I, Anup S Deshpande, hereby submit this work as part of the requirements for the degree of:

Master of Sciences

in:

Mechanical Engineering

It is entitled:

Computer Joystick Control and Vehicle Tracking System in Electric Vehicles

This work and its defense approved by:

Chair: Dr. Ernest L. Hall

Dr. Manish Kumar

Prof. Dan Humpert
Computer Joystick Control and Vehicle Tracking System in Electric Vehicles

A thesis submitted to the
Division of Graduate Studies and Advanced Research
of the University of Cincinnati

in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in the Department of Mechanical, Industrial and Nuclear Engineering
of the College of Engineering and Applied Science

August 2010

by

Anup S Deshpande

Bachelor of Engineering, Government College of Engineering, Aurangabad – MH, India

Thesis Advisor and Committee Chair: Dr. Ernest L. Hall
Abstract

The desirability of developing an autonomous vehicle and the rising demand for efficient use of energy in automobiles motivate the research on optimum solution to computer control of energy efficient vehicles. This thesis work describes three control methods - mechanical, hydraulic and electric that have been used to convert an electric vehicle into a ‘drive by wire’ vehicle using computer control. It also describes a vehicle tracking system used to track the route taken by the vehicle. Computer interfacing and control of basic automobile operations like steering, braking and speed have been implemented and will be described in detail in various chapters. A computer system with a joystick and a Galil three axis motion controller are used for this purpose. The motion controller is interfaced with a computer software program on the input side and with actual hardware (speed motors, steering system, and braking system) on the output side. WSDK (Windows Servo Design Kit) serves as an intermediate tuning layer between the motion controller and the computer program for tuning and parameter settings. The software program for this is developed in C#.NET. Voltage signals sent to the motion controller can be varied through the software program to control the steering motor, activate the hydraulic brakes and vary the vehicle’s speed.

This vehicle is now available as a test bed for research work on various autonomous operations and has been configured for its basic functionality including street legal operations. A 1000 mile test while running in a hybrid mode has also been conducted successfully. The vehicle was also tested in computer control mode with a keyboard and joystick as input devices. Currently the vehicle is being tested in various safety studies and is being also used as a test bed for experiments in control courses and research studies. The significance of this research work lies
in providing a test resource for autonomous operations, for disabled mobility studies and for
greater understanding of conventional driving controls.
Acknowledgements

I would like to thank Dr. Hall for his guidance and encouragement throughout my graduate studies and research work at the University of Cincinnati. I am grateful to him for his support during this period and for the knowledge that I could gain by working with him and other fellow students from the Robotics lab and 2007 DARPA team members. I would also like to thank Dr. Manish Kumar and Prof. Dan Humpert for their help during my thesis and research work.

I am also thankful to the faculty of University of Cincinnati, especially College of Engineering and Applied Science for their direct or indirect support during my graduate studies.

Finally, I would like to express my gratitude towards my family, friends and everyone because of whom I could successfully complete my studies at University of Cincinnati.
# Table of Contents

Chapter 1: Introduction ............................................................................................................................. 1

Chapter 2: Literature Review ..................................................................................................................... 4
  2.1 Autonomous vehicles .................................................................................................................. 4
  2.2 Drive-by-wire cars .................................................................................................................. 5

Chapter 3: Overview of Control Systems .................................................................................................. 6
  3.1 Introduction to control systems .......................................................................................... 6
  3.2 Open loop systems .............................................................................................................. 6
  3.3 Closed loop systems ............................................................................................................ 7
  3.4 Servo control systems ......................................................................................................... 8
  3.5 Servo control system elements .......................................................................................... 8
  3.6 Digital control servo system .............................................................................................. 10

Chapter 4: Computer Control .................................................................................................................. 12
  4.1 Computer control of steering system ............................................................................... 13
    4.1.1 Working of the computer controlled steering system ................................................ 15
    4.1.2 Tuning of servo control system ................................................................................. 16
  4.2 Computer control of brakes .............................................................................................. 19
    4.2.1 Design modifications in existing braking system ..................................................... 20
      4.2.1.1 Hydro-mechanical system ................................................................................. 20
      4.2.1.2 Design modifications ....................................................................................... 22
    4.2.2 Implementation of computer control of brakes for front wheels ................................ 22
      4.2.2.1 Interfacing of front wheel brakes with computer control system .................... 23
      4.2.2.2 Additional brake controller for manual braking ............................................. 24
List of Tables

Table 5.1 Joystick axis Z values ................................................................. 39
Table 5.2 Steering values for different joystick axis positions ......................... 41
Table 5.3 Speed values for different joystick axis Y positions .......................... 43
Table 5.4 Speed values for different joystick axis Y positions .......................... 46
List of Figures

Figure 3.1 Open loop control system ................................................................. 6
Figure 3.2 Closed loop control system ............................................................. 7
Figure 3.3 Command signal, feedback signal and resultant signal ...................... 8
Figure 3.4 Elements of servo system ............................................................... 9
Figure 3.5 Functional elements of a computer controlled servo system ............... 10
Figure 4.1 Computer control of basic operations ............................................. 13
Figure 4.2 Computer control of steering mechanism ....................................... 14
Figure 4.3 Photo showing the motor, gearbox and the steering column attached to the rack and pinion steering mechanism ................................................................. 16
Figure 4.4 Start-up menu of WSDK 4.04 ......................................................... 17
Figure 4.5 Tuning methods ............................................................................ 18
Figure 4.6 Step response for given values of $K_p$, $K_i$ and $K_d$ ......................... 19
Figure 4.7 Schematic diagram of a dual-diagonal disc braking system ............... 20
Figure 4.8 Hydraulic control cylinders ............................................................ 21
Figure 4.9 Signal flow for computer control of brakes .................................... 24
Figure 4.10 Wiring diagram showing manual brake controller operation ........... 24
Figure 4.11 Wiring diagram for brake control ............................................... 25
Figure 4.12 SPDT switch in dashboard for brake control ................................. 26
Figure 4.13 GE SX Speed Controller schematic ............................................ 27
Figure 4.14 Circuit for manual speed control ............................................... 29
Figure 4.15 Relationship between accelerator pedal movement and resistance .... 29
Figure 4.16 SPDT switch in dashboard for speed control ................................ 30
Figure 4.17 Computer control of speed ......................................................... 31
Figure 4.18 Relationship between resistance, voltage and speed ..................... 32
Figure 5.1 Block Diagram showing the design ................................................................. 35
Figure 5.2 Model - View - Controller Design Pattern .................................................. 36
Figure 5.3 Main screen of the joystick control program .................................................. 37
Figure 5.4 Joystick with different control axes ............................................................ 38
Figure 5.5 Speed control monitoring on virtual dashboard ........................................... 47
Figure 5.6 Brake control monitoring on virtual dashboard .......................................... 49
Figure 5.7 Brake and speed control monitoring on virtual dashboard ......................... 50
Figure 5.8 Steering control monitoring on virtual dashboard .................................... 51
Figure 5.9 Steering control screen when the vehicle is about to take a right turn .......... 52
Figure 5.10 Working of Vehicle Tracking System Module ........................................ 55
Figure 5.11 Route generated by the Vehicle Tracking System .................................... 56
Figure 5.12 Route generated by the Vehicle Tracking System in MS Excel format ....... 56
Figure 6.1 Testing the vehicle ....................................................................................... 57
Figure 7.1 Block diagram of complete computer control system ............................... 58
Chapter 1: Introduction

An experimental ‘all terrain vehicle’ was modified to ‘drive by wire’ for the 2007 DARPA Urban Challenge and now serves as a test bed for further research in energy efficiency, computer control of hardware, safety and educational purposes. To be street legal, the vehicle was adapted to have the required safety glass windshield, head, tail and turn signal indicator lights, followed by an inspection by the State Patrol. The successful inspection then permitted the registration, license and insurance of the vehicle. The conversion of the electric vehicle to drive by wire mode required several design level changes in the steering system, speed control functionality and braking system. Since the highest speed reached by this 2100 lbs vehicle is 45 mph with its two 7.5HP electric motors, safety was one of the of primary concern in this project. To ensure this safety, an emergency stop easily accessible to the driver and a back up signal were added to the vehicle. A wireless remote emergency stop was also made available for further safety and a chase vehicle was always used during road testing.

