Motion Control Terminal Blocks: The Next Step in Distributed Motion Control

J. Randolph Andrews
Douloi Automation, Inc.
3517 Ryder Street
Santa Clara, CA 95051-0714
(408) 735-6942
RandyA@douloi.com

Paper Presented at the
1999 Incremental Motion
Control Systems and
Devices Symposium

Abstract

The IBM PC standard has had a great impact on motion control and machine control. The PC standard is changing as a new category of flat-panel industrial computer is gaining popularity. These industrial computer appliances are often expansion slot limited or slot-less making the solution of placing a motion controller card into the PC unfeasible. This new trend mandates the use of distributed motion controllers. Communication and controller options for distributed controllers are discussed for different controller categories including a new category, the motion control terminal block. Distributed controller requirements to speed machine development are discussed.

Introduction - Control System Partitioning

From a high vantage point, the control system design of a motion controlled machine might be considered "all that bridges the gap between an operator interface and a properly moving motor shaft." This scope of responsibility is shown in Figure 1.

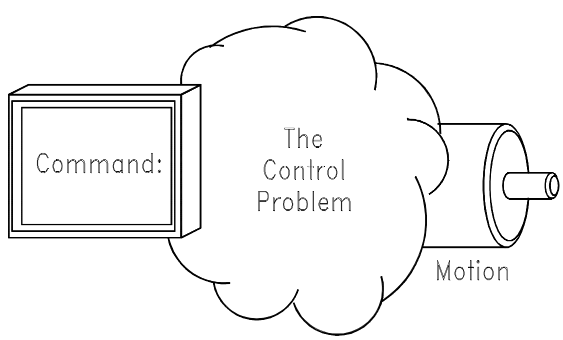

Figure 1. Control Problem Extent

Figure 1 shows an operator console on the left and a revolving motor shaft on the right. Between the console and the motor shaft are a number of elements and most likely some physical extent. Elements in between often include computers, motion controllers, motor drives, amplifiers and/or power supplies. These elements have varying amounts of intelligence and are available as resources to share the motion control responsibility.

In general, information flowing from the left side of the picture to the right side becomes less abstract and higher in frequency. This is illustrated in Figure 2.

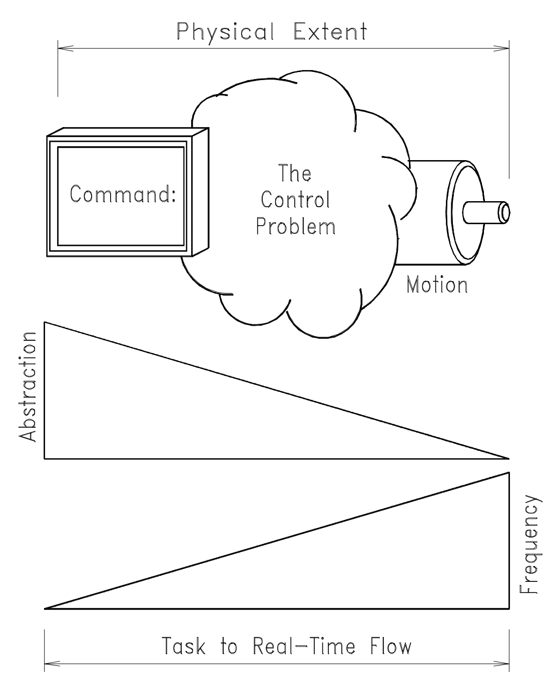

Figure 2. Abstraction/Frequency Transition

For example, a single command on the left, “perform transfer”, becomes a series of moves sent from the computer to the motion controller, a higher frequency series of coordinates representing profiled real-time motor positions, and an even higher frequency time history of voltages required to realize desired currents in the motor. The elements described do not necessarily mean there are multiple physical packages in the system. Some controllers may encompass all of these elements internally.

Motion control system design is the task of filling the bubble. Decisions being made include what elements are responsible for what jobs, where will those elements be located, and how time critical will their jobs be. Where in the system, for example, is multiaxis coordination responsibility? The time-critical question in motion control is often a challenging real-time requirement. A typical PC-Based logic control systems might make control decisions once every 10 milliseconds. Motion control has a much higher frequency real-time requirement. Motion controller sample rates are increasing from a traditional value of 1 kHz to new, higher levels such as 8 kHz.

