Robotic machining of aluminum alloy: evaluation of performance and part accuracy

Jingxin Zhang

2022

LUND UNIVERSITY

LTH

FACULTY OF ENGINEERING

MASTER THESIS

DIVISION OF PRODUCTION AND MATERIALS ENGINEERING

LUND UNIVERSITY

CODEN:LUTMDN/(TMMV-5344)/1-80/2022

Supervisor: Dr. Mathias Haage
Co Supervisor: Anders Robertsson, Professor
Examiner: Jinming Zhou, Professor

Author: Jingxin Zhang
Lund, Sweden 2022

Avdelningen för Industriell Produktion
Lunds Tekniska Högskola
Lunds universitet
Box 118
221 00 Lund
Sverige

Division of Production and Materials Engineering
LTH, School of Engineering
Lund University
Box 118
SE-221 00 Lund
Sweden

Printed in Sweden
Media-Tryck
Lund University

Foreword

First of all, I would like to thank my supervisors Dr.Anders Robertsson and Dr.Mathias Hagge for giving me support and feedback during this thesis project. Thank Dr.Jinming Zhou for

giving me advice and comments during the project. I would also like to thank Anton Tetov Johansson for the tool, workpiece, and force sensor calibration.

All of the work presented henceforth was conducted in the Robot Lab at Lund University. The project and associated methods were approved by the supervisors, Mathias Hagge and Anders Robertsson, from Computer Science and Control Departments, and examiner, Jinming Zhou, from Industrial Production Department.

This thesis is an original, unpublished, independent work by the author, Jingxin Zhang.

Lund 2022-06-09

Jingxin Zhang

Abstract

The automated machining is typically performed by CNC (Computer Numerical Control) machinery due to the high stiffness and accuracy provided by these machines, which are expensive and comparably big. For softer materials, such as aluminum, machining is possible using the less stiff industrial robot arm. This is attractive because the robot arm has a much lower cost than CNC combined with a larger work volume. Machining packages are now available for several robot brands, enabling robots to read and execute CNC G-code. It is particularly beneficial for small and medium enterprises. Robot arms are, however, less stiff and strong, resulting in less accuracy.

In this thesis, the idea was that by using the IRB 2400 robot to mill typical process paths to evaluate the performance and part accuracy in robotic machining of aluminum alloy. A pre-study towards a standard proposal for measuring robotic machining performance was performed.

Proper process parameters for robotic milling were applied, and the robot's task was to repeatedly mill parts in different postures. The path accuracy, microstructure, and milling forces were analyzed. Results showed that the posture of the robot has a significant effect on path accuracy. With a proper posture, the path accuracy could be under 0.01mm.

Keywords: Robotic machining, Path accuracy, Performance, Posture.

Table of Content

1. Introduction

The robotic arm is a part of the evolution of the industrial world like mechanization. The first robotic arm which was called Unimate #001, was designed and made by Devol and Engleberger in 1959. Unimate #001 was installed at a General Motors plant to assist a hot-die machine. By 1966, many robots were developed to assist with welding and other applications in the fast-growing automotive industry. The first industrial robot with 6 electro-mechanically driven axes was built by KUKA in 1973. The Swedish company released the first all-electric and microcomputer-controlled industrial robot IRB-6 in 1974, see in Figure 1. IRB-6 and its sibling IRB-60 also allowed continuous path motion, which gave themselves opportunities to do arc-welding and machining.

Figure 1: *ABB IRB-6*

In recent years, industrial robots are experiencing a fast-growing era. Dedicated controllers, new programming languages, offline computer-aid methods, and improved ability to be well programmed. They are popular in many industries. Industrial robots perform efficiently and stably in grinding, pre-machining, painting, welding, assembly, transmission, and cleaning processes. These operations demand high flexibility and little external loaded force would be applied. For example, in the grinding process, the robotic arm moves the grinding tool which is mounted on a flexible tool holder over the desired surface. In the case of a large volume workpiece, the robot is a perfect option to manage it.

Compared with CNCs, the high flexibility in addition to lower-cost make robots have a great tendency to be used in machining industries. Moreover, robots themselves are equipped with handling functions, which brings more benefits because CNCs need operators to load and unload parts every time. Normally, a workpiece always needs several different process procedures that need to be machined in several machines. It causes a long set-up time and results in a long cycle time. From this point of view, robotic machining can develop the automation level, efficiency and decrease labor costs as well.

Despite robots being so welcomed in the industry, they are still not well prepared for processes such as cutting and milling. According to the International Federation of Robotics (IFR), 72.7% of all industrial robots are used for handling and welding. Only 2.0% of industrial robots are used for the machining processes [1].

This is due to numerous reasons, for example, poor structural stiffness and unexpected vibrations are the main obstacles. Others like gear ratio, chatter, and backlash are also potential influences. They together affect the path accuracy and repeatability of the robot, resulting in poor part performance.

Stiffness is the extent to which an object resists deformation in response to an applied force. It is an important indication for informing the stability of the robot. For a 6R industrial robot, every joint stiffness is different from each other. Their stiffness also highly depends on the load magnitude and the posture i.e. the angle value of every joint. There are usually two ways to find out how much the stiffness is, establish a theoretical stiffness model or measure it experimentally. However, a stiffness model is extraordinarily complicated and the result of it has a large difference from the actual one because the model is highly related to the surrounding work environment. In addition, it is also hard to simplify a stiffness model because it is based on a varying Jacobian matrix. Besides, different robots must have a diverse stiffness model, which makes it even harder to establish a generic model for all 6R industrial robots. So, in such a circumstance, the experimental measurement is applied in this study.

As for the vibration in milling, it is composed of force vibration and self-excited chatter. The forced vibration could be caused by the clash between milling tool teeth and the workpiece, see Figure 2. The clash causes a time-varying external force that acts on the milling tooltips. The unbalance of a rotary member or a servo motor instability is also an example of forced vibration [2]. The self-excited chatter is caused by the repeated milling of a previous wavy-cut surface, resulting in micro-variations of chip thickness. As the chip thickness varies, so does the force [2]. Therefore, a time-varying force is generated. One of the best ways to improve robot machining performance is from the external force point of view because it is the basic issue. Either reducing the force or controlling it in an acceptive range would minimize the errors of accuracy to an extent.

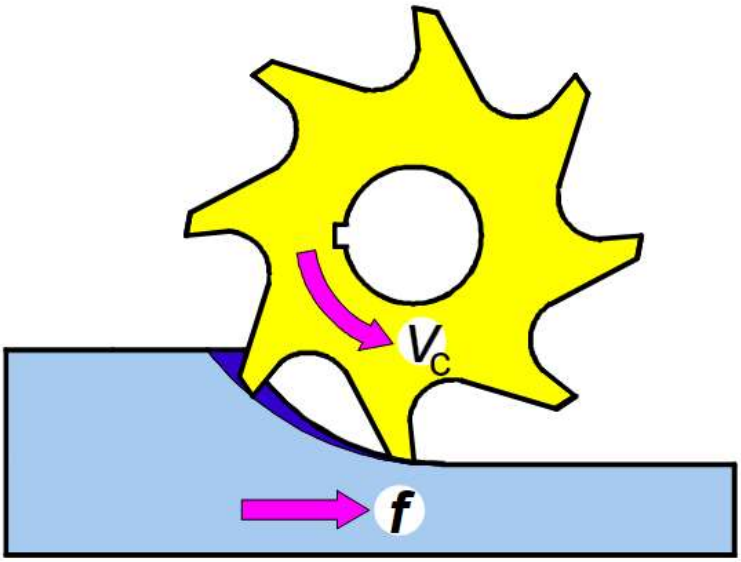


Figure 2: *Climb milling process*

In this study, a test methodology for performing a robotic machining performance test and analysis of test metrics would be researched with the use of the ABB IRB 2400 robot and other measure equipment. The purpose is to find out how well the path accuracy could the robot give under different work environments such as various workpiece placement, different machining process data, or different cutting features.

2. Problem statement

Automated machining (that is deflashing, deburring, milling, etc) is typically performed by CNC machinery due to the high stiffness and accuracy provided by these machines. For softer materials, such as aluminum, machining is possible using the less stiff industrial robot arm. This is attractive because the robot arm has a much lower cost than CNC combined with a larger work volume. Machining packages are now available for several robot brands, enabling robots to, for instance, read and execute CNC G-code directly.

Since the study of robot milling is in its early stage, a capable standard to evaluate how well the robot could perform in the milling process is not established until now. What we are doing now is finding a proper way to evaluate the performance of the robot machining. The basic information is from CNC evaluation standards like NAS 979 and ISO 10791-7. The milling features, the dimensions, the process parameters, and other requirements will be certainly adjusted according to the specific robot working environment.

3. Objective

The goal of the study is to evaluate the performance and part accuracy in robotic machining of aluminum alloy. In addition, this work is a pre-study of robotic machining. The result of this study could be a benefit in a general sense for further research in robotic machinings, such as deflashing, trimming, deburring, surface milling, drilling, and other flexible manufacturing processes based on an industrial robot.

4. Literature review

Before starting the machining process, it is very important to calibrate the workcell. According to the definition of repeatability and accuracy that is given by ISO 9283:1998, the typical accuracy claimed for an industrial manipulator is about ±1 mm, but values of 0.3 mm could be reached with accurate compensation. Repeatability instead ranges in between 0.1–0.03 mm [3]. Among all sorts of errors, the most evident and easily compensable source of errors is given by the measurement and correction of errors through the workcell calibration. Normally, the factory calibration is done by the manufacturer and it will keep the calibrated state by a very accurate resolver that is mounted on the motor axis. Workpiece and tool calibrations are needed for a new setup. Both of them could be accomplished through the offline program (OLP).

J.D Barnfather et al. [4] conducted experiments to quantify the performance of a Fanuc F200iB hexapod-format robotic machine tool to determine what machining challenges in terms of achievable tolerances. The authors captured the data of relation to various features and error tolerances. The result was that the magnitude of error has a strong relation with feature types. For example, the error of a cylinder was usually lower than 100 μm but of a nozzle, it was lower than 50 μm. The study result also showed that a large proportion of machining errors observed are systematic in nature.

The major hurdle preventing the adoption of robots for the machining processes is the fact that the stiffness of today's industrial robot is much lower than that of a standard CNC machine according to H.Z. et al. [5]. A robot manipulator could be deformed because of its compliance under an external force. The sources of the compliance are at the manipulator base, gearboxes, motors, links, and other transmission elements. The main challenge is to measure the stiffness at the condition of varying joints' values. H.Z. et al. made an intricate joint stiffness model which was based on feed-forward experiments. A deformation compensation system was developed based on the model to minimize the errors. The result shows a less than 0.1 mm surface error was achieved and the contact force-induced position error could be reduced by more than 60% using the ABB IRC5 robot.

R. Kelaiaia [6] used the same method as H.Z. et al.. The procedure was collecting data, sending it to a direct geometric model, then to the deformation model that was designed based on static stiffness values. The data were analyzed by deformation and corrected direct geometric models and then sent to a model of evaluation errors. A compensation model which calculates the jacobian matrix of the robot so that a modified result output. The trajectory errors were improved from 10 μm to 4 μm using a Delta robot.

The primary reason for deviations is the large process forces impact the robot body. Since it is not easy to establish a time-varying stiffness model, it is a great benefit in controlling the process forces instead [7]. O. Sörnmo et al. present a novel adaptive force controller, based on a derived model of the machining process and an identified model of the robot dynamics. But also they employed a stiffness model to continuously modify the robot trajectory to compensate for the deviations to satisfying results. The modeling and control design was based on the process parameters like feed rate. They aimed to control the force by adjusting the velocity of

the end-effector. A maximum allowed force and the amount of removing material were found and applied in this paper.

P. Ericsson [2] mentioned that in robotic milling, the excited vibrations are divided into two main categories: forced vibration and self-excited chatter. The former is generated by the interaction between the milling tool and the workpiece. A time-varying external force would act on the milling head. In the case that the frequency is close to the natural one, large vibrations due to resonance are likely to occur. As for the self-excited chatter could be caused by a wavy surface or the relative vibration between tool and workpiece. Consequently, small wave patterns and varying depth are about to occur. Reduced force or increased stiffness is recommended.

Ericsson also proved that a robot with a reducer can decrease the position error by approximately 7.5%. The rotation error could be corrected by 4.5% as well. Another way to revise the error is to increase the gear ratio. It was shown to reduce the mean position error by approximately 13 % and the mean rotation error by approximately 8 %.

Lin Yang et al. [8] proposed that it is reasonable to consider performing machining in the regions of the robot workspace where the kinematic, static, and even dynamic performances are highest, thereby reducing machining errors and exhausting the advantages of the robots. They believed that there is usually an optimum initial place for a workpiece where the robot has better performance. They studied kinematic modeling for Comau Smart5 NJ 220-2.7 robot from different positions. Every joint angle and its deformation were calculated to find out the stiffness performance concerning the workplace. The result showed that the maximum deformation difference value between optimized and unoptimized postures was around 0.5 mm and that the maximum joint angle difference was 1.5 milirad.

G. Janez et al. [9] also had the same idea. They tried to analyze the accuracy from static, kinematic, and dynamic points of view respectively. The structural stiffness of the robot was defined as a static factor and manipulability as a kinematic one and damping ratio, structural inertia, as well as presence of a resonate state as three dynamic factors. But the result showed that it is not possible to set the establishment of individual factors without interaction effects. As a result, the author used ACMA XR701 to make holes and evaluated the deviation for all factors. It showed that the larger the hole's diameter is, the lower accuracy is. For the maximum one, deviations from nominal diameter range from 0.23 to 1.86 mm, while for cylindricity, the tolerance interval is 0.16 to 0.87 mm.

Alexander Verl et al. [10] studied the theory of allied robots in machining. They found some optimized ways for increasing the accuracy of robot machining. They suggested that simulation of robotic systems could be interfaced with analytical or numerical models to compute robot motion adaptations for increased tool path accuracy. Process redundancy is also a good method to improve accuracy. An ideal suggestion was that make robot systems learn from actual experience and keep updating themselves during the process. In such conditions, developed input devices like controllers could facilitate the programming experience. Process optimization algorithms were mentioned to create a cloud-service platform for the auto-check.

The self-learning robot system will probably be realized in the near future. In the current situation, it is possible to provide feedback to the robot online, i.e., in real-time, to correct its performance. Tomas Kubela et al [11] investigated the causes of accuracy errors. They

proposed that there are always environment-dependent errors, robot-dependent errors, and process-dependent errors. An online system was developed to compensate for the robot trajectory and the actual positions of the robot based on some sensor information. A laser tracker was mounted on the robot. It collected the data of the robot's actual movement and sent it to an industrial PC, where the data was processed and provided signals to control the velocity and position. Inside the robot was also an initial controller. It received control signals and then compensated itself. The final accuracy was about 0.1 mm and could be further improved.

Compared with online methods, offline [1] methods deal mainly with identified robot parameters for correction of the kinematical model and/or adapting the path-planner based on actual robot properties. In most cases, offline programming is not directly transferred to the servo controller, as the programmed positions will most likely differ in the real practical environment [12]. What offline methods can do to improve accuracy is usually to find a better posture for each specific process. Offline software can provide information about the joint area and singularity position. Additionally, a simulation model can quantify the vibration error along with a specified trajectory.

The result could be much improved if combine both online and offline methods according to A. Klimchik et al. [13]. Besides the drawbacks of robots mentioned before, they realized that the main obstacle to robots utilization in machining is their relatively low accuracy (around 0.7 mm) and repeatability (around 0.2 mm) compared to the CNC machines. A method combining online and offline was designed to modify the trajectory to make robots follow the desired one. In the offline system, an error compensator is included which gets estimated loading data from an actual technological process. The error compensator analyzes the data first by an actual robot model including geometrical and stiffness models. Joint coordinates as a result imported to the nominal robot model and then output Cartesian coordinates to modify the trajectory. The modified trajectory would be sent to the online system, which includes the nominal robot model and actuators. The actuators give joint coordinates orders to the actual manipulator and finally, an obtained trajectory comes out as a result. Such a method allows for reaching a 95% error reduction level.

When it is speaking of the performance, it is necessary to analyze the workpiece. One of the important parameters is surface quality. B. Majedi [14] claimed that the geometry of the tooltip, the feed rate, and the kinematics of the machining process are the main parameters affecting the roughness. Additionally, the geometric errors of gear trains have a major effect on the surface quality and generate major irregularities on the machined surfaces. Surprisingly, the author proved that irregularity on the machined surface due to the low-frequency vibration of the robot is small. The vibration is mainly from the tool. To compensate for it, the author proposed a compliant tool holder. It is believed that, when the tool is cutting extra material, the tool would be displaced in the feed direction so that the feed speed will decrease for a while.

The five-axis milling machine is commonly used in the industrial world. One of the first tests of the accuracy of a five-axis milling machine was the NAS 979 [15], which was developed 49 years ago [16]. For example, as **Figure 3** shows, a milling test for material 7075-T6 Aluminum is designed to mill a certain shape and evaluate its, for instance, width and depth of cut.

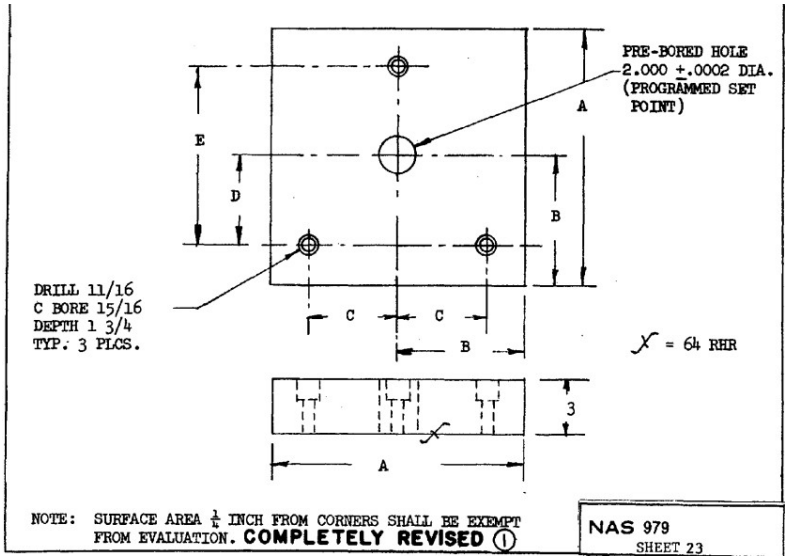


Figure 3: *NAS 979 milling test example[15]*

The ISO 10791-7 standard was established in 1998 and consists of two machining tests. The first one is a test of making a complex test piece using an end mill cutter. The second one is a flat surface machining test performed by face milling [16], which will be introduced in section 6.3. To evaluate the performance, several indexes are about to be measured after the process. For example, the cylindricity of the bored holes, straightness of the sides, roundness of the circle, and parallelism of the sides to the planes. **Figure 4** shows the measurement of the geometric test.

Object and references to the drawing		Tolerances Nominal size			Measuring instruments	Measured deviations
		80	160	320		
Central hole	Cylindricity of the bored hole C	0,010	0,010	0,015	CMM[a]	
	Perpendicularity between the hole C axis and datum plane A	0,010	0,010	0,015	CMM	
Square	Straightness of the side B	0,005	0,008	0,015	CMM or straightness reference artefact and linear displacement sensor	
	Straightness of the side F					
	Straightness of the side G					
	Straightness of the side H					
	Perpendicularity of the side H to datum plane B	0,010	0,010	0,020	CMM or squareness reference artefact and linear displacement sensor	
	Perpendicularity of the side F to datum plane B					
	Parallelism of the side G to datum plane B	0,010	0,010	0,020	CMM or height gauge and linear displacement sensor	
Diamond	Straightness of the side K	0,005	0,008	0,015	CMM or straightness reference artefact and linear displacement sensor	
	Straightness of the side L					
	Straightness of the side M					
	Straightness of the side N					
	Angularity of 30° angle of side K to datum plane B	0,010	0,010	0,020	CMM or sine bar and linear displacement sensor	
	Angularity of 60° angle of side L to datum plane B					
	Angularity of 30° angle of the side M to datum plane B					
	Angularity of 60° angle of the side N to datum plane B					
Circle	Roundness of the contouring circle P	0,015	0,015	0,020	CMM or linear displacement sensor or roundness-measuring instruments	
	Concentricity of the external circle P and datum hole C	0,025	0,025	0,025		
Sloping faces	Straightness of the face I	0,005	0,008	0,015	CMM or straightness reference artefact and linear displacement senor	
	Straightness of the face J					
	Angularity of 3° of the side I to datum plane B	0,010	0,010	0,020	CMM or sine bar and linear displacement sensor	
	Angularity of 93° of the side J to datum plane B					

Figure 4: *Contouring test piece geometric tests*

5. Fundamental knowledge of robotics

In this section, the relevant theory is going to be introduced. In the 5.1 section, robot coordinate system is discussed. The kinematic analysis of a serial robot is introduced in section 5.2. The stiffness of the robot and how much it could affect the milling process performance are discussed in the 5.3 section.

5.1 Robot Coordinate system

5.1.1 Base frame

The base frame indicates the frame which is fixed on the robot base. The origin of the base frame locates at the center of the base. The X-axis is always pointing straight forward and Z-axis is perpendicular to the base surface. According to the Cartesian coordinate system, the Y-axis is pointing in the left direction. If there is only one robot working in an area, the base frame can also act as a world frame. It is a coordinate system that can be used to acknowledge a robot where it is. The following introduced frames are defined also based on the base frame.

5.1.2 User frame

User frame is often defined by the user to align a specific Cartesian coordinate system of a workpiece. It is translated or rotated concerning the world or base frame. The following introduced frames are user frames.

5.1.3 Tool Center Point, TCP

When a robot is moving to a given target, it is the tool center point that will follow the exact target points to accomplish the path movement. During the movement, rotation and translation of every joint and link are related to the position of Tool Center Point, TCP. TCP is usually defined to be the center of the tool e.g. the welding gun and milling tool.

5.1.4 Tool frame

To define a TCP, a tool frame is required. It is a coordinate system fixed on the tool. Usually, the origin center of a tool frame coincides with the TCP. A tool frame is very important because it decides every joint's moving angle. A proper tool frame also can avoid singularity.

It is easy and quick to define a tool frame virtually. For example, import CAD models of the tool and other bits which are mounted on the TCP into software like RobotStudio. Create a new frame based on the wanted point and it is possible to translate or rotate the frame.

However, in the case of lacking a CAD model, doing the tool calibration to create a tool frame is a better method. In this study, a four-point with an elongate Z-axis calibration is used. The idea is that manipulate the TCP pointing on the same position from four different directions, which gives the robot four different vectors to find the position. The extra Z-axis should be calibrated for the reason that making milling tools always point down. The procedure is illustrated in **Figure 5**.

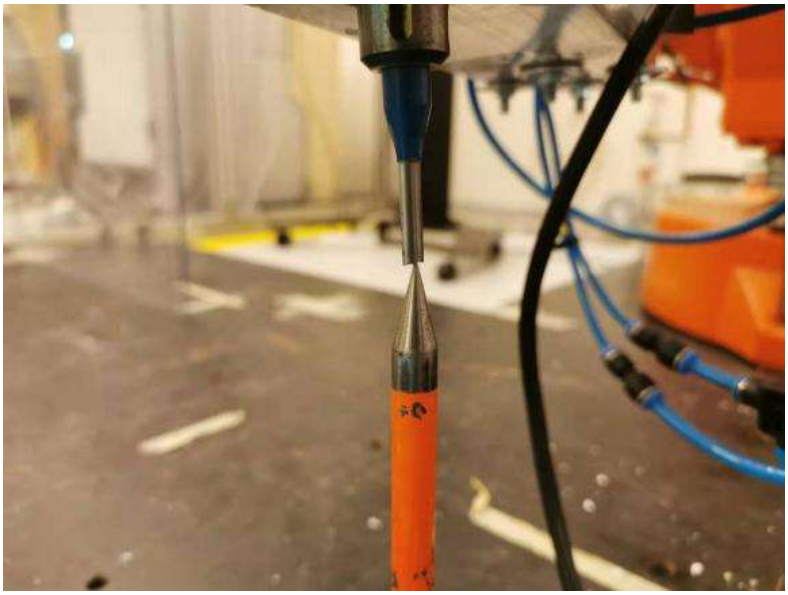


Figure 5: *Milling tool calibration*

5.2 Kinematic analysis

Robot mechanisms are systems of rigid bodies that are connected by joints. The description of the position and orientation, velocity, and acceleration of bodies is called kinematic analysis. In a so-called serial robot mechanism system, one joint connects two rigid bodies. To fully understand and predict the end-effector motion, it is crucial to set up an adequate kinematic model. The complexity is dependent on the prediction precision, and the articulated robot is regarded as complicated due to how serial build-up introduces modeling challenges [2]. The model can be used for motion planning. In this study, it is available to use to design milling paths.

