1-1-2008

Robotic machining from programming to process control: a complete solution by force control

Zengxi Pan

University of Wollongong, zengxi@uow.edu.au

Hui Zhang

ABB Corporate Research, Shanghai, China

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>

Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Pan, Zengxi and Zhang, Hui: Robotic machining from programming to process control: a complete solution by force control 2008, 400-409.
<https://ro.uow.edu.au/infopapers/1585>

Robotic machining from programming to process control: a complete solution by force control

Abstract

This paper aims to present the critical issues and methodologies to improve robotic machining performance with flexible industrial robots. Design/methodology/approach – A complete solution using active force control is introduced to address various issues during the robotic machining process.

Programming complex couture parts without a CAD model is made easy by using force control functions such as lead-through and path-learning. The problem of process control is treated with a novel methodology that consists of stiffness modeling, real-time deformation compensation for quality and controlled material removal rate for process efficiency.

Experimental results showed that higher productivity as well as better surface quality can be achieved, indicating a promising and practical use of industrial robots for machining applications that is not available at present.

Keywords

Robotic, machining, from, programming, process, control, complete, solution, force, control

Disciplines

Physical Sciences and Mathematics

Publication Details

Pan, Z. & Zhang, H. 2008, "Robotic machining from programming to process control: a complete solution by force control", *Industrial Robot*, vol. 35, no. 5, pp. 400-409.

Robotic Machining From Programming to Process Control— A Complete Solution by Force Control

Zengxi Pan, Hui Zhang

Abstract

This paper presents a critical issues and methodologies to improve robotic machining performance with flexible industrial robots. A complete solution using active force control is introduced to address various issues during the robotic machining process. Programming complex contour parts without a CAD model is made easy by using force control functions such as lead-through and path-learning. The problem of process control is treated with a novel methodology that consists of stiffness modeling, real-time deformation compensation for quality and controlled material removal rate (CMRR) for process efficiency. Experimental results show that higher productivity as well as better surface quality can be achieved, indicating a promising and practical use of industrial robots for machining applications that is not available at present.

Keywords: force control, PbG, CMRR, deformation compensation

1. Introduction

Cleaning and pre-machining operations are major activities and represent a high cost burden for casting producers. The cleaning operations account for 20-40% of the overall casting manufacturing cost. Subsequent machining operations may lead to expenditure equivalent to a further 40% of the casting production costs. Machining processes, such as cleaning, milling, grinding, deburring, and saw cutting are promising applications for industrial robot with the drive from foundry automation. From a robotic machining point of view, two types of machining processes could be distinguished. The first type, typically cleaning and deburring, usually has a very complex 3D curved cutting path, a crucial cycle time requirement, and relative low surface accuracy requirement. Today, most of the deburring operations are done manually in an extremely noisy, dusty and unhealthy environment. Therefore, automation for these operations is highly desirable. The second type is milling process, in which robot moves in a simpler path with lower feed speed (20~30 mm/s), while heavily engaging with the workpiece. The controller must be accurate enough to maintain the surface quality under large and varied machining forces. This type of machining is currently conducted by CNC machines, which can be justified economically only for large batch sizes.

This research proposes a robotic machining strategy for the foundry industry with small to medium batch sizes. The strategy is a complete solution addressing the difficulties for both types of machining applications from programming to process control. Based on an active force control platform, different control strategies are implemented including lead-through, path-learning, CMRR, and deformation compensation to facilitate various process requirements. Experimental results showed that higher productivity as well as better surface finish can be achieved, indicating a promising and practical use of industrial robots for machining applications that is not possible at present.

This paper is organized in six sections. Following this introduction section, section two describes several major challenges for robotic machining process. Section three provides the introduction of an active force control platform, which forms the foundation for various control strategies. Section four addresses the programming issues for a complex contoured part, which is made easy and efficient through two force control strategies, lead-through and path-learning, robotic programming. Section five presents two realtime process control techniques. The real-time deformation compensation improves the accuracy of the robotic machining operation, while the controlled material removal rate greatly reduces the process cycle time. Experimental results are presented at the end of section four and section five. A summary is provided in Section six.

