Open Drive Interfaces for Advanced Machining Concepts
By Scott C. Hibbard
Indramat Division of the Rexroth Corporation
January, 1995

MACHINE CONTROL EVOLUTION

In machine control technology, there has been a most obvious evolution in the area of the CNC control. Since the inception of the CNC in the 1950's, its program storage capability has evolved from memory-less tape-reader-only units to network-aware giga-byte drive-equipped systems. Man-machine interfaces have gone from jumping digits on NIXIE tubes to full color graphics on active matrix LCDs. With shop floor programming, those displays are even used to automatically create a part program from a diagram. Processing speeds have gone from sub-second to sub-millisecond; accuracy from thousandths to millionths. Instead of precise manual tool setting, automatic gauging is available.

Even with these developments, all the control aspects of a machine tool inevitably boil down to one central focus: the control of tool or workpiece motion. The performance of today's most advanced CNC is still dependent upon the actuators that control axes motion. The controllability, response, speed range and dynamics of the chosen actuator ultimately determine the robustness of the machine. Thus, it comes as no surprise that the actuators used to produce axes motions have evolved as well.

Feed axes on early machine tools consisted of hydraulic actuators. As power electronics developed, electric actuators began to dominate the scene. Although stepper motors were used on smaller machines, the majority of cases saw the use of DC brushed motor technology, both permanent magnet for low to mid-horsepower applications, and wound-field motors for higher horsepower. Earlier drives used Silicon Controlled Rectifiers, restricting them to response times dictated by the power line frequency. With the advent of reliable power transistors, Pulse Width Modulation (PWM) technology introduced a higher level of performance. The next evolution came in the 1970s, when that PWM technology was applied to brushless permanent magnet servo motors.

Digital Drives Revolution

The latest round of development has centered on digital technology. Analog electronics was traditionally used for servo amplifiers because of its speed. As microprocessor technology matured, followed by Digital Signal Processors (DSPs), processing speed has attained a level where it can match the performance of analog devices. Developers looked to them to replace the analog components, heralding a wave of drives that can provide wider speed ranges, more precise speed control, adaptive control, tuning parameters, and diagnostics.

Digital technology has not affected just axis drives. Spindle drives often require a horsepower that makes permanent magnet motors prohibitively expensive. Induction motors offer an economical alternative in high horsepower, but come with one significant disadvantage. Unlike permanent magnet motors, induction motors do not have a defined relationship between electrical current and torque, which prevented any precise speed control of them. With the advent of digital drives, this restriction has been overcome. A complex “Vector Control” algorithm can be applied to an induction motor based system to yield tight speed control.

The Drive-Control Interface

Through all of this, our industry has been at first blessed, then cursed, by the standard used to interface CNCs and drives. That standard, defined under EIA’s RS-431, dictates that an analog signal should be employed, where the magnitude of the voltage represents speed, the polarity the direction. This “Velocity Command” signal is normally scaled so that 7 to 10 volts represents full speed of the motor or other actuator, hence the term “±10 Volt Command”.

We have been blessed in that this interface scheme has been very portable. It applied itself to the first hydraulic servos, through the wave of SCR drives, then into the brushless technology. It permitted CNC developers to increase the level of product performance without regard to the actuator technology used. Application-specific control solutions could be developed and easily take advantage of any appropriate motion medium. Likewise, servos evolved independently of what controlled them.

Alas, all good things come to an end. As servos began to evolve into digital-based devices, they soon exceeded the capabilities of the ±10 volt command. When the standard began to restrict this evolution, the blessing turned to a curse. To circumvent the restrictions of the analog velocity command interface, various proprietary CNC-drive interfaces emerged. Although they each solved some of the problems, they also created new ones. Foremost was that the openness and portability of RS-431 was lost. Products were developed that depended upon not just a particular type of servo technology, but even a particular vendor’s product. The situation clearly demanded for a new standard.

North American & European trade associations, as well as international standards organizations, have worked on a solution but found that a next generation drive interface is not an easy issue to address. There are many potentials for compromise. For a user to determine whether an interface technology is appropriate for his current and future needs, a more thorough understanding of the background issues is required. The best way to start is with an understanding of a traditional system.

TRADITIONAL CONTROL ARCHITECTURE

To simplify this description, electric servo systems will be used as the example. Other motion systems, such as hydraulic or pneumatic, can be viewed in much the same way. Traditional machine control systems are composed of three main physical components:

1. The motion control, ranging from a single axis control for simple applications to a fully equipped CNC for more complex machines.
2. A servo amplifier, or controller, that receives commands from the motion control and translates them into power switching that is applied to:
3. The servo motor, which normally has one or more feedback devices on it that provide position and velocity information to both the servo amplifier and motion control.

