Modelling and Compensation of Positioning Errors in Robotic Machining Systems

by

Jing Liu

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering
University of Manitoba
Winnipeg

Abstract

Industrial robots are widely used in manufacturing operations such as drilling, welding, and painting. In recent years, robots have gained significant attraction for aerospace machining (material removal) as well. Compared to conventional CNC machine tools, robots have higher versatility and a lower cost. However, robotic arms suffer from low structural stiffness, which leads to deflection errors under machining loads, as well as vibrations during high speed motions of the arm. To tackle these issues, this thesis presents a systematic framework for the prediction and pre-compensation of positioning errors to improve the accuracy of machining robots.

The kinematic and compliance models of a Stäubli RX-90 industrial robot are first formulated mathematically. In the developed model, it is assumed that the links of the robot are rigid, and therefore all flexibilities originate from the joints. A cutting force model is developed to predict the machining forces exerted on the robot's end-effector (i.e. milling tool) during 3-axis milling operations. The predicted forces are combined with the compliance model of the robot to determine the structural deflections during machining.

In order to reduce the residual vibrations of the robot in high speed motions, the concept of input shaping is introduced. It is shown that input shaping can distort the reference toolpath, which leads to the deviation of the actual machining path from the desired trajectory, also known as contour error.

Finally, a systematic framework is proposed to predict and compensate for positioning errors in robotic machining. The developed model can determine the contour errors caused by both cutting force-induced deflections and input shaping distortions. The model then adjusts the joint commands to compensate for the predicted errors. The entire framework has been programmed in MATLAB, and simulation results prove that the proposed framework can significantly reduce contour errors in robotic machining.

Table of Contents

List of Tables

List of Figures

Nomenclature

Symbols

$\boldsymbol{a}$	Approach direction of tool orientation
a	Link length
a	Edge contact length
A_j	Amplitude of the jth impulse in the shaping sequence
c	Feed rate
C_θ	Joint compliance matrix
C_x	Cartesian space compliance matrix
d	Link offset
D	Diameter of the milling cutter
$\boldsymbol{e}$	Tooltip tracking error
$\boldsymbol{F}$	External static force vector
F_a	Axial cutting force
F_c	Resultant cutting force
F_r	Radial cutting force
F_t	Tangential cutting force
F_x	Instantaneous horizontal force
F_y	Instantaneous normal force
F_z	Instantaneous axial force
G	Transfer function

h	Local chip thickness
J	Jacobian matrix
j	Tooth number
k_c	Cutting force coefficients
$\boldsymbol{M}$	Robot reference frame
$\boldsymbol{n}$	Normal direction of tool orientation
N	Number of teeth on the cutter
n	Spindle speed
$\boldsymbol{p}$	Position vector
P_a	Actual tooltip position
P_r	Reference tooltip position
$\boldsymbol{R}$	Orientation matrix
Rot	Rotational operator
$\boldsymbol{s}$	Sliding direction of tool orientation
s	Laplace variable
T	Transformation matrix
$\boldsymbol{T}$	Tool frame
t	time
T_d	Period of residual vibrations
t_j	Time delay of the *j*th impulse in the shaping sequence
$Trans$	Translational operator
V_t	Linear velocity
$\boldsymbol{W}$	Workpiece frame

Wq	Work in joint term
W_x	Work in Cartesian term
τ	Joint torque
α	Twist angle
β	Helix angle on the cutter
δ	Dirac impulse function
δX	Deflection of the end-effector
ζ	Damping ratio
φ	Immersion angle
$\boldsymbol{\varepsilon}$	Tooltip contour error
ω_d	Damped natural frequency
ω_n	Natural frequency
ω_t	Angular velocity
ϑ_i	Reference commands of the *i*th joint

Subscripts

a	axial
ac	actual
c	compensated
d	desired
m	mean value
r	radial
ref	reference

s, shp	shaped
t	tangential
w	represented in the workpiece frame {W}

Acronyms

CAD	Computer Aided Design
CE	Contour Error
CNC	Computer Numerical Control
D-H	Denavit-Hartenberg
SMC	Sliding Mode Control
TCP	Tool Center Point
ZVD	Zero-Vibration-Derivative shaper

Acknowledgment

I'm very grateful to my research supervisor, Dr. Matt Khoshdarregi. This thesis was completed under the detailed guidance of him. His patient instructions helped me open up my research ideas and prevented me from many detours. His genuine attitude towards academic research also taught me how to be a good student. All his mindful advice will benefit me throughout my life.

I would also like to express my gratitude to my labmates at the Intelligent Digital Manufacturing Laboratory and Dr. Ehsan Jalayeri for his professional advice. They were always there to help me when I had difficulties with my research. I am deeply inspired by them.

I also want to thank my beloved family: my parents, Mingqi and two cats, who provided me with all the support I needed to make my study a smooth one. When I was faced with the uncertainty of life choices, they are behind me all the time.

1 Introduction

Over the past few decades, computer numerical control (CNC) machines have been extensively used for industrial machining operations such as milling, drilling, and grinding. However, despite their high accuracy and stiffness, CNC machines are not always the best option since they are expensive to operate and have a relatively small workspace. Industrial robots have gained attraction for machining soft materials due to their cheaper price and larger workspace. The advancement in technology has led to increased mechanical capabilities and faster production rates of industrial robots, which has resulted in a worldwide growth in the use of robotics in automotive and metalworking industries.

One of the main advantages of machining robots is flexibility and versatility. It is easy for a robot to switch from one task to another, e.g. drilling, trimming, etc. Various operations can be conducted using the Tool Center Point (TCP). Machining robots can operate with multiple serial axes and thus have a larger workspace than CNC machines. A medium-sized machining robot, for example, has a working volume of 7 to 8 cubic meters. These strengths enable machining robots to process items of almost any size and complexity. Furthermore, lower costs, greater work volume, various add-on functions such as computer vision systems, all make industrial robots a more attractive option.

However, there are several fundamental challenges in robotic machining operations. One major difficulty is that the achievable accuracy and tolerance of final parts are limited. Robot accuracy is affected by many elements. In robotic machining applications, in particular, there are many sources of error that can be categorized into two major groups: geometrical errors and non-geometrical errors.

Geometrical errors, commonly known as metrology errors, are caused by the discrepancy between the nominal and actual dimensions of a robot's structural components (e.g. links). Assembly errors, e.g. in the form of misalignments of components, may also lead to positioning errors at the TCP. Geometrical errors stemming from imperfections in link dimensions and joint misalignments can typically be reduced by amending the model parameters used in the controller. There are many calibration methods that can directly detect and correct the difference between the actual and theoretical geometric parameters.

Non-geometrical errors originate from many sources. For example, external force or moment applied to the end-effector during contact tasks such as machining can result in significant deformation of robot joints and thus deflection errors at the TCP. The interaction between the machining tool and workpiece causes so-called compliance errors, which are the most common reason for the violation of part tolerance in robotic machining. Compliance errors can be larger than 1 mm in machining metals with industrial robots, which violates typical part tolerance (<0.2 mm). In order to improve the accuracy and quality of robotic machining operations, this thesis focuses on the modeling and compensation of compliance errors as one of the dominant sources of positioning errors in robotic milling.

Compliance errors depend substantially on the configuration of the robot. The size of the errors is related to the stiffness of the robot and the magnitude of the external forces exerted on it. The stiffness of commonly used articulated industrial robots is significantly smaller than 1 N/µm, and varies considerably in the workspace as a function of pose and reach. This thesis assumes each joint of the robot as a linear torsional spring since the flexibility of robots stems from the compliance of the gear reductions in the joints. Since the reflected force and torque on each joint depends on the kinematic chain, the deformation of each component varies in the workspace even for constant TCP forces. In machining processes, cutting forces change considerably along the machining toolpath due to variation in chip thickness (removed material). These factors complicate the compensation procedure significantly.

Another category of non-geometrical errors is the errors caused by the control system, e.g. due to the tracking errors between the commanded and actual joint angles. High speed industrial robots exhibit large vibrations during rapid movements. To reduce residual vibrations, reference joint commands are typically shaped (filtered) to avoid the excitation of the structural modes of a robot. However, in a path following task, filtering methods distort the nominal reference trajectory and lead to deviations between the actual and desired path.

This thesis proposes a method for modeling and pre-compensation of contour errors induced by command shaping methods.

In present-day research, there are typically two strategies for compensation of compliance errors in machining robots. If the compliance error is not very large and the robot controller is accessible with the software, the compliance error can be compensated through modifying the robot model as shown in Figure 1.1. This is a methodology for the on-line reinforcement of the requisite model.

In the actual motion of the robot, the conversion from the joint coordinate frames to the global Cartesian space is a rough approximation due to uncertainty in kinematic parameters. Therefore, the theoretical kinematic model and the stiffness matrix applied in the robot controller need to be fine-tuned to more accurately match the actual machining scenario. This approach is difficult to implement in practice because most robot controllers for commercial industrial robots have a closed architecture and do not allow modifications.

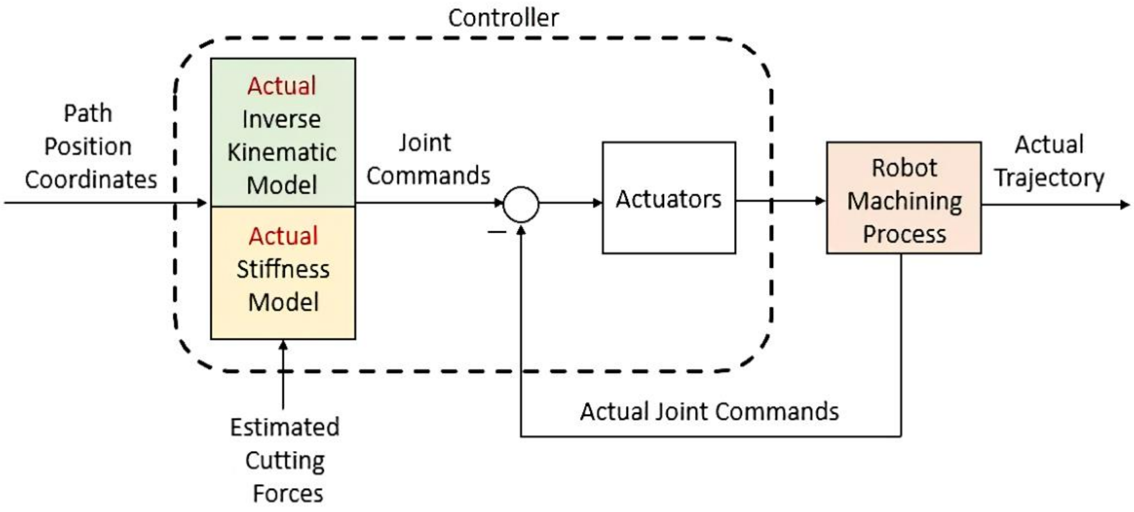


Figure 1.1: Robot Model Modification Method

Another option is to pre-compensate for contour errors by adjusting the input commands for the machining trajectory. This approach prior to the compensation involves some offline computation, as depicted in Figure 1.2. In the Cartesian space, the intended trajectory is interpolated into separate reference coordinate points. Before inputting the reference coordinates into the robot controller, the path is corrected by the appropriate contour error prediction and compensation model, which then brings the actual output trajectory closer to the desired trajectory.

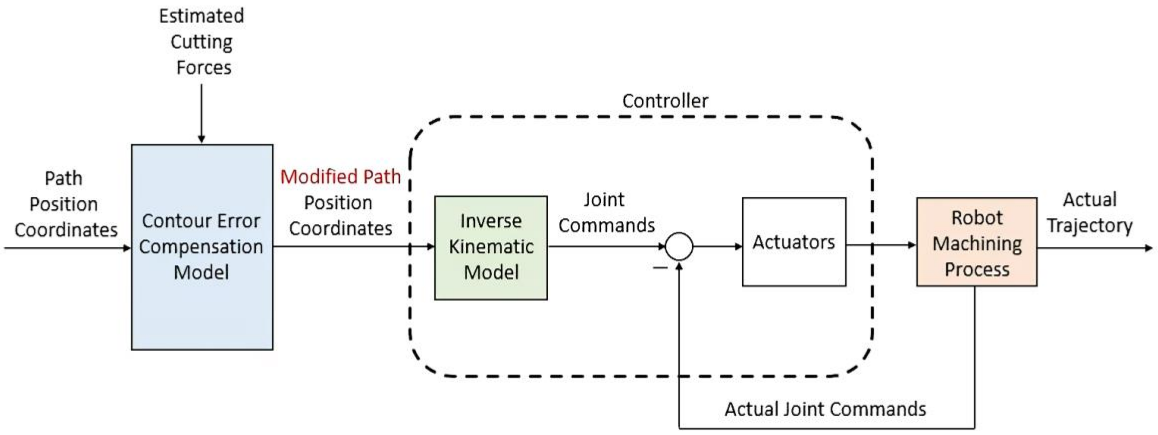


Figure 1.2: Input Trajectory Modification Method

This method is centered on the prediction of the deviation between the desired and actual path, which is the so-called contour error along the trajectory. The stiffness matrix describing the distribution of stiffness properties in the workspace, as well as the predictive model of the cutting forces applied to the end-effector during machining, are important factors in predicting profile errors. Both the aforementioned can be derived from algorithms in a specialized machining process.

The noticeable strength of the technology is that it can be used to compensate for all types of robotic contour errors, including geometric and non-geometric errors. Hence, it is very appealing for industrial purposes. To efficiently implement this method into robotic machining, this thesis presents substantial modifications to this technique, enabling it to effectively compensate for cutting forces as well as other sources of contour errors. The compensation methodology proposed in this thesis follows the approach shown in Figure 1.2, i.e. by modifying the path and correcting the input commands.

The chapters of the thesis are organized as follows: Chapter 2 reviews related literature and prior research on the modeling and error compensation of robotic machining systems. The kinematics and compliance modeling of industrial robots are presented in Chapter 3. Chapter 4 develops a systematic model to predict positioning errors in robotic machining. The proposed method for pre-compensation of contour errors is presented in Chapter 5, and the conclusions and possible future directions are provided in Chapter 6.

2 Literature Review

2.1 Overview

Since the invention of robots in the 1940s, the field of robotics has been constantly growing. After several decades of development, robotics has now become an essential part of industrial and personal environments. Robots are able to replace humans in dangerous, tedious, and boring jobs. Machining applications with industrial robots such as milling, drilling, and grinding are not uncommon nowadays. Industrial robots have some unique advantages over traditional machining systems. One robot can do various kinds of machining by simply replacing the end-effector with different tools, which is cheaper and more energy efficient. Robots are capable of working tirelessly and can be used in dangerous conditions with no concern for life safety. These advantages allow machining robots to work with items of almost any shape, size, and complexity.

However, the low stiffness compared to conventional machine tools will result in positioning errors. The usage of robots in many areas is limited due to their poor accuracy. The long kinematic chain and structure flexibility of robots are mainly responsible for the inaccuracy. In addition to the existence of contact forces during operations that can cause deformation of the robot structure and loss of positioning accuracy, there may be critical low-frequency chatter during robotic machining like milling and deburring, which is more likely to occur due to low stiffness. Moreover, when machining parts with complex curved surfaces, the depth and width of the cut are non-uniform, resulting in large variations in cutting forces, which will produce uneven deformation.

This chapter briefly reviews the state-of-art technologies and remained challenges in modeling and compensation of deflection errors in robotic machining. Section 2.2 lists the methodologies of modeling robot kinematics and compliance. Section 2.3 focuses on various sources of positioning errors in the machining process. Estimation and compensation approaches for improving contouring accuracy are presented in Section 2.4. A brief summary is presented in Section 2.5 at the end of the chapter.

2.2 Robot Kinematics and Compliance Modeling

The low stiffness of robots causes numerous challenges when it comes to performing machining operations, therefore, the stiffness of a robot must be studied and analyzed to improve its machining capabilities. Robot stiffness describes the ability of a robot to resist deformation in response to an external force.

It is important to determine which parts of the robot cause the most deformation. Since the Cartesian stiffness matrix at the end-effector varies with the robot's pose, it cannot be modeled as a fixed matrix. Instead, the stiffness matrix must be expressed as a function of joint stiffness and robot pose, and thus the kinematics of the robot must first be modeled.

Kinematic modeling and model identification are important topics in the robotic theory and have been widely discussed in the literature. Denavit and Hartenberg [2] introduced a systematized representation of the kinematic chains that are well known as the D-H convention, which is a comprehensive description of the kinematic properties of robot mechanisms provided by specific equations and matrix algebra.

Models based on the D-H parameters are quite intuitive and commonly used in the rigid kinematic analysis of robots. Simulation software can be developed based on the solutions of forward and inverse kinematics derived through the D-H convention [3]. Sicliano and Khatib [4] provide the latest knowledge on different aspects of rapidly evolving robotics, as well as the comprehensive methodology of the D-H convention and mutations of it. Different D-H convention variants are compared in the studies of sizing mechanical constraints guiding the system in the book by McCarthy and Soh [5].

Research on robotic machining was first reported in the 1990s. Although there has been continuous research on robotic machining all over the world since then, poor accuracy is still the key constraint for the widespread use of robotic machining [6]. As stated in [7], the stiffness of the robot is determined by the compliance of the joints, actuators, and so forth, which is not time-dependent. To keep the computation simple, it is commonly assumed that the linkage is rigid and considering only the joint stiffness since most robots have greater joint flexibility relative to the links.

Abele et al. [8] describe how to model the robot structure and identify its parameters by focusing on the stiffness of the system; they presented a method for computing Cartesian stiffness on account of the combination of polar moment of stiffness and Jacobian matrices. The Jacobian matrix represents the interrelationship between joint velocities and Cartesian velocities of the end-effector [9]. Several enhanced stiffness models that contain passive and active stiffness are proposed for flexible joints and rigid link robots [7][10]. The regular stiffness formula is improved by considering changes in the configuration of the robot and the resulting active stiffness.

Compliance is the inverse of stiffness, which refers to flexibility and suppleness in robotics. Both the terminology is intended to describe the compliant or non-stiff characteristics of robots. The Cartesian space stiffness matrix is usually used to define the industrial robot compliance model, which can be calculated by loading the robot in the desired configuration [11]. Optimization techniques based on static and dynamic stiffness models are introduced in [12]. The difference between the static and dynamic stiffness is whether to take the inertial and viscous effects of mass and damping components into account.

Bruder et al. [13] proposed a compliance identification method based on the Koopman operator theory. This method does not need the manual tuning of training parameters. Enhanced accuracy is achieved by using deflected circular trajectories to analyze the joint compliance [14].

The most commonly used analytical method for deducing the Cartesian space stiffness matrix is by calculating the Jacobian matrix [15], which can analyze the compliance effects of each joint. It is also possible to determine the stiffness parameters of the robot experimentally using static or dynamic loading [16].

2.3 Positioning Error Sources in Robotic Machining

There are two major sources of error in robotic machining: cutting force-induced errors and motion errors [17]. Typical motion errors are, for example, kinematic and dynamic errors, intrinsic to the controller of the robot. Cutting forces in excess of hundreds of Newton may easily lead to errors larger than 1mm, especially in milling operations [8]. As opposed to motion errors, machining forces only produce errors in contact cases and are a major contributor to surface errors. When robots are required to perform accurate operations such as machining, the positioning accuracy can be severely hampered by the deflection errors.

Any deformation of the robot's articulated joints will result in deviation of the robot's end positions. Even if the same force is applied, the robot tool tip displacements are varied based on the compliance model in different workspace positions. In addition to static deflections, robots may exhibit low- and high-frequency vibrations as well [18]. The displacements and oscillations are even more pronounced at high speeds, making the robot's trajectory accuracy worse.

The cutting force in industrial machining cannot be determined by a single formula, as it is not a fixed value. Its variation can be analyzed by modeling of the machining process. The final form of cutting force prediction models in different operations is introduced systematically in Altintas' studies in metal cutting mechanics [19]. Nikolaos and Aristomenis [20] proposed a computer-aided-design (CAD) based simulation method to calculate the cutting force of each edge in the machining process.

Cutting force prediction models have been extensively used and improved in today's research, e.g. in electrochemical machining [21], drilling [22], grinding [23], turning [24][25], and milling [26]. Both static and dynamic deformations in machining have been modeled by the mechanistic model proposed in [27] and [28].

Besides deflection errors generated by machining forces, contour errors caused by command shaping (filtering) are also considered in this thesis. These errors are caused by trajectory distortions due to a time delay in shaped reference commands. Input shaping is an efficient method first presented by Singer and Seering [29] to reduce inertial vibrations by creating a command signal that cancels its own vibration. Therefore, the vibration excited by the latter part of the command signal can cancel out the vibration of the former part. Input shaping filters have been successfully applied to the input commands to avoid residual vibrations in CNC machines [30] and industrial robots [31].

Input shaping is achieved by convolving original joint commands with a sequence of self-canceling impulses, called an input shaper. Input shapers effectively block the harmonics of the command that coincide with the structural modes of the robot. The derivative constraints used for zero-vibration-derivative (ZVD) shaper are derived mathematically in [32]. Input shapers avoid exciting the structural natural frequency, but they may distort the reference toolpath [30]. The length of the shaped command is not the same as the length of the original unshaped command. This time delay may cause contour errors in multi-axis toolpaths, which is one of the sources of positioning errors compensated in the thesis.

2.4 Contour Error Estimation and Compensation

The purpose of robot error compensation is to compensate for contour errors, taking into account all aspects of the process to improve actual manufacturing accuracy. Estimation of contour errors is a key factor in compensation. The calculation is either from the polar [33] or the Cartesian coordinates [34]. In either coordinate system, the deviation between the actual path and the desired path can be computed simultaneously and accurately to obtain an array of contour errors along the path.

