Motion's Digital Future

Dan D'Aquila
Pacific Scientific

Great technology is often underutilized prior to triggering an explosion of market acceptance. Think of the Internet, laptop PCs, and mobile phones. Such is the case with open standard digital interfaces between servo drives and controllers.

The German Machine Tool Builders Association initiated the development of an open standard digital interface in 1986. Originally a solution for computer numerical control equipment, the digital interface can benefit any application containing both a digital control and a digital drive. This includes applications in the packaging, paper, film, foil, food and beverage, robotic, material handling, assembly, and specialty machine industries.

Shifting to an open, digital, servo interface provides several options from which to choose.

Fifteen years later, that particular open standard digital interface, SERCOS, and others like it (MACRO and FireWire) are rapidly gaining momentum. Although most drives are digital, most products sold still have analog interfaces. However, manufacturers such as Delta Tau, Indramat, Kollmorgen, Nyquist, ORMEC, Pacific Scientific, and Rockwell Automation have introduced a number of controllers and drives with open digital interfaces. This new generation of motion controllers is almost all completely digital. A few analog outputs remain—a concession through which an otherwise digital controller commands an otherwise digital drive. However, this outdated interface technology doesn't make good sense in many applications.

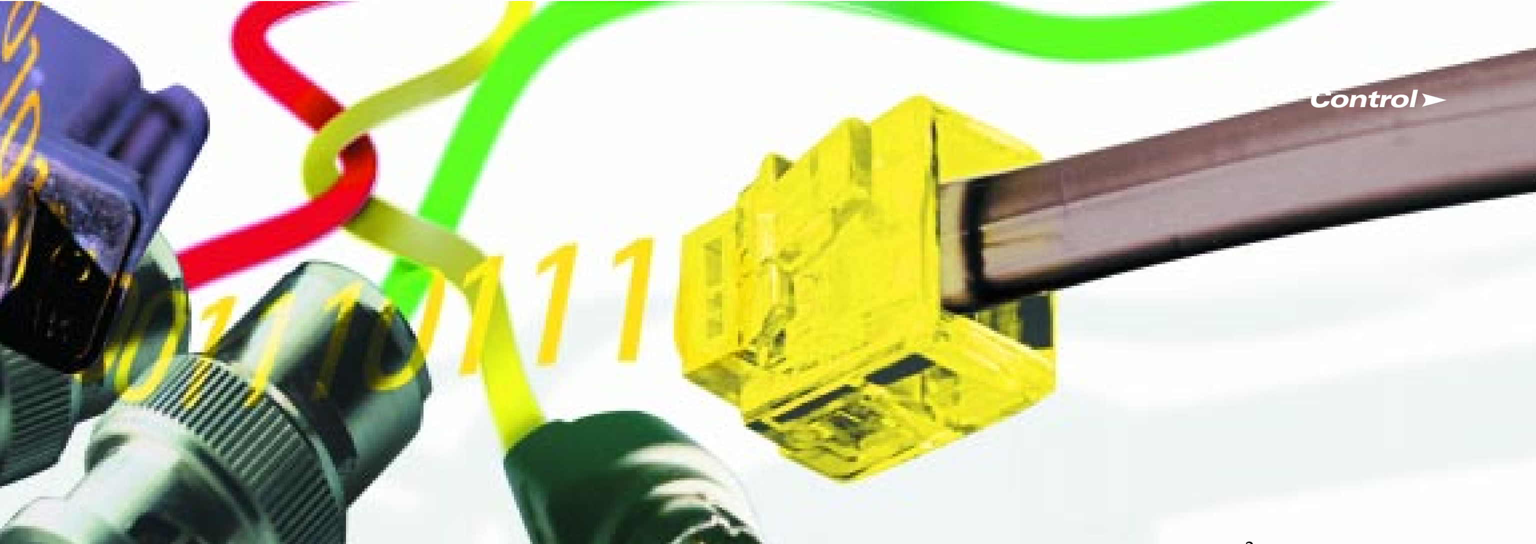
Open but Archaic

Traditional motion controllers used servo drives with a ±10-volt analog signal. This open, simple, and universally accepted control system allowed original equipment manufacturers (OEMs) and end users to integrate drives and controllers from different vendors.