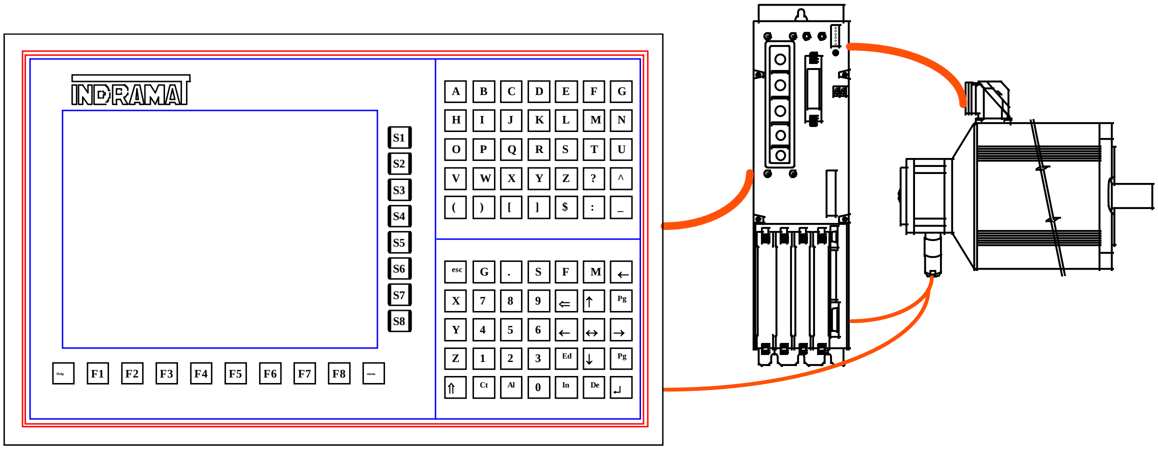
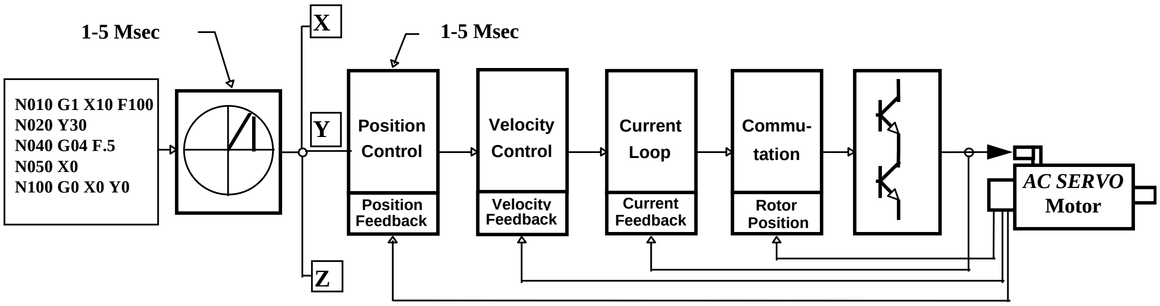


Figure 1. Physical Motion Control Architecture

Another way of viewing the control system is by the functions performed. Many years ago position control was simply a matter of reading a potentiometer on a slide and applying motion until the desired position was reached. Today's demands for fine position and velocity accuracy, even under widely varying load, have caused the architecture to evolve to something more complex.



```
N010 G1 X10 F100
N020 Y30
N040 G04 F.5
N050 X0
N100 G0 X0 Y0
```

Figure 2. Motion Control Block Diagram

The motion control contains a program that details a series of moves the machine should make. If the control is performing linear or circular interpolation, the motion control translates the path motions in the program into their individual axis movements and speeds. Once the control determines the total movement of each axis, as well as the velocity for those movements, it breaks the moves into small segments at a very exact, repeated rate. This "update" rate is typically 1 to 5 milliseconds. The control then applies the interpolated moves to a section of the control known as the position loop. The position loop compares the requested, or commanded, position from the interpolator with the actual position of the axis as read by a position feedback device. The difference in position, or position lag, is applied as a velocity command to the servo. This velocity command is compared with an actual velocity to generate a velocity error. The control uses the magnitude of this error to generate a torque command that is once more used in a closed loop. The result is that power semiconductors are switched to apply the desired current to a motor. In the case of a permanent magnet motor used on traditional axis drives, torque is proportional to current so the control is straightforward. For induction motors, a vector algorithm must be inserted to determine what amount of current to apply to which motor winding to achieve the desired torque.

Interface Point

Responsibilities in this control system have traditionally been split between the position and velocity loops. The motion program execution, interpolation and position control have all been performed in the CNC, while the remainder of the functions have been performed in the servo drive package. The interface itself has been the aforementioned RS-431 ±10 Volt velocity command signal. As described earlier, this division of responsibility has enjoyed the benefits of open systems and development independence between drive and CNC.

Restrictions

The reasons for necessary change are many, slowly accumulating over the years.

Resolution

The first is resolution. The size of the position and resolution values that a CNC has to deal with are impacted by two factors: the smallest addressable value and the highest addressable value. For example: If a CNC must control axes with a travel range of 60 inches at a precision of 0.0001", it must be capable of internally storing a number equal to 60/0.0001, or 600,000. Likewise, if an axis must travel at 300 IPM, and be programmable down to 0.1 IPM, the CNC must be able to handle a number equal to ±3000.