In a typical CNC machine system, the axes are aligned with the work-space coordinate system, and each of them is independent. This means moving along one axis will not affect others. For example, in **Figure 6**, the machine can point to x-direction by only moving along the guide rail above the workspace, and meanwhile keeping y- and z-direction still.

However, for a serial chain, it is not possible to move the end-effector by only involving only one joint, for a rotation or a translation of an axis always causes multi-joint movement. The rotation of one axis alters the behavior of every consecutive axis and therefore, industrial robots are said to be kinematic non-linear [2]. During the movement, the stiffness, the compliance, and the mass distribution keep changing. In the case of a large payload being applied, the torque and force direction are also changed. As a result, the position accuracy is under the effects of all of the factors, and kinematic errors appear.

The Denavit-Hartenberg (DH), is a set of parameters that defines the reference frames for each joint in the kinematic chain, illustrated in **Figure 7**. It is a convenient way of modeling a robotic system. After defining the parameters of the robot and summarizing them in a table like shown

in **Table 1**, a virtual model can be built up based on them with the support of Matlab, illustrated in **Figure 8**.

Table 1: *D-H parameters*

Link	θ_i	d_i	a_i	α_i
1	0	0	0	$\pi/2$
2	$\pi/2$	0	0	$\pi/2$
3	0	0	a_3	0
4	0	0	a_4	0
5	0	0	0	$-\pi/2$
6	$-\pi/2$	0	0	$-\pi/2$

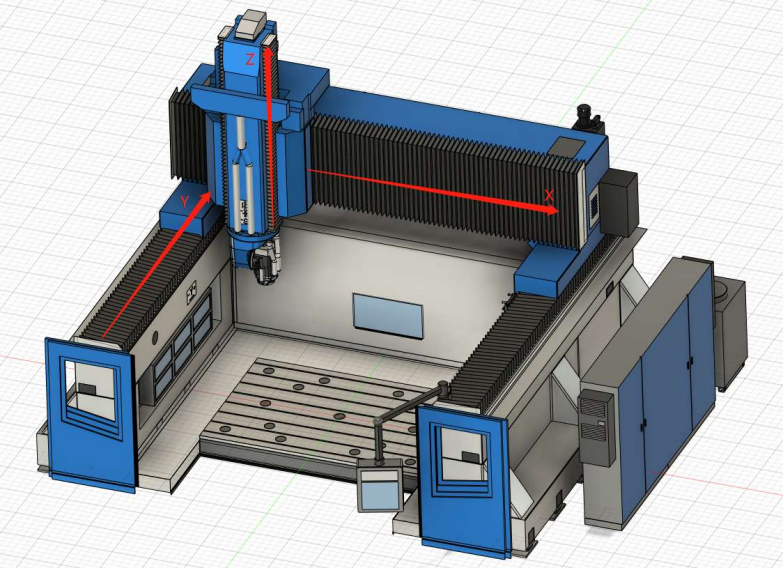


Figure 6: *Comi laborshape 3020 CNC machine model*

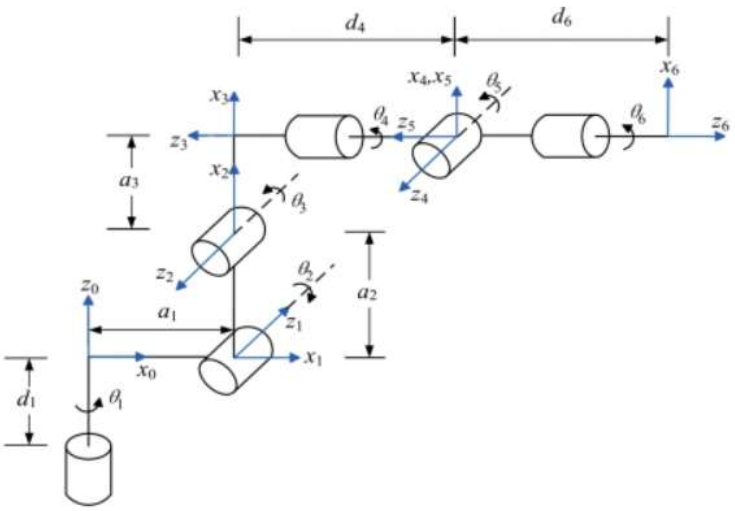


Figure 7: *Denavit-Hartenberg convention [2]*

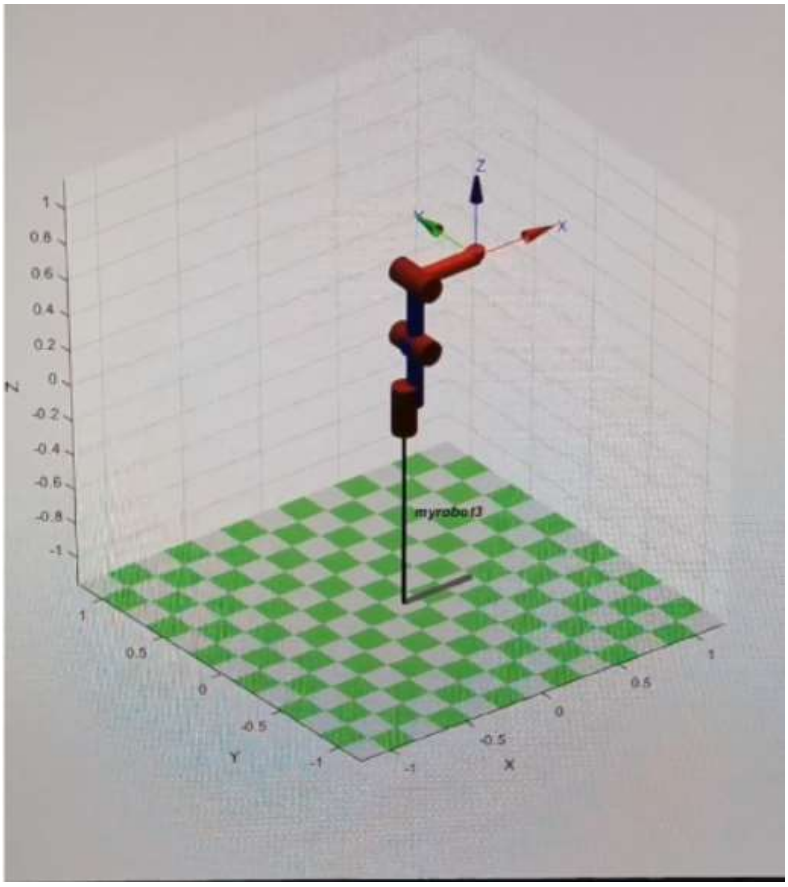


Figure 8: *The model built with DH parameters*

5.3 Stiffness analysis

For serial manipulators, the conventional stiffness model is expressed as [8]:

$$K = J^{-T} K_\theta J^{-1} \tag{1}$$

Where K and K_θ are the Cartesian and joint stiffness matrices respectively and J is the Jacobian matrix.

Then, the enhanced stiffness model is expressed as [8]:

$$K = J^{-T}(K_\theta - K_C)J^{-1} \tag{2}$$

Where K_c denotes a complementary stiffness matrix. For a given wrench applied on the robot end-effector, K_c was verified to be negligible in the workspace with better kinematic performance. According to Eqs. (1) and (2), K_θ is defined as [8]:

$$K_\theta = \begin{matrix} k_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & k_2 \end{matrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \tag{3}$$

where k_i (i = 1, 2, ..., 6) is the stiffness of Ji, K_{11}, K_{12}, K_{21}, and K_{22} are 3×3 submatrices of K_θ.

From Eqs. (1) and (2), it can be seen that calculation of the inverse Jacobian matrix is involved, which inevitably introduces a calculation error when the robot is close to singularities. To solve such a problem, a compliance model was proposed and defined as [8]:

$$C = JK_\theta^{-1}J^{-T} = \begin{bmatrix} C_{tt} & C_{tr} \\ C_{tr}^T & C_{rr} \end{bmatrix} \tag{4}$$

where C_{tt}, C_{tr}, and C_{rr} are the 3×3 translational, coupling, and rotational compliance submatrices, which are not the inverse of the stiffness submatrix [8].

By neglecting the rotational deformation and the cutting moment applied to the tool, the overall compliance of the robot is proportional to the volume of the translational compliance ellipsoid. Thus, the performance index of the robot stiffness can be defined as [8]:

$$k_{stiff} = \frac{1}{\sqrt[3]{detC_n}} = \frac{1}{\sqrt[3]{\det(J_{11}K_{11}^{-1}J_{11}^T + J_{12}K_{22}^{-1}J_{12}^T)}} \tag{5}$$

where J_{11} and J_{12} are 3×3 submatrices of J, which can be written as [8]:

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{6}$$

From Eq. (5), it can be seen that the stiffer robot posture can be obtained when the index tends to be larger.

The extent of the displacement depends on the value of the stiffness and the external forces as well. Eq. (6) shows the relationship between displacement δ and force.

$$\delta = \frac{F}{k} \tag{7}$$

So, the larger the external force is applied, the more displacement occurs.

6. Research method and experiment design

In this section, research method, experiment equipment and their work principle will be introduced. **Figure 9** shows all the devices used in this experiment at an assembled state. They are ABB IRB 2400, a force sensor, the milling tool, and the Nikon measurement system.

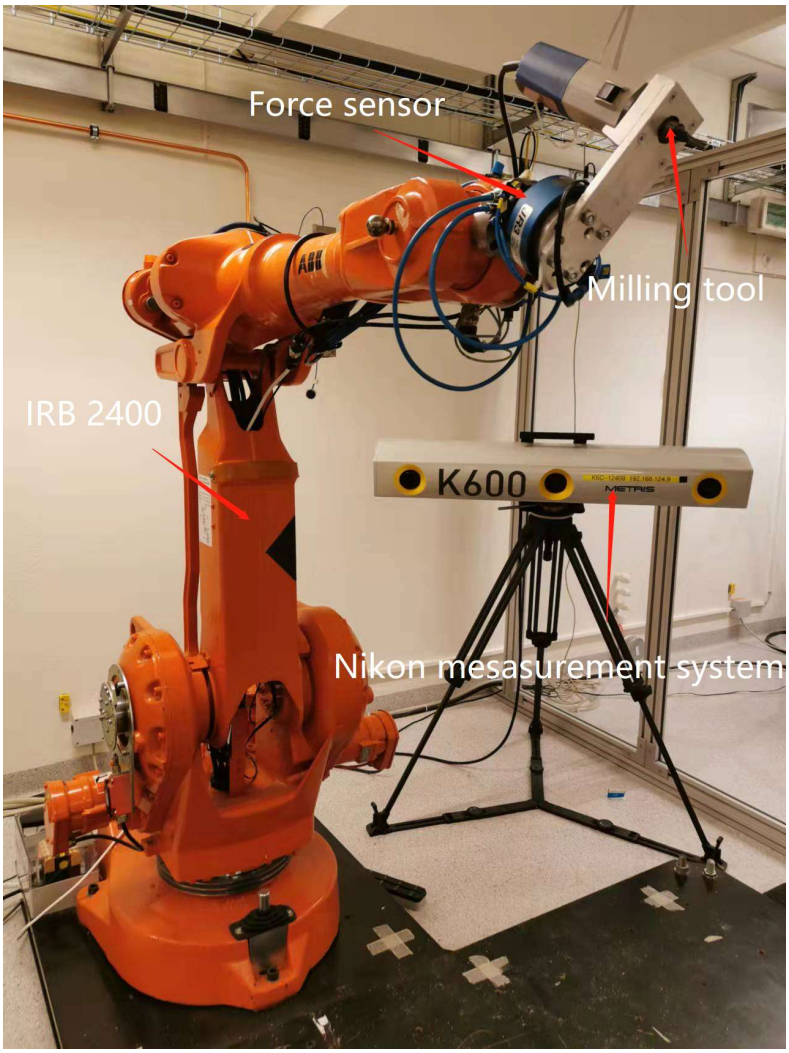


Figure 9: *Experiment setup: IRB 2400, force sensor, milling tool, Nikon measurement system*

6.1 Research method

The process of the experiment is first to build a CAM model by using Fusion 360 and then generate a RAPAID code used to control the robot. If the simulation in RobotStudio has no problem, the next step is to send the program to the robot. While the robot doing the process, both the force sensor and the Nikon metrology system are capturing data at 1000Hz and 100Hz frequencies respectively. All the data captured will finally be transferred to a laptop to be processed by using Matlab. The illustration of the procedures is shown below in **Figure 10**.

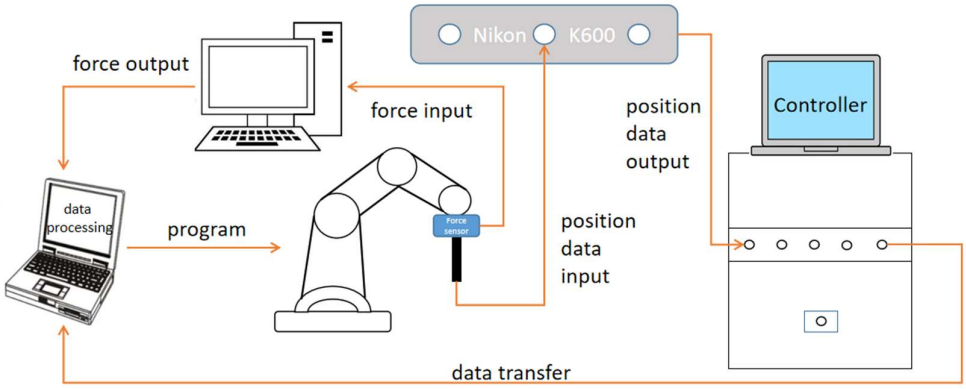


Figure 10: *Illustration of the experiment*

The method of designing the experiments is first to find proper process parameters which are suitable for industrial robot milling. Several different MRRs are tested like 1.8, 1.44, 0.6, and 0.48 cc/min in Location 1. The details of choosing machining parameters will be introduced in section 6.3.3.

After the process parameters are confirmed, they are used to mill a square in three different positions as **Figure 11** shows. The distance between the robot's center and the center of Location 1 is 1350mm, 1569mm of Location 2, and 1570mm of Location 3. To get data not only from a 2D plane but also from a 3D space, Location 3 is designed to be perpendicular to the horizontal plane. Thus, it is more comprehensive to evaluate the milling performance of a robot.

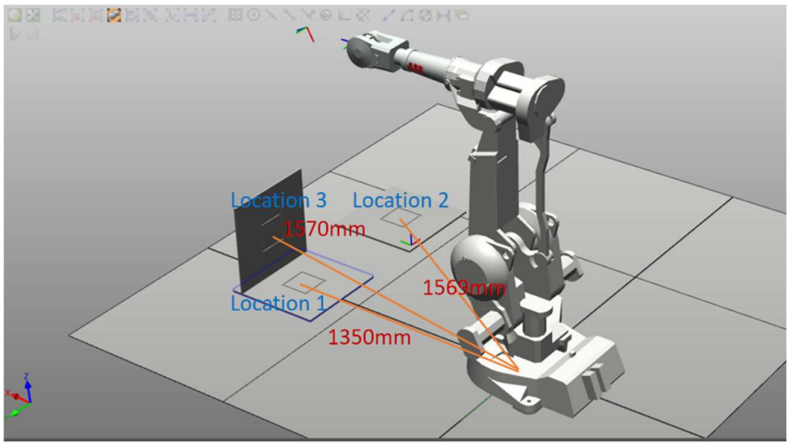


Figure 11: *Design of the experiments*

For each Location, the process is repeated twice. One for air movement, which means that the milling tool cuts nothing but air. And the other one is for the real cut i.e. cutting aluminum material. The reason for such a design is for comparing the performance of the process with and without cutting material. The performance of air motion includes the errors of the robot itself and the vibrations of the spindle etc. Therefore, by comparing them, we can know more precisely how the robot acts during the process.

6.2 ABB IRB 2400 robot

The ABB IRB 2400 robot is widely used in the industry world. It is a typical 6R serial robot, designed specifically for manufacturing industries that use flexible robot-based automation. The robot has an open structure that is specially adapted for flexible use and communicates extensively with an external system. The robot can be equipped with optional software for application support, for example, gluing and welding, communication features like network communication, and advanced functions such as multi-tasking, sensor control, etc. The manipulator axes of IRB 2400 are shown in **Figure 12**, which is provided by the ABB manufacturer. From the product specification of IRB 2400, its dimensions are provided as **Figure 13** shows.

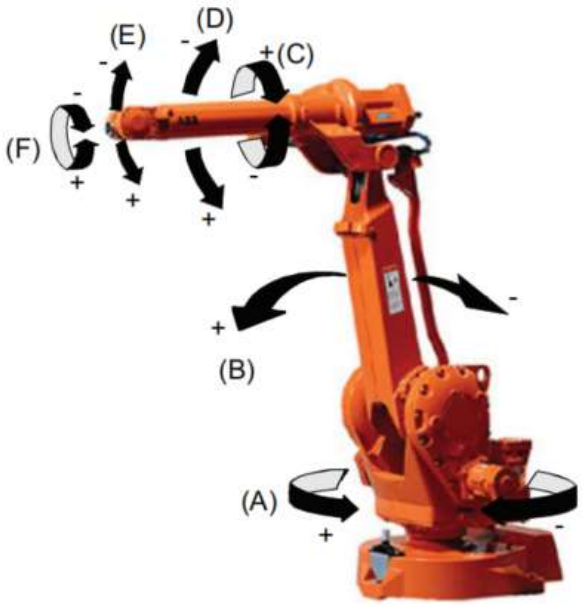


Figure 12: *IRB 2400 manipulator axes*

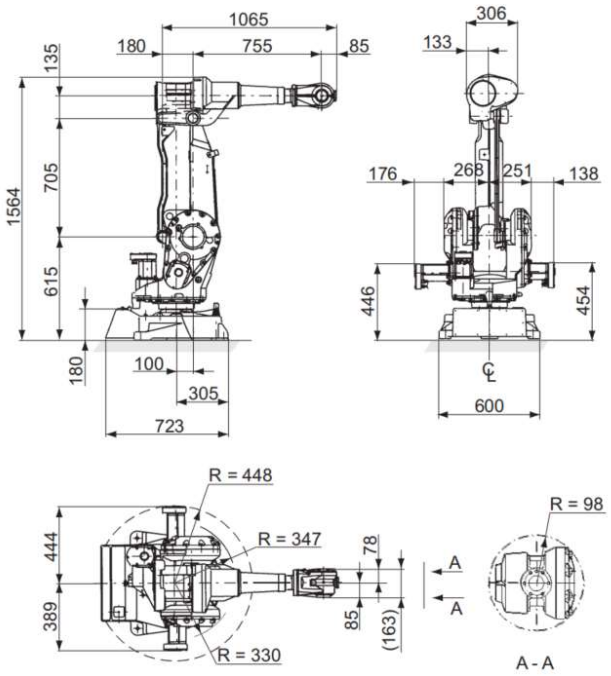


Figure 13: *Dimensions of IRB 2400/16*

6.3 Milling tool, work material, and machining parameters

6.3.1 Milling tool

The type of the spindle is Solectro UFM 500 whose speed range is 11000-25000 rpm. A mechanical structure is used to connect the spindle and the force sensor. And a tool changer is applied to mount the force sensor on the robot's original TCP, tool0. The parameters of the milling tool are summarized in Table 2.

Table 2: *Milling tool parameters*

Flute diameter	Overall length	Shoulder length	Flute length	Number of flutes	Front angle	Back angle
mm	mm	mm	mm	mm	º	º
5	45	26	21	2	0	10

6.3.2 Work material and milling features

The workpiece is EN AW-5083 Case and flat milled: 10 mm thickness, 500×500 mm^2. AW-5083 is Aluminum that is mainly made of AlMg4.5Mn0.7. Aluminum 5083 is widely used in the industry, for example, in shipbuilding and cars. The properties of the material are collected in **Table 3**.

To generate a couple of generic paths, several milling features are selected according to BS ISO 10791-7:2014, as shown in **Figure 14**. Due to the thickness of the prepared workpiece, only the first 5 mm thickness of the drawing is milled in this experiment.

Table 3: *Workpiece material properties*

Modulus of elasticity E **GPa**	**Shear module G** **GPa**	**Tensile Yield Strength σ** **MPa**	**Stretch limit Rp** **0.2MPa**	**Hardness H** **HBW**
71	27	228	115	75

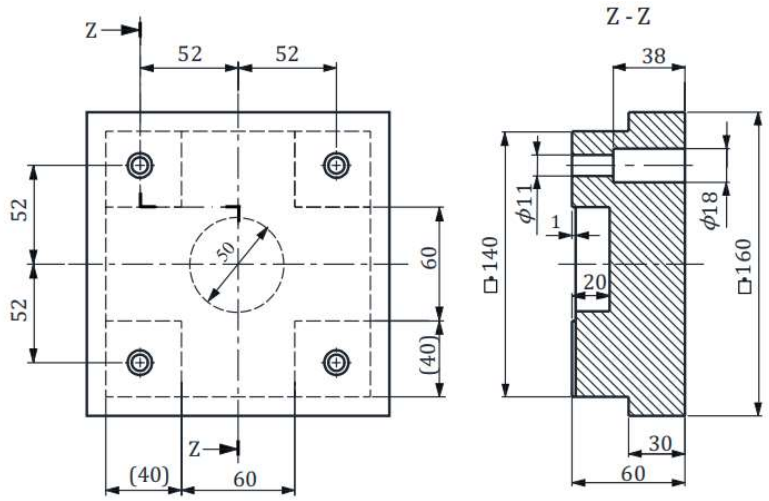


Figure 14: *ISO 10791-7, M1-160*

6.3.3 Machining parameters

6.3.3.1 Cutting speed, v_c

Based on the workpiece material, which is Aluminum 5000 series, and milling tool material, which is high-speed steel, a range of cutting speed is recommended for semi-finishing as 250-800 m/min. Cutting speed has a significant influence on cutting heat. It has been proved that the higher the cutting speed is, the larger amount of heat created. In this study, we use dry cut because it is difficult to add lubricant or coolant equipment to the robot. Therefore, it is necessary to control the temperature in a reasonable state. In addition, considering the stiffness of the robot, a low cutting speed is a better option. As a result, 300 m/min is chosen.

6.3.3.2 Spindle speed, N

Spindle speed, N in round per minute (RPM) is determined based on cutting speed v_c and the tool diameter D. The unit of v_c and D are m/min and mm respectively.

$$N = \frac{1000 v_c}{\pi D} \qquad (8)$$

The milling tool diameter is 5 mm. The spindle speed is 19099 RPM.

6.3.3.3 Feed rate, v_f

Feed rate, v_f is related to spindle speed N, feed per tooth f_z, and the number of cutting tooth Z.

$$v_f = f_z \times N \times Z \tag{9}$$

In this study, the feed rate is the moving speed of TCP. Regarding the low stiffness of the robot joints, it is important to decide on a proper feed for good performance.

We decide to determine a proper feed rate through workpiece quality and sound spectrum, two aspects.

It is necessary to mention that when it begins milling the first layer, the feed rate was 2 mm/s, the cutting width was 5 mm and the cutting depth was 1mm. A great vibration and large noise occurred. In addition, 1.8 cc/min and 1.44 cc/min Material Removal Rate (MRR) are also tested. The result shows a fatal performance, which will be analyzed in the Data analysis section. So that we define its MRR 0.6 cc/min as a limit.

Therefore, during the test experiments, we tested different feed rates and cutting widths but kept the cutting depth remaining at 1 mm because 0.5-1 mm is recommended in such a case.

6.3.3.4 Workpiece quality

For the circle feature, we can see from Figure 15 and Figure 16, there is little difference between 2 mm/s and 8 mm/s. The milling trace of the higher feed one is less than the lower one, especially around the circle. But in this case, it is hard to conclude which one is better.

Figure 15: *Circle feature with vf = 2 mm/s*

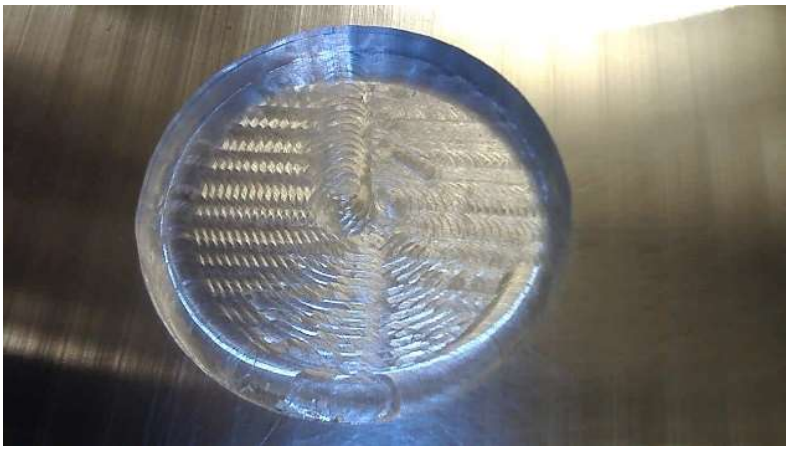


Figure 16: *Circle feature with vf = 8 mm/s*

However, for the square feature, as shown in **Figure 17** and **Figure 18**, it is much clearer that the quality of the higher feed one is better than the lower one. In this case, we can conclude that a 4 mm/s feed rate is better for this process.

