# PEOPLE MOVER PROJECT

## (CONTROL SYSTEM PART)


A REPORT SUBMITTED TO DR. A. K. M. ABDUL MALEK AZAD of ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTMENT of BRAC UNIVERSITY IN FULFILLMENT of THE REQUIREMENTS for THESIS WORK

**Group members**

- **Shayla Azad Bhuyan, ID: 09221152**
- **Choudhury Tanzia Siddqui, ID: 09221154**
- **Sazia Afrin Chowdhury, ID: 09221155**
- **MD Kazi Zaferullah, ID: 09221186**

## Declaration

We hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researchers are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

28/4/11

Signature of the Supervisor

Signature of the Author

① Zafor

② Tanjia

③ Ezaaz

④ Layla 28.04.11

## Abstract

People-Mover is an electrical self-motivated medium. The term is generally used only to describe systems serving relatively small areas such as airports, downtown districts or theme parks, but is sometimes applied to considerably more complex automated systems. Normally, the train runs autonomously form terminals to terminals. Its speed and movement can be controlled wirelessly.

Practically, people movers typically consist of driverless trains with up to about four cars each capable of carrying 20 to 100 passengers who are mostly standing. They have been successfully used for surface transportation in airports for over thirty years. A new category of automated people mover called personal rapid transit (PRT) is being implemented at London's Heathrow International Airport.

In this project we designed and implemented this independent rail system which is planned for people transportation from system specifications down to fully working system as well as hardware and software.

Main goal of this project is to convert the theoretical knowledge into practical system. On that aspect this project is perfect, because here all features of Electrical Engineering (control system, circuit designing and implementation, wireless communication, micro electronics etc.) as well as some mechanical engineering features are being used.

# Content

## List of Figures

## List of Table

# Chapter 1

## Introduction

The PeopleMover project which we have done is one of the first examples of project-based education completed by the Katholieke Universiteit Leuven, Belgium. As it is mentioned earlier, it features all the branches of electrical engineering with some mechanical engineering so it is very helpful for students to understand, use and compare their educational knowledge with practical experience.

As we all know People-mover is a different rail way system from the available system that we have in our country. It is an independent rail way that can be controlled distantly to take passengers from a terminal to another terminal. There are three teams having four students each. The diverse tasks of speech recognition and acoustic noise reduction for the remote supervisor, wireless transmission by radio link, sensors (infrared, magnetic and solar cells), power electronics, and the speed control system. So knowledge about motor, power electronics circuits, power supply, design of a control system for speed control and collision protection, an automatic light control system, digital signal processing, wireless communications, modulation techniques, beacons and antennas, sensors, design and assembly of printed circuit boards is required for accomplishing this project.

To make this project easier, finer and more educative this whole project is divided into three parts.

Group's are-

1. Control system group
2. Electronics group

3. Communication group

Now as it is a multi-disciplinary design project, so it has both advantages and disadvantages over the normal conventional process that has theoretical courses accompanied by exercise sessions. During the implementation of a real system, lots of practical problem arise which are usually masked in a simulation-oriented project. In addition with students learn to design something independently in any way rather than following a fixed way. This independent tendency force repetition in some case of doing the whole thing from first step of the design and at the end some new and different things come out under a same logo. Moreover it needs huge amount of teamwork within the 3 teams. And lastly the focus on synthesizing instead of only analyzing early on leads to a better understanding in later advanced courses.

On the other hand there are some disadvantages as well for this type of work. Since students have different opinion, multiple solutions may arise, which increase the supervising work and need to asses all the techniques. It will take much time in some case to integrate the different groups' works.

In this paper, the design of the "speed control system" is written in detail.

In the second chapter this project would be described. The whole system would be discussed briefly on that chapter.

The third chapter is about PID controller. How PID controller works, what are the elements of this controller, how it can be tuned, what is the optimum output etc.

Next on fourth chapter pulse width modulation will be discussed. What are the applications, which way we are going to use it in our work etc.

After that on the fifth chapter the speed control system will be given. Main principle of controlling speed is to vary voltage.

Then the sixth chapter will have the main material of this project that is microcontroller, where all the information about microcontroller and how this is used in this project will be given.

In the seventh chapter all the result that has been gathered so far will be shown and discussed.

Afterward on eighth chapter the conclusion will be on the ninth chapter.

And finally what is required in future to make this work smoother is written

At the end of paper will finish by some acknowledgement, references and appendix.

# Chapter 2
# Project Overview

## 2.1 Background and history

The PeopleMover project is one of the first examples of project-based education at the Katholieke Universiteit Leuven (K.U.Leuven), Belgium. Senior-level bachelor students in electrical engineering perform this project to develop their practical skills in subjects which are traditionally taught in purely theoretical courses. The project resembles the electronic and control aspects of a people mover, a self-directed but distantly supervised train that takes passengers between airport terminals. Groups consisting of four students tackle the diverse tasks of speech recognition and acoustic noise reduction for the remote supervisor, wireless transmission by radio link, finite state machine design for main control, sensor integration including infrared, magnetic, and solar cells, power electronics fabrication, and speed control system implementation. During initial brainstorming sessions, the students specify the layout of the entire system, the tasks facing each team, and the interfacing between different teams. Starting from these specifications, each team designs and simulates an electronic circuit, which is implemented on a printed circuit board (PCB). Finally, all PCBs are integrated into a small-scale model train, shown in Figure 1.Our multidisciplinary design project complements traditional education, consisting of theoretical courses and exercise sessions. In such a "real" application, the students encounter practical problems that do not arise in a simulation-only environment. Moreover, students must address multiple design aspects, which require iteration in the design cycle, sometimes even backtracking to the first steps of the design.

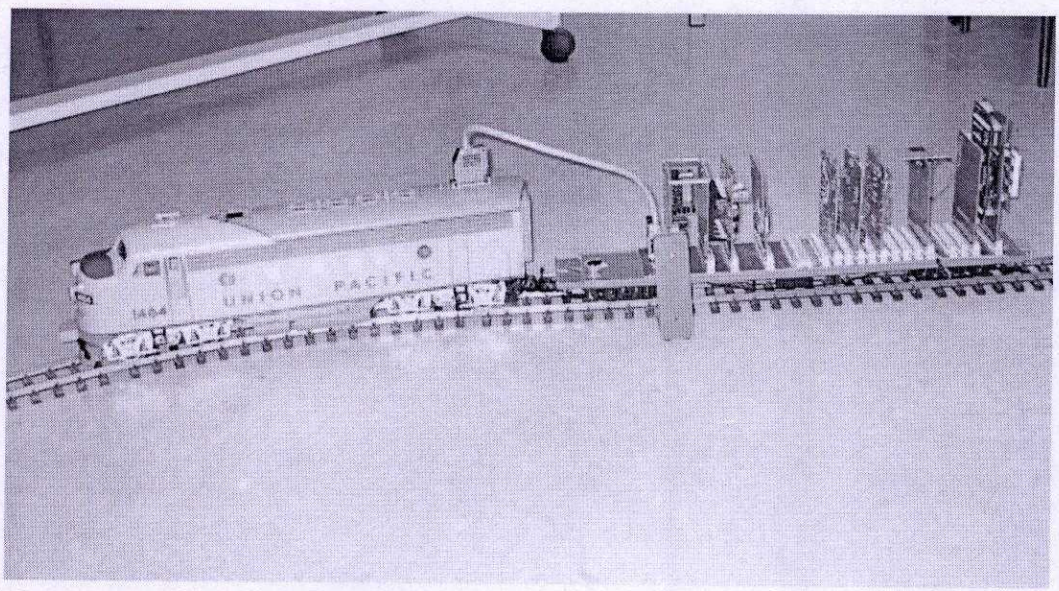



Fig1: First group peoplemover

## 2.2 Outline

The above figure shows the schematic view of the entire project. Here there is a track which has all the contents that a rail train track have. We have a bridge, tunnel etc. in the middle we can see a controller who will control the system. We can see the infra-red beacon, magnet beacon, control tower to complete this system. This controller may be needed to control if there a sudden accident occurs.

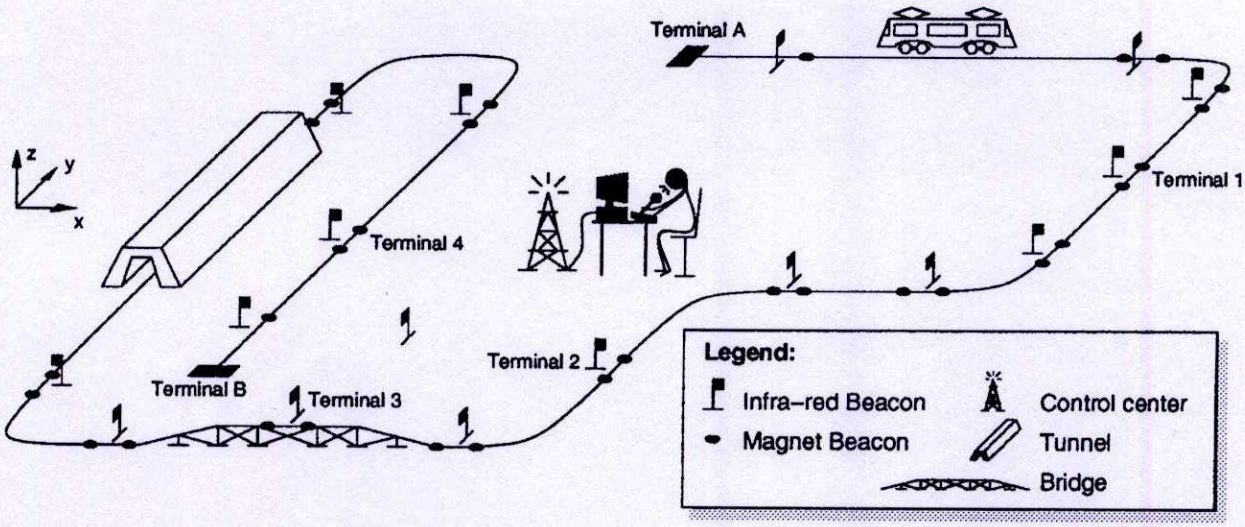



Fig2: Schematic view of peoplemover

## 2.3 System-level overview

The fig2 represents the whole system by a block diagram. In this figure the work would be done by all three groups is clearly defined.