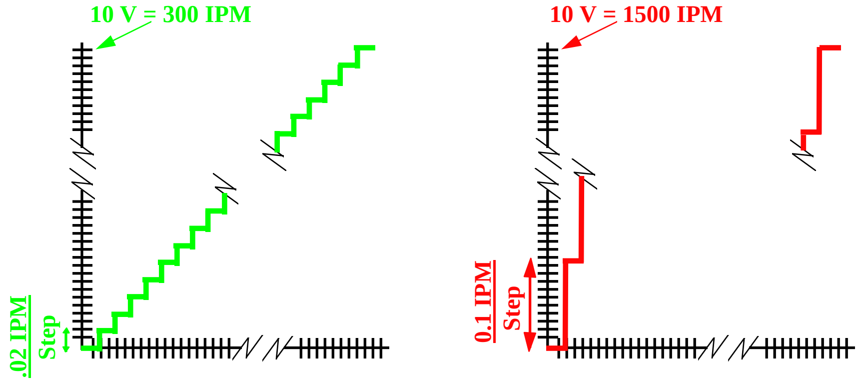


Figure 3. Analog Conversion Limitations

Sooner or later, these numbers are limited by resolution of the analog interface to the axis drive. CNCs usually employ a Digital to Analog Converter (DAC) to convert the internal numbers into an analog signal proportional to speed. DACs in use today are usually in the 12 to 14 bit range, with a resolution of 1 in 16000 at best. If full voltage on the DAC must represent rapid traverse speed, it can then be determined what the smallest addressable speed value is. This figure is important for more than just the obtainable resolution in speed control. Because this value is in the middle of a control loop, its resolution affects the stability of the control loop, and therefore the stiffness that can be achieved. In traditional machines where 300 IPM may have been the rapid traverse rate, a 14 bit DAC could address down to 0.02 IPM. This was more than sufficient for both programming resolution and the stiffness desired. However, today’s cutting speeds, much less rapid traverse speeds, are exceeding 300 IPM. For example, if 1500 IPM is the top speed of a system, now the DAC resolution limits the system to 0.1 IPM. This starts to impede both the programmability and stiffness of the system, and therefore impacts performance.

Hardware impact

Although not often recognized, the hardware requirements for an analog interface based control system can cause design and cost problems as the number of controlled axes increases. The digital to analog converters take up real-estate and introduce variable costs in a control. The required connectorization further complicates the system and increases costs.

Additional Information

Another restriction of the analog interface is the amount of information it can handle. Very simply, only one type of information can be transmitted, and in only one direction. However, control architecture has progressed to a point where much more information could be sent back & forth. Torque limit commands, torque monitoring information and diagnostic information are but a few examples. In each case, vendors have been forced to resort to discrete signals or a proprietary interface to convey this information. Although the problem at hand has been solved, it has involved a proprietary solution, and suffered the control system from a cost impact proportional to the information required.

Distributed control

Distributed control, where the control components are placed as close as possible to the device being controlled, is becoming an accepted practice. One example of distributed control use is on high production transfer machines. Up through the 1970's the typical transfer machine had a large number of electrical cabinets assembled in a line, sometimes up on a "catwalk" to conserve floor space. A huge amount of discrete wiring spewed from these cabinets over to the machine. In current systems, distributed I/O, distributed motion control, and now even distributed PLCs have significantly reduced the cabinet volume and wiring.

Similar situations present themselves in motion control systems. In larger machines a number of axes of motion may be most conveniently controlled from one motion controller. On the other hand, the axis motors are far enough apart that it would be more convenient to place their associated amplifiers close to the motors. The problem in achieving this is that the connection that must be extended is the ± 10V command. This signal is sensitive to electrical noise, which greatly impacts the practical application of distributed servo control.

DIGITAL DRIVE CHALLENGES

In addition to the restrictions when the ± 10V interface is used with analog based drives, a number of new issues arise when considering using the same interface with digital technology. A few examples:

Parametric Adaptation

One frustrating aspect of analog drives was that scaling and loop gain characteristics were determined by passive components in the drive. Regardless of whether the components were resistors, capacitors, or potentiometers, the only way to alter or inventory scaling and loop gain values was at the drive. Digital drives heralded a new way of adjusting these variables -- messaging. But how? Most early digital drives either came with a keyboard and readout on their front panel, or a serial port hookup to a laptop. In either case the data was only accessible via a vendor-specific solution, and a costly one at that.

Diagnostic Information

Analog drives were restricted to a few LED indicators and I/O points that gave a health check of the drive. A digital drive can contain full diagnosing, with plain language reporting. How can this information be conveyed to the upstream equipment? Like the problem of parametric adaptation, it was solved through proprietary interfaces and added hardware.

Drift Offset

A nagging problem with analog systems has been their susceptibility to temperature change and noise. Analog components change as temperatures increase, and a servo that stood still with a zero volt command at 50 degrees would drift when it warmed to 75 degrees. Although position loop systems compensated for this drift, the system was left with an offset. This made it almost impossible to maintain synchronization of axes over a wide temperature range. Grinding systems and multi-motor gantry axis have suffered from this problem in the past. Applications outside the machine tool industry, such as line shafting on converting machinery, had not benefited from servo technology due to this restriction as well.

Early digital drives promised to solve this. Their internal loop closures had no analog components, and hence no drift. But they communicated with -- you guessed it -- ±10V command. That command had to be interfaced with offset-prone DACs, and was itself sensitive to voltage drops bringing all the offset problems with it. Some vendors put the entire motion controller inside the drive to circumvent the problem, but that was no solution in the CNC world. Other created a “shut-off” for the command signal, interpreting everything below a level such as 50 milli-volts as “stop”. Unfortunately this prevents position loop closure from working.

