Chapter1 Introduction

1.1 Motivation

Although vehicles were created in 1886, vehicles are still mostly driven by humans at present. However, humans are fallible, like feel sleepy while driving at night or look away from the road to do something, many tr affic accidents have happened. According to statistics of Ministry of Transportation, at least 3 thousand people are injured in traffic accidents every year in Taiwan. In addition to traffic accidents, traffic congestion is also a serious problem. One way to solve this problem is to use current roads more efficiently by removing as much human involvement as possible through intelligent controller and automation. Therefore , the main objective of development of the intelligent transportation system (I T S) is to assist drivers in avoiding collision and reduce traffic congestion.

ITS is comprised of a number of technologies, including information processing, communications, control, and electronics. Joining these technologies to our transportation system o f ITS are Advanced Traffic Management System (ATMA), Advanced Traveler Information Systems (ATIS), Advanced Public Transportation System (APTS), Advanced Vehicle Control and Safety System (AVCSS), Commercial Vehicle Operations (CVO), Emergency Management S ystem (EMS) and Electronic Payment System & Electronic Toll Collection (EPS&ETC) [1]. In this thesis, we will focus on the AVCSS.

Among the various technical components in AVCSS, the Adaptive Cruise Control (ACC) system is an automatic control syste m for throttle and brake control to achieve automatic vehicle following in longitudinal direction or maintain the desired cruise speed when no vehicle ahead. When there are no vehicles or obstacles in the same lane, the ACC system works like a conventional cruise control system that maintains a pre -selected speed, if a vehicle with slow speed or an obstacle is ahead in the same lane, the ACC system automatically will reduce its speed and maintain the safety distance. If the lead ing vehicle keeps slowing down, or if another object is detected, the system sends a signal to the engine or braking system to decelerate. Then, when the road is clear, the system will sends a signal to the engine to re-accelerate the vehicle back to the set speed . When this preceding vehicle or obstacle disappears, the ACC system automatically accelerates back to the desired cruise speed. This is achieved through a radar headway sensor, digital signal processor and a speed controller.

Although ACC systems have been in market since 1995, it is still as an optional device for luxury vehicles. One of the reasons is that the ACC system is not cheap. One of the expensive components is controller which may be a notebook or car computer. We want to design a chip to replace the controller a nd have the aid of semiconductor manufacture to cost down. And then ACC system could be popularized to par vehicle. In order to determine if it is feasible to replace the controller with a chip , we use FPGA to test and verify it. Besides we do our best to use not expensive sensors and popular chips and to grab the signal from vehicle itself by avoiding installing sensors on it.

1.2 Literature review

Automatic driving is to design a controller to mimic the human driver's behavior. Therefore we must understand this behavior to certain degree. In the ACC system, the main objective is to follow a preceding car and maintain the desired safety distance chosen by driver, therefore collision prevention [2-5] is one of the key technologies in automatic vehicle following. How to define the safety distance to avoid collision is also an important task [6-11].

ACC system generally uses computer to replace the human driver. It can not only eliminate the human reaction time to improve driving safety but also reduce workload. Many companies are developing ACC system to provide a safe, reasonable and comfortable drive [2]. The main factors depending on driving comfort are de/acceleration and jerk. Some comparisons of driving behavior with and without ACC were discussed in [6,8] and [12]. The results showed the difference between computer driving and human driving. It must be noticed that adaptive cruise control system is a service to help the driver, not a replacement of the driver. The driver is still in charge of the car at any moment, regardless if the ACC system is active or not [13].

Driving behavior in vehicle following has been an active area of research since the early 50's [8]. In vehicle following the human drivers act as a controller. They sense the velocity and distance of the leading vehicle and then control the acceleration or brake pedal accordingly. In the complex traffic situation and various driving style, it is hard to design a perfect controller that can deal with all situations. Therefore we develop a controller that can implement cruise control and adaptive cruise control. The widely used controllers for adaptive cruise control systems are PID controller, fuzzy logic and neural network controller as described below.

1.2.1 PID controller

Ioannou proposed a PID controller to achieve steady state vehicle spacing for a pre-selected time headway [14], and used relative speed and distance information from the preceding and following vehicles in order to choose proper control action for smooth vehicle following and for maintaining a desired safety distance specified by the driver [15]. He also proposed throttle and brake control systems for automatic vehicle following that guarantee smooth vehicle following even when the leading vehicle exhibits erratic speed behavior [16]. Some stop and go cruise controllers [17,18] were proposed for low vehicle speed during congested traffic flow. There are a lot of techniques to perform ACC. Conventional methods based on analytical control generate good results but with high design and computational costs since the application object, a car, is a nonlinear element and its full mathematical representation is impossible [19-22]. Other ways to reach a human-like speed control is the application of artificial intellig ence techniques [23,24]. One of these techniques is the fuzzy control that allows an approximate human reasoning and an intuitive control structure [25].

1.2.2 Fuzzy logic neural network controller

The strength of the fuzzy logic approach was that no c haracterizations of the nonlinear and often unobservable vehicle dynamics were necessary. This allowed the controller to be transportable to other vehicles with different dynamics. Protzel proposed the fuzzy logic controller in driver assistance system, wh ich aids in automatically following a leading car at a desired distance [26]. He also used the fuzzy logic controller to replace the inverse dynamics of the vehicle in speed control. It releases us from mathematically establishing the nonlinear inverse dyn amics of the vehicle model. He introduced the concept of the type of driver, but lack the consideration of the different vehicle dynamics. In other words, the controller only suits for one type of vehicle. Therefore , there are many fuzzy logic controllers combined with neural-network to adapt to all kinds of vehicles [27-29].Various controllers mentioned above can be used to satisfy the same objectives, but they may require different variables to be sensed or different conditions to be met.

In this thes is we introduce a fuzzy logic controller to serve as the ACC system. The motivation to use the fuzzy logic is that it can be used to design a controller of nonlinear systems without precise knowledge of its mathematical model. In addition, experts experiences can be easily implemented by a fuzzy logic controller.

1.3 FPGA based controller

A Field Programmable Gate Array (FPGA) , which is made up of Combinational Logic Blocks (CLB), is a digital integrated circuit that can be programmed to do any type of di gital function. These blocks are made up of an array of digital AND, OR and INVERT gates. The CLBs are arranged in an array to implement different design. Each block is planed to perform a logic function that can be interconnected, so that the complete log ic function can be implemented.

There are many advantages of an FPGA over a microprocessor chip for fuzzy systems:
(1) An FPGA has the ability to be reprogrammed very easily and quickly. Therefore FPGA is suitable for fast implementation and quick hard ware verification. As compared to the dedicated fuzzy hardware, FPGA based system is more flexible than fuzzy chips. We can make changes if the design is incorrect . They can be easily reconfigured with new design. Hence it is often used to be a prototype chip to verify the function is correct or not.
(2) An FPGA is programmed just using support software and a download cable connected to a host computer. Once they are programmed, they can be disconnected from the computer and will retain their functionality until the power is removed from the chip. A Read Only Memory (ROM) type of a chip that is connected to the FPGA programmable inputs can also program the FPGA upon power-up. This means that when a board is in place in a remote location, the

chip can keep running while the designer updates the design back at a lab. Once the designer updates the design he or she can program another ROM chip, take it to the site and replace the old ROM chip. The next power-up the chip will be reprogrammed to the new design.
(3) An FPGA is described or modeled using Hardware Description Languages like Very High Speed Integrated Circuit Hardware Description Language (VHDL), Verilog, etc and verified by simulation. VHDL is now one of the most popular standard HDLs and can be used to describe the behavior or structure of the digital system.

The thesis presents the design and implementation of a digital fuzzy logic controller on a FPGA using VHDL.

1.4 Brief sketch of the contents

The thesis is organized as follows: Overall structure of vehicle longitudinal control system is presented in Chapter 2. In Chapter 3, we will discuss the peripheral interface of the system. In Chapter 4, we explain and implement the fuzzy logic controller. The experimental results are given in Chapter 5. Finally, the conclusion is described in Chapter 6.

Chapter 2 Overall Structure of Vehicle Longitudinal Control System

The main objectives of the vehicle longitudinal control are distance control and speed control [31]. Distance control system shown in Figure 2-1 makes the vehicle s following a preceding vehicle and maintain a safety distance. For situation (I) the relative distance between these vehicles is too large. Then the following vehicle speeds up till they keep the desired distance and the same speed finally. For situation (II) the relative distance between these vehicles is not enough. Then the following vehicle slows down till they keep the desired distance and the same speed finally. It is the main purpose of Adaptive Cruise Controller (ACC). Cruise control system shown in Figure 2-2 makes the vehicle keeping the desired cruise speed set by driver. For situation (I) the speed of this vehicle is not enough. Then the vehicle speeds up till they keep the desired speed finally. For situation (II) the speed of this vehicle is too high. Then the vehicle slows down till it keeps the desired speed finally .It is the main purpose of cruise controller (CC).

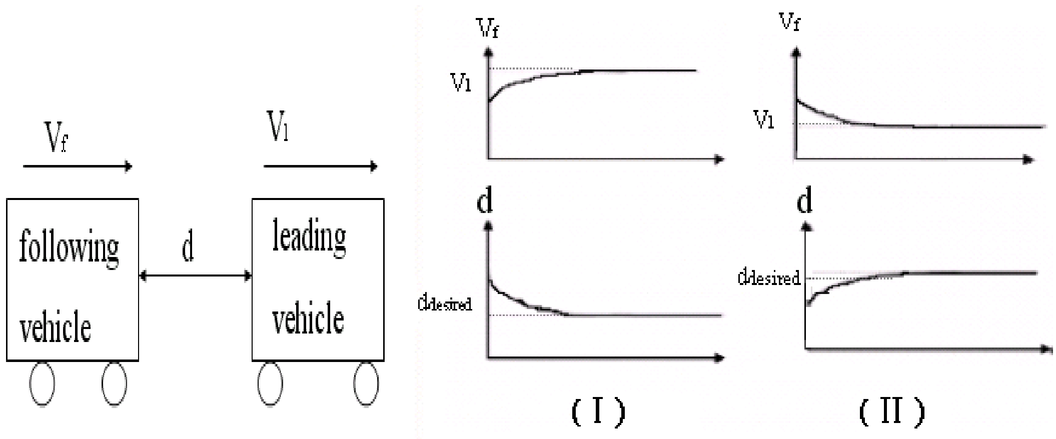


Figure 2-1 Distance control system

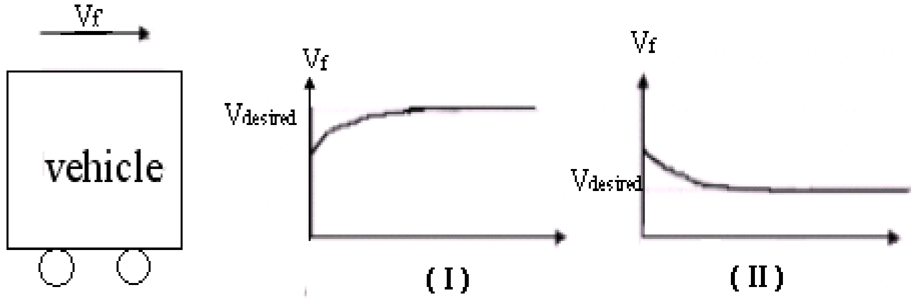


Figure 2-2 Cruise control system

The flow chart of the intelligent cruise control system is shown in Figure 2 -3. The process is described by following steps:

Step 1: Set cruise control speed

Step 2: Receive input from the sensors

Step 3: Check if it endangers now, if yes, go to step 4, if no, go to step 5.

Step 4: Perform safety controller and back to step 2.

Step 5: Check if any obstacle is ahead, if yes, go to step 8, if no, go to step 6.

Step 6: Cruise controller determines how to control speed.

Step 7: Perform speed control and back to step 2

Step 8: ACC controller determines how to control distance.

Step 9: Perform distance control and back to step 2.

Why do we need 0.13 , 0.4, 0.5 or 0.8 sec delay in the flow chart as show in Figure 2-3? For 0.13 sec it is because we need quickest reaction in unsafe situation and 0.13 is the shortest period of scanner. According the general cruise controller, the delay of slow down is about 0.5 sec and the one of speed up is about 0. 4 sec a t 40km/hr. For 0.8 sec , it is under the consideration of trustiness of laser scanner. If the delay is too short, changing distance is also small. Besides there must be measure inaccuracy for laser scan and this inaccuracy is not large. We can’t make sure that the distance is changing due to inaccuracy or not in short delay. Therefore we increase delay time to make sure that distance is truly changing.

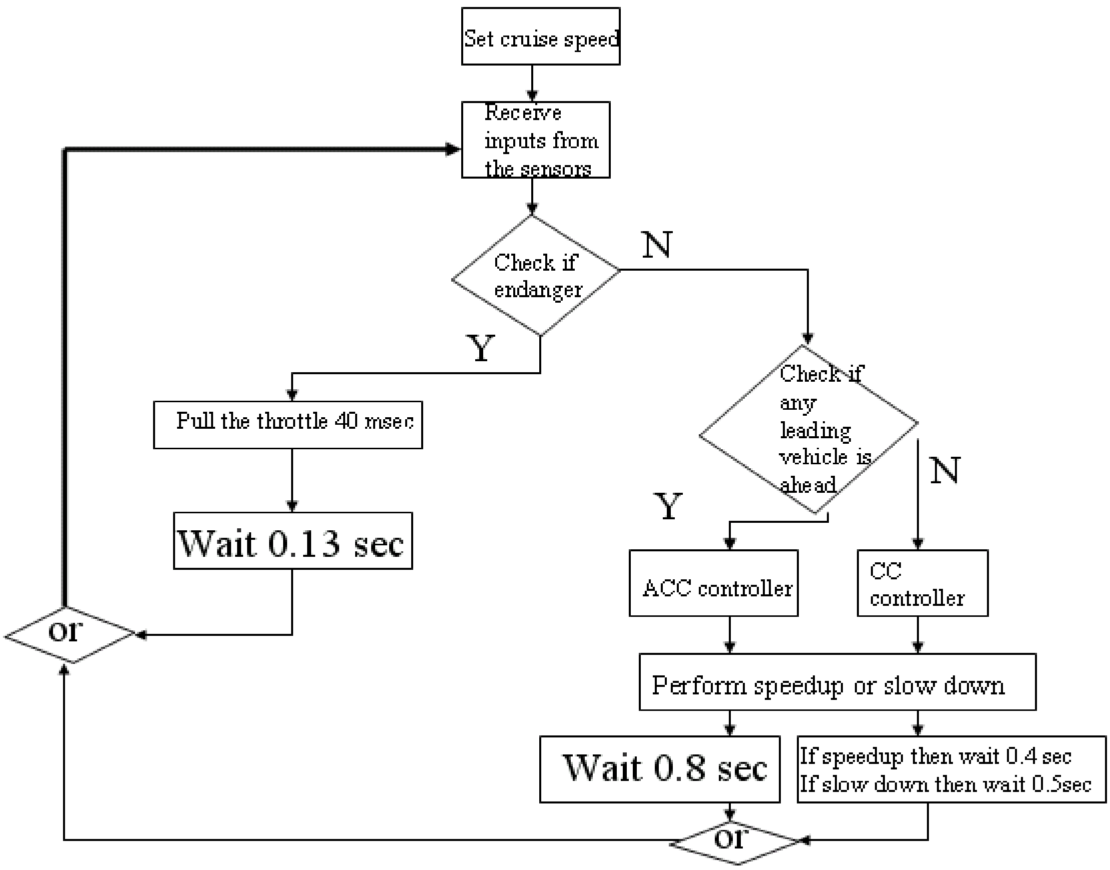


Figure 2-3 Overall flow chart of intelligent cruise control system

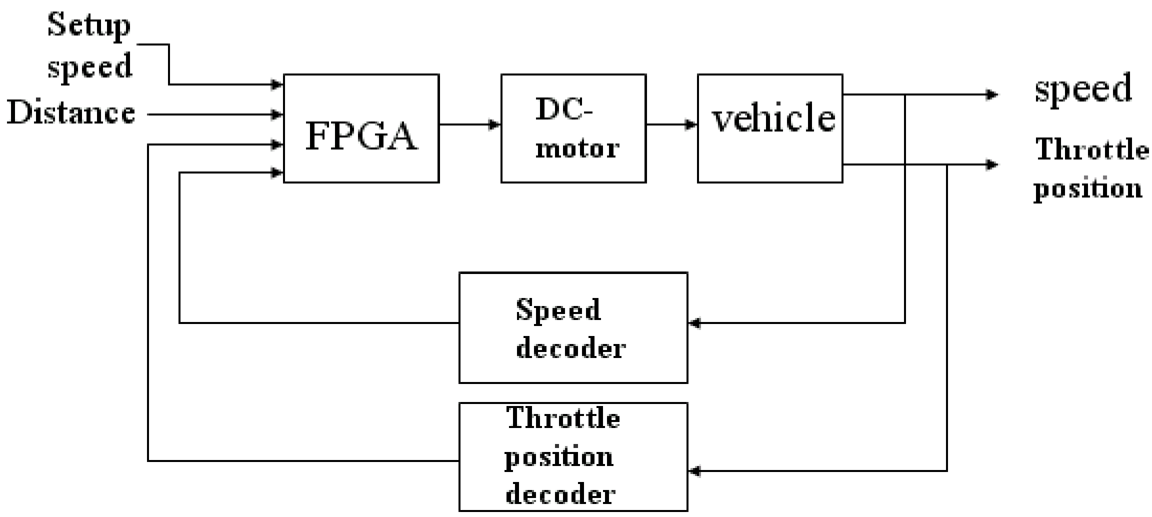


Figure 2-4 Overall structure of experimental system

Figure 2-4 shows the overall structure of experimental system. There are two inputs of distance and setup speed in the system. The distance sensor is responsible to detect the real-time distance. If it is more than 40 meters, the controller will operate in the cruise control (CC) mode. If it is less than 40 meters, the controller will change into the adaptive cruise control mode (ACC). In ACC mode it will provide the most important information for the controller to determine the vehicle should speedup or slow down. Besides, the distance gap between the last distance and now distance tells the degree of approach or leave. The setup speed instrument is a PS-2 number keyboard. If the controller operates in the CC mode, the setup speed instrument will provide the target speed. If the recent speed is lower than target speed, the controller will send a command to speedup. On the contrary if the recent speed is higher than target speed, the controller will order the vehicle to slow down.

Here we grab two kinds of feedback values which are very important in CC mode: throttle position voltage and real-time speed. The throttle position voltage provides the rough steady-state voltage that controller should hold around. Because of the delay of the vehicle, the steady-state voltage may approach and the speed still does not catch up. The controller will think he should still speedup. Hence it will cause the great overshoot .The controller can know the above steady-state voltage to avoid speedup or slow down too much. That is to say it can avoid the overshoot too much. If the controller holds at the above voltage level , and then the controller will execute the fine tuning according to the real-time speed till the target speed is equal to the real-time speed.

How do we control speed of the vehicle? There must be an interface to connect the FPGA and vehicle. Here we use a DC-motor to control the throttle to adjust the mass of air into engine. In other words, if we control the throttle, we will control the speed indirectly. However under the consideration of safety we do not have the interface to control the brake.

After all the brake is the most important safety device and people have the great responsibility to control vehicle.

The platform of experimental system is SAVRIN 2400 c.c . as show in Figures 2-5 and 2-6. Table 2-1 shows the sample specification of the plant.

Table 2-1 The specification of plant.

Mitsubishi Savrin 2.4	
Engine type	L4 DOHC 16V VVT+DMM
Exhaust	2400 cc
Horsepower (hp/rpm)	150/6250
Torsion (kgm/rpm)	19.2/3000
Transmission	INVECS-II SPORTS-MODE 4 A/T
Weight	1640 kg

Figure 2-5 The front view of plant

Figure 2-6 The lateral view of plant

Chapter 3 Peripheral Interface

3.1 Setup speed instrument

At the first step, we should set up the cruise control speed. Hence we must create an input subsystem. Under the consideration of convenience and communication with computer, we use PS-2 number keyboard. We show the appearance and pin function of it in Figure 3-1.

Pin	Signal	Description
1	Keyboard DATA	Keyboard data
2	NC	No connection
3	GND	Ground
4	VCC	+5V
5	Keyboard Clock	Keyboard clock
6	NC	No connection

Figure 3-1 Appearance and the pin function of PS-2 number keyboard

Afterward we have to identify the keyboard data waveforms. Here we use number 0 to 9 to setup our target speed. In Figure3-2, we show number "1" waveform for example. The upper pattern figures out the overall waveform from Pin 1. There are apparently three parts in a number. The under pattern shows the signals from Pin 1 (the upper line) and Pin 5 (the under line). From the pattern we find Pin 1 and Pin 5 will vary only when the keyboard works. The other times they will stay at "high" level. Besides according the PS-2 keyboard protocol, we should grab the Pin 1 data when the Pin 5 falling edge. It is a synchronous method. However, it is easily affected by noise and there are too much vibrations in the vehicle. Therefore we still deal with the Pin 1 data with asynchronous method. It tries to grab the data at the center of the bit not trigger by edge. Hence it is less sensitive to noise. However, no thing is perfect. The expense of using asynchronous method is spending more logic elements then synchronous method.

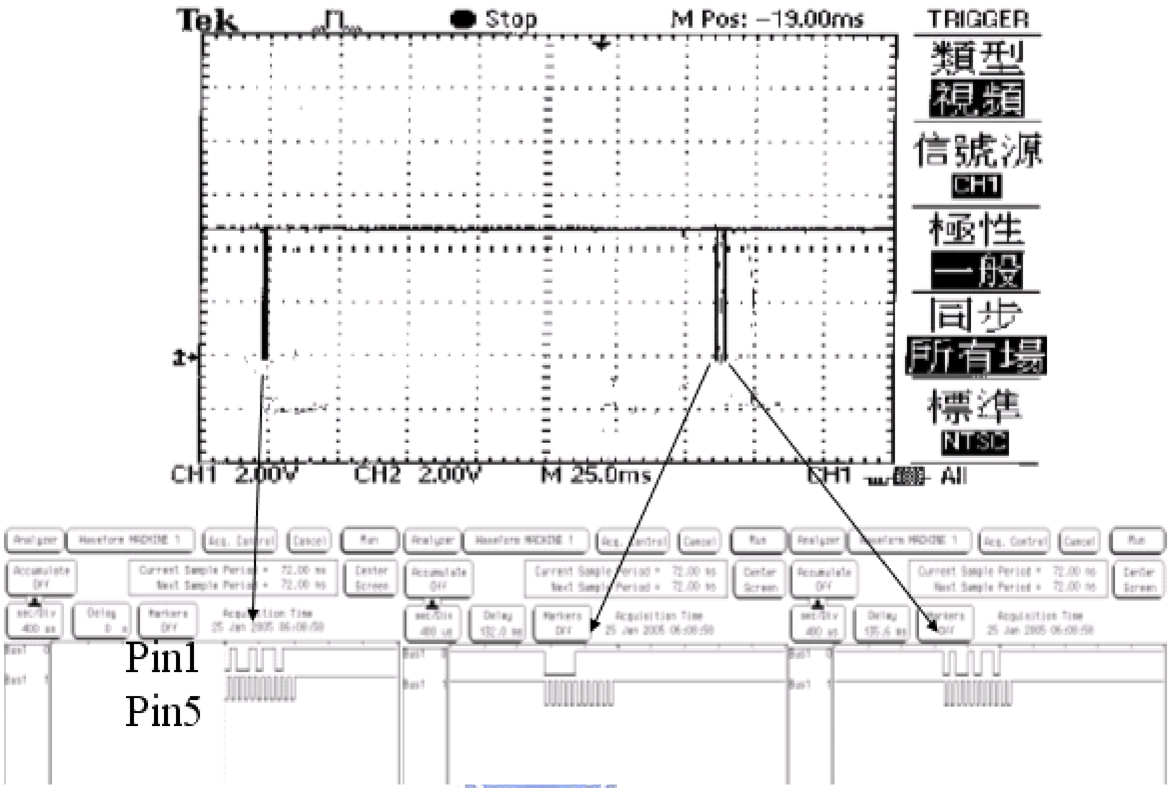


Figure 3 -2 The waveform of number “ 1 ”

The main object to build the input instrument is to setup the speed we want without speedup or slow down to the desired speed practically. Some of the cruise controller can only keep the speed just now. If we want to change the setup speed, we should first disable the cruise controller. And then we have to speedup or slow down to the speed we want. Finally we have to re-setup the cruise controller again. It is very inconvenient.

3.2 Speed sensor

In the ACC system one of the most important feedback signal is the speed of our vehicle right now. Therefore it is a very important task to decode the speed from the velocity sensor. In Figure 3-3, it shows where we draw out the speed signal. We measure the speed signal practically and then list in Table 3-1. It indicates that the relation between speed and frequency of speed signal is almost linear. On the assumption of linear relation we can infer from the speed signal to speed. Besides the speed from velocity sensor is between 12V and 13 V, we need to turn the voltage to 5V. Here we use a chip named 7805 to arrive at the target.

