AN RCS APPLICATION EXAMPLE: TOOL CHANGING ON A HORIZONTAL MACHINING CENTER

John C. Fiala, Albert J. Wavering

Robot Systems Division, National Bureau of Standards
Gaithersburg, Maryland

Abstract

A large six-axis manipulator performs material handling tasks for a horizontal machining center in the National Bureau of Standards' Automated Manufacturing Research Facility. The manipulator is controlled by the Real-time Control System (RCS), a hierarchical control system developed by the Robot Systems Division at NBS. This manipulator is used to load and unload parts for the machine and, recently, to change tools on the machine. This paper describes the development of tool changing as a new RCS application, including the creation of new commands to be sent to RCS from a workstation controller and the techniques used to program manipulator movements. The integration of force sensing to monitor tool insertion and removal as well as the addition of a changeable end effector for handling tools are discussed.

I. Introduction

Development of the Horizontal Workstation (HWS) began in 1981 as the first of the Automated Manufacturing Research Facility's machining workstations [1,10]. Today, the HWS is a highly flexible subsystem that has the ability to change the tool set-up of its machining center. The HWS consists of a horizontal machining center (HMC), an industrial robot, programmable fixtures for holding workpieces [11], and a material buffering device (MBD) for interfacing the workstation to the material handling system [9]. All of the equipment is coordinated by the Horizontal Workstation Controller (HWSC) [7,8].

The industrial robot, a five-thousand pound, six-axis, hydraulic manipulator, is controlled by the Real-time Control System (RCS) [15]. A brief overview of RCS is given in Section II. In the HWS, RCS is mainly responsible for effecting the loading of part blanks from the

MBD to the fixtures, the refixturing of partially machined parts, and the unloading of finished parts into the MBD. It has been noted, however, that a limitation with workstations is that the tool magazine of the machining center can hold only enough tools to make a few parts [14]. This greatly reduces the workstation's flexibility. To overcome this limitation, the robot can be used to automatically reconfigure the workstation by changing the tools in the magazine.

This paper describes the development of tool changing as an RCS application. Section III discusses the hardware developed for tool loading and shows how it is easily integrated into the system. The addition of new commands to the controller is discussed in Section IV. Finally, Section V describes the control data generated for the new application.

II. The Real-time Control System

RCS is a Robot Systems Division product of the National Bureau of Standards [2,3]. The system can be described as a collection of control processes organized in a hierarchy. These control processes execute non-sequentially in a distributed computing environment. The processes communicate through fixed-field, common memory buffers.

A control process in RCS is a functional component that communicates with other processes in the system only through explicitly-defined interfaces. The principle control structure of RCS is shown in Figure 1. Task, Subtask, E-move, and Prim are the processes involved in decomposing commands from the workstation into simple actions. The gripper control processes interface the end effectors [6,12] to the controller. These end effectors are connected to and disconnected from the robot by the quick change device [5].

The control processes of RCS are distributed among various processors. Most of the system executes on a set of processors residing on a single bus as shown in Figure 2. However, the vision system and active pedestal are complex enough to require their own hardware systems. RCS sends commands to the pedestal via the factory network [13] and makes sensory processing requests of vision via the high-speed 589 DMA interface. The division of the computational work load between several processors ensures that every process can complete one cycle of execution in the time between updates to the robot controller.

The lowest level of RCS, the Robot Interface process, transmits position updates to the robot controller every 40 ms. The Prim(itive) control process computes the knot points of trajectories and sends them to the Robot Interface. Prim also processes the sensor inputs appropriate to its operation, e.g. joystick inputs. The E(lemental)-

move process coordinates its subprocesses, i.e. Prim, the grippers, and the quick change, and queries the vision system. The commands to E-move from Subtask take the form of generic manipulation tasks such as Move-to-Object, Locate-Object, and Grasp-object. Subtask generates these commands as a decomposition of more general commands from the Task level like Get-Object or Place-object. Subtask is also responsible for coordinating the movements of the active pedestal and the robot. The Task process receives its commands from the workstation controller via the factory network. These commands define the principle operations of the robot in the workstation, e.g. Transfer, Move, Refixture. The commands include parameters such as the type of object to be moved, and the source and destination locations for the object.

