Robotic Machining: A Force-Control-Based Fast Programming Method

Jingguo Ge, Fusheng Tan, Hui Zhang
ABB Robotics, Corporate Research Center
Shanghai, China
jingguo.ge@cn.abb.com, simon-fusheng.tan@cn.abb.com, hui.zhang@cn.abb.com

***Abstract*— A force-control-based robotic machining system is introduced in this paper. Firstly, the machining process is analyzed. Three key technologies: lead-through teaching, path learning and template-based automatic program generation are described and then a fast programming method is proposed. It is proved practically that it can shorten the programming time from days to minutes in robotic machining programming.**

***Keywords*—Force Control, Force/Torque Sensor, Robot Machining, Lead-Through Teaching, Path Learning, Automatic Program Generation**

I. Introduction

Great efforts have been made by scholars and engineers world wide to apply force control technology in industrial robot application. And a lot of papers and reports have been published in the area of robot force control machining [1-7]. But programming for machining process is still a hard work for robot programmer at field. Currently, the machining process is still a labor intensive domain. Developed countries can not afford the labor cost in this domain and tried their best to transfer it to developing country, but the labor cost in developing country, such as China, is increasing rapidly. So the market has the requirement to use robot to instead of worker. Major robot vendors in the world have started competence in the domain of robot machining.

In Small and Medium Enterprises (SME) robots are not commonly found. An important reason besides the high investment is the necessary environment, especially the programming capabilities. To work with today's systems, the SMEs need to set up a robots department with programming engineers and trained service personal. These financial efforts do not pay up till now.

Traditional way of programming robot at field is jogging robot and then teaching targets one by one, but for machining process, it always need to teach dozens of even hundreds of targets in one path. And sometimes the work-piece requires very high surface quality and it need the user to teach targets very precisely at the corner of work piece. With this way, programming for a machining process always need several days. That's a very tedious and time-consuming process.

In this paper, a fast robot programming method is proposed, which distinctly differ from the traditional programming way with the capability of reducing programming time from days to minutes in the robot machining programming area. In the following sections, the method will be described in detail.

II. Force Control Robotic Machining System

The standard solution in market for force-controlled-based robotic machining system is the solution based on PushCorp device [8] (external force control tooling). It is reported that some robot companies had offered such solutions [9]. It has the benefit of faster response but limited to one dimension compliance and external controller.

ABB has offered the solution that integrated force control devices with six dimensions integrated with robot controller. The proposed force-control-based robotic machining system, as shown in Figure 1, includes mainly the following hardware:

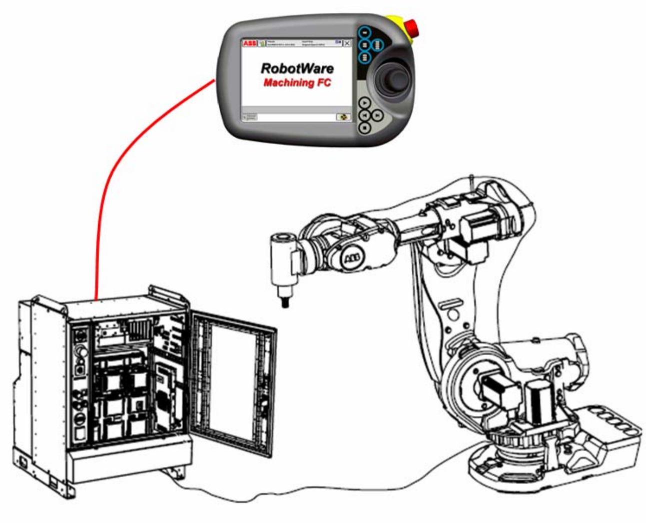

Figure 1. Force-control-based robotic machining system

- An ABB Robot (IRB6600, 4400, 2400, 140 et al.) with IRC5 controller and Flex Pendant (or Teach Pendant)
- An axis computer DSQC635 with support for PMC card (PCI Mezzanine Card),
- A PMC card. It is a data acquisition card. The card is mounted in the slot of axis computer. It is used to read data from F/T sensor and transfer data to the controller.
- A Robot mounted or room fixed F/T (force/torque) sensor. The sensor type can be ATI OMEGA160, THETA or DELTA according to the payload of manipulator, which is a 6 DOF (Degree of Freedom) sensor (forces in three axes direction and torques around three axes). It is

mounted on the wrist of the manipulator normally (room fixed is also possible) and connected to a PMC card through a signal cable.