Unfortunately, the analog interface also has numerous disadvantages. In demanding applications, its susceptibility to noise (electromagnetic interference, or EMI) and drift limit machine performance. In addition, analog fundamentally limits the transfer of information. There's usually just one command, and it goes only from the controller to the drive. Some drive manufacturers slightly mitigate this by adding a serial port and discrete I/O to configure the drive, monitor its operation, and perform other non-real-time functions. However, as serial protocols vary widely among manufacturers, this contradicts the analog interface's open advantages.

Finally, analog interfaces force costly wiring arrangements. The cost of cable alone makes it expensive; shielded motor/feedback cable can cost $30 to $40 per meter (m). Moreover, each axis in an analog control system typically requires 10–12 wires or more[1]: wires for the motion card, analog

[1] For a similar perspective, see "A Word to the Wires," *Motion Control*, December 2000.



commands, a common, I/O, and position feedback. A modest-sized machine can have more than 100 wires leading from the controller to the drives. Without significant planning when making these connections, the machine may be far too susceptible to EMI.

The large number of wires, coupled with a lack of diagnostic capabilities, often leads to troubleshooting nightmares. Service engineers might spend several shifts locating the source of a problem in an analog system. When this occurs, any up-front cost savings of an analog system are wiped out within the first few minutes of lost production time. This is also true of machines that are reconfigured to adapt to different product configurations. Rewiring an analog system, if done frequently, doesn't make sound economic sense.

Bye-Bye, Black Box

Recognizing the analog interface's limits, major automation suppliers have offered a digital controller/drive interface. Historically, this solution demanded standardizing on a single manufacturer's proprietary technology for all products in an application. While single-source solutions may offer some benefits, many OEMs and end users resent the constraints of being locked into a single supplier. That supplier might not offer the single best product for the application, forcing them to compromise on performance, cost, size, or service.

Today, OEMs and end users want the flexibility to integrate products from different suppliers and the benefits of a digital interface. Applied correctly, products with these combined properties improve productivity and quality, while reducing cost.

The Road to Digital

There are many open interfaces such as Ethernet, Profibus, and DeviceNet. However, companies that want products with a digital interface tend to examine three different open standards: **SERCOS** (SErial Real-time COmmunications System, standardized as IEC 61491/EN61491); **MACRO** (Motion And Control Ring Optical, nonproprietary but not standardized); and **FireWire** (developed by Apple; standardized as IEEE 1394, an open consumer market standard).

SERCOS

SERCOS communicates information among the controller, servo drives, actuators, and I/O using a fiber-optic ring. A ring connects up to 254 devices to a single master (although practical considerations often limit this to 8 or 16 axes). With SERCOS, the feedback and the motor connect only to the drive. Secondary devices, such as limit switches and encoders, can also connect to the drives, with all information passed through the fiber-optic ring.

The SERCOS interface works by specifying how a collection of messages, called "telegrams," traverse the ring in a fully deterministic[2] manner. These telegrams carry information between the controller and the drives. At the start of each cycle, the controller posts a master synchronization telegram (MST) on the SERCOS interface ring at precise intervals based on the programmed ring update time, which is generally 1 microsecond. This "heartbeat" coordinates message reception (although each drive uses the MST to synchronize its own internal clock and activities). Drives respond to the controller by posting amplifier telegrams that provide information such as axis position. At the end of each cycle, the controller posts a master data telegram that contains all the information from the controller to the drives.

The SERCOS interface standard defines a set of more than 400 normal parameters to specify control, drive, and I/O communications. SERCOS allocates 32,767 identification numbers (IDNs) for standard commands. A product must incorporate a subset of these commands. In addition, the standard defines 32,767 vendor-defined IDNs that can incorporate special features. All SERCOS products must pass a conformance test to ensure they perform according to the standard; this guarantees interoperability among products from various vendors.

