North American Academic Research

Journal homepage: http://twasp.info/journal/home

Research

Design of Motion Profile of Stepper Motor for Elevator Using Arduino Uno

Barun Kumar*1, Prof. Wang shi ying*2, Atta-Ur-Rehman*3 , Md Ahmed Sanuar Hossain*4, Khan shahnewaz sohagh*5

[1-4]Department of Mechanical Engineering, Taiyuan University of Technology, China.

[2]Department of E Engineering, Taiyuan University of Technology, China.

*[*]**Corresponding author***

***Accepted**: 20 April, 2020 ; **Online**: 24 April,2020*

***DOI** : https://doi.org/10.5281/zenodo.3764537*

***Abstract** :Elevators are one of the most important inventions in human history, are one of the key players in shaping urbanization of our society and required technological advancements to continue their safe and efficient use. Nowadays, many motion profiles are used in an elevator to improve its efficiency and comfortably of human beings. This paper will present the design and simulation of the motion profile of a stepper motor which comprises the use of the Arduino microcontroller and Arduino code to produce optimize motion profile output and it will use in the elevator system to get its high efficiency and accuracy. A timer interrupts program is written in Arduino IDE for stepper motor in point-to-point motion method for a given time to reach its constant acceleration/deceleration and maximum speed. A graph of this motion profile is plotted in excel from the time output of the serial plotter of Arduino IDE versus the speed that is given in the program of the motor.*

***Keywords**: Elevator, Stepper motor, motion profile, Arduino Uno*

1 INTRODUCTION

An elevator is a vertical-transport device that is very common in the present day. Elevators are used in our daily life to move goods or peoples vertically in high buildings such as shopping centers, working offices, hotels, and many more places. Elevators appear as simple rope or chain hoists. During the middle ages, the elevator operated by animal and human power or by water-driven mechanisms [1]. The elevator, as we know it today, was first developed during the 1800s and relied on hydraulic plungers for lifting capability. Later, the cab was affixed to a hollow plunger that lowered into an underground cylinder. Elevators are known more commonly as lifts in British English and another Commonwealth English. With the introduction of electricity, German Inventor Werner Von Siemens integrated electric motor into the elevator. As the motor mounted at the bottom of the cab, this design required a gearing scheme to climb shaft walls fitted with racks. By 1903, this design had advanced into the gearless traction electric elevator [2]. Electromagnet technology replaced manual rope-driven switching, braking, push-button controls, and various complex signal system. During this age and before microcontroller, the performance of elevators is very poor. The speed control of the motor is done by switching load resistor and timing circuitry [3][4][5]. The landing of the elevator is performed by sliding slowly when the car approaches the door which produces vibration and terrifying acceleration/deceleration, which reduces the comfort of riding an elevator. And also give shocks to mechanical parts of the system and shorten their life.

According to Farahani and Hekmatfer (2009), the science of locating a facility otherwise referred to as a facility location problem has attracted much attention over the last five decades. Although waste bins are kept at strategic places to reduce the exposure of waste but these bins always get filled quickly on timely basis because there is little or no existing technology to display the location of these bins to be monitored and evacuated. As a community's tolerance for dump-sites decreases, they are moved farther from densely populated areas, requiring garbage trucks to drive farther distances to dispose wastes. Also, the complexity of collection routes increases the cost of managing waste collection and disposal. Therefore, any improvement in the monitoring of waste bins and routing system of waste collection and disposal can improve the efficiency of solid waste management.

