

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Robotic Machining from Programming to Process Control

Zengxi Pan<sup>1</sup> and Hui Zhang<sup>2</sup>

<sup>1</sup>*Faculty of Engineering, University of Wollongong, Australia*

<sup>2</sup>*ABB Corporate Research China*

## 1. Introduction

Cleaning and pre-machining operations are major activities and represent a high cost burden for casting producers. Robotics based flexible automation is considered as an ideal solution for its programmability, adaptivity, flexibility and relatively low cost, especially for the fact that industrial robot is already applied to tend foundry machines and transport parts in the process. Nevertheless, the foundry industry has not seen many success stories for such applications and installations due to the several major difficulties involved in robotic machining process with a conventional industrial robot. (Pan, 2006)

The first difficulty is the generation of robot motion for a complex workpiece. Secondly, the lower stiffness of articulated robot manipulator presents a unique disadvantage for machining of casting parts with complex geometry, which has non-uniform cutting depth and width. As a result, the machining force will vary dramatically, which induces uneven robot deformation. The third difficulty is the deformation caused by the interaction force between the tool and the workpiece, especially for milling process, which generates large cutting forces. The stiffness for a typical articulated robot is usually less than 1 N/ $\mu\text{m}$ , while a standard CNC machine very often has stiffness greater than 50 N/ $\mu\text{m}$ . As a result, force induced deformation is the major source of the inaccuracy of finished surface. The fourth difficulty is chatter/vibration occurred during the machining process.

Most of the existing literature on machining process, such as process force modelling (Kim, Landers & Ulsoy, 2003), accuracy improvement (Yang, 1996) and vibration suppression (Budak, & Altintas, 1998) are based on the CNC machine. Research in the field of robotic machining is still focused on accurate off-line programming and calibration. As the chatter analysis was discussed in a separate paper (Pan & Zhang et, al, 2006), our focus here is to address the first three major issues in robotic machining process.

This chapter is organized in six sections. Following this introduction section, section two presents an active force control platform, which is the foundation of various control strategies for solving difficulties in robotic machining processes. Section three addresses the programming issues for a part with complex contour. With two force control strategies, lead-through and path-learning, robot programming is made easy and efficient. Section four and five present two real-time process control techniques. The Controlled Material Removal Rate (CMRR) greatly reduces the process cycle time of the robotic machining operation,

while the real-time deformation compensation improves the quality and accuracy. The focus of these two sections will be the implementation of advanced control strategies and further analysis of robot stiffness modelling, as the preliminary research outcomes for CMRR and deformation compensation have been already introduced in (Wang, Zhang, & Pan, 2007). Experimental results are presented at the end of these sections. A summary and discussion is provided in section six.

## 2. Force Control Platform

The active force control platform is the foundation of the strategies adopted to address various difficulties in robotic machining processes. It is implemented on the most recent ABB IRC5 industrial robot controller which is a general controller for a series of ABB robots. The IRC5 controller includes a flexible teach pendant with a colourful graphic interface and touch screen which allows user to create customized Human Machine Interface (HMI) very easily. It only takes several minutes for a robot operator to learn the interface for a specific manufacturing task and it is programming free. An ATI 6 DOF force/torque sensor is equipped on the wrist of the robot to close the outer force loop and realize implicit hybrid position/force control scheme. The system setup for robotic machining with force control is shown in Fig. 1.



Fig. 1. System setup for robotic machining with force control

The force controller provides two major functions to make the entire programming process collision free and automatic. First function is lead-through, in which robot is compliant in selected force control directions and stiff in the rest of the position control directions. To change the position or orientation of the robot, the robot operator could simply push or drag the robot with one hand. The second function is called path-learning, in which robot is

compliant in normal to the path direction to make the tool constantly contact with the workpiece. Thus, an accurate path could be generated automatically.

During the machining process, the force controller provides two more functions to achieve deformation compensation and CMRR. In both case, robot is still under position control, that is, stiff at all directions. Deformation compensation is achieved by update the target position of position loop based on the measured process force and robot stiffness model, while robot feed speed is adjusted to maintain constant spindle power consumption for CMRR. These two strategies are complementary to each other since CMRR adjusts robot speed at feed direction and deformation compensation adjusts the reference target at the rest of the directions. The detailed control strategies for process control of robotic machining will be explained in section four and five respectively.

### 3. Rapid Robot Programming

Although extensive research efforts have been carried out on the methodologies for programming industry robots, still only two methods are realistic in practical industrial application, which are, on-line programming (jog-and-teach method) and off-line programming (Basanez & Rosell, 2005)(Pires, et al., 2004). On-line programming relies on the experience of robot operators to teach robot motions by jogging the robot to the desired positions using teaching device (usually teach pendent) in real setup. Off-line programming generates the robot path from a CAD model of the workpiece in a computer simulated setup. The idea of programming by demonstration (PbD) has been proposed long time ago, while requirement of additional hardware devices and complicated calibration process make it unattractive in practical applications. The major advantage of the PbD method proposed here is that no additional devices and calibration procedures are required. The only sensor implemented for force feedback is an ATI 6 DOF force/torque sensor. This simple configuration will minimize the cost and simplify the complexity of the programming process greatly.

#### 3.1 Lead-Through

Lead-through is the only step requires human intervention through the entire PbD process. The purpose of lead-through is to generate a few gross guiding points, which will be used to calculate the path frame in path-learning as shown in Fig. 2. The position accuracy of these guiding points is not critical because these guiding points are not the actual points/targets in the final program and they will be updated in automatic path-learning. However the orientation of these points should be carefully taught since it will determine the path frame and will be kept in the final program.

Theatrically all six DOFs could be released under force control and the user can adjust both position and orientation of the robot tool at the same time. In practice, we found it is almost impossible to adjust the tool orientation accurately by push/pull with a single hand. Thus, a force control jogging mode is created, under which the operator could push/pull the robot tool to any position easily and change the robot tool orientation using the joystick on the teach pendent. Since this jogging is under force control, collision is avoided even when the tool is in contact with the workpiece. As the instant position and orientation of the robot tool is displayed on the teach pendant, the operator could make very accurate adjustment on each independent rotation axis.

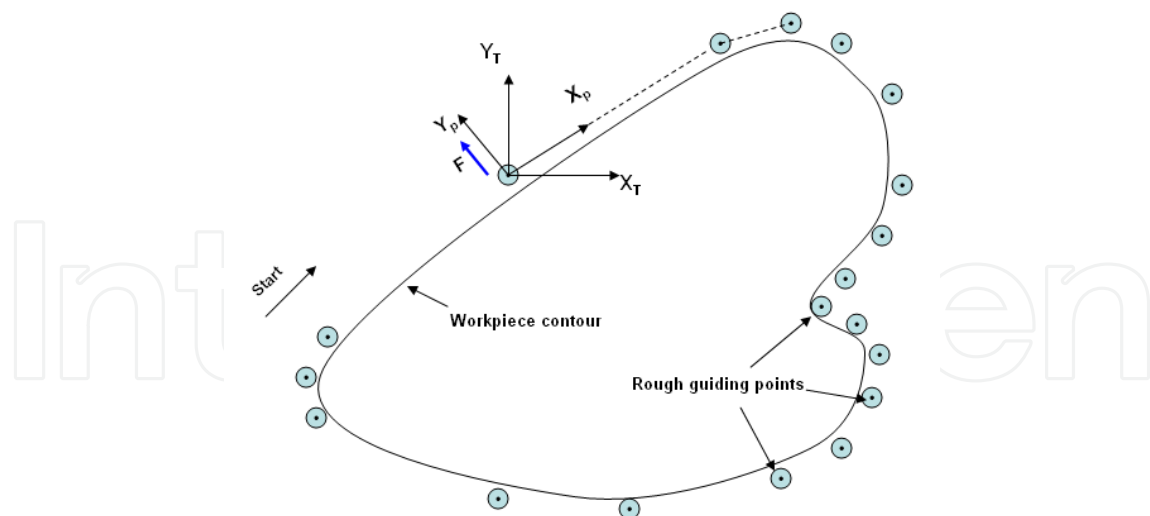


Fig. 2. Lead-through and path learning

### 3.2 Automatic Path-Learning

A robot program based on gross guiding points taught in lead-through is then generated. This program path, consisted of a group of linear movements from one guiding point to the next, is far different from the actual workpiece contour. The tool fixture would either move into the part or too far away from it.