This control hierarchy was developed for moving in free space, that is, in regions where there is limited contact between the manipulator and its environment. Thus, almost all commands are decomposed, at the Prim level, into sequences of Goto-Point or Go-Thru-Point commands. This is sufficient for most operations of the robot, but for assembly operations such as tool changing, force sensing is needed.

III. Tool Changing Hardware

The feasibility of changing tools with the robot was addressed by Rippey and Vranish [4]. They designed hardware and tested it with a simple teach/playback controller. The main components developed were a gripper for grasping tools, a tray for transporting and holding tools, and a load/unload position on the horizontal machining center.

The tool gripper, shown in Figure 3, was designed to grasp the standard tool holders in which tools are mounted. The tool gripper fingers locate around the flange of the tool holder. This provides a very stable grasp and leaves a space between the fingers for the tool drum fingers to grip the holder during the exchange of the tool between HMC and the robot. As demonstrated in Figure 4, the tool trays hold the tools upright so that the robot can easily obtain this grip. (For more on the gripper and tool tray designs see [4,5,6].)

The load/unload station of the tool drum can be seen in Figure 5. The large, white plate is a low-friction Teflon* surface that is used to control the position of the tool gripper fingers. The fingers are compliantly attached to the gripper. By pressing the fingers against the strike plate, the robot is steadied and small errors in robot position will not affect the exchange, Figures 6 and 7. The tool pocket has a mechanism for actuating the tool drum fingers, and sensors for detecting the position of its fingers and presence of a tool. In addition, a chamfered tab is used to acquire a notch of a tool

holder to secure its orientation.

To load a tool, the robot moves the tool straight into the tool pocket, the chamfered tab acquiring the inboard notch of the tool holder, until the gripper reaches the strike plate. Then the tool drum fingers grip the tool, the robot releases the tool and slides outward from the tool drum along the strike plate. When the robot's fingers are free of the tool flange, the robot moves off the strike plate to a safe position. To unload a tool, this process is reversed.

During testing of this mechanical system, Rippey and Vranish found that some form of sensing was needed to prevent disasters. For example, if the robot tries to insert a tool into a pocket while the tool drum fingers are closed, the robot will drive into the closed fingers and cause extensive damage. A force sensor used to prevent disasters must be capable of surviving substantial loads so that it is not accidentally destroyed by the hydraulic robot. The sensor should be able to detect several directions of force and torque as well. To provide this sensing, a JR^{3*} custom force/moment sensor was installed between the fingers and actuator of the tool gripper, Figure 3. The stainless steel, six-axis device has a 900-pound capacity in the axial direction, the direction in which the robot is most stiff.

Integrating the force sensor into RCS required getting conditioned sensor readings into the Prim control level. To do this, unused quick change signal lines were connected to the force sensor. The corresponding lines across the quick change interface were fed into strain gage conditioners. The conditioned signals were plugged into the A/D port of the Prim/E-move board, where they are read by routines in the Prim process.

Figure 1 shows that each gripper is controlled by a separate control process for that gripper. A new control process for the tool gripper was created and compiled onto the computer board with the other gripper control processes. The modularity of RCS allows this to be done simply, with minimal modification to existing control processes. An interface to the Tool Gripper process was created in E-move. E-move sends commands to the Tool Gripper and receives status back. RCS has provisions for quickly creating these communication buffers and incorporating them into a control process [2]. Gripper control signals were brought into the Tool Gripper process in the same way the force sensor was integrated into Prim.

IV. Control Process Programming

Four new Task-level commands were added to RCS to enable the robot to perform tool changing. If it were not desired to monitor

forces during tool changing motions, the application could have been easily added using the existing Transfer, Move and Position commands. As noted previously, however, such force monitoring is required to prevent disasters when the robot moves in contact with fixed workstation equipment, and the new commands make this possible. Two of the new Task commands, Insert-tool and Release-tool, provide for loading a tool into the drum. The other two, Acquire-tool and Remove-tool, were created for unloading tools. Four corresponding commands were also added to the Subtask and E-move levels of RCS. Two new Prim level commands, Go-while-not-detected (Go-While) and Calibrate-sensor, provide the desired guarded move capability.