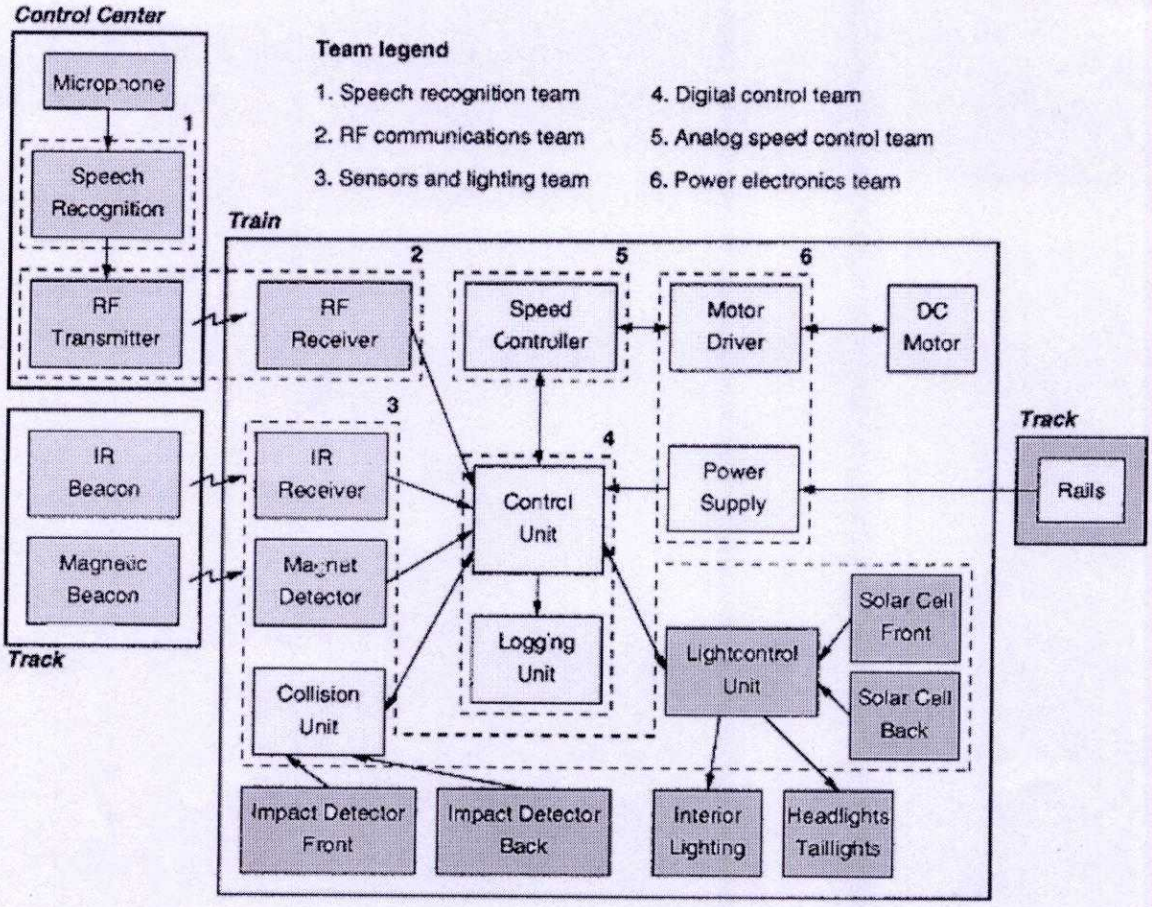



Fig3: System-level overview of the Peoplemover system

It shows a system-level overview of the PeopleMover system. At the top level, there are three main functional groups: the control center, the track, and the train. In the control center, the voice commands of the supervisor are picked up with a microphone in the presence of some interfering signal. The speech recognition unit translates the voice commands into a binary code, which is then amplitude-modulated onto a radio frequency (RF) carrier around 433 MHz and sent to the RF receiver located on the train. After detection, the RF receiver sends the commands directly to the control unit of the train. This signal path allows the PeopleMover to be operator-controlled, in case of emergencies. In normal mode, it runs autonomously. It therefore gathers information coming from beacons located along the track. For reasons of surety, there are infrared (IR) beacons and magnetic beacons. The IR beacons continuously broadcast an identification

code acting as electronic traffic signs. The IR receiver, located on the train, detects these codes and sends them to the on-train control unit. The ID code of the IR beacons is encoded in a pulse sequence. Every IR beacon corresponds to a specific predefined command (e.g., reduce speed when the train is entering a curve, stop in a terminal, etc.). As such, the beacons act as electronic traffic signs, indicating speed limitations and the locations of the terminals along the track. This speed can be reduced by the operator. Missing a beacon can have severe consequences. Therefore, every beacon is backed up by a pair of magnetic beacons located on the track before and after each IR beacon. All magnetic beacons are identical: they do not have an ID and therefore, the magnet detector on the train cannot distinguish between them. They only serve as surety back up since detecting two consecutive magnetic beacons without an IR beacon between indicates that an IR beacon has been missed. In this case, the train must reduce speed and go on at the slowest speed until it detects a new IR beacon and can resume its normal program. The beacon information, the on-train control unit is aware of its location and is able to decide upon the wanted speed and acceleration of the PeopleMover. The wanted speed is passed on digitally to the speed controller, an analog speed servo system, measuring and controlling both speed and imposed torque of the PeopleMover's electrical motor. A driver stage takes care of the power delivery to the electrical motor. To test the speed control system's ability to keep the velocity steady under different load conditions, a bridge with a ramp up and down has been included in the test track of Figure. The control unit is also able to send measured data (coming from the sensors or from the speed controller) or internal status information to a logging unit that stores this information in an onboard memory. The logging information can be used to debug any faulty behavior or to verify the correct operation of the PeopleMover.. To avoid the damage when the PeopleMover would collide with any foreign object on the track or with an end-of-track buffer, two optical impact detectors (front and rear) have been mounted on the PeopleMover. The collision unit processes these signals and sends a collision signal to the control unit, which in its turn can take the necessary action, i.e., stop the train by removing all

power from the motor. In addition, a light control unit manages the illumination of head and taillights of the train, as well as the interior lighting of the train wagons. Whereas the headlights are switched according to the PeopleMover's direction, the interior lighting is adapted to the available daylight. This daylight is being measured using two solar cells, front and rear, mounted on the train. To test this control unit, a tunnel has been included in the track setup. All other components are shut down. All of these requirements are specified in a system.

## 2.4 Inspiration

There are a lot of factors that inspire us to have an interest on this project. First of all project is done by students for using their theoretical knowledge. This project is perfect for implementing our theoretical knowledge into practically. As well as it is a very innovative idea for our country. We can use this peoplemover in our country for any short ranged area, for example small community, large university campus etc.

# Chapter 3

# PID Controller

## 3.1 Introduction to PID Controller

PID is the short term of "proportional, integral and derivative". A PID controller is a controller that contains elements of these three functions. In the literature on PID controllers, acronyms are also used at the element level:

The proportional element is referred to as the "P element," the integral element as the "I element," and the derivative element as the "D element."

The PID controller first came in market in 1939 and since then it has been the most widely used controller. A survey in 1989 shows that more than 90% of the controller used in process industries are PID controllers and advanced version of the PID controller.

PID controller is a method of feedback control that uses PID control as the main tool. The basic structure of conventional feedback control systems is shown in this Figure, using a block diagram representation.

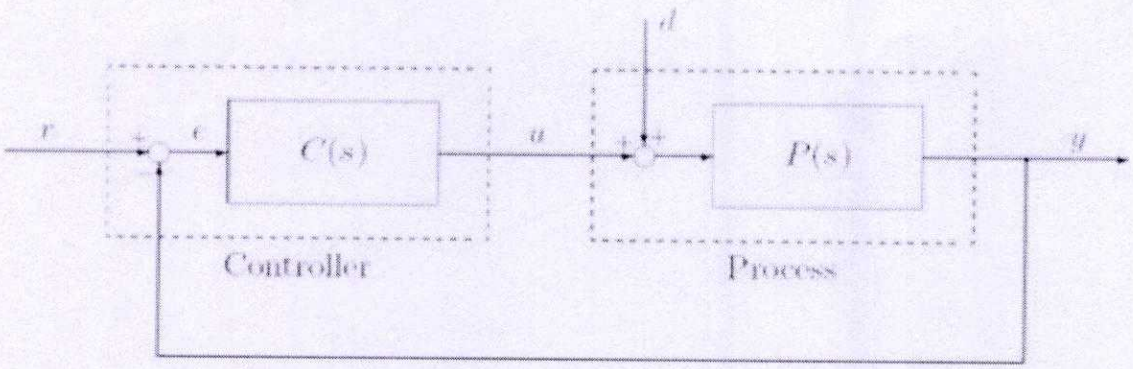



Fig4: Conventional feedback control system

Earlier PID controls system had the structure of the figure above, where it is mostly used as a compensator C(s).When it was used in this way, the three elements of PID controller produce with the following nature:-

**P element:** proportional to the error at the instant time which is the present error.

**I element:** proportional to the integral of the error up to the instant time which can be interpreted as the accumulation of the past error.

**D element:** proportional to the derivative of the error at the instant time, which can be interpreted as the future error.

So the PID controller can be understood as a controller that works with the present, past and future of the error into consideration. After the digital implementation started there are some certain changes in PID controller. but this change does not have influence on the designing or analysis of PID controller.

## 3.2 Control system and theory

In PID control system, this system works as a close loop system. It takes an error then compares it, measure it, make a correction on it, then give a more accurate output.

It uses some constant and there are some certain theory with which it works. The PID control system is named after three terms which correct errors.

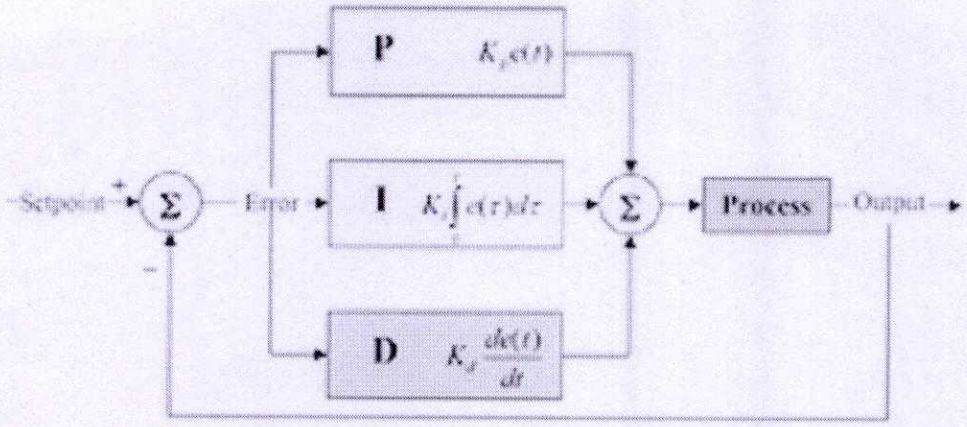



Fig5: PID controller

And finally by summing them the manipulated variable (MV) or output comes. Hence

$$O(t) = P_{out} + I_{out} + D_{out};$$

Where

$P_{out}$, $I_{out}$ and $D_{out}$ are the contribution of the output from the PID controller from each of the three terms as they are described bellow

## 3.2.1 Proportional Gain

The proportional gain makes a change to the output that is proportional to the current error value. It can be adjusted by multiplying the error by a constant $K_p$, which is also known as proportional gain.