During the automatic path-learning, the robot controller is engaged in a compliant motion mode, such that only in direction  $Y_p$ , (Fig. 2.) which is perpendicular to path direction  $X_p$ , robot motion is under force control, while all other directions and orientations are still under position control. Further, it can be specified in the controller that a constant contact force in  $Y_p$  direction (e.g., 20 N) is maintained. Because of this constrain, if the program path is into the actual workpiece contour, the tool tip will yield along the Y axis until it reaches the equilibrium of 20N, resulting a new point which is physically on the workpiece contour. On the other hand, if the program path is away from the workpiece, the controller would bring the tool tip closer to the workpiece until the equilibrium is reached of 20N.

While the robot holding the tool fixture is moving along the workpiece contour, the actual robot position and orientation are recorded continuously. As described above, the tool tip would always be in continuous contact with the workpiece, resulting a recorded spatial relationship that is the exact replicate between the tool fixture and the workpiece. A robot program generated based on recorded path can be directly used to carry out the actual process.

### 3.3 Post Processing

After tracking the workpiece contour, the data from logging the robot position have to be filtered and reduced to generate a robot program. The measurements around sharp corners are often influenced by noise due to high dynamic forces, which has influence on the contact force. By using a threshold for the maximum and minimum acceptable contact force, the measurements influenced by this type of noise are removed. This is called force threshold filtering.

The amount of the targets from automatic path-learning are disproportionately large since the robot controller can record the points as fast as every 4 ms. An approach, namely deviation height method, is used to approximating the contour by straight-line segments. As shown in Fig. 3, a straight line is made from a certain starting point on the contour to the current point. The deviation height is calculated between the line and each of the intermediate points. The deviation height is the length of the normal vector between the point and the line. The current point is displaced along the contour until the deviation height exceeds a certain limit. The previous point is then used as starting point for the next line segment. This continues until the whole contour is approximated with straight-line segments. From the reduced data, a robot program is generated in a standard format. The user could specify tool definitions, desired path velocity and orientation of the tool.

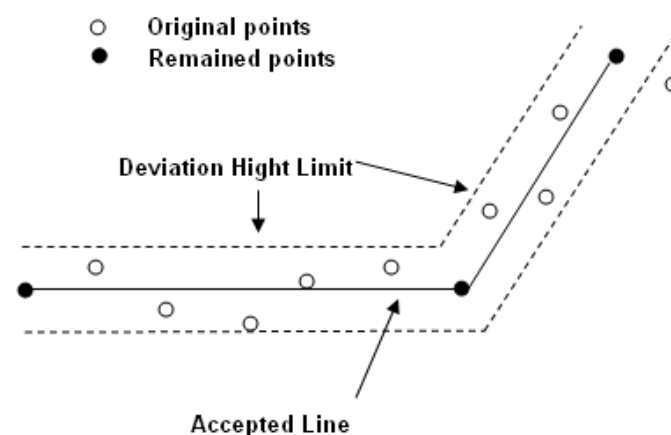


Fig. 3. Deviation height method

### 3.4 Experimental Results for PbD

With force control integrated in IRC5 controller, PbD method is available for a group of ABB industrial manipulators. An automatic deburring system using IRB 4400 manipulator is designed to clean the groove of a water pump to guarantee a seamless interface between two pump surfaces, as shown in Fig. 4.

A 2 mm cutting tool, driven by ultra high speed ( $\sim 18,000$ rpm) air spindle is adopted to achieve this task. Since the groove is only about 5 mm wide and has contoured 2D shape, manually teaching a high quality program to clean the complete groove is almost impossible. Due to the process requirement, the cutting tool is always perpendicular to the surface of water pump. During path-learning, a contact force normal to the edge of 10 N is used, while the robot path learning velocity is set at 5mm/s. As shown in Fig. 5, the curvature of recorded targets changes dramatically along the path. The blue points represent the targets in the final cutting program, while the red points represent the offset targets in the test program. The average robot feed speed during the cutting process is about 10 mm/s, while the exact feed speed is determined by the local curvature, which is slower at sharp corner, to ensure a smooth motion throughout the path. The point reduction technique is performed on the filtered measurements. A deviation height of 0.2mm reduced the thousands of points recorded by the robot controller every 4 ms to about 300 points.

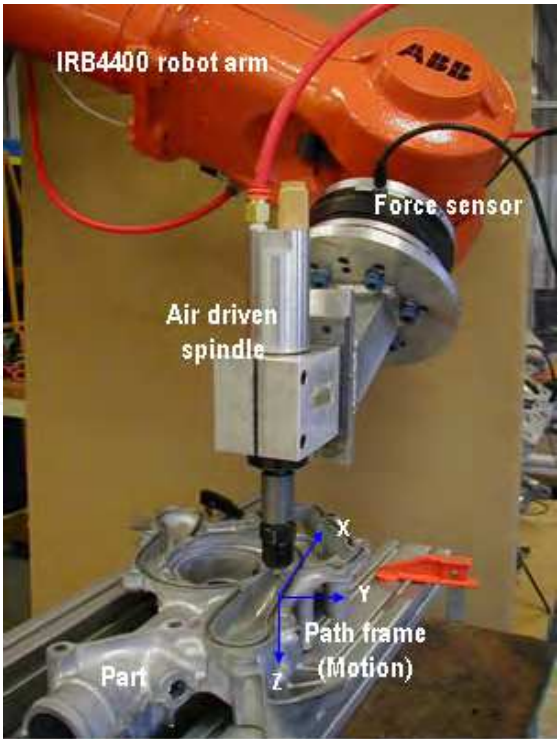


Fig. 4. Experimental setup for PbD

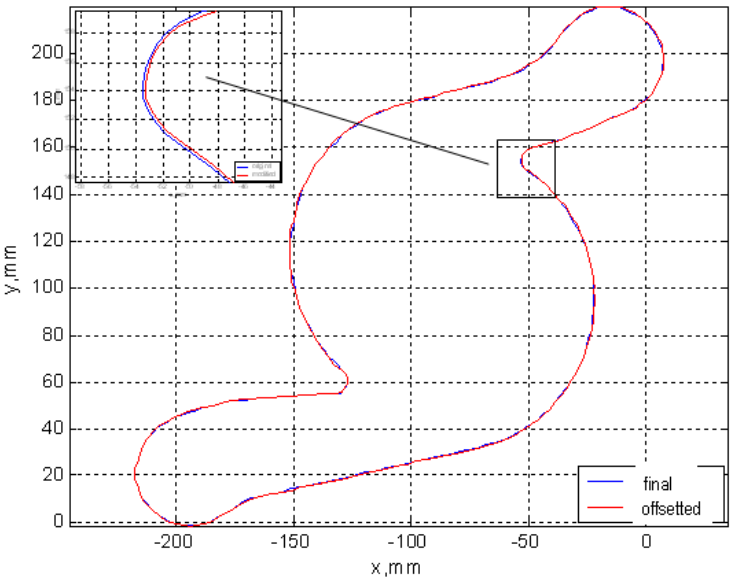


Fig. 5. Results from path-learning

With this programming strategy, generating a program for a water pump with complex contour, including more than three hundred robot target points, could be completed within one hour instead of several weeks by an experienced robot programmer. During this programming procedure, the operator is only involved with the first step of teaching the gross movement of the robot, while the bulk of the procedure is automated by the robot controller.