The motivations behind the study and research work in this field are as follows

1) Choosing a fully electric vehicle for implementing computer control is a step towards energy efficiency and environmental awareness.

2) Computer control of basic automotive operations can help in exploration of new ways for computer control of other automotive operations.

3) Computer control of automotive operations with further research work to address existing unresolved issues in this field is an effort towards providing driving capabilities for people with physical limitations.
4) Implementation of mathematical models like Artificial Neural Networks (ANN) and computer vision algorithms can also be used to detect faults and assist users with obstacle avoidance.

5) Advanced object oriented programming techniques introduce greater scope in development of efficient hardware communication, user controls and HMI (Human Machine Interaction).

6) Electric and hybrid vehicles are the future of eco-friendly automobiles and robotics is the science for developing modern day automation systems. The combination of these technologies can result in new solutions to a variety of automotive applications.

The study of autonomous operations in automobiles can improve the reliability, both in terms of the safety and precision of its operation as well as efficient use of energy. The problem of autonomous vehicle control may be modeled as a dynamic programming problem. However, the gap between theory and practice is still rather large. Various efforts have been made to minimize this gap, which include experimental studies conducted at the Intelligent Ground Vehicle Contest or IGVC (www.igvc.org) that has been held since 1993 and has resulted in many technical advances on small sized unmanned ground vehicles (UGVs). Research on larger experimental vehicles has been motivated by the DARPA Grand Challenges in 2004 and 2005 and the Urban Challenge in 2007. Thrun describes the winning vehicle for the 2005 Grand Challenge. Urmson, et al. describe the 2007 winning team and vehicle.

This thesis work describes three computer control methods for electric vehicles – mechanical, hydraulic and electrical. For definiteness, an example of an experimental “self constructed” vehicle is used which was modified for various design requirements at the Robotics Research Center, University of Cincinnati.
In Chapter 2, important research work done in past in this area is discussed. Chapter 3 gives an overview of control systems. Chapter 4 describes the actual implementation work of this project. A discussion of the design changes implemented for computer control, implementation of computer control of steering, speed and braking operations is given. The software program developed in C#.NET for joystick control and its communication with the motion controller and vehicle tracking system is described in Chapter 5. Finally, experimental results have been discussed in Chapter 6 and conclusions and future recommendations have been discussed in chapter 7. The Appendix contains a listing of the design work done for computer control of other systems in the vehicle.
Chapter 2 : Literature Review

2.1 Autonomous vehicles

Autonomous and computer controlled vehicles have been the topic of research in Japan, Germany and US since the seventies. The Tsukuba Mechanical Engineering Lab in Japan conducted a test of an autonomous vehicle on a dedicated, clearly marked course and it achieved speeds of up to 30 km/h (20 miles per hour), by tracking white street markers\(^2\). In the 1980s a vision-guided Mercedes-Benz robot van in a similar test at the Universität der Bundeswehr in Munich, Germany, achieved 100 km/h on streets without traffic\(^2\). In 1995, the Carnegie Mellon University ‘Navlab’ project reported 98.2\% autonomous driving on a 5000 km trip\(^2\). The car had steering wheel controlled by neural networks. However, the brakes and throttle were still human controlled\(^2\).

Recently, owing to the DARPA Grand Challenge, a lot of research has been done on autonomous vehicles at various universities. The 2004 and 2005 DARPA competitions allowed teams to compete in fully autonomous vehicle races over rough unpaved terrain and in a non-populated suburban setting\(^3\). The latest 2007 DARPA challenge involved autonomous cars driving in an urban environment\(^3\). Companies like Mercedes and General Motors are also involved in active research in the field of autonomous vehicles. General Motors has also stated that they plan to start testing their autonomous vehicles by 2015 and can be on road by 2018\(^2\).
2.2 Drive-by-wire cars

All the major car manufactures in the world have started introducing their models with drive-by-wire technology since late nineties. Mercedes, with their Benz F200 Imagination Concept introduced this technology in 1996. Little joysticks, called ‘sidesticks’ in the doors and the center console for steering and braking replaced the steering wheel, brake pedal and accelerator pedals\textsuperscript{14}.

In 1995, BMW began their research work on drive-by-wire technology (under the project Z22) to determine its feasibility in controlling a vehicle electronically. Z22 became visible to the outside world only after 2000\textsuperscript{15}. 
Chapter 3 : Overview of Control Systems

3.1 Introduction to control systems

A control system is a device or set of devices used to control, command, direct or regulate the behavior of another device or system of devices\(^8\). It can also be defined as a combination of components (electrical, mechanical, thermal, or hydraulic, electronic) that act together to maintain actual system performance close to a desired set of performance specifications\(^9\).

Control Systems has its applications in almost every industry from construction, transportation and logistics, manufacturing, to defense, aerospace, robotics, home appliances, and telecommunication. Depending on mode of the control, control systems can be mainly categorized in to open-loop systems and closed-loop systems.

3.2 Open loop systems

![Open loop control system](image)

Figure 3.1 Open loop control system

Open loop systems are systems in which no part of the output is provided as a feedback control signal to control the final system output. These systems use the input variable to obtain the desired system output using ‘the open loop transfer function’. As no output feedback is provided to the system, there is no output modification based on any system feedback. Open loop systems
are simple, quick in response and with low possibility of system oscillations. Examples of these systems are mobile phones and irrigation sprinklers.

### 3.3 Closed loop systems

Closed loop systems are systems in which a part of the output is provided as feedback signal to control the final system output. An additional measure of the actual output is used to compare the actual output with desired output. In other words, closed loop systems maintain a prescribed relationship between the system variables (i.e. reference input and desired output) by comparing functions of these variables and use the difference between values of these variables as means of control.

Closed loop systems find their application where there is a need for reduced sensitivity to parameter variations, stabilization of unstable systems, guaranteed performance, etc. Examples of closed loop systems are thermostats, cruise missiles, automobile engine governors, etc.
3.4 Servo control systems

A servo system is a feedback control system or a closed loop motor control system that combines various components to accurately control a machine’s operation depending upon the system feedback. The system is generally composed of a servo drive, a motor, and a feedback device. The controller receives an outside command signal (reference input) and a feedback signal (measured output) from the motor which get subtracted and a difference signal (measured error) is sent which controls the torque, velocity and position of the motor shaft.

The feedback continuously reports the real time status, which is constantly compared with the input command value. Differences between the command position and feedback signals are automatically adjusted by the closed-loop servo system to get desired output value. This closed-loop provides the servo system with accurate, high performance control of a machine.

