An Investigation of Kinematic Redundancy for Reduced Error in Micromilling

by

Yogesh Madhavrao Chukewad

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2014 by the
Graduate Supervisory Committee:

Joseph Davidson, Co-Chair
Angela Sodemann, Co-Chair
Veronica Santos

ARIZONA STATE UNIVERSITY

August 2014

ABSTRACT

Small metallic parts of size less than 1mm, with features measured in tens of microns, with tolerances as small as 0.1 micron are in demand for the research in many fields such as electronics, optics, and biomedical engineering. Because of various drawbacks with non-mechanical micromanufacturing processes, micromilling has shown itself to be an attractive alternative manufacturing method. Micromilling is a microscale manufacturing process that can be used to produce a wide range of small parts, including those that have complex 3-dimensional contours. Although the micromilling process is superficially similar to conventional-scale milling, the physical processes of micromilling are unique due to the scale effects. These scale effects occur due to unequal scaling of the parameters from the macroscale to the microscale milling. One key example of scale effects in micromilling process is a geometrical source of error known as chord error. The chord error limits the feedrate to a reduced value to produce the features within machining tolerances. In this research, it is hypothesized that the increase of chord error in micromilling can be alleviated by intelligent modification of the kinematic arrangement of the micromilling machine. Currently, all 3-axis micromilling machines are constructed with a Cartesian kinematic arrangement with three perpendicular linear axes. In this research, the cylindrical kinematic arrangement is introduced, and an analytical expression for the chord error for this arrangement is derived. The numerical simulations are performed to evaluate the chord errors for the cylindrical kinematic arrangement. It is found that cylindrical kinematic arrangement gives reduced chord error for some types of the desired toolpaths. Then, the kinematic redundancy is introduced to design a novel kinematic arrangement. Several desired toolpaths have been numerically simulated to evaluate the chord error for kinematically redundant arrangement. It is concluded that this arrangement gives up to 5 times reduced error for all the desired toolpaths considered, and allows significant gains in allowable feedrates.

DEDICATION

To my parents

ACKNOWLEDGMENTS

I would like to thank my colleague Nikita Rode, and friends Rohit Singwal, and Mukul Saha whose encouragement and friendship has brought happiness to my work.

I am thankful to committee: Drs. Joseph Davidson and Veronica Santos, whose knowledge contribution, effort, and thoughtful consideration has made this thesis possible. I would like to express my gratitude to my advisor and mentor: Dr. Angela Sodemann for her support and guidance throughout this project and my stay at ASU.

Finally, I am thankful to my family: Aai, Pappa, Ganesh, Aarti, and Dr. Laxman whose love and support continues to inspire me to pursue my goals.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Micromilling is the preferred method of manufacture for many types of small metallic parts. These small metallic parts are currently in high-demand in many industries which require either the metallic parts themselves, or molds for making small plastic parts through methods such as micro-injection molding [1]. These industries include biomedicine, the electronics field, and consumers of small optical parts like small lenses [2]. The biomedical engineering requires microparts for making devices for studying various processes in the human body [3,4], and for making human body implants [5,6]. The metallic parts are typically less than 1mm in size, and have sub-micron tolerances. Other methods besides micromilling can be used to produce these parts: micro-EDM, micro-ECM [7], and laser micro-machining [8] are examples of non-mechanical methods available to produce these parts. Other mechanical methods available for producing small metallic parts include micro lathe turning [9] and micro-injection molding [1]. However, these methods have low material-removal-rates, making them undesirable for mass-production of small metallic parts.

In this chapter, the process of micromilling is addressed first. Then, the scale effects that make the micromilling a different physical process from macroscale milling are explained. The concept of trajectory generation for micromilling is then presented, which is followed by the concept of kinematic redundancy adopted from robotics. The research objectives of this study are presented, along with the hypotheses that are to be tested to achieve the research objectives. Finally, the thesis overview is given to understand the flow of this thesis.

1.1 MICROMILLING

On the surface, the micromilling process is abstractly equivalent to the macroscale milling process, as in both the processes, the spinning toothed endmill removes the material from the workpiece. Being similar to macroscale endmills, the microendmills are typically made of tungsten carbide with 2 or 4 flutes. However, in micromilling, the endmill size is smaller than 1 mm, and currently it is as small as 5 microns in diameter [10], and spindle speeds are one or two orders of magnitude higher than the macroscale milling.

1.2 SCALE EFFECTS

In addition to the endmill size and spindles speeds affecting the micromilling process on the surface, the ‘scale effects’ also make the microscale milling process different from the macroscale milling. These scale effects arise due to unequal scaling of physical parameters from macroscale to microscale milling processes.

One key example of the scale effects is the increased ratio of tool size to the feature size [11] in micromilling process. This increased ratio of tool size to the feature size leads to a machining error known as chord error. Apart from scale effects, the desired toolpath, machining feedrate and the trajectory generation loop update time also affect the chord error. Altering the desired toolpath for achieving the reduced chord error is not desirable. On the other hand, the trajectory generation loop update time cannot be reduced due to several limitations on the system. Therefore, the only reasonable way to reduce the chord error is reducing the feedrate. However, reduced feedrate has two important negative effects on the micromilling process:

1. Reduced feedrate increases machining time, thus decreasing the part production rate and adding undue time cost to a mass-production operation and
2. Reduced feedrate can cause decreased tool life due to rubbing of the tool on the workpiece without cutting when the minimum chip thickness is violated.

Therefore, it becomes necessary to increase the feedrate for reducing the cost of micromilling, without increasing the chord error. The existing micromilling machines typically consist of 3 axes. In these machines, there are three linear axes in X, Y and Z directions. Typically, XY plane lies in the plane of the table on which a workpiece is fixed. The workpiece is usually fixed on a linear axis which is placed perpendicularly on the top of another linear axis, forming an XY plane. The third linear axis is placed along the Z axis, which means, all the three linear axes are mutually perpendicular to each other. Therefore, this arrangement can also be called *Cartesian kinematic arrangement*. A 3-axis micromilling machine is capable of producing parts in 2.5-dimensions. The parts produced by this arrangement have depth, but the parts with contours outside of XY, XZ, or YZ planes cannot be produced with Cartesian kinematic arrangement.

A micromilling machine is usually run with the help of Computer Numerically Controlled (CNC) systems. In CNC systems, a controller performs the trajectory generation from a toolpath desired by the user. The trajectory generation consists of determining the velocities of the motors attached to the joints. The controller performs this trajectory generation in real-time to compensate for the process errors. Therefore, the real-time trajectory generation requires reading the real-time positions of the axes and thus, updating the velocities of the axis motors. This process of performing the real-time trajectory generation creates a trajectory loop apart from the standard velocity loop that operates on individual motors. This dual-loop control is illustrated in Figure 1.1.

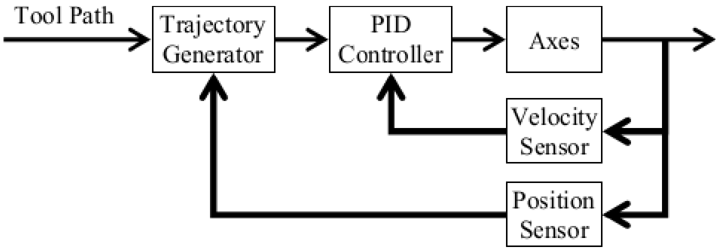

Figure 1.1: Diagram of the two control loops involved in CNC control

The two controls loops as shown in Figure 1.1 do not operate at the same speed. The outer loop of trajectory generation operates at a slower rate than the inner velocity determination loop. The time required to compute the trajectories limits the rate at which the velocity commands are sent to the axis motors. The *controller trajectory generation loop update time* corresponds to the time required for trajectory generation, plus the communication delay between the controller and the machine.

The spindles with very high speeds are used in a micromilling machine. The speeds of commercially-available spindles range from 50,000 RPM to 250,000 RPM. The spindle used in micromilling setup at Arizona State University is a high speed air spindle with maximum speed of 80,000 RPM. The electric spindles are also commercially available for micromilling process. Spindles with speeds of 1,000,000 RPM have recently become an active area of research [12]. However, high speed spindles require sufficiently high feedrates, which can be obtained by microendmills as small as 5 microns in diameter [10].

1.3 TRAJECTORY GENERATION

The process of trajectory generation involves computing the joint velocities according to the desired toolpath. Trajectory generation is performed in real-time to compensate for the process errors. The trajectory generation controller loop update time depends upon the time required to generating the joint velocities. However, the non-uniform B-spline (NURBS) definition increases the computational load on the process of trajectory generation, thus increasing the trajectory generation controller loop update time. Reducing this update time has become an active area of research [13, 14, 15].

1.4 KINEMATIC REDUNDANCY

Path-planning in manufacturing is generating the path for the tool to follow. A CAM (Computer-Aided Manufacturing) software performs the path-planning by taking the desired part as the input and thus calculating the toolpath. In robotics, this process includes the process of inverse kinematics. The path-planning should not be confused with the trajectory planning, which is the stage after path-planning. Once the desired toolpath is generated by path-planning, the velocities of the tool to follow the desired toolpath are generated with the process of trajectory planning. In micro-manufacturing, the trajectory planning depends upon the number of kinematics equations to be solved.

Path planning becomes difficult when the number of inverse kinematics equations are less than the total number of unknowns. This is the case when inverse kinematic equations are under-constrained. In robotics, this case is termed as kinematic redundancy. Though this case is widely addressed in robotics, it has not yet been addressed in micromanufacturing. A CAM software can generally handle maximum of five axes of motion. These axes typically include three linear axes and two rotational axes. The inverse kinematic equations have to be well-constrained to be processed by the CAM software. A kinematically redundant robot was studied for milling at the macroscale by Andres et al. [16]. Six rotary joints on a linear track were addressed in this study. However, the study did not address the benefits of kinematic redundancy in milling, and did not apply to chord error in micromilling. In robotics, the kinematically redundant manipulator is typically used to avoid obstacles in the desired end-effector path. It is also used to avoid singularities in several cases. For this reason, Mi et al. [17] considered a kinematically redundant manipulator to make the end-effector perform specific macroscale operations such as welding, painting and soldering.

As the case of kinematic redundancy occurs when the set of inverse kinematic equations is under-constrained, additional constraints are required to make this set well-defined. Without the additional constraints, the CAM software would yield infinite number of solutions for the given set of inverse kinematic equations. These additional constraints can then be used for optimization of parameters. This additional constraint was used by Zhang and Wang [18] to develop an artificial neural control for a redundant manipulator for avoiding obstacles. Tatlicioglu et al. [19] used the additional constraints for avoiding singularity, joint limit and for bounding of impact forces and potential energy.

1.5 RESEARCH OBJECTIVES

The enhancement of high-speed, high-precision milling at the macroscale is widely studied. However, the same at microscale has not yet been sufficiently studied due to incomplete understanding of scale effects impacting the micromilling. In order to improve the micromilling speed and precision, these scale effects must be properly addressed. This study aims to attain a new knowledge by addressing the limitations in the existing knowledge. More specific limitations are as follows:

- Existing micromilling research does not fully take into account the chord error which is less significant at macroscale.
- Existing micromilling research does not take into account the effect of kinematic arrangement on the high speed and high precision production.

This research will discuss these limitations, and their impact on the precision and speed of micromilling. The following four objectives will be specifically addressed in this thesis.

Objective 1: To achieve new knowledge which enhances the understanding of key factors affecting the high-speed, high-precision micromilling

Objective 2: To utilize new knowledge to understand the need of a new mechanism which can be used to enhance high-speed, high-precision micromilling

Objective 3: To apply the knowledge of robotics for developing a new kinematic arrangement for compensation of chord error in micromilling.

1.6 HYPOTHESES

Reduced feedrate is one approach to achieve reduced chord error. However, because of high speed spindles used in micromilling, the feedrate also needs to be relatively high. So, it becomes necessary to introduce a way to reduce the chord error without negatively affecting the feedrate.

It is hypothesized that the kinematic arrangements used at the macroscale is not desirable to be used at the microscale as it does not necessarily compensate for the scale effects. A better kinematic arrangement is introduced to compensate for the chord error. Productivity can be enhanced and high-speed spindles can also be used through increased feedrate achievable with new kinematic arrangements.

Following three hypotheses are tested to achieve previously stated objectives:

Hypothesis 1: The existing Cartesian kinematic arrangement used for macroscale and microscale milling is not optimal for the reduction of chord error.

Hypothesis 2: A cylindrical kinematic arrangement reduces the chord error for some types of desired toolpaths.

Hypothesis 3: Chord errors can always be reduced by introducing a kinematically redundant arrangement into the design of a micromilling machine

1.7 THESIS OVERVIEW

In Chapter 2, the background information about chord error is presented, and the information on the kinematic arrangement of present micromilling machine is given. The chord error for cylindrical arrangement is analytically derived, and the trajectory planning for micromilling is presented in Chapter 3. Chapter 4 reports the algorithm developed to numerically evaluate the hypotheses through experimentation setup, and the experimental results are given. The results are discussed and the conclusions are made in Chapter 5. Future work is outlined in Chapter 6. Then, the references are included, which is followed by an appendix that reports the code developed to perform numerical simulations.

CHAPTER 2

BACKGROUND INFORMATION

This chapter presents the details of the concepts that are necessary to understand the analysis and experimentation performed in this research. Section 2.1 defines the concept of chord error, the key process error addressed in this thesis. Section 2.2 discusses the kinematic arrangement that is currently being used in industry for micromilling. This section also discusses the dimension of the joint space and the work space. Finally, the concept of kinematic redundancy is addressed and the significance of kinematically redundant arrangement is also discussed.

2.1 CHORD ERROR

In the process of trajectory generation, the velocities of the joints are calculated during the process of machining. As the velocities of the joints are constant within the sampling interval, the endmill in Cartesian arrangement traverses a linear path within that interval as shown in Figure 2.1. Figure 2.1 also illustrates the chord error along with other machining errors in a micromilling operation.

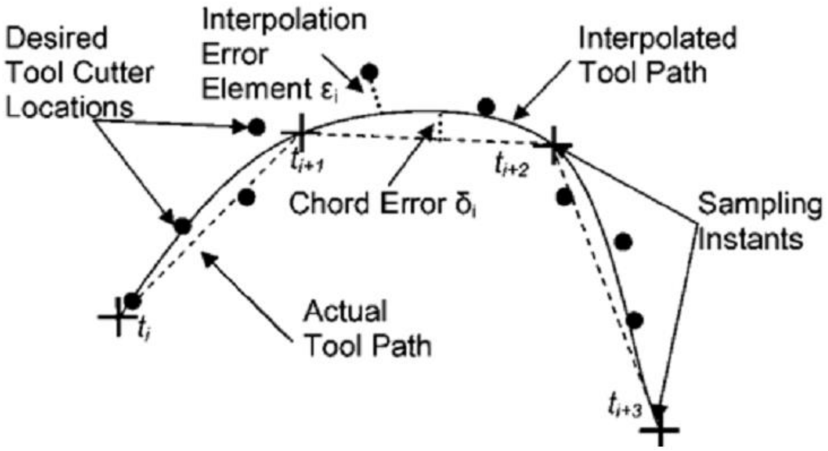

Figure 2.1: Chord Error and Interpolation Error in a micromilling toolpath [15]

According to Figure 2.1, an interpolated toolpath is generated from the desired tool cutter locations. This interpolated toolpath then serves as the desired toolpath for the endmill. The actual toolpath shown in the Figure 2.1 is the toolpath traversed by the endmill in Cartesian arrangement in micromilling operation.

Stoker in 1969 [21] was the first person to define the chord error in geometry. Stoker's definition of chord error was then adopted in Cartesian arrangement milling operation by Sun et al in 2006 [22]. Chord error is defined as the maximum Euclidean distance between the interpolated toolpath and the actual toolpath.

The chord error δ incurred in Cartesian arrangement micromilling operation can be calculated according to Equation 2.1 as determined by Sun et al [22], where Ts is the sampling time, ρ is the radius of curvature of the desired toolpath, and f is the feedrate.

$$\delta = \rho - \sqrt{\rho^2 - \frac{f^2 Ts^2}{4}} \tag{2.1}$$

From Equation 2.1, it can be seen that the chord error can be reduced-

- By reducing the radius of curvature of the desired toolpath,
- By reducing the machining feedrate, and/or
- By reducing the sampling time.

Apart from the relation in Equation 2.1, the feedrate is directly proportional to the speed of the spindle that holds the tool, and the feedrate has to be sufficiently high to maximize the tool life. Reduced feedrate also reduces the production rate, which is also not desirable.

Feedrate optimization is one of the techniques to achieve reduced error. This technique is well addressed in the literature [20, 22, 23, 24]. This literature mainly addresses the milling operation at macroscale. According to this technique, the feedrate can be reduced to get the chord error within the specified tolerances. As an upper limit on the chord error is imposed, the maximum feedrate depends only upon the radius of curvature of the desired toolpath. Equation 2.2 [20] gives the maximum feedrate f_{max} that can be reached to get the chord error below δ_{max}. The variables ρ and Ts are the radius of curvature of the desired toolpath, and the system's sampling time respectively.

$$f_{max} = \frac{2}{Ts}\sqrt{\rho^2 - (\rho - \delta_{max})^2} \tag{2.2}$$

This study proposes that the maximum feedrate not only depends upon the radius of curvature, sampling time and maximum chord error, but also depends upon the kinematic arrangement of the operation. This study concentrates on increasing the feedrate without affecting the chord error, by introducing the Cylindrical and kinematically redundant arrangements.

2.2 KINEMATICS

Kinematics of a mechanism involves the mechanism which determines the mechanical relation between the endmill and the workpiece. The standard Cartesian kinematic arrangement for a three-axis micromilling machine is as shown in Figure 2.2.

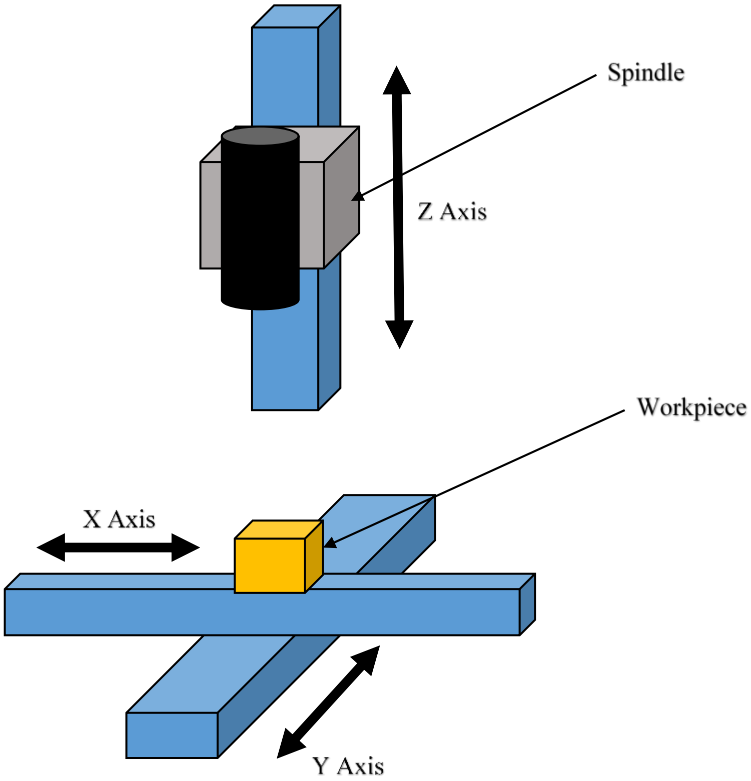

Figure 2.2: Standard Cartesian kinematic arrangement for the micromilling machine

The linear joints along X and Y axes are placed orthogonal to each other and in series, as shown in Figure 2.2. The workpiece is then placed on the top joint which is along the X-axis. Another linear joint along the Z-axis is attached in parallel with the series of the joints along X and Y axes. Then the endmill through the spindle is attached to this Z-axis. The endmill is not shown here for the sake of simplicity.

In the arrangement shown in Figure 2.2, the joint space consists of three linear joints, therefore it has three degrees of freedom. On the other hand, the workspace i.e. the space in which endmill moves is also three-dimensional. The endmill also rotates about its own axis because of a rotary joint present in the spindle. However, this rotation and the joint are not taken into the account as they do not affect the positioning of the endmill relative to the workpiece.

If Q and R denote the joint and workspaces respectively, then the forward kinematic relation, f can be represented by Equation 2.3.

$$f : Q \rightarrow R \tag{2.3}$$

The three-dimensional motion of the endmill can be easily controlled using the three joints, as there are three unknowns for three joints and three equations for desired toolpath in XYZ space. Therefore, the Equation 2.3 for this motion points out a one-to-one mapping.

Multiaxis machining is a very commonly used process in manufacturing, in which there are 4 or more axes to rotate the tool and/or the table to which the workpiece is fixed. However, multiaxis machining is not considered as a kinematically redundant process, as the number of axes never exceeds the dimension of the workspace. Therefore, the 4-axis and 5-axis kinematic arrangements which are very common in the industries meet the main purpose of altering the position and/or orientation of the workpiece relative to the tool after completion of a machining operation. The automated alteration of the workpiece position and/or orientation reduces the amount of labor work.

If another joint is introduced in 3-axis mechanism, the dimension of the joint space, Q changes from three to four. If this extra joint does not lead to a motion that changes the dimension of the workspace, R remains same as three. Therefore the mechanism is said to be *kinematically redundant*. In the forward kinematic equations, there are now four unknowns and three equations, hence, it leads to infinitely many solutions for the joint variables. The solution set for the joint variables can be made unique, if one more independent equation is introduced. This additional equation is referred as a *constraint*, which is usually used to optimize a parameter. This is explained in mathematical form with the help of Equation 2.4. Matrix A is a $n * m$ rectangular matrix with rank n therefore $m > n$. X is a column matrix of the unknowns. Therefore, matrix X has the dimension $m * 1$. The matrix B then has dimension $n * 1$. Then Equation 2.4 represents the matrix form for solving linear equations.

$$A_{n*m}X_{m*1} = B_{n*1} \tag{2.4}$$

Equation 2.4 leads to infinitely many solutions. Therefore, there are infinitely many solutions to the set of equations. If $m - n$ independent linear equations are introduced in matrix A in Equation 2.4, its dimension and rank then become $m * m$ and m respectively, and hence the dimension of matrix B becomes $m * 1$. This leads to the set of linear equations to have unique solution. Therefore, in kinematics, $m - n$ independent constraints have to be introduced to make the joint velocities have unique values.

CHAPTER 3

ANALYSIS

This chapter will present the concept of kinematic arrangements in detail. It will then present the kinematic arrangements that are hypothesized for this research. It will then define the chord error and present the analytical derivation of chord error for a micromilling machine which consists of a cylindrical joint and a linear joint which lies on any diameter of the rotary joint. First, the terminology to be utilized in derivation is defined. Next, the mathematical definitions are presented for key elements of the terminology. An expression for chord error is then derived. Finally, trajectory planning for all the kinematic arrangements considered in this analysis is addressed.

3.1 KINEMATIC ARRANGEMENTS

It has been hypothesized in this research that the kinematic arrangement plays an important role with the accuracy of the toolpath. The kinematic arrangement for the standard Cartesian micromilling machine as was shown in Figure 2.2 in the last chapter can be altered by changing the number of joints, by replacing the linear joints by rotational joints, or by both. The joints in a micromilling machine are connected to each other in series or in parallel. So, another way of altering the kinematic arrangement is to change the way in which the joints are connected.

One example of an alternate kinematic arrangement for a micromilling machine is to replace the linear Y axis with a rotational joint, as shown in Figure 3.1.

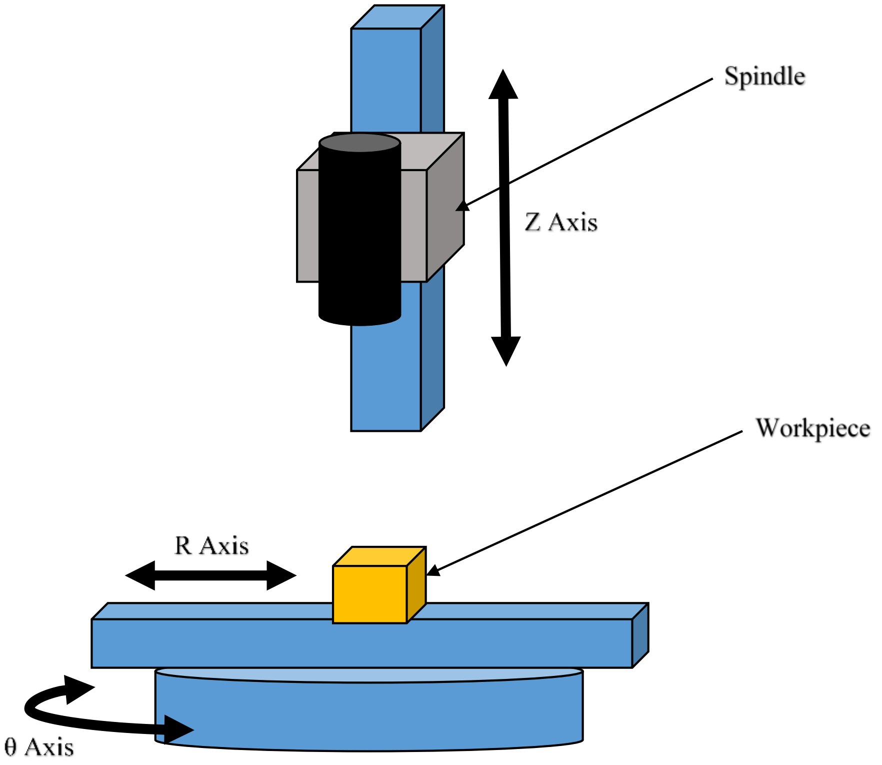

Figure 3.1: Cylindrical kinematic arrangement for a micromilling machine

It is hypothesized here that one of the benefits of an alternate kinematic arrangement is a variation in the type of toolpath achievable by the machine.