#### 4. Controlled Material Removal Rate

The MRR in machining process is usually controlled by adjusting the tool feedrate. In robotic machining process, this means regulating robot feed speed to maintain a constant MRR. Machining force and spindle power are two variables proportional to MRR, which could be used to control robot feed speed. With 6-DOF force sensor fixed on robot wrist, the cutting force is available on real-time. Most spindles have an analog output whose value is proportional to the spindle current. With force feed back or spindle current feed back, MRR could be regulated to avoid tool damage and spindle stall.

In most cases, the relationship between process force and tool feedrate is nonlinear, and the process parameters, which describe the nonlinear relationship, are constantly changing due to the variations of the cutting conditions, such as, depth-of-cut, width-of-cut, spindle motor speed, and tool wearing condition, etc. Most of the time, conservative gains have to be chosen in order to maintain the stability of the close-loop system, while trading off the control performances.

Three different control strategies, PI control, adaptive control and fuzzy control, are designed to satisfy various process requirements. PI control is easy to tune and is very reliable. Adaptive control provides a more stable solution for machining process. Fuzzy control, which provides a much faster response by sacrificing control accuracy, is the best method for applications require fast robot feed speed



Fig. 6. Robotic end milling process setup

##### 4.1 Robot Dynamic Model

A robotic milling process using industrial robot is shown in Fig. 6. The cutting force of this milling process is regulated by adjusting the tool feedrate. Since the tool is mounted on the robot end-effector, the tool feedrate is controlled by commanding robot end-effector speed. Thus, the robot dynamic model for this machining process is the dynamics from the command speed to the actual end-effector speed. The end-effector speed is controlled by the robot position controller. A model is identified via experiments for this position controlled close-loop system, which represents the dynamics from command speed to actual end-effector speed.

The dynamic model identified is given as

$$\frac{f(s)}{f_c(s)} = \frac{63s^2 - 45800s + 4330000}{s^3 + 575s^2 + 98670s + 4313000} \quad (1)$$

Where  $f(s)$  is the actual end-effector speed,  $f_c(s)$  is the commanded end-effector speed.

The dynamic model Eq. (1) is a stable non-minimum phase system, and its root locus is shown in Fig. 7.

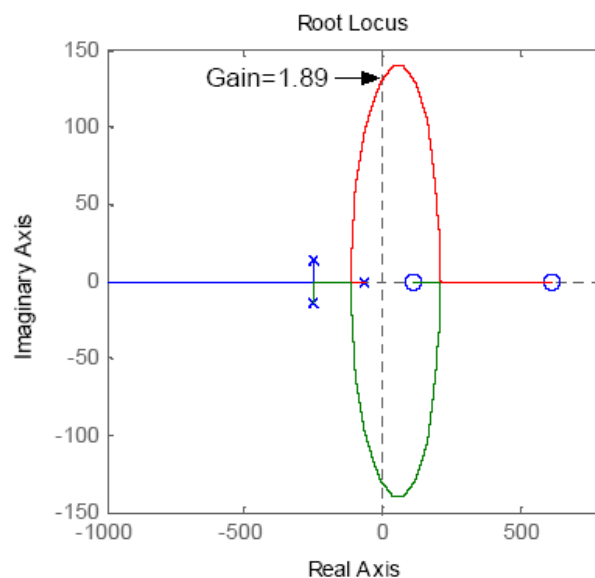


Fig. 7. Root locus of robot dynamic model

#### 4.2 Process Force Model

MRR is a measurement of how fast material is removed from a workpiece; it can be calculated by multiplying the cross-sectional area (width of cut times depth of cut) by the linear feed speed of the tool:

$$MRR = w \cdot d \cdot f \quad (2)$$

Where  $w$  is width-of-cut (mm),  $d$  is depth-of-cut (mm),  $f$  is feed speed (mm/s).

Since it is difficult to measure the value of MRR directly, MRR is controlled by regulating the cutting force, which is readily available in real-time from a 6-DOF force sensor fixed on the robot wrist. The relationship between the machining process force and the tool feed speed is nonlinear and time-varying, as shown in the following dynamic model (Landers & Ulsoy, 2000)

$$F = K_c d^\alpha f^\beta w^\gamma \frac{1}{\tau_m s + 1} \quad (3)$$

Where  $K_c$  is the gain of the cutting process;  $\alpha$ ,  $\beta$  and  $\gamma$  are coefficients, and their values are usually between 0 and 1.  $\tau_m$  is the machining process time constant. Since one spindle revolution is required to develop a full chip load,  $\tau_m$  is 63% of the time required for a spindle revolution. (Daneshmend & Pak, 1986) Since  $\tau_m$  is much smaller than the time constant of robot system, it is ignored here in the MRR controller design. Let,

$$K = K_c w^\gamma \quad (4)$$

K is considered as a varied process gain. Then, the force model is rewritten as a static model:

$$F = K d^\alpha f^\beta \quad (5)$$

The depth-of-cut,  $d$ , depends on the geometry of the workpiece surface. It usually changes during the machining process, and is difficult to be measured on-line accurately. The cutting depth is the major contributor that causes the process parameter change during the machining process.  $K$ ,  $\alpha$  and  $\beta$  depend on those cutting conditions, such as, spindle speed, tool and workpiece material, and tool wearing condition, etc, which are pretty stable during the cutting process. If the tool and/or the workpiece are changed, these parameters could change dramatically. But they are not changing as quickly as the depth-of-cut  $d$  does during the machining process as explained above. A force model, which is only valid for the specific tool and workpiece setup in ABB robotics lab is identified from experiments as

$$F = 23 d^{0.9} f^{0.5} \quad (6)$$

Eq. (6) models the process force very well from milling experimental data. The tool feedrate  $f$  is chosen as the control variable, i.e., to control the process force by adjusting the feed speed.