The SERCOS, MACRO, and FireWire fiber-optic rings provide noise immunity, while SERCOS and MACRO allow drives to be distributed long distances from the controller without degrading performance. The SERCOS interface permits passing complete parameters and commands to intelligent drives. It also makes comprehensive status and diagnostic information available. In addition, users can collect configuration values

[2] Determinism is knowing the slowest possible response time with absolute certainty. In a deterministic network, each node communicates in the same order, whether it has new data or not. Determinism means that a response to an event always happens within a certain time frame—for example, 50 msec. If the system always responds in that time, it's considered deterministic to within 50 msec. True determinism in a network or bus ensures that each node, such as multiple servo axes, operates in perfect time with the others. This is done via some means of synchronization, such as a clock tick or a token.

unique to their applications such as motor current and bus voltage. Configuration wizards help users establish values that can't be predetermined or read during power-up (e.g., tuning parameters).

MACRO

MACRO's operation is based on standard fiber distributed data interface networking technology. MACRO operates in a ring topology and transmits data serially. Each station on the ring has an "in" port for receiving data and an "out" port for transmitting data. Nodes residing at each station can be amplifier axes, I/O banks, or communication interfaces to other devices. A station can have one or more nodes, allowing for multiaxis amplifiers with a single I/O port.

Data packets (groups of 96 bits of serial data), sent out by a motion controller (master node), are targeted for a specific amplifier or slave node. If the data packet isn't meant for a specific node, it passes on unchanged. If it is for that node, the node copies the data packet's contents (typically commands), replaces that command data with feedback, and transmits the data packet.

Transmission across the MACRO ring requires no software intervention. The information sent from one node to another is written to a memory location, and the MACRO hardware takes care of the rest. MACRO passes position, velocity, torque, phase current, and pulse-width modulation commands from a controller to an amplifier (Tables 1 and 2). By transmitting data over the ring at 125 megabits per second (Mbps), MACRO facilitates closure of high-performance servo loops across the ring, allowing the controller and amplifier to split the motion control tasks between themselves at any level.

A MACRO communication system's only requirement is to equip each component with a MACRO interface. If the manufacturer doesn't build this interface into the component, an adapter can facilitate the MACRO link. Delta Tau, MACRO's developer, believes MACRO is one of the easiest fieldbus technologies to use. By having the bandwidth close to the proportional-integral-derivative (PID) loops between the motors and the controller, commissioning a machine is almost identical to what end users traditionally did with analog. In fact, a MACRO-based machine uses traditional PID tuning (unlike SERCOS's specialized data telegrams).

Because the MACRO data packet is always the same shape, there's no need for many of the packetizing protocols used by technologies that operate under standardized communication procedures (e.g., FireWire, USB, and Ethernet). As such, MACRO doesn't require a relationship between the controller and drive manufacturers to operate efficiently. It's an open system, always isochronous, and Delta Tau encourages companies to download the specification (*www.macro.org*) and design it into their products.

Register #	3 (16-bit)	2 (16-bit)	1 (16-bit)	0 (24-bit)
Master command	Motor flag command			Torque command
Slave feedback	Motor flag status			Position feedback

Table 1 Typical torque command packet.

Register #	3 (16-bit)	2 (16-bit)	1 (16-bit)	0 (24-bit)
Master command	Motor flag command	Phase C voltage command	Phase B voltage command	Phase A voltage command
Slave feedback	Motor flag status	Phase B current	Phase A current	Position feedback

Table 2 Typical direct PWM command packet.

FireWire

Unlike SERCOS and MACRO, FireWire (IEEE 1394) originated in the consumer realm and is now widely used in PCs and electronics such as digital camcorders (Sony calls its version of 1394 "i.LINK"). Consequently, the chip set enjoys a low cost ($10 to $20). Manufacturers such as Nyquist Industrial Controls believe that because it meets high-speed PC data communication requirements, uses an open standard, and offers all the benefits of digital communication, 1394 has all the features needed for servo/controller/application communication.