It is necessary to mention that the quality is mostly determined by the force, which would cause vibration. So, to have an even force and control the vibration, it is better to have the same Material Removal Rate (MRR)during the process. In this case, the MRR is 0.48 cc/min.

Figure 17: *Square feature with vf = 2 mm/s*

Figure 18: *Square feature with vf = 4 mm/s*

6.3.3.5 Sound spectrum

An experienced machinist often uses milling sounds to judge what is occurring in the cutting zone. It is necessary to use sound to support finding proper data because the sound tells us whether it is good for the workcell, including milling bits and workpiece.

For the circle feature, the sound spectrum of feed 2 mm/s and 8 mm/s are shown in **Figure 19** and Figure 20 respectively. The yellow lines are the maximum dbFs for each frequency during the measurement. The frequency features like 1.1 kHz, 1.4 kHz, 2.2 kHz, etc. are similar. It means each point of the milling tool creates its frequency during the process by impacting the workpiece.

The energy difference between the peak and trough of feed at 8 mm/s is larger than that of 2 mm/s. It indicates that the impact between the non-cutting edge and the workpiece is more gentle i.e. the non-cutting edge has less impact on the workpiece. As a result, it causes less vibration and a smaller force, which is better for the process.

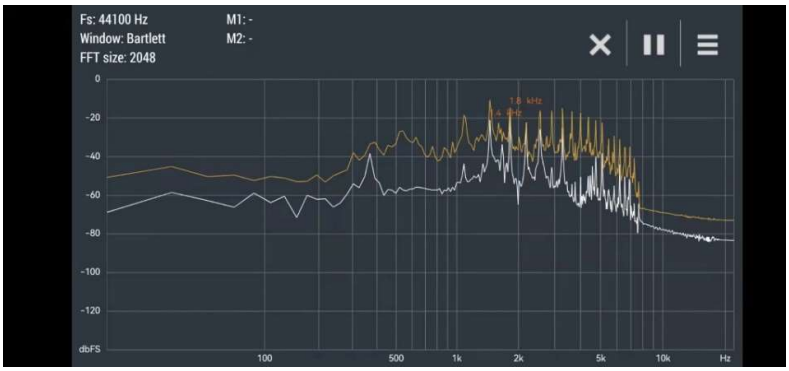


Figure 19: *Sound spectrum of circle feature with vf = 2 mm/s*

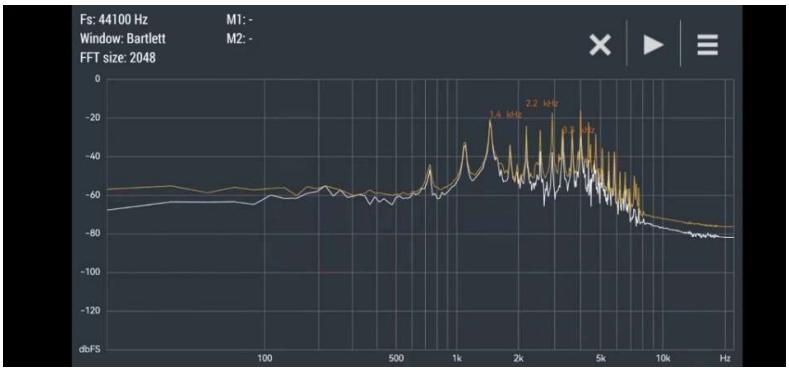


Figure 20: *Sound spectrum of circle feature with vf = 8 mm/s*

As for the square feature, the sound spectrums of v_f = 2 mm/s and v_f = 4 mm/s are shown in **Figure 21** and **Figure 22** respectively. They illustrate the same trend as circles.

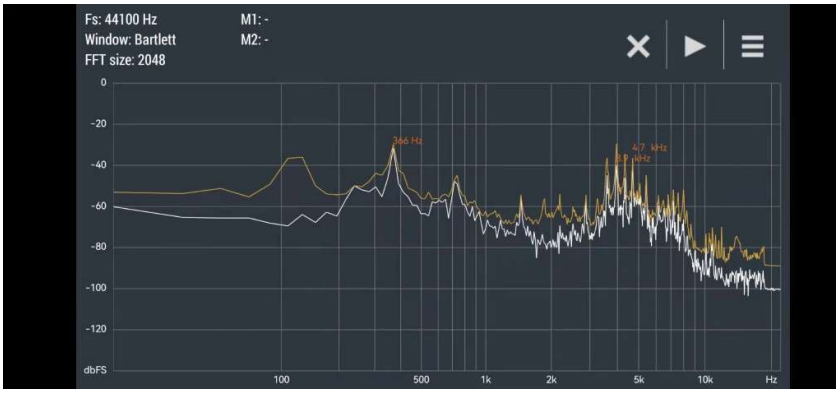


Figure 21: *Sound spectrum of square feature with vf = 2 mm/s*

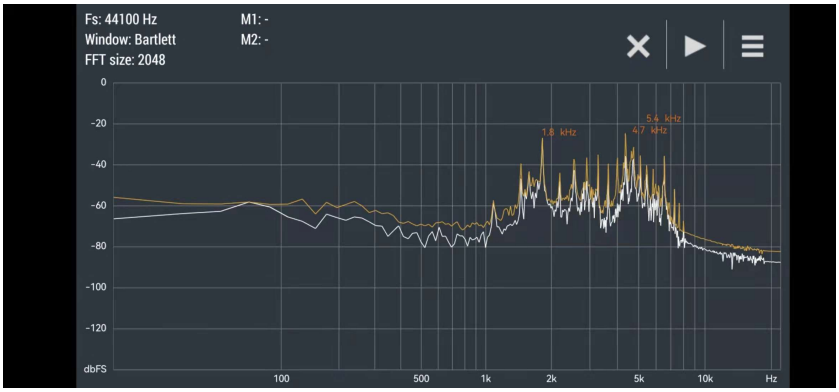


Figure 22: *Sound spectrum of square feature with vf = 4 mm/s*

6.3.3.6 Parameter summary

According to what has been discussed above, we can summarize proper machining parameters as **Table 4** shows.

Table 4: *Machining parameters for the process*

Feature	N RPM	v_c m/min	v_f mm/min	a_p mm	a_e mm	MRR cc/min
Circle	19000	300	480	1	1	0.48
Circle	19000	300	96	1	5	0.48
Square	19000	300	240	1	2	0.48
Square	19000	300	96	1	5	0.48

6.4 Measurement system

6.4.1 Nikon metrology system and its calibration

The standard instrumentation of 2D sensors is only able to measure in one or two directions, resulting in limited information available or complicated instrumentation plans. Instead, Nikon Krypton K600 is an optical coordinate measuring machine as a 3D sensor. Three cameras triangulate the locations of infrared LEDs. Two outer cameras calculate y and z positions while the middle calculates the x position. With such a measurement, it supports up to ± 0.0008 mm resolution.

The principle of the system is that the controller signals strobe to light up the LEDs which are glued on the tool, and camera to capture LED positions. Then the camera triangulates the LED positions. Finally, the camera returns LED coordinates to the controller. Krypton defines a coordinate system with the returned data. The basic principle is shown in **Figure 23**.

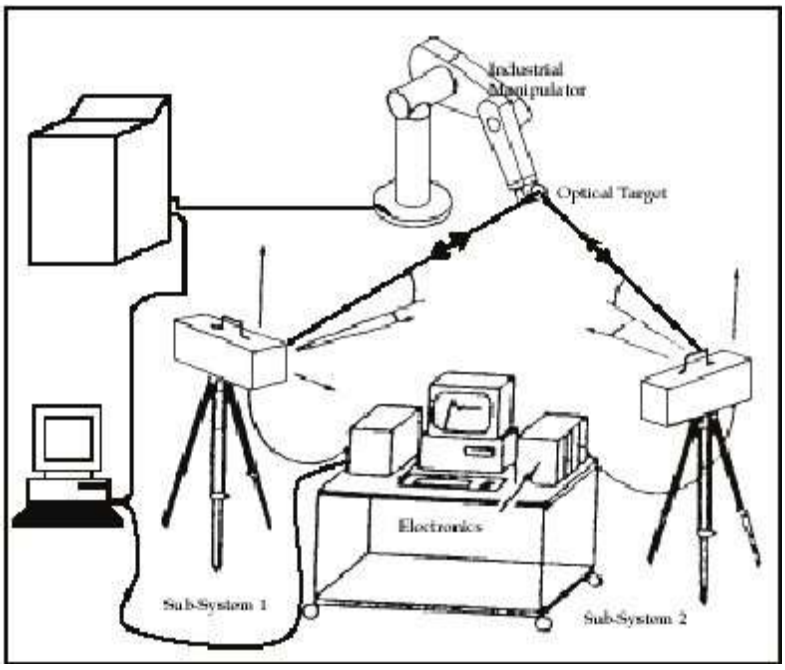


Figure 23: *The measurement system including the Optortrac & motor angles and currents acquisition hardware[17].*

As **Figure 9** shows, three cameras point to the three LEDs respectively. Thus the point cloud read by the cameras has an angle between the real process plane, which makes the path figure look like a tilted rhombus in a 3D space.

The method is first to find three points from the point cloud, assuming they are in the same plane, and then normalize it getting a normal vector. With the normal vector, the angle between the plane and the x-y plane can be calculated. After rotating the point cloud plane until it is parallel to the x-y plane and translating the start point close to (0,0), the calibration is done. All of these steps are done with Matlab. The code is also attached in Appendix (B).

6.4.2 Force/torque sensor and its calibration

The JR3 multi-axis load cell, also commonly known as a 6 degree-of-freedom force-torque sensor, contains analog and digital electronics systems. The strain gauge signals are amplified and combined to produce analog representations of the forces along three orthogonal axes (x,y, and z) as well as the moments or torques about those axes. In most models, the analog data is converted to digital form by electronic systems contained within the sensor.

However, the force sensor should be calibrated to fit the situation of the milling process before they are used to collect data.

As mentioned before, the force sensor needs a couple of mechanical connections with the milling tool and the robot. Therefore, the extra mass is applied to it, which would influence the data precision. To calibrate it, Our solution to this problem was to use a force sensor that was attached to the robot and connected to a computer. The sensor readings were then captured and all calculations were done in Matlab. The robot movement and program were done in the RobotStudio. Only joints 4, 5, and 6 are used to describe the translation between the tool frame

and the frame at the sixth joint and thus being able to translate forces measured at the sensor to the forces acting on the tool. The joint angles and the sensor readings are summarized in **Table 5**. The Matlab code and RobotStudio RAPID code are attached in Appendix (A). The result of the orientation error would be involved in the calculation of the force data processing.

Table 5: *Force sensor calibration data*

Joint angle	x	y	z	xtorque	ytorque	ztorque
0,90,0	2.079684	21.829126	2.435652	-15.844142	-0.067321	-0.058229
0,60,0	4.124119	20.973083	1.865139	-15.82905	-0.020078	-0.058229
0,30,0	5.886563	20.350505	0.285257	-15.825905	0.008267	-0.050465
0,0,0	6.521042	20.117039	-1.996796	-15.816785	0.005905	-0.050465
0,-30,0	5.886563	20.117039	-4.761589	-15.853262	-0.020078	-0.058229
0,-60,0	4.476608	20.895261	-6.341472	-15.853262	-0.031889	-0.042702
0,-90,0	2.150182	21.751305	-6.473129	-15.880618	-0.057872	-0.042702
0,90,0	2.009186	21.906950	2.435652	-15.844142	-0.057872	-0.058229
30,90,0	1.304209	19.805750	2.523423	-15.816785	-0.043699	-0.050465
60,90,0	0.599231	18.171484	0.241371	-15.789429	-0.043699	-0.042702
90,90,0	0.317240	17.626729	-2.303995	-15.798548	-0.072045	-0.050465
120,90,0	0.528733	18.015839	-4.805475	-15.825905	-0.081494	-0.050465
150,90,0	1.163213	19.572282	-6.736442	-15.862379	-0.064959	-0.058229
180,90,0	2.009186	21.829126	-6.517014	-15.862379	-0.069683	-0.058229
0,90,30	1.445204	20.895261	2.698965	-15.798548	-0.069683	-0.058229
0,90,60	0.387738	20.350505	1.952910	-15.816785	-0.046062	-0.065993
0,90,90	-0.0669729	20.428328	1.294626	-15.835023	-0.064959	-0.065993
0,90,120	-1.515702	21.128727	0.943541	-15.853262	-0.074407	-0.065993

0,90,150	-1.868191	22.218239	0.153600	-15.853262	-0.067321	-0.058229

6.4.3 Microcamera

To see the microstructure of the machined surface and get a clear vision of the process, a microcamera is applied **Figure 24**. It is mounted on the milling tool so that it can move with it all the time during the process.

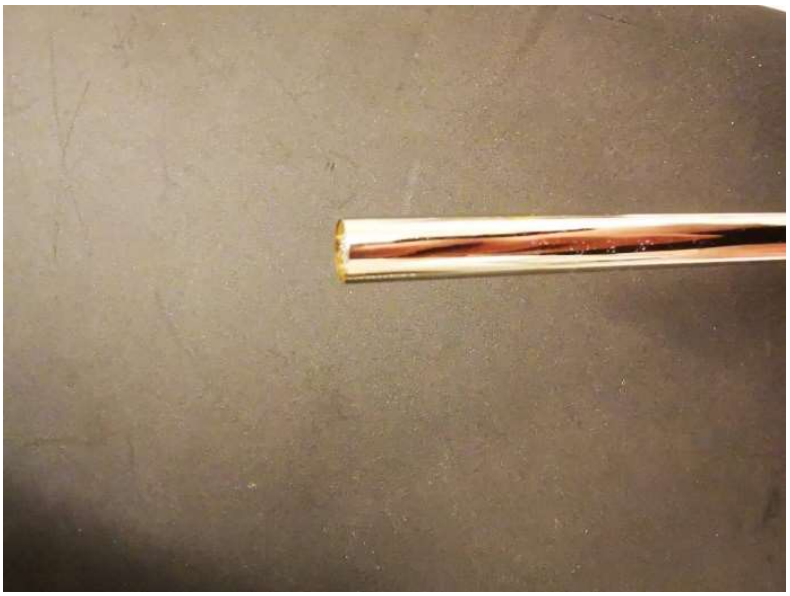


Figure 24: *The microcamera*

7. Data processing

During the process, the force sensor is always working by collecting forces and torques in x-, y-, and z-directions with a certain frequency which is 1000 Hz. Also, the Nikon metrology system is recording the positions of three LEDs with a frequency of 100 Hz. The position data from the Nikon metrology system and the force data from the force sensor should be processed to show the expected results.

7.1 Path accuracy error data processing

To simplify the descriptions, different features of a square are illustrated in **Figure 25**.

One of the goals is to know how large path accuracy errors could be during the process. The method is, for example, L1, using the air motion L1 y-coordinates minus the real cut ones. The Matlab code is attached in Appendix (C).

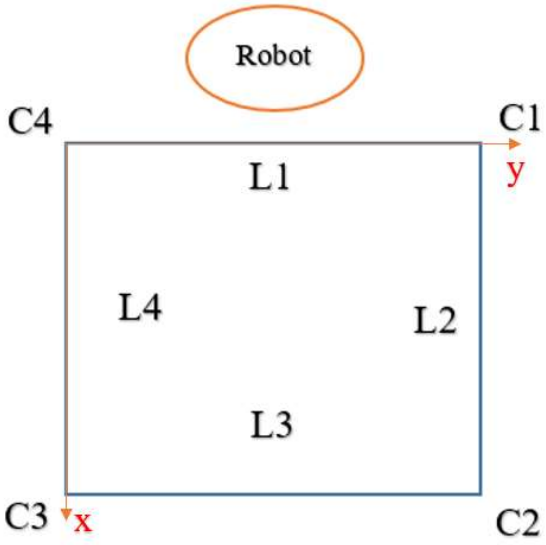


Figure 25: *Features of the square*

7.2 Tool frame orientation

As introduced before, three LEDs generate their coordinate system for themselves. The method of finding how large the tool frame coordinate system could be tilted during the process is to find the normal vector of each z-axis at the moment, and then calculate the angle between them.

For examole, vector z(t) = LED1(x1(t),y1(t),z1(t)) -LED2(x2(t),y2(t),z2(t)) and vector z(t-1) = LED1(x1(t-1),y1(t-1),z1(t-1)) - LED2(x2(t-1),y2(t-1),z2(t-1)). The angle between them is the tool tilted angle in this moment.

The Matlab code is attached in Appendix (D).

8. Results

In this section, the collected data will be analyzed. The trends of the path accuracy error, forces, and tool orientation are going to be interpreted combined with what has happened in the real milling process.

8.1 Milling with 1.8 cc/min and 1.44 cc/min MRR

8.1.1 Milling square line features with 1.8 cc/min MRR

When it is milling Line 1 with 1.8 cc/min MRR, we can see from **Figure 26** that the accuracy error ranges from -0.09 mm to 0.04 mm. The average value of the error is -0.03 mm.

In the beginning, it goes up to 0. It is because the tool mills from the point which is the one after plunging, which is not exactly lying on Line 1. So the tool tries to adjust itself from the effect of the plunging deviation. Therefore, the force varies in a big amplitude. The tool moves at a distance, around 0.05 mm in a sudden, which is the reason for the trend at the beginning.

Based on **Figure 27** we know that the force following is quite regular, ranging from 3 N to 55 N, which indicates that the process should remain at a regular vibration level. As a fact, we can see that the error ranges in a small distance, 0.02 mm. And the orientation of the milling tool stays at 0.01°, as **Figure 28** shows. Some larger forces make larger accuracy errors (**Figure 29**). The cause of the larger and smaller forces could be the local evenness of the workpiece or the evenness of the tool bit. But the most significant influence should be the changing feed speed. It is impossible to maintain the speed exactly at 4 mm/s. It could range to a small extent, 3.95 - 4.05 mm/s for instance. The source of the cause is the stiffness of the robot in such a position, which means how much force can the joints resist without deformation. There is a slight difference in the stiffness between a position to position, causing different force-based deviations.

The big error both in path accuracy and tool orientation, at last, is caused by the changing feed direction. It is moving to the first corner, so the feed direction is going to be changed. As a result, a little more material is going to be removed. which increases the force.

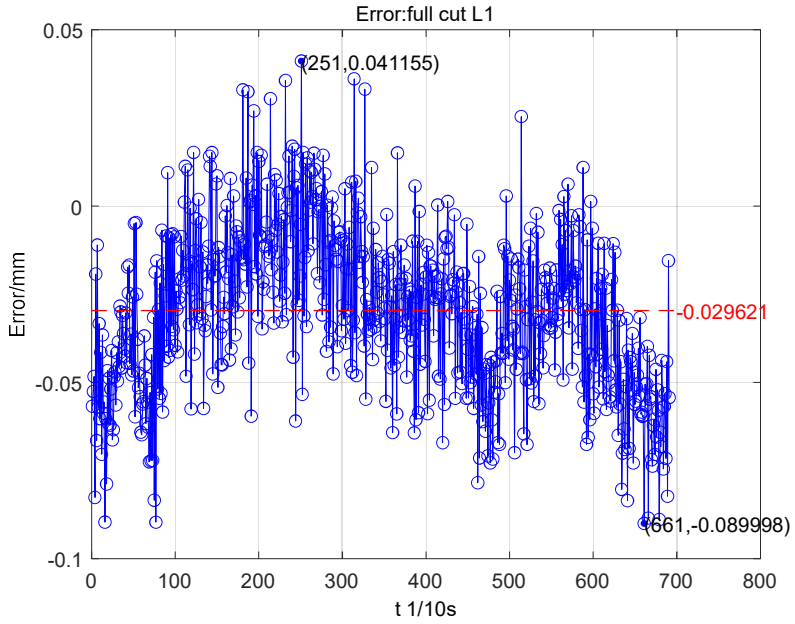


Figure 26: *Path accuracy error of Line 1 (MRR=1.8cc/min)*

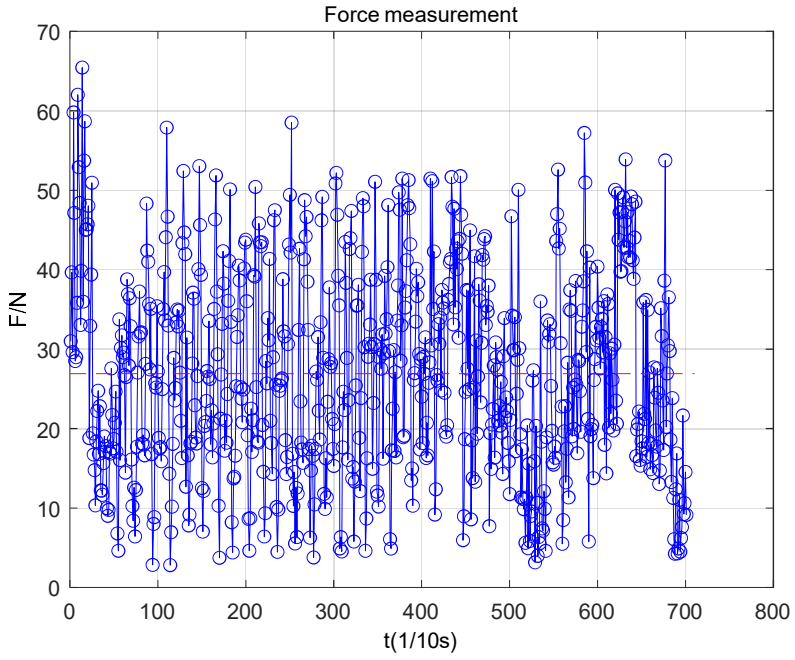


Figure 27: *Resultant force of Line 1(MRR=1.8cc/min)*

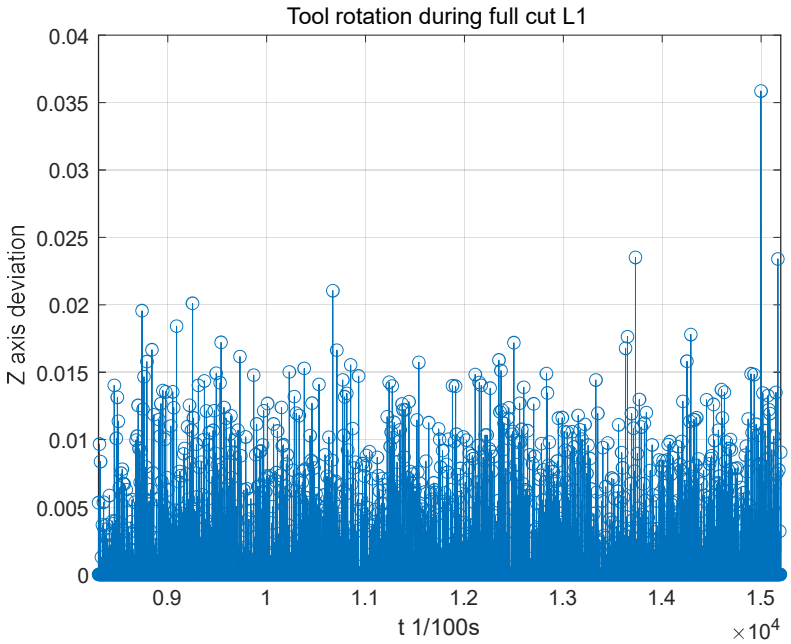


Figure 28: *Tool orientation of Line 1 (MRR=1.8cc/min)*

Figure 29: *Local path of Line 1(MRR=1.8cc/min) (Red-air motion; blue-real cut)*

When it comes to Line 2, the average path accuracy error becomes -0.89 mm, as shown by **Figure 30** which is almost 30 times Line 1's. According to the data collected, the forces caused during the processing Line 2 are similar to Line 1. So that the difference indicates the stiffness of the robot in the y-direction is much less than in the x-direction. A low stiffness always causes poor process performance. It would explain why there are more milling trace marks on Line 2 than on Line 1, as **Figure 31** shows.