With the on-going development of technology and economy, High-rise buildings of urban required fast speed, comfortable and efficient transportation system of lift without discomforting human beings and to avoid delay-related damages. For this, an improved and optimized constant acceleration and constant deceleration control are needed for a lift which fulfills the need for the processing speed, efficiency, and accuracy of the elevator system. Currently, there are four kinds of acceleration and deceleration curves in use and they are trapezoid curve, linear curve, exponential curve, and S-curve. The linear curve is applied when there is no need for high acceleration. An exponential curve has weak stability at higher speeds. S-curve has fast responded time and high accuracy at fast speed and maximum acceleration/deceleration but difficult to proceed in microcontroller because the computational power of a microcontroller is limited [6]. Stepper motors and microcontrollers are used in many fields of applications like position control, speed control and measurement applications due to its robust structure, absence of brushes, better control capability, and constant acceleration/deceleration. There are different techniques are used to control the stepper motor such as fuzzy logic, neural network, and microcontroller. But, the microcontroller is one of the easiest techniques for controlling the speed and position of the stepper motor by changing the control program with serial port programming by a personal computer [7][8]. Step motors, which are also called stepping motors/steppers, can be defined as "electromagnetic incremental-motion actuators which convert digital pulse input to analog output motion". Thus, step motors can be viewed as electrical motors that are driven by digital pulses rather than a continuous voltage or current [9]. Step motors translate a train of pulses into shaft revolutions and each pulse equals one rotary increment or one step angle. The advantages of using stepper motors are low cost and provide precise positioning and repeatability of movement [10]. Generally, the stepper driver is required for controlling a stepper motor, which controls the micro-stepping and helps to smooth out the motor's rotation. Arduino Uno is a basic single-board microcontroller and an open-source electronics platform based on easy-to-use hardware and software and also to design, develop and test complex electronics prototypes and products. It is inexpensive, simple and clear programming environment [11][12][13].

The objective of this paper is to present a low cost and optimized motion profile of stepper motor to use in elevator system to improve the speed, efficiency and accuracy of the elevator. This project describes the design and simulation of programming code in Arduino IDE for generating such motion profile. It also records the limitations and possible suggestions for enhancements in future models. The system design consists software and hardware design which described in the rest of this paper.

2 METHODOLOGY

In this project, an optimized motion profile is generated for lift motion and using Arduino Uno microcontroller, a program is written in Arduino IDE for a stepper motor which is controlled by TB6560 driver to produce same motion profile as required motion curve.

Lift kinematic is the study of the motion of a lift car. Jerk is the time rate of change of acceleration/deceleration and its units are m/s3 and deg/s3 for linear and rotary applications respectively. It is used to minimize backlash effects, mechanical wearing, and tear, and provide a smooth and stable profile. The maximum acceleration and jerk of this motion should not be discomforting the limits of human beings i.e. it should be optimum velocity, acceleration and jerk profiles that can be obtained at given human constraints. Generally, there are three different journey conditions for lift: a) the lift reaches full speed, b) the lift reaches full acceleration but not full speed, and c) the lift does not reach full speed and acceleration. For achieving minimum journey time and efficiency of lift, the first condition is very suitable for lift motion profile.

If a is acceleration & v is rated speed then time taken to accelerate for lift from rest to rated speed is given by:

$v = u + at$ ……………….[where u is initial velocity which is zero]

$v = at$

$t = v/a$ ……………………………(i)

Also, the distance travelled during constant acceleration is given as:

$v2 – u2 = 2as$……………..[where u is initial velocity, which is zero]

$v2 = 2as$

$S = (v2 / 2a)$ …………………..(ii)

In practice, the magnitude of acceleration and deceleration are approximately equal. If D is the distance between top and bottom floor then there will be distance:

$D – 2(v2 /2a) = (D – v2 / a)$……….(iii)

to be travelled at rated speed as shown in figure.

The time taken to do this will be given as:

$(D – v2 /a) / v = (D/v) – (v /a)$……(iv)

Therefore, the total time taken for the journey is given by:

$$(D/v) - (v/a) + 2(v/a) = [\, D/v + v/a] \ldots\ldots (v)$$

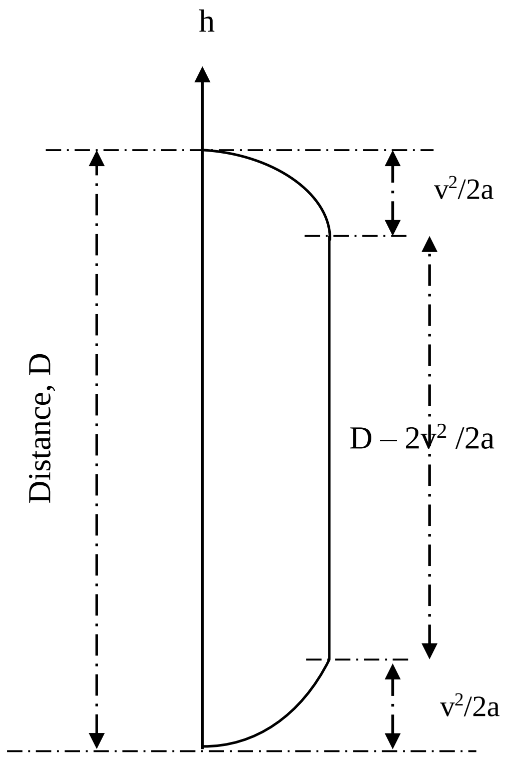


Figure 1 : Lift Journey distance details

S-curve is one of the most important motion profiles that are trying to use in elevator controllers. It increases the comfort during acceleration and deceleration by providing soft starting and stopping motion with gently increasing the acceleration and speed. It stops at zero speed and zero acceleration, which reduces the shock and vibration of the mechanical parts of the lift. This is also known as the 'direct landing' feature of the elevator controllers [14][15]. It is a geometrical method that is based on finding the area under the speed-time curve, equating it to the total distance and then finding the sum of the seven-time parts. It is assumed that the journey is split into seven parts: d1, d2, d3, d4, d5, d6, and d7 as shown in below figure. It is also assumed that the time duration for each part is t1, t2, t3, t4, t5, t6, and t7 respectively. The total distance of the journey is D, the rated speed is v, the rated acceleration is a and the rated jerk is j. The S-curve consists of 7 different phases of motion as shown in figure 2 below. In Phase-I, the acceleration increases linearly until it reaches the maximum acceleration. In phase-II, the profile accelerates at its maximum acceleration rate until it must start decreasing as it approaches the maximum velocity. In phase-III, the acceleration linearly decreases until it reaches zero. In phase-IV, the velocity is constant until deceleration begins. The Phase-V, Phase-VI & Phase-VII are just opposite of Phase-III, Phase-II & Phase-I respectively. The optimum velocity, acceleration & jerk for ideal lift are 2.5 m/s, 1 m/s2, 2 m/s3 respectively.