1394 defines two bus categories: *backplane* and *cable*. The *backplane* bus provides an alternative serial communication path for parallel bus devices plugged into the backplane. A "bus

Feature
Type of communication and network topology
Physical media
Maximum cable length
Communication modes
Servo loop modes supported
Transmission speed
Update rates
Level of determinism
Number of masters
Maximum number of nodes
Physical standard
Control standard
Cost
Flexibility
Hot pluggable
Supports supervisory controls and additional I/O
Number of vendors supporting technology (worldwide)
Web site

bridge" connects between buses: a 1394 to peripheral component interconnect interface within a PC, for example. Sixteen-bit addressing provides up to 64,000 nodes in a system, with up to 16 cable hops between each. Six-bit node IDs allow up to 63 nodes to be connected to a single bus bridge (the limit for a conventional FireWire-to-PC adapter card). Ten-bit bus IDs allow up to 1,023 bridges in a system.

Each node normally has three connectors (the standard allows 1–27). As many as 16 nodes can be daisy chained up to 4.5 m through the connectors, for a total standard cable length of 72 m. IEEE 13946 specifies fiber solutions with up to 100 m through the connectors. Extra devices can be connected in a leaf-node configuration. Physical addresses are assigned on bridge power-up (bus reset) and whenever a node is added or removed from the system. Device ID switches aren't required, and the nodes are hot pluggable.

Like SERCOS, 1394 defines a standard for "isochronous" data transfers to guarantee network determinism (to within 5 microseconds, according to 1394). 1394 uses isochronous transfers to allocate bandwidth to servos on the network, guaranteeing timely transmission of torque commands, position feedback, and high-speed I/O status during each loop update. Nyquist uses FireWire for the position command update (trajectory) and application information. The velocity, current, and position loops are closed inside the controller or drive at much higher rates. The asynchronous messages for the position updates are transfered over FireWire within 1 μsec. Also, like SERCOS, asynchronous transfers can enable or disable the drive, dynamically adjust tuning parameters, modify drive setup, monitor system variables in real time, transfer diagnostic messages, and reset errors.

FireWire	MACRO	SERCOS
Serial, tree topology (point to point)	Serial, ring topology	Serial, ring topology
Twisted-pair copper or fiber optic (new)	Fiber optic or twisted-pair copper	Fiber optic
Copper: 10 m	Glass: 3,000 m; copper: 100 m	Glass: 800 m (node to node; max. ring length is 200,000 m)
Deterministic: asynchronous and isochronous	Deterministic	Deterministic/Cyclic for managing real-time data; noncyclic for non-real-time data
FireWire connection is between application and controller; controller runs position, velocity, torque; trajectory and other application info travels over FireWire	Position, velocity, torque, direct PWM, phase current, I/O devices	Position, velocity, torque, block mode (drive internal interpolation)
400 Mbps	125 Mbps	2/4/8/16 Mbps
4 kHz for position and velocity loop; 16 kHz for current loop (update rates not applicable to FireWire because of topology)	Variable; default rates about 10 kHz; rules do not allow speed <2 kHz.	Selectable from 2 μsec and up; 2 kHz for 20 axes; 4 kHz for 10 axes
16	16	1 per ring
63 nodes, with any number of servos per node; standard products (Nyquist) have 4 servos per node.	Type 1 MACRO supports 224 nodes	254 per ring; multiple rings allowed
IEEE 1394	Available at no charge; offered as on open system, nonproprietary protocol; no agency standard (yet)	IEC 61491; EN61491
OSI Level 1 and 2; OHCI 1394	Unnecessary; data format for MACRO is the same as used for motor control; based on a memory map, the drive protocol is usually a magnitude (similar to send an output to DACs in an analog system)	IEC 61491; EN61491
Projected cost is $10 per node (Nyquist)	Hardware cost about $75 per receiving station; multiple axes may be interconnected to a single physical MACRO port	??
High—no specific use is being made of dedicated drives, software layers, or protocols	High—125-Mbps throughput rate and homogeneous packet shape allow MACRO fieldbus to pass many different types of data; motors and I/O share same fieldbus in MACRO-based machines	High—any device that passes SERCOS conformity test can connect to any other like device
Yes	Yes	Yes
Yes	Yes	Yes
8 of 10 PC manufacturers	There are MACRO vendors worldwide	88 suppliers of servo drives and/or controllers.
www.1394ta.org	*www.macro.org*	*www.sercos.com*

Table 3 Comparable characteristics of open digital interfaces.

How Fast Is Fast?

Nyquist notes that performance of any fieldbus, FireWire included, depends not only on the speed of the bus but also on how the PC and the remote node handle the data. Often, most PC processing time is taken up by data handling. With standard open host controller interface (OHCI) chip sets, the effective practical data transmission speed for a 1-kilobit data package is about 100 kHz. Because of the direct memory access transfers that OHCI supports, this speed no longer depends on the operating system's interrupt latency.

Delta Tau claims that MACRO eliminates the need to add "complex data telegrams to economize the interface due to traditionally limited transmission rates. The efficiency of standardized communication protocols must be divided by the lost communication time associated with the header data information included in every packet."

Delta Tau emphasizes MACRO's 125-Mbps rate of data transmission. The company states that "in the case of a system with 16 amplifiers and two eight-axis controllers, updates to each amplifier and controller occur at less than 25 millisecond intervals (>40 kHz). This is more than fast enough to close high-performance servo loops across the MACRO ring, allowing the flexibility to choose distributed intelligence or centralized control."

When comparing SERCOS with other digital interfaces, it can be difficult to account for the impact of structural differences. For example, the new 16-Mbps SERCON816 ASIC can achieve a data transfer rate of 16 Mbps. A quick comparison to a MACRO system (125 Mbps) or FireWire (400 Mbps) may appear to favor the other systems.

However, in a SERCOS setup, its data transfer rate describes a *position command update*. With a SERCOS interface, the velocity, current, and position loops are closed inside the drive at much higher rates than the position command is updated. For example, typical SERCOS drives update the position loop at 2 kHz, the velocity loop at 4 kHz, and the

continued on p. 33

Comparing Drive Interface Standards: Analog vs. SERCOS

The connection between the motion control interface in the PC-based control system and the servo drive unit that's actually controlling the motor can come in two flavors: analog (Figure A) and SERCOS (Figure B). Table A lists the trade-offs to consider when you're deciding which flavor to use. Visit *www.sercos.com* for details about the SERCOS interface.

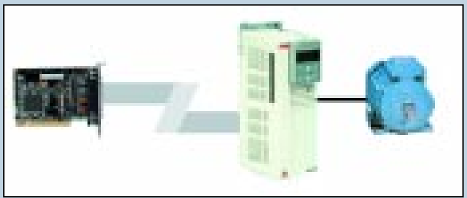


Figure A. The analog interface uses a separate wire for each signal between the controller and the drive. There may be up to 25 wires for each drive.

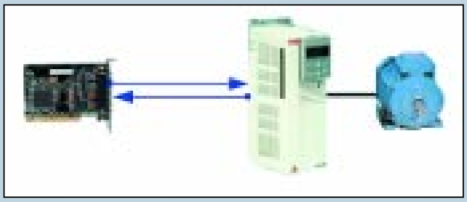


Figure B. The SERCOS interface uses a single fiber-optic loop to carry all of the signals between the controller and the drive.