Cheng and Lee [35] put forward a real-time algorithm for the calculation of contour errors in XY planes using the line-segment approximation. The estimation of spatial contour errors is more sophisticated. An iterative approach to calculate three-dimensional contour errors of an arbitrary smooth path is introduced by Khalick and Uchiyama [36]. This algorithm also works for complex contours with large curvature.

Many researchers have tried to compensate for the various errors in industrial manufacturing processes and have achieved great results. A motion control strategy that incorporates a combination of vibration avoidance and friction compensation technologies is introduced in [37], while sliding mode control (SMC) can be implemented to compensate for disturbances and significantly reduce tracking errors in machining [38].

Another option to compensate for friction characteristics is to use feedforward friction compensator, which is discussed both analytically and experimentally in [39]. Friction and torque are modeled in the compensator with simple numerical modeling, and the calculated friction torque is added to the command [40]. Moreover, a method based on the tool dynamics that can eliminate the intrinsic contour errors induced in the coordinate frame transformation is proposed in [41].

To compensate for machining errors, Huang et al. [42] proposed a dynamic method on the basis of the coarse-to-fine principle, which can track the path at high speed. Khoshdarregi et al. [43] correct the axis commands to pre-compensate the predicted contour errors in conventional CNC machines. Similarly, the deviation of the tooltip is also pre-compensated in [44]. Zeng et al. [45] proposed a positional error compensation method that does not require identification or modification of the kinematic parameters. The exact amount of compensation of each joint is calculated using a body frame in [46].

Schneider et al. [47] introduced a modular approach that can be applied for compensation both offline and online. To improve the efficiency of compensation, a new method that contains two-dimensional manifolds is proposed [48]. The two-dimensional manifolds can reduce the workspace's dimensionality. An artificial neural network is used by Nguyen et al. [49] to compensate for the non-geometric errors in robotic machining.

Although various compensation methods mentioned above are efficient, most of them are applied to conventional CNC machine tools. There is still a need to develop compensation strategies for robotic machining systems. This thesis provides an analytical method for pre-compensating contour errors, which can be utilized to compensate for various sources of error in robotic machining.

2.5 Summary

A brief review of the topics of modeling and compensation of positioning errors in robotic machining is presented in this chapter. Several methodologies are proposed to develop the kinematic and compliance model of industrial robots. Various sources of positioning errors are discussed, in which cutting force-induced errors are most significant. Command shaping techniques for vibration avoidance are reviewed; such techniques lead to a considerable delay to the path, which also induces contour errors. To estimate such errors, prediction models for cutting forces and tooltip displacements in real-time are introduced. This thesis proposes an integrated technique to compensate for contour errors due to both forces induced deflections and trajectory distortions caused by commands shaping techniques.

3 Modeling of Robot Kinematics and Compliance

3.1 Introduction

Robot kinematics is the study of the movement of the multi-degree of freedom structure of the robotic system. By applying geometry methodologies to the robot, the dimensions of the kinematic chain can be related by a set of kinematic equations, which is significant in planning and controlling robot movement. The positioning accuracy of industrial robotic manipulators depends upon a kinematic model that describes the robot geometry in a parametric form [50].

Besides the kinematic model, the compliance model of the industrial robot is also a key factor in the accurate prediction of the end-effector coordinates. Compliance is opposite to stiffness; compliance model of a robot is a numerical reflection of the flexibility and suppleness of the robot, and can intuitively and reliably predict the deformation of the robot structure under any external force.

This chapter is organized as follows: Section 3.2 briefly introduces the configuration of a Stäubli RX-90 industrial robot studied in this thesis. Section 3.3 develops the forward and inverse kinematic model of the robot and provides validations. Section 3.4 derives the compliance model of the robot with some assumptions: The links of the robot are assumed as infinitely stiff and the joints are providing pure rotation. The investigations presented in this chapter are summarized in Section 3.5.

3.2 Experimental Platform: Stäubli RX-90 Industrial Robot

In this thesis, modeling and analyses have been performed based on a Stäubli RX-90 industrial robot shown in Figure 3.1. This robot has 6 rotary joints and is a typical example of 6-degree of freedom robotic manipulators used widely in the industry.

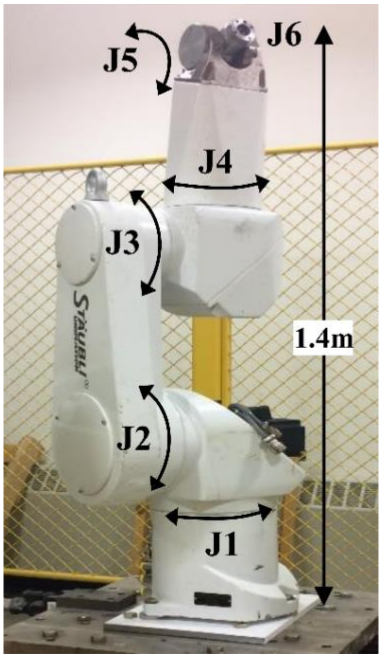


Figure 3.1: Configuration of Stäubli RX-90 Industrial Robot [51].

The nominal payload is 6 kg and the maximum payload is 11 kg. The arm assembly has a reach of 985 mm with a repeatability of 0.02 mm. The overall dimensions of the Stäubli RX-90 industrial robot are shown in Figure 3.2.

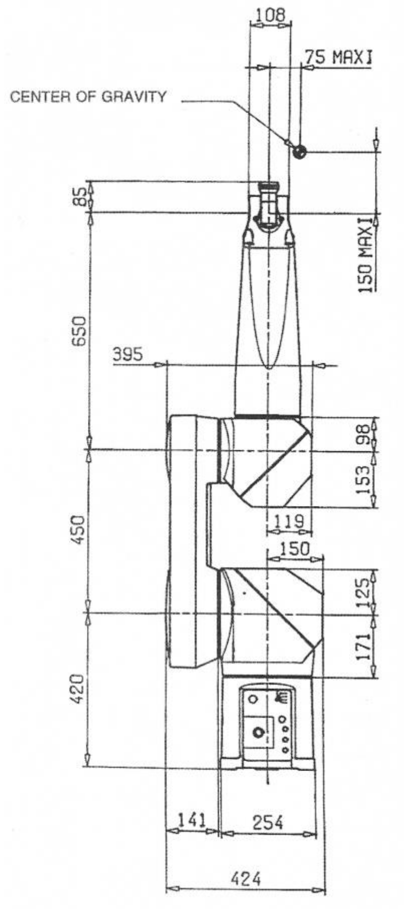


Figure 3.2: Overall Dimensions (mm) of Stäubli RX-90 Industrial Robot [51].

3.3 Kinematic Modeling

An industrial robot is made up of several linkages, which are joined together by a series of joints. If a coordinate system is fixed to each link, a homogeneous transformation matrix can be used to relate the positions and orientations between these coordinate systems. This transformation matrix is commonly used to solve kinematic problems in industrial robots.

There are two cases of kinematic problems: forward and inverse kinematics. A forward kinematic problem computes the position and orientation of the end-effector with known joint parameters of the robot. Forward kinematics is typically straightforward for serial robots and leads to a unique solution for the end-effector position given a set of joint angles.

An inverse kinematic problem is to compute the angles of each joint that achieve a specified position and orientation of the end-effector with knowledge about the dimensions of the links. Inverse kinematics is typically more complicated than forward kinematics for serial robots. General robot configurations with rotary and translational joints normally require cumbersome numerical solutions. However, there are certain robot designs that have analytic inverse kinematics, although these robots still face multi-solution problems, which require further validity checks and optimization.

3.3.1 Transformation Parameters and Matrices

The study of the robot's posture and motion is key to the overall kinematics of a robot, which is closely related to the length of the links, their interrelationship, and the motion types of each joint. Therefore, when investigating the position of the end-effector related to the base, it is necessary to analyze the interaction between the two connected joints, that is, to attach the reference coordinate frame to the links of a spatial kinematic chain.

A common convention for selecting reference frames in robotics is the Denavit and Hartenberg (D-H) convention, which was introduced by Jacques Denavit and Richard S. Hartenberg [2].

The reference coordinate frames of Stäubli RX-90 industrial robot are laid out as provided in Figure 3.3, which follows the D-H convention. Z1 …… Z6 are the axes of rotation for the joints, within which Joint 2 rotates perpendicular to Joint 1. Joint 4 intersects with Joint 5 and Joint 6 to form a “wrist” at the end of the robot arm.

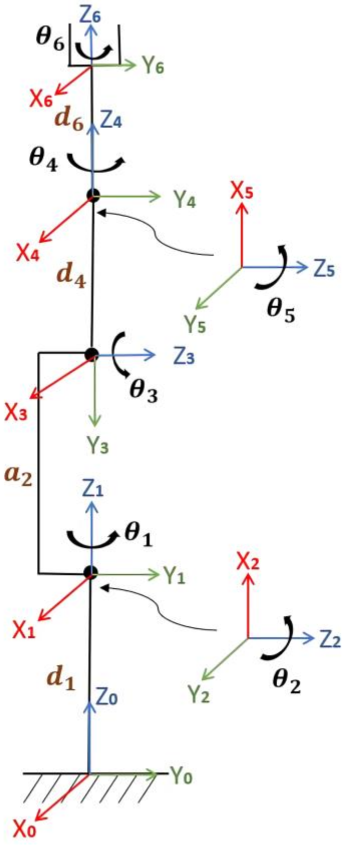


Figure 3.3: D-H Convention Frame Layout for Stäubli RX-90.

The following four transformation parameters, known as D-H parameters, are used to translate the coordinates from one link to the next:

d_i: link offset. The distance along the common axis.

θ_i: joint angle. The amount of rotation about the common axis.

a_i: link length. The length of the common normal.

α_i: twist angle. The amount of rotation about the common normal axis.

Eighteen parameters are required to completely establish the transformation frame of a 6-joint robot. The D-H parameters of the Stäubli RX-90 industrial robot are provided in Table 3.1, where $i = 1,2,\dots,6$ is the number of links. Table 3.2 provides the constant values of the non-zero link lengths and link offsets.

Table 3.1: D-H Parameters of Stäubli RX-90 Industrial Robot

$\boldsymbol{i}$	$\boldsymbol{a_{i-1}}$ [deg]	$\mathbf{a}_{\boldsymbol{i-1}}$ [mm]	$\boldsymbol{d_i}$ [mm]	$\boldsymbol{\theta_i}$ [deg]
1	0	0	d_1	θ_1
2	-90	0	0	θ_2 - 90
3	0	a_2	0	θ_3 + 90
4	90	0	d_4	θ_4
5	-90	0	0	θ_5
6	90	0	d_6	θ_6

Table 3.2: Constant D-H Parameter Values

Parameter	$\mathbf{d_1}$	$\mathbf{a_2}$	$\mathbf{d_4}$	$\mathbf{d_6}$
Values [mm]	420	450	650	85

The coordinate transformations along a serial robot consisting of n links form the kinematic equations of the robot. According to the modified D-H convention, the transformation matrix is given by the following order of operations [2]:

$$
\begin{aligned}
{}^{n-1}T_n &= Rot_{x_{n-1}}(\alpha_{n-1}) \cdot Trans_{x_{n-1}}(\alpha_{n-1}) \cdot Rot_{z_n}(\theta_n) \cdot Trans_{z_n}(d_n) \\
&= \begin{bmatrix} \cos\theta_n & -\sin\theta_n & 0 & a_{n-1} \\ \sin\theta_n \cos\alpha_{n-1} & \cos\theta_n \cos\alpha_{n-1} & -\sin\alpha_{n-1} & -d_n \sin\alpha_{n-1} \\ \sin\theta_n \sin\alpha_{n-1} & \cos\theta_n \sin\alpha_{n-1} & \cos\alpha_{n-1} & d_n \cos\alpha_{n-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.1}
$$

This matrix represents the pose (position and orientation), where the first three columns describe rotation and the last column represents the translation of a frame. Once the transformation matrix is determined by known parameters, the D-H notation gives a standard homogeneous transformation methodology to derive the kinematic equations of the robot.

$$^{0}T_{6} = {^{0}T_{1}} * {^{1}T_{2}} * {^{2}T_{3}} * {^{3}T_{4}} * {^{4}T_{5}} * {^{5}T_{6}} \tag{3.2}$$

Eq. (3.2) is the required forward kinematic equation. The final expression of it is a square matrix of size 4 as follow:

$$^{0}T_{6} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The forward kinematic equation is composed of three-unit vectors ***n***, ***s***, ***a*** that describe the tool orientation in the so-called approach, sliding, and normal directions. The last column is the position vector ***p***, which presents the end-effector coordinates in the robot frame.

3.3.2 Forward Kinematic Model

According to Eq. (3.1) and the D-H parameters in Table 3.1, the transformation from each coordinate system to the previous coordinate system of Stäubli RX-90 industrial robot can be described as follow:

$$
\begin{aligned}
&{}^{0}T_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
&&{}^{1}T_2 = \begin{bmatrix} \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&{}^{2}T_3 = \begin{bmatrix} -\sin\theta_3 & -\cos\theta_3 & 0 & a_2 \\ \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
&&{}^{3}T_4 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ 0 & 0 & -1 & -d_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&{}^{4}T_5 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\theta_5 & -\cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
&&{}^{5}T_6 = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ 0 & 0 & -1 & -d_6 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.4}
$$

The D-H parameters of the Stäubli RX-90 industrial robot presented in Table 3.2 are substituted into the above matrices. ${}^{0}T_6$ is then formed by the multiplication of all individual transformation matrices as shown in Equation (3.2). The final expression of the forward kinematic model in Eq. (3.3) is then:

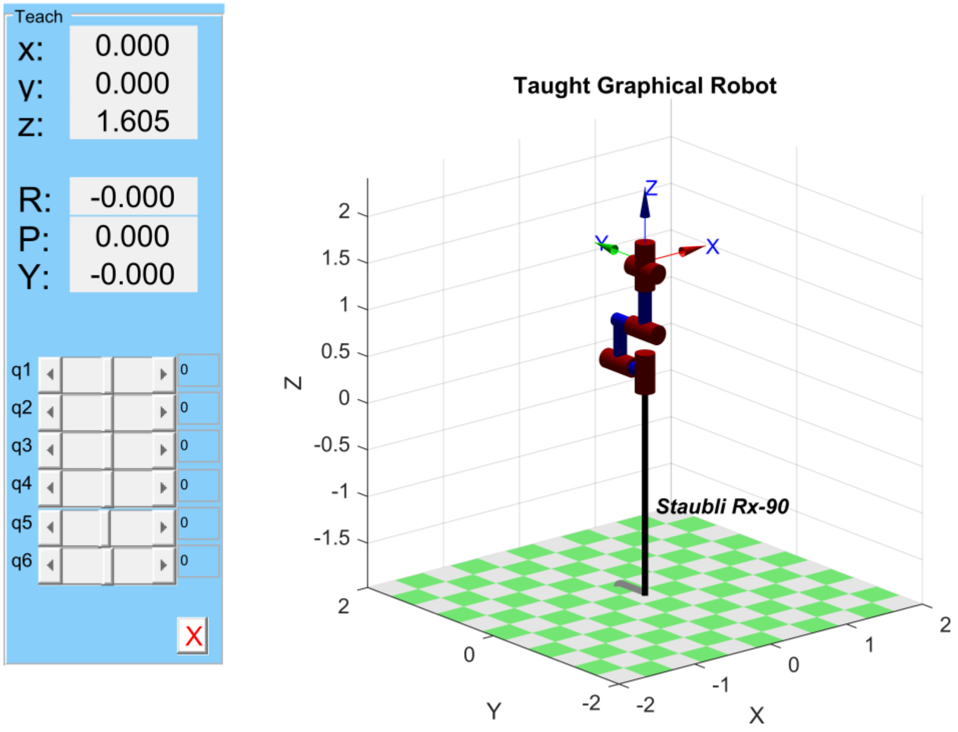
$$
\begin{aligned}
n_x &= -s_1\left(c_4s_6 + s_4c_5c_6\right) - c_1\left[c_{23}\left(s_4s_6 - c_4c_5c_6\right) + s_{23}s_5c_6\right] \\
n_y &= c_1\left(c_4s_6 + s_4c_5c_6\right) - s_1\left[c_{23}\left(s_4s_6 - c_4c_5c_6\right) + s_{23}s_5c_6\right] \\
n_z &= s_{23}\left(s_4s_6 - c_4c_5c_6\right) - c_{23}s_5c_6 \\
s_x &= -s_1\left(c_4c_6 + s_4c_5s_6\right) - c_1\left[c_{23}\left(s_4c_6 + c_4c_5s_6\right) - s_{23}s_5s_6\right] \\
s_y &= c_1\left(c_4c_6 - s_4c_5s_6\right) - s_1\left[c_{23}\left(s_4c_6 + c_4c_5s_6\right) - s_{23}s_5s_6\right] \\
s_z &= s_{23}\left(s_4c_6 + c_4c_5s_6\right) + c_{23}s_5s_6 \\
a_x &= c_1\left(s_{23}c_5 + c_{23}c_4s_5\right) - s_1s_4s_5 \\
a_y &= s_1\left(s_{23}c_5 + c_{23}c_4s_5\right) + c_1s_4s_5 \\
a_z &= c_{23}c_5 - s_{23}c_4s_5 \\
p_x &= c_1\left[s_{23}\left(d_4 + d_6c_5\right) + a_2s_2 + d_6c_{23}c_4s_5\right] - d_6s_1s_4s_5 \\
p_y &= s_1\left[s_{23}\left(d_4 + d_6c_5\right) + a_2s_2 + d_6c_{23}c_4s_5\right] + d_6c_1s_4s_5 \\
p_z &= d_1 + c_{23}(d_4 + d_6c_5) + a_2c_2 - d_6s_{23}c_4s_5
\end{aligned}
\tag{3.5}
$$

where $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $c_{ij} = \cos(\theta_i + \theta_j)$, $s_{ij} = \sin(\theta_i + \theta_j)$.

When $\forall\theta_i = 0$ (i=1, 2…6), the end-effector position and orientation of Stäubli RX-90 industrial robot can be computed by Eq. (3.5) as

$$ {}^{0}T_{6}\left(\theta_i = 0\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1.605 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6} $$

The developed forward kinematic model has been validated by comparing the results against those given by MATLAB’s robotic toolbox. In this case, every joint is set to 0, and the visualization of the graphical robot is shown in Figure 3.4.



Teach

x: 0.000
y: 0.000
z: 1.605

R: -0.000
P: 0.000
Y: -0.000

q1 0
q2 0
q3 0
q4 0
q5 0
q6 0

Figure 3.4: Visualization of the Robot at $\forall\boldsymbol{\theta}_{\boldsymbol{i}} = \mathbf{0}$.

The end-effector position and orientation are provided on the output interface: End-effector coordinate is (0, 0, 1.605) and pitch, roll, and yaw angles are $(0, 0, 0)$. Hence the tool orientation and position vectors given by the MATLAB robotic toolbox are as following:

$$ {}^{0}R_{6}\left(\theta_i = 0\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7} $$

$$ {}^{0}D_{6}\left(\theta_i = 0\right) = \begin{bmatrix} 0 \\ 0 \\ 1.605 \end{bmatrix} \tag{3.8} $$

It is obvious that Eq. (3.7) and Eq. (3.8) equal to Eq. (3.6) at corresponding columns and rows. Therefore, the derived forward kinematic code of Stäubli RX-90 industrial robot is validated.

3.3.3 Inverse Kinematic Model

Inverse kinematics is the problem of solving for the joint parameters given a known pose at the end-effector. In this section, the inverse kinematics of the Stäubli RX-90 industrial robot is investigated. A closed-form analytic solution exists in this case. Therefore, pure algebraic analytical methods are used instead of numerical methods; this greatly simplifies the computation.

The position of the origin of the wrist frame relative to the base, i.e. D^{w}, can be determined as

$$ D^{w} = \begin{bmatrix} p_{xw} \\ p_{yw} \\ p_{zw} \end{bmatrix} = D^{t} - \begin{bmatrix} d_6 a_x \\ d_6 a_y \\ d_6 a_z \end{bmatrix} = \begin{bmatrix} p_x - d_6 a_x \\ p_y - d_6 a_y \\ p_z - d_6 a_z \end{bmatrix} \tag{3.9} $$

where the parameters are defined in Table 3.2 and Eq. (3.5). D^{w} represents the position of the wrist origin, which coincides with the origin of Link 4. Hence it can be written as the 4th column of the ${}^{0}T_{4}$ transformation matrix, i.e.

$$\begin{bmatrix} D^w \\ 1 \end{bmatrix} = [{}^0T_4]_4 = \begin{bmatrix} p_{xw} \\ p_{yw} \\ p_{zw} \\ 1 \end{bmatrix} = \begin{bmatrix} d_4c_1s_{23} + a_2c_1s_2 \\ d_4s_1s_{23} + a_2s_1s_2 \\ d_1 + d_4c_{23} + a_2c_2 \\ 1 \end{bmatrix} \tag{3.10}$$

To solve the inverse kinematics, first pre-multiply Eq. (3.2) by $\left({}^0T_1\right)^{-1} = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, the positional equations associated with Stäubli RX-90 industrial robot can now be obtained as

$$[{}^1T_2 * {}^2T_3 * {}^3T_4]_4 = [{}^1T_4]_4 = \left({}^0T_1\right)^{-1} * \begin{bmatrix} D^w \\ 1 \end{bmatrix} \tag{3.11}$$

Equating the left-hand side and right-hand sides of Eq. (3.11) yields

$$\begin{bmatrix} a_2 + d_4c_3 \\ d_4s_3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} (p_{xw}c_1 + p_{yw}s_1)s_2 + \left(p_{zw} - d_1\right)c_2 \\ (p_{xw}c_1 + p_{yw}s_1)c_2 - \left(p_{zw} - d_1\right)s_2 \\ p_{yw}c_1 - p_{xw}s_1 \\ 1 \end{bmatrix} \tag{3.12}$$

Similarly, by pre-multiplying these terms with $({}^1T_2)^{-1}$ and $({}^2T_3)^{-1}$, and following trigonometric relationships, $\theta_1, \theta_2, \theta_3$ can be solved as

$$\theta_1 = Atan2\left(\frac{-p_{xw}}{p_{yw}}\right) + Atan2\left(\pm\sqrt{\frac{{p_{yw}}^2 + {p_{xw}}^2}{0}}\right) \tag{3.13}$$

$$\theta_2 = Atan2\left[\frac{(p_{xw}c_1 + p_{yw}s_1)(a_2 + d_4c_3) - \left(p_{zw} - d_1\right)d_4s_3}{(p_{xw}c_1 + p_{yw}s_1)d_4s_3 + \left(p_{zw} - d_1\right)(a_2 + d_4c_3)}\right] \tag{3.14}$$

$$\theta_3 = Atan2\left\{\frac{\pm\sqrt{4a_2^{\,2}d_4^{\,2} - [p_{xw}^{\,2} + p_{yw}^{\,2} + \left(p_{zw} - d_1\right)^2 - a_2^{\,2} - d_4^{\,2}]^2}}{p_{xw}^{\,2} + p_{yw}^{\,2} + \left(p_{zw} - d_1\right)^2 - a_2^{\,2} - d_4^{\,2}}\right\} \tag{3.15}$$

Both θ_1 and θ_3 give dual-valued functions, so there will be 4 different combinations of θ_1 and θ_3. Hence θ_2 also leads to 4 different values for any desired wrist position.