2. Challenges

Robotics based flexible automation is considered as an ideal solution for its programmability, adaptivity, flexibility and relatively low cost, especially for the fact industrial robots are 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 several major difficulties associated with robotic machining processes utilizing a conventional industrial robot [1].

The first difficulty is the generation of robot motions for a complex workpiece. Traditionally, online programming methods have conventionally been carried out by skilled workers guiding the robot through the desired path using a teach pendant, namely the jog-and-teach method. Although the concept is simple, it is not feasible for many machining processes especially for deburring process, which has a great number of teaching points and requires high positioning accuracy. In such cases, an operator must constantly guide the robot through motions accurately which is usually a very time-consuming task. Sometimes robots take longer to program and set-up than just completing the machining task. In addition, the quality of the program is limited by the skills of the operator and once the program is generated, it is almost impossible to make further amendments. Offline programming method, which extracts the robot targets from CAD data of a workpiece, is another choice [2, 3]. Although off-line programming is more accurate and flexible, it is only cost-effective for large batch sizes. Since it relies heavily on the modeling of the robot and workpiece, additional calibration procedures are usually inevitable to meet the process accuracy requirement.

Today, both on-line and off-line programming methods are still too expensive, time-consuming and difficult for companies producing parts in medium to small batch sizes. The complexity of programming becomes one of the major hurdles preventing automation of machining processes using industrial robots. Thus, efficient techniques for automatic robot programming must be applied. We will address this issue by presenting a programming by guiding (PbG) method which could minimize this burden.

The second difficulty is the deformation caused by the interaction force between tool and workpiece, especially for milling process which generates large cutting forces [4]. The stiffness for a typical articulated robot is usually less than $1 \text{ N}/\mu\text{m}$, while a standard CNC machine very often has stiffness greater than $50 \text{ N}/\mu\text{m}$. As a result, force induced deformation becomes a major source of the inaccuracy of finished surface. A perfect robot program without considering contact and deformation will immediately become flawed as the robot starts to execute the machining task. Unlike multi-axis CNC machine centers, such deformations are coupled and varied even subjected to the same force at different workspace locations. Such coupling results in deformations not only in the direction of reaction force and can generate some counter-intuitive results.

Thirdly, the lower stiffness also presents a unique disadvantage for the machining of casting parts with complex geometry having non-uniform cutting depth and width. As a result, the machining force will vary dramatically, which induces uneven robot deformations. What this means in one example is that the flatness of the machined plane is so inferior that it renders the robotic process unable to meet the minimum requirement. In general practice, machine tools maximize the material removal rate (MRR) during roughing cycles by applying all of the available spindle power to the machining process. However when machines use carbide tools for roughing operations, the available spindle power is the limiting factor on the MRR. In conventional robot programming and process planning practice, the cutting feed rate is constant even with significant variation of cutting force from part to part, which dictates a conservative cutting feed rate without violating the operational limits. Therefore, it is desirable to maximize MRR and minimize cycle time by optimizing the machining feed speed based on a programmed spindle load. By optimizing the feed speed in real time, one could compensate for conservative assumptions and process variations to help reduce the cycle time. Every part, including the first, is optimized automatically, eliminating the need for manual part program optimization.

The fourth difficulty is chatter/vibration occurring during the machining process [5, 6, 7]. Chatter/vibration becomes a more serious problem in robotic machining process due to the low stiffness and coupled

structure of industrial robots. Robotic engineers and technicians are often frustrated in dealing with elusive and detrimental chatter issues without a good understanding or even a rule of thumb guideline of the problem. Very often, to get their process working correctly, one has to spend a tremendous amount of time on trial and error for the sheer luck of stumbling a golden setup or has to sacrifice the productivity by settling on conservative cutting parameters much lower than the possible machining capability.

Most of the existing literature on machining process, such as process force modeling [8], accuracy improvement [9] and vibration suppression [10] are mostly based on the CNC machine. Research in the field of robotic machining is still focused on accurate off-line programming and calibration. In the literature, a number of references can be found concerning one or two of the challenges mentioned above. However, system coping with all the above challenges must be available in order to enable a large-scale penetration of robots into the area of machining processes. This paper presents the functional structure of such a system.