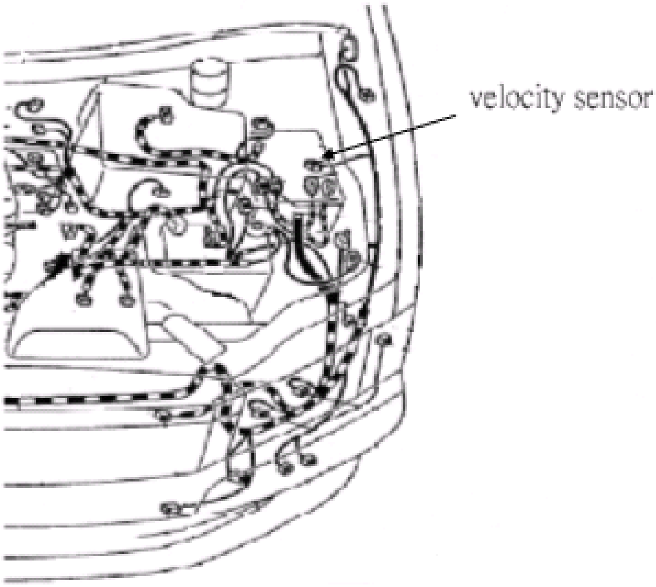


Figure 3-3 The location of speed sensor [36]

Table 3-1 Relation between speed and frequency signal

Velocity (Km / hr)	Frequency signal (Hz)
10	7
20	14
30	21
40	28
50	35
60	42
70	49
80	56
90	63

3.3 Throttle position sensor

The other feedback of the system is the throttle position. Figure 3-4 shows the location of the throttle position sensor. The throttle position is displayed with analog voltage from 0V to 5V. When the vehicle is in static state, the throttle position voltage is about 0.6V. As the throttle is opened more, the throttle position voltage will be increased. It means the more air-gasoline into the engine and more force applies on the vehicle. Thus if we want to exploit the relation, we will need an instrument to change analog signal to digital signal. The most frequent A/D converter is ADC0804 encoding any smaller analog voltage span to the full 8 bits of resolution shown in Figure 3 -5. The reason we use it is its conversion Time < 100 us, easy interface to most microprocessors, TTL compatible inputs and outputs and 0V to 5V analog voltage input range (single + 5V supply). With the position information, we can adjust the throttle to the approximate stable position first. The action can help us spending the shorter time to arrive at the rough setup speed, then designed fuzzy logic controller is applied to avoid great overshoot. Table 3-2 exhibits the relation between the stable speed, throttle position voltage and ADC0804 transformation.

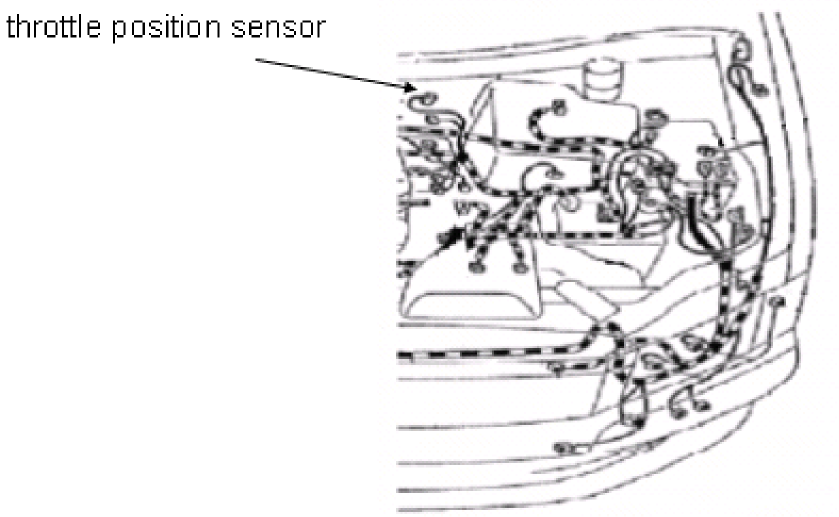


Figure 3-4 The location of the throttle position sensor [36]

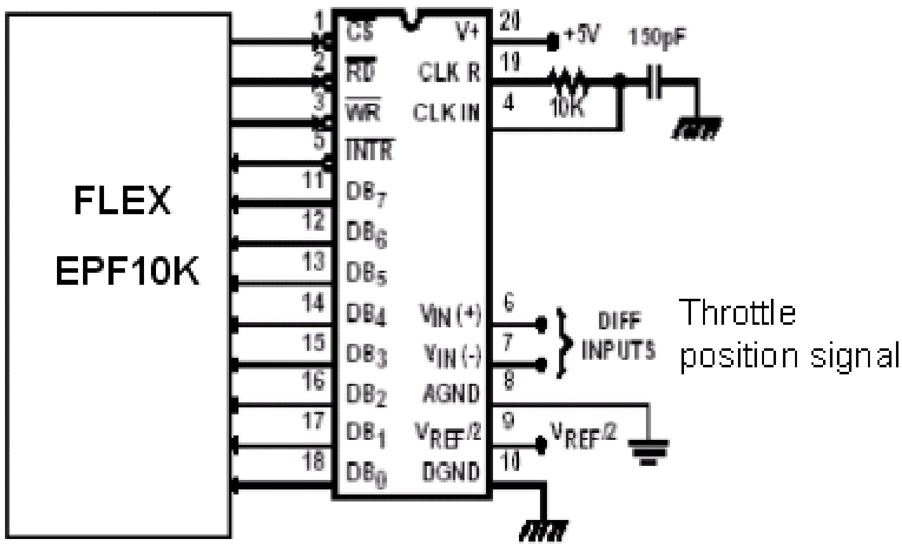


Figure 3-5 The connection between FPGA and ADC0804

Table 3-2 The relation between the stable speed , throttle position voltage and ADC0804 transformation

Speed (km/hr)	Throttle position voltage(V)	ADC0804
40	1.1	00110111
50	1.2	00111100
60	1.2	00111100
70	1. 25	00111110
80	1.325	01000010
90	1.35	01000011

3.4 Distance sensor

The distance sensor is the most important sensor in ACC system. Because the scanning range of laser scanner is larger than ultrasonic scanner, sound wave scanner and light wave scanner, we choose it. In Figure 3 -6, we show the appearance of it. The interface between FPGA and the laser scanner is RS -232. Due to different voltage levels shown in Figure 3 -7 between each other, there must be a voltage changed circuit. We make use of MAX232 shown in Figure 3-8 to achieve our requirement.

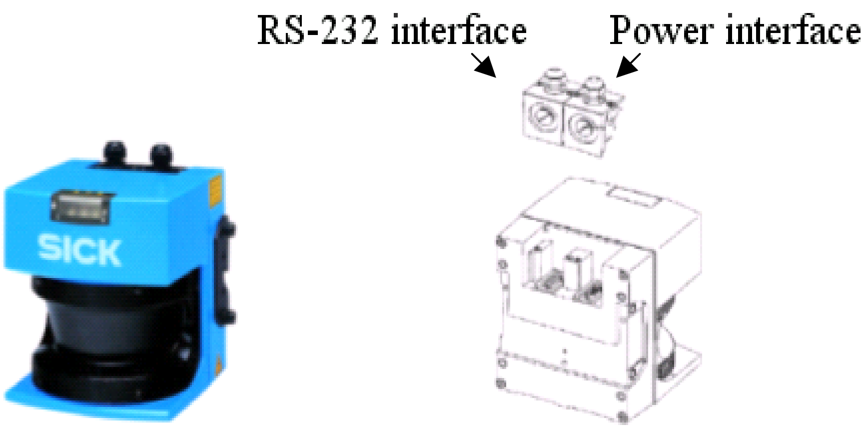


Figure 3-6 The appearance of distance sensor [37]

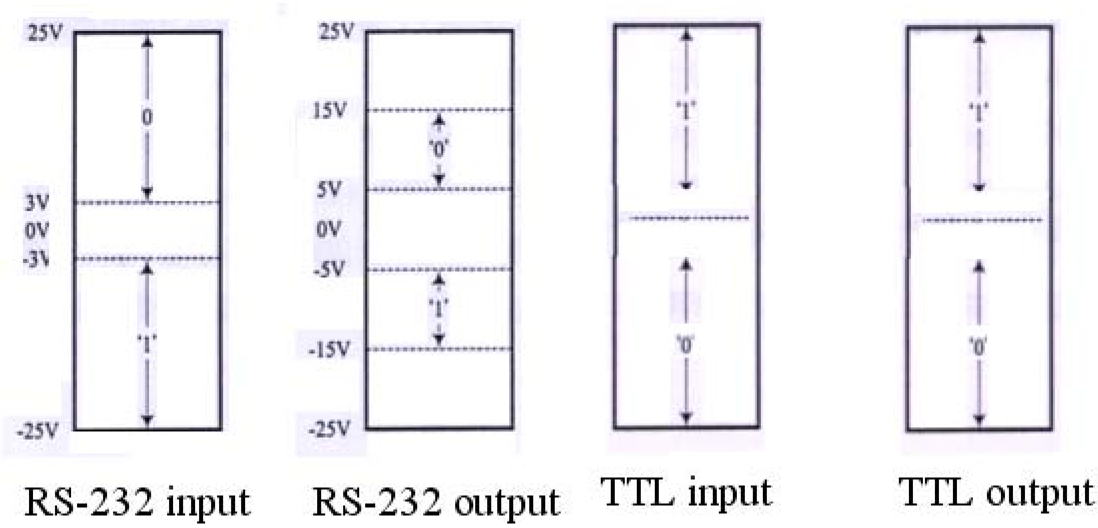


Figure 3-7 Voltage levels of RS-232 and TTL

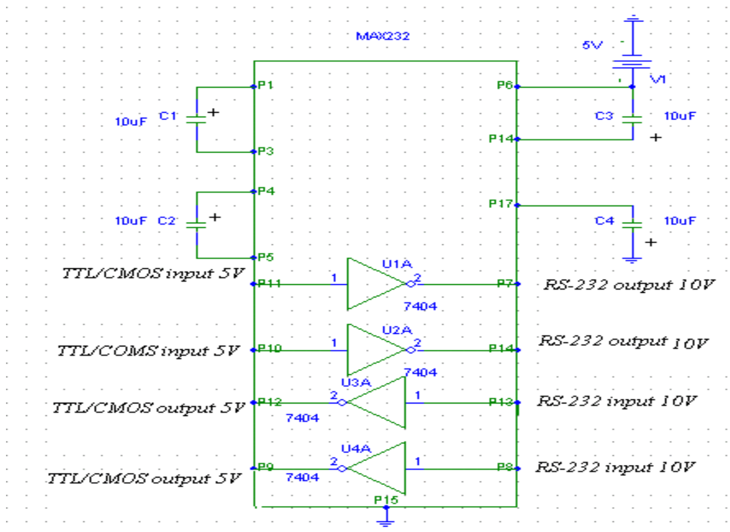


Figure 3 -8 MAX232 and operating circuit chart

The shorter period scanner works, the quicker rea ction we can act. Therefore, we configure our laser scanner to shorter scan period before operating the machine, change the baud rate from 9600 to 38400, and narrow scan angles and the resolution of each angle. A fter these setup steps are operated the laser scanner scans once about per 0. 13 second as shown in Figure 3-9, which is grabbed from the oscilloscope.

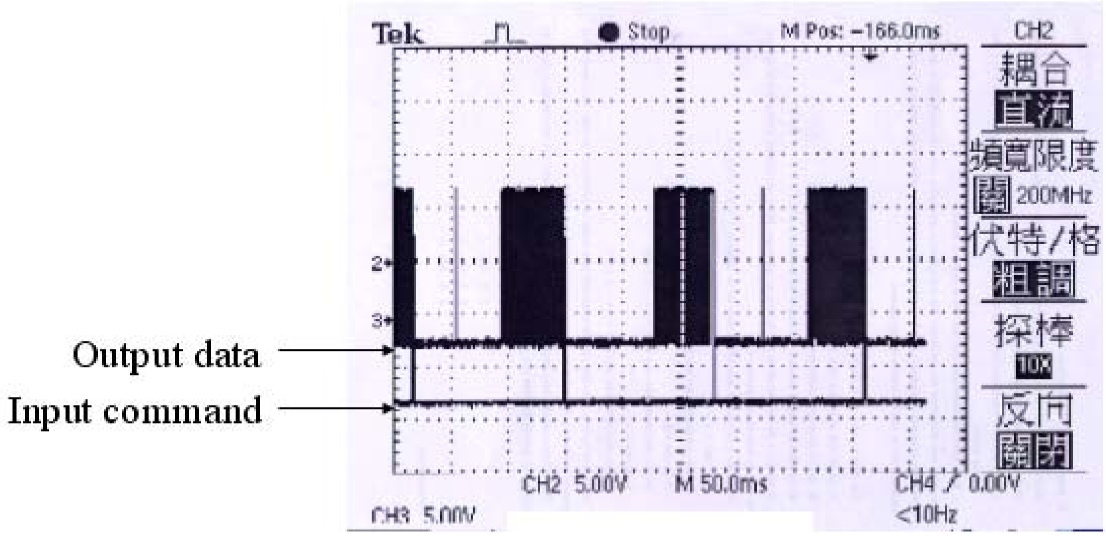


Figure 3-9 The waveform of laser scanner

If we only grab the distance from angle 90 degree, the respondent distance may be infinite. It might be because of the laser light absorbed by deep color objects or pass through glass without reflection or other reasons. In order to increase the reliability, we extract the distance of angle 88 ,89,90,91 and 92 degrees. Then we only deal with the shortest distance among thes e. There are still chances all of the distances missed even through this step. Therefore we have to think another way.

There are just two conditions which cause the distance varies drastically. One is the front car change lane and the distance will change fiercely. The other one is the detective laser light gone temporarily. Afterward the distance will change severally instantaneously. Hence if the response distance changes fiercely, the controller should not think that there are no cars in front of itself . It will hold the last distance and check the next distance again. If the next distance is more than 40 meters, the controller will really believe there are no vehicles in front of itself. On the contrary, it will determine what to do according to the real-time data.

However, there is still a problem during making a turn [30]. When the vehicle makes a turn, the directions that the sensor and the car move forward are not the same actually. We show a diagram to explain it in Figure 3 -10. Hence, we should kno w the angle between the moving forward and right front of car and then we grab the going forward distance through laser scanner.

We install an angle sensor in the axis of steering wheel like shown in Figure 3-11. It is because there is a relation between s teering wheel turning angle and the tire turning angle. Although the rate changes with speed, it is about 25 degrees of steering wheel to one degree tire. The interface of steering wheel angle sensor is CAN (Controller Area Network) bus. CAN is a serial bus protocol being primarily intended for transmission of control related data between a number of bus nodes and often used in automotive and general industrial applications. Capable of high-speed (1 Mbits/s) data transmission over short distances (40 m) and low-speed (5 kbits/s) transmissions at lengths of up to 10,000 m, the multi -master CAN bus is highly fault tolerant, with powerful error detection and handling designed in.

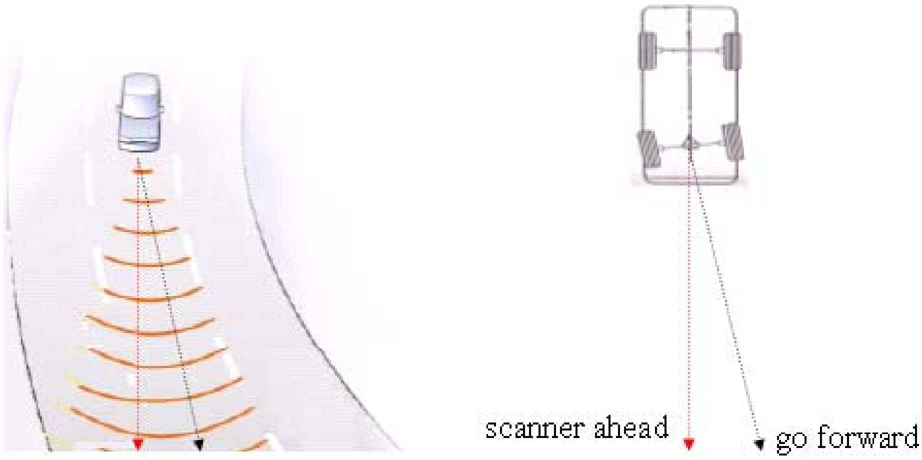


Figure 3 -10 The directions that the sensor and the car move forward when a vehicle makes a turn [38]

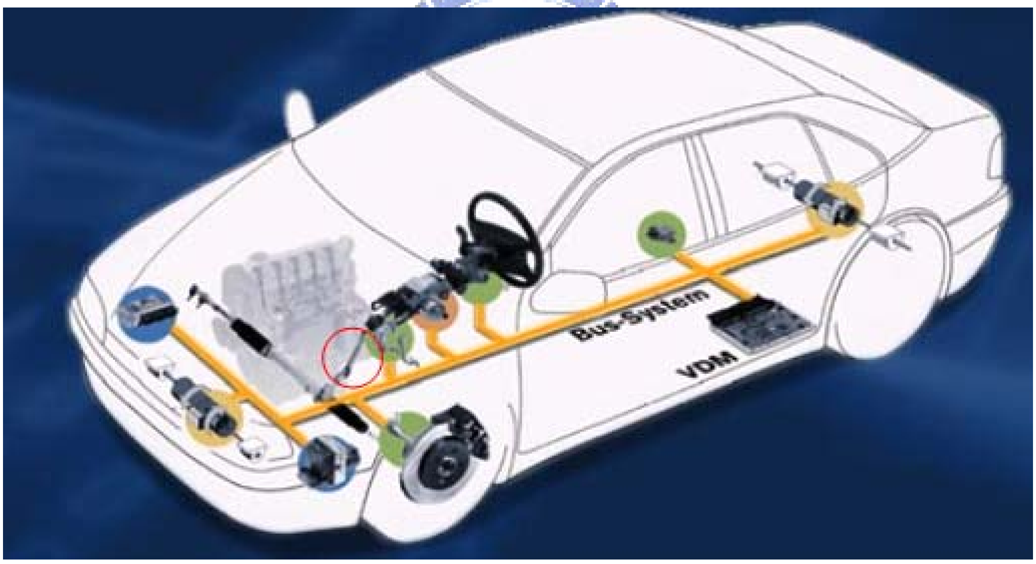


Figure 3 -11 The location of wheel angle sensor [38]

According to CAN bus protocol there are three components in CAN decode circuit , a CAN transceiver, a CAN controller and a central processing unit . T raditionally, these components are PCA82C250 , SJA1000 and 80C51 , respectively shown in the Figure 3 -12. The PCA82C250 support s a differential bus signal representation as described in the international standard for in -vehicle high -speed applications (ISO 11898) using the CAN protocol. In other words, it changes the differential signal to digital signal. Figure 3 -12 illustrates the interface capability of the SJA1000 for the connection to a variety of microcontrollers, like 80C51 etc., and CAN transceiver circuits. For connection to the host controller, the SJA1000 provides a multiplexed address/data bus and additional read/write control signals. The SJA1000 could be seen as a peripheral memory mapped I/O device for the host controller. The 80C51 microcontroller i s responsible to receive the data from SJA1000 and then convey the received data to FPGA. Finally FPGA is responsible to calculate the tire turning angle according to the data from 80C51 and then grabs the distance at this angle. We set the right forward degree "zero" .Then we just considerate the range from +7 to -7 degrees because CC or ACC system is often used in highway which doesn 't have large scale turn.

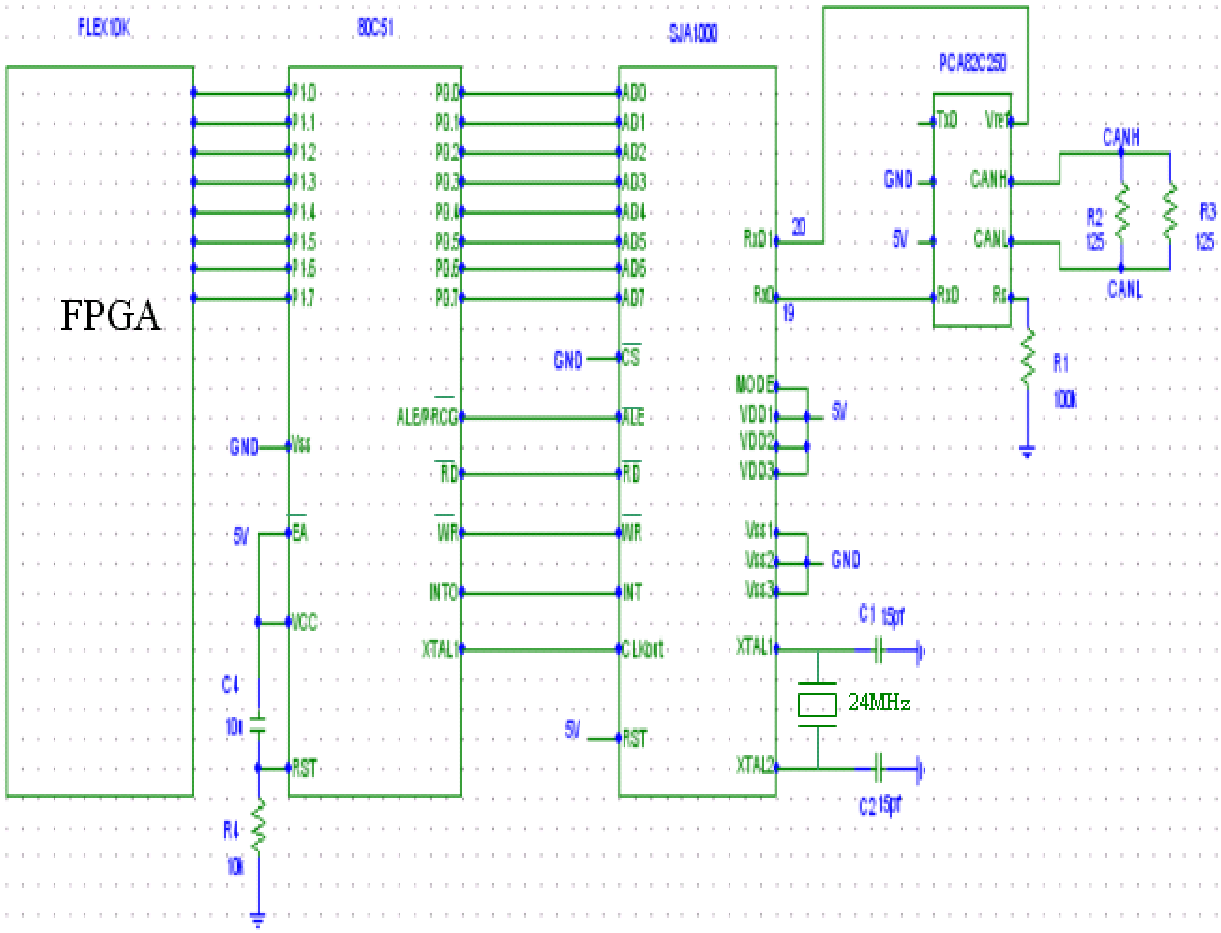
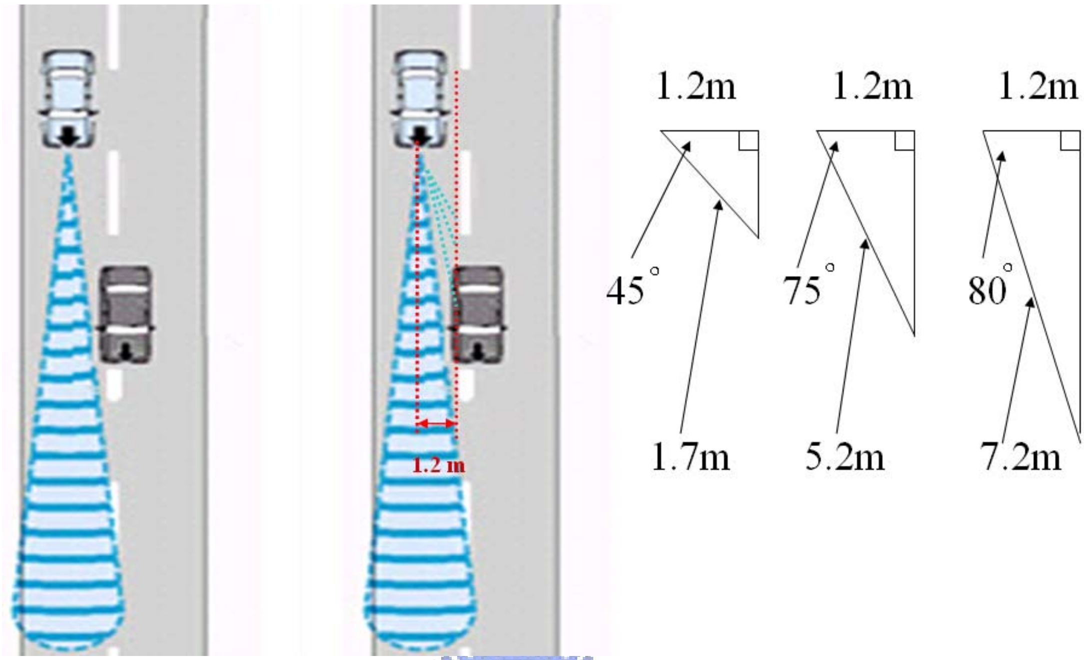


Figure 3-12 The connection chart of PCA82C250 、SJA1000 、80C51 and FPGA

There is another common case as shown in Figure 3-13(a). A vehicle may be not in the right front of our vehicle. It is in the lateral of our vehicle moving lane .The solution carried out is to check the d istance of angle 45, 75 and 80 degree s if it is over 1.7m, 5.2m and 7.2m respectively shown in Figure 3 -13(b). If any distance of angle 45 , angle 75 and angle 80 is shorter than 1.7m , 5.2m and 7.2m respectively, we will deal with the situation under the sh ortest distance among these distances. (The reason to select 1.7m, 5.2m and 7.2m is showed in Figure 3-13(c). The 1.2m represents the distance between center line and the side of vehicle) If the danger situation is not happened, we will operate in the normal mode. The remedial measure is trying to scan a lane, not just a fan-shaped. Because the laser scanner scans from right to left , there are some problems in scanning the left side of the land. Therefore we only consider the right half lane.



(a) (b) (c)

Figure 3-13 Another problem and the solution of it [38]

3.5 Throttle interface

Here we introduce how we control the speed. Figure 3-14 shows the location of throttle body and in Figure 3-15 shows the structure of throttle body .We exploit the principle that the throttle position can adjust the mass of gasoline and air into the engine. If the throttle position was pulled opener, the more fuel will be in the engine and the vehicle will run faster. If the throttle was set free, the less fuel will be in the engine and the vehicle will slow down. Besides, the throttle is adjusted by a tighter rope. If the pedal pulls tighter than DC -motor does, the throttle will be controlled by pedal. If the DC -motor pulls tig hter than pedal does, the throttle will be controlled by DC -motor as shown in Figure 3-15. When FPGA handles this vehicle, we suppose driver's foot is not on the pedal, and throttle is not affected by the driver. Throttle position is fully adjusted by DC -motor. Therefore, we can control the speed through adjusting the throttle position indirectly.