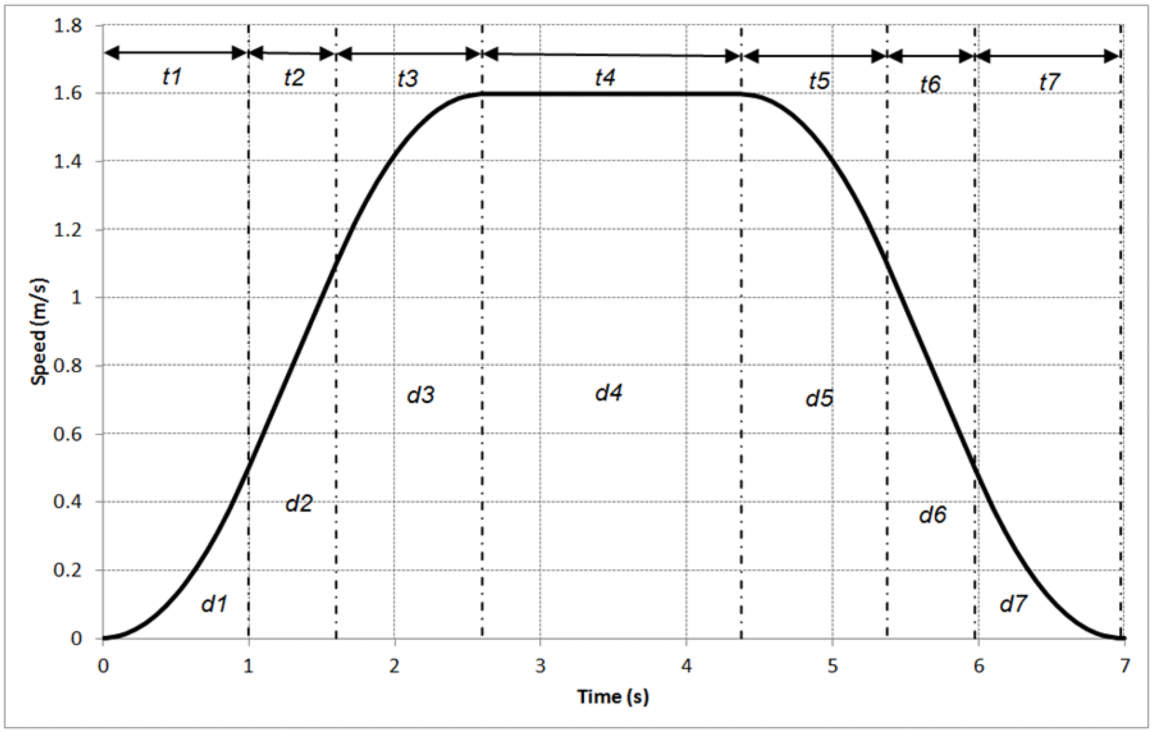


Figure 2: S-curve profile

As the S-profile is symmetry, the following applies:

$t_1 = t_3 = t_5 = t_7$ $t_2 = t_6$

Also, the following distances are equal as:

$d_1 = d_7$ $d_3 = d_5$ $d_2 = d_6$

And the respective time 't' is given after each successive section as:

$t_1 = \frac{a}{j}$ $t_2 = \frac{v}{a}$ $t_3 = \frac{a}{j} + \frac{v}{a}$ $t_4 = \frac{d}{v}$ $t_5 = \frac{d}{v} + \frac{a}{j}$ $t_6 = \frac{d}{v} + \frac{v}{a}$

$t_7 = \frac{d}{v} + \frac{a}{j} + \frac{v}{a}$

The condition for the lift reaches its full speed is given by below equation:

If $d \geq (a^2 * v + v^2 * j) / (j * a)$ then Journey Time = $\frac{d}{v} + \frac{a}{j} + \frac{v}{a}$

Timer/Counter Interrupts allow to perform a task at very specifically timed intervals regardless of what else is going on code. Interrupts are useful for calculating the time between two events, sending out a signal of a specific frequency, periodically checking for incoming serial data and many more. Arduino Uno has three timers: timer0, timer1 & timer2. Timer0 & timer2 are 8-bit timers, meaning they can store a maximum counter value of 255 whereas timer1 is a 16-bit timer, which can store a maximum counter value of 65535.

The Arduino Uno clock runs at 16 MHZ, and each tick of the counter represents 1/16,000,000 of a second i.e. ~63ns. So, for an example, a counter will take 10/16,000,000 seconds to reach a value of 9 (counters are 0 indexed). In many applications, the counter speed of 16MHZ is too fast. The control of speed of the timer counter is done by a Prescaler. The Prescaler can equal to 1, 8, 16, 256 & 1024. It dictates the speed of timer according to the below equation:

Timer Speed (Hz) = [Arduino clock speed(16MHz)] / Prescaler

Interrupt Frequency (Hz) = [Arduino clock speed (16MHz)] / [Prescaler * (compare match register + 1)]

Or, Compare Match Register = [16,000,000 Hz / (Prescaler * desired interrupt frequency)] – 1

If an interrupt of frequency of 1Hz required every second, then:

Compare Match Register value is given as:

= [16,000,000/ (Prescaler * 1)] – 1

= [16,000,000 / (1024*1)] – 1 where, Prescaler of 1024 is selected for Timer1 interrupt

= 15, 624

The most important registers that used with the timers are:

- TCCRX – Timer/Counter Control Register: here the clock source is used to set.

- TCNTX – Timer/Counter Register: here the value of timer's counter is stored.

- OCRX – Output Compare Register: value is set and compared with TCNTx to trigger the events;

- ICRx – Input Capture Register: used to measure time intervals between external events.

- TIMSKx – Timer/Counter Interrupt Mask Register: used to enable/disable timer interrupts.

- TIFRx – Timer/Counter Interrupt Flag Register: to signal the existence of an interrupt request.