NEXT GENERATION INTERFACE CRITERIA

From past experiences, a “wish list” of attributes can be produced for an analog interface replacement:

1. It must be an open system. Users of the products including the bus system should not be restricted to a given supplier for both drives and controls, as no one supplier can provide all solutions in both arenas. Likewise, designers of custom controls should have an open solution to design into their products, so that they may make use of the widest possible family of solutions. Such a system also permits the development of controls and drives independently of one another.
2. The bus must be technology independent, permitting use of various drive technologies such as brushless DC, vector control, VFAC, stepper, hydraulic, or pneumatic.
3. It must be economical. The total package cost should be no more for a control system with a next generation drive-control interface than it is today for an analog interface system.
4. It must support high speed and high resolution operations.
5. It must support access to internal data in a standardized format -- variables as well as diagnostics.
6. It should support single axis and multi-axis. The cost should not affect single axis, yet it should support synchronized multi-axis systems.
7. It should support distributed control.
8. It must afford the same or better troubleshooting aids as are available today.

Intermediate Solutions

In well intentioned attempts to overcome the ±10V command deficiencies, a number of intermediate solutions have appeared on the market. A couple of more popular ones are Current Command and Power Stage Interfaces.

Current Command Interface

The current command interface moves the split between CNC and Drive "one loop to the right". This affords some advantages, including:

- Less dependence on the lack of resolution of 14 bit DACs. A current command split into 16000 parts affects loop performance much less than a velocity command divided into that many parts.
- Some of the more often changed scaling and loop characteristic points are brought into the CNC control. This reduces the number of non-accessible optimization variables in the servo system.
- It automatically allows the control to operate the servo in either constant velocity or constant torque mode.

Disadvantages still persist:

- Some optimization components still remain in the drive; inaccessible to all but screwdrivers and soldering irons.
- There is still only one line of communication. Diagnostic messaging must be done via additional discrete wiring.
- The location of the velocity loop in the CNC automatically places a much higher processing demand there.
- Distributed control continues to be difficult, as the signal is still noise sensitive.
- The interface does not address other motion control methods, including Hydraulic, Pneumatic, stepper, & VFAC

Power Stage Interface

Another scheme moves the interface point to the "far right". In this scheme, the CNC provides power transistor switching information to a power bridge.

Advantages:

- It is a very low cost solution.
- Almost every single drive system variable is accessible inside the CNC.
- Power electronics are isolated from sensitive signals.

Disadvantages:

- Because the signals are sensitive, the interface is limited in length, impeding distributed control.
- Although the drive is greatly simplified, the CNC is correspondingly complicated.
- Additional information such as bridge temperature, motor temperature and feedback must still be transmitted discretely.
- The interface is highly technology dependent. For example, A CNC must be designed to operate a permanent magnet sinusiodally wound brushless DC motor.

No changes in technology, such as hydraulic, pneumatic, VFAC, DC, etc., can be effected without impacting the CNC. An immediate example is the vector drive technology spoken about earlier. This method of motor control is becoming increasingly more commonplace on machine tool spindle drives. For this technology to be supported under a power stage interface scheme, the entire vector control algorithm must be coded into the CNC.

A New World Standard

By 1987 the pressures had begun to build for a new standard. Machine tool builders were increasingly enticed by new technology in drives, only to discover that the advantages afforded by the new technology were achievable only via a proprietary interface. This restricted the family of control solutions that could employ the new technology.

Based upon pressures from its member companies, the VDW (German Machine Tool Builders Association) formed a joint working group with the ZVEI (German Electrical Standards Association). The charter of this working group was to develop a next-generation open interface between drives and controls that included the expanded opportunities afforded by digital technology. The interface was intended to permit continued drive technology independence; meaning drive technology should have no impact on CNC technology, and vice-versa.

The original committee member companies included major names in CNCs, drives, and machine tools:

ABB
AEG
AMK
Baumueller
Bosch
Gildemeister
Index
Indramat
Siemens

After over 40 man-years of combined effort, the working group published a specification for an interface scheme entitled SERCOS, for SErial Real-time COmmunication System. A look behind some of the earlier tasks of the working group will shed some light on the specification itself.

Interface Placement

The first issue tackled was “what interface placement point or points should this new bus support?” A goal was set that this be the only interface required between drive and motor electronics. This meant that the bus not only should transfer command information, like the old ±10V command, but also feed back information. Since one of the restrictions placed on the group was to continue the technology independence between drives and

CNCs, the power stage interface was ruled out. Both velocity and torque (current) interfaces were deemed acceptable for an open system.

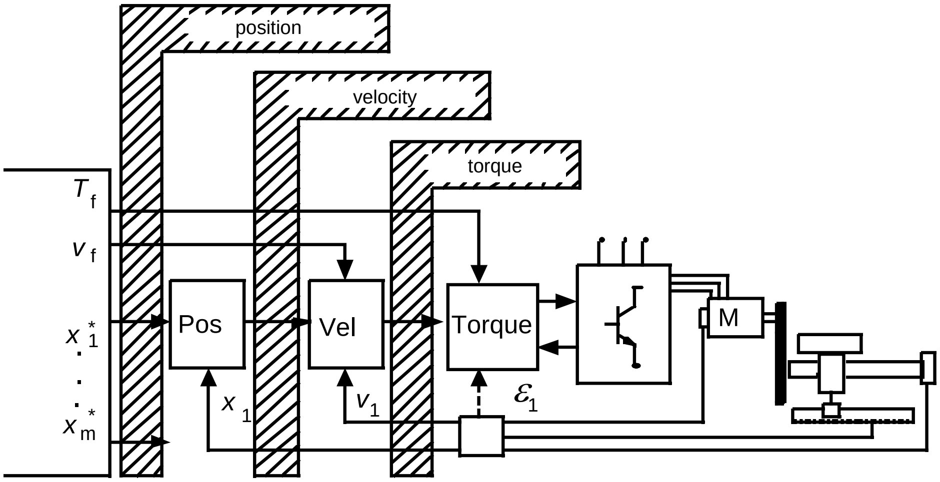


Figure 4. Interface Placement

A new and interesting interface placement was also contemplated, that of “position”. This was a radical departure from traditional technology, where the position loop was always tightly integrated with the interpolation software in the CNC. It was conceivable, however, that a digital drive could have the horsepower to close a position loop as well. This would reduce the number of loop closures in the CNC to zero. In addition to eliminating the need for active feedback devices at the CNC, it also would reduce the processor demands there.

