ADACS: An Advanced Deburring and Chamfering System

Keith A. Stouffer and Robert Russell, Jr.
Intelligent Systems Division
National Institute of Standards and Technology, USA

Summary This paper describes a numerically controlled finishing system, the Advanced Deburring and Chamfering System (ADACS). ADACS employs a user friendly, Graphical User Interface (GUI) to off-line program finishing and chamfering edges and prompt the operator for part cutting parameters. Given the operator-designated finishing and chamfering edges, ADACS extracts the features to generate a part finishing process plan. Collisions with the environment are detected and avoided through simulation of the cutting path before the path is downloaded to the actual hardware. ADACS interprets the finishing plan to generate motion trajectories that are tightly coupled to the tooling control. Because position control is too inaccurate for the required tolerances, ADACS uses active force control in the tool to compensate for any nominal error along the finishing path. This allows the cutter to apply a constant force to the edge. The ADACS approach to finishing parts can significantly reduce the manufacturing cost of a part by reducing rework and scrap, and improving process consistency and quality. A prototype ADACS has been built and successfully applied to aerospace test parts.

1 INTRODUCTION

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 finishing operation is a critical step in the manufacturing of parts made with hard metals. The primary finishing processes are deburring and chamfering. In the past, and still presently, the finishing step has been performed manually at a burr bench with a hand held spindle grinder. As expected, there are problems with the manual finishing of parts. Manual finishing is inconsistent and inaccurate. If a part is damaged beyond repair, thousands of dollars spent on the initial machining processes can be lost. Manual finishing is also very time consuming. In Engel et. al. (1), Pratt & Whitney has estimated that 12% of their total machining hours are devoted to manual deburring and chamfering of parts after they have been machined. Additionally, manual finishing increases health care costs that result from lacerations from sharp edges and cumulative trauma disorders. The most notable of these health–related illnesses is carpal tunnel syndrome.

Automation of the finishing process would prove to be very beneficial. Presently, manual finishing accounts for 12% of the total labor cost and approximately 10%–30% of the manufactured parts need rework after the manual finishing process. By automating the finishing and chamfering process, tolerances could be held to less than 0.07 mm (0.003 in), the finishing costs could be reduced as much as 50%, and the rework rates could be nearly eliminated.

Traditionally, robotic automation has been accomplished through the use of teach programming. This method of programming is only acceptable for simple paths and large part runs. When parts become complex or when only small numbers are produced, this method becomes impractical. Teach programming is tedious, time consuming, and prone to inaccuracies. For complex geometries, such as arcs and splines, hundreds of points need to be taught along the surface for a robot to perform the trajectory accurately. Therefore, a usable autonomous finishing system must have the capability to use CAD models to quickly and accurately generate the necessary finishing trajectories based on this knowledge.

The National Institute of Standards and Technology (NIST) and United Technologies Research Center, under Navy funding, have developed an 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:

- operator–controlled, off–line graphical user interface exploiting CAD part models to off-line program and test 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
- feature and part based chamfering process model
- tightly coupled coordination of tool forces and motion control to achieve ramping and smoothing

In the following sections, the requirements, features, and components of the ADACS system will be discussed. Section 2 presents the functional requirements for deburring and chamfering. Section 3 describes the details of the ADACS control structure. Section 4 reviews two implementations of the ADACS system. A robotic implementation of the system has been integrated at NIST in Gaithersburg, Maryland and a machine tool implementation of the system is being integrated at Pratt & Whitney in East Hartford, Connecticut for automated finishing of aerospace parts.

2 ADACS DATA FLOW

The machining process takes a workpiece and transforms it into a part. Given a machined part, the finishing operation removes any remaining material left from the machining operation. In Stouffer et al. (2), the data flow of the ADACS system is designed. Figure 1 reviews the data flow within a typical ADACS finishing operation.