- A Cable between F/T sensor and PMC card
- A tool for machining. The tool or work piece is mounted on the flange of the sensor to make the sensor can sense the forces or torques applied on them.

III. PROCESS ANALYSIS

The machining process has its special characteristics which differ from the other industrial processes. For machine tending process, the robot is mainly used to pick and place part [10]. But machining process is different. Grinding/polishing and deburring are two kinds of widely used machining processes. Take them as an example. For grinding/polishing processes, the robot is required to press on the work piece surface with a constant force while moving. The movement will follow the surface and adjust to surface variation. For the deburring process, the robot is required to go along the programmed path precisely, but it have to slowdown when the contact force increases (cutting bur) and resume the speed when contact force disappears.

Although different machining processes, such as grinding/ polishing, deburring/deflashing, milling, et al., is different to each other, the common characteristics among them still can be found.

A. Need to teach mass targets

Generally speaking, a machining program needs to teach more targets than a machine tending program. A machining path with dozens of even hundreds of targets is common. This made machining programming, as mentioned above, very tedious and time-consuming.

B. A machining path can be divided into three phases

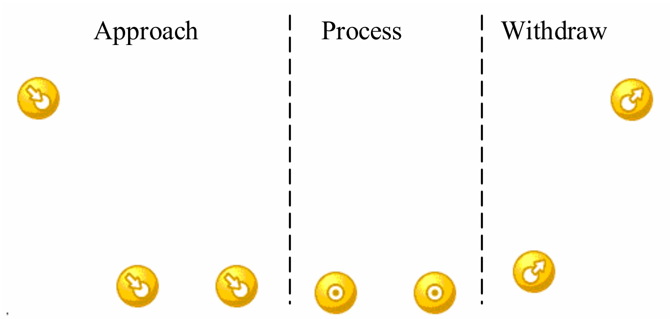

Figure 2. Illustration of a typical machining path (side view)

A typical path of machining is shown in Figure 2. It can be divided into three phases, i.e., the approach phase, the process phase and the withdraw phase. The approach phase is phase before process. During this phase, the tool does not contact with work piece and robot is position-controlled. It is used to initializing and setting force control parameters or switching the robot controller to force control state. The process phase is the real working phase that material is removed from the work piece. During this phase, the tool may be required to press on the work piece tightly. It can be force-controlled (for grinding/polishing), or position-controlled (for deburring). The withdraw phase is the post processing phase. It is be used to clean resource of the controller or switching the robot controller from force control state to position control state [11].

IV. KEY TECHNOLOGIES

The lead-through and path learning technologies proposed by Hui Zhang overcome the shortcoming of mass targets in robotic machining programming [12]. A template-based automatic program generation method is proposed in this paper to fully use the three phases' characteristic of machining path. Combines the three technologies together, a fast programming method is proposed in this paper. Lead-through method can simplify the teaching of robot target and decrease the requirement for precision of taught targets. Secondly, the path learning can improve the accuracy of path with less interaction between human and machine. Thirdly, the automatic program generation liberates robot engineer from writing code line by line with creating robot program on-line. These three technologies can decrease robotic machining programming time from days to minutes.

A simple introduction about the three technologies will be given in the following sections.

A. Lead-through teaching

Lead-through teaching means the robot can be led by hand instead of the traditional way of jog with the joy stick. The robot can be led linearly or rotationally. This made jogging robot much easier. Since the robot movement is controlled by hand directly, it is collision free.

Both of the lead-through teaching and the path learning utilize the feedback signal of F/T sensor. In this system, when user push, drag or rotate the tool or work piece mounted on the sensor's mounting plate, the F/T sensor senses force and torque applies on the sensor, the PMC card samples and transfers the data to controller, then the controller can make decision and drive motors to move to the directions of applied forces or torques.