To solve for the final three joint values, i.e. θ_4, θ_5 and θ_6, the four distinct solutions obtained for the first three joints are represented by their corresponding matrix, $({}^0T_3)^{-1}$, i.e.

$$\left({}^0T_3\right)^{-1} = \begin{bmatrix} c_1c_{23} & s_1c_{23} & -s_{23} & d_1s_{23} + a_2s_3 \\ -c_1s_{23} & -s_1s_{23} & -c_{23} & d_1c_{23} + a_2c_3 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

Since

$$\left({}^0T_3\right)^{-1} * {}^0T_6 = {}^3T_6\left(\theta_4, \theta_5, \theta_6\right), \tag{3.17}$$

substituting Eq. (3.16) into Eq. (3.17) yields

$$\begin{aligned} &\begin{bmatrix} c_1c_{23} & s_1c_{23} & -s_{23} & d_1s_{23} + a_2s_3 \\ -c_1s_{23} & -s_1s_{23} & -c_{23} & d_1c_{23} + a_2c_3 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -s_4c_6 - c_4c_5s_6 & c_4s_5 & d_6c_4s_5 \\ s_5c_6 & -s_5s_6 & -c_5 & -d_4 - d_6c_5 \\ c_4s_6 + s_4c_5c_6 & c_4c_6 - s_4c_5s_6 & s_4s_5 & d_6s_4s_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.18}$$

The left-hand side of Eq. (3.18) is known based on the previously derived equations. Therefore, the wrist angles can be obtained as

$$\theta_4 = Atan2\left(\frac{c_1a_y - s_1a_x}{a_xc_1c_{23} + a_ys_1c_{23} - a_zs_{23}}\right) \text{ and } \theta_4 \pm 180^\circ \tag{3.19}$$

$$\theta_5 = Atan2\left(\frac{\pm\sqrt{(c_1a_y - s_1a_x)^2 + \left(a_xc_1c_{23} + a_ys_1c_{23} - a_zs_{23}\right)^2}}{a_xc_1s_{23} + a_ys_1s_{23} - a_zc_{23}}\right) \tag{3.20}$$

$$\theta_6 = Atan2\left(\frac{c_1s_{23}s_x + s_1s_{23}s_y + c_{23}s_z}{-c_1s_{23}n_x - s_1s_{23}n_y - s_{23}n_z}\right) \text{ and } \theta_6 \pm 180^\circ \tag{3.21}$$

θ_4, θ_5 and θ_6 provide dual values for any given θ_1, θ_2 and θ_3 but not independent of one another. If θ_4, θ_5 and θ_6 represent one solution, then $\theta_4 \pm 180^\circ$, $-\theta_5$ and $\theta_6 \pm 180^\circ$ will represent the 'flipped' solution. In summary, the inverse kinematics, i.e. θ_i s, can be computed by Eq. (3.13)-(3.15) and (3.19)-(3.21).

To validate the developed inverse kinematic equations of the Stäubli RX-90 industrial robot, a MATLAB code was developed to solve the inverse kinematic equations. It is worth noting that in the code, the end-effector is assumed to be perpendicular to the ground to represent 3-axis robotic milling. In this case, joint 4 remains zero in all solutions. As a test point, joint variables $\theta_i = [70^\circ\ 20^\circ\ 60^\circ\ 0^\circ\ 50^\circ\ 100^\circ](i = 1, 2, \ldots, 6)$ were first substituted into Eq. (3.5), leading to ${}^0T_6$ as:

$${}^0T_6 = \begin{bmatrix} 0.8872 & -0.3797 & 0.2620 & 0.2716 \\ -0.4417 & -0.5355 & 0.7198 & 0.7461 \\ -0.1330 & -0.7544 & -0.6428 & 0.9557 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

The inverse kinematic model was then used to determine the joint variables. The given solutions are shown in Table 3.3. There are two possible solutions in the workspace.

Table 3.3: Joint Variables Solved by Inverse Kinematic Model

	θ_1	θ_2	θ_3	θ_4	θ_5
1	70°	20°	60°	0°	100°
2	-110°	-20°	-60°	0°	-100°

From Table 3.3, it can be seen that the first solution is the same as the given joint angles. The other solution can be tested by substituting Eq. (3.5), which also gives a square matrix that is consistent with the given pose. This suggests that the derived inverse kinematic model is correct.

3.4 Compliance Modeling

There are two values that depict the machining performance of an industrial robot, pose repeatability, and pose accuracy [52]. Pose repeatability describes a robot's ability to return to the same posture, while pose accuracy describes the capability of a robot to precisely reach an ideal position in three-dimensional space. Nowadays, many industrial robots commonly have excellent pose repeatability, but poor accuracy.

For industrial robots with large driving torques, the geometric and elastic deformations of the system render it difficult to achieve high positioning accuracy. The deformation of the robot, when subjected to external loads, is mainly due to the link's lack of rigidity. There are many parameters that determine the stiffness of an industrial robot, e.g. compliance of its joints, geometric and material properties of the links, etc.

Normally, deformations caused by joint compliance is the most dominant type of deformity in robotic machining due to the long kinematic chain. Therefore, the following analyses of the robot compliance model consider only joint compliance, and the links are considered rigid.

3.4.1 Jacobian Matrix

When the joints of the robot move with certain known velocities, the velocity of the end-effector can be determined based on the kinematic model. Jacobian matrix provides a relationship between the joint velocities and the Cartesian velocities of the end-effector. In a 6-axis industrial robot, the Jacobian matrix has six columns. There are a variety of methods for deriving the Jacobian matrix such as differential and vector methods. By definition, the Jacobian matrix contains the first-order partial derivatives of a vector-valued function. Consider a set of equations with ***n*** variables $x_1, \ldots, x_n$, written as:

$$\begin{cases} y_1 = f_1\left(x_1, x_2, x_3, x_4, x_5, x_6\right) \\ y_2 = f_2\left(x_1, x_2, x_3, x_4, x_5, x_6\right) \\ \qquad \vdots \\ y_6 = f_6\left(x_1, x_2, x_3, x_4, x_5, x_6\right) \end{cases} \tag{3.23}$$

Eq. (3.23) can be written in the vector form as:

$$F = F\left(X\right) \tag{3.24}$$

Taking the differential:

$$\begin{cases} \delta Y_1 = \dfrac{\delta f_1}{\delta x_1}\delta x_1 + \dfrac{\delta f_1}{\delta x_2}\delta x_2 + \ldots + \dfrac{\delta f_1}{\delta x_6}\delta x_6 \\ \delta Y_2 = \dfrac{\delta f_2}{\delta x_1}\delta x_1 + \dfrac{\delta 2}{\delta x_2}\delta x_2 + \ldots + \dfrac{\delta f_2}{\delta x_6}\delta x_6 \\ \qquad \vdots \\ \delta Y_6 = \dfrac{\delta f_6}{\delta x_1}\delta x_1 + \dfrac{\delta f_6}{\delta x_2}\delta x_2 + \ldots + \dfrac{\delta f_6}{\delta x_6}\delta x_6 \end{cases} \tag{3.25}$$

which in the vector form can be expressed as:

$$dY = \frac{\delta F}{\delta x} dX \tag{3.26}$$

Jacobian matrix is the 6x6 partial derivative matrix $\frac{\delta F}{\delta x}$ in Eq. (3.26). Hence the above equation can be rewritten as:

$$dY = J\left(X\right)dX \tag{3.27}$$

Dividing both sides of the above equation with a time differential dt yields:

$$\dot{Y} = J\left(X\right)\dot{X} \tag{3.28}$$

Eq. (3.28) gives the relation between velocities in the X frame and velocities in the Y frame. Note that the Jacobian matrix is a time-varying transformation.

In robotics, the Jacobian matrix helps to determine the velocity of the end-effector when the joint velocities are known. Based on Eq. (3.28):

$$\dot{X} = J\left(q\right)\dot{q} \tag{3.29}$$

where $J(q)$ is the robot's Jacobian matrix, and $\dot{X}$ and $\dot{q}$ are column vectors representing the end-effector and joint velocities, respectively. The end-effector velocity comprises two parts: linear velocity (V_t) and angular velocity (ω_t). Hence the above equation can be expanded as:

$$\begin{bmatrix} v_t \\ \omega_t \end{bmatrix} = \left[J\right] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_6 \end{bmatrix} \tag{3.30}$$

The Jacobian corresponding to linear velocity, i.e. J_v, is the first derivative of the position functions for x, y, and z of the end-effector, and can be derived as:

$$J_v = \begin{bmatrix} \frac{\delta x}{\delta q_1} & \frac{\delta x}{\delta q_2} & \frac{\delta x}{\delta q_3} & \cdots & \cdots & \frac{\delta x}{\delta q_n} \\ \frac{\delta y}{\delta q_1} & \frac{\delta y}{\delta q_2} & \frac{\delta y}{\delta q_3} & \cdots & \cdots & \frac{\delta y}{\delta q_n} \\ \frac{\delta z}{\delta q_1} & \frac{\delta z}{\delta q_2} & \frac{\delta z}{\delta q_3} & \cdots & \cdots & \frac{\delta z}{\delta q_n} \end{bmatrix} \tag{3.31}$$

In the kinematic model of Stäubli RX-90 industrial robot derived in the previous section, let the transformation between frame i and the base be described as

$${}^{0}T_i = \begin{bmatrix} {}^{0}i_i & {}^{0}j_i & {}^{0}k_i & {}^{0}p_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.32}$$

The number of columns in the Jacobian matrix is equal to the number of joints of the robot, i.e.

$$J(\theta) = [J_1 \ J_2 \cdots J_6] \tag{3.33}$$

The column vectors are stated as [57]:

$$J_i = \begin{bmatrix} {}^{0}k_{i-1} \times \left({}^{0}p_n - {}^{0}p_{i-1} \right) \\ {}^{0}k_{i-1} \end{bmatrix} \tag{3.34}$$

Substituting the vectors from Eq. (3.32) into Eq. (3.34), the column vectors of the Jacobian matrix J_i, $i = 1,2, \ldots \ldots, 6$, are obtained as

$$J_1 = \begin{bmatrix} -s_1 \left[s_{23} \left(d_4 + d_6 c_5 \right) + a_2 s_2 + d_6 c_{23} c_4 s_5 \right] - d_6 c_1 s_4 s_5 \\ c_1 \left[s_{23} \left(d_4 + d_6 c_5 \right) + a_2 s_2 + d_6 c_{23} c_4 s_5 \right] - d_6 s_1 s_4 s_5 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.35}$$

$$J_2 = \begin{bmatrix} c_1\left[c_{23}\left(d_4 + d_6 c_5\right) + a_2 c_2 - d_6 s_{23} c_4 s_5\right] \\ s_1\left[c_{23}\left(d_4 + d_6 c_5\right) + a_2 c_2 - d_6 s_{23} c_4 s_5\right] \\ -s_{23}\left(d_4 + d_6 c_5\right) - a_2 s_2 - d_6 c_{23} c_4 s_5 \\ -s_1 \\ c_1 \\ 0 \end{bmatrix} \tag{3.36}$$

$$J_3 = \begin{bmatrix} c_1\left[c_{23}\left(d_4 + d_6 c_5\right) - d_6 s_{23} c_4 s_5\right] \\ s_1\left[c_{23}\left(d_4 + d_6 c_5\right) - d_6 s_{23} c_4 s_5\right] \\ -s_{23}\left(d_4 + d_6 c_5\right) - d_6 c_{23} c_4 s_5 \\ -s_1 \\ c_1 \\ 0 \end{bmatrix} \tag{3.37}$$

$$J_4 = \begin{bmatrix} d_6 s_5\left(s_1 c_4 + c_1 c_{23} s_4\right) \\ d_6 s_5\left(c_1 c_4 - s_1 c_{23} s_4\right) \\ d_6 s_{23} s_4 s_5 \\ s_{23} c_1 \\ s_{23} s_1 \\ c_{23} \end{bmatrix} \tag{3.38}$$

$$J_5 = \begin{bmatrix} d_6\left(c_1 c_{23} c_4 c_5 - c_1 s_{23} s_5 - s_1 s_4 c_5\right) \\ d_6\left(s_1 c_{23} c_4 c_5 - s_1 s_{23} s_5 + c_1 s_4 c_5\right) \\ -d_6 (c_{23} s_5 + s_{23} c_4 c_5) \\ -c_1 c_{23} s_4 - s_1 c_4 \\ -s_1 c_{23} s_4 + c_1 c_4 \\ s_{23} s_4 \end{bmatrix} \tag{3.39}$$

$$J_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c_1 s_{23} c_5 - s_1 s_4 s_5 + c_1 c_{23} c_4 s_5 \\ s_1 s_{23} c_5 + c_1 s_4 s_5 + s_1 c_{23} c_4 s_5 \\ c_{23} c_5 - s_{23} c_4 s_5 \end{bmatrix} \tag{3.40}$$

Substituting Eq. (3.35)-(3.40) into Eq. (3.33) provides the Jacobian matrix of Stäubli RX-90 industrial robot.

3.4.2 Compliance Formulation

The Jacobian matrix relates the Cartesian and joint velocities as shown in Eq. (3.29). Multiplying both sides of Eq. (3.29) yields:

$$dX = J\left(q\right)dq \tag{3.41}$$

where dX is the vector of infinitesimal displacements at the end-effector, and dq is the corresponding joint displacements. If τ represents the vector of joint torques, the relationship between the torques at the joints and the joint displacements can be stated as:

$$dq = C_\theta \cdot \tau \tag{3.42}$$

where C_θ is the constant joint compliance matrix of diagonal form as follows:

$$C_\theta = \begin{bmatrix} C_{\theta 1} & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{\theta 2} & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{\theta 3} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{\theta 4} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{\theta 5} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{\theta 6} \end{bmatrix} \tag{3.43}$$

where $C_{\theta i}$, $i = 1, \ldots 6$ are joint compliances.

Similarly, the relationship between the Cartesian external static force F and the end-effector displacement is denoted as

$$dX = C_x(\theta) \cdot F \tag{3.44}$$

where $C_x(\theta)$ is the Cartesian space compliance matrix. Substituting Eq. (3.42) and Eq. (3.44) into Eq. (3.41) yields

$$C_x(\theta) \cdot F = J\left(q\right) \cdot C_\theta \cdot \tau \tag{3.45}$$

During the robot movement, the work in all coordinate frames has to remain the same for all dq. The differential work in Cartesian terms, W_x, and the corresponding work in Joint terms, W_q, can be written as:

$$dW_x = F^T dX \qquad , \qquad dW_q = \tau^T dq \tag{3.46}$$

Equating the work terms in Eq. (3.46) and substituting dX from Eq. (3.41), after transposing both sides, yields

$$\tau = J\left(q\right)^T \cdot F \tag{3.47}$$

Substituting Eq. (3.47) into Eq.(3.45), the Cartesian compliance matrix finally equals

$$C_x(\theta) = J\left(q\right) \cdot C_\theta \cdot J\left(q\right)^T \tag{3.48}$$

Eq. (3.48) is significant for the establishment of the robot compliance model since it is able to convert the joint compliance C_θ into the Cartesian compliance $C_x(\theta)$ without calculating the inverse Jacobian matrix.

As defined in Eq. (3.43), constant joint compliance matrix C_θ is diagonal, thus each element of the Cartesian space compliance matrix $C_x(\theta)$ can be rewritten in a generalized form as:

$$C_x(i,j) = \sum_{n=1}^{N} J_{n,i} \cdot C_{\theta n} \cdot J_{n,j} \tag{3.49}$$

where N is the total number of joints, and $C_{\theta n}$ is the compliance of the nth joint.

Methods for identifying joint stiffness can be divided into two categories: static load tests [54] and dynamic load tests [53]. The static load test is conducted by measuring the static deformation of the end-effector under a constant external load while the robot is in a fixed posture. The joint stiffness parameters are then calculated based on the relationship between the external load and the resultant deformation. Dynamic load experiments are performed by applying dynamic forces to the robot ends to measure and process vibration signals, and then calculating dynamic parameters such as robot inertia and stiffness.

In this thesis, the joint compliance values of the Stäubli RX-90 industrial robot have been adopted from [54] and are provided in Table 3.4. The identification method used in [54] is based on static loading tests.

Table 3.4: Joint Compliances of Stäubli RX-90 Robot [54]

Joint number	**1**	**2**	**3**	**4**	**5**
Compliance $C_{\theta i}$ [rad/Nm]$\cdot 10^{-7}$	9.03	11.17	14.74	27.65	97.58

Hence the constant joint compliance matrix C_θ is given as below:

$$C_\theta = \begin{bmatrix} 9.03 & 0 & 0 & 0 & 0 \\ 0 & 11.17 & 0 & 0 & 0 \\ 0 & 0 & 14.74 & 0 & 0 \\ 0 & 0 & 0 & 27.65 & 0 \\ 0 & 0 & 0 & 0 & 97.58 \end{bmatrix} \cdot 10^{-7} \tag{3.50}$$

As mentioned in the previous section, the Jacobian matrix is pose-dependent and not fixed during the movement of the end-effector. Therefore, the Cartesian space compliance matrix $C_x(\theta)$ expressed in Eq. (3.49) is a function of joint angles and related to the interpolation of the path.

3.5 Summary

This chapter starts with a brief introduction of the general structure of the Stäubli RX-90 industrial robot, as well as the description of the coordinate frames attached to the joints and their corresponding transformation parameters. The kinematics of the robot is analyzed in terms of homogeneous transformation matrices. A forward and inverse kinematic model of Stäubli RX-90 industrial robot was developed following the D-H convention. A simple validation of the derived kinematic model was also provided.

Joint flexibility is the dominant factor that affects the positioning accuracy of machining robots. Therefore, a Cartesian compliance model of the Stäubli RX-90 industrial robot was developed and the parameters of the model were introduced. An analytical study of the Jacobian matrix was performed, and the Jacobian matrix of the Stäubli RX-90 industrial robot was derived. The developed kinematics and compliance models provide the theoretical basis for the deflection prediction and compensation models presented in the next chapter.

4 Prediction of Toolpath Distortion in Robotic Machining

4.1 Introduction

The kinematic and compliance model of the robot was obtained in Chapter 3. The most significant factor contributing to errors in robotic machining is the low stiffness of robot joints. When cutting forces are exerted on the robot, the robot's joints will deform, resulting in the deviation of the end-effector from the desired path (i.e. contour errors). When machining forces and the stiffness model of the robot are known, the resultant contour errors can be predicted. The accuracy of machined parts can be improved considerably if contour errors are compensated ahead of time. This chapter predicts cutting forces during sample 3-axis milling processes and analyzes the deflection caused by the flexibility of the Stäubli RX-90 industrial robot.

In addition to deflections, joint compliance in robots leads to large structural vibrations at the end-effector during high speed movements. Vibrations must be avoided as they can cause inaccuracy and damage to manufactured parts. Input shaping of reference commands can significantly reduce vibrations, but it may distort the toolpath. This chapter predicts and compensates for the distortions caused by the input shaper.

The chapter is organized as follows: Section 4.2 develops a prediction model of cutting forces with given operation parameters. Section 4.3 estimates the tooltip deflections due to the cutting forces. Section 4.4 introduces the technique of input shaping for vibration avoidance. Section 4.5 predicts trajectory distortions caused by input shaping. The works of the chapter are summarized in Section 4.6.