3 EXPERIMENTAL SET UP

A. ARDUINO UNO:

Arduino Uno R3 is one kind of ATmega328P based microcontroller board. It includes the whole thing required to hold up the microcontroller; just attach it to a PC with the help of a USB cable, and give the supply using AC-DC adapter or a battery to get started. The term Uno means "one" in the language of "Italian" and was selected for marking the release of Arduino's IDE 1.0 software. The R3 Arduino Uno is the 3rd as well as most recent modification of the Arduino Uno. Arduino board and IDE software are the reference versions of Arduino and currently progressed to new releases. Arduino Uno R3 has ATmega328P based microcontroller, which operating voltage is 5V. The Input voltage is up to 7V-12V and the input/output limit voltage is 6V-20V. It has 14 digital input/output pins, 6 PWM pins, and 6 analog input/output pins. It provides 20mA of DC current for each input/output pin. It used 3.3V DC current pin is 50mA. It has 32KB of flash memory and 0.5KB is used for the boot loader. Its SRAM and EEPROM are 2KB and 1KB respectively. Its inbuilt processing speed of the clock is 16 MHz.

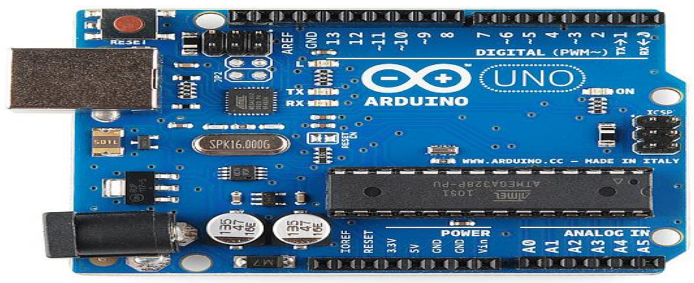


Figure 3 - Arduino Uno

B. Stepper motor:

In this project, 42BYGH47-401A stepper motor (150mm x 100mm x 43mm) is used which is generally used in mechanical motion control, Like CNC milling machine, laser cutter, 3D printer, robot platform, etc. It is 2-phase stepper motor of step angle 1.8° and speed of 1000 rpm. It requires 24V DC power supply of Phase Current of 1.5A/Phase. It has phase resistance of 1.6ohm/Phase and phase inductance of 2.8mH/Phase with holding torque of 5500g.cm. It has 4 wire.

Figure 4: Stepper motor

C. TB6560 Driver:

The TB6560 Stepper Motor Driver is an excellent microstepping driver that use TOSHIBA TB6560 Chip, based on pure-sine current control technology. It has self-adjust current control parameters for different stepper motors which make to run the stepper motors with smaller noise, lower heating, smoother movement and have better performances at higher speed. It is suitable for driving 2-Phase and 4-Phase hybrid stepper motors.

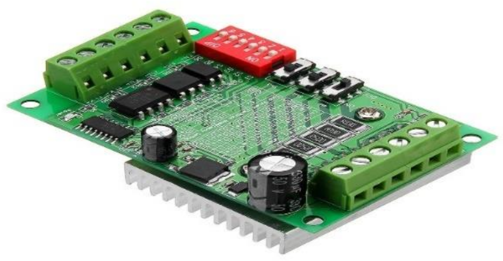


Figure 5: Motor Driver

It has low cost and a High-Speed torque. It has seven output current choices and the maximum of 3200 steps/rev. it works on supply voltage up to +32 VDC and outputs current up to 3.0 Amp. It has pulse frequency up to 20 kHz and over-voltage and short-circuits protection. It is suitable for many stepping motors and can be used in X-Y-Z tables, labeling machines, laser cutters, engraving machines, pick-place devices, and more. It is specially used for low vibration, high-speed and high precision.