The equipped F/T sensor made the robot much intelligent to sense the force applied on it (the sensor's mounting plate). The high performance of the controller, whose control cycle is 4ms, made it have good response performance.

B. Path learning

Path learning is to learn a path along the taught path. It is an automatic process.

Figure 3 shows the principle diagram of path learning.

User input the task performance criteria (the accuracy, minimum and maximum distance between targets) and process parameter (the force magnitude, speed). the controller drive the motor move along the taught targets and press tightly on the work piece surface. At the same time, the targets is sampled and recorded.

During learning:

- The robot will follow the previously taught path.
- The robot will move along the work piece with specified force and speed. The robot will press tightly on the work piece with a constant force. The force direction is being

adjusted every 4ms.

- A series of targets will be recorded. The recorded targets are optimized according to the preset performance criteria. More targets will be kept in the corner and less in the straight edge.

After learning, an accurate path which closely fits the contour will be created.

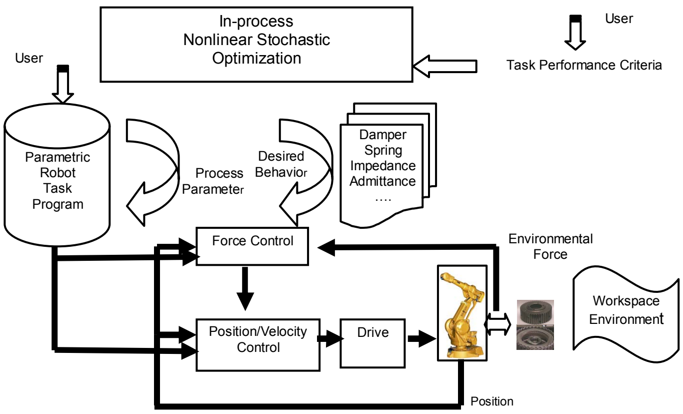

Figure 3. Path learning principle diagram

There are three key points in path learning.

1) Path coordinate system In path learning, the force is controlled on the normal direction of work piece surface. In order to make the force direction always perpendicular to the work piece in run time, a dynamic coordinate system is setup based on tool coordinate system, i.e., the path coordinate system. The coordinate system is defined as illustrated in Figure 4.

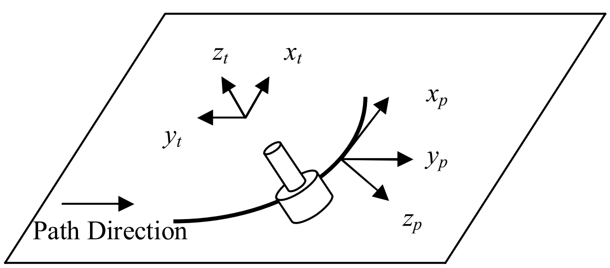

Figure 4. Path coordinate system definition

Use x_t, y_t and z_t to represent the axes of the tool coordinate system, use x_p, y_p and z_p to represent the axes of the path coordinate system. Then the definition of path coordinate system is,

- x_p is defined as the tangent of the path.
- y_p is defined as the cross product of x_p and z_t. $y_p = x_p \times z_t$(assume the z direction of tool coordinate system does not parallel with the tangent of the path).
- z_p obey the right hand rule. $z_p = x_p \times y_p$.

The path coordinate system is adjusted in controller every 4ms when robot moving along the programmed path.

2) Force direction setting Since the force is a space vector, it has magnitude and direction. Its magnitude can be set by user interactively. But its direction in path coordinate system is very hard to set for user. So the force direction needs to be set automatically.

The force can be applied on each direction among the six directions of path coordinate. It can be positive x_p+, y_p+, z_p+, or negative x_p-, y_p-, z_p-. Take robotic grinding as an example, the machining pattern can be side grinding or face grinding (see Figure 5).