3.5 Servo control system elements

A servo control system has various elements like controller, motor, amplifier, and encoder. In our system the controller will be a Galil DMC 1000 controller. Fig 3.4 shows these elements.
Motor: A motor converts the incoming current into torque which produces motion. The desired motion in our case will be steering of the wheels depending upon the input current with the extremes being a full left or full right turn of the steering column. In a motion controller, each axis of motion requires a motor sized properly to move the desired load at the desired speed and acceleration. The motor may be a step or servo motor, brush type or brushless, rotary or linear. For our application a rotary, brush type DC servomotor is used to steer the vehicle through required angle.

Amplifier: An amplifier is a device that modifies (usually increases) the amplitude of an incoming signal. In our circuit, the power amplifiers convert input voltage (which is +/-10 V) from the controller into current required to drive the motor. For best performance the amplifier should be configured for a current mode of operation with PID compensation.

Encoder: An encoder translates motion into electrical pulses which are fed back into the controller. Encoders typically provide two channels in quadrature, channel A and channel B. This type of encoder is called a quadrature encoder. The channels A and B are 90 degrees out of
phase and due to this relationship, the resolution of the encoder is increased to 4N quadrature counts/rev. (N is the number of pulses generated by the encoder per revolution).

**Controller:** A controller is the main element of a servo system. It accepts both input signals and encoder signals and finds the error between them. The error signal is then passed through different filters to obtain the output control signal which is then fed to the amplifier.

### 3.6 Digital control servo system

![Functional elements of a computer controlled servo system](image)

(a) block diagram (b) components of a controller

A typical computer-controlled servo system is shown in Figure 3.5. It has three main elements, a digital controller, an amplifier, and a motor with encoder as its integral part. The controller is a unit which consists of a digital filter, a zero-order-hold (generally called ZOH) and a Digital to Analog Converter (DAC). The main purpose of the controller is to stabilize or compensate the system which tends to become unstable because of the difference between desired output and
actual output. This compensation is usually achieved by adjusting the parameters of the filter in the controller. The controller accepts both encoder feedback and commands signals from the computer and finds the difference (error) between them. The error signal then passes through the components of controller i.e. PID digital filter, ZOH and DAC to generate control signals required to control the amplifier (AMP). The amplifier amplifies the input signals from the controller to generate the required current to drive the motor. The encoder is usually mounted on the motor shaft. When the shaft rotates, it generates electrical impulses, which are processed into digital position information by the encoder. This position information is then directly fed back by the encoder into the controller which uses this as a feedback signal to calculate the error.
Chapter 4: Computer Control

The computer control of three main operations (steering, braking, and speed) in the test vehicle was implemented using a motion controller and a software program to send commands to the motion controller. Motion controllers are the devices which control position and/or velocity of machines using hydraulic pumps, linear actuators or electric motors in our case. Motion controllers can take computer commands as input and provide electrical signals to generate the required torque in the electric motors as output. The basic idea in this design is to use motion controllers to provide the desired power output required to operate different systems in the vehicle.

A three axes motion controller (Galil DMC 1000) is interfaced with a computer software program on the input side and electric motors and hydraulic systems on the output side. A three axes motion controller implies that the motion controller can generate three different output signals. The voltage signal sent to the motion controller can be varied through the software program for desired results in controlling the steering motor, activating the hydraulic brakes and varying the vehicle’s speed. Input signals to the motion controllers will be sent using a joystick as input device. The motion controller output axes X, Y and Z will provide required output voltage to the brake control, speed control and steering control systems respectively. Figure 4.1 depicts this design.
4.1 Computer control of steering system

The computer control of steering system involves two main steps:

1) **Reading the current position of the motor shaft**

To know the magnitude and direction of torque to be generated by the steering motor, one needs to first read the current position of the steering motor shaft (and thereby, the current position of the steering column). This is achieved by using a servo control system with an encoder, which is an integral part of the motor. The encoder provides a continuous feedback of the position information to the controller to generate desired output.
2) **Generating the required torque**

Based on the position of the steering motor shaft, the controller determines the magnitude and direction of the torque required to turn the motor through the requested angle.

Figure 4.2 shows the circuit diagram for the computer control of the steering mechanism. As discussed above, it involves a servo motor system and a motion controller. The motion controller serves as an interface between the computer program, which provides the input signals and the servo system, which generates the required output by comparing it with the feedback signals.

![Figure 4.2 Computer control of steering mechanism](image)

The output from the motion controller is fed to a power amplifier which amplifies the control signal to generate the required torque. As discussed earlier, the controller has three elements, namely the PID Digital Filter, the Digital-to-Analog Converter (DAC), and the Zero Order Hold.
(ZOH). The Digital-to-Analog Converter (DAC) converts a 14-bit number to an analog voltage signal. The range of numbers for input is 16384 and the output voltage generated can be $\pm 10V$ maximum. The torque that needs to be generated by the steering motor to operate the steering system and therefore to effectively steer the vehicle is calculated by conducting a system analysis. It is found that a signal of $\pm 10V$ from the controller with proper amplification is enough to generate this torque. The computer program is designed to provide input signals for generating the required voltage output from the controller. The input device for the computer program can be either a keyboard or a joystick. However, the joystick axis can offer better and precise control of the steering mechanism.

### 4.1.1 Working of the computer controlled steering system

The C#.NET software program is designed such that it captures the joystick axes position every 100 milliseconds and creates a motion controller-interpretable command depending upon the position of joystick axis Z, which is the axis responsible for steering system. The motion controller executes these commands and generates the respective output signal for the power amplifier. The power amplifier amplifies the voltage signal from the controller to drive the steering motor. The encoder generates electrical pulses on the movement of the motor shaft. These pulses are processed into digital position information. This position information is then fed directly into the controller. The controller simultaneously accepts encoder feedback signals and computer commands on the basis of which it calculates the error between them. This error signal is then passed through the digital filter, ZOH and DAC to generate control signals to control the amplifier. The power amplifier then provides the voltage to generate the required torque.
To match the number of rotations of the motor shaft with that of the steering column for a given
torque value, the motor output is geared down to the required number of rotations using a
gearbox. For our purpose a gearbox with gear ratio of 10:1 was used (i.e. every 10 rotations of
steering motor shaft will rotate the steering column by 1 rotation). The gearbox is connected to
the steering column as shown in Figure 4.3 which also shows the motor and the steering column.

![Figure 4.3](image)

Figure 4.3 Photo showing the motor, gearbox and the steering column attached to the rack and
pinion steering mechanism

### 4.1.2 Tuning of servo control system

A major challenge faced while working with any closed loop system is tuning for stability and
good transient response. A stable closed loop system requires appropriate combination of
proportional, integral and derivative gains of a PID controller. The tuning of the system
parameters in our case was done using a software tool WSDK (Windows Servo Designer Kit)
provided by Galil Motion Inc. The analytical values of $K_p$, $K_i$ and $K_d$ which are the proportional,
integral and derivative gains respectively of the PID controller, were tuned and tested for stability in the real system with the help of WSDK tool, version 4.04.

Following screens show the steps involved in the tuning process.

![Figure 4.4 Start-up menu of WSDK 4.04](Galil Motion Control - Servo Design Kit)

(Galil DMC-1000 Manual Rev. 2.0xf, www.galilmc.com)

From the above screen, the ‘Tuning’ menu is selected to tune the system parameters.
One can choose ‘Manual Method’ to adjust the values of $K_p$, $K_i$ and $K_d$. The manual method is an iterative process in which different combinations of $K_p$, $K_i$ and $K_d$ are tried until one obtains a stable system response. After every execution, the program plots a Bode frequency plot for stability analysis (or the step response plot for transient analysis) for given combination of system parameters. The rise time and overshoots, if any, can be observed from the step response plot and the above steps can be repeated to get satisfactory values of $K_p$, $K_i$ and $K_d$.