### 4.3 MRR Control Strategy

In roughing cycles, maximum material removal rates are even more critical than precision and surface finish. Conventionally, feed speed is kept constant in spite of variation of depth-of-cut during the pre-machining process of foundry part. This will introduce a dramatic change of MRR, which induces a very conservative selection of machining parameters to avoid tool breakage and spindle stall. The idea of MRR control is to adjust the feed speed to keep MRR constant during the whole machining process. As a result, a much faster feed speed, instead of conservative feed speed based on maximal depth-of-cut position, could be adopted. Fig. 8 illustrates the idea of MRR control while depth-of-cut changes during milling operation. (Pan, 2006)

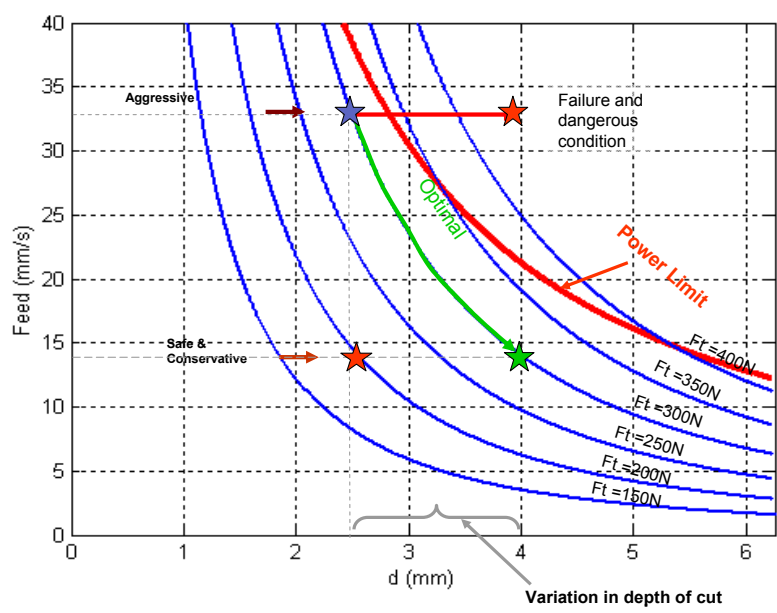


Fig. 8. Controlled material removal rate

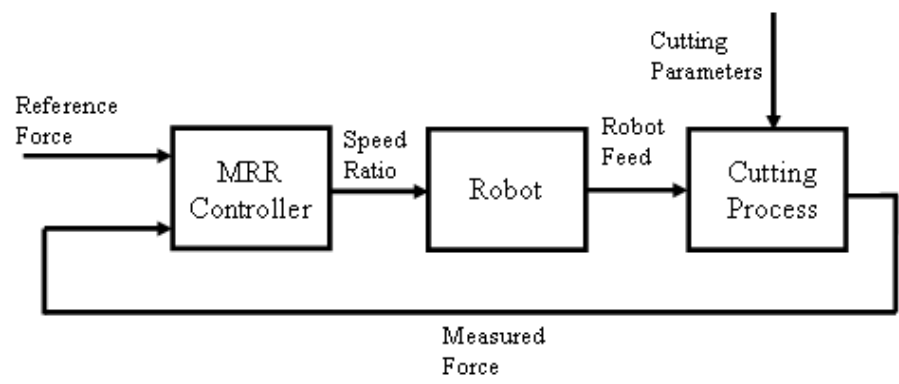


Fig. 9. The force control loop for CMRR

4.3.1 Force Control Sturcture

The block diagram of CMRR is shown in Fig. 9. The cutting force is controlled by varying the robot end-effector speed in tool feed direction. The difference between the reference force and the measured cutting force is input to the MRR controller. In actual implementation, the robot motion is planned in advance based on a pre-selected command speed. The output of MRR controller is a term called speed\_ratio, which is a ratio (e.g. from 0 to 1) of the actual robot feed speed to interpolate the reference trajectory in order to adjust the tool feedrate. Thus the command speed is the greatest speed robot can move. If the measured cutting force is larger than reference force, robot will slow down; otherwise robot will speed up until it reaches command speed. The CMRR function may implement several control approaches under the indirect force control framework. Three different control strategies, classical control (PI), adaptive control, and fuzzy logic control, will be introduced bellow.

### 4.3.2 PI Control

The cutting force model is nonlinear as described in Eq. (5), for controller design, it can be rewritten as

$$F = Kd^\alpha f^\beta = K_f f^\beta \quad (7)$$

Where  $K_f = Kd^\alpha$ . The effects of parameters  $K$ ,  $d$ , and  $\alpha$  to the process force are lumped into one parameter, force process gain  $K_f$ .

Define

$$F' = (F)^{1/\beta} \quad (8)$$

Together with Eq. (7), we get

$$F' = (F)^{1/\beta} = (K_f)^{1/\beta} f = kf \quad (9)$$

Where  $k = (K_f)^{1/\beta}$  is time-varying. Instead of controlling cutting force  $F$ , we control  $F'$  to follow the new command force, i.e.,  $F'_r = (F_r)^{1/\beta}$ , which is equivalent as controlling  $F$  to follow the original reference force  $F_r$ . By using Eq. (9), the nonlinear system is exactly linearized, and the linear system design technique can be applied to design a controller for the nonlinear system. PI type control is selected to achieve null steady-state error. The derivative term is not desirable due to the large noise associated with force readings.

The PI control in is given as

$$G_c = K_p + \frac{K_i}{s} \quad (10)$$

We put the zero of PI controller at -66.5 to cancel the slow stable pole of the robotic dynamic model. Since the zero of the PI controller is fixed, the proportional and integral gains will be given as

$$K_p = 0.015\alpha, K_i = \alpha \quad (11)$$

Where  $\alpha$  will be chosen to make the open loop gain of the whole system at the desired value. The magnitude of open loop gain, defined as  $kK_p$  determines the stability of the system. Conservative  $K_p$  and  $K_i$  are selected to ensure system still stable while the force process gain  $k$  takes the maximal value. The desired system response is that small overshoot for command feed speed.

### 4.3.3 Adaptive Control

Since depth-of-cut and width-of-cut are likely to change dramatically due to the complex shape of workpiece and varied bur size, the force process gain  $k$  will vary dramatically during the machining process. The fixed-gain PI control will surely have problems to



$$K_I = \frac{28.84}{V_r \hat{k}}, K_P = \frac{0.432}{V_r \hat{k}} \quad (14)$$

Where  $\hat{k}$  is the on-line estimation of  $k$  in Eq. (9). Eq. (14) is used as the self-tuning rules for the PI controller, which aims to maintain the open loop gain at 28.84.

The following standard recursive linear least square (RLS) method is used to identify  $k$  and  $\beta$  of Eq. (9)

$$\begin{aligned} k(t) &= \frac{P(t-1)x(t)}{\lambda + x^T(t)P(t-1)x(t)} \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + k(t)[y(t) - \hat{\theta}(t-1)x(t)] \\ P(t) &= \frac{1}{\lambda} [I - K(t)x^T(t)]P(t-1) \end{aligned} \quad (15)$$

Where  $\hat{\theta}(t) = (\ln \hat{k}(t) \quad \hat{\beta})$ ;  $y(t) = \ln F(t)$ ;  $x(t) = (1 \quad \ln f(t))^T$ ;  $t = 1, 2, 3, \dots$  is the sampling point;  $\lambda$  is the forgetting factor, which is usually chosen between 0.95 and 0.99. The on-line identified  $\hat{k}$  and  $\hat{\beta}$  are used in Eq. (9) and Eq. (14) respectively as the adaptive rules.

#### 4.3.4 Fuzzy Logic Control

Although PI control and adaptive control provide stable and zero static error solutions for MRR control, they are only feasible for applications with slow feed speed, such as end milling and grinding. Their response is limited by the open loop gain to maintain a stable performance. For deburring applications, where the cycle time is critical, faster feed speed up to 200 mm/s is usually required. Also, the variation of material to be removed (bur size) is more dramatic in deburring process. Even with the largest stable gain, the PI and adaptive controller could not response fast enough to prevent spindle stall or robot vibration. Derivative term (change of force) must be included in the controller to predict the force trend and achieve faster response. Since the force/spindle current signal is very noisy, it is not practical to expand the PI control to a complete PID controller. A more intuitive control method must be adopted here to address this problem since the change of force information is only critical at the moment when the cutting tool start to engage a large bur.