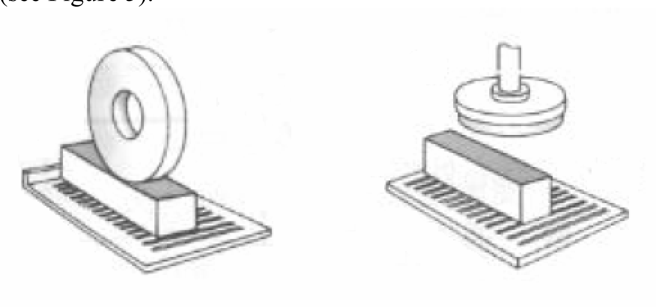

Figure 5. Side grinding (left) and face grinding (right)

Consider the basic fact that no matter side grinding or face grinding, the tool have to approach to the work piece and then move along the surface to remove material. So the space relationship is a restrict condition to judge the force direction. In Figure 6, mark the three targets as A_1, A_2 and A_3, the plane decided by A_1, A_2 and A_3 is perpendicular to the surface, A_1 is the above the surface, A_2 and A_3 is close to the surface. Mark the vector from A_1 to A_2 as v_1 and the vector from A_2 to A_3 as v_2, then,

$$v_3 = v_1 \times v_2 \quad (1)$$

$$v_n = v_2 \times v_3 \quad (2)$$

v_n is almost the normal direction of the surface, find the vector among x_p+, y_p+, z_p+, x_p-, y_p- and z_p- which has the minimum angle to v_n. Then it is the direction of force.

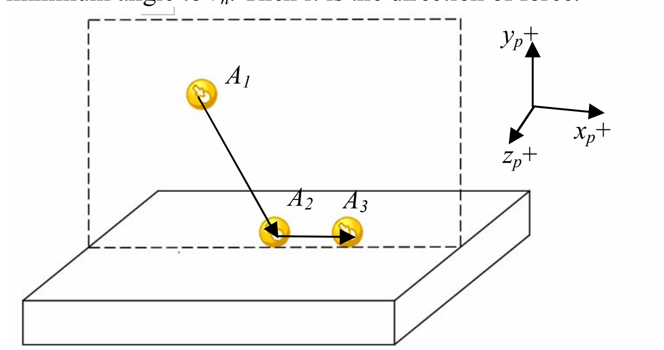

Figure 6. Force direction judgment

3) Mapping between taught targets and learned targets After learning, the path can have more or less targets than the taught path. It depends on the required precision. The parameters for learned targets will be map to and interpolate from the nearest taught targets in path length, which include speed, force, zone, et al.

For the benefit of the path learning, the user can teach a path roughly then use the automatic path learning to get an accurate path. It saves lots of time by using the function.

C. *Automatic program generation*

The lead-through teaching and path learning can get position and pose information of all targets. The other information, such as the force, speed and zone can be input by user interactively.

As for the three phases' characteristic of machining process, the robot machining path program can have the same structure. It makes it feasible to create machining program with a predefined template after get all the information of targets. The targets information includes the target's position, pose, movement type, speed, force et al. The template describes the sequence of the robot actions.

V. Software Design

Based on the key technologies mentioned above, an application has been implemented on the ABB IRC5 robot system to simplify robot machining programming. The Figure below shows the architecture of the application, it is a three layer application, see Figure 7.

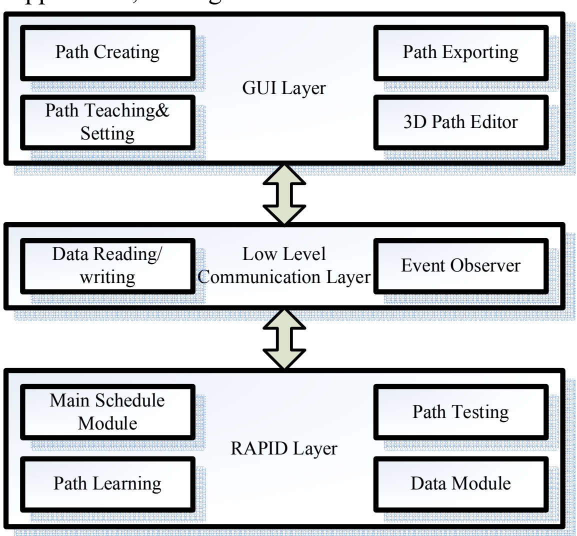

Figure 7. Three layer architecture of the machining application

Figure 8 shows the class diagram designed for the machining cell.