If we have a look at **Figure 33**, it is clear that two deviated angles appear (as **Figure 32** shows). It happens also because of the low stiffness. When a force is large enough to make the robot joints have a slight elastic deformation, the joints would keep staying at such a position until the servo motor gives it a torque to resist the force. And then the robot can move again along the trajectory that has been programmed already. However, due to the time delay, the robot will move straight to the later position i.e. jump over a distance without programmed speed. So, a speed vacuum appears. In this experiment, the time delay is extremely short, less than 1/100 seconds.

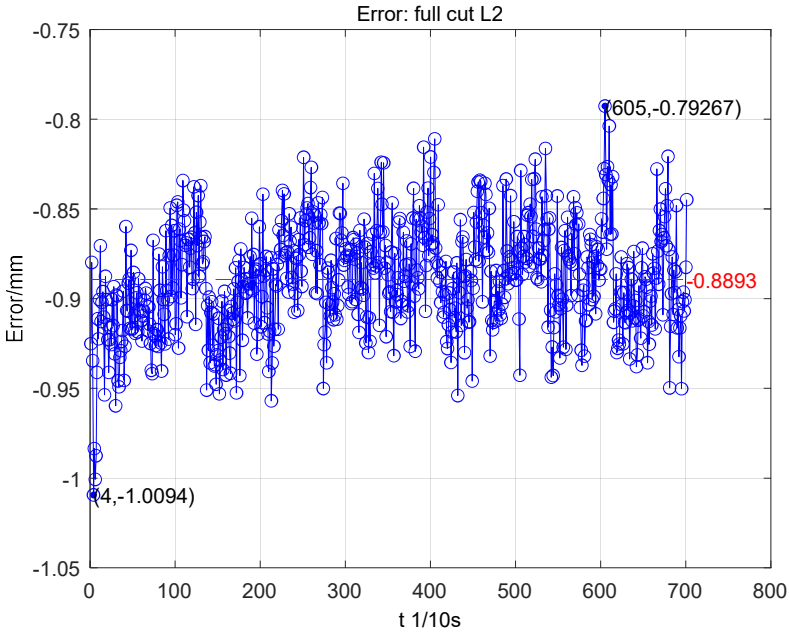


Figure 30: *Path accuracy error of Line 2 (MRR=1.8cc/min)*

Figure 31: *Milling traces on Line 2*

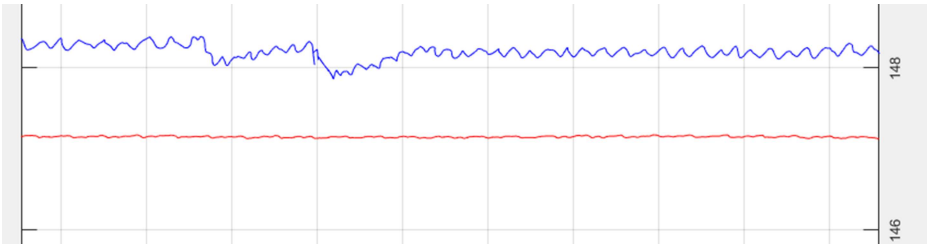


Figure 32: *Local path of Line 2(MRR=1.8cc/min) (Red-air motion; blue-real cut)*

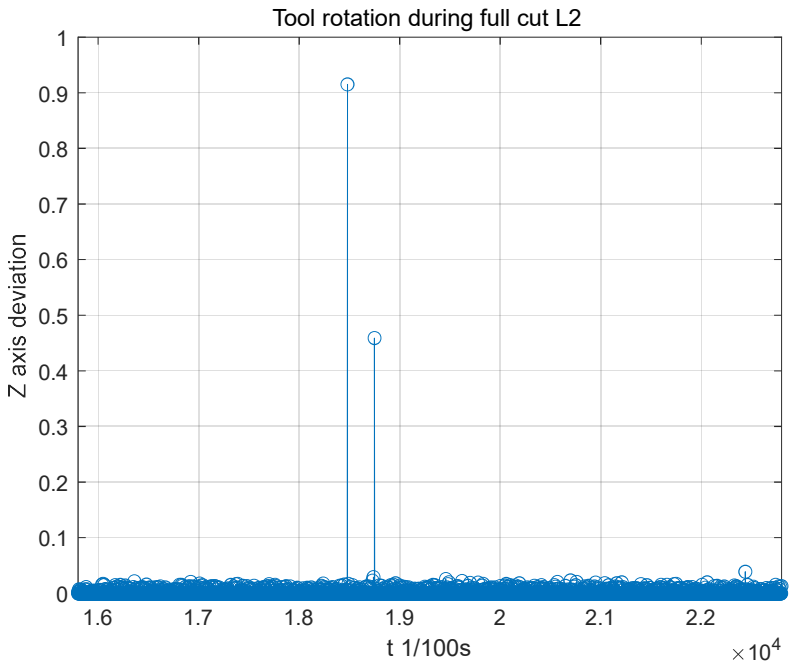


Figure 33: *Tool orientation of Line 2 (MRR=1.8cc/min)*

The path errors and forces of Line 3 are contrary to Line 1. The average path error is close to 0.05 mm and the maximum error is about 0.18 mm according to Figure 34. Comparing **Figure 26** and **Figure 34**, we know that the robot has weaker resistance to +x-direction force along Line 1 but a stronger one along Line 3. So we assume that after the robot reaches a certain position, the stiffness of the joints to resist +x-direction force becomes smaller. But before this position, the robot has a higher value for such stiffness. It means the further distance the TCP goes, the more likely it tends to return to its best performance working area, and vice versa. The assumption will be confirmed in later sections.

There are more speed vacuum phenomena that happen during milling Line 3 according to **Figure 35** and **Figure 36**. Besides the relatively low stiffness in +x-direction, the wear of the tool and the measurement system errors are also reasons for it.

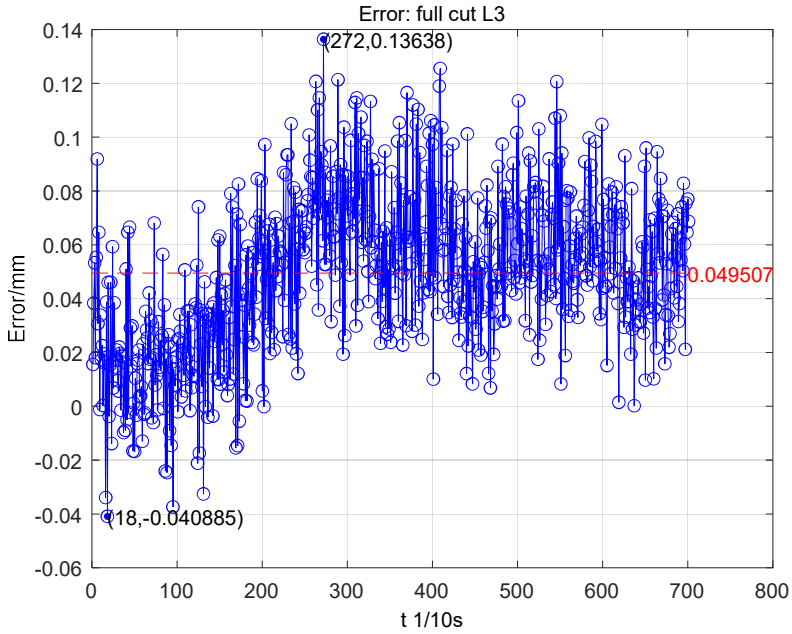


Figure 34: *Path accuracy error of Line 3 (MRR=1.8cc/min)*

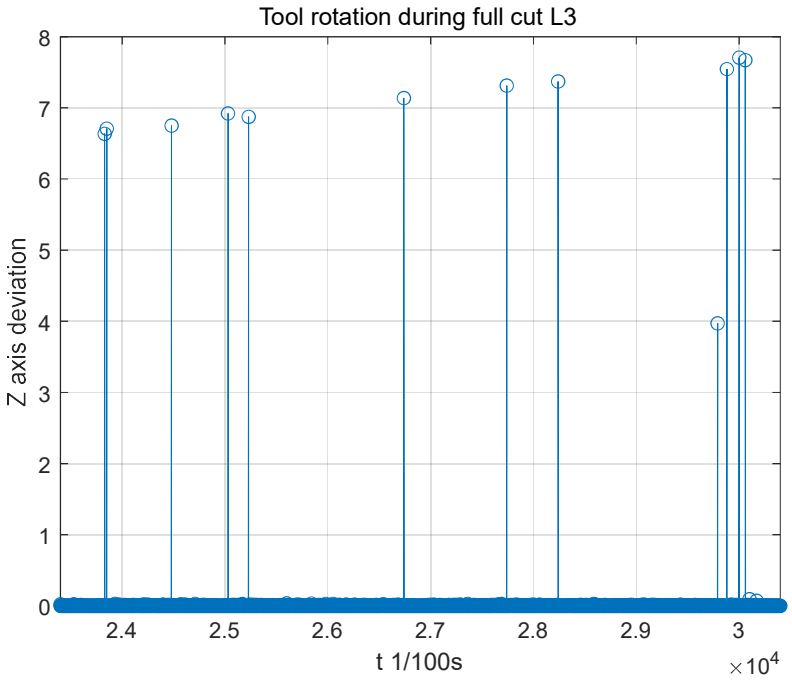


Figure 35: *Tool orientation of Line3 (MRR=1.8cc/min)*

Figure 36*: Local path of Line 3(MRR=1.8cc/min) (Red-air motion; blue-real cut)*

The performance of Line 4 is very like Line 2, according to Figure 37 and **Figure 38**. It indicates that in such a working area and certain postures, the robot milling performance is steady in both

±x-directions. However, the deviation value of the process in the y-direction is still about 20 times larger than that in the x-direction.

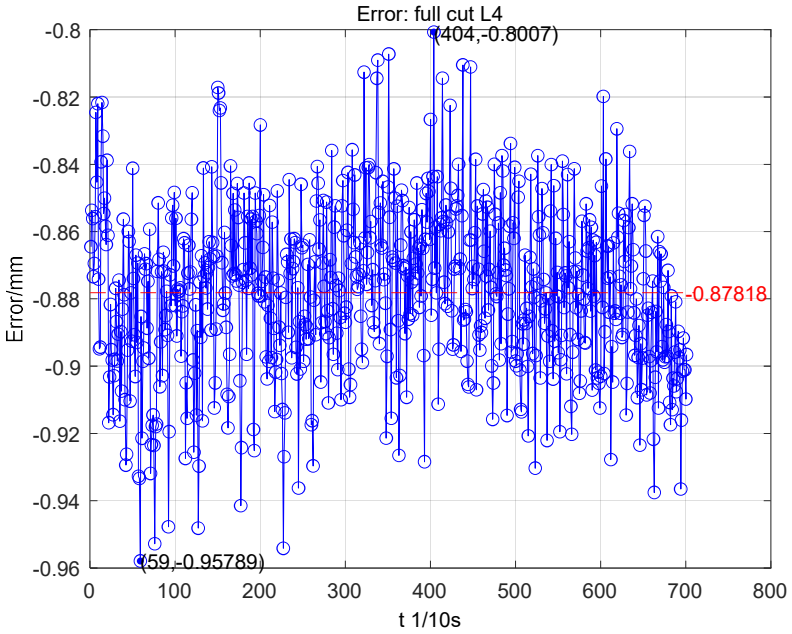


Figure 37: *Path accuracy error of Line 4(MRR=1.8cc/min)*

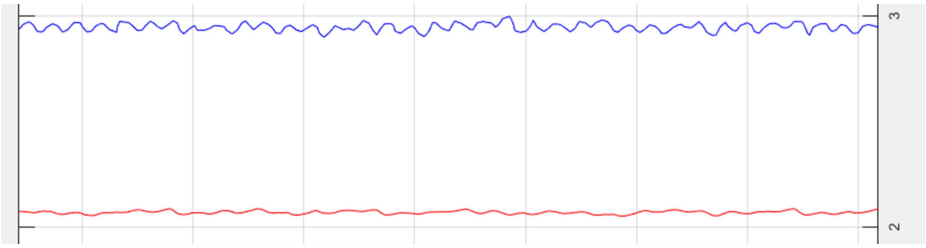


Figure 38: *Local path of Line 4(MRR=1.8cc/min) (Red-air motion; blue-real cut)*

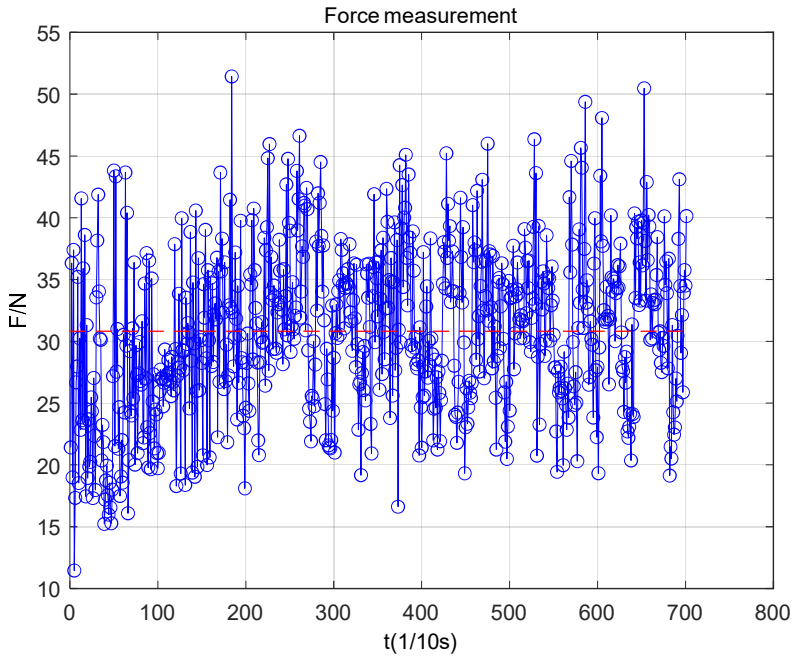


Figure 39: *Resultant force of Line 4(MRR=1.8cc/min)*

8.1.2 Milling square line features with 1.44 cc/min MRR

As for a smaller MRR process, path accuracy, forces, and tool orientation will be discussed in this subsection. The results will be compared with the same feature of 1.8 cc/min MRR.

The path accuracy error of Line 1 with 1.44 cc/min MRR is around -0.05 mm based on **Figure 40** and **Figure 41**, which is 0.03 mm larger than that with 1.8 cc/min MRR. Many errors that are larger than 0.15 mm happen after 25s. The reason behind them is the uneven mill in the first milling process. A varying width thickness i.e. cutting width has been generated by the previous cut. As a result, a changing amount of material is being milled in such circumstances, resulting in vibrations that are out of normal amplitude.

What should be noticed is the force generated during the process, as **Figure 42** shows. The force is even 5N larger than that of 1.8 cc/min MRR. One of the reasons that cause it is the unevenness of the force distribution, which means only 20% of the tool diameter is involved. In addition to the slight vibration, larger forces than we expected would impact the tool.

Since the force here does not have a big difference, the tool orientation should be similar to the one of 1.8 cc/min MRR, which is confirmed by **Figure 43**.

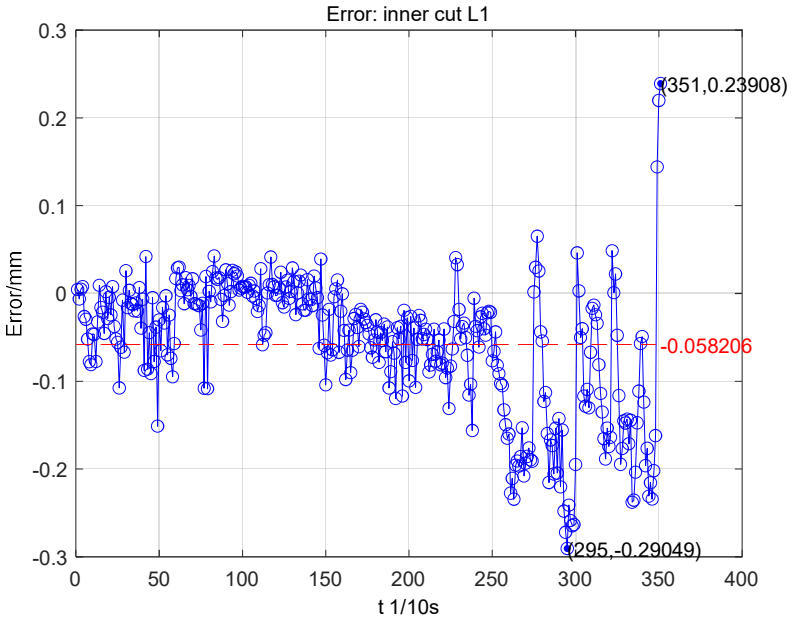


Figure 40: *Path accuracy error of Line 1(MRR=1.44cc/min)*

Figure 41: *Local path of Line 1(MRR=1.44cc/min) (Red-air motion; blue-real cut)*

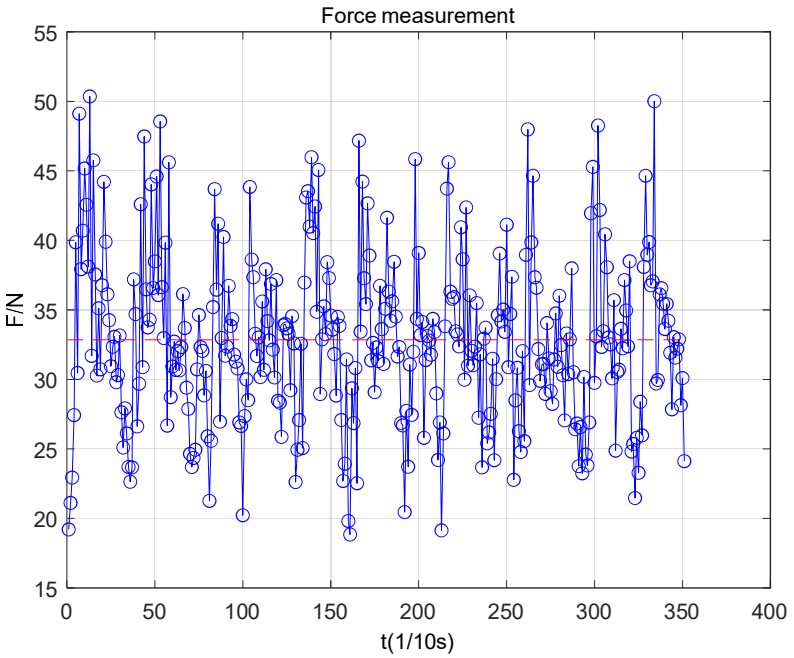


Figure 42: *Resultant force of Line 1(MRR=1.44 cc/min)*

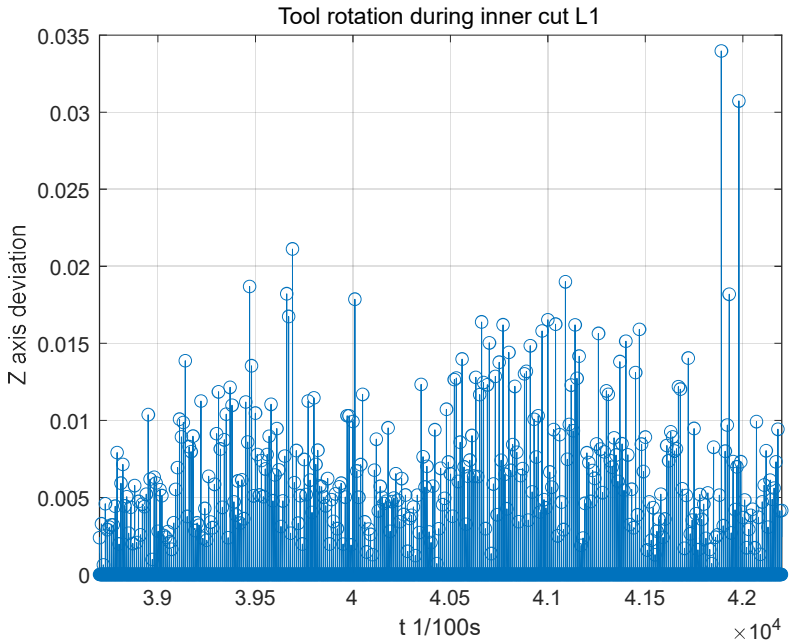


Figure 43: *Tool orientation of Line1 (MRR=1.44cc/min)*

As has been expected, the process performance of Line 2 with 1.44 cc/min MRR is worse than that with 1.8 cc/min MRR (**Figure 44** and **Figure 45**). Besides the relatively low stiffness on this line, the machined uneven features make it worse.

Some speed vacuum also happens during this part of the process (**Figure 46**). And what it has caused on the path accuracy can be seen in **Figure 45** as an example.

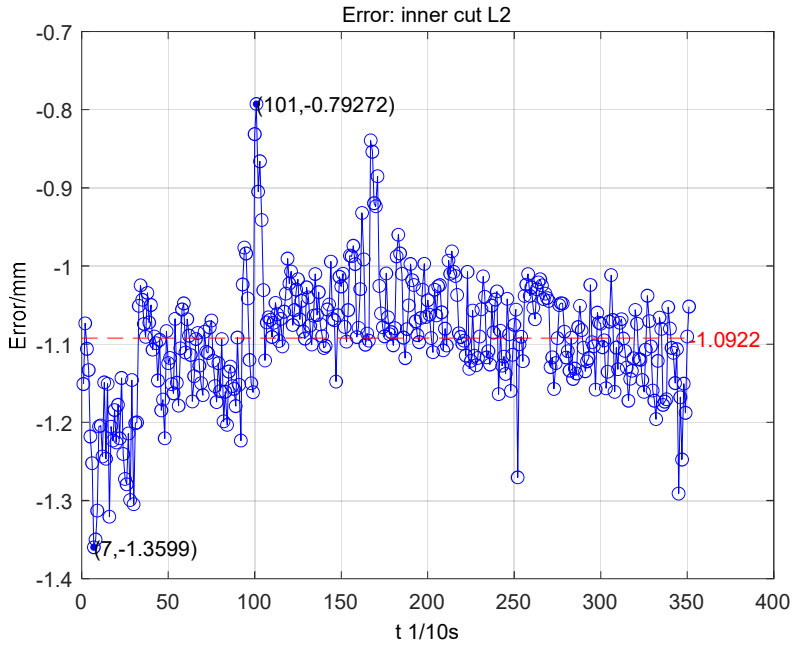


Figure 44: *Path accuracy error of Line 2(MRR=1.44cc/min)*

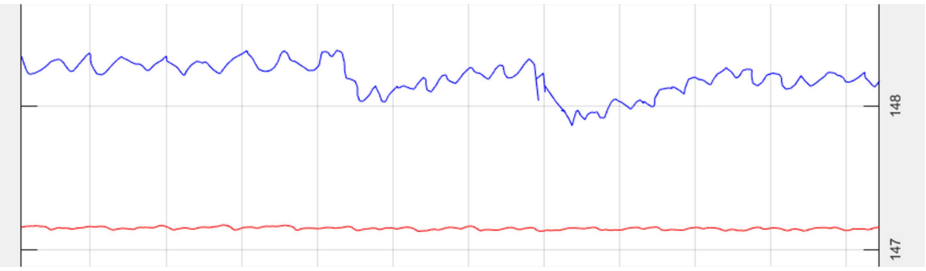


Figure 45: *Local path of Line 2(MRR=1.44cc/min) (Red-air motion; blue-real cut)*

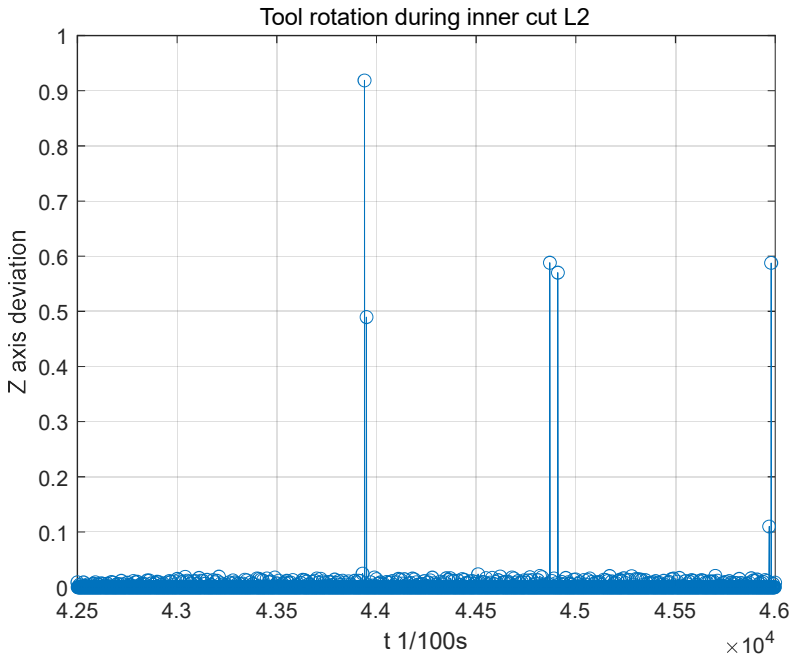


Figure 46: *Tool orientation of Line2 (MRR=1.44cc/min)*

When the process went to Line 3 with 1.44 cc/min MRR, the milling tool finally could not endure the force and the high temperature any longer. The cutting edges were broken during the process. **Figure 47** can tell when and where the broken point is. After it entered Line 3 6.8s, it had a sudden wide vibration and lost its milling function. Figure 48 shows how the path was affected by it. More uncontrollable vibration happened after the broken point as shown in **Figure 49**.