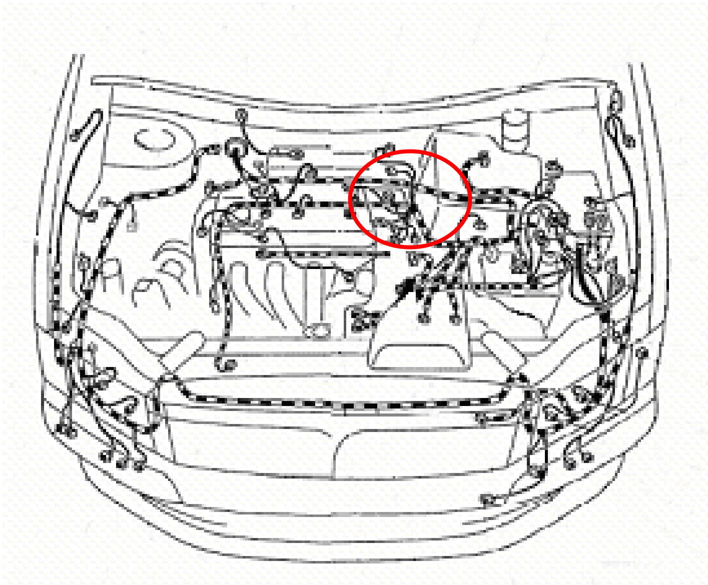


Figure 3-14 The location of throttle body [36]

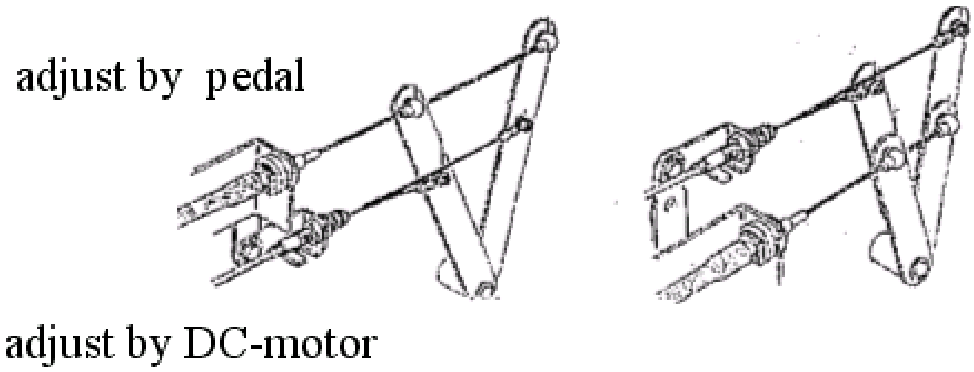


Figure 3-15 The structure of throttle body [39]

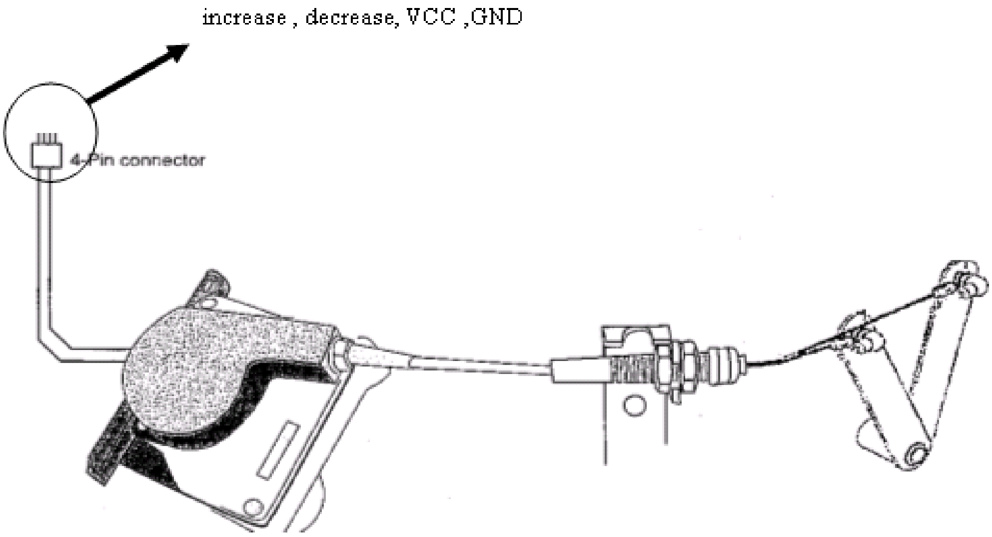


Figure 3-16 The method how DC-motor adjusts the throttle [39]

Next, our problem is to control the throttle position. We use a DC-motor to pull or set free the throttle as Figure 3-16 shows. If the DC-motor pulls the throttle, the position of throttle will be opened and the air-gasoline mixture can flow into the engine. Hence the vehicle will be speeded up. If the DC-motor sets free, the position of throttle will be closed and the air-gasoline mixture can not flow into the engine. Hence the vehicle will be slowed down. Finally, we can achieve our goal through the way. We finish our short-distance target to control the speed.

While the DC-motor is started, it needs almost 0.7 A current which can not be supplied by FPGA itself along. Thus we must have a driver circuit to drive the motor. Here we use a chip named TLP250. It is because TLP250 can allow the different voltage in the left side and right side and these different voltages are provided from independent voltage source. The left side voltage of TLP250 is 3V and the right side voltage of TLP250 is 12V. Besides if there is a digital signal which is 0V or 3V between Pin 2 and Pin 3, then there will be a digital signal which is 0V or 12V between Pin 7 and Pin 6. Figure 3-17 shows the pin assignment of TLP250.

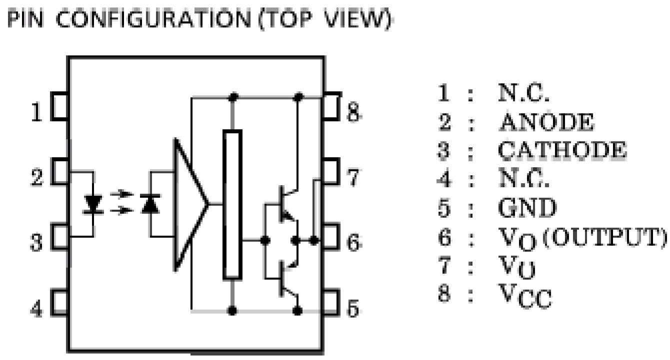


Figure 3-17 the pin assignation of TLP250 [40]

We should pay attention to the DC-motor because it is not a typical DC-motor. If there is a constant signal to a DC-motor, typical it will run endless. However, it is not true in our case. In our case the DC-motor will not move if we arrive the minimum or maximum boundary. The DC-motor will automatically cut off the command signal into it at the minimum or maximum boundary. Hence the typical DC-motor driver, H-type circuit, is not suitable.

Following the TLP250 is the class B output stage which consists of a complementary pair of transistors (an npn transistor and a pnp transistor) connected in such a way that both cannot conduct simultaneously. It can amplify the current to about 1 A which can drive the DC-motor. Figure 3-18 shows the layout of the driver circuit. Here we use the pulse to drive the motor. When the signal is high, the motor is driven. When the signal is low, the motor is unmoved. Here we don't use typical PWM signal because PWM is good at holding exact position but it is harmful to this motor.

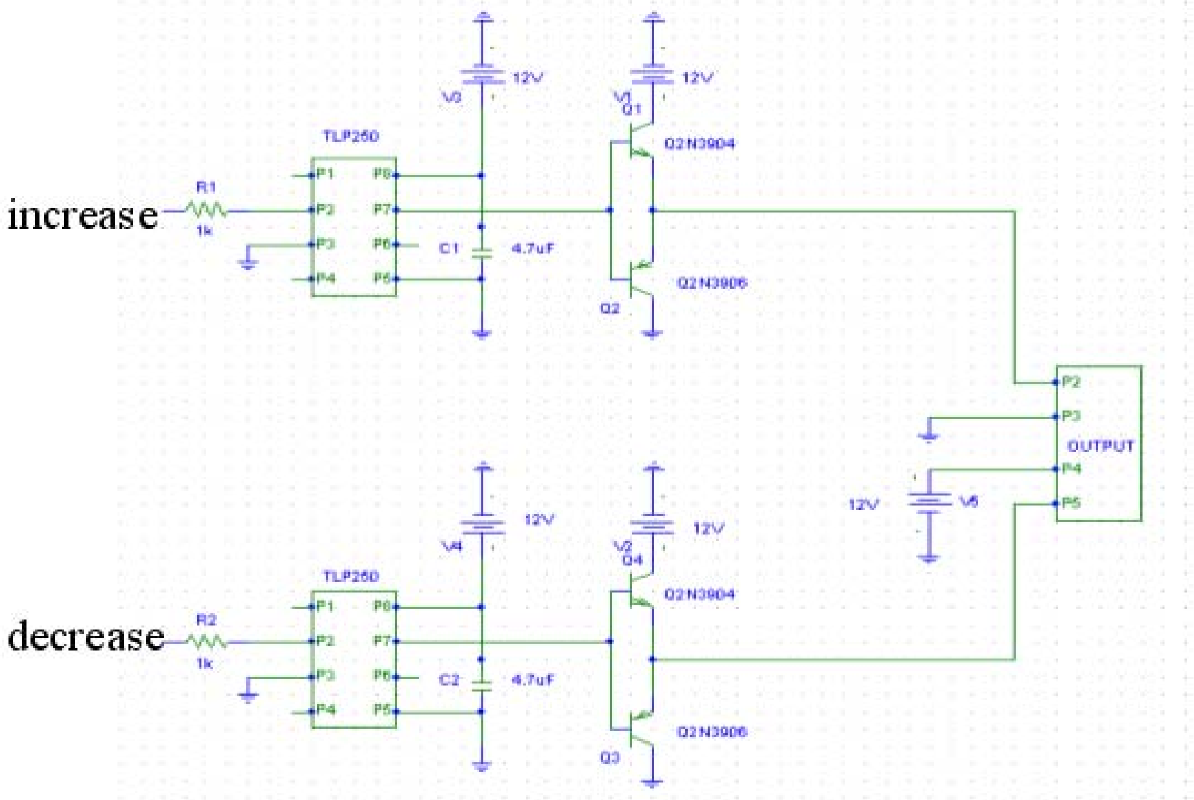


Figure 3-18 The layout of driver circuit

3.6 Disable device

Although the ACC system can handle most conditions, we could not forget that ACC system is a service to help the driver, not a replacement of the driver. The driver is still in charge of the car at any moment, regardless if the ACC system is active or not. Furthermore it is possible that the sensor may be out of control or lost its function because of the touch between two lines or power source is not stable or the abrasion of DC-motor,…..,etc. Hence we need a way to disable the ACC controller and change the control right to driver. The sample way is to cut off the power of DC-motor which will relax the throttle fully and then the throttle position is adjusted by driver. The next step is to choose a rule to decide whether one should turn off power or not. The driver will step on the brakes with great exertion when they encounter an emergent condition. For this reason we install a load cell which is the same function as pressure sensor on the brakes pedal to measure the pressure. The more pressure is on it, the higher voltage will output from the load cell. Besides the load cell needs an amplifier to amplify the output signal of it. Here we use ADAM-3016 of ADVANTECH to be the amplifier. Figure 3-19 shows the appearance of load cell and the circuit to determine whether one should turn off the power of DC-motor or not. Because some drivers are used to step on the brakes slightly, we have to adjust the R5 to decide the boundary voltage. If we relax the load cell the controller will re-accelerate according to real-time condition. However common cruise controller will be disabled once you put your leg on brake. And then, if you relax the brake you will have to re-accelerate by yourself. Besides, if we really want to disable the CC and ACC function, we can either push the reset key on the development board or turn off the power of it.

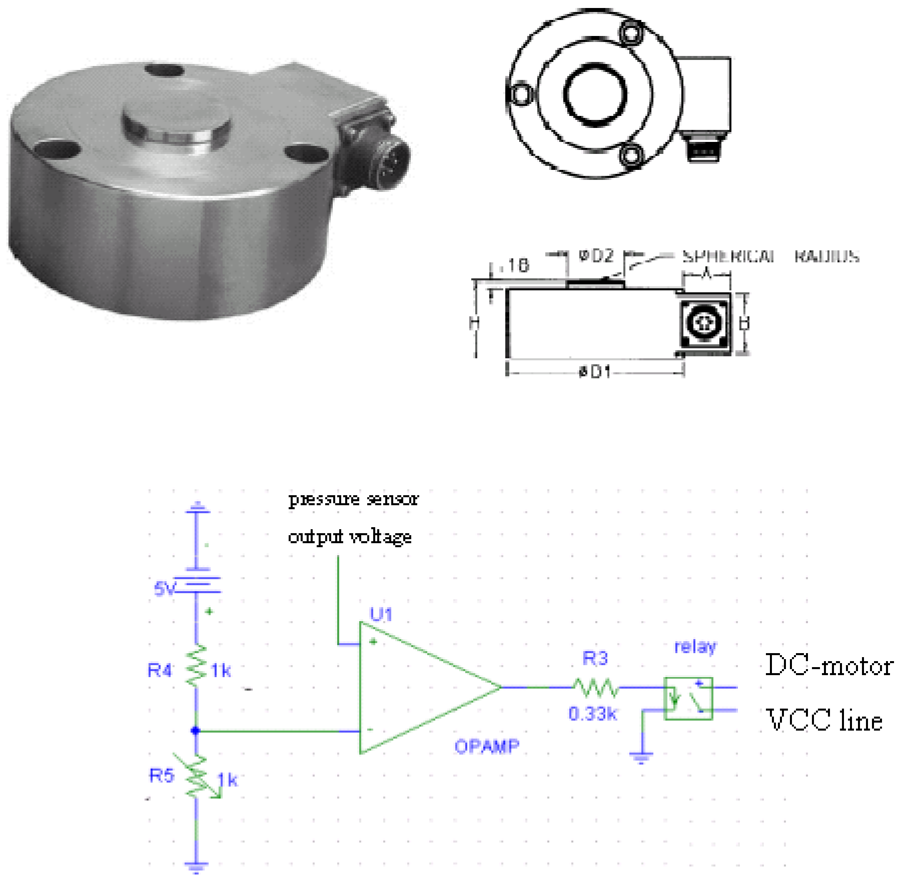


Figure 3-19 The appearance of load cell and the circuit to deter whether turn off the power of DC-motor or not [41]

Chapter 4 Fuzzy Logic Controller

Fuzzy logic control is a useful methodology for system control in the presence of uncertainties and disturbances. The independence of expert knowledge of the controlled plants is a remarkable advantage in fuzzy logic controller (FLC) design. Besides, the control actions are usually smoothed through the well-built control rules. The basic principles of fuzzy logic were introduced by Zadeh in 1965 [42]. Because of the advantages such as easy implementation, suitability for complex dynamic systems, and high flexibility and robust nature, fuzzy controllers have been implemented in many fields.

The characteristic of FLC is that it adopts the linguistic control strategy to control plants without realizing their mathematical models. The linguistic control strategy of FLC is constructed according to the operator experience and/or expert knowledge. Experiences show that the FLC yields results superior to those obtained by traditional control algorithm in the complex situation where the system model or parameters are difficult to obtain.

Typically, fuzzy controllers are based on four well-known stages : a fuzzification interface, a rule base, an inference engine, and a defuzzification interface as shown in Figure 4-1. More detail descriptions for each stages are stated below.

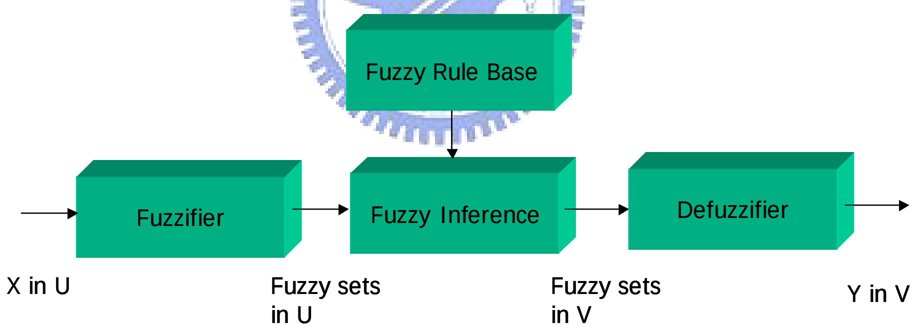


Figure 4-1 Fuzzy system architecture

4.1 Fuzzy set and set-theoretical operators

Definition 4.1 *Fuzzy Set*: Let U be a collection of objects, for example, $U = R^n$, and be called the universe of discourse. A fuzzy set F in U is characterized by a membership function $\mu_F : U \to [0,1]$, with $\mu_F(u)$ representing the grade of membership of $\mu \in U$ in the fuzzy set F.

Definition 4.2 *Support, Fuzzy Singleton*: The support of a fuzzy set F is the point(s) $u \in U$ at which $\mu_F(u)$ achieves its maximum value. If the support of a fuzzy set F is a single point in U at which $\mu_F = 1$, the F is called a fuzzy singleton.

Definition 4.3 *Intersection, Union, and complement*: Let A and B be two fuzzy sets in U. The intersection A $\cap$ B of A and B is a fuzzy set in U with a membership function defined for all $u \in U$ by

$$\mu_{A \cap B}(u) = \min\{\mu_A(u), \mu_B(u)\} \tag{4.1}$$

The union of A $\cup$ B of A and B is a fuzzy set in U with the membership defined for all $u \in U$ by

$$\mu_{A \cup B}(u) = \max\{\mu_A(u), \mu_B(u)\} \tag{4.2}$$

Usually, the intersection and union operators are denoted by $\wedge$ and $\vee$, respectively. The complement $\bar{A}$ of A is a fuzzy set in U with the membership function defined for all $u \in U$ by

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \tag{4.3}$$

4.2 Fuzzifiers

The fuzzifier stage transforms crisp input from real values into fuzzy sets. Here we introduce two fuzzifiers as following:

1. Singleton fuzzifier: the singleton fuzzifier maps a real valued point $x^* \in U$ into the fuzzy singleton *A* in *U*, in which the membership value is 1 at x^* and 0 at other points in *U*, i.e.,

$$\mu_A(x) = \begin{cases} 1 & x = x^* \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

2. Triangular fuzzifier: the triangular fuzzifier maps $x^* \in U$ into the fuzzy set *A* in *U*, in which the membership function is written as:

$$\mu_A(x) = \begin{cases} (1 - \dfrac{|x_n - x_n^*|}{b_n}) & \text{if } |x - x^*| < b_i, i = 1,2,\ldots n \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Here b_i are positive parameters.

Finally, we summarize the above fuzzifiers. The singleton fuzzifier greatly simplifies the computations involved in the fuzzy inference engine for all membership functions. If the computations are more complex, it means the more CLB will be needed and the price will raise. It is the thing we do not want. Hence we use singleton fuzzifiers in the simpler CC controller. Although the triangular fuzzifier is more complex than singleton, it can restrain noise in the input which singleton fuzzifier cannot. Hence we balance these methods. We use discrete triangular fuzzifier like Figure 4-2 in the more complex ACC controller.

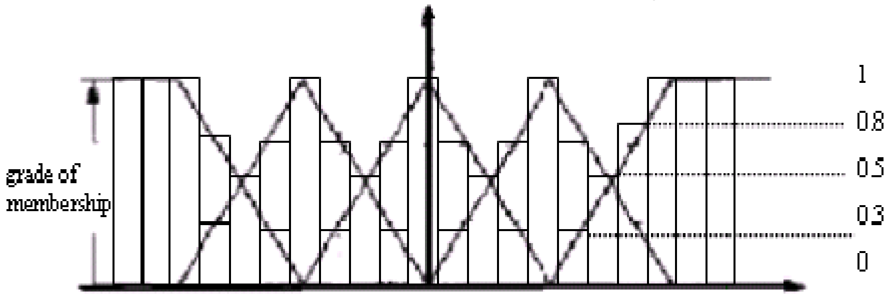


Figure 4-2 Discrete triangular fuzzifier

4.3 Defuzzifiers

The defuzzifier is defined as a mapping from a fuzzy set D in $V \subset R$ to a crisp point $y^* \in V$. Hence, the task of the defuzzifier is to specify a point in V that represents the fuzzy set *D*. There are three types of defuzifiers introduced below.

1. Center of area Defuzzifier (COA)

The center of gravity defuzzifier specifies y^* as the center of the area covered by the membership function of *D*.

$$y^* = \frac{\int_V y\mu_D(y)dy}{\int_V \mu_D(y)dy}, \tag{4.6}$$

where $\int_V$ is the conventional integral.

In the case of discrete universe, the (4.6) will be changed to

$$y^* = \frac{\sum_v y\,\mu_D(y)}{\mu_D(y)} \tag{4.7}$$

2. Center Average Defuzzifier

Let $\bar{y}^l$ be the center of the lth fuzzy set and w_l be its height. The center average defuzzifier presents y^* as

$$y^* = \frac{\sum_{l=1}^{M} \bar{y}^l w_l}{\sum_{l=1}^{M} w_l} \tag{4.8}$$

3. Mean of maximum (MOM)

The MOM defuzzifier generates a control action that represents the mean value of all local control actions whose membership functions reach the maximum. In the case of a discrete universe , the control action may be expressed as

$$y^* = \sum_{j=1}^{m} \frac{y_j}{m} \tag{4.9}$$

where y_j is the support value at which the membership function reaches the maximum value and m is the number of such support values.

Finally, we summarize the above defuzzifiers. The COA strategy has been shown to yield superior results. Furthermore, the MOM strategy yields a better transient performance and simplifies the computations, while the COA strategy yields a better steady-state performance (lower mean square error). For these reasons we implement defuzzifier by COA method.

4.4 Fuzzy rule bases

The fuzzy rule base consists of fuzzy **IF-THEN** rules. It is the core of the fuzzy system in a sense. And all other stages are used to implement these rules in a reasonable and efficient manner. The general form of the fuzzy control rules in our case is a multi-input-single-output system (MISO). Hence, the fuzzy rule base comprises the following fuzzy **IF-THEN** rules:

Rule i: **IF** x_l is A_1^i and …and x_n is A_n^i **THEN** y is D^i (4.10)

The canonical fuzzy **IF-THEN** rules in the form of (4.10) includes the following ones:

(1) Partial rules:

IF x_1 is A_1^i and …and x_m is A_m^i **THEN** y is D^i (4.11)

(2) Or rules

IF x_1 is A_1^i and …and x_m is A_m^i or x_{m+1} is A_{m+1}^i and … x_n is A_n^i **THEN** y is D^i (4.12)

(3) Singles fuzzy statement

y is D^i (4.13)

4.5 Fuzzy inference

The fuzzy inference is a reasoning method using the fuzzy theory, and whereby the expert knowledge is presented using linguistic rules. The fuzzy inference is introduced as following.

Max-min Inference: $\mu_D(y) = \max_{l=1}^{M}[\sup_{x\in U} \min(\mu_A(x), \mu_{A_1^l}(x_1)..., \mu_{A_n^l}(x_n), \mu_{D^l}(y))]$ (4.15)

The Max-min inference is the most commonly used fuzzy inference in the fuzzy system and other fuzzy applications. Figure 4-3 shows the Max-min inference.

In order to simplify the computations and implement it more easily in the fuzzy inference engine, we introduce the singleton method. It is because the result is a region in tradition fuzzy inference and the result is a singleton in singleton inference. Figure 4-4 shows this inference [33]. The result is just a singleton, whose degree is determined by max-min inference, and location is determined by the section in the fuzzifier. For example the two inputs map to the locations, I_1 and I_2, according to fuzzifier, and then the degrees of them are operated through Max-min inference.

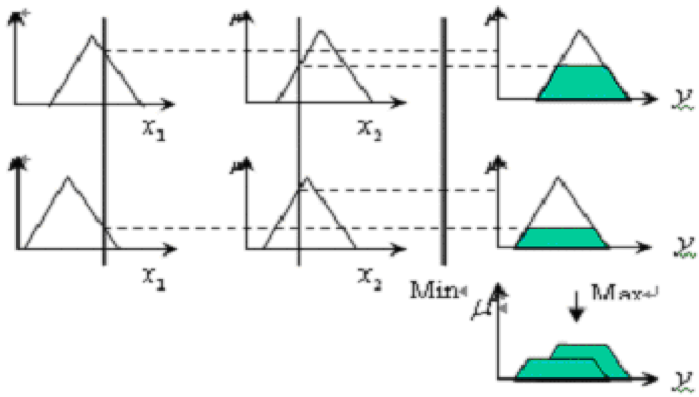


Figure 4.3 Max-min inference

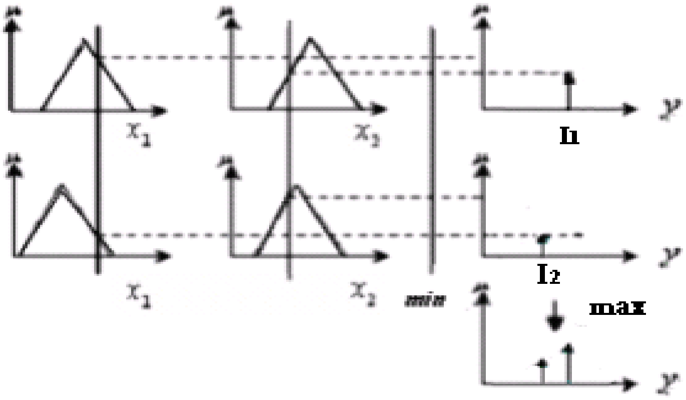


Figure 4.4 Max-min singleton inference

4.6 Longitudinal controller

In this section, we design a FLC to realize the ACC for a vehicle. Before we start our plan, let's review the overall architecture in Figure 4-5. The first thing in our controller is to check whether the real-time distance is inside the safety distance or not. Here the safety distance is up to traffic rule. If the real-time distance is inside safety distance, throttle will be set free rapidly and repeat as soon as possible.

