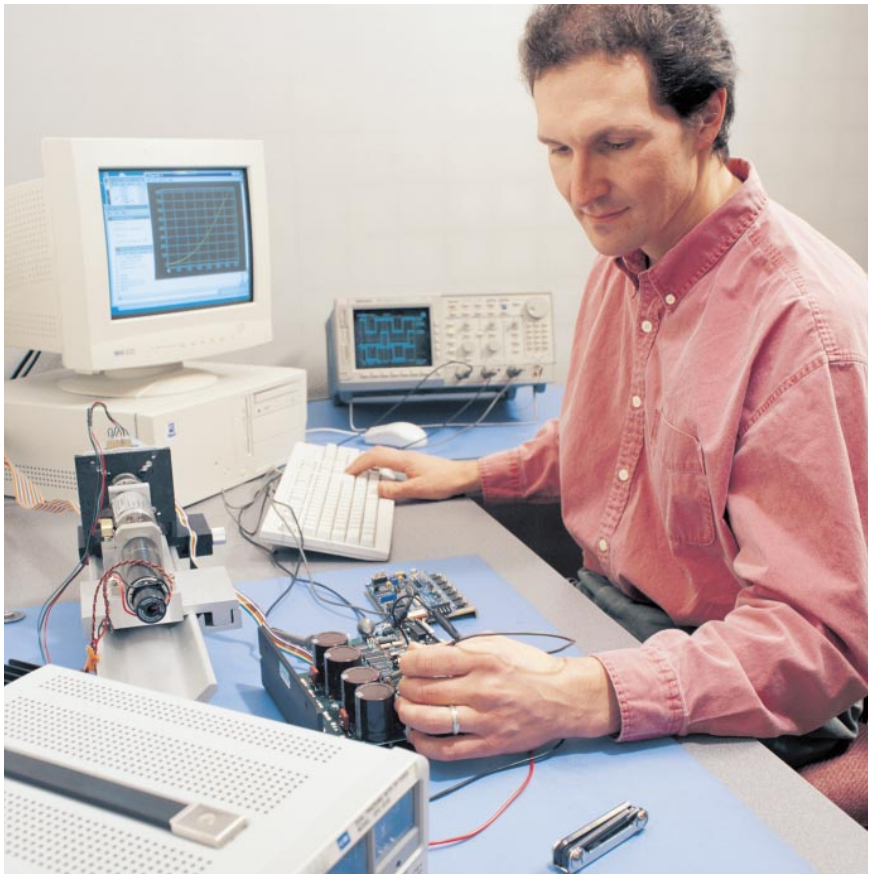
Motion control in THE DIGITAL AGE

Motion processors have the ability to do for machines what Pentium processors have done for personal computers. Find out what other engineers know about these devices and what they can do for you.

Chuck Lewin
Performance Motion Devices Inc.



Motion control in the digital age is played out on the desktops of engineers working at PCs. Gone are the temperamental analog circuits that had to be kept in line with trim pots and tedious adjustments. In their place are PC-based software tools and motion processors that convert simple move commands to efficient motion algorithms.

Not long ago, analog electronics ruled the world of motion control. It was a difficult time for everyone. Back then it was common to see technicians poring over control racks, trying to adjust PID gains, offsets, and other parameters. But that was before the development of motion processor ICs.

Now motion control is primarily a digital affair. Thanks to advances in chip technology, designers sitting at PCs enter literal commands on-screen, and using software simulators, measure every last detail about each motion axis right at their desks. At the press of a button, machines come to life, automated by programmable motion processors that produce signals optimized for motors and drives.

What's a motion processor?

A motion processor is an integrated circuit, implemented on one or more chips, that provides motion control functions. Operating in tandem with general-purpose ICs, motion processors automate all sorts of machines, generating move profiles, closing servo loops, and performing motor commutation.

The power of a motion processor is its programmability. Using a special command language, designers working on PCs can set critical move parame-

ters, such as velocity, destination, and servo gain, and download them to the motion processor unit (MoPU) over a serial link.