Based on the data from Figure 40 and Figure 47, the assumption is further confirmed: The further distance the TCP goes, the more likely it tends to return to its comfortable working area, and vice versa.

Figure 50 to **Figure 54** are showing the data of Line 4 with 1.44 cc/min MRR. The analysis of the features in the figures has been mentioned before as Line 4 with 1.8 cc/min MRR.

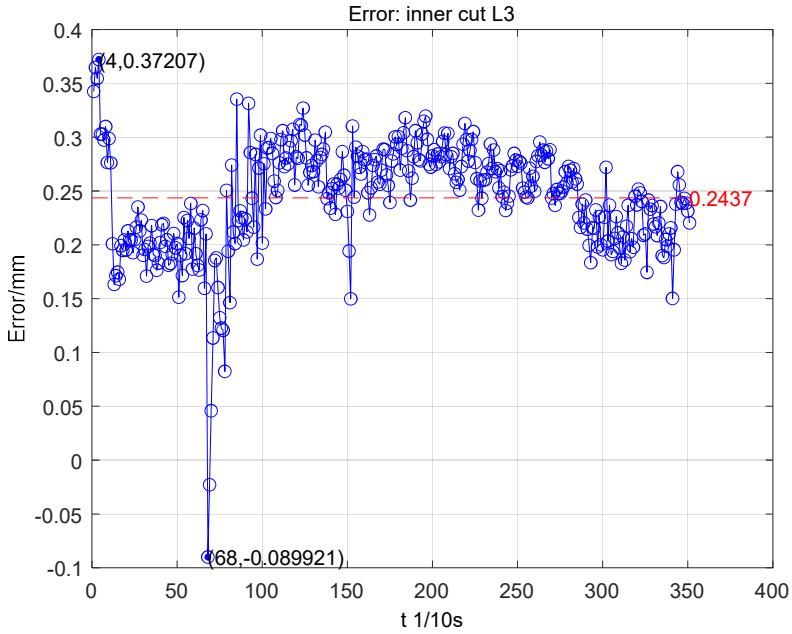


Figure 47: *Path accuracy error of Line 3(MRR=1.44cc/min)*

Figure 48: *The broken point on Line 3 (1.44 cc/min MRR) (Red-air motion; blue-real cut)*

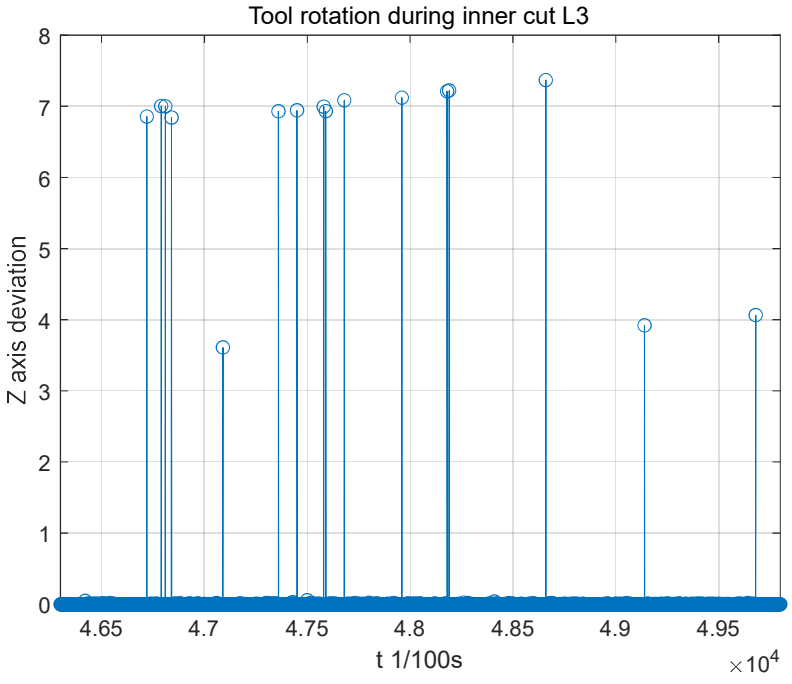


Figure 49: *Tool orientation of Line3 (MRR=1.44cc/min)*

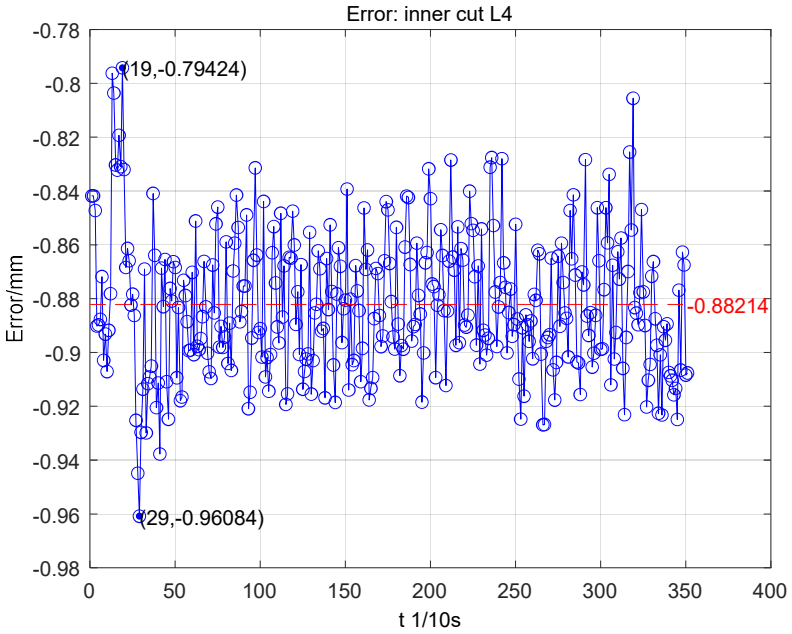


Figure 50: *Path accuracy error of Line 4(MRR=1.44cc/min)*

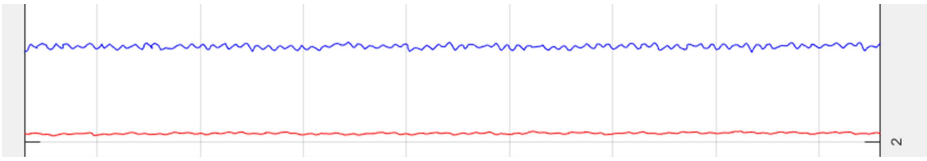


Figure 51: *Local path of Line 4(MRR=1.44cc/min) (Red-air motion; blue-real cut)*

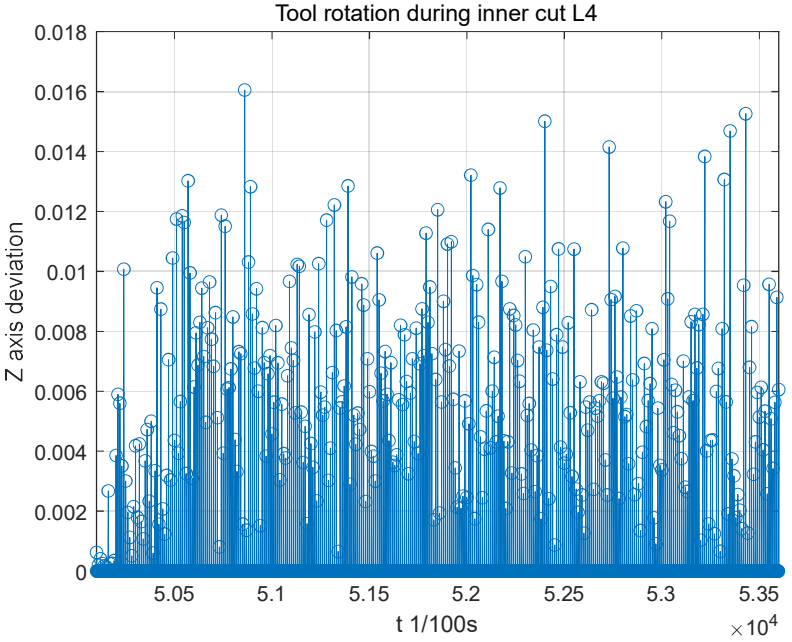


Figure 52: *Tool orientation of Line4 (MRR=1.44cc/min)*

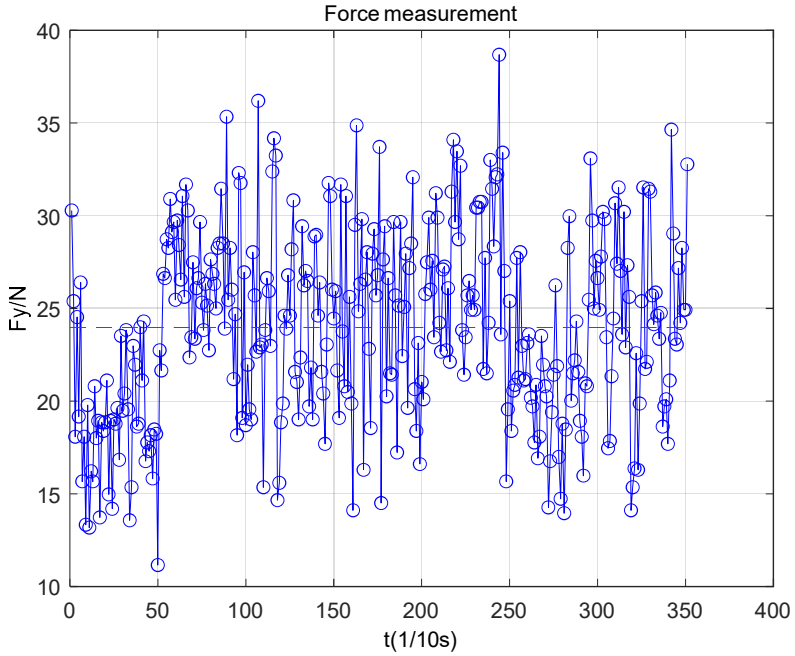


Figure 53: *Y-direction force of Line 4(MRR=1.44 cc/min)*

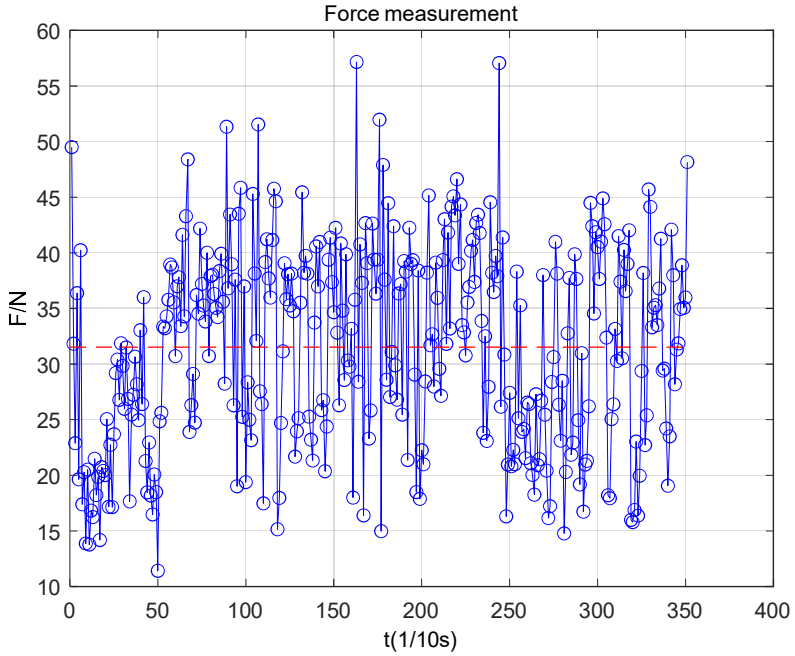


Figure 54: *Resultant force of Line 4(MRR=1.44 cc/min)*

8.1.3 Milling square corner features

As for the corner feature, the feed rate direction keeps changing, resulting in changing force directions. Meanwhile, due to the difference in path accuracy between x-direction and y-direction, the amount of material is increasing during C1 and C3 or it is decreasing during C2 and C4. This means a varying amplitude force is going to be generated when milling the

corners. **Figure 55** illustrates the path of C1 with 1.8 cc/min MRR and 1.44 cc/min MRR. We can see that the path error becomes larger and larger during C1.

The force of the 1.8 cc/min MRR one varies a lot from 5N to 55N and keeps this trend until C1 is finished, as **Figure 56** shows. As has been expected, the force of 1.44 cc/min MRR is quite close to the one of 1.8 cc/min MRR, but its amplitude of it is almost half. It is because only 20 percent of the tool is involved and less material is being milled.

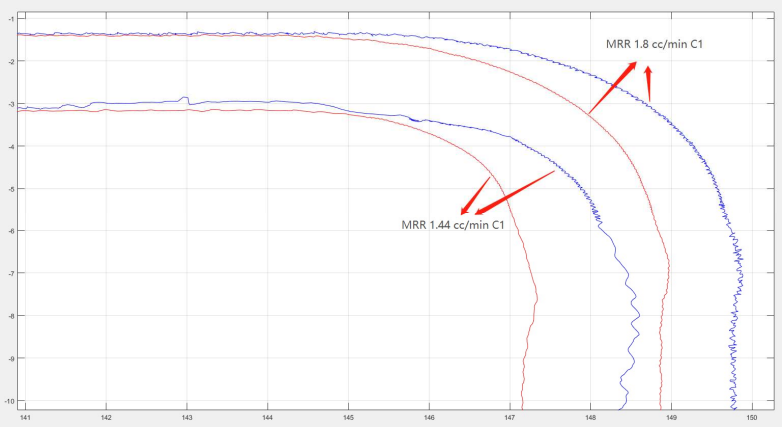


Figure 55: *Path accuracy error of C1 (MRR 1.8 cc/min and MRR 1.44 cc/min)(Red-air motion; Blue-real cut)*

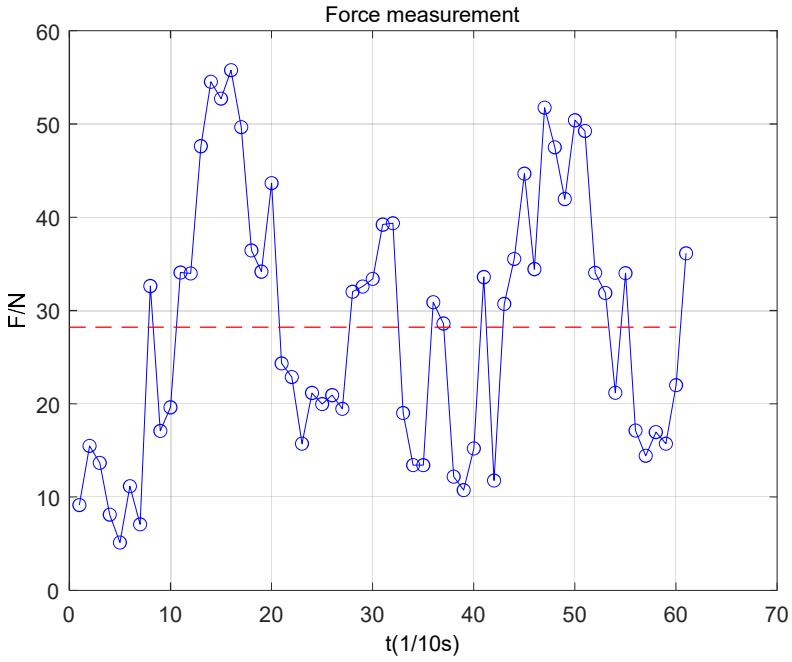


Figure 56: *Resultant force of C1 (1.8 cc/min MRR)*

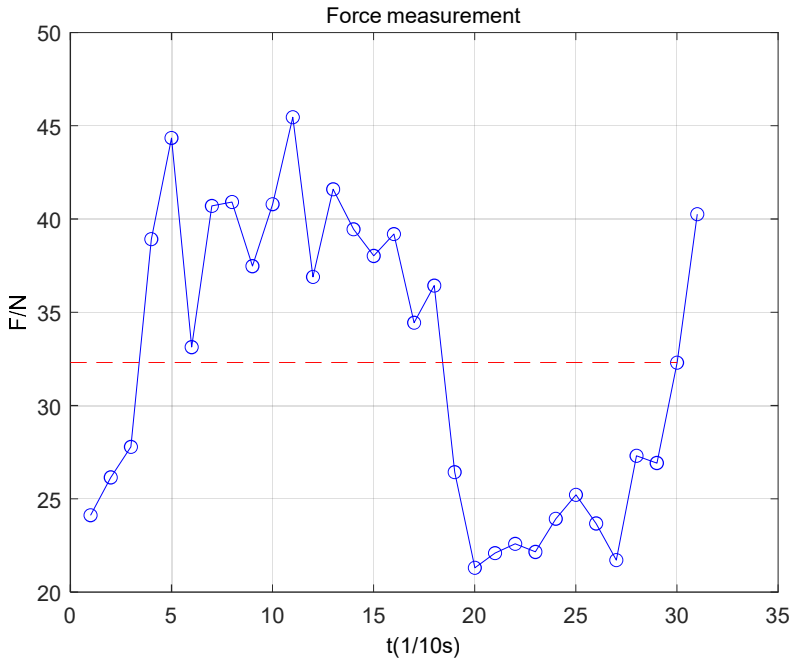


Figure 57: *Resultant force of C1 (1.44 cc/min MRR)*

The path accuracy error of C2 has a contrary trend to C1's if we look at **Figure 58**. But the force of 1.8 cc/min becomes smaller, see **Figure 59**. The amplitude decreases to 30N and the largest force reduces to 45N, which is 15N smaller than C1's. It is due to the process being from the weaker stiffness side to the stronger one, thus there is more and more resistance to prevent the path from deviation. Therefore, less force generated by the vibration. On the contrary, the resistance to prevent the path from deviation becomes less and less during milling the C1, thus the force is naturally larger. If comparing **Figure 60** and **Figure 57**, it is not hard to find out the force also decreases, though the difference is not as clear as C1's.

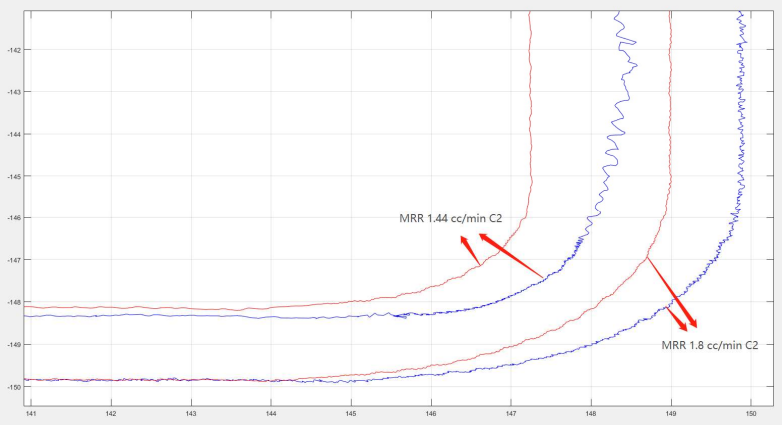


Figure 58: *Path accuracy error of C2 (MRR 1.8 cc/min and MRR 1.44 cc/min)(Red-air motion; Blue-real cut)*

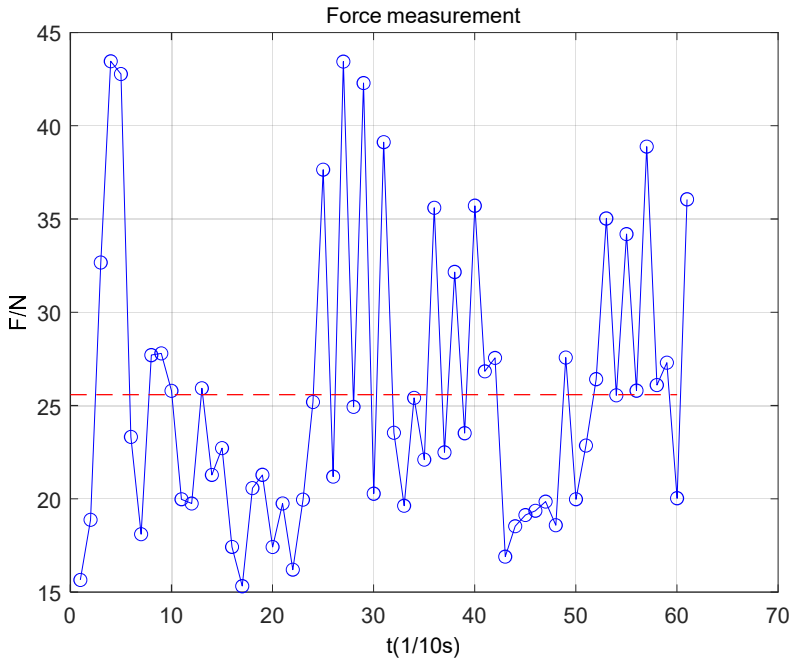


Figure 59: *Resultant force of C2 (1.8 cc/min MRR)*

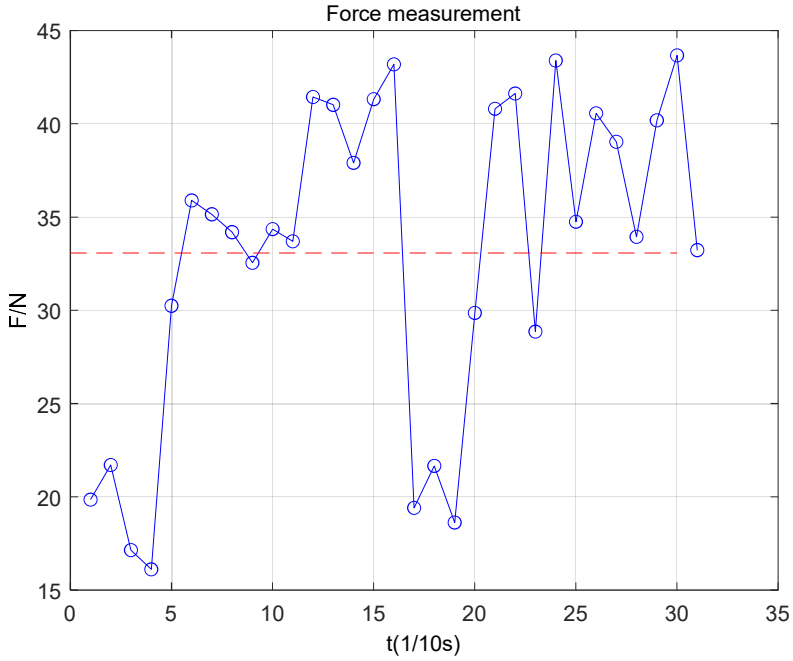


Figure 60: *Resultant force of C2 (1.44 cc/min MRR)*

Figure 61 to **Figure 66** illustrate the path accuracy and the force for C3 and C4 features. Their values and trends of them are similar to C1 and C2.

