PROCESS DEPENDENT PATH PLANNING FOR MACHINING WITH INDUSTRIAL ROBOTS

A Dissertation
Presented to
The Academic Faculty

By

Keith Ng

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering
College of Engineering

Georgia Institute of Technology

August 2022

PROCESS DEPENDENT PATH PLANNING FOR MACHINING WITH INDUSTRIAL ROBOTS

Thesis committee:

Dr. Shreyes Melkote, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Stephen Balakirsky
Georgia Tech Research Institute
Georgia Institute of Technology

Dr. Steven Liang
School of Mechanical Engineering
Georgia Institute of Technology

Date approved: July 26, 2022

ACKNOWLEDGMENTS

I would like to the many people who supported me in my research endeavors and ensured that I was able to write this thesis. First is Dr. Shreyes Melkote, my advisor, who provided unending support, feedback, and guidance throughout my time at Georgia Tech. I would also like to thank my other committee members, Dr. Stephen Balakirsky and Dr. Steven Liang for their time and insight in serving on my reading committee.

My research would not have been possible without the support of Boeing, for funding this project, and Allison Brown and Dr. Phil Freeman from Boeing. I also greatly appreciate the help of Steven Sheffield, Matt Caroll, and the rest of the staff at the Montgomery Machining Mall for their machining expertise.

I'd like to thank my seniors, Dr. Vinh Nguyen and Dr. Toni Cvitanic, for frequently proving support during their personal time. I'd also like to thank my labmates and officemate, Lindsey Lanzillotta, for their encouragement and help. Last but not least, I'd like to thank my parents, sister, and friends for their various forms of support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADS Automation Device Specification

C^n continuity n-order parametric continuity

CAM Computer Aided Manufacturing

CIRC circular

CMM Coordinate-Measuring Machine

CNC Computer Numerical Control

CP Complete Pose

CSV Comma-separated Values

DoC Depth of Cut

DoF Degree of Freedom

DPP Decoupled Partial Pose

FEA Finite Element Analysis

FIR Finite-Impulse Response

G^n continuity n-order geometric continuity

HSS High-Speed Steel

IPM Inches per Minute

KRC4 KUKA Robot Controller 4

KRL KUKA Robot Language

LIN linear

LIPB Linear Interpolation with Polynomial Blends

P2P Point to Point

PI Proportional Integral

PID Proportional Integral Derivative

RPM Revolutions per Minute

RSI Robot Sensor Interface

SGM Sensor Guided Motion

SLIN spline-linear

SPL spline

T-Mac Tracker-Machine control sensor

TiN Titanium Nitride

SUMMARY

The use of industrial robots in machining operations, such as milling, is an area of growing interest due to potential workflow and efficiency benefits. However, the inherent mechanical design of robot manipulators results in low stiffness and easy-to-excite dynamics when compared to the traditionally used CNC machines. While research exists to compensate for deficiencies in robot manipulators, such as trajectory planning, online and offline error compensation, no integrated solution combining process-force compensation, robotic trajectory planning, and online error compensation exists, as would be required for industrial settings. This thesis introduces a deflection-limited trajectory planning algorithm for curvilinear slotting and linear peripheral milling cuts. The research purpose is to develop a solution involving a variable feed rate trajectory that limits the deflection-induced part errors when milling with an industrial robot. Thus, given a set of points to be approximated into a path, the methodology in this thesis generates a process-aware trajectory in which feed-rate has been adjusted to meet a user-specified deflection limit. The trajectory is formatted to be compatible with a closed-loop feedback and communication system with the industrial robot. Experiments are conducted using a large (range of 2855 mm), industrial robot milling system controlled by a closed-loop, laser tracker feedback system. Experimental data supports that the deflection-limited variable feed rate trajectory provides better part accuracy and surface roughness than the constant feed rate case. Furthermore, the variable feed rate trajectory executed by the closed-loop system maintains better positional accuracy than the open-loop, native robot controller using native motion types. Thus, the merit of a process dependent trajectory planner is argued, and future work for improvements and use-case generalization is suggested.

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Statement

Machining processes are used extensively for high accuracy and high precision manufacturing, including for aerospace applications. While modern CNC machines allow production of complex shapes with good dimensional accuracy and repeatability, this is achieved by the specific, purposeful design of CNC machines, prioritizing stiffness and error minimization [1]. This purposeful design and its benefits come at the cost of process flexibility and size limitations. For example, milling tools have been placed on mobile robotic platforms, such as legged walking robots and robot arms on wheeled vehicles, which allows for the machine tool to move to the workpiece [2–4]. Typically, with a CNC machine, the workpiece must be moved to and fit within the relatively small machine workspace. Additionally, a typical CNC machine tool used in industrial manufacturing, especially aerospace, has high capital cost compared to an industrial robot [5, 6]. Therefore, machining with industrial robotic systems has become an area of interest for processes that require larger work volumes and/or more flexible workflows [7].

A major trade-off in using industrial robotic manipulators for subtractive manufacturing, such as milling, however, is significantly lower accuracy arising from kinematic errors and lower stiffness of the manipulator joints and links when compared to conventional CNC machine tools [8]. As machining processes generate significant forces, the lower stiffness of an industrial robot translates to into additional lost accuracy. This can be compensated for with different methods, including trajectory planning, offline error compensation, and online error compensation. While some trajectory planning methods seen in CNC machine research can be applied to robotic manipulators, namely 5th order Bézier curve blends, such trajectory planning methods generally do not account for machine deflection; machine deflection is inherently a more pronounced issue in robotic manipulators than CNC machines due to the structural design. As robotics is a broad field with systems of varying shapes and sizes, traditional robot path and motion planning generally focuses on satisfying constraints on the motion, such as jerk optimization and obstacle avoidance, and does not specifically consider process (task) induced effects on the motion of the robot. Some research has been done to improve robot path accuracy in processes that produce significant forces, but the compensation for such forces is generally independent of the trajectory planning and path approximation of curvilinear paths to be executed by the robot [9–11].

The need for error compensation arises from industrial robots' relative lack of native path accuracy (~ 1 mm) compared to CNC machine tools (~ 0.01 mm) [12]. Online, external, real-time feedback using metrology devices, such as a laser tracker, that have a greater measurement accuracy and resolution than the robot joint encoders can be used to overcome this limitation. Using a Proportional Integral Derivative (PID) controller with tuned gains, industrial robots have been shown to achieve kinematic errors below their repeatability rating (0.08 mm) [13]. However, the position and orientation (pose) accuracy, particularly under external loads, is still lower than that of CNC machine tools.

Offline error compensation exists to alleviate weaknesses of industrial robots, such as low path execution accuracy and stiffness. Training data can be used to reduce path error, but requires a a sample trajectory to be run first, which increases process time [14]. This is counterproductive to the workflow benefits industrial robots can bring. Stiffness models can also be used to optimize pose to minimize deflection-induced errors in milling, as milling is a five Degree of Freedom (DoF) operation and typical six DoF robotic manipulators have a redundant degree of freedom, making many Cartesian positions and orientations non-unique in terms of robot joint angles [13]. In addition to suppressing robot deflection, offline error compensation can minimize vibrations that are more easily excited in a robot manipulator [15]. These error compensation strategies are, however, applied separately, generally after robot trajectory planning and path approximation. This often requires a manual, intermediate step to incorporate error compensation into the traditionally planned robot trajectory.

Despite its advantages, robotic machining is not widely used in production environments, primarily due to the limitations discussed above but partially due to the the abundance of research in CNC machining. The ubiquity of CNC machines has resulted in different comprehensive Computer Aided Manufacturing (CAM) packages for CNC programming. Current research in robotic machining does not provide an integrated, model-based approach to trajectory planning that accounts for process forces and resulting deflections and that can be combined with real-time feedback to correct for any unmodeled effects. It is also not well understood how a process-aware trajectory planner would impact the accuracy of a machined part. Thus, an investigation combining robotic trajectory planning, mechanistic process-force prediction, and online error compensation is needed to enable the feasibility of robots in industrial machining environments. In this thesis, a process force induced deflection-limited trajectory planning algorithm combined with real-time pose-error correction is investigated. Once the impact of this combined approach is better understood, recommendations can be made regarding trajectory planning and operation of industrial robots in practical machining applications.

1.2 Research Objectives

Motivated by the problems discussed above, the goal of this research is to develop a novel, deflection-limited trajectory planning algorithm combined with real-time pose-error correction and to evaluate the performance of such an approach.

The specific objectives of this research are as follows:

1. Develop an offline trajectory planning algorithm that:

    (a) Maximizes the robot feed rate (velocity) while meeting a user-specified constraint on the maximum static deflection experienced by the robot during milling.

(b) Maintains both parametric and geometric continuity of the trajectory up to and including the second derivative (C^2 and G^2, respectively).

2. Integrate the varying feed rate trajectory with a real-time, closed-loop, metrology-aided feedback system for correcting errors in robot pose.

3. Evaluate the varying feed rate trajectory for improvements in machined part accuracy and surface finish.

1.3 Proposed Approach

The overall approach is summarized in Figure 1.1. The time-varying forces generated during milling are applied to the robot manipulator, which experiences an instantaneous static deflection. Within the offline component of this approach, the forces are estimated using a static mechanistic milling force model that predicts the time-varying milling forces and the corresponding static deflections of the robot at points along the tool path. The feed rate of the trajectory is then varied to meet a user-specified deflection limit of the robot end effector.

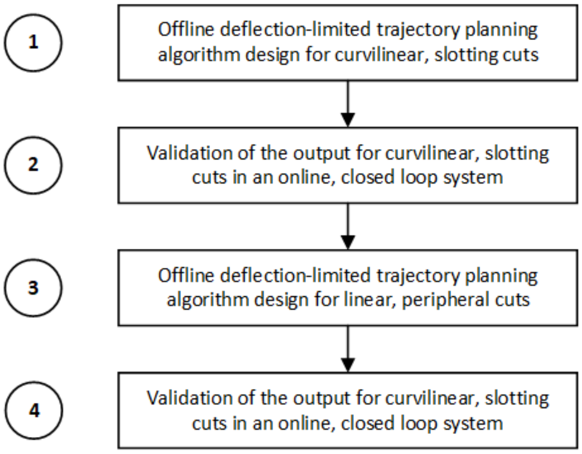

1
Offline deflection-limited trajectory planning algorithm design for curvilinear, slotting cuts
2
Validation of the output for curvilinear, slotting cuts in an online, closed loop system
3
Offline deflection-limited trajectory planning algorithm design for linear, peripheral cuts
4
Validation of the output for curvilinear, slotting cuts in an online, closed loop system

Figure 1.1: Overall research approach

This approach is first applied to curvilinear slotting cuts in which path approximation is applied for the non-linear sections of the path. The performance of the offline deflection-limited trajectory planning algorithm is then experimentally evaluated by implementing it on a six DoF industrial robot with real-time laser metrology-aided closed-loop feedback control system. The process is repeated for linear peripheral cuts in which the radial Depth of Cut (DoC) increases with the length of cut.

1.4 Thesis Outline

The remainder of this thesis is organized into the following chapters:

Chapter 2 reviews prior work and existing literature on trajectory planning and error compensation in both the CNC machining and robotics fields. Chapter 3 introduces the offline deflection-limited trajectory planning algorithm for curvilinear slotting cuts and discusses the experimental results. Chapter 4 adapts the algorithm introduced in Chapter 3 to linear peripheral cuts in which the radial DoC increases with the length of cut and evaluates its performance through an experiment. Finally, Chapter 5 draws conclusions, makes recommendations, and notes potential areas of interest for future work.

CHAPTER 2

LITERATURE REVIEW

In this chapter, a literature review of existing research relevant to the problem addressed in this thesis is presented. The review is organized into four sections: 1) trajectory planning in CNC machines, 2) traditional robotic trajectory planning, 3) process dependent trajectory planning, and 4) error compensation methods.

2.1 Trajectory Planning in CNC Machines

CNC machines are the industry standard in subtractive manufacturing, and as a result, it is necessary to compare industrial robots to CNC tools. However, due to the inherent structural differences between CNC machines and robot manipulators, not all research in trajectory planning for CNC machines is analogous and transferable to industrial robots. For example, 5-axis CNC milling machines are commonly used in industrial machining settings, as milling is a 5-DoF operation. However, robot manipulators commonly have 6-DoF, allowing an extra DoF for pose optimization. Thus, much of the research in overcoming limitations in 3 or 5-DoC, such as the trajectory generator with singularity avoidance proposed in [16], is simply unnecessary. With a typical 6-axis robot manipulator with revolute joints, end effector trajectories need not be modified to avoid singular configurations. Even so, in singular configurations, one DoF can be lost while still executing the milling operation. Additionally, due to the comparatively large workspace provided by industrial robot manipulators, a region of the workspace far from singularity can simply be chosen using the methodology introduced in [17].

Another common area of interest in trajectory planning for CNC machines is feed rate optimization. An appropriate feed rate must be chosen such that the machining operation achieves an acceptable part accuracy and finish. A feed rate that is too high can result in poor accuracy, poor finish, and chatter. On the other hand, a feed rate that is too low increases manufacturing time, an important consideration in industrial applications. Thus, the goal of feed rate optimization is to minimize the time required under a set of constraints – a goal shared with robotic machining applications. Recent research in time-optimal feed rate planning generally focuses on constraining the acceleration along with higher-order derivatives or other criterion, such as chord error [18–20]. These algorithms do not account for process forces or machine deflection. However, compliance in industrial robot is a much more pronounced issue, with deflections that are much greater ($\sim$1 mm) than those of CNC machines ($\sim$0.01 mm) [12]. Thus, what is desired in feed rate optimization for robotic milling systems is a constraint on the deflection of the end effector.

Characteristics of the acceleration and jerk experienced by computer controlled machines, such as a CNC machine tool or an industrial robot, can be shaped in other ways, such as through the geometric path of the trajectory. While the geometric path alone cannot limit the magnitudes of the experienced acceleration and its derivatives, the geometric path can influence the profile and continuity of the acceleration and its derivatives. A 5-axis CNC machine tool transitioning between different linear toolpath segments will experience velocity, acceleration, and jerk discontinuities at the transition points. As such, it is desirable to approximate the path in order to maintain continuity up to acceleration (C^2 continuity), and thereby also limit jerk to a finite value. This is typically achieved with quintic B-splines (or Bézier curves) as such curves can guarantee continuity up to the second derivative [21–24]. While implementation varies, a similar use of quintic Bézier curves is often adopted in robotic path planning.

2.2 Traditional Robotic Trajectory Planning

Indeed, the quintic B-spline introduced in [21] with continuous tool position, velocity, and acceleration is a commonly used as the foundation for other B-spline trajectories in robotics [25, 26]. Despite the inherent mechanical differences between typical CNC machine tools and robot manipulators, acceleration continuity (or the continuity of its derivatives) is dependent on the trajectory itself, not the machine. Thus, research in robotic trajectory planning often builds upon the quintic B-spline by adding additional constraints on the formulation, such as jerk (C^3) continuity, or on the implementation, such as efficient algorithms for online or cheap micro-controller use [26–28]. On the other hand, methods exist for novel robotic path planning, such as using geodesics and regarding the joints of a robot as linearly dependent, but such methods become difficult to implement with more path constraints, such as obstacles or desired part accuracy [29, 30]. For a similar reason, trajectory planning in the robot joint space is typically used when the exact path taken is not important. As a result, path planning in the operational, Cartesian space is appropriate for robotic manufacturing.

Trajectory planning algorithms tailored towards robotic manufacturing exist, although such algorithms generally do not consider process forces. For example, the use of B-spline based path smoothing can be used to minimize vibrations and path error in robotic additive manufacturing, where accuracy is essential but there are no significant forces [31]. In the context of machining, such trajectory planning algorithms generally utilize variants of the discussed spline parameterizations to achieve various goals, such as tracking of complex 3D geometry or integration with a Finite-Impulse Response (FIR) filter for smooth feed rate adjustment of the parametric curve [32, 33]. While such methods can increase accuracy, they ignore or only indirectly address a fundamental component of machining: generated forces, which excite robot vibrations and cause end effector deflection. Process forces are more commonly addressed in other forms of error compensation, such as offline pose optimization schemes and online feed-back systems, which are reviewed next.

2.3 Offline Error Compensation

In one study by Schnoes and Zach [9], a KUKA KR240 industrial robot is used for machining a rectangular pocket, and a procedure is introduced for selecting the placement of the workpiece in the workspace and compensating for robot static deflections caused by generated process forces. Predicted process forces are determined using a mechanistic model and used to predict robot static deflection. The workpiece placement is then determined by optimizing the desired trajectory for the mean deflection within a bounding box defined in the robot reachability range. Finally, a compensated path is created by taking the ideal path and offsetting the predicted static deflection. This method effectively considers the process forces when milling with industrial robots but ignores several aspects key to machining and robotic accuracy. No feed rate optimization is implemented and no smoothing of the path is applied – the geometric path of the rectangular pocket consists of linear segments only, creating a path that does not guarantee acceleration (C^2) continuity. As discussed in [21–26], discontinuities in the acceleration can excite dynamics, which are particularly devastating to accuracy in robot manipulators. Furthermore, the compensated trajectory is exported as machine code for KUKA robot controllers and thus is susceptible to the trajectory modification behavior (path approximation and velocity reduction) and relatively poor accuracy of the native robot controller.

In a similar study by Reinl et al. [10], offline compensation of the path deviation during the milling of a 90° path with a KUKA KR210 industrial robot was investigated. Once again, forces are predicted using a mechanistic model, but additional parameters, such as the modal damping and natural frequencies, are included in the model of the robot dynamics. However, as before, offline execution of a compensated trajectory by the robot (i.e., without any external metrology devices) presents a bottle-neck in improving accuracy: the native robot controller.

One method to account for the native robot controller's behavior to reduce execution errors in an offline compensation scheme is to use a sample trajectory first run as training data for a second, compensated run, as done in [14]. However, this adds significant setup time which reduces its feasibility for the manufacturing of parts that may change in design or are not high volume.

Rather than correcting for path and trajectory errors, other forms of offline error compensation exist. In a similar vein as the workpiece placement optimization determined by mean deflection discussed in [9], modal properties of the robotic system, such as modal frequency, stiffness, and damping coefficient, can be sampled to create a data-driven model to predict vibrations, a key weakness of robotic manipulators [34]. Accurate prediction of vibrations enables identification and selection of areas of the workspace that are suitable for a given operation. However, in some cases, the workpiece may be constrained to a small region of the reachable space of the robot. Then, stiffness optimization can be performed by optimizing the configuration of the robot joints instead [13]. This is enabled by the fact that a typical six-axis revolute joint robot has an extra degree of freedom that generally allows for multiple configurations per end effector pose, and the stiffest configuration can be chosen. This procedure allows for minimization of deflections without needing to know any information about the process and its forces.

2.4 Online Error Compensation

One effective method of error correction that does not need information regarding the cause of error (i.e., process induced effects) is the use of a PID controller. A study from Shi et al. uses a FARO laser tracker to determine the end effector pose at a greater accuracy than the native controller of the KUKA KR5 [35]. Closed-loop feedback to correct for path errors is achieved using the laser tracker measurements. Such a method requires little information about the process and experimental conditions and thus does not require modeling of forces and robot dynamics, making it often effective for zero or constant load tasks. However, in the presence of significant vibrations, corrections sent from the PID controller to the robot may do little to alleviate execution inaccuracy without tuned gains.

Optimized gains have been shown to reduce vibration and improve accuracy in robotic milling, but vibration suppression and stiffness improvement in robotics is generally done with trajectory planning and pose optimization, as previously discussed [15]. These methods can be extended to the online case, for example, with an online B-spline trajectory planner to reduce interpolation error [36]. Additionally, while PID controllers have been shown to be effective in robotic milling for a variety of applications, including aircraft wing fiberglass shims, other, specialized, closed-loop controllers can be utilized to incorporate further information about the process or system [37]. Gonzalez et al. present one such controller which compensates for robot compliance with force data (in contrast to laser tracker data) and a model of the robot system [38].

Xiong et al. introduce an integrated methodology using an offline feed rate planner based on predicted cutting forces coupled with an online Proportional Integral (PI) controller [11]. This method incorporates the feed rate planning aspect of the trajectory planning with error compensation but neglects any path planning of the geometric path. This is due to the experiments run being linear, down-milling operations of constant radial DoC. Clearly, non-continuous, linear movements are not feasible in industrial machining, which is often employed for producing complex part geometry. Extending this methodology to curvilinear paths is essential for feasibility in many manufacturing applications, and an approach will be introduced in the following chapters of this thesis.

2.5 Summary

From the literature survey presented in this chapter, it is clear that strategies exist for increasing accuracy through trajectory planning in CNC machines and robot manipulators as well as offline and online error compensation. However, no combined solution exists in which 1) a curvilinear path is smoothed to maintain G^2 and C^2 continuity, 2) the feed rate is varied according to predicted process forces and robot deflection, 3) external feedback is utilized to overcome inherent kinematic accuracy deficiencies of robot manipulators. Therefore, a methodology incorporating these three key aspects is introduced in the following chapters. An integrated solution to robotic machining is desired to promote widespread adoption of industrial robotic machining by overcoming inherent weaknesses of industrial robots.

CHAPTER 3
DEFLECTION-LIMITED TRAJECTORY PLANNING FOR CURVILINEAR SLOTTING CUTS

3.1 Introduction

In this chapter, a deflection-limited trajectory planning algorithm for curvilinear slot milling (henceforth referred to as slotting) cuts made with industrial robots is presented. The algorithm seeks to vary the feed rate of the robot along the desired path to satisfy the user-specified robot deflection limit, and in doing so, consider process related constraints. In addition to satisfying a static deflection limit, the algorithm guarantees both parametric and geometric continuity of the trajectory up to and including the second derivative (C^2 and G^2, respectively). In the following sections of this chapter, the geometric path approximation, milling force and robot deflection prediction, and varying feed rate trajectory generation methodology are introduced, followed by experimental validation, discussion of the results, and conclusions.