4.2 Modeling of Cutting Forces in Robotic Milling

Milling is one of the most commonly used operations for machining custom parts [55]. There are many factors that affect milling forces, e.g. the material of the workpiece and the geometry of the tool. Consider an endmill with no helix angle. For a differential element at the cutter-workpiece interface, the local tangential, radial, and axial cutting forces can be obtained as [56]:

$$\begin{cases} \Delta F_t = k_{tc} h \cdot \Delta a \\ \Delta F_r = k_{rc} h \cdot \Delta a \\ \Delta F_a = k_{ac} h \cdot \Delta a \end{cases} \tag{4.1}$$

where k_{tc}, k_{rc} and k_{ac} are cutting force coefficients, and Δa is the differential height along the tool axis. h is the local chip thickness and can be obtained as a function of the angle of immersion ϕ and feed rate c:

$$h = c \sin \phi \tag{4.2}$$

Based on the equilibrium diagram shown in Figure 4.1, local forces can be projected onto the tool coordinate system as:

$$\begin{cases} \Delta F_x = -\Delta F_t \cos \phi - \Delta F_r \sin \phi \\ \Delta F_y = \Delta F_t \sin \phi - \Delta F_r \cos \phi \\ \Delta F_z = \Delta F_a \end{cases} \tag{4.3}$$

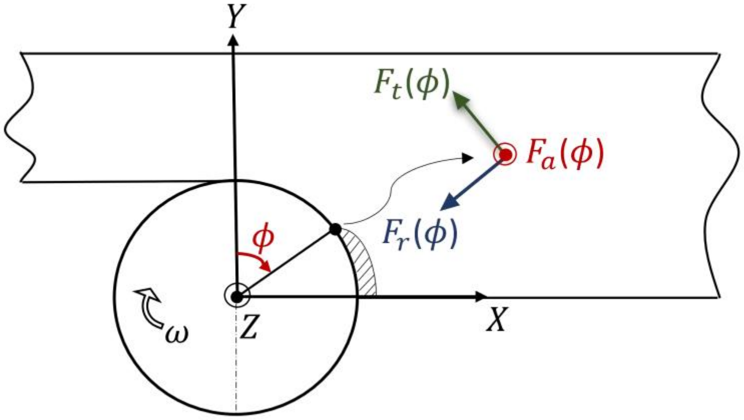


Figure 4.1: Local Cutting Forces in the Milling Process.

Summing up the local cutting forces contributed by all active edges and all elements along the tool axis gives the total forces in the tool coordinate system:

$$F_x = \sum_{j=1}^{N} F_{xj}\left(\phi_j\right), F_y = \sum_{j=1}^{N} F_{yj}\left(\phi_j\right), F_z = \sum_{j=1}^{N} F_{zj}\left(\phi_j\right) \tag{4.4}$$

where j denotes tooth number and N is the total number of flutes on the endmill. Finally, the instantaneous resultant cutting force on the cutter, F_c, is obtained as:

$$F_c = \sqrt{F_x^2 + F_y^2 + F_z^2} \tag{4.5}$$

Figure 4.2 shows a sample simulation of half-immersion down milling operation with the following parameters:

Cutting Conditions: $a = 10\,mm,\ c = 0.1\frac{mm}{rev}/tooth\,, n = 4000\,rpm.$

Tool geometry: $D = 10\,mm, N = 4, \beta = 30°.$

Cutting constants: $k_{tc} = 1800\,N/mm^2, k_{rc} = 540\,N/mm^2, k_{ac} = 800\,N/mm^2.$

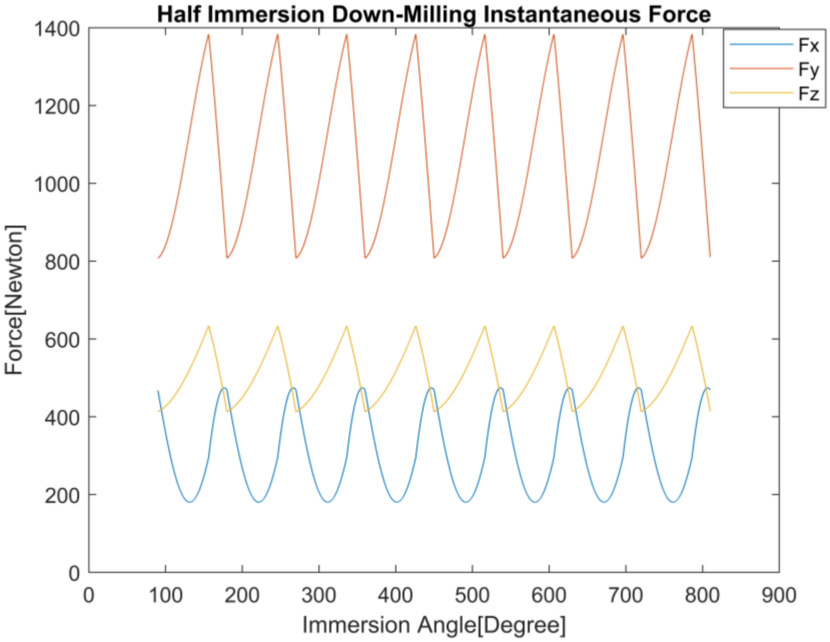


Figure 4.2: Sample Simulation of Instantaneous Cutting Forces.

For predicting robot deflections during machining, it is sufficient to consider the average of cutting forces. The mean values of cutting forces for the simulated case are denoted as F_{xm}, F_{ym} and F_{zm}, and are shown in Figure 4.3.

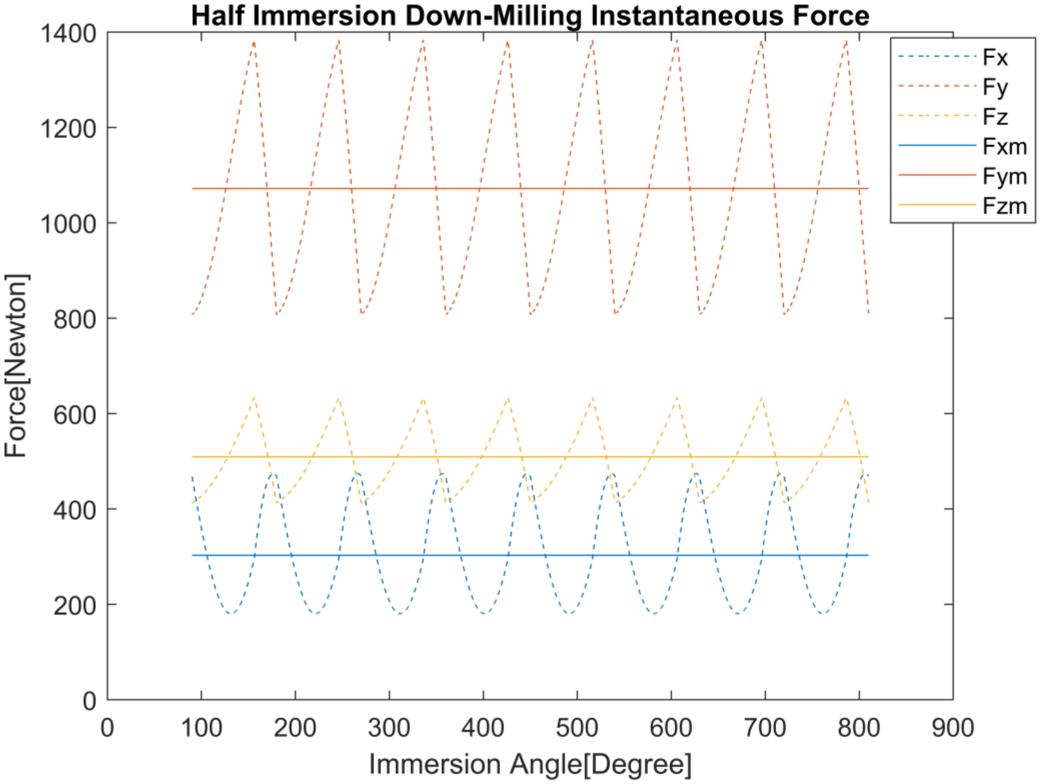


Figure 4.3: Mean Cutting Forces.

The quasi-static external force vector F in Eq. (4.6) is used in the next section for the prediction of tooltip displacements induced by cutting forces:

$$F = \begin{bmatrix} F_{xm} \\ F_{ym} \\ F_{zm} \end{bmatrix} \tag{4.6}$$

For the simulated case, the force vector F is

$$F = \begin{bmatrix} 302.9372 \\ 1071.9 \\ 509.3078 \end{bmatrix} [N] \tag{4.7}$$

Measurement of the cutting force allows for the estimation of the tooltip deflections in the machining process. However, it is worth noting that the elements in vector F in Eq. (4.6) are represented in the tool frame; further coordinate system transformations are required before using the force vector in the deformation prediction model. This is studied in the next section.

4.3 Prediction of Tooltip Displacements Due to Cutting Forces

The relationship between the Cartesian force vector F and the deflection of the end-effector δX is given by

$$\delta X\left(t\right) = C_x\left(\theta\right) \cdot \mathrm{F}\left(t\right) \tag{4.8}$$

where $C_x(\theta)$ is the Cartesian compliance matrix and derived in Eq. (3.48) as:

$$C_x(\theta) = J\left(q\right) \cdot C_\theta \cdot J\left(q\right)^T \tag{4.9}$$

Here, C_θ is the diagonal matrix of joint compliances. Hence the tooltip displacement can be presented in a more intuitive form by combining the above equations as:

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} \left(t\right) = J\left(q\right) \cdot C_\theta \cdot J\left(q\right)^T \cdot \begin{bmatrix} F_{xm} \\ F_{ym} \\ F_{zm} \end{bmatrix} \tag{4.10}$$

Since the Jacobian matrix is dependent on the robot pose, the deflection of the tooltip is different across the workspace. During a milling operation, the direction of machining forces also varies as a function of the machining path. Therefore, the machining model developed in the previous section must be combined with the compliance model presented in Chapter 3 to determine the instantaneous deflections of the robot.

Figure 4.4 illustrates the composition of tooltip displacement δX . δx , δy and δz are the projection of δX in the robot's coordinate system.

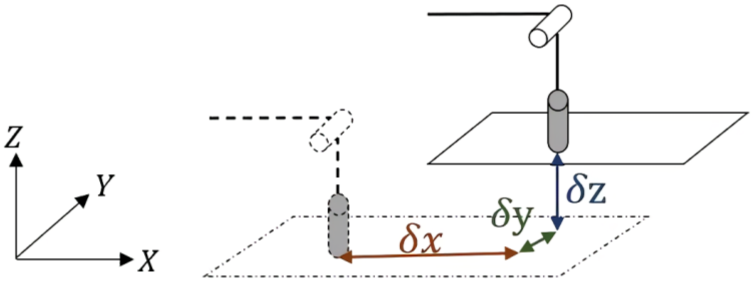


Figure 4.4: Illustration of Deflection Errors at the Tooltip.

As shown in Figure 4.5, the cutting force vector F obtained in Eq. (4.6) is expressed in the tool frame $\{T\}$. The origin of the moving workpiece frame $\{W\}$ is at the end of tool tip. Robot reference frame $\{M\}$ coincides with the robot base frame $\{X_0\ Y_0\ Z_0\}$. Coordinate systems $\{W\}$ and $\{M\}$ are assumed to be parallel to each other, so the cutting force expression in the $\{X_W\ Y_W\ Z_W\}$ directions of frame $\{W\}$ is the same as in $\{X_M\ Y_M\ Z_M\}$ directions of frame $\{M\}$. Green solid circle in Figure 4.5 is the desired circle and the dash-line is the actual trajectory after taking deviation into account.

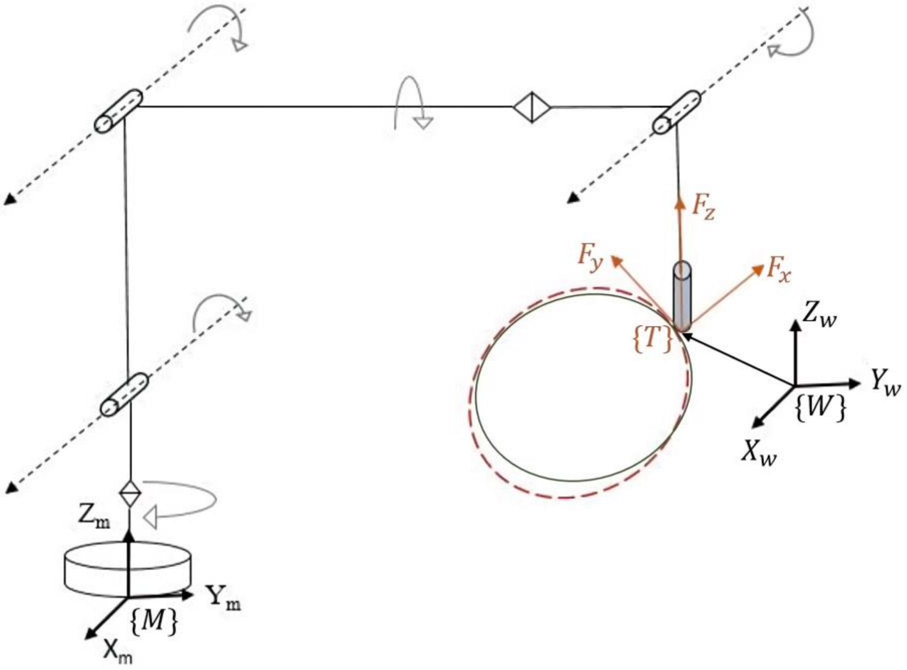


Figure 4.5: Tool Frame {T}, Moving Workpiece Frame {W}, and Robot Reference Frame {M}.

The cutting force vector F in the frame $\{T\}$ with respect to the frame $\{M\}$ is given by [57]

$$F\{M\} = F\{T\} \cdot Rot_{Z_m}(\alpha_i) \cdot Rot_{Z_m}(\gamma) \tag{4.11}$$

where angle $\alpha_i = \frac{\pi}{2} - \phi_i$ as defined in Figure 4.1, is the complementary angle of the immersion angle ϕ of the tooth *i*, and γ is the angle between $\{X_W\}$ direction and cutting tool feed direction.

The sample milling forces simulated in the previous section are assumed to act on the tool during milling of a circular path with a diameter of 300 mm and a feed rate of 100 mm/s. The cutting forces are transformed into the robot reference frame by Eq. (4.11) as shown in Figure 4.6, where the projected forces onto the frame $\{M\}$ are denoted as $F_x\{M\}, F_y\{M\}$

and $F_z\{M\}$. Note that cutting force vectors in robot frame are time-varying even when machining forces are constant.

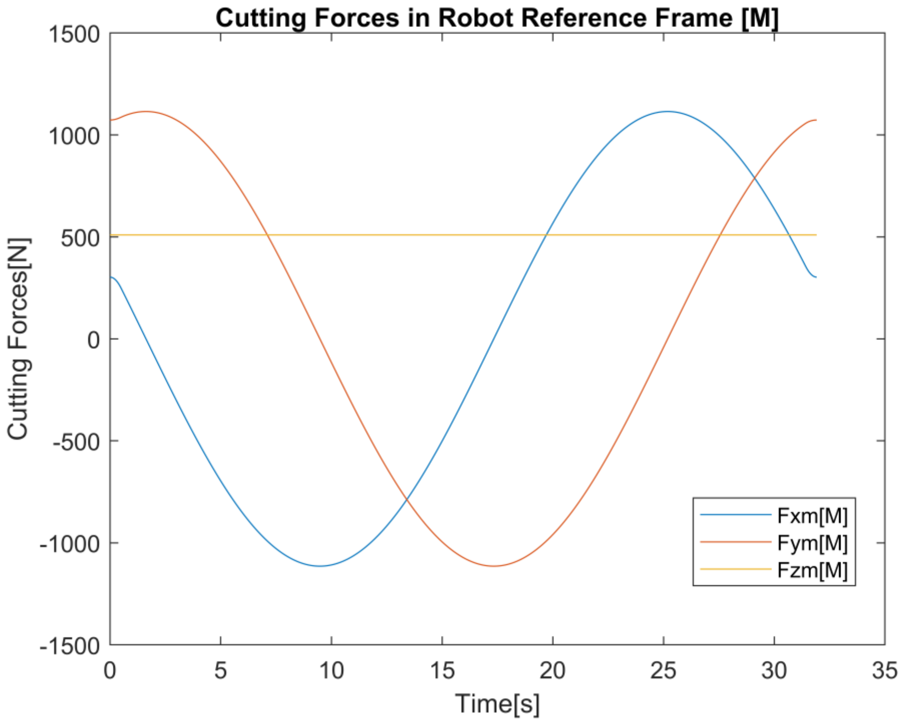


Figure 4.6: Cutting Force Vector in Robot Reference Frame.

The following procedure has been developed to calculate tooltip deflections under machining forces (Figure 4.7):

Step 1: Extract the desired end-effector coordinate from the programmed trajectory. The coordinate vectors are expressed in the robot reference frame $\{M\}$.

Step 2: Calculate the vector of joint angles, θ, from the inverse kinematic model developed in the previous chapter. If there are multiple solutions, choose the one closest to the joint angles at the previous pose.

Step 3: For the calculated joint commands θ at each point, obtain the robot's Jacobian matrix J_θ.

Step 4: Calculate the Cartesian space compliance matrix using Eq. (3.48) .

Step 5: Calculate the external cutting force vector F in the tool frame $\{T\}$ using the machining model.

Step 6: Find the cutting force vector in the robot reference frame $\{M\}$.

Step 7: Determine the linear deflection along the axes of the industrial robot's tooltip $\delta X = [\delta_x \quad \delta_y \quad \delta_z]^T$ induced by cutting force F.

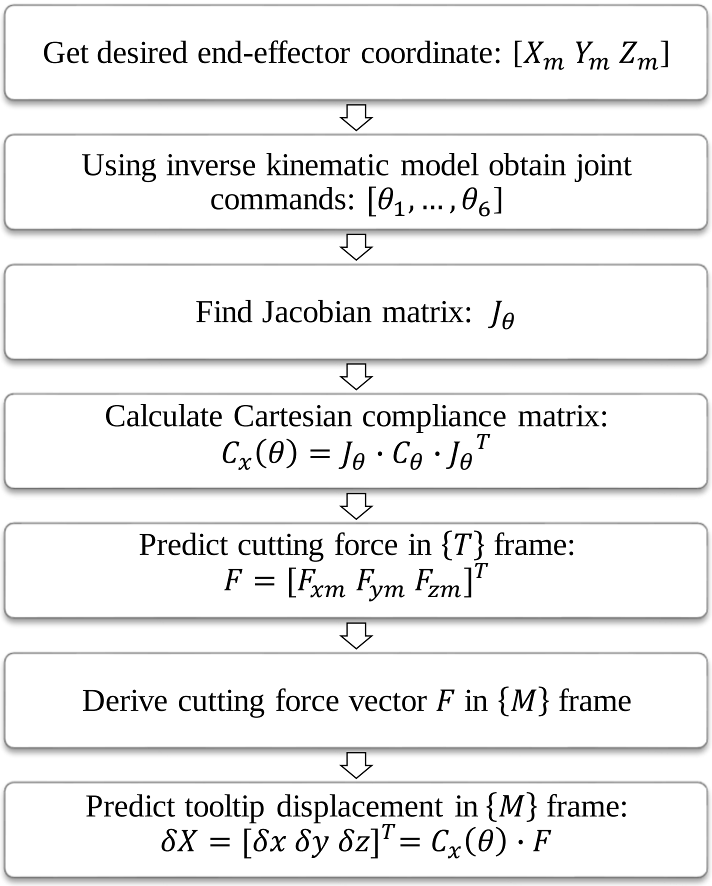


Figure 4.7: Procedure for Prediction of Tooltip Displacements due to Machining Forces.

Figure 4.8 shows the simulated deflections at the tooltip for the case study presented in Figure 4.6. In this simulation, the milling tool travels along a circular toolpath as shown in Figure 4.5.

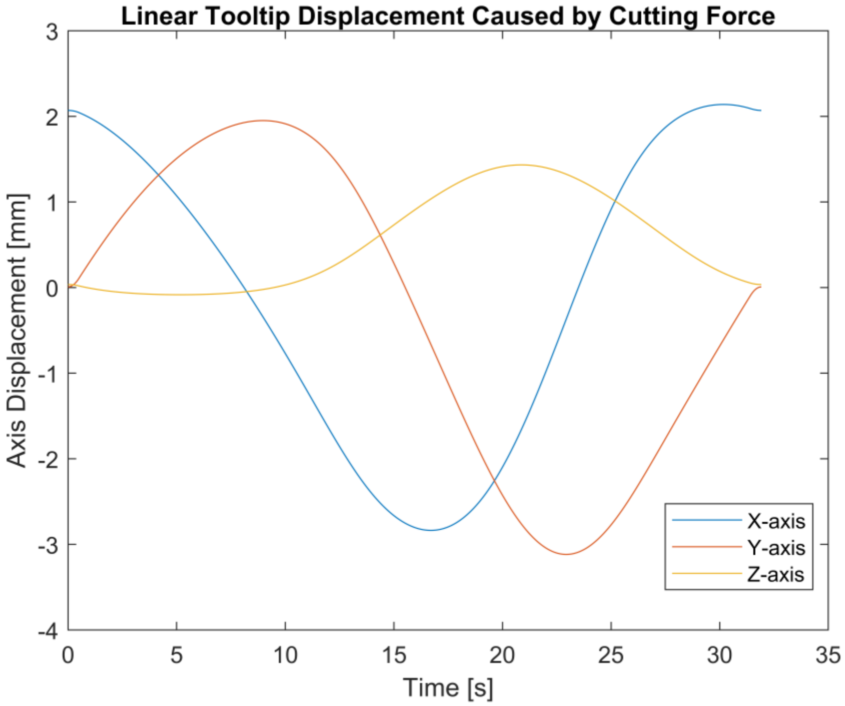


Figure 4.8: Tooltip Displacements Due to Cutting Forces.

During the simulated operation, the mean displacements are 1.6 mm, 1.5 mm, and 0.6 mm in X, Y and Z direction, respectively. Max displacements are 2.8 mm along X axis, 3.1 mm along Y axis, and 1.4 mm along Z axis, respectively.

4.4 Input Shaping for Vibration Avoidance

Due to their long kinematic chain, robots experience large vibrations during high speed movements. Input shaping is a command modification technique that alters reference joints commands of the robot to avoid the excitation of structural modes. Consider a flexible system with ω_n as natural frequency and ζ as damping ratio; the second-order transfer function of the system can be written as:

$$G(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2} \tag{4.12}$$

Apply an impulse with the magnitude of A_1 to this system at the time t_1; the system will vibrate at its damped natural frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$. Then apply a second impulse with the magnitude of A_2 to this system at the time t_2. Theoretically, if the time and magnitude of the second impulse are properly chosen, the second impulse can completely eliminate the vibration induced by the first impulse. Such a sequence of zero vibration impulses is called an input shaper and is illustrated in Figure 4.9.