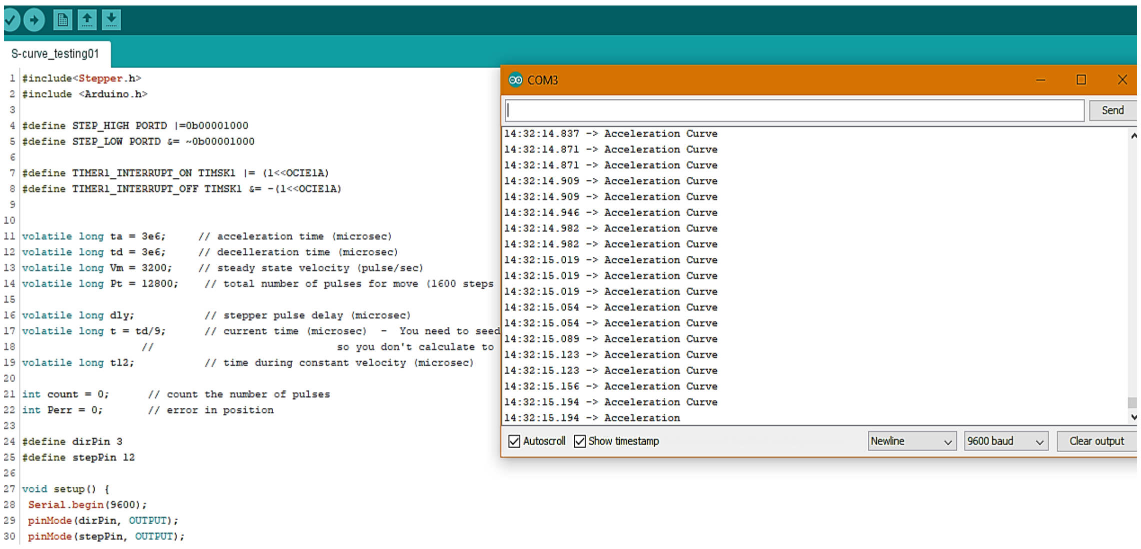
D. Power supply:

In this project, the LRS-35-24 power supply device is used for powering stepper motor through TB6560 driver. It is AC-DC power supply which consists of an electronic circuit with many components such as transformers (used to step the AC Voltage level up or down), rectifiers (used to convert AC Sinewave to pulsating DC wave), filters (used to remove the pulsation from a DC wave and produce a constant current), and regulators (to help maintain the fixed output voltage). It is very cheap and easily available in China.

Figure 6:

E. Arduino Uno IDE

Arduino Integrated Development Environment (IDE) is a software that can be used in Windows, macOS & Linux. The Arduino IDE supports C and C++ language using special rules of code structuring. The Arduino IDE has its own software library and used to write and upload programs to Arduino compatible boards. It is easy to use for a beginner in coding.



S-curve_testing01

```
#include<Stepper.h>
#include <Arduino.h>

#define STEP_HIGH PORTD |=0b00001000
#define STEP_LOW PORTD &= ~0b00001000

#define TIMER1_INTERRUPT_ON TIMSK1 |= (1<<OCIE1A)
#define TIMER1_INTERRUPT_OFF TIMSK1 &= -(1<<OCIE1A)


volatile long ta = 3e6;     // acceleration time (microsec)
volatile long td = 3e6;     // decelleration time (microsec)
volatile long Vm = 3200;    // steady state velocity (pulse/sec)
volatile long Pt = 12800;    // total number of pulses for move (1600 steps

volatile long dly;           // stepper pulse delay (microsec)
volatile long t = td/9;      // current time (microsec)  -  You need to seed
                   //                              so you don't calculate to
volatile long t12;           // time during constant velocity (microsec)

int count = 0;      // count the number of pulses
int Perr = 0;       // error in position

#define dirPin 3
#define stepPin 12

void setup() {
 Serial.begin(9600);
 pinMode(dirPin, OUTPUT);
 pinMode(stepPin, OUTPUT);
```

COM3

```
14:32:14.837 -> Acceleration Curve
14:32:14.871 -> Acceleration Curve
14:32:14.871 -> Acceleration Curve
14:32:14.909 -> Acceleration Curve
14:32:14.909 -> Acceleration Curve
14:32:14.946 -> Acceleration Curve
14:32:14.982 -> Acceleration Curve
14:32:14.982 -> Acceleration Curve
14:32:15.019 -> Acceleration Curve
14:32:15.019 -> Acceleration Curve
14:32:15.019 -> Acceleration Curve
14:32:15.054 -> Acceleration Curve
14:32:15.054 -> Acceleration Curve
14:32:15.089 -> Acceleration Curve
14:32:15.123 -> Acceleration Curve
14:32:15.123 -> Acceleration Curve
14:32:15.156 -> Acceleration Curve
14:32:15.194 -> Acceleration Curve
14:32:15.194 -> Acceleration
```

