AN ADVANCED DEBURRING AND CHAMFERING SYSTEM (ADACS) BASED ON THE ENHANCED MACHINE CONTROLLER (EMC)

Keith Stouffer, Fred Proctor, Elena Messina and Jim Albus
Intelligent Systems Division
National Institute of Standards and Technology
Building 220, Room B124
Gaithersburg, Maryland 20899 USA

ABSTRACT

An autonomous finishing workcell for deburring and chamfering high precision machined parts has been developed by engineers at the National Institute of Standards and Technology (NIST) working with United Technologies Research Center (UTRC) and Pratt & Whitney. This Advanced Deburring and Chamfering System (ADACS) is an application of the NIST Enhanced Machine Controller (EMC). The EMC is an open architecture hierarchical controller suitable for a variety of high-fidelity real-time control systems. Within ADACS, a feature-based process planning system generates the deburring and chamfering paths based on parameters and edges selected by a manufacturing engineer from a solid model Computer Aided Design (CAD) definition of the part. Simulation of the paths as well as collision detection is performed before the generated tool paths are downloaded to the robot/machine tool, and finishing tool. The ADACS uses active force control in the tool to control stiffness of the tool normal to and tangential to the chamfer edge. The ADACS is being installed in a commercial application for finishing jet engine components at Pratt & Whitney.

INTRODUCTION

The finishing operation is a critical step in the manufacturing of parts manufactured from hard metals. After a part has been machined, a finishing operation is usually required to remove excess material, or burrs, to bring the part within tolerance of the specification. The primary finishing processes are deburring and chamfering. In the past, and still presently, the finishing operation has been performed manually at a burr bench with a hand held spindle grinder. This hand-crafting of parts tends to be expensive, inconsistent and inaccurate. Manual finishing can account for 10%–20% of the total labor cost and approximately 10%–30% of the manufactured parts need rework after the manual finishing process.

Automation of the finishing process would prove to be very beneficial. By automating the finishing and chamfering processes, tolerances could be held to less than 0.08 mm (0.003 in), the finishing costs could be reduced as much as 50%, and the rework rates could be nearly eliminated.

The National Institute of Standards and Technology (NIST) and United Technologies Research Center (UTRC), under Navy ManTech funding, have developed the Advanced Deburring and Chamfering System (ADACS) which is capable of processing aerospace parts made from hard materials such as titanium and inconel. For aerospace parts, the ADACS must produce a precision 45 degree break edge, or chamfer, for part edge geometries such as modified and full radii.

Features of ADACS include:

- open architecture controller
- operator–controlled, off–line graphical user interface exploiting CAD part models to off-line program and simulate finishing trajectories
- automated extraction of features from edge data
- active tooling to compensate for small position errors and to maintain a constant cutting force on the edge
- tightly coupled coordination of tool and motion commands to achieve ramping and smoothing

An open system built from open component technology was the major design paradigm used to achieve the ADACS system requirements for integration, flexibility, and extensibility. System design using an open architecture reference model with well-defined interfaces offers a sound approach to implementation that can be adapted to satisfy future requirements. The ADACS used the Enhanced Machine Controller (EMC) for its control architecture. The EMC is an open-architecture reference model with well-defined interfaces. This paper will show how the ADACS

The research developed during the ADACS project was supported by the U.S. Navy ManTech Program.

was designed and implemented using the EMC architecture.

EMC ARCHITECTURE

Open architecture controllers offer many benefits to users of machine tools, robots and coordinate measuring machines, which will ultimately reduce the life-cycle costs of installing, operating, and maintaining manufacturing equipment. Aside from those benefits resulting from basing a controller on common operating systems and computing platforms, the main feature of an open architecture is the public availability of interfaces to controller functionality. These interfaces allow third parties who are not associated with the original equipment manufacturers to provide enhancements to the functionality of the machine. Efforts to standardize the interfaces to open architecture machine tool controllers are underway both in the United States and abroad. In the United States, the Department of Energy and NIST have cooperatively undertaken this standards effort (Proctor et. al. 1996).