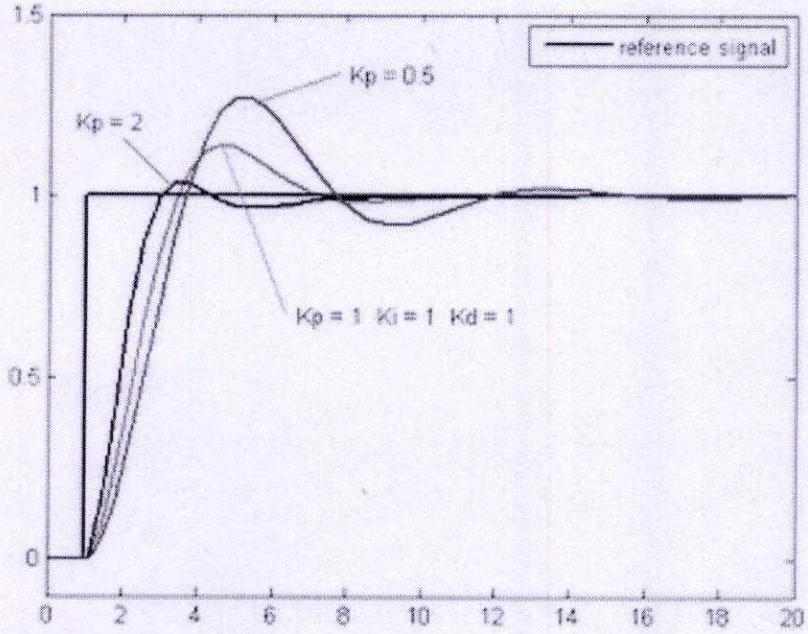



Fig6: PV vs. time, for three values of $K_P$ ($K_I$ and $K_d$ held constant)

So the proportional output is given by:

$P_{out} = K_P e(t)$

Where

$P_{out}$: Proportional term of output

$K_P$: Proportional gain, a tuning parameter

PV: Process value (or process variable), the measured value

e : Error

t : Time or instantaneous time (the present)

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output, response to a large input error, and a less responsive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

## 3.2.2 Integral Gain

The role of integral gain is proportional to both the magnitude of the error and the duration of the error. Summing the sudden error over time (integrating the error) gives the collected offset that should have been corrected previously. The collected error is then multiplied by the integral gain and added to the controller output. The magnitude of the contribution of the integral term to the overall control action is obtained by the integral gain $K_I$

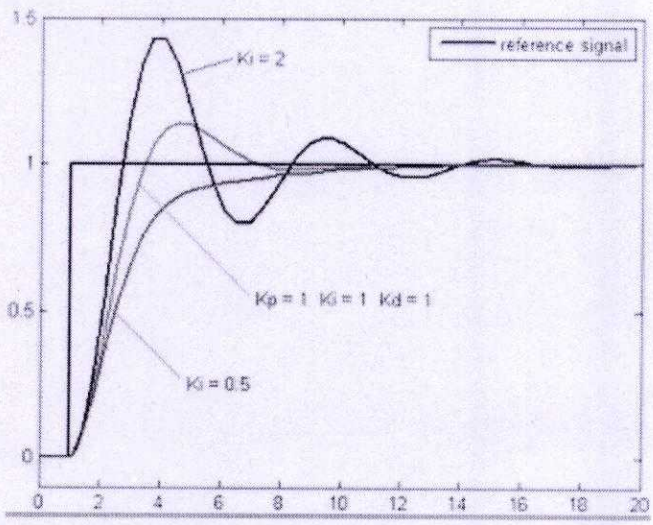



Fig7: PV Vs time, for three values of $K_i$ ($K_p$ and $K_d$ held constant)

The integral output is given by:

$$I_{out} = K_i \int_0^t e(\tau)\, d\tau$$

Where

$I_{out}$: Integral term of output

$K_i$: Integral gain, a tuning parameter

PV: Process value (or process variable), the measured value

e : Error

t: Time or instantaneous time (the present)

τ: a dummy integration variable

When the integral and proportional output is being added accelerates the movement of the process towards set point and removes the residual steady-state error that occurs with such controllers which is only proportional controller. However, since the integral term is responding to accumulated errors from the past, it can cause the present value to overshoot the set point values. These values cross over set point and then create a deviation in the other direction.

## 3.2.3 Derivative Gain

The rate of change of the process error is calculated by determining the slope of the error over time that is first derivative with respect to time and multiplying this rate of change by the derivative gain $K_d$. This magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, $K_d$.

The derivative output is given by:

$$D_{out} = K_d \frac{d}{dt} e(t)$$

Where

$D_{out}$: Derivative term of output

$K_d$: Derivative gain, a tuning parameter

PV: Process value (or process variable), the measured value

e: Error

t: Time or instantaneous time (the present)

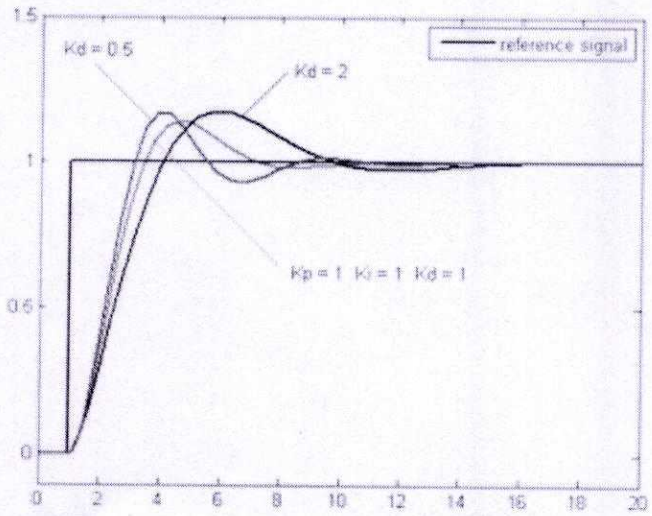



Fig8: PV Vs time, for three values of $K_d$ ($K_p$ and $K_i$ held constant)

This derivative output slows the rate of change of the controller and this effect is most noticeable close to the controller set point. Therefore, derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller process stability. However differentiation of a signal amplifies noise and after that this term in the controller is highly sensitive to noise in the error term, and can cause process to become unstable if the noise and the derivative gain are sufficiently large.

## 3.3 Final Result

The proportional, integral, derivative outputs are summed to calculate the output of the PID controller. Defining u (t) as the controller output the final form of the PID algorithm is:

$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t)$$

Where the parameters are:

### Proportional gain, Kp:

Larger values typically mean faster response since the larger the error, the larger the proportional term compensation. An excessively large proportional gain lead to proportional gain will lead to process instability and oscillation.

### Integral gain, $k_i$:

Larger values imply steady state errors are eliminated more quickly. The trade-off is larger overshoot; any negative error integrated during transient response must be integrated away by positive error before reaching steady state.

### Derivative gain, $K_d$:

Larger values decrease overshoot, but slow down transient response and may lead to instability due to signal noise amplification in the differentiation of the error.

## 3.4 Performance

If the PID controller parameters (e.g. proportional gain, integral gain and derivative gain) are chosen incorrectly, the controlled process input can be unstable, i.e. its output diverges, with or without oscillation, and is limited only by saturation or mechanical breakage. Instability is caused by excess gain, particularly in the presence of significant lag.

Generally, stability of response is required and the process must not oscillate for any combination of process conditions and set point, though sometimes marginal stability is acceptable or desired.

## 3.5 Best output

The best behavior on a process change or set point change varies depending on the application.

Two basic requirements are:

1. Regulation e.g. disturbance rejection – staying at a given set point;
2. Command tracking e.g. implementing set point changes.

These two refer to how well the controlled variable tracks the desired value. Specific criteria for command tracking include rise time and settling time. Some process must not allow an overshoot of the process variable beyond the set point if, this would be unsafe. Other process must minimize the energy expended in reaching a new set point.

# Chapter 4

# Pulse width modulation

## 4.1 Introduction

A powerful technique for controlling analog circuits with a processor's digital outputs is Pulse width modulation (PWM). PWM is used in a wide variety of applications, arranging from measurement and communications to power control and conversion.

PWM is a commonly used technique for controlling power to inertial electrical devices, made practical by modern electronic power switches. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast pace. The longer the switch is on compared to the off periods, the higher the power supplied to the load is. The PWM switching frequency has to be much faster than what would affect the load, which is to say the device that uses the power. Typically switching have to be done several times a minute in an electric stove, 120 Hz in a lamp dimmer, from few kilohertz (kHz) to tens of kHz for a motor drive and well into the tens or hundreds of kHz in audio amplifiers and computer power supplies. The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on. The main advantage of PWM is that power loss in the switching devices is very low. When a switch is off there is practically no current, and when it is on, there is almost no voltage drop across the switch. Power loss, being the product of voltage and current, is thus in both cases close to zero. PWM works also well with digital controls, which, because of their on/off nature, can easily set the needed duty cycle. PWM has also been used in certain communication systems

where its duty cycle has been used to convey information over a communications channel.

## 4.2 Applications

Pulse width modulation is usually used in three sides:

1. Telecommunication;
2. Power delivery;
3. Voltage regulation;
4. Audio effects and amplification.

But here we are going to use PWM for voltage regulation.

## 4.3 Voltage regulation

PWM is also used in efficient voltage regulators. By switching voltage to the load with the appropriate duty cycle, the output will approximate a voltage at the desired level. The switching noise is usually filtered with an inductor and a capacitor.

One method measures the output voltage. When it is lower than the desired voltage, it turns on the switch. When the output voltage is above the desired voltage, it turns off the switch.

Figure 9 shows three different PWM signals. Figure 1a shows a PWM output at a 10% duty cycle. That is, the signal is on for 10% of the period and off the other 90%. These three PWM outputs encode three different analog signal values, at 10%, 50%, and 90% of the full strength. If, for example, the supply is 9V and the duty cycle is 10%, a 0.9V analog signal results.

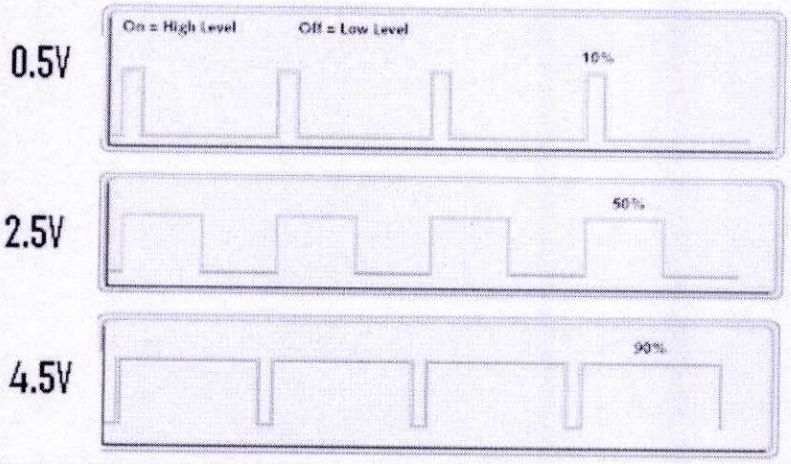



Fig9: PWM signals of varying duty cycles

## 4.4 Advantages of PWM

One of the advantages of PWM is that the signal remains digital all the way from the processor to the controlled system; no digital to analog conversion is necessary. By keeping the signal digital, noise effects are minimized. Noise can only affect signal if it is strong enough to change a logic 1 to a logic 0 or vice versa.

PWM finds application in a variety of systems. PWM is economical, space saving, and noise immune. And it's now in your bag of tricks

# Chapter 5

# Speed control

## 5.1 Introduction to speed control

The reason why a motor needs speed controller is to take a signal, representing the demanded speed and drive this motor at that speed. The controller may not actually measure the speed of the motor and if it dose it is called a feedback speed controller or closed loop speed controller and if not it is called an open loop speed controller. The feedback system is more complicated system. Motors come in a variety of forms, and the speed controller's motor drive output will be different dependent on these forms. The speed controller presented here is designed to drive a simple motor.