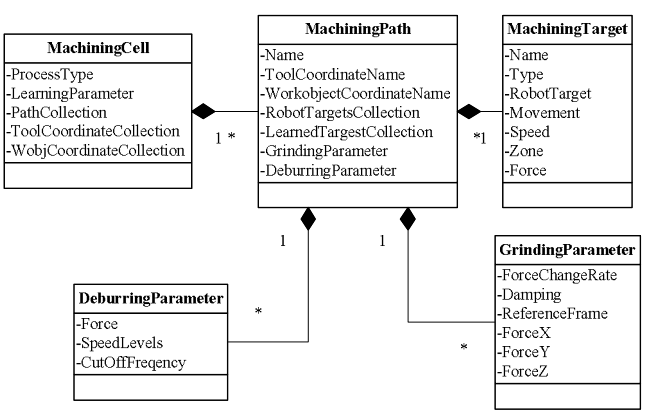

Figure 8. Class diagram of machining cell

The application has the following features:

- Wizard guided programming style with graphical user interface. User can create program intuitively by following the wizard step by step without learning more.
- GUI support for the lead-through teaching and path learning. User can create a machining path in minutes.
- 3D path editor, conceptual path editor and fine tuning editor. All of them can be used to teach path and modify path. The 3D path editor is more intuitive to show space relationship of targets, the conceptual path editor is much suitable to create path, the fine tuning editor is more suitable for fine tuning and set those parameter except position and pose. In Figure 9, the user can modify the path such as add points, delete points or modify position. After the path points are taught, the user can view the path in 3D view in Figure 10 or do fine tuning in Figure 11.

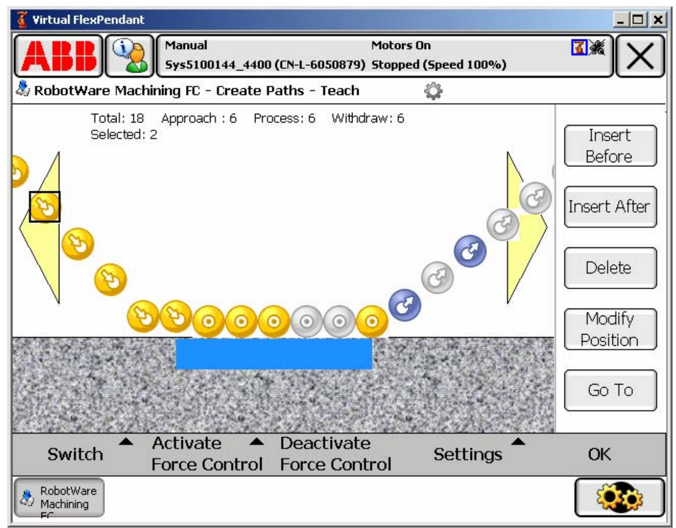

Figure 9. Snapshot of conceptual view of machining path. Press Insert Before/After button to insert point at selected point. Press delete button to delete selected point. Press Modify Position button to record current point value. Press Go To button to move robot to selected point position.

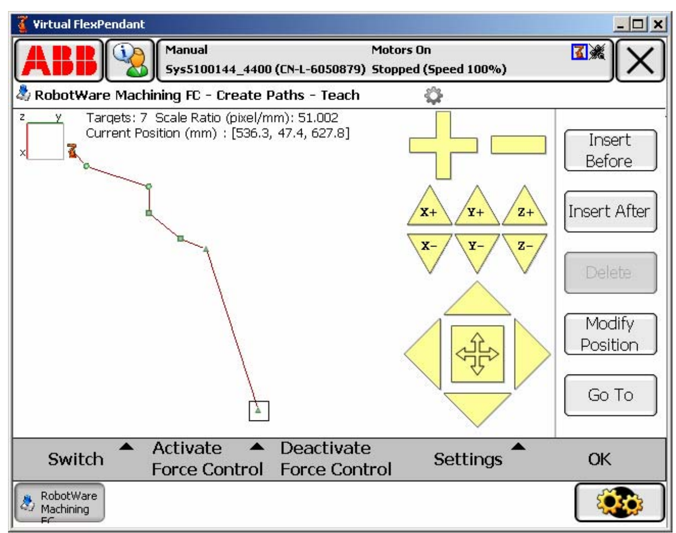

Figure 10. Snapshot of 3D path editor. User can press left-top cub to rotate the 3D view. User can also press right graphics button to modify the view. Left figure is the view of the path.