The standard Cartesian kinematic arrangement consists of only linear joints. If the joint velocities are held constant, the endmill can only trace a path that is a straight line. In contrast, the micro-parts that can be manufactured by micromilling often consist of curved surfaces. So, in the two-dimensional case, micro-parts can be said to have curvilinear edges. Thus, the curvilinear edges are interpolated by the Cartesian micromilling machine as straight lines between any two subsequent sampling points.

In the alternate cylindrical kinematic arrangement shown in Figure 3.1, the endmill can be made to follow a certain curvilinear path. This curvilinear path is referred to as an Archimedean spiral. The cylindrical parametric coordinates of a point on an Archimedean spiral are given by Equations 3.1 and 3.2.

$$r = u \tag{3.1}$$

$$\theta = a + b * u \tag{3.2}$$

Figure 3.2 shows an example of Archimedean spiral drawn between two points (2,2) and (5,1).

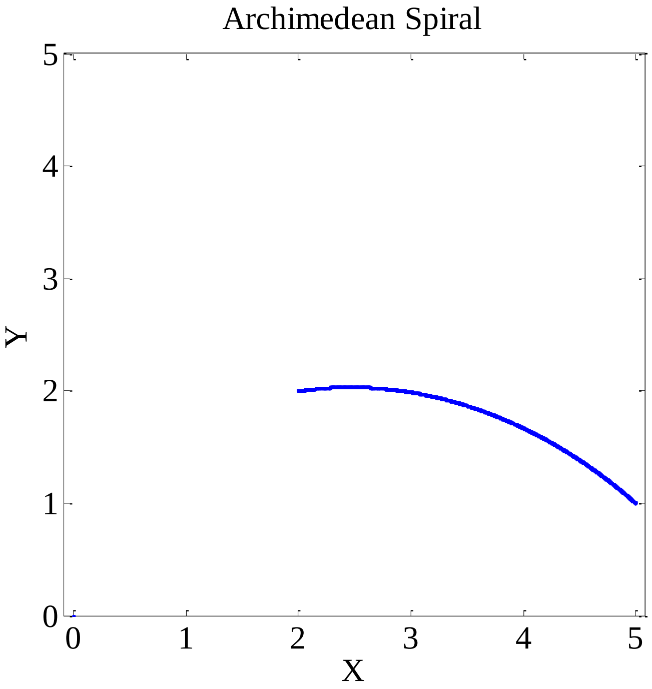

Figure 3.2: Archimedean Spiral

In the example shown in Figure3.2, the center of the spiral is at the origin (0,0). A position vector is a vector that is drawn from the origin to a given point. If the magnitudes of position vectors drawn to two given points are equal, the Archimedean spiral is equivalent to a circular arc. If the magnitudes of the position vectors are different and the slopes of the position vectors are same, the Archimedean spiral is equivalent to a straight line.

The standard Cartesian Arrangement and the new cylindrical arrangement can be combined into a single machine with a kinematically redundant arrangement, as shown in Figure 3.3.

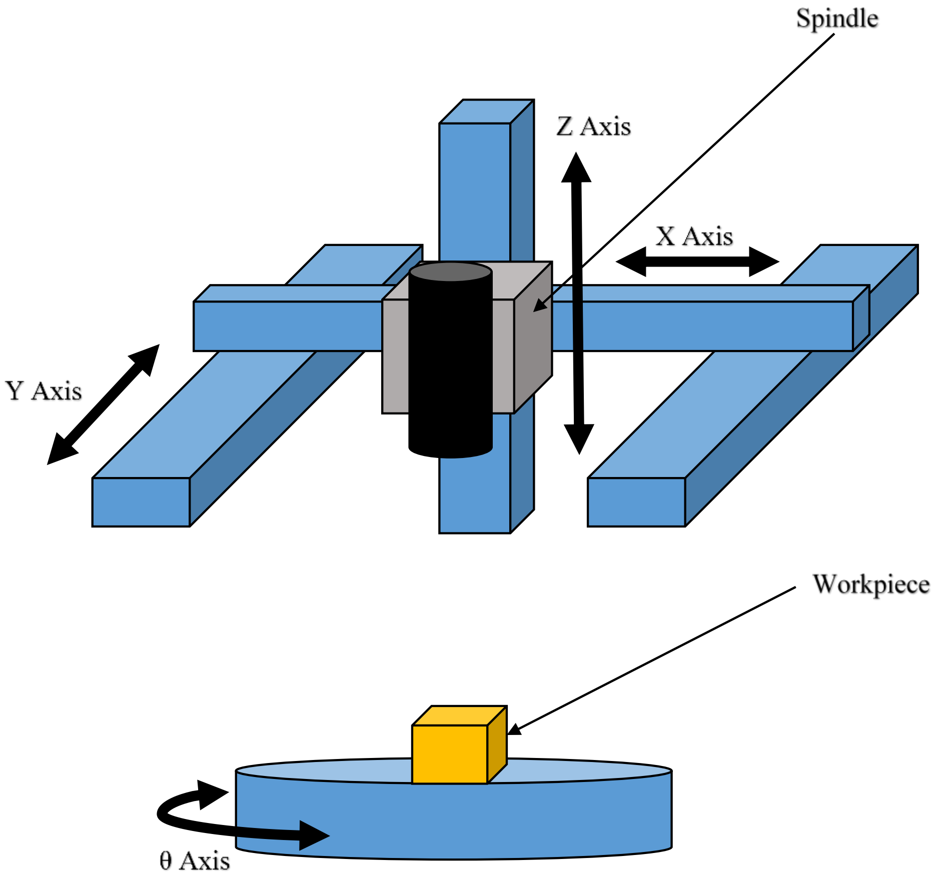

Figure 3.3: Proposed kinematically redundant micromilling machine

The proposed kinematically redundant arrangement has three linear joints and one rotary joint, whereas the dimension of the task space remains to be three. Thus, the arrangement is kinematically redundant. In this arrangement, the endmill is attached to the linear joint along the Z-axis which is attached to another linear joint which is along X-axis. This joint is then attached to one more linear joint which is along Y-axis, forming a standard Gantry mechanism. The set of all these linear joints is then placed in parallel with a rotary joint. The workpiece is placed on this rotary joint.

As discussed in Section 2.2, an additional dimension of the joint space leads to introduction of a free parameter which allows for an additional constraint equation. The constraint equation in this research is the chord error equation.

3.2 DEFINITION OF CHORD ERROR

The concept of chord error is well-defined in the literature for the standard Cartesian kinematic arrangement. In order to evaluate the effectiveness of the proposed alternate kinematic arrangements, the concept of chord error for the new arrangements has to first be defined.

In the definition of chord error, the term “Desired toolpath” is defined as the curve the endmill is supposed to follow. This desired toolpath is discretized to get the sampling points. These points on the desired toolpath are equidistant along the curve. Because of limitations of the hardware as well as the kinematic arrangement, the endmill fails to exactly follow the desired toolpath. So, the “actual toolpath” can be defined as the path followed by the endmill. “Chord error” is then defined as the maximum Euclidean distance between the desired and actual toolpaths between two subsequent sampling points, such that the chord error segment is always perpendicular to the actual toolpath. The desired and actual toolpaths along with four subsequent sampling points and chord error are illustrated in Figure 3.4.

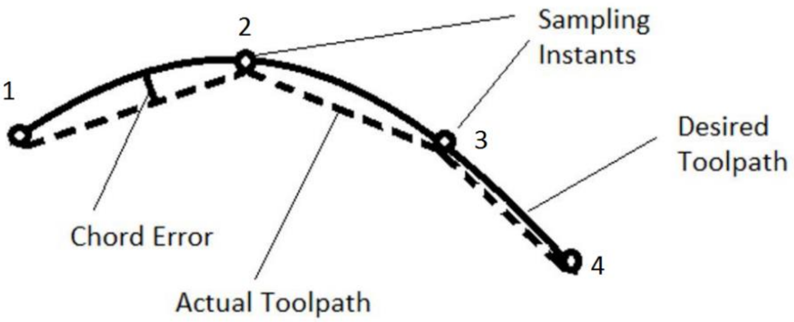

Figure 3.4: Desired toolpath, actual toolpath, sampling instants and chord error

In both of the proposed kinematic arrangements, one of the axes used in the kinematic arrangement is rotary. Thus, the actual toolpath is not necessarily a straight line between two subsequent sampling instants.

3.3 ANALYTICAL DERIVATION OF CHORD ERROR FOR CYLINDRICAL MICROMILLING MACHINE

In this section, the chord error is analytically derived for the Cylindrical Arrangement. The axes velocities are assumed to be constant between two subsequent sampling points. The linear axis velocity is denoted by $\dot{r}$, whereas that of the rotary axis is denoted by $\dot{\theta}$. The variable T is the sampling time of the control system. The sampling time of the control system depends on the capability of the hardware, the efficiency of the programming, and the complexity of the calculations that have to be performed between sampling points. Figure 3.5 shows the desired toolpath and the actual toolpath along with the intermediate points.

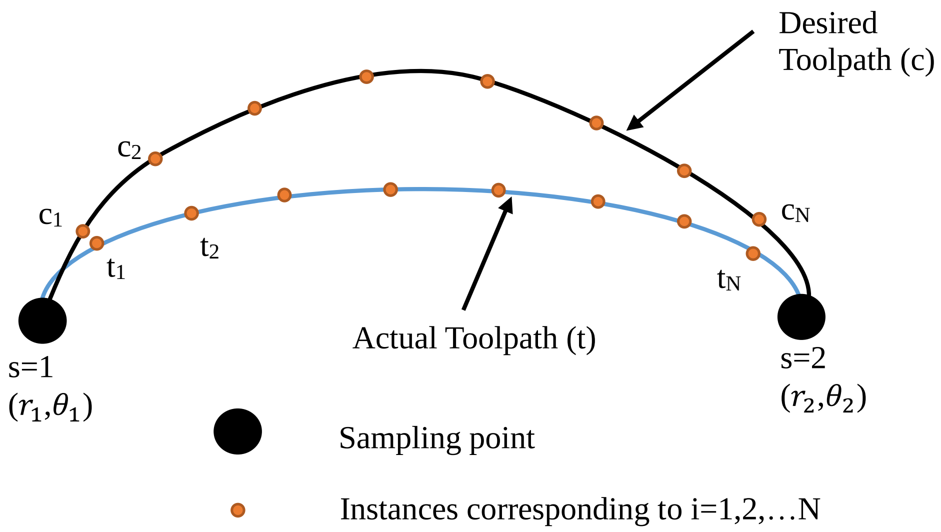

Figure 3.5: Desired toolpath, actual toolpath and intermediate points

The variables (r_1,θ_1) and (r_2,θ_2) are the polar coordinates of sampling points 1 and 2, then $\dot{r}$ and $\dot{\theta}$ are given by Equations 3.3 and 3.4 respectively.

$$\dot{r} = \frac{r_2 - r_1}{T} \tag{3.3}$$

$$\dot{\theta} = \frac{\theta_2 - \theta_1}{T} \tag{3.4}$$

The polar coordinates of a point on the toolpath at any time instant t_i, between subsequent sampling points can be defined as in Equations 3.5 and 3.6.

$$r(t_i) = r_1 + t_i\,\dot{r} \tag{3.5}$$

$$\theta(t_i) = \theta_1 + t_i\,\dot{\theta} \tag{3.6}$$

The Cartesian coordinates of the same point can be written according to Equations 3.7 and 3.8.

$$x(t_i) = r(t_i)\cos(\theta(t_i)) \tag{3.7}$$

$$y(t_i) = r(t_i)\sin(\theta(t_i)) \tag{3.8}$$

The $r(t_i)$ and $\theta(t_i)$ from Equations 3.5 and 3.6 respectively can be substituted into Equations 3.7 and 3.8 to get Equations 3.9 and 3.10 respectively.

$$x(t_i) = (r_1 + t_i\,\dot{r})\cos(\theta_1 + t_i\,\dot{\theta}) \tag{3.9}$$

$$y(t_i) = (r_1 + t_i\,\dot{r})\sin\left(\theta_1 + t_i\,\dot{\theta}\right) \tag{3.10}$$

As the chord error segment is perpendicular to the toolpath, its equation is derived by finding out its slope. The partial derivatives of Cartesian coordinates given by Equations 3.9 and 3.10 with respect to t_i are given by Equations 3.11 and 3.12 respectively.

$$\frac{\partial x}{\partial t_i} = -\dot{\theta}(r_1 + t_i\,\dot{r})\sin\left(\theta_1 + t_i\,\dot{\theta}\right) + \dot{r}\cos(\theta_1 + t_i\,\dot{\theta}) \tag{3.11}$$

$$\frac{\partial y}{\partial t_i} = \dot{\theta}(r_1 + t_i\,\dot{r})\cos\left(\theta_1 + t_i\,\dot{\theta}\right) + \dot{r}\sin(\theta_1 + t_i\,\dot{\theta}) \tag{3.12}$$

Now, the slope *m* of the chord error segment is given by Equation 3.13

$$m = \frac{-\frac{\partial x}{\partial t_i}}{\frac{\partial y}{\partial t_i}} \tag{3.13}$$

Partial derivatives given by Equations 3.11 and 3.12 can be substituted into Equation 3.13 to get the slope represented by Equation 3.14.

$$m = \frac{\dot{\theta}(r_1 + t_i\,\dot{r})\sin\left(\theta_1 + t_i\,\dot{\theta}\right) - \dot{r}\cos(\theta_1 + t_i\,\dot{\theta})}{\dot{\theta}(r_1 + t_i\,\dot{r})\cos\left(\theta_1 + t_i\,\dot{\theta}\right) + \dot{r}\sin(\theta_1 + t_i\,\dot{\theta})} \tag{3.14}$$

The general equation of the chord error segment can be written as Equation 3.15.

$$y = mx + b \tag{3.15}$$

The line given by Equation 3.15 passes through the point whose coordinates are given by Equations 3.9 and 3.10.

$$b = y(t_i) - mx(t_i) \tag{3.16}$$

Let $(x(c_i), y(c_i))$ be the point on the curve which is closest to a point $(x(t_i), y(t_i))$ which is given by Equations 3.9 and 3.10. Therefore, the chord error segment also passes through $(x(c_i), t(c_i))$.

$$y(c_i) = mx(c_i) + y(t_i) - mx(t_i) \tag{3.17}$$

Therefore, the chord error which will be denoted (δ_i) at any time instant t_i, is given by Equation 3.18.

$$\delta_i = \sqrt{(x(c_i) - x(t_i))^2 + (y(c_i) - y(t_i))^2} \tag{3.18}$$

Equation 3.17 is then used to substitute the value of $y(c_i)$ into Equation 3.18.

$$\delta_i = \sqrt{(x(c_i) - x(t_i))^2 + (mx(c_i) + y(t_i) - mx(t_i) - y(t_i))^2} \tag{3.19}$$

Equation 3.19 is simplified to get Equation 3.20.

$$\delta_i = \sqrt{(x(c_i) - x(t_i))^2 + m^2(x(c_i) - x(t_i))^2} \tag{3.20}$$

Equation 3.20 can be further simplified to get Equation 3.21.

$$\delta_i = \sqrt{(1 + m^2)}\big(x(c_i) - x(t_i)\big) \tag{3.21}$$

At sampling point 1 as illustrated in Figure 3.4, $x(c_i)$= $x(t_i)$ and $y(c_i)$= $y(t_i)$. Also, t=c=0 can be assumed at the same point, as it is the starting point of the toolpath. The values of the points *(x, y)* at the toolpath starting point is defined in Equations 3.22 and 3.23.

$$x(c_0) = r_1 \cos\theta_1 \tag{3.22}$$

$$y(c_0) = r_1 \sin\theta_1 \tag{3.23}$$

Therefore, at any time instant t_i, the coordinates of a point on the curve can be written according to the Equation 3.24 and Equation 3.25.

$$x(c_i) = (r_1 + \int_0^{c_i} \frac{\partial r(c)}{\partial c}\, dc) \cos(\theta_1 + \int_0^{c_i} \frac{\partial \theta(c)}{\partial c}\, dc) \tag{3.24}$$

$$y(c_i) = (r_1 + \int_0^{c_i} \frac{\partial r(c)}{\partial c}\, dc) \sin(\theta_1 + \int_0^{c_i} \frac{\partial \theta(c)}{\partial c}\, dc) \tag{3.25}$$

Equation 3.24 is then used to substitute $x(c_i)$ into Equation 3.21 to get the chord error at instant t_i which is given by Equation 3.26.

$$\delta_i = \sqrt{1+m^2}\left(\left(r_1 + \int_0^{c_i} \frac{\partial r(c)}{\partial c}\, dc\right)\cos\left(\theta_1 + \int_0^{c_i} \frac{\partial \theta(c)}{\partial c}\, dc\right) - (r_1 + \dot{r}t_i)cos\big(\theta_1 + \dot{\theta}t_i\big)\right) \tag{3.26}$$

The maximum of δ_is of all the instants between two sampling points is termed as the chord error. This maximum δ is the expression for chord error for a cylindrical coordinate system. Equation 3.26 gives a very complex expression. However, it can be inferred that this expression gives zero value for the case of circular desired toolpath. For a circular arc, $\frac{\partial r(c)}{\partial c}$ and $\dot{r}$ are zero, and $\int_0^{c_i} \frac{\partial \theta(c)}{\partial c}\, dc = \dot{\theta} t_i$, which lead to zero chord error for circular arc. As it is difficult to conclude anything about other types of curves, the necessity of numerical simulation arises.

3.4 TRAJECTORY PLANNING

Trajectory planning involves determining the velocity of the joints. The time taken to execute a trajectory between two consecutive sampling points is the sampling time of the control system. In this study, it is assumed that the joint motors have very high acceleration. The stages being used in the micromilling setup at Arizona State University have acceleration of 1g, so the total of acceleration and deceleration time within a sampling interval for these motors is less than 5% of the sampling time. Therefore, the velocities of the joints are assumed to be constant throughout a sampling interval.

3.4.1 Determination of sampling points

Sampling points are specific points on the desired toolpath which are recognized by the system in an interval that corresponds to the sampling time. Sampling points are defined by the feedrate of the machine. If the sampling points are chosen far apart, they will correspond to a large feedrate, and vice versa. To keep the chip thickness constant, and hence to stabilize the machining process, the feedrate is kept as close to a constant value as possible.

The curve equation is the only information needed about the desired toolpath for trajectory generation. The convenient way to input a curve is in its parameterized form. The parameter of the curve can be used to determine the Cartesian location of the curve at each sampling point. In order to keep the feedrate approximately constant, the sampling points must be evenly spaced along the curve. So, the curve is parameterized by arc-length prior to the trajectory generation stage.

The parametric equation of the curve is represented by $r(t)$, where t is called the *curve parameter*. Then, $\dot{r}(t)$ represents the derivative of $r(t)$ with respect to the parameter t. The length of the arc of the curve between $t = 0$ to t is represented by s. The process of parameterization by arc length is to express the curve as a function of the arc length s, rather than the general parameter t.

Figure 3.6 shows the flow diagram for parameterization of the curve by arc-length.

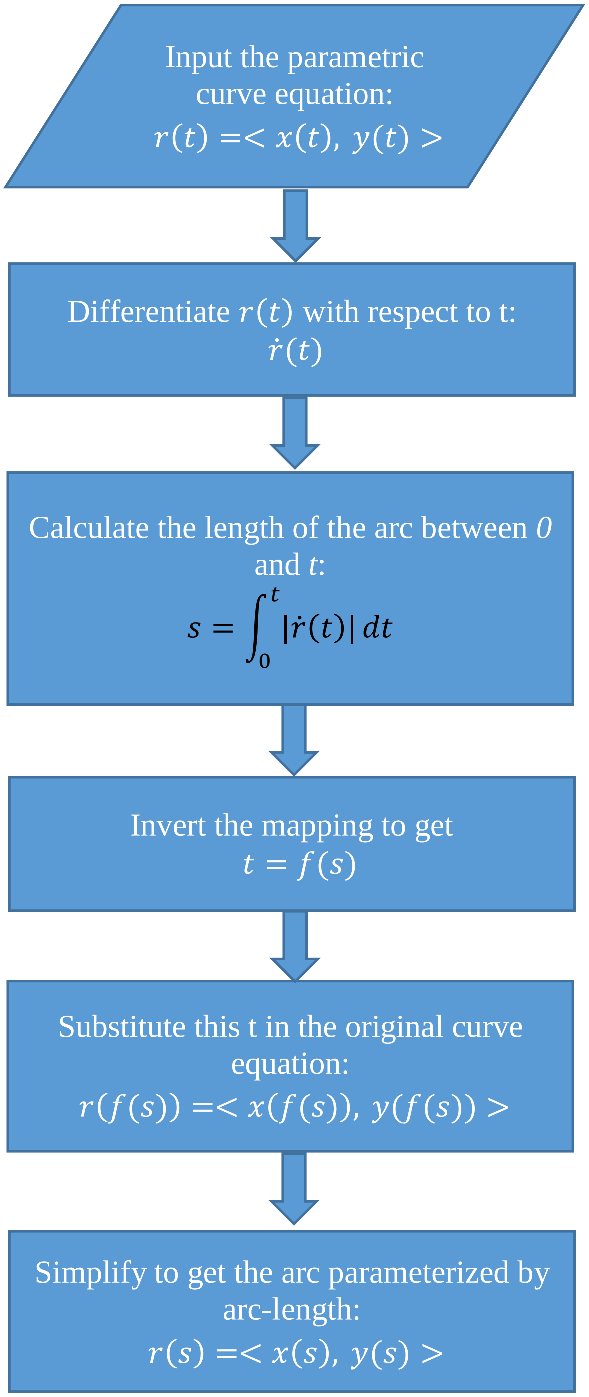
Input the parametric curve equation:
$r(t) =< x(t), y(t) >$
Differentiate $r(t)$ with respect to t:
$\dot{r}(t)$
Calculate the length of the arc between *0* and *t*:
$s = \int_0^t |\dot{r}(t)|\, dt$
Invert the mapping to get
$t = f(s)$
Substitute this t in the original curve equation:
$r(f(s)) =< x(f(s)), y(f(s)) >$
Simplify to get the arc parameterized by arc-length:
$r(s) =< x(s), y(s) >$

Figure 3.6: Flow-Diagram for parameterization of the curve equation by arc-length

The initial stage in parameterization of the curve involves inputting the $r(t)$. Once $r(t)$ is inputted, the next task is to differentiate it with respect to t to get $\dot{r}(t)$. Then, $\dot{r}(t)$ is integrated with respect to t with limits from 0 to t. This finite integral gives the relation of the arc-length s to the parameter *t*. The mapping can be inverted to get t as function of s. Then the $t = f(s)$ can be substituted in original equation of the curve to get the equation parameterized by arc-length.

Once the desired toolpath curve is parameterized by arc-length, the next task is to determine the coordinates of the sampling points. Figure 3.7 presents a flow diagram for determining the *x* and *y* coordinates of sampling points. The variable $Time$ in the Figure 3.7 is denotes the time for which the numerical simulation for calculating the chord error is run. The variable T is the sampling time that depends upon the CNC systems. Two variable arrays $X(i)$ and $Y(i)$ represent the *x* and *y* coordinates of the sampling points respectively. The total number of sampling points is finally represented by the variable- N.

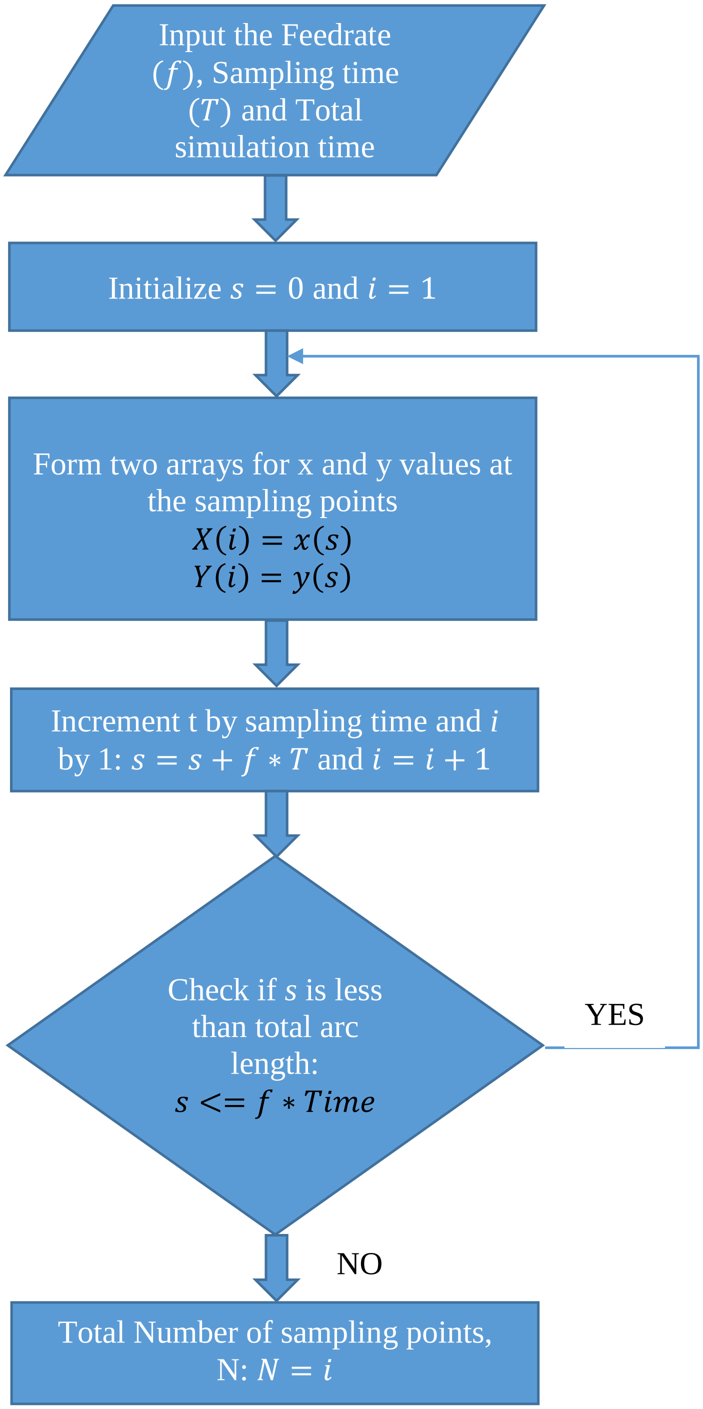
Input the Feedrate (f), Sampling time (T) and Total simulation time
Initialize $s = 0$ and $i = 1$
Form two arrays for x and y values at the sampling points
$X(i) = x(s)$
$Y(i) = y(s)$
Increment t by sampling time and i by 1: $s = s + f * T$ and $i = i + 1$
Check if s is less than total arc length:
$s <= f * Time$
YES
NO
Total Number of sampling points, N: $N = i$

Figure 3.7: Flow Diagram for determining *x* and *y* coordinates of sampling points

First, the feedrate, the total simulation time and the sampling time are inputted. Two arrays $X(i)$ and $Y(i)$ are formed to store the *x* and *y* coordinates of the sampling points. The curve parameter is increased by the feedrate times the sampling time after every loop. The curve parameter is kept increasing until it reaches the total curve length, i.e. feedrate times total simulation time. The total number of sampling points is stored in the variable N which was updated with the help of a counter.