As the chatter analysis is discussed in a separate paper [7], our focus in this paper is to address the first three major issues confronting robotic machining processes: 1) To generate robot programs for complex 3D curvatures easily without experienced technician and CAD model; 2) To improve the machining quality with the low stiffness, and a low accuracy robot; 3) To improve the robotic machining efficiency by providing real time optimization to maximize material removal rate.

3. Force Control Platform

The active force control platform forms the foundation of strategies developed to address various difficulties mentioned. It is implemented on the most recent ABB IRC5 industrial robot controller [11], which is a general controller for a series of ABB robots. The IRC5 controller includes a flexible teach pedant 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 to realize implicit hybrid position/force control scheme. The system setup for robotic machining with force control is shown in Figure 1. The flexible force controller could be configured differently to satisfy various application needs. The block diagram of the force control loop is shown in Figure 2.

Figure 1 System Setup for Robotic Machining with Force Control

The force control is implemented as an implicit hybrid position/force control [12, 13]. In order to obtain the accurate external contact force, a gravity compensation algorithm is necessary to compensate the payload – tool and gripper on the robot wrist from the measured force. By running a single system routine, the mass and centre of mass can be calculated by a nonlinear least square routine. Then, the gravity of payload at any tool position/orientation could be estimated accurately and then offset from the measured force.

Figure 2 The Force Control Loop

Whilst the conventional position control is realized in joint space, force control is implemented in Cartesian space. The difference between the reference force and the measured contact force is the input to the force controller. If a certain direction is set to be under force control, the force deviation calculates a correction to the robot's nominal position and changes the reference position and speed given by a robot trajectory generator.

The force controller provides two major functions to make the entire programming process collision free and automatic. The first function is “lead-through”, in which the robot is compliant in selected directions (force control directions) and stiff in the rest of directions (position control directions). In order to change the position or orientation of the robot, the robot operator could simply push or drag the robot by hand. The second function is called “path-learning”, in which the robot is compliant in normal to path direction to make the tool constantly contact with work piece. 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, the robot is still under position control, that is, stiff in all directions. Deformation compensation is achieved by update the target position in the position loop based on the measured process force and robot stiffness model, while robot feed speed is adjusted to maintain constant cutting tool spindle power consumption for CMRR. These two strategies are complementary to each other since CMRR adjusts robot speed at feed direction only and deformation compensation adjusts the reference target at the rest of directions. The detailed control strategies will be explained in section 5.

4. Easy Robot Programming

Programming by Guiding (PbG) aims at solving the persistent problem of programming robotic applications. To be a successful strategy, it must satisfy the requirements for potential robot operators, who usually have the knowledge about the machining process and know the basic robot operations, such as jogging a robot and writing simple robot programs.

The proposed programming method should not require such operators to learn further new skills such as programming with high level computer language like C++ or Java. The method should allow the operator to program robot motions with minimum efforts at the actual setup so that the process requirement, motion path are presented together and the operator can provide in-situ and timely judgment for any particulars in programming the robot. Further, it is desirable to free the operator from having to guide the robot in three dimensions for all the program points in a complex path during programming operations in an effort to reduce the complexity of programming desired motions of the robot.

To facilitate the programming process, an artificially tangible tool (dummy tool) with the same dimensions as the real process tool is usually desirable. For example, in the deburring process with an end milling tool, moving the tool with sharp cutting edge along the workpiece surface can create undesirable friction and damage to the part's surface. However, a cylindrical shape with the same dimension would eliminate the problem and greatly enhance the programming experience.

4.1 Lead-through

Lead-through is the only step that requires human intervention throughout the entire PbG process. The purpose of lead-through is to generate a few number of gross guiding points and prepare for the next step. These guiding points will be used to calculate path frame in path-learning as shown in Figure 3. The position accuracy of these guiding points is not critical because they are not the actual points/targets in the final program and they will be updated by the automatic path-learning. However the orientation of these points should be carefully taught since they will determine the path frame and will be kept in the final program.

Figure 3 Lead-through and Path Learning

Theoretically 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 that this is almost impossible to adjust the tool orientation accurately by pushing/pulling with a single hand. Thus, a force control jogging mode is created, under which the operator can push/pull the robot tool to any position easily and change the robot tool orientation using the joystick on the teach pendant. 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 can make very accurate adjustment on each independent rotational axis.

