Motion Control for the Newcomer

Designing a system to boost your machine's performance may be easier than you think

Jacob Tal and Todd Shearer, Galil Motion Control

For engineers whose experience has centered on mastering the mechanical operation of their machines, the task of having to suddenly integrate a new motion control system can be very daunting.

Faced with decisions on everything from component selection to software programming, it's not surprising that engineers designing their first motion control system often ask a raft of questions. Why should I use a stepping motor instead of a dc servo motor? When is a linear motor a better choice than a lead screw? Do I want centralized control or distributed control?

But though motion control can be complex, with technologies constantly changing, engineers shouldn't view it as a black art either. With good planning, as well as reliable support from vendors or integrators, you can design successful systems that will substantially boost your machine's performance.

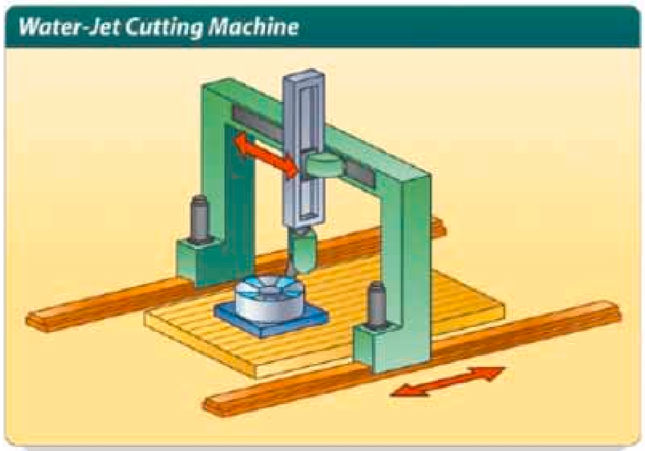
It all starts in the planning phase, which consists of all the decisions and calculations made up to the actual purchase of the equipment. Here you need to address the common thread that applies to all motion control applications: Something needs to be moved a specified distance or speed. Moving that load depends on three major components–the mechanics and couplings, the motor and amplifier, and the motion controller.

Mechanics and Mechanical Coupling

When it comes to choosing these components, you first need to determine the type of motion that you'll be commanding. Among the most common power transmission options are:

- Lead screws translate the rotary motion of a motor into linear motion. Performance of the lead screw ranges from a simple plastic nut on a threaded shaft–good for millimeter-type accuracy-to very high-precision ball screws with

More on page S29 and S30



One common machine type that novice motion control designers encounter is an X-Y Cartesian gantry, used for such applications as water-jet cutting, pick and place, or material dispensing.

Galil's DMC-21X3 controller with sandwiched amplifiers is designed for Ethernet applications.

Stitching It All Together

To understand how meeting a motion control challenge works in the real world, Galil engineers point to the thinking behind a system for an automated quilt stitcher.

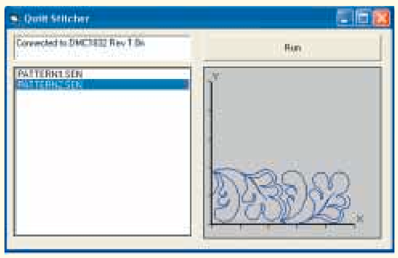
In this application (see Figure 1), a two-axis gantry (X-Y) moves a reciprocating needle axis (Z) around the surface of the quilt. A PC-based user interface allows intricate patterns to be drawn and loaded to the controller. Among the prime system requirements:

- X-Y coordinated to follow any two-dimensional path
- Constant stitch length (regardless of speed)
- X-Y path specified by CAD
- Graphical User Interface

Engineers chose lead screws for the X-Y gantry since high speeds and high precision are not required. The Z axis consists of a custom-machined cam for reciprocating the needle.

This application requires a host PC, so a bus-based controller is appropriate. The PCI bus allows for rapid transfer of the path data from the PC to the controller—in this case, the DMC-1832 model, which contains three axes and I/O.

Among other key components, engineers selected a Galil NEMA 23-sized brushless motor because all three axes require less than 0.3 Nm of continuous torque. The Galil AMP-19540, a four-axis brushless amplifier (500W per axis), is properly sized for the motor. Like many vendors, Galil offers a Motion Component Selector program for choosing the right motors and amplifiers.



(Figure 1) In this application, engineers had to control a two-axis gantry moving a reciprocating axis around the surface of a quilt.

(Figure 2) Once the Quilt Stitcher Visual Basic application is launched, the user chooses a stitch pattern, represented by a two dimensional image on the right.

Software considerations

Engineers used the Galil CADtoDMC and ActiveX toolkit for the PC-based user interface. The CADtoDMC program translates standard .dxf CAD files into Galil motion files, while the ActiveX toolkit provides ready-made GUI components for the case of a Visual Basic or C++ environment.

Once engineers selected these building blocks, they needed to go through several application steps, including:

- **Vector.** This application requires that the X-Y axes be coordinated to follow a vector path. CAD2DMC uses the Galil vector mode (VM), which coordinates two axes in a Cartesian coordinate system (Figure 2).
- **Gearing.** In this application, stitch length must be constant regardless of the speed. The gearing function allows one or more axes to be slaved to a master axis. Here, the Z axis is geared to the vector motion of the X-Y axes.
- **Graphical User Interface.** The interface written for this example, in addition to the sample Galil code, is shown in Figure 2. When the Quilt Stitcher Visual Basic application is launched, the user clicks on a stitch pattern file, and a two dimensional image of the pattern is displayedf (with the DMCMove ActiveX Toolkit object).

For more on this and other typical applications, visit http://rbi.ims.ca/3868-531.

recirculating bearings capable of sub-micron accuracy. The advantage of the lead screw is that you can select a wide range of pitches and accuracies. The disadvantage is that backlash can decrease that accuracy.

- Gear reducers may be your best choice when a rotational motion is required. However they may have too high an inertia for a motor to drive. The rule of thumb is that the inertial ratio between the load and motor should be less than 10:1. A higher gear ration can decrease that mismatch, but it will also decrease the speed of the load. Selection of the gearhead depends on the accuracy needed and the available budget. With gearheads, too, you may lose some accuracy due to backlash.
- Direct drive represents the simplest form of transmission. Consider this option when the load can be coupled directly to the servo motor. The two most common types are the linear motor and the direct drive rotary motor. Since no coupling is required, there is no backlash to decrease accuracy. However, be prepared to pay more, versus gear-reduced or lead-screw equivalents.

Motor and Amplifier Choice

Stepping motors and dc servo motors dominate motion control applications, though you'll also find such alternatives as voice coil motors and piezo-ceramic motors. Most motor companies offer special software and online selector guides to help you size particular motors. Selection of the proper motor type depends on the required performance and the available budget. For example:

- Stepping motors and drives perform well in applications that require low speed, high torque and low cost. These motors can run open-loop (no position-sensing device required), and require little tuning for optimum performance. The stepper amplifier determines the resolution of the motor by dividing the step pulses into full, half or even micro steps (up to 256× or greater).
- (Brush or brushless) dc servo motors will deliver higher performance, versus a stepper motor, but you'll also face higher costs and greater complexity. These motors can achieve speeds upwards of 6,000 rpm. However, since they are typically used in a closed-loop positioning system, as a result, they require more tuning or setup from the amplifier and controller. You pair these motors with either analog or digital drives. With the digital option, you get such advantages as programming and positional control, but you'll also pay more.
- Piezo-ceramic motors, while used in a much smaller percentage of applications, may be a good choice in applications that require very high accuracy but limited travel.

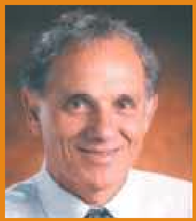
More vendors offer automatic tuning, such as Galil's WSDK servo tuning package shown here.

System Architecture and the Motion Controller

Here's where the motion control challenge may begin to get more complicated. Newcomers to motion control design need to become familiar with the brains of the system, especially the motion controller, PC, and software.

Newcomers to motion control design need to become familiar with the brains of the system, especially the motion controller, PC, and software.

The motion controller determines the motion of the axes, or motors, in the system. Engineers generally can choose from three options: centralized control, true distributed control, and flexible distributed control. For a look at the differences, got to page S33:



Jacob Tal, co-founder of Galil

"Motion control has not become too complicated. There are plenty of standard components and design tools that shorten and simplify the design process."
—*Jacob Tal*

Prime Educational Tool: **http://rbi.ims.ca/3868-531**

More than 10,000 engineers all over the world have attended motion control seminars conducted by Jacob Tal, co-founder of Galil. While motion control choices have multiplied, Tal still believes that ordinary engineers can master the technology and find ways to simplify design.

A good place to start is Galil's own website (**http://rbi.ims.ca/3868-531**). Among its offerings:

- **MotionCode toolkit.** Each MotionCode application details the procedure for integrating a controller into a particular machine type, and gives the engineer recommendations on components and downloadable software code.
  http://rbi.ims.ca/3868-532

- **"Jacob Talks" Web tutorials.** Engineers can view more than 20 popular tutorials free, dealing with such topics as "Tuning Servos for Best Performance" and "Servo Amplifier Basics."
  http://rbi.ims.ca/3868-533

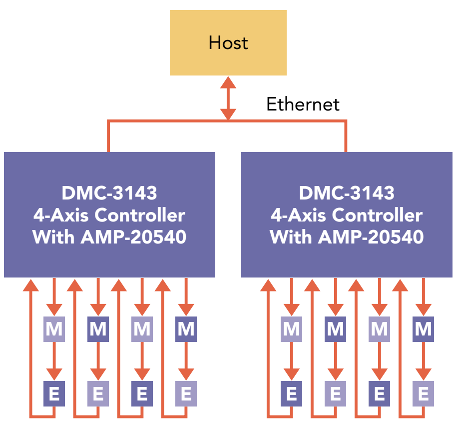
- **Ask the Doctor' forum.** This online help forum, monitored by Galil engineers, has attracted nearly 800 active members who post questions and information daily. Topics include: tuning, connecting hardware, and DMC programming.
  http://rbi.ims.ca/3868-534

- **Servo Trends.** This quarterly publication, featuring technical articles, product information, and application information, is now available online.
  http://rbi.ims.ca/3868-535

- **Galil Web-based training.** Dr. Tal's popular "Motion Control Made Easy" seminar forms a self-paced course and includes two modules running two hours each.
  http://rbi.ims.ca/3868-536

- **Galil application notes.** Galil application engineers have written nearly 200 application notes, available for downloading. New notes appear frequently.
  http://rbi.ims.ca/3868-537

- Centralized control features one motion controller (PC-based or stand-alone) connected to amplifiers and motors located elsewhere on the machine. This is a cost-effective choice on multi-axis machines, where the axes are grouped close together. The disadvantage: Increased wiring due to the fact that all signals must come back to the central controller.
- True distributed control includes multiple single-axis controllers (or controller/amplifier packages) located throughout the system and capable of communicating with each other. This architecture reduces cabling, since the controller/amplifier is closer to the axes. However, multiple axes can be more costly. As a result, operators may find it more difficult to tightly synchronize motion across axes.
- Flexible distributed control combines the preceding options. Axes are grouped together into small islands of control, with each island communicating with the others. Amplifiers may be packaged with these controllers or stand-alone. The benefit is that wiring and costs are minimized.



The Galil DMC-31x3 iontroller can be used in a flexible distributed control network, where "islands" of axes are grouped together over the Ethernet.

Communication Structure

Monitoring the performance of machines is essential to motion control, and engineers must choose a communication structure consistent with the control architecture and the types of data and software in the system. Some of the major options:

- Centralized control. Typical communication types within the PC are the PCI bus, ISA bus or PC/104 bus. If the application requires a great deal of data to be passed between the host software and the PC, a centralized control on the PCI bus is a good choice. Host software could be a Visual Basic or C++ program, or special automation software, such as LabView or Think&Do.
- Distributed and stand-alone communication. Typical communication types include:

Ethernet, CanBus, FireWire, and RS-232. Oftentimes, the type selected depends more on the controller and drive used in the application.

Phase Two: Connecting the Components

During this phase, when all of your components have been arrived, you'll need to address such tasks as system tuning, motion programming, and the user interface.

First, you must tune the amplifier and controller to match the requirements of the motor. The motor needs a certain current to drive the load, and the amplifier gain (A/V) must be adjusted to that value. Similarly, you must calibrate the motion controller so it outputs the maximum command signal for peak torque of the amplifier and motor.

In addition, you have to tune the motion controller to insure optimum positional performance of the motor. If you're using a stepper motor, you can skip this procedure. But if you've selected servo motors using optical encoders, you'll need to tune a filter on the controller (typically PID) to make sure that the motor moves to a commanded position with minimal overshoot and settling time. Luckily for novice designers, most motion control and amplifier companies have software to help auto-tune these filters.

Motion Programming

Next, you need to address the motion program, which is the code that coordinates all the motion, I/O, and logic functions of the machine. The format of this code depends on the type of machine being designed. Among the options to consider:

- Stand-alone controller. If there is no PC in the system, you can write the program in the native control language. This program, which controls all machine functions, is stored on a controller's on-board memory.
- PC-based system. In this application, you can write a host program to operate the entire machine. Using one of the software packages mentioned earlier, you will be able to write complete automation programs to communicate on a real-time basis with the controller.
- Combination programming. A program is written in the native language and executed on the controller. Concurrently, the host PC communicates with the card for status updates. You'll find this method equally effective with bus style controllers (PCI, ISA) and stand-alone controllers (Ethernet, FireWire, serial).

The new Galil CDS-3310 is an example of an Ethernet based single axis controller and amplifier that can be used in a fully distributed control system.

Human Machine Interface (HIM) and User Access

The final phase in your programming duties involves the customer interface. This can be an HMI touch screen where the user enters data, or a PC terminal where motion status is continuously refreshed. If your machine operates in an industrial environment and requires user access, you might choose an industrial PC or a NEMA-enclosed HMI designed for rugged and harsh environments. If, on the other hand, your motion system is for a lab process, then a standard desktop PC should work just fine for user access.

Mechanics, motors and amplifiers, system architecture, programming chores. Dealing with all these parts of motion control design may seem confusing, but engineers are successfully solving these challenges every day. The biggest hurdle on the first control system is to know what to use and why. With that knowledge, the newcomer will know which companies to contact. Those companies in turn can help the engineer size and select components, as well as address key questions in the design and commissioning process.

Jacob Tal is chairman and co-founder of Galil Motion Control and a well-known author on motion control topics. Todd Shearer is a senior application engineer and the company's Northeast Region Technical advisor.