Analog	SERCOS
The cost of the two interface options is roughly equivalent.	
Almost all drive manufacturers support the analog interface standard.	Not as many drive manufacturers support the SERCOS interface standard.
The analog interface delivers commands to the drive as fast as the motion control board generates them. Typically, new commands are generated every 800 µsec.	The SERCOS interface delivers commands to the drive every 2 msec. Faster SERCOS hardware is under development but isn't yet widely available.
The signals between the motion control board and the drive are analog values in the range of ±10 V.	The signals between the motion control board and the drive are in the form of digital messages.
The analog interface standard has been accepted and used for decades.	The SERCOS standard was developed in 1989.
The analog interface supports one drive. The exact number of analog interfaces supported varies among hardware suppliers. Typically it's 2–16.	The SERCOS interface supports up to 254 drives on an individual fiber-optic ring. The exact number of drives supported varies among hardware suppliers. Typically, hardware suppliers support eight drives per ring.
The analog interface has very limited drive diagnostics capabilities. Only amplifier fault detection is available.	The SERCOS interface supports a rich set of diagnostic capabilities for the attached drives.
The analog interface, which uses wires, is susceptible to electrical noise.	The SERCOS interface, which uses fiber-optic cables, is immune to electrical noise.
The analog interface is limited to relatively short distances between the motion control card and the drive.	The SERCOS interface can go long distances between the motion control card and the drive.
The analog interface has many wires to be connected. There is plenty of opportunity to make a wiring mistake.	The SERCOS interface has only two fibers to connect. It's difficult to make a wiring mistake.
The analog interface cannot support additional I/O on the drive.	The SERCOS interface allows for dedicated I/O devices to be added to the ring.

Table A. Trade-off considerations for the analog and SERCOS interfaces.

Make Contact!

Ovidiu Adam is a software engineer for Entivity, Inc. Prior to Entivity's formation, he spent six years as Steeplechase Software's senior driver development engineer. Contact him at 935 Technology Drive, Suite 200, Ann Arbor, MI 48108; tel: (734) 205-5129; ovidiua@entivity.com; www.entivity.com.

continued from p. 24

current loop at 16 kHz (and that's with the old 4-Mbps SERCON ASIC).

Because SERCOS's intelligent drive performs many of the motor control functions that MACRO and FireWire systems relegate to the controller, an equivalent comparison of functions can't be made with surface comparisons. Ultimately, SERCOS performs well in applications requiring fast process (for example, a seven-axis, horizontal-flow-wrapping machine requiring registration control and wrapping 1,400 items per minute).

More than likely, those of you who've explored these standards have heard proponents of a particular standard state that it's the only real "open" standard and the true possessor of a particular unique and valuable feature that makes it most suited for servo control. Likewise for each standard's detractors. An impartial examination shows that, like other technology, each standard has its pros and cons and fulfills a market need. The Web sites I've listed at the end of this article will enable you to review more detailed information and make an informed decision.

Most importantly, no matter which open digital interface technology is selected, all provide advantages over analog technology:

- High speed, high response.
- Complex commands and drive parameters that can be transmitted to digital drives and I/O devices via a high-speed serial communications bus.
- Fiber-optic connections that eliminate EMI.
- No digital-to-analog conversions.
- Serial cables that replace complex, bulky, and costly analog cabling. This can eliminate hundreds of signal terminations and significantly reduce wiring costs.
- Detailed diagnostic messages that are easily transferred from digital drives to the control and are accessible from remote locations.
- The elimination of potentiometer setup, parameter drift, and drive tweaking.

When combined and applied to the right machine, these benefits improve productivity, reduce downtime, and increase flexibility. With companies around the world now feeling a financial pinch, you can't afford not to explore going digital. *MC*

Make Contact!

Dan D'Aquila is a product marketing manager for Pacific Scientific's Wilmington, Mass., drive facility. He has been with PacSci 14 years in a variety of capabilities, including engineering, sales, marketing, and product management. Dan graduated with a BSE from Northwestern University in Boston, Mass. Contact him at 110 Fordham Road, Wilmington, MA 01887-2165; tel: (978) 988-9800.

*The author gratefully acknowledges the following companies and persons who contributed information to this article: Rod Dorschner, Rockwell Automation (*www.rockwellautomation.com*); George Ellis, Kollmorgen Corp. (*www.kollmorgen.com*); Ronald Larson, SERCOS N.A. (*www.sercos.com*); Jeffery Schatz, Delta Tau Data Systems (*www.deltatau.com*); Eric Smeets, Nyquist Industrial Controls (*www.nyquist.nl*); ORMEC (*www.ormec.com*); and Pacific Scientific (*www.pacsci.com*).*