CHAPTER 4
CONTROL THEORY

4.1 Overview

Pneuman's mechanical structure has 25 degrees of freedom (DOF). Each DOF is actuated by a direct current (DC) motor and has a sensor for feedback. The embedded computer analyzes the information from the sensor and controls the corresponding DC motor to achieve the desired output. Each DOF, with its own DC motor and sensor, constitutes a closed loop system. There are two different types of sensors used on *Pneuman*; potentiometers provide absolute joint position for 19 of the 25 DOF and incremental optical encoders are used on the drive wheels and stero head. The details regarding the use of each sensor will be discussed in the following sections.

Although a control system provides a way to achieve a desired output, the ways to determine what the desired output should be are also considered. For example, if a particular joint is positioned at 0 degrees and the desired position is 90 degrees, how should the joint move from the initial to the final position? Do you simply command the controller to position the joint at 90 degrees as fast as possible? Will that cause too much mechanical strain on the joint? What if you wanted it to move “smoothly” over a period of 5 seconds? These issues will also be discussed in detail.

4.2 Control of Direct Current (DC) Motors

Essentially, a DC motor consists of a stator, a rotor, and a commutator. The stator is the housing of the motor and contains magnets, bearings, etc. The rotor is the rotating part of the motor and contains a coil of wire through which current flows. The coil of wire in the rotor connects to the commutator and receives current through brushes. The commutator insures that the current flows in the proper direction while the rotor turns [26].

In understanding this topic, it is important to have an understanding of the operation and mathematical model of a DC motor. When current flows through the coil of wire in the rotor, a torque is created that causes the rotor to spin. The relationship between the motor output torque and the current is given by

$$\tau_m = k_m i_a \tag{4-1}$$

where τ_m is the output torque, k_m is the torque constant, and i_a is the rotor coil current. Consequently, the amount of torque generated is proportional to the current flowing through the wire. However, there is a limit to the amount of torque a given motor can produce. The coil of wire in the rotor is an inductor, and the voltage across the inductor is

$$v = L \cdot \frac{di}{dt} \tag{4-2}$$

where v is the voltage across the coil, L is the inductance of the coil, and $\frac{di}{dt}$ is the changing current across the inductor. This coil-induced voltage opposes the voltage that is applied to the motor, causing a *decrease* in current through the rotor. This is called the *back-emf voltage* and the negative feedback eventually causes the motor to settle at a steady state point of operation [26].

Changing the voltage across the motor terminals will vary the current flowing through the coil thereby changing the torque produced by the motor. However, this technique is not used to control *Pneuman's* motors. Instead, a constant voltage is pulsed through the motor coil. This pulsing, or pulse width modulation (PWM), changes the *average* current through the motor over time. The average current is proportional to the *duty cycle* of the PWM signal. The *duty cycle* is determined by

$$\% \text{ duty cycle } = \frac{\text{high time}}{\text{period}} \cdot 100\% . \quad (4\text{-}3)$$

See Figure 4-1 [26].

‘

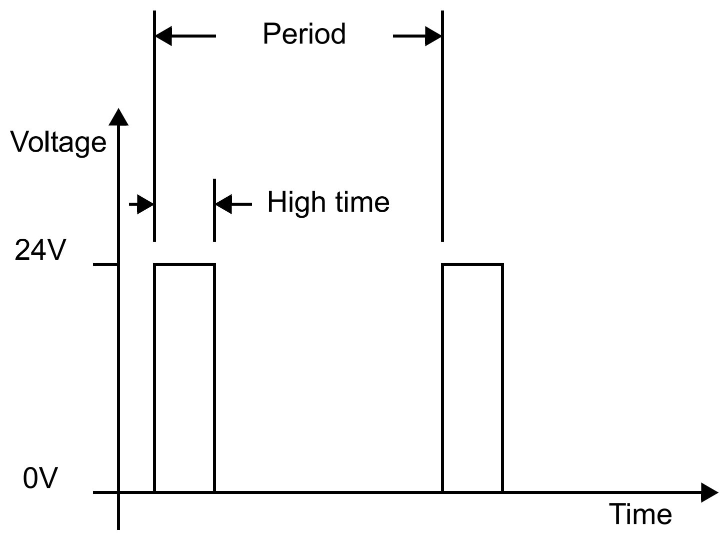

Figure 4-1: Pulse width modulation

4.3 Control System Implementation

4.3.1 Analog Feedback Control

Nineteen of *Pneuman's* joints use analog potentiometers for feedback. They operate as absolute position encoders, providing a voltage reference indicating the joint angle. This voltage signal is fed into an analog to digital converter, providing eight-bits of resolution over the potentiometer's operational range of 300 degrees. Therefore each bit corresponds to 1.17 degrees of movement, which is acceptable for *Pneuman's* designated purpose as an experimental research platform. See Figure 4-2.