Fuzzy control is a very popular approach for performing the task of controller design because it is able to transfer human skills to some linguistic rules. Therefore, fuzzy control is often applied to some ill-defined systems or systems without mathematical models. In this robotic machining situation we use a Mamdani type fuzzy PD control law to regulate the machining force. In Mamdani method, fuzzy logic controller (FLC) is viewed as directly translating external performance specifications and observations of plant behavior into a rule-based linguistic control strategy.

A FLC is a control law described by a knowledge base (defined with simple IF . . . THEN type rules over variables vaguely defined -- fuzzy variables) and an inference mechanism to obtain the current output control value. The designed FLC has three inputs, force difference, filtered change of force difference, and previous output speed\_ratio, and one output change of speed\_ratio. The inputs are divided in levels in accordance with the observed sensor

characteristics and fuzzyfied using triangular membership functions.(Galichet & Foulloy, 1995) The output is fuzzyfied in the same way. The rule base is constructed using a methodology similar to that in the work of (Li, & Gatland, 1996). The rule base consist three groups of rules:

- 1) Force limit rule: Basic rules to speed up or slow down robot based on the difference of measured force and reference force. This group of rules perform similarly to classical control method.
- 2) Force trend rule: This group of rules are specially implemented to detect the large burs by evaluate the trend of force difference. Proper set of force trend rule could reduce overshoot of cutting force and achieves fast response.
- 3) System failure protection rule: Used for safety purpose. When speed\_ratio is already on lowest stage and process force is still high, robot will stop to avoid motor overload and robot vibration.

FLC generates change of speed\_ratio through evaluating various rules. Instead of changing speed\_ratio continuously as in classical PID control, speed\_ratio is set to several stages. The reason behind this is that continuously adjusting feed speed is not desirable for machining process because it increase tool wear and deteriorate surface quality. Since a too slow feed speed will change the chip generation mechanism, that is, tool becomes rubbing instead of cutting the workpiece; the minimal feed speed is also set. Although ideally more stages means more control accuracy, five stages (0.2, 0.4, 0.6 0.8, 1.0) would be enough for most applications. A special case is two-stage switching control which has only low or full speed. Two-stage switching control, which sacrifices control accuracy to achieve faster response, is a very attractive control method for many deburring process. One such example will be presented in the next session.

#### 4.4 Experimental Results for CMRR

Experimental studies are conducted for an end milling process to verify the stability and performance of the proposed PI control and adaptive control algorithm. The robot used in the milling process is the ABB IRB 6400, the same robot on which we have done the parameter identification. The setup of robotic end milling process is shown as Fig. 6.

During the end milling experiment, a spindle was hold by the robot arm, and an aluminum block (AL2040) is fixed on a steel table. The cutting depth of the process was changed from 1 mm to 3 mm with a step of 1 mm. Both fixed gain PI control algorithm and self-tuning PI control algorithm, proposed, were tested with the same experimental setup. The control system performance and stability are compared for these two controllers. The experiment results for fixed-gain PI controller and for self-tuning PI controller are shown in Fig. 11 and Fig. 12, respectively.

The reference force was set at 250 N for the experiments. When the cutting depth is 1mm, both controllers are saturated with a full command speed at 30 mm/s. When the cutting depth changed to 2 mm, the fixed-gain PI controller started to vibrate, but still stable. When the cutting depth changed to 3 mm, the fixed-gain PI controller became unstable, just as predicted in the simulation results. On the other hand, the self-tuning adaptive controller maintained the stability and performance for all the cutting depths.

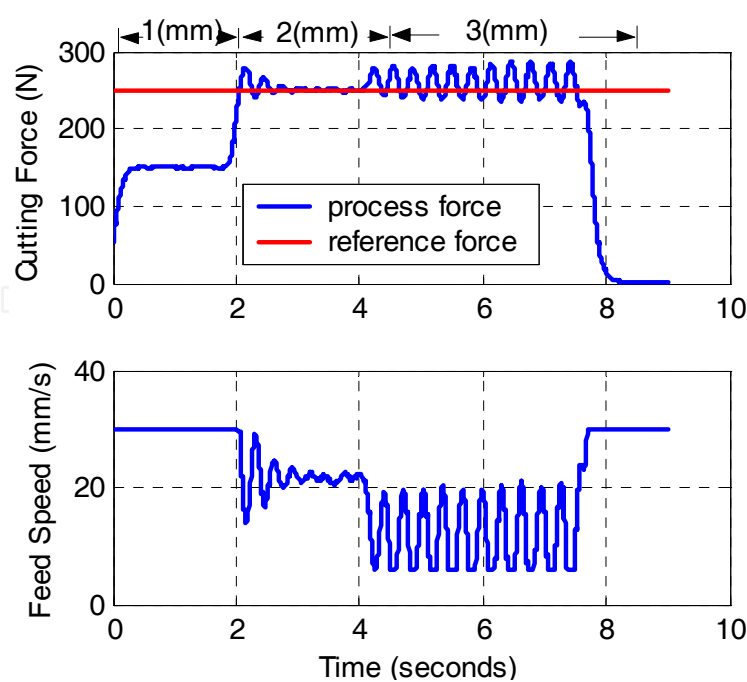


Fig. 11. Fixed-gain PI control experiment result

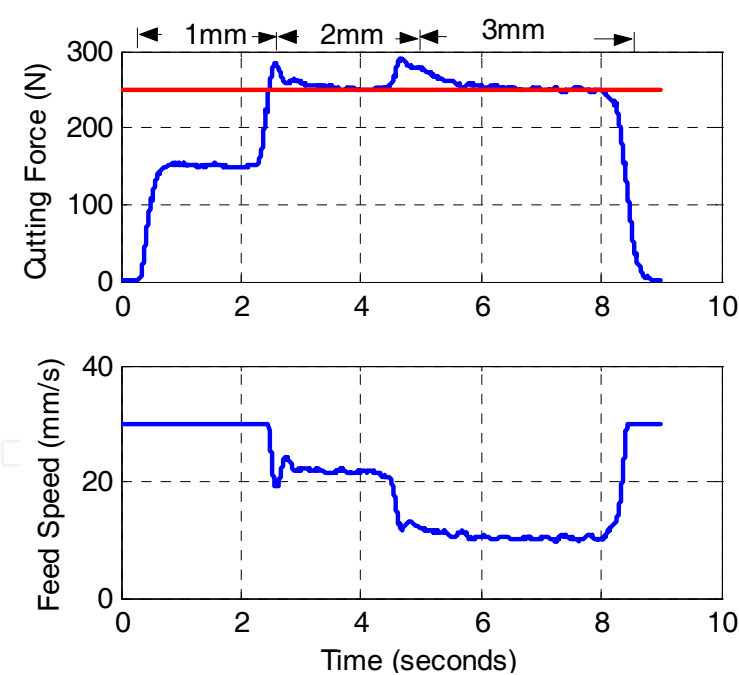


Fig. 12. Self-tuning PI control experiment result

The FLC is tested in another setup for robotic deburring with a grinder. There two burs located in the middle of the cast steel workpiece. A single cut with straight path is supposed to remove the burs. The limit of this system is the spindle power, which is equivalent to about 300 N. Without the CMRR function, the spindle will stall at the bur location and the entire system setup will be damaged. Since the bur location and size are not predicable,

normally the command feed speed is set to be a very conservative value, such as 30~40 mm/s. With FLC MRR control, command feed speed is set to 100ms. Two-stage switch control (0.5, 1.0) is sufficient to keep the system under spindle limit. The motor current signal (blue) is also recorded for comparison purpose. It could be shown that after a linear conversion (a gain and an offset), spindle current is equivalent to machining force signal. Either signal could be used for feedback here. Note that the force measurements in the experiments were filtered with a low-pass filter before used. (Fig. 13)