Transmission Medium

The next subject addressed was the communications medium. Fiber optics was chosen due to its inherent noise immunity, especially important around large current drive systems. A ring architecture was settled upon in order to reduce the number of components required on a CNC. This allowed a theoretical unlimited expansion of axis count on a CNC without the need for additional hardware. If a B-axis is a machine option, adding it requires nothing more than opening the ring and placing the new drive in the ring. Practically, the bus is limited to 254 drives on a single ring.

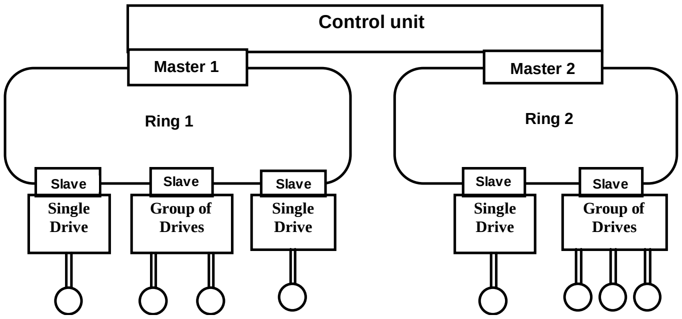


Figure 5. SERCOS interface Topology

Communications Structure

Using a serial bus to break control loops seemed to many to be an impossible task. Much background work was necessary to ensure strict synchronization of multiple axes, and a predictable update time at each axis. A master/slave communication structure was selected, where the CNC acts as a master, the drives as slaves. The drives are only permitted to respond to CNC queries.

Telegram Format

All communication between CNC and drive is performed via a defined set of "telegrams". Each telegram has a corresponding identification, or "Ident" number. It was agreed that not only should all parametric data, such as scaling and loop gains be set up this way, but also even the so-called "real-time" loop closure information. This scheme allowed the committee to standardize the most common interface data. For example, a control developer can always depend on ID No. 00036 being the velocity command value. To avoid restricting developers to just the standardized data, a large group of telegrams were also allocated for "vendor specific use".

System Of Units & Variable Format

Wherever possible, tables of acceptable units such as revolutions, inches, millimeters, etc. are defined for standardized telegrams. The format of the byte values is also defined. This is another step toward predictable operation of any servo system a control vendor may choose.

Cyclic Operation

The next task, a very difficult one, was to devise a scheme whereby loop data such as command signals could be transferred over a serial channel. Many thought this to be impossible, because of the inherent delays in a serial transmission. Closed loops expect instantaneous update of information. After looking at the big picture, however, a solution was developed. In CNC position loop software, the position error is determined in a software routine that takes time. The system works because the position error updates are designed to occur at predictable points in time. The same logic was applied in SERCOS. Methods are specified to keep the jitter on the serial link down to a low level, then an internal timing sequence is used to ensure that all drives in a loop act upon their command signal at the exact same moment, and all acquire their feedback information at the exact same moment. The result is transparent to the user.

The SERCOS interface cycle time is specified in a flexible format of 62.5 microseconds, 125, 250, 500, and then multiples of 1 millisecond. The amount & type of data contained in a cycle is also variable. This flexibility permits a designer to vary cycle time, content and number of drives to achieve the particular project's requirements. More data can be sent faster to a smaller number of drives. Slowing down the rate somewhat permits a higher density of drives per ring. When necessary, multiple rings can be employed. Looking at a typical cycle of operation illustrates the master/slave concept in use during cyclic operation. A cycle begins with the transmission of a “Master Synchronization Telegram”(MST) from the master (CNC). The MST is used as a “Time mark” for all slaves (drives) to determine when to talk on the bus, when to acquire feedback signals, and so forth. At a predetermined time after the end of the MST, the first drive in the system places its data on the bus in a Drive Telegram (AT). Each drive follows in turn, all synchronized off the MST. The drives are instructed during a SERCOS initialization phase when they should transmit their message with respect to the MST. After the last drive has placed its data on the bus, the master sends out a Master Data Telegram, or MDT. The MDT is one long message with space set aside for each drive in the ring. The drives have been previously instructed where their data is located within the MDT. As the MDT is received by a drive, it “fast forwards” to the start location for its information and retrieves it.

After the MDT is sent, another MST is emitted by the master control, signaling the beginning of another cycle.