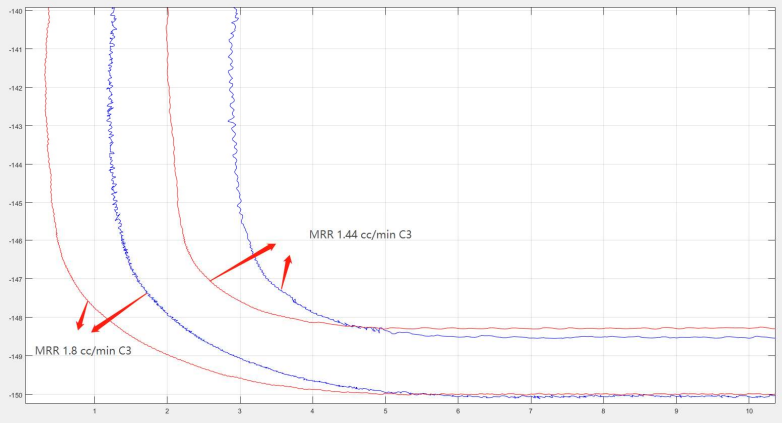


Figure 61: *Path accuracy error of C2 (MRR 1.8 cc/min and MRR 1.44 cc/min)(Red-air motion; Blue-real cut)*

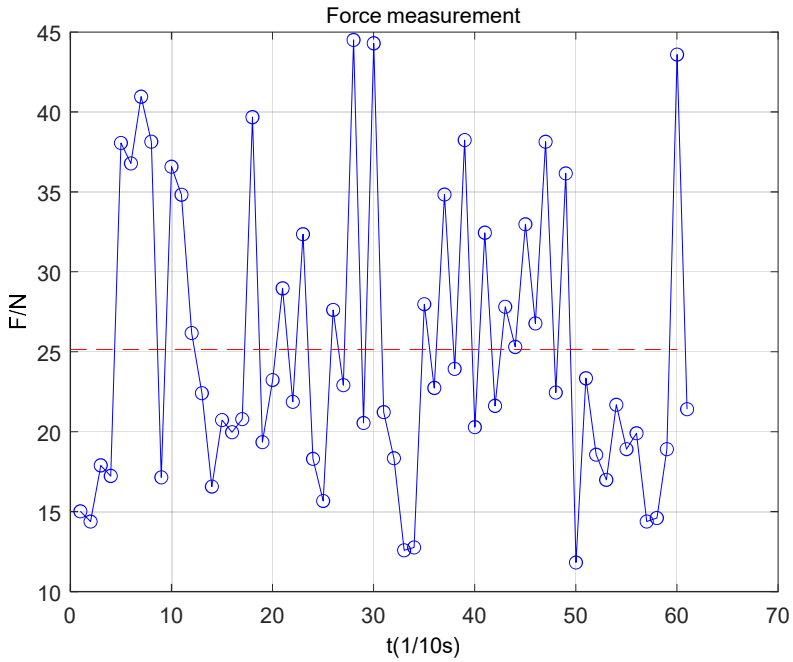


Figure 62: *Resultant force of C3 (1.8 cc/min MRR)*

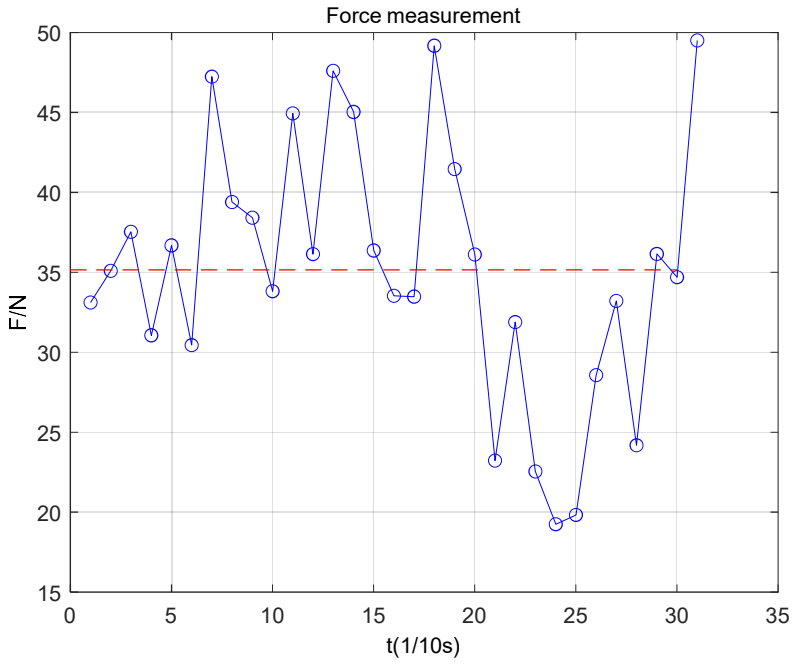


Figure 63: *Resultant force of C3 (1.44 cc/min MRR)*

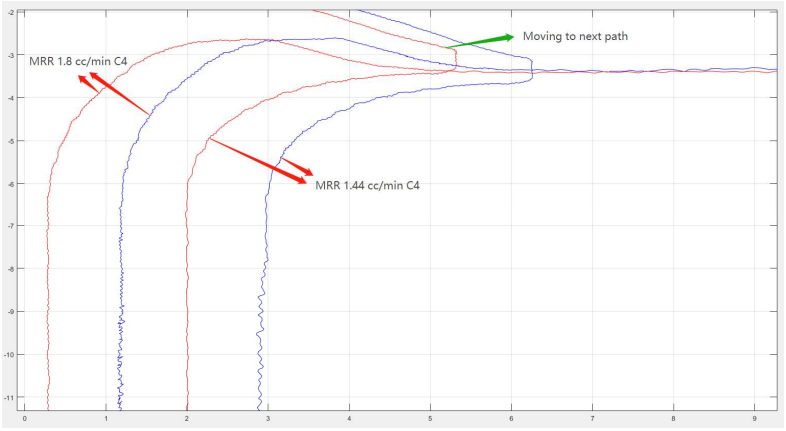


Figure 64: *Figure 50: Path accuracy error of C4 (MRR 1.8 cc/min and MRR 1.44 cc/min)(Red-air motion; Blue-real cut)*

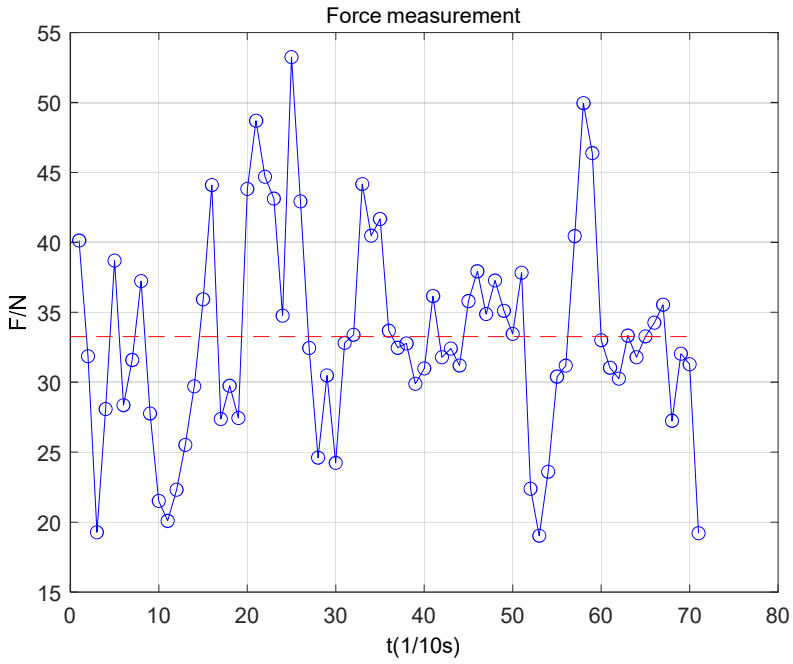


Figure 65: Resultant force of C4 (1.8 cc/min MRR)

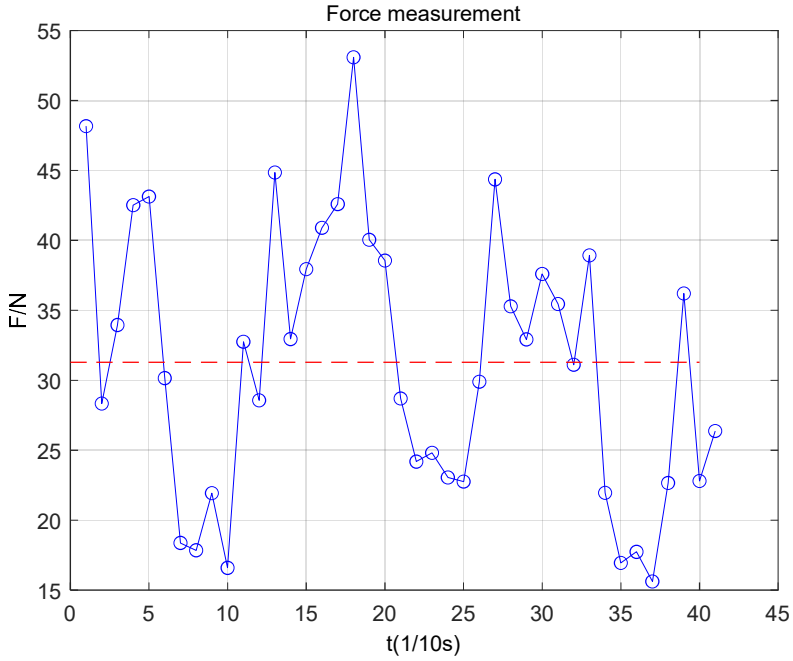


Figure 66: *Resultant force of C4 (1.44 cc/min MRR)*

8.2 Milling with 0.48 cc/min MRR

In this section, the performance of the process is illustrated in three locations with the proper process data(0.48 cc/min).

8.2.1 Location 1

When the speed rate and cutting depth become smaller, we can see from **Figure 67**, **Figure 68**, and **Figure 69** that the accuracy becomes much higher. The errors are in the range of 0-0.04mm on Line 1 and 3, and of 0-0.1mm on Line 2 and 4. The reason has been discussed in section 10.1. These data prove that the stiffness on the x-axis is stronger than that on the y-axis.

The errors of partial cut (**Figure 69**) are very similar to that of full cut (**Figure 68**). As a fact, if we have a look at **Figure 70**, **Figure 71**, **Figure 72**, and **Figure 73**, we know that the forces and the force-based tool vibration of full cut and partial cut are close. Though some speed vacuum phenomena happened on Line 3 and 4, the amplitudes of them are all smaller than 1.5º, which are light enough to be ignored. In general, the tool vibration keeps at the average value of 0.002º. And the forces of lines keep in the range of 19N to 21N on average.

From these data, we can conclude that when the proper process parameters are applied, the accuracy errors become much smaller and even. Besides, the forces keep low and regular so that the tool vibrates lighter.

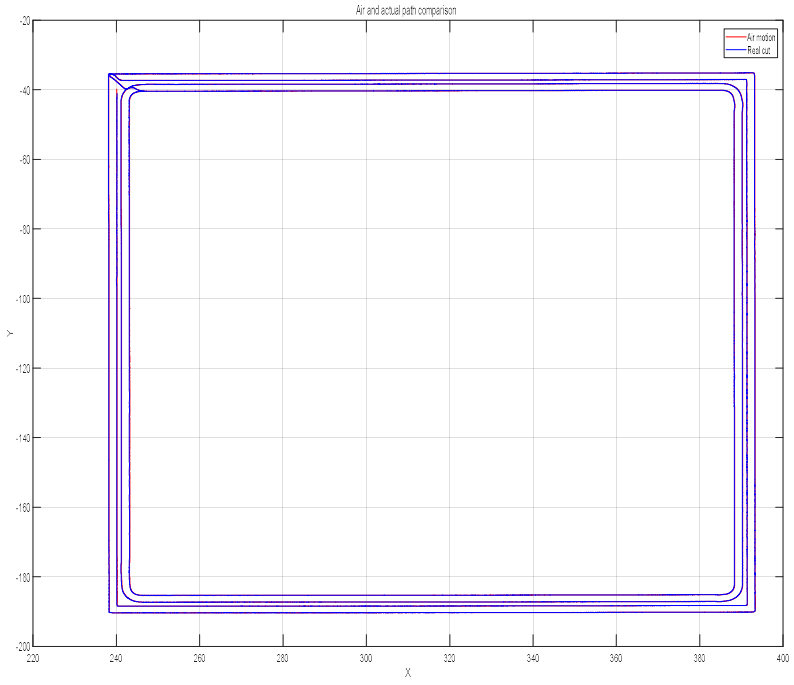


Figure 67: *Air motion and real cut paths (Location 1)*

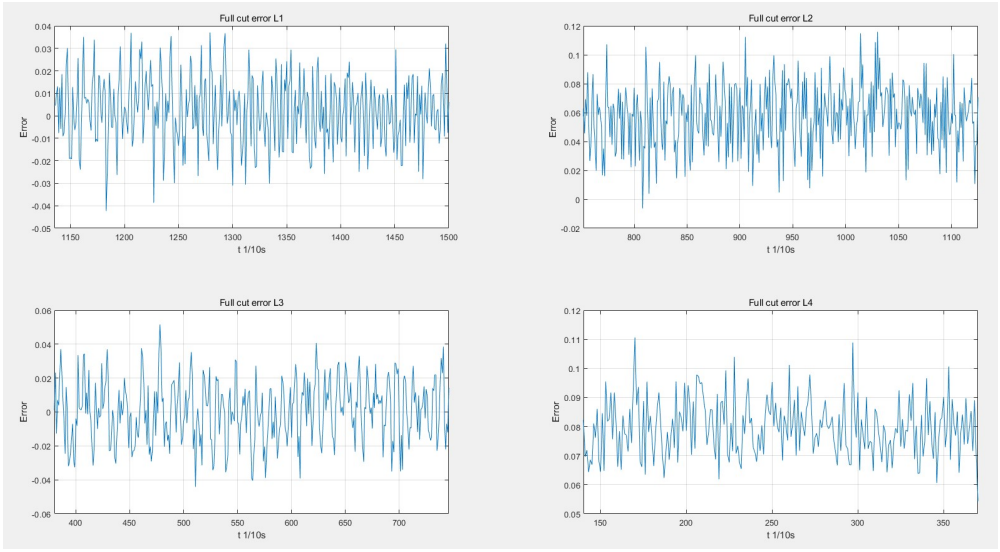


Figure 68: *Accuracy errors of Lines (Location 1, a_e=5mm)*

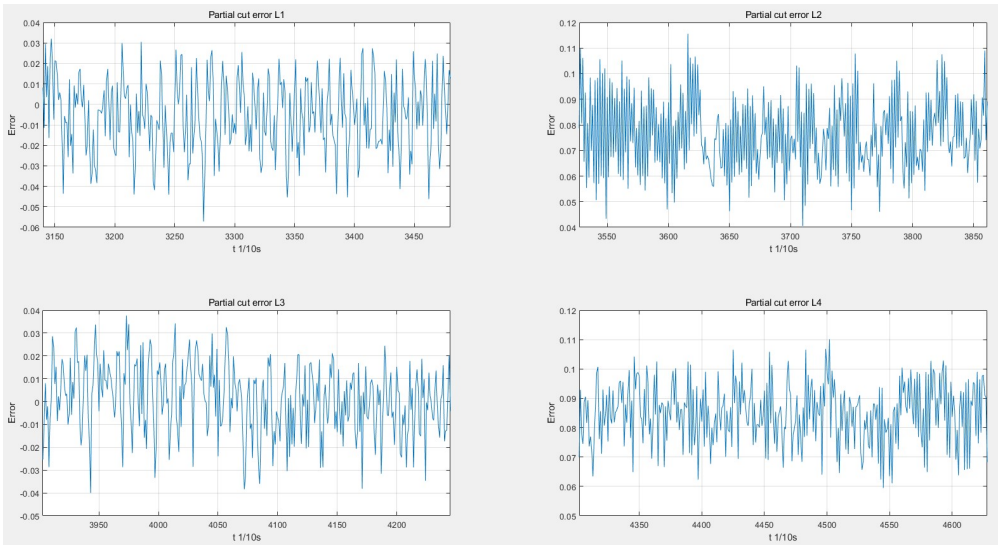


Figure 69: *Accuracy errors of Lines (Location 1, a_e=2mm)*

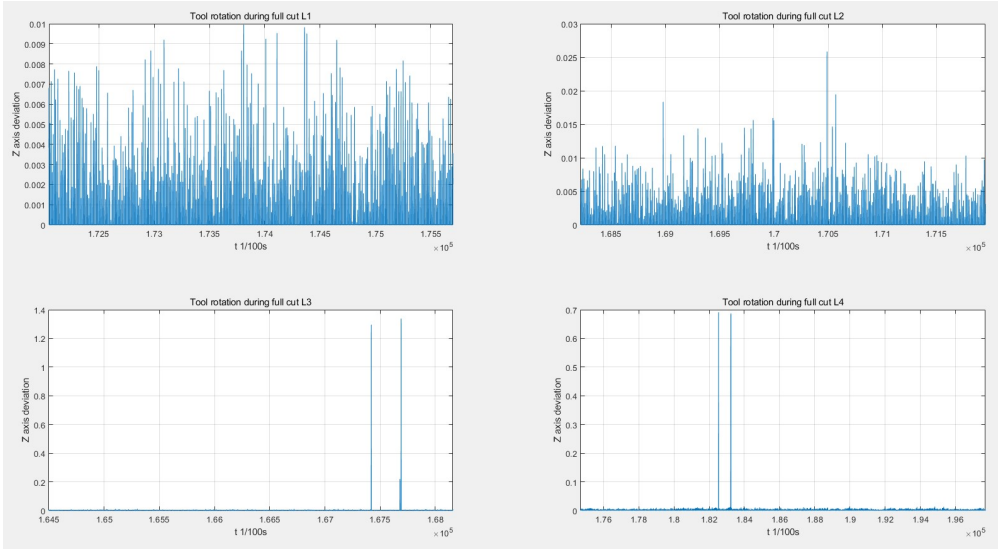


Figure 70: *Tool rotation during the full cut(Location 1, a_e=5mm)*

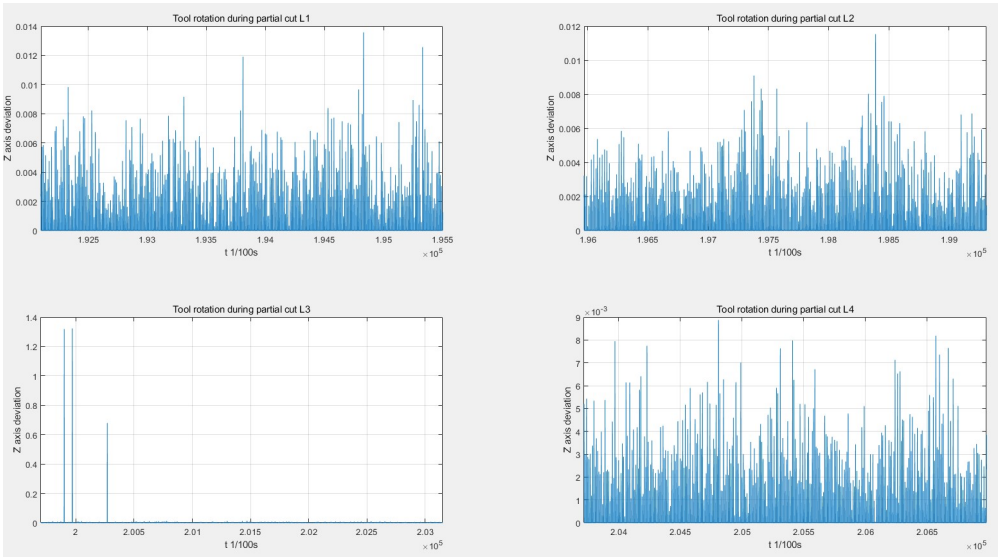


Figure 71: *Tool rotation during the full cut(Location 1, a_e=2mm)*

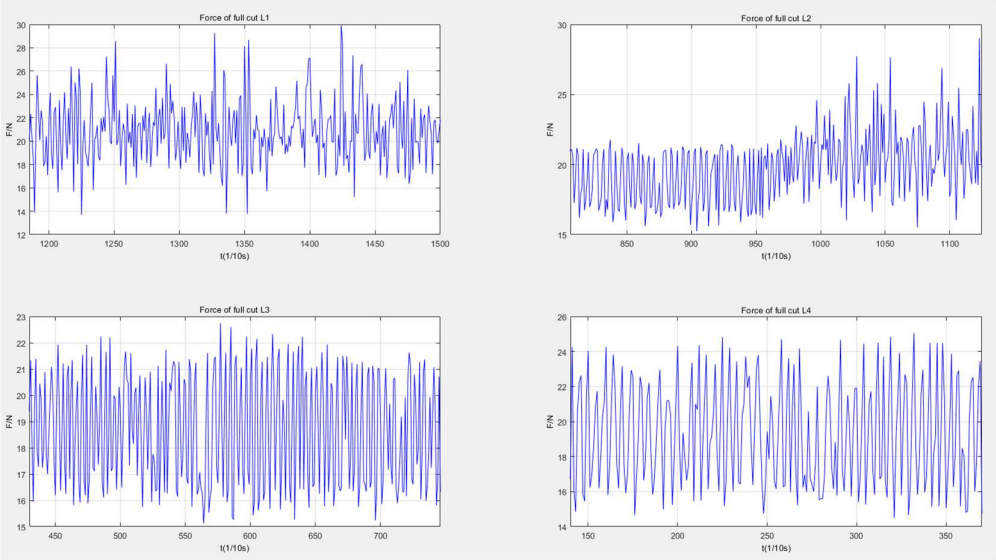


Figure 72: *The resultant force of lines (Location 1, a_e=5mm)*

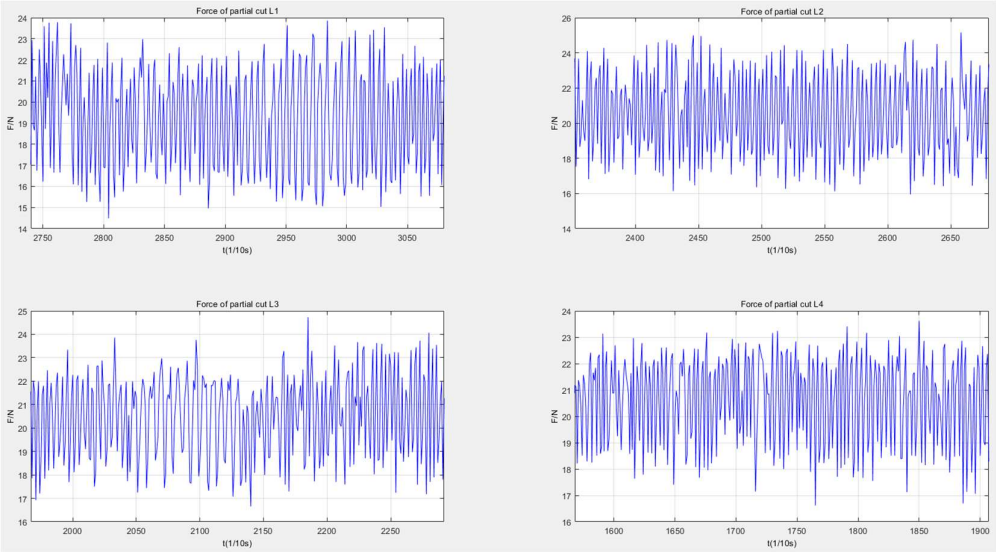


Figure 73: *The resultant force of lines (Location 1, a_e=2mm)*

8.2.2 Location 2

As for the milling process in Location 2, the accuracy trend (**Figure 74** to **Figure 76**) is quite different from that in Location 1. For Line 1, the error varies from 0.06mm maximum to -0.04mm minimum. The amplitude of the error is about 1mm, so as the other lines. The most likely reason behind this is the changed posture. As mentioned before, when the posture is changed i.e. the joints' rotation angles are changed, the stiffness of the robot also changes. At a posture that has less resistance to the forces coming from the x or y direction, the tool will follow the force direction, which causes the trend in **Figure 75** and **Figure 76**.