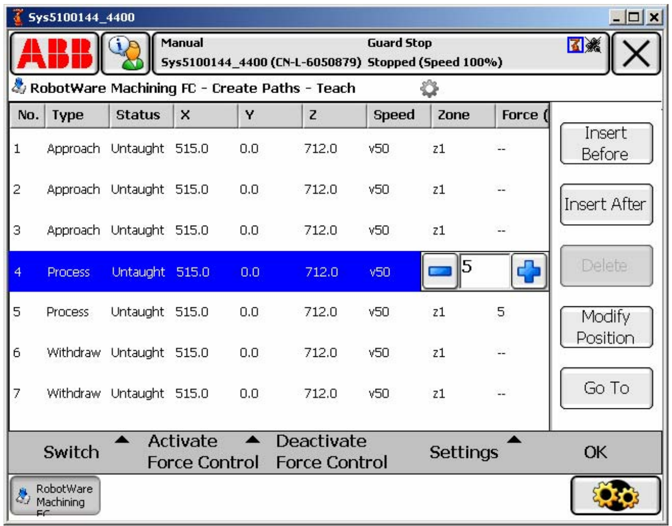

Figure 11. Snapshot of fine tuning editor view. Select a list in the column. Press force area and it is editable. Press plus or minus button to fine tune the value.

- Force data chart in real time allows display of actual force and speed variations during production. This makes it easier for operator to monitor the process and diagnose program at field. See Figure 12.

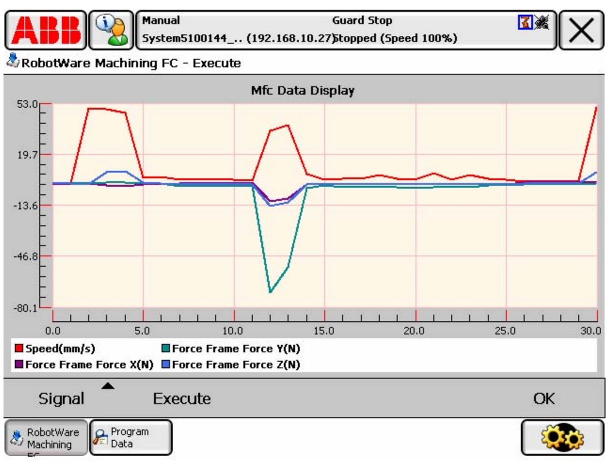

Figure 12. Snapshot force& speed chart in real time. Use can monitor force value and speed value during machining process

- Safe to operate. If the controller detected collision occurred it would quit force control mode and stop immediately.

- The created path program can be easily integrated into user's programs.

Figure 13 shows the RAPID (ABB robot language) code generated from the template for robotic grinding process.

```
MODULE mfcPath1

VAR robtarget T1Approach1 := ...
VAR robtarget T1Approach2 := ...
VAR robtarget T1Approach3 := ...
VAR robtarget T1Process1 := ...
VAR robtarget T1Process2 := ...
VAR robtarget T1Withdraw1 := ...
VAR robtarget T1Withdraw2 := ...

PROC RunPath1 (bool bRecover, bool bSpindleOn)
        FCDeact;

        IF bRecover = true THEN
                FCCalib  tool0_LD \Recovery;
        ELSE
                FCCalib  tool0_LD;
        ENDIF

        IF bSpindleOn = true THEN
                SpindleOn;
        ENDIF

        MoveL T1Approach1, v50, z1,tool0\wobj :=wobj0;
        MoveL T1Approach2, v50, fine,tool0\wobj :=wobj0;
        FCPress1LStart  T1Approach3,v50  \Fx:=0  \Fy:=0  \Fz:=-5,50
\ForceFrameRef:=FC_REFFRAME_WOBJ  \ForceChange:=50
\DampingTune:=100  \TimeOut:=5,  \UseSpdFFW,  z1,tool0
\wobj:=wobj0;

        FCPress1L T1Process1, v50, 5, z1;
        FCPress1L T1Process2, v50, 5, z1;

        FCPressLEnd  T1Withdraw1,v50,\ForceChange
:=50,\ZeroContactValue := 2.5;
        MoveL T1Withdraw2, v50, z1,tool0\wobj :=wobj0;

        IF bSpindleOn = true THEN
                SpindleOff;
        ENDIF

ENDPROC

ENDMODULE
```