## 5.2 Theory of DC motor speed control

A simplest method to control the rotation speed of a DC motor is to control its driving voltage. The higher the voltage is the higher speed the motor tries to reach. In many applications a simple voltage regulation would cause lots of power loss on control circuit, so a pulse width modulation method (PWM) is used in many DC motor controlling applications. In the basic Pulse Width Modulation (PWM) method, the operating power to the motors is turned on and off to modulate the current to the motor. The ratio of "on" time to "off" time is what determines the speed of the motor. When doing PWM controlling, we have to keep in our mind that a motor is a low pass device. The reason is that a motor is mainly a large inductor. It is not capable of passing high frequency energy, and

hence will not perform well using high frequencies. Reasonably low frequencies are required, and then PWM techniques will work. Lower frequencies are generally better than higher frequencies, but PWM stops being effective at too low a frequency. The idea that a lower frequency PWM works better simply reflects that the "on" cycle needs to be pretty wide before the motor will draw any current (because of motor inductance).

The speed of a DC motor is directly proportional to the supply voltage, so if we reduce the supply voltage from 9 Volts to 4.5 Volts, the motor will run at half the speed. The speed controller works by varying the average voltage sent to the motor. It could do this by simply adjusting the voltage sent to the motor, but this is quite inefficient to do. A better way is to switch the motor's supply on and off very quickly. If the switching is fast enough, the motor doesn't notice it, it only notices the average effect. Speed controllers to drive a motor. When the switch is closed, the motor sees 9 Volts, and when it is open it sees 0 Volts. If the switch is open for the same amount of time as it is closed, the motor will see an average of 6 Volts, and will run more slowly accordingly. As the amount of time that the voltage is *on* increases compared with the amount of time that it is *off*, the average speed of the motor increases.

## 5.3 Reversing DC Motor

To reverse a DC motor, the supply voltage to the armature must be reversed, or the magnetic field must be reversed. In a series motor, the magnetic field is supplied from the supply voltage, so when that is reversed, so is the field, therefore the motor would continue in the same direction. We must switch either the field winding's supply, or the armature winding's supply, but not both.

In our case we use a signal from microcontroller to reverse the DC motor. We don't do it physically or by using any H-bridge, other than that we give a certain voltage to a pin in the microcontroller. When we give high voltage the train move in forward direction and when it is low it would reverse the direction.

## 5.4 Feedback Speed Control

To stop an accident or sudden errors on going forward, we need to have feedback control of the motor speeds. This means that the actual speed of each wheel is measured, and compared with all the other wheels. Obviously to go in a straight line, the motor speeds must be equal. However, this does not necessarily mean that the speed demand for each motor should be the same. The motors will have different amounts of friction, and so a 'stiffer' motor will require a higher speed demand to go as fast as a more free-running motor.

To perform the same function as described above in software requires that the software has digital representation of the speed of each wheel, and can finely control the width of the PWM signal sent to each wheel. To get the speed of each wheel, an optical encoder must be used as in the analogue method, but the output of it must be sent to the microcontroller. This is achieved using a counter, clocked by the speed controller, which the microcontroller can read, and can clear. At regular intervals, the microcontroller must read the counter, and then clear it. The interval depends on the maximum speed of the robot, the diameter of the wheel, the number of slots in the speed encoder's disc, and the number of bits of the counter.

# Chapter 6

# Hardware

## 6.1 Introduction

People Mover is a hardware based project; we need a lot of things to complete this project. But for control group, we do not need so much of them. As our work basically to control speed automatically with changing mode and feedback we use microcontroller to do this job done for us. We write a code, implement it on microcontroller and use it to do this task. We use ATMEGA32 to do this work. In addition we do the motor driver circuit as well. We did it by using L293D. So for basic control purpose we use microcontroller only but as we design the motor driver circuit so we use the L293D.

## 6.2 Microcontroller

### 6.2.1 Introduction to microcontroller

By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote

controls, office machines, appliances, power tools, and toys. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

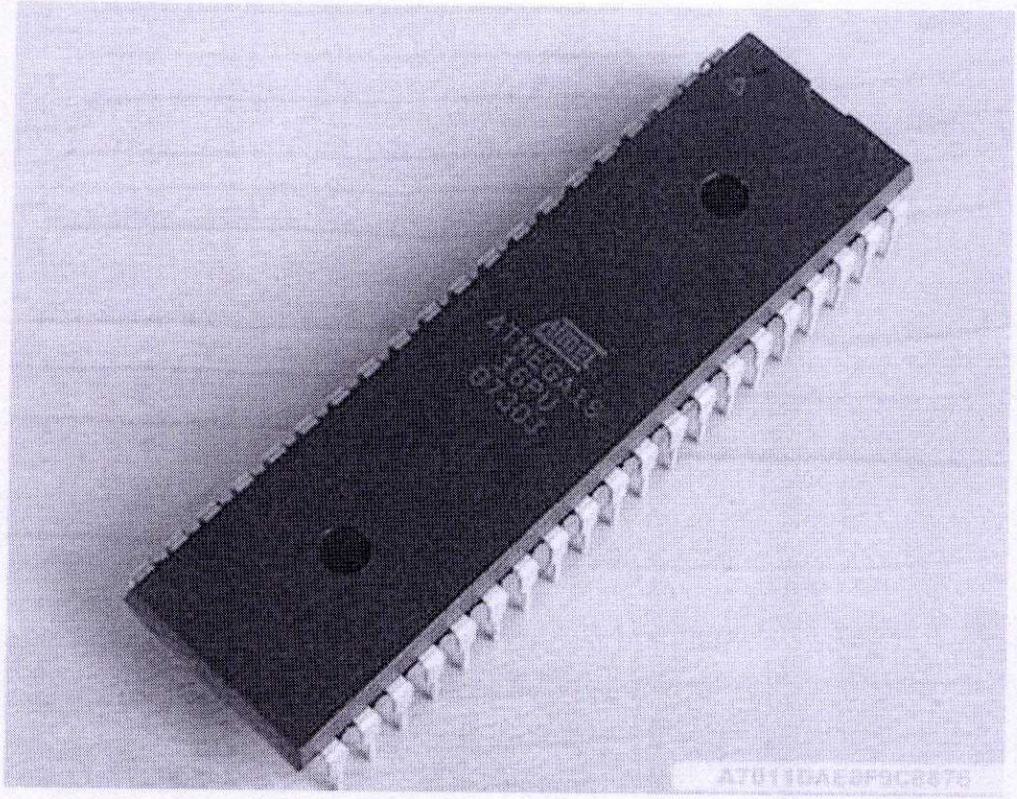


Fig10: ATMEGA16

In this project we are using ATMEGA16 which contains 16 Kbytes of in-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbytes SRAM, 32general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes.

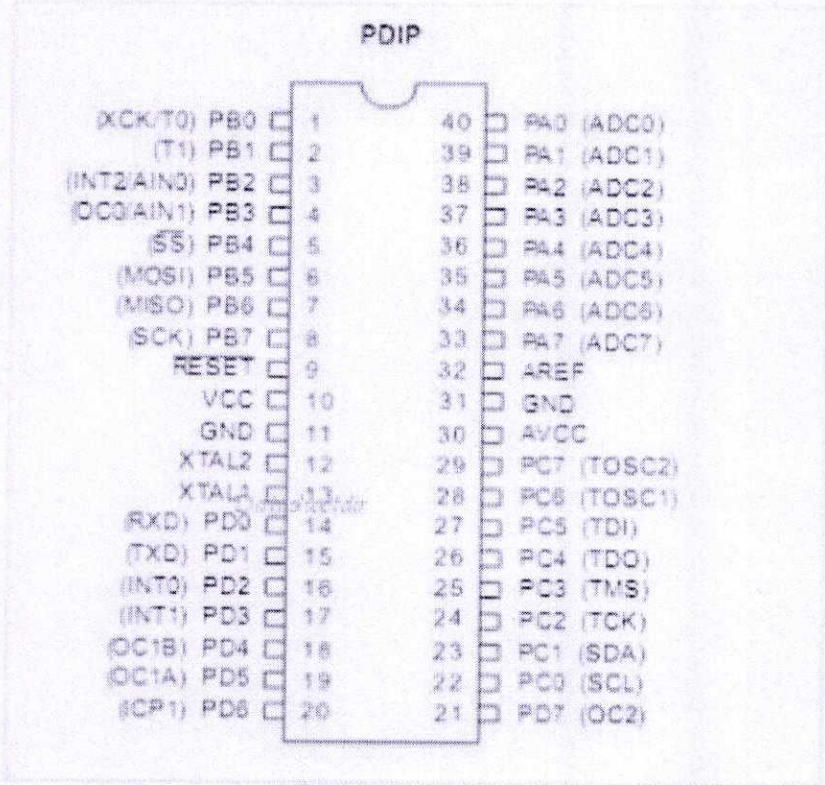



Fig11: Layout of ATMEGA16

The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM; Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping.

## 6.2.2 Designing a PID Using Microcontroller

Making a proportional integral and derivative controller is control system group's task. It can be done by making circuit. But it would be more complex that time, because we then need to make a circuit by using Op-Amps, registers, inductors,

diodes, capacitors, and lots of others. It is not an easy job to make it as well as there are some problems. Any of the item can be burnt or can be raised any problem. So it is not possible all time, that's why we prefer to use microcontroller and write a code to design a PID controller. We write a code for proportional gain as we know that it cans response to the error in a fastest way. Our code will be included in the appendix section where we show how our code works.

## 6.3 Advantages of microcontroller

It is clearly an advantage to use one chip over a lot of elements. And it is not the only reason we use microcontroller. It is a high-performance, low-power needed chip. It has real time counter with separate oscillator as well as four PWM channels and 32 Programmable I/O Line. It needs only 2.7 - 5.5V for operating ATmega16. Frequency is in the range of 0 - 8 MHz. It is not expensive as well. Though ATMEGA16 is not available in market but there are lots of other microcontrollers with better features.

## 6.4 Driver Circuit

### 6.4.1 Introduction

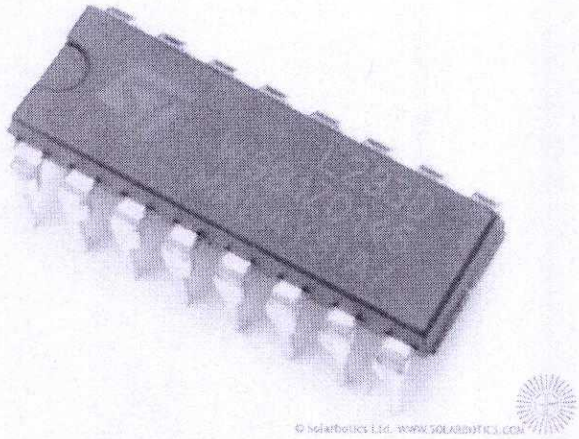


Fig12: L293D

All motors require drive circuitry which controls the current flow through the motor windings. Now we have code to run proportional controller but we need that driver circuit by which we are going to use the pwm signal to drive the motor. This includes the direction and magnitude of the current flow. The motor cannot commutate the windings (switch the current flow), so the control circuit and software must control the current flow correctly to keep the motor turning smoothly. The circuit is a simple half-bridge on each of the three motor windings. To do this job we use L293D chip

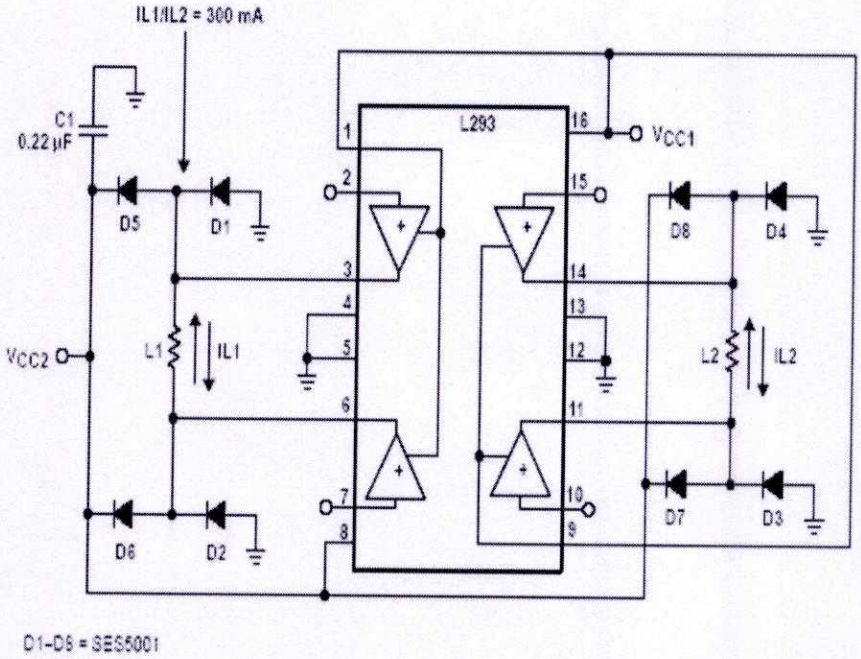



Fig13: L293D (inside circuit)

The L293 and L293D are quadruple high-current half-H drivers. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. This device are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high voltage loads in positive-supply applications. To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included. This device is suitable for use in switching applications at frequencies up to 5 kHz. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

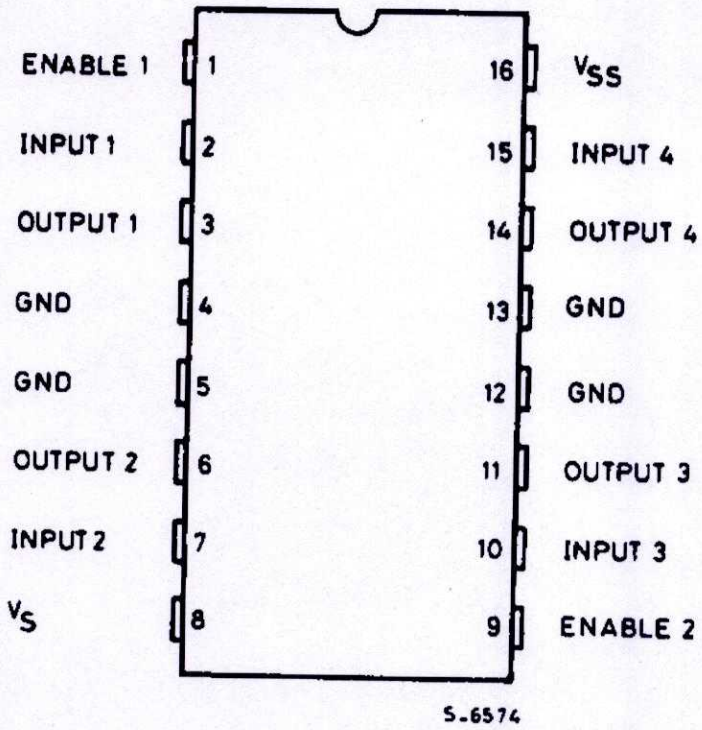



fig14: Pin configuration of L293D

## 6.3.2 Designing Motor driver circuit

As Stated before L293D is a dual H-Bridge motor driver, so with one IC we can interface two DC motors which can be controlled in both clockwise and counter clockwise direction and if we have motor with fix direction of motion then we can make use of all the four I/Os to connect up to four DC motors.

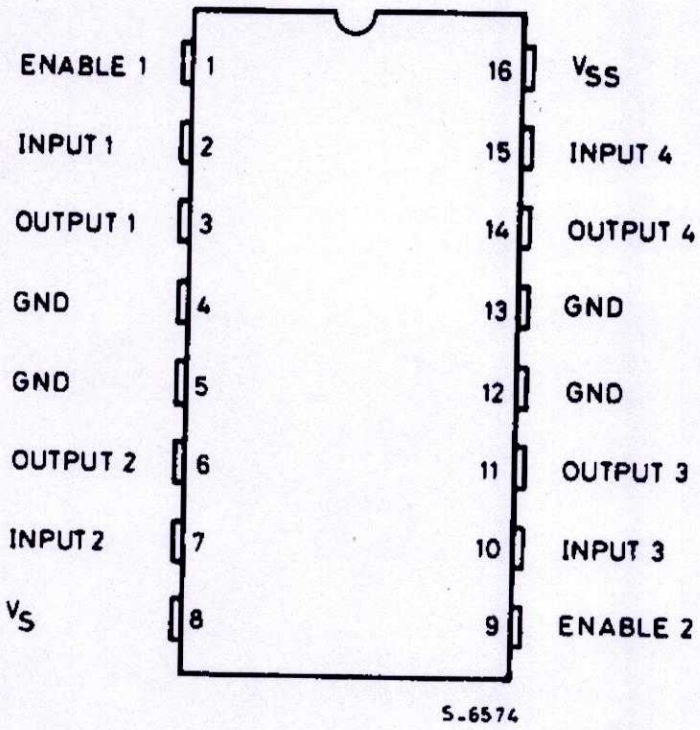



fig14: Pin configuration of L293D

## 6.3.2 Designing Motor driver circuit

As Stated before L293D is a dual H-Bridge motor driver, so with one IC we can interface two DC motors which can be controlled in both clockwise and counter clockwise direction and if we have motor with fix direction of motion then we can make use of all the four I/Os to connect up to four DC motors.

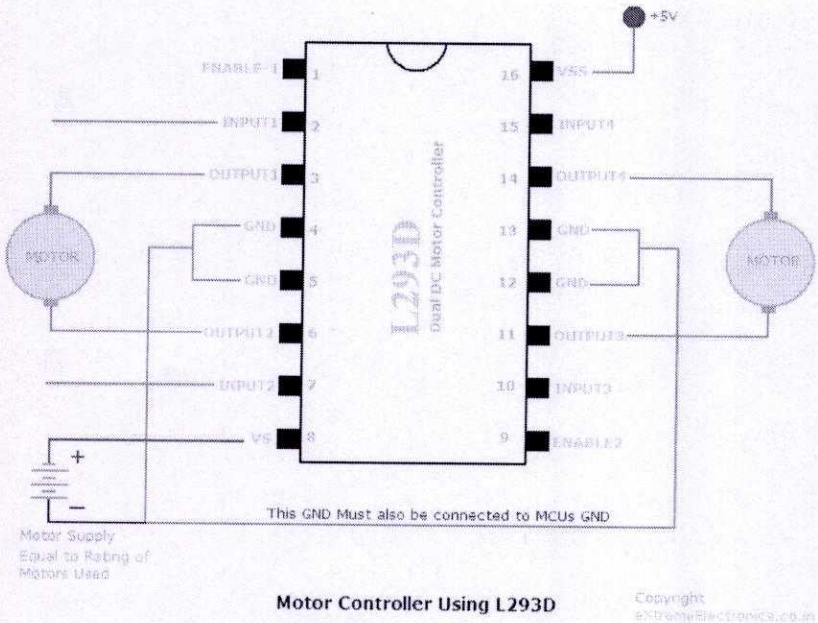



Fig15: Controlling motor by using L293D

Now mostly like shown in the figure we connect our circuit in this way. Enable-1 is given 5v from power supply to enable 1 side of the chip. Then Input1 & Input2 is given the output of microcontroller from the 18$^{th}$ & 19$^{th}$ pin of it. It is the proportionally gained output of microcontroller. Now the Output1 and Output2 pins are connected to the input pins of the motor of the train. The 4$^{th}$ & 5$^{th}$ pins are used for heat sink, but for now we make them ground. On the 8$^{th}$ pin 9v has been given. The 16$^{th}$ pin is vss, so we give 5v there. The output voltage we get from the L293D is the amplified form of voltage given to it from microcontroller. Theoretically it should be 2 times more then the given voltage but in reality it is approximately 2V. Here is the Table which shows for which voltage we get what voltage.

| Input (from microcontroller) | Output (of driver circuit) |
|---|---|
| 0.0v | 0.0v |
| 0.5v | 0.5v |
| 1.0v | 1.5v |
| 1.5v | 2.5v |
| 2.0v | 3.70v |
| 2.5v | 4.70v |
| 3.0v | 5.95v |
| 3.50v | 6.98v |
| 4.0v | 8.20v |
| 4.7v | 9.00v |

Table1

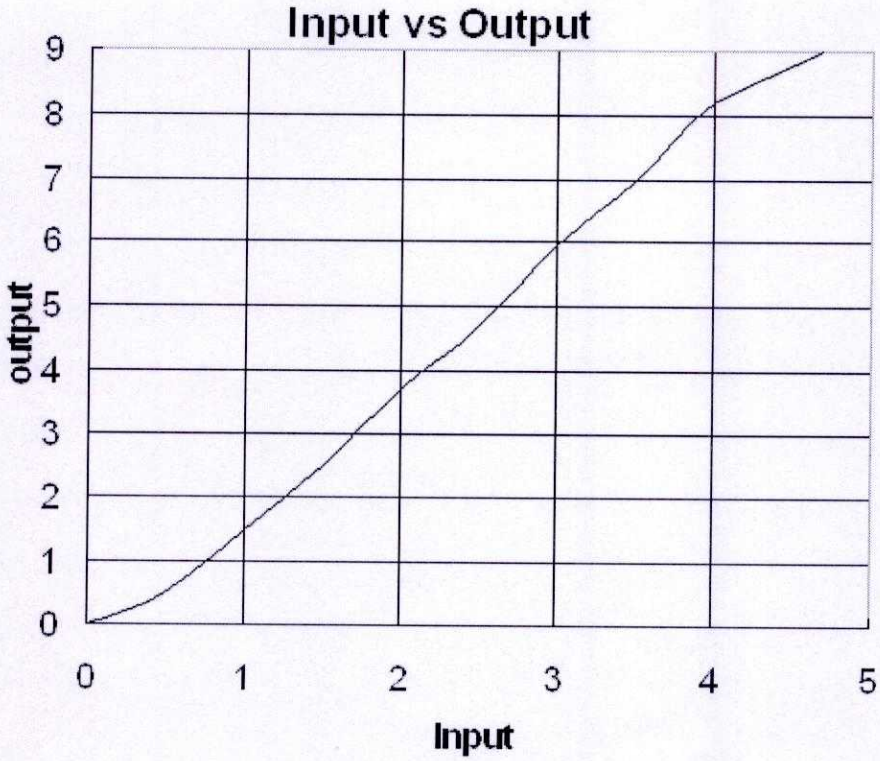



Fig16: Linear representation of I/O Vs O/P

And this is the circuit figure which we made for our project with L293D.