In the early 1990s, the Manufacturing Engineering Laboratory at NIST began the EMC program to develop a modular definition of components for machine control (Proctor and Michaloski 1993). The intent of this program was to document the interfaces to these modules to the degree that would allow independent third parties to provide interoperable products. The development of this modular architecture grew out of NIST's experience developing controllers based on the Real-time Control System (RCS) reference model architecture and NASREM reference model architecture (Albus 1991). NIST and the Department of Energy national laboratories have combined their efforts in this area under the support of the Technologies Enabling Agile Manufacturing (TEAM) program, and are undertaking a formal review of the interface specification that has resulted from recent implementations of interface-based controllers. The development of the EMC architecture, shown in Figure 1, was the first step toward defining an interface specification. In this figure, boxes indicate the individual modules for which interfaces have been defined and validated. These include Task Sequencing, Trajectory Generation, Servo Control, and Discrete Input/Output. The Operator Interface, shown at the side, does not require any specific interfaces itself, but can be developed using only the interfaces provided by the other modules. Implementations of the operator interface are only required to make use of the messages to the controller and data provided by the controller: no additional interfaces are required to be defined in order to incorporate an operator interface into an EMC controller.

The interface specifications are formalized in the C++ programming language using header files. The specification consists of messages into each module, and world model data provided by each module. Both the messages and world model data are implemented using C++ classes.

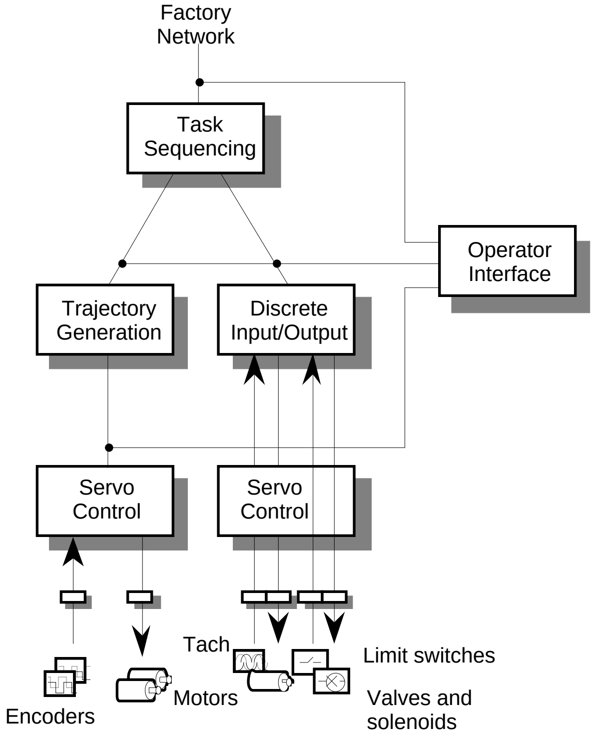

Figure 1. The EMC Architecture.

Class definitions alone are not sufficient to describe the interfaces. The specification needed to include the expected behavior of the control modules in response to each control message, and their effect on the world model of each control module. This information is provided in manual-style pages accompanying the C++ class definitions, using Hypertext Markup Language (HTML) format.

Supplementing the message specification is a model of data transfer, the Neutral Manufacturing Language (NML) (Shackleford and Proctor 1996). This model provides for "mailboxes" of data, with one or more readers and writers. Each module is modeled as a cyclic process, which reads its input command from its supervisor, reads the status of its subordinates (or sensors), and computes and sends outputs to its subordinates (or actuators).

The interface specification is divided into two parts: commands that each module will perform, and status that each module will maintain. Both commands and status are derived from the NML message base class, and require a unique identifier and zero or more data fields representing the parameters to the command or fields in the status. During the development of the specification, the intent was to analyze the general requirements of each module in terms of which commands it should be responsible for carrying out, and what world model status it should be responsible for maintaining.

ADACS CONTROL STRUCTURE

The ADACS control structure, shown in Figure 2, is based on the Unified Telerobotic Architecture Project (UTAP) architecture (Russell et. al. 1995). UTAP is an EMC compliant architecture that defines open interfaces between modules of the system (Lumia et al. 1994). The integration of these modules is performed by placing wrappers around the software that accept commands that are specified in the UTAP document. A detailed description of the UTAP architecture is beyond the scope of this paper and is left to the reader to further investigate.