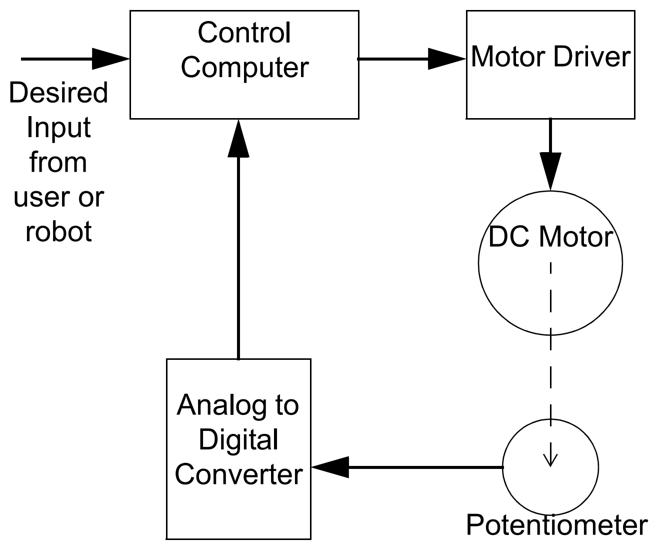

Figure 4-2: Block diagram of analog control system.

All of the joints utilizing a potentiometer use a discrete approximation of the proportional, derivative, and integral (PID) control law, with gravity compensation (except for the steering mechanisms), implemented in software. This robust control law was selected due to its simplicity and good performance. The discrete PID controller is implemented with the following equation:

$$\mu(n) = k_P \cdot e(n) + k_I \sum_{N=0}^{n} e(n) + k_D[e(n) - e(n-1)], \qquad (4\text{-}4)$$

where $\mu(n)$ is the motor control signal output, updated at the sampling time n, k_P is the proportional gain, k_I is the integral gain, k_D is the derivative gain, and $e(n)$ is the position error at the sample time n. All of the joints have the same sampling rate of 100 Hz, and all of the gains are individually tuned for maximum performance [28], [29].

The potentiometers used as joint angle sensors may have nonlinear characteristics. For example, the potentiometer may physically rotate 90 degrees, but due to the nonlinear characteristics the analog value does not indicate a change of 90 degrees. See figure 4-3.

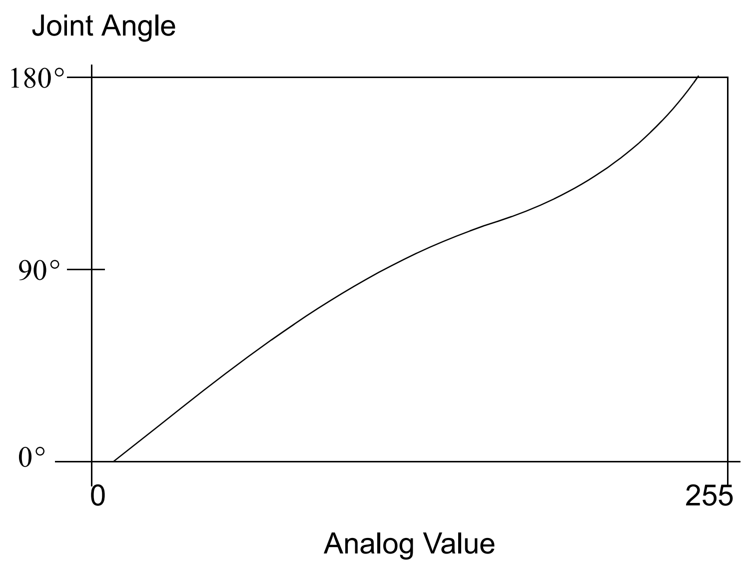

Figure 4-3: Output of uncalibrated potentiometer

Therefore, all of the joints must be calibrated to get the most accurate measurements possible. Ideally, a large data set collected over the complete range of motion should be collected and used for an accurate calibration. However, collecting data over the complete range of motion for each DOF is not feasible due to difficulties in obtaining accurate position measurements without sophisticated tools. For this reason, three data points are collected and used to calibrate each joint.

