rule), then a consequent (THEN portion of a rule) was performed. An example of this would be if the input vector indicated that white pixels were present in the upper right of the camera frame and nowhere else, the vehicle would turn slightly left. If none of the antecedents matched the input vector, a default action was performed. All rules needed to be programmed manually. If an unanticipated situation were encountered, the vehicle's behavior would be unpredictable. To account for these situations, a new rule will need to be added to the algorithm and the source code recompiled.

The third task was to navigate a more complex outdoor obstacle course. Because of the limitations of the stimulus response algorithm, an artificial neural network to control navigation was introduced. The network was trained with a set of sample input vectors. The weights of the network were adjusted until the network could produce the proper action for each sample input. After training, the neural network was able to adapt and select the correct action even for previously unseen input vectors. Also because of the constantly changing lighting conditions outdoors, the neural network used for image processing needed to be improved for this task.

The last task was to continue evolving the design through testing and perform continuous improvement to the navigation, image processing and motor control algorithms. This involved collecting additional data for all neural networks, determining the best network topology for each neural network, and re-training them to reduce network errors. The motor control algorithm was further refined, and the vehicle's hardware design was modified as needed.

2.2 Team Members

Table 2-1 below shows the team members, team member responsibilities and the hours spent on each task for project Zenna.

Team Member	***Major***	***Responsibilities***	***Hours***
Mike Nasers	MS Computer Science	Camera Interface	10
		Image Processing	150
		Scene Analysis	150
Mohammad Jallad	MS Computer Science	Decision Logic	200
		Motor Control/Interfacing	100

Table 2-1 Team Member Responsibilities

3 Image Processing

3.1 Image Processing Algorithm

The purpose of the image-processing algorithm is to process camera data into a suitable for use by the artificial neural network used to control navigation. The first step is to capture a frame of data from the Panasonic DV Camcorder. The frames are captured using a Java Class provided by LTU student Tom Burke that used JNI to call C++ classes used by Windows XP to process image data. Each frame of data is stored in a 120 x 160 array of pixels.

The next step in the algorithm is to determine the red, green and blue (RGB) values of each pixel. These values are used as inputs into an artificial neural network used to determine the color of each pixel. The possible colors for each pixel are: white, orange, green, yellow and black (all colors except white, orange, green and yellow). If it is determined that a pixel is green, the RGB is input into a second artificial neural network. This network determines if the shade of green of the pixel is from a blade of grass or from the ramp that the vehicle must cross.

After the color of all pixels is determined, the image is reduced from 120 x 160 pixels to 12 x 16 pixels. This is done by combining pixels into 192 10 x 10 blocks. Each block contained a color value (one for white, two for orange, etc). The value of a block is based on the most dominant color in the block. For example, if the majority of a block is white, the value of the block is set to one. The values of all the blocks are used as input into a different artificial neural network that determines the appropriate direction for the vehicle.

3.2 Color Determination Artificial Neural Networks (ANNs)

Artificial Neural Networks are used to determine the color of each pixel of a frame. The ANNs use the RGB (0-255) components obtained from the Camera. ANNs were chosen because they provided benefits over using RGB threshold values. One benefit is that the ANNs can recognize and distinguish different colors better in all lighting conditions. Another benefit is that the ANNs are much less sensitive to glare and other types of noise. In fact, the ANNs are able to eliminate all but the most intense cases of glare from the sun. These cases were encountered during testing at different times of the day depending on where the test track was set up.

Two neural networks are used to determine the color of each pixel. Both networks have three layers. The primary one, shown in figure 3-2, has three input neurons, five hidden neurons and four output neurons. Each output neuron represents a single color. If all output neurons are zero, the pixel is considered to be black. The second, smaller, network determines the shade of green of a green pixel. This information is used to determine if the vehicle is traveling on grass or on the ramp. This network uses three input neurons, three hidden neurons and one output neuron.

Note: *The main color determination ANN contains an output neuron for yellow. Yellow detection is not needed for IGVC but was used for a training task. A second version of the ANN was produced for the IGVC that only had 3 output neurons. However, it did not perform as well as the network with 4 output neurons and was therefore replaced. The color yellow is not considered in the navigational algorithm.*

3.3 Color Artificial Neural Network (ANN) Training

Both color detection networks were trained by an algorithm using the ES (1 + 1) Evolutionary Strategy with a 1/5 rule. This method was chosen over the commonly used back-propagation algorithm because it has a much shorter training time. This strategy starts with a parent neural network containing randomly initialized weights. The main color determination ANN uses 44 weights. The ANN used to separate the different greens uses 16 weights. The weights are modified using a Gaussian distribution function multiplied by a learning rate to create an initial child neural network.