Once programmed, the MoPU will follow the commands of a PC or an embedded microprocessor, executing one move after another in response to an "update" signal. During each move, the processor makes sure that system variables, including servo lag and encoder position, stay within range. When the move ends, the MoPU sends an interrupt signal to initiate the start of the next cycle.

Motion processors work with most motors, including dc servos, permanent magnet brushless, and steppers. While some MoPUs handle just one type of motor, others take on a variety. Another distinction is the number of motion axes, with one, two and four-axis chips being the most common. The advantage of a multiaxis MoPU is that it facilitates synchronization, letting users trigger several axes at exactly the same instant.

Like all computer technology, motion processors are evolving rapidly. Lately, the biggest trend seems to be the integration of PLC-like functions. Through programmable I/O, MoPUs can be commanded to start and stop moves in response to external signals and events. They can also be programmed to trigger other devices, like sensors, in synchrony with machine cycles. For example, at the end of a move, when the machine reaches a certain po-

If you build it

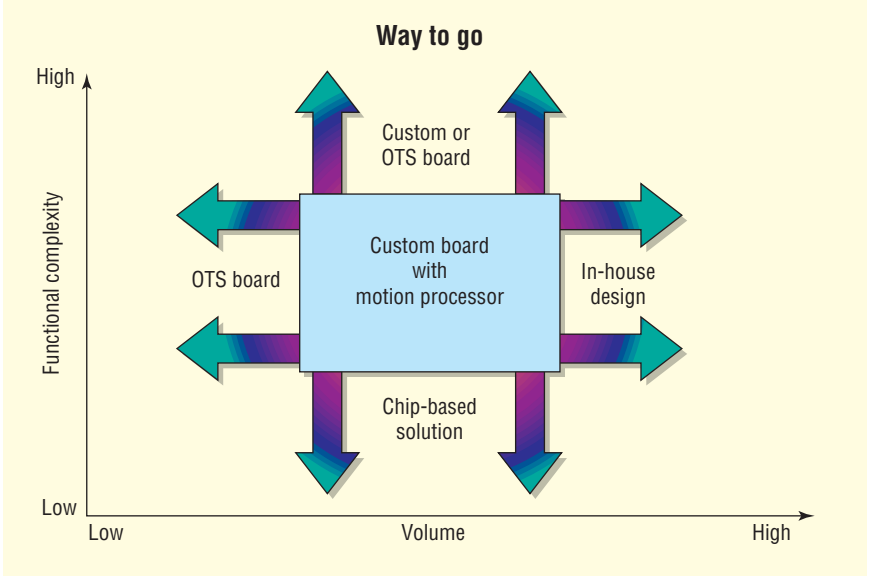
When it comes to implementing a motion controller, there are several options: a stand-alone controller, an off-the-shelf (OTS) motion card, a general-purpose microprocessor or DSP (digital signal processor) with a homemade motion processing circuit, or you can use an off-the-shelf motion processor IC.

Despite the focus on core competencies, many companies opt to develop their own controllers, using DSPs and general-purpose microprocessors. Though it may take weeks, engineers hand-code the complex algorithms required for trajectory generation and servo loop control.

The attraction of such an approach seems obvious: The resultant product is low in cost and optimized to the needs of the user. But cost is relative, especially when you balance it against the time (lost business) and cost of development, and the inconveniences that you know will crop up when one of the original designers leaves the company. Looking at it this way, the "buy or build" decision isn't so straightforward.

While there are no absolutes in making this decision, the accompanying table may provide some insight.

A motion processor is an integrated circuit geared for motion control. Some are single-chip, others, as in the case of the Navigator MC2100 processor for brushed servo motors, consist of several chips.



Volume and functional complexity usually determine what sort of motion control solution will work best for a particular application.