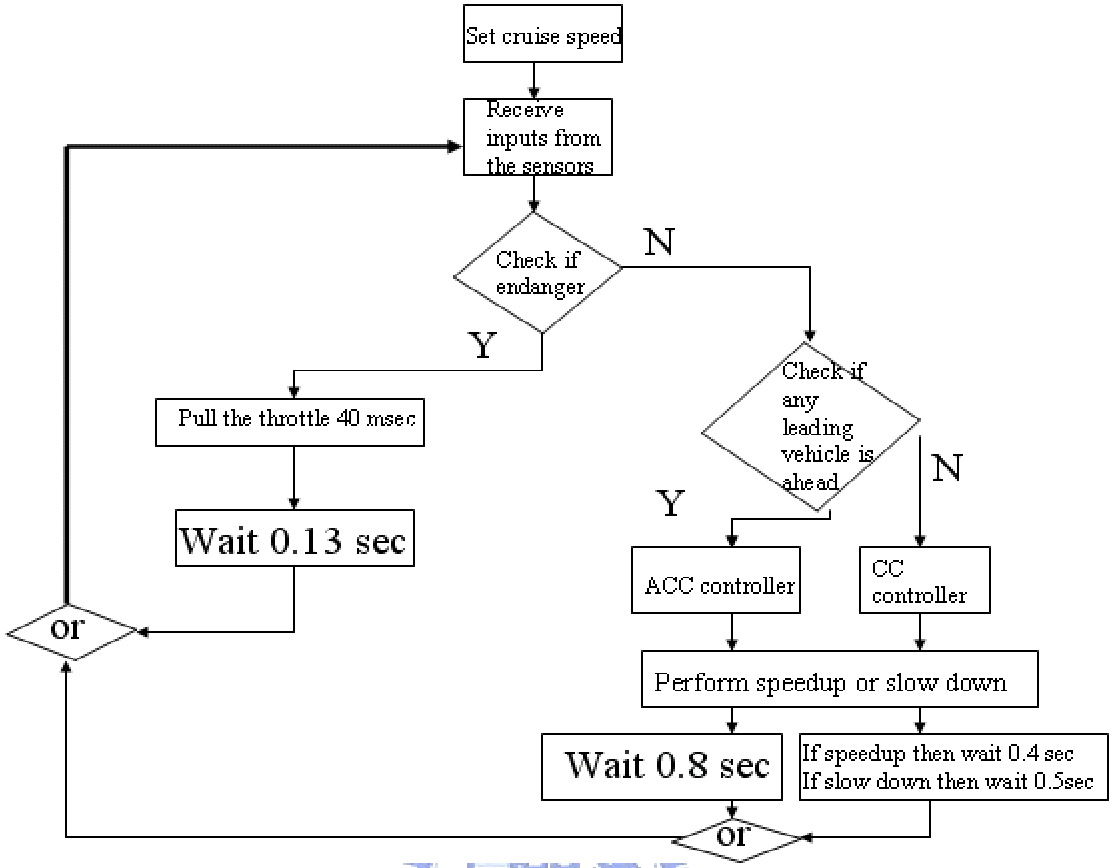


Figure 4-5 Overall architecture

The main idea is to control the speed over the position of throttle which is adjusted by the DC-motor. It is because throttle position controls the mass of air-gas intermix into the engine. For this reason, we can control the speed by DC-motor indirectly. The fuzzy logic controller is designed to determine the DC-motor should pull or push and should work how much time. The architecture is different from commercial system which uses P controller because it performs fuzzy theory to improve performance. In our system it keeps CC mode or ACC mode as other commercial systems. We change between CC and ACC by checking if the distance between our vehicle and forward vehicle is shorter then 40 m. If a vehicle is less than 40 m away from our car, we will change into the ACC mode. If there is no vehicle less than 40 m away form, even there is no vehicle in the lane, we will change into the CC mode.

4.6.1 Cruise Control (CC) Controller

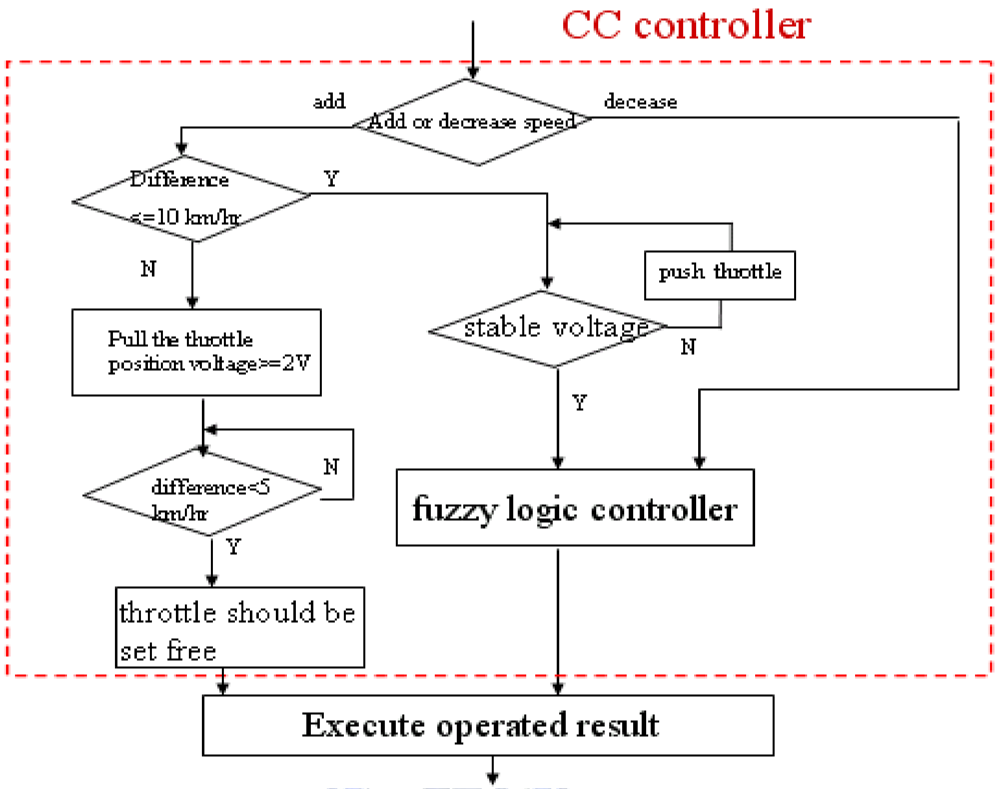


Figure 4-6 CC system flow chart

First, we focus on the CC controller and let's see its flow chart as shown in Figure 4-6. We roughly cut the flow chart into two parts, control by throttle voltage and control by FLC. The action which is controlled by throttle voltage attempts to reach the rough steady-state voltage and rough stable speed through the throttle. Here we design fuzzy logic controller only to make a fine adjustment. Therefore, first we need to control throttle to rough target speed. If we control speed only dependent on FLC, the target speed will also be reached. However, it adjusts quiet slowly and has large overshoot due to delay. We refer to the CC system in reference [31] as shown in Figure 4-7.

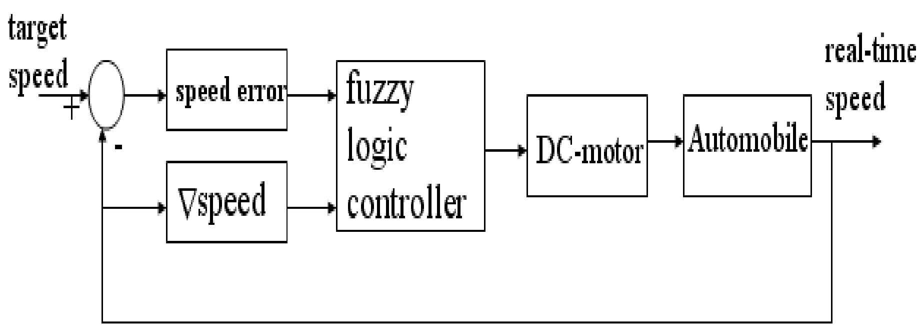


Figure 4-7 The block diagram of CC system

Fuzzifier: We define each situation of two input variables, speed error and speed variation in one cycle, of the FLC [31]. First we cut each input into 5 intervals. Speed error is cut into “DM: decrease middle, DS: decrease small, ZO: zero, AS: add small, AM: add middle”. Speed variation in one cycle is cut into “MS: middling slow, LS: little slow, ZO: zero, LQ: little quick, MQ: middling quick”. Finally, we build a ROM to memorize the input- interval map. From now on, we just substitute the input variables and then we can know the plan interval.

Fuzzy rule base: After we define the input intervals, we have to determine which actions should be taken under the conditions. The actions are fuzzy rules. They are designed according to our experiences. We establish the fuzzy rule in Table 4-1 to check how to do under the situations.

Table 4-1 The fuzzy rule of CC system

		-5	-2	0	2	5 (km/hr)
	speed error / ∇ speed	DM	DS	ZO	AS	AM
-2	MS	IB	IB	IM	IS	ZO
-1	LS	IB	IM	IS	ZO	DS
0	ZO	IM	IS	ZO	DS	DM
1	LQ	IS	ZO	DS	DM	DB
2	MQ	ZO	DS	DM	DB	DB

(km/hr · one period)

IS: increase small IM: increase medium IB :increase big ZO: zero DS: decrease small DM: decrease medium DB: decrease big

We refer to the singleton fuzzifier in [34] to greatly simplify the computations. For example if the speed falls into the DM and the speed variation in one cycle falls into the MS, the degree of IB will be 1 and the degree of others will be zero. This way is just grabbing the input and mapping to the output without any fuzzy inference and defuzzifier. We set up the output as Figure 4-8 shows.

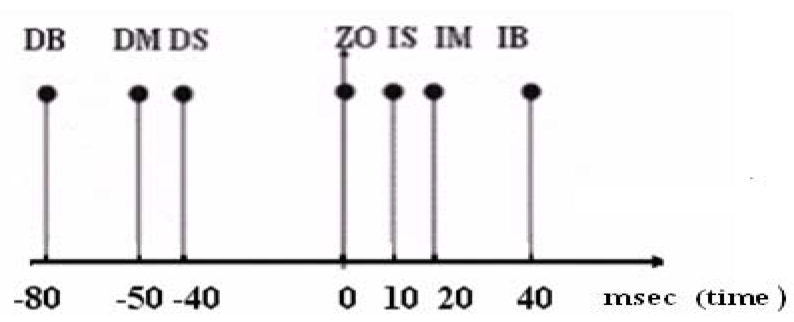


Figure 4-8 Fuzzy output value (CC)

4.6.2 Adaptive Cruise Control (ACC) Controller

Let's see ACC flow chart in Figure 4-6. Before flowing into fuzzy logic controller we roughly cut the flow chart into two parts, control by throttle and control by FLC. If we control speed only dependent on FLC, in most situations at the moment controller should speedup controller increases the throttle voltage from 0.6V which is the lowest throttle voltage. It is because in most situations the throttle has been set free before we speed up. Increasing from 0.6V is too late to catch up to the front vehicle and then we will lose the vehicle or controller will create too large acceleration. The large acceleration will cause the relative distance too close and cause the distance to oscillate largely. If we raise voltage to its rough stable value rapidly at the moment FLC transforms determination from slow down to hold or speedup, we will catch up to the front vehicle and the controller won't create large momentum and avoid large distance oscillation. FLC will grab the information from sensors and then make an adjustment. Here we choose our safety distance according to the half of real-time speed. If the real-time speed is 60 km/hr, the ideal distance will be 30 m. It is a varying number, not a fixed number.

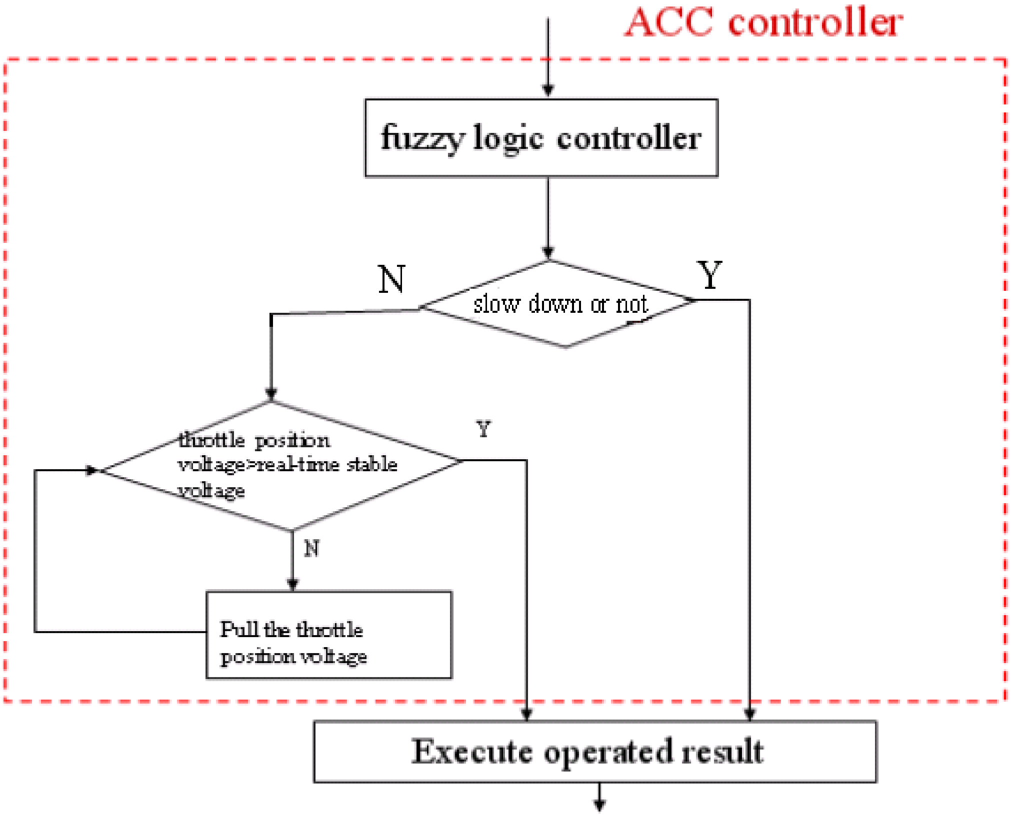


Figure 4-9 ACC system flow chart

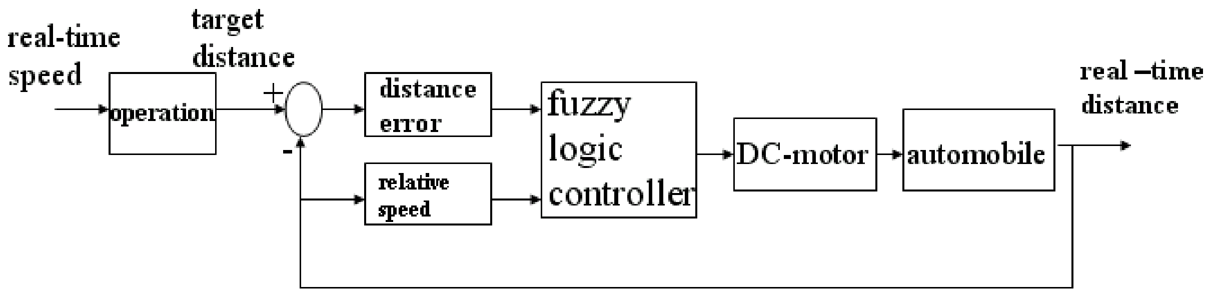
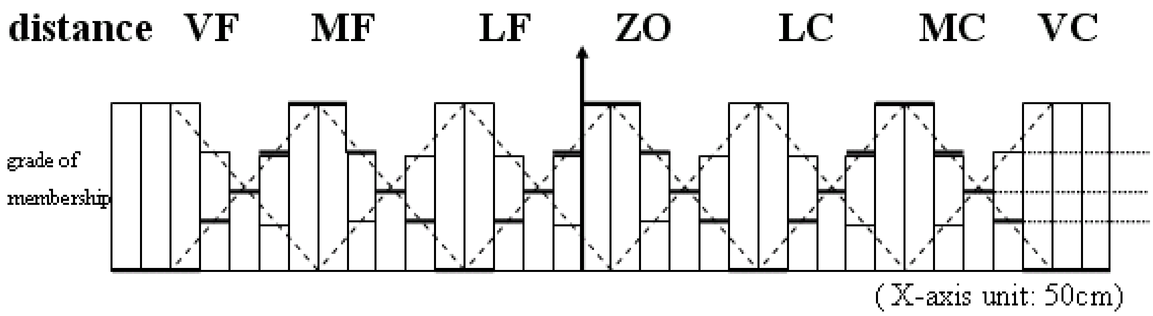


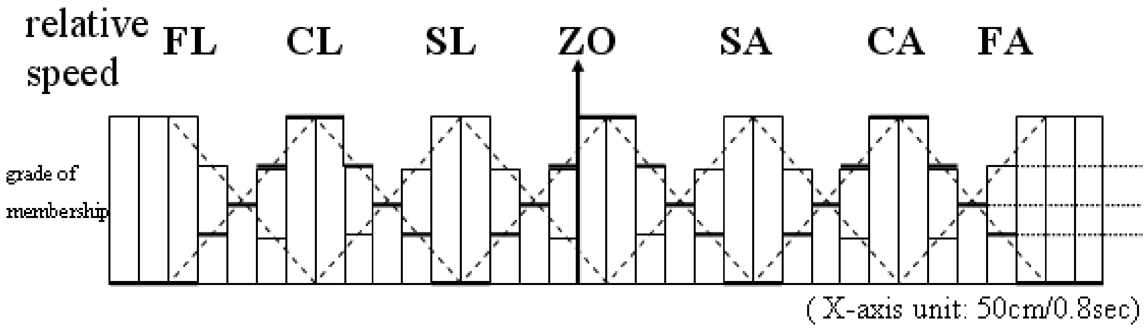
Figure 4-10 The block diagram of ACC system

Fuzzifier: We define each situation of two input variables, distance and relative speed, of the FLC [31] as shown in Figure 4-10. First we use the triangular membership to represent the degree of this situation. Because it is difficult and it needs a lot of CLB to detect every input membership degree [32], we cut each interval to some sub-intervals. Besides, we cut the degree to 5 level values: 0, 0.3, 0.5, 0.8 and 1. Finally, we build a ROM to memorize the input- degree map. From now on, we just input the substitute variables and then we can know the degrees and situations.



VF: very far MF: medium far LA: little far ZO: zero

LC: little close MC: medium close VC: very close



FA: fast approach CA: common approach SA: slow approach

ZO: zero FL: fast leave CL: common leave SL: slow leave

Figure 4-11 Membership function for ACC input variables

Fuzzy rule base: After we define the input degrees and conditions, we have to determine which actions should be taken under the conditions. The actions are fuzzy rules. They are formed according to our experiences. We establish Table 4-2 to check how to do under its current situations.

Table 4-2 The fuzzy rule of ACC system

d v	VF	MF	LF	ZO	LC	MC	VC
FL	IF	IF	IF	IB	IM	IS	ZO
CL	IF	IF	IB	IM	IS	ZO	DS
SL	IF	IB	IM	IS	ZO	DS	DM
ZO	IB	IM	IS	ZO	DS	DM	DB
SA	IM	IS	ZO	DS	DM	DB	DF
CA	IS	ZO	DS	DM	DB	DF	DF
FA	ZO	DS	DM	DB	DF	DF	DF

IF: increase far IB: increase big IM: increase medium IS: increase small ZO: zero

DS: decrease small DM: decrease medium DB: decrease big DF: decrease far

The output degree will belong to 5 level values: 0,0.3,0.5,0.8 and 1 due to the fuzzy input.

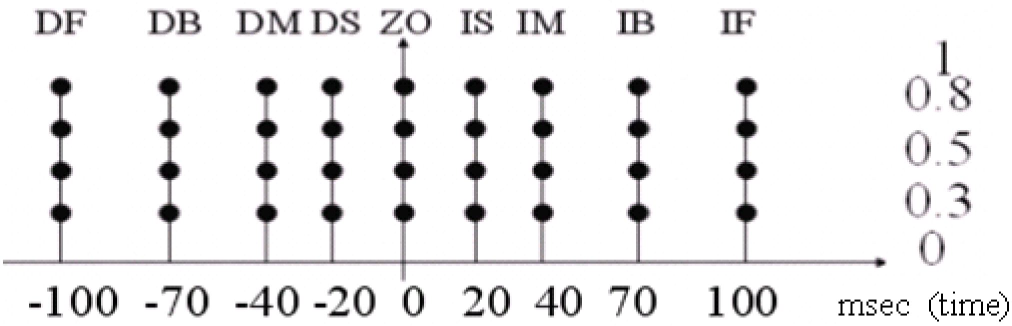


Figure 4-12 Fuzzy output value (ACC)

Fuzzy inferences: We select the max-min and singleton reasoning methods to carry out the algorithm. In order to explain it clearly, we take Figure 4-13 as an example. In Figure 4-13, a, b, c and d are membership degrees.

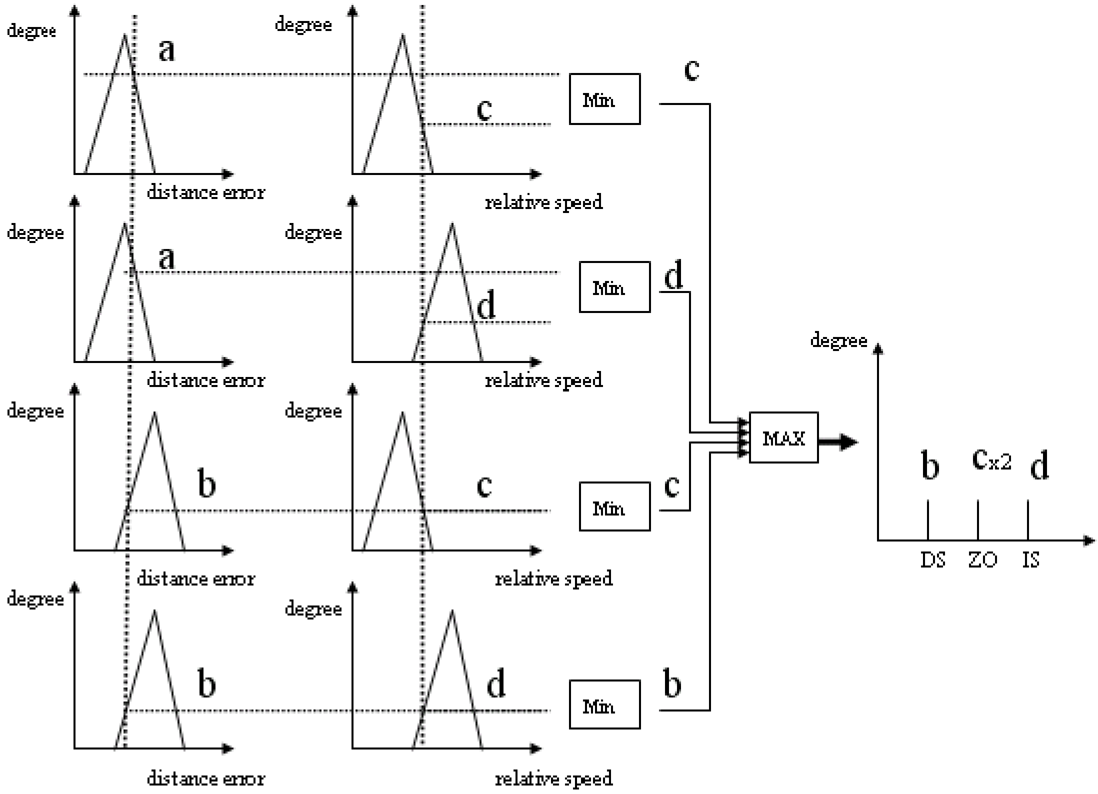


Figure 4-13 Max-min and singleton inference

Defuzzifier : We use the most frequently used defuzzifier method, the center-of-gravity defuzzification, in ACC system

Figure 4-14 shows the block diagram chart to implement the COA defuzzifier as (4.7). Firstly, we focus on the up half part. The symbol wi means the i-th degree and yi means the i-th output variable value. The first step is to multiply wi by yi . And then, the multiplicative result is accumulated into the register 1. The register accumulates the multiplicative result four times and then the accumulative result is pushed to the register 2. After the register 2 catch the data, the register 1 is cleared to restart the other accumulative cycle. The register 3 is to accumulates the degree four times and then push the accumulative result to the register 4. After the register 4 catch the data, the register 3 is cleared to restart the other accumulative cycle. Finally, the register 2 is considered as a numerator and the register 4 is considered as a denominator .There is a division to excise the operation .The final result is created after the operation. The result greater than zero means we should make the DC-motor pull the throttle. On the contrary, the result smaller then zero means we should make the DC-motor push the throttle. The scale means how long should the DC-motor works.