The safety issue is always a concern in automation process. In the lead-through scenario, although the operator is very close to the robot or even in touch with the robot, they are in a safe environment because they are the only person within the robot workspace and has complete control of the robot through the teach pendant which has an emergency stop button and hold-to-run mechanism. If any thing goes wrong in the robot controller, the operator can stop the robot by releasing the hold-to-run mechanism or hit the emergency stop button.

4.2 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 , which is perpendicular to path direction X_p , (Figure 3) robot motion is under force

control, whilst all other directions and orientations the robot is 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 in the actual workpiece contour, the tool tip will yield along the Y axis until it reaches the equilibrium of 20N, resulting in a new point that 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.

Since this method uses the path direction of gross guiding points to approximate the actual normal to workpiece contour direction, it is valid only when the normal direction does not change too much between the two neighbouring guiding points. As a result, more guiding points need to be taught at sharp corner to limit the approximation error while fewer points are required at the place with small curvature.

Whilst 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 in a recorded spatial relationship that is an exact replicate between the tool fixture and the workpiece. A generated robot program based on recorded path can be directly used to carry out the actual process. Whilst the robot is executing the actual process, the robot controller is not required to engage in any force control behaviour, unless such control would benefit the process in one way or the other.

4.3 Post Processing

After automatic path-learning, the position data logged by the robot controller will be filtered and reduced to generate a robot program. The measurements around sharp corners are often very noisy due to the high dynamic forces. The maximum and minimum acceptable contact force is set up as a threshold to remove this type of noise. The number of the targets from automatic path-learning is disproportionately large since the robot controller records position data as fast as every 4 ms. One approach, namely deviation height method, is used to reduce the redundant points and approximate the contour by straight-line segments. The deviation height limit determined by process requirement is set as the error bound for the reduced robot path. As shown in Figure 4, a point will be remained in the path only if there is a certain intermediate point exceeding the deviation height limit. All the intermediate points will be removed from the path. This approach can reduce the length of the point data to 5~10% of the original. A robot program is then generated from the reduced data.

Figure 4 Deviation Height Method

4.4 Complete PbG Procedure

For most applications, such as a deburring process with a milling tool, it is usually required that the tool to maintain certain angle with the work surface. This requirement is achieved through the teaching of guiding points where the operator can determine the required angle at each target taught. This is where the lead-through programming method is desirable for intuitiveness and convenience. Whilst the robot is in

compliant continuous motion, this desired angle is preserved as the orientation of the robot is still in closed-loop position control. In the case that the desirable angle between two guiding points is different, the industrial robot controller will provide an interpolated orientation between the two points, which is still preserved in step of compliant path learning for the same argument, resulting a continuous and smooth transition between the two distinctive point.

In a real application, for safety reasons it is always beneficial to verify the program before starting the spindle and conduct the cutting. For this purpose, besides the final cutting program, PbG generates a test program with a small offset away from the work piece. Without turning on the spindle, the robot operator can run this test program to verify the correctness of the final program with lower feed speed using a dummy tool and master piece. Ideally, the dummy tool will follow the contour of the workpiece with a consistent offset.

The complete programming procedures are summarized as follows:

- 1) Teaching a small number of selected gross guiding points between the robot fixture and the workpiece, where the guiding points can be obtained by lead through teaching.
- 2) Generate a robot program based on the points taught.
- 3) Executing continuous movement with the robot based on the taught program whilst engaging the robot in compliant force control in the direction appropriate in the particular process, usually normal to the path, so that the tool fixture is always in contact with the workpiece and follows the desired contour of the workpiece.
- 4) Continuously recording a spatial relationship of the robot fixture relative to the workpiece during movement of the robot which is under compliant force control.
- 5) Post-processing of recorded points by filtering the noise and reducing the redundant points.
- 6) Generating the robot motion program based on the reduced points which is the exact desired program for the given setup. Both final cutting program and test program will be generated.
- 7) Verify the program.

4.5 Experimental Results

With force control integrated in IRC5 controller, PbG method is available for all ABB industrial manipulators. An automatic deburring system using an 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 Figure 5.