In our case, a stable system response was observed at $K_p = 6$, $K_i = 0$ and $K_d = 64$ after trying various combinations of these parameters and adjusting them based on system overshoots from the step response plot. Fig 4.6 shows the step response plot for these values.
To make sure that the system gives a stable response consistently, the above values of $K_p$, $K_i$ and $K_d$ are passed to the motion controller through the software program every time the computer control mode is enabled.

### 4.2 Computer control of brakes

Since the 2100 pounds test vehicle can attain a maximum speed of 45 mph, it called for a sturdy and reliable braking system. The vehicle already had a manually operated mechanical hydraulic braking system. To implement computer control of brakes, a new trailer brake assembly was introduced which could be triggered by computer signals. For safety reasons, both manual and computer control systems were kept functional all the time.

The implementation of computer control of the braking system involved modifying the existing braking system for computer control and implementation of actual computer control
4.2.1 Design modifications in existing braking system

4.2.1.1 Hydro-mechanical system

The existing brake system on the jeep is a typical hydro-mechanical system with disc brakes on both the front and rear wheels. Figure 4.7 shows a schematic diagram of the existing dual-diagonal brake system on the vehicle.

![Figure 4.7 Schematic diagram of a dual-diagonal disc braking system](http://vnc.thewpp.ca/stuff/bentley/ep0niks.ctech.ca/vw/eva2/SU02/ch1.1.html)
The dual-diagonal braking system has a master cylinder with brake fluid, brake lines and disc brakes on each of the four wheels. The master cylinder, which is operated mechanically by the brake pedal, has two separate chambers. There are two different hydraulic circuits for right front and left rear wheel braking and for left front and right rear wheel braking. Each chamber builds up required hydraulic pressure in each of these two circuits depending upon the position of the brake pedal.

When the driver applies the brakes, i.e. when the brake pedal is pushed, the brake fluid is forced under pressure into the two master cylinder chambers which activates the two hydraulic circuits. The pressure is transmitted to the disc brakes on the respective wheels through the brake lines and generates the force required to press the brake shoes against the wheels causing the braking effect. The advantage of the dual–diagonal braking system is that in case of failure of one of the two hydraulic circuits, the other circuit will still be able to supply some braking force to one front and one rear wheel.

Figure 4.8 Hydraulic control cylinders
4.2.1.2 Design modifications

Since the vehicle was also going to be used as a test bed for various computer controls and autonomous algorithms, safety was one of the prime concerns. The idea was to have manual brake control active all the time even if the vehicle is under computer control. This is accomplished by having rear brakes always under manual control and the front brakes under computer control. This design makes sure that in case of failure of the computer control system, the manual control of brakes will still be possible.

To achieve this, the output of the brake lines running from the two chambers of the master cylinder were combined and fed to the two rear wheels and a new trailer brake system is introduced for front wheel braking. The trailer brake system can be activated by electric signals causing the front wheel braking. This design makes the rear wheel braking independent of any electronics and computer algorithm. So when the driver pushes the brake pedal to apply brakes, the hydraulic pressure is transmitted to the rear wheels causing rear wheel braking.

4.2.2 Implementation of computer control of brakes for front wheels

In order to implement computer control for the front wheel brakes, the following components were introduced into the braking system.

1. Brake Actuator: A Dexter DX series brake actuator with a 1600 psi rating is used which works on a 12V DC power supply.
2. Brake Controllers: Two Dexter Predator DX2 electric controllers are used. A brake controller comprises of an inbuilt manually operated lever which allows variable braking effort if required. It works on a 12V electrical system. One of the brake controllers is slightly modified and used for computer control. It receives output signals from the motion controller, amplifies them and provides the necessary signals to the brake actuator required for activation. The other brake controller is installed in the dashboard for additional safety and is operated manually.

4.2.2.1 Interfacing of front wheel brakes with computer control system

As mentioned earlier, the front wheel brakes are modified to be operated by computer control. The brake actuator gets actuated when it gets electric signals from the motion controller. As the voltage required for actuation of the brake actuator is more than the voltage supplied by motion controller, the signals from motion controller need to be amplified. This is achieved by modifying a brake controller. The electric signals required for actuation of brake actuator, which were originally generated by operating a manual lever, are now generated by electric signals from the motion controller. In other words, a manual brake controller lever is replaced by input signals from the motion controller. Thus the computer commands to the motion controller are converted into electric signals and supplied to the modified brake controller, which amplifies these signals and forwards them to the brake actuator. The actuator in turn pumps the brake fluid from a reservoir into the front brake lining thereby causing the hydraulic braking action. Fig 4.9 shows the signal flow from motion controller to brake actuator.
4.2.2.2 Additional brake controller for manual braking

While designing the computer control of brake system, probability of hardware and electrical system failure also needs to be taken into account. To cope with such unforeseen situations, an additional manual brake controller is made available on the dashboard of the test vehicle. This brake controller runs on a 12V electrical system and is connected to the brake actuator directly without any computer control. A schematic of this arrangement is shown in Figure 4.10.

When the brake controller lever is operated, it varies the voltage supplied to the brake actuator, varying the pressure on the hydraulic fluid that is pumped from the reservoir into the front brake lining. Thus the brake controller lever position determines the braking force that presses the brake shoes against the front wheels.
4.2.2.3 Working of the computer controlled braking system

Figure 4.11 shows the circuit for computer and manual control of the front wheel brakes. Terminals ACMDX and ground of the Galil motion controller breakout board are connected to the modified impulse brake controller. When the X axis of the computer joystick is operated, which is the axis responsible for brake control, the respective computer command is sent to the motion controller which then generates proportionate electrical signals in the range 0-5V. These signals are forwarded to the brake controller which amplifies them and sends a voltage signal ranging from 0-10V to the brake actuator to activate the hydraulic pump. The braking fluid in the hydraulic pump creates necessary hydraulic pressure and transmits it via the brake lining to the brake calipers and brake shoes.

Figure 4.11 also shows the circuit for manual brake controller. The manual brake controller, as described in the previous section also works in the same way except that the voltage required to
activate the brake actuator is generated by manually operating the brake lever. Both the brake controllers are powered by a 12 V battery supply.

To facilitate toggling between the two types of controls, i.e. manual and automatic, the output signals from the modified brake controller and the manual brake controller are routed to the brake actuator through a single pole double throw switch (SPDT).

Fig 4.12 shows the SPDT switch provided in the dashboard. When the switch is on automatic side, current flows through the modified brake controller, completing the circuit for motion controller signals. With this, the brakes are under computer control and can be operated by the joystick. When the switch is on manual side, current flows through the manual brake controller and the brakes can be operated manually. Irrespective of the position of the SPDT switch, control of the rear brakes by the brake pedal will always be possible.
4.3 Computer control of speed

The test vehicle is powered by two 7.5HP electric motors for speed which are controlled by a GE SX speed controller\(^7\). Just as in computer control of the braking system, the implementation of computer control of speed involved modifications in the speed controller circuit and actual implementation of computer control.

4.3.1 Existing speed control system

Figure 4.13 GE SX Speed Controller schematic

As mentioned earlier, the torque required to turn the wheels is provided by two 7.5HP electric motors each of which is controlled by a GE SX speed controller. Figure 4.13 shows a schematic diagram of the speed controller and the speed control circuit. The speed controller has a central circuit board called ‘OSC card’ and has 20 pins (from P1 to P20). These pins are connected to various switches and terminals in the electric circuitry of the vehicle. For example, the following functions are controlled by the corresponding pins: forward-reverse circuit terminals (P4 & P5), seat switch (P6), accelerator start switch (P3), regenerative braking potentiometer input (P13).

