Manipulating Program Equations

Setting the Stepper Motor Rotation Speeds

The NEMA 23 stepper motor functions based on a sequence of electrical pulses that influence a magnetic field which induces precise single step movements that rotates a shaft which in turn is mechanically coupled to the rotarod apparatus. The rate of rotation of the stepper motor shaft and rotarod is determined by the delay time between electrical pulses A series of calculations for the delay times, based on the desired rotarod rotation speed, are performed by the Arduino MEGA program as shown below.

Delay Time Calculation for Constant Rotation Speed

Our stepper motor completes one full rotation after receiving 800 pulses. To determine the amount of time between pulses, we recall that rotation speed $\omega = \Delta\theta/\Delta t$. In our case, we solve for Δt, the period of the pulses needed for our desired rotation speed. We further recognize that in one period our stepper will move one step or $\Delta\theta = 1/800$ of a rotation. It is also important to note that the period of our pulses will be twice our desired delay time since each pulse is composed of half of a period in the low state followed by half or a period in the high state. Since our rotation speeds are given in revolutions per minute or RPM, our calculation must also convert from minutes to seconds. The final result is:

$$delayT = \frac{60}{1600 * RPM}$$

Where *delayT* is the computed delay between pulse state transitions, and *RPM* is our desired stepper motor rotation speed in RPM.

Notes On Radius

Our design utilizes a 3GT timing pulley mounted on the axle of our stepper motor and a custom 3D printed pulley mounted on the rotarod axle. These two pulleys are then connected a non-slipping belt composed of a rubber hose. Due to the difference in radius sizes between the manufactured timing pulley and 3D printed pulley, a gearing ratio is necessary. This was done through the following equation:

$$RPM_{Stepper} = RPM_{Rotarod}*(7.39/4.00)$$

Delay Time Calculation for Variable Rotation Speed

To calculate the equation for ramping (increasing/decreasing) the rotation speed of the stepper motor, two equations were used. One is the delay time equations as seen above and the other is to determine what RPM is desired in a given moment. The equations are listed below:

$$delayT = \frac{60}{1600 * RPM}$$

$$RPM = RPMi + \left(\frac{RPMf - RPMi}{Total\ Ramping\ Time}\right) * ElapsedTime$$

Derivation of RPM equation

To demonstrate how the second equation was derived, take a case in which we desired the stepper motor increase from 4 to 60 RPM in 300 seconds. The RPM at a given time in the ramping sequence from 4 to 60 RPM in 300 seconds can be visualized on the graph in Figure N.

The equation of the graph can be used to determine the RPM at a given time:

$$RPMx = RPMi + \left(\frac{RPMf - RPMi}{Total\ Ramping\ Time}\right) * ElapsedTime$$

Thus, we are left with two equations:

$$delayT = \frac{60}{1600 * RPM}$$

$$RPM = RPMi + \left(\frac{RPMf - RPMi}{Total\ Ramping\ Time}\right) * ElapsedTime$$

Combining these equations can create a formula that can be embedded into the code so that the program will change the delay time autonomously based on the elapsed time:

$$delayT = \frac{60}{1600(RPM + \left(\frac{RPMf - RPMi}{Total\ Ramping\ Time}\right) * elapsedTime)}$$

Writing Sequential Programs

Because of ease of programming and ability to manipulate its capabilities, we found that the device is capable of developing novel modes such as a sequence of fluctuating speeds and rotational directions. In order to program this, we used while statements that constrained its activity based on the elapsed time. Thus, within mode 6, a sequential program from an increase in forward rotating speed followed by a decrease and a subsequent reversal in direction was coded. To develop your own sequential program, make sure to include a time condition in the while loop to execute different speeds at different times.