3.4.2 Determination of joint velocities

Once the sampling points are determined, the next task is to determine the joint velocities at those points which is commonly known as trajectory generation. The joint velocities at each point are chosen in such a way that the endmill reaches the next sampling point after one sampling time. Therefore, these velocities depend upon the sampling time apart from the curve. In this section, the method of determining joint velocities for standard, cylindrical and kinematically redundant arrangements are described.

3.4.2.1 Determination of joint velocities for Standard Kinematic Arrangement

The standard kinematic arrangement used for micromilling is a Cartesian arrangement as was illustrated in Figure 2.1. The joints involved in this arrangement are the linear joints along X, Y and Z axes. As this research concentrates on geometry involved in X-Y plane, commonly known as 2.5-D in manufacturing, the linear joint along Z axis is neglected in this analysis.

The joint velocities along X and Y axes are represented by two arrays $VX(i)$ and $VY(i)$ respectively. The variable i corresponds to a counter.

Figure 3.8 shows the flow diagram for determining the velocities of the joints along X and Y axes at the sampling points in standard kinematic arrangement.

First, the sampling time is inputted. Then with the help of two arrays $X(i)$ and $Y(i)$, the velocities of the two linear joints are calculated. At every sampling point, the velocity of the linear axis along X axis is equal to difference of x coordinate of next sampling point and that at the current point, whole divided by the sampling time. The velocity of linear axis along Y axis is calculated in similar way. These two velocities are stored in two arrays $VX(i)$ and $VY(i)$ respectively. The loop is updated with the curve parameter s and the counter i until the velocities at all sampling points are determined.

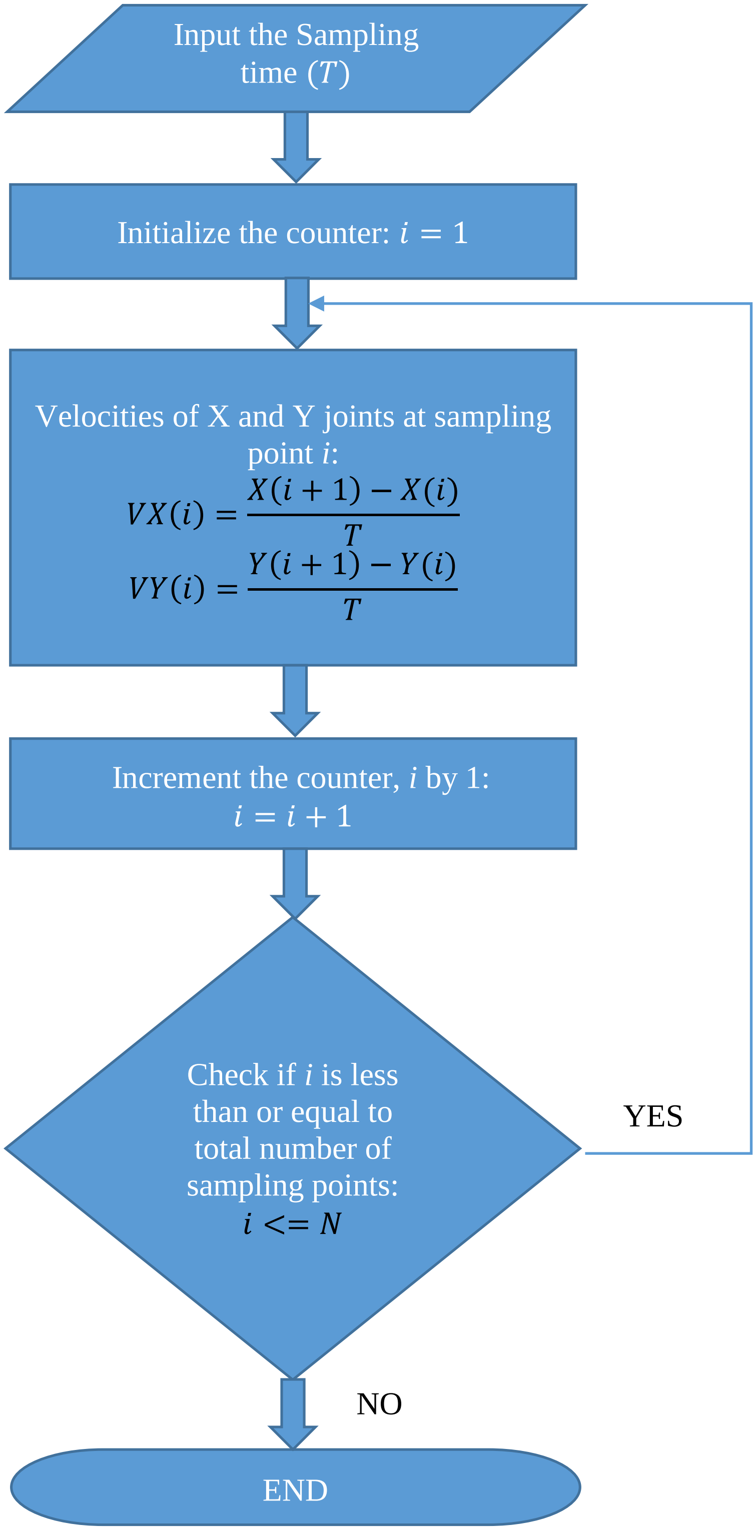
Input the Sampling time (T)
Initialize the counter: $i = 1$
Velocities of X and Y joints at sampling point i:
$VX(i) = \frac{X(i+1) - X(i)}{T}$
$VY(i) = \frac{Y(i+1) - Y(i)}{T}$
Increment the counter, i by 1:
$i = i + 1$
Check if i is less than or equal to total number of sampling points:
$i <= N$
YES
NO
END

Figure 3.8: Flow-Diagram for determining joint velocities for standard kinematic arrangement at the sampling points

3.4.2.2 Determination of joint velocities for Cylindrical Kinematic Arrangement

The cylindrical kinematic arrangement used for micromilling was shown in Figure 3.1. The Z axis is neglected in this analysis for the same reason as explained in Section 4.1.2.1. The cylindrical coordinate of the sampling points can be written according to Equations (4.1) and (4.2), where $R(i)$, $TH(i)$ are the cylindrical coordinates of the sampling point *i* and $\tan^{-1}(X(i), Y(i))$ is the two-argument arctangent.

$$R(i) = \sqrt{X(i)^2 + Y(i)^2} \tag{4.1}$$

$$TH(i) = \tan^{-1}(X(i), Y(i)) \tag{4.2}$$

The velocity of the rotary joint at the sampling points is represented by an array $VTH(i)$, and that of the radial linear axis represented by $VR(i)$. Figure 3.9 shows the flow diagram for determining the velocities of the rotary joint and the linear joint in the cylindrical kinematic arrangement shown in Figure 3.1.

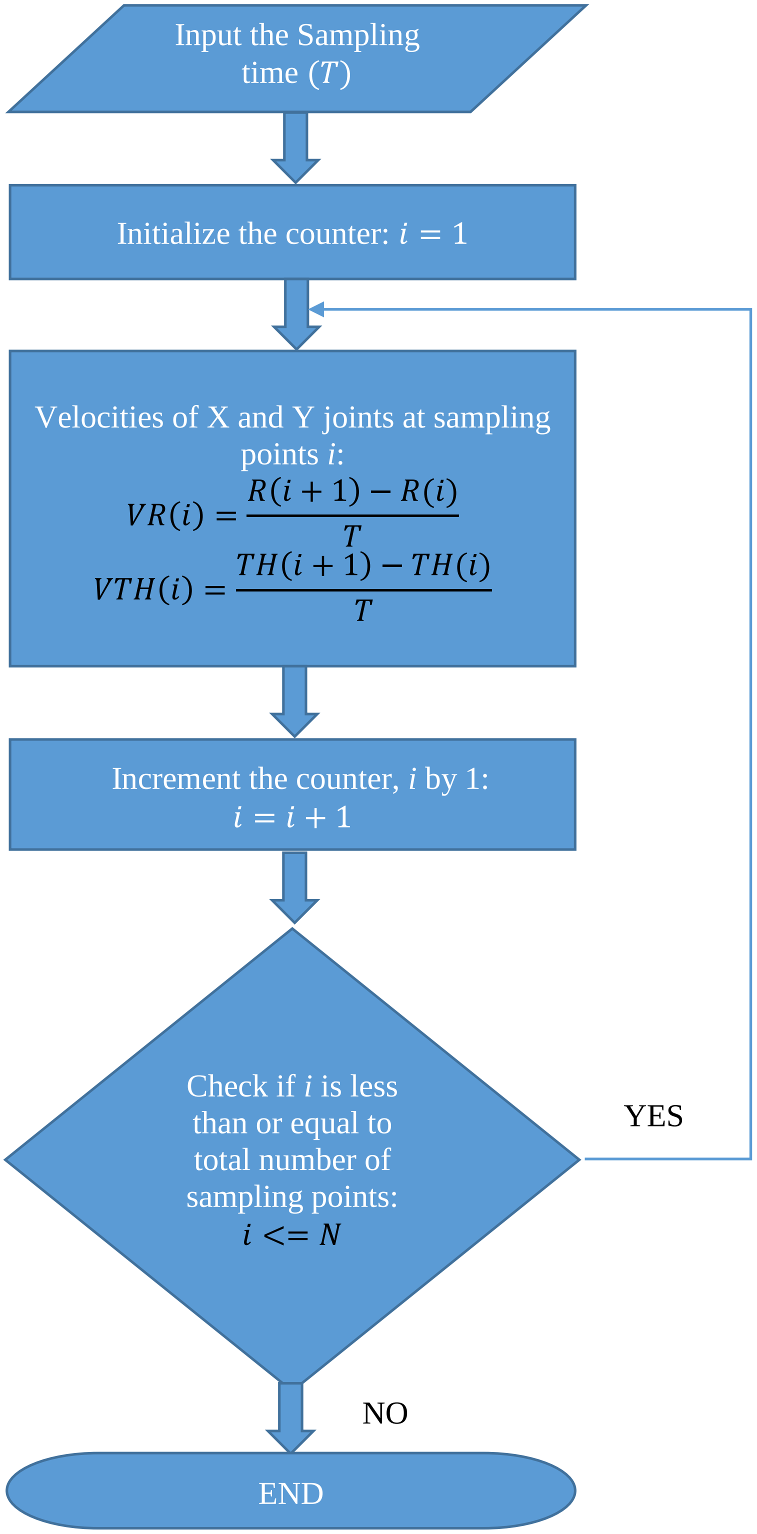
Input the Sampling time (T)
Initialize the counter: $i = 1$
Velocities of X and Y joints at sampling points i:
$VR(i) = \frac{R(i+1) - R(i)}{T}$
$VTH(i) = \frac{TH(i+1) - TH(i)}{T}$
Increment the counter, i by 1: $i = i + 1$
Check if i is less than or equal to total number of sampling points: $i <= N$
YES
NO
END

Figure 3.9: Flow-Diagram for determining joint velocities for Cylindrical kinematic arrangement at the sampling points

The process is similar to that of standard kinematic arrangement, however, the axes involved in this case are different from those in the case of the standard kinematic arrangement. The velocities of the axes are chosen according to Figure 3.8 in such a way that the endmill reaches the next sampling point in a time equal to the sampling time. The velocities of rotary and linear joints at the sampling points are stored in the arrays $VTH(i)$ and $VR(i)$ respectively.

3.4.3 Trajectory planning of Kinematically Redundant Arrangement

For the trajectory planning of kinematically redundant arrangement, as the feedrate is constant, same sampling points as determined in Section 3.4.1 are considered. However, the kinematically redundant arrangement involves more axes than that in each of the standard and cylindrical kinematic arrangements. Therefore, the joint velocities determination is not as simple as that presented in Section 3.4.2.

In kinematically redundant arrangement, both the linear joints, along X and Y axes and the cylindrical joint about Z axis are working simultaneously. This arrangement assumes:

1. The linear joints are attached to the endmill which is at the top of the system.
2. The cylindrical joint is attached to the workpiece.

Figure 3.10 shows two adjacent sampling points, and an intermediate point A.

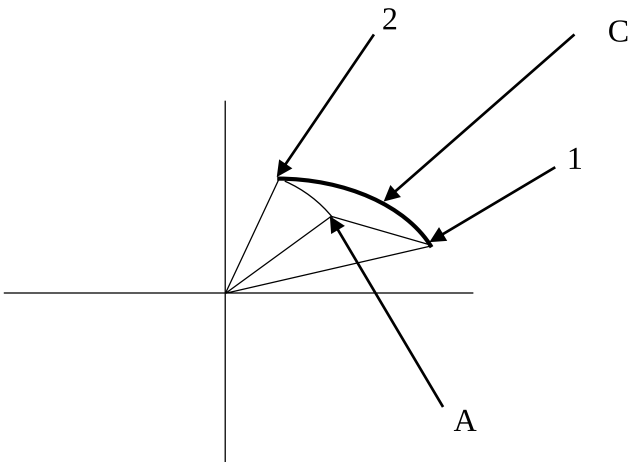

Figure 3.10: Start and End points, Desired Toolpath ‘C’ with Intermediate point A

In Figure 3.10, Point 1 is the starting point of the path, the endmill is right above point 1 initially. The desired toolpath is shown by C. This approach proposes that the endmill travels to point A from point 1 with the help of two linear joints. At the same time, the rotary joint holding the workpiece rotates point 2 such that it overlaps with point A. Note that the point A is assumed to be fixed with x-y plane, and it does not move with any of the three joints.

When the point 2 on the workpiece overlaps point A on the fixed frame, the endmill completes the path as it has reached point 2 from point 1. In this approach, point A is a free choice. It will be shown later in Chapter 4 that the shape of the toolpath depends on the choice of point A. So, this choice is used to minimize the chord error. Figure 3.11 explains joint velocities for kinematically redundant arrangement.

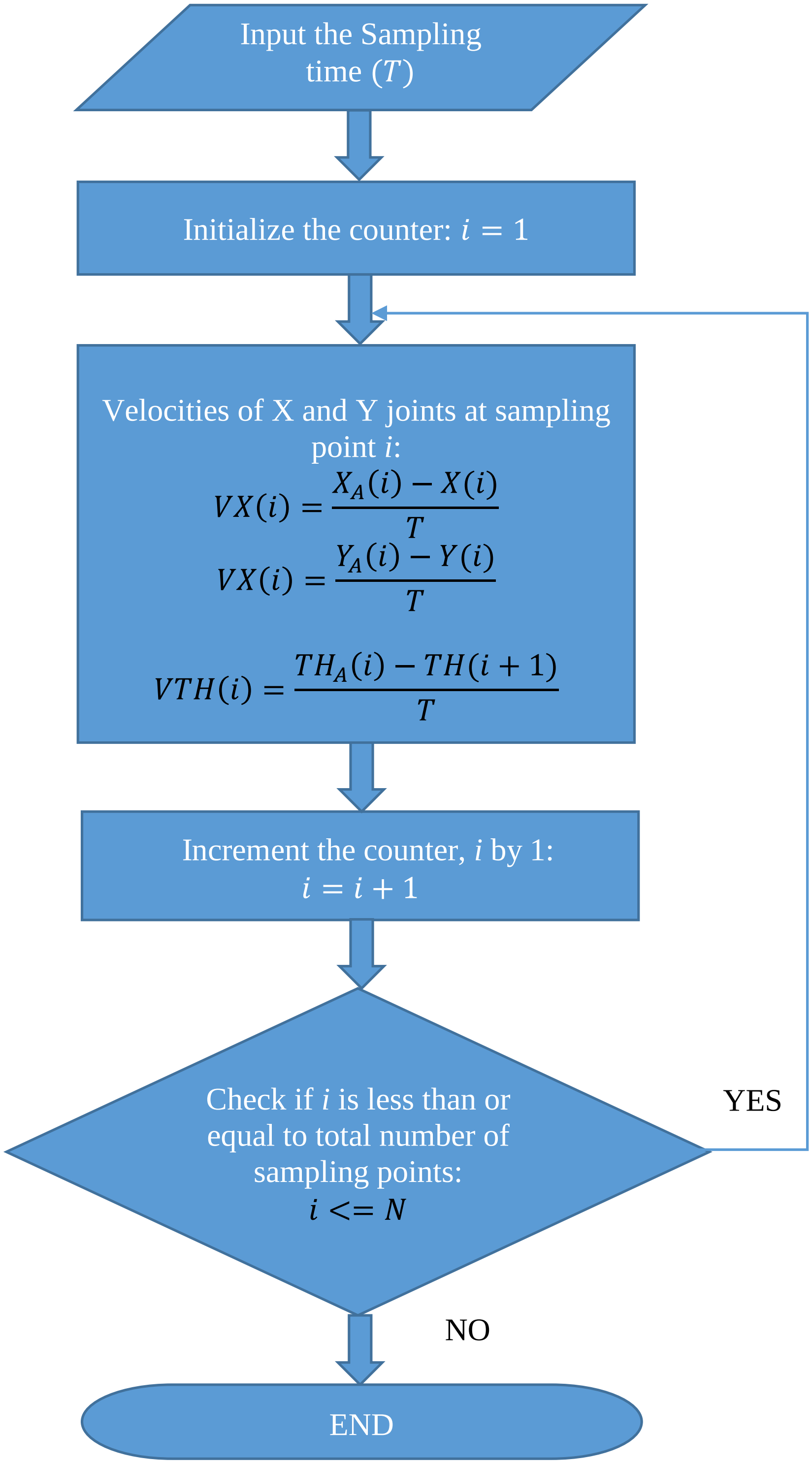
Input the Sampling time (T)
Initialize the counter: $i = 1$
Velocities of X and Y joints at sampling point i:
$VX(i) = \frac{X_A(i) - X(i)}{T}$
$VX(i) = \frac{Y_A(i) - Y(i)}{T}$
$VTH(i) = \frac{TH_A(i) - TH(i+1)}{T}$
Increment the counter, i by 1:
$i = i + 1$
Check if i is less than or equal to total number of sampling points:
$i <= N$
YES
NO
END

Figure 3.11: Flow-Diagram for determining joint velocities for Kinematically Redundant Arrangement at the sampling points

The joint velocities in a sampling interval depend on the consecutive sampling points as well as the intermediate point A. The loop of determining the joint velocities is continued until all sampling intervals are covered.

CHAPTER 4

NUMERICAL SIMULATION

In previous chapters, the analytical facet of the effects of the kinematic arrangements on the chord error was addressed. This chapter presents a numerical simulation used to evaluate the chord error for standard, cylindrical and redundant kinematic arrangements. First, the simulation setup is explained by listing all the intermediate stages. The chapter also explains the algorithm involved in the intermediate stages. After the numerical simulation setup is addressed, a detailed explanation is given on the desired toolpath and parameters selections for the numerical experiments performed. Finally, the experimental results of the evaluation of the chord errors are presented.

4.1 SIMULATION SETUP

This section presents the flow chart and algorithm to determine the actual toolpath by taking the trajectory planning results into account. Once the actual toolpath is determined, this section explains the algorithm for determining the chord error for each sampling interval. The algorithms are presented in form flow charts for better visualization. All the stages in the numerical simulation are shown in Figure 4.1.

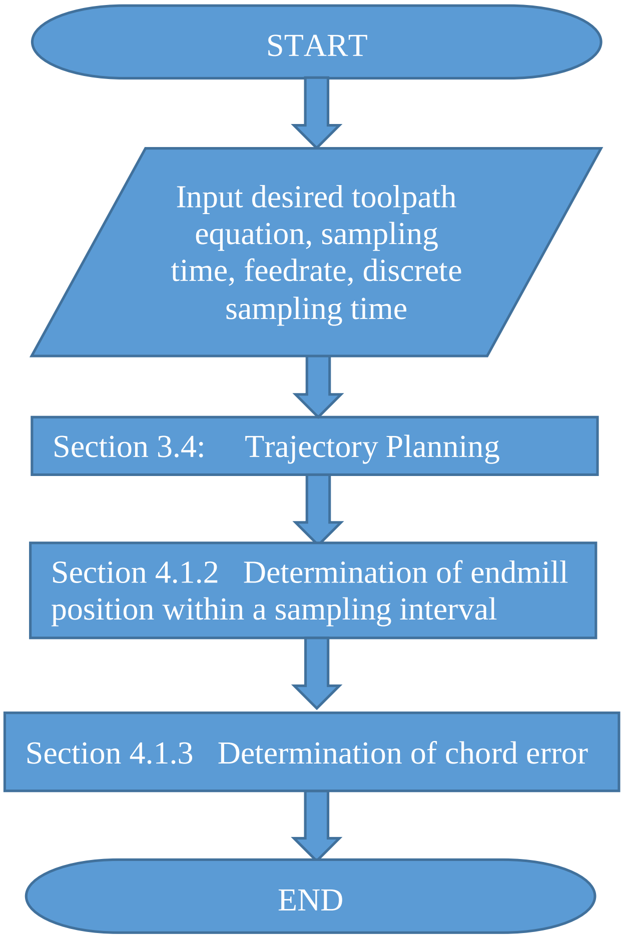
START
Input desired toolpath equation, sampling time, feedrate, discrete sampling time
Section 3.4: Trajectory Planning
Section 4.1.2 Determination of endmill position within a sampling interval
Section 4.1.3 Determination of chord error
END

Figure 4.1: All the stages in the numerical simulation

First, all the parameters including desired toolpath equation, sampling time, feedrate and discrete time are inputted. The joint velocities are calculated during trajectory planning as explained in Section 3.4. The trajectory planning is done in real-time, during the process of machining. However, for the simulation purposes, the trajectory planning is performed ahead of all other stages in the numerical simulation. The joint velocities are then used to determine the actual toolpaths within all the sampling intervals. Once the actual toolpath is determined, the chord errors are calculated for all the sampling intervals.

4.1.1 Determination of Desired Toolpath between two sampling points

Once the joint velocities are determined at the sampling points, the next task is to find out the desired and actual toolpath within two sampling points. In numerical simulation, the desired or actual toolpath need not be determined at all intermediate points, rather it is desirable to divide a sampling interval further into several sub-intervals. As one sampling interval corresponds to a sampling time T, another discrete time Td is chosen corresponding to small discrete instants. The variable Td is chosen to be fifty times smaller than the sampling time T, the reason behind this will be explained later in this chapter.

This section explains the determination of the desired toolpath between the consecutive sampling points. The desired toolpath determination process is independent of the kinematic arrangement. So, this algorithm remains same for all the arrangements.

Figure 4.2 shows the flow diagram for determining the desired toolpath between two consecutive sampling points.