Figure 5 Experimental Setup for Robotic Programming

A 2 mm cutting tool, driven by ultra high speed ($\sim 18,000$ rpm) air spindle is used to achieve this task. Since the groove is only about 5 mm wide and has a contoured 2D shape, manually teaching a high quality

program to clean the complete groove is almost impossible even for very experienced robot operator. 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, whilst the velocity is 5mm/s. As shown in Figure 6, the curvature of recorded targets after path learning 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 corners, to ensure a smooth motion throughout the path. The point reduction technique is performed on the filtered data. A deviation height of 0.2mm reduced the thousands of points recorded by the robot controller every 40ms to about 300 points.

Figure 6 Results from Path-Learning

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

5. Process Control

5.1 Robot Stiffness Model

As was stated before, one of the focuses for this paper is to improve the robotic machining accuracy by reducing machining force induced deformations. While thermal induced error is the largest error component for CNC machining, motion error due to machining force contributes most of the total machining errors in robots. [14] For example, a 500N cutting force during a milling process will cause a 1 mm position error for a robot in comparison to an error of less than 0.01mm for a CNC machine. In order to achieve higher dimensional accuracy, the deformation due to the interactive force must be compensated.

Since force measurement and subsequent compensation is carried out in 3-D Cartesian space, a stiffness model, which relates the force applied at the robot tool tip to the deformation of the tool tip in Cartesian space, is crucial to realize deformation compensation. The model should be accurate enough for the prediction of robot structure deformation under arbitrary load conditions. At the same time, it needs to be simple enough for real time implementation. Detailed modeling of all the mechanical components and connections will render a model too complicated for real-time control, and difficult for accurate parameter identification.

Industrial robotic systems are designed to achieve high positioning accuracy and high strength. Elastic properties of the arms are insignificant. Therefore, the dominant contribution factor for a large deflection of the manipulator tip position is the joint compliance, e.g., due to gear transmission elasticity. Modeling of robot stiffness could be reduced to six rotational stiffness coefficients in the joint space. From the control

point of view, this model is also easy to implement, since all industrial robot controllers are decoupled to SISO joint control at the servo level. As a result, the joint deformation could be directly compensated on the joint angle references passed to the servo controller. Note here that the axis of the force sensor coincides with the axis of joint 6, the stiffness of the force sensor and its connection flange could be modeled into joint 6. Figure 7 shows the structure of a 6-DOF ABB IRB 6400 robot with black arrows representing the location of compliant joints.

Figure 7 Structure of 6-DOF ABB IRB 6400 manipulator

Next, we will derive the stiffness model in Cartesian space based on joint compliance parameters.

In joint space, the stiffness model could be represented as:

$$\boldsymbol{\tau} = \mathbf{K}_q \cdot \Delta \mathbf{Q} \quad (1)$$

Where: $\boldsymbol{\tau}$ is the torque load on the each joint; $\mathbf{K}_q$ is a 6×6 diagonal joint stiffness matrix; $\Delta \mathbf{Q}$ is the 6×1 joint deformation vector.

While in Cartesian space:

$$\mathbf{F} = \mathbf{K}_x \cdot \Delta \mathbf{X} \quad (2)$$

Where $\mathbf{F}$ is the 6 D.O.F. force/moment vector, $\Delta \mathbf{X}$ is the 6 D.O.F. deformation of the robot in Cartesian space, the first three components represent positions and last three components represent orientations, and $\mathbf{K}_x$ is a 6×6 stiffness matrix in Cartesian space.

From the definition of the Jacobian matrix, we have:

$$\Delta \mathbf{X} = \mathbf{J}(\mathbf{Q}) \cdot \Delta \mathbf{Q} \quad (3)$$

Where $\mathbf{J}(\mathbf{Q})$ is the Jacobian matrix of the robot.