The ADACS uses an around–the–arm control approach (Murphy and Proctor 1990). One manipulator is used for gross positioning and another, an active tool, for fine positioning and force control. Either a robot or machine tool is used as the gross positioner and the Chamfering and Deburring End-of-arm Tool (CADET), an active tool designed by UTRC, is used as a fine positioner. Each has its own controller that is coordinated by a workcell supervisor (Stouffer et. al. 1993). This type of control allows for the inaccuracies of the robot/machine tool because the active tool can make up for small positioning errors (Guptil and Stahura 1987).

The ADACS is designed to have two discrete operator interfaces (Stouffer and Russell 1995). The Developer Interface is for the manufacturing engineer who knows the required manufacturing processes such as the required chamfer depths and on which edges they occur. This is where the part program is produced for use on the shop floor. A CAD package is used in which the manufacturing engineer can select edges and input machining parameters required to finish the edge. Next, the part program is run through a workcell simulation to verify that no unexpected collisions will occur between the hardware and the environment and that the desired trajectories are performed. The part program is used by shop floor personnel operating the actual workstation through the User Interface. This interface allows the operator to select a part program, run it and change various settings (feed rate, force, etc.) on the fly if required.

The Developer Interface allows the manufacturing engineer to produce the part program that will be run on the factory floor. ProManufacture and ProEngineer, CAD/CAM packages developed by ParametricTechnologies, are used to create the tool paths for the specific features that need to be finished. This data is then run through a post processor that creates tag points for use within the Deneb Robotics workcell simulation package, Telegrip. These tag points are placed on the features of the CAD model of the part. This allows the engineer to now create a feature based program.

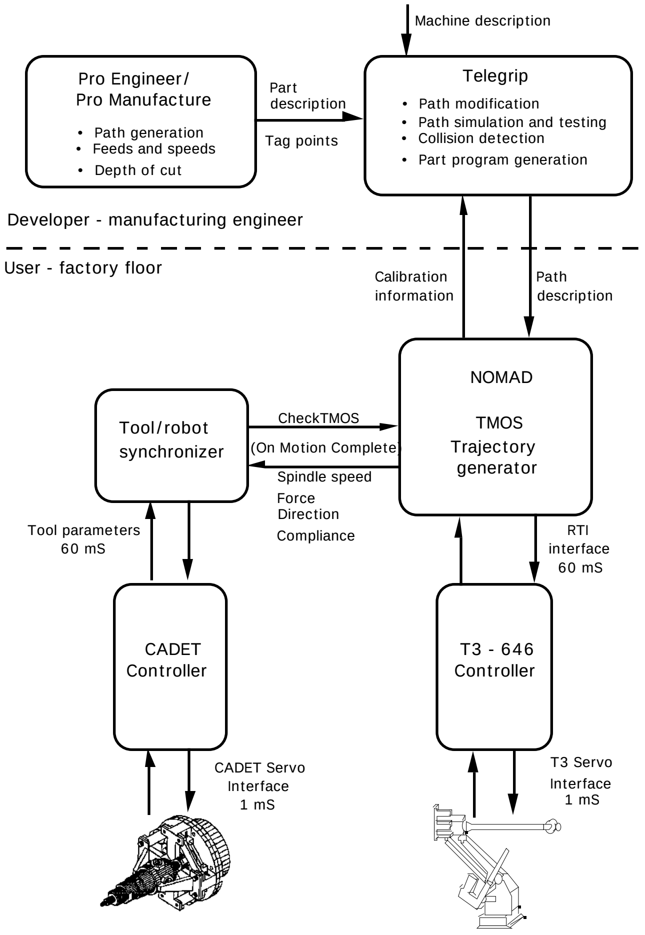

Figure 2. The ADACS Control Structure.

The engineer selects what features to chamfer, what order to chamfer them in, and programs any intermediate clearance points required to allow the hardware to move from one feature to the next without collisions with the environment. Telegrip is a workcell simulation software package that allows a part program to be simulated before it is downloaded to the actual hardware. This allows the programmer to preview the paths being performed to determine if the hardware is actually doing what it is expected to do. Collision detection is also performed at this level to check for any unwanted collisions between the hardware and the environment. A post processor is then run on the program generated within Telegrip to be interfaced with the Trellis–developed NOMAD motion control system. This allows the program generated in Telegrip to be executed on the actual workcell.

