TECH TIPS
Helpful insights for designing with motion control from
Chuck Raskin, PE, CMCS
27 September 1992

#1: Getting Systems To Respond to Small DAC Changes

A reader asked why the system in Figure 1 doesn't respond to single DAC step changes. The required rotary load base speed is 75 +/-0.25 Rpm, DAC resolution is 16 bit, the motor top speed is 2000 Rpm, and the system was also required to make 45-degree move increments in 20 milli-seconds. The motor package was operating in the current mode.

The motor had enough power to perform the required operations, but the user wasn't taking into account some subtle things about the motion control package (motion control CPU, motor amplifier and motor).

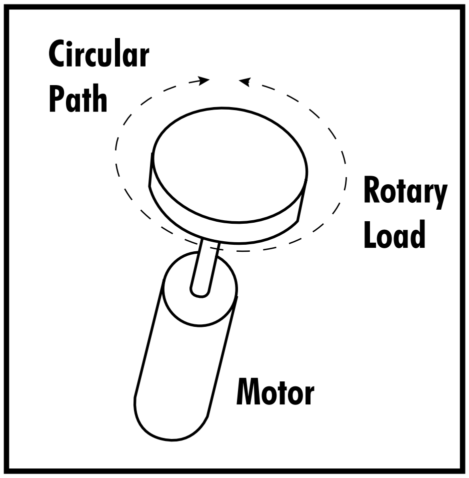

***Figure 1:** Rotary load system fails to respond to small DAC changes*

The first problem was the speed regulation of the motor and amplifier package as a stand-alone item. The ability of the amplifier to maintain the motor's rotation about the requested speed setpoint without excessive oscillation (see Figure 2) was critical to this operation. Operation of the motor package in the current mode was not going to yield the best stability for the high speed indexing due to system friction, etc. Therefore, we reccommended placing the unit into the voltage mode. Here's how this would work.

If we were to apply a fixed DC voltage such as a battery to the motor amplifier (no motion control CPU), and then monitor the motor's speed with a digital strobotac (1 Rpm resolution), any variation occurring in the motor's speed would be a direct result of the motor and motor amplifier's combined stability or regulation. By knowing the speed regulation of the driving package, we can determine if the system will fit the required stability criteria. In this case, the user was operating the system in the voltage mode with no tachometer, which meant that the system stability was over 1percent (+/-10 Rpm).

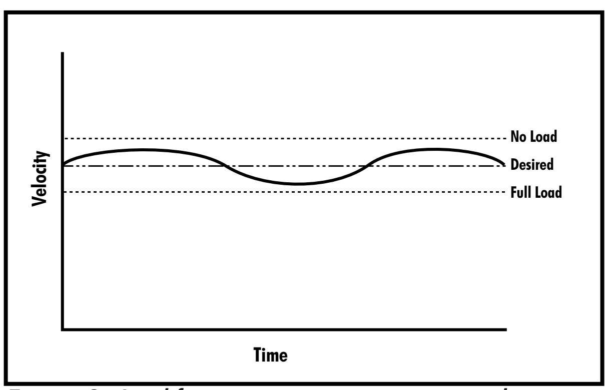

***Figure 2:** Amplifier maintains motor rotation without excessive oscillation*

Consideration of the motor and its amplifier's speed regulation will determine how reactive the gain structure of the system must be to maintain smooth motion. Since the motor is to operate at 3.75n percent of its maximum speed, I recommend using either a gearbox to raise the motor's operating Rpm; a lower speed motor winding to reduce the motor's top speed and therefore, its instability; a motor which was designed specifically for low Rpm use; or a velocity voltage feedback loop, such as a tachometer, to increase motor stability at low Rpms.

The second problem was the use of a 16-bit DAC. There's a tendency to think that the higher the DAC resolution, the more stable—or controllable—the system becomes. The ability to control, however, is not a function solely of the DAC resolution, but also of the motor amplifier's sensitivity or "gain" to the voltage applied at its signal input pins.

There are 16,384 steps on each side of the 16 bit DAC range (+/- 10 VDC). Since this system is using a voltage drive, each DAC step results in an attempt to change the motor's speed by 0.122 Rpm (l/O vdc = 2000 Rpm). If the gain of the motor amplifier package is 1 millivolt, then the 15 bit DAC controller will cause a change in the motor's rpm every 1.6384 DAC steps (see Figure 3).