3.2 Algorithm Overview

An overview of the proposed deflection-limited trajectory generation algorithm is shown in Figure 3.1. Given user inputs related to the robot deflection constraint and the desired geometry of the robot end effector path, a set of discrete points defining the user-specified path shape is approximated by a smooth path, one with a uniquely defined first derivative at every point on the path. This is done using Linear Interpolation with Polynomial Blends (LIPB), in which linear segments of the path are connected by a suitable polynomial curve [25]. The predicted static milling forces and corresponding robot deflections are calculated for each linear segment and the feed rate iteratively lowered until the user-specified deflection limit is satisfied. Finally, motion through the curved sections is refit to ensure C^2 continuity, and the trajectory is output as a Comma-separated Values (CSV) file to be read by the closed-loop robot control architecture.

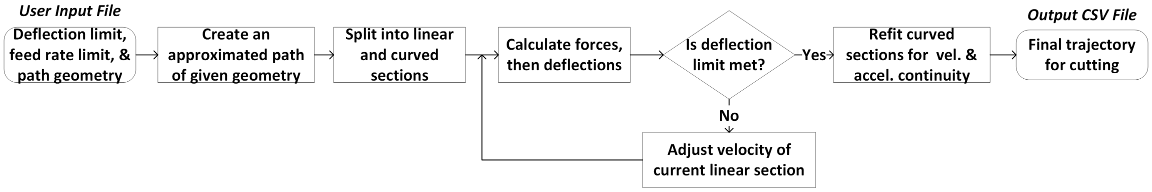
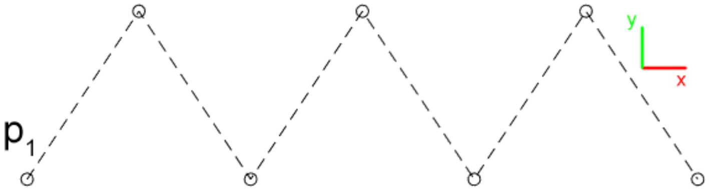

User Input File
Deflection limit, feed rate limit, & path geometry
Create an approximated path of given geometry
Split into linear and curved sections
Calculate forces, then deflections
Is deflection limit met?
Yes
No
Adjust velocity of current linear section
Refit curved sections for vel. & accel. continuity
Output CSV File
Final trajectory for cutting

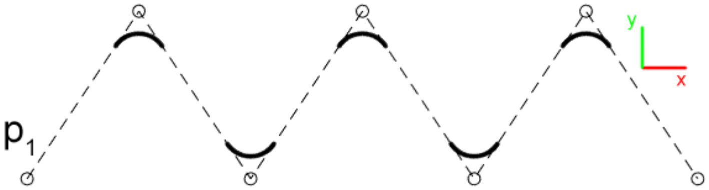
Figure 3.1: Curvilinear slotting cut trajectory planning algorithm overview.

For purposes of illustrating how the algorithm works, a triangular wave path consisting of linear segments with polynomial blends will be considered in this chapter. A square wave path can also be constructed using the same methodology. The algorithm proceeds as follows:

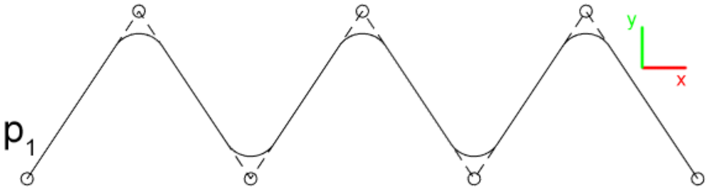
1. Create a XY planar path of corner points $p_i = (x_i, y_i, z_0)$ defining a triangular wave.



2. Approximate the path by creating circular arcs of a specified radius. (See Section B.1.)

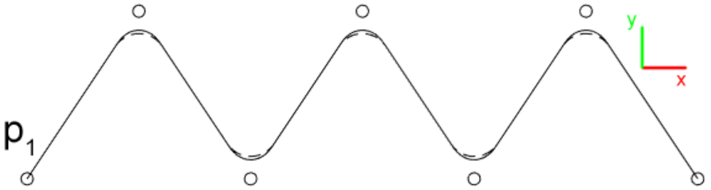


3. Create linear sections connecting the circular arcs.



4. Replace the circular arcs with varying-radius curves, where the radius varies as a 4th order polynomial of angle θ.

   Note: A circular arc for blending linear sections is undesirable as the transition will cause unbounded jerk at the junction points.



   Skip to step 7 if a constant feed rate trajectory is desired.

5. Calculate milling forces and static robot deflections at every point for each linear section.

6. Iterate by lowering the feed rate of the linear section until the user-specified deflection limit is met.

7. With the geometric path of the blending curve defined, use a 5th order polynomial of the tangential displacement $s(t)$ as a function of time to smoothly vary the feed rate along the curved blend and maintain C^2 continuity of the joined sections.

   Note: The use of a 5th order polynomial defines the motion timing law and ensures continuity of the tangential velocity and acceleration at the junction points.

8. Define angularly spaced points along the varying radius curve $r(\theta)$ at $[\theta_0, ..., \theta_f]$ where θ_0 and θ_f correspond to the start and end points of the varying radius curve.

Determine the mapping $\theta(s(t))$ by interpolating the angularly spaced points θ and the corresponding tangential displacements $s(t)$.

9. Using $\theta(s(t))$, determine the interpolated angular positions θ corresponding to equally spaced time instances $t = [0, ..., t_f]$

10. Use $r(\theta)$ to determine the corresponding radial distances to the angular positions determined in the above step and convert the (r, θ, z) values back to Cartesian x, y, z.

11. Output final varying feed rate trajectory as a CSV for use in the closed-loop, real-time control system.

The following sections of this chapter describe in detail the above methodology and reasoning for each step in the above algorithm.

3.3 Path Approximation and Trajectory Planning

3.3.1 Terminology

Before proceeding further, it is important to define and clarify the different terms that will be used in this chapter. As is generally defined in the robotics field, a *path* is a locus of points in the joint or Cartesian space that the manipulator follows during motion; in other words, a pure geometric description of the motion. A *trajectory*, however, is a path with a specified timing law, for example, a path with a specified velocity/acceleration profile [29].

Geometric continuity, naturally, is dependent on the geometric path alone. Suppose we have a path composed of m parameterized segments, where u is the parameter:

$$p(u) = p_k(u) = \begin{bmatrix} p_{k,x}(u) \\ p_{k,y}(u) \\ p_{k,z}(u) \end{bmatrix} \qquad \begin{matrix} k = 0, ..., m-1 \\ 0 \leq u \leq 1 \end{matrix}$$

Then it is said that two segments meet with n-order geometric continuity (G^n continuity) if and only if they share common unit vectors $\left(\frac{d^jp}{du^j}/\left|\frac{d^jp}{du^j}\right|\right)$ where $0 \leq j \leq n$. In this thesis, a G^2 continuous path is desired, that is the unit tangent $\left(\frac{dp}{du}/\left|\frac{dp}{du}\right|\right)$ and unit curvature $\left(\frac{d^2p}{du^2}/\left|\frac{d^2p}{du^2}\right|\right)$ are the same at all points where the different segments meet.

If the requirement is strengthened to ensure two segments that meet share vectors $\frac{d^jp}{du^j}$ where $0 \leq j \leq n$, and not just the unit vectors, then the two segments are said to meet with n-order parametric continuity (C^n continuity). A path with C^2 continuity would guarantee continuity of position, velocity, and acceleration, which is sought for the robot trajectories considered in this thesis. Note that the relationship between G^n continuity and C^n continuity means that a G^n curve ensures the existence of a C^n continuous parameterization, but a G^n curve may not necessarily be C^n continuous [25].

3.3.2 Path Approximation

A geometric path is usually not fully specified by the user due to complexity reasons. Instead, a reduced number of parameters, such as extremal or intermediate way points along the path, and geometric primitives (i.e., lines and circular arcs) are specified [29]. Given a set of way points specified in the Cartesian space, the simplest construction of a path is to linearly connect the points. However, this is usually undesirable since the instantaneous change in direction at the intersection of two non-colinear segments can excite robot vibrations and reduce path accuracy. Thus, the path at such junctions between segments is typically interpolated or approximated with curves. In *interpolation*, the curve exactly passes through the way points, whereas in *approximation*, the curve does not exactly pass through the way points but instead falls in a neighborhood defined by a prescribed tolerance. The two methods are illustrated in Figure 3.2.

If it is not required that the path exactly pass through the way points, interpolation is generally preferred for the following reasons [25]:

- When the path must fit many points, the free parameters characterizing the curve may not be sufficient to obtain an exact interpolation.

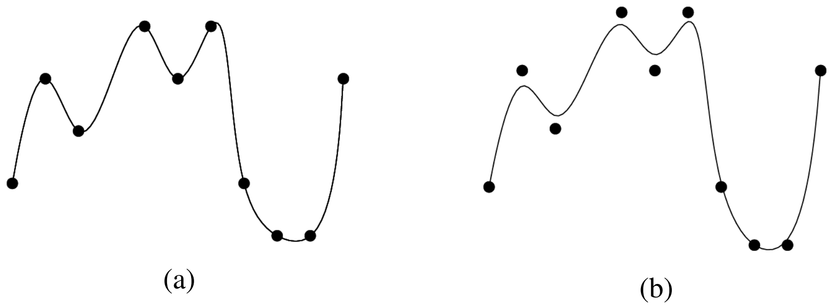


Figure 3.2: Interpolation (a) and approximation (b) of a set of way points [25].

- When the goal is to capture the general ”shape” of the curve defined by the points (as seen in Figure 3.2) but avoid fast oscillations (those that create undesirable behavior in the curvature and acceleration profiles) between adjacent points.

One common interpolation method used to satisfy the second reason above is *smoothing B-splines*, since this method reduces the curvature and acceleration along the trajectory [25].

3.3.3 Path Approximation with Bézier Curves

A standard approach for approximating a set of via-points is to use LIPB in which the polynomial blends are either 4th or 5th order Bézier curves. As shown in Figure 3.3, given a desired point q_k to be approximated with tolerance δ when blending two linear segments, a blending Bézier curve can be constructed as follows [25]:

1. Additional points q'_k and q''_k are found by determining the intersection of the lines $\overline{q_{k-1}\,q_k}$, $\overline{q_k\,q_{k+1}}$, and a sphere of radius δ centered about q_k.

For the 4th order case:

2. The unit tangent vectors $\hat{\mathbf{t}}$ and curvature vectors $\hat{\mathbf{n}}$ are defined as:

$$\hat{\mathbf{t}}_{0,k} = \hat{\mathbf{n}}_{0,k} = \frac{q_k - q'_k}{\delta}, \quad \hat{\mathbf{t}}_{4,k} = \hat{\mathbf{n}}_{4,k} = \frac{q''_k - q_k}{\delta} \tag{3.1}$$

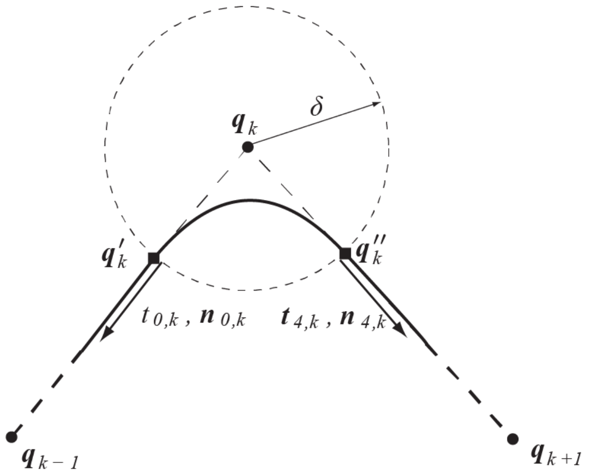


Figure 3.3: Linear interpolation with a polynomial blend [25].

3. We desire to interpolate the two additional points q'_k and q''_k and thus set the Bézier control points to be:

$$p_{0,k} = q'_k, \quad p_{4,k} = q''_k \tag{3.2}$$

4. Imposing the condition that the first and second derivatives of the Bézier curve are oriented along the tangent and curvature vectors at the endpoints, respectively, yields the following set of equations:

$$a\alpha_k^2 + b\alpha_k + c = 0$$
$$\text{where} \quad \begin{cases} a = 4 - \frac{1}{4}|\hat{\mathbf{t}}_{4,k} + \hat{\mathbf{t}}_{0,k}|^2 \\ b = 3(p_{4,k} - p_{0,k})^T \cdot (\hat{\mathbf{t}}_{0,k} + \hat{\mathbf{t}}_{4,k}) \\ c = -9|p_{4,k} - p_{0,k}|^2 \end{cases} \tag{3.3}$$

5. The largest value of α_k satisfying the quadratic equation is considered.

6. Finally, the Bézier curve is defined by the control points p_k, which can be transformed

into a standard polynomial form.

$$b_k(u) = a_{0,k} + a_{1,k}u + a_{2,k}u^2 + a_{3,k}u^3 + a_{4,k}u^4, \quad 0 \leq u \leq 1$$

$$\begin{cases} p_{0,k} = q'_k \\ p_{1,k} = q'_k + \frac{1}{4}\alpha_k \hat{\mathbf{t}}_{0,k} \\ p_{2,k} = q_k \\ p_{3,k} = q''_k - \frac{1}{4}\alpha_k \hat{\mathbf{t}}_{4,k} \\ p_{4,k} = q''_k \end{cases} \Rightarrow \begin{cases} a_{0,k} = p_{0,k} \\ a_{1,k} = -4p_{0,k} + 4p_{1,k} \\ a_{2,k} = 6p_{0,k} - 12p_{1,k} + 6p_{2,k} \\ a_{3,k} = -4p_{0,k} + 12p_{1,k} - 12p_{2,k} + 4p_{3,k} \\ a_{4,k} = 1p_{0,k} - 4p_{1,k} + 6p_{2,k} - 4p_{3,k} + 1p_{4,k} \end{cases} \tag{3.4}$$

For the 5th order case:

The 5th order case follows the same steps in derivation and results in the following equations:

2. The unit tangent vectors $\hat{\mathbf{t}}$ and curvature vectors $\hat{\mathbf{n}}$ are given by:

$$\hat{\mathbf{t}}_{0,k} = \hat{\mathbf{n}}_{0,k} = \frac{q_k - q'_k}{\delta}, \quad \hat{\mathbf{t}}_{5,k} = \hat{\mathbf{n}}_{5,k} = \frac{q''_k - q_k}{\delta} \tag{3.5}$$

3. Points to be interpolated are given by:

$$p_{0,k} = q'_k, \quad p_{5,k} = q''_k \tag{3.6}$$

4. After applying constraints in a manner similar to the 4th order case, we get:

$$a\alpha_k^2 + b\alpha_k + c = 0$$
$$\text{where} \quad \begin{cases} a = 256 - 49|\hat{\mathbf{t}}_{5,k} + \hat{\mathbf{t}}_{0,k}|^2 \\ b = 420(p_{5,k} - p_{0,k})^T \cdot (\hat{\mathbf{t}}_{0,k} + \hat{\mathbf{t}}_{5,k}) \\ c = -900|p_{5,k} - p_{0,k}|^2 \end{cases} \tag{3.7}$$

5. The largest value of α_k satisfying the quadratic equation is considered.

6. The resulting Bézier curve is:

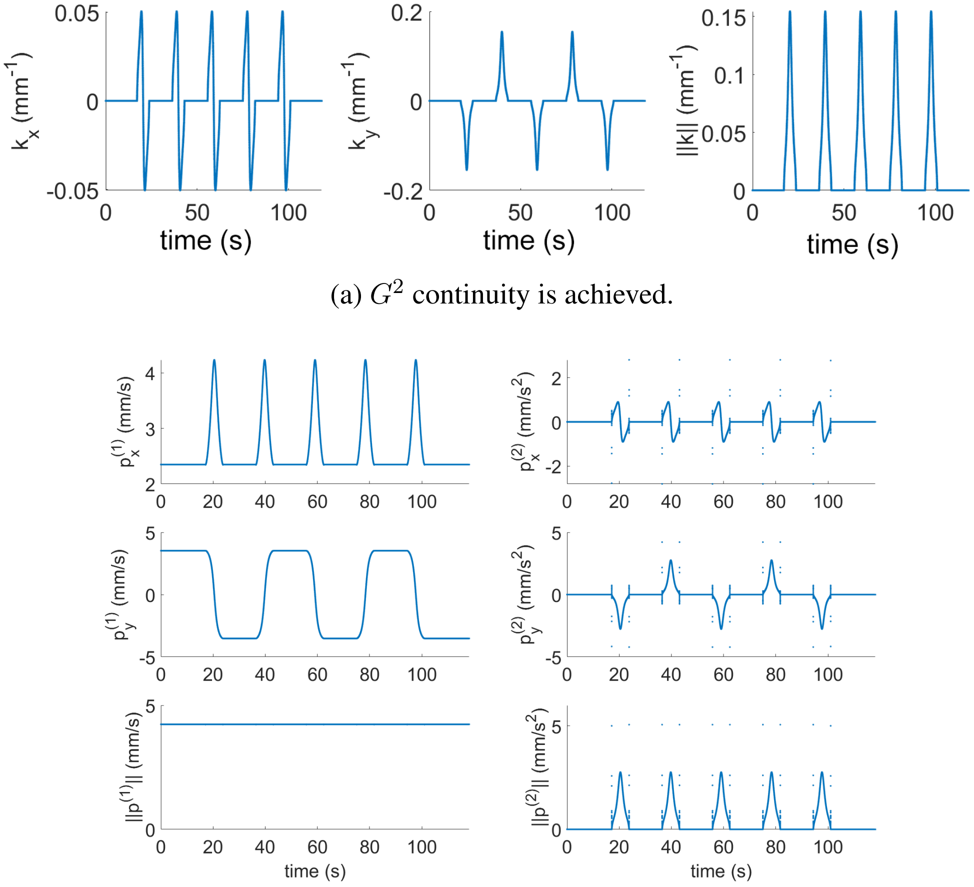
$$b_k(u) = a_{0,k} + a_{1,k}u + a_{2,k}u^2 + a_{3,k}u^3 + a_{4,k}u^4 + a_{5,k}u^5, \quad 0 \leq u \leq 1$$

$$\begin{cases} p_{0,k} = q'_k \\ p_{1,k} = p_{0,k} + \frac{\alpha_k}{5}\hat{\mathbf{t}}_{0,k} \\ p_{2,k} = 2p_{1,k} - p_{0,k} \\ p_{5,k} = q''_k \\ p_{4,k} = p_{5,k} - \frac{\alpha_k}{5}\hat{\mathbf{t}}_{5,k} \\ p_{3,k} = 2p_{4,k} - p_{5,k} \end{cases} \Rightarrow \begin{cases} a_{0,k} = p_{0,k} \\ a_{1,k} = -5p_{0,k} + 5p_{1,k} \\ a_{2,k} = 10p_{0,k} - 20p_{1,k} + 10p_{2,k} \\ a_{3,k} = -10p_{0,k} + 30p_{1,k} - 30p_{2,k} + 10p_{3,k} \\ a_{4,k} = 5p_{0,k} - 20p_{1,k} + 30p_{2,k} - 20p_{3,k} + 5p_{4,k} \\ a_{5,k} = -p_{0,k} + 5p_{1,k} - 10p_{2,k} + 10p_{3,k} - 5p_{4,k} \\ \qquad + p_{5.k} \end{cases} \tag{3.8}$$

A 4th order Bézier curve obtained using the above method ensures a G^2 continuous trajectory, but the trajectory may not be C^2 continuous. Consider the triangle wave introduced in this chapter approximated with linear segments and 4th order Bézier curves. Figure 3.4a and Figure 3.4b show an example of such a trajectory with G^2 continuity but only C^1 continuity. Figure 3.4b shows the velocity $p^{(1)}$ and acceleration vectors $p^{(2)}$ and a clear discontinuity can be seen in the acceleration profile. Figure 3.4a shows continuity of the components of the curvature vector $\mathbf{k}$ with magnitude κ (often referred to simply as *curvature*).

With a 5th order Bézier curve, it is possible to define a C^2 continuous trajectory. Again, consider the triangle wave introduced this chapter but this time approximated with linear segments and 5th order Bézier curves. Figure 3.5a and Figure 3.5b show an example triangle wave trajectory with G^2 and C^2 continuity.

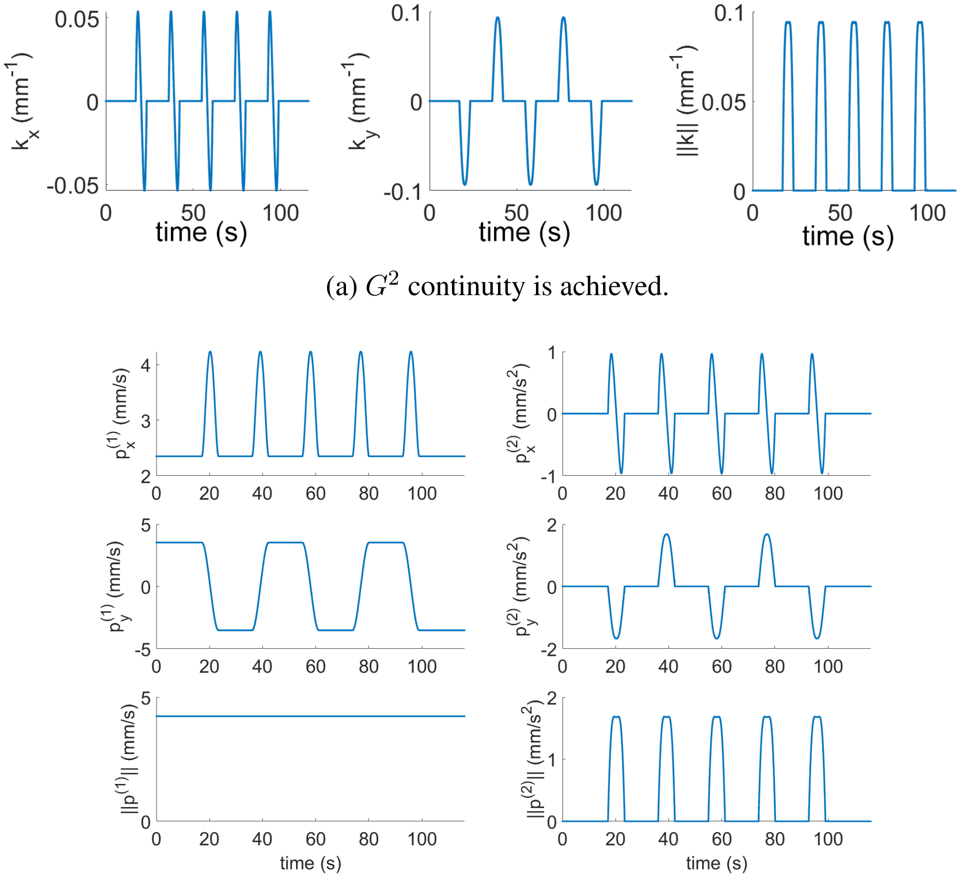
While this method achieves the goal of both G^2 and C^2 continuity, alternative curves can be used in blending the linear sections to achieve different characteristics, such as in



(a) G^2 continuity is achieved.