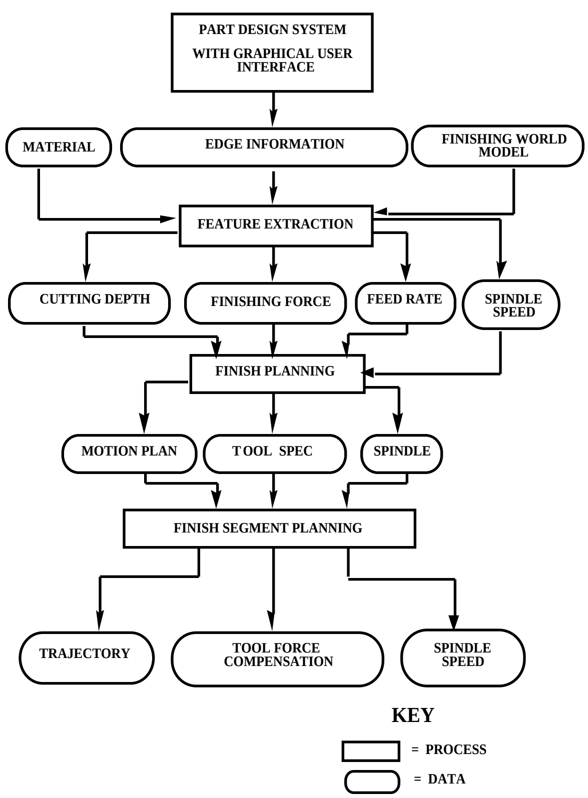

Figure 1. ADACS Finishing Process Model

At the top level, or part design system, we will assume that a CAD model of the part exists in a boundary representation. Using a graphical rendering of the part model and a mouse, an operator selects the part edges that require finishing. Edge information is extracted from the CAD data. The collection of edge information as well as material and finishing model information is sent to the feature extraction subsystem.

The feature extraction subsystem accepts part edge information and extracts part feature information. Edge knowledge is determined from the collection of edge information, including concavity, starting and ending tool orientation, and the state transition. State transitions include the transition from free–space to contact as well as continuous versus discontinuous features. In addition, the state transition from free–space to contact imparts ramping requirements on the motion and tooling control to provide blending of the edge.

The finish planning subsystem generates the required cutting force based on a process model that relates the cutting force to the inputted chamfer depth, feed rate, and spindle speed.

The finish segment planning subsection then generates motion primitives based on the chamfer features defined in the system world model. The system must account for proper machine setup and fixturing, proper tooling, and account for any interference from the fixturing. Each motion primitive must then in turn be transformed into coordinated position and force–control motion segments. These segments are then in turn transformed into a series of set points, or trajectories, and are downloaded to the robot controller. The required tool force compensation along with the spindle speed are sent to the tool controller.

3 ADACS CONTROL STRUCTURE

In Murphy and Proctor (3), the ADACS is described as using an around the arm control approach. One manipulator is used for gross positioning and another, the 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) is used as a fine positioner. As described in Hollowell (4), both have their own controllers that are supervised by a workcell controller. This type of control, as described in Guptill and Stahura (5), allows for the inaccuracies of the robot/machine tool because the tool can make up for small positioning errors.

The ADACS control structure, shown in Figure 2, is based on the Unified Telerobotic Architecture Project (UTAP) architecture. UTAP is an Enhanced Machine Controller (EMC) compliant architecture that defines open interfaces between modules of the system. The integration of these modules described above is performed by placing wrappers around the software that accept commands that are specified in the UTAP doc-