The three data points form two lines; the slopes and y-axis intercepts of each line are the calibration parameters for each DOF. The slopes are determined from the following equations:

$$\text{slope}_A = \frac{\Delta\text{Joint Position}}{\Delta\text{Analog Value}} = \frac{\text{Position 1} - \text{Position 0}}{\text{Analog 1} - \text{Analog 0}} \tag{4-5}$$

$$\text{slope}_B = \frac{\Delta\text{Joint Position}}{\Delta\text{Analog Value}} = \frac{\text{Position 2} - \text{Position 1}}{\text{Analog 2} - \text{Analog 1}} \tag{4-6}$$

and the intercepts from

$$\text{Intercept A} = \text{Position 1} - (\text{slope}_A \cdot \text{Analog 1}) \tag{4-7}$$

$$\text{Intercept B} = \text{Position 1} - (\text{slope}_B \cdot \text{Analog 1}) \tag{4-8}$$

For example, each drive wheel is calibrated at -90, 0, and +90. The corresponding analog values are recorded and used to calibrate the joint. The calculated calibration lines are then used to interpolate joint position between the calibration points. See Figure 4-4.

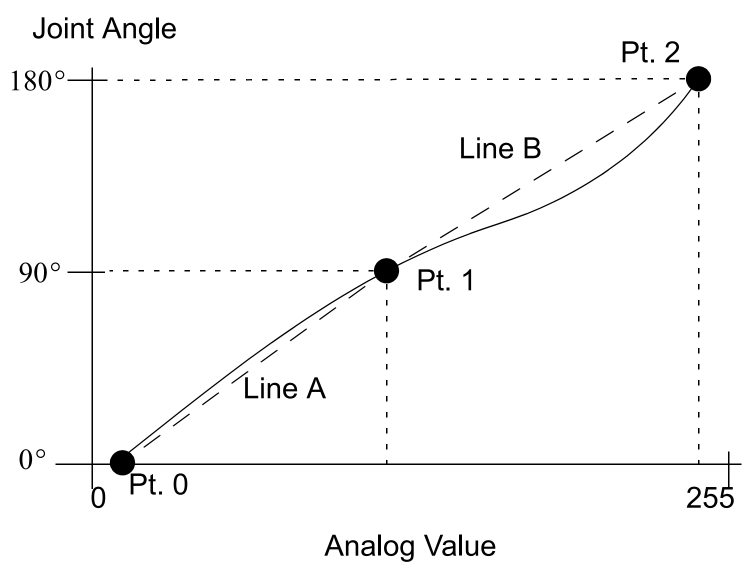

Figure 4-4: Calibrated potentiometer plot

4.3.2 Digital Feedback Control

Pneuman's drive wheels and stereo head actuators each use incremental optical encoders for feedback. These non-contact sensors permit a full 360 degrees of rotation, a requirement for the drive wheels. The wheel encoders have a resolution of 0.18, allowing for precision distance measurement. The stereo head convergence optical encoders have a resolution of 0.036, which is needed for precision stereo vision. Each of the encoders connects to a National Semiconductor LM629 motion control integrated circuit (IC). This specialty-purpose controller interfaces directly to an optical encoder and outputs a signed-magnitude PWM signal for motor control. See Figure 4-5.