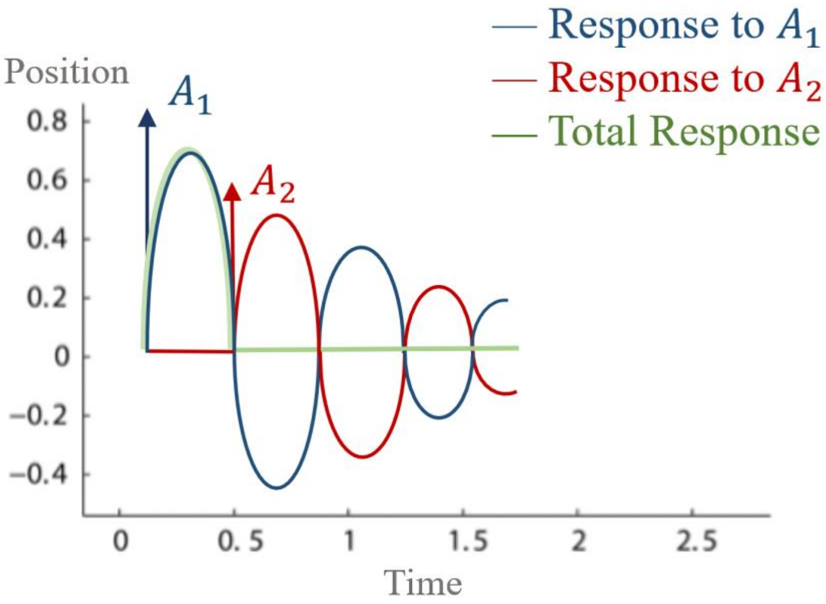


Figure 4.9: Two-Impulse Sequence with Self-canceling Vibrations.

Input shaper design is the process of determining the magnitudes and time locations of the sequence of impulses, which depend on the system parameters: natural frequency and damping ratio. The most commonly used input shaper is the three impulse sequence known as Zero-Vibration and Derivative (ZVD) shaper. The amplitudes and time locations in a ZVD shaper can be obtained as [30]:

$$\left[\begin{Bmatrix} A \\ t \end{Bmatrix}_j\right] = \left[\begin{Bmatrix} \dfrac{1}{K^2+2K+1} \\ 0 \end{Bmatrix}_1, \begin{Bmatrix} \dfrac{2K}{K^2+2K+1} \\ 0.5T_d \end{Bmatrix}_2, \begin{Bmatrix} \dfrac{1}{K^2+2K+1} \\ T_d \end{Bmatrix}_3\right] \tag{4.13}$$

where $K = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$ and $T_d = 2\pi/\omega_n\sqrt{1-\zeta^2}$.

It can be shown that if an input shaper is mathematically convolved with any reference command, the so-called shaped command will not cause any vibrations. For example, assume $\theta(t)$ is the original unshaped command for a joint of the robot, and the designed ZVD input shaper is $InShp(t) = A_1\delta(t) + A_2\delta(t-t_2) + A_3\delta(t-t_3)$. The shaped command $\theta_{shp}(t)$ can be derived by:

$$\begin{aligned} \theta_{shp}(t) &= \theta_{shp,1}(t) + \theta_{shp,2}(t) + \theta_{shp,3}(t) \\ &= A_1\theta(t) + A_2\theta(t-t_2)\mathbf{1}\langle t-t_2\rangle + A_3\theta(t-t_3)\mathbf{1}\langle t-t_3\rangle \end{aligned} \tag{4.14}$$

where $\mathbf{1}\langle t-t_n\rangle = 1$ at $t \geq t_n$ and $\mathbf{1}\langle t-t_n\rangle = 0$ at $t < t_n$ for $n = 2,3$.

For example, the Stäubli RX-90 industrial robot has a dominant natural frequency at about 18 Hz with estimated damping of 0.1 [31]. Given the system parameters: $\omega_n = 18$ and $\zeta = 0.1$, a ZVD input shaper is designed with magnitude and time locations given by Eq. (4.13):

$$\left[\begin{Bmatrix} A \\ t \end{Bmatrix}_j\right] = \left[\begin{Bmatrix} 0.3344 \\ 0 \end{Bmatrix}_1, \begin{Bmatrix} 0.4877 \\ 0.1754 \end{Bmatrix}_2, \begin{Bmatrix} 0.1778 \\ 0.3508 \end{Bmatrix}_3\right] \tag{4.15}$$

The designed shaper is mathematically convolved with a step joint command at joint 1, i.e. $\theta_1(t) = 1$ rad. The original and shaped commands are shown in Figure 4.10. Note that the shaped command has 0.35 s time delay with respect to the original step command. Such delays can distort the trajectory in multi-axis toolpaths. It should be mentioned that this thesis focuses only on the modeling and compensation of distortions caused by input shaping. Detailed analysis of robot vibrations and the effect of input shaping in vibration suppression have been studied by Newman et al. in [31].

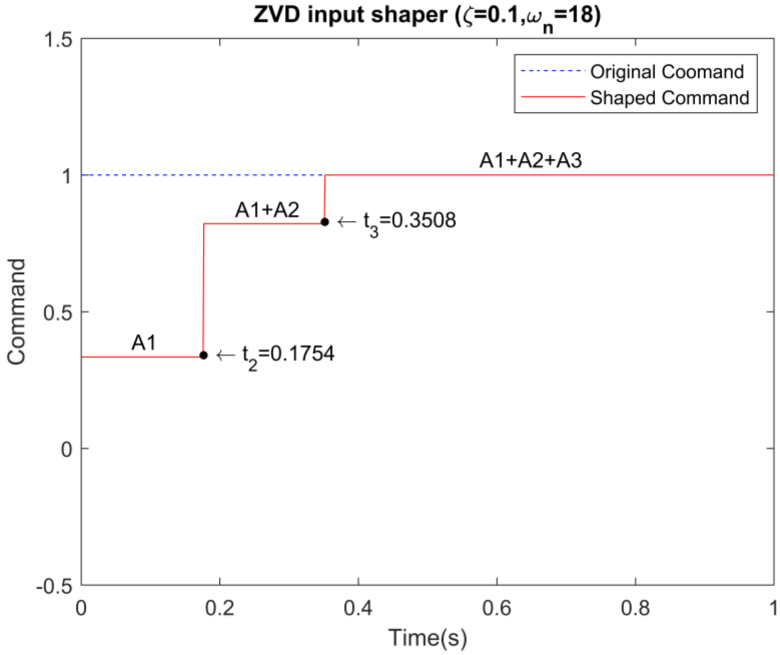


Figure 4.10: Original and ZVD-shaped Reference Commands for : $\boldsymbol{\omega_n = 18}$ and $\boldsymbol{\zeta = 0.1}$.

4.5 Trajectory Distortion Caused by Input Shaping

As can be seen in Eq. (4.14), the ZVD input shaper shifts the reference commands in time, causing a delay equal to the length of the shaper, i.e. t_3. In robotic machining tasks involving several joints, the added time delay will lead to considerable path distortions even if all joints are shaped with the same shaper.

For example, consider the circular toolpath generated in Section 5.2. Assume that Stäubli RX-90 industrial robot is to contour this path while the end-effector (tool) is always in the vertical (Z) direction. Figure 4.11 shows the unshaped and ZVD-shaped joint commands for contouring this circular path. It can be seen that the shaped commands have a small delay with respect to the original commands. Small differences in the commands of each joint will eventually accumulate to a large shift in the end-effector position.

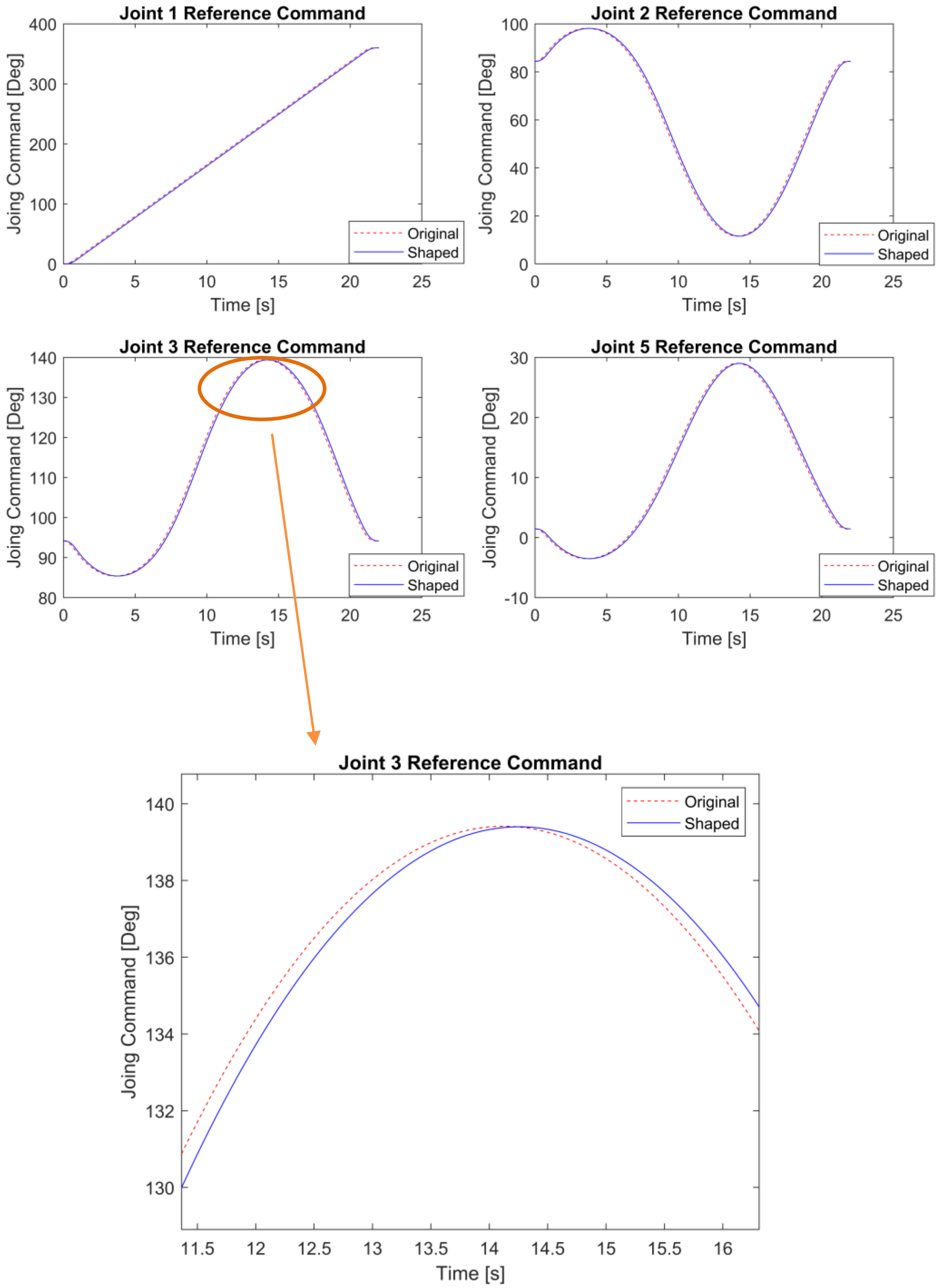


Figure 4.11: Original and Shaped Reference Commands.

In order to determine the geometry of the shaped toolpath, the shaped joint commands are given as inputs to the forward kinematic model developed in Chapter 3. The original and shaped toolpaths are shown in Figure 4.12. It can be seen that there are some deviations between the original (desired) and shaped toolpath. These deviations are called contour errors. Modeling and compensation of contour errors are presented in Chapter 5.

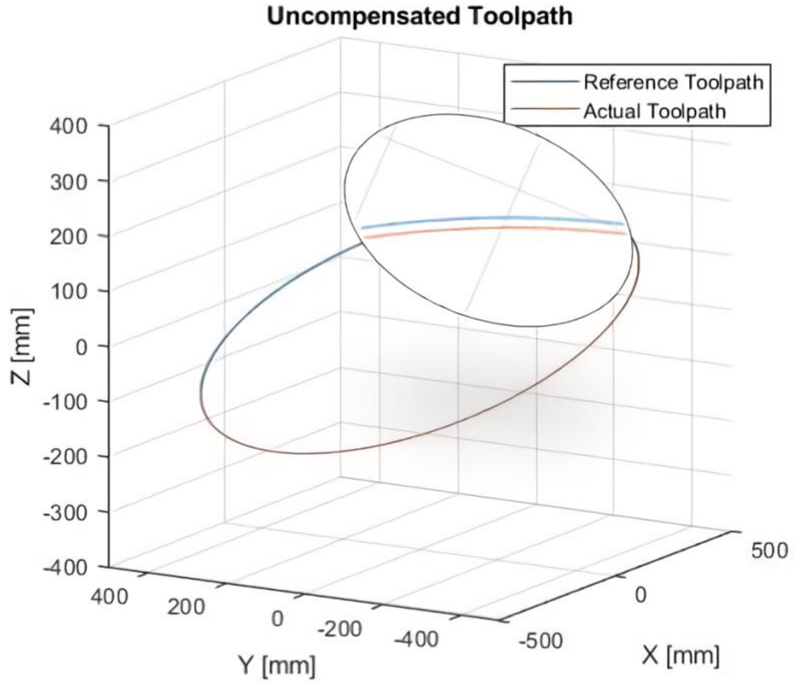


Figure 4.12: Trajectory Distortion Caused by Input Shaping.

4.6 Summary

There are many sources of positional errors in robotic machining operations. This chapter focuses on the positioning errors that arise from cutting forces and input shapers. Cutting forces are always present in robotic machining processes and cause deflections of the arm. Input shaping is a widely used method to reduce vibrations, but it can cause delays and path distortions. Prediction and compensation of contour errors is the most important step for improvements of accuracy in robotic machining. This task is investigated in the next chapter.

5 Compensation of Positioning Errors from Different Sources

5.1 Introduction

The purpose of positioning error compensation is to correct for contour errors to improve the machining accuracy of a robot. Many factors can contribute directly to positioning errors in robotic machining, such as deflections, input shaping delays, environmental effects, etc. This chapter proposes techniques for the compensation of contour errors caused by cutting force induced deflections and input shaping. The compensation model is designed to correct for the predicted contour errors and to improve robot accuracy by adjusting reference commands prior to machining.

The chapter is organized as follows: Section 5.2 presents a numerical method to model contour errors caused by different sources. Section 5.3 introduces a compensation model to correct for contour errors induced by cutting forces and input shaping. Section 5.4 proposes a unified framework for combining and compensating both errors in a single step. A brief conclusion of this chapter is summarized in Section 5.5.

5.2 Estimation of Contour Errors

To present the contour error prediction and compensation model, a sample robotic milling task is studied in this section. As shown in Figure 5.1, the trajectory of machining is a three-dimensional circle with a radius of 300 mm and with its center located at the origin. The normal vector of the plane of the path is [0.2 0.3 0.5].

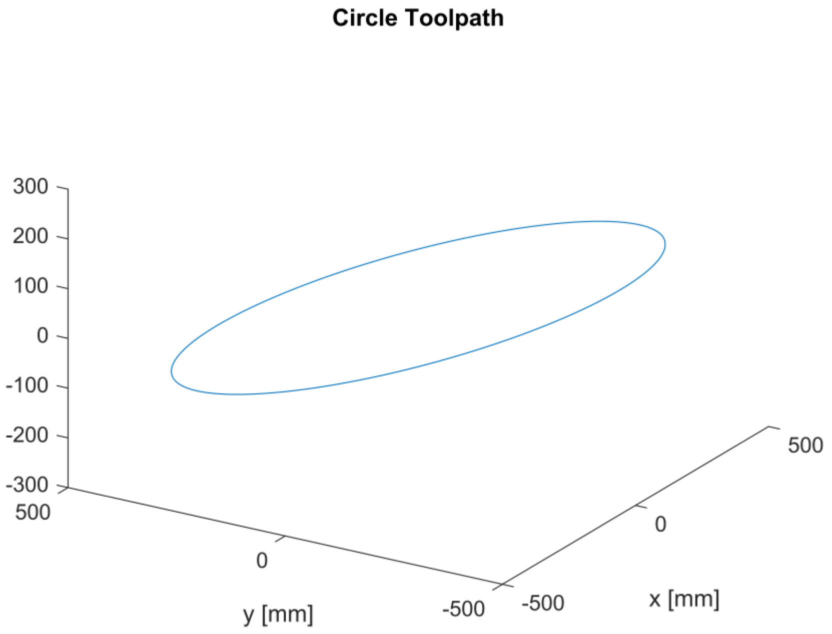


Figure 5.1: Spatial Circular Toolpath Used for Contour Error Analysis.

To generate the machining trajectory, a systematic model is developed as shown in Figure 5.2. This model is able to generate reference commands for each joint of the robot for the smooth contouring of linear and circular segments. First, the path is interpolated at 1 ms time intervals using a trapezoidal velocity profile with the maximum velocity and acceleration of 100 mm/s and 200 mm/s2 respectively. This step considers the motion along the path regardless of its shape, and outputs the desired location of the tool along the path at each interpolation time. Figure 5.3 shows the time-stamped location of the tool in the Cartesian space for the circular toolpath shown in Figure 5.1.

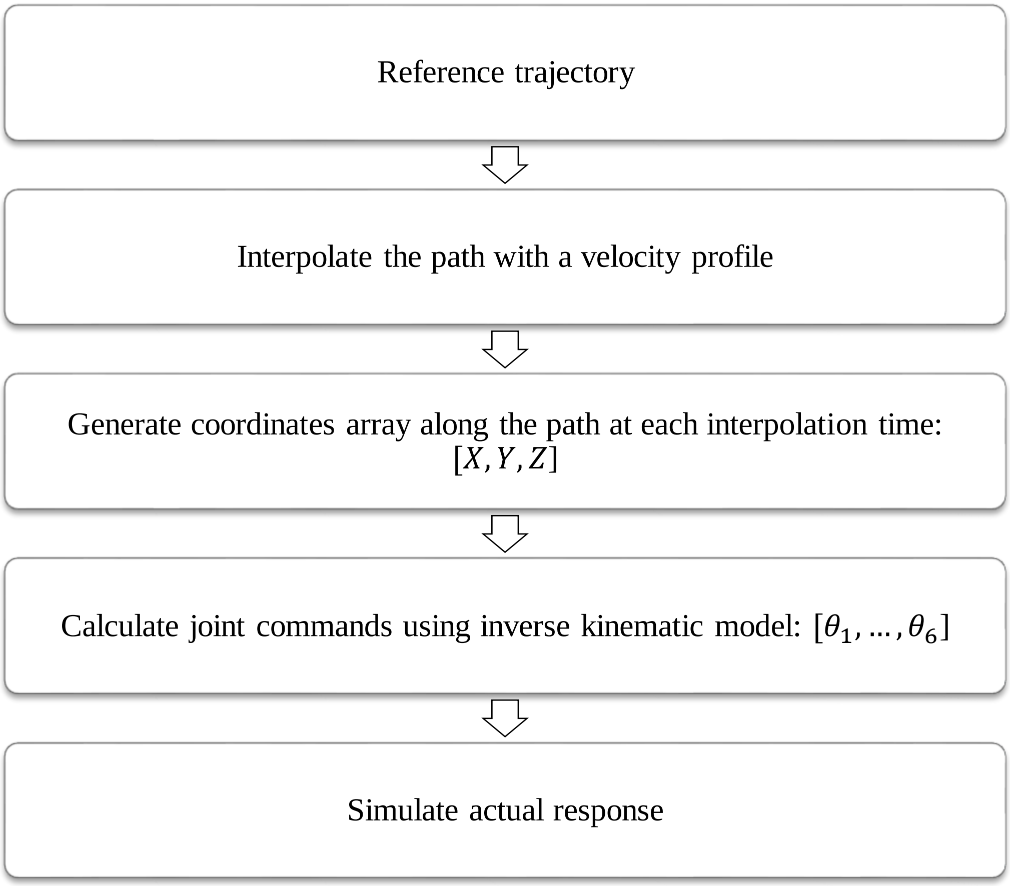


Figure 5.2: Algorithm for Trajectory Generation and Path Simulation.

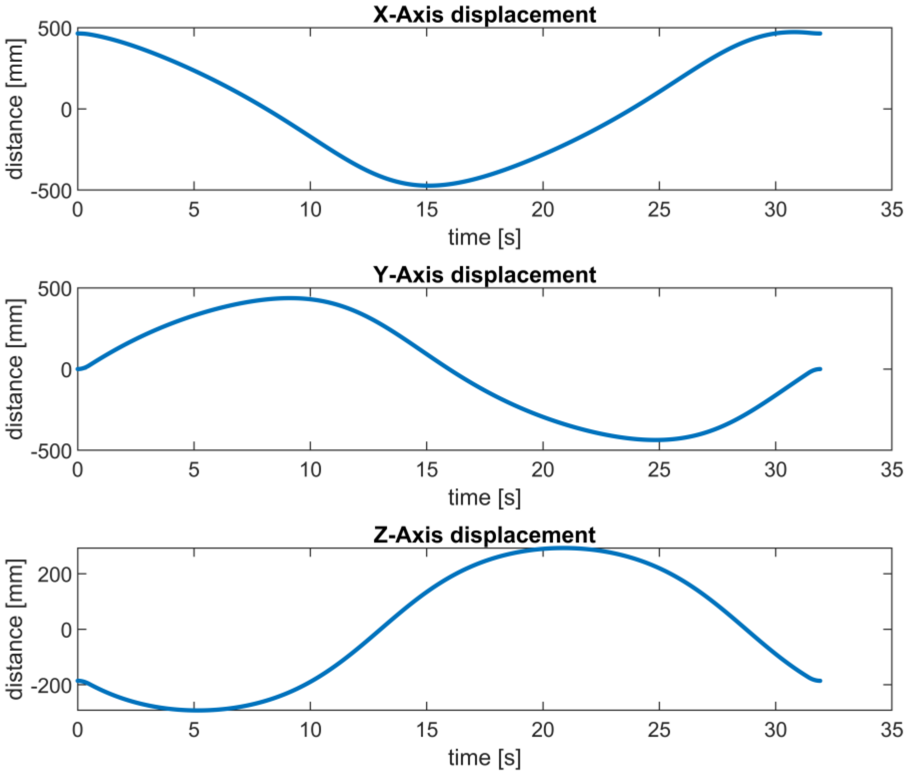


Figure 5.3: Time-Stamped Cartesian Coordinates of the Tool Along the Circular Path.

Figure 5.3 provides the Cartesian coordinates of the end-effector at each interpolation time. The corresponding joint commands (θ_i) obtained from the inverse kinematic model (Chapter 3) are shown in Figure 5.4.

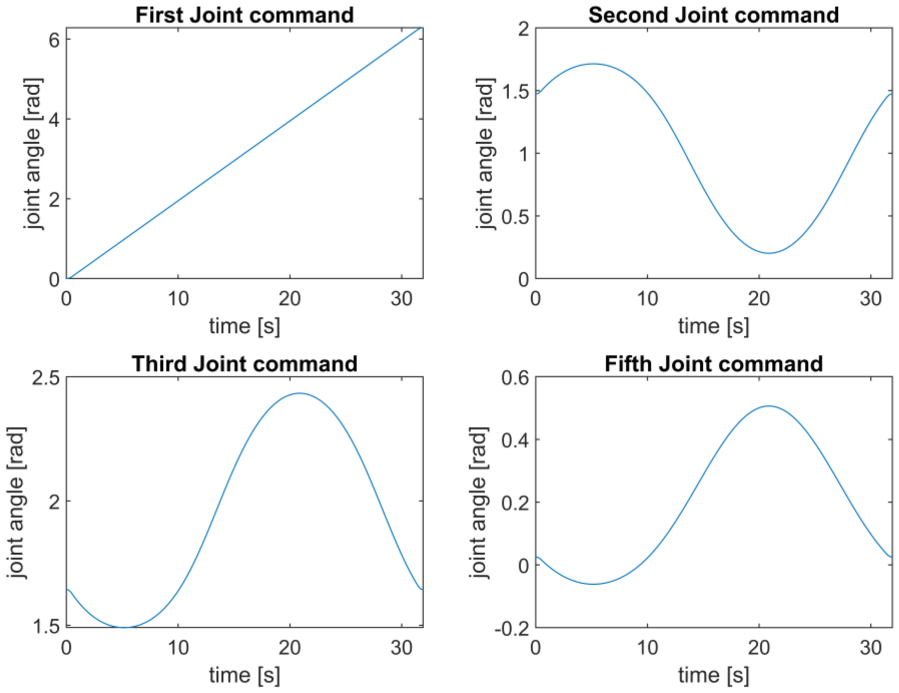


Figure 5.4: Joint Commands of Joint 1,2,3 and 5.

After generating the reference trajectory, the next step is to model the deviations of the robot from the desired path, also known as contour errors. At each point along the machining path, contour error is defined as the shortest distance between the reference and the actual trajectory. In Figure 5.5, the desired and actual contours are represented by solid and dash lines, respectively. At the time t, the desired location of the tooltip is $P_r = [x_r \quad y_r \quad z_r]$, while the actual position is $P_a = [x_a \quad y_a \quad z_a]$. The tracking error is defined as follows:

$$e = P_r - P_a = [x_r - x_a \quad y_r - y_a \quad z_r - z_a] \tag{5.1}$$

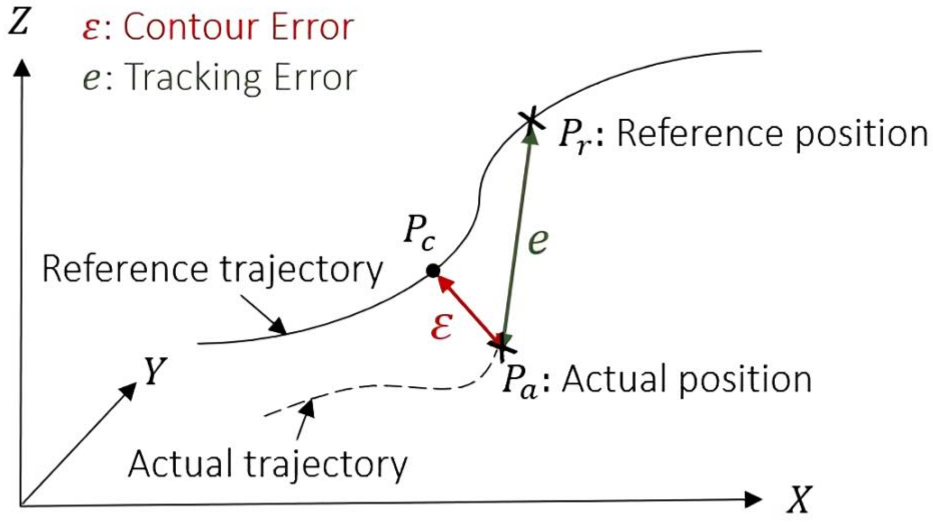


Figure 5.5: Definition of Tracking Error and Contour Error.

Let $P_c = [x_c \quad y_c \quad z_c]$ be the point on the reference trajectory closest to the actual position P_a. The distance between P_c and P_a is contour error ε, which is a spatial vector defined as:

$$\varepsilon = P_c - P_a = [x_c - x_a \quad y_c - y_a \quad z_c - z_a] \tag{5.2}$$

Tracking error is a time-dependent variable whereas contour error is a purely geometrical parameter representing the accuracy and tolerance of the machined part.

In Chapter 4, tooltip deflections due to cutting forces were calculated (Figure 4.8). Figure 5.6 shows the corresponding three-dimensional visualization of the deflection errors in the robot reference frame {M}.

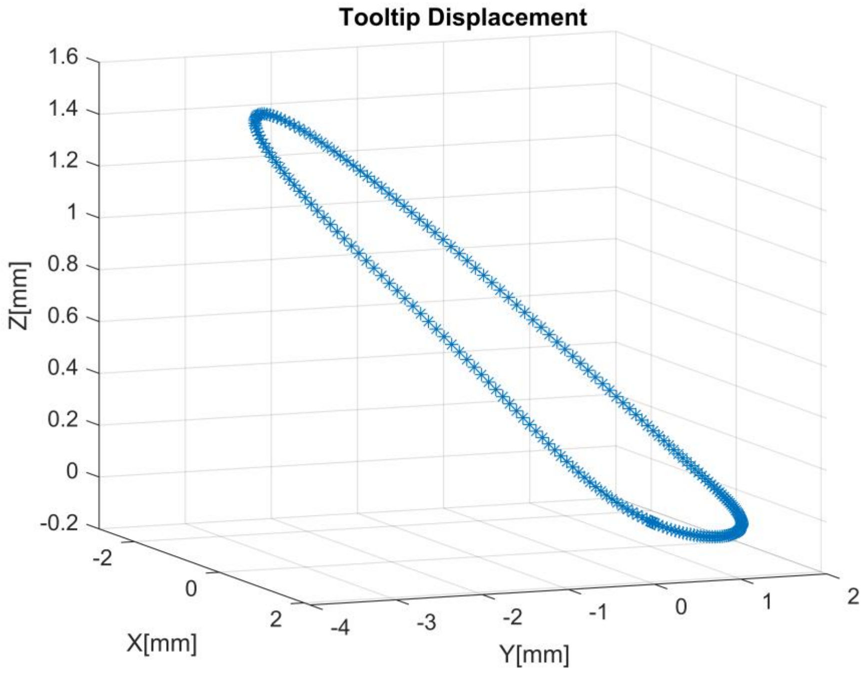


Figure 5.6: Three-Dimensional Plot of Tooltip Deviations Due to Cutting Forces.

From the predicted tooltip deviations, it is possible to determine the actual tool coordinates (P_{xa}, P_{ya}, P_{za}):

$$\begin{cases} P_{xa} = P_{xd} - \delta x \\ P_{ya} = P_{yd} - \delta y \\ P_{za} = P_{zd} - \delta z \end{cases} \tag{5.3}$$

where (P_{xd}, P_{yd}, P_{zd}) are the coordinates of the desired path at any given moment *t*, and δx, δy and δz are the tooltip displacements obtained from the prediction model (Figure 4.7). Figure 5.7 compares the desired and actual toolpath for the sample circular milling operation.

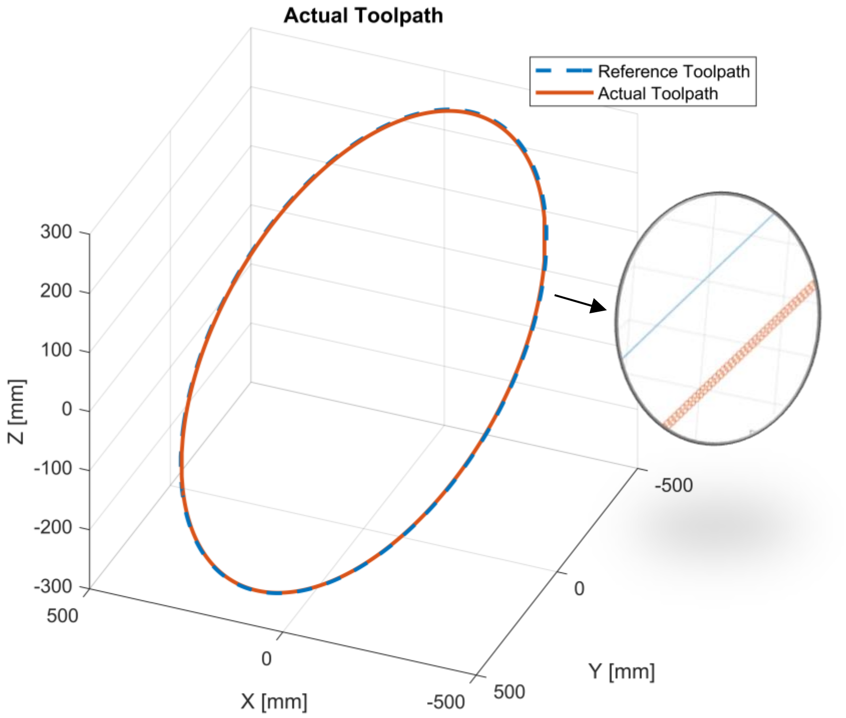


Figure 5.7: Reference and Actual Trajectories Subject to Force-Induced Deflections

The zoomed view in Figure 5.7 clearly shows the distance between the actual path and the desired path. As explained previously, cutting forces generated by the contact between the workpiece and the tool during the robotic machining process contribute to this discrepancy.

Using the definition of contour error presented in Figure 5.5 and Eq. (5.2), contour errors for the path shown in Figure 5.7 have been obtained. Figure 5.8 presents the instantaneous contour error due to cutting force-induced deflections.

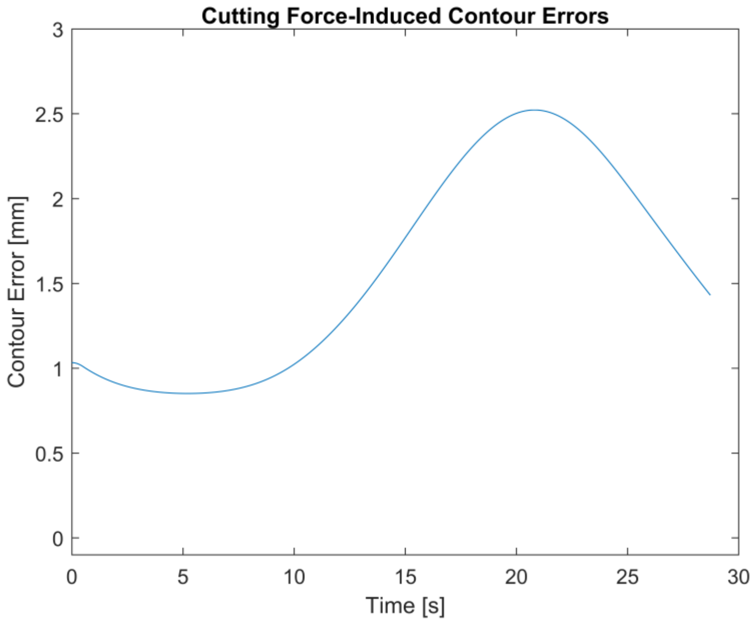


Figure 5.8: Cutting Force-Induced Contour Errors.

The mean value of the cutting force-induced contour errors is 1.6 mm, and the maximum value is 2.5 mm. Simulation results show that, as stated in [57], cutting forces can result in contour errors greater than 1mm, which is one of the major sources of positioning errors in robotic machining. Such large errors may violate the required tolerance for machined parts and thus must be prevented. This is studied in the next section (5.3).

In addition to cutting forces, path distortions caused by input shaping can also contribute to positioning errors as investigated in Section 4.5. Figure 5.9 shows the calculated contour errors caused by the implementation of the ZVD input shaper defined in Eq. (4.15). Contour errors have been calculated based on the original and shaped trajectories simulated in Figure 4.12.

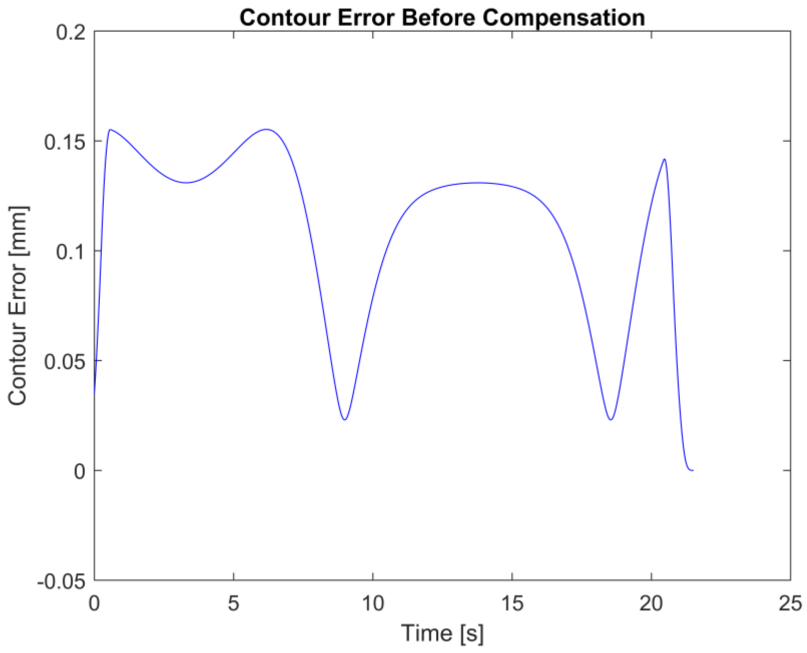


Figure 5.9: Contour Error Caused by Input Shaping.

The mean value of the input shaping-induced contour errors is 0.1 mm, and the maximum value is 0.16 mm. Compared with cutting forced-induced errors (>1mm), this distortion is much smaller. Nevertheless, achieving the required part tolerance in aerospace machining (<50 micrometers), requires compensation of both types of error.

The predicted positioning errors induced by cutting forces and input shapers can be avoided to a great extent. A systematic framework for the compensation of contour errors in robotic machining is presented in the following section.

5.3 Compensation of Positioning Errors

As shown in Figure 5.10, there is an acceptable tolerance zone around the desired machining path (solid blue line). However, the actual trajectory (the red circle marker line) may land outside this tolerance band due to positioning errors caused by cutting forces and input shaping.

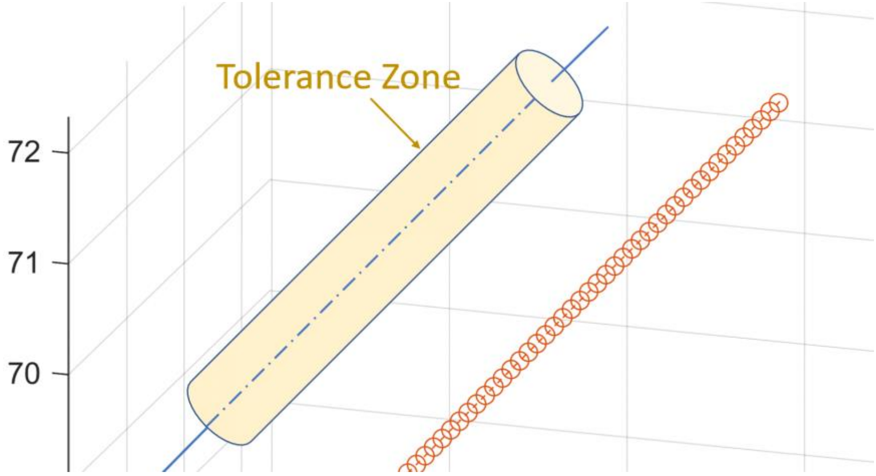


Figure 5.10: Acceptable Tolerance Zone Around a Machining Toolpath**.**

In order to bring the actual trajectory inside the tolerance zone, the deviation is estimated and compensated by adjusting the desired toolpath. Figure 5.11 illustrates the general idea of the developed compensation method. Assume that the black dash-dot circle is the desired toolpath. Due to structural deflections and input shaping distortions, the actual response (red dash-line) will deviate from the desired path. However, as illustrated by the green solid line, the reference path can be modified according to the predicted deviation such that the actual response of the system follows the desired path (i.e. exact circle). In this case, the actual response will be closer to the desired circle even though the compensated reference path is not a perfect circle.

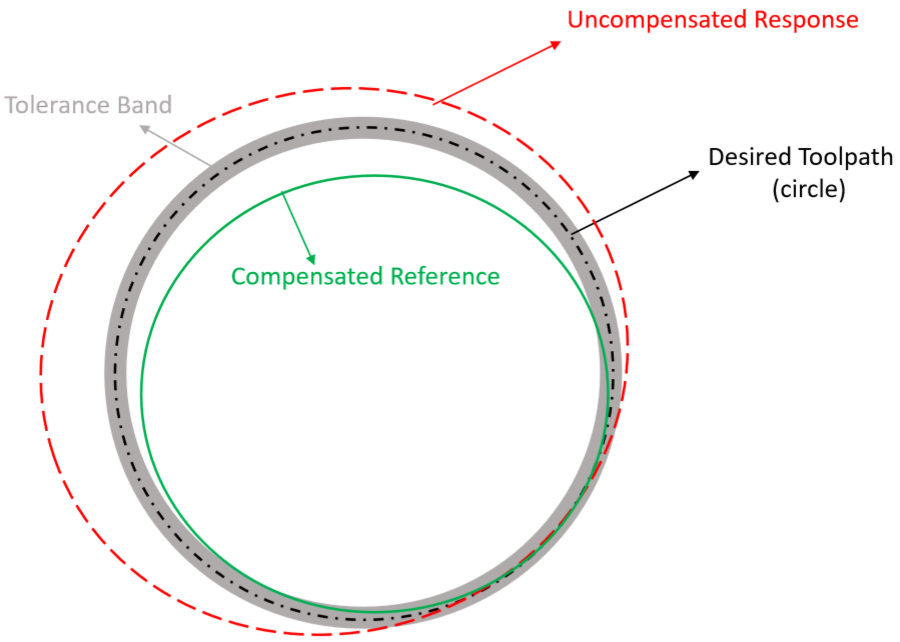


Figure 5.11: Illustration of Contour Error Compensation.

The flowchart of the developed contour error compensation framework is presented in Figure 5.12. Given a desired toolpath for robotic machining, the path is first interpolated at 1 ms intervals with a trapezoidal velocity profile (Figure 5.3). The inverse kinematic model of the robot is then used to determine the corresponding joint commands that put the tool at the desire location at each time step (Figure 5.4). The error prediction model introduced in Section 4.3 is used to determine the tooltip deviations (Figure 4.7), and the actual tool trajectory is determined accordingly (Figure 5.7). The resultant contour errors between the desired and actual trajectory are calculated, as shown in Figure 5.8 and Figure 5.9.