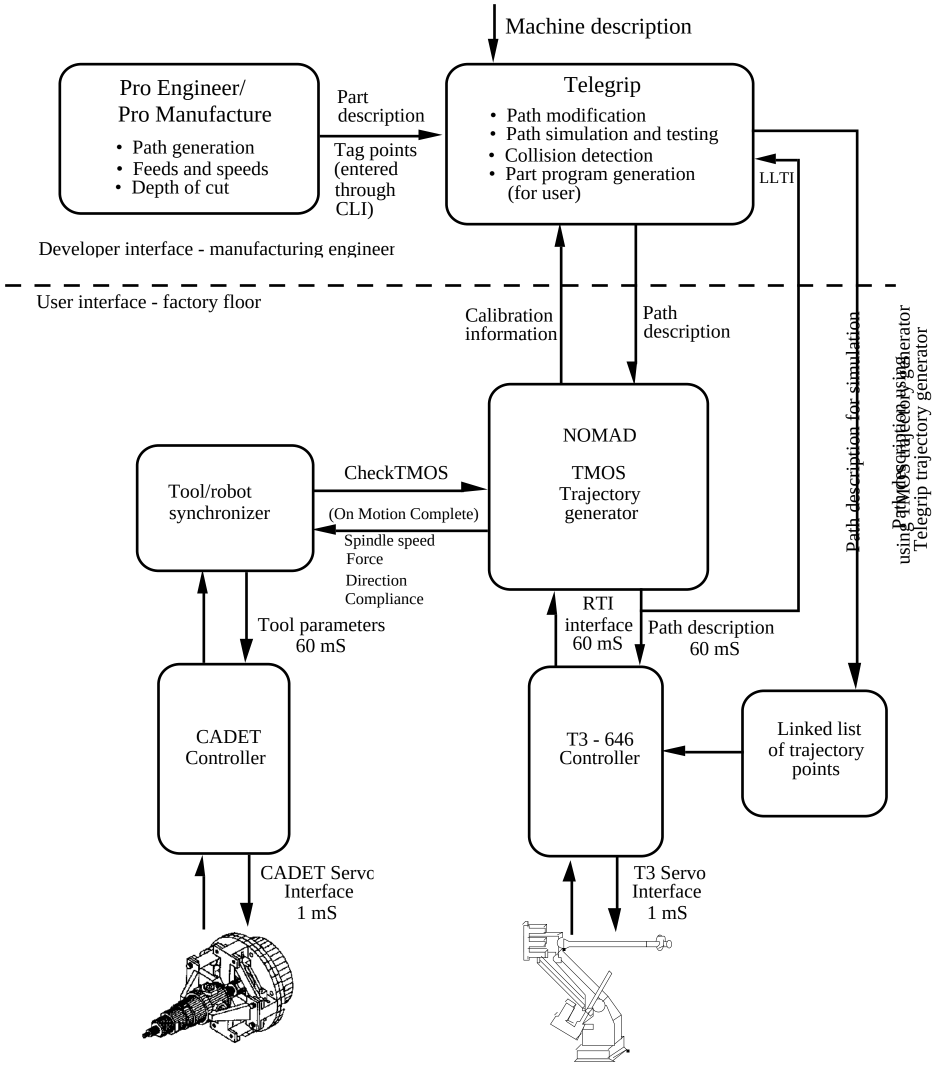

Figure 2. ADACS Control Structure

ument. The description of the UTAP and EMC architectures is beyond the scope of this paper and is left to the reader to further investigate. The UTAP architecture is described in greater detail in Lumia et al. (6). The EMC architecture is described greater detail in Proctor and Michaloski (7).

3.1 Graphical User Interfaces

The ADACS is designed to have two discrete operator interfaces. One 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 second operator 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 part program is interpreted by the controller and the data flow that was described in the previous section occurs.

3.1.1 Developer Interface

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 Parametric Technologies, 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. The engineer selects what features to chamfer and in what order to chamfer them in as well as programming intermediate clearance points 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 allow the programmer to view the paths being performed to determine if the hardware is actually doing what it is expected to do. Collision detection is also perfomed 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.

The Developer interface can also be used to produce trajectory points every 60 ms that can be downloaded directly to the robot controller, although the NOMAD interface is preferred. The actual trajectories generated by the NOMAD motion software can also be imported back into Telegrip for simulation to validate that the NOMAD controller is producing the correct motions.

Calibration information obtained from calibration procedures in the NOMAD system can also be uploaded to the Telegrip software to update the models of the workcell if required to update the model of the workcell with the actual workcell.

3.1.2 User Interface

The part program, generated by the manufacturing engineer, is used by shop floor personnel operating the actual workstation through the second operator interface. This interface allows the operator to select a part program produced by the manufacturing engineer, run it and change various settings (feed rate, force, etc.) on the fly if required. The user receives graphical feedback of the current feed rate, force, and chamfer depth. The part program is interpreted by the controller and the data flow that was described in Section 2 occurs.

3.2 Motion Trajectory Generation

NOMAD is a software package that assist 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 tool synchronizer 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 motion generator to determine if the motion setpoint has been reached.

3.3 Tooling Compensation

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. It is shown in Asada and Slotine (8) that chatter is reduced when the tangential and normal stiffnesses differ by a factor of 10, as shown in Figure 3.

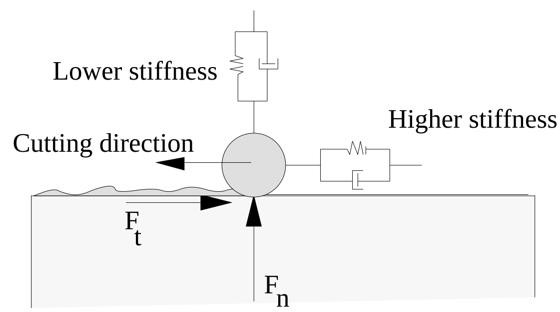

Figure 3. Tool Chamfering Stiffness

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.

4 ADACS IMPLEMENTATIONS

In this section, we provide a short overview of two ADACS prototypes, and how the ADACS system is used to finish parts. A robotic implementation of the system has been integrated at NIST in Gaithersburg, Maryland and a machine tool implementation of the system is being integrated at Pratt & Whitney in East Hartford, Connecticut for automated finishing of aerospace parts.

4.1 World Model

The main components of the world model for ADACS consists of device kinematics, feature knowledge, chamfer knowledge, and tooling force compensation.

Of most interest to this paper is the software development of the feature knowledge. 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. By defining a free–space and none–left edge objects, determining features during edge state transition was greatly simplified.

4.2 Robotic Implementation

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 demonstrates automated finishing on Sikorsky helicopter components.