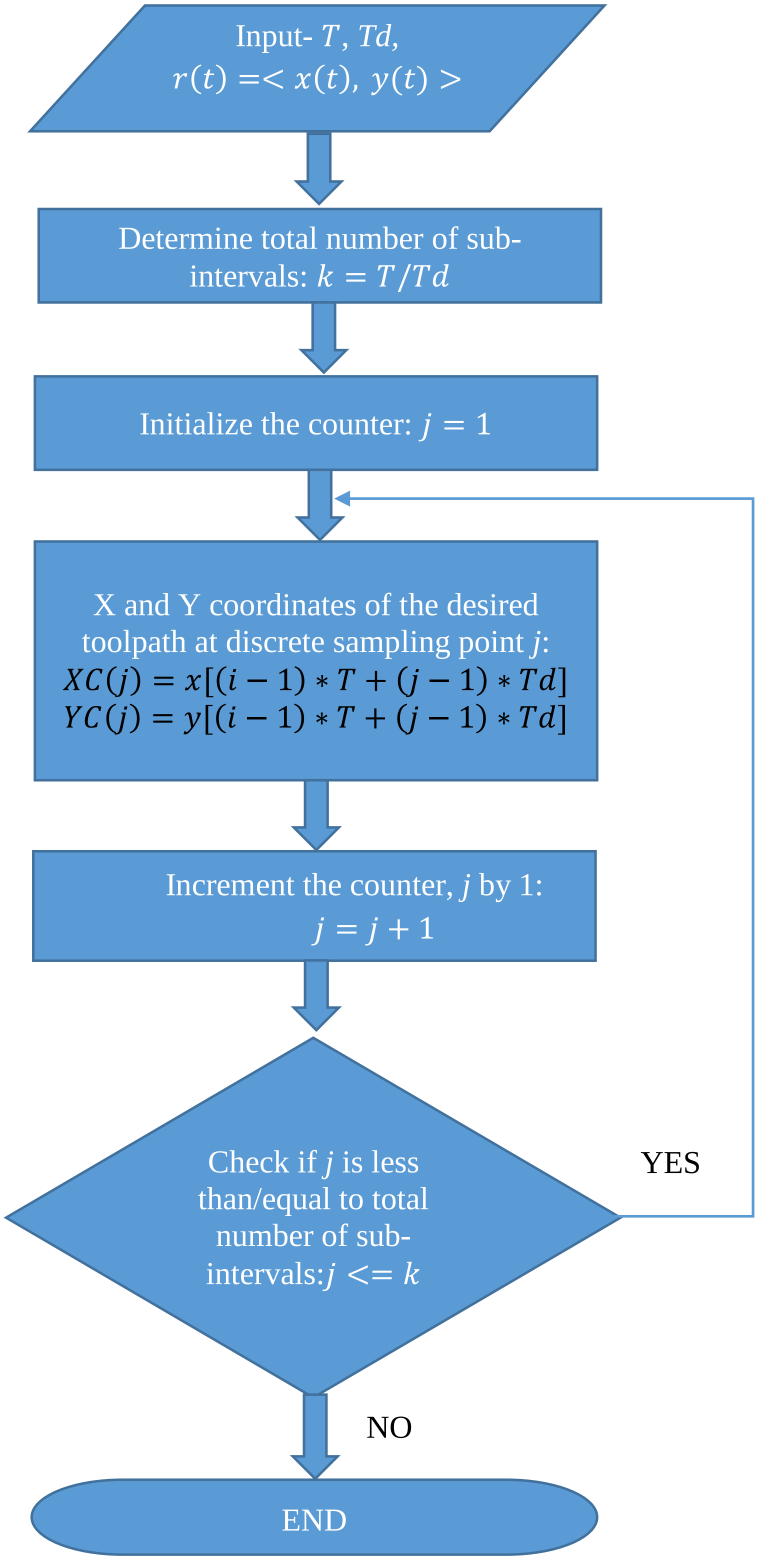
Input- T, Td,
$r(t) =< x(t), y(t) >$
Determine total number of sub-intervals: $k = T/Td$
Initialize the counter: $j = 1$
X and Y coordinates of the desired toolpath at discrete sampling point j:
$XC(j) = x[(i-1) * T + (j-1) * Td]$
$YC(j) = y[(i-1) * T + (j-1) * Td]$
Increment the counter, j by 1:
$j = j + 1$
Check if j is less than/equal to total number of sub-intervals:$j <= k$
YES
NO
END

Figure 4.2: Flow-Diagram for determining the desired toolpath between two sampling points

First, the sampling time, discrete time and the original parametric desired toolpath equation is inputted. The number of sub-intervals is then determined with the help of total simulation time and sampling time. Once the parameters at the sub-intervals are known, with the help of desired toolpath equation, the desired toolpath position can be determined at those discrete points. The coordinates of sampling points were determined in Section 4.1.1. Starting with a sampling point, the desired toolpath parameter can be increased by discrete time until it reaches next sampling point.

4.1.2 Determination of Endmill Position between two sampling points

In this section, endmill position at discrete points between two consecutive sampling points is determined. The joint velocities are held constant between two consecutive sampling points. The system can sense the desired joint velocities after every sampling time. The algorithm for determination of endmill position within a sampling interval is not same for all the three kinematic arrangements.

4.1.2.1 Determination of endmill position for Standard Kinematic Arrangement

The standard kinematic arrangement consists of all linear joints. Because all joint velocities must be constant between two consecutive sampling points, the endmill traverses a straight line between two consecutive sampling points.

Figure 4.3 shows the flow diagram for determining the endmill position at the discrete points between any two consecutive sampling points.

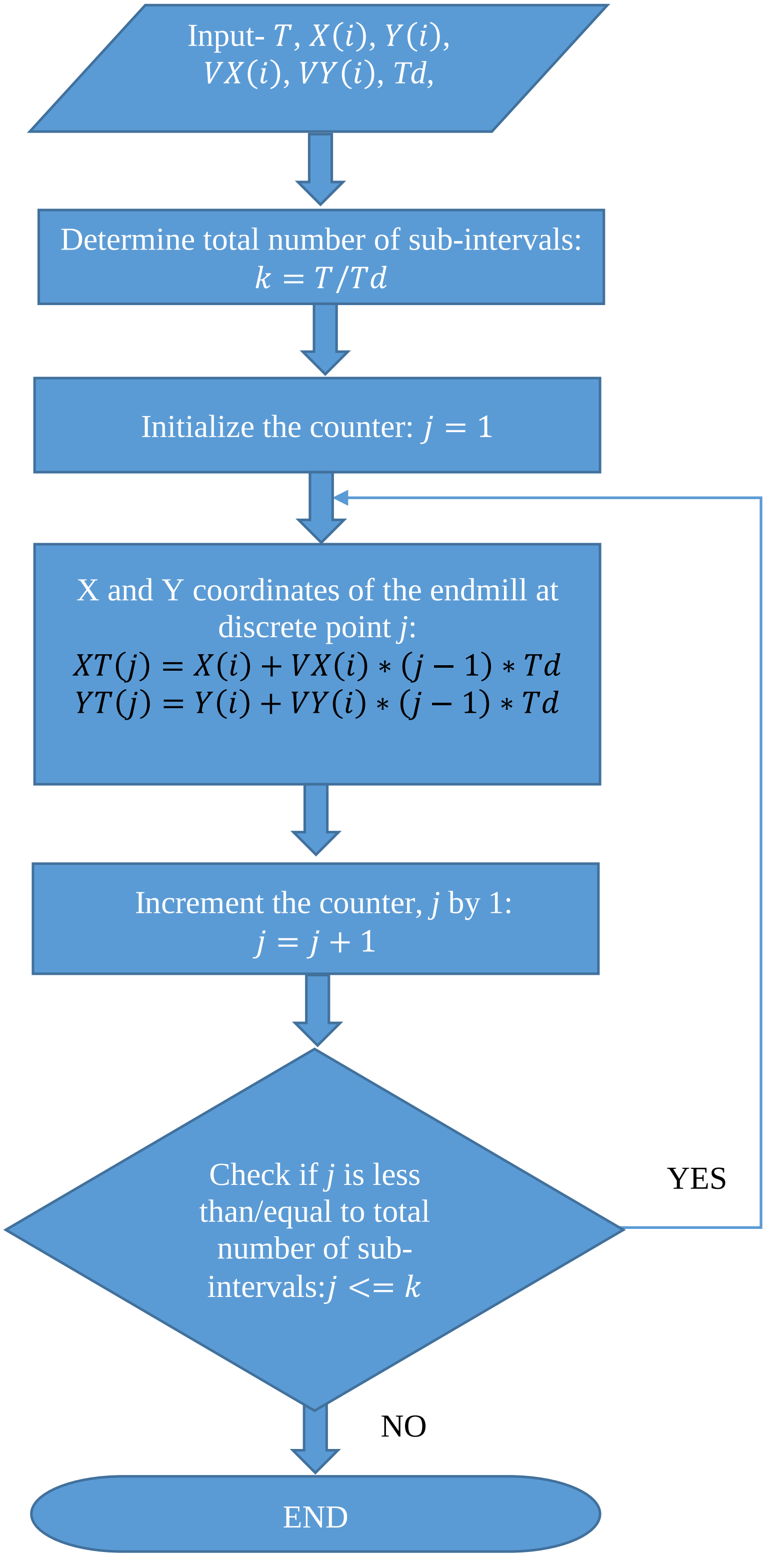
Input- T, $X(i)$, $Y(i)$, $VX(i)$, $VY(i)$, Td,
Determine total number of sub-intervals: $k = T/Td$
Initialize the counter: $j = 1$
X and Y coordinates of the endmill at discrete point j:
$XT(j) = X(i) + VX(i) * (j - 1) * Td$
$YT(j) = Y(i) + VY(i) * (j - 1) * Td$
Increment the counter, j by 1: $j = j + 1$
Check if j is less than/equal to total number of sub-intervals: $j <= k$
YES
NO
END

Figure 4.3: Flow-Diagram for determining the endmill position between two sampling points for standard kinematic arrangement

First, the number of sub-intervals with the help of total simulation time and the sampling time is determined. As the velocities of the linear joints along X and Y axes are constants, equation of motion is then applied separately to both X and Y directions to get X and Y coordinates of the discrete points. The process is repeated until X and Y coordinates of the endmill position at all discrete points are determined.

4.1.2.2 Determination of endmill position for Cylindrical Arrangement

The Cylindrical Kinematic Arrangement consists of a rotary joint and a linear joint over it. The endmill position at every discrete point is determined by finding out the cylindrical coordinates at that point. The rotary and the linear joint have constant velocities within a sampling interval. The cylindrical coordinates are updated with the help of equation of motion.

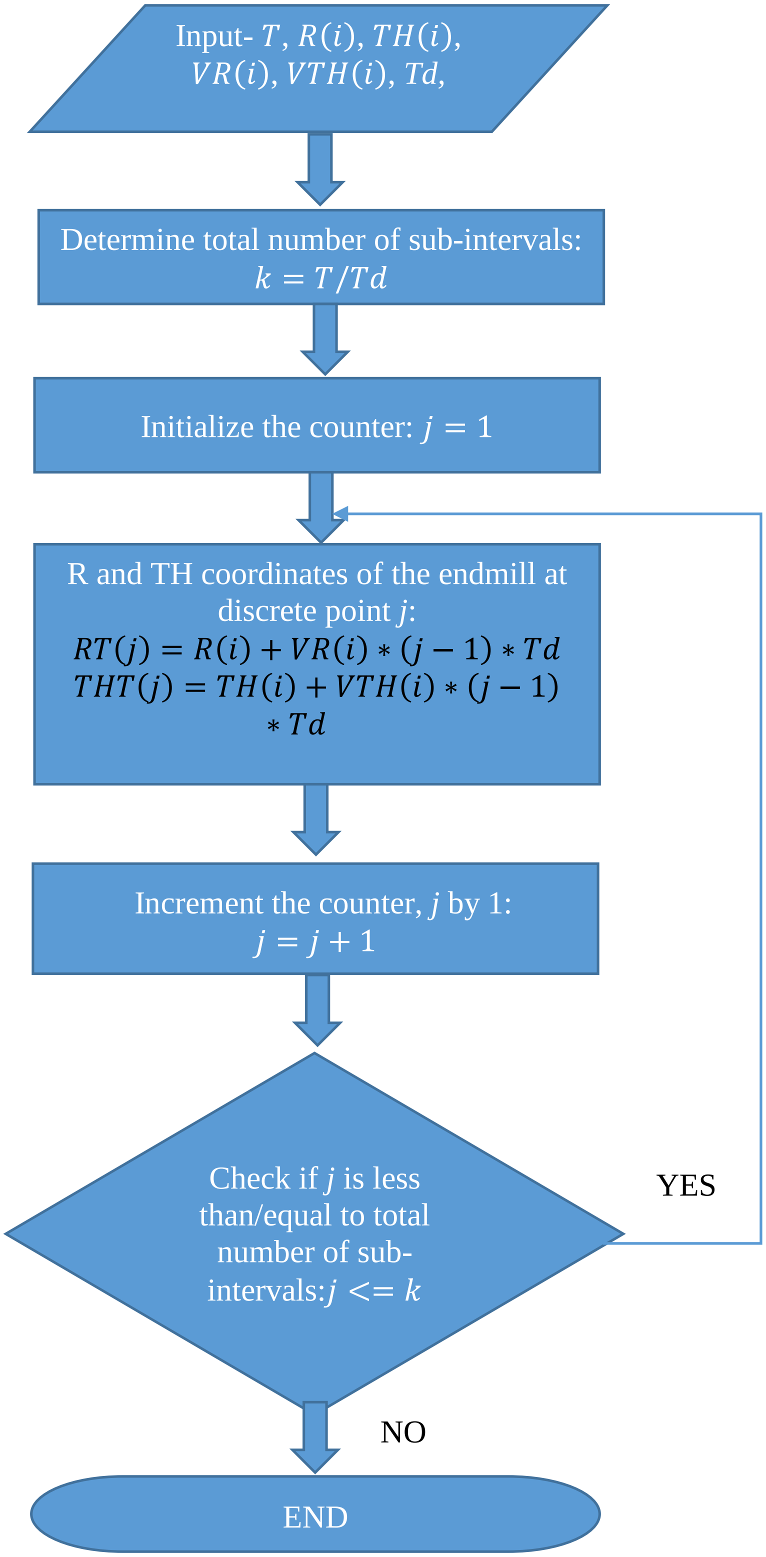
Input- T, $R(i)$, $TH(i)$,
$VR(i)$, $VTH(i)$, Td,
Determine total number of sub-intervals:
$k = T/Td$
Initialize the counter: $j = 1$
R and TH coordinates of the endmill at discrete point j:
$RT(j) = R(i) + VR(i) * (j - 1) * Td$
$THT(j) = TH(i) + VTH(i) * (j - 1) * Td$
Increment the counter, j by 1:
$j = j + 1$
Check if j is less than/equal to total number of sub-intervals:$j <= k$
YES
NO
END

Figure 4.4: Flow-Diagram for determining the endmill position between two sampling points for cylindrical kinematic arrangement

Figure 4.4 shows the flow diagram for determination of the endmill position within a sampling interval for cylindrical kinematic arrangement. First, the number of sub-intervals is determined. Equation of motion is then applied separately for the motion of both the rotary and linear joints to get R and TH coordinates of the discrete points. The process is repeated until R and TH coordinates of the endmill position at all discrete points are determined.

4.1.2.3 Determination of endmill position for Kinematically Redundant Arrangement

In this arrangement, both the endmill and workpiece are in motion. So, to obtain the toolpath on workpiece, relative motion has to be addressed. For the purpose of obtaining the relative motion in numerical approach, the rotary joint is kept motionless, and only two linear joints attached the endmill are moved. At each discrete point, the toolpath is shifted by rotating it by -θ as if the rotary joint is rotated by θ.

This is illustrated in Figure 4.5 which shows the flow diagram for determining the endmill position between two sampling points for the kinematically redundant arrangement.

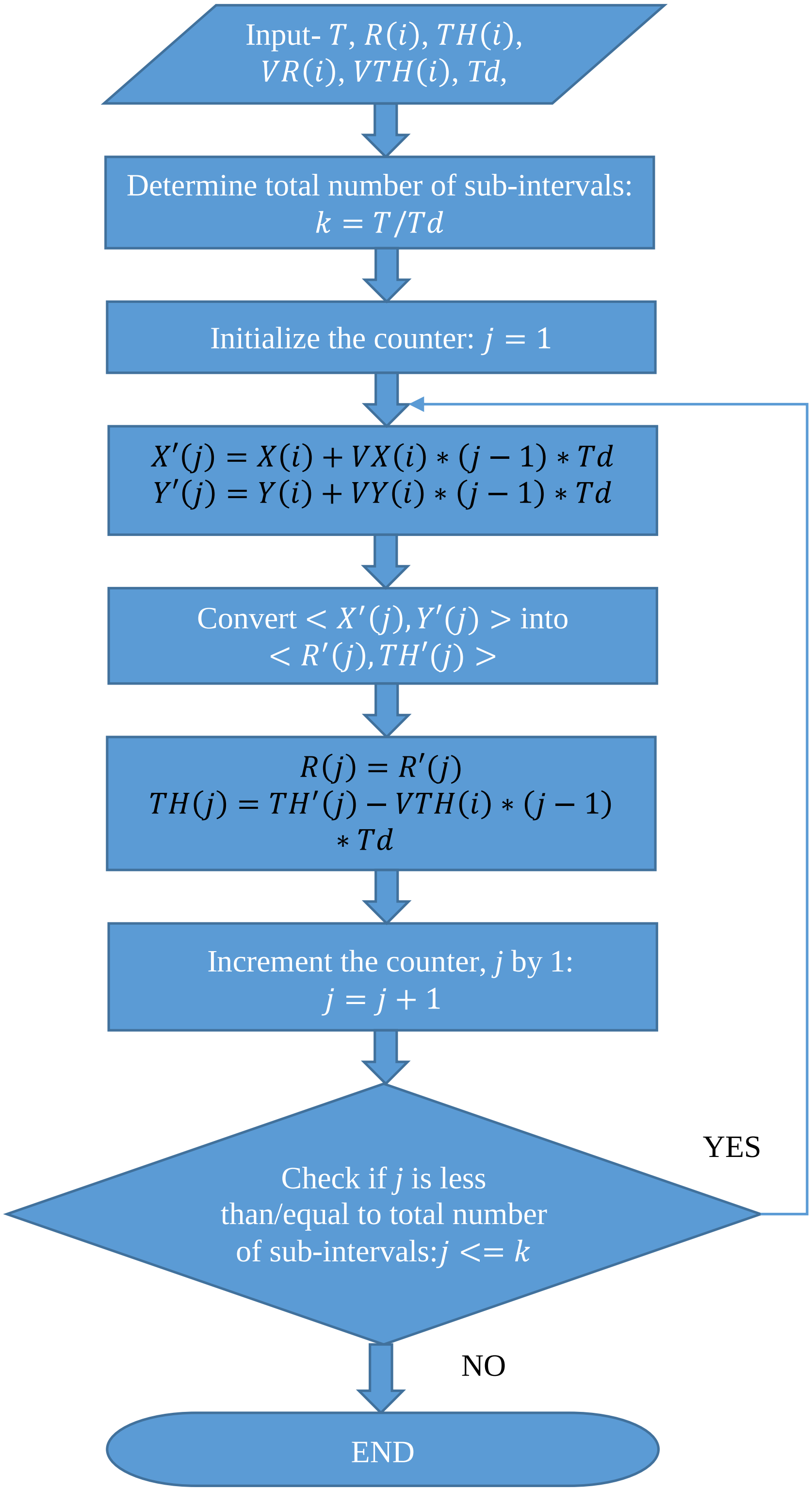
Input- T, $R(i)$, $TH(i)$,
$VR(i)$, $VTH(i)$, Td,
Determine total number of sub-intervals:
$k = T/Td$
Initialize the counter: $j = 1$
$X'(j) = X(i) + VX(i) * (j - 1) * Td$
$Y'(j) = Y(i) + VY(i) * (j - 1) * Td$
Convert $< X'(j), Y'(j) >$ into
$< R'(j), TH'(j) >$
$R(j) = R'(j)$
$TH(j) = TH'(j) - VTH(i) * (j - 1)$
$* Td$
Increment the counter, j by 1:
$j = j + 1$
Check if j is less
than/equal to total number
of sub-intervals:$j <= k$
YES
NO
END

Figure 4.5: Flow-Diagram for determining the endmill position between two sampling points for kinematically redundant arrangement

4.1.3 Determination of chord error

This section presents the algorithm for determining the chord error as defined in Chapter 3. The chord error depends on the desired toolpath and endmill position between two consecutive sampling points. So, this algorithm is unique for all the three kinematic arrangements.

The goal of this algorithm is to find the length of the longest perpendicular distance between the desired and actual toolpaths between two sampling points. Figure 4.6 shows the desired and actual toolpaths drawn between two sampling points: *s*=1 and *s*=2.

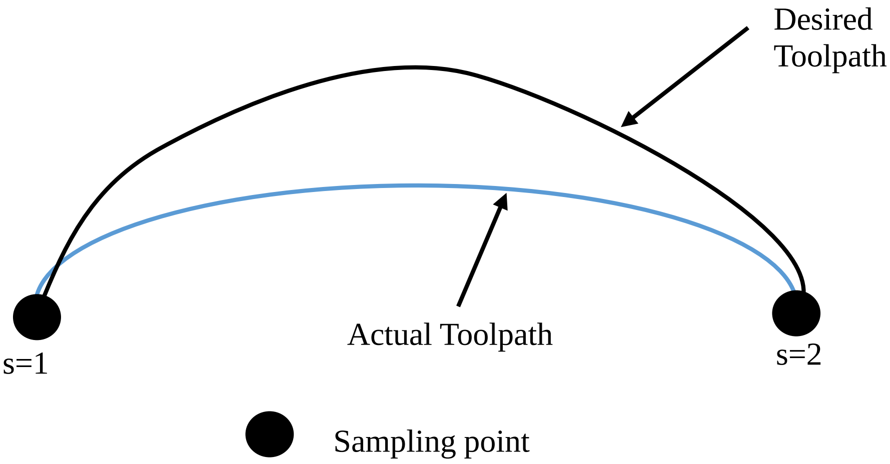

Figure 4.6: Desired and actual toolpaths between two chosen sampling points

Now, the algorithm is developed to find the chord error, and illustrative plots for the above case are presented. Figure 4.7 shows the flow diagram for the first step in determining the chord error.

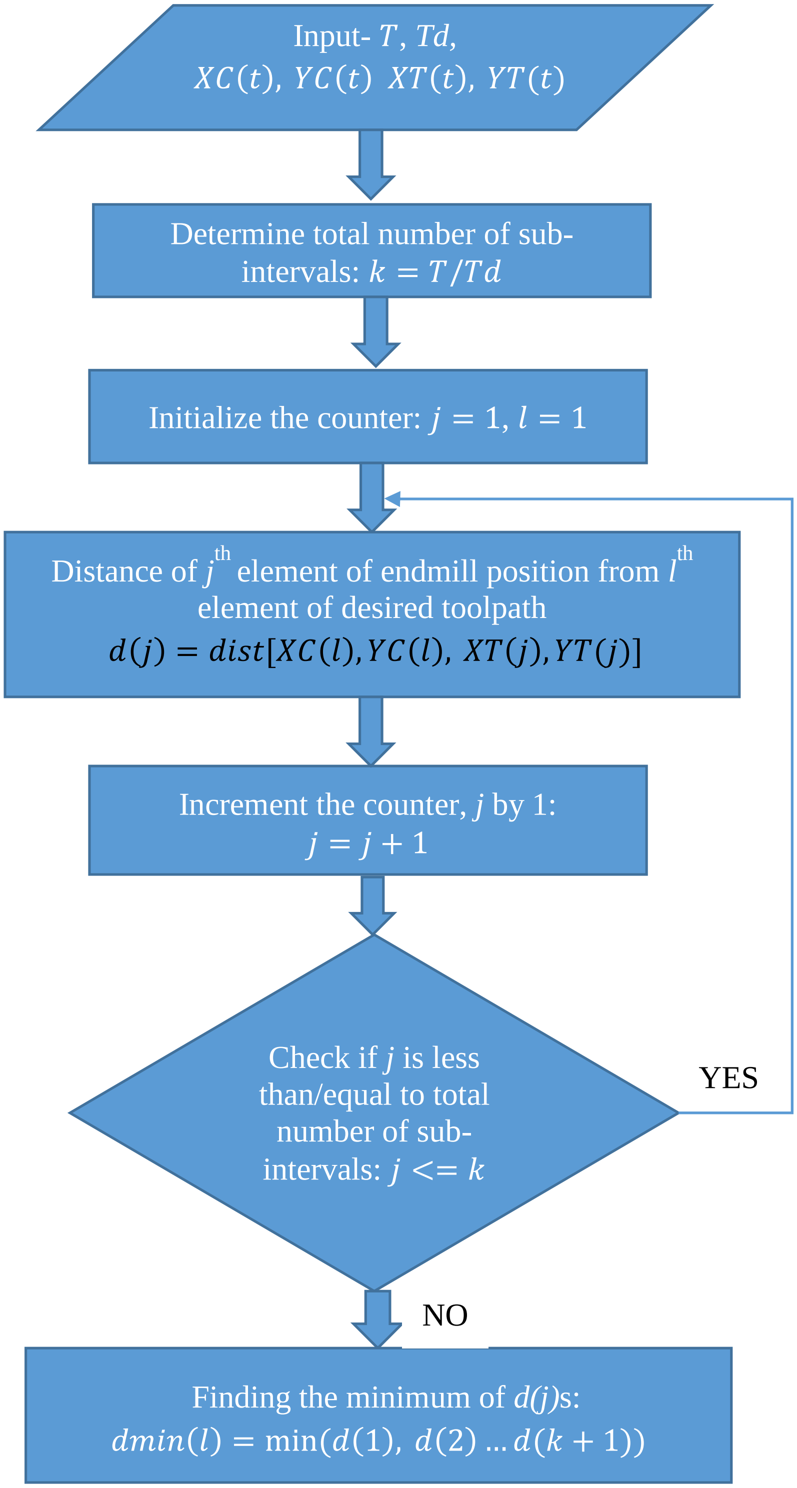
Input- T, Td,
$XC(t)$, $YC(t)$ $XT(t)$, $YT(t)$
Determine total number of sub-intervals: $k = T/Td$
Initialize the counter: $j = 1$, $l = 1$
Distance of j^{th} element of endmill position from l^{th} element of desired toolpath
$d(j) = dist[XC(l), YC(l), XT(j), YT(j)]$
Increment the counter, j by 1:
$j = j + 1$
Check if j is less than/equal to total number of sub-intervals: $j <= k$
YES
NO
Finding the minimum of $d(j)$s:
$dmin(l) = \min(d(1), d(2) \dots d(k + 1))$

Figure 4.7: Flow-Diagram for determining the length of the chord perpendicular to the actual toolpath for every discrete point

The algorithm shown in Figure 4.7 determines the length of the chord perpendicular to the actual toolpath for the first discrete point. First, the sampling time, discrete sampling time, the array of desired toolpath position at all the discrete points are inputted. The number of sub-intervals is then calculated with the help of total simulation time and the sampling time. The only point on desired toolpath considered in this stage of algorithm is the one that corresponds to the first discrete point. Distance between this point and all the discrete points on actual toolpath are calculated. The chord that corresponds to the shortest distance is chosen for further analysis. This algorithm is then repeated for all discrete points on the desired toolpath.