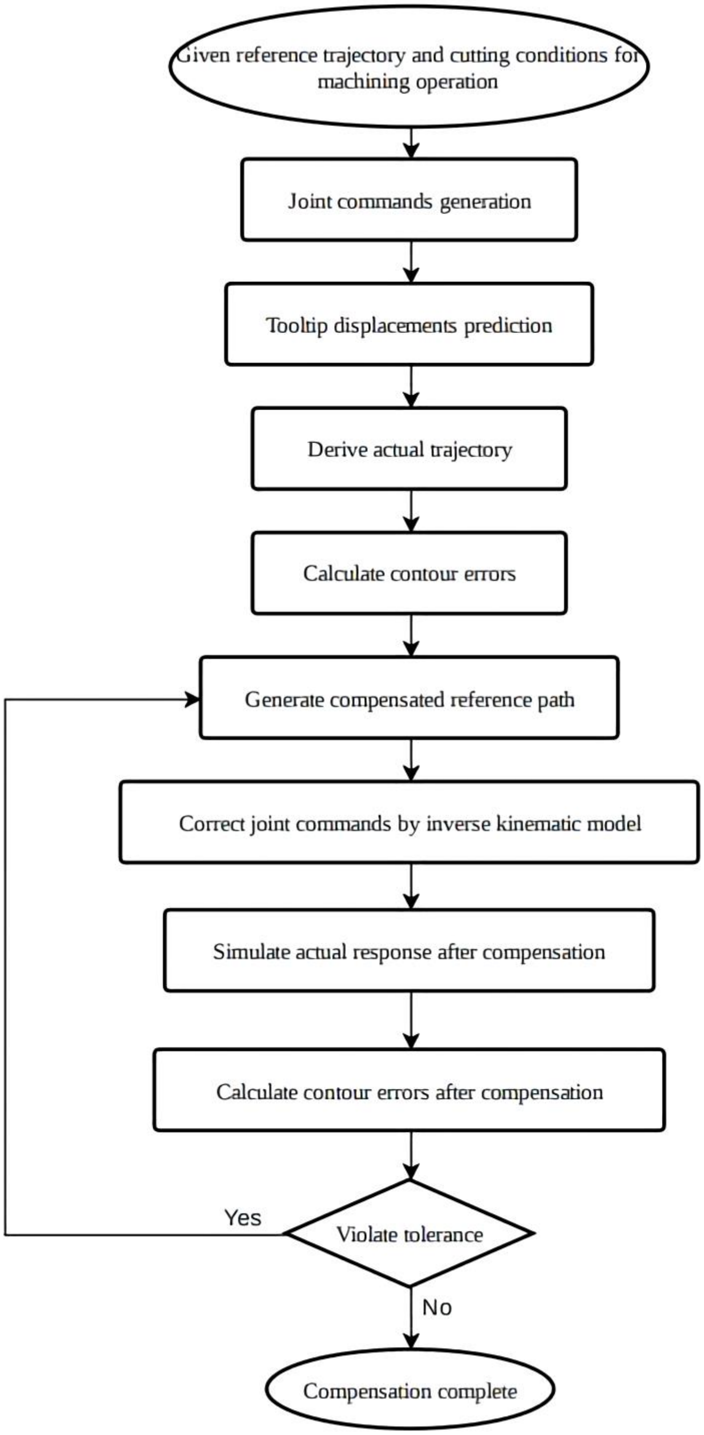


Figure 5.12: Procedure for Contour Error Compensation.

The compensated reference path can be obtained as follows. Assume $\begin{bmatrix} X_M & Y_M & Z_M \end{bmatrix}^T$ is the interpolated coordinates of the desired trajectory (black dash-dot circle in Figure 5.11), also called the original reference path. The path can be compensated by correcting for contour errors, i.e.

$$\begin{bmatrix} X_{compensated} \\ Y_{compensated} \\ Z_{compensated} \end{bmatrix} = \begin{bmatrix} X_M \\ Y_M \\ Z_M \end{bmatrix} + \begin{bmatrix} CE_x \\ CE_y \\ CE_z \end{bmatrix} \tag{5.4}$$

where $\begin{bmatrix} X_{compensated} & Y_{compensated} & Z_{compensated} \end{bmatrix}^T$ are the coordinates of the compensated path (green solid line in Figure 5.11) in the robot reference frame {M}, and $\begin{bmatrix} CE_x & CE_y & CE_z \end{bmatrix}^T$ are the contour errors along the robot axes (Figure 5.8 and Figure 5.9).

After obtaining the compensated path, the corresponding joint commands are calculated using the inverse kinematic model developed in Chapter 3. The actual response of the robot to the corrected joint commands is then simulated using the forward kinematic model. The new contour errors are calculated; if the after-compensation contour error meets the final part tolerance requirements, the compensation procedure is completed; otherwise, a new compensated reference path based on the new contour errors is generated. It's worth noting that in the proposed compensation technique, all of the calculations are carried out offline before sending commands to the robot controller. Processing contour errors in advance makes the real-time operation more efficient by avoiding extra computational burden on the controller.

5.3.1 Compensation of Cutting Force-Induced Errors

To verify the effectiveness of the presented error compensation method, a set of 8 observation points have been selected randomly along the simulated milling toolpath (Figure 5.7). Table 5.1 provides the joint angles and the tooltip positions (in the robot reference frame {M}) for each of the observation points.

Table 5.1: Desired Joint Angles and End-effector Position for Observation Points

Point	θ_1[rad]	θ_2[rad]	θ_3[rad]	θ_5[rad]	X[mm]	Y[mm]	Z[mm]
1	1.4698	1.6581	1.5244	-0.0410	42.5149	419.5080	-268.7108
2	1.4798	1.6558	1.5259	-0.0401	38.3769	420.5603	-267.6869
3	1.4898	1.6534	1.5274	-0.0392	34.2230	421.5832	-266.6391
4	1.4998	1.6510	1.5289	-0.0383	30.0534	422.5764	-265.5672
5	1.5098	1.6485	1.5305	-0.0374	25.8680	423.5398	-264.4710
6	1.5198	1.6459	1.5321	-0.0365	21.6668	424.4728	-263.3504
7	1.5298	1.6433	1.5338	-0.0355	17.4501	425.3751	-262.2051
8	1.5398	1.6407	1.5355	-0.0345	13.2178	426.2464	-261.0350

The corresponding contour errors projected along the X, Y, and Z axes of the robot reference frame {M} are shown in Table 5.2. The fourth row in Table 5.2 is the total contour error. It is seen clearly that considerable errors (>1 mm) exist at each observation point along the path.

Table 5.2: End-effector Errors Before Compensation

Point	1	2	3	4
CE-X[mm]	0.4427	0.4283	0.4139	0.3993
CE-Y[mm]	1.8274	1.8323	1.8372	1.8421
CE-Z[mm]	-0.1241	-0.1265	-0.1267	-0.1280
CE[mm]	1.8843	1.8859	1.8875	1.8892

Point	5	6	7	8
CE-X[mm]	0.3846	0.3697	0.3547	0.3396
CE-Y[mm]	1.8469	1.8516	1.8563	1.8610
CE-Z[mm]	-0.1292	-0.1305	-0.1318	-0.1330
CE[mm]	1.8909	1.8927	1.8945	1.8964

The contour error compensation algorithm in Figure 5.12 has been used to correct for the errors. The corrected joint angles and tool coordinates of each observation point are shown in Table 5.3. The resultant contour errors at the observation points after compensation are also provided in Table 5.4.

Table 5.3: Joint Angles and End-effector Position After Compensation

Point	θ_1[rad]	θ_2[rad]	θ_3[rad]	θ_5[rad]	X[mm]	Y[mm]	Z[mm]
1	1.4692	1.6581	1.5214	-0.0381	42.9575	421.3354	-268.8349
2	1.4792	1.6583	1.5229	-0.0372	38.8052	422.3926	-267.8123
3	1.4892	1.6559	1.5244	-0.0364	34.6369	423.4204	-266.7658
4	1.4992	1.6535	1.5259	-0.0355	30.4527	424.4185	-265.6952
5	1.5092	1.6486	1.5275	-0.0345	26.2525	425.3866	-264.6003
6	1.5192	1.6461	1.5291	-0.0336	22.0366	426.3244	-263.4809
7	1.5292	1.6435	1.5307	-0.0326	17.8048	427.2314	-262.3369
8	1.5392	1.6408	1.5324	-0.0317	13.5574	428.1074	-261.1680

Table 5.4: End-effector Errors after Compensation

Point	**CE-X[mm]**	**CE-Y[mm]**	**CE-Z[mm]**	**CE[mm]**
1	0.0033	0.0069	0.0060	0.0097
2	0.0032	0.0069	0.0060	0.0097
3	0.0032	0.0068	0.0060	0.0096
4	0.0031	0.0068	0.0060	0.0096
5	0.0031	0.0068	0.0060	0.0095
6	0.0031	0.0068	0.0060	0.0095
7	0.0030	0.0067	0.0060	0.0095
8	0.0030	0.0067	0.0060	0.0095

Time-stamped contour errors along the toolpath before and after compensation are compared in Figure 5.13. The blue dash-line is the contour error before applying compensation. It can be seen that the original path can cause contour errors larger than 2 mm. After compensation, contour errors have been reduced to below 0.03 mm. This clearly shows the effectiveness of the developed compensation method provided an accurate estimate of the error is known beforehand.

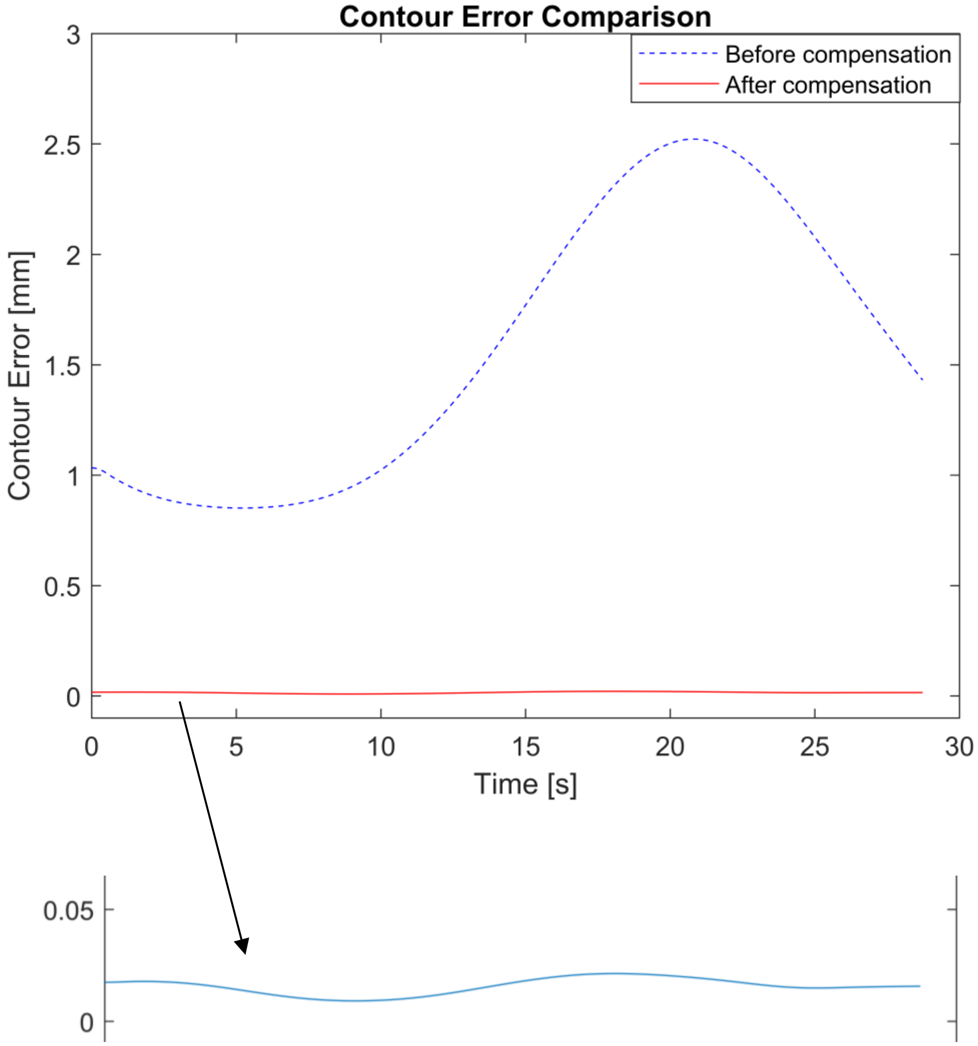


Figure 5.13: Contour Error Comparison Before and After Compensation.

5.3.2 Compensation of Input Shaping Distortions

As investigated in Chapter 4, input shapers allow robots to follow a path without significant vibrations, but the 'shaped' trajectory may exhibit distortions relative to the original path. Similar to cutting force induced contour errors, path distortions caused by input shaping can be predicted and corrected in advance. Figure 5.14 illustrates the framework for the compensation of contour errors caused by input shaping.

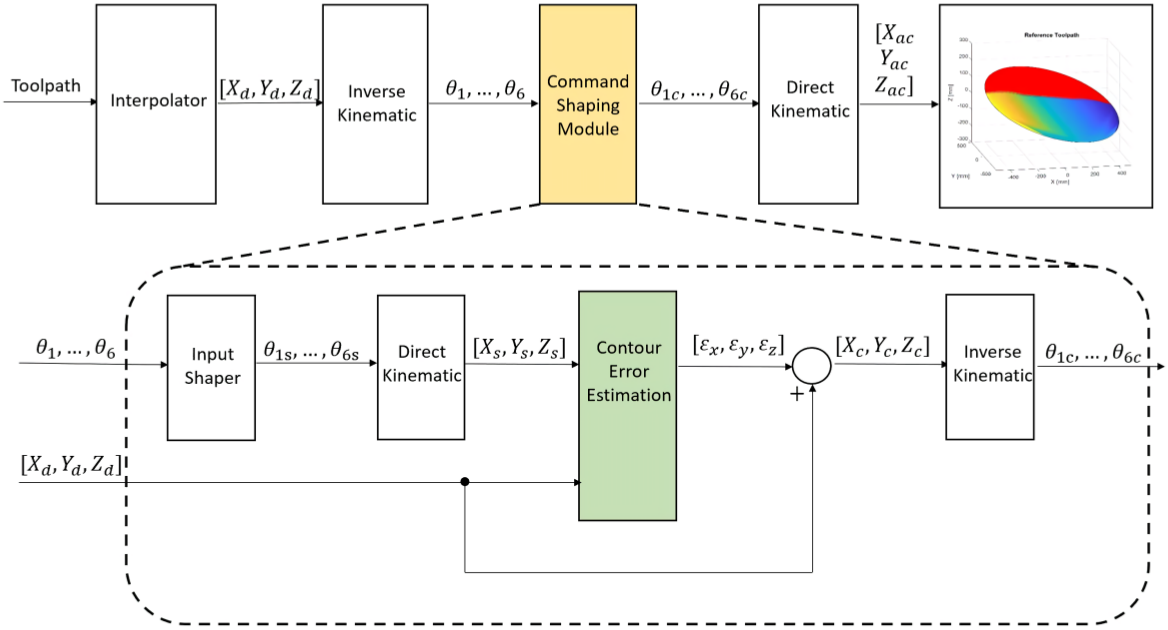


Figure 5.14: Block Diagram for Compensation of Input Shaping Distortions.

As shown in Figure 5.14, the original joint commands $\theta_1, \theta_2 \ldots \theta_6$ are first passed through the input shaping module to eliminate problematic harmonics of the reference trajectories. The shaped commands $\theta_{1s}, \theta_{2s} \ldots \theta_{6s}$ result in end-effector coordinates X_s, Y_s, Z_s, which can be obtained by the direct kinematic model. In the contour error estimation module, the desired (X_d, Y_d, Z_d) and shaped tool coordinates (X_s, Y_s, Z_s) are compared to find the contour errors $\varepsilon_x, \varepsilon_y, \varepsilon_z$ induced by the input shaper. Similar to Eq. (5.4), the predicted contour errors are used to obtain the compensated trajectory, i.e. X_c, Y_c, Z_c. The compensated joint commands $\theta_{1c}, \theta_{2c} \ldots \theta_{6c}$ are also obtained using the inverse kinematic model. These commands are then sent to the robot controller.

Figure 5.15 compares the contour errors before and after compensation for the ZVD shaper simulated in Figure 5.9. It can be seen that the proposed compensation technique can nearly eliminate all contour errors caused by input shaping.

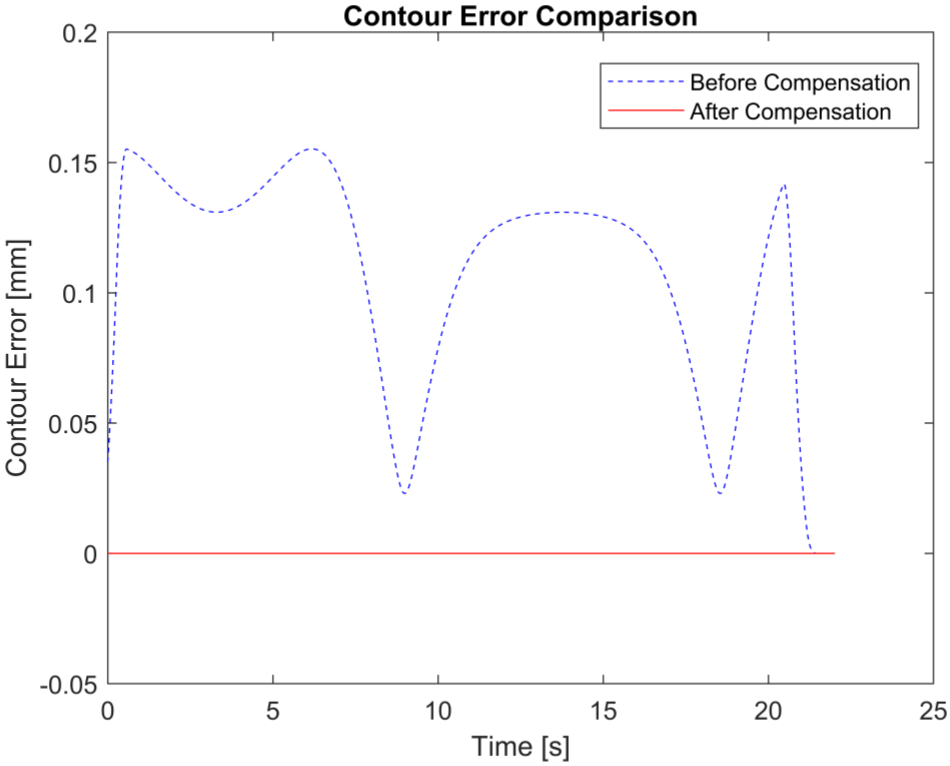


Figure 5.15: Compensation of Input Shaping Distortions.

5.4 Unified Compensation Framework for Higher Efficiency

Compensation of contour errors caused by cutting forces and input shaping distortions was investigated individually in the previous section. In order to improve the efficiency of the compensation technique, this section presents a unified framework that combines and compensates both errors together. This approach enhances computational efficiency as well as the accuracy of the proposed compensation technique. Figure 5.16 presents the block diagram of the unified compensation framework.

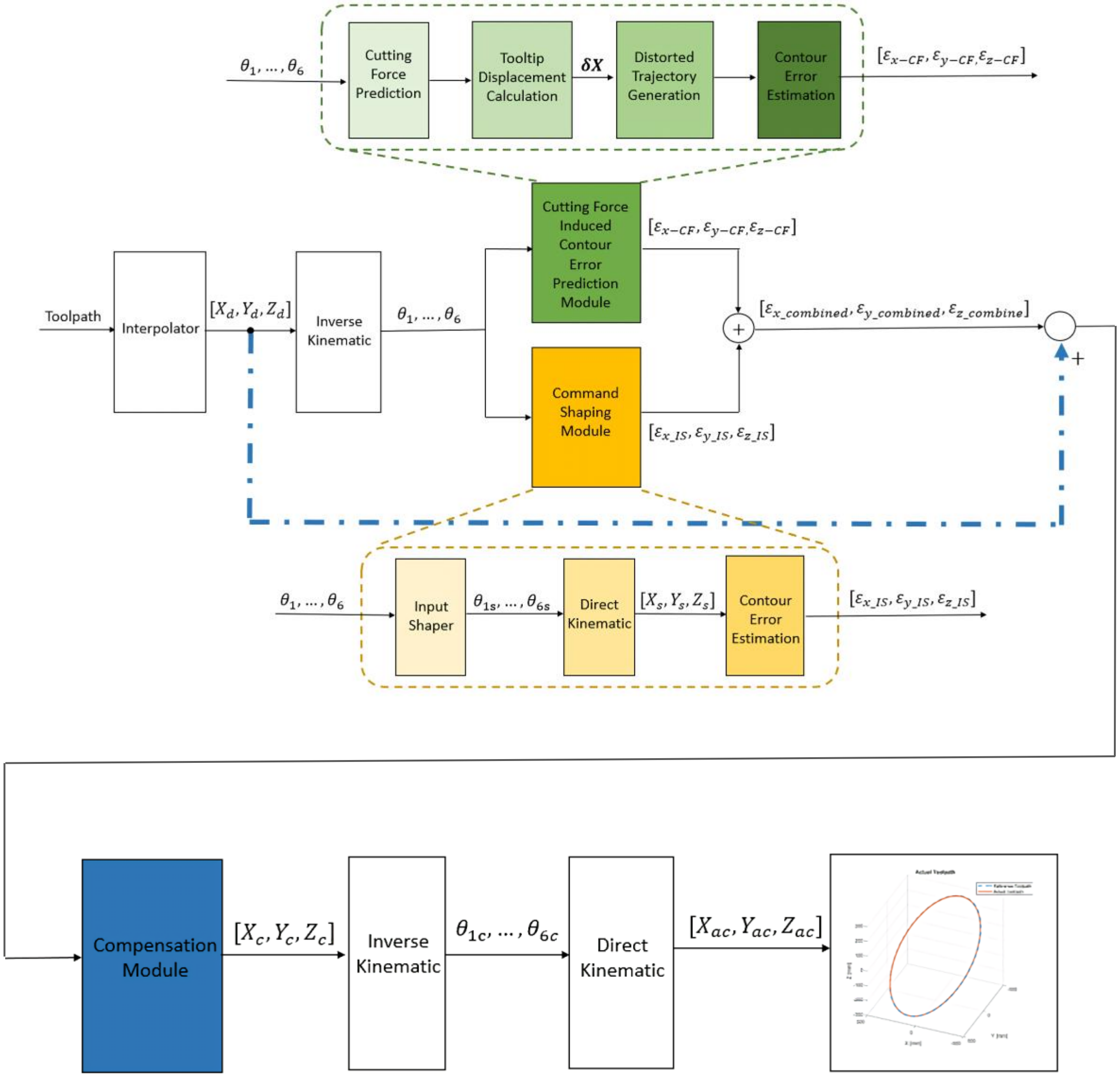


Figure 5.16: Block Diagram of the Unified Compensation Module.

In Figure 5.16, the green module predicts the contour errors caused by cutting forces and the yellow module determines the path distortions due to input shaping. Figure 5.17 shows the individual contour errors as well as their summation (combined contour error) that is to be compensated. The trend of the combined error is dominated by the cutting-force induced errors; however, both errors must be compensated to achieve the desired accuracy.