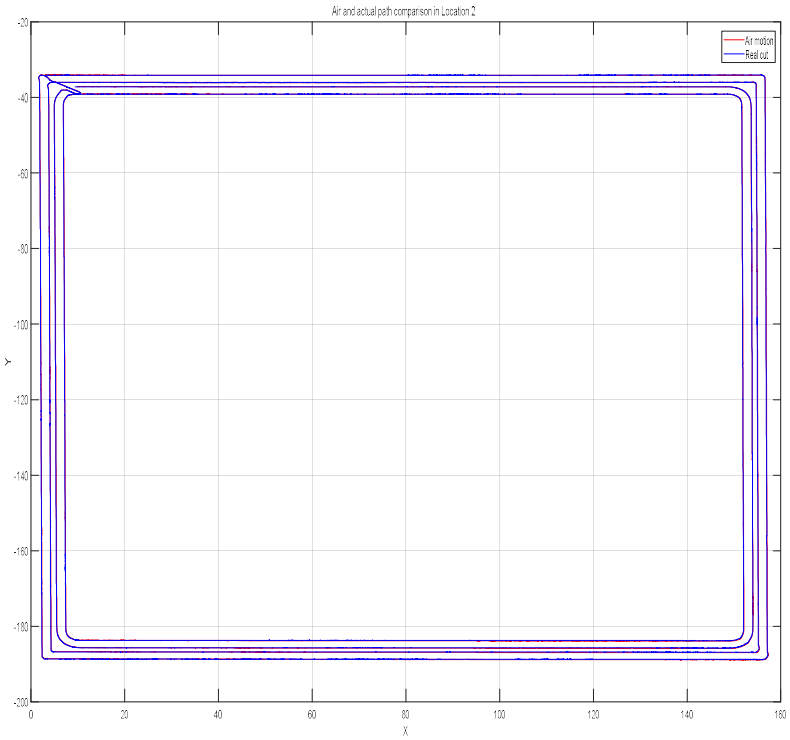


Figure 74: *Air motion and real cut paths (Location 2)*

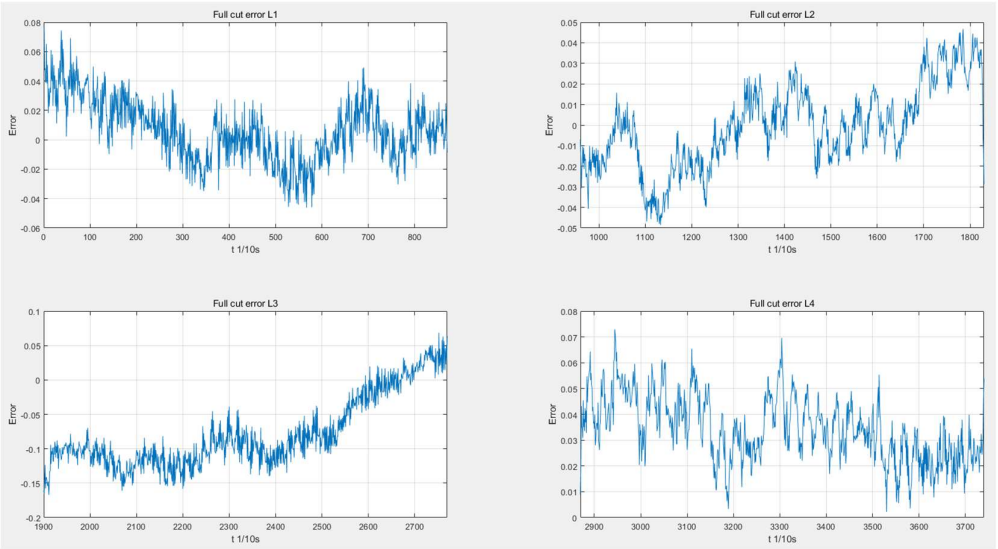


Figure 75: *Accuracy errors of Lines (Location 2, a_e=5mm)*

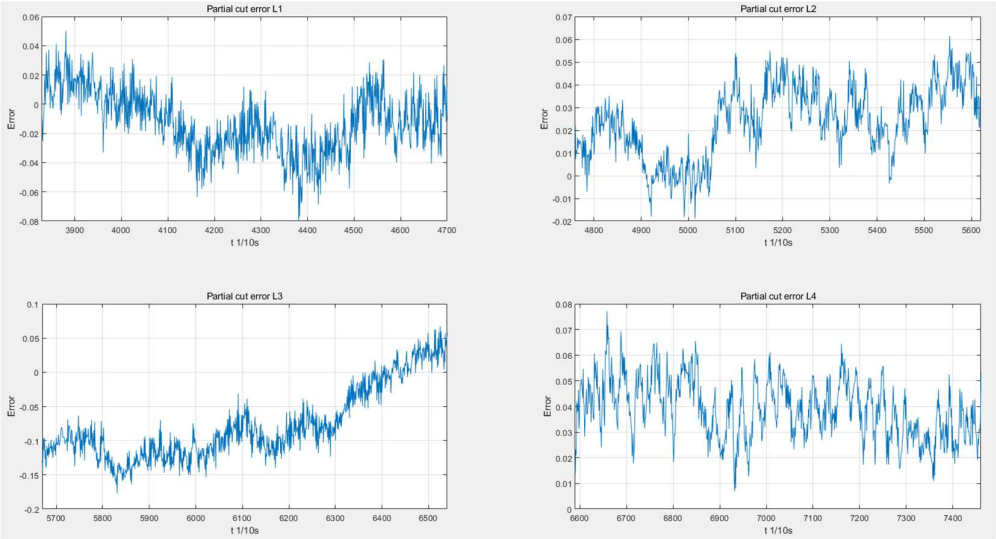


Figure 76: *Accuracy errors of Lines (Location 2, a_e=2mm)*

The average value of tool vibration is around 0.0002º, as shown in **Figure 77** and **Figure 78**. Although there are speed vacuum phenomena on Line 1, they do not exceed 0.008º during the full cut process (a_e=5mm). However, during the partial cut (a_e=2mm), the vibration is clearly larger. It is because the tool has space to vibrate after the full cut. Unlike the full cut process, the tool bit is completely enclosed by aluminum material.

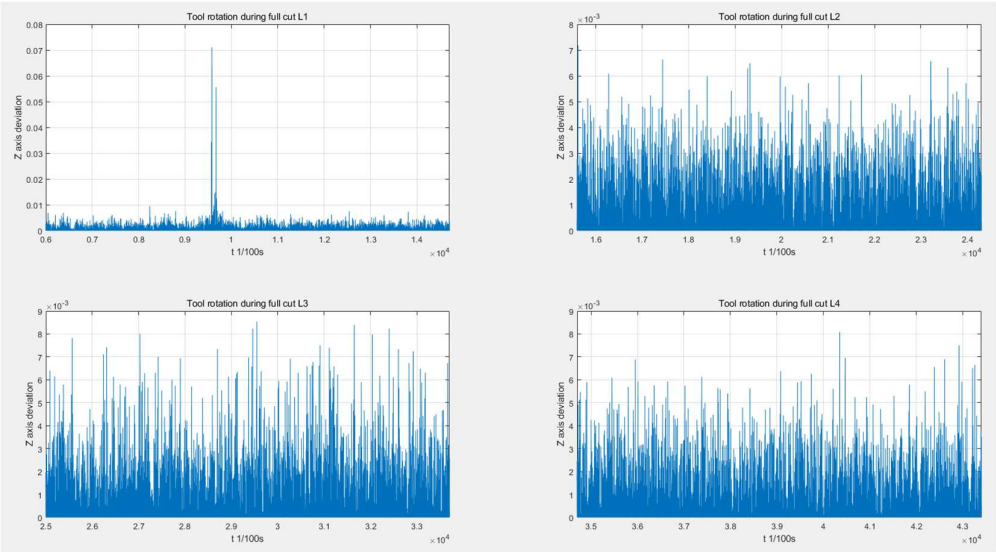


Figure 77: *Tool vibration (Location 2, a_e=5mm)*

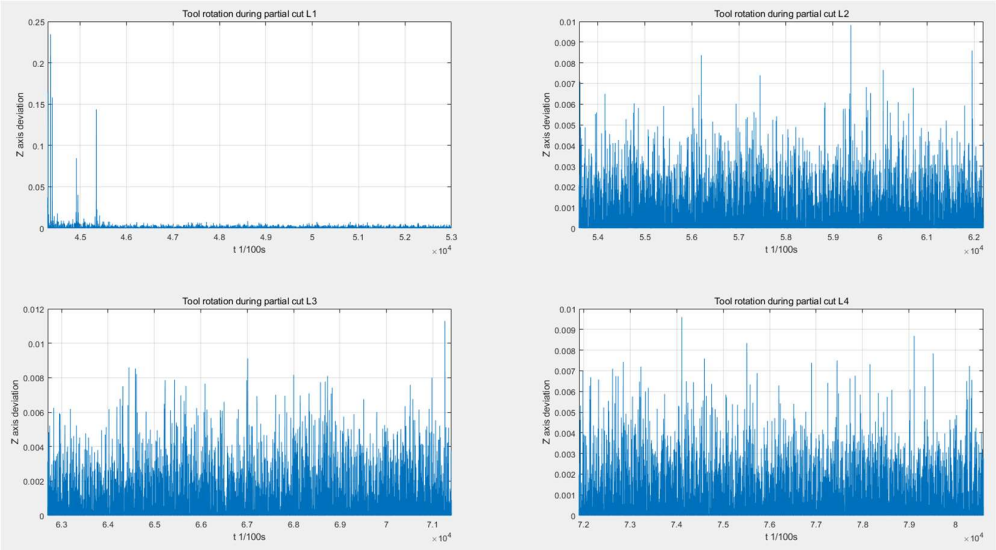


Figure 78: *Tool vibration (Location 2, a_e=2mm)*

The forces that are shown in **Figure 79** and **Figure 80** illustrate that during the process the force is very even at about 22N. We can see there are few forces applied on Line 1 both in the full cut and the partial cut. They could be the explanation for the speed vacuum that happened above.

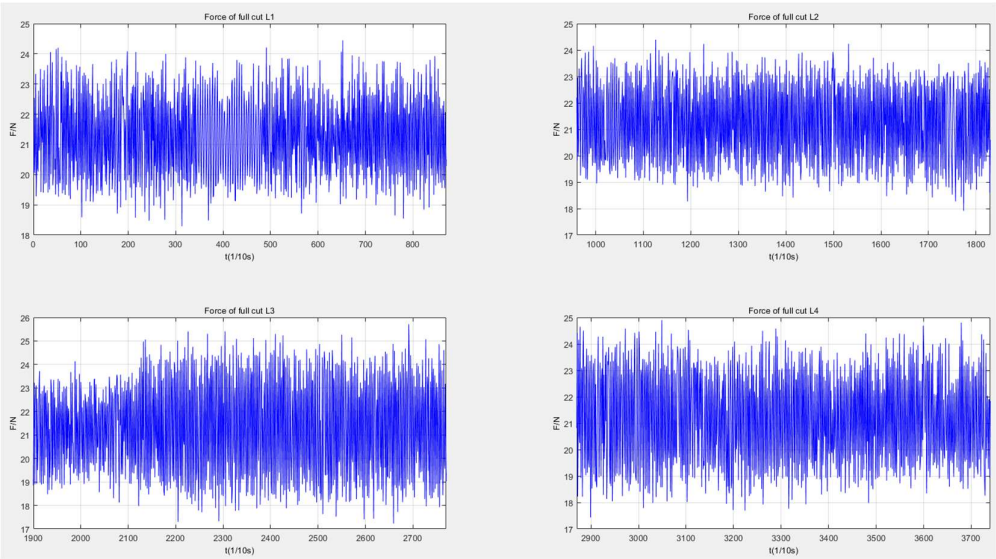


Figure 79: *The resultant force of lines (Location 2, a_e=5mm)*

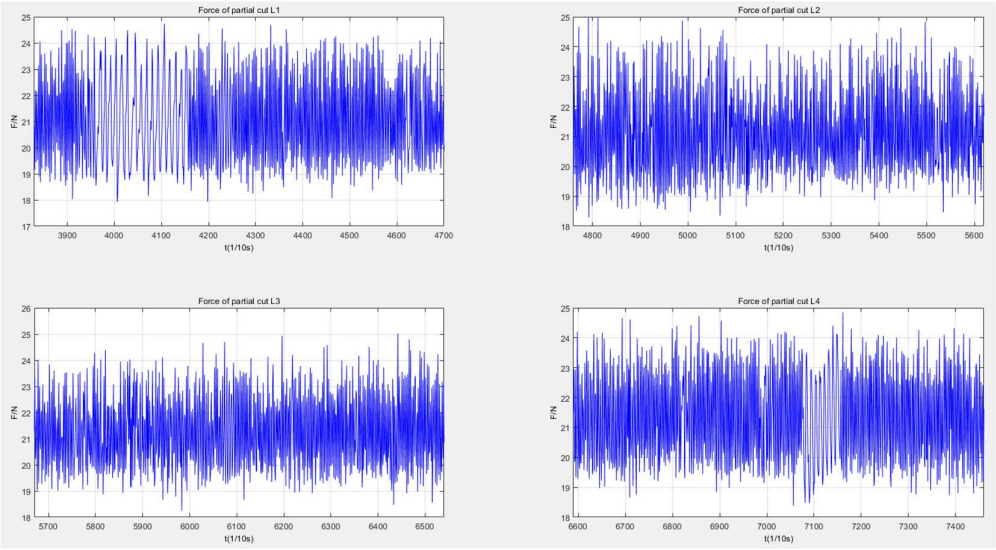


Figure 80: *The resultant force of lines (Location 2, a_e=2mm)*

8.2.3 Location 3

It is not difficult to know that the process in Location 3 is more steady than that in Location 2 and the errors are lower than that of Location 1 if we have a look at **Figure 81**, **Figure 82**, and **Figure 83**. The reason behind this could probably be the tool weight is much larger than the generated force. So that the external force is not capable to make any displacement to the tool on the global z-axis.

The maximum error does not exceed 0.08mm during the full cut and the average error is around 0.02mm. It means the robot milling can reach the accuracy class IT 8~7, which is the middle finishing machining class for CNC, in such a case.

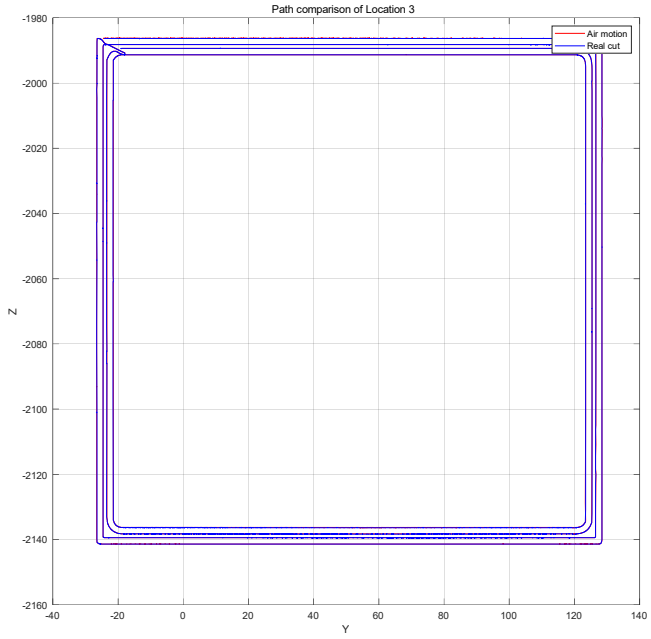


Figure 81: *Air motion and real cut paths (Location 3)*

There are two big errors on Line 2 both in full cut and partial cut. In the full cut, its value is around 0.02mm, and 0.05mm in the partial cut. Based on **Figure 82** and **Figure 83**, it can be seen that they happen in the same positions. The probable reason could be the big grains in the raw material, which makes the material harder than the other places, causing a little larger vibration and more displacement. **Figure 84** and **Figure 85** illustrate that when the vibration is larger than 0.005º, a big error occurs. It is also the same for Line 4.

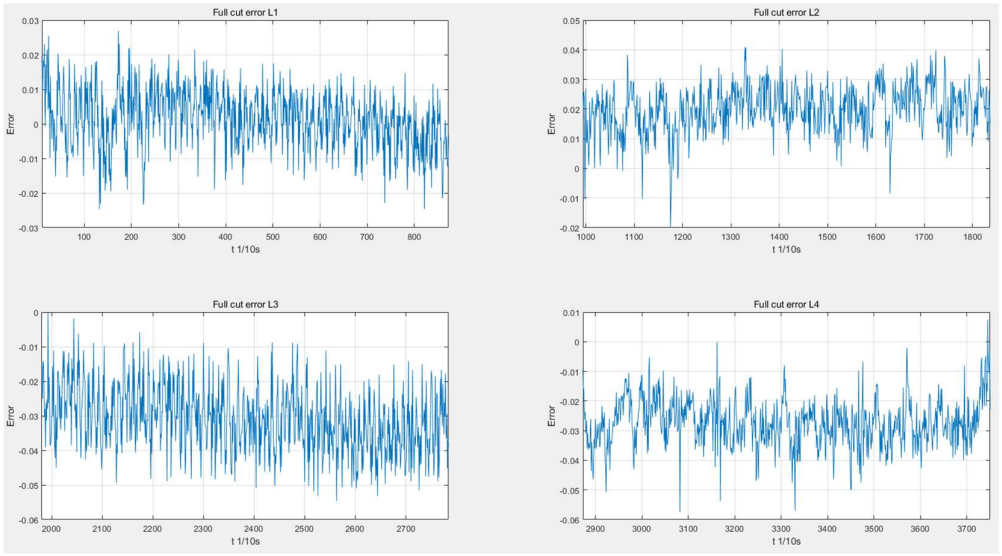


Figure 82: *Accuracy errors of Lines (Location 3, a_e=5mm)*

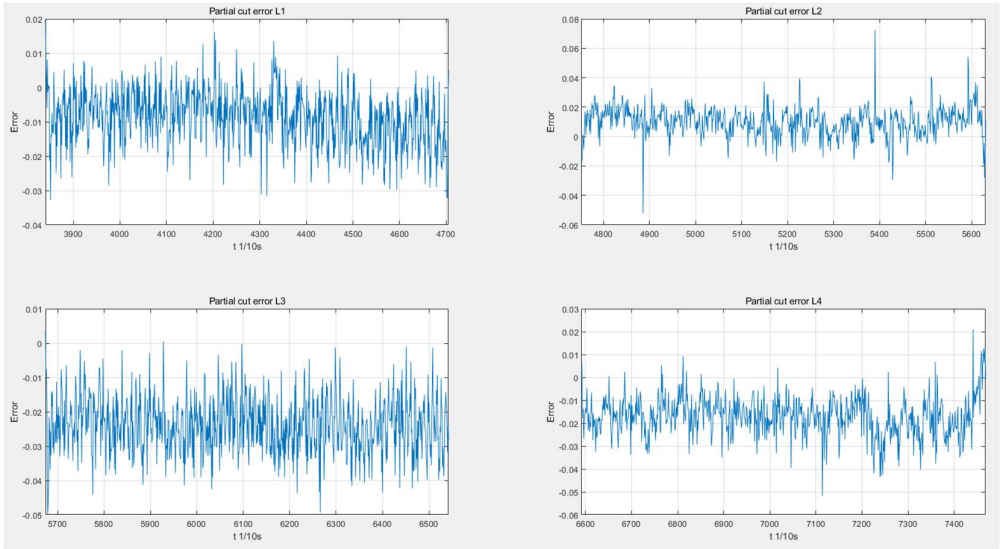


Figure 83: *Accuracy errors of Lines (Location 3, a_e=2mm)*

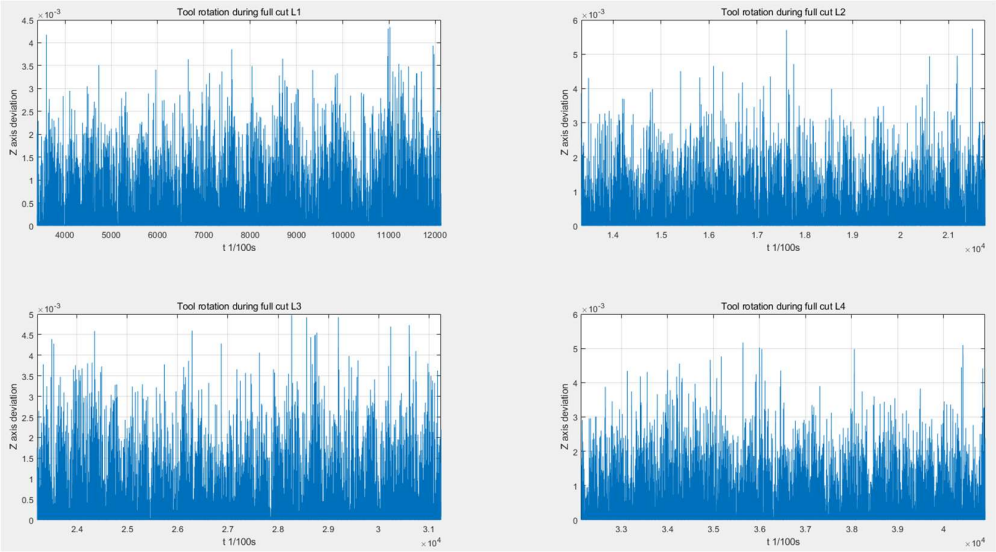


Figure 84: *Tool vibration (Location 3, a_e=5mm)*

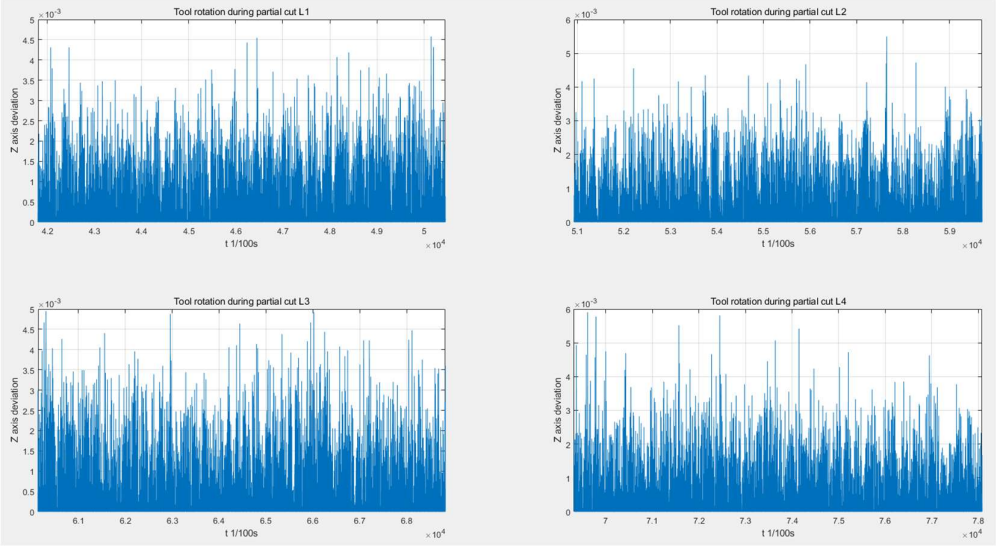


Figure 85*: Tool vibration (Location 3, a_e=2mm)*

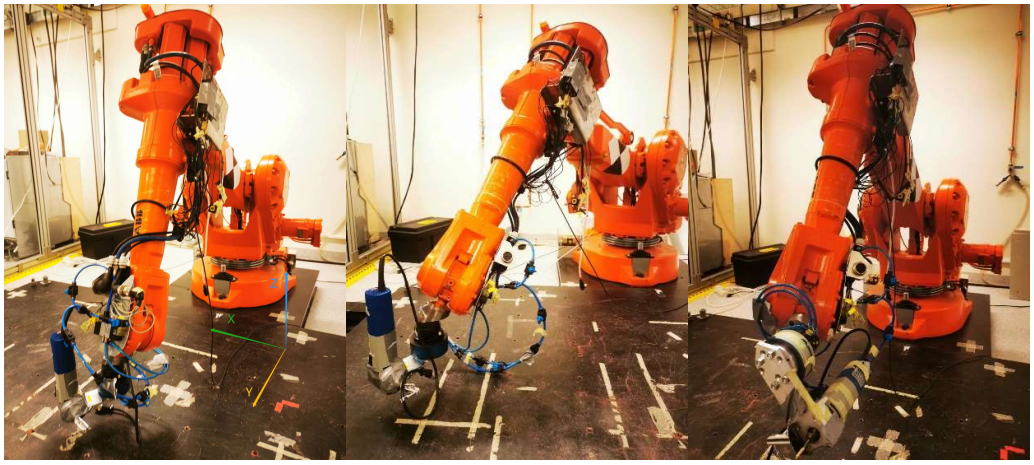
9. Discussion and conclusion

The cause of the different performances shown above is mainly because of the posture of the robot arm. The joint values of the three locations are summarized in **Table 6**, and the actual postures are shown in **Figure 86**.

If we have a look at the posture of, for example, Location 1, especially joint 4 and 5, it is not hard to know the stiffness on the x-axis could be much smaller than that on the y-axis. It is because the joint 5 is now pointing to the x-axis, which means it is capable of rotating in the x-o-z plane (workpiece frame). So, when a horizontal external force comes from whatever direction, the first part that has a displacement must be along the x-axis. This is also the reason why Line 2 and Line 4 have larger errors than Line 1 and Line 3.

Table 6: *Joint values in each location*

Joint value (°) / Location	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Location 1	2.36	62.66	66.78	110	-27	-53
Location 2	-33	75.7	49.98	-44	19.64	65.74
Location 3	-2.35	51.39	55.11	-34	-45	-95



(1) (2) (3)

Figure 86: *The postures of three locations, (1)Location 1, (2)Location 2, (3)Location 3*

Figure 87 summarized the average, maximum, and minimum path errors of the process with 1.8 and 1.44 cc/min MRR. It clearly shows when the milling force points to a low-stiffness direction, a large deviation will occur, like in Line 2 and Line 4. The difference could be large as 1mm. However, if the force points in a high-stiffness direction, a great performance can be achieved as small as 0.026 mm. Since the parameters for this case were out of the limits that the robot could manage, the path accuracy is not good enough to accomplish a finish milling task. Besides, the milling force could be very high, as shown in **Figure 88**. These data clearly show us what is the consequence if we choose the parameters that are beyond the limits of the robotic milling process.