For multiaxis coordinated machines, the flow of information in the abstract control picture includes a fork. At some point, the singular intent to move an XY mechanism requires flow of information to two destinations, the X and Y motors of the mechanism. This fork in the flow of information is being called a “coordination node”. An important aspect of the system design is the decision regarding what element contains the coordination node, and where the node is located in the abstraction/frequency transition.

Coordination requirements come in varying degrees of sophistication. A machine’s multiaxis coordination requirement might be any of the following:

- Concurrent Motion
- Coordinated Vector Motion
- Linear and Circular Interpolation
- Kinematic Coordination
- Dynamic Coordination

Concurrent motion is the requirement that independent motors be moving at the same time with no particular coordination beyond the possible need to begin motion at the same time. A group start does not create a real-time communication problem for the host because many independent axis controllers support a "broadcast" style group start command.

Coordinated vector motion requires that all axes in a coordinated group begin motion at the same time, have scaled but time-matched velocity profiles, and that all motion ends at the same time. It’s possible to produce linear interpolation with independent axis controllers. The host must calculate the necessary profile parameters to have the independent axes coincidentally achieve the appropriate ratioed velocities. Although possible, it’s not generally desirable to achieve vector coordination this way since the host program (and programmer) now have more responsibility.

Linear and circular interpolation puts a greater demand on axis coordination. This is possible with independent axis controllers but is much more difficult. It is necessary for the positions of the individual axes to be described at higher frequencies than the previous cases. By using linear or curve-fit interpolation between host submitted points, the host communication rate can be greatly reduced from the controller sample rate. Profile accuracy problems can occur if the host communication rate is too low.

Kinematic Coordination requires performing a real-time mapping between an ideal space where motion is described and a physical joint space that actually exists, most likely having a different geometry than the ideal space. Similar to linear and circular interpolation it is possible to reduce the mapping frequency to less than the controller sample rate through the use of interpolation at the motors. Beyond circular interpolation, kinematic coordination adds the additional coordination node load for the mapping function. Generally the mapping function is geometric in nature requiring transcendental math and is most conveniently done with floating point operations.

Dynamic coordination is often sensor based and includes coordination modes such as electronic camming. In this situation, an encoder driven by an external “master” represents the angular position of a cam. The controlled slave axis much inspect the value of the master axis, calculate what the cam displacement would be based on that angle, and then servo to that coordinate. This calculation is best made every controller sample period. In this coordination mode the question cannot be answered ahead of time “Where will the slave need to be?”. That question must be answered in real-time, every motion controller sample period, because the servo commanded position is sensor based. Although it is possible to use interpolation techniques to reduce the coordination node communication rate, this would degrade performance. As well as servo bandwidth, it is necessary to have profiler bandwidth. If a group of axes have electronic camming responsibility, they must all have access to the master encoder information in real-time. This level of coordination requires a coordination node with sample rate communication.

In the course of choosing system elements and delegating the coordination node responsibility, consideration must be given to the type of coordination required and what communication demands that places in the system partitioning being considered.

Historical Trends

How have motion control systems been partitioned in the past? What has historically been inside the bubble between the console and the motor shaft?

Early NC Controls

Early motion controllers were for numerical control of machine tools. Multi-cabinet, refrigerator sized NC machine controls occupied a significant portion of the total machine’s floor space. What filled the gap between console and motor shaft was effectively one very large cabinet containing all of the elements.

PLC Directed Motion

PLC based motion control solutions place motion control modules in the I/O rack of the PLC itself. The motion controller has a relationship with the PLC at the PLC scan period which might be 10 milliseconds or so. The PLC might connect to an HMI console through a distributed I/O network. Power amplifiers and drives reside in the control cabinet near the PLC.

Traditional Stand-alone Motion Controllers

Traditional stand-alone motion controllers had sufficient on-board intelligence and I/O to solve a machine control problem. In comparison to PLCs, the emphasis of a stand-alone motion controller was less on logic and more on motion. Often stand-alone controllers were programmed using sequential commands in threads like computers rather than concurrent “ladder logic rungs” characteristic of the state machine PLC model. Being stand-alone these controllers encompass most of the bubble less the console on the left and the drives and motors on the right.