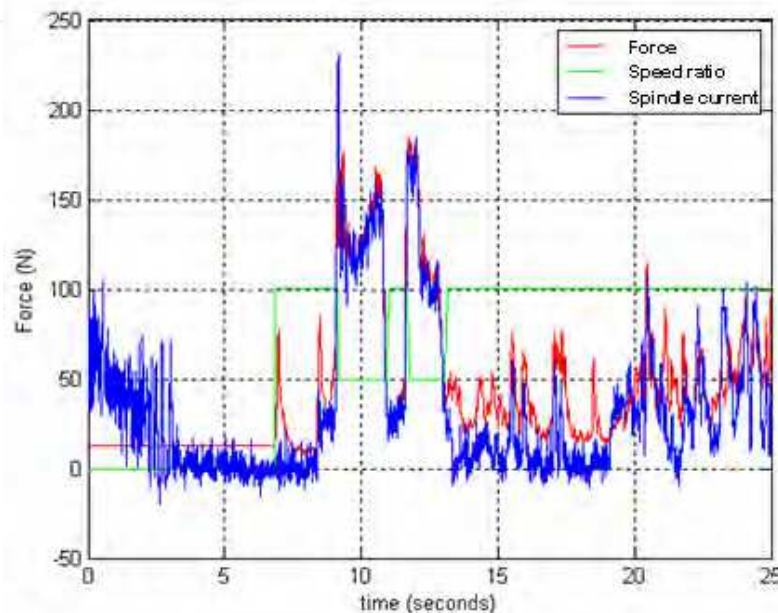


Fig. 13. FLC MRR control result

## 5. Robot Deformation Compensation

Among the many sources of errors of machine tools, thermal deformation and geometric errors are traditionally known as key contributors. For example, by studying a large amount of data, (Bryan, 1990) reported that thermal errors could contribute as much as 70% of workpiece errors in precision machining. RTEC techniques for geometric and thermal errors have successfully improved machine tool accuracy up to one order of magnitude (Donmez, 1986) (Chen, 1993).

After the geometric and thermal errors are compensated for, cutting force induced errors become the major source of machine tool errors. (Bajpai, 1972) and (Kops, et al., 1994) attempted to overcome the errors due to deflection using the relationship between workpiece deflection and the depth-of-cut applied at the final pass. However, most of the current error compensation research has not considered the cutting force induced errors. The following argument has been used to justify the neglect of the cutting force induced errors: in finish machining, the cutting force is small and the resulting deflection can be neglected.

However, in robotic machining process, due to the low stiffness of the industrial robot, the force induced deformation of the robot structure is the single most dominant source of workpiece surface error. Offline calibration strategies are often used to improve accuracy while sacrificing operation cycle time. The workpiece is calibrated with a distance sensor,

usually LVDT or laser sensor before and after the machining process. The surface error is measured and calculated to update the tool/workpiece data of the next cut. Although offline calibration could improve robot path error as well as force induced error, the process cycle time is increased, mostly doubled. With force sensor attached on the robot wrist, force information is ready on real time. If an accurate stiffness model could be established, the force induced error could be compensated online by updating the robot targets.

### 5.1 Robot Stiffness Modeling

A robot stiffness model, which relates the force applied on the robot tool end point to the deformation of the tool end point in Cartesian space, is crucial for robot deformation compensation, since the force measurement and control is fulfilled in Cartesian space while the robot position control is implemented in joint space.

The proposed model must be accurate enough for a great improvement of the surface error, as well as simple enough for real-time implementation. Detailed modelling of all the mechanical components and connections will bring a too complicated model for real-time control; and difficulties for accurate parameter identification.

The sources of the stiffness of a typical robot manipulator are the compliance of its joints, actuators and other transmission elements, geometric and material properties of the links, base, and the active stiffness provided by its position control system (Alici & Shirinzadeh, 2005).. As commercial robotic systems are designed to achieve high positioning accuracy, elastic properties of the arms are insignificant. The dominant influence on a large deflection of the manipulator tip position is joint compliance, e.g., due to reducer elasticity (Pan et al., 2006).

The conventional formulation for the mapping of stiffness matrices between the joint and Cartesian spaces, was first derived by (Salisbury, 1980) and generally has been accepted and applied.

$$K_x = J(Q)^{-T} K_q J(Q)^{-1} \quad (16)$$

Where  $K_q$  is a 6×6 diagonal joint stiffness matrix, which relates the motor torque load  $\tau$  on six joints to the 6×1 joint deformation vector  $\Delta Q$ ,

$$\tau = K_q \cdot \Delta Q \quad (17)$$

$J(Q)$  is the Jacobian matrix of the robot;

$K_x$  is a 6×6 Cartesian stiffness matrix, which relates the 6 D.O.F. force vector in Cartesian space  $F$  to the 6 D.O.F. deformation of robot in Cartesian space  $\Delta X$

$$F = K_x \cdot \Delta X \quad (18)$$

Eq. (16) can be derived from the definition of Jacobian matrix in Eq. (19) and the principle of virtual work in Eq. (20).

$$\Delta X = J(Q) \cdot \Delta Q \quad (19)$$

$$F^T \cdot \Delta X = \tau^T \cdot \Delta Q \quad (20)$$

For articulated robot,  $K_x$  is not a diagonal matrix and it is configuration dependent. This means: first, the force and deformation in Cartesian space is coupled, the force applied in one direction will cause the deformation in all directions; second, at different positions, the stiffness matrix will take different values.

(Chen & Kao, 2000) introduced a more complex model using a new conservative congruence transformation as the generalized relationship between the joint and Cartesian stiffness matrices in order to preserve the fundamental properties of the stiffness matrices.

$$K_x = J(Q)^{-T} (K_q - K_g) J(Q)^{-1} \quad (21)$$

With

$$K_g = \left[ \frac{\partial J^T(Q)}{\partial Q} \cdot F \right] \quad (22)$$

where  $K_g$  is a 6×6 matrix defining the changes in geometry via the differential Jacobian;  $F$  is external applied force.

The second model is more difficult to implement as the differential Jacobian is not available in the robot controller. The difference between these two models is the additional  $K_g$  in the second model.  $K_g$  accounts for the change in geometry under the presence of external load. IRB6400, a typical large sized industrial robot has a payload of 150kg, which will cause about 3 mm deformation considering its stiffness is around 0.5N/μm. From our calculation,  $K_g$  is negligible compared to  $K_q$  as this is a relative small deformation compared to the scale of robot structure.

Thus, the conventional formulation is selected in this research for stiffness modelling. In this model, robot stiffness is simplified to six rotational stiffness coefficients, that is, equivalent torsional spring with stiffness  $K$  as each joint is actuated directly with AC motor. Also from the control point of view, this model is the easiest to implement, since these are the 6 degree of freedom of the robot, which could be directly compensated by joint angles. Since the axis of force sensor is coincide with the axis of joint six, the stiffness of force sensor and its connection flange could be modelled into joint six.