Figure 4.8 shows the desired and actual toolpaths discretized according to the discrete sampling time, *Td*. This discrete sampling time is chosen such that the line segment corresponding to shortest distance out of the distances between any discrete point on the desired toolpath to all the discrete points on the actual toolpath, makes an angle of 90±1 degree with the actual toolpath.

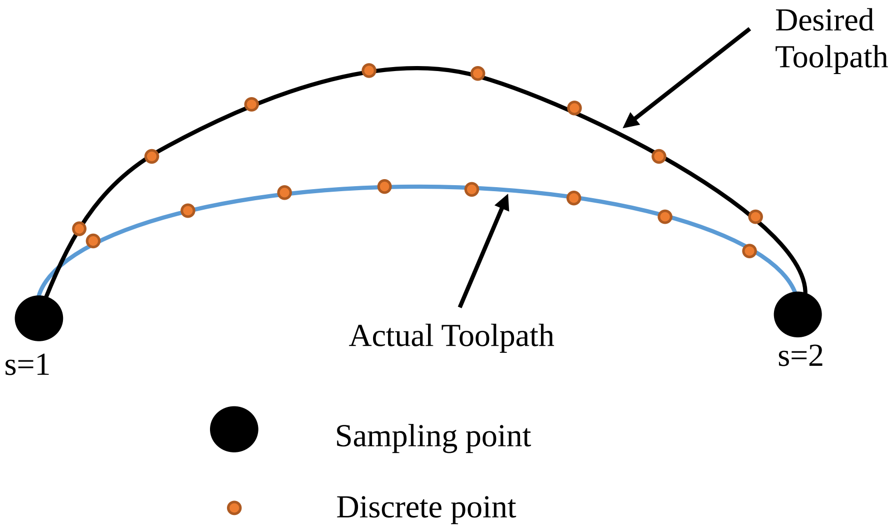

Figure 4.8: Discretization of desired and actual toolpaths

Figure 4.9 shows the segments joining a discrete point on the desired toolpath with all the discrete point on the actual toolpath

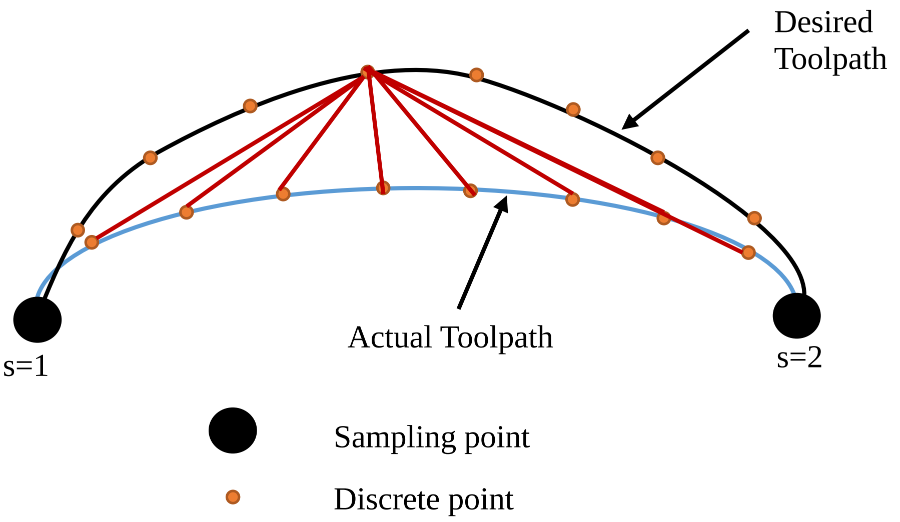

Figure 4.9: Segments joining a discrete point on the desired toolpath with all the discrete points on the actual toolpath

Figure 4.10 shows the shortest segments out of the segments shown in the Figure 4.9.

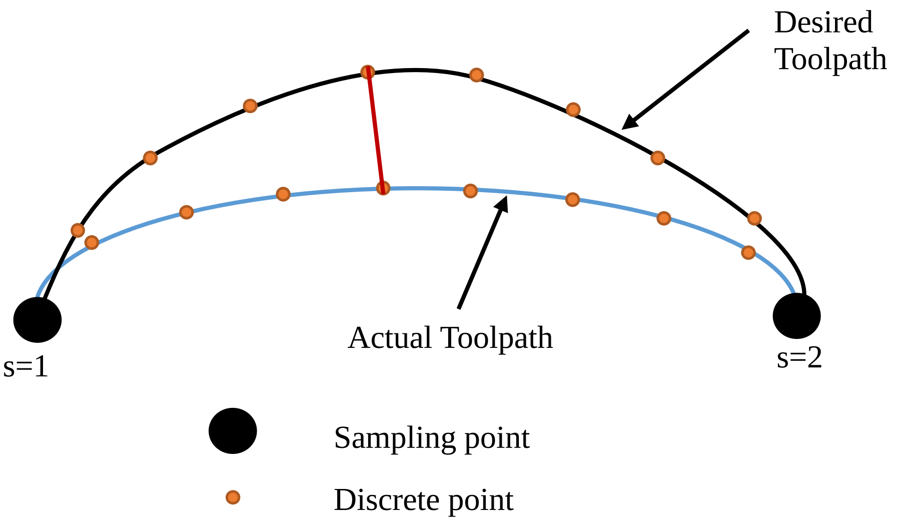

Figure 4.10: Shortest of the segments joining a discrete point on the desired toolpath with all the discrete points on the actual toolpath

The analysis is repeated to find the shortest segments for all the discrete points on the desired toolpath, which is shown in the Figure 4.11.

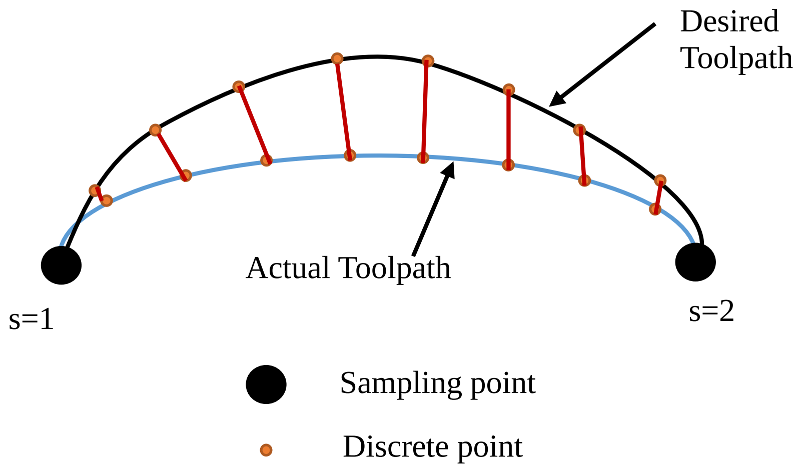

Figure 4.11: Shortest segments for all the discrete points on the desired toolpath

According to geometry, the shortest line segment drawn from a point to a curve is perpendicular to that curve. As the segments shown in the Figure 4.11 are drawn from each discrete point on the desired toolpath, those are perpendicular to the actual toolpath. This complies with the fact that the chord has to be perpendicular to the actual toolpath.

Figure 4.12 shows the flow diagram for second and last part of algorithm for determining the chord error by finding the segment that corresponds to the maximum length.

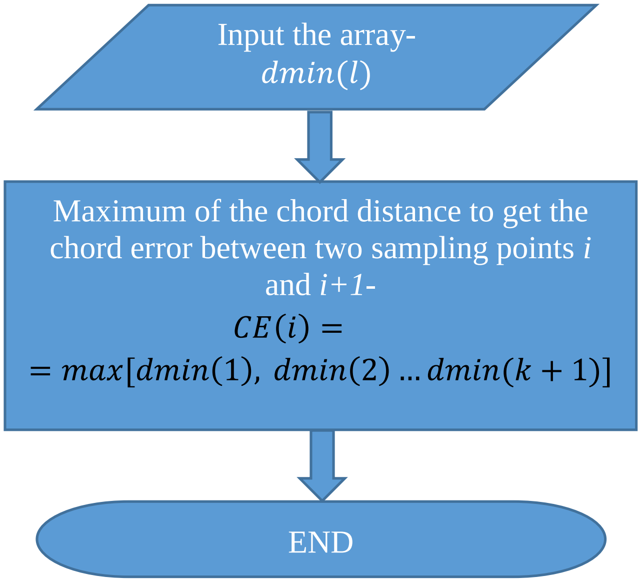
Input the array-
$dmin(l)$
Maximum of the chord distance to get the chord error between two sampling points *i* and *i+1*-
$CE(i) =$
$= max[dmin(1), dmin(2) \ldots dmin(k+1)]$
END

Figure 4.12: Flow-Diagram for determining the chord error by finding maximum of the chord lengths

First, the array of chord error segments for discrete points which was generated from the algorithm shown in Figure 4.7 is inputted. Then the element corresponding to largest chord error segment length is selected. The largest of the distances shown in Figure 4.11 corresponds to the chord error between these two toolpaths. This chord error is shown in Figure 4.13.

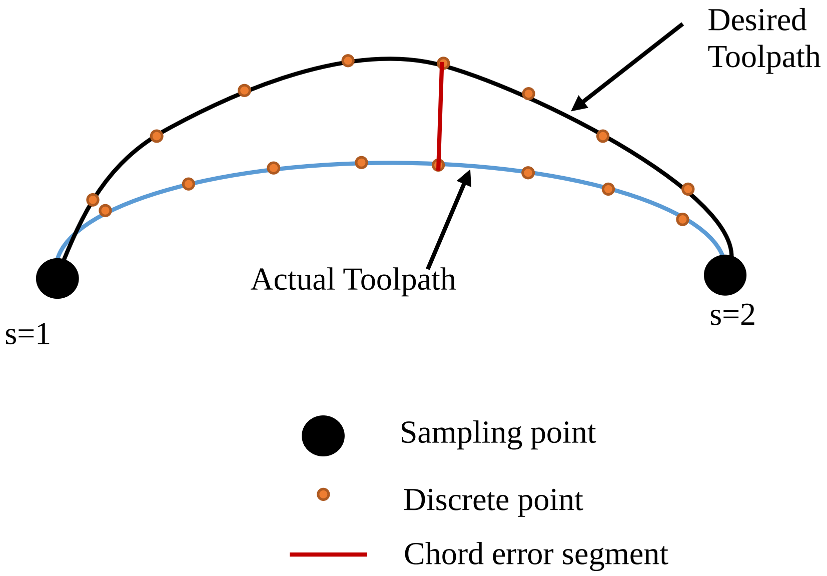

Figure 4.13: Chord error segment

4.1.4 Evaluating chord error for the kinematically redundant arrangement

This section discusses the effect of kinematically redundant arrangement on the chord error. First of all, two sampling points are arbitrarily taken into account. A trajectory is then generated between these two sampling points. This trajectory is treated as a desired toolpath. A limit on the allowable chord error is denoted by variable δ. So, the task is to trace this desired toolpath such that the chord error is less than the limit δ.

An initial guess is made on the actual toolpath. Depending upon the chord error for this guess, more accurate toolpaths are generated with successive iterations. Figure 4.14 shows an Archimedean spiral which is to be traced using the kinematically redundant arrangement. Figure 4.15 shows the desired toolpath along with the initial guess of actual toolpath. Figure 4.16 includes the actual toolpaths after next two iterations. Finally, Figure 4.17 shows the trend of chord error with number of iterations.

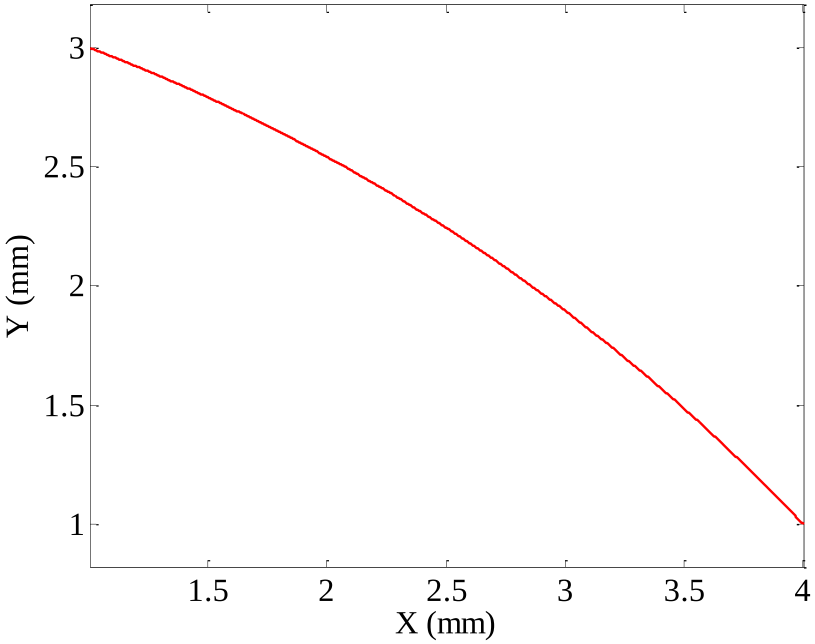

Figure 4.14: Desired toolpath

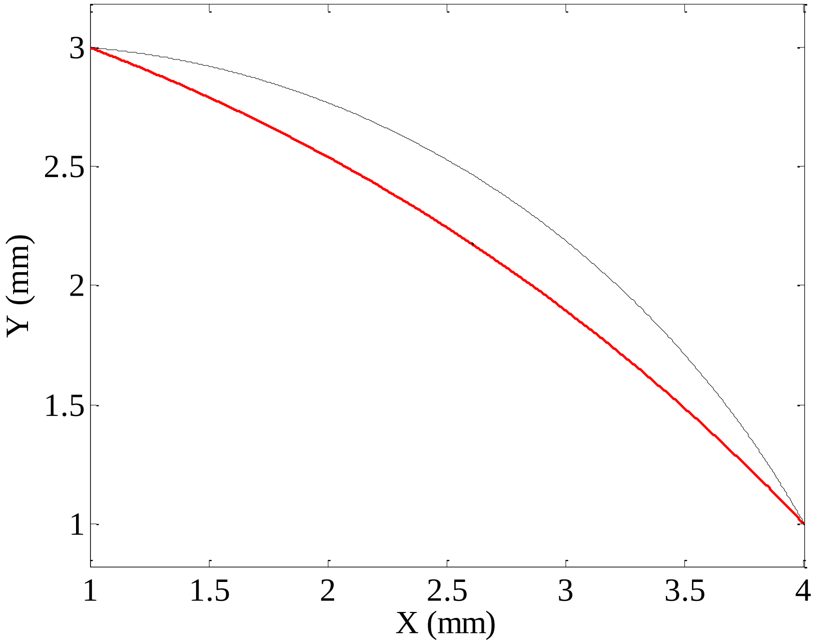

Figure 4.15: Desired toolpath with initial guess

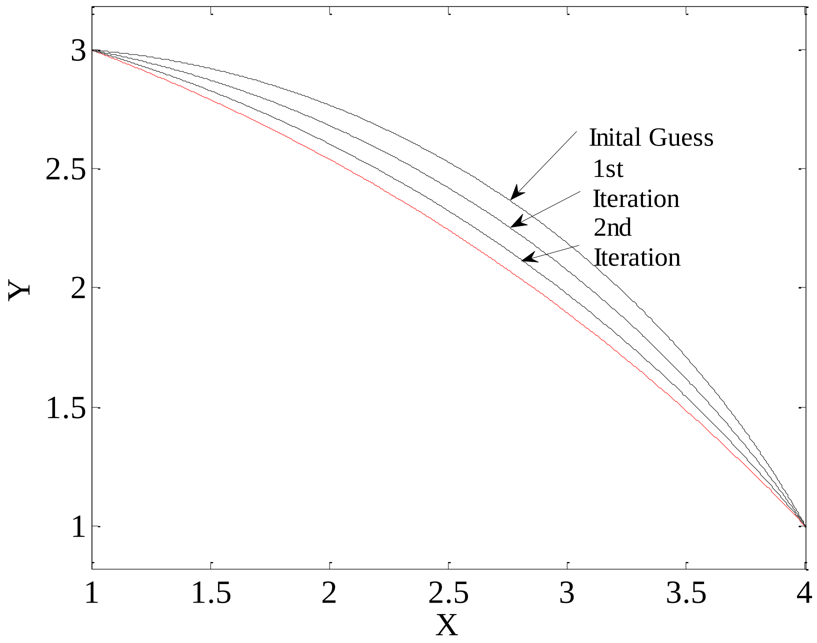

Figure 4.16: Subsequent iterations of possible trajectories with initial guess

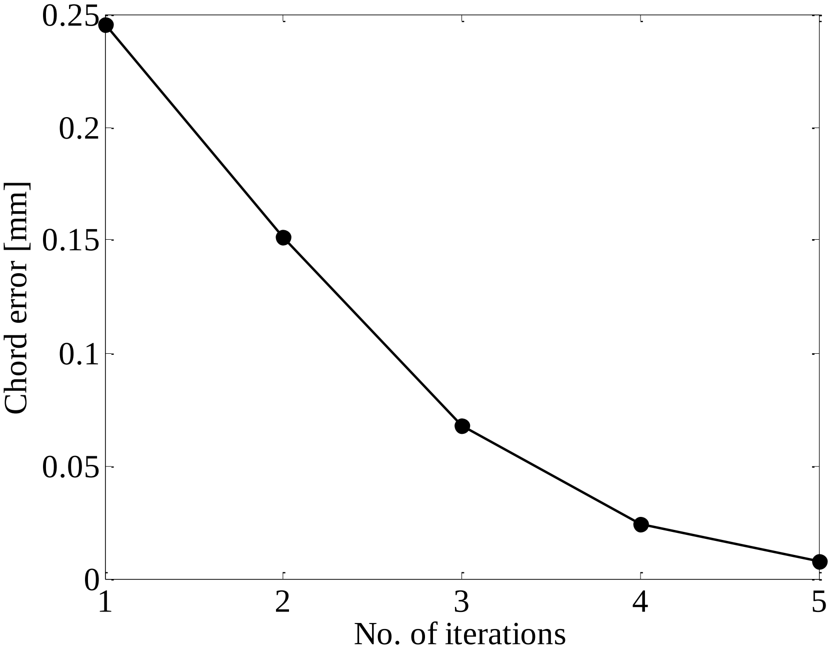

Figure 4.17: Chord error vs number of iterations

.

The analysis for evaluating the chord error for kinematically redundant kinematic arrangement presented the effect of redundant kinematic arrangement on the actual toolpath and the chord error. According to results shown in Figure 4.16, the actual toolpath can be optimized to approach the desired toolpath such that the chord error goes on decreasing. This optimization was obtained through two iterations. Figure 4.17 showed that how the chord error decreased with number of iterations for an Archimedean spiral. The process of iterations can be continued until the chord error falls below the maximum allowable chord error for micromilling.

4.2 EXPERIMENTAL DESIGN

This section presents the experimental design for the numerical simulations performed to analyze the effect of kinematic arrangements on chord error. Section 4.2.1 explains how the desired toolpaths were selected for the numerical simulation experiments. Sections 4.2.2 and 4.2.3 present the design of experiments for comparison of the standard kinematic arrangement with the cylindrical kinematic arrangement and the kinematically redundant arrangement, respectively.

4.2.1 Selection of desired toolpaths

While analyzing the differences in the chord errors for standard and cylindrical kinematic arrangement, the selection of desired toolpath is important to consider as many curve features as possible. The desired toolpaths considered in this analysis include- a line, circular arc, circle involute, logarithmic spiral, a power function and Archimedean spiral. A line is considered as it is expected to show zero error for standard kinematic arrangement. Similarly, a circular arc is expected to show zero error for the cylindrical kinematic arrangement. A circle involute is not expected to show zero error in either of these arrangements. The unique feature about the logarithmic spiral is that its radius of curvature increases linearly along the arc length. A power function is chosen such that only rotary joint velocity is constant along the arc length. It is used to observe the behavior of the chord error along the arc length for these two arrangements. Finally, an Archimedean spiral is chosen as it has the special characteristics of being a locus of points whose r coordinate linearly varies with its θ coordinate. It is an interesting characteristic because the actual toolpath generated by a cylindrical kinematic arrangement within a sampling interval is always an Archimedean spiral.

4.2.2 Parameter Selection

In these experiments, the independent variable is the kinematic arrangement of the micromilling machine, and the dependent variable is the chord error. The variable parameter is the desired toolpath. One of the parameters that is kept constant in this analysis is the sampling time. Sampling time depends upon the computer and the hardware used during the processing, and so it is kept because it is not related to the independent variable; the same sampling time can be achieved on a micromilling machine regardless of the kinematic arrangement.

The second parameter that is kept constant in this analysis is the feedrate. If the feedrate is varied, the distance between two consecutive sampling points is also varied. Figure 4.18 illustrates the effect of feedrate on the chord error in case of standard kinematic arrangement.

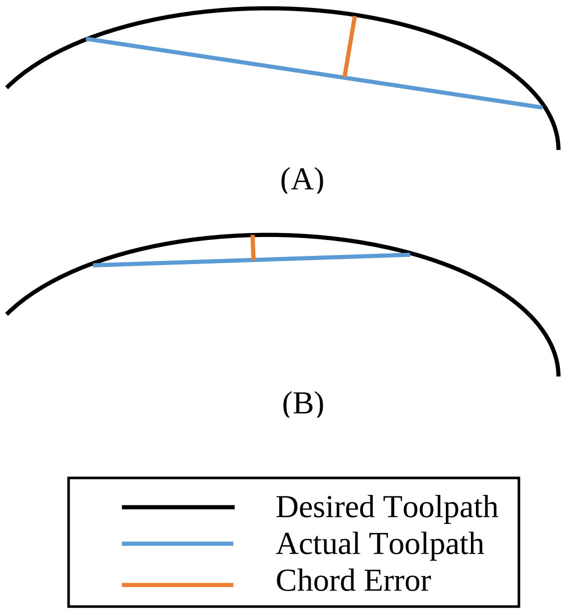

Figure 4.18: Illustration of effect of (A) faster feedrate and (B) slower feedrate, on the chord error

In Figure 4.18 (A) and (B), curves in black represent the desired toolpath, blue lines represent the actual toolpath between two sampling points. The intersection of actual toolpath with the desired toolpath corresponds to two sampling points. Therefore, the blue lines correspond to the length that the endmill travels relative to the workpiece in one sampling time. Equation 4.1 gives the relation between length of the blue segment and the feedrate.

$$feedrate = \frac{Chord\ Length}{sampling\ time} \tag{4.1}$$

The red line segments in Figure 4.18 are the chord errors for the corresponding cases. With the help of Figure 4.19 and Equation 4.1, it can be shown that the chord error depends upon the feedrate.

Decreased feedrate decreases the production rate. In addition, changing the feedrate causes uneven chip thickness, and hence it disturbs machining stability. Therefore, changing the feedrate during machining is not preferred. Second parameter that is kept constant in this analysis is therefore the feedrate. Keeping the feedrate constant for all of the desired toolpaths analyzed, the sampling points are determined. In the case of the standard kinematic arrangement, the feedrate is calculated to equate to the velocities of the linear joints along X and Y axes according to Equation 4.2. The variables $\dot{x}$ and $\dot{y}$ are the velocities of linear joints along x and y axes respectively.

$$feedrate = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{4.2}$$

4.2.2.1 Parameters for Experiment 1: Comparison of Cartesian and Cylindrical Kinematic Arrangements

Chord errors for several desired toolpaths using both standard and cylindrical kinematic arrangements are compared. Table 4.1 summarizes the parameters with their values used for the numerical experimentation.

Table 4.1: Parameters and their values for standard and cylindrical kinematic arrangement

Parameters	**Standard Kinematic Arrangement**	**Cylindrical Kinematic Arrangement**
Number of joints working at a time	2	2
Involved joints	Linear joints along X and Y axes	Linear joint along radial axis and rotary joint about Z axis
Center of the joints	(0,0)	(0,0)
Sampling time	5ms	5ms
Feedrate	10 mm/s	10 mm/s

4.2.2.2 Parameters for Experiment 2: Comparison of Cartesian, Cylindrical and Kinematically Redundant Kinematic Arrangements

For this analysis as well, sampling time and feedrate are kept constant. Therefore, the same sampling points which were determined according to Section 4.2.2 are used. Table 4.2 lists the parameters and their values used for the kinematically redundant arrangement.

Table 4.2: Parameters and their values for Kinematically Redundant Arrangement

Parameters	**Standard Kinematic Arrangement**	**Cylindrical Kinematic Arrangement**	**Kinematically Redundant Arrangement**
Number of joints working at a time	2	2	3
Involved joints	Linear joints along X and Y axes	Linear joint along radial axis and rotary joint about Z axis	Linear joints along X and Y axes, and rotary joint about Z axis
Center of the joints	(0,0)	(0,0)	(0,0)
Sampling time	5ms	5ms	5ms
Feedrate	10 mm/s	10 mm/s	10 mm/s

Table 4.3 lists the limit values that will be imposed on the chord error while minimizing it within sampling intervals for the desired toolpaths considered. These limiting values are the maximum allowable limit of the chord error. The numerical simulation for minimizing the chord error stops once the chord error incurred falls below the limiting value. These limiting values on the chord error were determined such that the chord error minimizing simulation does not take more than one minute per one sampling interval on a basic configured computer system.