At steady state, after compensating for the gravity force of the tool, the robot joint torques will exactly balance external forces applied on the tool tip. The principle of virtual work gives us:

$$\mathbf{F}^T \cdot \Delta \mathbf{X} = \boldsymbol{\tau}^T \cdot \Delta \mathbf{Q} \quad (4)$$

From (1), (3), (4), when the robot is not at singular position, we have:

$$\mathbf{K}_x = \mathbf{J}(\mathbf{Q})^{-T} \mathbf{K}_q \mathbf{J}(\mathbf{Q})^{-1} \quad (5)$$

For an articulated robot, $\mathbf{K}_x$ is not a diagonal matrix and it is configuration dependent. This means that: first, the force and deformation in Cartesian space is coupled, in other words, the force applied in one direction will cause the deformation in all possible directions; second, the stiffness is also a function of robot kinematics $\mathbf{J}(\mathbf{Q})$, it changes significantly in the entire workspace. However, even though at different

locations, the stiffness matrix will take different values (see Table 1 for one example), these changes can be sufficiently modeled by Eq. (5), with the assumption that $K_q(i,i)$, representing the stiffness of joint i , is a constant value. Thus, if K_q can be measured accurately, the deformation of robot TCP under external force at any location in the workspace could be estimated as,

$$\Delta X = J(Q)K_q^{-1}J(Q)^T \cdot F \quad (6)$$

Table 1 One example of the Cartesian Stiffness Matrix K_x from Eq. (5), (Since, $K_x=F/\Delta X$, Units are N/mm, N/rad, N·mm/mm, and N·mm/rad respectively)

2.86E+02	-8.78E+02	8.12E+02	1.49E+06	3.01E+05	3.97E+05
-8.78E+02	2.39E+02	5.48E+03	-1.85E+05	-5.46E+05	-4.09E+05
8.12E+02	5.48E+03	4.91E+02	-4.56E+06	4.31E+05	8.62E+05
1.49E+06	-1.85E+05	-4.56E+06	7.53E+07	5.85E+08	-7.62E+08
3.01E+05	-5.46E+05	4.31E+05	5.85E+08	1.68E+08	3.20E+08
3.97E+05	-4.09E+05	8.62E+05	-7.62E+08	3.20E+08	1.30E+08

Experimental determination of joint stiffness parameters is critical in fulfilling real-time position compensation. In this model, the joint stiffness is an overall effect contributed by motor, joint link, and gear reduction units. It is not realistic to identify the stiffness parameter of each joint directly by disassembling the robot; the practical method is to measure it in Cartesian space.

Figure 8 Experiment Setup of Robot Stiffness Measurement

To be able to measure small deformations in 3-D space, the end-effector is equipped with a sphere-tip tool shown in Figure 8. The tool tip is set to a fixed point in the workspace, and the manipulator joint values are recorded. A given load in the range of 100N~400N is applied to the tool, causing the sphere-tip to move away from the original point. The original and deformed positions are measured with ROMER, a portable CMM 3-D digitizer, and the 3-DOF translational deformations are calculated. From, Eq. (6), K_q could be solved by least square method.

The same procedure is repeated at several different locations in the robot workspace, the deviation of the results is small, which means a set of constant model parameters could model the robot deformation with a small error. (Figure 9)

Figure 9 Error of Stiffness Modeling

5.2 Robot Deformation Compensation

The major position error sources in robotic machining process can be classified into two categories, (1) Machining force induced error, and (2) motion error (kinematic and dynamic errors, etc.). The motion error, typically in the range of 0.1 mm, is inherent from the robot position controller and would appear even in non-contact cases. While the machining force in the milling process is typically over several hundreds of Newton, the force-induced error, which could easily go up to 1 mm, is the dominant factor of surface error. Our objective is to estimate and compensate the deformation in real time to improve the overall machining accuracy.

The existing research of robot deformation compensation is focused on gravity compensation, deflection compensation of large flexible manipulators, etc. Not much attention has been paid to the compensation of process force induced robot deformations due to the lack of understanding and model of robot structure stiffness, the lack of real time force information and limited access to the controller of industrial robot.

Figure 10 Principle of Real-time Deformation Compensation

The block diagram of real time deformation compensation is shown in Figure 10. After filtering the force sensor noise and compensating the gravity of the spindle and the cutter, the force signal was translated into the robot tool frame. Based on the stiffness model identified before, Eq. (6), the deformation due to the machining force is calculated in real time and the joint reference for the robot controller is updated accordingly.