From system analysis we found that pins P7 and P9 are connected to a potentiometer and control the speed of the two traction motors. The accelerator pedal of the vehicle is connected to a variable resistor (or potentiometer) between pins P7 and P9. The terminal P9 is attached to the resistor, while the other terminal P7 (generally called wiper) is attached to a center tap of the variable resistor. When the wiper slides along the resistor, it varies the resistance in the circuit from maximum to minimum. The accelerator pedal or throttle is connected to the wiper (i.e. P7). Pressing the accelerator pedal slides the wiper over the resistor which increases resistance in the circuit as well as the speed of the vehicle, gradually. Thus at zero throttle position (zero speed), resistance in the speed control circuit is minimum and at full throttle (full speed), it is maximum. Figure 4.14 shows the circuit for manual control of speed.
The relationship between accelerator pedal movement (or speed) and variation in resistance is shown in figure 4.15. For simplicity in design calculations, the relationship is assumed to be linear.

4.3.2 Modification of existing circuit for computer control

As discussed in the previous section, pins P7 and P9 of the OSC card form a potentiometer system and provide required current to the two traction motors. These pins are connected to the
throttle through one end of a 6 pin connector whose other end is connected to the OSC card as shown in Figure 4.14. The remaining pins of this 6 pin connector are used for various other terminals like reverse forward switch, seat switch, etc. In manual control of speed, the voltage across P7 and P9 was found to be in the range of 3.5V (zero speed) to 0.5V (maximum speed). To implement computer control, the motion controller was required to generate a voltage signal in the above range by sending electrical signals to pins P7 and P9. For this, the throttle connections to pins P7 and P9 were disconnected and replaced by breakout board output axis Y and ground terminal respectively. The computer program is designed to send commands to the motion controller which can generate a voltage in the range of 3.5V to 0.5V. A Single Pole Double Throw (SPDT) switch is provided as shown in Figure 4.16 to be able to toggle between manual and computer control modes of the speed. Two inputs of the SPDT switch are connected to the throttle and the breakout board output axis Y while its output is connected to the pin P7 of SX controller through the 6 pin connector. There is also a common ground between the throttle and Galil circuit which is connected to pin P9 of the SX controller through the same 6 pin connector. By changing the position of SPDT switch, the desired speed control mode (automatic or manual) can be selected.

Figure 4.16 SPDT switch in dashboard for speed control
4.3.3 Working of the computer controlled speed

Figure 4.17 shows the circuit for computer and manual controls of speed. As discussed in the earlier section, to be able to toggle between manual and computer control modes, an SPDT switch is provided in the dashboard. The SPDT switch takes input from terminal P7 of the throttle circuit and output axis Y (ACMDY) of Galil breakout board. The output of the SPDT switch is given to the pin P7 of OSC card of the SX speed controller. The ground terminal is connected to P9 which is common for both the throttle and the Galil circuit.

When the SPDT switch is on the automatic side and joystick axis Y (axis responsible for speed control) is moved, respective computer commands are sent to the motion controller which then generates proportional voltage signals in the range of 3.5V to 0.5V depending on the actual joystick axis position. When the joystick axis is in neutral position, the computer program sends a command to the Galil motion controller to generate 3.5V signal (zero speed signal) and when
the joystick axis Y is in extreme position, it sends a commands to the Galil motion controller to generate 0.5V signal (full speed signal). All intermediate joystick axis Y positions send respective commands to generate a voltage which is less than 3.5V and greater than 0.5V. The Galil motion controller then passes these signals to the OSC card of speed controller through the SPDT switch and thus speed is controlled by joystick.

When the control mode is changed to ‘manual’, the Galil motion controller is out of circuit and the OSC card takes signals only from throttle circuit. In this mode of operation, when the accelerator pedal is pressed, the pin P7 connected to the wiper slides over the resistor P9, increasing the resistance, decreasing the voltage and increasing the speed gradually. The relationship between resistance, voltage and speed is shown in Fig 4.18 below.

![Figure 4.18 Relationship between resistance, voltage and speed](image-url)
Chapter 5: Software Implementation in C#.NET

5.1 System design

As discussed earlier, the joystick control program used for computer control of various automotive operations in our vehicle is the main interface between the motion controller (including all the connected hardware components) and the human operations. The joystick, which is connected to the computer system, takes commands from the user which are processed by the software program and forwarded to the motion controller. Depending on the type of the command, the motion controller then generates the required voltage signals for the appropriate hardware system (speed motors, braking system or steering motors). The software program for this is developed in C#.NET. The advantages of using the .NET framework include greater flexibility in developing the code required for hardware communication, features like custom controls, tools available in Visual Studio for developing interactive GUI etc. The program mainly uses Microsoft DirectX, a collection of Application Programming Interfaces (APIs) which is used in programming tasks involving multimedia communication, game programming etc.

The DirectX component is used to process data from input devices (joystick in our case) as it provides all the classes and functions required to develop communication between a C# program and a joystick. The software program is thus capable of acquiring a joystick, updating the status of the joystick, polling the joystick, etc. The classes for communication between the motion controller and the software program are developed using ‘DMCdll.net.dll’ provided by Galil. A DLL (Dynamic Link Library) file, also called a class library is a collection of classes and can be
included in a program reference to use the functionality provided by its member classes. The
DMCdnet.dll contains all the classes required to establish communication between a C#.NET
program and a Galil motion controller.

The C# program is divided into three functional modules which are designed to handle three
principal tasks as follows:

**Joystick Control Module** - This part of the program is responsible for establishing
communication between the program and joystick and uses DirectX API classes.

**Virtual Dashboard Module** - This part of the program is responsible for calculating and
displaying current speed of the vehicle (mph) and current braking pressure (psi) on the virtual
dashboard which is a custom control developed in C#.NET.

**Vehicle Tracking System Module** - This part of the program is responsible for logging the
motion controller commands into a database which are later plotted in the form of an XY line
chart to obtain the route covered by the vehicle.

Figure 5.1 shows the design of the system. There are three main layers defined, the user input
device layer, the software program layer and the hardware layer. As discussed earlier, the
software layer has three functional modules and a central module. Fig 5.1 also shows flow of
commands between different layers and modules of the system.
5.1.1 MVC Design Pattern (Model – View - Controller)

The design pattern implemented while designing the joystick control software system for our jeep is MVC design pattern (Model–View-Controller). This design pattern separates the code into three layers based on the responsibilities to be handled by that part of the code. MVC design pattern offers several advantages in the development of a multi module code. For example, it provides better capabilities for organization and reuse of the code. It also provides separation of different layers (presentation, data, and controller) in the code which makes code maintenance easier.
**Model** – The different modules that are responsible for different major tasks in the code are separated in model layer. In our program, the Joystick Control Module, Virtual Dashboard Module and Vehicle Tracking System Module represent the model layer.

**Controller** - The controller layer contains the code for responding to user actions, controlling the GUI (Graphical User Interface) and interacting with the model layer. Generally all the event handler code is placed in controller. The Main Module represents this layer in our code.

**View** – This is the presentation layer and is responsible for taking user inputs from GUI, displaying and updating GUI. All the user interface screens and dashboards in our program form the view layer. Fig 5.2 shows the MVC design pattern implementation for our program. It shows the separation of the code into module, controller and view.