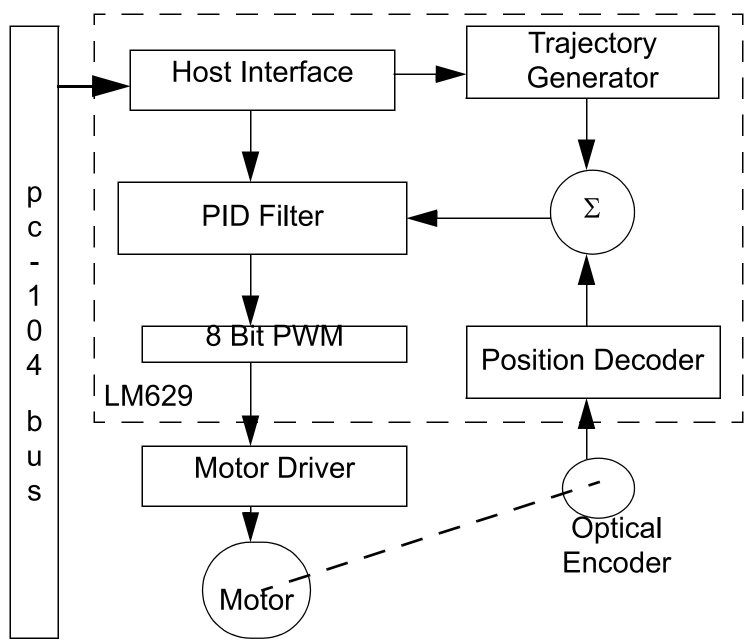

Figure 4-5: LM629 PID controller block diagram.

The LM629 is a specialty purpose micro controller which interface directly to a quadrature optical encoder for feedback. *Pneuman’s* main computer issues commands to the LM629 via the pc/104 bus, and the IC generates the desired motion. The PID filter is given by the equation

$$\mu(n) = k_P \cdot e(n) + k_I \sum_{N=0}^{n} e(n) + k_D[e(n') - e(n'-1)] \qquad (4\text{-}9)$$

where $\mu(n)$ is the motor control signal output, updated at the sampling time n, $e(n)$ is the position error at the sample time n, n' indicates the derivative sampling rate, k_P is the proportional gain, k_I is the integral gain, and k_D is the derivative gain [28].

The proportional term contributes a restoring force proportional to the positional error. The integral term provides a restoring force that is summed over time, insuring that the static error is zero. Therefore, even if there is a constant load on the motor, zero error will still be achieved. The final derivative term provides a damping force, which is proportional to the rate of error change [28].

4.4 Joint Trajectory Generation

4.4.1 Software Trajectory Generation

The desired overall motions of a manipulator may be considered a multidimensional trajectory, which is a history of position, velocity and acceleration versus time. While a qualitative description of a trajectory appears trivial (i.e., make the end-effector go from point A to point B), a quantitative description is more difficult. Questions such as, “How fast should the manipulator move?” and, “What if there is an obstacle in the way?” need to be addressed. Even though a quantitative description is not trivial to compute, an end user of a robotic system should not have to deal with all details of the desired motions. Instead, a goal position and orientation may be given and the control system calculates the best way to get there.

There are a number of ways to move a robot from point A to point B, but they all share a common attribute; they allow the robot to move “smoothly.” A motion may be considered smooth if it is continuous and differentiable. This type of motion decreases wear on the mechanics, reduces vibrations, and generally improves the performance of a manipulator [29].

Calculating a smooth trajectory requires that some constraints be placed on the paths between the points along a trajectory. These constraints guarantee a smooth path will be executed, and they must meet the following conditions:

$$\theta(0) = \theta_0, \tag{4-10}$$

$$\theta(t_f) = \theta_f, \tag{4-11}$$

which are the initial and final joint position values at the initial and final times, respectively. Two other constraints are given by:

$$\dot{\theta}(0) = 0, \tag{4-12}$$

$$\dot{\theta}(t_f) = 0, \tag{4-13}$$

indicating that the initial and final velocities are zero.

These four constraints necessitate a function with four coefficients, a cubic polynomial. A cubic polynomial has the following form:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \tag{4-14}$$

with velocity and acceleration given by:

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2, \tag{4-15}$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t. \tag{4-16}$$

By taking the previous constraints and combing them with the cubic and the derivatives we get a system of four equations and four unknowns; therefore we can solve for the cubic polynomial coefficients:

$$a_0 = \theta_0, \tag{4-17}$$

$$a_1 = 0, \tag{4-18}$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0), \tag{4-19}$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \tag{4-20}$$

where θ_0 is the initial position, θ_f is the final position, and t_f is the amount of time allotted to complete the trajectory [29].