(b) C^2 continuity is not achieved (C^1 continuous).

Figure 3.4: G and C continuity of linear segments blended with 4th order Bézier curves



(a) G^2 continuity is achieved.

(b) C^2 continuity is achieved.

Figure 3.5: G and C continuity of linear segments blended with 5th order Bézier curves

the curvature, acceleration, and jerk profiles. Thus, an alternate method for creating a curve that achieves G^2 and C^2 is proposed in this thesis.

3.3.4 Path Approximation with a Varying-radius Curve

Circular corners are common in manufacturing for a variety of reasons, including stress distribution, part aesthetics, and manufacturing limitations. A circular corner between two linear segments also allows for the tool to change directions without stopping and is a basic movement type found in many CNC machine tools and robot manipulators. However, a transition from a linear segment to a circular segment necessarily is G^2 discontinuous as the path instantaneously changes from zero curvature to non-zero curvature. Approximating the triangle wave with circular corners of radius 12.7 mm (0.5 in) yields the continuity plots seen in Figure 3.6. Note that the 2-norm of the curvature vector, also denoted as curvature κ, for a circle is the reciprocal of the radius, $\kappa = 1/R$.

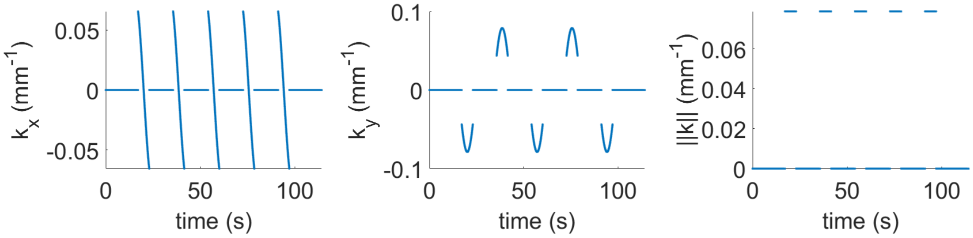


Figure 3.6: G^2 discontinuity of linear segments with circular corners.

As noted earlier, G^2 continuity is desirable in robot motion planning since it is a necessary condition to ensure C^2 continuity [25]. Discontinuities in acceleration (in the case of a C^2 discontinuous trajectory) create an unbounded jerk that can excite undesirable robot vibrations. Thus, a pseudo-circular corner with radius varying as a function of angle $r(\theta)$ is proposed in this thesis.

Consider a function $r(\theta)$ where $0 \leq \theta \leq \theta_f$. As the discontinuous curvature of linear segments with circular blends creates a similarly discontinuous radial acceleration, we want $r(\theta)$ such that the start and end points of the blending curve have continuous radial acceleration at the blend points with the linear segments; that is, $a_r(0) = a_r(\theta_f) = 0$. Note that θ is a function of time t.

To impose this requirement on the proposed varying-radius function, we derive an equation for acceleration defined in cylindrical coordinates as follows:

$$\begin{aligned}
\vec{r} &= r\,\hat{e}_r + z\,\hat{e}_z && (3.9)\\
\vec{v} &= \dot{\vec{r}} \\
&= \dot{r}\,\hat{e}_r + r\,\dot{\hat{e}}_r + \dot{z}\,\hat{e}_z \\
&= \dot{r}\,\hat{e}_r + r\dot{\theta}\,\hat{e}_\theta + \dot{z}\,\hat{e}_z && (3.10)\\
\vec{a} &= \dot{\vec{v}} \\
&= \ddot{r}\,\hat{e}_r + \dot{r}\,\dot{\hat{e}}_r + \dot{r}\dot{\theta}\,\hat{e}_\theta + r\ddot{\theta}\,\hat{e}_\theta + r\dot{\theta}\,\dot{\hat{e}}_\theta + \ddot{z}\,\hat{e}_z \\
&= \ddot{r}\,\hat{e}_r + \dot{r}\dot{\theta}\,\hat{e}_\theta + \dot{r}\dot{\theta}\,\hat{e}_\theta + r\ddot{\theta}\,\hat{e}_\theta - r\dot{\theta}\dot{\theta}\,\hat{e}_r + \ddot{z}\,\hat{e}_z \\
&= (\ddot{r} - r\dot{\theta}^2)\,\hat{e}_r + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\,\hat{e}_\theta + \ddot{z}\,\hat{e}_z && (3.11)
\end{aligned}$$

Setting the radial acceleration term to zero, we get:

$$a_r = (\ddot{r} - r\dot{\theta}^2) = 0 \tag{3.12}$$

In addition, the varying radius curve is defined such that:

- $r(\theta)$ is symmetric about the halfway angle $\theta_f/2$
- The curve starts and ends at the same point as a circle of radius r_0. That is, $r(0) = r(\theta_f) = r_0$

This yields the following varying-radius function of θ:

$$r(\theta(t)) := A\,\theta^2(\theta - \theta_f)^2 + r_0, \qquad 0 \le \theta \le \theta_f \tag{3.13}$$

where A is the fitting constant, θ_f is the absolute value of the range of angles (or the angle corresponding to the end point of the blend and hence the start point of the second linear segment), and r_0 is the nominal (minimum) radius. Since the function is symmetric about the center angle, $\theta_f/2$, enforcing the desired constraint on the start point will also enforce the constraint at the end point of the blend. Taking the time derivatives:

$$\dot{r}(\theta(t)) = 2A\theta(\theta_f - \theta)^2\dot{\theta} - 2A\theta^2(\theta_f - \theta)\dot{\theta} \tag{3.14}$$

$$\begin{aligned}\ddot{r}(\theta(t)) &= 2A\theta^2\dot{\theta}^2 + 2A(\theta_f - \theta)^2\dot{\theta}^2 + 2A\theta(\theta_f - \theta)^2\ddot{\theta} \\ &\quad - 2A\theta^2(\theta_f - \theta)\ddot{\theta} - 8A\theta(\theta_f - \theta)\dot{\theta}^2\end{aligned} \tag{3.15}$$

Substituting Equation 3.14 and Equation 3.15 back into the expression for radial acceleration (Equation 3.12):

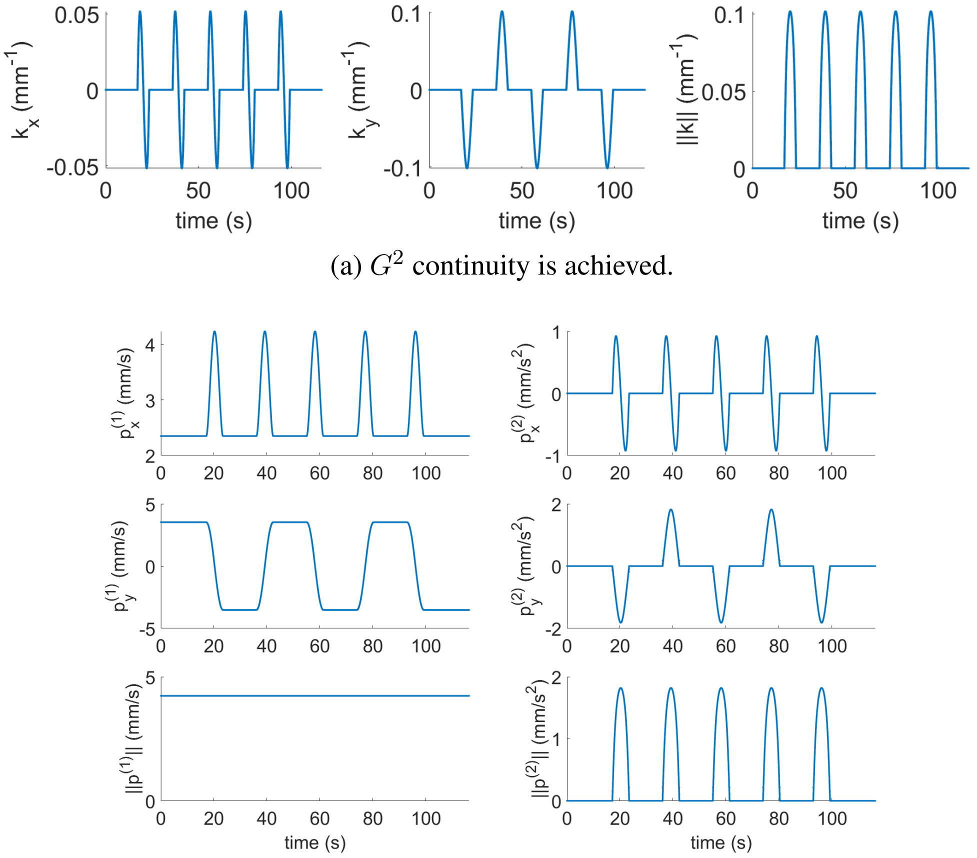
$$\begin{aligned}a_r &= 2A\theta^2\dot{\theta}^2 - r_0\dot{\theta}^2 + 2A\dot{\theta}^2(\theta - \theta_f)^2 + 8A\theta\dot{\theta}^2(\theta - \theta_f) \\ &\Rightarrow A = \frac{r_0\dot{\theta}^2 + a_r}{12\theta^2\dot{\theta}^2 - 12\theta\theta_f\dot{\theta}^2 + 2\theta_f^2\dot{\theta}^2}\end{aligned} \tag{3.16}$$

Enforcing $a_r = 0$ at $\theta = 0$ gives $A = \frac{r_0}{2\theta_f^2}$ and

$$r(\theta(t)) := \left(\frac{r_0}{2\theta_f^2}\right)\theta^2(\theta - \theta_f)^2 + r_0, \qquad 0 \le \theta \le \theta_f \tag{3.17}$$

Using this method for path approximation with the timing law presented later in Section 3.5, it can be seen in Figure 3.7a and Figure 3.4b that both G^2 and C^2 continuity can be achieved.

A comparison of the circular blend, the 4th order Bézier curve blend, the 5th order Bézier curve blend, and the varying-radius curve blend derived above is shown in Figure 3.8. The path taken by the 5thorder Bézier curve and the varying-radius curve are similar, but not identical, which is supported by the similar curvature vectors seen in Figure 3.5a and Figure 3.7a. Furthermore, the jerk profiles are different, with the varying-radius curve



(a) G^2 continuity is achieved.

(b) C^2 continuity is achieved.

Figure 3.7: G and C continuity of linear segments with varying-radius blends.

blend exhibiting a lower maximum jerk and a smaller range of jerk values, as shown in Figure 3.9.

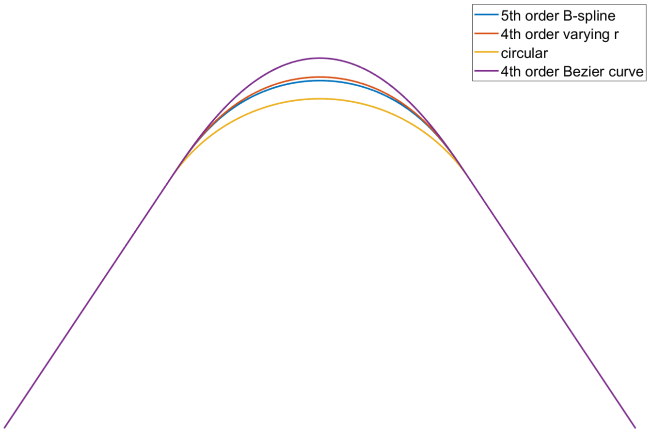
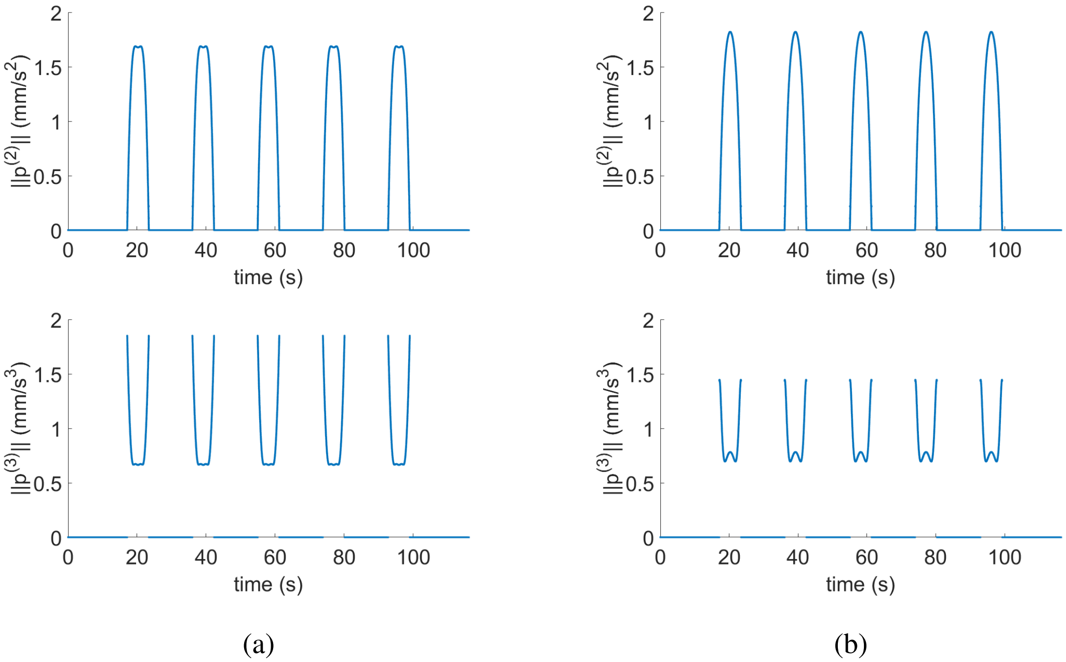


Figure 3.8: Comparison of the different linear segment blending methods.

3.4 Milling Force and Robot Deflection Prediction

Assuming a constant, nominal feed rate specified by the user, milling forces must first be predicted to vary the feed rate to satisfy the prescribed deflection limit for the milling cut. Milling forces are predicted using a well-established static mechanistic milling force model described in [39] and summarized in Algorithm 1. A schematic of the milling process as modeled in the milling force model is shown in Figure 3.10. Additionally, an example of the simulated static milling forces in a half-immersion down milling cut are shown in Figure 3.11. For the purposes of this thesis, the force model assumes the tool and workpiece are perfectly rigid relative to the compliance of the robot. In addition, the effects of tool run-out are not modeled, although this can be accounted for as discussed elsewhere in literature [40].

Since the end milling tool, workpiece, and spindle are assumed to be rigid, the path accuracy due to process forces is a direct function of the instantaneous static deflection of the robot along the path. From the known pose of the robot at each point along the milling



(a)

(b)

Figure 3.9: Magnitudes of the acceleration and jerk for the (a) 5th order Bézier curve and (b) varying-radius curve blended paths.

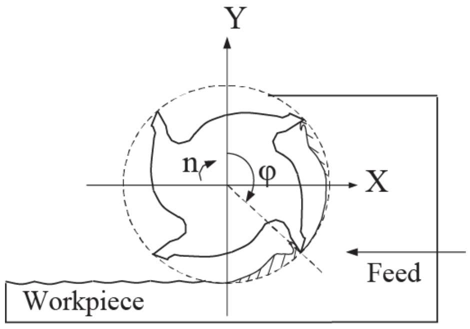


Figure 3.10: A schematic of a down milling cut as modeled in the milling force model [39].

Algorithm 1 Pseudo-code for Milling Force Simulation [39].

```
Inputs
    Cutting conditions                          $a, c, n, \varphi_{st}, \varphi_{ex}$
    Tool geometry                               $D, N, \beta$
    Cutting constants                           $K_{tc}, K_{rc}, K_{te}, K_{re}$
    Integration angle                           $\Delta\phi$
    Integration height                          $\Delta a$
Outputs
    Cutting force history                       $F_x(\varphi), F_y(\varphi), F(\varphi)$
    Cutting torque and power history            $T_c(\varphi), P_c(\varphi)$
Variables
      $\varphi_p = \frac{2\pi}{N}$                              Cutter pitch angle
      $K = \frac{2\pi}{\Delta\phi}$                             Number of angular integration steps
      $L = \frac{a}{\Delta a}$                                  Number of axial integration steps
    for $i = 1$ to $K$                                          Angular integration loop
      $\varphi(i) = \varphi_{st} + i\Delta\phi$                 Immersion angle of flute's bottom edge
      $F_x(i) = F_y(i) = F_t(i) = 0.$                           Initialize the force integration registers
      for $k = 1$ to $N$                                        Calculate the force contributions of all teeth
        $\varphi_1 = \varphi(i) + (k-1)\varphi_p$               Immersion angle for tooth k
        $\varphi_2 = \varphi_1$                                 Memorize the present immersion
        for $j = 1$ to $L$                                      Integrate along the axial depth of cut
          $a(j) = j \cdot \Delta a$                             Axial position
          $\varphi_2 = \varphi_1 - \frac{2\tan\beta}{D}a(j)$    Update the immersion angle due helix
          if $\varphi_{st} \leq \varphi_2 \leq \varphi_{ex}$    If the edge is cutting,
                                                                then
            $h = c\sin\varphi_2$                                Chip thickness at this point
            $\Delta F_t = \Delta a(K_{tc}h + K_{te})$           Differential tangential force
            $\Delta F_r = \Delta a(K_{rc}h + K_{re})$           Differential radial force
            $\Delta F_x = -\Delta F_t\cos\varphi_2 - \Delta F_r\sin\varphi_2$   Differential feed force
            $\Delta F_y = \Delta F_t\sin\varphi_2 - \Delta F_r\cos\varphi_2$    Differential normal force
            $F_x(i) = F_x(i) + \Delta F_x$                      Sum the cutting forces contributed by all active
            $F_y(i) = F_y(i) + \Delta F_y$                        edges
            $F_t(i) = F_t(i) + \Delta F_t$
          else
        next $j$
      next $k$                                                  Resulting cutting force values at immersion angle
                                                                  $\varphi(i)$
      $F(i) = \sqrt{F_x^2(i) + F_y^2(i)}$                       Resultant cutting force
      $T_c(i) = \frac{D}{2}F_t(i)$                              Cutting torque
    next $i$
    Plot $F_x(i), F_y(i), F_i, T_c(i)$ with varying immersion $\varphi(i)$
stop
end
```

Description of Inputs:

axial depth of cut a, feed per tooth c, spindle speed n, cutter entry and exit angles ϕ_{st}, ϕ_{ex} cutter diameter D, number of teeth N, helix angle β

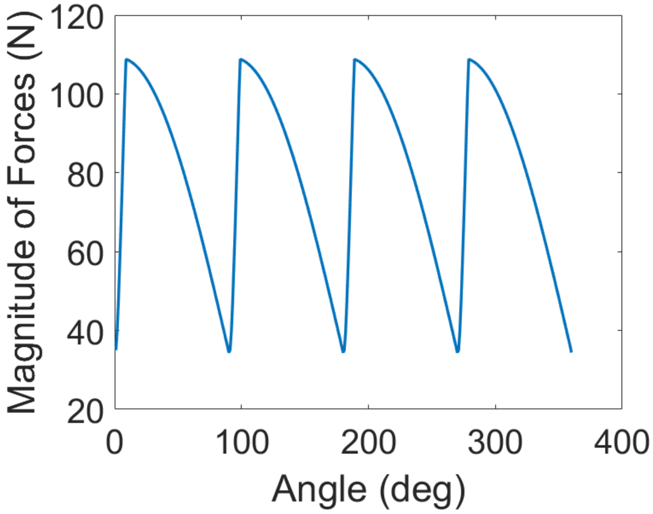


Figure 3.11: Simulated resultant force for a half-immersion, down milling cut in 6061 Al. *Note: FR = 4.23 mm/s (10 IPM), D = 19.05 mm (0.75"),* $\beta = 30°$*, N = 4 flutes, n = 2000 RPM, a = 2.54 mm (0.1")*

path, the joint configuration corresponding to each point is found using numerical inverse kinematics and the solution closest to the known initial joint configuration is chosen for determining the corresponding Cartesian stiffness of the robot.

To determine the static deflection experienced by the robot at a given pose, a static joint stiffness model is required. To this end, each joint is modeled as a linear torsional spring where all compliance is due to the rotational compliance of the individual joint, known as a six parameter model. This is preferable as it entails a simpler experimental calibration and does not require explicit knowledge of the friction and structural properties of the robot links [13]. The model is calibrated using both the Complete Pose (CP) method [17] and the Decoupled Partial Pose (DPP) method [41], which follow the same experimental approach but DPP decouples the first three (arm) and last three (wrist) joints.