![Figure 5.2 Model - View - Controller Design Pattern](image-url)
Fig 5.3 shows the main graphical user interface of the software program. The program offers different options like Speed Control, Brake Control, Speed and Brake Control where users can monitor joystick control of speed and braking operations along with their corresponding joystick signal values on the virtual dashboard. Users can choose to monitor both speed and brake control operations on the same screen by choosing the ‘Brake control and Speed Control’ option.

For safety reasons, the program is designed in such a way that the user has to enable the joystick by clicking the ‘Enable Joystick’ button every time the program is started. This ensures that in case any of the joystick axes are not in neutral position, starting the program will not immediately start sending joystick commands to the jeep. By clicking the same button the joystick can also be disabled.
5.2 Joystick Control Module

For our joystick control program, we are using a Logitech dual action gamepad joystick with 4 axes and 12 buttons. Fig 5.4 shows three axes on the joystick which control different operations. Operating X axis sends commands to the Galil motion controller to control brake; operating Y axis sends commands to the Galil motion controller to control speed and operating Z axis sends commands to the Galil motion controller to control steering. The fourth axis, ZR, is not used currently.

![Joystick with different control axes](http://www.logitech.com/repository/95/jpg/450.1.0.jpg)

Figure 5.4 Joystick with different control axes

The main function of the Joystick Control Module is to establish communication between the joystick and the Main Module of the program. User command (in other words, joystick axis position information) is obtained from the joystick axis and provided to the Main Module. The Main Module processes this information and converts it into a motion controller compatible
command. The motion controller receives this command and generates appropriate electric signals required for the respective system i.e. brake, steering or speed control system.

5.2.1 Steering control calculations in program

When joystick axis Z (steering control axis) is moved from left to right, it generates a numeric signal in the range of 0 to 65535 with neutral position generating a signal of 32767 and all intermediate positions generating signal values proportional to the axis position. The value 65535 is the highest number which can be represented by an unsigned 16 bit binary number. This numeric signal data needs to be converted to useful command information which can be supplied to the motion controller. The signal value that needs to be supplied to the motion controller to produce desired steering effect is in the range of -40000 to 40000. This value is obtained from various calculations like system analysis results, gearbox ratio and voltage value required to generate the desired torque. Table 5.1 shows the values of the joystick axis Z in 3 main positions.

<table>
<thead>
<tr>
<th>Joystick axis Z position</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>0</td>
</tr>
<tr>
<td>Center (default)</td>
<td>32767</td>
</tr>
<tr>
<td>Right</td>
<td>65535</td>
</tr>
</tbody>
</table>

Table 5.1 Joystick axis Z values

Following calculations are performed in the Main Module code to convert the joystick axis signals to motion controller signals.

\[
\text{initial\_steer\_value} = \frac{\text{axisZ value}}{32.767}
\]
If \( (initial\_steer\_value) > 1000\),

\[
final\_steer\_value = (initial\_steer\_value) \times 20 \quad \text{(5.1)}
\]

If \( (initial\_steer\_value) \leq 1000\),

\[
final\_steer\_value = ((initial\_steer\_value \times 40) - 40000) \quad \text{(5.2)}
\]

If the joystick axis Z is in a position that sends a value of 0 (leftmost position), then

\[
initial\_steer\_value = 0 / 32.767 = 0
\]

As per equation (5.2) above (since \(0 \leq 1000\)),

\[
final\_steer\_value = ((initial\_steer\_value \times 40) - 40000)
\]

\[
= ((0 \times 40) - 40000)
\]

\[
= -40000
\]

The vehicle will take a **full left turn**.

If the joystick axis Z is in a position that sends a value of 65535 (rightmost position), then

\[
initial\_steer\_value = 65535 / 32.767 = 2000
\]

As per equation (5.1) above (since \(2000 > 1000\)),

\[
final\_steer\_value = (initial\_steer\_value) \times 20
\]

\[
= (2000 \times 20)
\]

\[
= 40000
\]

The vehicle will take a **full right turn**.

If the joystick axis Z is in a position that sends a value of 32767 (neutral position), then
initial_steer_value = 32767 / 32.767 = 1000

As per equation (5.2) above (since 1000 <= 1000),

final_steer_value = ((initial_steer_value * 40) – 40000)

= ((1000*40)-40000)

= 40000 - 40000

= 0

The vehicle will travel in a straight line path.

Table 5.2 shows different steering values for different joystick positions.

<table>
<thead>
<tr>
<th>Joystick axis Z position</th>
<th>Joystick Value</th>
<th>Motion controller steering value</th>
<th>Effect on vehicle steering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leftmost</td>
<td>0</td>
<td>-40000</td>
<td>(full left turn)</td>
</tr>
<tr>
<td>Between neutral and left</td>
<td>16384</td>
<td>-20000</td>
<td>(half left turn )</td>
</tr>
<tr>
<td>Center (default )</td>
<td>32767</td>
<td>0</td>
<td>(no turn)</td>
</tr>
<tr>
<td>Between neutral and right</td>
<td>49151</td>
<td>20000</td>
<td>( half right turn)</td>
</tr>
<tr>
<td>Rightmost</td>
<td>65535</td>
<td>40000</td>
<td>(full right turn )</td>
</tr>
</tbody>
</table>

Table 5.2 Steering values for different joystick axis positions

5.2.2 Speed control calculations in program

Just like axis Z, when joystick axis Y (speed control axis) is moved up from its neutral position, it generates a numeric signal in the range of 32767 to 0 with all intermediate positions generating signal values proportional to the axis position. If the joystick axis is moved down from its neutral position, all other behavior is similar except that the signal range is 32767 to 65535. The signal value that needs to be supplied to the motion controller is in the range of 3.5 (zero speed) to 0.5(maximum speed), which is the voltage value to be supplied to the SX speed controller to run
two 7.5 HP motors. The following calculations are performed in the Main Module code to convert joystick axis Y signals to motion controller signals.

\[
\text{initial\_speed\_value} = \frac{\text{axis Y value}}{32.767}
\]

If \(\text{initial\_speed\_value} < 1000\),

\[
\text{final\_speed\_value} = \left[0.5 + \frac{\text{initial\_speed\_value}}{333.333}\right]
\]

If \(\text{initial\_speed\_value} \geq 1000\),

\[
\text{final\_speed\_value} = \left[0.5 + \frac{2000 - \text{initial\_speed\_value}}{333.333}\right]
\]

If the joystick axis Y is in a position that sends a value of 0 (axis Y moved all the way up), then

\[
\text{initial\_speed\_value} = \frac{0}{32.767} = 0
\]

As per equation (5.3) above (since \(0<1000\)),

\[
\text{final\_speed\_value} = \left[0.5 + \frac{\text{initial\_speed\_value}}{333.333}\right]
= \left[0.5 + 0\right]
= 0.5
\]

A voltage of 0.5 V will be generated to give \textbf{full speed} to the vehicle.

If the joystick axis Y is in a position that sends a value 32767 (neutral position), then

\[
\text{initial\_speed\_value} = \frac{32767}{32.767} = 1000
\]

As per equation (5.4) above (since \(0\geq1000\)),

\[
\text{final\_speed\_value} = \left[0.5 + \frac{2000 - \text{initial\_speed\_value}}{333.333}\right]
= \left[0.5 + \frac{2000-1000}{333.333}\right]
\]
\[
= [0.5 + 3]
= 3.5
\]

A voltage of 3.5 V will be generated to give zero speed to the vehicle.

If the joystick axis Y is in a position that sends a value of 65535 (axis Y moved all the way down), then

\[
\text{initial\_speed\_value} = \frac{65535}{32.767} = 1000
\]

As per equation (5.4) above (since 1000\(\geq\)1000),

\[
\text{final\_speed\_value} = [0.5 + (2000 - \text{initial\_speed\_value}) / 333.333]
\]

\[
= [0.5 + 0]
= 0.5
\]

A voltage of 0.5 V will be generated to give full speed to the vehicle.

Table 5.3 shows different speed values for different joystick axis Y positions.

<table>
<thead>
<tr>
<th>Joystick axis Y position</th>
<th>Value</th>
<th>Motion controller speed value</th>
<th>Effect on vehicle speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>0</td>
<td>3.5</td>
<td>Full speed (45 mph)</td>
</tr>
<tr>
<td>Between neutral and up</td>
<td>16384</td>
<td>1.75</td>
<td>Half the speed (22.5 mph)</td>
</tr>
<tr>
<td>Neutral (default)</td>
<td>32767</td>
<td>0</td>
<td>Zero speed (0 mph)</td>
</tr>
<tr>
<td>Between neutral and down</td>
<td>49151</td>
<td>1.75</td>
<td>Half the speed (22.5 mph)</td>
</tr>
<tr>
<td>Down</td>
<td>65535</td>
<td>3.5</td>
<td>Full speed (45 mph)</td>
</tr>
</tbody>
</table>

Table 5.3 Speed values for different joystick axis Y positions
5.2.3 Brake control calculations in program

Just like axes Y and Z, when joystick axis X (brake control axis) is moved from left to right, it generates a numeric signal in the range of 0 to 65535 with neutral position generating a signal of 32767 and all intermediate positions generating signal values proportional to the axis position.

The signal value that needs to be supplied to the motion controller in this case is in the range of 0 (zero braking force) to 10 (maximum braking force), which is the voltage value to be supplied to the brake controller to activate brake actuator, and in turn, the hydraulic pump. The following calculations are performed in the Main Module code to convert the joystick axis X signals to motion controller signals.

\[
\text{initial\_brake\_value} = \frac{\text{axis Y value}}{32.767}
\]

If \(\text{initial\_brake\_value} < 1000\),

\[
\text{final\_brake\_value} = \left[5 - \frac{\text{initial\_brake\_value}}{200}\right]
\]  \(\text{(5.5)}\)

If \(\text{initial\_brake\_value} \geq 1000\),

\[
\text{final\_brake\_value} = \left[5 - \frac{2000 - \text{initial\_brake\_value}}{200}\right]
\]  \(\text{(5.6)}\)

If the joystick axis X is in a position that sends a value of 0 (leftmost position), then

\[
\text{initial\_brake\_value} = 0 / 32.767 = 0
\]

As per equation \(\text{(5.5)}\) above (since \(0<1000\)),

\[
\text{final\_brake\_value} = \left[5 - \frac{\text{initial\_brake\_value}}{200}\right]
\]

\[
= 5 - 0
\]

\[
= 5
\]
A voltage of 5V will be generated to give **full baking force**.

If the joystick axis Y is in a position that sends a value 32767 (neutral position), then

\[
\text{initial_{brake\_value}} = \frac{32767}{32.767} = 1000
\]

As per equation (5.6) above (since 0\(\geq\)1000),

\[
\text{final\_brake\_value} = \left[5 - \frac{(2000 - \text{initial\_brake\_value})}{200}\right]
\]

\[
= \left[5 - \frac{(2000-1000)}{200}\right]
\]

\[
= \left[5 - 5\right]
\]

\[
= 0
\]

A voltage of 0V will be generated to give **zero baking force**.

If the joystick axis Y is in a position that sends a value of 65535 (rightmost position), then

\[
\text{initial\_brake\_value} = \frac{65535}{32.767} = 2000
\]

As per equation (5.4) above (since 1000\(\geq\)1000),

\[
\text{final\_brake\_value} = \left[5 - \frac{(2000 - \text{initial\_brake\_value})}{200}\right]
\]

\[
= \left[5 - \frac{(2000-2000)}{200}\right]
\]

\[
= \left[5 - 0\right]
\]

\[
= 5
\]

A voltage of 5V will be generated to give **full baking force**.

Table 5.4 shows different braking values for different joystick axis X positions.
<table>
<thead>
<tr>
<th>Joystick axis Y position</th>
<th>Signal value</th>
<th>Motion controller speed value</th>
<th>Effect on vehicle Braking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leftmost</td>
<td>0</td>
<td>5</td>
<td>Full braking force</td>
</tr>
<tr>
<td>Between neutral and left</td>
<td>16384</td>
<td>2.5</td>
<td>Half the braking force</td>
</tr>
<tr>
<td>Neutral (default)</td>
<td>32767</td>
<td>0</td>
<td>Zero braking force</td>
</tr>
<tr>
<td>Between neutral and right</td>
<td>49151</td>
<td>2.5</td>
<td>Half the braking force</td>
</tr>
<tr>
<td>Rightmost</td>
<td>65535</td>
<td>5</td>
<td>Full braking force</td>
</tr>
</tbody>
</table>

Table 5.4 Speed values for different joystick axis Y positions

5.3 Virtual Dashboard Module

The main purpose of the Virtual Dashboard Module is to provide the user with options to monitor the joystick control operations on a virtual (computer screen) dashboard. The operations that can be monitored are speed control, steering control and brake control.

5.3.1 Speed Control Module

Figure 5.5 shows the screen to monitor the speed control operations. The Dashboard Control Module comprises custom controls. Custom controls are the controls that can be developed as per specific graphical user interface requirements using inbuilt user controls like textboxes, datagridviews and timers. They abstract away the implementation code from the program and can be used as plug-in components in any code. The speed control screen in the virtual dashboard is a custom control and consists of a gauge and a needle. The operating speed range for our vehicle is 0-45 mph hence the gauge is calibrated to display 0-45 mph speed.
When the speed command from the joystick is passed to the motion controller, the same set of commands is passed to the Speed Control Module as well. The Speed Control Module converts this joystick value into a speed value in mph using simple mathematical calculations and displays it on the speed gauge. The speed gauge display is updated at the same frequency as that of the joystick state update frequency. This ensures that every time the joystick state of a particular axis changes (joystick axis Y in this case), it is reported to the virtual dashboard which in turn displays the respective speed on the speed gauge. As the joystick signals sent to the motion controller and the virtual dashboard are the same, the speed displayed has higher accuracy and
matches with the actual speed of the vehicle. The screen also displays joystick axes position values for all four axes.