5.3 Controlled Material Removal Rate

In pre-machining processes, maximum material removal rates are even more important than precision and surface finish for process efficiency. 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:

$$MRR = w \cdot d \cdot f \quad (7)$$

Conventionally, feed speed is kept constant in Fig. 4(a) of the variation of depth of cut and width of cut during foundry part pre-machining process. Since most foundry parts have irregular shapes and uneven depth of cut, this will introduce a dramatic change of MRR, which would result in a very conservative selection of machining parameters to avoid tool breakage and spindle stall. The concept of MRR control is to dynamically adjust the feed speed to keep MRR constant during the whole machining process. As a result, a much faster feed speed, instead of a conservative feed speed based on maximal depth of cut and width of cut position, could be adopted.

Since the value of MRR is difficult to measure, the MRR is controlled by regulating the cutting force, which is readily available in real-time from a 6-DOF strain gage force sensor fixed on the robot wrist. Placing the analysis of the material removal process on a quantitative basis, the characterization of cutting force is important for research and development into the modeling, optimization monitoring and control of metal cutting.

As the feed speed f is adjusted to regulate the machining force, the MRR could be controlled under a specific spindle power limit avoiding tool damage and spindle stall. Also, controlled MRR means predictable tool life, which is very important in manufacturing automation. Figure 2 shows the block scheme of machining force control with controlled material removal rate (CMRR).

$$F_c(s) = K \cdot w \cdot d \cdot f \frac{1}{\tau_m s + 1} \quad (8)$$

where τ_m is the machining process time constant. Since one spindle revolution is required to develop a full chip load, τ_m is 63% of the time required for a spindle revolution [16]. Since the force control is implicitly implemented, the control loop bandwidth is limited by a position servo control, which is around 10 Hz for an industrial robot. The force process gain may be seen as $\theta = K \cdot w \cdot d$, which is sensitive to the process

inputs. With the proper selection of reference feed speed f_r and reference force F_r , a PI controller is adopted to regulate the cutting force F_c , while force process gain θ changes.

In the previous sections, the robot deformation subject to an arbitrary process force loading is modeled and the model parameter is experimentally measured. With this model, the online deformation scheme is implemented on the robot controller. Secondly, the concept for controlled material removal rate is presented and implemented. In this section, the experimental results are presented to validate the aforementioned schemes.

Figure 12 shows the setup of a milling test. A spindle is fixed on the robot arm and the workpiece is fixed on a steel table. For illustration, a 6063 aluminum block is used for testing purpose.

Figure 12 Experimental Setup for Robotic Milling

Tests on an aluminum block with the depth of cut changed from 2 mm to 3 mm shows, when force control is activated, the cutting force is regulated in spite of the variance of depth of cut. Figure 13 shows the cutting force for both position control (top) and force control. Three curves represent the machining force at three directions. As you can see, in position control, machining force increased dramatically when depth of cut changed from 2 mm to 3 mm, while in force control, after a short transient time the machining force is regulated at the setup value 160N. The milling test of aluminum with variation of width of cut shows similar results.

Figure 13 Force Control Result of Variant Depth of Cut

As a result, the feed speed could always be setup as fast as the limit of spindle power. In a foundry milling or deburring process, the robot movement will not be constrained to a very conservative speed to avoid tool breakage or spindle stall. The cycle time decreased by CMMR is typically around 30% to 50% for different workpieces.

In the deformation compensation test for milling an aluminum block, a laser displacement sensor is used to measure the finished surface. The surface error without deformation compensation demonstrates counter-intuitive results; an extra 0.5mm was removed in the middle of the milling path. Conventional wisdom would indicate that a flexible machine would also cut less material due to deformation, since the normal force during cutting will always push the cutter away from the surface and cause negative surface error. However, in the articulated robot structure, the deformation is also determined by the structure Jacobian, in a lot of cases, a robot could end up cutting more material than programmed. The coupling of the robot stiffness model explains this phenomenon, the force in feed direction and cutting direction will result in positive surface errors in that robot configuration. Since the feed force and the cutting force are the major components in this setup, the overall effect will cut the surface 0.5 mm more than the commanded depth. In our definition, a negative surface error means that less material was removed than that of the commanded position. The result after deformation compensation shows a less than 0.1 mm surface error, which is in the range of the robot path accuracy. (Figure 14) Further test conducted on a foundry cylinder head workpiece shows that the surface accuracy improved from 0.9mm to 0.3mm, which is below the 0.5mm target accuracy for a pre-machining application.