A sigmoid activation function is used to calculate the output of each neuron. This output is compared with the expected output for all samples and the sum of the squared error is calculated using the following formula:

$$\sum_{x=1}^{x=N} \left(y_d - y_x \right)^2$$

Where Y_d is the desired output vector for input sample *x*, Y_x is the actual output vector of the sample and *N* is the total number of samples. The error for both networks is calculated. If the child network's error is lower, it becomes the parent. Otherwise the parent is kept, the child is discarded and new child network is produced. If a child is kept, this is considered to be success. Periodically, the number of successes is checked. If the success rate is greater then 1/5, the learning rate is increased. Otherwise the learning rate is decreased. The algorithm continues until a network with an acceptable error is found. Through testing, it was found that an error less then .001 produced acceptable results.

A program was written in Java to collect the RGB components used for training. One of the colors the ANN was being trained to collect was placed in front of the camera. After being instructed by the user, the program would collect RGB data at 100 random points in a 20 x 20 pixel section of the 120x160 pixel frame. The average of the 100 values was written to a file along with the color value of the sample. This was used as one input into the ANN. Several samples were collected for all colors under different lighting conditions. Twenty samples for each color were collected except for white. For white, it was found that using 25 samples produced better results. These values were determined through testing.

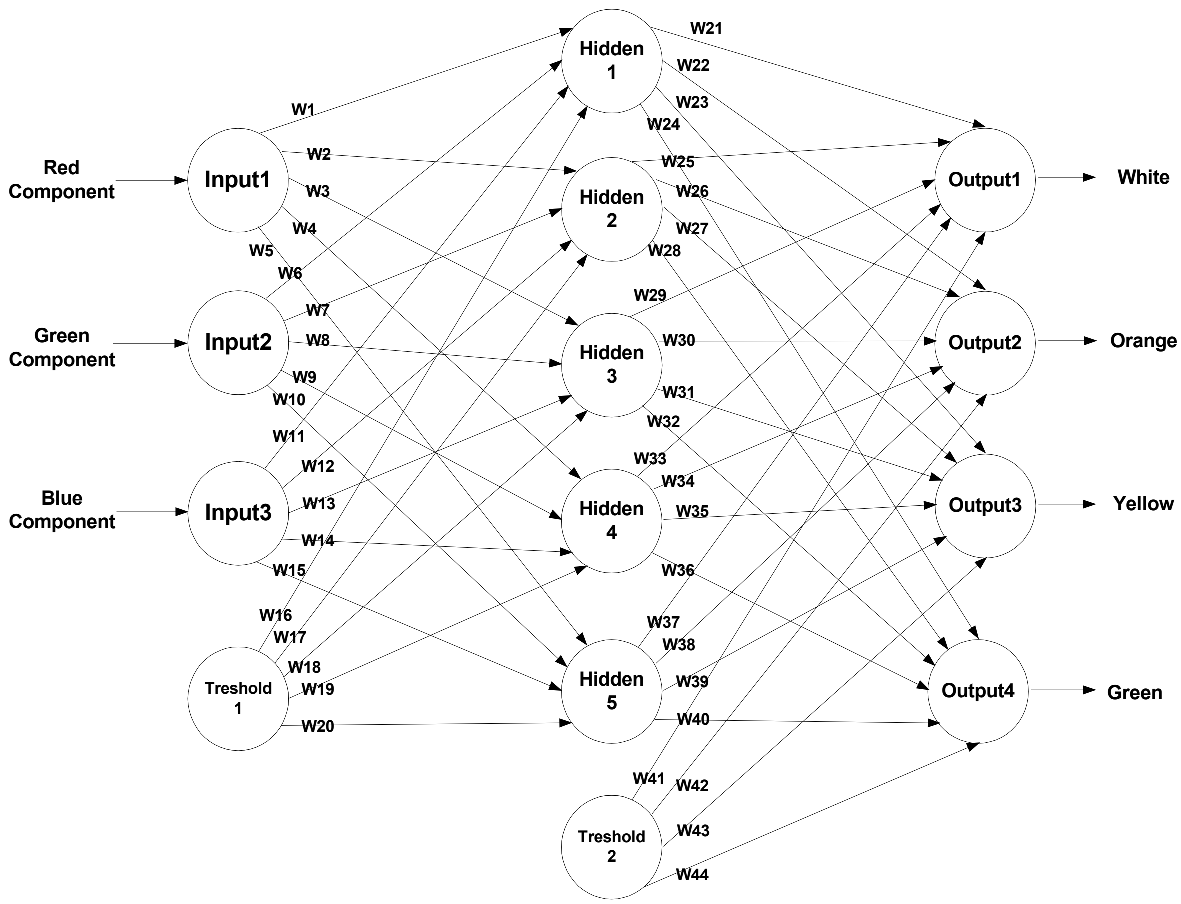

Figure 3-2 ANN Used for pixel color determination

4 Hardware Design

4.1 Motor Control

Drive Train

Two 12-volt electric motors are used to power the vehicle. The motors are Fisher Price Hot Wheels gearbox motors. These motors are capable of carrying 70 lbs at speeds up to 5 mph. The motors combined have the capability to carry the IGVC required 20 lbs payload at speeds up to 12 mph. The motors can climb ramps having a maximum angle of 20 degrees. The actual speed of the vehicle is limited by software to 5 mph in accordance with the IGVC maximum speed requirement. Each motor has independent speed and direction control. Each is also directly connected to a wheel on a fixed axle and differentially steers the robot giving it zero turn radius capability. Figure 4-1 contains a picture showing the vehicles motors.

Figure 4-1 12 Volt Electric Motors

Motor to Wheel Coupling