5.3.2 Brake Control Module

Fig 5.6 shows the screen to monitor the brake control operations. The brake control screen is also a custom control and consists of a brake gauge with a needle to display the braking pressure value in psi units. When the joystick axis X is moved from its neutral position to pass a braking command to the motion controller, the same command is also passed to the virtual dashboard. The Virtual Dashboard Module converts this joystick value into braking force value in psi units and displays it on the brake gauge. Just like speed gauge, the brake gauge screen is also updated at the same frequency as that of the joystick state update frequency to display the most updated braking value. The screen also displays joystick axes position values for all four axes.
Figure 5.6 Brake control monitoring on virtual dashboard

Figure 5.7 shows both speed and brake controls on the same screen. If the user wants to monitor speed and brake simultaneously, this option can be selected from the main screen by clicking ‘Speed Control and Brake Control’ button.
Figure 5.7 Brake and speed control monitoring on virtual dashboard

5.3.3 Steering Control Module

Figure 5.8 shows the screen to monitor the steering control operations. The steering control screen contains a virtual steering wheel image which takes input from the joystick axis Z. When the joystick axis Z is moved from its neutral position to pass a command to the steering motors, the same set of commands is also passed to the virtual dashboard. Depending on the joystick axis Z value, the virtual dashboard calculates the angle through which the virtual steering wheel needs to be rotated to match the position of the actual steering wheel and displays the new position of the steering wheel on the screen. The virtual steering screen is updated at the same frequency as that of the joystick state update frequency. The screen also displays joystick axes position values for all four axes.
Figure 5.8 Steering control monitoring on virtual dashboard

Figure 5.9 shows the virtual steering screen when the vehicle is about to take a right turn. This is also evident from the Z axis reading in this case. The Z axis, which has a value of 32767 for neutral position shows a value of 34058, which is slightly greater than 32767 and hence the vehicle will start to take a right turn.
Figure 5.9 Steering control screen when the vehicle is about to take a right turn

5.4 Vehicle Tracking System Module

A vehicle tracking system is developed as a part of the main software program to track the route covered by the vehicle. This is achieved by capturing the joystick speed and steering commands and plotting them with the help of Dundas Charts for .NET. The Dundas Charts for .NET, a product by the Dundas Data Visualization, Inc. is a charting component that offers extensive charting solutions for .NET. When the joystick is operated to steer or accelerate the vehicle, the computer commands are recorded and saved in a database in the form of x and y coordinates. A timer control from the .NET visual studio toolbox is used for this purpose. The frequency of the timer can be set as per the precision requirements and in our case it is set to 1 second so that the positions of the joystick axes Y and Z are captured every second. The Vehicle Tracking System
plots these values in the form of an X-Y line chart. At the end of a trip, the track report can be generated by clicking ‘Generate Track Report’ from the Menu.

To generate the route navigated by the vehicle, the two parameters required are distance traveled by the vehicle and angle through which the vehicle takes a turn. These values are calculated from the joystick command data by doing some arithmetic and trigonometric calculations.

5.4.1 Displacement calculations

In joystick control mode, the speed of the vehicle is controlled by the joystick axis Y position with zero speed at 32767 (neutral position) and maximum speed at 0(uppermost position). This joystick speed command is already converted into mph in the Speed Control Module so the same speed is used for the displacement calculations. The value of speed is converted into distance by dividing it by time. The following example shows the displacement calculation when the speed is 40 mph for one second.

Vehicle speed = 40 mph  

\[
\text{Vehicle displacement per second} = \frac{\text{Vehicle speed}}{3600}
\]

\[
\text{Vehicle displacement in m/s} = \left(\frac{\text{Vehicle speed} \times 1608}{3600}\right)
\]

From equation (1) and (2),

Vehicle displacement for given vehicle speed = \(\frac{40 \times 1608}{3600}\)

\[= 17.8 \text{ meters}\]
5.4.2 Angle calculations

In joystick control mode, the steering angle of the vehicle is controlled by the joystick axis Z position with 0° at 32767 (neutral position), -45° at 0 (leftmost position) and 45° at 65535 (rightmost position).

The following example shows the angle calculation when the joystick axis Z value is 65535

\[
\text{joystick\_axis\_Z\_position} = 65535
\]

From (3) and (4),

\[
\text{vehicle\_steering\_angle} = \frac{(\text{joystick\_axis\_Z\_position} - 32767)}{728} = 45^\circ
\]
axis values. The Vehicle Tracking System Module retrieves this information from the database and connects the series of x and y coordinates to generate the route covered by the vehicle. To be able to generate the route separately for each trip, the program is designed to overwrite the existing database values every time the joystick control program is restarted.

![Diagram of Vehicle Tracking System Module]

Figure 5.10 Working of Vehicle Tracking System Module

Figure 5.11 shows the route generated by the Vehicle Tracking System for a set of angle and displacement values.
The database values can also be exported to MS Excel to obtain the route in Excel format. Figure 5.12 shows the same route generated in Excel format.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>0</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>75</td>
</tr>
<tr>
<td>18</td>
<td>89</td>
</tr>
<tr>
<td>38</td>
<td>88</td>
</tr>
<tr>
<td>55</td>
<td>78</td>
</tr>
<tr>
<td>72</td>
<td>67</td>
</tr>
<tr>
<td>88</td>
<td>56</td>
</tr>
<tr>
<td>105</td>
<td>45</td>
</tr>
<tr>
<td>122</td>
<td>34</td>
</tr>
</tbody>
</table>
Chapter 6: Experimental Results

After the computer control implementation, the vehicle was extensively tested in electric power mode in which it could cover a distance of around 100 miles per charge. Fig 6.1 shows the vehicle while testing. The vehicle has six 12 V batteries and thus runs about 100 miles on 72 volts. A 1000 miles hybrid test was also performed in which a gasoline powered generator was used to charge the set of batteries. Finally, the vehicle was tested for drive-by-wire mode in which it was completely controlled by a joystick. To ensure safety during driving operations, the drive by wire tests were first performed by putting the vehicle on jack stands. After the jack stand test results were found to be satisfactory and agreeable to the expected results, the vehicle was actually tested on ground. The results of the drive-by-wire test, which included controlling steering, acceleration and braking operations using a joystick showed acceptable results. The speed and braking force readings on the virtual dashboard also showed a close match with the actual speed and braking force values. The videos showing jack stand tests can be found at

1. http://www.youtube.com/watch?v=7-39URz4vL8 - Steering control
2. http://www.youtube.com/watch?v=kTlAuS7KPSo - Speed control

Figure 6.1 Testing the vehicle
Chapter 7: Conclusion and Future Recommendations

7.1 Conclusion

The electric vehicle was converted into a drive-by-wire vehicle by modifying the existing steering, braking and speed control systems. A software program and a Galil motion controller were used to control the vehicle by computer commands and a computer joystick was used as an input device for this purpose. A virtual dashboard system was developed to monitor the computer control of speed, braking and steering operations on computer screen. A vehicle tracking system was also developed as a part of the software program to track the vehicle route using the joystick commands.

![Virtual Dashboard Module](image)

Figure 7.1 Block diagram of complete computer control system

Fig 7.1 shows the block diagram of complete computer control system. It shows the signal flow from the input device to the wheels, steering column and brakes via the computer software and the motion controller. The input signals are also sent to the virtual dashboard to monitor speed.
and brake control on the screen and to the Vehicle Tracking System Module to track the route navigated by the vehicle.

Currently, the vehicle is being used as a test bed for studies and research related to control theory, autonomous vehicle technologies and various computer controls.

### 7.2 Future recommendations

Currently, the electric vehicle has speed, braking and steering operations under computer control. The other operations like forward–reverse, turn signals and wind shield wipers can also be brought under computer control by interfacing the respective electric systems with the Galil motion controller. System design for computer control of forward–reverse switch is shown in Appendix A.
References


Appendix A

Hardware design for computer control of forward-reverse switch

Figure 1. Hardware connections for computer control of forward-reverse motion

Voltage readings for three modes:

Neutral – 1.1V
Forward – 0V
Reverse – 0V
<table>
<thead>
<tr>
<th></th>
<th>OUT1 (F)</th>
<th>OUT2 (N)</th>
<th>OUT3 (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ON</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FWD</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>REV</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Truth table for reverse-forward switch connections
Figure 2. Flowchart for computer control of reverse-forward switch