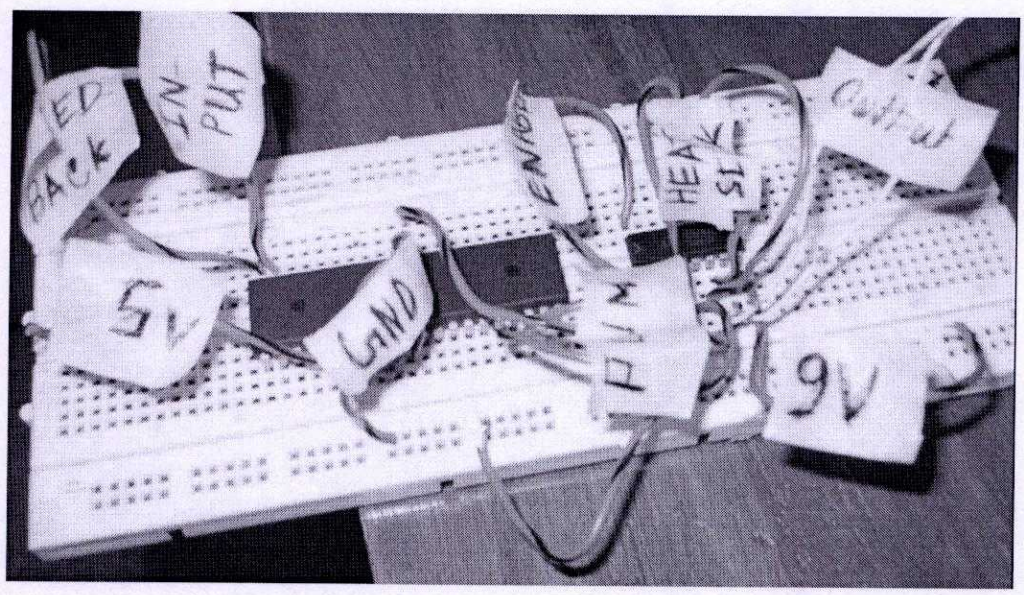



Fig17: Project Circuit on breadboard

## 6.4 Advantages of driver circuit

Provide bidirectional current simultaneously 600mA. Voltage range is 4.5V to 36V. Give amplified output voltage. Two motor can be run together. Easy to maintain

# Chapter 7

# Project Construction

## 7.1 Introduction to our project

This is a totally different project then any other project on many points. First of all this is a project, where work is divided into three groups as stated before, so not just working on our portion, we have to maintain a good connection with other two groups and after all other work done we have to commission all the work together. The whole project is described on chapter two before. Now in this chapter we are going to discuss mainly on control system work what we have done so far but we will try to give a flavor of whole project we did.

For the controlling purpose we need to design a PID controller. Since it is not an easy job to do in this level, we go for proportional controller. This is a process where our People Mover would move in such a proportion constantly whether the train is going up-hill or down-hill it will maintain a constant speed so that passengers in the train do not fell uncomfortable and make the system as perfect as possible.

Now after planning of controlling speed in a certain constant way we need to implement this for our people mover. To do this we use a chip named L293D, which stated before in this paper.

In this chapter we are going to discuss about how everything is done in our project.

## 7.2 Circuit design

As stated earlier we use microcontroller (ATMEGA16) to make the proportional controller. We write a code in c language and implement it for our project. While writing this code we have to think of some major points. They are-

- It has to be accurate
- It has to have a point by point variation
- Cannot exceed the memory capacity of microcontroller
- Has to have variable feedback, references, direction changing proficiency.

The control system will work in such a way so that the speed of the people mover does not change. Before doing this to make the thing more accurate we do some experiment. We give the microcontroller an analogue voltage, it takes that voltage and converts it into digital value by the built in ADC (Analogue to Digital Converter) that it has. And again it would rather give digital value but the DAC (Digital to Analogue Converter) in the microcontroller convert it into the analogue value and show us that. So to make the code more accurate we first find out for every decimal change of analogue voltage what could be the digital value for both input and feedback. Here two tables have the data of this experiment.

| Analog value | Digital value |
|---|---|
| 0.0 | 00000011 |
| 0.5 | 00001101 |
| 1.0 | 00011011 |
| 1.5 | 00101000 |
| 2.0 | 00110101 |
| 2.5 | 01000011 |
| 3.0 | 01001111 |
| 3.5 | 01011101 |
| 4.0 | 01101010 |
| 4.5 | 01111000 |
| 4.7 | 01111111 |

Table2

Conversion on analog to digital value

| Digital value | Analog value |
|---|---|
| 000000010101 | 0.0 |
| 000001101100 | 0.5 |
| 000011011000 | 1.0 |
| 000101000001 | 1.5 |
| 000110101100 | 2.0 |
| 001000011000 | 2.5 |
| 001010000011 | 3.0 |
| 001011101110 | 3.5 |
| 001101011010 | 4.0 |
| 001111000110 | 4.5 |
| 001111111111 | 4.7 |

Table3

Conversion of digital value to analog value

These are the digital value which we got in decimal format. We need these values to write down our code and have more accurate values.

Now the primary work is done, it is time to write the code. As described before this is an close loop like the figure shown below.

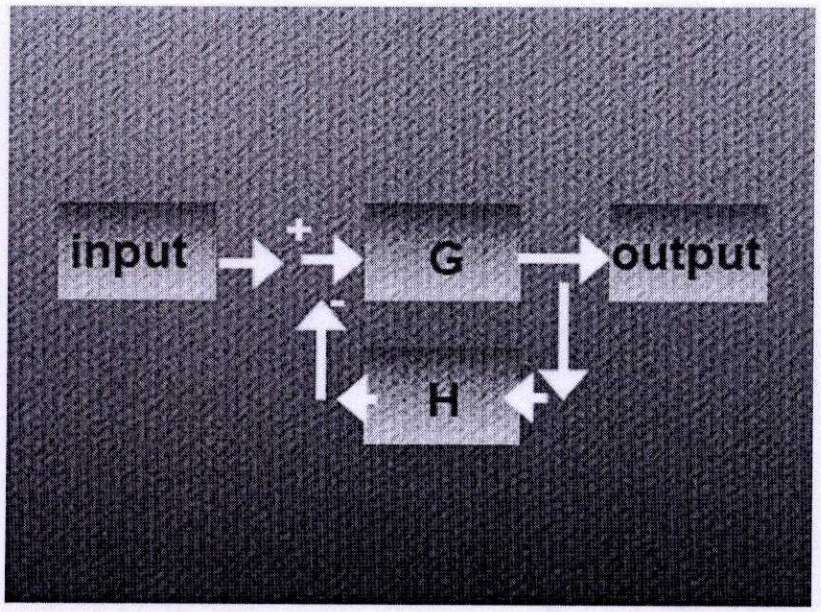



Fig18: Example of a close loop system

The reference (input) is given by the instructor depending on which speed he wants to run the train. Depending on that voltage initially the train will start moving. At first there is no proportional gain ($k_p$) so we get output straight forward and at the same time we will get feedback voltage (H). Our microcontroller takes that value, compare it with the input and at that instant if the feedback is less than the input, the value of output voltage will decrease but if the feedback is greater then input, output we get from the microcontroller will be less then the input voltage. Moreover is both the feedback and input is same then output would follow the input.

For example if we want the train's speed is half of its full speed, we will give 2.5V as input voltage. At first there is no feedback so output will follow the input directly. After some time feedback will be the same 2.5V, so output is 2.5V. The same thing will happen if the rail line is straight. It means if there is no hump which makes the train to change its speed, train will follow the initial speed and on that time proportional gain ($k_p$) is zero. However in the case of having humped and other distraction in the line is not that simple. On those case if the input voltage is half, then while going up-ward feedback will be less. It expresses that

the input voltage have to be more than it was given so that it can maintain the half speed steadily. After that when the train put itself on the bridge and start running on the plane path feedback will increase. So subsequently, comparing feedback and input, output voltage will reduce then the previous value. Later, when train goes downward, to maintain the constant speed feedback will be more than before. Thus microcontroller will give a lower output voltage after comparing both the feedback and input. The output voltage value will change with a fixed ratio. This ratio is the proportional gain. To find out the gain we did another experiment. We measure the value of RPM (Rotation per Minute) of the train for changing the value of input of the motor.

This table has the values of different rpm for different voltage. As this is a small motor and the battery we used is less powerful so we can not go beyond 3V as an input voltage.

| Voltage | RPM |
|---|---|
| 3.5v | 62 |
| 5.0v | 81 |
| 5.5v | 99 |
| 6.5v | 110 |
| 7.0v | 133 |
| 7.5v | 140 |
| 8v | 149 |
| 8.5v | 158 |
| 9v | 171 |

Table4

Voltage Vs Rotation per Minute

And this is the graph which we designed by using matlab.

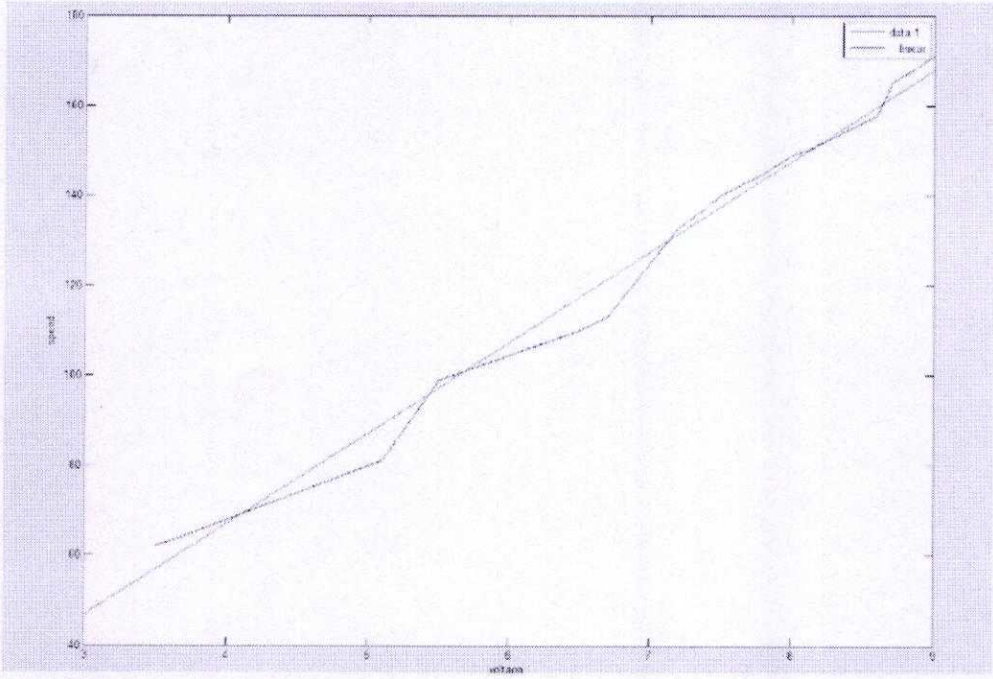


Fig19: Voltage Vs RPM

So now we have the voltage to use to run the train as input voltage for it. These two outputs of microcontroller ($18^{th}$ & $19^{th}$) are straight away connected to L293D ($2^{nd}$ & $7^{th}$). Then the amplified voltage output ($3^{rd}$ & $6^{th}$) of this chip is connected to the train. Finally the train will move in any desired speed we want.