Discussion:

The speed-regulation and sensitivity of the motor amplifier package dictate how the motion control computer will need to handle the system. For continuous motor velocities that operate in the bottom 10 percent range of the rated Rpm, I recommend the use of a motor designed for low-speed operation. If that is not an alternative because of cost, size or availability, then use some form of gearing to boost the motor's Rpm into a more suitable operating region. Consideration should then be given to the mode of operation of the motor package, voltage or current. Since this system requires high-speed indexing, the voltage mode would be preferential, or one should use a techometer velocity loop.

Technology 80 Inc. • 658 Mendelssohn Avenue North • Minneapolis, MN 55427 USA
612/542-9545 • 612/542-9785 (fax) • www.tech80.com • info@tech80.com

TECH TIPS

#1: Getting Systems To Respond to Small DAC Changes

The third problem facing the high-speed, small-move indexing requirement is that the PID gain structure used requires error to generate DAC voltage. (Error is noted as the difference between where the computer calculates it should be, and where the encoder counter indicates it actually is.) In order to make the required move of 45 degrees, the trajectory generator will be required to move 250 counts (500 line encoder x 4 = 2000 counts per revolution / 8). The National LM628 chip will generate a DAC output based on the update formulas shown in Example. A relationship exists between smallest move size and the system resolution.

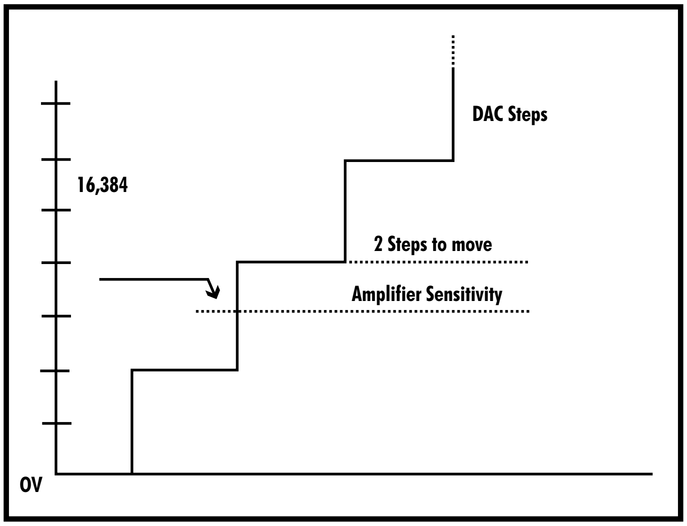

Figure 3: *Increasing DAC resolution and amplifier sensitivity to volatage applied at the signal input can produce stable systems*

Knowing how the gain structure works, we'll use the Kd parameter as the prime factor in handling the move. Observing the CPU's control signal with an oscilloscope, apply a 10 Vdc step response to the motor amplifier package. Record the time it takes to go from rest to the system's rated velocity. Set the derivative sample time between 5 and 10 percent of the response time noted in the step response test just accomplished. Finally, adjust the Kd PID term until the system indexes stably. Use the maximum velocity, and acceleration limits to ensure the move performs within the required profile time (20ms). The key is to not exceed the required time profile by more than 5 percent since inertial loading goes up by square functions of the allotted time. In other words, small changes in time can cause large changes in torque loading, which can cause unstable reactions and require larger motor torques. Also, the Kd term will be used in a gain lead fashion allowing the motion CPU to "kick" the indexing system into place. Ki will only be used to "tweek" the system into final position in very small doses, and Kp will probably not be needed for the small moves.

The fourth and final obstacle is to ensure that the system update time is capable of handling the system speed. In our system, the update period is 256 usecs., which will result in 78 update periods from the start to the end of the move (20/.256). If the update period were 1 msec., there would only be 20 update calculations performed in the move. The number of updates and the system resolution must be properly coordinated with a given gain structure in order to ensure move stability. The more updates performed in a given move profile, the more stable the moves can be performed, providing that the resolution has been properly determined for the move size, the acceleration rate, and the top velocity requirement. ■

About the Author:

In his more than two decades in the industry, **Chuck Raskin, PE, CMCS**, has contributed to many industry publications, including ***Machine Design, Motion Control,*** & ***PCIM***, and is the author of the ***Designing With Motion Handbook***. Chuck is currently the manager of technical services for Technology 80, a frequent speaker on motion control issues and a board member of the American Institute of Motion Engineers (AIME).