With the Jacobian $J(\theta)$, which defines the relationship between end effector wrenches and velocity to joint torques and velocity respectively, the deflection of the end effector ΔX at a given robot configuration subject to a given milling force F can be found [17].

$$\Delta X = J(\theta)(K_\theta - K_c)^{-1}J(\theta)^T F \tag{3.18}$$

where K_θ is the diagonal joint stiffness matrix and K_c is the complementary stiffness matrix, which captures the change in geometry under external loads [42]. The experiments in this thesis are run in a region of the workspace where the configurations are far from singularity. When the robot is far from singularity, i.e., the inverse of the condition number $\kappa(J^{-1}) = ||J^{-1}|| \cdot ||J||$ is large, K_c is negligible and the previous equation reduces to:

$$\Delta X = J(\theta) K_\theta^{-1} J(\theta)^T F \tag{3.19}$$

Equation 3.19 uses the CP method and considers all six joints together. In contrast, the DPP method separates the arm and wrist joint stiffnesses, resulting in Equation 3.20.

$$\Delta X = J(\theta) K_{\theta_{123}}^{-1} J(\theta)^T F + J(\theta) K_{\theta_{456}}^{-1} J(\theta)^T F \tag{3.20}$$

This thesis utilizes the DPP method for calculating robot deflection from this point on. With static deflection predicted, the feed rate of the trajectory can now be varied to satisfy the user-specified deflection limit.

3.5 Motion Timing Law

As the robot moves along the curvilinear path defined earlier, the static deflection of the robot end effector due to the milling forces varies with robot stiffness, which changes continuously with robot pose along the path. Note that the maximum predicted milling force per revolution along the linear sections of the path stays constant. In contrast, as the robot changes direction around a corner, the changing forces acting on the robot significantly impacts the deflection of the robot as it is stiffer in some directions than others. However, by limiting the feed rate around the corner segments (while maintaining C^2 continuity), the magnitude of the deflection during the corner is constrained to be less than the maximum of the magnitudes at the end and start of the linear segments before and after the curve, respectively.

Thus, it is only necessary to explicitly determine the feed rate for the linear segments of the path, which are at constant velocity. For the linear segments, the feed rate is reduced iteratively until the magnitude of the maximum deflection experienced by the robot end effector in the linear segment is less than or equal to the user-specified deflection limit. Then, a feed rate profile for the corner sections must then be chosen such that C^2 continuity is maintained and the deflection-limit is not violated.

While changes to the feed rate for the linear segments do not alter the geometric path (and the G^2 continuity of the path), the velocity, and thus acceleration, or C^2 continuity, of the path is not ensured at the blend points. To ensure C^2 continuity at the blend points, an arc length s based parameterization of the varying-radius segments is used. Note that if a different parameterization of the varying-radius curve, say of θ, were used, an undesirable behavior of the feed rate profile may arise as the radius is non-constant and the radius r and angle θ change at different rates.

A 5th order polynomial motion timing law defined in terms of the arc length s is used for the curved segment:

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{3.21}$$

Six constraints on the displacement s, tangential velocity (or feed rate) $\dot{s}$, and tangential acceleration $\ddot{s}$ are imposed on the start and end points of the varying-radius curve segment to ensure C^2 continuity of the path. The constraint values are chosen to be the same as the adjacent linear sections, thereby enforcing continuity. Assuming $t_i = 0$, the constraints are as follows:

$$
\begin{aligned}
s(0) &= s_0 = a_0 \\
s(t_f) &= s_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\
\dot{s}(0) &= \dot{s}_0 = a_1 \\
\dot{s}(t_f) &= \dot{s}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\
\ddot{s}(0) &= \ddot{s}_0 = 2a_2 \\
\ddot{s}(t_f) &= \ddot{s}_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3
\end{aligned}
\tag{3.22}
$$

Where

$s_0 = 0$

$s_f =$ the length of the path

$\dot{s}_0 =$ the velocity at the end of the previous, linear section

$\dot{s}_f =$ the velocity at the beginning of the following, linear section

$\ddot{s}_0 = 0$

$\ddot{s}_f = 0$

The six coefficients a_i can then be found by solving the linear system, where t_f is chosen based on the time it would take to traverse the curve at a constant velocity given by $min(\dot{s}_0, \dot{s}_f)$. We now have a geometric description of the varying-radius curve $r(\theta)$ (as derived in the previous section) and the motion timing law $s(t)$. To implement the trajectory defined by these two quantities, a functional relationship or mapping between θ and s is required.

The arc length of a curve is given by $L = \int ds$ which in polar form, for the varying-radius curve, is given by:

$$
\begin{aligned}
L &= \int \sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2}\, d\theta \\
s(\theta) &= \int_0^{\theta} \sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2}\, d\theta
\end{aligned}
\tag{3.23}
$$

Since the closed form of Equation 3.23 may not exist, it needs to be computed numerically. It follows then that a closed form of the functional inverse $\theta(s)$ may not exist. Thus, from the numerically computed values of $s(\theta), \quad 0 \leq \theta \leq \theta_f$, the functional inverse $\theta(s)$ is found by interpolating the appropriate value of θ for s specified by the 5th order polynomial $s(t)$. In other words, the varying-radius curve is now parameterized as $r(\theta(s(t)))$. The varying-radius curve, specified in cylindrical coordinates, can easily be converted back to Cartesian coordinates. (i.e., $x = r\ cos\,\theta$, $y = r\ sin\,\theta$.)

With the linear and curved segments reconnected, C^2 continuity can be confirmed for the triangle wave example, as shown in Figure 3.12.

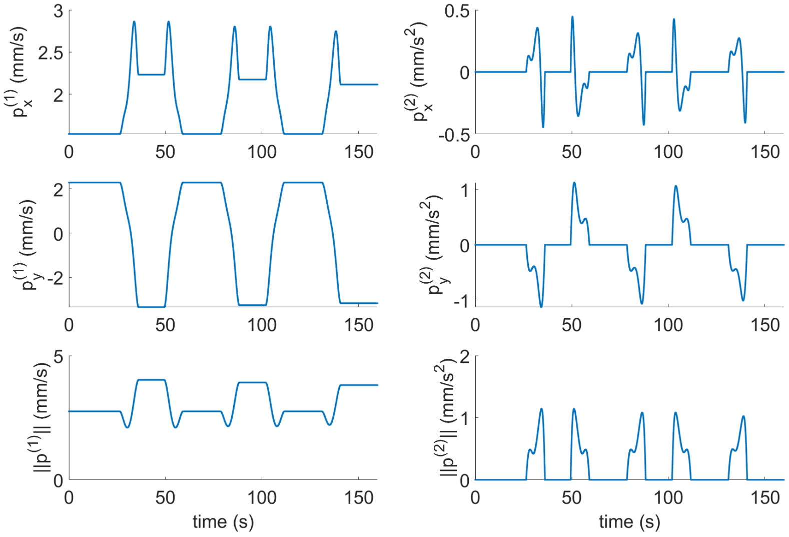


Figure 3.12: C^2 continuity of the varying feed rate triangle wave trajectory.

3.6 Experimental Procedure

3.6.1 Experimental Calibration

To utilize the milling force prediction and static deflection models, the cutting force coefficients ($K_{tc}, K_{rc}, K_{ac}, K_{te}, K_{re}, K_{ae}$) in the force model must first be calibrated and the robot joint stiffnesses K_θ established.

To obtain the cutting force coefficients, the mechanistic model calibration approach described in [39] was used. Linear slotting cuts were performed on 25.4 mm (1 in) thick 6061 aluminum with an axial depth of cut of 1.27 mm (0.05 in) using a 2 flute, 19.05 mm (0.75 in) mill diameter, Titanium Nitride (TiN) coated High-Speed Steel (HSS) square end mill (detailed specifications can be found in Table A.1 and Table A.2). The calibration cuts were performed on an Okuma MILLAC 44V 3-axis CNC milling machine with a Kistler Type 9257B Multi-Component Dynamometer mounted inside, as shown in Figure 3.13.

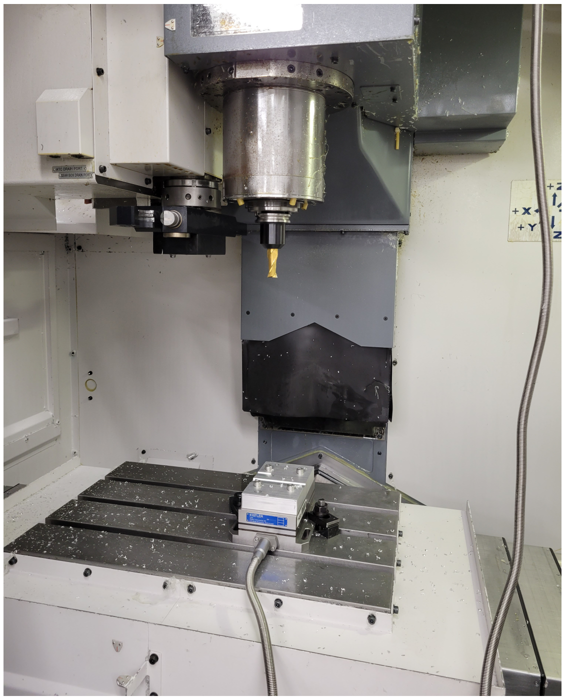


Figure 3.13: The dynamometer and calibration workpiece mounted inside the CNC machine.

The speeds and feeds used in the cutting coefficient calibration experiments are shown in Table 3.1 (Note that feed per tooth is the same between the different sets of spindle speeds used.)

Table 3.1: Speeds and feeds used for the cutting coefficient calibration experiments.

Speeds (RPM)	Feed Rates (mm/s)				Feed Rates (IPM)			
1000	0.64	1.69	2.96	4.23	1.5	4	7	10
2000	1.27	3.39	5.93	8.47	3	8	14	20
3000	1.59	4.23	7.41	10.58	3.75	10	17.5	25

The dynamometer data is processed using MATLAB, and the average force during full engagement is calculated. An example of one test (from the 2000 Revolutions per Minute (RPM) and 8.467 mm/s feed rate experiment) is shown in Figure 3.14.

The mechanistic model in [39] assumes a different coordinate system than the dynamometer coordinate system, as shown in Figure 3.15, and thus a transformation is applied to the dynamometer data and reversed in sign to determine the forces acting on the tool. The average forces experienced by the tool in the tool frame are shown in Table 3.2.

Table 3.2: Average forces experienced by the tool in the tool frame.

Speed (RPM)	Feed (mm/s)	Avg ${}^{T}F_{T,x}$ (N)	Avg ${}^{T}F_{T,y}$ (N)	Avg ${}^{T}F_{T,z}$ (N)
1000	0.64	-13.45	17.91	-6.66
1000	1.69	-21.97	37.88	-7.17
1000	2.96	-28.00	57.08	-10.71
1000	4.23	-31.20	74.57	-13.40
2000	1.27	-10.39	15.93	-4.12
2000	3.39	-17.52	33.97	-7.27
2000	5.93	-24.93	55.03	-10.38
2000	8.47	-32.40	74.63	-13.39
2500	1.59	-10.55	17.95	-3.17
2500	4.23	-18.65	37.03	-6.45
2500	7.41	-27.41	59.30	-9.50
2500	10.58	-38.384	80.994	-13.79

By applying a linear regression to the average cutting forces as a function of the feed per tooth c, the contribution from the edge forces can be separated as in Equation 3.24.

$$\overline{F}_q = \overline{F}_{qc}c + \overline{F}_{qe} \tag{3.24}$$

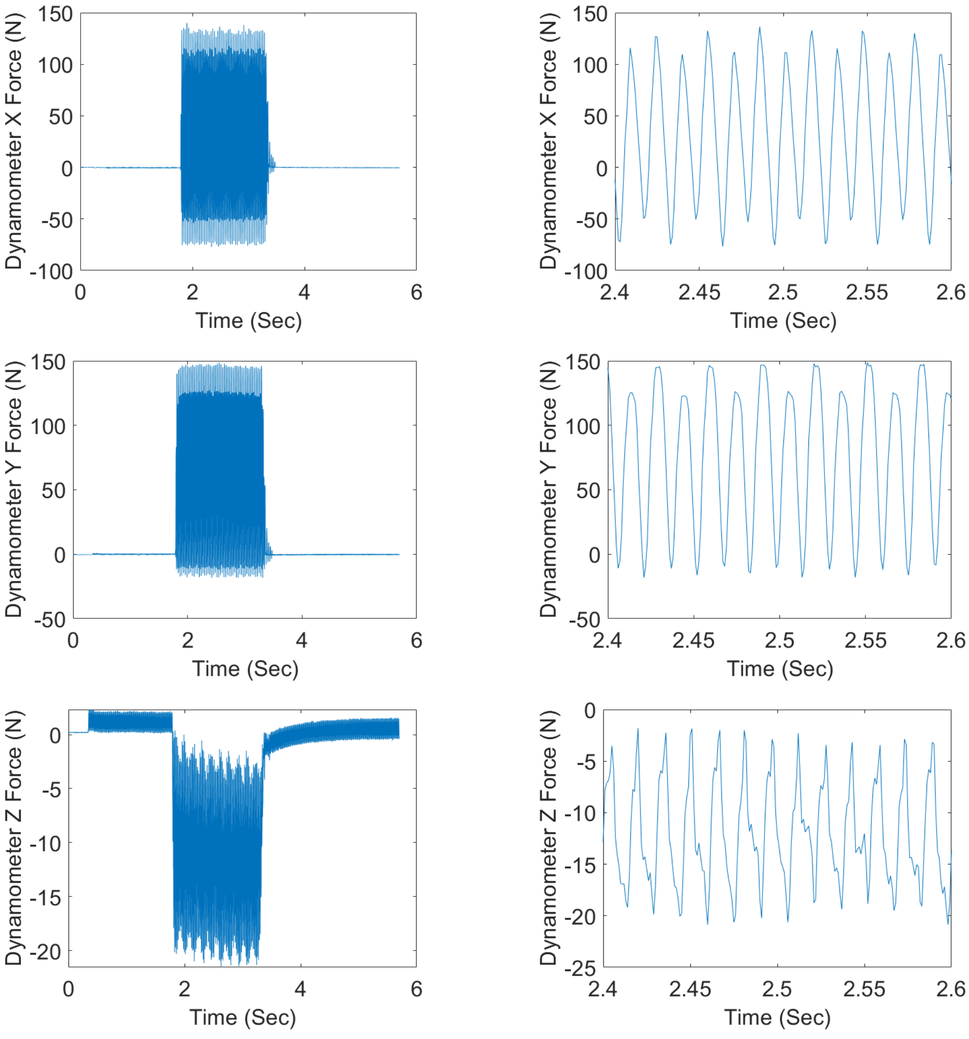


Figure 3.14: Force data measured by the dynamometer (a) in its entirety and (b) zoomed into several revolutions during full engagement.

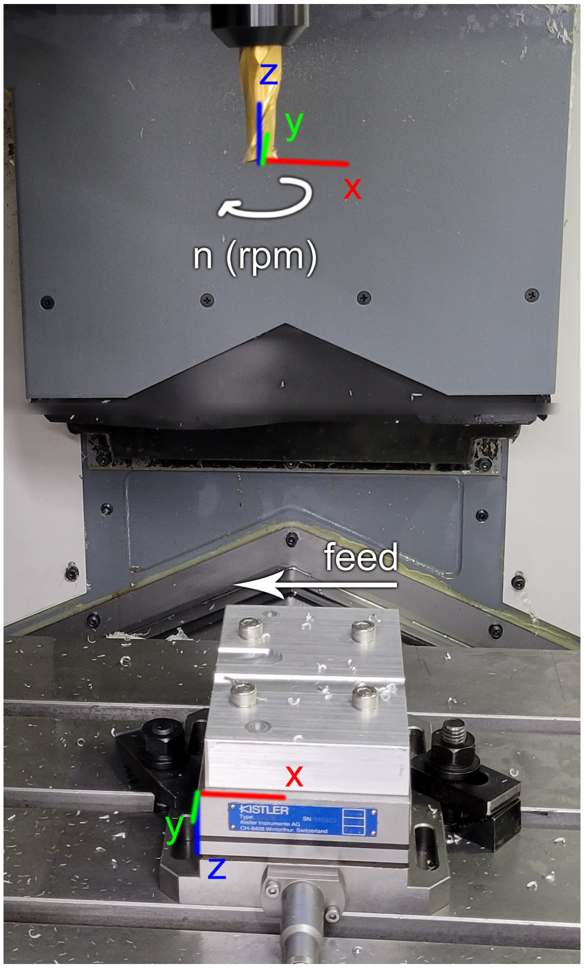


Figure 3.15: The coordinate system of the dynamometer and mechanistic model.

For N flutes and an axial depth of cut a, the cutting coefficients are as in Equation 3.25.

$$\begin{aligned} K_{tc} &= \frac{4\overline{F}_{yc}}{Na}, & K_{te} &= \frac{\pi\overline{F}_{ye}}{Na} \\ K_{rc} &= \frac{-4\overline{F}_{xc}}{Na}, & K_{re} &= \frac{-\pi\overline{F}_{xe}}{Na} \\ K_{ac} &= \frac{\pi\overline{F}_{zc}}{Na}, & K_{ae} &= \frac{2\overline{F}_{ze}}{Na} \end{aligned} \tag{3.25}$$

Table 3.3: Calibrated cutting coefficients for the tool-workpiece combination used in experiments.

Speed (RPM)	$K_{tc}\ \left(\frac{N}{m^2}\right)$	$K_{te}\ \left(\frac{N}{m}\right)$	$K_{rc}\ \left(\frac{N}{m^2}\right)$	$K_{re}\ \left(\frac{N}{m}\right)$	$K_{ac}\ \left(\frac{N}{m^2}\right)$	$K_{ae}\ \left(\frac{N}{m}\right)$
1000	8.21 e8	11.91 e3	2.55 e8	14.93 e3	-0.82 e8	-3.75 e3
2000	8.57 e8	7.42 e3	3.19 e8	8.45 e3	-1.05 e8	-2.13 e3
2500	9.19 e8	8.80 e3	4.02 e8	6.73 e3	-1.19 e8	-1.05 e3

The axial cutting coefficients K_a are negative because the model presented in [39] assumes that the axial force acts upwards in the z direction on the tool, in the tool frame. However, in the linear slotting cut performed, the workpiece is pulled upwards, exerting a downwards force on the tool. This is purely a matter of convention.

Note that we have assumed the force model coefficients to be constant during the evaluation experiments, as the calibration experiments capture the range of feeds per tooth used in the evaluation experiments. In actuality, the coefficients vary with the chip thickness, and thus the predicted cutting forces may be slightly inaccurate for the deflection-limited trajectory.

The joint stiffnesses for the industrial robot used in this work have been determined in previous work using the methodology presented in Section 3.4 and are listed in Table 3.4 [13]. The robot configurations used for the static stiffness calibration can be found in Table A.3. The stiffness values for each joint are taken to be constant, but the resulting Cartesian stiffness varies with the changing pose along the trajectory.

With the joint stiffnesses established, the assumption of the tool and spindle being rigid compared to the robot can be qualified. A Finite Element Analysis (FEA) was performed on a model of the tool and spindle, which yielded a minimum stiffness of 231 MN/m.

Table 3.4: Calibrated joint stiffness values $\left(\frac{MNm}{rad}\right)$ for the KUKA KR500-3.

Method	K_{θ_1}	K_{θ_2}	K_{θ_3}	K_{θ_4}	K_{θ_5}	K_{θ_6}
CP	3.93	3.46	2.77	1.57	0.24	0.17
DPP	2.86	2.36	1.67	0.24	0.16	0.14

In comparison, the maximum Cartesian stiffness of the robot in the configurations used for stiffness calibration was 12 MN/m, or an order of magnitude lower [13]. Thus, the assumption of a rigid tool and spindle is valid for this application.

3.6.2 Experimental Setup

The robotic milling system used in this thesis utilizes a 6-DoF industrial robot (KUKA KR500-3) with a milling spindle (Suhner Max 40B) and a laser tracker target, as shown in Figure 3.16. The KUKA KR500-3 has a rated maximum payload of 500 kg, pose repeatability of 0.08 mm, and a reach of 2825 mm [43]. The robot operates on the KUKA Robot Controller 4 (KRC4) software, which runs programs written in KRL, and paths can be executed natively within KRL. To incorporate external feedback from the 6-DoF laser tracker (Leica AT960), KUKA Robot Sensor Interface (RSI) is used to facilitate real-time corrections of the robot pose.

Mounted on the robot is a Suhner Max-40B milling spindle, which has a maximum power of 7.5 kW and maximum torque of 400 Nm [44], and a Tracker-Machine control sensor (T-Mac), which is used to determine the tool tip position. The 6-DoF pose of the T-Mac is determined by a Leica AT960 laser tracker, which can measure at a frequency of 1 kHz with an error of < 0.0254 mm [45]. We assume that the T-Mac is rigidly mounted to the spindle, and thus a constant transformation matrix relating the pose of the T-Mac to the tool tip can be defined.

An overview of the real-time control architecture is shown in Figure 3.17. The robot and a central computer are connected by and communicate via the EtherCAT protocol to minimize latency and ensure real-time communication [46]. The feedback control algo-

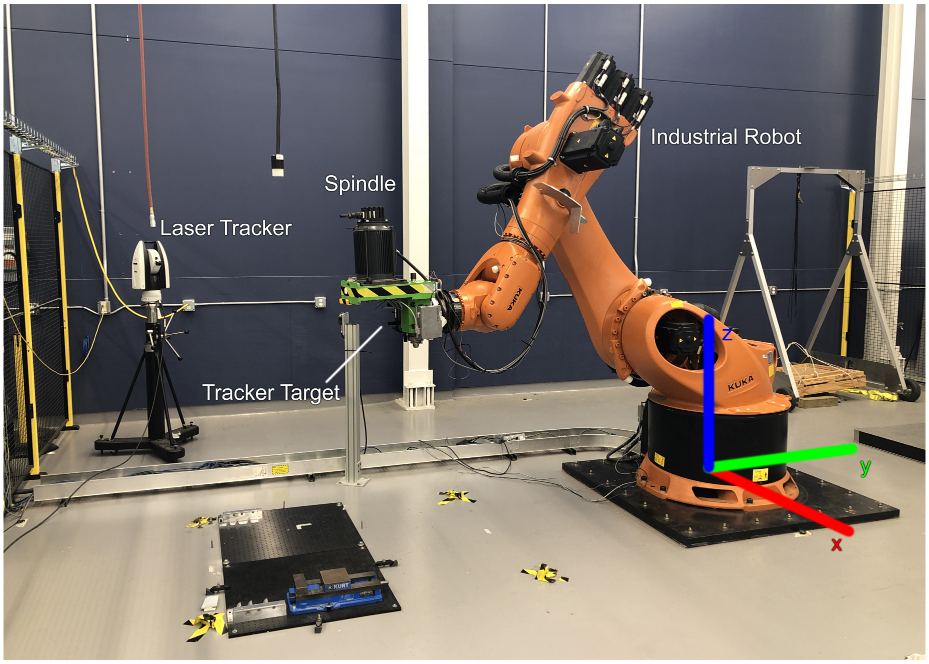


Figure 3.16: The robotic milling system used.

rithm is executed by the central computer, which is a Windows PC running TwinCAT C++, a real time software environment [47]. For purposes of this thesis, real-time is characterized by a strict adherence to issuing commands from the central computer to the robot controller within the system cycle time (4 ms for KUKA RSI).