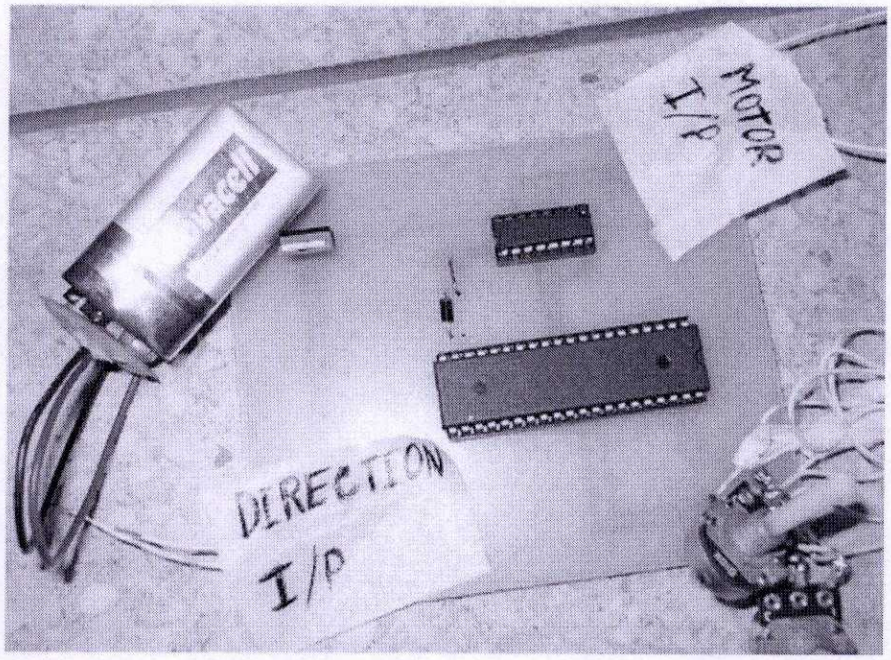


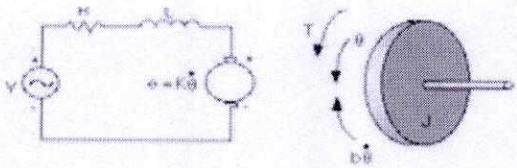

Fig20: Final circuit on printed circuit board (PCB)

# Chapter-8

# Result

## 8.1 Simulation result

Aim of a project is to learn a lot about that project. On every aspect how the project can be done successfully we need to find out that one as well. So to do that, in our first semester we did the whole project in matlab simulink. Our main focus was on developing a dc motor close loop circuit on matlab, and found that what are the output curves and whether it is similar to the theoretical curve. We start to make a dc motor which will work as a power driver in the system.



$$J\frac{d^2\theta}{dt^2} = T - b\frac{d\theta}{dt} \implies \frac{d^2\theta}{dt^2} = \frac{1}{J}(K_t i - b\frac{d\theta}{dt})$$

$$L\frac{di}{dt} = -Ri + V - e \implies \frac{di}{dt} = \frac{1}{L}(-Ri + V - K_e\frac{d\theta}{dt})$$

These two transfer functions are used to design the dc motor. this is an example of an open loop circuit.

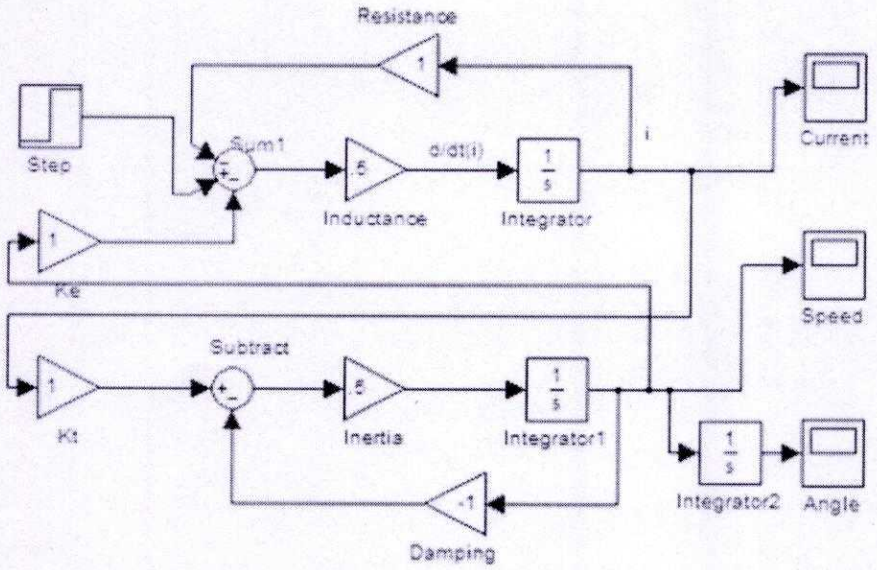
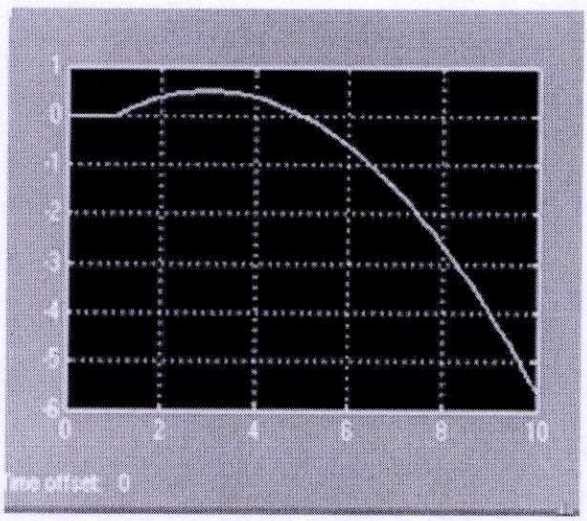
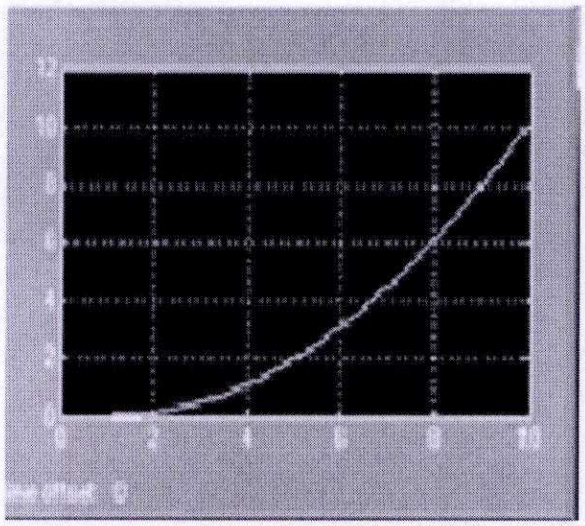



Fig21: Open loop system (mahlab simulink)

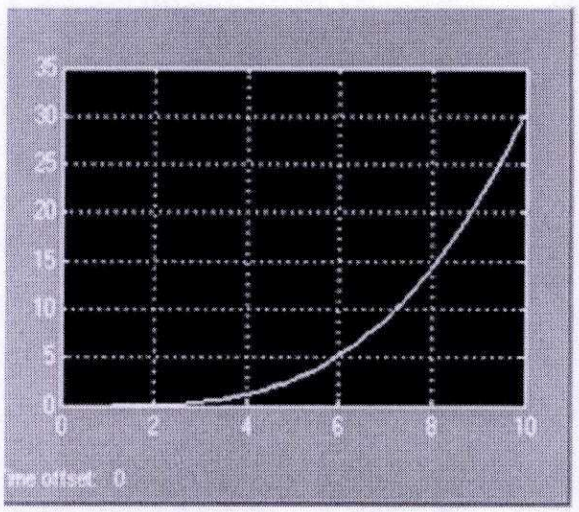
As this is an example of open loop circuit so its output will give the open loop response.




Current



Speed



Position

Fig16: Outputs of open loop (matlab simulink)

After that we start working with close loop system. And the circuit will be

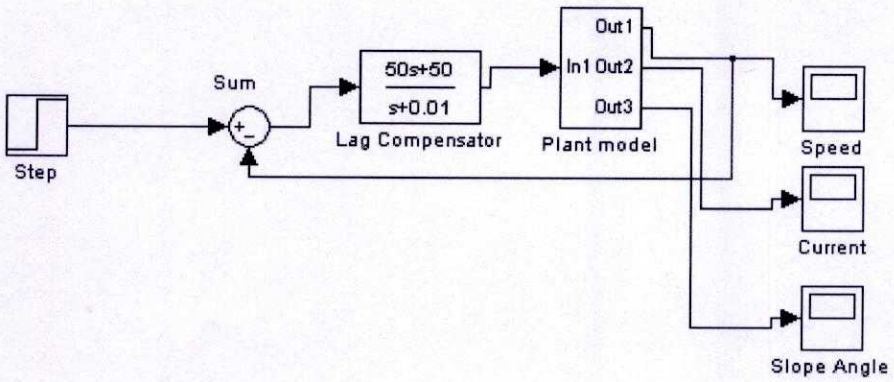



Fig17: DC motor (close loop by mathlab simulink)

Plant model is the subsystem of a DC motor(open loop)

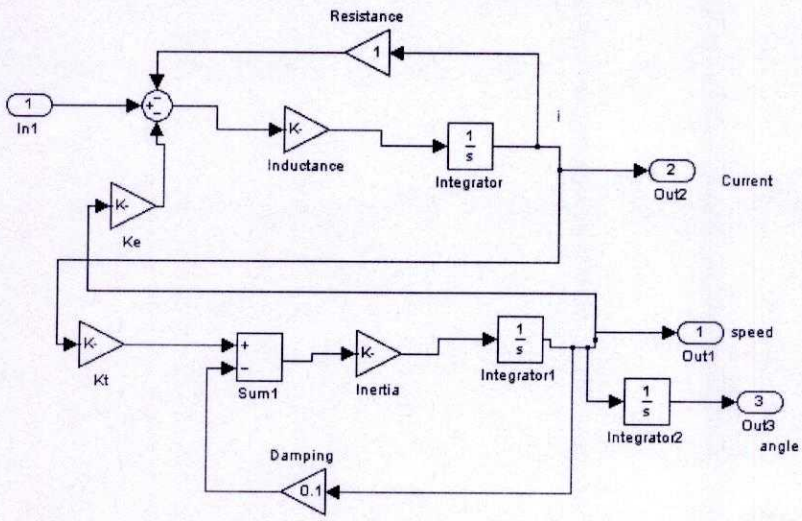



Fig18: Plant model(inside of it)

and the output of this circuit is

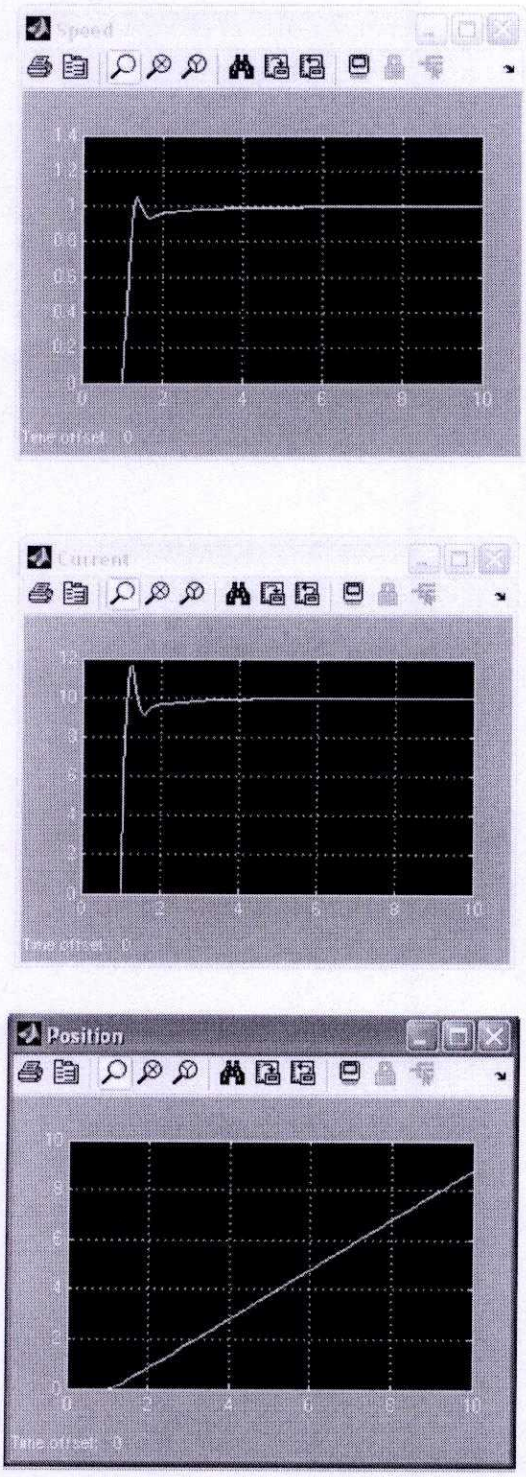



Fig19: Outputs of close loop

All of these curve with the theory. So it seems that we were successful to sort out our mathematical model is running properly.