## 5.2 Parameter Identification of the Stiffness Model

Experimental identification of the robot stiffness model parameters, joint stiffness of six joints, is critical in fulfilling real-time position compensation. In our model, the joint stiffness is an overall effect contributed by motor, joint link, and gear reduction units. It is not realistic and accurate to identify the stiffness parameter of each joint directly by disassembling the robot as the assembly process will affect the stiffness of the robot arm. The practical method is to measure it in Cartesian space.

The setup of robot stiffness measurement is shown in Fig. 14. The cutting tool at the end-effector is replaced by a sphere-tip. When robot is driven to a fixed position in the workspace, the joint angles of the robot are recorded. A weight is applied on the tool tip to generate a deformation. The position of the sphere-tip is measured by ROMOR CMM machine before and after the weight is applied to and the 3-DOF translational deformation is calculated. The applied force is measured by 6 DOF ATI force/torque sensor. A pulley is used to generate force on other directions than vertical down direction.



Fig. 14. Methodology of robot stiffness measurement

Given the kinematic parameters of the robot, the Jacobian matrix at any robot position could be calculated using robotics toolbox for MATLAB. Table 1 shows the IRB6400 kinematic model in Denavit-Hartenberg parameters.

The same procedure is repeated at multiple positions in the robot workspace and with different loads. From the relationship of

$$F = J(Q)^{-T} K_q J(Q)^{-1} \cdot \Delta X \quad (23)$$

$K_q$  could be solved by least square method, given  $F$ ,  $J(Q)$  and  $\Delta X$ . Only the first three equations from Eq. (23) are used in calculation as the orientation and torque are hard to measure accurately in the setup. The calibration results show that the standard deviation of the stiffness data is small, which means constant model parameter is adequate to model the deformation of robot. The deviation in the entire work space is less than 0.04mm.

Axis	$\theta$	d	a	$\alpha$	Home
1	$q_1$	900	188	$-90^\circ$	0
2	$q_2$	0	950	0	$-90^\circ$
3	$q_3$	0	225	$-90^\circ$	0
4	$q_4$	1300	0	$-90^\circ$	$180^\circ$
5	$q_5$	0	0	$90^\circ$	0
6	$q_6$	200	0	0	0

Table 1. DH model of IRB 6400

5.3 Real-time robot deformation compensation

The major sources of position error in robotic machining process can be classified into two classes, 1) machining force oriented error, and 2) motion error (kinematic, measurement and servo errors, etc.). The motion error is inherent from robot position controller and will appear even in non-contact movement. While the machining force in the milling process will typically over several hundreds of Newton, the force oriented error, which will easily go up to 0.5mm, is the dominant factor of surface error. Our objective here is to measure the deformation through a viable way and compensate it online to improve the overall machining accuracy.

To our best knowledge, none of the existing research has addressed the topic of online compensation of process force oriented robot deformation due to the lack of real-time force information and limited access to the controller of industrial robot.

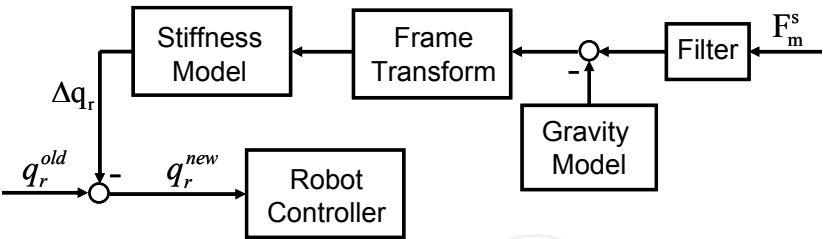


Fig. 15. Block diagram of real-time deformation compensation

The block diagram of real time deformation compensation algorithm is shown in Fig. 15. After the force sensor noise is filtrated, gravity compensation must be conducted to remove the force reading from the weight of spindle and tool. Since the robot may not always maintain a wrist down position, a general gravity compensation algorithm is developed to remove the gravity effects for any robot configuration. The algorithm takes measurement of gravity force at 15 distinctive robot configurations and uses least square method to calculate the mass and center of mass coordinates. This information is then updated to the robot tool data and the robot will always offset the gravity from the force reading at any robot configurations.

The force signal read from the sensor frame is then translated into the robot tool frame. Based on the stiffness model identified before, the deformation due to machining force is calculated online and the joint reference for robot controller is updated accordingly.

5.4 Experimental Results

The experimental tests on both standard aluminum block and real cylinder head workpiece have been conducted to verify the results of proposed real-time deformation compensation method.

5.4.1 Aluminum block end milling test

A 150mm×50mm 6063 aluminum alloy block is used for end milling test. Table 2 lists the detailed parameters for the experiment.

Test	End milling
Spindle	SETCO,5HP, 8000RPM
Tool type	SECO $\Phi$ 75mm, Square insert×6
Cutting fluid	- (Dry cutting)
Feed rate	20 mm/s
Spindle speed	3600 RPM
DOC	3 mm

Table 2. Parameters for end milling



Fig. 16. Setup of aluminum end milling and surface scan

A laser distance sensor is used to measure the finished surface of aluminum block as shown in Fig. 16. The surface error without deformation compensation demonstrates anti-intuitive results, on average extra 0.4mm material was removed from the aluminum block, which is not possible for a CNC machine since the cutting force normal to the workpiece surface will always push the cutter away from the surface and cause negative surface error (cut less). The coupling of robot stiffness model explains this phenomenon. When end milling using square inserts, the machining force in the robot feed direction and the cutting direction (around 300N each) are much larger than the force in the normal direction (around 50N). At this specific robot configuration, the force in feed and cutting direction will both push the cutter into the workpiece, which results in positive surface error (cut more). Since the feed force and cutting force are the major components in this setup, the overall effect is that the surface is removed 0.4 mm more than commanded depth. On the other hand, the result after

deformation compensation shows a less than 0.1 mm surface error, which is in the range of robot path accuracy.

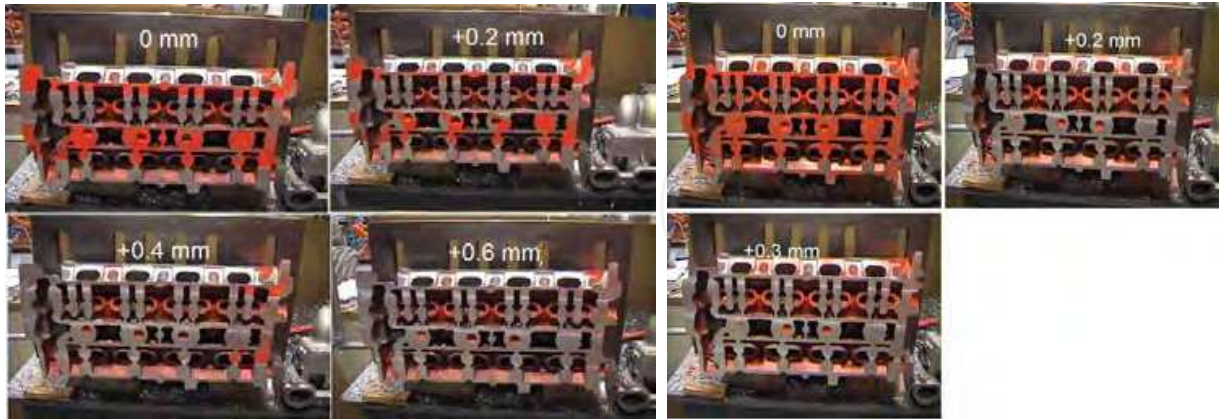


Fig. 17. A. Cylinder head part, surface error of end milling in position control; B. Cylinder head part, surface error of end milling in force control

#### 5.4.2 Cylinder Head End Milling Test

A real cylinder head workpiece is also utilized here for deformation compensation test, using the same end milling parameters as listed in Table 2. To better visualize the surface error, the surface is covered by orange paint after the end milling. Then the tool is moved 0.1mm closer to the workpiece surface each time, until all the paint on the surface are cleaned. As shown in Fig.17A, under position control, the tool touches the surface at -0.3mm, and clean the surface at 0.6mm, the total surface error is 0.9mm. Under the force control, the tool touches the surface at -0.1mm, and clean the surface at 0.3mm, the total surface error reduced to 0.4mm, as shown in Fig. 17B.