Figure 7: Program in Arduino Uno IDE

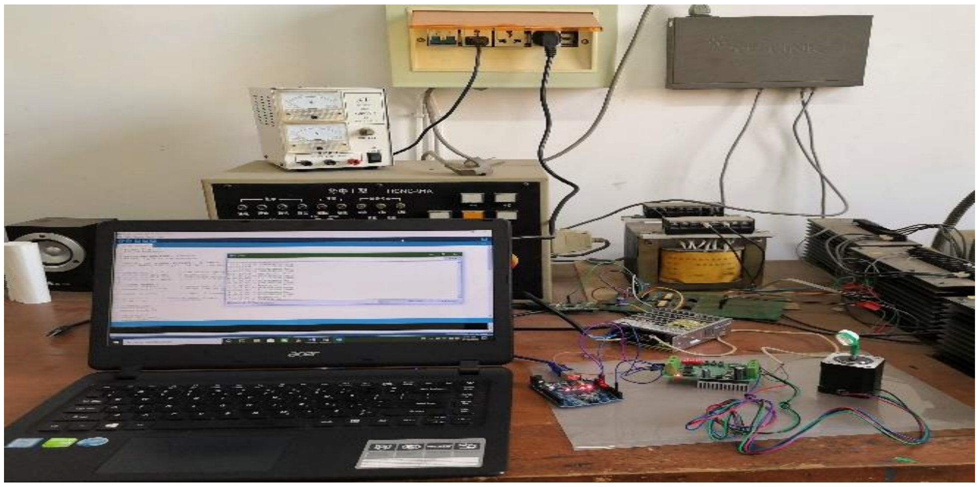


Figure 8: Experimental Setup

III. RESULT

The optimize motion profile (which is a subset of an S-Curve) for Stepper motor is generated using the Arduino IDE program. The out-put data of Arduino Serial Plotter is converted into speed from steps/s for the stepper motor. The output data is graphed in the excel sheet to visualize the profile. It is very useful in the elevator system as the motor had its full speed at a constant velocity curve which improves the efficiency of the elevator. The constant acceleration/deceleration curve is also fond very smooth with the timer interrupt program of Arduino. It also helps to reduce the jerk of the elevator system by reducing residual vibration. So, the microcontroller-based control system can be used to any elevator system to improving its precise, comfortable and efficient lift transportation system.