The trajectories of *Pneuman's* drive wheels are determined using this method. This simple trajectory generation scheme was chosen because the steering assembly does not require additional constraints on the velocities and accelerations. The amount of time required to execute any given trajectory is determined by taking the ratio of the desired movement over the overall range of motion and multiplying by the time allowed for the full range of motion:

$$t_{allotted} = \frac{\Delta\theta_{given}}{\theta_{total}} \cdot t_{total} \tag{4-21}$$

with θ_{total} and t_{total} varying for the different joints. The steering joints all use 180 degrees and 3 seconds, respectively.

For example, a steering trajectory executed with a starting position of -45 degrees and an ending position of 30 degrees will have the following parameters:

$$t_{allotted} = \frac{\Delta\theta_{given}}{\theta_{total}} \cdot t_{total} = \frac{75°}{180°} \cdot 3s = 1.25s \tag{4-22}$$

$$a_0 = \theta_0 = -45° \tag{4-23}$$

$$a_1 = 0 \tag{4-24}$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) = 144 \tag{4-25}$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) = -76.8 \tag{4-26}$$

and the cubic is

$$\theta(t) = -45t + 144t^2 - 76.8t^3 \tag{4-27}$$

and the plot of the above trajectory is shown in Figure 4-6.

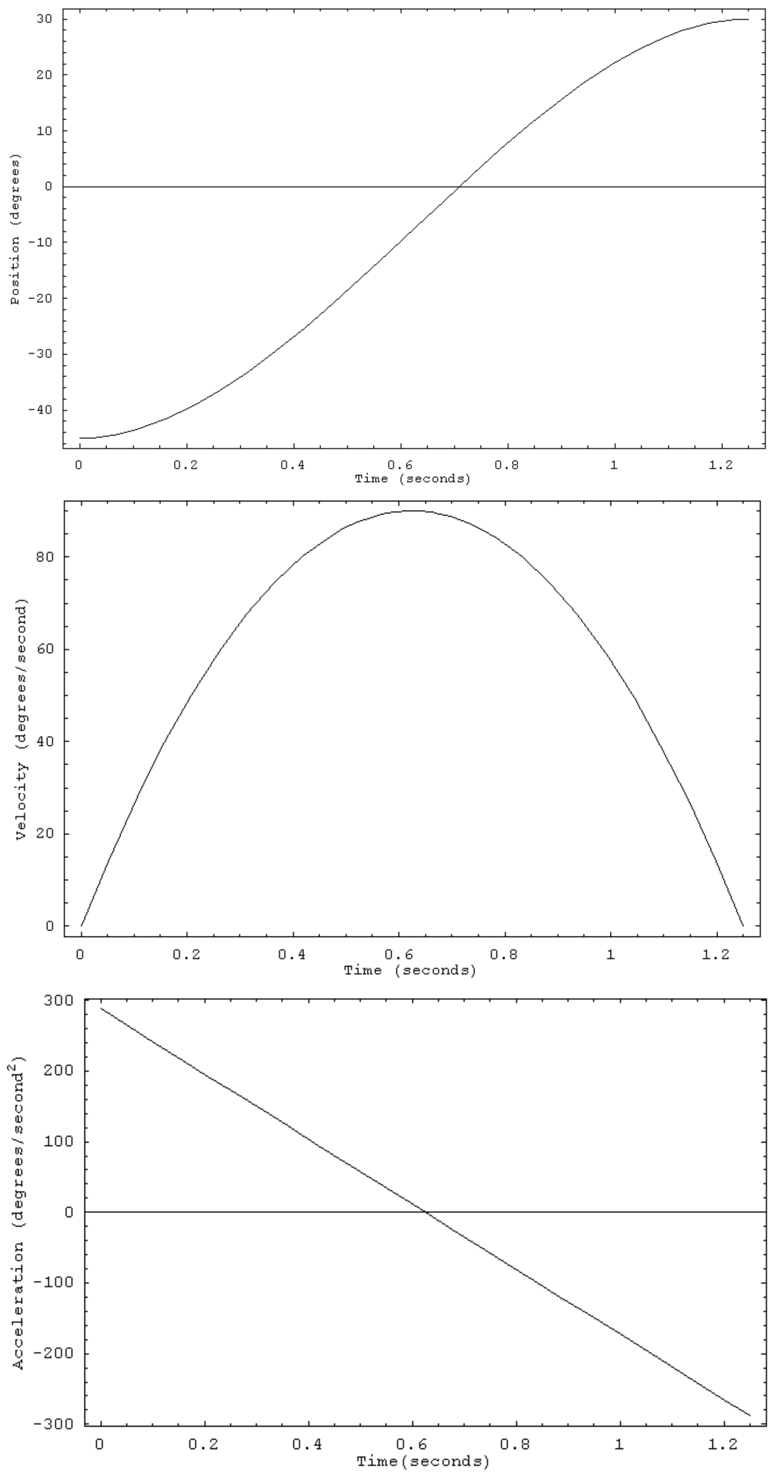

Figure 4-6: Plot of Desired Cubic Trajectory from -45 degrees to 30 degrees in 1.25 seconds without via points

The previously described method may be applied if the starting and ending velocities are zero. However, if intermediate via points are needed where the velocities are not zero, the cubic coefficients are determined by:

$$a_0 = \theta_0 , \tag{4-28}$$

$$a_1 = \dot{\theta}_0 , \tag{4-29}$$

$$a_3 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f , \tag{4-30}$$

$$a_4 = -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0) , \tag{4-31}$$

where θ_0 is the starting position, $\dot{\theta}_0$ is the starting velocity, θ_f is the final position, and $\dot{\theta}_f$ is the final velocity of the segment. Although the steering and drive assemblies do not use this technique, the rest of *Pneuman's* joints benefit from the ability to use via points. See Figure 4-7 for a cubic trajectory with via points; the first segment from -45 degrees to 30 degrees occurs in 1.5 seconds with a via point velocity of 20 degrees/second, and the second segment from 30 to 75 degrees occurs in 3 seconds with and ending velocity of 0 degrees/second [29].

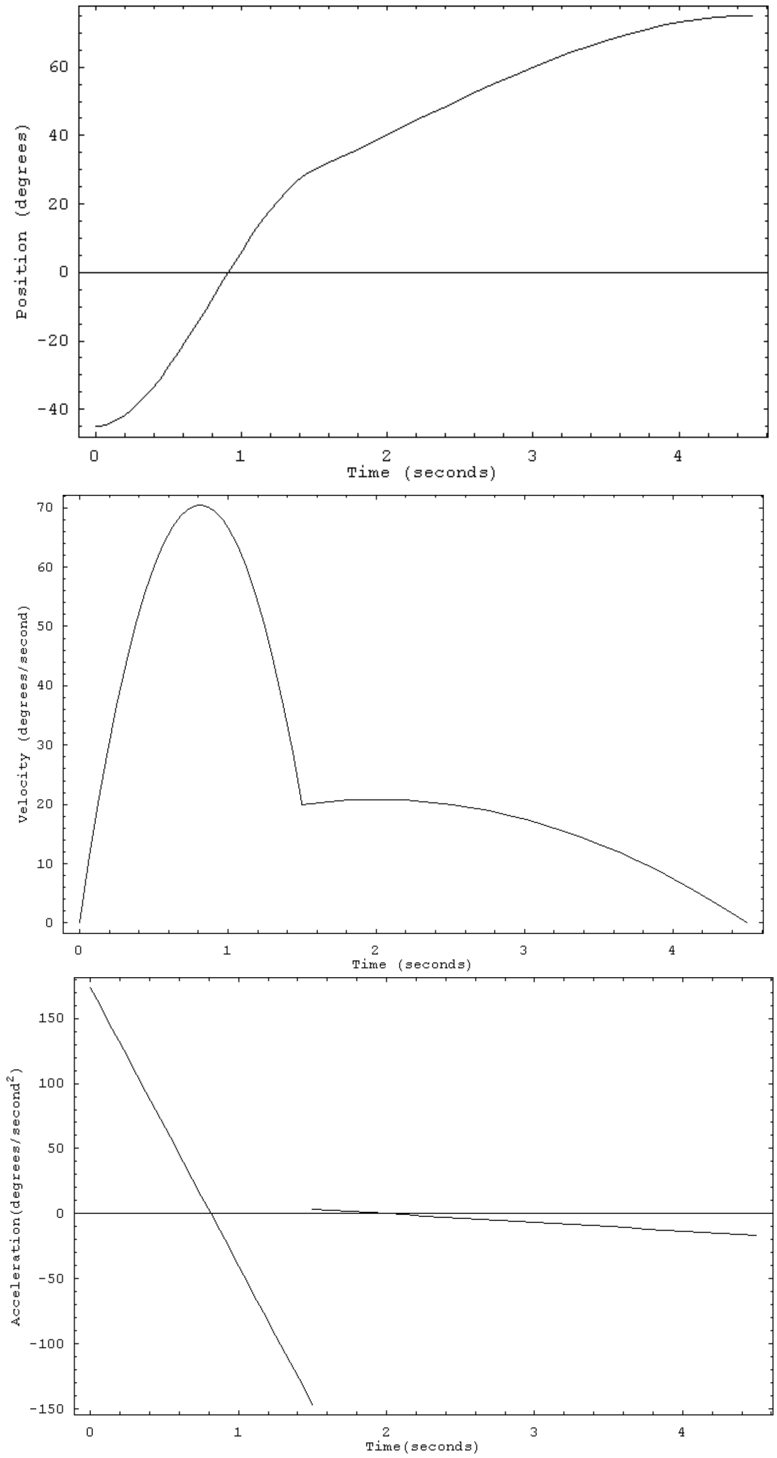

Figure 4-7: Trajectory with via points.

4.4.2 Hardware Trajectory Generation

The LM629 motion control IC does not use cubic trajectory generation. Instead, an alternate generation scheme with a trapezoidal profile is used. In positional control mode, the profile is generated by specifying the desired values of acceleration, maximum allowable velocity, and desired final position. The motion controller uses this information to affect the move by accelerating continuously until the maximum velocity is reached or deceleration must begin to stop at the desired final