After a start command is sent by the central computer to the robot, the laser tracker transmits its pose measurement of the T-Mac to the central computer, which uses a PID controller to compute desired error correction signals. These correction signals are sent from the central computer to the robot via RSI, which sends information, such as joint configuration, back from the robot controller. Within the central computer, values such as the tracker measurements and robot joint configuration are pulled from TwinCAT using the Automation Device Specification (ADS) protocol, which stores these values in a log file.

In Sensor Guided Motion (SGM), the commanded trajectory is stored on the central computer and motion commands are sent as Cartesian position corrections through RSI to the robot controller (KRC4). In other words, the robot has no future knowledge of the path. In contrast, in KRL guided motion, the robot controller (KRC4) commands the robot from

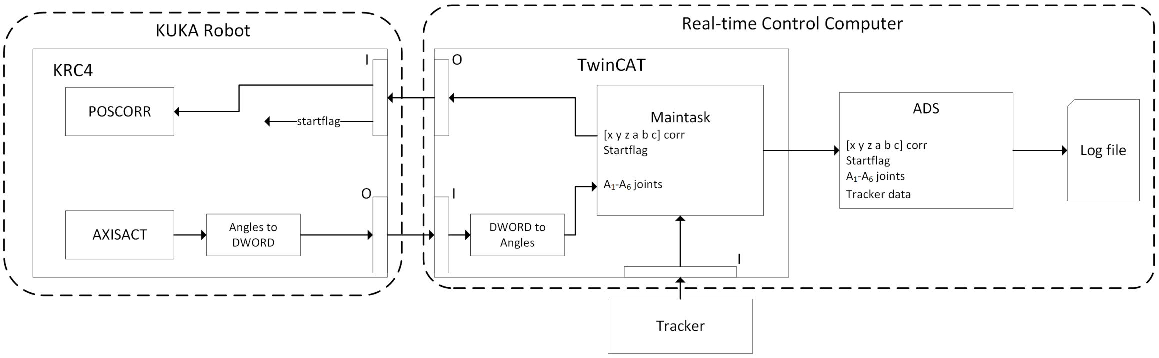


Figure 3.17: An overview of the real-time control architecture.

a trajectory directly uploaded to and stored on the robot controller.

To evaluate the performance of the deflection-limited trajectory planner presented earlier, approximated triangular paths of length 304.8 mm (12 in), height 76.2 mm (3 in), and three waves are cut into 25.4 mm (1 in) thick 6061 aluminum blocks with a 2 flute, 19.05 mm (0.75 in) diameter, TiN coated HSS square end mill. The cutter and workpiece are the same as those used in the calibration experiments. However, the workpiece length and width are now 406.4 mm (16 in) and 152.4 mm (6 in), respectively, to account for mounting hardware. The nominal feed rate is set to 4.23 mm/s (10 IPM) with a spindle speed of 2500 RPM. Three trajectories are evaluated:

1. The constant feed rate trajectory using the variable-radius path approximation.

2. The varying feed rate trajectory using the variable-radius path approximation.

3. Native path and trajectory planning capability of the KUKA KR500-3's KRC4 controller using the spline feature.

Trajectories 1 and 2 were run with closed loop, real-time control utilizing proportional and integral gains of 0.05 and 0.3, respectively, for corrections in the Cartesian directions, and a 0.02 integral gain for angular motion corrections. As derivative gains are sensitive to noise, and the industrial robot undergoes significant vibration during milling, the derivative gains were set to zero.

Using the trapezoidal rule for numerical integration, the discrete transfer function is given as:

$$\frac{U[z]}{E[z]} = K_p + K_i \frac{T_s}{2} \frac{z+1}{z-1} + K_d \frac{1}{T_s} \frac{z-1}{z} \tag{3.26}$$

which can be implemented as:

$$u[k] = u[k-1] + a\,e[k] + b\,e[k-1] + c\,e[k-2]$$
$$\text{where} \quad \begin{cases} a = K_p + K_i\,\frac{T_s}{2} + K_d\,\frac{1}{T_s} \\ b = -K_p + K_i\,\frac{T_s}{2} - K_d\,\frac{2}{T_s} \\ c = K_d\,\frac{1}{T_s} \end{cases} \tag{3.27}$$

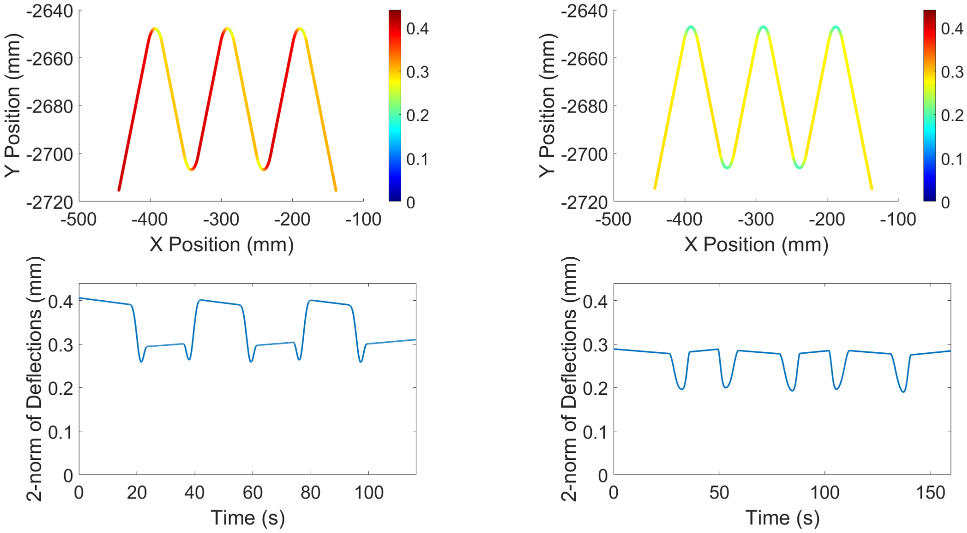
Note that $u[k]$ is the reference signal, $e[k]$ is the error signal, T_s is the sampling period, and K_p, K_i, K_d are the controller gains. Additionally, due to the black-box nature of the KUKA robot controller, we lack an accurate plant model of the robot. With tuned controller gains, the closed loop feedback is effective, but inaccuracies still exist in the executed trajectory.

3.7 Results and Analysis

The triangle wave trajectories discussed in this section are of the T-Mac referenced in the robot base frame, shown in Figure 3.16. The predicted static deflections of the constant and varying feed rate trajectories are shown in Figure 3.18. The deflection limit for the varying feed rate trajectory was set to 0.29 mm.

As expected, the robot deflections of the constant feed rate trajectory vary as the direction of milling force and the robot pose change. In the varying feed rate case, the deflection is reduced and far more consistent throughout the operation. To achieve this result, the feed rate is determined as shown in Figure 3.19.

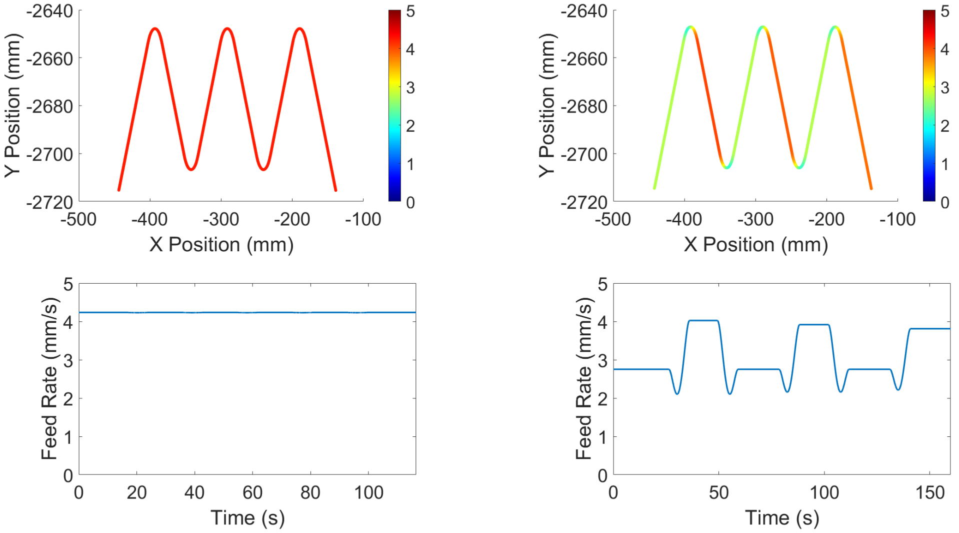
While the user-specified deflection limit can also be satisfied by simply traveling at a constant velocity equal to the minimum velocity in the varying feed rate trajectory, this can



(a)

(b)

Figure 3.18: Predicted static deflections of the (a) constant feed rate and (b) varying feed rate curvilinear trajectories.



(a)

(b)

Figure 3.19: Prescribed feed rate for the (a) constant feed rate and (b) varying feed rate curvilinear trajectories.

add a significant amount of time to the process. The varying feed rate trajectory satisfies a user-specified level of accuracy while allowing for the highest material removal rate possible. For example, as seen in Figure 3.19b, the downwards linear (-y) sections have a feed rate of 4.02 mm/s compared to the 2.75 mm/s of the upwards linear sections (+y) – 46.18% faster.

To confirm the commanded feed rate, the velocity was calculated from the laser tracker data and plotted. However, due to vibrations of the robot system, the data exhibits a lot of noise and was therefore filtered using a moving average window of 0.1 s and is shown in Figure 3.20.

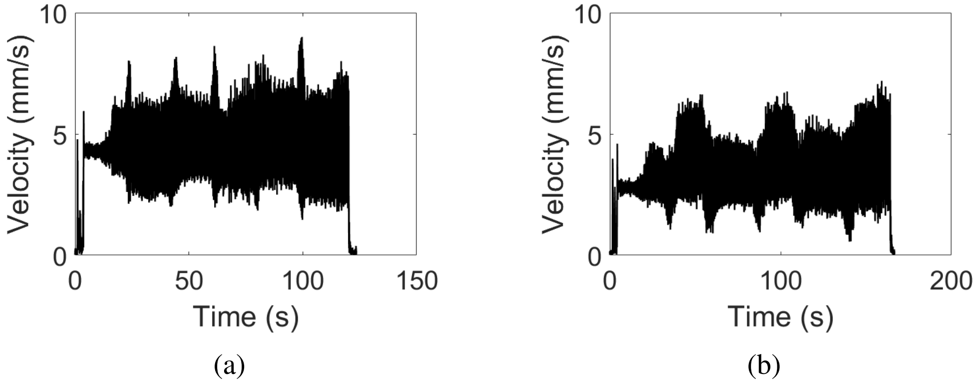


Figure 3.20: Measured velocities of the (a) constant feed rate and (b) varying feed rate curvilinear trajectories.

Despite the noise, the general trend of the commanded feed rates can be seen in the measured data. Additionally, despite the fact that the velocity was commanded as a constant 4.23 mm/s in the constant feed rate trajectory shown in Figure 3.20a, five spikes can be seen as the robot changes direction around the corners while milling. These spikes are not present in the varying feed rate trajectory as the robot stiffness variation has been accounted for when adjusting the feed rate.

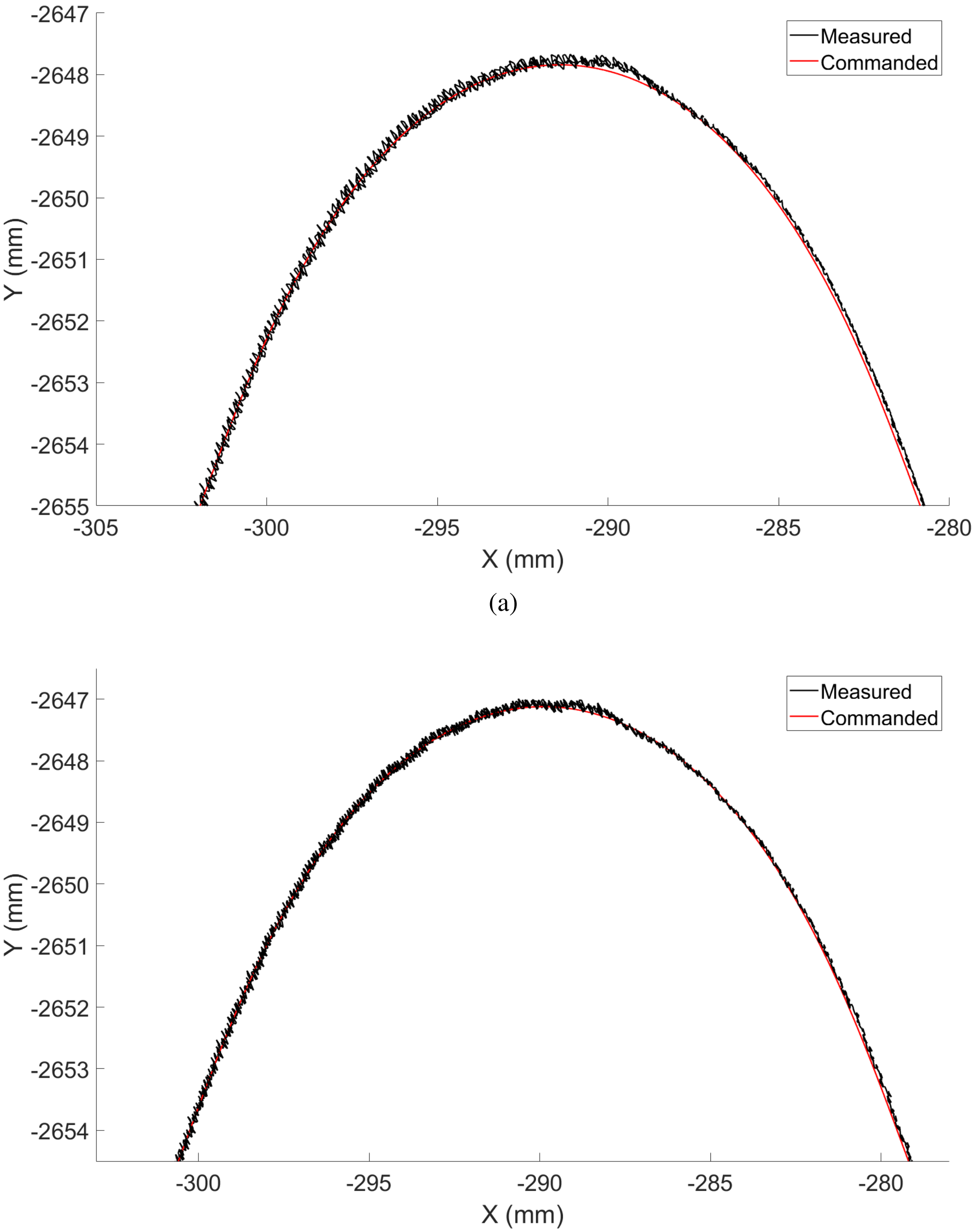
A close-up of the behavior around the corner, shown in Figure 3.21, reveals that the robot experiences greater vibrations in the first half of the corner (+y direction) than the second half (-y direction). This is consistent with the deflection (and thus stiffness as velocity is constant in the constant feed rate trajectory) predictions shown previously in Figure 3.18a. Because this is accounted for in the model, it can be seen that the vibrations in the first half of the corner of the constant feed rate trajectory are greater than those in the varying feed rate trajectory. Furthermore, in the second half of the corner, the constant feed rate trajectory overshoots the commanded trajectory slightly while the varying feed rate trajectory better tracks the commanded path, as shown in Figure 3.21. Specifically, a ”bump” can be seen at the top of the curve, and a small offset in the second half of the corner (-y direction).

To quantify the improvement in the machined part, shown in Figure 3.22, flatness measurements of the walls of the linear sections of the milled triangular paths were made with a Keyence XM-5000 Coordinate-Measuring Machine (CMM). Flatness is defined as the distance between two parallel planes that contains the measured surface. In a perfect scenario with zero deflections and no unmodeled effects, these sections should ideally be perfectly flat. The flatness measurements are shown in Table 3.5 as defined in Figure 3.23.

Table 3.5: Flatness measurements (mm) of the walls of the curvilinear slotting cuts.

	Seg. 3 (↓)	Seg. 5 (↑)	Seg. 7 (↓)	Seg. 9 (↑)	Seg. 11 (↓)
Constant feed rate, ~ 0.4 mm max deflection					
Upper Wall	0.136	0.081	0.080	0.102	0.132
Lower Wall	0.166	0.197	0.127	0.079	0.099
Varying feed rate, 0.29 mm deflection limit					
Upper Wall	0.083	0.072	0.092	0.062	0.118
Lower Wall	0.121	0.107	0.095	0.059	0.097
Improvement (constant $-$ varying case)					
Upper Wall	0.053	0.009	-0.012	0.040	0.014
Lower Wall	0.045	0.090	0.032	0.020	0.002

The improvement may be a bit suppressed because both experiments are run with closed-loop feedback, mitigating some of the positional error that would be caused by robot deflection. Unfortunately, it is difficult to decouple the effect of the varying feed rate from the effect of closed-loop feedback control because of the inability of the robot to maintain a constant axial DoC in open-loop operation due to kinematic errors. However, looking at



(a)

(b)

Figure 3.21: Zoomed-in views of the measured (using laser tracker) (a) constant feed rate and (b) varying feed rate positional data around the corner.

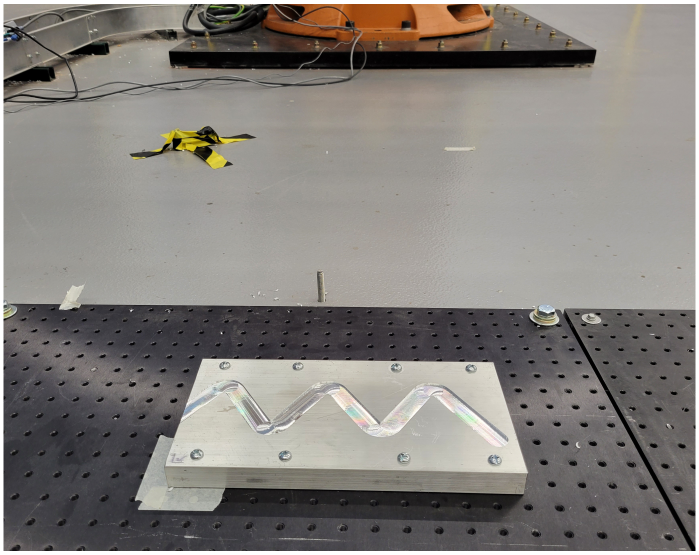


Figure 3.22: The machined part.

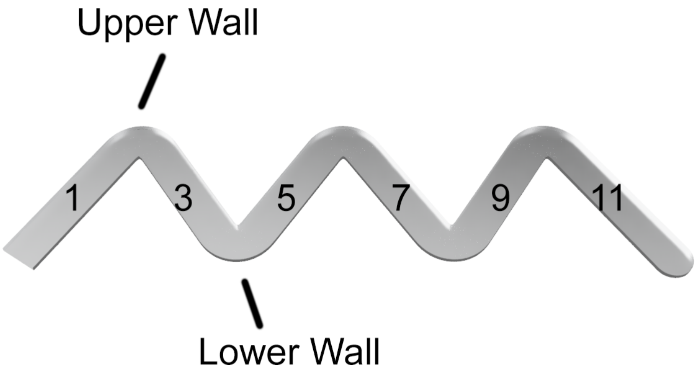
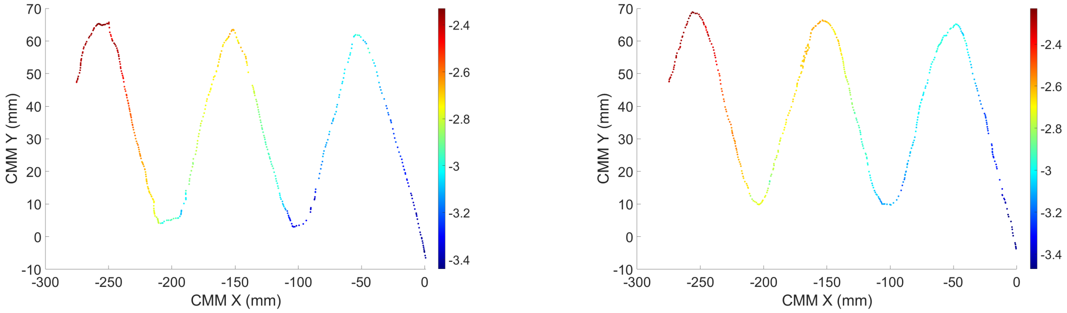


Figure 3.23: Definition of the segments and walls.

the machined floor of the slot gives additional insight. Using the same CMM, a scan of the slot floor was made. We expect a more consistent cut floor in the varying feed rate case. The axial DoC varies in both the constant and varying feed rate cuts due to various reasons, including: the mounting platform of the workpiece not being perfectly parallel to the robot base, the ± 0.30 mm thickness tolerance of the workpiece, and kinematic errors along the path. Despite this, it can be seen in Figure 3.24 that the varying feed rate trajectory results in less variation in cut depth due to the robot changing direction, particularly around the lower, ”$\cup$” shaped corners (segments 4 and 8). The mean and standard deviation of the axial DoC for each section can be seen in Table 3.6. Furthermore, the work pieces used in the constant and varying feed rate tests were mounted on the breadboard at identical locations, so any differences in the DoC can be attributed to the different trajectories. Note that the measurements of the slot shown in Figure 3.24 are taken with a handheld CMM, and as such, the path is not exactly on the center line of the slot.



