TECH TIPS

Helpful insights for designing with motion control from Chuck Raskin, PE, CMCS

# #2: Correcting Axis Mispositioning

This problem bothered me. Computer control mispositioning an axis can be caused by literally anything: couplings, noise, bad encoder, bad CPU, and the list goes on. I led the user through a series of tests and checks to determine what was wrong, and I wound up uncovering four separate problems, each producing a different effect on the operation at various speeds and accelerations.

Figure 1 shows the motor system operating in a tangential mode. In Figure 2, we see the motor system controlling a set of nip rollers, which pull the product along. To make the required rate for any profile, the control had to accelerate from zero to 2000 rpm in under 50 milliseconds. This was just over the system step response capability of 43 milliseconds.

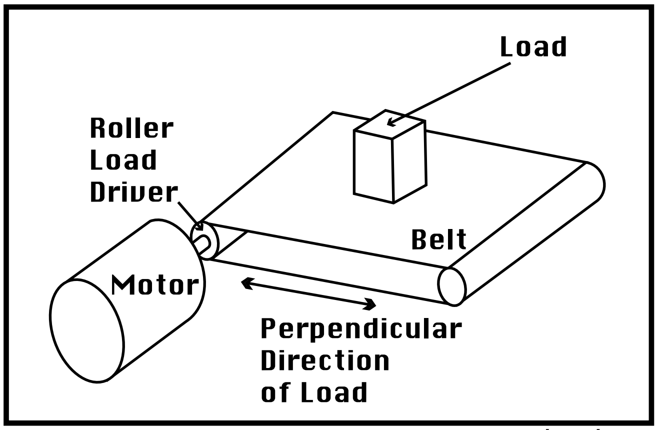


***Figure 1:** Motor system operating in a tangential mode*

The system could resolve 1000 counts per inch and could operate in voltage mode with a low inertia pancake style motor well suited for this application. The encoder was a 500-line unit (2000 counts per revolution).

I noted three problems:

1. The accumulated position error at low speed ranged in the vicinity of 1/32 inch per index.

2. The accumulated error at maximum speed ranged in the vicinity of 1 inch per index.

3. The system did not respond properly to the PID terms, as my experience indicated it should have.

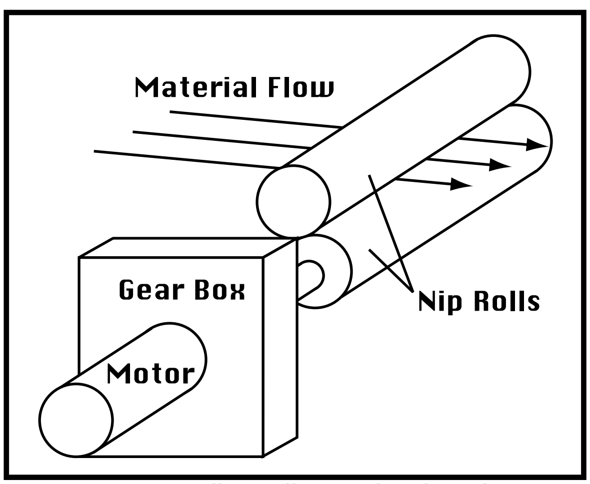


***Figure 2:** Nip Rollers pulling product through system*

In general, to troubleshoot software problems, you approach them one at a time. Since each problem you encounter usually alters operational flow. However, when dealing with a machine, all visible symptoms can be related in one form or another to bugs, which require a slightly different troubleshooting tactic.

In this case we eliminated the possibility of noise-induced mispositioning by first observing the encoder signals with a storage oscilloscope they were clean. Because of this, I decided to retune the system and to make the index rate before finding out why the system mispositioned. Therefore, if the reason for the mispositioning was due to mistuning, returning would correct the problem immediately. I had the user lower the maximum velocity to 500 rpms and the acceleration (and deceleration) to one second. We gradually raised the value of the Kp PID term until the motor began to oscillate. The Kp equaled 50 at the point of oscillation, yet the maximum allowable Kp value was 32000. I suspected that the system leaned more toward a rotary type and decided to begin tuning with the Kd gain function. Setting the Kd sample time to 5 percent of 43 milliseconds (approximately 2 milliseconds), we began incrementing the axis and raising the Kd value. At the point of axis instability, we lowered Kd until the axis again stabilized.

