Controller Interfaces for Robotic Surface Finishing Applications

Presented at: The 4th Annual RIA Grinding, Deburring and Finishing Workshop
Novi, Michigan October 1998

Lester E. Godwin
Vice President
PushCorp, Inc.

Introduction

Creating an effective automated processing work cell from the myriad of component equipment available today is, to say the least, a daunting task. From a controls technology standpoint, there are three major design decisions to consider. First one must decide what processing equipment will be necessary to accomplish the task at hand. Second one must choose a control strategy to monitor, control and synchronize the process equipment. Finally, an operator interface scheme must be chosen to allow an operator to effectively run and monitor the work cell. The resolution of each of these design decisions directly impacts the cost, performance, and usability of the final, complete work cell.

The purpose of this paper is to provide information so that an integrator or end-user can effectively answer the three questions above. In doing so, options regarding interfacing process equipment, control strategies, and operator interfaces will be examined and differentiated with respect to their respective strengths and weaknesses. It is hoped that, with this information, an integrator or end-user can more easily decide on an effective control scheme in designing an automated work cell.

Process Equipment

The process equipment used in automated surface finishing work cells typically falls into four categories: the manipulators, end-of-arm tooling, fixtures, and safety equipment. Each piece of equipment in these categories has its own set of sensory and control requirements. These sensory and control requirements as well as the process equipment requirements must be evaluated with respect to cost and viability of implementation.

Manipulators

Manipulators in an automated surface finishing work cell generally serve to either manipulate a part over a piece of processing equipment (i.e., Part-in-Hand processing) or manipulate a finishing tool over a part (i.e., Tool-in-Hand processing). In either of these cases the manipulator can be either dedicated fixed automation or a flexible robotic motion platform. The sensory and control requirements for either of these manipulators involve monitoring and controlling the manipulator’s motion status so that it can be coordinated with other pieces of processing equipment. For example, a grinding motor must be activated at a specific point in space to polish a part or a gripper must be actuated to grasp a part to be ground. In this case the signals can be simple digital on/off signals toggled at the appropriate time by the manipulator controller.

Fixed automation manipulators, are most often controlled by either a small, dedicated, local controller or directly controlled by the central work cell controller. When the work cell controller controls the manipulator, the work cell controller is burdened with every detail of the lowest level of the manipulators operation. This is often unacceptable because manipulators demand fast, real-time responses to feedback signals, and the high-level work cell controller has insufficient resources to quickly react. As a result, a small, high-speed local controller is often used as the manipulator's controller. This small controller handles the real-time requirements of the manipulator and interfaces to the work cell controller through a lower speed, high level interface that serves to synchronize the manipulator with the other process equipment.

The ultimate example of a local manipulator controller is the controller of a robotic manipulator. The robot controller is designed to handle the demanding real-time requirements of robotic servo control. However, as computer technology has progressed, some robot controllers now have sufficient excess resources and flexibility to serve as the central controller for the entire work cell.

End-of-Arm Tooling

A robotic manipulator is basically useless without some sort of end-of-arm tooling (EOAT). For Part-in-Hand surface finishing applications, the EOAT may be some sort of gripper mechanism to hold a part. For Tool-in-Hand applications the EOAT is often an active or passive compliant device with an attached motor to perform the finishing operation.

The control interface for the Part-in-Hand gripper usually involves simple digital I/O for monitoring proximity switches and controlling pneumatic valves or electrical actuators. Proximity sensors are used to monitor the gripper's status as well as to verify that the part is present and secure. With Part-in-Hand applications most of the control complexity lies in the interface to the stationary buffing and/or grinding equipment.

Backstand tooling sophistication can vary from a relatively simple single set-point manually adjusted compliant device with a single speed pneumatic motor to a fully active, servo-motor driven precision tool. The control interface complexity will also vary accordingly. Since the manual backstand has only a single compliant force setting, the only required control interface would be digital I/O to turn the electric or pneumatic motor on and off.

Tool-in-hand applications utilize an active or passive adjustable force device to control the media contact force on the part. This arrangement is used where the part is too large to be manipulated in a reasonable manner. Therefore an active adjustable force device requires a control interface to adjust the force during the process. The passive device uses analog signals to set and vary the pressure in the pneumatic actuator. The active device has slightly more demanding interface requirements. Its interface can be an analog, digital or RS-232 serial interface. An analog interface provides a simple and direct means to adjust the device's compliant force. With an analog interface the resulting compliant force is proportional to the input voltage which usually ranges from minus to plus 10 volts DC. While the analog interface is electrically and conceptually simple to implement, it does require that the work cell have an digital-to-analog (D-to-A) output module so that the analog voltage can be controlled programmatically. In fact for the passive device, two modules are required and calculations are necessary to determine the proper output voltage. These interface modules tend to be rather expensive, so many work cell implementations use a digital interface exclusively since most robot controllers come with digital I/O included.

Using a digital interface to an active force device allows one to select one of several pre-configured force levels by turning on or off a series of individual I/O lines. By using standard digital I/O, any PLC or robot work cell controller can control the compliant force with inexpensive and readily available digital output modules. The only drawback to this control interface method is that only a given number of force levels are available for selection at any given time – as opposed to the analog interface where any possible force can be commanded at any time.

For example, the digital interface available on the PushCorp, Inc., FCU100 active force device controller is comprised of three digital inputs. These three inputs, taken together, form a 3 bit binary number ranging from zero to seven. (See Table 1.) As one can see the combination of bits determine which force level is chosen. A special case is, when all the bits are OFF, force level “one” is selected by default. The compliant force associated with each force level is configured separately so that force level “one” represents a specific desired compliant force. For example, force level “one” may set a force of 10.1 pounds, force level “two”: 5.7 pounds, force level “three”: –21.9 pounds, etc. The force levels are configured via the controller’s front panel (FCU100-1) or the serial port (FCU100-3) with the aid of PushCorp supplied software.

Bit 2	***Bit 1***	***Bit 0***	***Force Level***
OFF	OFF	OFF	1
OFF	OFF	ON	1
OFF	ON	OFF	2
OFF	ON	ON	3
ON	OFF	OFF	4
ON	OFF	ON	5
ON	ON	OFF	6
ON	ON	ON	7

Table 1. Converting Binary Bits to Force Levels

The final available interface for active force tools is a RS-232 serial interface. A serial interface is very flexible and allows a great deal of information to be exchanged with the device. Because of this, the serial interface is often used to configure the operation even if other interfaces are used for control during part processing. For the PushCorp FCU100 force tool controllers, the serial interface is capable of configuring and controlling every aspect of the device. The drawback to serial interfaces is that most PLC and robot controllers do not easily support serial communications without special programming. Therefore the serial interface is most often used with a PC compatible computer acting as a temporary configuration platform or in some cases as the central work cell controller.

To use a PC as a configuration platform, PushCorp interface software running in the Microsoft Windows™ environment is installed on a portable or laptop computer. This computer is then plugged into the active force tool controller and the user configures the force levels, position limits, etc. to whatever is needed for the finishing application. Once the proper configuration is obtained, the laptop is unplugged and the active force device is subsequently controlled from either the analog or digital interfaces. This mode of operation has several advantages. First, a single portable computer can be used to configure several work cells. Secondly, using a portable PC to configure the active force tool provides some degree of security by preventing unauthorized personnel from independently adjusting configuration parameters. Finally, using the serial port for configuration allows one to use the flexible serial interface for quick and easy application setup while still allowing the use of the easier-to-implement analog or digital interfaces during actual part processing.

It should be noted that, with any of these interfaces, the force applied by an active or passive tool can be changed at any time. This means that one has the ability to change forces "on-the-fly" as the robot moves through its programmed path. This opens up a whole range of processing capabilities where the force can be changed based on part features or some sort of process feedback.

Most finishing applications require some type of motor to work the abrasive media. In the past pneumatic motors have been widely used. But more recently, servo motor technology has become very popular due to its clean, efficient, and low maintenance performance. While simply opening or closing a valve can control pneumatic motors, servo motors require electronic controllers which share many of the same control interface characteristics as the active force tools.

Typically servo motor systems are composed of the motor itself, a servo amplifier, servo controller and power supply. Given the advances in electronic integration over the last several years, many or all of these components may be combined into a single servo control unit. Interfaces to the servo control unit include digital inputs to enable the drive system and digital outputs for various status and fault signals. To control the motor speed a ±10 volt DC analog interface is used where the velocity of the motor is proportional to the input voltage. Again, similar to the active force tool, many of the newer servo controllers have an RS-232 serial interface, which is used to configure the servo system.

Fixtures

Fixtures are pieces of equipment whose function is to firmly hold either the part being processed or processing tools themselves in place. Tool fixtures hold disk pads, abrasive wheels, router bits, drill bits, or chamfering tools so that the robot can reliably pick up and replace the tools. These fixtures can be relatively simple trays to cradle and present the tools or the fixtures may be more complex tool changing mechanisms such as the PushCorp Disk Manager which actively removes a spent abrasive disk and inserts a new one. A part fixture holds a part to be processed securely in position so that the robot can move in a consistent manner from part to part.

Most simple fixtures have at most, sensors to verify that the tool or part is in the correct position. These sensors are often digital proximity sensors due to their ruggedness and reliability. More complex fixtures may have some control requirements to actuate clamps or similar type capture mechanisms. The most complex fixtures, such as the PushCorp, DM100 Disk Manager, may have significant control requirements to handle the tool change sequence or the shuttling of a part into and out of the work cell. In these cases a small, local PLC controller may be used to perform the low-level fixture control. This local PLC is, in turn, communicates to a central work cell controller through a high level digital interface or, as is becoming more common, a distributed I/O network such as DeviceNet, Genius Bus, ModBus Etc.

Safety Equipment

The final major category of equipment to be interfaced with the control system is safety equipment and circuitry. Operator safety is the foremost consideration in designing any automated work cell and incorporating safety related issues should at all levels of the control system. However the difficulty in interfacing safety equipment to the work cell controller is that the controller cannot directly control much of the safety system logic. According to electrical codes, the work cell safety mechanisms should be fail safe and not depend on any intermediary computer controller.

Work cell safety equipment is composed primarily of work cell barriers to keep personnel out of the workspace and emergency stop circuitry to quick bring the work cell to a benign state. These two elements often must interact in a complex way during operation. Since the work cell controller cannot be involved, this interaction must be accomplished with relay logic. Further complicating matters is that the relays used to implement the safety logic must be guided contact, fail-safe safety relays. All this combines to make the safety equipment integration very complicated and costly.

The safety equipment's interface to the work cell controller is primarily a one-way street where the controller monitors the system's state. When an operator initiates an emergency stop, the hard-wired safety logic operates to immediately stop all moving elements of the work cell. In the case of motorized manipulators, power is removed from the motors and brakes are applied. The central work cell controller's primary duty is to reset itself to a known safe state so that the work cell can be re-started. Since an emergency stop can be initiated for most any reason, it is often difficult but also very important that the controller verify that the equipment is indeed in a known safe state before the re-start can proceed.

Controller Implementation Strategies

Once the appropriate hardware is selected for an automated work cell, one then needs to decide on a control implementation strategy to integrate the equipment into a complete system. The controller implementation strategy includes assigning each piece of equipment to either a local controller or central work cell controller, deciding on the most effective control interface for each piece of equipment, and finally deciding how the operator will interface with the work cell.

Robot / PLC Integration

With the current state-of-the-art in controller technology there are a number of ways to implement any work cell. The overlap in functionality between robot controllers, PLC controllers, and PC compatible controllers is becoming greater and greater so that it is becoming difficult to go wrong in choosing any of these technologies. However, there are still situations where each technology is particularly adept. The following sections describe some controller strategies showing how the various technologies may be most effectively used.

Robot Controller as Sole Work Cell Controller

Robot controllers, like most other computer equipment, continue to get faster and more capable with every model. Five years ago robot controllers barely had sufficient power to manage just their servo system. Now, multitasking robot controller can perform complex motion trajectories and handle a significant amount of control logic for peripheral equipment. While robot controllers cannot match the scan times of most PLCs, for small work cells with limited, low-speed I/O control requirements, using the robot controller as the sole work cell controller is definitely a viable option. This setup can simplify work cell implementation since all wiring comes to a central point, and all software development is performed on a single platform.

Robot Master with PLC Slave(s)

If one finds that the robot controller has insufficient power to handle specific pieces of peripheral equipment, then one might consider using one or more small PLCs as local controllers. With this setup, the robot controller serves as the central work cell controller, performing the primary work cell logic sequencing. The robot controller is connected to the peripheral equipment local controllers either through a limited number of digital interface lines or possibly even a high-level control network such as DeviceNet or CanBus. The local controllers function in a slave mode to the robot controller and monitor and control the peripheral equipment. This arrangement relieves the robot controller of the potentially demanding control details of managing the peripheral equipment.

A prime example of this strategy is an automated finishing work cell with a robot and a PushCorp, DM100 Disk Manager to automatically change out the abrasive disks. In this cell, the Disk Manager has a micro PLC installed to control the detailed operation of the changer. The robot in turn controls the Disk Manager through eight control lines. The control lines to the Disk Manager serve to initialize the unit and initiate a change cycle. Control lines returning to the robot from the Disk Manager indicate status information such as change cycle complete and error conditions. The limited I/O to and from the Disk Manager allows one to control the changer at a functional level rather than having to deal with more complicated, low-level logic to monitor individual sensors and actuate motors.

PLC Master with Robot Slave(s)

Another control strategy is to have a PLC acting as the central work cell controller with one or more robots operating as slave local controllers. This control strategy is often used in production line operations that involve several robots working together. The PLC work cell controller orchestrates the overall cell operational logic and sends the robot controllers high level commands such as “Move to Home”, “Start Cycle”, etc. The robot controller executes these high level commands and responds with status and error information. Often the robot controller is responsible for controlling its EOAT because of the required synchronization with the robot motion. This arrangement forms a true hierarchical control system.

Distributed Control

Contrasting the hierarchical control system is the distributed control system strategy. In a distributed control system, all the controllers in the system, whether PLC, PC, or robot controllers have equal status in terms of control. This control strategy is often used in very large installations where there are a number of somewhat isolated control stations. These stations are networked together via high-speed data link and operate relatively independently of one another. The advantage of distributed control is that the independent workstations can be implemented and changed relatively quickly without impacting the other workstations. Also, distributed control reduces the need to bring all the wiring back to a central termination point, which reduces wiring complexity and cost.

Operator Interfaces

The final control interface decision to be made is regarding the operator interface. There are entire books on good operator interface design, however a good interface boils down to providing the operator with clear and timely cell status information. The amount of information should be limited to that which is most important at any given time while allowing the operator the option to view more detailed status reports. Likewise the operator controls should be limited to only those which are relevant at a given time. The controls should be clearly labeled and should in every case possible not allow the operator to injure themselves or the equipment.

Given these requirements, the choice of operator interface depends on a number of factors including cost, work cell complexity and preference. Often it is desirable to have a “one-button” interface solution. The “one-button” solution implies that the work cell controller should be capable of detecting and correcting any possible problem situation. Therefore the level of programming and cost for the “one-button” is prohibitive for all but the most trivial work cells.

	One-Button Interface	Multi-Button Panel	GUI
Reconfigure Interface	Easy	Difficult	Easy
Software Cost	Very Expensive	Least Expensive	More Expensive
Hardware Cost	Least Expensive	Most Expensive	More Expensive
Access to Information	Bad	Good	Excellent
Operator Training	Little	Extensive	Little

Table 2. Comparison of Operator Interface Strategies

Moving up from the “one-button” solution is the hard-wired control panels commonly used for the last fifty years. While these interfaces have served well for relatively small systems, they become expensive to implement and all but impossible for the operator to comprehend on large, complex work cells. The reason for this is that these panels are incapable of adapting to different situations during cell operation. These interfaces present the operator with all status information and all possible operator input choices all of the time. Therefore the operator often finds himself overwhelmed all the flashing lights and buttons and finds it difficult to take appropriate action.

This information over-load as well as flexibility are the prime motivations of the move toward Graphical User Interfaces (GUIs) for work cell controllers. The GUI conveys status in a clear manner that can be adapted to present the operator only the most pertinent information at any given time. Using various input devices such as touch-screens, light pens, and mice the operator can easily opt to view more detailed information.

Another benefit is that any sort of gauge, switch, lever, analog slider, or indicator can be quickly reproduced on the operator display. These input devices can be displayed or hidden at any time to limit the operators choices only to those that are valid for the current work cell state. All this combines to make the work cell’s operation much more easily understood by the operator and make operator errors much less likely.

Conclusion

As stated in the introduction, creating an effective automated surface finishing work cell from the myriad of component equipment available today is truly a daunting task. First one must decide what manipulators, end-of-arm tooling, fixtures, and safety equipment must be used to accomplish the process. Second one must decide on how to implement the control strategy to monitor, control and synchronize the process equipment. Can only the robot controller be used? Is one or more PLCs required? What about distributed control? Finally, a user interface scheme must be chosen to allow an operator to effectively run and monitor the work cell.

Several options for each of these questions has been examined and differentiated with respect to their respective strengths and weaknesses. With this information, an integrator or end-user should be well armed to begin the implementation of a successful robotic automation system.

Related Publications

1 Lawley, T. J. " The Automation of a Contour Surface Grinding System: Controls Problem Definition. " 1991.

2 Lawley, T. J., Erlbacher, Edwin A., "Robotic Grinding of Parts to Remove Machine Tool Scallops and Mismatch," Maintaining and Supporting an Aircraft Fleet Conference, Dayton, Ohio, June 1992.

3 Graf, Tim, "Practical Methods for Robotic De-burring and Finishing Applications," Conference Proceedings Volume 2, International Robots & Vision Automation Conference, Detroit, Michigan, April 1988.

4 Erlbacher, E. A. " Robotic Surface Finishing Cell for Contoured Scalloped Parts," Ph.D. Dissertation, University of Texas at Arlington, 1992.

5 Davis, James C., "Path Planning Methods for Robotic Grinding of Complex Contoured Parts," M. S. Thesis at the University of Texas at Arlington, 1993.

6 Godwin , L. E., “Design and Implementation of a Modular Object-Oriented Integration Architecture for Real Time Systems,” ," Session Twenty of the International Robots & Vision Automation Conference, Detroit, Michigan, April 1993.

Bibliography

1 National Center for Manufacturing Sciences, Automatic Inspection Systems for Edge and Surface Finishing, Technical Report NCMS-88-MPM-9, April 1989.

2 Whitney, D.E., and Tung, E.D., " Robot Grinding and Finishing of Cast Iron Stamping Dies," ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 114, March 1992, pp. 132-140.

3 Reinhart, Robert W. " Deburring and Polishing - A New Frontier For Robot Application, " Technical Paper MS82-129, Society of Manufacturing Engineers, 1982

4 Graf, Tim " Deburring, Finishing, and Grinding Using Robots and Fixed Automation: Methods and Applications, " In Proceedings International Robots & Vision Automation Conference, pages 20.1 - 20.22, 1993.

5 Backes, P.G., Zimmerman, W., Leahy, M. B., " Telerobotics Application to Aircraft Maintenance and Remanufacturing, " submitted to IEEE ICRA, 1993.

6 Coons, S.A., " Surfaces for Computer-Aided Design of Space Forms, " M.I.T. Project MAC TR-41, Cambridge, Mass., June 1967.

7 Murphy, Karl N., Proctor, Frederick M. " An Advanced Deburring and Chamfering System, " In Proceedings, Third International Symposium on Robotics and Manufacturing, July, 1990.

8 Turner, Ed, Red, W. Edward, " Inaccuracy Mapping of 2-D Surfaces for Robot Off-Line Programming, " System Automation Laboratories, Bringham Young University.

9 Luckemeyer, James A., Franke, Ernest A., " Robotic Inspection and Refurbishment of Aircraft Canopy Transparencies, " Proceedings, Maintaining and Supporting an Aircraft Fleet, Dallas, 1992.

10 Shukla, Deepak, Bagchi, Amit, Paul, Frank, " CAD Based Simulation for Polishing of Two Dimensional, Polygonal Components, " Proceedings ASME - WAM Dallas, TX, November, 1990.

11 Craig, John, *Introduction to Robotics* Addison-Wesley Publishing Company, 1986.