(a) (b)

Figure 3.24: Measured axial DoC as a colormap on the measured XY path of the (a) constant feed rate and (b) varying feed rate cuts.

As flatness is a metric based on the maximum deviations, the average distance from the nominal plane of the linear section walls is also computed and shown in Table 3.7. The distance from a point (x_0, y_0, z_0) to a plane $Ax + By + Cz + D = 0$ is given by:

$$d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}} \tag{3.28}$$

Table 3.6: Slot depth metrics (mm) of the measured cut floor.

Segment	Constant Feed Rate		Varying Feed Rate		Difference	
	Mean	Std. Dev.	Mean	Std. Dev.	Δ Mean	Δ Std. Dev.
2 (Curv. $\cap$)	-2.375	0.037	-2.262	0.027	-0.113	0.010
3 (Lin. $\downarrow$)	-2.584	0.099	-2.523	0.091	-0.061	0.008
4 (Curv. $\cup$)	-2.900	0.126	-2.742	0.056	-0.158	0.070
5 (Lin. $\uparrow$)	-2.791	0.071	-2.682	0.066	-0.109	0.005
6 (Curv. $\cap$)	-2.701	0.035	-2.609	0.023	-0.092	0.012
7 (Lin. $\downarrow$)	-2.916	0.114	-2.870	0.092	-0.046	0.022
8 (Curv. $\cup$)	-3.244	0.111	-3.112	0.054	-0.132	0.057
9 (Lin. $\uparrow$)	-3.108	0.047	-3.008	0.033	-0.100	0.014
10 (Curv. $\cap$)	-3.033	0.042	-3.009	0.038	-0.024	0.004
11 (Lin. $\downarrow$)	-3.295	0.107	-3.333	0.091	0.038	0.016

As the magnitude of predicted deviations are lower in the varying feed rate case, it follows that the deviation from nominal cut wall would also be lower. An example of the measured wall points in comparison to the nominal wall location is shown in Figure 3.25.

Table 3.7: Average distance (mm) of the measured wall to nominal plane.

	Seg. 3 ($\downarrow$)	Seg. 5 ($\uparrow$)	Seg. 7 ($\downarrow$)	Seg. 9 ($\uparrow$)	Seg. 11 ($\downarrow$)
Constant feed rate, ~ 0.4 mm max deflection					
Upper Wall	0.498	0.475	0.424	0.516	0.458
Lower Wall	0.259	0.407	0.201	0.411	0.187
Varying feed rate, 0.29 mm deflection limit					
Upper Wall	0.422	0.386	0.392	0.428	0.418
Lower Wall	0.247	0.326	0.203	0.355	0.177
Improvement (constant $-$ varying case)					
Upper Wall	0.076	0.089	0.032	0.088	0.040
Lower Wall	0.012	0.081	-0.002	0.056	0.010

The quality of the cut floor was also inspected using an optical profilometer (Zygo Zegage Pro) to measure the surface roughness. A three-point average was taken for surface roughness measurements in each type of section of the triangular path, namely the linear upward, top curved, linear downward, and bottom curved sections. The measured values of interest are the arithmetic mean roughness height (S_a), root mean square roughness height (S_q), and maximum peak-to-valley roughness height (S_z). From the deflection and feed rate plots shown in Figure 3.18 and Figure 3.19, we expect the linear upward (+y) sections

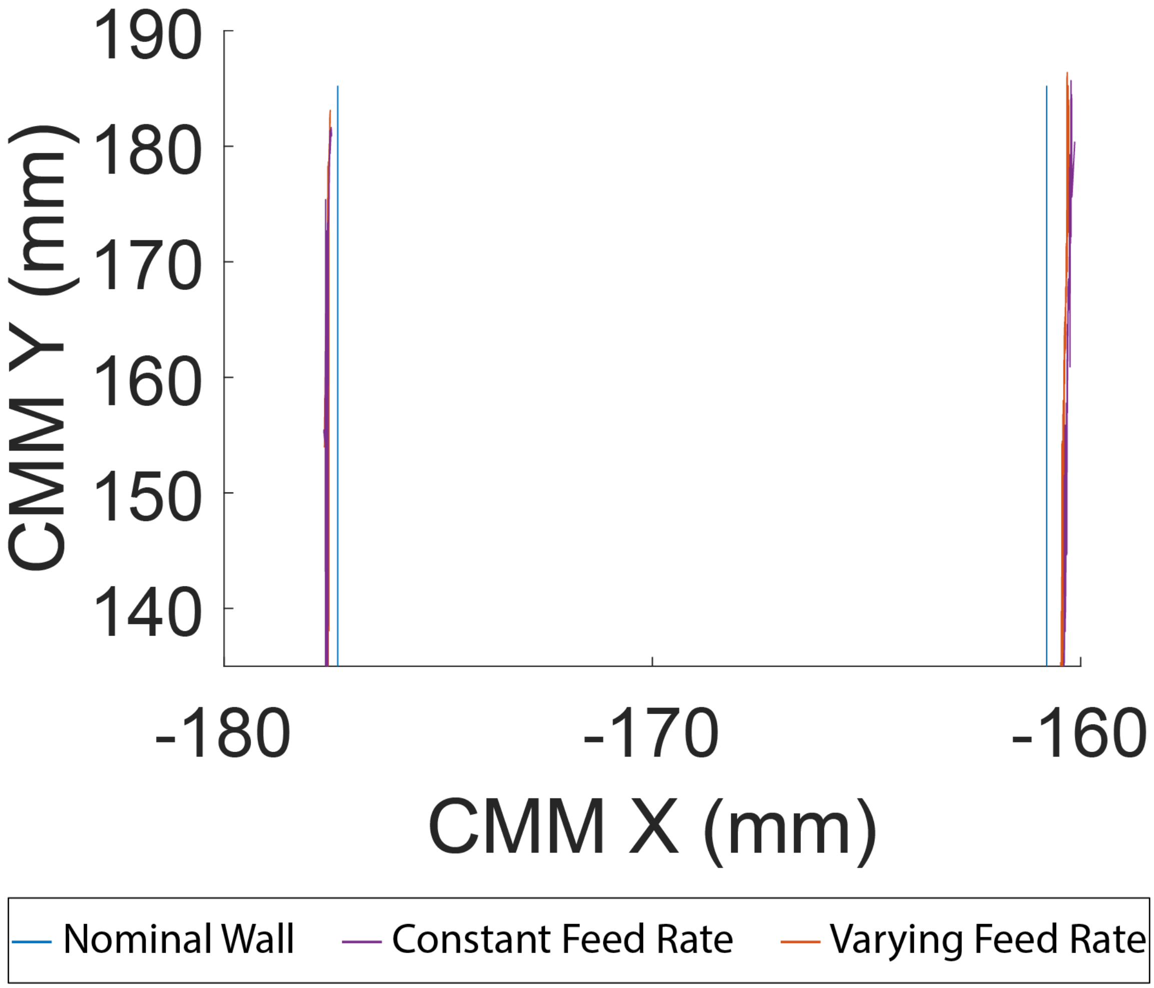


Figure 3.25: Measured wall location vs. the nominal wall location for Segment 3.

to have the greatest amount of change between the constant and varying feed rate cases. The measurements and improvement can be seen in Table 3.8. All optical profilometer measurements are given in Table A.4.

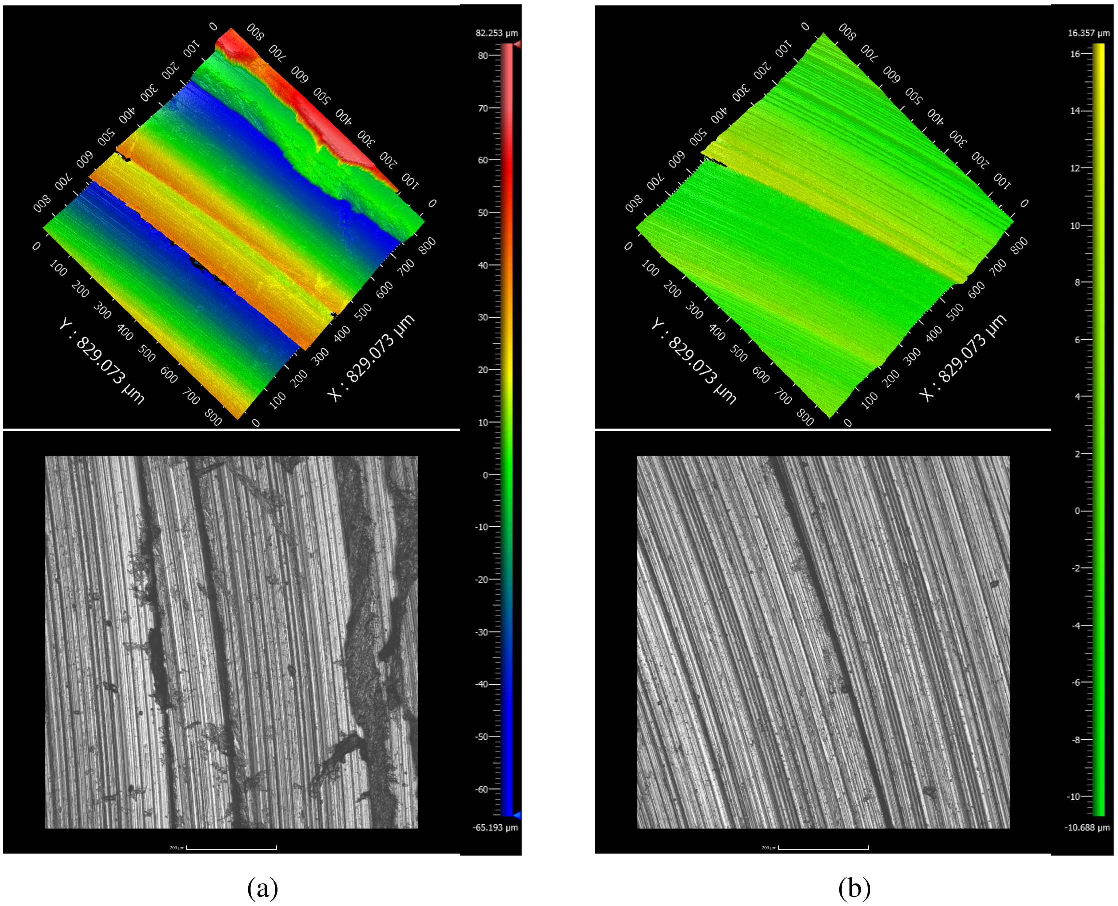
Table 3.8: Average of surface roughness measurements (μm) of the curvilinear slotting cuts.

	Constant Feed Rate			Varying Feed Rate		
	S_a	S_q	S_z	S_a	S_q	S_z
Seg. 3 (Linear ↓)	7.73	9.34	76.13	8.64	10.85	67.07
Seg. 4 (Bottom Curve)	18.99	23.15	121.89	7.08	8.70	48.92
Seg. 5 (Linear ↑)	13.94	16.78	112.31	8.12	9.76	54.69
Seg. 6 (Top Curve)	16.45	20.01	133.60	5.75	7.18	62.51
Improvement (constant − varying case)						
Seg. 3 (Linear ↓)	-0.91	-1.51	9.06			
Seg. 4 (Bottom Curve)	11.91	14.45	72.97			
Seg. 5 (Linear ↑)	5.82	7.02	57.72			
Seg. 6 (Top Curve)	10.70	12.83	71.09			

From the data, the largest improvements are found in the curved regions of the path, and representative images from the optical profilometer can be seen in Figure 3.26. Visually, it can be confirmed that the varying feed rate trajectory yields a much smoother milled surface in segments where the feed rate had to be significantly lowered to meet the deflection limit.

As a comparison, the native path approximation and trajectory planner of the KUKA KR500-3's KRC4 controller programmed using curvilinear path options in KRL is also evaluated. KRL defines four motion types: Point to Point (P2P), linear (LIN), circular (CIRC) and spline (SPL) motion. Of these four, P2P, LIN, and CIRC motions use approximate positioning, which entails the following [48]:

- The actual motion may not pass through the programmed points. Additionally, the approximated path is difficult to predict and generating the desired path can be complicated and time consuming.
- Often, the velocity may be reduced by the KRC4 controller in a manner that is hard



(a)

(b)

Figure 3.26: 3D roughness of segment 6 of the (a) constant feed rate and (b) varying feed rate cases.

Note: The colormap ranges from -65.193 μm (blue) to 82.253 μm (red). Zero is green.

to predict.

- The path can change depending on other constraints, i.e., velocity, acceleration, or time.

The SPL motion type, however, passes through the programmed points, maintains the programmed velocity in most cases, and does not change the path based on settings like velocity and acceleration. Therefore, the key points of the approximated triangle wave, namely the first point, last point, and curve start and end points, were programmed in KRL using the SPL motion type for comparison with the proposed varying radius path described earlier in this chapter. Note that the use of KRL, instead of SGM, allows for a path to be executed natively by the robot controller without external feedback control (termed open-loop in this thesis). With no constraints on the geometry of the spline path, the robot controller takes a path far from the desired shape, as seen in Figure 3.27.

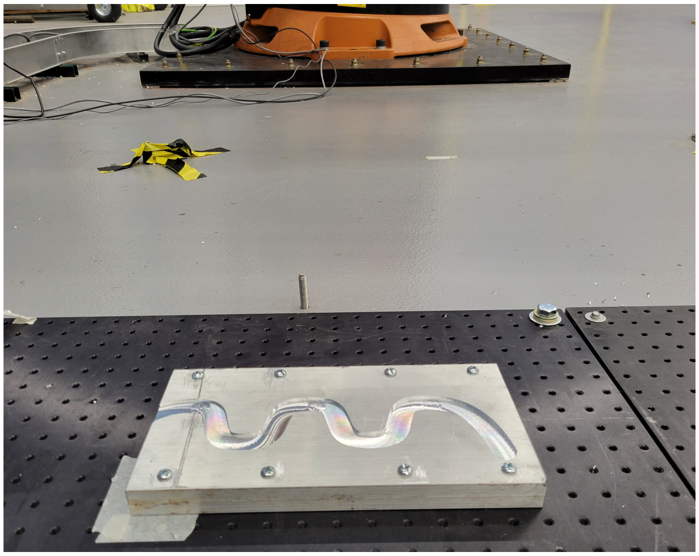


Figure 3.27: Machined part using KRL's SPL motion only.

An additional specification to the SPL motion type can be used to better define the path geometry. The spline-linear (SLIN) motion can be used within spline blocks to maintain the benefits of the SPL motion type discussed earlier, so long as the SLIN segments are not successive. By utilizing this specification, the only difference between the path generated by KRL and the deflection-limited trajectory planner is in the corners of the triangular wave. Thus, a direct comparison of the varying-radius path for the curved sections proposed in this thesis and the native KRL SLIN path can be made. Figure 3.28 shows laser tracker data of the path executed by the robot running the KRL SLIN trajectory overlaid on top of laser tracker data of the varying feed rate trajectory obtained with a deflection limit of 0.29 mm.

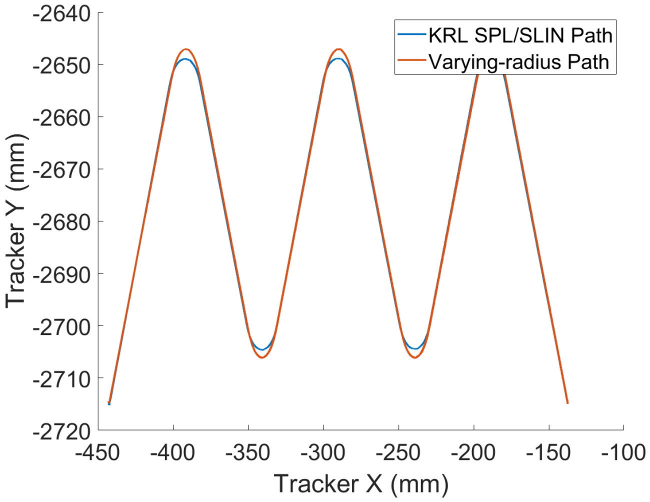
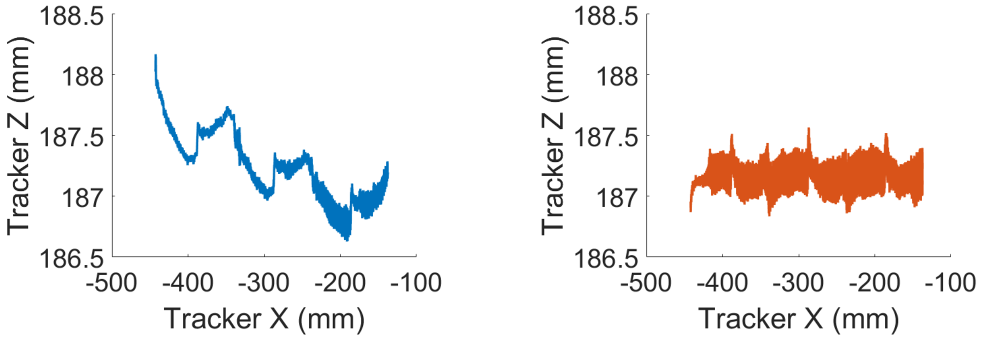


Figure 3.28: A comparison of laser tracker data for KRL SLIN motion versus the varying-radius trajectory.

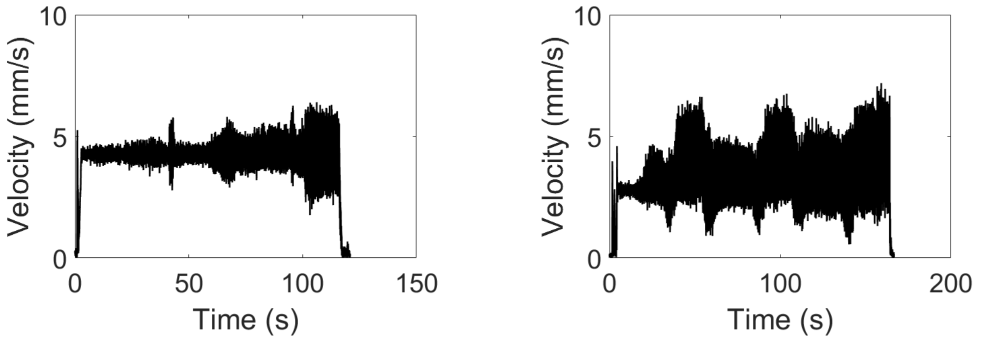
It can be seen that the trajectories are different, but the criteria which the KUKA robot controller's native trajectory planner uses to build the spline connecting the linear segments is unknown because of the proprietary nature of the KRC4 controller. In contrast, the trajectory planning algorithm presented in this chapter ensures G^2 and C^2 continuity with preferable jerk characteristics compared to that of traditional Bézier curves. The part machined using the KRL SLIN trajectory cannot be directly compared to that machined using the varying feed rate trajectory since the open loop, KRL guided trajectory failed to maintain a constant axial DoC, as shown in Figure 3.29. The closed-loop trajectory does, however, maintain a fairly constant axial DoC, with fluctuations in the plot due to robot vibrations.



(a) (b)

Figure 3.29: A comparison of the variation in the robot Z position in the (a) open-loop SLIN trajectory and the (b) closed-loop varying feed rate trajectory.

The SLIN trajectory maintains the constant feed rate commanded, as shown in Figure 3.30, and exhibits a tighter curved path approximation at the corner. However, undesirable characteristics and deviation can also be seen around the corners of the SLIN path. A zoomed-in view is shown in Figure 3.31, where the deviation in the SLIN path can be seen after cresting the peak of the curve (moving in the +x direction).



(a) (b)

Figure 3.30: A comparison of the measured velocity (feed rate) in the (a) open-loop SLIN trajectory and the (b) closed-loop varying feed rate trajectory.

3.8 Summary

A deflection-limited trajectory planning algorithm for curvilinear slotting cuts was developed to minimize the impact of varying robot stiffness in robotic milling. Analysis showed

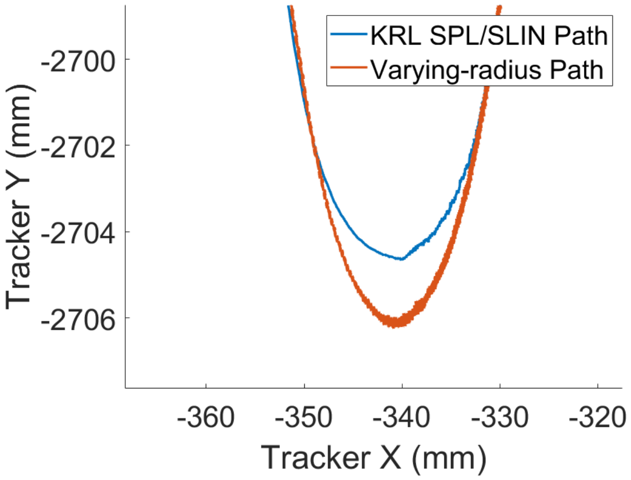


Figure 3.31: Zoomed-in view of the measured positions of the open-loop SLIN trajectory and the closed-loop varying feed rate trajectories around segment 4.

some improvement, particularly around the corners, where the constant feed rate trajectory tended to overshoot and cause greater vibrations. However, the effect of the trajectory planner could not be decoupled from the closed-loop feedback control, which corrects instantaneous deviations in robot pose and likely suppressed some of the differences between the constant and varying feed rate cases. Additionally, it was shown that although a KRL guided trajectory with SLIN motion can recreate a path that is similar, the constraints used in creating the spline around the corners are unknown and unpredictable due to the proprietary nature of the native robot controller. For manufacturing applications, in which part and path accuracy is important, the black-box behavior of the native robot controller is undesirable. Under such conditions, the proposed deflection-limited, varying-radius trajectory planning algorithm may be preferable.