Calibration information obtained from calibration procedures in the NOMAD system can also be uploaded to the Telegrip software to calibrate the models of the software simulated workcell (world model) with the actual workcell.

The part program, generated by the manufacturing engineer, is used by shop floor personnel operating the actual workstation through the User Interface. This interface allows the operator to select a part program produced by the manufacturing engineer, run it, and modify selected parameters (feed rate and force) on the fly if required to meet the required tolerances. The user receives graphical feedback of the current feed rate, force, and chamfer depth.

NOMAD is a software package that assists in producing machine controllers. The trajectory generation software within NOMAD, the Trellis MOtion System (TMOS), provides high level C interfaces to general motion control of machines. TMOS is used for the ADACS trajectory generation. It is designed to be a part of an open system that allows other trajectory generators to be swapped in and out with no disturbance to the system as a whole. The trajectory set points generated by the motion controller are either downloaded to the robot/machine tool controller or to the Telegrip software for simulation. A synchronization process couples the commands that are sent to the CADET with the position of the robot/machine tool. This is performed by cyclic polling of the trajectory generator to determine if the motion setpoint has been reached.

To remove material from a part manufactured from a hard material, a hard cutter must be used. Hard cutters require compliant tool holders, either passive or active, to reduce chatter and to account for inaccuracies in the planned trajectory. Robot arms, unlike structurally stiff machine tools, have a relatively low stiffness that allows large amplitude resonances that cause chatter. Chatter is reduced when the tangential stiffness is approximately 10 times stiffer than the normal stiffnesses, as shown in Figure 3 (Asada and Slotine 1986). When following an edge, robot accuracy is not sufficient to keep a hard cutter on the edge. Therefore, the normal direction of the tool must be made to be compliant, so that the cutter will remain in contact with the edge and apply the necessary normal force to achieve the required break edge. Compliance can either be implemented passively (with spring and damper system) or actively through force control.

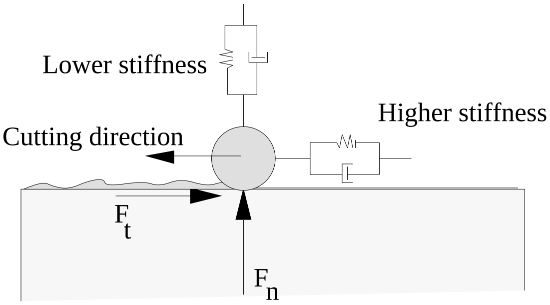

Figure 3. Tool Chamfering Stiffness

ROBOTIC IMPLEMENTATION

The main components of the world model for the ADACS consists of device kinematics, feature knowledge, chamfer knowledge, and tooling force compensation. ADACS uses the base class concept of C++ to define a chamfer edge object. A chamfer edge includes the typical data definitions of a starting, entry, and ending position and orientation. However, the chamfer edge applies the C++ virtual function to include functions to derive direction, concavity, orientation and input and output format interface. Depending on the feature, the default definitions of the virtual function might be overridden.

A robotic implementation of the ADACS system has been integrated at the Advanced Manufacturing Research Facility (AMRF) located at NIST in Gaithersburg, Maryland. This implementation was used to demonstrate automated finishing on mock Sikorsky helicopter components. For the robotic implementation of ADACS, a Cincinnati Milacron T3–646 six-axis electric robot, shown in Figure 4, was used as a macropositioner and the CADET was used as a micropositioner and force control tool.