Figure 13. Automatic generated RAPID routine

VI. Discussion and Conclusion

An experiment system is setup using the ABB IRB140 robot and ATI Delta F/T sensor, as shown in Figure 14. The developed application, which is based on lead-through teaching, path learning and automatic program generation technologies, is tested in laboratory, the experiments shows it has the following advantages.

- *Reduced programming time.* Up to 90% faster to program grinding applications by allowing the robot to "feel" the surface. Automatic path learning saves up to 90 % programming time in deburring applications.

- *Shorter cycle time.* Up to 20% faster deburring applications as the robot adapts to surface defects.

- *Longer tool life.* Up to 20% longer tool life as there is consistent wear by avoiding tool and work piece collisions.

Figure 14. Robotic grinding experiment system

The developed application has been shown in GIFA 2007 (International Foundry Exhibition and Technical Forum, GIFA is the abbreviation of its Germany name) in Germany and dozens of installation has been setup in customer factory. The feedback has shown its great potential in improving the productivity, and these new technologies are really very helpful to user and can reduce programming time evidently. But the product is still not so good enough to meet the habits of engineers at field.

In the following developing plan, usability features and machining process knowledge at field would be considered into the product. Some selection items are the followings: to support graphics simulation offline for the machining process, to support multi-points selection in fine tuning editor view, to consider more safety issues.

ACKNOWLEDGMENT

Special thanks to ABB United States Corporate Research Center and Sweden Corporate Research Center. They had done lots of research and pre study work to make this application package completed.

REFERENCES

[1] JN. Pores, “Programming Industrial Robotic and Automation Equipment,” Vol. 16, No. 1, Industrial Robot, 2000

[2] M, Ostring, S. Gunnarssson, M. Norrlof, ”Closed-loop identification of an industrial robot containing flexibilities,” 11, Control Engineering Practice, 2003, pp.291-300

[3] S. Nahavandi, M. J. Uddin, Y. Nasu, H. Trinh, M. Saadat, ”Automated robotic grinding by low-powered manipulator,” 23, Robotics and Computer-Integrated Manufacturing, 2007, pp.589-598

[4] V. Mallapragada, D. Erol and N. Sarkar, “A New Method of Force Control for Unknown Environments,” Vol. 4, No.3, International Journal of Advanced Robotic System, 2007, pp.313-322

[5] S. Satori, “Polishing of Mould Surfaces,” Vol 43, No. 2, Periodica Ploytechnica SER. Mech. Eng., 1999, pp.113-120

[6] Z.X.Gan, H.Zhang and J.J.Wang, “Behavior-Based Intelligent Robotic Technologies in Industrial Application, ” Lecture Notes in Control and Information Sciences 362,2007, pp.1-12.

[7] Zengxi Pan, Hui Zhang, Zhengqi Zhu and Jianjun Wang, “Chatter Analysis of Robot Machining Process,”Journal of Materials Processing Technology, 173(2006), pp.301-309.

[8] http://www.pushcorp.com/

[9] http://www.roboticsonline.com/public/articles/articlesdetails.cfm?id=2341

[10] J.G.Ge and X.G.Yin, “An Object Oriented Robot Programming Approach in Robot Served Plastic Injection Molding Application, ” Lecture Notes in Control and Information Sciences 362,2007, pp.91-97.

[11] ABB application manual, Force Control for Machining, 3HAC027595-001_revB_en. 2007.

[12] Zhang Hui, Brogardh Torgny, Wang Jianjun, Zhang George and Gan Zhongxue, “Method of and Apparatus for Automated Path Learning,” U.S. Patent, PCT/US2006/004677.