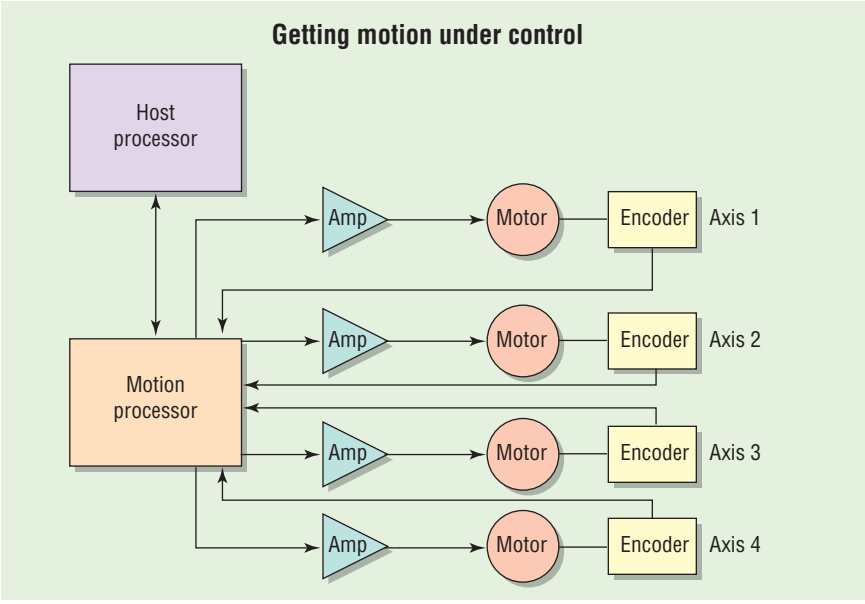
sition, the MoPU can "fire" a position sensor, causing it to take an important measurement.

Such flexibility is a good reason to consider designing your own motion card using off-the-shelf MoPUs. When you go that route, as opposed to purchasing a generic motion board, you get to pick and choose, and pay for, just the features and peripherals you want. Whatever you need, whether it's an unusually large number of I/O points or nonstandard card dimensions, you'll have a better chance of achieving it by designing your own card.

Inside out

Like any standard CPU (central processing unit) microprocessor or microcontroller unit (MCU), motion processor ICs are based on a modular architecture. Various circuit functions, each occupying distinct areas of the chip, work together to achieve a common goal — make a mechanical system move to a desired position or along a precise path. In all, there are five major circuit modules.

- Quadrature decode and index capture — This is where feedback signals go. Primary functions include a digital filter and a quadrature decoder for each axis. Though once exclusive to servo control chips, QDIC modules are becoming common in all MoPUs, including step-motor ICs.
- Host I/O interface — As the communications port, this module serves as a dedicated interface to the host processor. Using this interface, the host PC or embedded micro can send commands almost instantaneously to the motion IC. Most MoPUs use a byte-wide I/O data bus.
- Trajectory profile generator — This is where the math takes place. The trajectory generator performs all calculations necessary to synthesize move profiles supported by the MoPU. Typical profiles include trapezoidal and velocity contouring; S-curve profiling and electronic gearing are available on advanced processors.
- Digital servo filter — This module, essentially a DSP core, provides servoloop feedback stabilization. It is usu-



Motion processors simplify the design of multiaxis motion systems, reducing them to a few basic elements. Most processors have the ability to drive everything from power ICs to stand-alone servoamplifiers, so you can control almost any size motor.

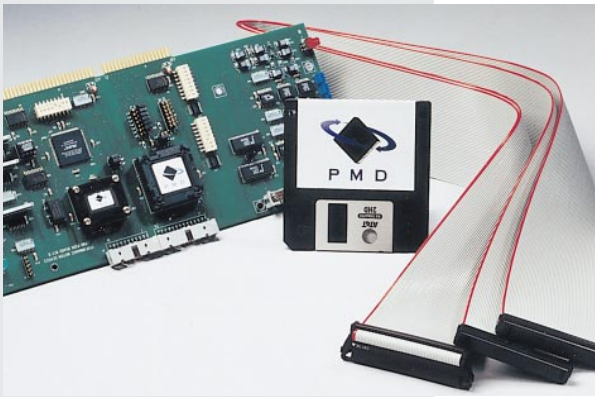
Get with the Program

Most motion processors implement some sort of packet-oriented command instructions. Programming is easy once you're familiar with the commands. Here's a typical sequence that performs a trapezoidal profile motion on a single axis.

SET_1; set axis #
SET_POS 2468; set final destination position
SET_VEL 3456; set maximum velocity
SET_ACC 543; set acceleration
UPDATE; make the move.

The command mnemonics (SET_POS) are generally transmitted to the motion processor in hexadecimal form. Data can be sent to or retrieved from the motion processor. In the latter case, the host may need to read critical information like current status and axis position.

Motion processor ICs are organized in such a way that the host program provides the "language" (branching and looping constructs) while the motion processor provides the motion instructions. Standard (C, C++, BASIC) languages are typically used to program move sequences. Most vendors provide a library of software routines compiled in one of these languages to simplify software development.



PC-based motion processor development kits simplify motion system programming, saving time and money. By letting designers enter and execute move commands, the new Navigator kit makes it possible to quickly evaluate multiaxis systems, incorporating motors, amplifiers, encoders, and limit switches.

If you build it

Point-to-point moves are common in a variety of machines, including medical, semiconductor, and packaging equipment, and in such manufacturing operations as board stuffing, drilling, tapping, and light assembly.

The two most important considerations in point-to-point motion are end-point positioning accuracy and speed. To optimize these variables, you have to pick the right motion profile – which means matching the desired path to the characteristics of the axis to be moved. Consider a single-axis system consisting of a simple load driven by a servo motor without compliance.

Here, the simplest profile we can use is a **constant velocity** "full on" profile without ramping. Though easy to program, such profiles don't track well because no motor can change the speed of a load instantly. As a result, the servo lag builds up and there is overshoot and oscillation at the end of the move. Because the controller must wait until the axis settles, the overall transfer time is excessively long resulting in lower machine throughput. What's more, the oscillation at the beginning and end of the motion can cause excessive mechanical wear.

Now try a ramped motion also known as a **trapezoidal** profile. Here the servo tracking accuracy is dramatically better. This is because the move profile does not exceed the ability of the motor to accelerate and decelerate. Tracking errors are virtually eliminated and there's less ringing at the end of the move, which means higher machine throughput.

Most of these benefits are lost, however, if there's compliance between the motor and load. Here the system is much more sensitive because the load can vibrate, as if connected by a spring to the motor, even if the motor is stable.

STARTING AND STOPPING

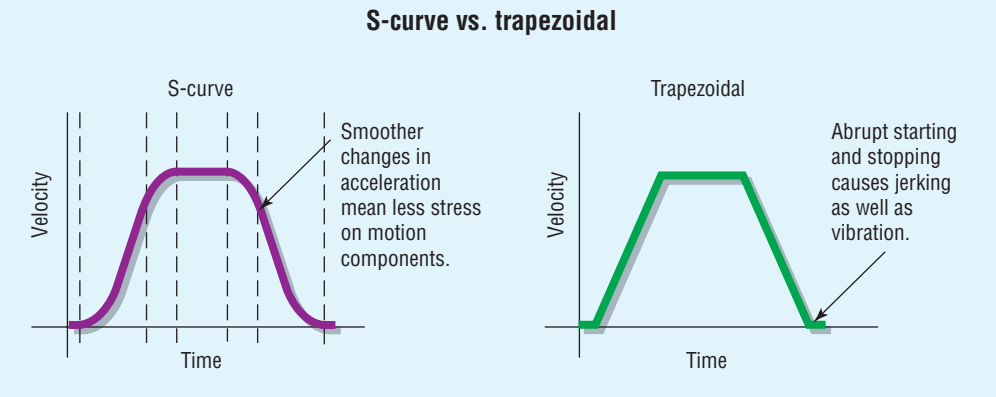
Profile	End-point oscillation	Comments
Constant velocity	Severe	Low-end profile useful for simple mechanisms
Trapezoidal	Moderate	Good overall profile for wide range of applications
S-curve	Small	Useful when there's compliance between load and motor
Parabolic	Moderate	Used mainly with step motors

To drive this more complex system calls for a more sophisticated **S-curve** profile. S-curve profiles are like trapezoidal profiles but with smoother velocity changes. They inject less vibration into the load because they accelerate and decelerate with less "jerk." Jerk, the second derivative of velocity, can be thought of as an energy spike that accompanies changes in acceleration or deceleration.

An example of S-curves in action is a car making a sudden stop. A compliant system, consisting of a heavy chassis on rubber tires, a stopping car is analogous to an accelerating or decelerating motor and spring-coupled load.

As for the car, a good driver knows to apply pressure to the brakes at a more or less constant rate, but just before the car comes to a complete stop, let off on the pressure. This final "smoothing" action keeps the car, and its passengers, from rocking back or oscillating – which is exactly what an S-curve does for a motor.

Another profile to consider for point-to-point moves, especially if they involve a step motor, is a **parabolic** profile. Unlike servo motors, step motors can't produce constant torque over their entire velocity range. At higher speeds, the available torque drops sharply. Parabolic profiles accommodate for this by maximizing torque (acceleration) at low speeds and minimizing it at high speeds.



Compared to trapezoids, S-curves put less stress on motors and drive mechanisms, as well as the load itself, because they produce smoother changes in velocity and acceleration. This, in turn, means less vibration and, in some cases, lower transfer times.

ally only present on servo motor ICs. Ordinary motion processors offer at least PID (proportional, integral, derivative) compensation, while advanced ones provide feedforward terms as well.

• Motor output generator and commutation — This is where the drive signal originates. The form of the signal depends on the type of motor supported. Step motor ICs, for example, provide step and direction outputs or direct two-phase commutation. Servo motor chips, on the other hand, produce a direct torque signal, usually in a PWM or DAC (digital-to-analog converter) compatible format.

Some servo processors generate commutation signals as well, calculating phase angle using Hall or encoder data. These devices output up to three phase signals per axis in either PWM or DAC format. For the smoothest moves, a MoPU employing sinusoidal commutation — rather than "6-step" or "trapezoidal" — is your best bet.

Putting MoPUs to work

Thanks to integration, designing a custom motion control board is a lot easier than it used to be. A typical board incorporates a minimum of devices — the fewer, the better — including a host microprocessor, a motion processor, and several motor amplifiers.

The advantages of this approach are many; lower overall cost, more compact design, and greater reliability because of the associated reduction in cabling. And virtually all the components you'll need are available off-the-shelf.

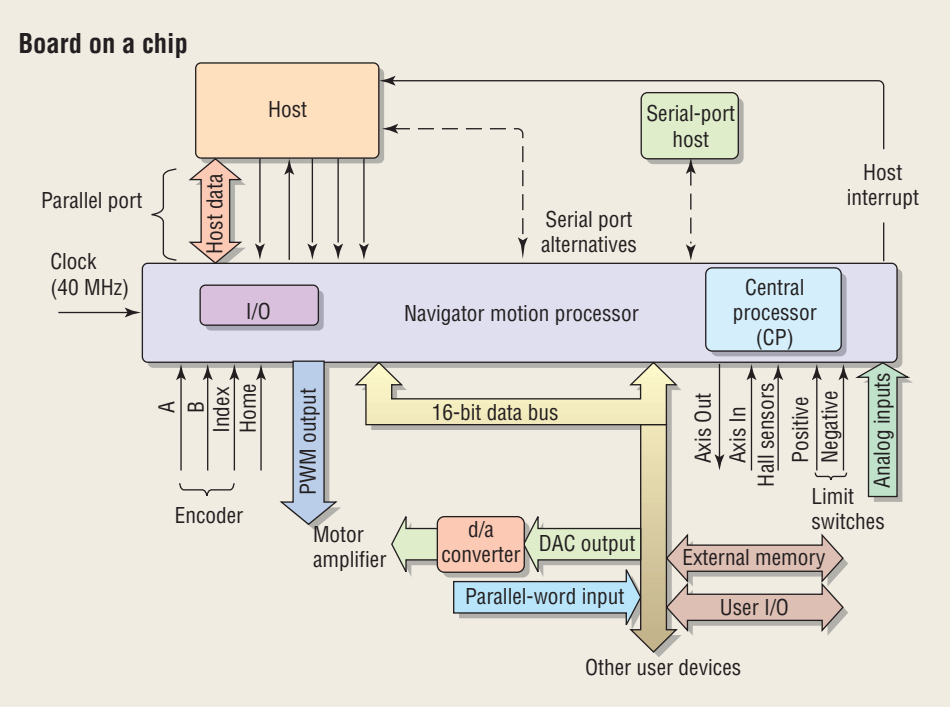
• Motion processor unit (MoPU) — These dedicated chips and chipsets have come a long way in the last ten years, incorporating functions that once took up the space of an entire board.

• CPU microprocessor — Designers can choose from a large number of embedded CPUs, and more are being developed all the time. The selection process is a matter of assessing your overall computational needs, the required number of digital I/Os, and whether or not the application calls for analog I/O.

• Dc-dc converters — These useful devices simplify power distribution. A typical machine controller might have several voltages, including 5 V for logic, 24 to 48 Vdc for the motor drive, and +/- 12 V for analog input/output.

• On-board amplifiers — Machines employing low-power motors (3 A) can often get by with IC-based amplifiers. For more power, you'll probably need MOSFETs or related driver chips.

• External amplifiers — Not all applications lend themselves to a single-board solution. Often, a more modular approach, using external amplifiers, is a better way to go. External amplifiers offer a wide range of power levels, extending to 1,000 W and beyond, and with a +/- 10 V analog interface you can tap into most commercial units.



Add a few electronic components to a motion processor and you essentially have a motion control card. Such cards can accommodate a variety of motors and, usually, anywhere from one to four axes. The host is typically a PC or embedded computer. All other necessary functions are integrated into the motion processor.

COUNT THE COST

	Custom board (MoPU)	OTS board	High-end OTS board	In-house design	Chip-based solution
Parts cost	low to med	high	high	low	low
Design cost	med	none	none	high	med
Quantity	med to high	low	low to med	high	med
Performance	med to high	low to med	high	low to high	low
Resource req.'s	low	low	low	high	low
Development time	low	low	low	high	low

Be aware of the implementation costs before you commit to any motion control solution.

Power for motion

One of the more daunting tasks associated with motion control is what to do about the amplifier. Designing an amplifier, whether it's for servos or steppers, requires detailed knowledge that few engineers possess.

To simplify the process, several chipmakers offer relatively easy-to-use "all-in-one" amplifier ICs. For drive applications up to 55 V and 3 A (continuous), these chips can reduce part counts by a factor of almost ten. When compared to discrete MOSFET designs, which require at minimum a driver chip, four MOSFETs, a charge-pump circuit, and a shoot-through protection circuit, amplifier ICs such as the LMD18xxx series from National Semiconductor offer a tremendous advantage.

Some of these chips even provide on-board current control in addition to half or full-bridge switching capability. Current control can substantially improve the high-frequency response of motion systems by individually controlling the current in each motor coil.

Little steps

Although a custom motion card is the surest route to an optimum solution, it may not be the most sensible in light of time constraints and economic uncertainties. In such cases, however, there's a strategy that minimizes time-to-market and engineering commitment, while keeping the option open to customization.

The approach boils down to purchasing a generic motion control card that uses an off-the-shelf motion processor. If you choose this type of board, you can easily convert to a custom card later. At that time, you can eliminate unnecessary features, saving cost as well as board space. And best of all, you can re-use your existing software.

This "buy now, shrink later" approach is becoming more and more common in motion control because it fits well in most business plans. When an engineered product is released to market, it is initially sold at a relatively low profit margin. Then a cost reduction phase is completed, letting the company sell the same product with the same functions, but at a higher margin. Switching to a custom card is not only feasible in this approach, it's one of the cost reductions that makes the approach work. ●

Chuck Lewin is Chief Technology Officer of Performance Motion Devices Inc., Lexington, Mass.

Next step...

Did you like this article?

For more information on motion processors, or call **(800) 568-7324.**