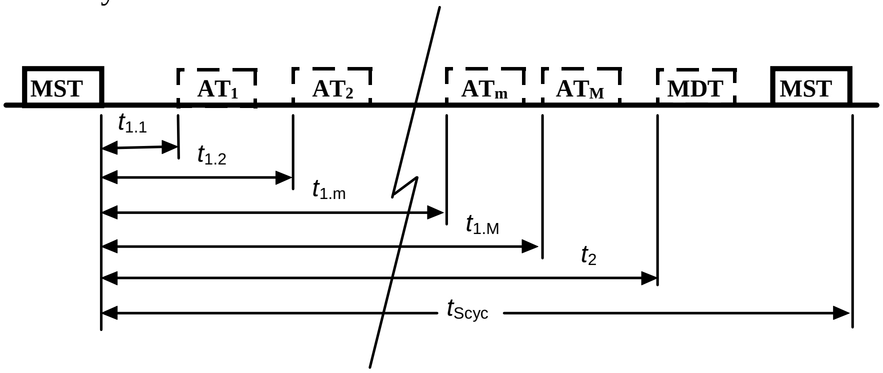


Figure 6. Timing Diagram for Cyclic Operation

AT Content

A drive telegram is composed of five main fields:

1. Beginning of Frame (BOF)
2. Drive Address (ADR)
3. Data record
4. Frame Check Sequence (FCS)
5. End of Frame (EOF)

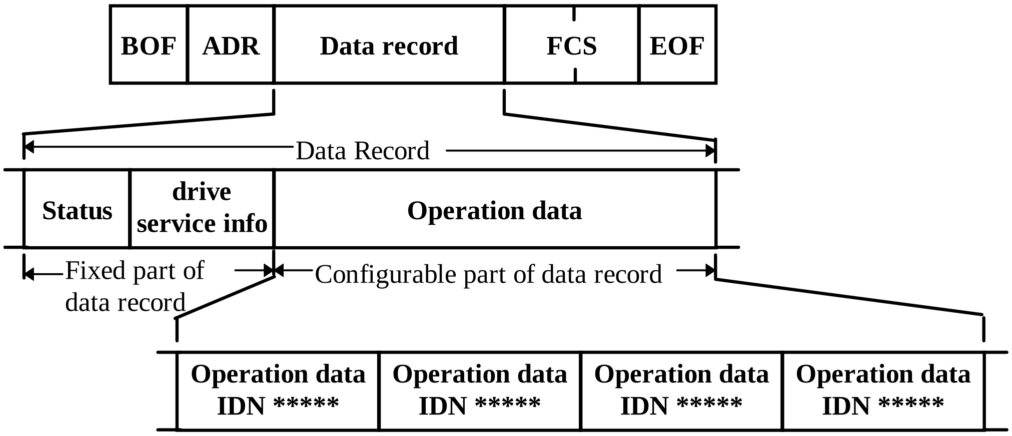


Figure 7. Drive Telegram Structure

The data record itself is composed of both fixed & variable data. The fixed data portion contains drive status information, such as whether the drive is ready to operate and other status bits. The variable portion of the data record contains the current value of 1 or more idents. For example: if the drive is operating in velocity mode, meaning it receives velocity information from the CNC, it may be configured with current actual velocity in the drive telegram. This variable portion may be up to 16 bytes long, permitting a number of idents to be transmitted on a cyclic basis.

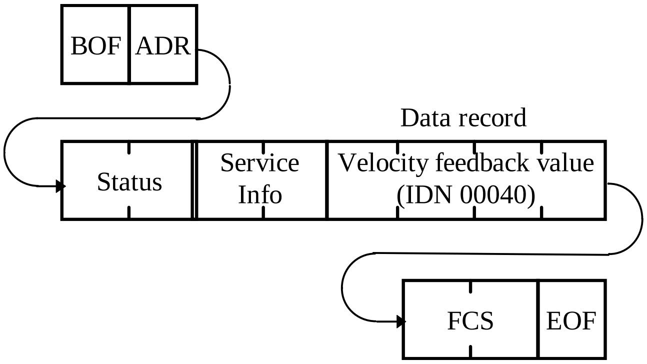


Figure 8. Drive Telegram Example

MDT Content

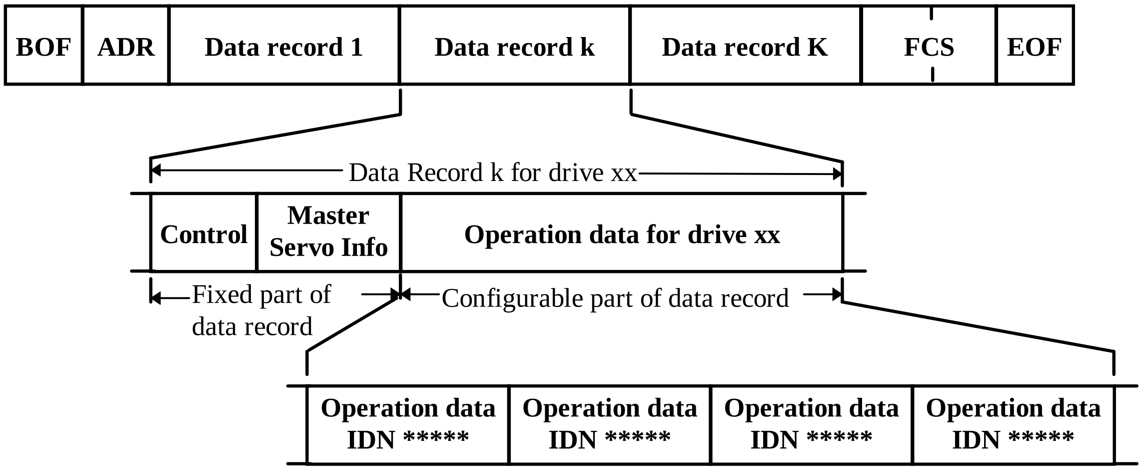


Figure 9. Structure of the Master Data Telegram

The Master Data Telegram is structured much the same as the AT, except that the data field includes a record for each drive on the ring. Like the AT, each data record in the MDT consists of a fixed and variable data portion. The fixed portion contains commands to enable, disable or halt the drive, among other things. The variable portion consists of one or more Idents of data chosen based on the application. Using the previous example, in velocity mode this data record would contain the velocity command for the drive.