Table 4.3: Imposed chord error limit for different desired toolpaths

Desired Toolpath	**Limits on the Chord Error [mm]**
A Straight line	$2.5\text{x}10^{-5}$
Circular Arc	$1\text{x}10^{-5}$
Circular Involute	$5\text{x}10^{-5}$
Logarithmic Spiral	$1\text{x}10^{-4}$
Power Function	$5\text{x}10^{-5}$

4.3 EXPERIMENTAL RESULTS

This Section presents the experimental results based upon the set up explained in Section 4.2. First, comparison of chord errors for standard and cylindrical kinematic arrangements is presented for different desired toolpaths. Effect of redundant kinematic arrangement on chord error is then discussed. Finally, the chord error for kinematically redundant arrangement is compared with that of standard and cylindrical kinematic arrangements.

4.3.1 Result of Experiment 1: Comparison of chord errors for Standard and Cylindrical Kinematic Arrangement

This section presents several desired toolpaths with different characteristics that are used to present the difference between the chord errors for standard and cylindrical kinematic arrangements. A few parameters as explained in Section 4.2 are kept constant for more precise analysis. All linear joints involved in this analysis are made to pass through the origin of the system. The rotary axis is also made to rotate about the Z axis.

The sampling time is kept constant to the value of 5ms. The feedrate is also set to the value of 10mm/s.

4.3.1.1 Line

A linear desired toolpath is important to evaluate as the standard kinematic arrangement is expected show zero chord error for this type of desired toolpaths. The parametric equation of the line used for this analysis is given by Equation 4.3. This linear segment is shown in Figure 4.16.

$$\begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \begin{bmatrix} u \\ 1 \end{bmatrix} \tag{4.3}$$

Figure 4.20 shows the velocities of the joints versus the arc length if this desired toolpath is traced with the help of standard kinematic arrangement. Similarly, Figure 4.21 shows the velocities of the joints vs arc-length for the case of cylindrical kinematic arrangement. Finally, Figure 4.22 shows the chord error plotted against the arc length for both the kinematic arrangements.

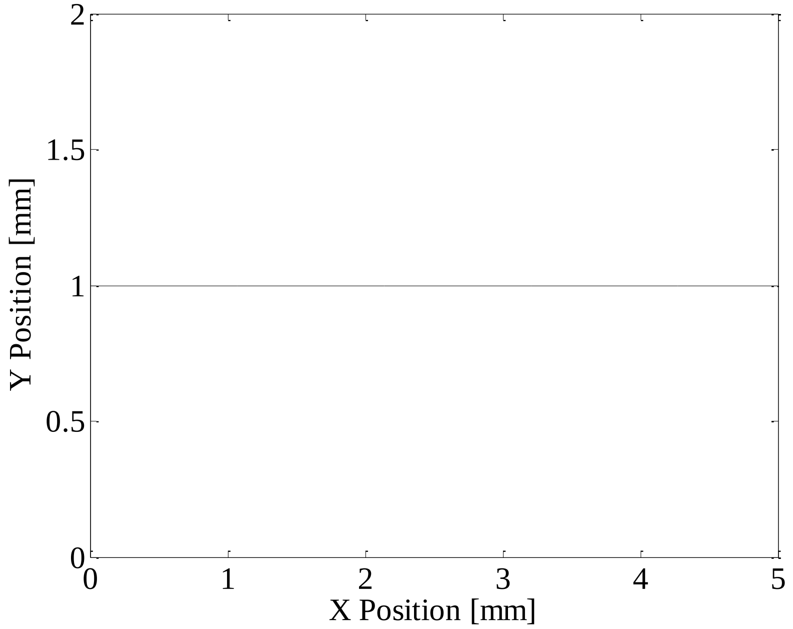

Figure 4.19: A linear segment

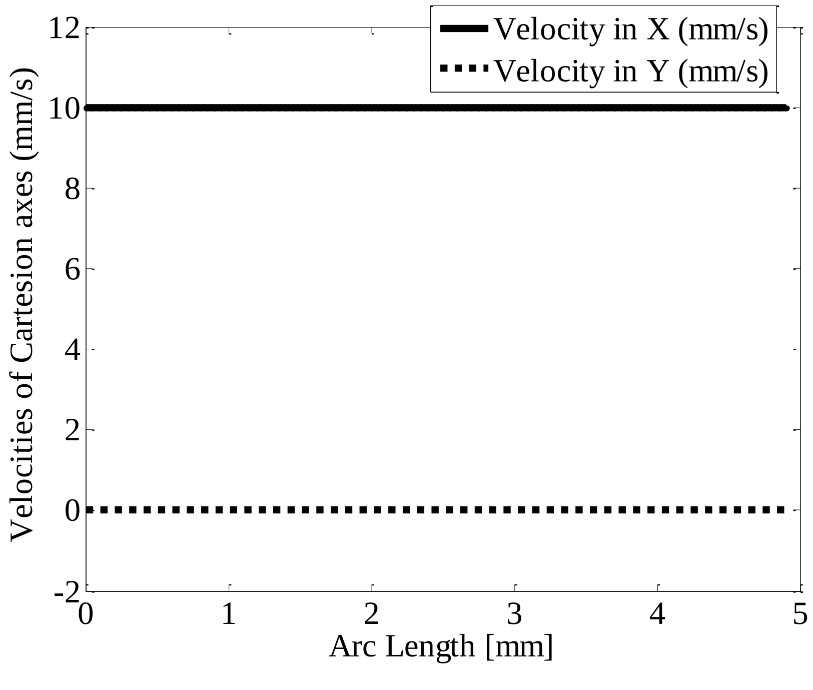

Figure 4.20: The standard arrangement joint velocities

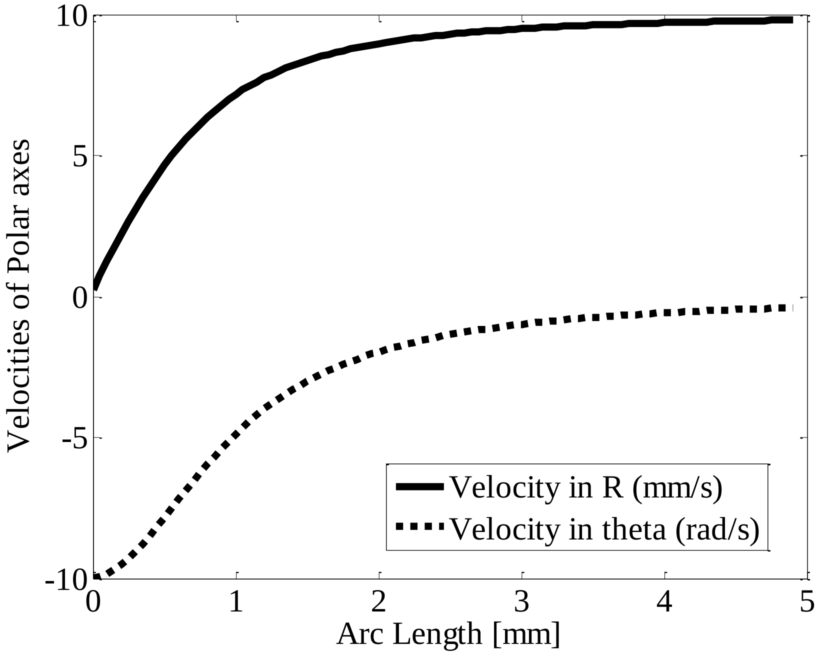

Figure 4.21: The cylindrical arrangement joint velocities

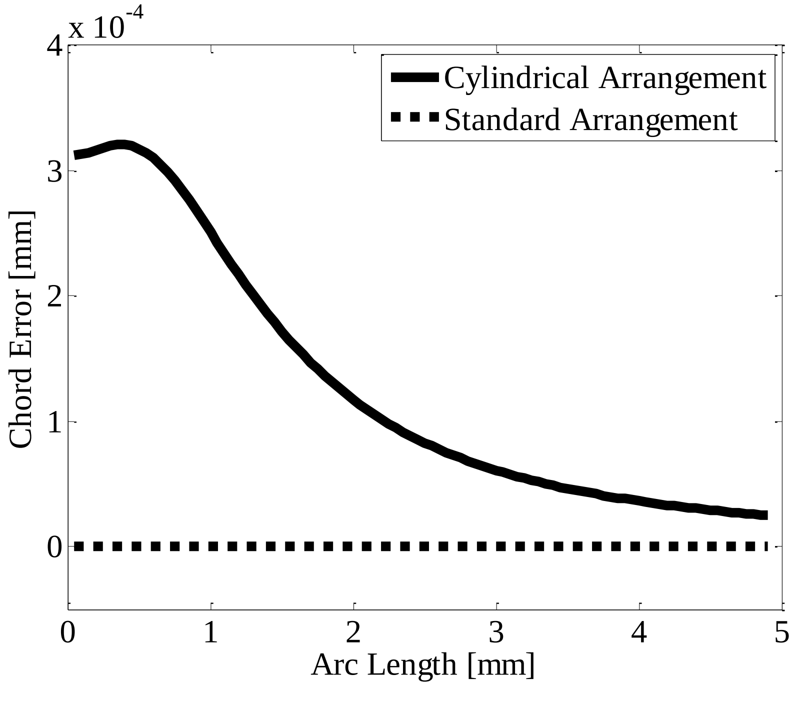

Figure 4.22: Chord error comparison for standard and cylindrical kinematic arrangements

4.3.1.2 Circular Arc

A circular arc is considered as in this case the cylindrical kinematic arrangement is expected show zero chord error. The circular arc considered in this analysis is shown in Figure 4.23. It is a 90-degree circular arc segment of 1 mm radius. Figure 4.24 and Figure 4.25 show the joint velocities versus the arc length in case of standard and cylindrical kinematic arrangements respectively. Finally, Figure 4.26 shows the chord error plotted against the arc length for both the kinematic arrangements.

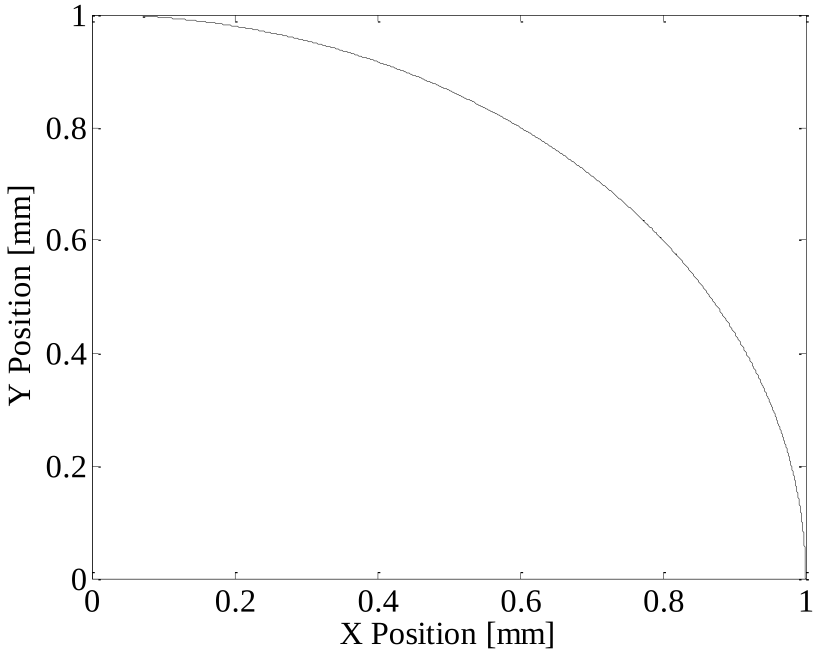

Figure 4.23: A circular arc segment

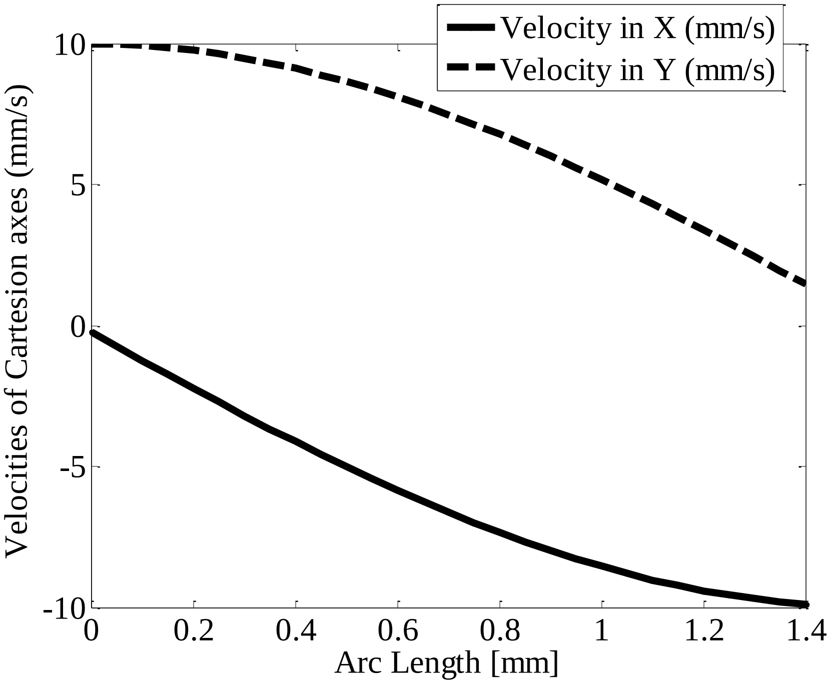

Figure 4.24: Standard arrangement joint velocities

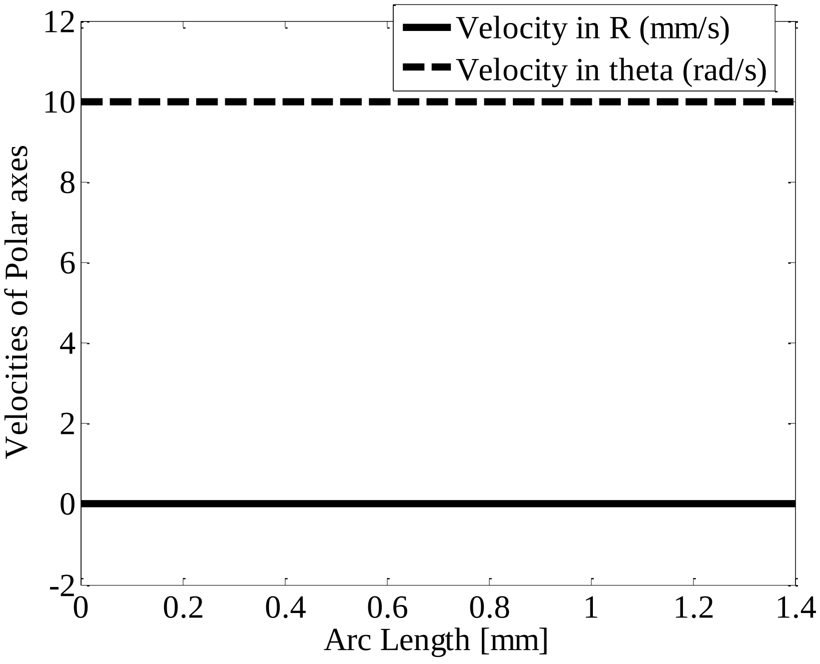

Figure 4.25: Cylindrical arrangement joint velocities

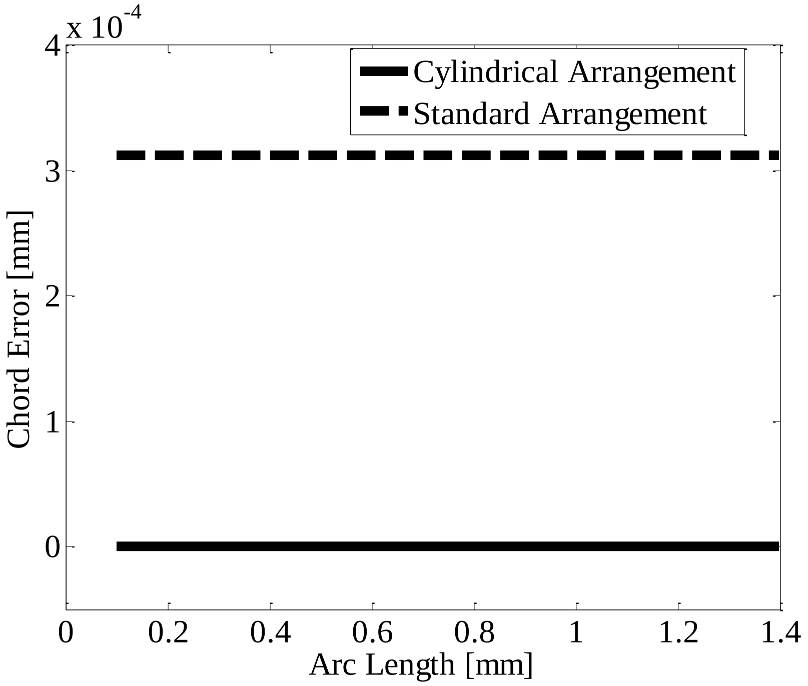

Figure 4.26: Chord error comparison for standard and cylindrical kinematic arrangements

4.3.1.3 Circle Involute

The circle involute is chosen as it is not expected to show zero error for any of the kinematic arrangements considered in this analysis. The parametric equation of the circle involute considered here is shown by Equation 4.4.

$$\begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \begin{bmatrix} 2(\cos u + u \cdot \sin u) \\ 2(\sin u - u \cdot \cos u) \end{bmatrix} \tag{4.4}$$

Figure 4.27 shows the circle involute corresponding to the Equation 4.4. Figure 4.28 and Figure 4.29 show the joint velocities versus the arc length for standard and cylindrical kinematic arrangements respectively. Figure 4.30 shows the chord error plotted against the arc length for both the kinematic arrangements.

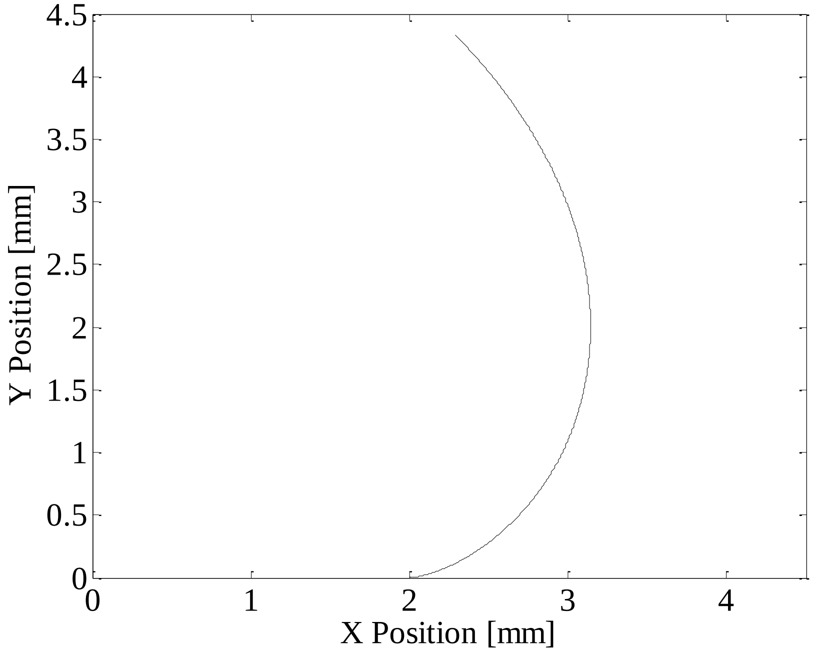

Figure 4.27: A circle involute

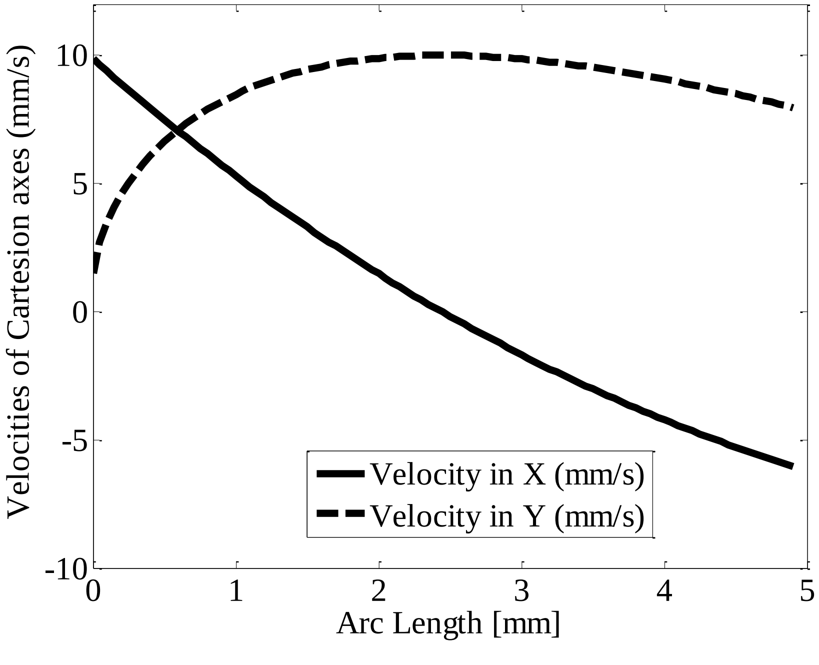

Figure 4.28: Standard arrangement joint velocities

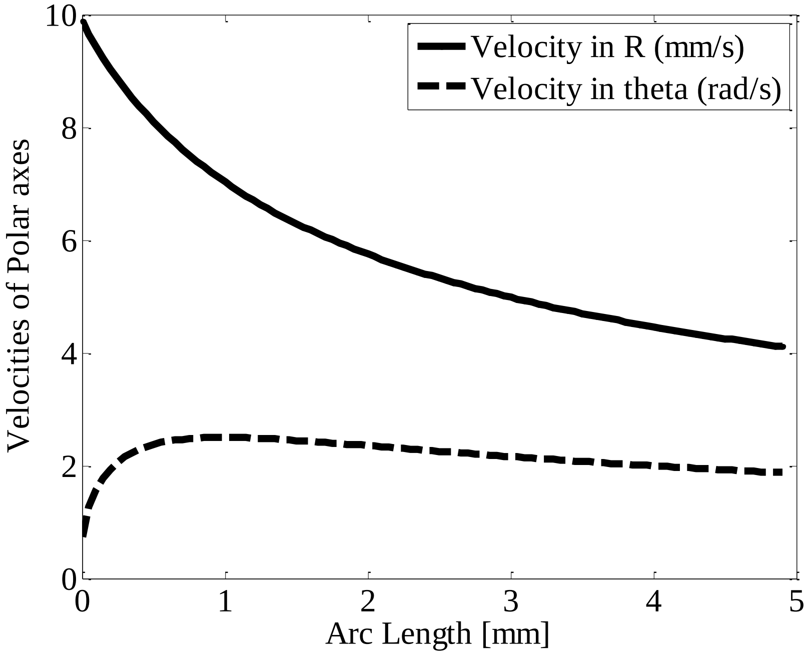

Figure 4.29: Cylindrical arrangement joint velocities

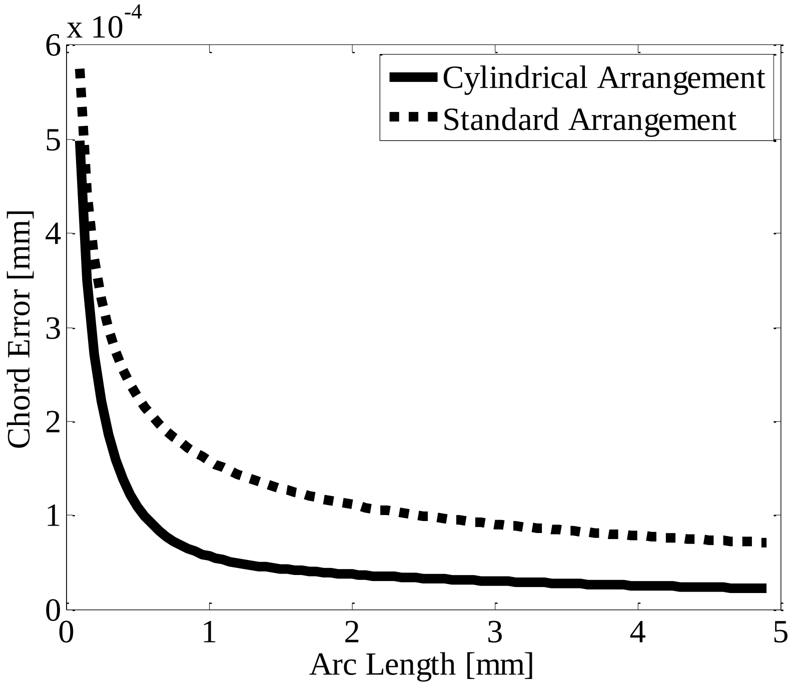

Figure 4.30: Chord error comparison for standard and cylindrical kinematic arrangements

4.3.1.4 Logarithmic Spiral

The next desired toolpath considered in this analysis is the logarithmic spiral. Its parametric equation is given by Equation 4.5.

$$\begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \begin{bmatrix} e^u \cos u \\ e^u \sin u \end{bmatrix} \tag{4.5}$$

This spiral is plotted, and as shown in Figure 4.31. Figure 4.32 and Figure 4.33 show the joint velocities for standard and cylindrical kinematic arrangements. Figure 4.34 shows the chord errors versus the arc-length for both the kinematic arrangements.