The Task-level Insert-tool command causes the robot to move the tool into the tool transfer location, monitoring the force during the final segment of the insertion. Release-tool directs the robot to open the gripper fingers, perform a force-guarded slide away from the tool along the strike plate, and depart from the strike plate.

Similarly, for tool unloading the command Acquire-tool directs the robot to make a force-guarded move to the strike plate followed by a guarded slide along the plate toward the tool. Acquire-tool also causes the gripper to close after the slide to the tool is complete. Remove-tool results in the withdrawal of the tool by the robot, again with forces being monitored to detect any jamming or improper contact.

The Horizontal Workstation Controller (HWSC) provides coordination between the robot tool changing motions and the opening and closing of the tool drum fingers. It accomplishes this by sequencing the issuance of commands to RCS and the High-Level Machine tool Controller (HLMC). The command and status protocol between the HWSC, RCS and the HLMC provides a natural means to effect the handshaking needed for a successful tool transfer.

Adding a command to a control process essentially means creating an additional branch possibility for its primary decision process. This branch represents an additional capability which executes if the new command is selected. Each command has a secondary decision process to determine what commands will be sent to subordinate processes based on sensory, status, and other conditions. This is how RCS performs task decomposition. Addition of new commands is facilitated by the use of modular, generic structures for decision processing routines. Adding a new command does not affect the execution of existing commands at all.

The E-move level performs most of the decomposition of the tool change commands. E-move breaks the commands down into elemental motion segments such as Delta-moves and Go-while moves. Delta-move and Go-while are Prim-level input commands which specify moves relative to a particular location and sensor-guarded moves, respectively. E-move also sends a Calibrate-sensor command to Prim prior to each guarded

move to negate the effects of tool weight and normal contact forces from the force readings. In addition, the E-move level coordinates gripper actions to grasp and release the tool appropriately.

At the Prim level, the Go-while and Calibrate-sensor commands were added to provide the guarded move capability needed for tool changing. Although created for tool changing, they may also be useful for other operations. Go-while allows a sensor to be monitored during a move, and if the sensor exceeds a threshold value then the motion is halted and an error status message is sent. The Prim level responds with a "done" status only if the robot successfully moves to the goal position without generating a detected error. The sensor to be monitored during a Go-while move is specified by name as one of the parameters contained in the input command from E-move. Any sensor integrated into the Prim level may be monitored, and individual force/moment components from the force sensor may be selected by naming them as different "sensors". The other new Prim command, Calibrate-sensor, takes a baseline reading from the force sensor with the robot in position prior to a Go-while and uses this to calculate the threshold values for the guarded move.

The new tool change commands which have been added allow for force monitoring during critical motions. Unlike the free-space commands, however, the Task-level tool changing commands are not generic—they may not be executed with an arbitrary object at an arbitrary location. The commands do, however, represent a first step in the development of such a set of generic commands for assembly-type tasks where contact with fixed objects is anticipated. The next step will be to define appropriate data structures and modify the decision processing to allow the object and location of an Insert or Remove command to be specified by the workstation controller.

V. Data Programming

Whenever the robot is to manipulate a new object, data which describes how to handle the object must be entered into RCS. This section describes the types of data which were entered to program tool changing. RCS provides special data structures and forms for entering the required information. The available data structures and the details of RCS data entry have been described in a previous paper [2] and will not be repeated here.