position. During the move, the values of maximum velocity and desired stopping position may be changed to alter the trajectory [28].

In velocity-only control mode, the values of acceleration and maximum velocity are used to generate the trajectory. The LM629 causes the motor to accelerate to the specified velocity until the maximum allowable velocity is reached, and maintains this velocity until commanded to stop. Again, the deceleration rate is equal to the acceleration rate. See Figure 4-8 for typical trajectories [28].

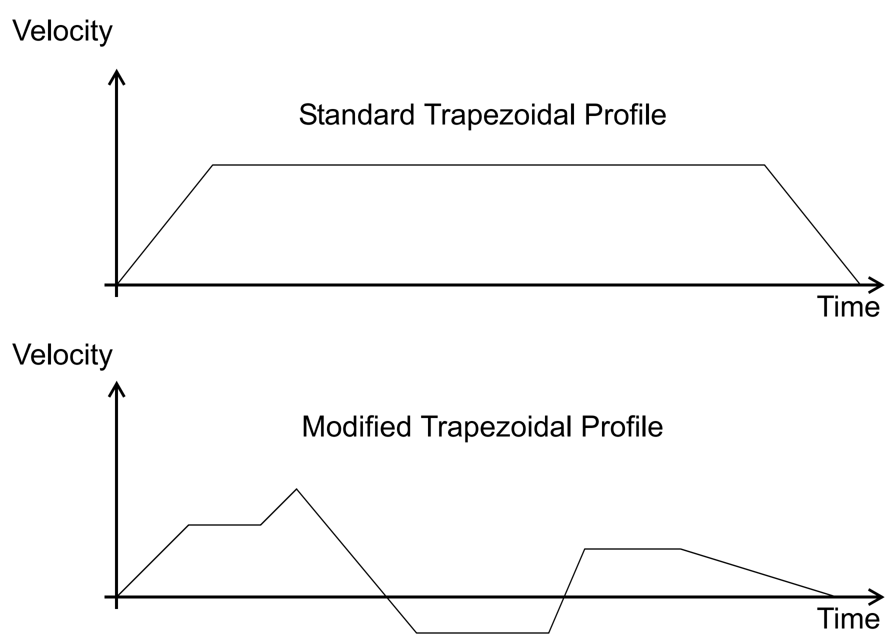

Figure 4-8: Typical LM629 Motion Trajectories