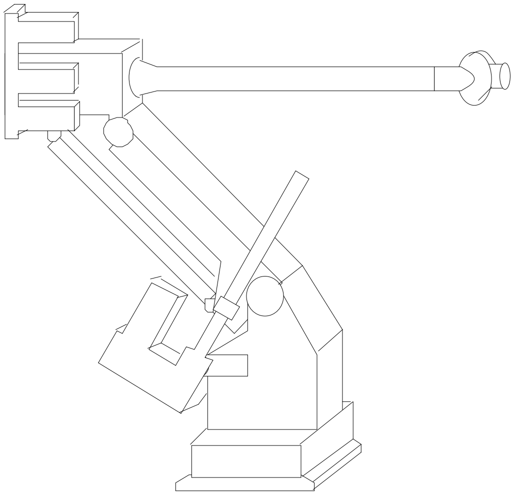

Figure 4. T3-646 Robot

The CADET, shown in Figure 5, incorporates a two-axis gimbal which permits movement and control of the tool tip in a direction perpendicular to the spindle axis over a 6.4 cm^2 (1 in^2) area (Engel et.al 1992). This ability to control the direction of the cutting force in real-time is unique to the CADET. The CADET's high bandwidth force servo receives force and direction commands from the machine controller telling the CADET where the edge is and what cutting force to maintain. This allows the CADET to maintain a constant cutting force on the edge while actively adapting to variations in edge location.

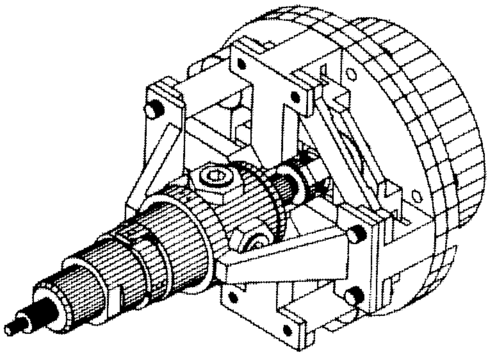

Figure 5. Chamfering and Deburring End-of-arm Tool (CADET)

The workcell supervisor and robot controller run on a 68040 processor running in a VME backplane. The LynxOS real-time operating system was chosen as the operating system because it is a POSIX compliant operating system. The NOMAD motion control software was used for generating trajectories for the robot. The CADET controller also runs on a 68040 processor in a VME backplane, but uses the VxWorks real-time operating system.

The ADACS makes use of various off the shelf software packages. These products offer hooks into their systems that allow them to be modified or enhanced. The NOMAD software package is used as the trajectory generation software for the ADACS and is also used as the operator interface for the factory floor User Interface. ProManufacture, a module that runs within ProEngineer, is the CAD/CAM software package used to create tool paths. Telegrip is a machine simulation software package that allows a part program to be simulated before it is downloaded to the actual hardware.

The ADACS also make use of various NIST developed control code. This archive consists of C and C++ code that supplies application-independent libraries, such as communication, tasking, vector math, etc., as well as application-specific routines, such as device kinematics, I/O drivers, etc. The application–independent services are designed to be platform–independent so that code is transparently portable across platforms. A set of shell commands are provided within the archive as a programming convenience. These commands automate much of the tedious programming chores and provide a consistent programming paradigm.

MACHINE TOOL IMPLEMENTATION

Shop schedules did not permit the use of a production CNC machine at Pratt & Whitney for its first machine tool application (Dansereau and Grot 1996). The CADET was integrated with a surplus Series 200 K&T 4-axis machine tool. The only modifications required to the K&T were the installation of 110 volts outlets for the CADET's support equipment , the addition of two external motors to the K&T controller to drive the encoders for CADET normal force and direction commands and a bracketed shelf to hold the Senotec unit and the two Kaman amplifiers.

A new controller was installed on the K&T by UTRC. The Delta Tau CNC machine controller was selected due to its relative flexibility and capability to process information from external devices, such as the CADET, and integrate this information with commands routinely used by standard CNC equipment.

The CADET successfully produced 0.53–0.63 mm (0.021–0.025 in) chamfers on all designated hole edges on the inner diameter surface of the test compressor case. The surface finish of 50 Ra was consistent with this type of finishing operation (hole edge chamfering) and was within the allowable blueprint requirements of 125 Ra max. The force used to produce the 0.53–0.63 mm (0.021–0.025 in) chamfers was 0.3 lb. at a feed rate of 27.5 mm/sec (1.1 in/sec), which is also consistent with this type of finishing operation. Overall the CADET performed as expected and programmed, producing consistent chamfers from hole to hole, even thought the hole positions varied up to 0.5 mm (0.020 in). This variation is common when performing repetitive tasks on similar features within one piece of hardware. This validates the CADET's ability to seek the part hole's edge and apply the correct amount of force to finish the edge, while taking into account the edge's variation in position.