4.2.1 Hardware

For the robotic implementation of ADACS, a Cincinnati Milacron T3–646 six-axis electric robot is used as a macropositioner and the CADET is used as a micropositioner and force control tool. A servo table is used to fixture the part to be chamfered. The T3-646 is shown below in Figure 4.

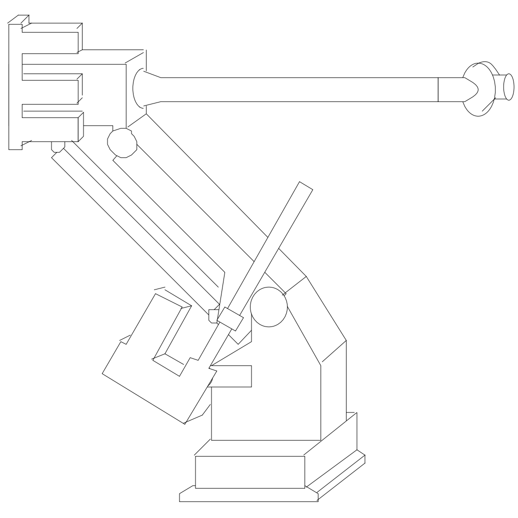

Figure 4. T3-646 Robot

A battery of tests was performed on the T3-646 to determine induced resonant frequencies and robot–induced dynamic path errors. These tests were performed to determine the tool performance requirements necessary to filter out these disturbances. A mock tool designed to simulate the mass distribution of the ADACS chamfering tool was mounted to the flange of the robot. An accelerometer was mounted to either of the five arms of the mock tool and was used in conjunction with an impact hammer to determine the robot dynamic characteristics. The accelerometer was mounted on the mock tool and the mock tool was struck in several directions and the hammer force and robot acceleration times were recorded using a data analyzer. These tests were performed to obtain the performance requirements for a new active tool being designed by the engineers at UTRC. The CADET, Figure 5, is a voice coil actuator active tool. The test procedures and results are discussed in detail in Engel et. al. (1).

The workcell 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.

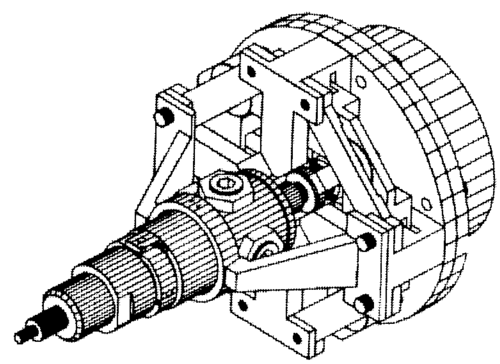

Figure 5. Chamfering and Deburring End-of-arm Tool (CADET)

4.1.2 Software

The ADACS makes use of various off the shelf software packages. These products offer hooks into their systems that allows 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.

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.

4.2 Machine Tool Implementation

At the writing of this paper, a machine tool implementation of the ADACS system is being integrated at Pratt & Whitney located in East Hartford, Connecticut for automated finishing of engine casings.

A machine tool platform could allow the finishing of a part to take place on the same machine in which it was manufactured. There would be no need to remove the part from the machine and perform the finishing operations elsewhere in the manufacturing facility, which can add hours to the manufacturing time.

4.2.1 Hardware

The CADET will be integrated with a K&T Series 200 CNC machine tool. This implementation will be performing automated finishing on engine casings for Pratt & Whitney.

4.2.2 Software

The K&T machine tool will run a general purpose Delta-Tau CNC machine tool controller based with additional requirements based on the capabilities of the CADET tool.

5 CONCLUSION

ADACS supplies a CAD–based graphical interface of a part, wherein the operator uses a mouse to select feature edges to chamfer and optionally supplies chamfer forces and cutting strategies. ADACS subsequently generates the finishing process model and performs the finishing operation that includes the following key features:

- graphically–instructed edge definition and automated edge extraction from CAD model
- feature recognition based on edge–to–edge transitions including free–space to contact, continuous versus discontinuous, and force information.
- feature–based knowledge to define material removal. Feature–based knowledge defines a process model based on part material, cutting force, depth of chamfer, feed rate and spindle speed. Tool wear estimation is also monitored.
- feature based generation of multiple finishing paths to prevent dead-reckoning problems, and scarring.
- tightly motion and tool forces control that allows linear and curvilinear (arc, ellipses, etc.). Ramping of tool forces and spindle speeds allows smooth transition from free–space to contact–space.
- active force control tooling to compensate for positioning errors.

From our experiences, the ADACS system has proven a flexible and useful system. It has applicability beyond part chamfering and deburring and in the future will be adapted to perform feature-based grinding and welding, as well as other edge–related feature applications.

6 REFERENCES

1. 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.

2. 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.

3. 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.

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

5. Guptill, 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.

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

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

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