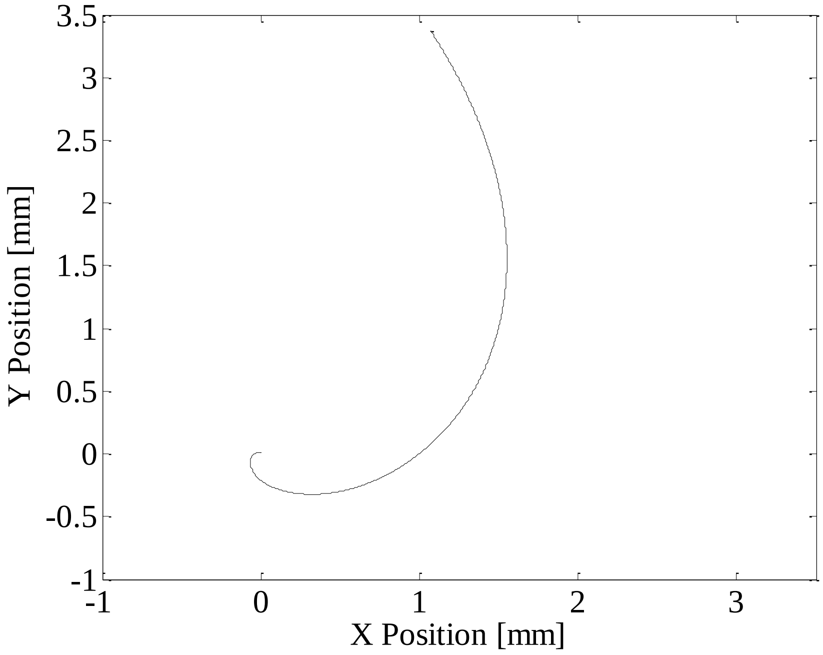

Figure 4.31: A logarithmic spiral

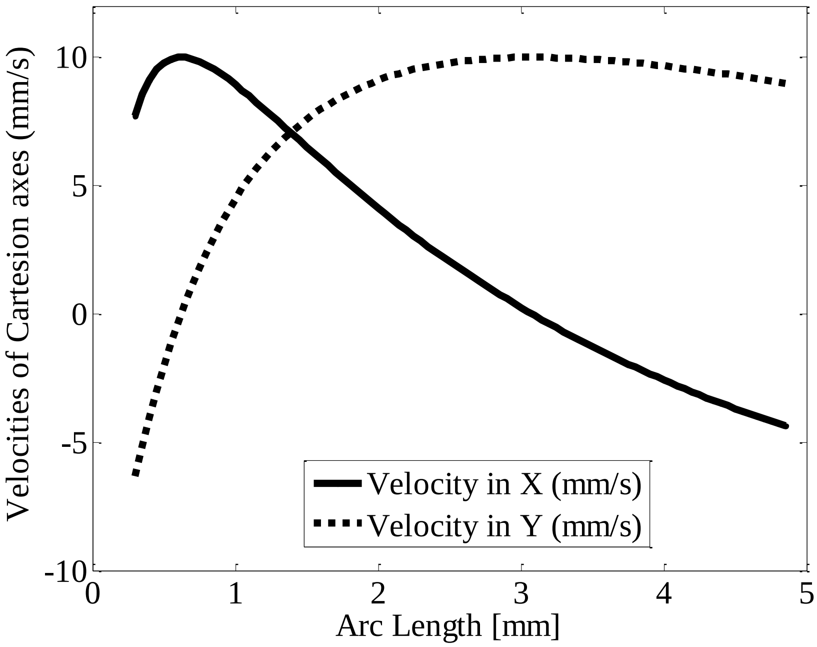

Figure 4.32: Joint velocities for standard kinematic arrangements

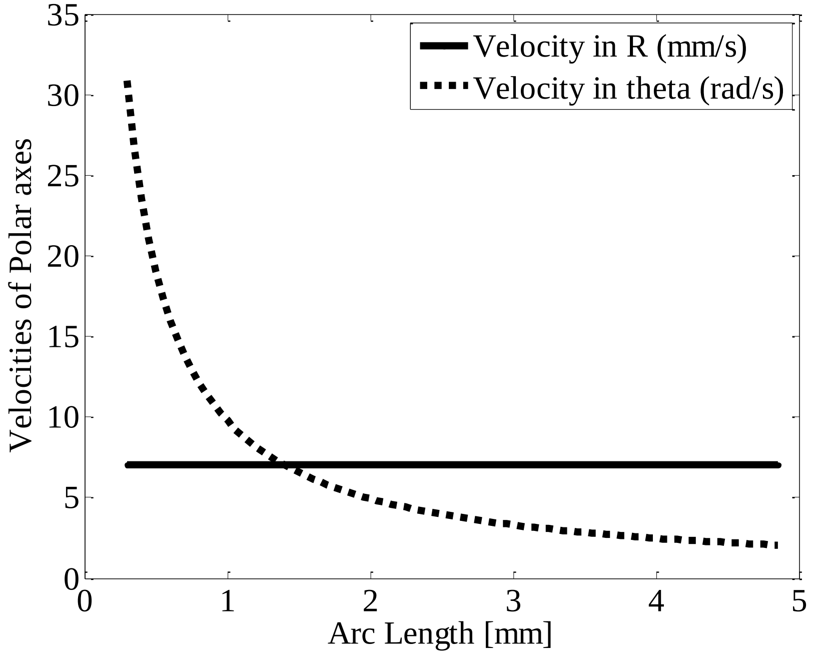

Figure 4.33: Joint velocities for cylindrical kinematic arrangements

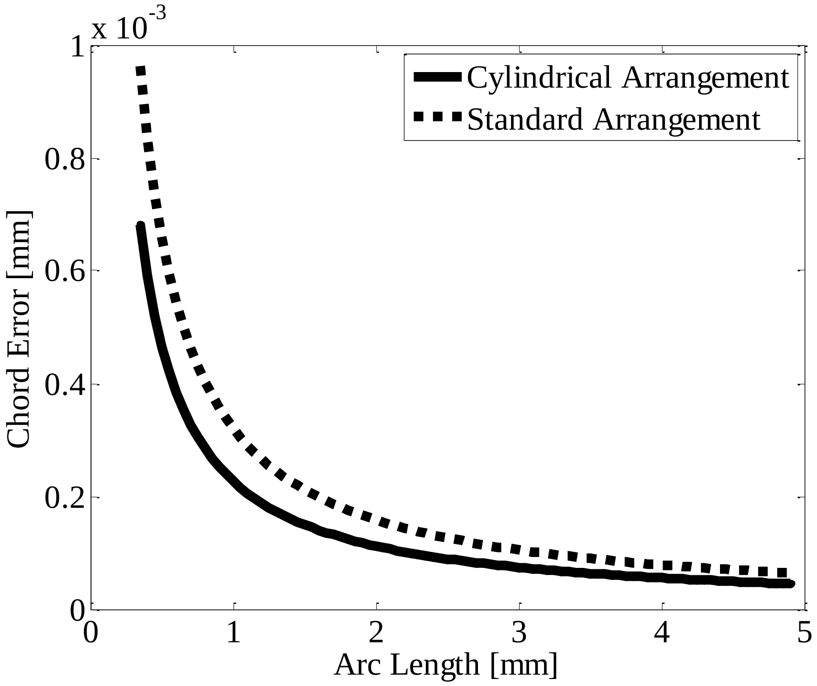

Figure 4.34: Chord error comparison for standard and cylindrical kinematic arrangements

4.3.1.5 Power function

The power function is made up to analyze the arrangements for an arbitrary case. The function considered in this analysis is represented by Equation 4.6.

$$\begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \begin{bmatrix} \dfrac{u^3}{3} \\ \dfrac{u^2}{2} \end{bmatrix} \tag{4.6}$$

Figure 4.35 shows the plot of this power function. Figure 4.36 and figure 4.37 show the joint velocities plotted against the arc length for standard and cylindrical kinematic arrangement respectively. Finally, Figure 4.38 shows the chord error against the arc length for both the kinematic arrangements.

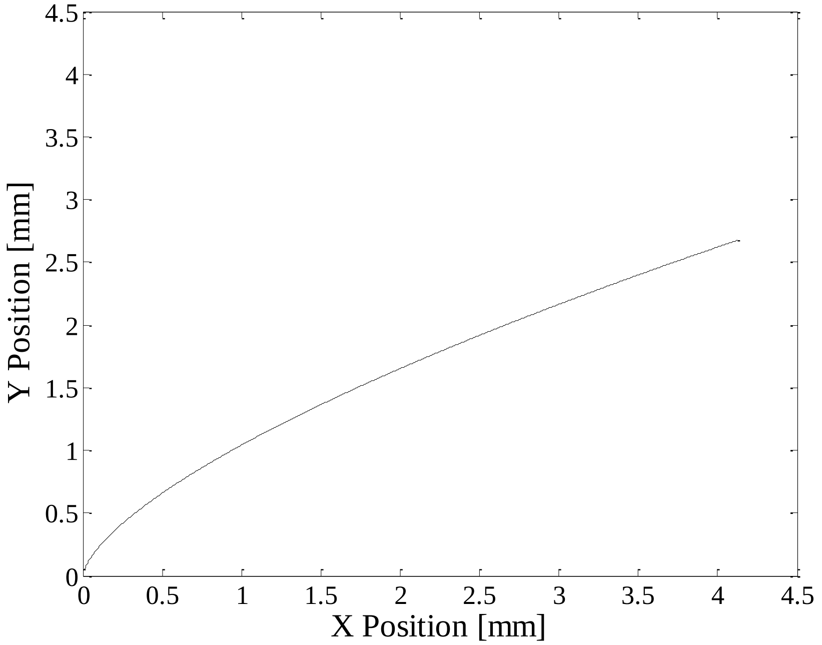

Figure 4.35: A power function

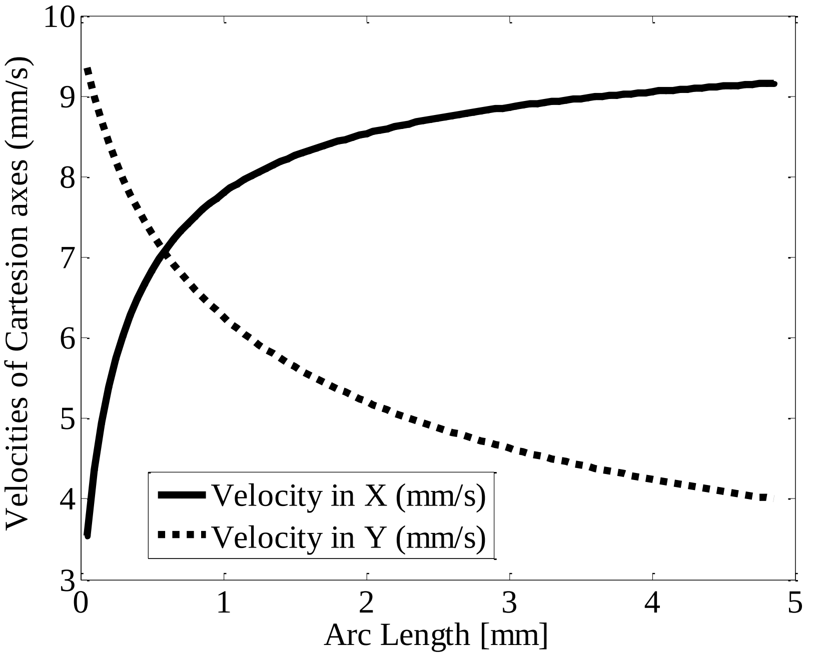

Figure 4.36: Joint velocities for standard kinematic arrangement

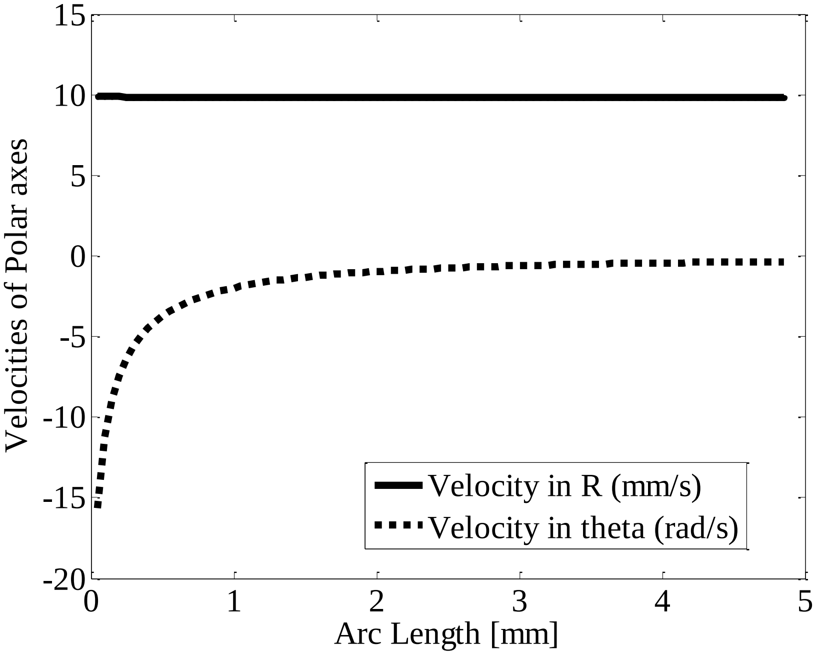

Figure 4.37: Joint velocities for cylindrical kinematic arrangement

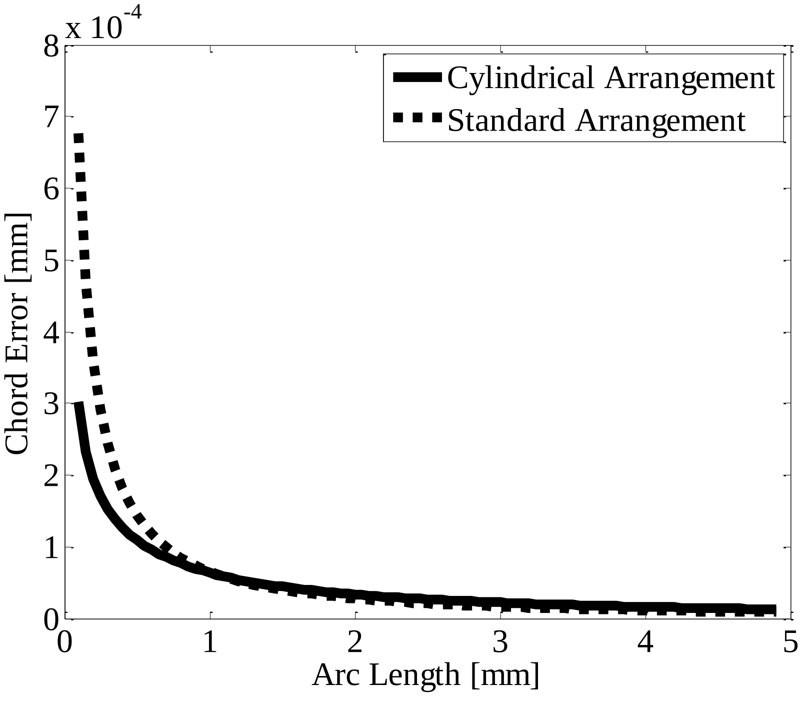

Figure 4.38: Chord error comparison for standard and cylindrical kinematic arrangements

4.3.2 Results of Experiment 2: Comparison of Standard Cartesian and Cylindrical Kinematic Arrangements with Kinematically Redundant Kinematic Arrangement

This section presents the comparison of the chord errors for all the three kinematic arrangements. A limit is placed on the chord error for the processing using kinematically redundant arrangement such that it does not exceed the limiting value. This limiting value is chosen by considering the simulation time for the corresponding desired toolpath.

The same desired toolpaths as presented in Section 4.2 are chosen here. The chord error comparison for the linear toolpath is presented in Figure 4.39. The chord error for the circular toolpath same is presented in Figure 4.40. The Figure 4.41 shows the error comparison for all the three arrangements for circle involute. The chord error for the logarithmic spiral is presented in Figure 4.42. The Figure 4.43 represents the chord error comparison for the power function.

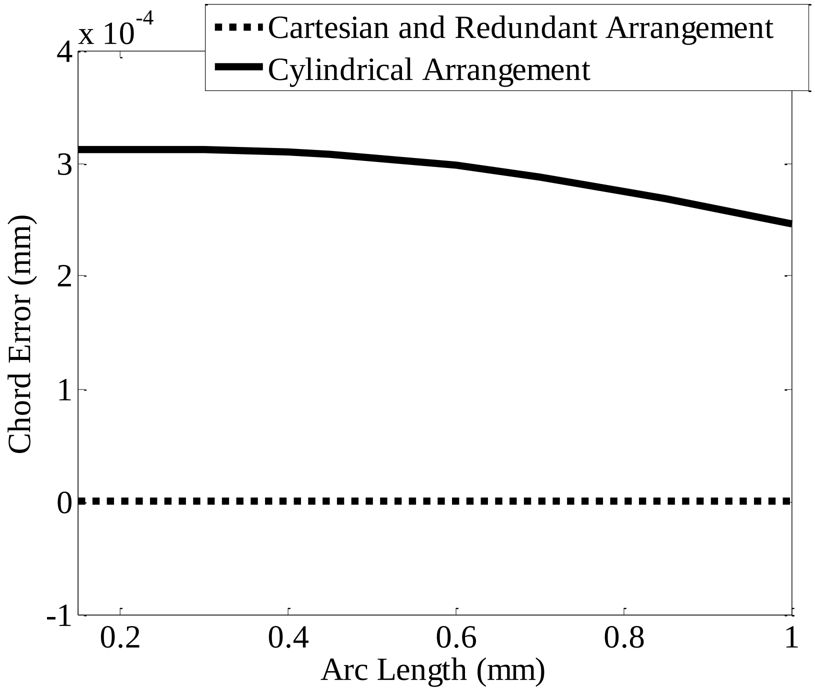

Figure 4.39: Chord error comparison for linear desired toolpath

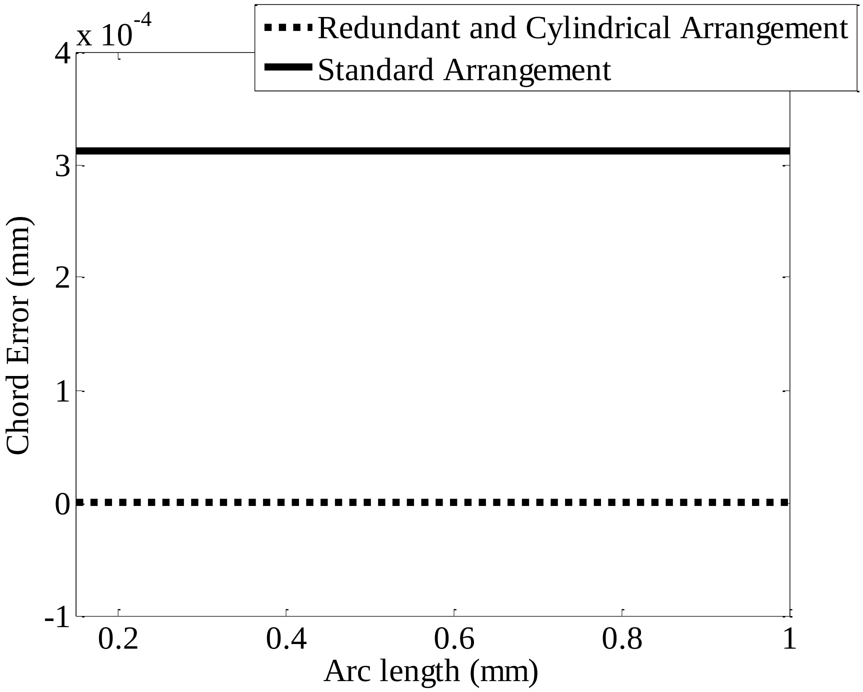

Figure 4.40: Chord error comparison for circular desired toolpath

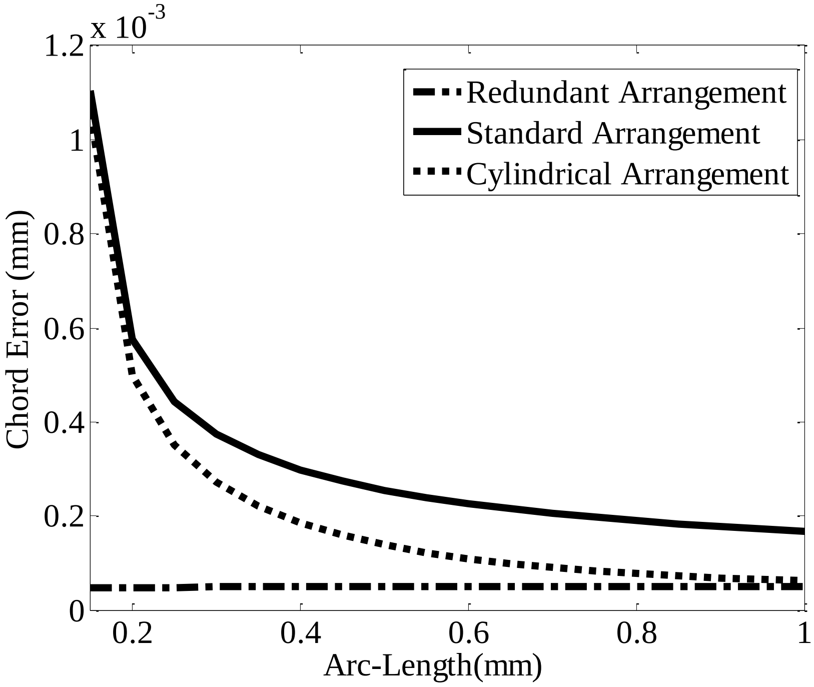

Figure 4.41: Chord error comparison for a circle involute

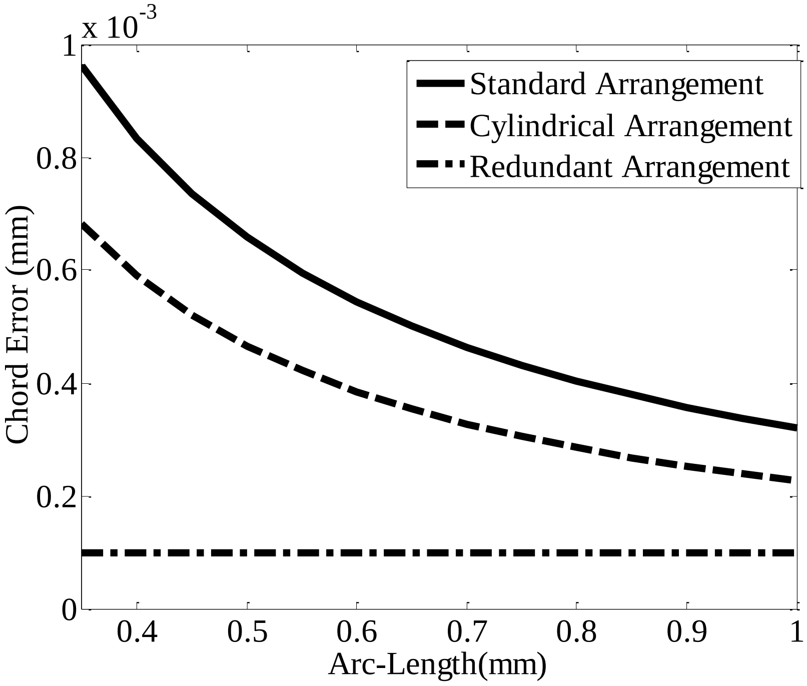

Figure 4.42: Chord error comparison for the logarithmic spiral

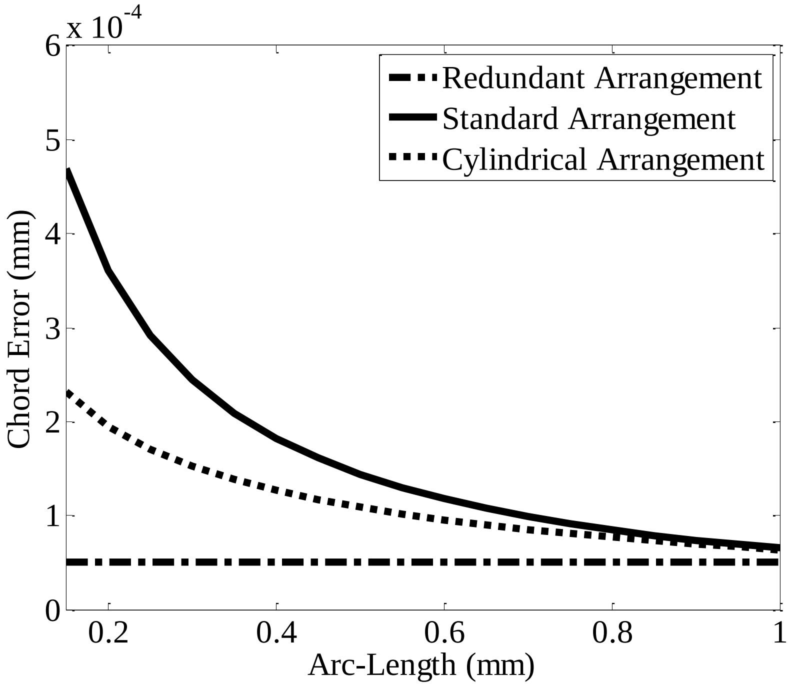

Figure 4.43: Chord error comparison for a power function

CHAPTER 5

DISCUSSION AND CONCLUSIONS

The body of research presented in this thesis has addressed the effect of kinematic arrangements in reduction of chord error in micromilling. This chapter will discuss the experimental results presented in Chapter 4. The effect of the kinematic arrangement on the chord error will also be discussed in details. Finally, the chapter will present the conclusions on the outline of the objectives and the hypotheses that were presented in the Chapter 1.

5.1 SIMULATION RESULTS DISCUSSION

First, the comparison of chord error for standard and cylindrical kinematic arrangements will be discussed. Later, the chord error for the redundant kinematic arrangement will be discussed in details. As the sampling time and the feedrate are kept constant throughout the experiments, it has become easier to analyze the effect of different kinematic arrangements on chord error.