CHAPTER 4
DEFLECTION-LIMITED TRAJECTORY PLANNING FOR LINEAR PERIPHERAL CUTS

4.1 Introduction

As the cutting tool is constrained by the workpiece material on either side during slotting cuts, we hypothesize that the effects of the deflection-limited trajectory planner will be easier to see in a peripheral cut of increasing radial DoC. This chapter focuses on testing this hypothesis by adapting the deflection-limited trajectory planning algorithm presented in Chapter 3 to enable its application to peripheral end milling cuts. As in the previous chapter, we assume that most of the deflection induced path inaccuracy stems from the comparatively low stiffness of the robot.

4.2 Algorithm Overview

The algorithm introduced in Chapter 3 is adapted as shown in Figure 4.1. Since the robot path considered in this chapter is linear, there is no need for path approximation anymore; the inclined linear path does not contain changes in direction as in the triangular wave path considered in Chapter 3. In the revised algorithm, the predicted milling forces and robot deflections are computed at key points every length ℓ, specified by the user, and the feed rate is adjusted iteratively until the deflection limit is met. The segments between the deflection-limited points are once again refit to ensure C^2 continuity. Note that as the path is fully linear, G^2 continuity is inherently ensured. The trajectory is then exported as a CSV file to be implemented in the same closed-loop, real-time control architecture introduced in Chapter 3.

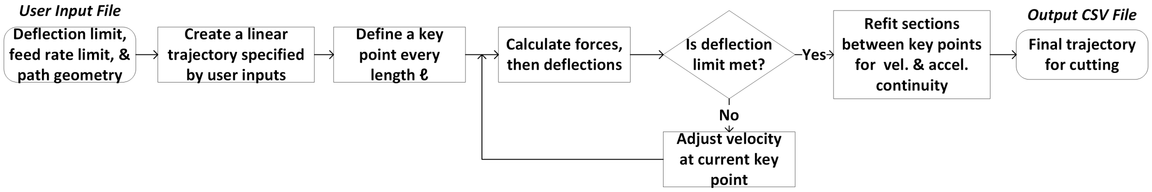

User Input File
Deflection limit, feed rate limit, & path geometry
Create a linear trajectory specified by user inputs
Define a key point every length ℓ
Calculate forces, then deflections
Is deflection limit met?
Yes
No
Adjust velocity at current key point
Refit sections between key points for vel. & accel. continuity
Output CSV File
Final trajectory for cutting

Figure 4.1: Linear peripheral cut algorithm overview.

The deflection-limited trajectory planning algorithm for linear peripheral cuts consists of the following key steps:

1. Create a linear path of user specified x-length and y-length equal to the cutting tool radius as seen in Figure 4.2.

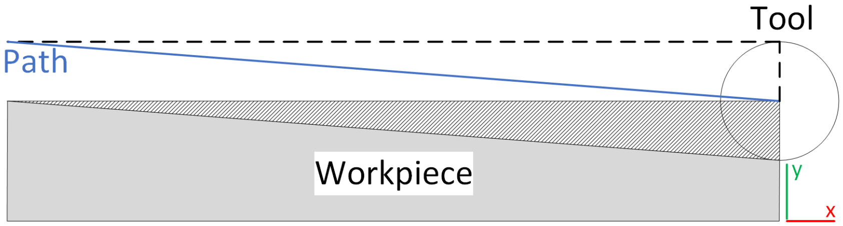


Figure 4.2: Top view of the toolpath and cut workpiece in the linear peripheral cut.

2. Define equally spaced key points separated by a length ℓ specified by the user
3. Calculate the peripheral milling forces and corresponding robot static deflections at every key point, assuming down milling
4. Vary the feed rate at the key points to satisfy the deflection limit
5. With a set of path positions and corresponding desired feed rates, use a 5th order polynomial on tangential displacement $s(t)$ to smoothly vary the velocity and ensure C^2 continuity
6. Output the final path and motion timing law as a CSV for use in the closed-loop, real-time robot control architecture.

4.3 Motion Timing Law

The 5th order polynomial $s(t)$ introduced in Section 3.5 is once again used to interpolate through the key points defined along the path by applying the following constraints:

- $s_0, s_f, \dot{s}_0, \dot{s}_f$ are obtained by the defined key points and the feed rates obtained in step 4 of the algorithm presented above.

  Note: t_f is the time it takes to traverse the section of length ℓ between key points at a constant velocity of $min(\dot{s}_0, \dot{s}_f)$

- $\ddot{s}_0, \ddot{s}_f$ are determined such that $\ddot{s}_0[i] = \ddot{s}_f[i-1], \quad \ddot{s}_f[i] = 2\overline{a} - \ddot{s}_0[i]$

  Note: we assume $\ddot{s}_0 = 0$ and $\overline{a} - \frac{\dot{s}_f - \dot{s}_i}{t_f}$

This method is sufficient as we simply want to ensure C^2 continuity and do not necessarily care about the shape of the velocity and acceleration profiles.

4.4 Experimental Work

The same industrial robotic milling system introduced in Chapter 3 is used to implement and evaluate the performance of the deflection-limited trajectory planning algorithm for linear peripheral milling cuts. The nominal feed rate is kept at 4.23 mm/s (10 IPM) with a spindle speed of 2500 rpm. The closed-loop, real-time feedback gains used in the experiments are unchanged at 0.05 for the proportional gain and 0.3 for the integral gain used to correct errors in the Cartesian directions. An integral gain of 0.02 is again used for correcting angular errors. The workpiece is the same 25.4 mm (1 in) thick 6061 aluminum cut with the same 2 flute, 19.05 mm (0.75 in) diameter, TiN coated HSS square end mill.

However, the workpiece now has a length of 203.2 mm (8 in) and a width of 50.8 mm (2 in). Additionally, the cutting path is linear with a x length of 203.2 mm (the length of the workpiece) and y length of 9.525 mm (0.375 in, half radial immersion). In these experiments, carried out on the experimental setup shown in Figure 4.3, down milling is used from zero to half radial immersion. The constant and varying feed rate trajectories are evaluated in the following section.

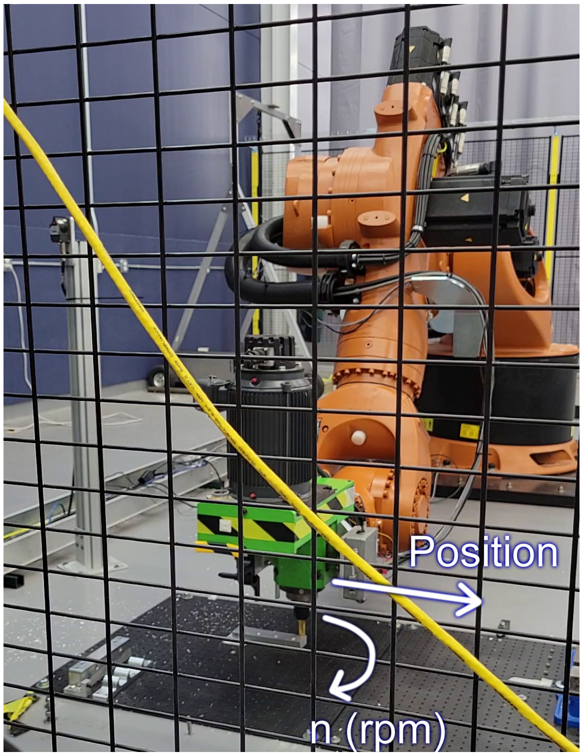
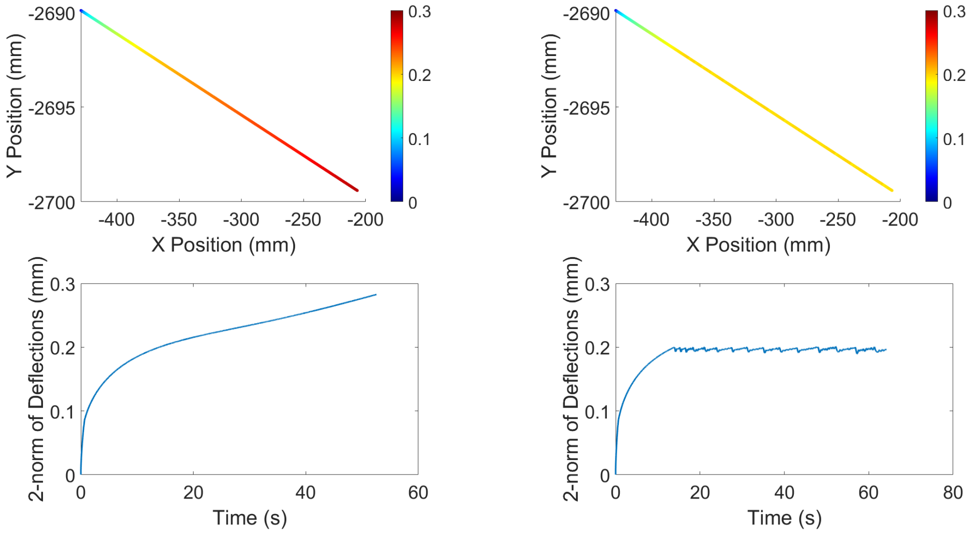


Figure 4.3: linear peripheral cuts using the robotic milling system.

4.5 Results and Analysis

The trajectories discussed in this section are still of the T-Mac referenced in the robot base frame. The predicted static deflections for the constant and varying feed rate trajectories are shown in Figure 4.4, where the deflection limit of the varying feed rate trajectory was set to 0.2 mm, and a uniform spacing of $\ell = 2.54$ mm (0.1 in) between the key points was used for checking robot deflections. The maximum static deflection predicted in the constant feed rate trajectory is 0.283 mm.

To achieve the deflection limit, the feed rate is varied as discussed earlier in this chapter and in Chapter 3. However, a wavy pattern can be seen in the varying feed rate trajectory's predicted deflections and prescribed feed rate, shown in Figure 4.4b and Figure 4.5b. This



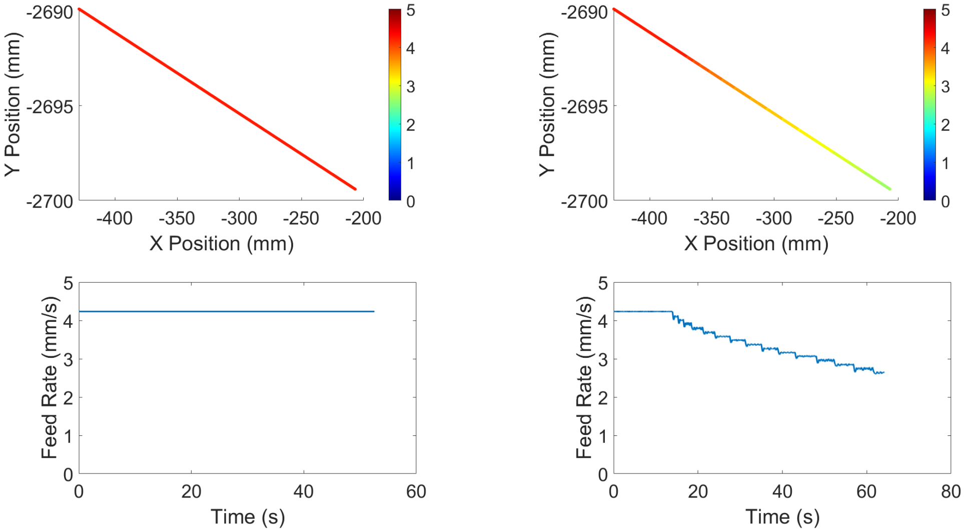
(a)

(b)

Figure 4.4: Predicted static deflections of the (a) constant feed rate and (b) varying feed rate, linear trajectories.

is attributed to the choice of t_f in the 5th order polynomial as $min(\dot{s}_0, \dot{s}_f)$, which prescribes more time than in a trapezoidal velocity profile case and allows the velocity profile to dip, as seen in Figure 4.5b. Taking the maximum of the two would result in a similar, though flipped, velocity profile and could potentially result in static deflections above the specified limit. As the goal here is to simply ensure C^2 continuity, the actual profiles of the velocity and acceleration were not a priority. Furthermore, given the physical limitations of the robot and the fact that the trajectory is still C^2 continuous, we expect that the velocity fluctuations in the prescribed trajectory will not create any undesirable behavior during execution of the trajectory.

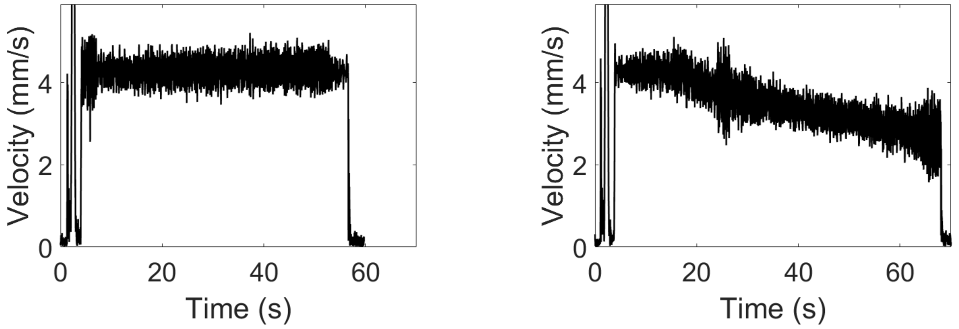
Using a moving average window of 0.1 s to smooth out spikes in the laser tracker data due to robot vibrations, the velocity computed from the time-stamped tracker position data is plotted in Figure 4.6. It can be seen that the nominal velocity of 4.23 mm/s is maintained in the constant feed rate trajectory during milling, while the varying feed rate trajectory exhibits a decreasing velocity commensurate with increasing radial immersion along the path (and correspondingly greater forces and deflections).



(a)

(b)

Figure 4.5: Prescribed feed rate of the (a) constant feed rate and (b) varying feed rate, linear trajectories.



(a)

(b)

Figure 4.6: Velocities of the (a) constant feed rate and (b) varying feed rate, linear trajectories computed from laser tracker data using a 0.1 s moving average window.

A close-up inspection of the tracker position data, shown in Figure 4.7, reveals some differences in the oscillation amplitudes between the constant and varying feed rate trajectories. Specifically, the vibration amplitude is slightly larger in the varying feed rate trajectory.

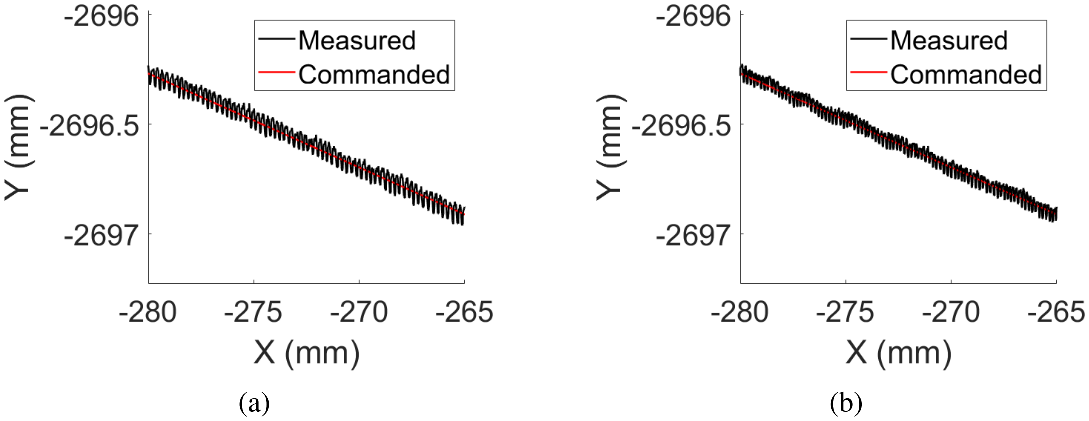


Figure 4.7: Zoomed-in view of the measured instantaneous positions of the robot (T-Mac) for the (a) constant feed rate and (b) varying feed rate, linear trajectories.

Flatness measurements of the cut wall and floor surfaces were made using the Keyence CMM and are shown in Table 4.1. Improvement is defined by the difference between the flatness values for the constant and varying feed rate trajectories.

Table 4.1: Flatness measurements (mm) of the walls and floor of the linear peripheral cut.

	Cut Wall Flatness	Cut Floor Flatness
Constant Feed Rate (0.283 mm max deflection)	0.218	0.066
Varying Feed Rate (0.2 mm deflection limit)	0.134	0.031
Improvement	0.084	0.035

Additionally, the average distance of the measured points along the wall and floor to the locations of the nominal wall and floor planes, respectively, were calculated and are shown in Table 4.2. The distance from a point (x_0, y_0, z_0) to a plane defined by $Ax + By + Cz + D = 0$ is given in Equation 3.28.

Table 4.2: Average distance (mm) of the actual wall and floor of the linear peripheral cut to their nominal planes.

	Cut Wall Avg. Dist.	Cut Floor Avg. Dist.
Constant Feed Rate (0.283 mm max deflection)	0.296	0.107
Varying Feed Rate (0.2 mm deflection limit)	0.171	0.104
Improvement	0.125	0.003

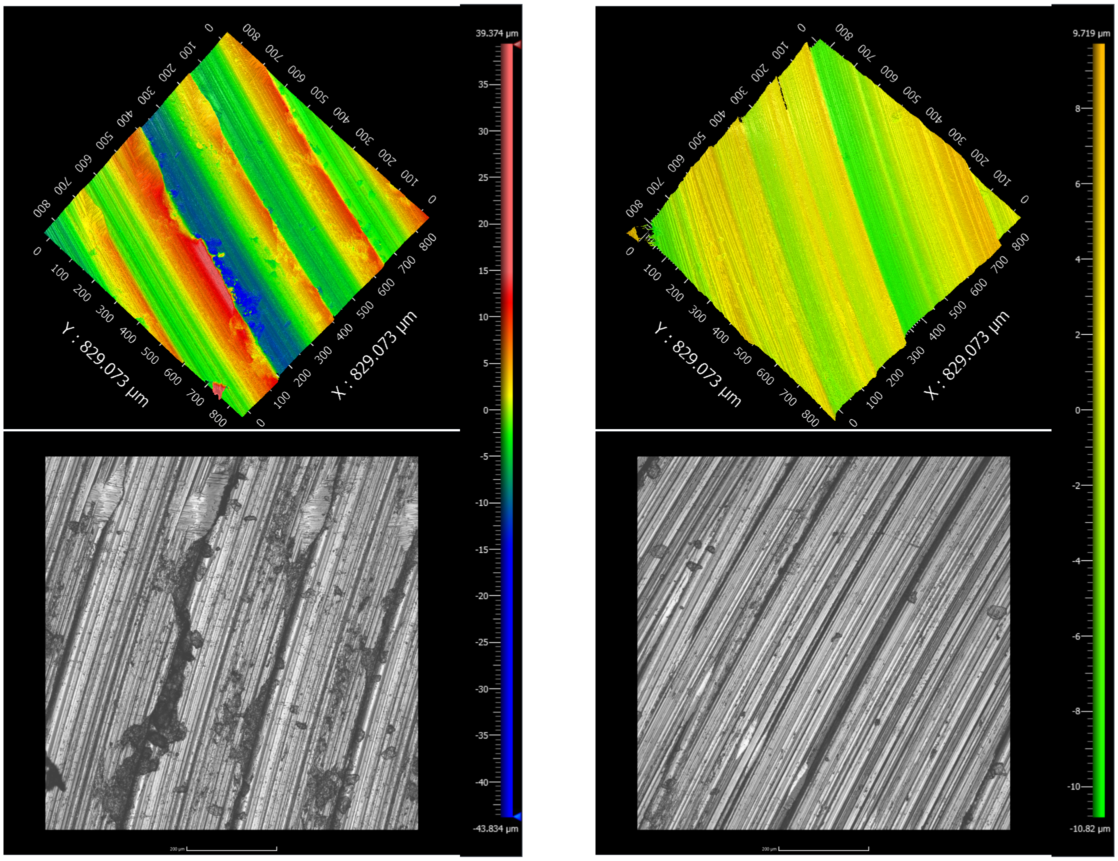
The surface roughness of the floor surface was inspected using the Zygo optical profilometer, and the three-point averages are reported in Table 4.3. The entirety of the measurements can be seen in Table A.5. An optical profilometer image can also be seen in Figure 4.8. Consistent with the CMM measurements, the cut floor surface can be seen to be smoother for the varying feed rate trajectories than for the constant feed rate trajectories. This is attributed to the well-known relationship between feed rate and surface roughness in milling processes [49].

Table 4.3: Average of surface roughness measurements (μm) of the linear peripheral cuts.

	S_a	S_q	S_z
Constant Feed Rate	4.25	5.33	70.00
Varying Feed Rate	3.20	3.98	28.04
Improvement	1.05	1.35	41.96

4.6 Summary

The deflection-limited trajectory planning algorithm introduced in Chapter 3 was adapted from curvilinear slotting cuts to linear peripheral cuts. Due to the inherently lower forces and lack of any direction changes in the linear path, the approach and results are seen to be different than in the curvilinear slotting case. Regardless, an improvement in the cut wall and floor flatness, as well as cut floor surface roughness was observed. Once again, the closed-loop, real-time control system that sends instantaneous position and orientation corrections to the robot likely reduced the effect of the algorithm. However, feedback must be



(a) (b)

Figure 4.8: Optical profilometer measurements of the linear peripheral cut surfaces obtained with the (a) constant feed rate and (b) varying feed rate trajectory.

Note: The colormap ranges from -43.834 μm (blue) to 39.374 μm (red). Zero is light green.

used to keep constant axial and radial DoC, as shown in Figure 3.29. Most importantly, the results of this chapter have shown that the deflection-limited trajectory planning algorithm can be extended to linear peripheral cuts. Additional development of the trajectory planner can further generalize the use cases to more complex paths.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

This chapter summarizes the original contributions and main conclusions of this thesis and suggests possible areas for future studies.