The data for new RCS applications falls broadly into two categories: 1) locational information and 2) object-related information. The locational information entered for tool changing consists of:

- The position and orientation (pose) of the robot at the tool transfer location
- The robot pose for a location near the tool transfer position, but safely back and away from the machine
- Offsets used for the tool changing motions, which are performed relative to the tool transfer pose
- The locations of the tool holders in the tool trays, which define a one-dimensional array
- Intermediate trajectories to define collision-free paths between the tool trays and the transfer location, including velocity and acceleration parameters for each segment

Creating the object-related data for tool changing required the definition of:

- An object called TOOL
- The approach, grasp, and depart grip sizes for TOOL
- The association of the tool tray array with TOOL
- Position offsets to be used for different grips of TOOL at different locations
- Approach and depart trajectories for different grips of TOOL at different locations, including points moved through and velocity/acceleration parameters

Although the tool change commands described contain a substantial amount of information about how and where to perform the operation, they are still data-independent in that locations, offsets, and the like are referred to symbolically. The control procedures do not contain any numeric values to describe movements, locations, or trajectory parameters. Only one robot pose, the tool transfer position, had to be taught for tool changing. The "safe" location was defined by adding an offset to this pose. The locations of tools at the MBD are defined by the tool tray array, which specifies the positions relative to a taught pose common to all tray configurations. The use of relative moves makes it relatively easy to fine tune the tool change procedure by making small adjustments to the transfer pose.

VI. Conclusion

The RCS-controlled tool changing operation described provides the workstation with the tooling flexibility needed to automate production of a wide variety of parts. Force sensing and guarded-move

capabilities were added to ensure the safe performance of this assembly-type task. A new gripper control process was also integrated into the system. The new commands and data which have been added provide easily-modified control of the robot tool-changing motions and a basis for implementing more generic RCS assembly commands.

VII. Acknowledgements

Development of the NBS Real-time Control System is partially supported by funding from the Navy Manufacturing Technology Program.

*Certain commercial equipment is identified in this paper to adequately describe the experimental facility. Such identification does not imply recommendation or endorsement by NBS.

References

[1] Simpson, J. A., Hocken, R. J., Albus, J. S., "The Automated Manufacturing Research Facility of the National Bureau of Standards," Journal of Manufacturing Systems, Vol. 1, No. 1.

[2] Barbera, A. J., et al., "RCS: The NBS Real-time Control System," Robots 8 Conf., Detroit, June, 1984.

[3] Fitzgerald, M. L., et al., "Real-time Control Systems for Robots," 1985 SPI Nat. Plastics Conf., Chicago, June, 1985.

[4] Rippey, W. G., Vranish, J. M., "Robotic Tool Changing in a Horizontal Workstation," Robots 9 Conf., Detroit, June, 1985.

[5] Vranish, J. M., "'Quick Change' System for Robots," Robots 8 Conf., Detroit, June, 1984.

[6] Bunch, R. W., Vranish, J. M., "'Split Rail' Parallel Gripper," Robots 9 Conf., Detroit, June, 1985.

[7] Scott, H., Strouse, K., "Workstation Control in a Computer Integrated Manufacturing System," Autofact 6 Conf., Anaheim, CA., Oct., 1984.

[8] Scott, H., Strouse, K., "Horizontal Workstation Controller Task Programming and Execution," AMRF Internal Report, Sept., 1985.

[9] Fishman, D., Scott, H., Bunch, B., "Integration of Material Buffering Devices in an Automated Factory," 2nd Int. Conf. Robotics and Factories of the Future, San Diego, CA., July, 1987.

[10] Albus, J. S., et al., "A Control System for an Automated Manufacturing Research Facility," Robots 8 Conf., Detroit, June 1984.

[11] Slocum, A. H., Peris, J., Donmez, A., "Development of a Flexible Automated Fixturing System," SME Conference on Flexible Manufacturing Systems, SME Technical Paper #MR 86-128, Feb., 1986.

[12] Myers, D. R., "Sensory-Interactive Control of an Instrumented Gripper," Internal Report, Robot Systems Division, 1984.