Figure 14 Deformation Compensation Results

6. Conclusion

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

Easy robot programming is characterized by two main modules: lead-through and automatic path-learning. Lead-through provides robot operator the freedom to adjust the spatial relationship between the robot tool fixture and the workpiece easily, whilst the robot automatically follows the workpiece contour, records the targets and generate the process program in path-learning. Since the final robot program 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 adjustment of the robot feed speed. Various control strategy, including PID, adaptive control and fuzzy logic control, could be implemented depending on the different cutting situations

Considering the chatter and vibration analysis presented in another work [7], these complete set of solutions will greatly benefit the foundry industry with small to medium batch sizes. Consequently a dramatic increase of successful setups of industrial robots in foundry cleaning and pre-machining applications will be seen in the very near future.

References

- [1] Zengxi Pan, "Intelligent robotic machining with force control", Ph.D. dissertation, Stevens Institute of Technology, NJ, USA, Jan 2006.
- [2] Luis Basanez, Jan Rosell, "Robotic Polishing Systems—From graphical task specification to automatic programming", IEEE Robotics & Automation magazine, Vol 12, pp. 35-43, Sept, 2005.
- [3] J. Norberto Pires, et al., "CAD interface for automatic robot welding programming", Industrial Robot: An International Journal, Vol 31, pp.71-76, 2004.
- [4] Hui Zhang, Zengxi Pan, Zhenqi Zhu, et al, "Machining with Flexible Manipulator: Toward Improving Robotic Machining Performance", IEEE/ASME AIM, pp. 1127-1132, July 24-28, 2005
- [5] Tlusty, J., and Polacek, M., 1963, "The Stability of Machine Tools against Self Excited Vibrations in Machining", International Research in Production Engineering, ASME, pp. 465-474.
- [6] Gasparetto, A., 1998, "A System Theory Approach to Mode Coupling Chatter in Machining", ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 120, pp.545-547.
- [7] Zengxi Pan, Hui zhang, et al., "Chatter analysis of robotic machining process", Journal of Material Processing Technology, Volume 173, Issue 3, Pages 301-309, April 2006.
- [8] Sung I. Kim, Robert G. Landers, A. Galip Ulsoy, 2003, "Robust Machining Force Control with Process Compensation," Journal of Manufacturing science and engineering, Vol 125, pp. 423-430
- [9] Seung-Han Yang, 1996, "Real-time compensation for geometric, thermal, and cutting force induced errors in machine tools," Ph.D. dissertation, The University of Michigan
- [10] E. Budak, Y. Altintas, 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
- [11] ABB Robot User Manual, "IRC5 Reference"

- [12] Raibert, M.H. and Craig, J.J., 1981, "Hybrid position/force control of manipulators", ASME Journal of Dynamic Systems, Measurement and Control, 102, 126–133, 1981.
- [13] Pagilla, P.R., and Yu, B., 2001, "Robotic Surface Finishing Processes: Modeling, Control, and Experiments," ASME Journal of Dynamic Systems, Measurement, and Control, vol. 123, pp. 93-102.
- [14] Sung I. Kim, Robert G. Landers, A. Galip Ulsoy, 2003, "Robust Machining Force Control with Process Compensation," Journal of Manufacturing science and engineering, Vol 125, pp. 423-430
- [15] Landers, R.G., Ulsoy A.G., 2003, "Model-based Machining Force Control", ASME Journal of Dynamics Systems, Measurement, and Control, Vol. 122, No. 3, pp. 521-527, 2003
- [16] L.K. Daneshmend, H.A. Pak, 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.

Author Information

Zengxi Pan Lecturer

School of Electrical, Computer and Telecommunication Engineering, University of Wollongong, Wollongong
NSW 2252, Australia

Zengxi@uow.edu.au

Tel: 61-2-4221-5498

Hui Zhang

ABB Corporate Research China

31 Fu Te Dong San Road, Shanghai, China 200131

Email: hui.zhang@cn.abb.com