## 8.2 Microcontroller result

After doing that matlab simulation, now it's time to built proportional controller. We are going to do this by writing a code in c language. So to write this First of all we build a flow chart for "P" controller.

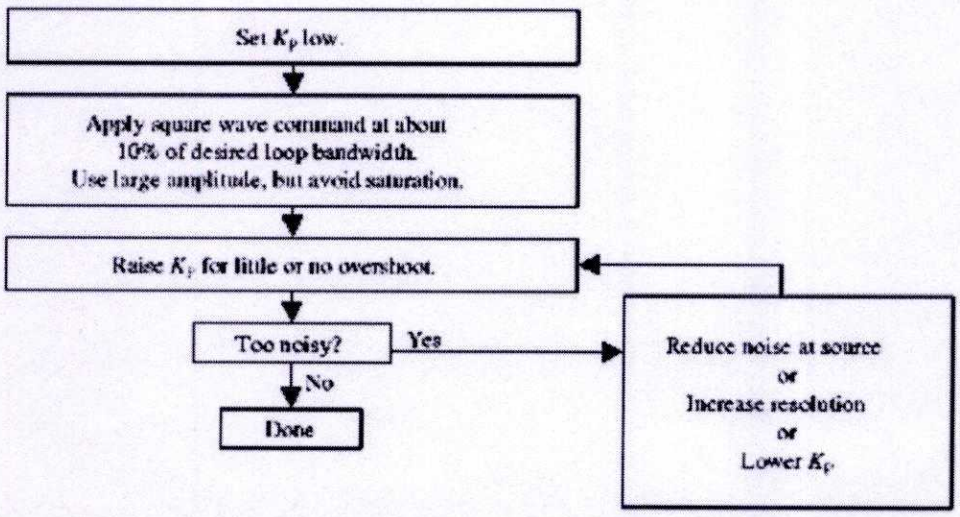



Fig20: Flowchart for proportional controller

Slowly we solve this p controller flow chart and found a c code for that. We are going to put that code on appendix. We got the value of input, error and feedback. It shows that the amount of input voltage is applied on the microcontroller and the amount of feedback is changed the error is balancing them which is a basic of proportional theory. All those data are shown below the table

| Input | Error | Feedback |
|---|---|---|
| 5 | 2.6 | 2.36 |
| 4.5 | 2.23 | 2.25 |
| 4 | 1.9 | 3.1 |
| 3.5 | 1.7 | 1.7 |
| 3 | 1.5 | 1.5 |
| 2.5 | 1.2 | 1.25 |
| 2 | 1 | 1 |
| 1.5 | .75 | ,75 |
| 1 | .712 | .28 |

Tabel5

Input, error & feedback from microcontroller

## 8.3 Servo system result

To see that our microcontroller is working appropriately as a "P" controller or not we use servo motor. We made to work as a proportional controller and finally we got these data.

| Input | Error | Feedback |
|---|---|---|
| 5 | 2.6 | 2.36 |
| 4.5 | 2.23 | 2.25 |
| 4 | 1.9 | 3.1 |
| 3.5 | 1.7 | 1.7 |
| 3 | 1.5 | 1.5 |
| 2.5 | 1.2 | 1.25 |
| 2 | 1 | 1 |
| 1.5 | .75 | ,75 |
| 1 | .712 | .28 |

Tabel6

Input, error &feedback from servo system

We can see from both of the chart that the theoretical and microcontroller output is almost same. So from these two charts we can say that the work so far we have done is ok.

# Chapter 10

# Conclusion

Our peoplemover project will use to take people from one place to another for a small place or a certain area. This is an independent vehicle which can also be controlled by a supervisor staying in the control tower. This is "real" project. As well as this can also be implemented in Bangladesh. It is an innovative idea for our country. It will open a new door for transportation in our country. The PeopleMover project and the speed control system task within it, is highly regarded. Students and professors similarly find this real application a healthy change from courses with purely theoretical lectures and tutorial-like exercise sessions. The goal of having the train run within prescribed specifications, and working toward this goal is a thrilling experience for all of the people involved. The aim of this project is to apply theoretical course material in a practical application, learn how to do multidisciplinary project work, and gain experience with communication between teams. These goals are more than fulfilled. Above all, the students learn in a playful way, with the ultimate reward of seeing the train riding smoothly along its tracks

# Chapter 9

## Future works

So far we have done with our proportional controller. As we don't have enough time and there is a shortage of microcontroller and some other problems we cannot finish the whole PID controller. To do this again we will build a flow chart for "I" controller, then we made a "C" code on basis of that. So we can finish the controller part. As mentioned before for time shortage we cannot finish the solar light control part as well, so in future we will work on it. Design and implement the complete PID controller which is not done yet. Then PC control real time system is our final desired work.

# References:


1. http://www.hackchina.com/en/s/pid.c
2. http://www.hackchina.com/en/r/117651/motorctrl.c__html
3. http://www.hackchina.com/?lang=en&q=microcontroller+based+pid+algorithms+c+code
4. http://www.hackchina.com/en/r/117651/motorctrl.c__html
5. http://www.hackchina.com/en/s/pid.c
6. http://www.hackchina.com/?lang=en&q=microcontroller+based+pid+algorithms+c+code
7. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1183680
8. ftp://ftp.esat.kuleuven.ac.be/sista/doclo/reports/04-71.pdf
9. http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf
10. http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/3/L293D.shtml
11. Andersen and J. Hasen, "Engineers of tomorrow: Knowledge, insight

Denmark, Sept. 2001and skills needed to work across borders," in *SEFI Conf.*, Copenhagen.


# Appendix

# Appendix

- Source code

```
#include<avr/io.h>
#include<util/delay.h>
#define set_FORWARD TCCR1A = 0x81
#define set_REVERSE TCCR1A |= (1<<COM1B1)
#define F_CPU 1000000UL
void InitADC()
{
ADMUX =(1<<REFS0) ;//| (1<<REFS1) ;                    // For Aref=AVcc;
ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1) | (1<<ADPS0); //Rrescalar div factor =128
}uint16_t ReadADC(uint8_t ch)
{  //Select ADC Channel ch must be 0-7
  ch=ch&0b00000111;
  ADMUX|=ch;
 //Start Single conversion
  ADCSRA|=(1<<ADSC);
  //Wait for conversion to complete
  while(!(ADCSRA & (1<<ADIF)));
  //Clear ADIF by writing one to it
  ADCSRA|=(1<<ADIF);
  return(ADC);
```

```
}
void delay_ms(unsigned int ms){
 while(ms){
 _delay_ms(1.000);
 ms--;
    }
}
////////////////////////////////////
int main(){
 unsigned int adc_val,adc_val1;
   InitADC();
   DDRD |= (1<<PD5) | (1<<PD4);
    TCCR1A |= (1<<COM1A1) | (1<<COM1B1) |(1<<WGM10) | (1<<WGM11) ;
    TCCR1B |= (1<<CS10) | (1<<WGM12);
   while(1){
    InitADC();
    adc_val=ReadADC(0);
  adc_val=adc_val/8;
     InitADC();
     adc_val1=ReadADC(1);
            adc_val1=adc_val1/8;
  //  set_FORWARD;
  TCCR1A |= (1<<COM1A1) | (1<<COM1B1) |(1<<WGM10) | (1<<WGM11) ;
 TCCR1B |= (1<<CS10) | (1<<WGM12);
/*
```

```
 if(adc_val1<=127 && adc_val1>=75)
 {    if(adc_val<=127 && adc_val>=0)
{     OCR1A=1012;
   OCR1B=0;   } }
 if(adc_val1<=74 && adc_val1>=45)
    if(adc_val<=127 && adc_val>=0){
      OCR1A=735;
            OCR1B=0;}
  if(adc_val1<=44 && adc_val1>=0)
    if(adc_val1<=127 && adc_val1>=0){
   OCR1A=412;
         OCR1B=0;}*/
//4.0
  if(adc_val1==106 || adc_val1==107 || adc_val1==105 || adc_val1==108 || adc_val1==109 ||
adc_val1==104)
         {   if(adc_val==127)
          {OCR1A=600;
         OCR1B=0;  }
          if(adc_val==115 || adc_val==116 || adc_val==114 || adc_val==117)
          {OCR1A=650;
             OCR1B=0;  }
 if(adc_val==106 || adc_val==107 || adc_val==105 || adc_val==108 )
         {  OCR1A=870;
             OCR1B=0;  }
         if(adc_val==93 || adc_val==94 || adc_val==92 || adc_val==95)
         {   OCR1A=920;
```

```
            OCR1B=0; }
  if(adc_val==79 || adc_val==80 || adc_val==81 || adc_val==78)
          { OCR1A=1023;
            OCR1B=0;}  }
////////////////////////////////////////////////////////////////
//3.50
   if(adc_val1==93 || adc_val1==94 || adc_val1==92 || adc_val1==95)
         {  if(adc_val==127)
          {       OCR1A=550;
                  OCR1B=0; }
  if(adc_val==115 || adc_val==116 || adc_val==114 || adc_val==117)
          {OCR1A=650;
           OCR1B=0;}
if(adc_val==106 || adc_val==107 || adc_val==105 || adc_val==108){
OCR1A=740;
 OCR1B=0;}
if(adc_val==93 || adc_val==94 || adc_val==92 || adc_val==95){
                    OCR1A=760;
                          OCR1B=0;}
          if(adc_val==79 || adc_val==80 || adc_val==78 || adc_val==81){
            OCR1A=850;
            OCR1B=0;}
                if(adc_val==67 || adc_val==68 || adc_val==66 || adc_val==69){
                      OCR1A=940;
                        OCR1B=0;}
```

# References:


1. http://www.hackchina.com/en/s/pid.c
2. http://www.hackchina.com/en/r/117651/motorctrl.c__html
3. http://www.hackchina.com/?lang=en&q=microcontroller+based+pid+algorithms+c+code
4. http://www.hackchina.com/en/r/117651/motorctrl.c__html
5. http://www.hackchina.com/en/s/pid.c
6. http://www.hackchina.com/?lang=en&q=microcontroller+based+pid+algorithms+c+code
7. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1183680
8. ftp://ftp.esat.kuleuven.ac.be/sista/doclo/reports/04-71.pdf
9. http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf
10. http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/3/L293D.shtml
11. Andersen and J. Hasen, "Engineers of tomorrow: Knowledge, insight

Denmark, Sept. 2001and skills needed to work across borders," in *SEFI Conf.*, Copenhagen.