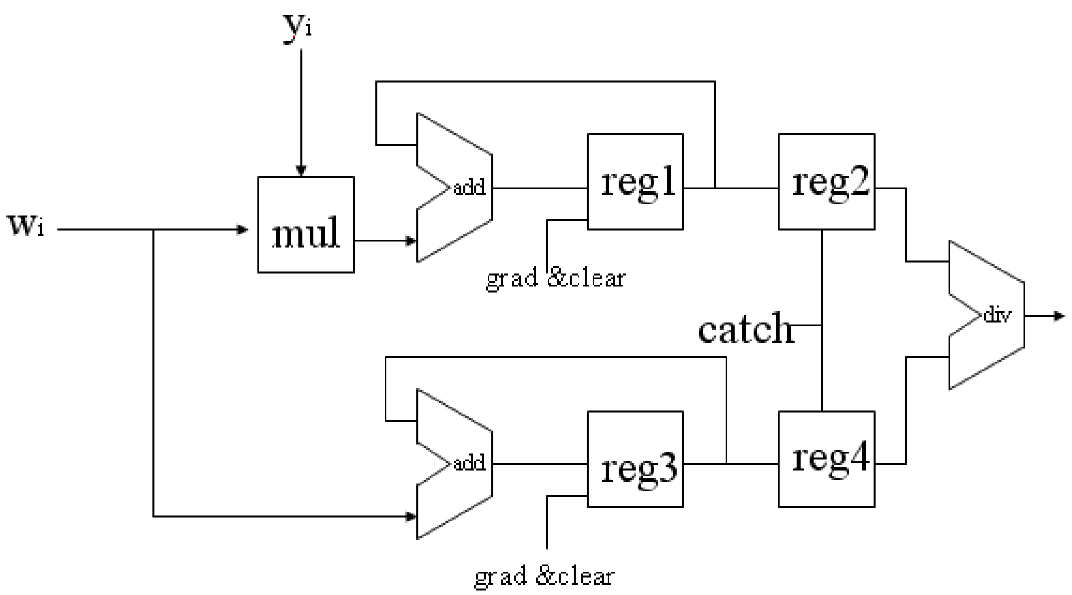


Figure 4-14 Block diagram to implement the COA defuzzifier

4.7 Practically implementation

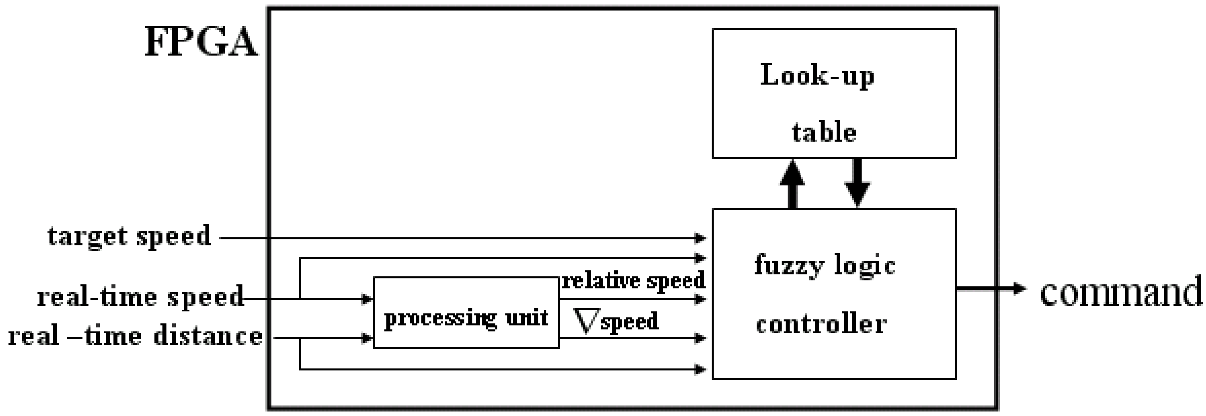


Figure 4-15 Practically implementation a FLC

In practical application, we implement FLC as shown in Figure 4-15 [35]. It is because the fuzzy logic is one-way. If the inputs of FLC are immobile, the operated results will be fixed, not fluctuated. Hence we can infer the output command from the input of FLC. In general we calculate the results from the input of FLC through fuzzy logic. Next we store the inputs and the outputs in the Look-up Table (LUT). It has the same results with calculated results step by step and doesn’t need the multiplication, division, fuzzy inference and etc. It saves not only time but also a lot of CLBs. Hence in our system we use the same way to implement the controller.

4.8 Development Board

The next question is what instrument to implement the Fuzzy logic controller. Here, we make use of the EPF10K100ARC240-1 of FLEX10K series. There are 200K gates and 189 I/O ports in this development board, as shown in Figure 4-16. It doesn't like some other development Boards with RS-232 interface or PS-2 keyboard interface or DSP interface. Therefore we have to program the driver of the peripheral devices in VHDL language. Table 4-3 shows different devices to compare the resources of each development board. From Table 4-3, we can know the resource of FLEX10KEPF10K100ARC is relatively poor. The used logic elements are about 3300, which is about 65% in EPF10K100ARC. The program may look like very long. However FLEX10KEPF10K100ARC is the superannuated FPGA. Besides 3300 is only about 30% in the Nios development board, which is not designed to be FPGA, and just about 12% in the Stratix EP1S25F780C5 development board, which is fully planed to be FPGA. Therefore the program is truly simple and direct.

Table 4-3 Three types of FPGA [43]

Device	Logic elements	Memory bits	I/O pins
FLEX10K EPF10K100ARC	4992	49152	189
NiosII REV1.0	10570	920448	427
STRATIX EP1S25F780C5	25660	1944576	598

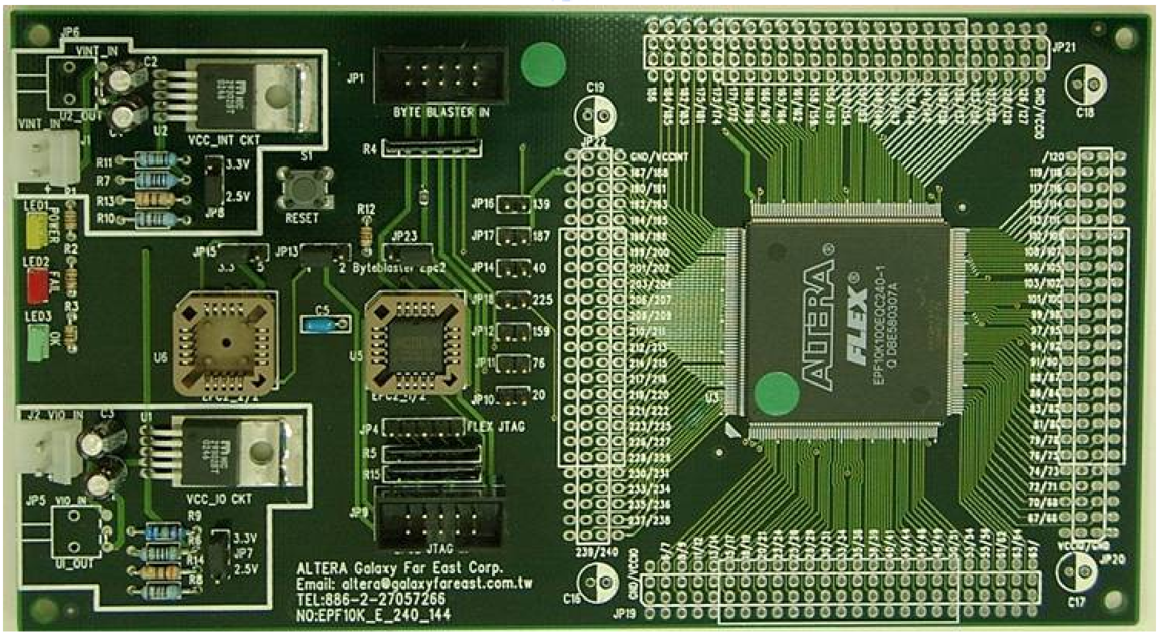


Figure 4-16 Development board-- EPF10K100ARC240-1 [43]

Chapter 5 Experimental Results

Before we start to exhibit our results, we have to explain the practical instruments, vehicle body and design circuits are listed in Appendix. In this chapter we use a device named dspace to record the speed (km/hr), throttle voltage (v) and relative distance (meter). In the chapter we show four different experiment results: hold at a fixed speed, change speed, adaptive cruise control and switch between each mode automatically. The relative distance in Sections 5.1 and 5.2 is farther than 40m so the command from controller will not be affected by it. Hence we only list the relative distance in the Sections 5.3 and 5.4.

5.1 Hold at a fixed speed

5.1.1 Hold at 40 km/hr

First we try to hold at 40 km/hr on a smooth road. Here we also record the speed and throttle voltage of general cruise control to be the reference. The results are displayed in Figure 5-1.

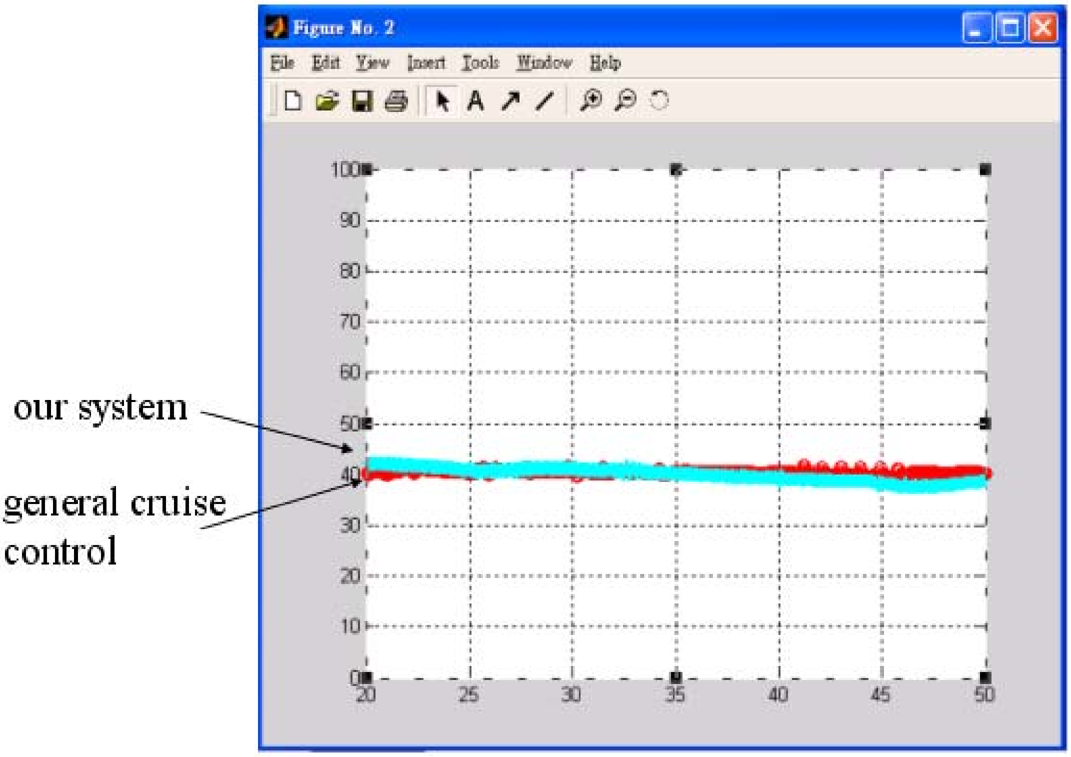


Figure 5-1 The speeds of our system and general cruise control which want to hold at 40 km/hr

The max speed difference of our system is about 2 km/hr. This result is little worse than general cruise control. However, the difference between 40km/hr and real-time speed is no more than 1 km/hr for the most part. Therefore the result is acceptable. And then we display the throttle voltages of our system and general cruise control which want to hold at 40 km/hr in Figure 5-2. We find that the throttle voltage is flatter than the general cruise control. Therefore general cruise control provides more momentum than our system which changes the throttle slowly. It means that the reaction time of our system is longer than that of general cruise control. On the contrary it means that driver will feel more comfortable in the vehicle which has installed our system.

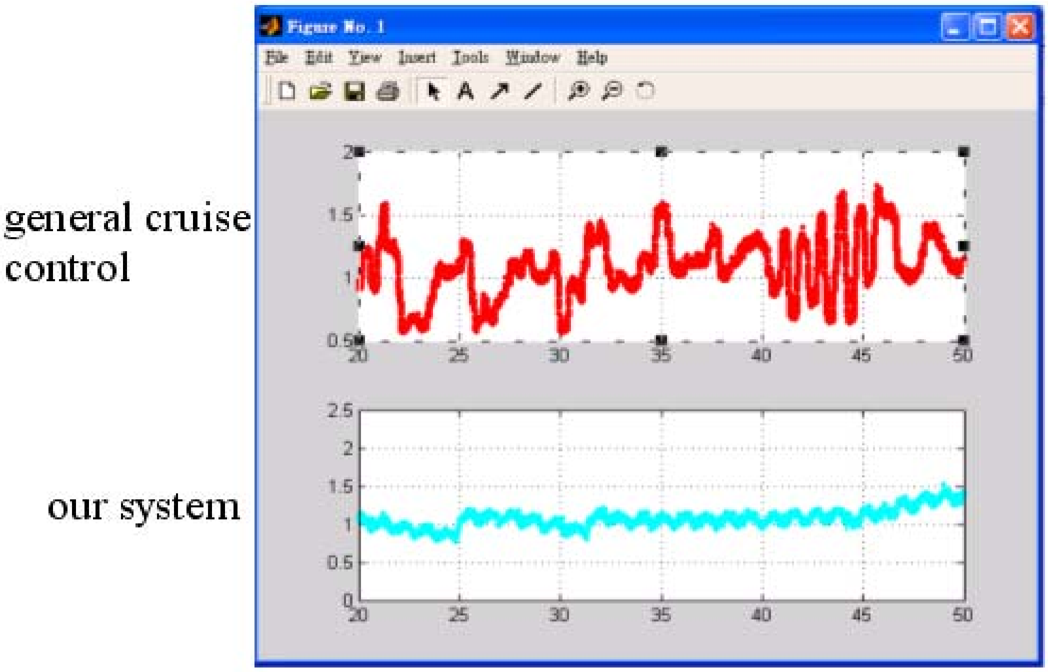


Figure 5-2 The throttle voltage of our system and general cruise control which want to hold at 40 km/hr

5.1.2 Hold at 60 km/hr

First we try to hold at 60 km/hr on a smooth road. Here we also record the speed, throttle voltage of general cruise control to be the reference. The results are shown in Figure 5-3. The max speed difference of our system is about 2 km/hr. This result is little worse than general cruise control. However, the difference between the desired speed (60km/hr) and real-time speed is within 1 km/hr for the most part. The result of our system almost overlaps the result of general cruise control. Therefore this result is great. The throttle voltages of our system and general cruise control held at 60 km/hr are shown in Figure 5-4. We find that the throttle voltage is flatter than one of the general cruise control. Therefore general cruise control provides more momentum than our system which changes the throttle slowly. It means that the reaction time of our system is longer than that of general cruise control. On the contrary it means that driver will feel more comfortable in the vehicle which has installed our system.

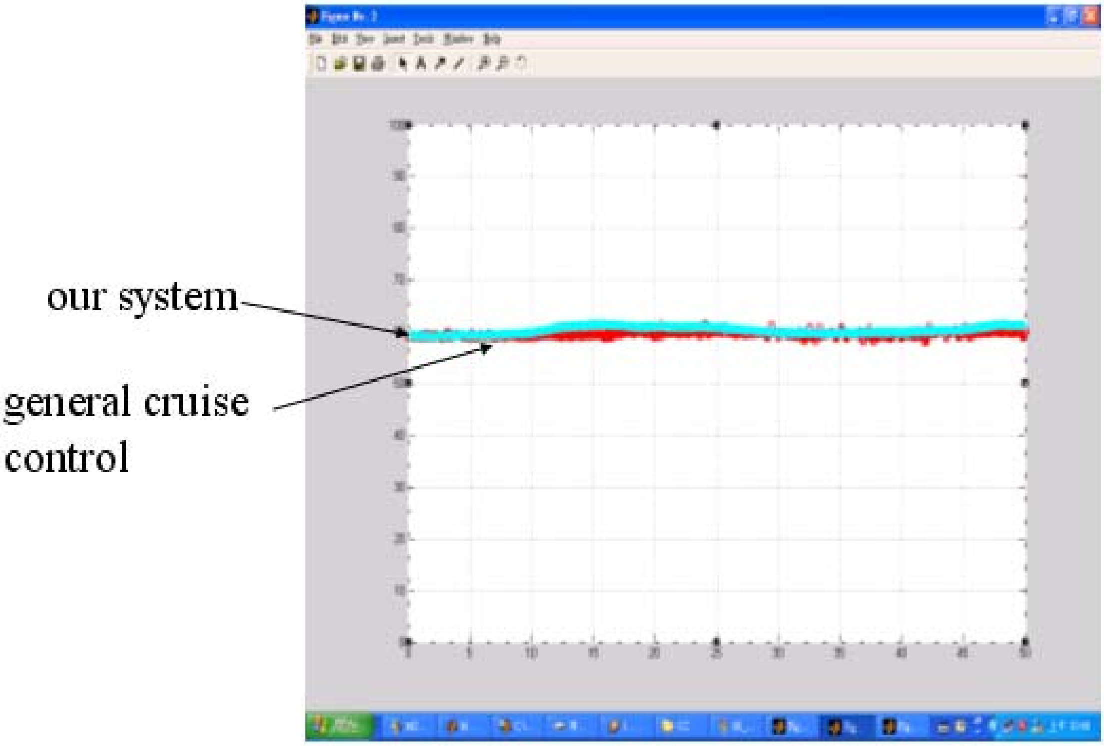


Figure 5-3 The speeds of our system and general cruise control which want to hold at 60 km/hr

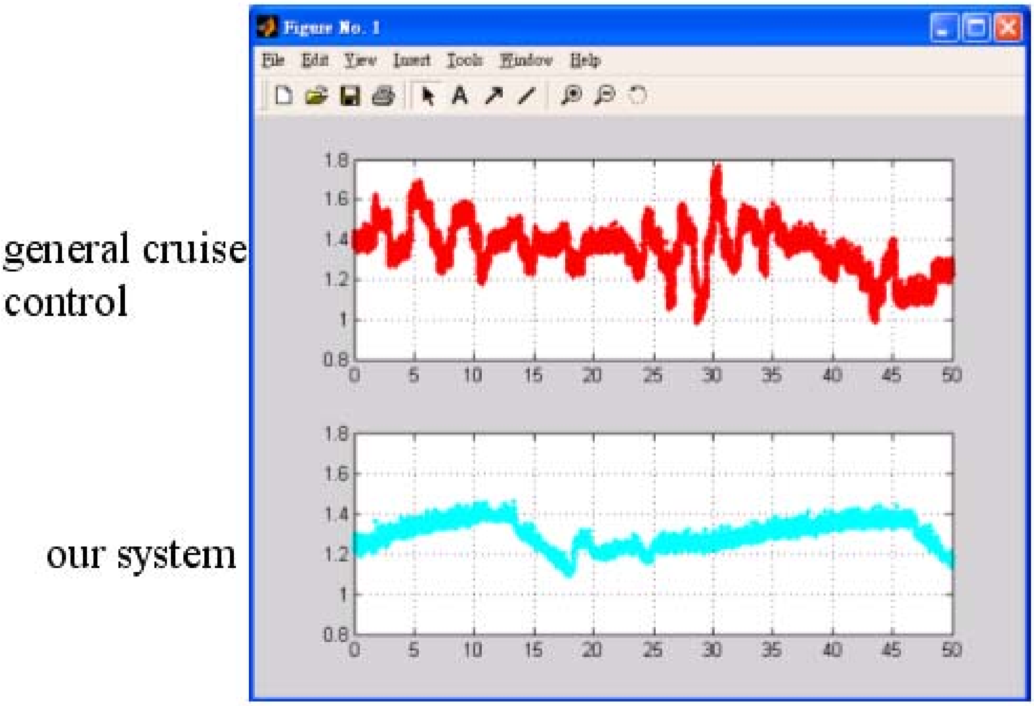


Figure 5-4 The throttle voltage of our system and general cruise control which want to hold at 60 km/hr

5.1.3 Hold at 80 km/hr

We try to hold at 80 km/hr on a smooth road. Here we also record the speed, throttle voltage of general cruise control to be the reference. The results are shown in Figure 5-5. The max speed difference of our system is about 2 km/hr. However, the difference between the desired speed (80km/hr) and real-time speed is within 1 km/hr for the most part. Therefore the result is great. The throttle voltages of our system and general cruise control held at 80 km/hr are shown in Figure 5-6. We find that these throttle voltages become smooth. It is because when the vehicle is at 80 km/hr, the inertia of it is quite big. It will be lightly affected by wind, hole, friction and slope has little effect on the car. It will not slow down quickly. Therefore it doesn't need pull throttle rapidly.

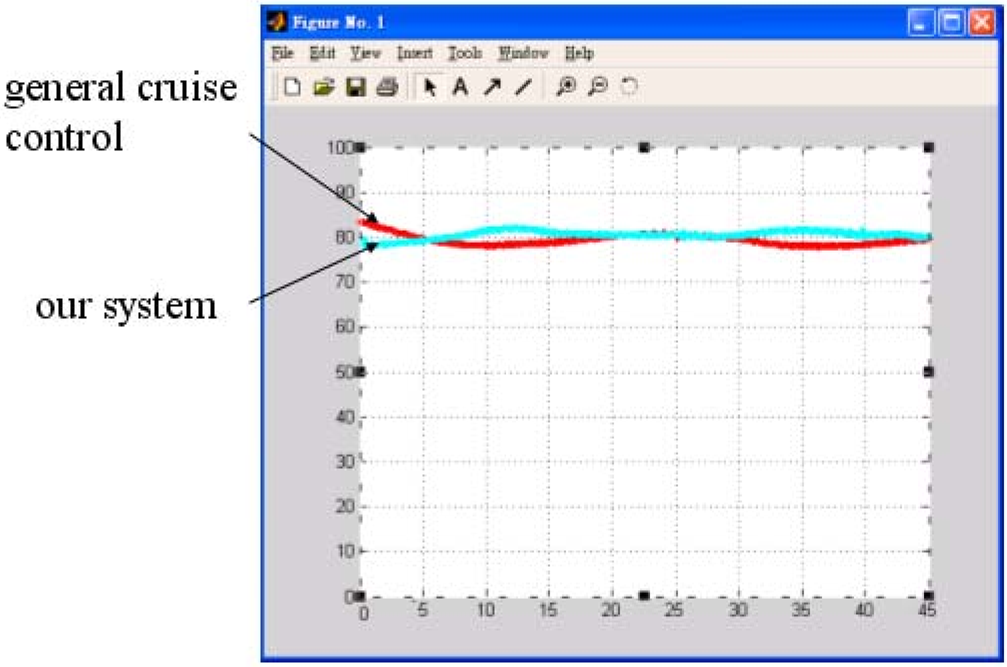


Figure 5-5 The speeds of our system and general cruise control which want to hold at 80 km/hr

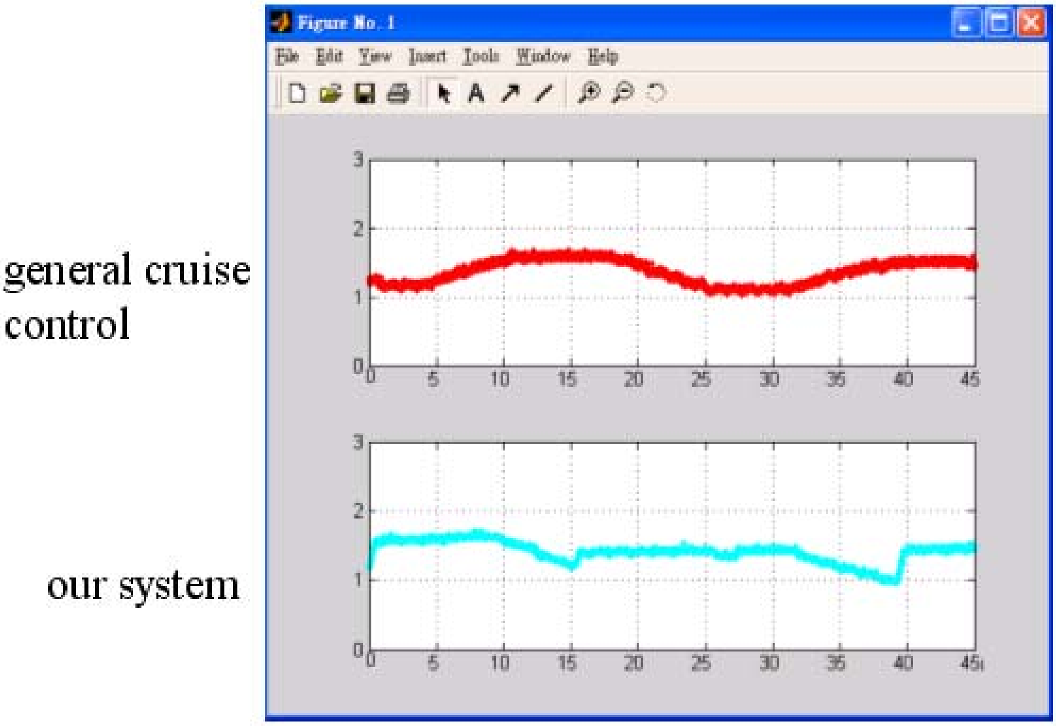


Figure 5-6 The throttle voltage of our system and general cruise control which want to hold at 80 km/hr

5.1.4 Holding at 80 km/hr on an about 20 degrees slope

We try to hold at 80 km/hr on an about 20 degrees slope. Here we also record the speed, throttle voltage of general cruise control to be the reference. The results are shown in Figure 5-7. The max speed difference of our system is about 5 km/hr. The throttle voltages of our system and general cruise control which want to hold at 80 km/hr are shown in Figure 5-8. Here, the general cruise control runs on about 20 degrees slope during 20^{th} to 50^{th} sec and its result is compared with our result during 0^{th} to 30^{th} also on about 20 degree slope. We find that the change of throttle voltage is larger than one of the general cruise control. It is because we want to build our system upon a comfortable drive style. Therefore we choose the delay of speedup, the delay of slow down and the small acceleration. Thus, the reaction period is longer than that in general cruise control and the variation of throttle voltage in a period is smaller than that of throttle voltage in general cruise control. When our vehicle runs on a slope, the speed slows down quickly but the throttle voltage increases slowly at the beginning. After the speed slows down more than 5 km/hr the throttle voltage increases largely. And then the action will cause a larger overshoot than general cruise control. It is a trade-off between comfortable drive style and quick reaction. If you want the comfortable drive style, you can choose longer period and smaller variation of throttle voltage in a period, like our system does. If you want to hold speed quickly, you can choose shorter period and larger variation of throttle voltage in a period, like general cruise control does. .