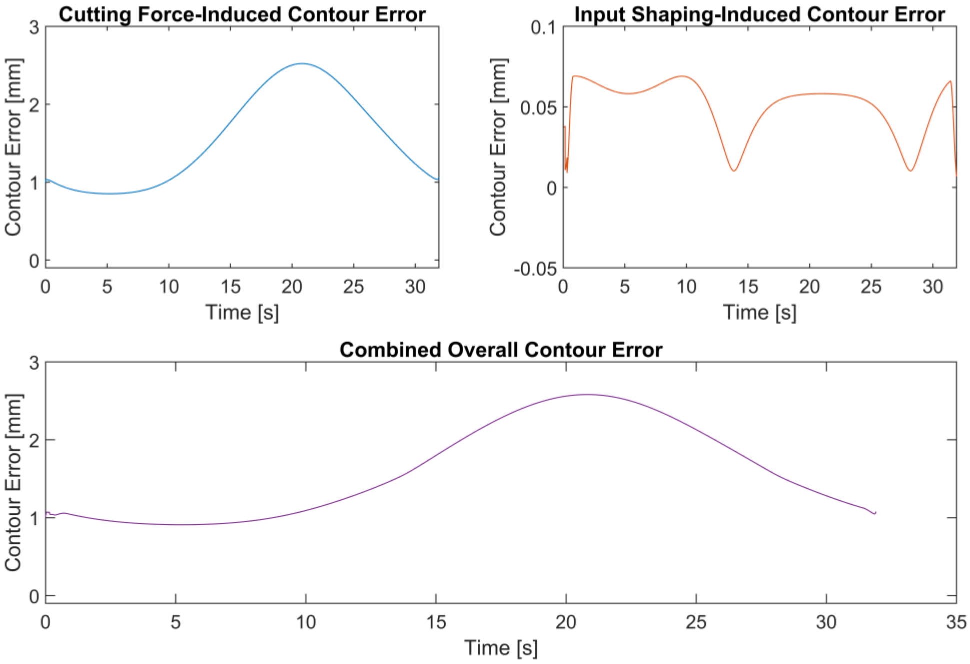


Figure 5.17: Combined Contour Error Due to Both Cutting Forces and Input Shaping.

The combined contour error calculated in Figure 5.17 is imported into the blue module (compensation) in Figure 5.16. This module calculates the compensated path as presented in Eq. (5.4). Figure 5.18 compares the total (combined) contour errors before and after compensation. It can be seen that the contour error has been reduced from 2.5 mm to less than 0.02 mm, which is within the tolerances allowed by most machining processes.

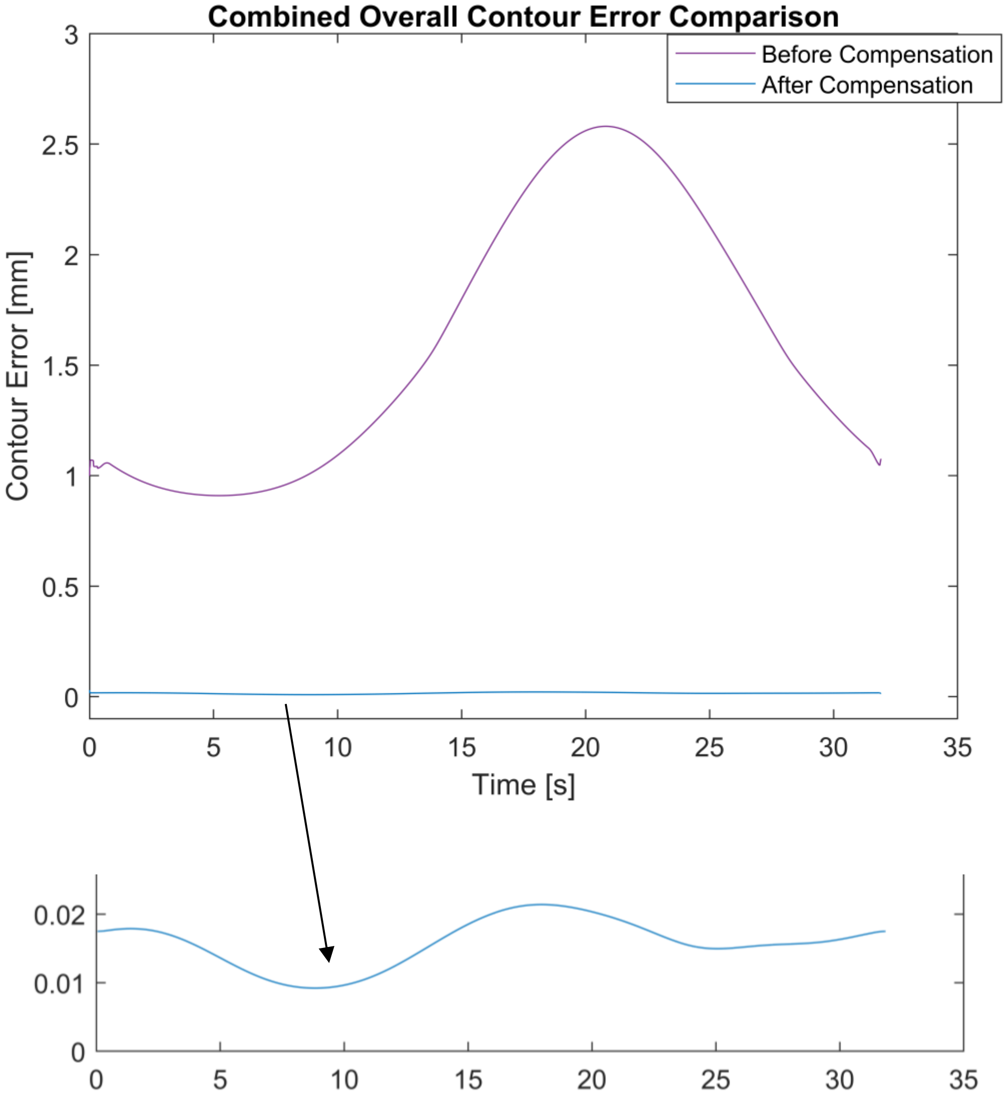


Figure 5.18: Combined Contour Error Before and After Compensation.

The unified compensation model predicts and compensates for both errors in a single step. This not only reduces the computational load but also enhances the accuracy and efficiency of the compensation technique.

Finally, it is worth mentioning that the compensation model developed in this section is generic and can be used to compensate for a variety of contour errors caused by different error sources. This compensation algorithm can be used to generate corrected reference trajectories to reduce both geometric and non-geometric errors, as long as contour errors can be accurately predicted in advance.

5.5 Summary

This chapter presents the modeling and calculation of contour errors with known reference and actual trajectories in robotic machining. A systematic compensation module was proposed to correct for contour errors by adjusting reference commands ahead of time. This compensation model works for different types of positioning errors such as cutting force-induced deflections and input shaping-induced trajectory distortions. The effectiveness of the developed compensation technique was verified through the simulation of the response of the Stäubli RX-90 industrial robot.

6 Conclusions

This thesis proposes a systematic framework for the prediction and compensation of contour errors in robotic machining systems. Two representative sources of error are considered in this thesis: structural deflections caused by the low stiffness of robots, and the path distortions caused by input shapers. Kinematics and compliance of a Stäubli RX-90 industrial robot are formulated mathematically, simulation algorithms are developed in MATLAB. A practical cutting force model is presented to predict the deflections of the robot arm due to machining forces. Contour error compensation models are developed to correct the deviation of the actual machining path from the desired reference. Simulation results show that the positioning error compensation methodology as proposed in this thesis is effective in reducing various contour errors in machining.

The contributions and conclusions of this thesis are summarized below:

(1) A brief overview of the Stäubli RX-90 industrial robot is given and its structural characteristics are analyzed. The D-H method is used to deduce the forward and inverse kinematic model of the robot. It is shown that the forward kinematic model has a unique solution while the inverse kinematic model can provide 6 solutions, some of which may not be feasible due to the configuration constraints of the robot.

(2) The compliance model of the Stäubli RX-90 industrial robot is developed and its Jacobian matrix is solved. The mapping of joint stiffness at the end-effector, i.e. the Cartesian compliance model, is derived and simulated in MATLAB. It is shown that even with constant joint compliance values, the robot compliance at the end-effector varies across the workspace.

(3) A prediction model for instantaneous cutting forces during robotic machining is discussed. Based on the estimation of the cutting force, a functional relationship between joint compliance and robot deformation is obtained. The tooltip displacement predicted by the model shows positioning errors exceeding 2 mm. It is shown that even with constant

cutting forces, instantaneous deflection at the end-effector varies as a function of robot pose and machining direction.

(4) The concept of input shaping for suppression of robot vibration is introduced. It is shown that input shaping can cause path distortions leading to deviations from the desired machining trajectory. The predicted distortions can exceed 0.15 mm; therefore, achieving machining accuracy below 100 um requires compensation of input shaping induced errors.

(5) A systematic error compensation model is developed to improve the accuracy of machining robots. The proposed model predicts and pre-compensates the contour errors caused by both cutting forces and input shaping. By adjusting the input joint commands, the positioning errors are corrected in advance, thus making the actual trajectory to more closely match the desired trajectory.

(6) It is shown that input shaping induced errors can nearly be eliminated completely due to their fully deterministic nature. Cutting force induced errors, on the other hand, depend strongly on the accuracy and completeness of the prediction model, and are more difficult to compensate.

(7) The entire framework and algorithms have been programmed in MATLAB. The simulation results prove that the developed error compensation model can significantly improve the accuracy of machining robots.

The following research areas can be investigated in future studied:

(1) In this thesis, joint stiffness is represented by a linear torsional spring whereas in practice, wear and backlash in gears introduce some nonlinearity into the model. The effect of nonlinearities in joint stiffness must be taken into account to improve the error prediction model.

(2) This study does not experimentally identify the real joint stiffness parameters of the robot but rather uses nominal parameters. Experimental identification tests can be designed to improve the accuracy of the compliance model.

(3) The developed error compensation model in this thesis is for the case of Cartesian (3-axis) machining. The case of general 5-axis milling and the associated compensation models for both position and orientation of the milling tool can be studied in future studies.

(4) While the focus of this thesis was on offline compensation of positioning errors, the best results can be achieved when offline and real-time error compensation techniques are combined. Real-time error compensation of cutting force induced errors is an important direction for future research.

Bibliography

[1] Reuleaux, F. (2013). *The Kinematics of Machinery: Outlines of a Theory of Machines*. Courier Corporation.

[2] Denavit, J., & Hartenberg R. S. (1955). A Kinematic Notation for Lower-pair Mechanisms Based on Matrices. *Journal of Applied Mechanics*, *22*(6), 215-221.

[3] Atique, Md. M. U., & Ahad, Md. A. R. (2014). Inverse Kinematics solution for a 3DOF robotic structure using Denavit-Hartenberg Convention. *2014 International Conference on Informatics, Electronics Vision (ICIEV)*, 1–5.

[4] Siciliano, B., & Khatib, O. (2016). *Springer Handbook of Robotics*. Springer.

[5] McCarthy, J. M., & Soh, G. S. (2010). *Geometric Design of Linkages*. Springer Science & Business Media.

[6] Chen, Y., & Dong, F. (2013). Robot machining: Recent development and future research issues. *The International Journal of Advanced Manufacturing Technology*, *66*(9), 1489–1497.

[7] Alici, G., & Shirinzadeh, B. (2005). Enhanced stiffness modeling, identification and characterization for robot manipulators. *IEEE Transactions on Robotics*, *21*(4), 554–564.

[8] Abele, E., Rothenbücher, S., & Weigold, M. (2008). Cartesian compliance model for industrial robots using virtual joints. *Production Engineering*, *2*(3), 339.

[9] Spong, M. W., & Vidyasagar, M. (2008). *Robot Dynamics And Control*. Wiley India Pvt. Limited.

[10] Dumas, C., Caro, S., Chérif, M., Garnier, S., & Furet, B. (2010). A methodology for joint stiffness identification of serial robots. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 464–469.

[11] Klimchik, A., Bondarenko, D., Pashkevich, A., Briot, S., & Furet, B. (2014). Compliance Error Compensation in Robotic-Based Milling. *Informatics in Control, Automation and Robotics, 283*.

[12] Cvitanic, T., Nguyen, V., & Melkote, S. N. (2020). Pose optimization in robotic machining using static and dynamic stiffness models. *Robotics and Computer-Integrated Manufacturing*, *66*, 101992.

[13] Bruder, D., Remy, C. D., & Vasudevan, R. (2019). Nonlinear System Identification of Soft Robot Dynamics Using Koopman Operator Theory. *2019 International Conference on Robotics and Automation (ICRA)*, 6244–6250.

[14] Youngsu Cho, Munyu Kim, Cheong, J., Do, H., & Kyung, J. (2017). Simultaneous identification of kinematic screw and joint compliance of elastic robot manipulators using deflected circular trajectories. *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 382–387.

[15] Milutinovic, D. (1997). Universal compliant device based on SCARA concept. *Robotics and Computer-Integrated Manufacturing*, *13*(4), 319–321.

[16] Yang, T., Fan, S.-H., & Lin, C.-S. (2003). Joint stiffness identification using FRF measurements. *Computers & Structures*, *81*(28), 2549–2556.

[17] Pan, Z., & Zhang, H. (2008). Robotic machining from programming to process control: A complete solution by force control. *Industrial Robot: An International Journal*, *35*, 400–409.

[18] Stelzer, M., Stryk, O. von, Abele, E., Bauer, J., & Weigold, M. (2008). *High speed cutting with industrial robots: Towards model based compensation of deviations*. 143–146.

[19] Altintas, Y. (2000). *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. Cambridge University Press.

[20] Tapoglou, N., & Antoniadis, A. (2012). CAD-Based Calculation of Cutting Force Components in Gear Hobbing. *Journal of Manufacturing Science and Engineering*, *134*.

[21] Wei, C., Zhao, L., Hu, D., & Ni, J. (2013). Electrical discharge machining of ceramic matrix composites with ceramic fiber reinforcements. *The International Journal of Advanced Manufacturing Technology*, *64*(1), 187–194.

[22] Wang, J., Zhang, J., & Feng, P. (2017). Effects of tool vibration on fiber fracture in rotary ultrasonic machining of C/SiC ceramic matrix composites. *Composites Part B: Engineering*, *129*, 233–242.

[23] Azarhoushang, B., & Tawakoli, T. (2011). Development of a novel ultrasonic unit for grinding of ceramic matrix composites. *International Journal of Advanced Manufacturing Technology*, *57*(9–12), 945–956.

[24] Rajasekaran, T., Palanikumar, K., & Vinayagam, B. K. (2012). Turning CFRP Composites with Ceramic tool for Surface Roughness Analysis. *Procedia Engineering*, *38*, 2922–2929.

[25] Sekhar, R., & Singh, T. P. (2015). Mechanisms in turning of metal matrix composites: A review. *Journal of Materials Research and Technology*, *4*(2), 197–207.

[26] Yuan, S., Fan, H., Amin, M., Zhang, C., & Guo, M. (2016). A cutting force prediction dynamic model for side milling of ceramic matrix composites C/SiC based on rotary ultrasonic machining. *The International Journal of Advanced Manufacturing Technology*, *86*(1), 37–48.

[27] Budak, E. (2006). Analytical models for high performance milling. Part I: Cutting forces, structural deformations and tolerance integrity. *International Journal of Machine Tools and Manufacture*, *46*(12), 1478–1488.

[28] Ammar, A. A., Bouaziz, Z., & Aghal, A. (2009). Modelling and simulation of the cutting forces for 2.5 D pockets machining. *Advances in Production Engineering & Management, 4*(4), 163-176.

[29] Singer, N. C., & Seering, W. P. (1990). Preshaping Command Inputs to Reduce System Vibration. *Journal of Dynamic Systems, Measurement, and Control*, *112*(1), 76–82.

[30] Altintas, Y., & Khoshdarregi, M. R. (2012). Contour error control of CNC machine tools with vibration avoidance. *CIRP Annals*, *61*(1), 335–338.

[31] Newman M., Lu K.Y. & Khoshdarregi, M. R. (2020). Suppression of Robot Vibrations Using Input Shaping and Learning-Based Structural Models. *Journal of Intelligent Material Systems and Structures*

[32] Kang, C.-G. (2011). On the derivative constraints of input shaping control. *Journal of Mechanical Science and Technology*, *25*(2), 549–554.

[33] Shyh-Leh Chen, Hung-Liang Liu, & Sing Ching Ting. (2002). Contouring control of biaxial systems based on polar coordinates. *IEEE/ASME Transactions on Mechatronics*, *7*(3), 329–345.

[34] Altintas, Y., & Sencer, B. (2010). High speed contouring control strategy for five-axis machine tools. *CIRP Annals - Manufacturing Technology*, *59*, 417–420.

[35] Ming-Yang Cheng, & Cheng-Chien Lee. (2005). On real-time contour error estimation for contour following tasks. *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 1047–1052.

[36] El Khalick M., A., & Uchiyama, N. (2011). Contouring controller design based on iterative contour error estimation for three-dimensional machining. *Robotics and Computer-Integrated Manufacturing*, *27*(4), 802–807.

[37] Kamalzadeh, A., & Erkorkmaz, K. (2007). Accurate tracking controller design for high-speed drives. *International Journal of Machine Tools and Manufacture*, *47*(9), 1393–1400.

[38] Ouyang, P. R., Acob, J., & Pano, V. (2014). PD with sliding mode control for trajectory tracking of robotic system. *Robotics and Computer-Integrated Manufacturing*, *30*(2), 189–200.

[39] Ruderman, M. (2014). Tracking Control of Motor Drives Using Feedforward Friction Observer. *IEEE Transactions on Industrial Electronics*, *61*(7), 3727–3735.

[40] Sato, R., & Tsutsumi, M. (2007). Friction Compensator for Feed Drive Systems Consisting of Ball Screw and Linear Ball Guide. In *Proceedings of the 35th International MATADOR 2007 Conference* (pp. 311–314).

[41] Uchiyama, N., Nakamura, T., & Yanagiuchi, H. (2009). The effectiveness of contouring control and a design for three-dimensional machining. *International Journal of Machine Tools and Manufacture*, *49*(11), 876–884.

[42] Huang, S., Bergström, N., Yamakawa, Y., Senoo, T., & Ishikawa, M. (2016). High-performance robotic contour tracking based on the dynamic compensation concept. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 3886–3893.

[43] Khoshdarregi, M. R., Tappe, S., & Altintas, Y. (2014). Integrated Five-Axis Trajectory Shaping and Contour Error Compensation for High-Speed CNC Machine Tools. *IEEE/ASME Transactions on Mechatronics*, *19*(6), 1859–1871.

[44] Zhang, K., Yuen, A., & Altintas, Y. (2013). Pre-compensation of contour errors in five-axis CNC machine tools. *International Journal of Machine Tools and Manufacture*, *74*, 1–11.

[45] Zeng, Y., Tian, W., & Liao, W. (2016). Positional error similarity analysis for error compensation of industrial robots. *Robotics and Computer-Integrated Manufacturing*, *42*, 113–120.

[46] Kim, S. H., & Min, B.-K. (2020). Joint Compliance Error Compensation for Robot Manipulator Using Body Frame. *International Journal of Precision Engineering and Manufacturing*, *21*(6), 1017–1023.

[47] Schneider, U., Drust, M., Ansaloni, M., Lehmann, C., Pellicciari, M., Leali, F., Gunnink, J. W., & Verl, A. (2016). Improving robotic machining accuracy through experimental error investigation and modular compensation. *The International Journal of Advanced Manufacturing Technology*, *85*(1), 3–15.

[48] Zhu, W., Li, G., Dong, H., & Ke, Y. (2019). Positioning error compensation on two-dimensional manifold for robotic machining. *Robotics and Computer-Integrated Manufacturing*, *59*, 394–405.

[49] Nguyen, H.-N., Le, P.-N., & Kang, H.-J. (2019). A new calibration method for enhancing robot position accuracy by combining a robot model–based identification approach and an artificial neural network–based error compensation technique. *Advances in Mechanical Engineering*, *11*(1), 1687814018822935.

[50] Stone, H. W. (1987). *Kinematic Modeling, Identification, and Control of Robotic Manipulators*. Springer Science & Business Media.

[51] *SCARA, 6-axis, FAST picker and collaborative robots / Stäubli*. (n.d.). Retrieved July 3, 2020, from https://www.staubli.com/en/robotics/

[52] Abderrahim, M., Khamis, A., Garrido, S., & Moreno, L. (2006). Accuracy and Calibration Issues of Industrial Manipulators. *Industrial Robotics: Programming, Simulation and Applications*.

[53] Bottin, M., Cocuzza, S., Comand, N., & Doria, A. (2020). Modeling and Identification of an Industrial Robot with a Selective Modal Approach. *Applied Sciences*, *10*(13), 4619.

[54] Slavkovic, N., Zivanovic, S., Kokotovic, B., Dimic, Z., & Milutinovic, M. (2020). Simulation of compensated tool path through virtual robot machining model. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *42*(7), 374.

[55] Kumar, K., Ranjan, C., & Davim, J. P. (2020). Drilling and Milling Operation. In K. Kumar, C. Ranjan, & J. P. Davim (Eds.), *CNC Programming for Machining* (pp. 55–60). Springer International Publishing.

[56] Mamedov, A., & Lazoglu, I. (2016). An evaluation of micro milling chip thickness models for the process mechanics. *The International Journal of Advanced Manufacturing Technology*, *87*(5–8), 1843–1849.

[57] Slavkovic, N., Milutinovic, D., & Glavonjic, M. (2014). A method for off-line compensation of cutting force-induced errors in robotic machining by tool path modification. *The International Journal of Advanced Manufacturing Technology*, *70*.