## 6. Conclusion

This chapter has addressed the critical issues in robotic machining process from programming to process control. Three major contributions, including rapid robot programming, controlled material removal rate, and online deformation compensation have been introduced in detail. The complete solution is achieved with force control strategy based on ABB IRC5 robot controller.

Rapid robot programming is characterized by two main modules: lead-through and automatic path-learning. Lead-through gives robot operator the freedom to adjust the spatial relationship between the robot tool fixture and the workpiece easily, while robot automatically follow the workpiece contour, record the targets and generate the process program in path-learning. Since the robot programming is generated at actual process setup, no additional calibration is required.

Online deformation compensation is realized based on a robot structure model. Since force induced deformation is the major source of inaccuracy in robotic machining process, the surface quality is improved greatly adopting the proposed method. This function is especially important in milling applications, where cutting force could be as large as 1000 N.

Regulating machining forces provides significant economic benefits by increasing operation productivity and improving part quality. CMRR control the machining force by realtime adjusting the robot feed speed. Various control strategy, including PID, adaptive control and fuzzy logic controller were implemented on different cutting situations Including the chatter and vibration analysis presented in (Pan & Zhang, et al, 2006), these complete set of solutions will greatly benefit the foundry industry with small to medium batch sizes. Dramatic increase of successful setups of industrial robots in foundry cleaning and pre-machining applications will be seen in the very near future.

## 7. References

- Alici, G.; Shirinzadeh, B. (2005). "Enhanced Stiffness Modeling Identification and Characterization for Robot Manipulators", *IEEE Transactions on Robotics*, Vol. 21, No. 4, August 2005.
- Donmez, M. A. et al. (1986). "A General Methodology for Machine Tool Accuracy Enhancement y Error Compensation", *Precision Engineering*, Vol. 8, No. 4, pp. 187-196.
- Bajpai, S. (1972) "Optimization of Workpiece Size for Turning Accurate Cylindrical Parts", *International Journal of Machine Tool Design and Research*, Vol. 12, pp. 221-228.
- Basanez, L. & Rosell, J. (2005) "Robotic Polishing Systems—From graphical task specification to automatic programming", *IEEE Robotics & Automation magazine*, Sep, 2005.
- Bryan, J. B. (1990). "International Status of Thermal Error Research", *Annals of the CIRP*, Vol. 39, No. 2, pp. 645-656.
- Budak, E.; Altintas, Y. (1998) "Analytical Prediction of Chatter Stability Conditions for Multi-Degree of Systems in Milling. Part I: Modeling, Part II: Applications," *Transactions of ASME, Journal of Dynamic Systems, Measurement and Control*, vol.120, pp.22-36
- Chen, J. S. et al. (1993). "Real Time Compensation of Time-variant Volumetric Error on a Machining Center", *ASME Journal of Engineering for Industry*, Vol. 114, pp. 472-479.
- Chen, S.F.; Kao, I. (2000) "Conservative congruence transformation for joint and Cartesian stiffness matrices of robotic hands and fingers", *The International Journal of Robotics Research*, Vol. 19, No. 9, September 2000, pp. 835-847.
- Daneshmend, L., & Pak, H., (1986) "Model Reference Adaptive Control of Feed Force in Turning," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 108, No. 3, pp. 215-222
- Galichet, S., & Foulloy, L. (1995) "Fuzzy controllers: synthesis and equivalences", *IEEE Transactions on System, Man and Cybernetics*.
- Kim, S., Landers, R., Ulsoy, A., 2003, "Robust Machining Force Control with Process Compensation," *Journal of Manufacturing science and engineering*, Vol 125, pp. 423-430
- Kops, L.; Gould, M.; & Mizrach, M. (1994). "A Search for Equilibrium between Workpiece Deflection and Depth of Cut: Key to Predictive Compensation for Deflection in Turning", *Proceedings 1994 ASME Winter Annual Meeting*, Vol. 68-2, pp. 819-825.
- Landers, R. & Ulsoy, A., (2000) "Model-based machining force control", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 122, no. 3, 2000, pp. 521-527.

- Li, H., & Gatland, H., (1996) "Conventional fuzzy control and its enhancement", IEEE Transactions on System, Man and Cybernetics.
- Pan, Z. (2006). "Intelligent robotic machining with force control", Ph.D. dissertation, Stevens Institute of Technology, NJ, USA, Jan 2006.
- Pan, Z.; & Zhang, H.; et al. (2006). "Chatter analysis of robotic machining process", Journal of Material Processing Technology, Volume 173, Issue 3, Pages 301-309, April 2006.
- Pettersen, T. et al. (2003) "Augmented Reality for Programming Industrial Robots", Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03), 2003.
- Pires, J. et al. (2004) "CAD interface for automatic robot welding programming", Industrial Robot: An International Journal, Vol 31, pp.71-76, 2004.
- Salisbury, K. (1980) "Active stiffness control of a manipulator in Cartesian coordinates", Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 87-97.
- Schmid, D. (1990) "Sensor Simulate Tools", The industrial robot, pp. 97-9, June, 1990
- Sugita, S. et al. (2003) "Development of robot teaching support devices to automate deburring and finishing works in casting", The International Journal of Advanced Manufacturing Technology, Springer-Verlag London, Dec 2003.
- Wang, J. Zhang, H. and Pan, Z. (2007) "Machining with Flexible Manipulators: Critical Issues and Solutions", Industrial Robotics: Programming, Simulation and Applications, ISBN 3-86611-286-6 Edited by Low Kin Huat, Feb 2007.
- Yang, S., (1996) "Real-time compensation for geometric, thermal, and cutting force induced errors in machine tools," Ph.D. dissertation, The University of Michigan

IntechOpen



## **Robot Manipulators New Achievements**

Edited by Aleksandar Lazinica and Hiroyuki Kawai

ISBN 978-953-307-090-2

Hard cover, 718 pages

**Publisher** InTech

**Published online** 01, April, 2010

**Published in print edition** April, 2010

Robot manipulators are developing more in the direction of industrial robots than of human workers. Recently, the applications of robot manipulators are spreading their focus, for example Da Vinci as a medical robot, ASIMO as a humanoid robot and so on. There are many research topics within the field of robot manipulators, e.g. motion planning, cooperation with a human, and fusion with external sensors like vision, haptic and force, etc. Moreover, these include both technical problems in the industry and theoretical problems in the academic fields. This book is a collection of papers presenting the latest research issues from around the world.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Zengxi Pan and Hui Zhang (2010). Robotic Machining from Programming to Process Control, Robot Manipulators New Achievements, Aleksandar Lazinica and Hiroyuki Kawai (Ed.), ISBN: 978-953-307-090-2, InTech, Available from: <http://www.intechopen.com/books/robot-manipulators-new-achievements/robotic-machining-from-programming-to-process-control>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen