



AISSIG

2004 IGVC Design Report



Team Organization:		
Steve Vachon <u>Degree:</u> MS Computer Science <u>Course:</u> Adv. Intelligent Systems	Vehicle - Electrical and Electronics, Vehicle Stability, Reliability and Safety controls, Radar Interface, Decision Module – High Level Control, Fail-safe controls.	150 Hrs.
Tom Burke <u>Degree:</u> MS Computer Science <u>Course:</u> Adv. Intelligent Systems	Vehicle – Mechanical Body, Camcorder Interface, Image processing and Data conversion (Feature Vector), Texture Identification using Artificial Neural Net.	150 Hrs.
Santosh Nair <u>Degree:</u> MS Computer Science <u>Course:</u> Collab. Research Project I	Graphical User Interface, Manual Vehicle Control, Decision Module - Fuzzy Logic Controller, Feature Vector classification - Artificial Neural Net, GPS Interface and Navigation Algorithm.	150 Hrs.

Faculty Advisor Statement

I, Dr. Chan-Jin Chung of the Department of Math and Computer Science at Lawrence Technological University, certify that the design and development on AISSIG has been significant and each team member has earned 3 credit hours for his work.

Signed,

Date

Dr. Chan-Jin Chung (chung@ltu.edu)

Introduction

As one of the four entries from Lawrence Technological University (LTU), we are happy to introduce *The AISSIG*. AISSIG is a low-cost autonomous vehicle, with primary focus on “intelligent software”. The name stands for “As I See, So I Go”. AISSIG relies very heavily on the “vision” as a key sensory input to the vehicle. The camcorder image is pre-processed to increase reliability. The other sensor being used is a RADAR, which gives extra reliability for the Obstacle detection. The Software developed for the Autonomous challenge uses Artificial Intelligence (AI) techniques like Evolutionary Computation using ES (1+1) with 1/5th rule, Artificial Neural Nets (ANN) and Fuzzy Logic Controller (FLC). AISSIG will compete in all the three competitions in this year’s IGVC.

Design Process

Last year LTU entered the competition with 2 vehicles named CogitoBot I and CogitoBot II. This year LTU is entering the competition with 4 entries, namely, **AISSIG, Deep Blue, Challenger and Zenna**. The LTU teams and our advisor evaluated last year’s mechanical structure and we decided to rebuild the vehicles from scratch due to inherent problems of slow speed and stability of those vehicles. This decision was much easier for us, due to the simplicity, low-cost and effort associated with the mechanical design. We decided to rewrite the software too. Since last year was our first time, we had some flaws in the software design. Though IGVC is a test for both Engineering and Computer Science skills, being students from Computer science, we focused our Design and Development effort on the software aspect of the vehicle to increase the performance reliability required to win the competition. Once the basic mechanical body was fabricated for all the four vehicles in our Robot Lab, each team took control of their vehicle and started building the software for Autonomous challenge and solving various problems as outlined in the IGVC Design doc. Each team made adjustments to mechanical design as fit for their approach.

For AISSIG, we first outlined and designed the software architecture needed to compete in the Autonomous challenge. Then we broke down the various software components into various software modules and distributed them amongst the three of us.

We developed the software for AISSIG primarily in Java. Due to the Object Oriented nature of the tool, we worked on each of our software objects fairly independently and then integrated all the classes by developing a Top-level class for GUI. We made changes to the mechanical design of the vehicle few times during the development process, primarily to improve stability on

ramp and sandpit. We also introduced special software modules to detect surfaces like ramps, grass and sandpit, using “texture classification” within the Vision processing module.

Mechanical Design



Robot Structure

The basic configuration of the 2004 robot is similar to the 2003 versions with two front-wheels and a caster wheel, for tank style differential steering. We maintained our design philosophy for a simple and robust vehicle, capable of being easily manufactured at low cost, from off the shelf components. A Medium Density Fiber (MDF) board of 0.25-inch thickness is used for the main structure. This material is easily machined and processed using standard household power tools and is lightweight. The new lightweight structure maintains the structural ability to carry a 20-pound payload.

Fig 1: Robot Structure

To build the robot, we cut the MDF panels to desired shape and drilled holes to fasten the various components together with bolts and PVC spacers.

Drive Train

One of the improved performance features of our new drive train design is the use of higher speed 12-volt electric motors. The new Fisher Price Hot Wheels toy gearbox motors are capable of carrying 70 lbs. at speeds up to 5 mph (as compared to 1mph last year). AISSIG uses 2 motors to provide 10 ft-lbs of torque per wheel at 144 RPM with a 111:1 gear ratio gearbox and a stall current of 15 amps. The actual speed of the vehicle is limited to 5mph using software control. Each motor has independent speed and direction control. It is directly coupled to a wheel on a fixed axle and differentially steers the robot giving it zero turn radius.

Wheel design



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. Since the plastic wheels did not provide enough traction, we added a rubber sheet to it as shown in fig 2.

Fig 2: Wheel design for traction

Electrical and Electronics

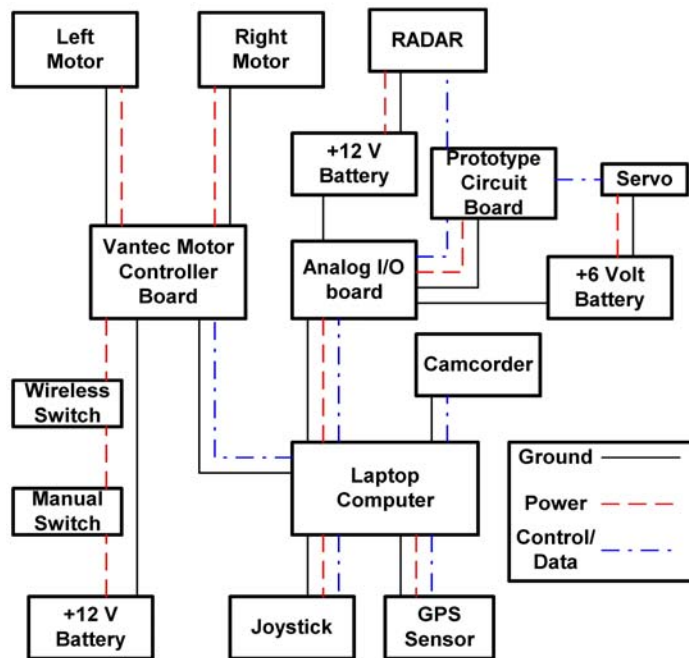


Fig 3: Electrical/Electronics Block Diagram

Motor Control

The Vantec CDFR-21 motor controller was quite reliable and effective for the 2003 robots and we continued to use them for 2004. The motor controller is interfaced to the main robot computer via the parallel port. The speed and direction of the motors are controlled by sending commands to the parallel port. 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, which is more than

sufficient to control our robots. An improved use of the Vantec CDFR-21 for the 2004 robots is that we increased the PWM rate from 338 Hz for the 2003 robot to 21.6KHz rate for the 2004 vehicle. This increases the speed control resolution and gives the robot control software finer control over the motor speed, which greatly improves the turning and speed precision.

RADAR

AISSIG uses a 24 GHz pulsed radar. It has a range of 20 meters and a beam width of 15

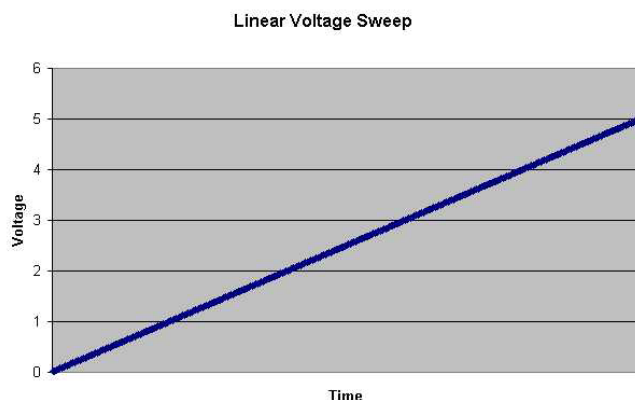


Fig 4: Linear Voltage Sweep

degrees. The radar is fed with a linear voltage sweep (see figure 4), which enables the radar to scan out in distance. A small circuit built up on a prototype board generates the linear voltage sweep. The data returned by the radar is I and Q data where the combined amplitude of the two signals will show if there is an object at that range. The raw data can be seen in figure 5. To get the combined data, you take the

magnitude of the I and Q data together. This is calculated by centering the I and Q data around 0, and then squaring each I and Q sample and adding them together. This can be seen in figure 5. The higher the A/D sample number, the farther the object is in range.

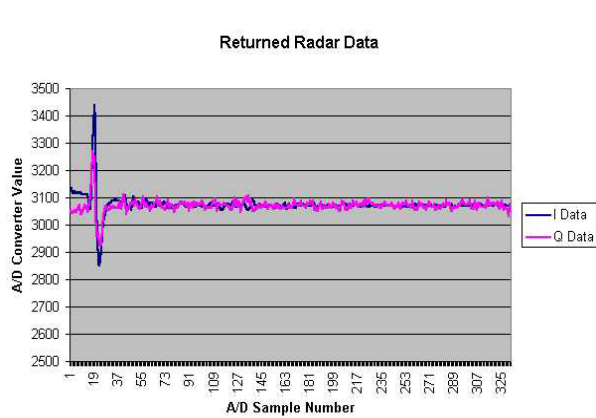


Fig 5:Raw Data from Radar

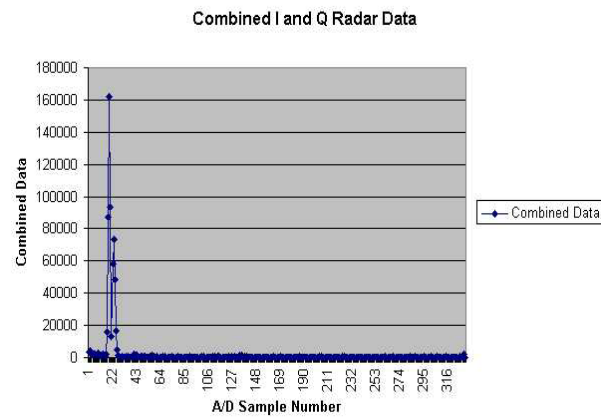


Fig 6:Combined I & Q Radar Data

Vision Sensors

The primary sensor used by AISSIG is a vision sensor provided by a Sony Digital 8 camcorder with an IEEE 1394 computer interface. This camcorder is used to detect lanes, potholes, and obstacles. A .5X wide-angle lens was successfully used on the camera to provide an increased field of view at a lower cost than using additional cameras. An increased field of view increased the amount of visual data available to the control software and hence improved the accuracy of the lane following algorithm. Adjusting the camera tripod mounted on the robot can make additional field of view adjustments. The adjustable iris on the camcorder provided a better adaptability to changing lighting conditions like ramp reflections and shadows. An IEEE 1394 interface increased the camera data bandwidth available and reduced the previous years web cam data bottleneck.

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. The GPS unit provides a wide range of features at a reasonable price. The built in features of the eTrex GPS unit provides all the functions necessary to navigate the robot. It allows us to set the destination coordinates and continuously provides feedback indicating the direction and distance to the desired waypoint. The eTrex unit is interfaced using the RS-232 serial port and uses one of the various protocols built into the etrex unit for data communication. The advertised accuracy of the GPS unit in WAAS mode is less than three meters, which is the actual accuracy of the vehicle.

Power System and E-stop

Figure 3 shows the schematic diagram of our robot vehicle's electrical power system. A 12 volt 7 amp hours sealed lead acid rechargeable battery produces main electrical power. These

batteries can power the vehicle for 3 hours. Actual lifetime of battery is lower due to the fact that batteries cannot sustain full discharge. A backup battery is always charged and ready to replace the weak battery. Other notable safety features of the robot’s electrical system is the resetting circuit breaker panel which provides 15 amp over current protection for each motor and the 30 for the overall system.

As per IGVC guidelines, a remote and mechanical emergency stop switches are provided to stop the vehicle. A low cost automotive keyless entry switch with a 50 feet range provides e-stop. 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 system is that the main power supply can only be restored using the mechanical push-pull switch even if it is stopped by remote switch. This prevents the robot from being restarted until a person is in physical proximity and can decide to restart the robot.

Software Design for Autonomous Challenge

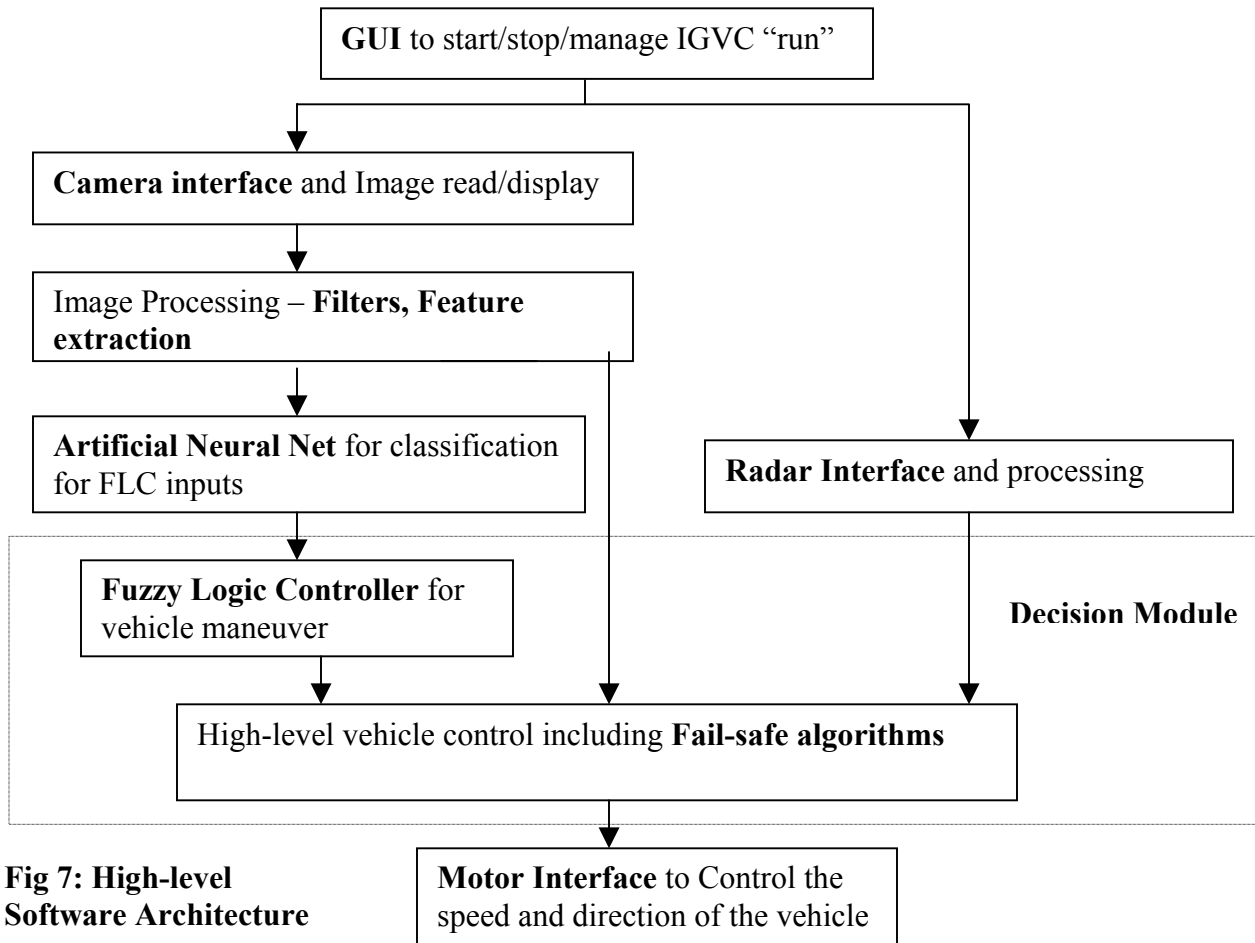


Fig 7: High-level Software Architecture

AISSIG software consists of many components that work together. Fig 7 outlines the software architecture for AISSIG's Autonomous challenge.

User Interface Design for the AISSIG software

The IGVC software is showcased under a systematic GUI module for better management and control of all the modules. Good UI also helps in testing and troubleshooting. A sample screen shown below provides an insight of the UI.

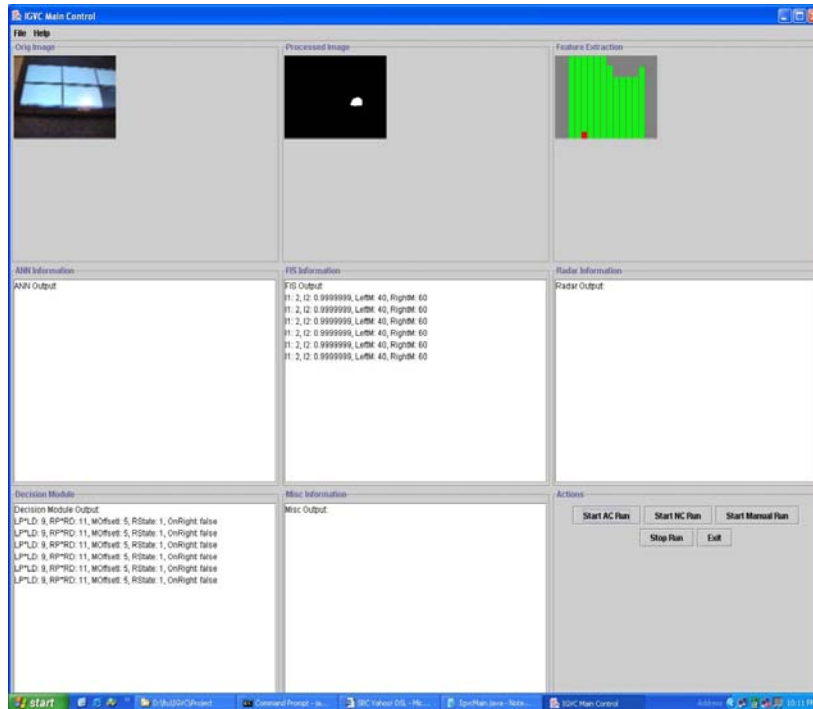


Fig 8:Graphical User Interface

collecting training data for ANN.

Vision Processing

Vision processing techniques are employed by AISSIG to detect the lane boundaries, an obstacle-free path and the surface of the path for better vehicle control using software.

The image from the camcorder (vision sensor) is processed in order to produce a compact representation of the obstacle free region within the lanes. This area represents all potential paths the vehicle can travel within the constraints of the competition. The processing is done in discrete steps as described below:

The UI was developed using Java Swing components. It has controls to start and stop the Autonomous and Navigation Challenge and another control to put the vehicle in manual control mode. There are many text area objects to show the various statuses during the run. The vehicle has a Manual Control Module to transport the vehicle manually using a joystick control. This helps in moving the vehicle around and also

- Video Capture using JavaDV: JavaDV (www.tomburke.net/JavaDV) serves as the capture interface between our Java environment and the camcorder, helping in grabbing the image frames.
- Low-Pass Filter Pre-Processing: A low-pass filter is then applied to the video image. This reduces noise in the image that would compromise the fidelity of our results.
- White / Orange region detection: The pre-processed image is then examined, pixel-by-pixel, to determine which pixels represent lanes and obstacles. If the color of the pixel is known to be in the range of colors attributed to lanes and obstacles, it is tagged. A new bitmap containing only the obstacles and lanes are created.
- Feature Vector creation - Array of floats created to represent clear path: The obstacle / lane bitmap is segmented into 16x12 pixel regions that are assigned a Boolean based on whether or not some threshold of white pixel count is met. This 16x12 grid of Booleans is then converted into an array of 16 float elements. Each element represents the relative distance to a lane or obstacle at a particular location in the camcorders field of view. This is the compact representation of the obstacle free region within the lanes that is used in the further processing and decision-making.



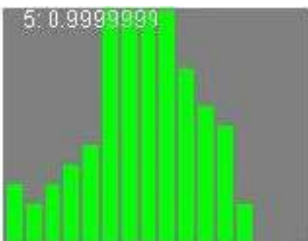
		
Image after low-pass filter	Image after white/orange region detection	Graphical representation of the feature vector - an array of floats used to represent possible paths for the robot to take.

Table 1: Image Processing stages

Surface Texture Classification

AISSIG exhibits different behaviors when traveling on grass, sand, or the ramp. The Texture classification module determines which surface the vehicle is currently on. It takes a 16 x 16 grayscale bitmap as input, transforms that bitmap into frequency space via a Fourier transform, samples that transform into a representative sample of 4 real numbers, and sends this sample to a Artificial Neural Net for classification.

The coefficient values in different regions of the Fourier transform will differ greatly depending on the amount of high frequency detail in the image. Image texture of grass, with lot of detail, will show Fourier coefficients with relatively high magnitude in the high frequency region and coefficients with low magnitude in the low frequency region. Sand, having an image texture with

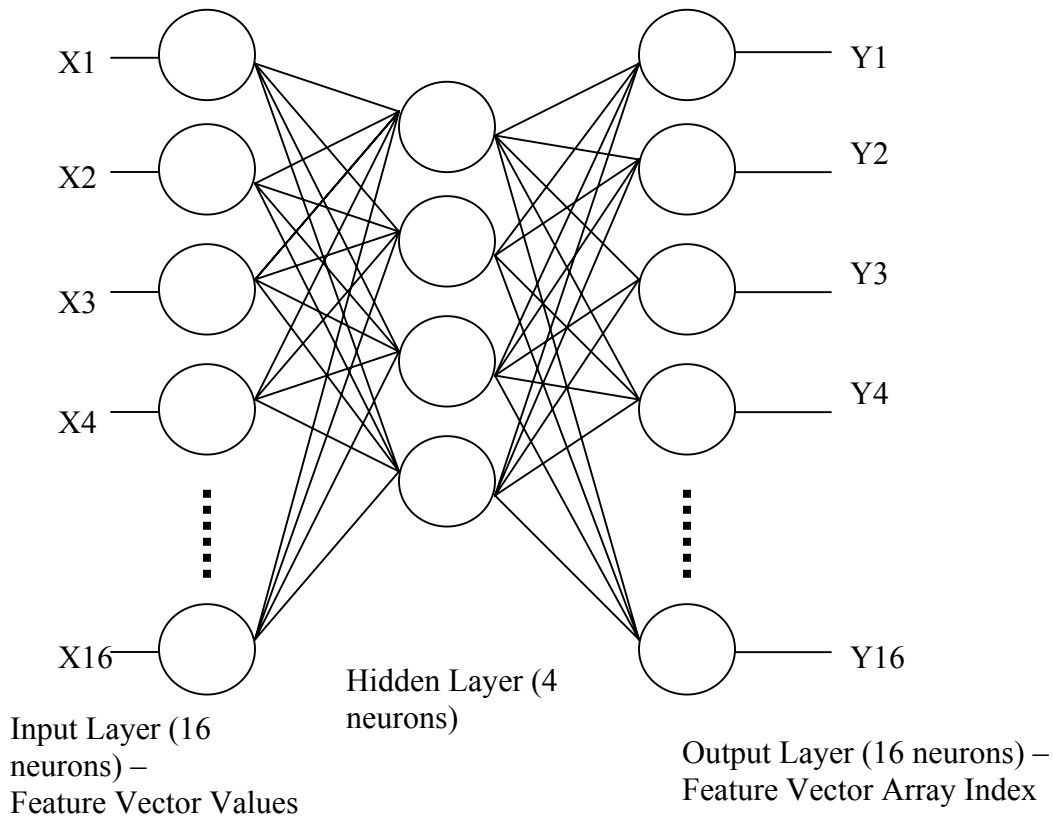
much less detail, will exhibit Fourier coefficients with low magnitude in the high frequency region, and coefficients with high magnitude in the low frequency region. The table below shows images representing the state of a grass and sand sample at each stage of the surface classification process.

Frame		
16 x 16 Sample		
Fourier Transform		
Neural Net Input	[4.86, 0.42, 0.29, 0.15]	[2.93, 2.46, 2.03, 0.65]
Neural Net Output	Sand	Grass

Table 2: Texture Classification illustration

Artificial Neural Net for Feature Vector Classification

The input from the Vision processing module comes in the form of a Feature Vector Array (As shown in the figure 10). The 16 feature vector values are used as inputs to the ANN and fed to the input neurons (as shown in fig 9).



As an example from feature vector in fig 10, Y4 = 1, and all other outputs = 0

Fig 9: ANN schematic diagram

The “Index of Feature Vector Array” represents the “Open Space” in the image frame, where the vehicle needs to steer to. The corresponding “Feature Vectors’ Value” represents the degree of openness in the image frame. For example, if the Vector Value is 1, that represents fully open area. A lower value represents some obstacle/Lane in that particular area. The array index and its value goes as inputs to FLC.

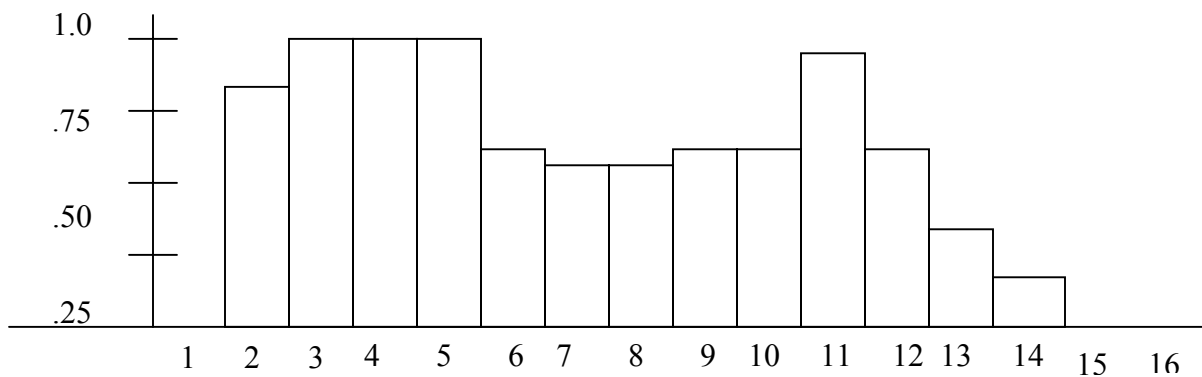


Fig 10: Feature Vector representation

ANN implementation works in two stages. First stage is to train the ANN, so called the “learning phase”. This is done by collecting sample feature vector data collected by manual runs and stored in a text file. The joystick-controlled interface is used to assist in this process.

A human expert then determines the desired output for each training sample. These training samples are then used to calculate the weights for each neuron connector. The weights for each neuron connector is calculated using Evolutionary Computation that uses ES(1+1) evolutionary strategy with 1/5th rule. During each iteration of evolution, the program calculates the error margin of ANN against the desired output. If not within the tolerance, a new ANN is evolved for next iteration. Once trained not only can the ANN recollect the outputs for the inputs that was used for training, but also those inputs that the ANN has not “seen” before. Accuracy of ANN depends highly on the training data set used during the “learning” process.

The example below shows a typical Feature Vector from the processed image. The 16 inputs to the ANN would be feature vector values (y-axis) for each array element. The output of ANN is the index of array which is most open, in this case “4”. The inputs to FLC inputs will be this Feature Vector Index (4) and Corresponding Feature Vectors’ Value (1.0)

Decision Module - Design of Fuzzy Logic Controller:

The output from ANN goes into the FLC as inputs, which calculates the speed and direction of vehicle to successfully maneuver through the lanes and obstacles. FLC outputs, along with Radar interface, and High-level controls with fail-safe mechanisms, together form the Decision module. The FLC is based on Sugeno-model.

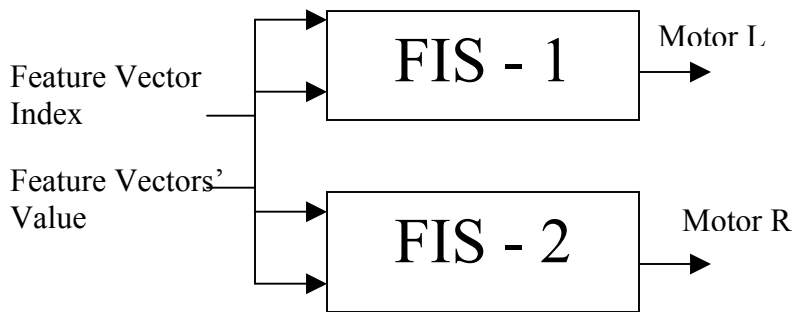
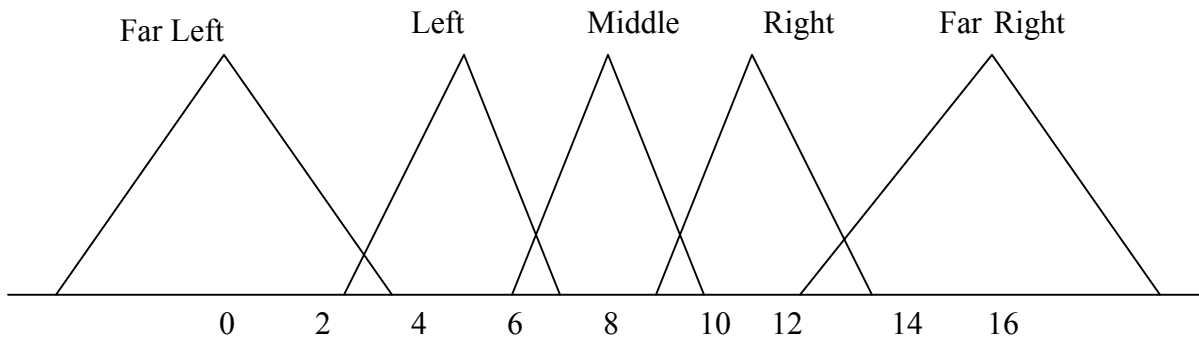


Fig 11: FLC to assist Decision Making

Input 1 – Feature Vector Index from ANN (Desired Center Position)



Input 2 – Feature Vector Value from ANN (Turn Factor)

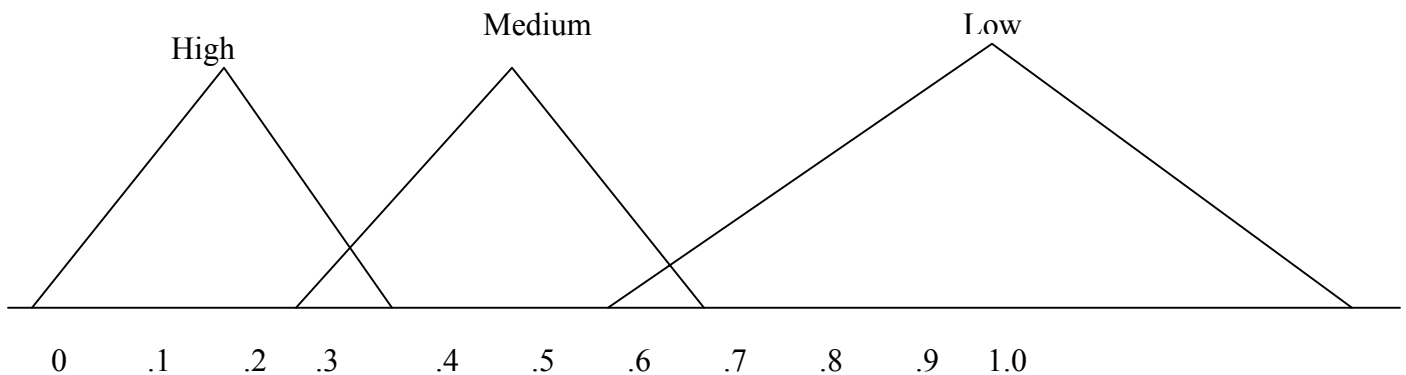


Fig 12: FLC – Fuzzy Sets for Inputs

FLC Rules:

These are the rules the Fuzzy Logic Controller uses to fuzzify the inputs and generate appropriate de-fuzzified outputs.

Vector Value Vector Index	High	Medium	Low
Far Left	Strong Sharp Left	Medium Sharp Left	Weak Sharp Left
Left	Strong Left	Medium Left	Weak Left
Middle	Slow Straight	Medium Straight	Fast Straight
Right	Strong Right	Medium Right	Weak Right
Far Right	Strong Sharp Right	Medium Sharp Right	Weak Sharp Right

Radar Interface

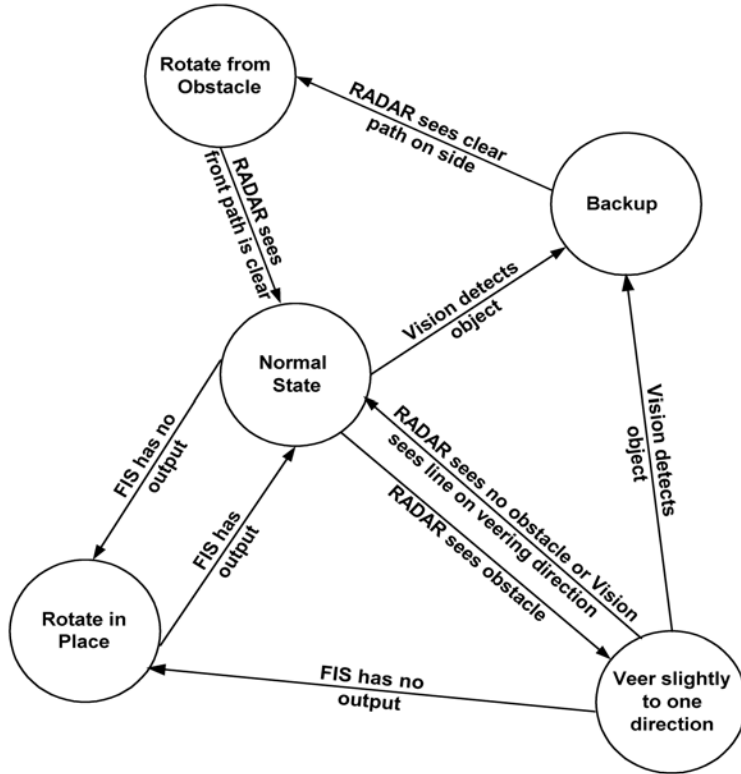
The radar is mounted on a servo, which allows the radar to scan about 100 degrees. The radar is reset to point forward at the start. If an obstacle is detected it will scan to either direction to find the extent of the obstacle. The information about open path is communicated to the decision module.

If the decision module goes into back up mode, the radar will point to one side or the other. The decision module will determine this side. The radar interface will then inform the decision module if the path is clear. Then the radar will point forward as the vehicle rotates in place. When the front of the vehicle is clear the radar interface will inform the decision module. Instead of using multiple radars, the radar is mounted on the servo to avoid interference issues that could be caused on multiple radars.

Decision Module - Overall vehicle control

The overall vehicle control is a state machine that takes the output of the Vision processing functions, the FLC, and the output from the Radar interface as inputs. The output is the power value to send to the motors. The state transitions can be seen in the figure 12. In the “normal state”, the decision module takes the output from the FLC and uses that to set the motor speed. When the vision data shows that there is an object directly in front of the vehicle (undetected by FLC), the decision module will start to move the motors in reverse. When the radar sees a clear path on the side of the vehicle, it will move to the “Rotate from obstacle state”. The decision module will cause the vehicle to rotate until the radar sees a clear path. Once a clear path is

detected the robot will go back into normal state. If the radar detects an object in the distance, it will try and determine, which side it is better to pass the object from. The decision module will



then add a slight bias to the output of the fuzzy logic controller to steer it towards the side it has deemed safe. Once the object is no longer detected, or if the vision sees a lane on the side vehicle is veering towards, the decision module will go back to the normal state. If the fuzzy logic controller produces no output, the decision module will rotate in place until the fuzzy logic controller produces output again. The “veer slightly state” will have the same actions as the normal state, the only difference being the slight bias it adds to the motors to steer clear from an obstacle.

Fig 12: State Transition Diagram for vehicle control

Software Design for Navigation Challenge

Once the map is given, we manually assign the waypoints in the order to be traversed (based on shortest possible distance). Then manually, connect the waypoints & calculate the angle between line of source-target waypoint and “North (y-axis)” as shown. Also calculate the shortest distance between each source-target waypoint pair (straight line).

In the example below: calculate angles a, b, c... and distance x, y, z... and so on.

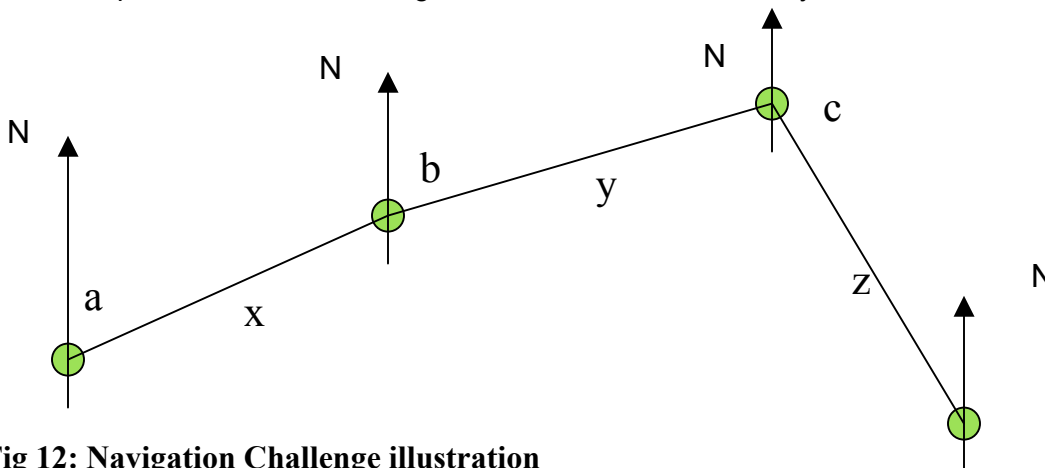


Fig 12: Navigation Challenge illustration

When the vehicle reaches a waypoint or is at the start, program the vehicle to rotate so that it faces “North”. Then, let the vehicle turn the required angle (a, b, c... in the previous example) Set the vehicle to motion. Poll for the vehicle position (or differential with target) every 5 seconds or so & compare against the shortest distance (x, y, z... in the previous example). If differential is greater than the shortest distance and some threshold, reposition the vehicle to face the target and start again.

Use radar data to avoid the obstacles and circumvent them. Current plan is to not use the camcorder for Navigation challenge to reduce processing and have better response time. This module is currently under development and we may make some design adjustments depending on the test results. **Summary**

Listed below are the tables of key specifications and performance statistics and expense to build AISSIG.

Vehicle Specifications Property	Specification Value	
Length x Width x Height Weight Motor Voltage Motor Stall Current Torque per Wheel Gear Ratio Wheel Diameter Wheel RPM	36" x 16" x (19" – 69") 40 lbs. 12V 15A 10 ft.-lbs 111:1 ~10" 144	
Vehicle Performance Property	Performance Prediction	Performance Results
Max. Speed Ramp Climbing Ability ($\mu = .3516$) Reaction Time Battery Life Object Detection Dead-ends and Traps Waypoint Accuracy	5 Mph 58 Degree Incline 1.75 hours < 10 feet	4.3 Mph 18 degree Incline .7 seconds 1 hour 3.6 feet ? < 10 feet

Vehicle Cost

Component	Purpose	Retail Cost	Cost to Team
Vantec CDFR-21	Motor Driver Board; provides PWM and directional control	\$250.00	\$250.00
2 – 12 VDC Motors	Integral component of the drive train	\$30.00	\$60.00
2 - 0.25" Particle Board	Robot Base	\$6.00	\$6.00
3 - 10" Wheels	Drive Train	\$8.00	\$24.00
Sony Digital 8 Camcorder	Lane Detection, Obstacle detection, Ramp, Grass, Sand-pit classification	\$500.00	\$0.00
Wide-eye Lens	To increase vision range	\$30.00	\$30.00
Camcorder Tripod stand	To mount the camcorder	\$30.0	\$30.00
Radar Sensor	Secondary Obstacle detection	\$100.00	\$0.00
Servo Motor	To set radar in rotary motion	\$30	\$0.00
Wireless Joystick	For transportation and ANN training	\$50	\$50
Garmin GPS 76	Data for Navigation	\$150.00	\$150.00
Sony Vaio Laptop	Software/Computation	\$1400.00	\$0.00
Miscellaneous	Connectors, batteries...	\$100.00	\$100.00
Total		\$2730.00	\$700.00

Conclusion

LTU is competing in IGVC for the second time with 4 entries. Our primary focus has been the software that controls the vehicle, and that's where the 4 entries primarily differ from each other. AISSIG software uses Artificial Intelligence techniques like Evolutionary Computation, Artificial Neural Net, and Fuzzy Logic Control. AISSIG will be competing in Navigational Challenge this year for the first time. AISSIG has been tested successfully to follow the lanes and avoid obstacles of different configuration.

Some of the areas where AISSIG can be improved are to have better drive system like using Chain or Treads to increase traction for stability in ramp climbing and sandpit.