[13] Jones, A. T., McLean, C. R., "A Proposed Hierarchical Control Model for Automated Manufacturing Systems," Journal of Manufac-

turing Systems, Vol. 5, No. 1.
[14] Zygmont, J., "Flexible Manufacturing Systems: Curing the Cure-all," High Technology, Oct., 1986.
[15] McCain, H. G., "A Hierarchically Controlled, Sensory Interactive Robot in the Automated Manufacturing Research Facility," 1985 IEEE Int. Conf. on Robotics and Automation, St. Louis, March, 1985.

Fig. 1. Hierarchy of Control Levels.

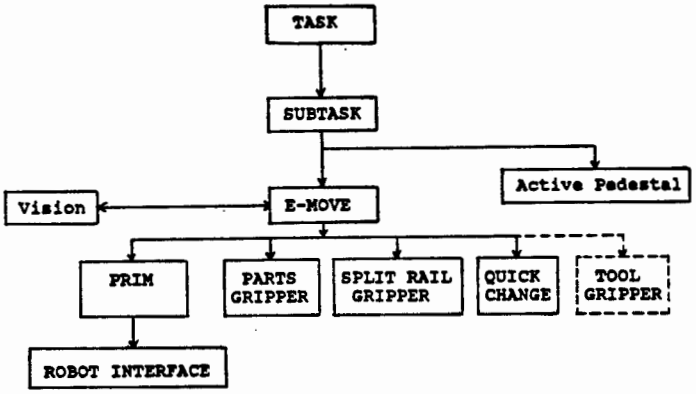
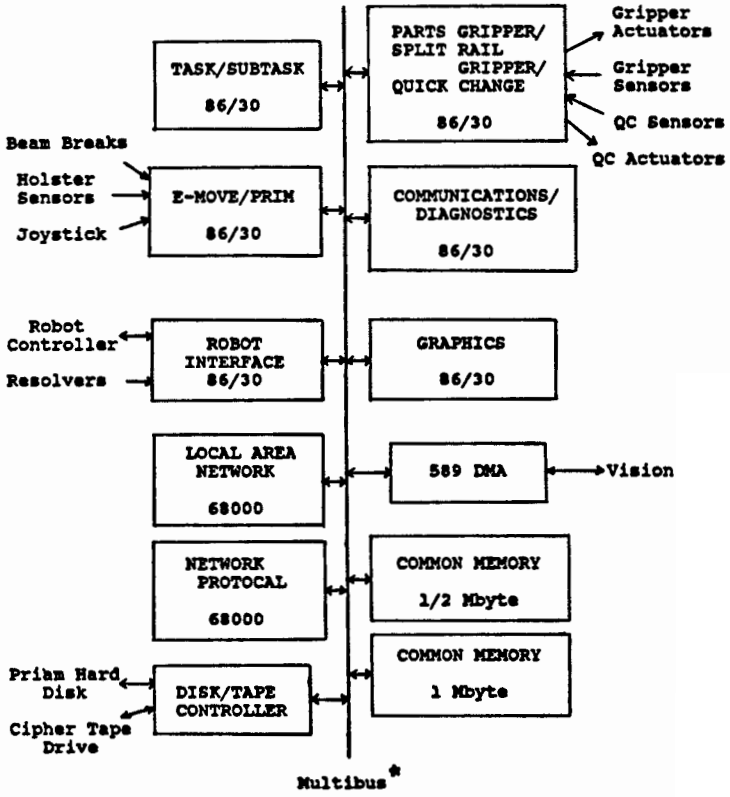


Fig. 2. RCS Hardware Configuration.



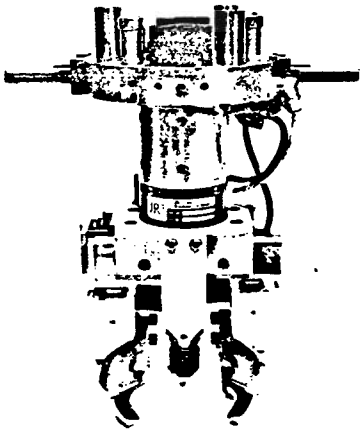


Fig. 3. Tool Gripper.