5.1 Original Contributions

A process-mechanics based methodology for prescribing the maximum allowable feed rate through a deflection-limited trajectory planner for robotic milling was presented in this thesis. While it is perhaps obvious that a lower feed rate will result in lower milling forces and deflections, and therefore better part accuracy, the originality of this thesis lies in the development of a formal methodology for incorporating process force induced deflection limits in robot trajectory planning. This contrasts with the common practice of using experience and empirically established tables to determine an appropriate feed rate in milling. From a trajectory planning perspective, a new, varying-radius method for generating a G^2 and C^2 continuous trajectory that blends two linear segments of a curvilinear path was presented and its performance compared to existing 4th and 5th order Bézier curve based trajectory planning methods. Note that while the industrial robotic system used in this work was a 6-axis serial link manipulator configured for milling, the approach presented is general and can be applied to other robotic manipulators used in other applications.

5.2 Main Conclusions

The main conclusions of this thesis are:

- The varying-radius based path approximation method yields an improved jerk profile with lower maximum jerk and smaller range of jerk values than the standard 5th order

Bézier curve method.

- At a constant feed rate, the robot tends to experience greater deflection around corners along a curvilinear path. This is due to the changing stiffness and direction of milling forces that is unaccounted for in the constant feed rate trajectory. The deflection is seen in the XY position data of the robot measured by a laser tracker as well as in the Z depth of the cut floor surface measured by a CMM.

- The greater variation in the cut floor surface position data for the constant feed rate trajectory is also seen in the surface roughness measurements, where the varying feed rate trajectory yields a smoother and more consistent surface.

- The walls of the curvilinear slotting trajectory showed some improvement in flatness, but improvements are limited due to the physical constraint imposed by the walls surrounding the cutter on either side.

- The linear peripheral cut saw similar improvements in the cut wall flatness and cut floor surface roughness. However, the lack of direction changes in the trajectory, a significant source of robot vibrations due to jerk, resulted in an inherently more consistent cut.

- Despite the coupled nature of the deflection-limited trajectory planning algorithm and the closed-loop, real-time feedback control system, the deflection-limited trajectory planning algorithm resulted in trajectories that yielded lower robot vibrations, better machining accuracy, and improved surface finish.

5.3 Future Work and Recommendations

The deflection-limited trajectory planning algorithm establishes the merit of process aware robotic trajectory planning in machining. Furthermore, it provides a basis for future developments that enable feasibility of industrial robotics in manufacturing tasks characterized by high process forces. The obvious next step is to further generalize this algorithm to more complex milling operations involving free-form peripheral tool paths characterized by rapid changes in tool orientation. The peripheral cuts presented in this thesis utilized a simple path and part geometry, an assumption that must be relaxed for more realistic industrial use cases. The current algorithm also takes advantage of knowledge of the path to be approximated, such as the fact that the linear sections are symmetric and that the path is periodic (in the case of the triangle wave path). Relaxing these assumptions would entail accounting for an arbitrary set of points to be approximated into a path that is C^2 continuous.

Another next step with potential merit is to incorporate a dynamic model of the milling forces and robot deflections, one that has inertial and damping terms, in place of the current static deflection model to better predict the instantaneous robot deflection during milling. Such dynamic models of conventional peripheral milling are available in the literature but have not been adapted for robotic milling [50–52].

Appendices

APPENDIX A

ADDITIONAL TABLES

Table A.1: Specifications of the 2 flute cutter used.

System of Measurement	Inch
Finish	Titanium Nitride (TiN) Coated
Material	High-Speed Steel
Mill Diameter	3/4”
Mill Diameter Tolerance	0.000” to 0.001”
Shank Type	Straight
Shank Diameter	3/4”
Length of Cut	1 5/16”
Overall Length	3 9/16”
Flute Type	Spiral
Number of Flutes	2
Flute Spacing	Equal
Helix Angle	30°
Cut Style	Square
End Mill Type	Center Cutting

Table A.2: Specifications of the 6061 Aluminum used.

Material	6061 Aluminum
Shape	Sheet and Bar
Thickness	1”
Thickness Tolerance	-0.012” to 0.012”
Tolerance Rating	Standard
Width Tolerance	-0.084” to 0.084”
Length Tolerance	-1/16” to 1/16”
Yield Strength	35,000 psi
Fabrication	Extruded
Temper	T6511
Heat Treatment	Hardened
Hardness	Brinell 95
Hardness Rating	Soft
Heat Treatable	Yes
Certificate	Material Certificate with Traceable Lot Number
Appearance	Plain
Temperature Range	-320° to 300° F
Specifications Met	ASTM B221, SAE AMS QQ-A-200/8
Flatness Tolerance	0.010” per in.
Elongation	12.5%
Material Composition	
Aluminum	95.1-98.2%
Chromium	0.4-0.8%
Copper	0.05-0.4%
Iron	0-0.7%
Magnesium	0.8-1.2%
Manganese	0-0.15%
Nickel	0-0.05%
Silicon	0.4-0.8%
Titanium	0-0.15%
Zinc	0-0.25%
Zirconium	0-0.25%
Other	0.15%

Table A.3: Robot configuration (deg) used for the static stiffness calibration [13]

Test No.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	19.67	-52.14	117.9	11.86	-66.08	-4.84
2	26.35	-32.54	123.49	-8.99	-90.81	-0.15
3	58.28	-10.5	105.98	42.39	-115.04	-2.57
4	44.71	-70.81	58.16	138.51	-41.63	-125.84
5	-23.13	-85.41	80.84	70.31	83.63	-71.78
6	2.39	-34.82	69.97	115.13	72.1	176.98
7	25.21	-55.7	130.03	169.88	72.17	-179.78
8	42.99	-40.39	97.42	-177.38	72.1	-179.8
9	42.99	-57.91	110.04	-155.79	51.1	-179.83
10	28.12	-55.4	84.06	-167.6	27.07	-179.85

Table A.4: Surface roughness measurements (μm) taken using an optical profilometer of the curvilinear, peripheral cut.

	Non-optimized			Optimized		
	S_a	S_q	S_z	S_a	S_q	S_z
Seg. 3 (Linear $\downarrow$)						
Sample 1	7.279	8.794	107.886	8.274	9.816	52.154
Sample 2	5.949	7.306	50.801	9.734	11.256	73.557
Sample 3	9.95	11.912	69.689	7.915	11.465	75.503
Seg. 4 (Bottom Curve)						
Sample 1	14.57	18.08	112.128	8.132	9.547	45.063
Sample 2	34.169	41.87	191.304	8.037	10.546	57.285
Sample 3	8.232	9.502	62.228	5.067	6.001	44.57
Seg. 5 (Linear $\uparrow$)						
Sample 1	18.888	22.05	171.265	12.543	14.269	62.094
Sample 2	12.104	15.171	84.333	4.431	5.666	45.068
Sample 3	10.84	13.105	81.332	7.371	9.34	56.615
Seg. 6 (Top Curve)						
Sample 1	16.357	19.103	106.05	3.427	4.701	47.381
Sample 2	11.737	15.01	147.294	9.561	11.817	113.097
Sample 3	21.263	25.924	147.446	4.254	5.015	27.045

Table A.5: Surface roughness measurements (μm) taken using an optical profilometer of the linear, peripheral cut.

	Non-optimized			Optimized		
	S_a	S_q	S_z	S_a	S_q	S_z
Sample 1	4.427	5.428	43.069	3.001	3.827	20.54
Sample 2	4.554	5.638	83.209	3.747	4.693	32.07
Sample 3	3.776	4.924	83.734	2.862	3.417	31.516

APPENDIX B
ADDITIONAL ALGORITHMS

B.1 Creating a Corner of Specified Radius

A circular corner of given radius is created using geometry. Suppose we have three points, P_1, P_2, P_c where P_1 is the first point, P_c is the corner point to be rounded, and P_2 is the second point. We define vectors $\vec{v}_1 := \overrightarrow{P_cP_1}$, $\vec{v}_2 := \overrightarrow{P_cP_2}$. Then the angle of the corner is

$$\alpha = \frac{cos^{-1}(\vec{v}_1 \cdot \vec{v}_2)}{2\,||\vec{v}_1||_2\,||\vec{v}_1||_2}$$

Let segment s be the segment from the corner to the tangent point in which the circular section meets the linear section. Given desired radius r, the length of s is given as

$$\ell_s = \frac{r}{|tan\,\alpha/2|}$$

Radius r must be chosen such that the length of the segment s is not longer than either of the linear sections, or $\ell_s \leq min(||\vec{v}_1||_2,\, ||\vec{v}_2||_2)$. Let $\vec{u}_i$ be the unit vector of $\vec{v}_i$, then the tangent points of the circle and linear segments are:

$$P_{t1} = P_c + \ell_s \vec{u}_1, \quad P_{t2} = P_c + \ell_s \vec{u}_2$$

To find the circle's center, we take the a perpendicular vector of $\vec{u}_1$ and $\vec{u}_2$ that points inward, $\vec{u}_{1\perp}$, $\vec{u}_{2\perp}$ and thus

$$P_{cc} = P_{t1} + r\vec{u}_{1\perp} = P_{t2} + r\vec{u}_{2\perp}$$

In this implementation, where we assumed a concave down shape, one way to find the right perpendicular vector, the one that points inward, is to take the average $\vec{u}_{avg} = (\vec{u}_1 + \vec{u}_2)/2$ and

$$\vec{u}_i = \begin{bmatrix} a \\ \\ b \end{bmatrix} \quad \Rightarrow \quad \vec{u}_{i\perp} = \begin{bmatrix} -b \\ \\ a \end{bmatrix}$$

$$\begin{cases} \vec{u}_{1\perp,2\perp} = -\vec{u}_{1\perp,2\perp} & \text{if } tan^{-1}\vec{u}_{avg} > 0 \end{cases}$$

Finally, we can find the start and end angles of the circular segment.

$$\theta_1 = tan^{-1}(\overrightarrow{P_{cc}P_{t1}}),\, \theta_2 = tan^{-1}(\overrightarrow{P_{cc}P_{t2}})$$

We also adjust θ by 2π such that there is no discontinuity in $\theta(t)$ when using the arctan function in implementation. For atan2, this means

$$\begin{cases} \theta_{1,2} = \theta_{1,2} + 2\pi & \text{if } \theta_2 - \theta_1 > \pi \\ \\ \theta_2 = \theta_2 + 2\pi & \text{if } \theta_2 - \theta_1 < -\pi \end{cases}$$

REFERENCES

[1] S. Kalpakjian and S. R. Schmid, "Manufacturing processes for engineering materials," in 5th ed. Pearson Education, Inc., 2008, ch. 14.3 Numerical Control, pp. 878–883, ISBN: 0-13-227271-7.

[2] J. Liu, Y. Tian, and F. Gao, "A novel six-legged walking machine tool for in-situ operations," *Frontiers of Mechanical Engineering*, vol. 15, no. 3, pp. 351–364, Sep. 2020.

[3] A. Olarra, D. Axinte, L. Uriarte, and R. Bueno, "Machining with the walkinghex: A walking parallel kinematic machine tool for in situ operations," *CIRP Annals*, vol. 66, no. 1, pp. 361–364, 2017.

[4] X. Zhao, B. Tao, and H. Ding, "Multimobile robot cluster system for robot machining of large-scale workpieces," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 561–571, 2022.

[5] A. Karim and A. Verl, "Challenges and obstacles in robot-machining," in *IEEE ISR 2013*, 2013, pp. 1–4.

[6] W. Ji and L. Wang, "Industrial robotic machining: A review," *The International Journal of Advanced Manufacturing Technology*, vol. 103, no. 1, pp. 1239–1255, Jul. 2019.

[7] Y. N. Hu and Y. H. Chen, "Implementation of a robot system for sculptured surface cutting. part 1. rough machining," *The International Journal of Advanced Manufacturing Technology*, vol. 15, no. 9, pp. 624–629, Aug. 1999.

[8] B. Denkena, B. Bergmann, and T. Lepper, "Design and optimization of a machining robot," *Procedia Manufacturing*, vol. 14, pp. 89–96, 2017, 17th Machining Innovations Conference for Aerospace Industry, MIC 2017, 6-7 December 2017, Garbsen, Germany.

[9] F. Schnoes and M. Zaeh, "Model-based planning of machining operations for industrial robots," *Procedia CIRP*, vol. 82, pp. 497–502, 2019, 17th CIRP Conference on Modelling of Machining Operations (17th CIRP CMMO).

[10] C. Reinl, M. Friedmann, J. Bauer, M. Pischan, E. Abele, and O. von Stryk, "Model-based off-line compensation of path deviation for industrial robots in milling applications," in *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2011, pp. 367–372.

[11] G. Xiong, Z.-L. Li, Y. Ding, and L. Zhu, "Integration of optimized feedrate into an online adaptive force controller for robot milling," *The International Journal of Advanced Manufacturing Technology*, vol. 106, no. 3, pp. 1533–1542, Jan. 2020.

[12] Y. Chen and F. Dong, "Robot machining: Recent development and future research issues," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9, pp. 1489–1497, Jun. 2013.

[13] T. Cvitanic, V. Nguyen, and S. N. Melkote, "Pose optimization in robotic machining using static and dynamic stiffness models," *Robotics and Computer-Integrated Manufacturing*, vol. 66, p. 101 992, 2020.

[14] *Trajectory-Dependent Compensation Scheme to Reduce Manipulator Execution Errors for Manufacturing Applications*, vol. Volume 1: Additive Manufacturing; Advanced Materials Manufacturing; Biomanufacturing; Life Cycle Engineering; Manufacturing Equipment and Automation, International Manufacturing Science and Engineering Conference, V001T05A009, Jun. 2021. eprint: https://asmedigitalcollection.asme.org/MSEC/proceedings-pdf/MSEC2021/85062/V001T05A009/6736411/v001t05a009-msec2021-63617.pdf.

[15] V. Nguyen, J. Johnson, and S. Melkote, "Active vibration suppression in robotic milling using optimal control," *International Journal of Machine Tools and Manufacture*, vol. 152, p. 103 541, 2020.

[16] S. Tajima and B. Sencer, "Real-time trajectory generation for 5-axis machine tools with singularity avoidance," *CIRP Annals*, vol. 69, no. 1, pp. 349–352, 2020.

[17] C. Dumas, S. Caro, S. Garnier, and B. Furet, "Joint stiffness identification of six-revolute industrial serial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 4, pp. 881–888, 2011, Conference papers of Flexible Automation and Intelligent Manufacturing.

[18] X. Beudaert, S. Lavernhe, and C. Tournier, "Feedrate interpolation with axis jerk constraints on 5-axis nurbs and g1 tool path," *International Journal of Machine Tools and Manufacture*, vol. 57, pp. 73–82, 2012.

[19] A. Bharathi and J. Dong, "Feedrate optimization for smooth minimum-time trajectory generation with higher order constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 82, no. 5, pp. 1029–1040, Feb. 2016.

[20] K. Zhang, C.-M. Yuan, and X.-S. Gao, "Efficient algorithm for time-optimal feedrate planning and smoothing with confined chord error and acceleration," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9, pp. 1685–1697, Jun. 2013.

[21] K. Erkorkmaz and Y. Altintas, "High speed cnc system design. part i: Jerk limited trajectory generation and quintic spline interpolation," *International Journal of Machine Tools and Manufacture*, vol. 41, no. 9, pp. 1323–1345, 2001.

[22] S. Tulsyan and Y. Altintas, "Local toolpath smoothing for five-axis machine tools," *International Journal of Machine Tools and Manufacture*, vol. 96, pp. 15–26, 2015.

[23] K. Zhang, J.-X. Guo, and X.-S. Gao, "Cubic spline trajectory generation with axis jerk and tracking error constraints," *International Journal of Precision Engineering and Manufacturing*, vol. 14, no. 7, pp. 1141–1146, Jul. 2013.

[24] Y. Altintas and K. Erkorkmaz, "Feedrate optimization for spline interpolation in high speed machine tools," *CIRP Annals*, vol. 52, no. 1, pp. 297–302, 2003.

[25] L. Biagiotti and C. Melchiorri, "Trajectory planning for automatic machines and robots," in Springer-Verlag Berlin Heidelberg, 2008, ch. 8 Multidimensional Trajectories and Geometeric Path Planning, pp. 341–414, ISBN: 978-3-540-85628-3.

[26] J. Peng, P. Huang, Y. Ding, and H. Ding, "An analytical method for decoupled local smoothing of linear paths in industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102 193, 2021.

[27] B.-J. Park, H.-J. Lee, K.-K. Oh, and C.-J. Moon, "Jerk-limited time-optimal reference trajectory generation for robot actuators," *The International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 4, pp. 264–271, Dec. 2017.

[28] Y. Xiao, Z. Du, and W. Dong, "Smooth and near time-optimal trajectory planning of industrial robots for online applications," *Industrial Robot: An International Journal*, vol. 39, no. 2, pp. 169–177, Jan. 2012.

[29] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Robotics modelling, planning and control," in Springer-Verlag London Limited, 2010, ch. 4 Trajectory Planning, pp. 161–189, ISBN: 978-1-84628-641-4.

[30] Y. Chen, L. Li, and X. Ji, "Smooth and accurate trajectory planning for industrial robots," *Advances in Mechanical Engineering*, vol. 6, p. 342 137, 2014. eprint: https://doi.org/10.1155/2014/342137.

[31] F. Xie, L. Chen, Z. Li, and K. Tang, "Path smoothing and feed rate planning for robotic curved layer additive manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101 967, 2020.

[32] A. Olabi, R. Béarée, O. Gibaru, and M. Damak, "Feedrate planning for machining with industrial six-axis robots," *Control Engineering Practice*, vol. 18, no. 5, pp. 471–482, 2010.

[33] M. Amersdorfer and T. Meurer, “Equidistant tool path and cartesian trajectory planning for robotic machining of curved freeform surfaces,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2021.

[34] V. Nguyen, T. Cvitanic, and S. Melkote, “Data-Driven Modeling of the Modal Properties of a Six-Degrees-of-Freedom Industrial Robot and Its Application to Robotic Milling,” *Journal of Manufacturing Science and Engineering*, vol. 141, no. 12, Oct. 2019, 121006. eprint: https://asmedigitalcollection.asme.org/manufacturingscience/article-pdf/141/12/121006/6434102/manu_141_12_121006.pdf.

[35] X. Shi, F. Zhang, X. Qu, and B. Liu, “An online real-time path compensation system for industrial robots based on laser tracker,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, p. 1 729 881 416 663 366, 2016. eprint: https://doi.org/10.1177/1729881416663366.

[36] L. Biagiotti, L. Moriello, and C. Melchiorri, “A repetitive control scheme for industrial robots based on b-spline trajectories,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5417–5422.

[37] V. Nguyen *et al.*, “Precision robotic milling of fiberglass shims in aircraft wing assembly using laser tracker feedback,” *SAE International Journal of Aerospace*, vol. 15, no. 1, pp. 87–97, Jan. 2022.

[38] M. K. Gonzalez, N. A. Theissen, A. Barrios, and A. Archenti, “Online compliance error compensation system for industrial manipulators in contact applications,” *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102 305, 2022.

[39] Y. Altintas, “Manufacturing automation: Metal cutting mechanics, machine tool vibrations, and cnc design,” in 2nd ed. Cambridge University Press, 2012, ch. 2.7 Mechanics of Milling Process, pp. 35–47, ISBN: 978-0-521-17247-9.

[40] ——, “Manufacturing automation: Metal cutting mechanics, machine tool vibrations, and cnc design,” in 2nd ed. Cambridge University Press, 2012, ch. 7.3.5 In-Process Detection of Tool Breakage, pp. 330–333, ISBN: 978-0-521-17247-9.

[41] A. Ambiehl, S. Garnier, K. Subrin, and B. Furet, “New method for decoupling the articular stiffness identification: Application to an industrial robot with double encoding system on its 3 first axis,” in *IEEE International Conference on Intelligent Robots and Systems*, Cited By :3, vol. 2017-September, 2017, pp. 1478–1483.

[42] S.-F. Chen, “The 6/spl times/6 stiffness formulation and transformation of serial manipulators via the cct theory,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 3, 2003, 4042–4047 vol.3.

[43] *Kr 360-3; kr 500-3 specification*, KUKA Roboter GmbH, Augsburg, Germany, 2013.

[44] *Max 40 datasheet*, Suhner Machining, Lupfig, Switzerland.

[45] T. Cvitanic, “Accuracy improvement of industrial serial manipulators for manufacturing applications,” Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, Aug. 2021.

[46] G. Prytz, “A performance analysis of ethercat and profinet irt,” in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 408–415.

[47] S. Blair, *Beckhoff and twincat 3 system development guide*, University of Strathclyde, Glasgow, Scotland, 2015.

[48] *Kuka system software 8.2 operating and programming instructions for system integrators*, KUKA Roboter GmbH, Augsburg, Germany, 2012.

[49] M. Elbestawi, F. Ismail, and K. Yuen, “Surface topography characterization in finish milling,” *International Journal of Machine Tools and Manufacture*, vol. 34, no. 2, pp. 245–255, 1994.

[50] B. Jiang, T. Zhang, P. Zhao, and J. Zhao, “Dynamic milling force model for a milling cutter under vibration,” *The International Journal of Advanced Manufacturing Technology*, vol. 109, no. 5, pp. 1297–1317, Jul. 2020.

[51] G. U. Pelayo, “Modelling of static and dynamic milling forces in inclined operations with circle-segment end mills,” *Precision Engineering*, vol. 56, pp. 123–135, 2019.

[52] F. Wang, G. Bi, and F. Ning, “Modeling of dynamic milling forces considering the interlaminar effect during milling multidirectional cfrp laminate,” *Journal of Reinforced Plastics and Composites*, vol. 40, no. 11-12, pp. 437–449, 2021. eprint: https://doi.org/10.1177/0731684420971760.