A unique design feature of our drive train is that the motors and wheels are easily removed and changed by simply removing the retaining cotter pin on the axle and separating the wheel from the motor. This ability allowed our team to test different wheels for the best traction and performance. A custom coupling plate was needed to connect the Fisher Price motors to the wheels because the motors are specifically designed to mate with the Hot Wheels vehicle wheels. Our test proved that these original Hot Wheels vehicle wheels did not provide the needed traction for outdoor use on grass, a wooden ramp, or sandy terrain. This issue was solved by designing a special motor to wheel coupling plate. This allowed us to use a wide variety of wheel assemblies. The best traction was obtained by using a 10.5-inch pneumatic mini bike wheel as the selected wheels for our vehicle.

Motor Control

The Vantec CDFR-21 motor controller proved to be very reliable and effective for the 2003 LTU IGVC vehicles, and it was selected for use as the motor controller for the 2004 vehicles. This motor controller is interfaced to the main computational computer via the parallel port. Writing commands to the parallel port controls the speed and direction of the motors. The CDFR-21 is capable of controlling two electric DC motors with a PWM output voltage range of 5 – 30 volts and continuous operating current of 14 amps per channel and 45 starting amps per channel. Figure 4-2 shows the Vantec Motor Controller.

Figure 4-2 Vantec Motor Controller

4.2 Electrical System

Onboard Computer

The vehicle uses a Toshiba Satellite 2415-S205-laptop computer for image processing, navigational computation and issuing motor commands. The laptop uses a 2.0 Gigahertz Intel Pentium 4 processor and has 256 MB RAM. Windows XP is used as the operating system. The laptop was chosen because its high processing rate was able to handle all computational tasks. A Belkin J1394 PCMCIA Firewire notebook adapter was needed to interface the DV Camera to the laptop.

Vision Sensor

The vehicles vision abilities are provided by a Panasonic digital video camcorder with an IEEE 1394 Firewire computer interface. This camcorder is used to detect lines, potholes, and obstacles. Adjusting the camera tripod mounted on the vehicle, which provides camera height, tilt, and direction adjustments, can make field of view adjustments. The distance at which the vehicle can detect objects depends on the angle the tripod is set. This detection distance is usually set to be around 6 feet. The camcorder was chosen over more inexpensive web cameras because of its ability to adapt quickly to the changing lighting conditions of the IGVC course. This prevents the vehicles navigational algorithm from interpreting sunlight reflections as an obstacle or line. The DV camcorder interfaces to the laptop computer using IEEE 1394 Firewire. An IEEE 1394 Firewire interface increased the camera data bandwidth, which is another advantage over a web camera. The camera frame rate is 20 frames per second. This gives the vehicle a reaction time of .7 seconds.