Computer Based Motion Control Solutions

As microprocessors made computers less expensive and available, motion controllers moved into computers as accessory boards. Popular standards included Multibus and STD bus styles. In general there was a multi-card passive backplane “rack” into which the boards slid along with a “processor board” that sent instructions directing the motion. The HMI was the computer console, and signals from the computer might travel to amplifiers in a separate control cabinet, or possibly the same cabinet as the rack mount computer.

The computer backplane provided a high bandwidth communication link between the host computer and the motion controls. Dynamic coordination could be accomplished with this approach even with independent axis controllers at the cost of giving the host computer motion sample rate responsibilities. Although motion control cards had individual processors running control laws for the motors, coordination could be done in the host. At this point in time, processors might have had clock rates of 4 MHz. In this approach, the coordination node was the processor board.

PC Based Motion Control

The popularity of the IBM PC standard created a new focus for computer board motion controllers. The “ISA” slot was available to support a long format board able to handle multiple axes of control. Generally the computer was placed in or near the cabinet. Controller signals travelled from the computer to cabinet wiring. Although “open architecture” has come to mean many different things, at this point in history, “open” meant “open up the computer case and put the motion controller inside”.

With increasing axis count and coordination responsibility delegated to the motion controller, more powerful processors were placed on to the control card. Processor clock rates moved into the 20 MHz to 120 MHz range. In these cases, the multiaxis control cards were the coordination nodes.

Higher axis count systems became technically feasible and were desirable for a number of reasons. More axes per board meant lower cost per axis. As well, coordination relationships could be established over a greater breadth of axes because of the high internal communication rate of the motion control card which served as the coordination node. Even if coordination is not required across all of the axes, synchronization requirements often still existed relating the completion of motion for one group of motors to the start of motion for another group. However, with these advances came some new problems.

Higher axis count means more physical signals. Getting a large number of electrical signals into and out of the back of a PC is a challenge. If the computer is not inside the control cabinet, those signals need to travel to the cabinet. A centralized controller approach is practical when the computer containing the controller is central to the machine wiring. If it's not, wiring difficulties occur.

Current Trends

A new and important trend in PC based motion and machine control is the flat-panel computer. This new category of industrial computer is a flat rectangular package with a screen as one surface. A typical flat-panel computer might be 14 inches wide, 12 inches tall, and 3 inches thick. Not much larger than a flat panel display, the computer is integrated into the display providing a complete system. Inside the flat-panel computer is a variety of built-in communication ports common in PCs such as serial ports, parallel ports, and USB ports (Universal Serial Bus). Additional ports that would normally be accessory cards are also built in, such as an ethernet adaptor. The computer might have a built-in hard disk and run Windows NT, or it might have solid state mass storage and run Windows CE.

If the flat-panel computer has any expansion slots at all, they are most likely mechanically smaller than the conventional “long slot” style found in desktop PCs. Unlike their historical desktop counterparts, a flat-panel computer is not intended to be “opened up” and modified with additional hardware components. This style of computer appears to represent the beginning of the “PC Computer Appliance” concept applied to industrial computing. The computer is used "as is", as a standard configuration component.

This important new category of machine control host computer mandates a motion control solution which is not slot-based. Rather than communication from the host to the motion controller through a high-bandwidth backplane, a distributed communication approach will be required.

The objective is to continue benefitting from PC based open control solutions and continue to have quality coordination even when there’s no more slot for an internal-style control card.

Distributed Communication Options

When discussing distributed communications, it is not necessarily the case that a network solution is the only kind of solution. Point-to-point solutions, such as a conventional stand-alone motion controller with an RS-232 link, can be considered distributed in the sense that the motion controller is not internal to the PC. There are a large number of communication options available. Of these options only a small number of representative types are discussed.

An important characteristic of the communication is it's real-time determinism in relationship to the real-time need. How important is it that information arrive in a timely manner? The communication link between a coordination node and the motors must have near sample rate performance for the more advanced coordination models. The following broad categories are used to discuss distributed communication options.

Classic General Purpose Standards

In this category are placed the familiar standards not specifically related to industrial applications or motion control. Members of this category include RS-232 and RS-422. This category has the following attributes:

- *Ubiquitous* - These standards are so common that support parts are readily available. RS-232 cables can be purchased at the local office supply store.

- *Familiar* - These standards are well understood and many people know how to work with them. The IBM PC popularized certain design aspects (such as the 16450 "UART" chip). Beyond the communication standard, even the devices used to implement it are effectively standard and available from many sources.

- *Slow* - Although communication rates for these standards have been enhanced with "high speed" versions, they remain slow in comparison with other options.

- *Relatively simple* - With approximately 10 device writes it is possible to emit a character from an RS-232 port.

The determinism of this group is subject to the method of use. Both deterministic and non-deterministic standards are available.

Classical Industrial Network Standards

In this category are networking standards that were designed for automotive/industrial type applications. These include Profibus, Seriplex, Honeywell SDS, CAN-Bus and the related derivations such as DeviceNet as examples. In most cases these standards were initially available for use with PLCs rather than PCs although PCs can support them through add-on communication cards (which consume that precious single slot in a flat-panel computer). Such add-on cards might range in cost from $600 to $1500 dollars depending on the communication standard. This category has the following attributes:

- *Less Familiar* - These standards usually require more specialized skill and training than the generic standards.

- *More robust* - Because they've been designed to work in industrial environments, the electrical "layer" of the standard is more rugged and immune to noise.

- *Relatively complicated* - Although complication varies among members of the category, as a whole more expertise is required to write a device driver for a motion controller (for example) than for the classic generic standards.

Determinism for this group varies depending on standard, but is well characterized as a technical specification.

Motion Control Specific Standards

This category includes networking solutions that are specific to the needs of Motion Control. These standards specifically address sample-rate frequency communication to a set of motors and “packet sizes” that are large enough to support a coordination node remote from the individual axis controls providing “coordination at a distance”. Attributes of this category include:

- *Even less familiar* - Knowledge to develope for these standards requires specialized knowledge.

- *Well suited to motion* - These standards appreciate the real-time requirements of motion control and are designed to accommodate that need.

- *Proprietary Solutions* - Some of these standards are advocated by individual companies and are not widely supported across manufacturers.

Examples in this category include Sercos, and proprietary solutions many of which are fiber optics based.

PC Based Standards

Beyond the standard of the IBM PC itself is a set of standards for related technology. In particular, PC based networking technologies are being embraced by industrial users. The characteristics of PC based networking solutions include:

- *Low cost or free* - Communication ports such as USB come built-in to modern computers. Ethernet cards are very inexpensive and often built in to the flat-panel computers.

- *Very high speed* - A slow ethernet card is still very much faster than most of the industrial networks.

The two most important examples in this category include ethernet and USB. Ethernet was initially criticized as being unsuitable for industrial applications because it was not deterministic and not as electrically rugged as the industrial network alternatives. In response to the determinism question, ethernet advocates had two responses. The first is that by being significantly faster, determinism is less important since even a delayed packet of information will arrive on ethernet sooner than on a slow-but steady industrial network alternative.

Secondly, it’s possible to use ethernet hardware standards with alternate control software than is used in business settings. Rather than a multi-point topology as might be used in an office network, ethernet can be used in a point-to-point master/slave topology. By eliminating contending devices on the network, the arbitration delays and non-deterministic aspects of ethernet can be eliminated providing a truly deterministic solution. The second response to the objection of electrical ruggedness has to do with what constitutes an industrial environment. Particularly in the semiconductor industry where automation is being deployed in clean rooms, the “industrial environment” of a clean room can be much friendlier than the front office. Ethernet I/O and motion controllers have become popular in the last year with major vendors and is being appreciated as a fast, low-cost, reliable, and maintainable approach to industrial communication.

Although promoted by a strong group of companies including Compaq, Intel, Microsoft, and NEC, The USB standard (Universal Serial Bus), has been slow in adoption. Nevertheless, it appears to be the future of PC expansion for flat-panel computers and similar computer appliances which have no slots. Even where slots are available, USB has a role as a viable expansion standard for all PCs being less intrusive.

In many respects, USB is not as well suited for motion control as other standards. The “frame rate” of a communication transaction appears to be about 1 millisecond. This is relatively slow, however the information that arrives can be a relatively large block. USB can support different communication models. Of the models most apparently suitable for motion control, one is not deterministic, and the model that is deterministic does not guarantee data integrity. Although data loss is acceptable for applications such as video or audio, data loss is never acceptable in a motion control application. Because of these limitations it is likely that a USB approach would not be suitable for a sample-rate frequency coordination node. A USB motion controller will have to have a higher level of coordination intelligence to meet current performance expectations of machine builders.

The other objection to USB is that the cable length limitation is approximately 5 meters. Unfortunately this is much shorter than the length of many machines where distributed control would be valuable. However 5 meters is long enough to get from a flat-panel computer to a control cabinet. Despite these limitations, USB is important now and will become very important in the next two years.

Distributed Controller Options

Given that there is a distributed communication cable available, What motion control solution can be placed on the other end of the cable? The following three broad controller categories include Smart Drives, Traditional stand-alone multiaxis controllers, and Motion Control Terminal Blocks.

Smart Drives

Smart Drives are the evolutionary result of advances in motor amplifier technology. No longer limited to simple torque or velocity modes, smart drives are able to perform independent motion patterns, synchronize motion with built-in I/O, and communicate to a host computer through a command set. Generally a smart drive will run one motor. Groups of smart drives can relate to each other most often through a Motion Control Specific network although coordination performance has historically been less than multiaxis controllers can provide.

For small axis count systems, such as 1 or 2 axes, smart drives can offer a more cost effective solution than multiaxis controllers with "dumb" and less expensive drives. When the number of coordinated axes is three or greater, a multiaxis controller approach offers better performance and is more cost effective.

Traditional Stand-alone Multiaxis Controllers

Stand-alone multiaxis coordinated motion controllers are generally "bookend" format boxes with flange mounts that bolt to the back of the control cabinet. In most cases, the controllers have on their front panel connectors for the different signal groups, such as D connectors or higher density connectors which are now available. Plugged into these connectors are cables that go from the motion controller to DIN rail mounted terminal blocks. These terminal blocks can be located at various locations inside the control cabinet to present screw terminals for interconnection to drives, sensors, power, and the other wiring required for integration.

Generally, but not always, these multiaxis controllers are intelligent but have no power amplifier muscle internal to the controller itself. These controllers serve as the coordination node in the control system and have the ability to be downloaded with independent control behavior. Although described as stand-alone, there is often a PC host involved to support an HMI or other services. Because of the local intelligence in a stand-alone controller there are less real-time communication requirements to the host computer. Many applications can be solved with the controller itself. In some industries, a PLC is included in the system design whenever there is I/O to be managed. Modern stand-alone controllers provide I/O as well as motion resources which in many cases eliminate the need for the PLC.

Stand-alone Motion Controllers are more expensive than their equally intelligent but unpackaged PC-Slot counterparts by generally a factor of 2. The cables and terminal break-out accessories required to conveniently wire the motion controller is an additional cost for both the stand-alone and the PC-Slot formats.

Motion Control Terminal Block

The category of Motion Control Terminal Block represents the evolutionary development of stand-alone multiaxis controllers. In this scenario, the bookend controller, cables, and DIN rail mounted terminal blocks are replaced by a motion control terminal block which is approximately the same size and shape as the passive terminal block element it replaces. However, instead of being one component used to connect through a cable to the motion controller, the smart terminal block is the motion controller all by itself. Attributes of a Motion Control Terminal Block include:

- *Small-Footprint* -The small size allows the Motion Control Terminal Block to occupy not much more space than the passive terminal block it replaces.

- *Multiaxis* - Because many of the more desirable communication standards do not have sample-rate performance, we need to look to the Motion Control Terminal Block to be the coordination node.

- *Din-Rail mounted with screw terminals* - By having screw terminal points directly on the controller, and having the controller in the normal position for accessing the wire harness, the extra parts and cost of cables and terminal blocks is eliminated and space is saved as well.

- *Multiple communication channels* - For reasons discussed below it is very desirable to have several communication channels in the controller to support diagnostics during actual machine operation.

- *Integrated I/O* - It's desirable to provide opto-isolated inputs and isolated high current outputs to eliminate the additional components and cost of a separate I/O rack.

- *I/O Expansion* - For systems where the built-in quantity of I/O is not sufficient, the I/O system should be expandable.

Although not defining attributes of a Motion Control Terminal Block, it is desirable to support multiple communication standards so as to flex with the networking standard chosen for the system.

The enabling technologies to support the Motion Control Terminal Block model are highly integrated “PC-on-a-chip” components, FPGA based high-density axis support hardware, and continued size reduction available through smaller surface mount parts.

The Motion Control Terminal Block approach reduces wiring complexity and cost while providing local intelligence for coordination.

Distributed Control Requirements

What’s required to effectively develope and deploy distributed motion control systems? The single most important characteristic a distributed motion controller should have is multiple concurrent communication channels. Traditionally stand-alone motion controllers have had one command interpreter, or one command interpreter and a separate interpreter for on-board programs. Although on-board programs can run a machine, it's more common for the controller to only take on a part of the control responsibility and also receive commands from a host. When a host is involved, it is very important to be able to see what the host is asking the motion controller to do. Even more important is having a “Window” into the motion controller during operation of the machine so as to be able to monitor in actual live operation the dynamic motor response, monitor I/O, and measure performance. This greatly aids diagnostics and development.

A second important characteristic of a distributed motion controller is the ability to be maintained remotely. Maintenance can include application software updates, firmware updates, and in the case of certain types of FPGA based controllers, hardware updates. Ideally, a controller revision of a motion application, controller firmware, and controller hardware can be emailed along with a software utility to perform the upgrade to a user anywhere in the world, or downloaded and used from an internet site. It can be very inconvenient to access motion controllers once installed in a system. Particularly in environments such as clean rooms, even the activity of opening up a PC cabinet can be an environmental contaminate. Controller upgrades need to be as non-intrusive and simple as possible.

A third requirement is integrated software tools. In the past, a traditional stand-alone controller might have a serial port, a description of the port settings, and a command list. It would be the responsibility of the user to come up with a way to create a command and send it over the serial cable to the controller. The sophistication of many of the network standards makes this one-sided approach no longer adequate. Motion controller vendors must accept responsibility for software on both ends of the cable, or work with third parties, such as PC-Based Control companies, to support the construction and transmission of commands over networks that are supported. Although the communication standards themselves may be complicated, their use should not have to be. The topology might be distributed, but from the control designers point of view all required resources should be together in one environment already integrated and ready to use.

Summary

Trends in slot-less appliance PCs, such as flat-panel computers, make a distributed motion control system more necessary and more beneficial than a traditional centralized control approach. The warning “PC ISA Slots are Going Away” is understated. For some computer categories all internal PC slots are going away. Accordingly the future of PC Based motion control will rely heavily on distributed control technologies and integrated software tools. The Motion Control Terminal Block provides the required distributed controller attributes in a coherent, complete, and cost effective package.

Bibliography

1) Think & Do Software Learning Guide, Manual Number PC-TND-M, 2nd Edition, Think & Do Software, Inc., 8/98

2) Instruction Manual for Motion Server & Binary Command Interpreter, Douloi Automation Inc., Santa Clara CA, February 1999

3) Instruction Manual for Motion Server & Servo Application Workbench, Douloi Automation Inc., Santa Clara CA, March 1999

4) Andrews, J. Randolph: "Motion Server - A Next Generation Motion Controller Architecture”, In *Proceedings of the Twenty Fifth Annual Symposium on Incremental Motion Control Systems and Devices, San Jose, CA*, 1996.

5) Cox, Brad: *Object Oriented Programming: An Evolutionary Approach*, Addison-Wesley Publishing, 1986, 1991

6) Ellis, George: *Control System Design Guide*, Academic Press, San Diego, 1991

7) Franklin, Gene & Powell, David: *Digital Control of Dynamic Systems*, Addison-Wesley, Massachusetts, 1981

8) Meyer, Bertrand: *Object-oriented Software Construction*, Prentice Hall, New York, 1988

About the Author

J. Randolph Andrews received his B.S. M.E. in 1981, B.S. E.E. in 1981 and M.S.M.E. in 1983 from the Massachusetts Institute of Technology.

Andrews spent 4 years at Hewlett Packard's corporate research laboratory in the Applied Physics Research Center as well as the Manufacturing Research Center.

The following 4 year period was spent with Galil Motion Control.

In July '91 Andrews founded Douloi Automation, Inc. to provide motion control hardware and software solutions for use with Microsoft Windows and Windows NT.

Professional interests include motion control, software/electrical/mechanical system design trade-offs, high abstraction programming and visual programming techniques and tools.