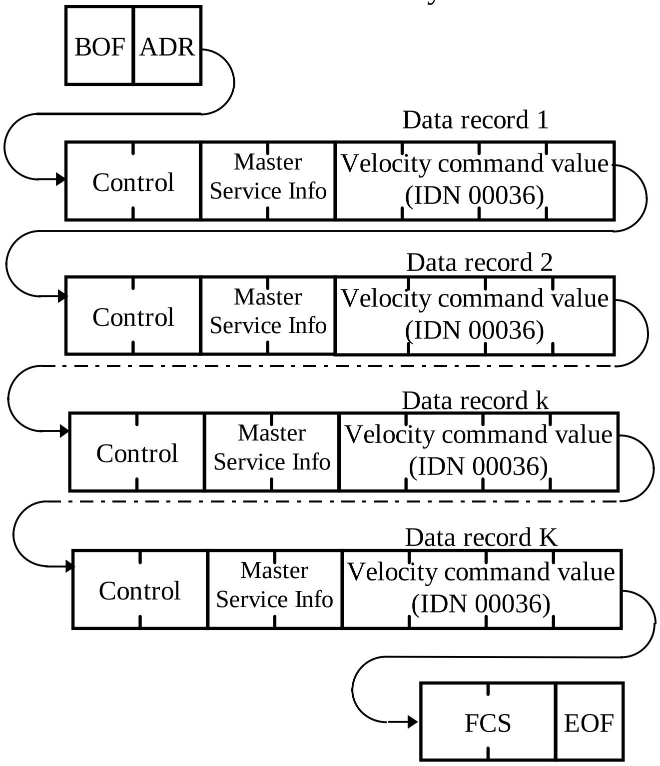


Figure 10. Example of a Master Data Telegram

Service Channel

A method for exchange of “non critical” information was next on the docket. Items such as diagnostics and loop gains did not need to be transmitted repeatedly so it was not necessary to impact loop performance by setting aside bandwidth for them. Instead two bytes of information were set aside for each drive in the ring to send non-critical information to the control. Long messages are broken into two byte chunks, sent over this service channel, and re-assembled on the other end.

Example

The best way to imagine how this system works is to take a real world example. One often used in describing the SERCOS interface involves a system with 8 drives and a two millisecond cycle rate. Under these conditions the following information can be transferred every cycle:

From control to each drive:
- 32 bit command value (e.g., velocity or position)
- 16 bit limit value (e.g., torque)

From each drive to control:
- 32 bit feedback value (e.g., velocity or position)
- 16 bit feedback value (e.g., torque)

This means that every two milliseconds one SERCOS interface ring can transport a very high resolution velocity command signal to 8 drives, plus a torque limit command to each of those drives that could be enabled at will. In addition, each cycle is returning two feedback values from each drive. This reduces the customary feedback wiring to the control to nothing.

In addition to all this real-time data being exchanged, the service channel provides what is the equivalent of an 8 Kilo-baud serial port to *each and every* drive on the ring. This can be used to set and archive performance variables, to read diagnostics, etc.

After seeing this example some individuals have concluded that the system is limited to two milliseconds or 8 drives. Neither is the case. Flexibility exists in the update rate (down to 62.5 milliseconds), the number of drives per ring (up to 254), the amount of data exchanged, and even the number of rings.

Does The SERCOS Interface Satisfy The Criteria?

A look back at the “Next Generation Interface Criteria” will confirm that SERCOS has indeed solved the issues.

1. SERCOS is open. The interface is not owned by any one company. It was designed by a consortium of suppliers. Enhancements to it are accepted by committee vote.

2. SERCOS is technology independent. It is as applicable to Brushless DC servos as it is to vector control, VFAC, hydraulic, pneumatic or stepper control.
3. SERCOS is economical. Costs to include an analog interface into a digital drive are comparable to the costs of a SERCOS interface.
4. SERCOS supports high speed in its data rates, and high resolution in its 32 bit data values.
5. Internal data access is dictated in a standardized format. A control builder can design a system that will function with any SERCOS compliant drive.
6. SERCOS is economical for a single axis installation, and provides the necessary synchronization for multi-axis systems.
7. Distributed control has received a boost from SERCOS. Control of an axis can now be reduced down to a point very close to the actuator.
8. Standardized telegrams afford the same troubleshooting capabilities we have enjoyed with analog systems, plus enhanced functions.

Does it work?

After all the hype, the question must be asked; “Does the SERCOS interface really work?” With thousands of units in the field for a number of years, the answer is easily *yes*. The first field installations of SERCOS were on high production transfer lines in the auto industry. Although this gave some the impression that SERCOS was a transfer line-only solution, the fallacy of that opinion has been confirmed as SERCOS has moved into the standard machine tool market and found it’s way into the packaging and converting industry. The distributed control capabilities afforded by SERCOS have enabled developers to control large numbers of axes with controls that are essentially nothing more than an industrial personal computer. The inherent axis synchronization of SERCOS has at long last permitted servos to replace drive shafts used in printing machines and paper and other material converting machines.