GPS Sensor

The Garmin eTrex Venture handheld Wide Area Augmentation System (WAAS) enabled GPS unit was selected as the GPS sensor for the navigation challenge portion of the IGVC. This GPS unit provides a wide range of features at a reasonable price. The built in features of the eTrex GPS unit provide 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 desired waypoint. All that is needed to use the navigational features of the eTrex unit is to interface using the RS-232 serial port and one of the various protocols built into the eTrex unit for sending and receiving data from the eTrex GPS unit. The advertised accuracy of the GPS unit in WAAS mode is less than three meters, which is the actual accuracy of the robot.

Power System

Figure 4-3 shows the schematic diagram of the vehicles electrical power system. A 12 volt 7 amp hours sealed lead acid rechargeable battery produces main electrical power. Predicted time required before a recharge is needed was calculated to be 1-3 hours at normal speed for a fully charged battery. This prediction was verified by experimental tests. The time varied based on the tasks the vehicle was performing. A backup battery is always charged and ready to replace a weak battery. Other notable safety features of the vehicles electrical system is the resetting circuit breaker panel which provides 15 amp over current protection for each motor and the 30amp over protection for the overall system. The laptop and GPS each have their own internal batteries.

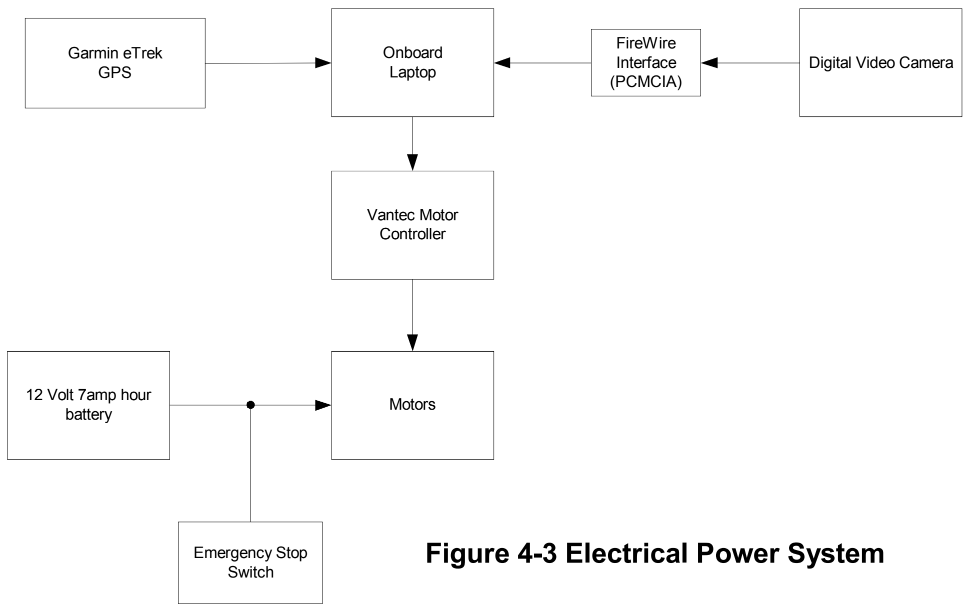

Figure 4-3 Electrical Power System

<u>Emergency Stop</u>

As required by IGVC contest rules, a remote and mechanical emergency stop switches are required to stop the vehicle. To maintain the off the shelf design philosophy, a low cost automotive keyless entry switch with a 50 feet range was selected as the remote E stop component. The remote E stop controls a relay that is in series with the mechanical push-pull E stop and the Vantec motor controller main power. Both of these switches must be closed before electrical power can be supplied to the Vantec motor control board. Another notable safety feature of the E stop systems is that the vehicles main power can only be restored after remote E stop shutdown by resetting the vehicle. This prevents the vehicle from being restarted until a person is in physical proximity and can decide to restart the vehicle.

Appearance and Configuration

Figure 4-4 shows the overall vehicle configuration. The vehicle has two levels. The top level is used as a platform for the laptop computer and as a base for the DV camcorders tripod. The mechanical emergency stop switch is also located on the top level to provide easy access. The two motors that power the vehicle are located in the rear of the vehicle. A free-moving caster wheel is located in the front to facilitate differential steering of the rear wheels. The 20 lbs payload is placed directly above the rear wheels to provide better traction. The lower level also holds the Vantec motor controller. Medium Density Fiber (MDF) board was used as the main structural component of the lower level. This material is easily machined and processed using standard household power tools. A .25-inch thickness was used. The top level is constructed from .25-inch thick oak. Building our vehicle consist of cutting the MDF panels to the desired shape and drilling holes to fasten the various components together with bolts and PVC spacers.

Figure 4-4 Final Configuration

5 Vehicle Control Software Design

Artificial Neural Networks (ANNs) were used to control our vehicle. ANNs, which closely model human brains, can be trained to produce the desired outputs. Using ANNs to command our motors was accomplished in three stages: collecting sample data, training ANNs, and testing their performances.

5.1 Collecting sample data

The process of collecting sample data went through a series of steps. First, a snapshot of an outdoor testing course was captured by the camcorder camera and then passed as 120 x 160 pixels image buffer to our image-processing algorithm. The image-processing algorithm, in turn, scaled down the collected image buffer to a 12x16 binary input vector. Next, the input vector along with the desired output (command) was saved to a disk. The same procedure was applied to more and more image frames captured while navigating the testing course until enough sample data was obtained. The number of collected sample data was 65. In the second stage, this sample data was passed to the control ANN to train it.

5.2 Vehicle Control Artificial Neural Network (ANN) Training

Using our saved sample data of binary input vectors and their desired outputs, the neural network was trained using an evolutionary strategy ES (1+1) along with modified 1/5 rule. This approach was already explained above in Section 3.2. The basic idea behind it was to update weights of the neural network in order to minimize neural network error. Updating neural network weights would stop whenever a satisfactory minimum error was achieved. Training neural network was a challenging task because the number of inputs entering ANN was relatively large. In order to enhance performance of our neural network, an optimizer function was introduced. This optimizer function was used to check for local peaks while searching for new weights attempting to minimize error of our ANN. Whenever a local peak was detected, the optimizer function re-initialized network weights and restarted the ES (1+1) training algorithm. In addition to the introduction of an optimizer function, various neural network structures with a different number of hidden neurons and output neurons were tested to see which structure would give the best performance. The final ANN topology we decided to use consisted of (12 x 16) inputs, 5 hidden neurons, and 8 output neurons (see figure 4-1). A total number of eight different commands were associated with the eight output neurons. Whenever one of the eight output neurons was fired, a command associated with the fired output neuron was sent to the vehicle motors.

5.3 Testing Neural Network

Testing of our neural network was done in three stages. First, several captured images of our testing course were passed to the neural network as binary input vectors where an output for each input vector was generated. Then, theses generated outputs were verified to see whether they would match the desired outputs. Second, the ANN control algorithm was used to command our vehicle on a testing navigation course that only included white lines. Finally, orange barrels were added to the navigation course, and the control algorithm performance was tested for dealing with obstacles.

5.4 Strengths and Weaknesses

The advantage of using neural network is the ability to train itself to generate the desired outputs by learning some given sample data. On other words, there is no need to define every possible condition and what action to go along with it. Neural networks, just like human brains, can learn how to act for a given condition. However, performance of a neural network depends largely on how efficient it is being trained. Neural network trainability is a challenging task especially when the number of inputs entering a neural network is large. In order to handle this challenge, different approaches were implemented. For example, ES (1+1) evolutionary strategy with an improved 1/5 rule was used to update neural networks weights. In addition, a new optimizer function was introduced which helped in dealing with local peaks.

5.5 Handling Dead Ends and Traps

The Artificial Neural Network (ANN) vehicle control algorithm managed efficiently to navigate Zenna between two lines or avoid an obstacle. However, this algorithm was not able to handle some special cases such as dead ends and traps. In order to fix this problem, we introduced a special functionality called the “deadEndDetector”. The “deadEndDetector” functionality was used to perform periodic check for a dead end or trap. If either a dead end or a trap were detected by this functionality, then an interrupt would be turned on. As a result, the normal vehicle control algorithm (ANN) was departed and a special function we called the “deadEndHandler” would take control of Zenna until the trap or the dead end was completely avoided. As soon as Zenna was out of the trap or the dead end, vehicle control was immediately returned back to ANN. The “deadEndHandler” managed to rescue Zenna from being trapped by commanding it to back up slightly, and to make zero radius turns until a clearance path was found.