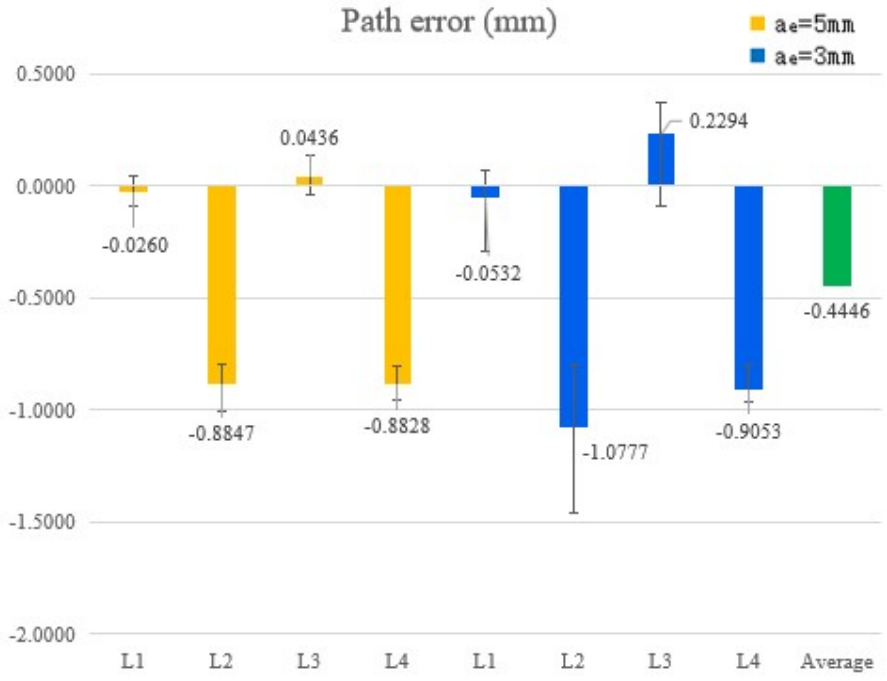


Figure 87: *The path error summary of MRR 1.8 and 1.44 cc/min*

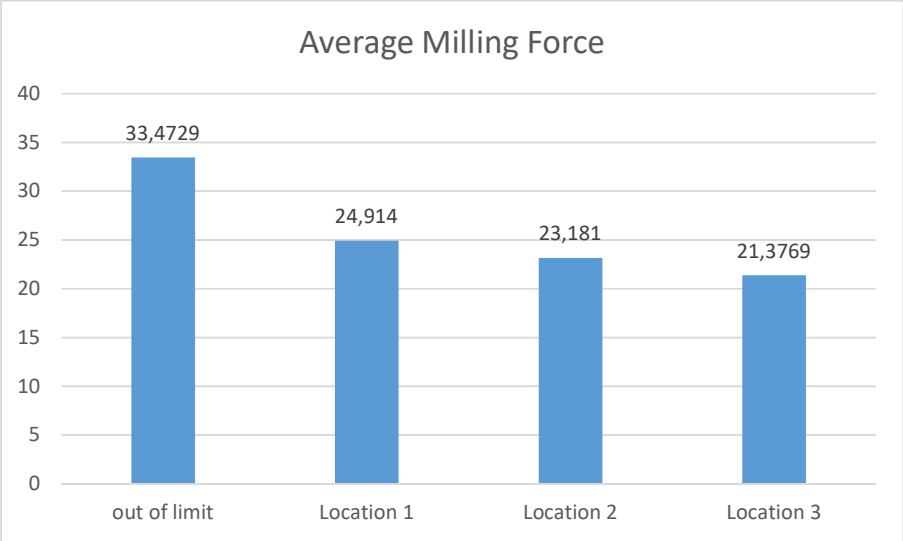


Figure 88: *The average milling force*

On the contrary, in the case that proper parameters are chosen, the path accuracy has been significantly improved, more than 10 times smaller than before, as shown in **Figure 89**. The errors are under 0.09 mm, and they can be nearly 0 along the high-stiffness direction. The average error is around 0.037 mm.

Better performance can be achieved if a more steady posture is chosen. The average path error is 0.0086 mm and 0.0096 mm in Location 2 and Location 3 respectively, as illustrated in **Figure 90** and **Figure 91**. Especially in Location 3, the path accuracy is quite steady and the largest error is not more than 0.04 mm. It means the robot does great in Location 3 with such a posture. It could be because the weight of the milling tool compensates for the external forces during the milling. Since the tool's weight is much larger than the milling forces, it can not be forced to displace itself.

To conclude, the evaluation method used in this study is capable of indicating how well a robot could be in the milling process. The important thing is to find the suitable process parameters that fit the robot. With a proper posture of the robot, the performance can be as good as what CNC does to aluminum.

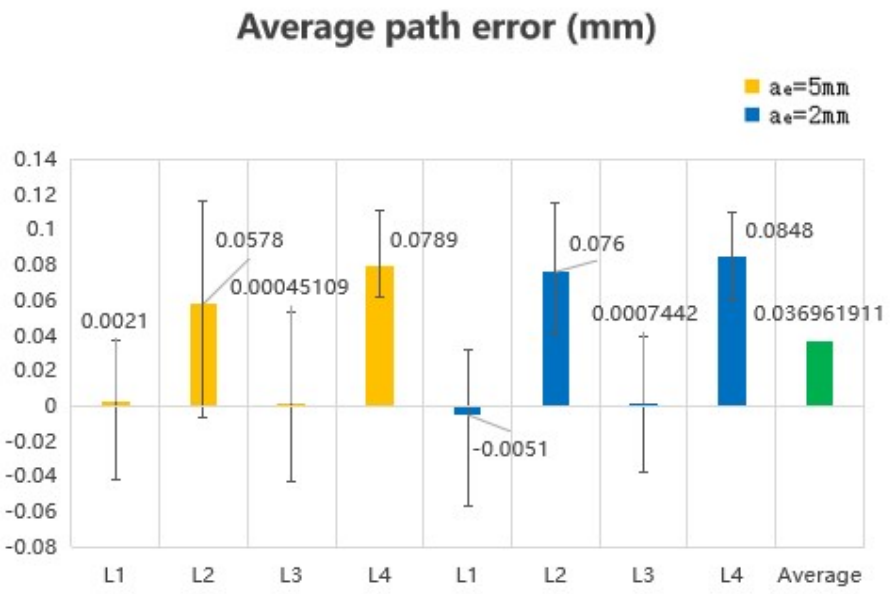


Figure 89: *The path error summary of Location 1, MRR 0.48 cc/min*

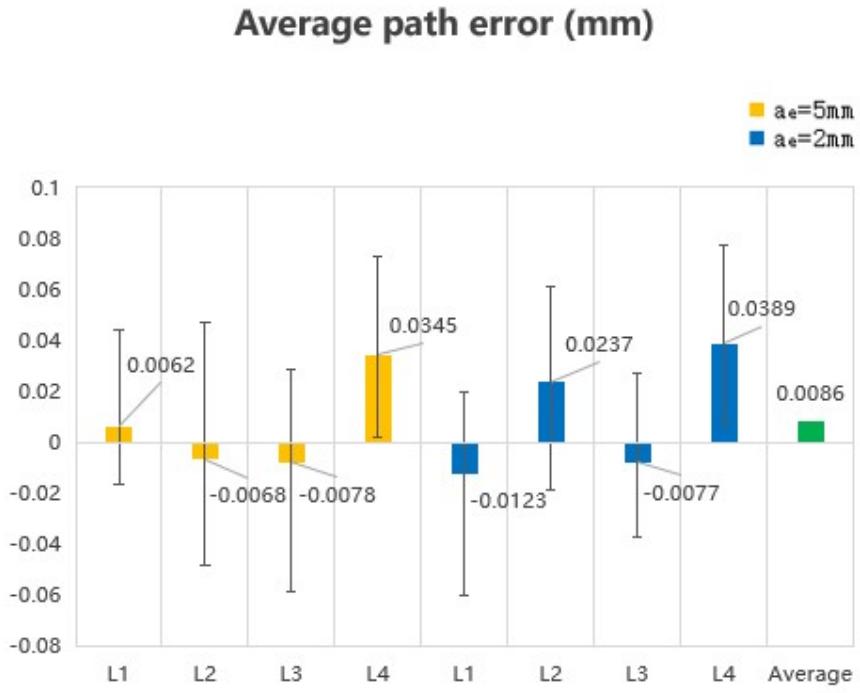


Figure 90: *The path errors summary of Location 2, MRR 0.48 cc/min*

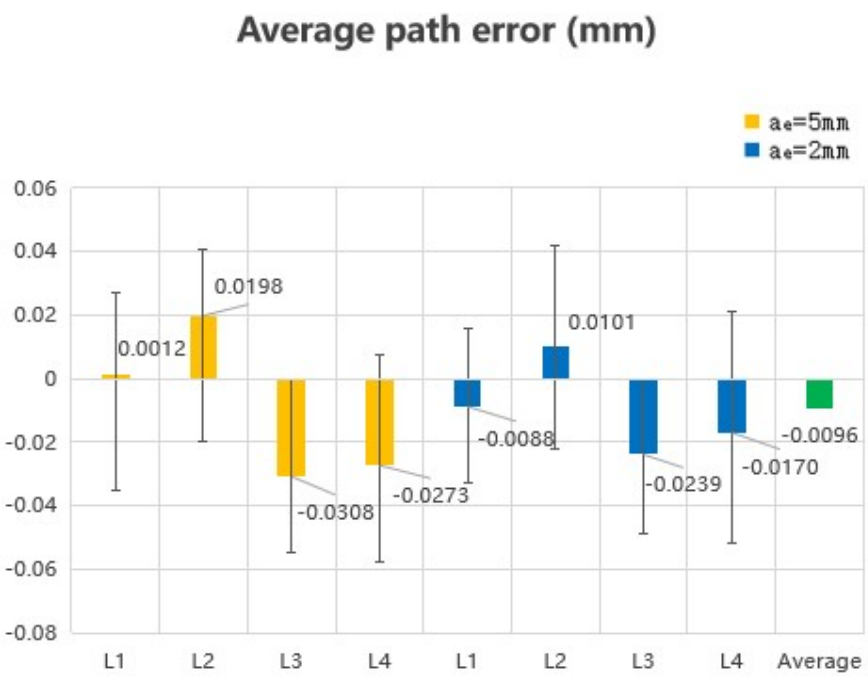


Figure 91: *The path errors summary of Location 3, MRR 0.48 cc/min*

References

[1] U. Schneider, M. Drust, M. Ansaloni, C. Lehmann, M. Pellicciari, F. Leali, J.W. Gunnink, A. Verl, Improving robotic machining accuracy through experimental error investigation and modular compensation, The International Journal of Advanced Manufacturing Technology 85(1-4) (2014) 3-15.
[2] P. Ericsson, Milling accuracy improvement of a 6-axis industrial robot through dynamic analysis, Skolan För Teknikvetenskap, Kungliga Tekniska högskolan, 2019, pp. 1-67.
[3] F. Leali, A. Vergnano, F. Pini, M. Pellicciari, G. Berselli, A workcell calibration method for enhancing accuracy in robot machining of aerospace parts, The International Journal of Advanced Manufacturing Technology 85(1-4) (2014) 47-55.
[4] J.D. Barnfather, M.J. Goodfellow, T. Abram, Achievable tolerances in robotic feature machining operations using a low-cost hexapod, The International Journal of Advanced Manufacturing Technology 95(1-4) (2017) 1421-1436.
[5] H.Z. Jianjun Wang, and Thomas Fuhlbrigge, Improving Machining Accuracy with Robot Deformation Compensation, Intelligent Robots and Systems, 2009, pp. 1-6.
[6] R. Kelaiaia, Improving the pose accuracy of the Delta robot in machining operations, The International Journal of Advanced Manufacturing Technology 91(5-8) (2016) 2205-2215.
[7] O. Sörnmo, B. Olofsson, A. Robertsson, R. Johansson, Increasing Time-Efficiency and Accuracy of Robotic Machining Processes Using Model-Based Adaptive Force Control, IFAC Proceedings Volumes 45(22) (2012) 543-548.
[8] Y. Lin, H. Zhao, H. Ding, Posture optimization methodology of 6R industrial robots for machining using performance evaluation indexes, Robotics and Computer-Integrated Manufacturing 48 (2017) 59-72.
[9] G. Janez, K. Timi, G. Karl, B. Miran, Accuracy improvement of robotic machining based on robot's structural properties, The International Journal of Advanced Manufacturing Technology 108(5-6) (2020) 1309-1329.
[10] A. Verl, A. Valente, S. Melkote, C. Brecher, E. Ozturk, L.T. Tunc, Robots in machining, CIRP Annals 68(2) (2019) 799-822.
[11] A.P. Tomas Kubela, Vladislav Singule, High Accurate Robotic Machining based on Absolute Part Measuring and On-line Path Compensation, Electrical Drives & Power Electronics, 2019, pp. 1-7.
[12] G. Bwegström, Method for calibration of off-line generated robot program, Automatic Control, Chalmers, 2011, pp. 1-113.
[13] A. Klimchik, A. Ambiehl, S. Garnier, B. Furet, A. Pashkevich, Efficiency evaluation of robots in machining applications using industrial performance measure, Robotics and Computer-Integrated Manufacturing 48 (2017) 12-29.
[14] B. Majedi, Robotic Machining Evaluation of the Positioning Accuracy and the Machined Surface Quality, Mechanical Engineering, The Université du Québec, 2014, pp. 3-129.
[15] A.I.a.o. America, NAS 979, Uniform Cutting Test - NAS Series Metal Cutting Specification, 1969.

[16] R. Wolny, Evaluation of Accuracy of Five-Axis CNC Milling Machine on the Basis of Test Piece Machining, Proceedings of the 29th International DAAAM Symposium 20182018, pp. 0164-0168.
[17] M. Abderrahim, A. Khamis, S. Garrido, L. Moreno, Accuracy and Calibration Issues of Industrial Manipulators, Industrial Robotics: Programming, Simulation and Applications2006.

Appendix

A. Force sensor calibration Matlab and Robotstudio code

A.1 Matlab code

```
clear all
clc

force = readtable('forcemeasurment.xls');

m = force(1,3)/9.82
% measurement points
anglesX = [60, 30, 0, -30, -60, -90];
anglesY = [60, 30, 0, -30, -60, -90];
anglesZ = [60, 30, 0, -30, -60];
Qx = rotz(90);
Qy = rotx(90);
Qz = roty(60);
% known orientations with origin in itself of end-effector
for k = 1:size(anglesY,2)
Qy = [Qy roty(anglesY(k))];
Qx = [Qx rotx(anglesX(k))];
end
for k = 1:size(anglesZ,2)
Qz = [Qz rotz(anglesZ(k))];
end
% calculating the normal forces at the end-effector
X = calibration_func(Qy(:,1:3), m);
for i = 4:3:21
Q_temp = Qy(:,i:i+2)
X = [X calibration_func(Q_temp, m)]
end
for i = 1:3:15
Q_temp = roty(90)*Qz(:,i:i+2)
X = [X calibration_func(Q_temp, m)]
end
for i = 1:3:21
Q_temp = roty(90)*Qx(:,i:i+2)
X = [X calibration_func(Q_temp, m)]
end
%calculate displacment
torqueVector = force(1,4:6)';
```

```
for i = 2:size(force,1)
torqueVector = [torqueVector; force(i,4:6)'];
end
d = X'\torqueVector
d = (-1)*d
function X = calibration_func(Q,m)
g = 9.82;
mg= m*g*[-1;0;0];
mg_prim = Q\mg;
F = -mg_prim;
X = [0 mg_prim(3) -mg_prim(2);
-mg_prim(3) 0 mg_prim(1);
mg_prim(2) -mg_prim(1) 0]
end
```

A.2 Robotstudio code

MODULE Force_sensor_cali

CONST robtarget
Home:=[[928.61214857,0,1497.500018623],[0.866025467,0,0.49999989,0],[0,0,0,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_0_90_0:=[[897.499962754,0,1381.387819174],[0.258818801,0,0.965925892,
0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_0_60_0:=[[928.61214857,0,1412.499981377],[0.49999989,0,0.866025467,0],
[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_0_30_0:=[[940,0,1455],[0.707106781,0,0.707106781,0],[0,0,0,0],[9E+09,9E+
09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_0_0_0:=[[928.61214857,0,1497.500018623],[0.866025467,0,0.49999989,0],[
0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_0_n30_0:=[[897.499962754,0,1528.612180826],[0.965925892,0,0.258818801
,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_0_n60_0:=[[855.000026683,0,1540],[1,0,0.000000157,0],[0,0,0,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_0_n90_0:=[[812.50001326,0,1528.612166977],[0.96592585,0,-
0.258818958,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_30_90_0:=[[897.499962754,36.806106541,1391.249990688],[0.249999755,0.

```
066987266,0.933012734,0.250000135],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09
,9E+09]];
    CONST robtarget
Target_60_90_0:=[[897.499962754,63.750037246,1418.193941843],[0.224143624,0.
129409457,0.836516238,0.482963158],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09
,9E+09]];
    CONST robtarget
Target_90_90_0:=[[897.499962754,73.612180826,1454.999976892],[0.183012558,0.
1830125,0.683012855,0.683012641],[0,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_120_90_0:=[[897.499962754,63.750025253,1491.806078929],[0.129409421,
0.224143645,0.482963021,0.836516317],[0,1,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_150_90_0:=[[897.499962754,36.806078169,1518.750025692],[0.066987211,
0.24999977,0.249999927,0.933012789],[0,1,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_180_90_0:=[[897.499962754,-
0.000023986,1528.612180826],[-0.000000042,0.258818801,-
0.000000157,0.965925892],[-1,2,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_0_90_30:=[[897.499962754,0,1381.387819174],[0.249999755,0.250000135,0
.933012734,0.066987266],[-1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_0_90_60:=[[897.499962754,0,1381.387819174],[0.224143624,0.482963158,0
.836516238,0.129409457],[-1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_0_90_90:=[[897.499962754,0,1381.387819174],[0.183012558,0.683012641,0
.683012855,0.1830125],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_0_90_120:=[[897.499962754,0,1381.387819174],[0.129409421,0.836516317,
0.482963021,0.224143645],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_0_90_150:=[[897.499962754,0,1381.387819174],[0.066987211,0.933012789,
0.249999927,0.24999977],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PROC main()

Path_calib2;
Stop;

ENDPROC
```

```
PROC Path_calib2()
        MoveJ Home,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_60_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_30_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_0_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_n30_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_n60_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_n90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_30_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_60_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_90_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_120_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_150_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_180_90_0,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_90_30,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_90_60,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_90_90,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_90_120,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
        MoveJ Target_0_90_150,v60,fine,tool0\WObj:=wobj0;
         WaitTime 3;
ENDPROC
ENDMODULE
```

B. Nikon metrology camera calibration Matlab code

```
clc
array_z = evalin('base','K6C_12409_3_1');
% array_x = evalin('base','K6C_12409_3_2');
% array_y = evalin('base','K6C_12409_3_3');

% % Find the start
% x0 = array_x(1,1:100:end);
% figure(1)
% plot(x,'-o');
% grid on

% plot

x = array_z(1,(83)*100:10:150*100)';
y = array_z(2,(83)*100:10:150*100)';
z = array_z(3,(83)*100:10:150*100)';

P = [x y z];
p1 = [array_z(1,10000),array_z(2,10000),array_z(3,10000)];
p2 = [array_z(1,60000),array_z(2,60000),array_z(3,60000)];
p3 = [array_z(1,70000),array_z(2,70000),array_z(3,70000)];
normal = cross(p1-p2, p1-p3);
z = [0,0,1];
theta = acos(dot(normal,z)/(norm(normal)*norm(z)))*180/pi;

figure(2)
surface = plot3(P(:,1), P(:,2), P(:,3), '-');

direction_y = [0 1 0];
direction_z = [0 0 1];
direction_x = [1 0 0];
rotate(surface, direction_y, -theta);
rotate(surface, direction_z, -55.1);
% rotate(surface, direction_x, 180);
axis equal
xlim([260 450]);
ylim([-180 10]);
grid on
hold on
view(0,90);
title('Position measurement');
xlabel('X');
```

```
ylabel('Y');
zlabel('Z');
```

C. Path accuracy error recognization

```
% Error recognization
 x = xlsread('E:\study in Sweden\Master Thesis\Data\Milling
test_1\Bad_milling_calibrated data.xlsx','x')';
y = xlsread('E:\study in Sweden\Master Thesis\Data\Milling
test_1\Bad_milling_calibrated data.xlsx','y')';
 %%
% Error full L1
t_full_L1 = 1:700;
tmax_full_L1 = max(t_full_L1);
full_L1_y = y(1,(2)*100:10:(2+69)*100)';
array_L1_y0 = xlsread('E:\study in Sweden\Master Thesis\Data\Air
motion\air motion benchmark full.xlsx','L1_y');
full_L1_y0 = array_L1_y0(1:10:end,1);
error_full_L1 = full_L1_y0 - full_L1_y;
merror_full_L1 = mean(error_full_L1);

figure(1)
plot(error_full_L1,'o-');
grid on
hold on
[error_full_L1max,I]=max(error_full_L1);
[error_full_L1min,O]=min(error_full_L1);
plot(t_full_L1(I),error_full_L1(I),'.','MarkerSize',10)
plot(t_full_L1(O),error_full_L1(O),'.','MarkerSize',10)
str_max_full_L1 = ['(' num2str(t_full_L1(I)) ','
num2str(error_full_L1(I)) ')'];
str_min_full_L1 = ['(' num2str(t_full_L1(O)) ','
num2str(error_full_L1(O)) ')'];
text(t_full_L1(I),error_full_L1(I),str_max_full_L1)
text(t_full_L1(O),error_full_L1(O),str_min_full_L1)
plot([0 710], [merror_full_L1 merror_full_L1],'--r');
str_mean_full_L1 = num2str(merror_full_L1);
text(tmax_full_L1,merror_full_L1,str_mean_full_L1,'color','r')

title('Error:full cut L1')
xlabel('t 1/10s')
ylabel('Error/mm')
%%
% Error full L2
```

```
t_full_L2 = 1:701;
tmax_full_L2 = max(t_full_L2);
full_L2_x = x(1,(2+69+6)*100:10:(2+69+6+70)*100)';
array_full_L2_x0 = xlsread('E:\study in Sweden\Master Thesis\Data\Air
motion\air motion benchmark full.xlsx','L2_x');
full_L2_x0 = array_full_L2_x0(1:10:end,1);
error_full_L2 = full_L2_x0 - full_L2_x;
merror_full_L2 = mean(error_full_L2);
```

D. Tool orientation measurement

```
% Orientation processing
clc
array_z = evalin('base','K6C_12409_3_1');
array_x = evalin('base','K6C_12409_3_2');
array_y = evalin('base','K6C_12409_3_3');
% voz = array_z(1:3,6500:end)-array_x(1:3,6500:end);
% voy = array_x(1:3,6500)-array_y(1:3,6500);

for i = 68*100:10:111321
    % generate three points A,B,C
    B = [array_z(1,i) array_z(2,i) array_z(3,i)];
    A = [array_x(1,i) array_x(2,i) array_x(3,i)];
    C = [array_y(1,i) array_y(2,i) array_y(3,i)];

    b = [array_z(1,i-1) array_z(2,i-1) array_z(3,i-1)];
    a = [array_x(1,i-1) array_x(2,i-1) array_x(3,i-1)];
    c = [array_y(1,i-1) array_y(2,i-1) array_y(3,i-1)];

%     % normal ABC plane have a point D
%     norm = cross(A-B, A-C).*0.01;
%     D = [norm(1) norm(2) norm(3)];

    % generate vectors
    AB = B-A;
    AC = C-A;
    ab = b-a;
    ac = c-a;
%     AD = D-A;
    ABmo = sqrt(AB(1)^2+AB(2)^2+AB(3)^2);
    ACmo = sqrt(AC(1)^2+AC(2)^2+AC(3)^2);
    abmo = sqrt(ab(1)^2+ab(2)^2+ab(3)^2);
    acmo = sqrt(ac(1)^2+ac(2)^2+ac(3)^2);
%     ADmo = sqrt(AD(1)^2+AD(2)^2+AD(3)^2);
```

```
    % calculate angle between AB and AC sigma; angle between AC and
AD gamma
    sigma = acos(dot(AB,AC)/(ABmo*ACmo))*180/pi;

%     gamma = acos(dot(AC,AD)/(ACmo*ADmo))*180/pi;

    alpha = acos(dot(AB,ab)/(ABmo*abmo))*180/pi;
    if alpha > 30
        alpha = alpha -90;
    end
    angle_z(i,:) = alpha;
end
```

E. Data access

All the data can be found in the link below:

https://1drv.ms/u/s!AoxHbAaPtVqVcIEFCfWJBTWcUwI?e=OTQUlR