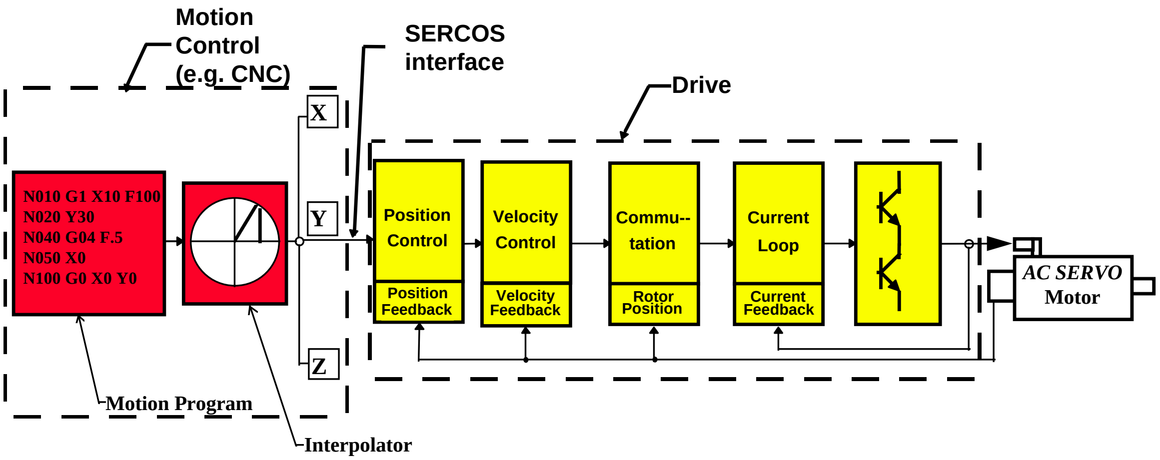


Figure 11. Intelligent Drive Topology

One of the most exciting capabilities of SERCOS lies in the position command mode of operation. Looking at a control block diagram, it can be seen that this places all the axis-specific loops in each drive, rather than in the central control. This concept has since been expanded by analyzing other control functions. Ultimately, all axis specific functions have been moved into SERCOS compliant drives. Examples:

- Probing. Many operations in machine tools involve capturing an axis' position when a signal is received. Tool or part gauging is an example. This has long been a time intensive task for a CNC to perform, as it must *immediately* capture the position of the moving axis. SERCOS compliant digital drives have been designed that route the probe signal to the drive rather than the CNC. When the drive senses probe actuation, it immediately saves the current axis position and signals the CNC via SERCOS that this has occurred. The CNC then can use the SERCOS service channel to request and receive the captured position.
- Referencing (homing) sequences. SERCOS compliant drives have had the homing routines incorporated in them, freeing the CNC from this task.
- Feed forward is an algorithm that compensates for position lag, or following error, in a classic position loop control. It is a necessity for high speed, coordinated axis machining. Because it is axis specific, feed forward can be built into a position control capable SERCOS drive.
- Reversal error compensation is a routine that overcomes lost motion caused by backlash in ballscrews and gearing. Again, being axis specific, this can be built into a SERCOS drive.
- More complex than reversal error is Lead Error Compensation. This involves a table that correlates programmed position versus actual position, compensating for inaccuracies in the mechanical drive train translation.
- Digital storage oscilloscope. Modern digital drives can include storage capabilities to capture a table of data such as position, velocity or torque without burdening the CNC. The data can then be transferred over the SERCOS service channel and displayed at the console.
- Move to positive stop. A feature popular in the high production industry, this involves moving an axis into a stall condition. With no additional hardware or software in the CNC, this feature can be included in the digital drive.

All of these functions were formerly in the control and were only developed and included if absolutely necessary for the application. With SERCOS, they can now be developed once, included in the drive, and be available to *any* control hooked to the drive. This equates to a form of “object oriented control development”!

Enhancements

Since the first SERCOS interface specification was published, a number of enhancements have occurred. Although SERCOS is easy to use and powerful, it demanded a large amount of front end development to first include in a product. The SERCOS group sponsored the development of an Application Specific Integrated Circuit (ASIC) that handles all the timing, handshaking, and configuration routines. This has reduced the front end development down to something similar to that required for a serial port or DAC.

Interpolation Mode

Work has occurred on an additional mode of operation for SERCOS. In this mode, a destination, ramp and velocity value are sent to a drive that is later triggered. Such a mode reduces the CNCs involvement in axis control even more, as it no longer must update information every cycle.

I/O Control

The SERCOS interface was designed specifically to solve the issues that occur when a serial bus is used to break high speed control loops. Such operation is not possible using other serial bus standards on the market, such as input/output busses. On the other hand, there is nothing in SERCOS to prevent it from controlling I/O. Just such enhancements have been proposed, supporting both discrete and analog devices.

The proof

As was stated at the beginning of this paper, all the control aspects of a machine tool inevitably boil down to one central focus: the control of tool or workpiece motion. In machine tools, the proof is how accurately that process takes place. As an example, XLO’s new high speed machining system, based on SERCOS products, has achieved some astounding results. Performing XY circular interpolation on a 68 mm (2.5”) diameter circle, at path speeds of ***20 Meters per Minute (800 IPM)***, a part measured accuracy of ***4 micrometers (0.00015”)*** has been achieved!

For more information

To learn more about SERCOS, contact SERCOS North America, at 1-800 5-SERCOS.