The CADET's performance in this CNC environment was excellent, although the testing and demonstration were executed under a relatively narrow scope based on the range at which most pieces of CNC equipment operate. With some development and factory hardening, the CADET has the potential to be used in a normal CNC production environment.

CONCLUSION

The ADACS supplies a CAD–based graphical interface of a part, wherein the operator uses a mouse to select feature edges to chamfer and supplies machining parameters and cutting strategies. The ADACS subsequently generates the finishing process model and performs the finishing operation. The key features of the ADACS are:

- open architecture controller
- operator–controlled, off–line graphical user interface exploiting CAD part models to off-line program and simulate finishing trajectories
- automated extraction of features from edge data
- active tooling to compensate for small position errors and to maintain a constant cutting force on the edge
- tightly coupled coordination of tool and motion commands to achieve ramping and smoothing

From our experiences, the ADACS system has proven to be a flexible and useful system. System design using the EMC open architecture reference model offered a sound approach to implementation that could be adapted to satisfy future requirements. ADACS systems have been applied to both robotic and CNC platforms with excellent results. With some development and factory hardening, the research developed in the ADACS has the potential to be used in a production environment.

REFERENCES

Albus, J.S., “Outline for a Theory of Intelligence,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No.3, May/June 1991.

Asada H., Slotine, J. J., Robot Analysis and Control, John Wiley and Sons, New York, New York, 1986.

Dansereau, R. and Grot, A. “Advanced Deburring and Chamfering System Development, Tasks IV and V Addendum - CADET Integrated with a CNC,” April 1996.

Engel, T. W., Roberts, R. K., Proctor, F. M., “Specification of an Active Force Control Tool for Performing Deburring and Chamfering on a Robot Platform,” Proceedings of the International Conference on Industrial Electronics, Control, Instrumentation, and Automation, San Diego, California, November 9-13, 1992.

Guptil, R., Stahura, P., “Multiple Robotic Devices: Position Specification and Coordination,” Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, March 31 - April 3, 1987.

Hollowell, R., “Hybrid Force/Position Control for Robotic Light Machining,” Robotics and Remote Systems Conference, Charleston, South Carolina, March 1989.

Lumia, R., Michaloski, J.L., Russell, R.T., Wheatley, T.E., “Unified Telerobotic Architecture Project (UTAP) Interface Document,” June 1994.

Murphy, K. N., Proctor, F. M., “An Advanced Deburring and Chamfering System,” Presented at Third International Symposium on Robotics and Manufacturing, British Columbia, Canada, July 1990.

Proctor, F.M., Michaloski, J., “Enhanced Machine Controller Architecture Overview,” NISTIR 5331, December 1993.

Proctor, F., Michaloski, J., Shackleford, W., Szabo, S. “Validation of Standard Interfaces for Machine Control,” Proceedings of the International Symposium on Robotics and Manufacturing - The World Automation Congress ‘96, Montpellier, France, May 27 - 30, 1996.

Russell, R., Michaloski, J., Stouffer, K., “The ADACS Implementation of the UTAP Architecture,” Proceedings of the 6th International Conference on Manufacturing Engineering, Melbourne, Australia, November 29 - December 1, 1995.

Stouffer, K. A., Michaloski, J., Russell, R., Proctor, F. “ADACS - An Automated System for Part Finishing,” Presented at IECON’93, Lahaina, Maui, Hawaii, USA November 15 - 19, 1993.

Stouffer, K., Russell, R. “ADACS: An Advanced Deburring and Chamfering System,” Proceedings of the 6th International Conference on Manufacturing Engineering, Melbourne, Australia, November 29 - December 1, 1995.

BIOGRAPHY

Keith Stouffer is a Mechanical Engineer at the United States National Institute of Standards and Technology (NIST). He has been with the Intelligent Systems Division (formerly the Robot Systems Division) since 1990, specializing in real-time control systems for automated robotic deburring. He has managed the Advanced Deburring and Chamfering System (ADACS) project since 1992.