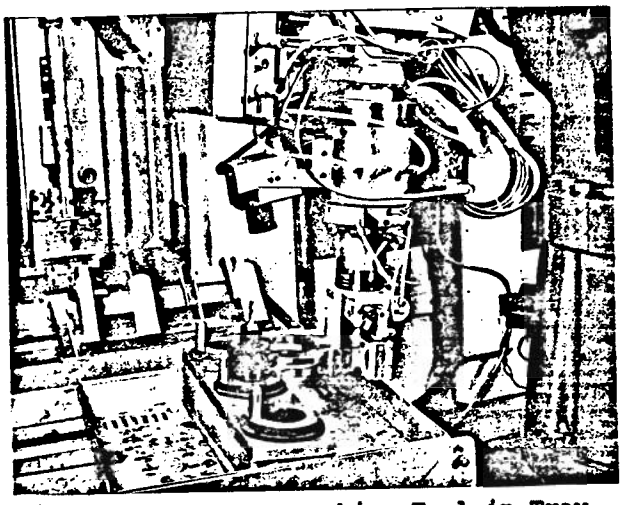


Fig. 4. Robot Approaching Tool in Tray.

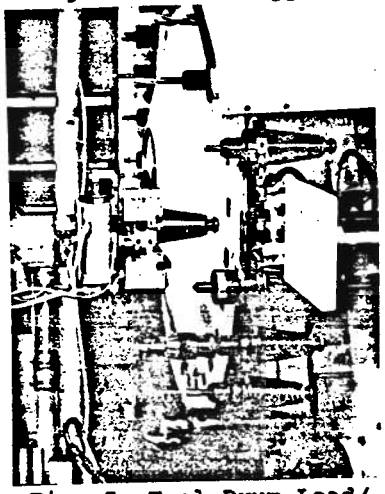


Fig. 5. Tool Drum Load/ Unload Station.

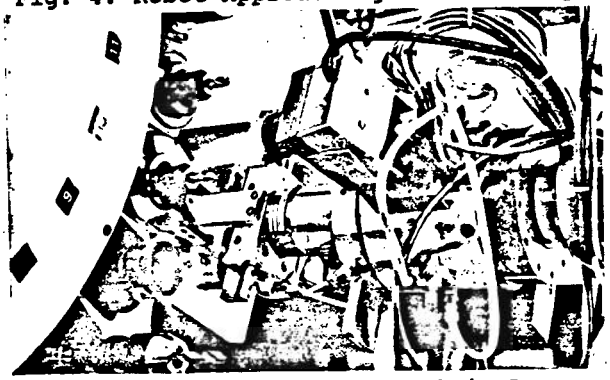


Fig. 6. Robot Gripping Tool in Drum.

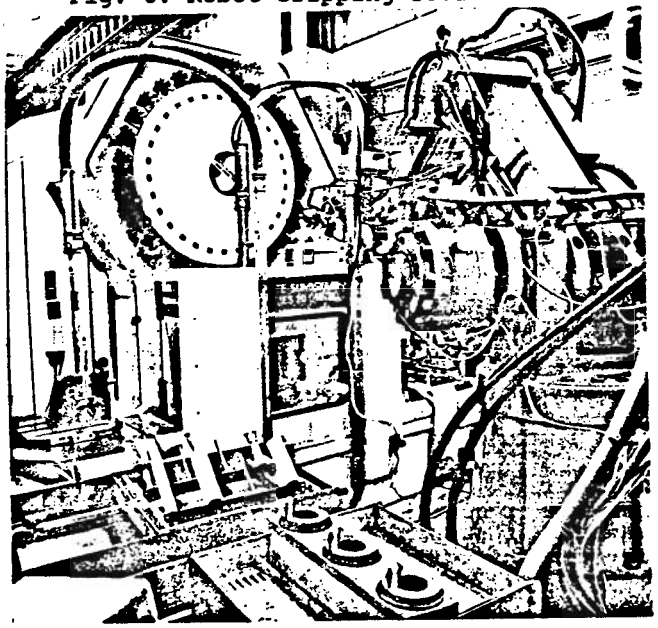


Fig. 7. Overall View of Tool Transfer.