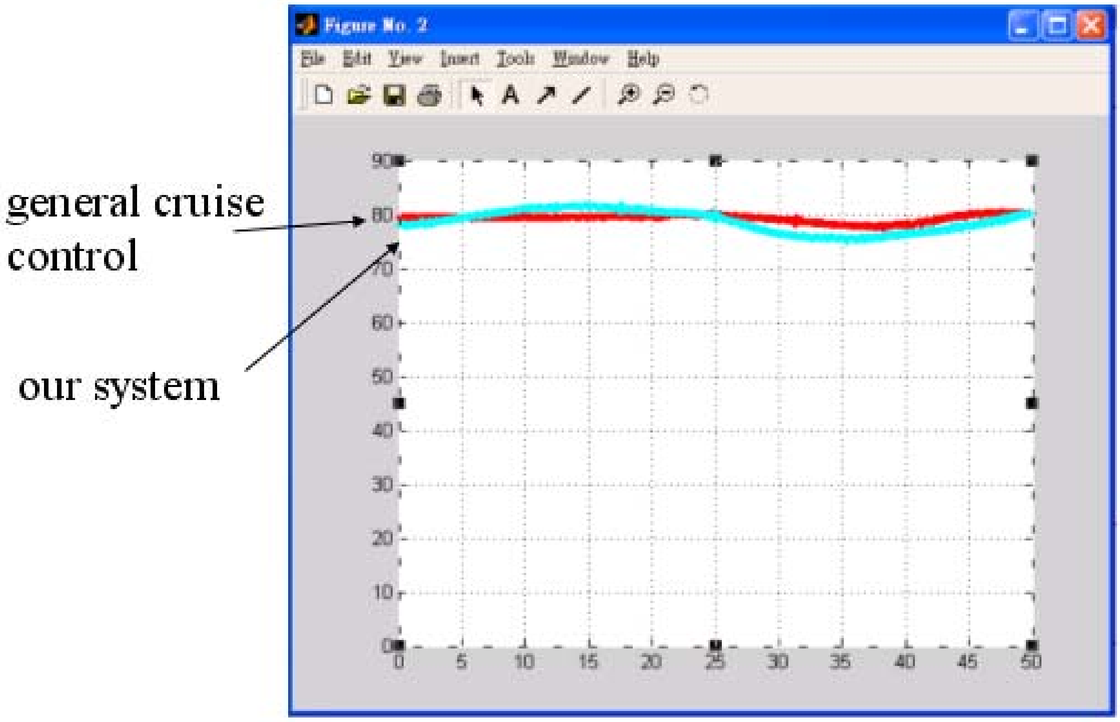


Figure 5-7 The speeds of our system and general cruise control which want to hold at 80 km/hr on an about 20 degrees slope

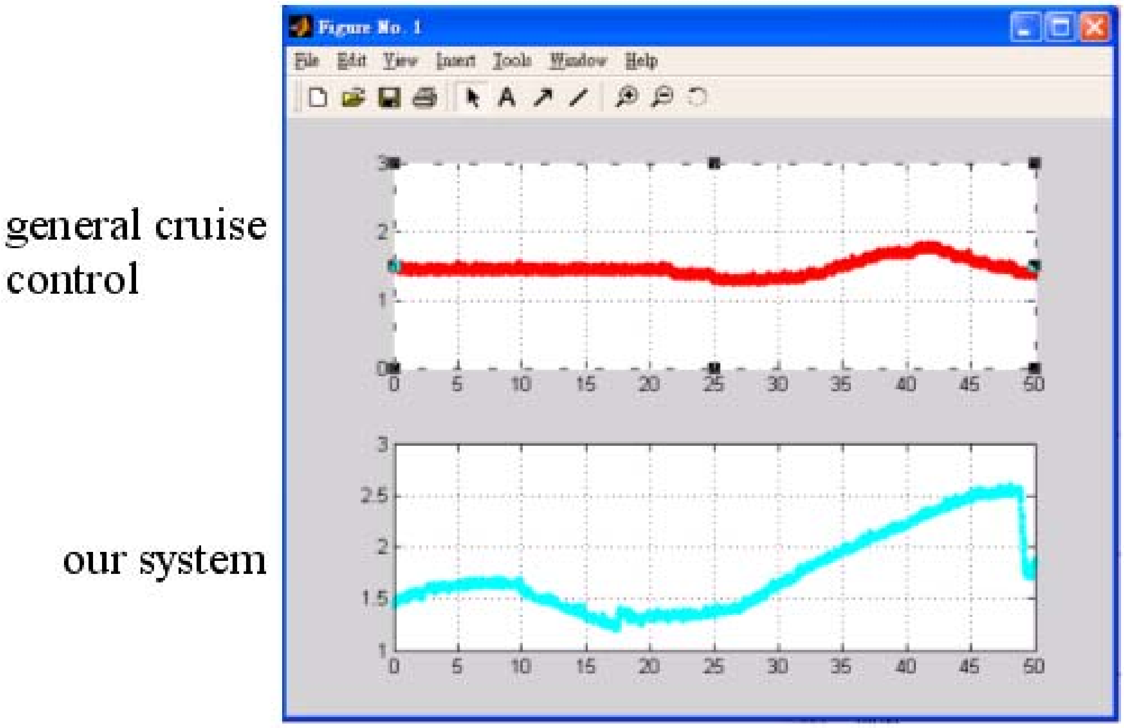


Figure 5-8 The throttle voltage of our system and general cruise control which want to hold at 80 km/hr on an about 20 degrees slope

5.2 Change speed

5.2.1 0 to 60 km/hr

Here we try to change our speed, and not hold a fixed speed. The general cruise control doesn't work when the speed of vehicle is lower than 40 km/hr. Hence it doesn't have this function so we don't have the reference to compare with. We test this function on a smooth road. First we try to speedup from 0 to 60 km/hr. In Figure 5-9 we can find out that the vehicle reaches to 60 km/hr spends for 13 sec. Maybe one may think it is too long. However under the consideration of comfortable drive, we don't like speedup fast. Hence we speed up with 2V throttle voltage as shown in Figure 5-10. If you want to speed up quickly you just need to enlarge this setup throttle voltage. We design the throttle voltage will not fall down until the difference between target speed and real-time speed is smaller than 5 km/hr. And then FLC will adjust throttle to target speed. The method can also be applied to other speeds.

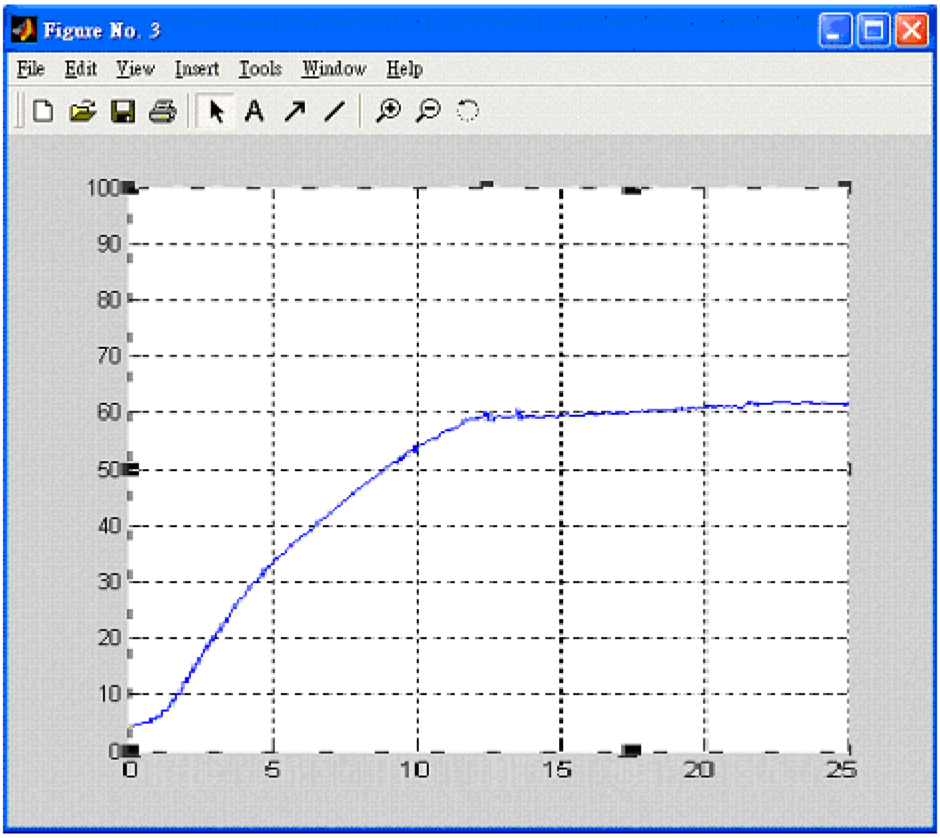


Figure 5-9 The Speed of speed-up from 0 to 60 km/hr

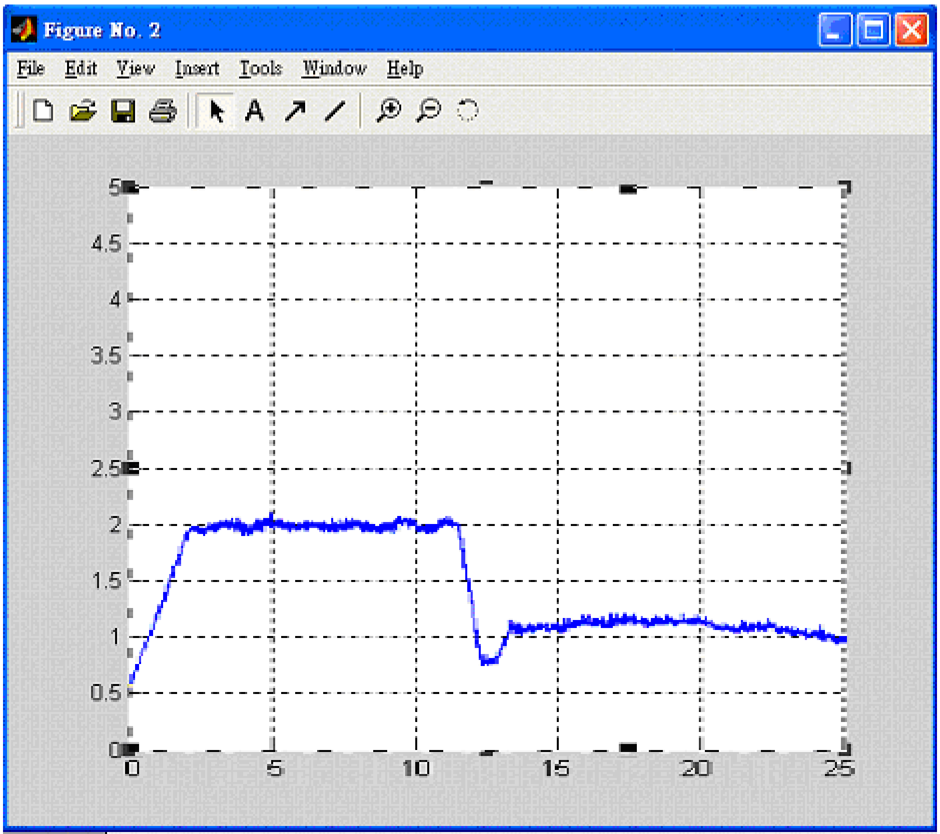


Figure 5-10 Throttle voltage of speed-up from 0 to 60 km/hr

5.2.2 60 to 80 km/hr

Here we try to change our speed from 60 to 80 km/hr. Although general cruise control works at 60 km/hr, it spends long time to speedup from 60 to 80 km/hr. It is because general cruise control only adjusts speed slightly when we press the acceleration button. Hence we don't memorize the speed which is accelerated by general cruise control to be the reference resource. We test this function at an about 10 degrees sloped road. In Figure 5-11 we can find out that the vehicle reaches to 80 km/hr spends for 35 sec. It is because the vehicle runs on the slope. If you want to speed up quickly you just need to enlarge this setup throttle voltage. In Figure 5-12, we design the throttle voltage will not fall down until the difference between target speed and real-time speed is smaller than 5 km/hr. And then FLC will adjust throttle to target speed. The method can also be applied to other cases.

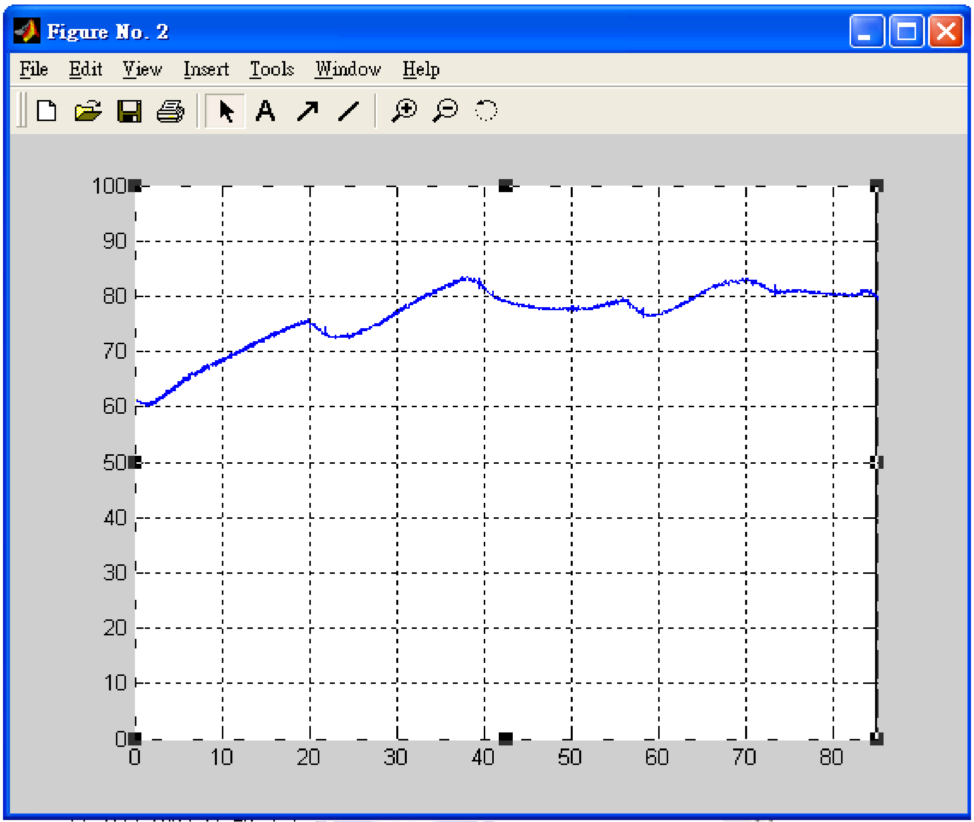


Figure 5-11 The speed of speed-up from 60 to 80 km/hr

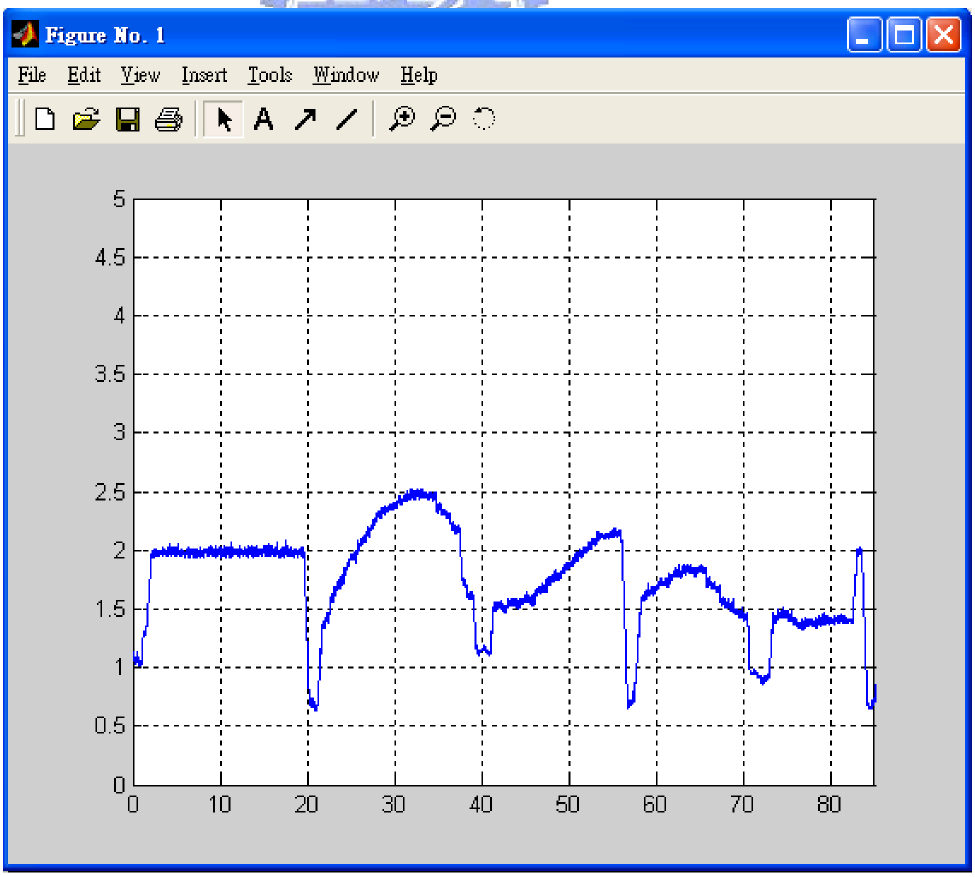


Figure 5-12 Throttle voltage of speed-up from 60 to 80 km/hr

5.2.3 80 to 60 km/hr

Here we try to change our speed from 80 to 60 km/hr. Although general cruise control works at 80 km/hr, it spends long time to slow down from 80 to 60 km/hr. It is because general cruise control only adjusts speed slightly when we press the deceleration button. Hence we don't memorize the speed which is decelerated by general cruise control to be the reference resource. We test this function at an about 5 degrees declivity. In Figure 5-13 we can find out the vehicle is down to 60 km/hr for about 15 sec and the undershoot is about 2 km/hr. In Figure 5-14, we design the throttle voltage will not rise to steady-state voltage until the speed is equal to our target speed. And it will be adjusted by FLC. The method is not only for the case. We also can slow down from 60 to 40 km/hr or other speeds via the same approach.

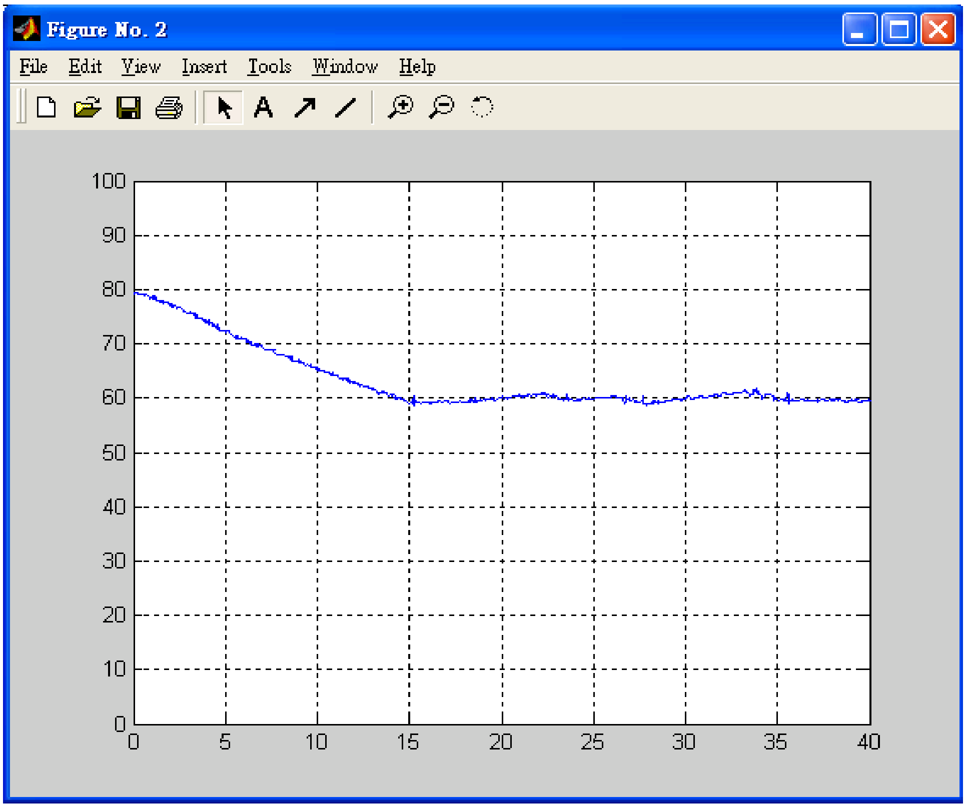


Figure 5-13 The speed of speed-up from 80 to 60 km/hr

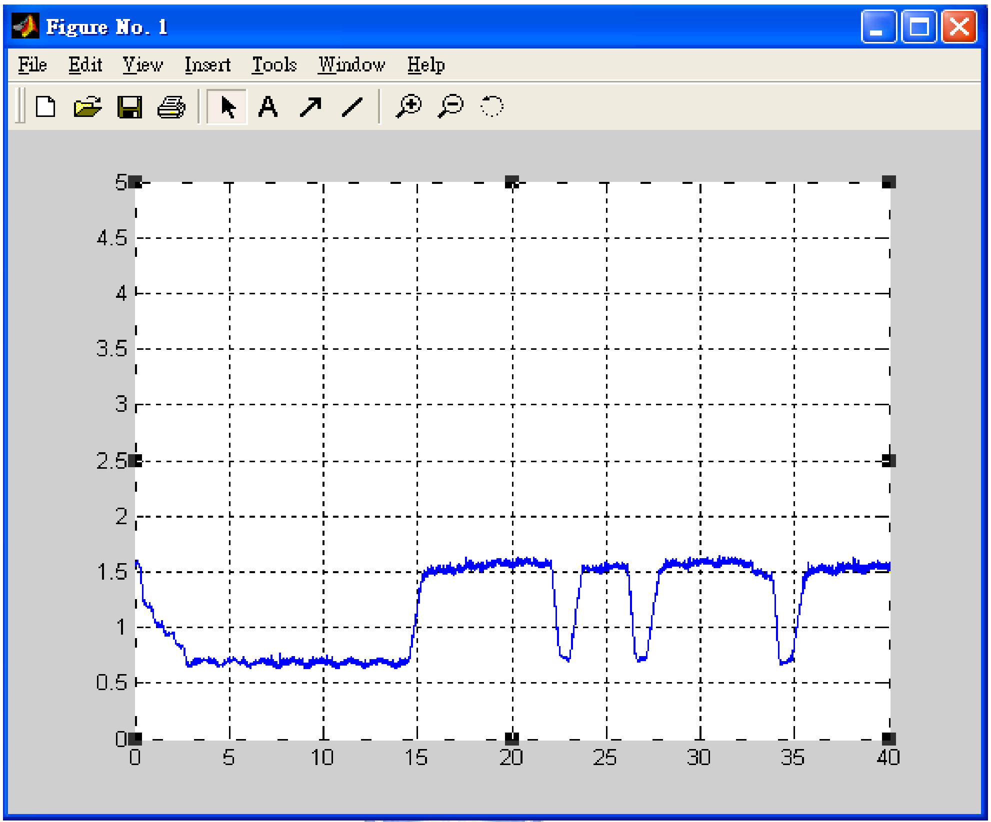


Figure 5-14 Throttle voltage of speed-up from 80 to 60 km/hr

5.3 Adaptive cruise control

5.3.1 The front vehicle with fixed speed

Here we arrange a vehicle in front of us. Figure 5-15 which is extracted from Figure 5-21 shows the real-time distance and ideal distance. The front vehicle tries to fix the speed at about 60km/hr in this section. Figure 5-16 displays the speed of our vehicle. As before, if FLC determines to slow down, the throttle voltage should be deceased by FLC. On the contrary, if FLC determines to speedup, the throttle voltage should be increased by FLC. Besides at the moment FLC changes determination from slow down to hold or speedup speed, the throttle voltage should be increased to rough steady-state voltage. The throttle voltage is shown in Figure 5-17. From Figures 5-15 to 5-17, we find the action obeys our rules and the control can adjust the speed and relative distance automatically. Most important is that it also makes the speed hold at about 60 km/hr and keeps the relative distance around ideal distance (about 30 m). It means the system is a stable system, not an unstable system.

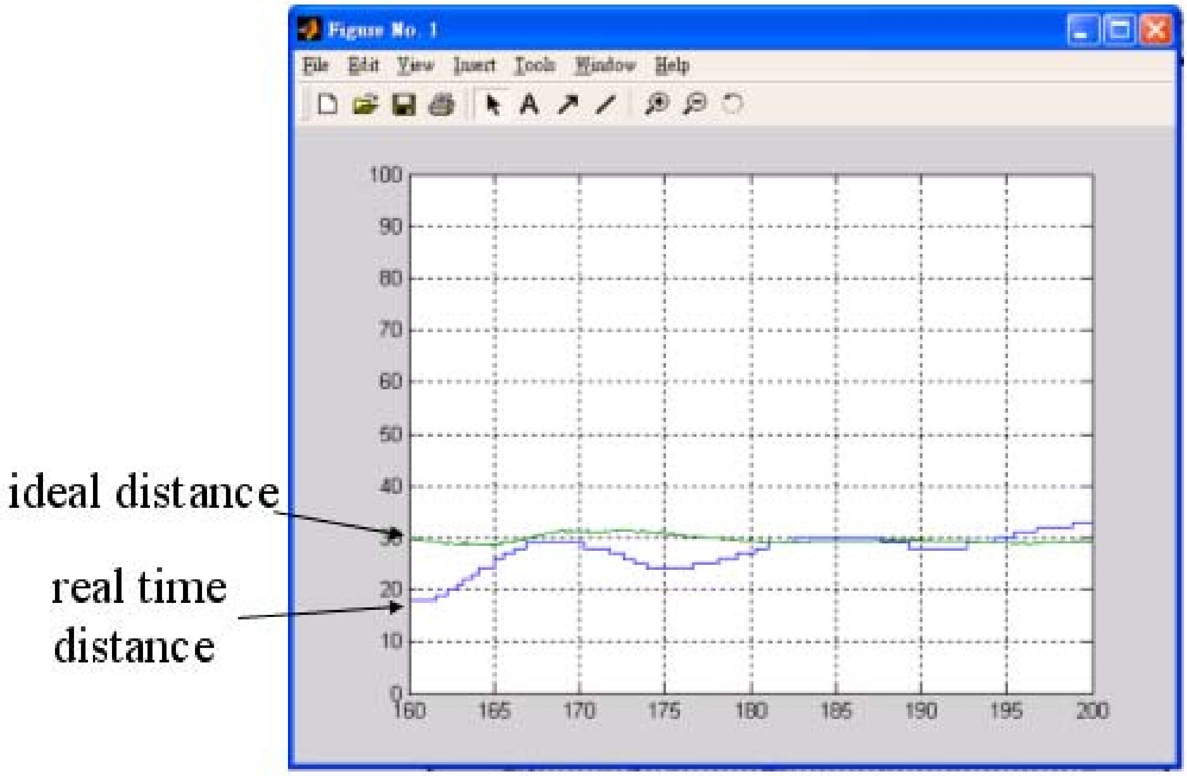


Figure 5-15 The relative distance between the front vehicle and ours

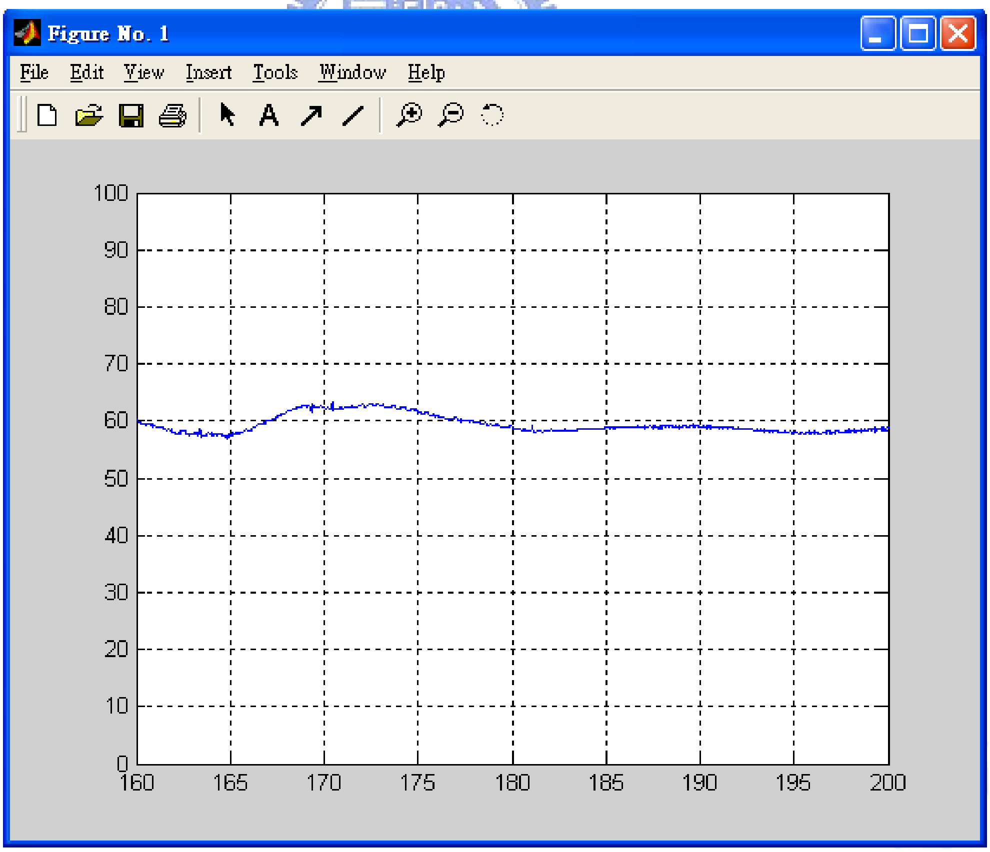


Figure 5-16 The speed of our vehicle

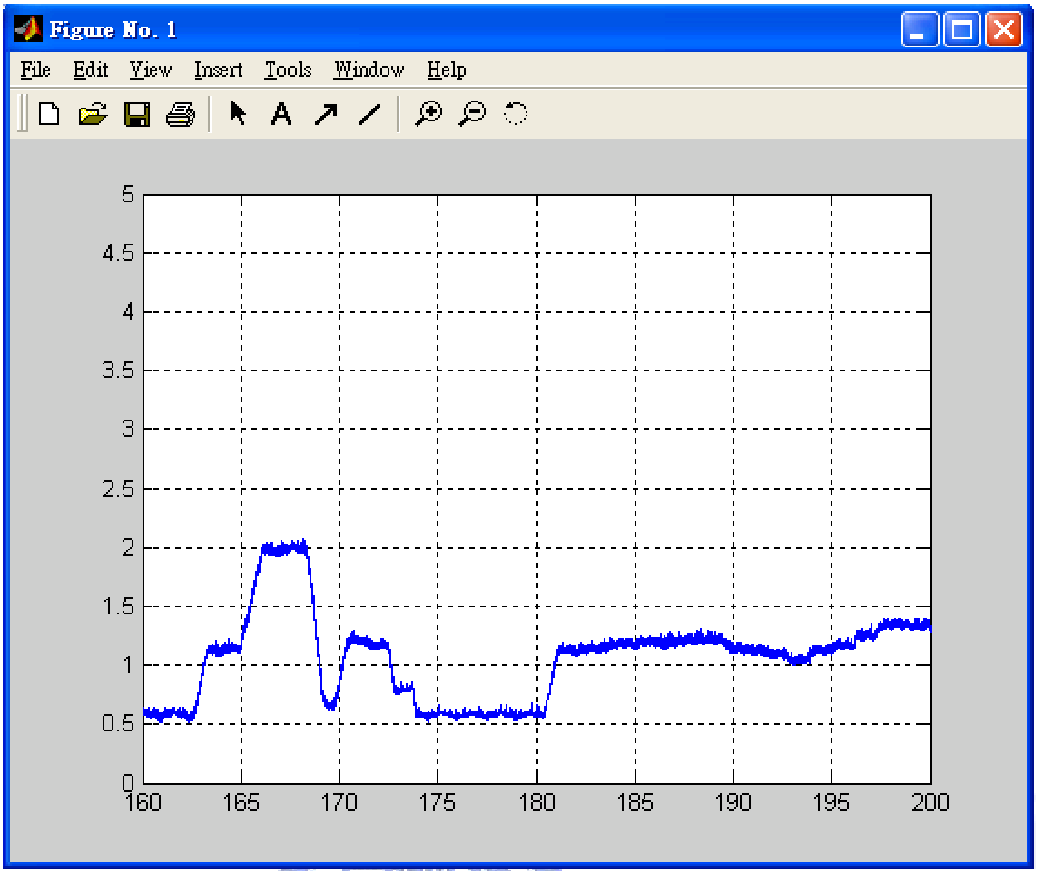


Figure 5-17 The throttle voltage of our vehicle

5.3.2 The front vehicle changes speed

We arrange a vehicle in front of us. The front vehicle drives at about 50km/hr. It is sometimes driven faster than 50 km/hr and sometimes driven slower than 50 km/hr. Figure 5-18 shows the relative distance between the front vehicle and ours. Figure 5-19 displays the speed of our vehicle. If FLC determines to slow down, the throttle voltage should be deceased by FLC. On the contrary, if FLC determines to speedup, the throttle voltage should be increased by FLC. Besides at the moment FLC changes determination from slow down to hold or speedup speed, the throttle voltage should be increased to rough steady-state voltage. The throttle voltage is shown in Figure 5-20. From Figures 5-18 to 5-20, we find the action obeys our rules and the control can adjust the speed and relative distance automatically. Most important is that it doesn’t make the relative distance too close. Therefore it is worthy to trust our system in general cases.

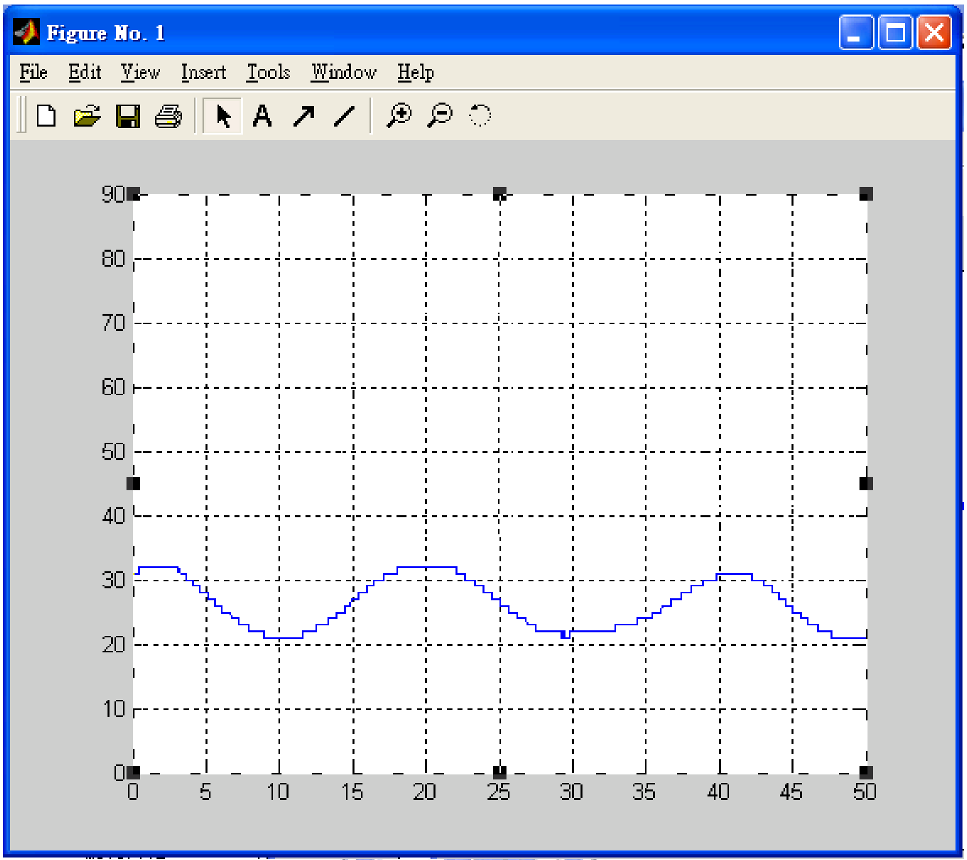


Figure 5-18 The relative distance between the front vehicle and ours

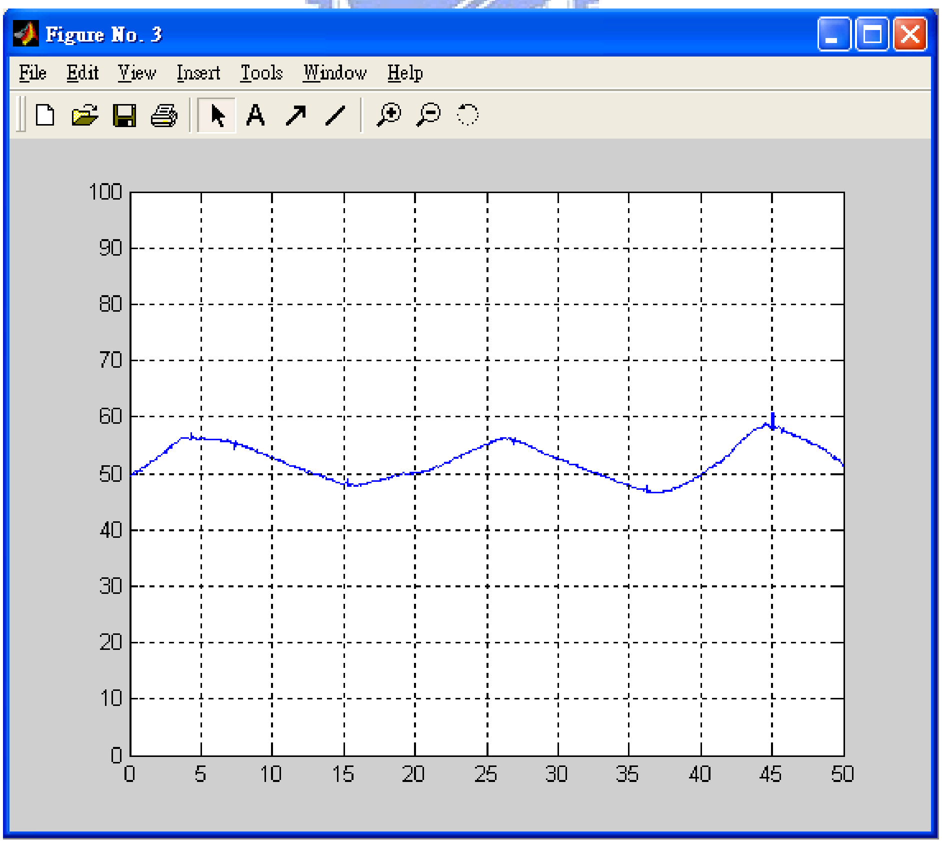


Figure 5-19 The speed of our vehicle

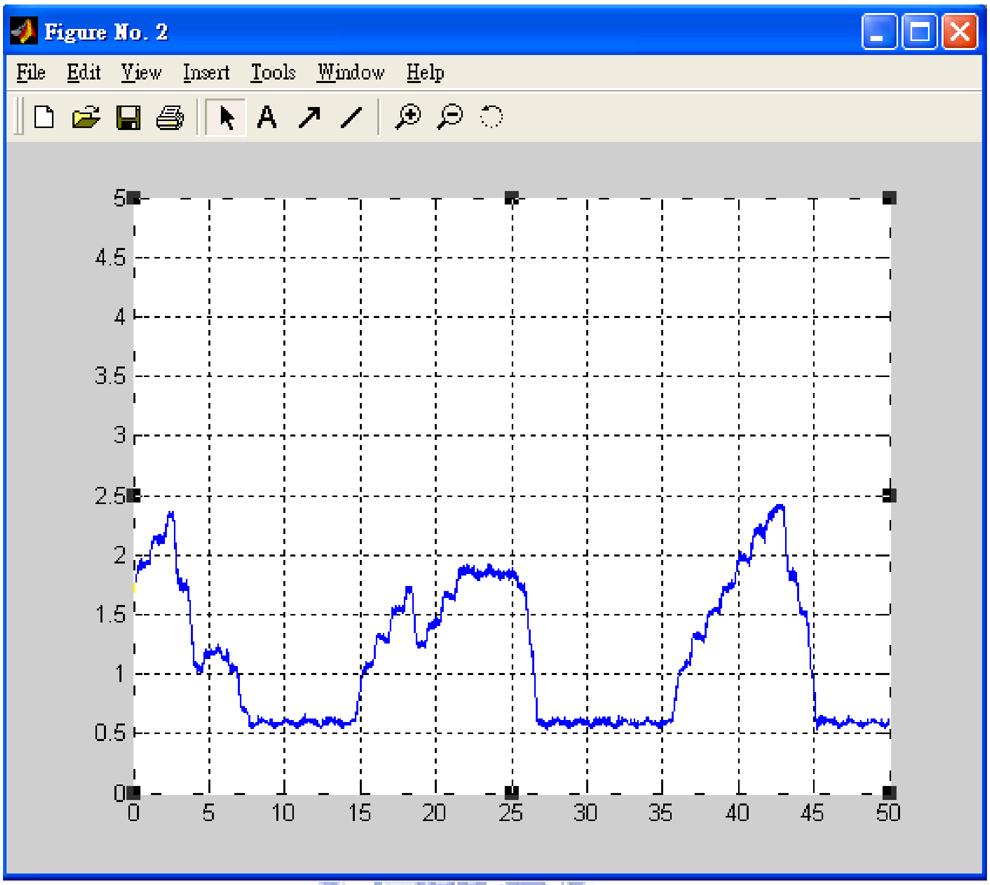


Figure 5-20 The throttle voltage of our vehicle

5.4 Switch between each mode automatically

We observe Figures 5-21 to 5-23. First we speed up from 0 to 50 km/hr and then controller tries to hold target speed, 50 km/hr. There is a vehicle whose speed is lower than ours. Therefore we approach it gradually. In order to catch up to front vehicle quickly, we change our target speed to 66 km/hr at about the 80th second. Afterwards at about the 90th second, controller starts to catch front vehicle. During the 98th to 110th second and the 142th to 152th second our vehicle is running on a sloped road, so controller lost front vehicle. At the intervals controller will hold the target speed till it catches front vehicle again. Finally, because front vehicle holds at about 60 km/hr, we can find out that controller attempts to keep at 60 km/hr and 30 meters gradually. We find out the actions during 0~90, 98~110 and 142~156 seconds obeys our CC system rules and the other seconds obey ACC system rules. Most important is that our system can change between each mode automatically.

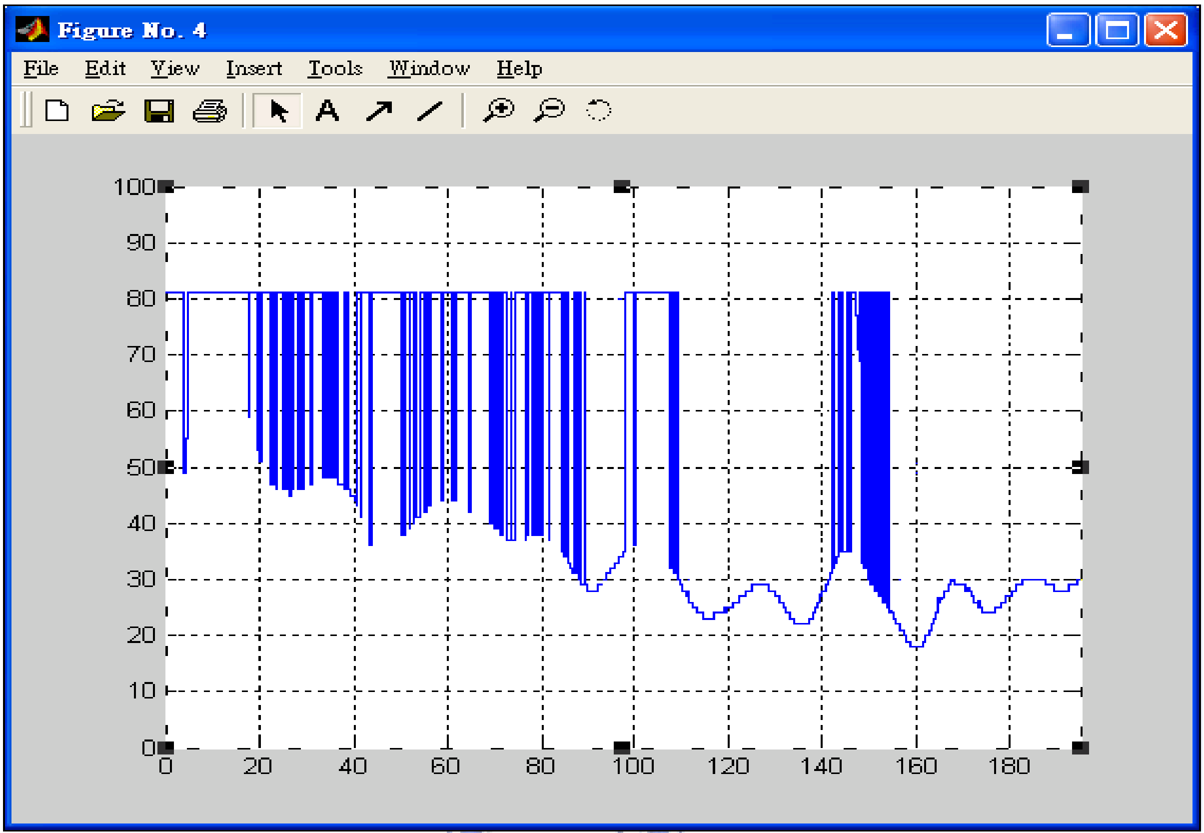


Figure 5-21 The relative distance between the front vehicle and ours

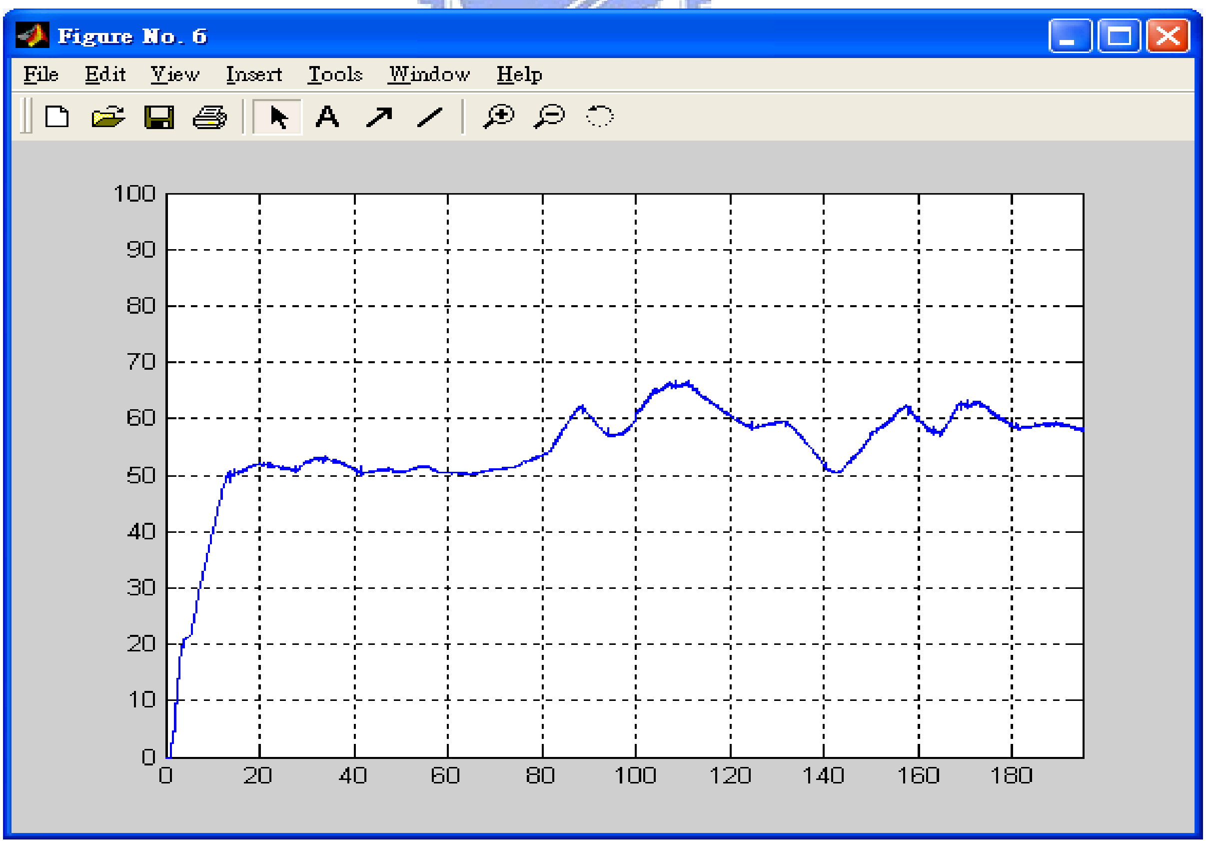


Figure 5-22 The speed of our vehicle

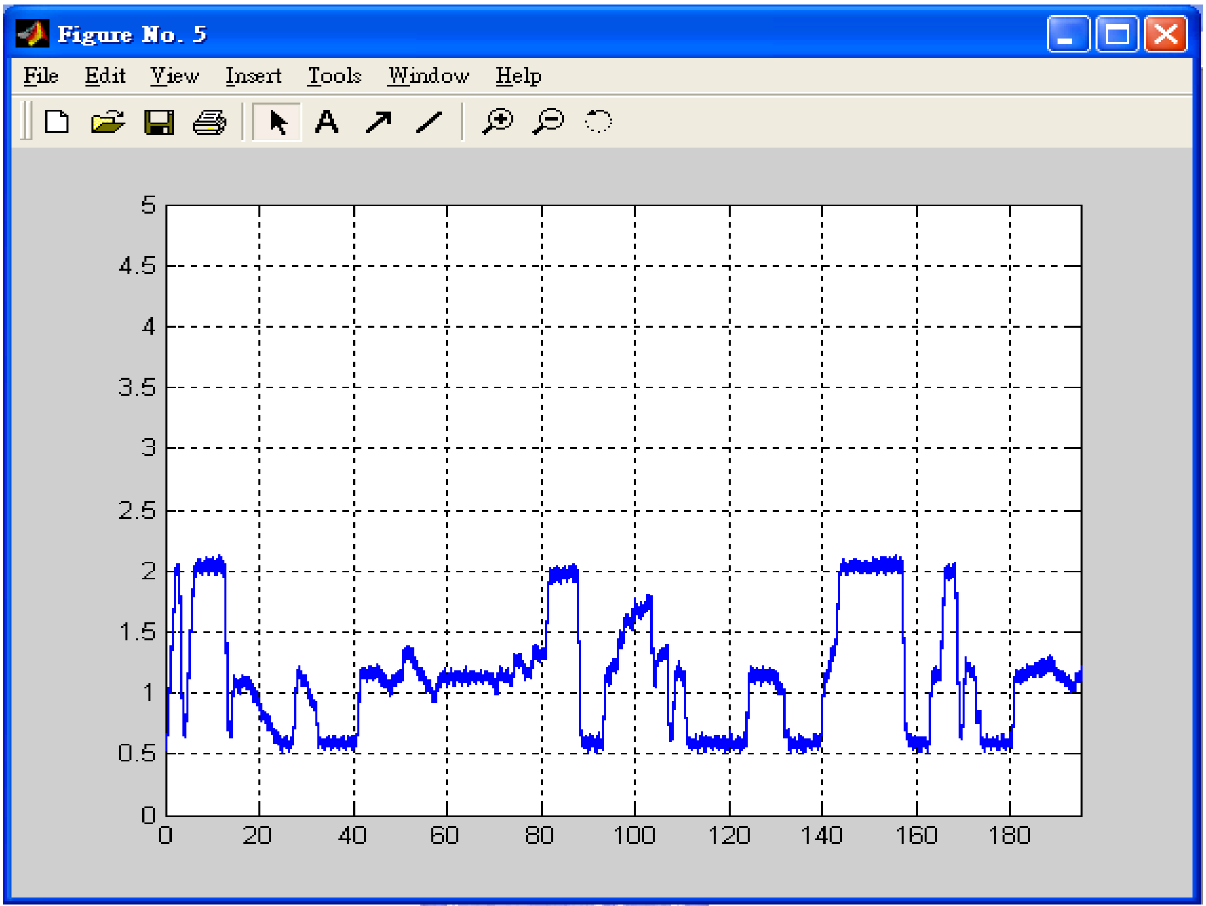


Figure 5-23 The throttle voltage of our vehicle

Figure 5-24 Experimental condition

Chapter 6 Conclusions

In this thesis, we proposed an FPGA-based Intelligent Cruise Control System that can hold at a fixed speed, change speed, operate in adaptive cruise control mode and switch between each mode automatically. Besides we consider some problems like making a turn, raising a slope and emergent situation. Although we only verify some sample functions, these functions can handle a vehicle on a monotone road, like free way. Most important is the drivers and controller are based on FPGA instead of car computer. Therefore we can infer that the Intelligent Cruise Control System based on FPGA is a solution to provide a practicable、comfortable and safety auxiliary drive system.

From these experiments, we understand that elementary measure technology is very important. If the sensors provide wrong or discontinuous data, even the complex controller hardly handles well. If the sensors provide correct and continuous data, even sample controller also can control the vehicle not bad. Besides if the sensors deliver data slowly, the reactive interval will be long. Hence controller can't have real-time data in time. Controller must increase the throttle variations to catch up to the front vehicle or controller may lose it. It may make passenger feel uncomfortable. On the contrary, if the sensors deliver data quickly, the reactive interval will be short. Controller can grad real-time data and make a determination in time. It is small throttle variation in every interval. It makes passenger feel comfortable. Although sensor is quite unapparent, it is the key component of Intelligent Cruise Control System.

When we write programs, we not only consider if the functions are correct but also think the used CLB can be decreased or not. At the same time the operating period is acceptable. Because the experiment platform is a vehicle, not inside a laboratory, we have to consider noise from shaking and wrong data from sensors.

In the future, we can increase a function to switch between normal parameters and special parameters. We can adjust the scan angles like human according to speed. If the speed is faster, the scan angles are narrower. If the speed is slower, the scan angles are wider. We can try to control brake to arrive other target "start and go". We also can combine neural network to adjust parameters to satisfy each type of vehicles. Besides we can setup a rule to fine adjust parameters according the voltage of battery, years of engine, years of DC-motor, etc. Therefore, there are still many topics that are deserved to be researched.

Appendix

Figure A-1 Laser scanner

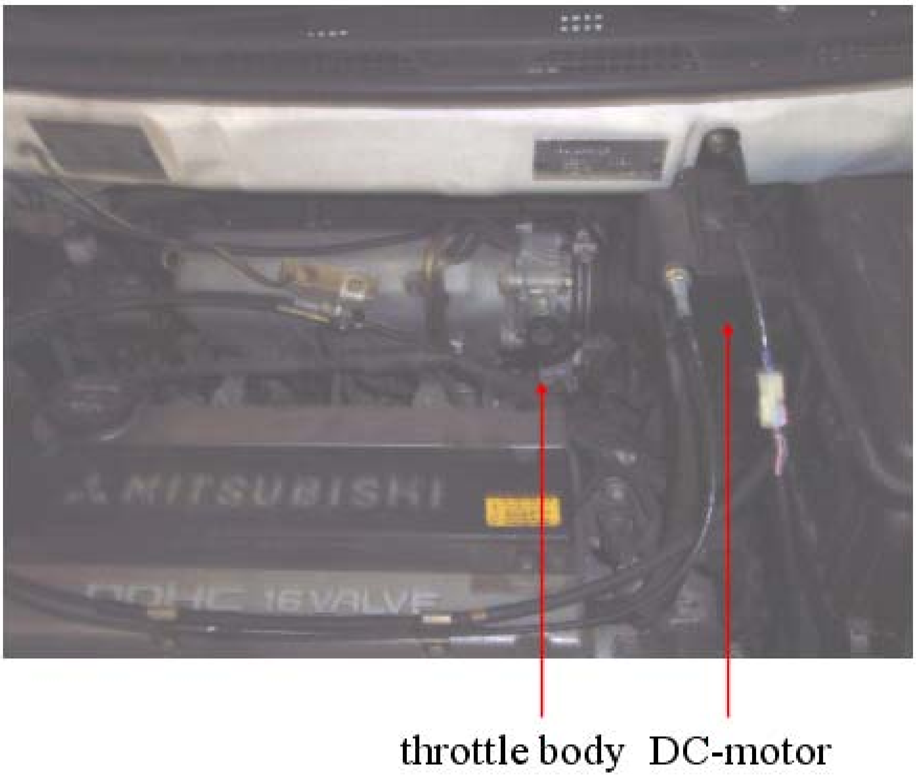


Figure A-2 Throttle body and DC-motor

Figure A-3 Steering wheel angle sensor

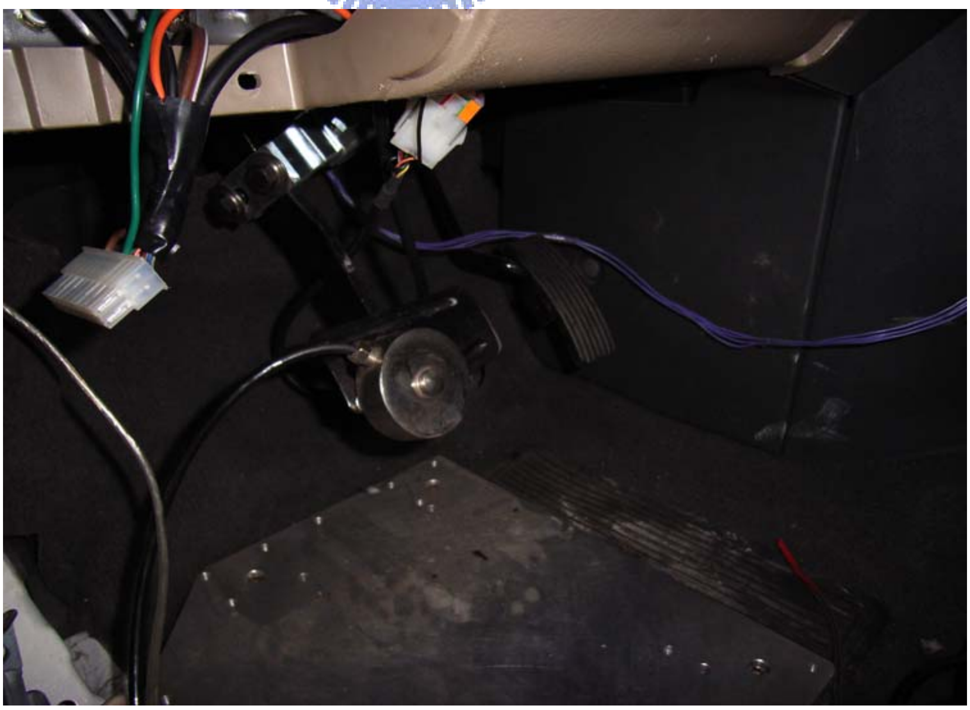


Figure A-4 Load cell

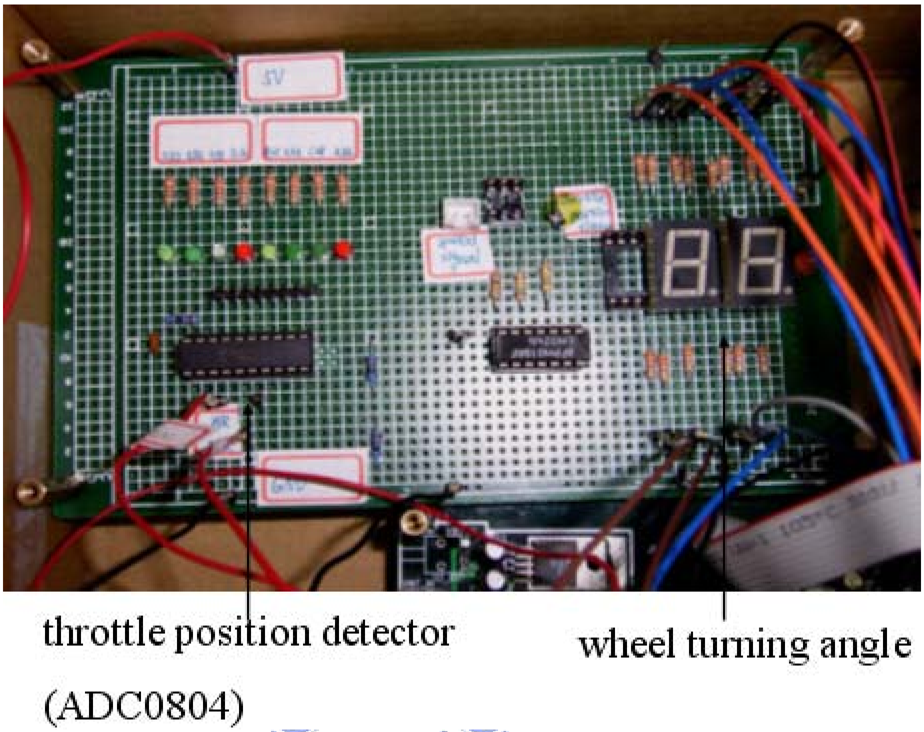


Figure A-5 Throttle position detector and the wheel turning angle display circuit

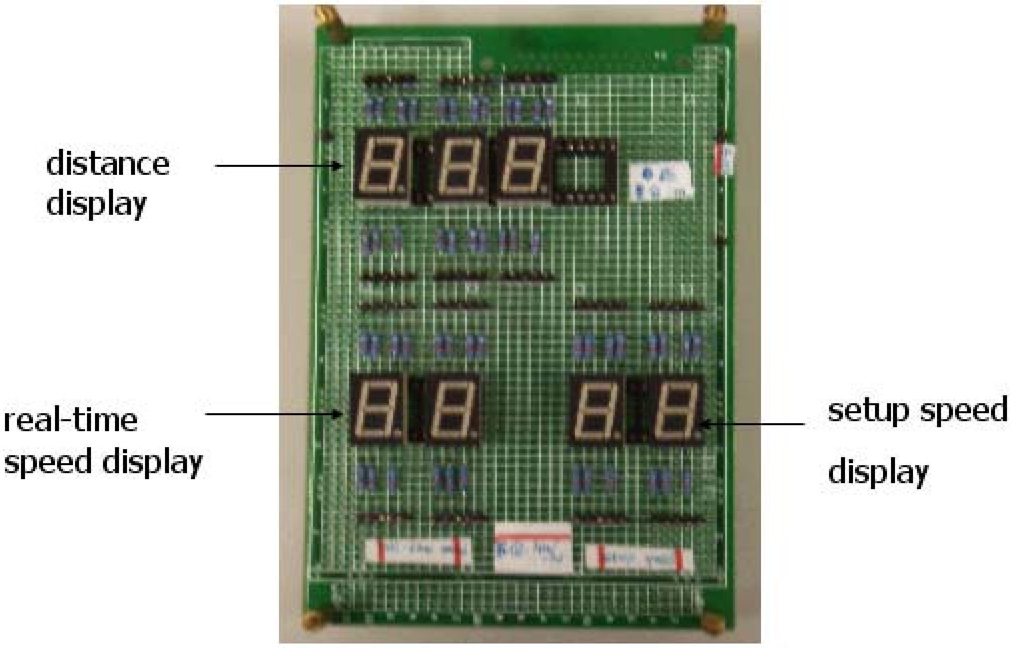


Figure A-6 Real-time, setup speed and distance display circuit

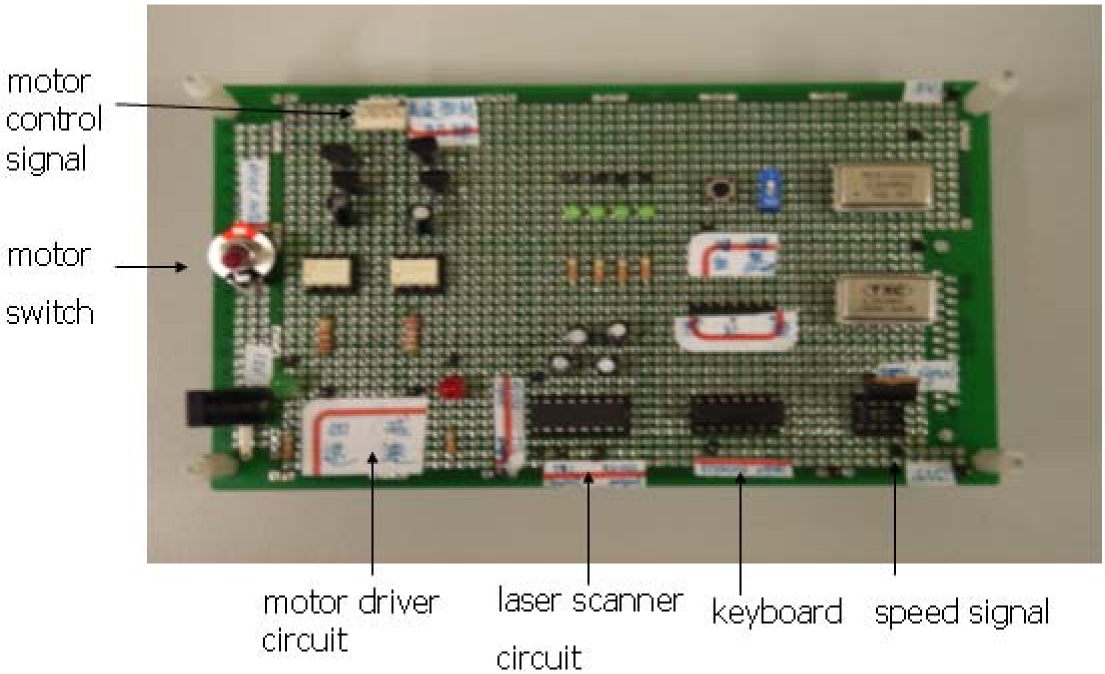


Figure A-7 Motor driver, laser scanner, keyboard and speed signal circuit

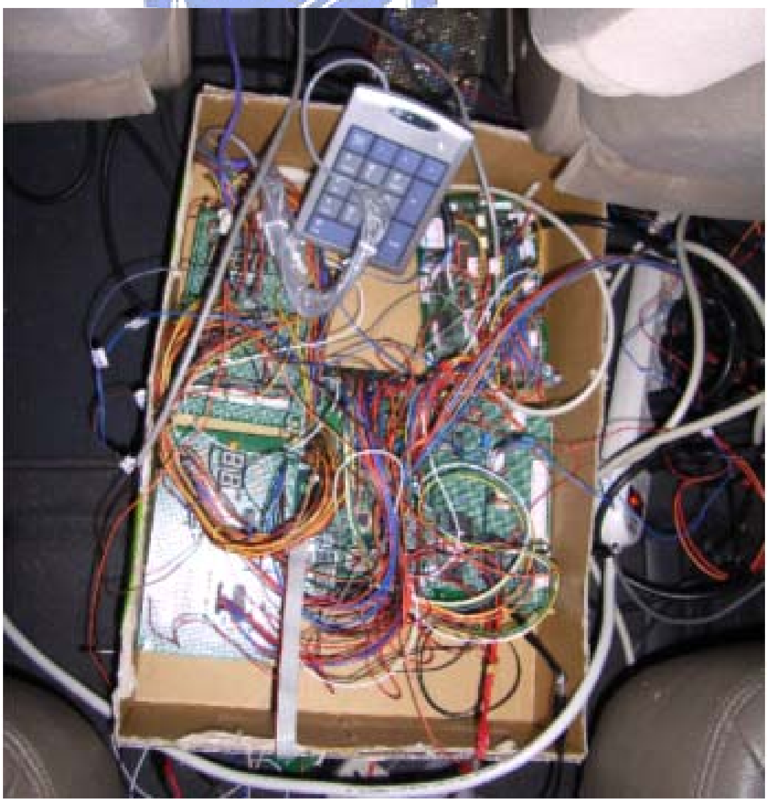


Figure A-8 All circuits

References

[1] Intelligent Transportation Society of Taiwan , http://www.its-taiwan.org.tw.

[2] W. D. Jones, “Keeping Cars from Crashing” IEEE Spectrum, Vol. 38, Issue 9,pp.40-45,Sept.2001.

[3] M. A. Goodrich and E. R. Boer, “Designing Human-Centered Automation: Tradeoffs in Collision Avoidance System Design” IEEE Transactions on Intelligent Transportation Systems, Vol.1, No. 1, pp. 40-54.March 2000.

[4] J. Mar, “An ANFIS Controller for the Car-Following Collision prevention System” ,IEEE Transaction on Vehicular Technology, Vol.39, No. 1,pp.82-85,Jan.2002.

[5] W. D. Jones, “Building Safer Cars 2002 IEEE”, IEEE Spectrum, Vol.39, No.1, pp.82-85, Jan. 2002.

[6]C. C. Chien and P. Ioannou , “Autonomous Intelligent Cruise Control”, IEEE Transactions on Vehicular Technology, Vo1. 42, No.4, November,1993.

[7]A. Kanaris, P. Ioannou and F.S. Ho, “Spacing and Capacity Evaluations for Different AHS Concepts”, Proceedings of the American Control Conference, Vol.3, pp. 2036-2040, 1997.

[8] C. C. Chien and P. Ioannou , “Autonomous Vehicle Following 1992”, Proc. 1992 American Control Conference, Chicago, pp. 1748-1752.

[9]D. Swaroop and K. R. Rajagopal, “A Review of Constant Time Headway Policy for Automatic Vehicle Following”, Proceeding of the IEEE Intelligent Transportation Systems Conference, pp. 65-69, August 2001.

[10] D. Yanakiev and I. Kanellakopoulos, “Nonlinear Spacing Policies for Automated Heavy-Duty Vehicles”, IEEE Transactions on Vehicular Technology, Vol. 47, No.4, pp. 1365-. 1377, November 1998.

[11] K. Santhanakrishnan and R. Rajamani, “On Spacing Policy for Highway Vehicle Automation”, Proceeding of the American Control Conference, Vol.3,pp.1509-1513,2000.

[12] M. Hoedemaeker and K. A. Brookhuis, “Behavioural adaptation to driving with an adaptive cruise control (ACC)”, Transportation Research, Vol. 1F, No.2, Dec. ,pp. 95-106.

[13] R. Holve and P. Protzel, “Adaptive Fuzzy Control for Driver Assistance in Car-Following” , Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing, pp. 1149-1153, Aug.1995.

[14] P. Ioannou , F. Ahmed-Zaid and D.H. Wuh, “A time Headway Autonomous Intelligent Cruise Controller: Design and Simulations”, Research Report UCB-ITS-PWP-94-07, University of Southern California, April 1994.

[15] Y. Zhang, E.B. Kosmatopoulos and P. Ioannou, “Autonomous Intelligent Cruise Control Using Front and Back Information for Tight Vehicle Following Maneuvers”, IEEE Transactions on Vehicular Technology, Vol.48, No.1, pp.319-328, 1999.

[16] P. Ioannou and Z. Xu, “Throttle and Brake Control Systems for Automatic Vehicle Following” ,IVHS Journal, Vol.1,pp. 345-377 ,1994.

[17] M. Persson, F. Botling, E. Hesslow and R. Johnsson, “Stop and go controller for adaptive cruise control”, Proceedings of the 1999 IEEE International Conference on Control Applications, Vol.2 , pp. 1692-1697,1999.
[18] P. Venhovens, K. Naab and B. Adiprasito, “Stop and go cruise control” ,Proc. of the 2000 FISITA World Automotive Congress, Seoul, Korea, June.
[19] J.I. Suarez , B.M. Vinagre and Y.Q. Chen, “Spatial path tracking of an autonomous industrial vehicle using fractional order controller” in Proc. ICAR 11th Int. Conf. advanced Robotics, Coimbra, Portugal ,pp.405-410, June 30-July, 2003.
[20] E.J. Rossetter and J.C. Gerdes, “Performance gurantees for hazard based lateral vehicle control” , in Proc. IMECE Conf., 2002.
[21] S. Bentalva et al., “Fuzzy path tracking control of a vhicle”, in IEEE Int. Conf. Intelligent Vehicles, pp.195-200, 1998.
[22] S. Sheikholeslam and C.A.Desoer, “Design of Decentralized Adaptive Controllers for a Class of Interconnected Nonlinear Dynamical Systems: Part I” ,Department of Electrical Engineering and Computer Sciences, Institute of Transportation Studies University of California , Berkeley ,CA, PATH Technical Memorandum 92-1, Feb.3 ,1992.
[23] R. Hovle, P.Protzel and K. Naab, “Generating fuzzy rules for the acceleration control of an adaptive cruise control system” in Proc. Fuzzy Information Processing Soc. NAFIPS. Biennal Conf. North American, pp.451-455,1996.
[24] ALVINN: An Autonomous Land Vehicle in a Neural Network , Advances in Neural Information Processing Systems, 1989. http://portal.acm.org/citation.cfm?id=89891
[25]M. Sugeno, “An introductory survey to Fuzzy control”, Info. Sci., Vol.36, 1985.
[26] P. Protzel, R. Holve, J. Bernasch and K. Naab, “Fuzzy Distance Control for Intelligent Vehicle Guidance”. Proc. of the 12th Annual Meeting of the North American Fuzzy Information Processing Society, Allentown PA, pp.87-91Aug. 22-25, 1993.
[27] N. C. Griswold, N. D. Kehtarnavaz and K.M. Miller, “A Transportable Neural Network Controller for Autonomous Vehicle Following”, Proceedings of the IEEE Intelligent Vehicle Symposium, pp. 195-200, Oct. 1994.
[28] Y. J. Ryoo and Y. C. Lim, “ Neural-fuzzy control system for vision-based autonomous vehicle” , Proceedings of the 1999 IEEE International Conference on Fuzzy Systems, Vol.3 pp. 1643-1648, 1999.
[29] H. Nomura, I. Hayashi and N. Wakami, “A learning Method of Fuzzy Inference Rules by Descent Method”, IEEE International Conference on Fuzzy Systems, pp. 203-210 ,1992.
[30] J.L. Arroyabe, G. Aranguren, J.L.Martin, L.A.L. Nozal,;; “Autonomous vehicle guidance with fuzzy algorithm ”Industrial Electronics Society, IECON 2000. 26th Annual Conference of the IEEE Vol. 3, PP. 1503 – 1508. 22-28, Oct. 2000.
[31]J. E. Naranjo, C. Gonzalez, J. Reviejo, R. Garcia, and T.de pedro, “Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping”, IEEE Transactions on Intelligent Transportation systems, Vol. 4, No.3 ,pp. 132-142, Sep. 2003.

[32]S.Singh, K.S.Rattan, “Implementation of a fuzzy logic controller on an FPGA using VHDL”North American Fuzzy Information Processing Society, NAFIPS, 22nd International Conference , PP:110 - 115 ,July 24-26, 2003.

[33] W. Chi ,C. K. Tse and Y.S. Lee “Development of a Fuzzy Logic Controller for DC/DC Converters :Design Computer Simulation and Experimental Evaluation” IEEE Transactions on power electronics Vol. 11, No.1, January 1996.

[34]R. Holve, P. Protzel , K. Naab, “Generating fuzzy rules for the acceleration control of an adaptive cruise control system” Fuzzy Information Processing Society, NAFIPS. Biennial Conference of the North American PP:451- 455 19-22 June 1996.

[35] M. McKenna and B. M. Wilamowski, “Implementing a Fuzzy System on a Field Programmable Gate Array”, Neural Networks, Proceedings. IJCNN '01. International Joint Conference on Vol.1, PP.189 - 194 15-19 July 2001.

[36]中華汽車 SAVRIN 工作手冊。

[37] Quick Manual for LMS communication setup.
http://team.caltech.edu/members/SICK/LMS_Quick_Manual_V1_1.pdf

[38] BOSCH world wed site, http://www.bosch.com.

[39] Electric operated cruise control AP500 installation manual.
http://www.autoalarm.ee/index.php?cruisecontrol

[40]TOSHIBA photocoupler GaAlAs IRED & PHOTO-IC TLP250.
http://www.toshiba.com/taec/components/Datasheet

[41] Precision Pancake Load Cells, http://www.sensotec.com.

[42]孫宗瀛、楊英魁，Fuzzy 控制 :理論、實作與應用，全華，1999。

[43] Altera world wed site, http://www.altera.com.

[44]W. H. Crouse, D. L. Anglin 著，劉崇富 譯，汽車學(二)----汽車驅動系統與底盤，高立，2001。

[45]林傳生，使用 VHDL 電路設計語言之數位電路設計，儒林，1998。