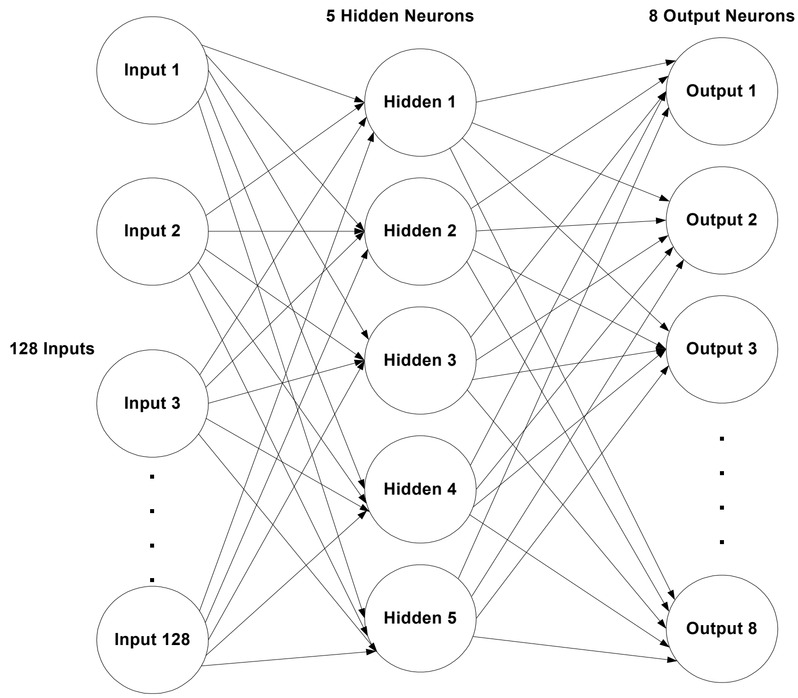

Figure 5-1 Vehicle Control Neural Network

6 Conclusions

The results of project Zenna showed that a variety of autonomous tasks can be developed and performed using an inexpensive vehicle platform as a base. The vehicle was designed to compete in the autonomous and navigation challenges of the IGVC. But because of its simple design, it can easily be converted/redesigned to perform many different applications related to artificial intelligence. Also, because a laptop is used for vehicle control, the intelligence can easily be modified or changed in order to evaluate the effectiveness of different artificial intelligence technologies. For example, the ANN used to determine the proper direction might be replaced with a fuzzy inference system or the two technologies may be combined to create a hybrid system. There is also plenty of room on the vehicle for the addition of new sensors such as a laser range finder. Team Zenna is certain that future LTU teams will be using this platform to implement artificial intelligence designs and demonstrate them at future IGVC events for the next several years.

7 Vehicle Parts and Cost Summary

Figure 7-1 contains the parts list and cost for team Zenna. The laptop, J1394 PCMCIA adapter and 6pin/4pin cable are student owned but included in the final cost. The total cost excluding these would be reduced to $1006.92.

2004 IGVC Zenna Team Bill of Materials			
Part Description	**Unit Price**	**Quantity**	**Total Price**
Chassis			
MDF Board 16" X 36" X .25"	$3.00	2	$6.00
Video Tripod	$19.99	1	$19.99
Drive Train			
12V Power Wheels Gearbox Motor Assy.	$15.00	2	$30.00
4" Tire & Wheel Assembly - 10.5" Knobby Tire	$23.99	2	$47.98
Wheel axle 3' x 7/16"	$8.47	1	$8.47
Swivel Caster wheel (4" x 2")	$12.00	1	$12.00
Wheel Adapter Plate	$0.00	2	$0.00
Sensors			
DV Camcorders	$412.66	1	$412.66
Garmin GPS 76 unit	$179.95	1	$179.95
IEEE1394 FireWire cable	$15.00	1	$15.00
Electrical			
Vantec electric Motor Speed Controller	$230.00	1	$230.00
Emergency stop push pull switch	$3.99	1	$3.99
Main Battery 12 Volt 7 Ah	$11.95	1	$11.95
Remote Control Keyless entry Switch	$37.85	1	$37.85
5' #14 - 2 conductor grey speaker wire	$0.22	5	$1.10
DPDT Relay Plug-In relay	$7.99	1	$7.99
8"X6"X3" Project Box	$6.99	1	$6.99
Computer			
Laptop Computer	$1,150.00	1	$1,150.00
Belkin PCMCIA J1394 Adapter	$45.00	1	$45.00
Belkin J1394 Cable 6pin/4pin	$29.99	1	$29.99
Miscellaneous			
Miscellaneous hardware components			$75.00
	Total Vehicle		**$2,331.91**

Figure 7-1 Vehicle Part List