We then increased the acceleration and deceleration ramps until we created a flat top slew rate in the velocity profile (see Figure 3). We continued this process, retuning the Kd if required and raising the maximum velocity until we met the worst case profile requirement. Upon completing this profiling operation, we met all requirements for stability, velocity, and product rate.

Next, we had to find the cause of the axis mispositioning. We marked all shaft coupling points and slowed the system down to a crawl. After operating the unit for a few minutes, the system deviated out of position. I immediately checked the program index routine.

Indexing a motion axis the same incremental amount time after time should be done in an incremental mode. The user in this

**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 #2: Correcting Axis Mispositioning

case made an absolute move from zero, zeroed the counters and then repeated the same absolute move profile. The problem here is that the counters zeroed at the actual stopping position, not at the desired stopping position. Because of this, the actual stopping position became the new zero position. Since it is rare that an axis will stop repeatedly at the precise no-error position called for, the axis will creep from its original home (0) position. It can do this in either the forward or reverse direction. We changed the software to do index moves in incremental mode rather than absolute mode, and the problem disappeared.

Now that the unit ran properly, we increased the system speed to once again make rate, and there it was again — mispositioning! We waited a moment to ensure a reasonably large deviation from the desired position and then stopped the operation. Upon inspecting the marks we had previously made at each shaft and coupling point, nothing had moved. Once again, we checked the encoder signals, and they were clean.

We decided to change the encoder—not to another brand, but to a unit with a higher acceleration rating. Problem solved! We found that the disk inside the encoder was slipping, causing mispositioning at higher acceleration rates.

Interestingly enough, the customer had done all of his homework as far as calculating all possible inertial, ratios, requirements, etc., but he had not checked the encoder data sheet to determine whether or not motor acceleration exceeded encoder-acceleration — it did.

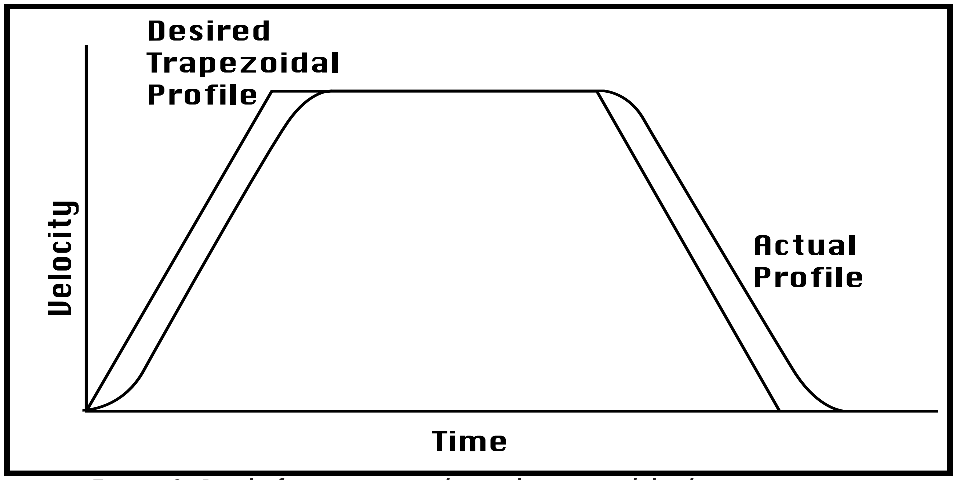


***Figure 3:*** *Results from increasing the acceleration and deceleration ramps*

At the end of this session, we found that:

1. The axes were improperly tuned.

2. The system acceleration rate was exceeding the encoder's capability, and

3. At the completion of each move, the user had zeroed the counter, resulting in mechanical creep ■

## 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 currently working on the fourth edition of the ***Designing with Motion Handbook***. Chuck is currently the manager of technical services for Technology 80 and a board member of the American Institute of Motion Engineers (AIME).