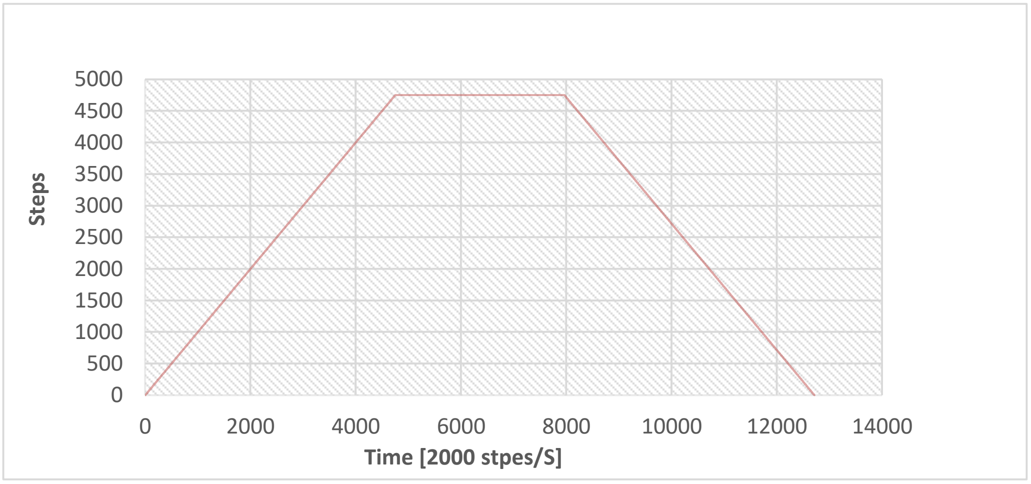


Figure 9: The Output Velocity Profile of project

IV. DISCUSSION

However, there are certain disadvantages of the microcontroller-based control system for the stepper motor. The speed of the stepper motor used to be low at the micro-stepping stage of motor because, at the higher speed of the motor, torque is low which may bring problems for the elevator system. To address this instability, a non-zero starting velocity is used. This takes the profile to instantly jump to a programmed velocity upon initial acceleration, and while deceleration. But, it produces the jerk at the starting point of the system. S-curve is third-order curve and have seven phase which is more complicated to calculate the stopping distance for a given set of profile values. It is very much difficult to calculate how long, and over what distance, the motion control profile previously took to accelerate to determine when to start decelerating.

V. CONCLUSION

In this paper the proposed motion profile of the stepper motor is successfully design with the Arduino IDE program and the output of this program is very applicable to the velocity profile for the elevator system. Initially, the parameters of the program are taken from Stepper motor data-sheet and converted it into a delay between acceleration/deceleration curve & constant velocity instruction to produce the required velocity of the stepper motor. Here Arduino acts as a hardware tool for controlling the motor motion profile. The out-put data of the serial plotter of Arduino IDE is converted into speed from step/S and plotted it in the Excel sheet.

VI. REFERENCES

K. Pradhasaradhi, K. A. Babu, and K. O. Prakash, "IMPLEMENTATION OF EMBEDDED ELEVATOR CONTROL SYSTEM USING BLUETOOTH MODULE," vol. II, no. Ii, pp. 216–221, 2016.

H. Singh, "Microcontroller Based Stepper Motor Drive for an Elevator System," vol. 1, no. 8, pp. 36–39, 2011.

R. Singh, P. Maurya, D. P. Patel, and P. Ranjan, "Microcontroller Based Elevator System," vol. VI, no. 1, pp. 160–162, 2017.

M. A. Balug, T. Spoljaric, and G. Vujisic, "Laboratory model of the elevator controlled by ARDUINO platform," 2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc., pp. 1562–1565, 2017.

A. Arun, J. Pratik, and S. Shinde, "Elevator Control System (Microcontroller Based)," vol. 3, no. 03, pp. 1510–1515, 2015.

M. Zeng, C.-Z. Hu, and P.-F. Hu, "Control Algorithm of Acceleration Curve for Stepper Motor," J. Control Syst. Eng., vol. 4, no. 1, pp. 32–39, 2016.

S. J. Parmar, "Design and Development of Stepper Motor Position Control using Arduino Mega 2560," vol. 3, no. 09, pp. 77–82, 2017.

D. O. Carrica, S. A. González, and M. Benedetti, "A high speed velocity control algorithm of multiple stepper motors," Mechatronics, vol. 14, no. 6, pp. 675–684, 2004.

K. Takashi, Stepping Motors - Takaeshi Kenjo.Pdf. 1984.

P. K. A. S. J P Gajipara, "Stepper Motor Drive for High Speed Control By High voltage & Constant current," Int. J. Adv. Eng. Res. Stud., pp. 1–3.

R. R. A, S. P, and G. R, “Modelling and Control of Hybrid Stepper Motor using LabVIEW,” Ijireeice, vol. 7, no. 5, pp. 49–55, 2019.
Q. F. -Qun, J. Xue-Dong, and Z. Shi-qing, “Design of stepping motor control system based on AT89C51 microcontroller,” Procedia Eng., 2011.
X. Wei, “Acceleration and deceleration control design of step motor based on TMS320F240,” Procedia Eng., vol. 15, pp. 501–504, 2011.
A. Jalali Yazdi, P. Forsythe, A. Ahmadian Fard Fini, and M. Maghrebi, “Optimization of flexible lift processes on high-rise building construction sites,” Autom. Constr., vol. 107, no. August, 2019.
P. J. Siripala and Y. A. Sekercioglu, “A generalised solution for generating stepper motor speed profiles in real time,” Mechatronics, vol. 23, no. 5, pp. 541–547, 2013.

Dedication

Not mentioned.

Conflicts of Interest

There are no conflicts to declare.