5.1.1 Chord errors for standard and cylindrical kinematic arrangements

The chord error plot for a linear segment was first presented in Section 4.3.1. As the actual toolpath for the standard kinematic arrangement within a sampling interval is always a straight line, the linear desired toolpath is expected to be followed perfectly using the standard kinematic arrangement. On the other hand, the cylindrical arrangement does not produce a linear toolpath unless the two sampling consecutive points lie along a straight line along the axes of the linear joint. However, no two consecutive sampling points in this study are along the axis of the linear joint of cylindrical arrangement. Therefore, the cylindrical arrangement was expected to give higher chord error compared to that given by standard kinematic arrangement. As can be observed in Figure 4.22, the chord error for standard kinematic arrangement is zero, whereas the chord error for cylindrical arrangement is comparatively higher and it reduces as the arc length increases. It approaches zero as the sampling points considered increase distance from the origin.

The next desired toolpath considered was the circular arc. As discussed earlier, the toolpath produced by the cylindrical kinematic arrangement between a sampling interval is an Archimedean spiral, and a circular arc is a special case of Archimedean spiral. Therefore a cylindrical kinematic arrangement was expected to show zero chord error for a circular arc. However, standard kinematic arrangement was expected to show non-zero constant errors, as the velocities of the cylindrical joints were constants along the arc length. The chord error for cylindrical kinematic arrangement was always zero, and the chord error for standard kinematic arrangement was always constant and non-zero, as can be observed in Figure 4.26.

The results for the circle involute were followed by the circular arc. The joint velocities for both the kinematic arrangements do not show any particular patterns. Therefore, the chord error for none of the arrangements was predictable. As shown in Figure 4.30, the chord errors for both the arrangements reduce with the increasing arc-length. However, the cylindrical arrangement showed less error as compared to the standard arrangement.

The next desired toolpath in the desired toolpath selection was the logarithmic spiral. In this case too, the joint velocities do not show any specific trend, except that the joint velocity of the linear joint in cylindrical arrangement was constant along the increasing arc length. The chord error as can be seen in Figure 4.34, was higher with standard arrangement as compared to that with cylindrical arrangement.

Similar to that of the logarithmic spiral was presented next, in which only the linear joint in cylindrical arrangement had constant velocity with change in arc-length. As shown in Figure 4.38, reduced chord error was seen with the cylindrical arrangement. However, the chord error for standard arrangement is lesser than for cylindrical one beyond 1mm of arc-length.

5.1.2 Chord errors for kinematically redundant arrangements

This Section discusses the results for chord error for kinematically redundant arrangements which were presented in Section 4.3.2. The desired toolpaths for which the chord errors were compared for standard and cylindrical arrangements in Section 5.1.1 will be discussed in this Section, which also includes the chord error for redundant kinematic arrangement.

For the linear toolpath, the standard kinematic arrangement had shown zero chord error. As was shown Figure 4.39, the redundant kinematic arrangement has also shown zero chord error, as the standard arrangement is a special case of redundant arrangement, when the rotary joint does not rotate at all. Similarly for the circular arc, the redundant kinematic arrangement can turn into a special case to work exactly as the cylindrical kinematic arrangement and yield zero error, as was shown in Figure 4.40.

For circle involute as a desired toolpath, the chord error was minimized until it fell below the maximum allowable limit. Therefore, for each sampling interval, the chord error was just below the limit, as can be seen in Figure 4.41. Similarly, for the logarithmic spiral and the power function, the chord errors were minimized to their corresponding limits, as shown in Figure 4.42 and Figure 4.43 respectively. These plots also conclude that the chord errors for redundant kinematic arrangement always remain less than that for standard and cylindrical arrangements, except for the case of any of the arrangements giving zero chord error. In this special case, the plot for chord error for redundant arrangement overlaps with that for the arrangement which shows zero chord error along the increasing arc length.

5.2 CONCLUSIONS

The research presented in this thesis addresses the enhancement of high level of precision required in micromilling. More specifically, the work studies the impact of the kinematic arrangement on the compensation of scale effects. This section discusses the conclusions for the objectives explained in the Chapter 1. Table 5.1 shows the maximum chord errors over 1 mm of arc length for each desired toolpath for all kinematic arrangements.

Table 5.1: Maximum chord errors for each desired toolpath for all kinematic arrangements

Desired Toolpath	**Cartesian Kinematic Arrangement [mm]**	**Cylindrical Kinematic Arrangement [mm]**	**Kinematically Redundant Arrangement [mm]**
A Straight line	0	3.13×10^{-5}	0
Circular Arc	3.12×10^{-4}	0	0
Circular Involute	1.1×10^{-3}	1.05×10^{-3}	4.98×10^{-5}
Logarithmic Spiral	9.62×10^{-4}	6.82×10^{-4}	9.94×10^{-5}
Power Function	6.77×10^{-4}	3.02×10^{-4}	5×10^{-5}

The Table 5.2 shows the mean chord errors over 1 mm of arc length for each desired toolpath for all kinematic arrangements.

Table 5.2: Mean and Standard Deviation chord errors for each desired toolpath for all kinematic arrangements

Desired Toolpath	**Cartesian Kinematic Arrangement [mm]**	**Cylindrical Kinematic Arrangement [mm]**	**Kinematically Redundant Arrangement [mm]**
A Straight line	0	$(2.93 \pm 0.19) \times 10^{-5}$	0
Circular Arc	$(3.12 \pm 0.00) \times 10^{-4}$	0	0
Circular Involute	$(3.13 \pm 2.18) \times 10^{-4}$	$(2.07 \pm 2.35) \times 10^{-4}$	$(4.85 \pm 0.19) \times 10^{-5}$
Logarithmic Spiral	$(5.37 \pm 1.90) \times 10^{-4}$	$(3.80 \pm 1.35) \times 10^{-4}$	$(9.88 \pm 0.05) \times 10^{-5}$
Power Function	$(2.07 \pm 1.58) \times 10^{-4}$	$(1.30 \pm 0.61) \times 10^{-4}$	$(4.99 \pm 0.00) \times 10^{-5}$

Table 5.3 shows the percentage mean reductions in chord errors for each desired toolpath over 1 mm of arc length for all cylindrical and kinematically redundant kinematic arrangements compared to Cartesian kinematic arrangement.

Table 5.3: Mean reductions in % of chord errors for each desired toolpath for cylindrical and redundant kinematic arrangements compared to Cartesian kinematic arrangement

Desired Toolpath	**Cylindrical Kinematic Arrangement**	**Kinematically Redundant Arrangement**
A Straight line	0.0	0.0
Circular Arc	100.0	100.0
Circular Involute	44.4	80.2
Logarithmic Spiral	29.2	79.4
Power Function	26.5	58.8

Objective 1: To achieve new knowledge which enhances the understanding of key factors enhancing the high-speed, high-precision micromilling

- The knowledge of key factors affecting the high speed and high precision milling was insufficient for application to the microscale.
- The key factors affecting the production rate have been addressed. These key factors include the sampling time, the allowable feedrate and the scale effects.
- Key scale effect which affects the production rate the most was identified as the chord error, and hence addressed.
- New knowledge to enhance high-speed, high-precision micromilling has been attained. According to this new knowledge, the key factors affecting the production rate also include the kinematic arrangement used for the micromilling.

Objective 2: To utilize new knowledge to understand the need of a new mechanism which can be used to enhance high-speed, high-precision micromilling

- New cylindrical kinematic arrangement was introduced to minimize the chord error.
- The chord error for the new cylindrical kinematic arrangement was determined for several desired toolpaths and compared with the chord error for the standard arrangement.
- It proved the Hypotheses 1 and 2 that the standard Cartesian Kinematic Arrangement is not optimal for reduction of chord error in micromilling, and the cylindrical kinematic arrangement gives reduced error for some types of the desired toolpaths.
- It was also found that the chord error largely depends upon the kinematic arrangement, and zero chord error can also be achieved for some of the desired toolpaths if proper kinematic arrangement is chosen.

Objective 3: To apply the knowledge of robotics for developing a new kinematic arrangement for compensation of chord error in micromilling

- Adopting concept of kinematic redundancy from robotics, a new kinematically redundant arrangement was introduced.
- It proved the Hypothesis 3 that the kinematically redundant arrangement always gives reduced chord error for all different types of desired toolpaths.

CHAPTER 6

FUTURE WORK

A number of paths of possible supplementary study have been identified, during this research. This Section will present the future work for extension of this study. Topics are organized according to the importance of the work.

REAL-TIME TRAJECTORY PLANNING

The study presented in this thesis presents the way of trajectory planning. This method of trajectory planning can also be used in real-time. Real-time trajectory planning compensates for dynamic errors which can occur during machining. However, for real-time trajectory planning, the time required for trajectory planning between two consecutive sampling points should be less than the sampling time of the system. Therefore, the code for the trajectory planning for kinematically redundant arrangement should be so optimized that it cannot take more than the sampling time of the system.

The detrimental obstacles in real-time trajectory planning are as follows:

- Effect of high spindle speeds has to be compensated by high feedrates. If the feedrate increases, while sampling time remains same, the distance between the two sampling points increases. In most of the cases, it leads to increase in the chord error. Larger the chord error is, longer is the time for trajectory planning.
- Advancing technology has also been producing the systems with very short sampling times. It also shortens the allowable time for the trajectory planning.

REFERENCES

[1] Hupert M. L., Guy W. J., Llopis S. D., Situma C., Rani S., Nikitopoulos D. E. and Soper S. A., 2006, "High-precision micromilling for low-cost fabrication of metal mold masters," in Proceedings of SPIE 6112, Microfluidics, BioMEMS, and Medical Microsystems IV, San Jose.

[2] Li J., Friedrich C.R. and Keynton R. S., 2002, “Design and fabrication of a miniaturized, integrated, high-frequency acoustical lens-transducer system,” Journal of Micromechanics and Microengineering, vol. 12, no. 3, pp. 219-228, 2002

[3] Singh V., Desta Y., Datta P., Guy J., Clarke M., Feedback D. L., Weimert J. and Goettert J., 2007, “A hybrid approach for fabrication of polymeric BIOMEMS devices,” Microsystem Technologies, vol. 13, no. 3, pp. 369-377.

[4] Sodemann A. A., Li M., Mayor J. R. and Forest C. R., 2009, "Micromilling of molds for microfluidic blood diagnostic devices," Proceedings of the Annual Meeting of the American Society for Precision Engineering, Monterey.

[5] Filiz S., Xie L., Weiss L. E. and Ozdoganlar O. B., 2008, "Micromilling of microbarbs for medical implants," International Journal of Machine Tools & Manufacture, vol. 48, no. 3, pp. 459-472.

[6] Mescher M. J., Swan E. E. L., Fiering J., Holmboe M. E., Sewell W. F., Kujawa S. G., McKenna M. J. and Borenstein J. T., 2009, "Fabrication methods and performance of low-permeability microfluidic components for a miniaturized wearable drug delivery system," Journal of Microelectromechanical Systems,vol. 18, no. 3, pp. 501-510.

[7] Chiou Y-C., Lee R-T., Chen T-J. and Chiou J-M., 2012, "Fabrication of high aspect ratio micro-rod using a novel electrochemical micro-machining method," Precision Engineering, vol. 36, no. 2, pp. 193-202.

[8] Allegre O. J., Perrie W., Bauchert K., Liu D., Edwardson S. P., Dearden G. and Watkins K. G., 2012, "Real-time control of polarisation in ultra-short-pulse laser micro-machining," Applied Physics A, vol. 107, no. 2, pp. 445-454.

[9] Lu Z., and Yoneyama T., 1999, "Micro cutting in the micro lathe turning system," International Journal of Machine Tools and Manufacture, vol. 39, no. 7, pp. 1171-1183.

[10] Performance Micro Tool, [Online]. Available: www.pmtnow.com. [Accessed 6 February 2013].

[11] Mayor J. R. and Sodemann A. A., 2007, "An investigation of scale effects on process planning for high-speed high-precision micromachining operations”, 2nd International Conference on Micromanufacturing, Greenville.

[12] Zwyssig C., Kolar J. W. and Round S. D., 2008, "Mega-speed drive systems: pushing beyond 1 million RPM," in PCIM Europe 2008. International Exhibition & Conference for Power Electronics Intelligent Motion Power Quality, Nuremberg.

[13] Zhiming X. C., Jincheng C. and Zhengjin F., 2002, "Performance evaluation of a real-time interpolation algorithm for NURBS curves," International Journal of Advanced Manufacturing Technology, vol. 20, no. 4, pp. 270-276.

[14] Vijayaraghavan A., Sodemann A., Hoover A., Mayor J. R. and Dornfeld D., 2010, "Trajectory generation in high-speed, high-precision micromilling using subdivision curves," International Journal of Machine Tools and Manufacture, vol. 50, no. 4, pp. 394-403.

[15] Mayor J. R. and Sodemann A. A., 2008, "Intelligent Tool-Path Segmentation for Improved Stability and Reduced Machining Time in Micromilling," Journal of Manufacturing Science and Engineering, vol. 130, no. 3, pp. 031121-031134.

[16] Andres J., Gracia L. and Tornero J., 2012, "Implementation and testing of a CAM postprocessor for an industrial redundant workcell with evaluation of several fuzzified Redundancy Resolution Schemes," Robotics and Computer-Integrated Manufacturing, vol. 28, no. 2, pp. 265-274.

[17] Mi Z., Yang J., Kim J. H. and Abdel-Malek K., 2011, "Determining the initial configuration of uninterrupted redundant manipulator trajectories in a manufacturing environment," Robotics and Computer Integrated Manufacturing, vol. 27, no. 1, pp. 22-32.

[18] Zhang Y. and Wang J., 2004, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," Systems, Man, and Cybernetics, Part B, vol. 34, no. 1, pp. 752-759.

[19] Tatlicioglu E., Braganza D., Burg T. C. and Dawson D. M., 2009, "Adaptive control of redundant robot manipulators with sub-task objectives," Robotica, vol. 27, no. 6, pp. 873-881.

[20] Yeh S-S. and Hsu P-L., 2002, "Adaptive-feedrate interpolation for parametric curves with a confined chord error," Computer-Aided design, vol. 34, no. 3, pp. 229-237.

[21] Stoker J. J., 1969, “Differential Geometry”, Wiley-Interscience.

[22] Sun Y., Wang J. and Guo D., 2006, "Guide curve based interpolation scheme of parametric curves for precision CNC machining," International Journal of Machine Tools and Manufacture, vol. 46, no. 3, pp. 235-242.

[23] Farouki R. T. and Tsai Y-F., 2001, "Exact Taylor series coefficients for variable-feedrate CNC curve interpolators," Computer-Aided Design, vol. 33, no. 2, pp. 155-165.

[24] Tsai Y-F., Farouki R. T. and Feldman B., 2001, "Performance analysis of CNC interpolators for time-dependent feedrates along PH curves," Computer Aided Geometric Design, vol. 18, no. 3, pp. 245-265.

APPENDIX A

MATLAB CODE

Following MATLAB code was written for comparison of Standard Cartesian and Cylindrical Kinematic Arrangements for circular desired path.

```
clc
clear

ts=50;                          %sample time
tss=1;                          %sample time for calculating the errors
time=5;                         %simulation time

t=0;                            %initial time for inputting the curve equation
i=1;                            %counter

%Inputting the curve in cartesion and converting it in to polar

while(t<=time)

     xy(i,1)= cos(t);   % x co-ordinate
     xy(i,2)=sin(t);   % y co-ordinate
     xy(i,3)=t;

     pos_polar(i,1)= sqrt(xy(i,1)^2+xy(i,2)^2); % r co-ordinate
     pos_polar(i,2)= atan(xy(i,2)/xy(i,1));     % theta co-ordinate
     pos_polar(i,3)=t;

     i=i+1;
     t=t+ts/1000;
end

%calculating the velocities of cartesion axes
i=1;

for t = 0: ts/1000 : time-2*ts/1000
    vel(i,1)=(xy(i+1,1)-xy(i,1))/(ts/1000); %velocity in x
    vel(i,2)=(xy(i+1,2)-xy(i,2))/(ts/1000); %velocity in y
    vel(i,3)=t;

    vel(i,4)=sqrt((vel(i,1))^2+(vel(i,2))^2);

    i=i+1;
end

%calculating the radius of the curvature

i=1;

for t = 0: ts/1000 : time-3*ts/1000
    acc(i,1)=(vel(i+1,1)-vel(i,1))/(ts/1000); %accl in x
    acc(i,2)=(vel(i+1,2)-vel(i,2))/(ts/1000); %accl in y
    acc(i,3)=t;
    i=i+1;
end

i=1;

for t = 0: ts/1000 : time-3*ts/1000

    a1=sqrt((vel(i,1))^2+(vel(i,2))^2);
    a2=(vel(i,1))*(acc(i,2));
    a3=(vel(i,2))*(acc(i,1));

    if (a2>a3)
        a4= a2-a3;
    else
        a4=a3-a2;
    end

    rc(i,1)=(a1^3)/a4;
    rc(i,2)=t;

    i=i+1;
end

%change in radius of curvature
i=1;

for t = 0: ts/1000 : time-4*ts/1000

    rc_change(i,1)=(rc(i+1,1)-rc(i,1))/rc(i,1);
    rc_change(i,2)=t;

    i=i+1;
end


%Formulating the velocities of polar axes

i=1;

for t = 0: ts/1000 : time-2*ts/1000
    vel_polar(i,1)=(pos_polar(i+1,1)-pos_polar(i,1))/(ts/1000); %velocity in
r
    vel_polar(i,2)=(pos_polar(i+1,2)-pos_polar(i,2))/(ts/1000); %velocity in
theta
    vel_polar(i,3)=t;

    vel_polar(i,4)=sqrt((vel_polar(i,1))^2+(vel_polar(i,2))^2);

    i=i+1;
end

%Determining the Tool position cac
```

```
ind=0;
iter=0;
for t=0:tss:(time*1000-ts)
  if mod(t,ts) == 0
      ind=ind+1;
      iter=t+1;
      k=0;
      pos_tool_cac(t+1,1)=xy(ind,1);
      pos_tool_cac(t+1,2)=xy(ind,2);

      pos_tool_cac(t+1,3)=t/1000;

  else
      k=k+1;
      pos_tool_cac(t+1,1)=pos_tool_cac(iter,1)+k*0.001*vel(ind,1)*tss;
      pos_tool_cac(t+1,2)=pos_tool_cac(iter,2)+k*0.001*vel(ind,2)*tss;

      pos_tool_cac(t+1,3)=t/1000;

  end
end


%Determining the Tool position pac

ind=0;
iter=0;
for t=0:tss:(time*1000-ts)
  if mod(t,ts) == 0
      ind=ind+1;
      iter=t+1;
      k=0;
      pos_tool_pac(t+1,1)=pos_polar(ind,1);
      pos_tool_pac(t+1,2)=pos_polar(ind,2);

      pos_tool_pac(t+1,3)=t/1000;

      pos_tool_pac(t+1,4)= pos_tool_pac(t+1,1)*cos(pos_tool_pac(t+1,2));
      pos_tool_pac(t+1,5)= pos_tool_pac(t+1,1)*sin(pos_tool_pac(t+1,2));

  else
      k=k+1;
      pos_tool_pac(t+1,1)=pos_tool_pac(iter,1)+k*0.001*vel_polar(ind,1)*tss;
      pos_tool_pac(t+1,2)=pos_tool_pac(iter,2)+k*0.001*vel_polar(ind,2)*tss;

      pos_tool_pac(t+1,3)=t/1000;

      pos_tool_pac(t+1,4)= pos_tool_pac(t+1,1)*cos(pos_tool_pac(t+1,2));
      pos_tool_pac(t+1,5)= pos_tool_pac(t+1,1)*sin(pos_tool_pac(t+1,2));

  end
end
```

```
%Calculation of the curve position

t=0;
j=1;

while(t<time)
    pos_curve(j,1)=cos (t); %t^2;curve position x
    pos_curve(j,2)=sin(t); %t+0.0001;curve position y
    pos_curve(j,3)= t;

    t=t+tss/1000;
    j=j+1;
end

%Error calculation pac

index = 0;

for i =0: tss: time*1000-2*ts
    x1=pos_tool_pac(1+i/tss,4);
    y1=pos_tool_pac(1+i/tss,5);

   if  mod(i,ts) == 0
       prev = i;
    end

    min=999;
    if mod(i,ts)~=0
    j=prev+tss;
    while (mod(j+1,ts)~=0)
      x2=pos_curve(1+j/tss,1);
      y2=pos_curve(1+j/tss,2);

      dist = sqrt((x2-x1)^2+(y2-y1)^2);
        if dist<min
            min=dist;
        end
    j=j+tss;
    end
    smallest(i,1) = min;
    end

    if mod((i+tss),ts) == 0
        max=0;
        for k = prev+tss:tss: i-1
            if smallest(k,1) > max
                max=smallest (k,1);
            end
        end
        index = index+1;
        max_interval_pac(index,1)=max;
        max_interval_pac(index,2)=(i+tss)/1000;
    end
end

%Error calculation cac

index = 0;

for i =0: tss: time*1000-2*ts
    x1=pos_tool_cac(1+i/tss,1);
    y1=pos_tool_cac(1+i/tss,2);

   if  mod(i,ts) == 0
       prev = i;
    end

    min=999;
    if mod(i,ts)~=0
    flag=1;
    j=prev+tss;
    while (mod(j+1,ts)~=0)
      x2=pos_curve(1+j/tss,1);
      y2=pos_curve(1+j/tss,2);

      dist = sqrt((x2-x1)^2+(y2-y1)^2);
        if dist<min
            min=dist;
        end
    j=j+tss;
    end
    smallest(i,1) = min;
    end

    if mod((i+tss),ts) == 0
        max=0;
        for k = prev+tss:tss: i-1
            if smallest(k,1) > max
                max=smallest (k,1);
            end
        end
        index = index+1;
        max_interval_cac(index,1)=max;
        max_interval_cac(index,2)=(i+tss)/1000;
    end
end
```

Following MATLAB script was written for evaluating chord error for circle using Kinematically Redundant Arrangement.

```
clc
clear

tic

ts=50; %ms
tss=0.025; %ms
time=1; %seconds

thk=0.1;

t=0;
i=1;

while(t<=time)

    xy(i,1)=cos(t);    % x co-ordinate
    xy(i,2)=sin(t);    % y co-ordinate

    pos_polar(i,1)=sqrt(xy(i,1)^2+xy(i,2)^2);
    pos_polar(i,2)=atan(xy(i,2)/xy(i,1));

    i=i+1;
    t=t+ts/1000;

end

%velocities

i=1;

for t=0:ts/1000:time-2*ts/1000

    vel_xyc(i,1)=(-
pos_polar(i,1)*cos(pos_polar(i,2))+pos_polar(i+1,1)*cos(pos_polar(i+1,2)))/(t
s/1000);
    vel_xyc(i,2)=(-
pos_polar(i,1)*sin(pos_polar(i,2))+pos_polar(i+1,1)*sin(pos_polar(i+1,2)))/(t
s/1000);

    vel_xyc(i,3)= (thk-pos_polar(i+1,2))/(ts/1000);
    vel_xyc(i,3)=t;

    i=i+1;
end

%toolpath
```

```
for i=1:length(vel_xyc)

   thk=pos_polar(i+1,2);

   [xdash, ydash, xdot, ydot, thdot]=toolpath(xy(i,1), xy(i,2), xy(i+1,1), 
xy(i+1,2), ts, tss, thk);

   [xdash_c, ydash_c]= curve(ts, tss, ts*(i-1)/1000);

   [error, tp_point, curve_point]=chorderror(xdash, ydash, xdash_c, ydash_c, 
(ts/tss)+1);

   [optimized_error, tp_point, curve_point, iteration] = 
optimization(xy(i,1), xy(i,2), xy(i+1,1), xy(i+1,2), xdash, ydash, xdash_c, 
ydash_c, ts, tss, pos_polar(i,1), pos_polar(i+1,1), thk, 0.00005);

   error_comparison(i,1)=error;
   error_comparison(i,2)=optimized_error;
   error_comparison(i,3)=iteration;

end

toc
```

MATLAB function for finding the actual toolpath is as follows:

```
function [xdash ydash xdot ydot thdot] = toolpath(x1, y1, x2, y2, T, tss,
thk)

subparts=T/tss;

%Start point- Cylindrical co-ordinates
r1=sqrt(x1^2+y1^2);
th1=atan(y1/x1);

%End point- Cylindrical co-ordinates
r2=sqrt(x2^2+y2^2);
th2=atan(y2/x2);

%Theta velocity
thdot=(thk-th2)/T;



%X-Y velocities
xdot= (-r1*cos(th1)+r2*cos(thk))/T;
ydot= (-r1*sin(th1)+r2*sin(thk))/T;

%Small discrete time
time=T/subparts;

%start point in the following loop
r(1,1)=r1;
th(1,1)=th1;

%start point in the following loop
x(1,1)=x1;
y(1,1)=y1;

xdash(1,1)=x1;
ydash(1,1)=y1;

%Loop for determining intermediate discrete points
for i=2:(subparts+1)

    x(i,1)=x(i-1,1)+xdot*time;
    y(i,1)=y(i-1,1)+ydot*time;

    r(i,1)=sqrt((x(i,1))^2+(y(i,1))^2);
    th(i,1)=(atan(y(i,1)/x(i,1)))-(i-1)*thdot*time;

    xdash(i,1)=r(i,1)*cos(th(i,1));
    ydash(i,1)=r(i,1)*sin(th(i,1));


end
```

```
    plot(xdash(:,1),ydash(:,1));
    xlabel('X')
    ylabel('Y')
    hold on
    axis equal



end
```

MATLAB function for finding the desired toolpath locations is as follows:

```
function [ xdash_c ydash_c ] = curve(ts, tss, current_t)

t=current_t;

subparts=ts/tss;

for i=2:(subparts+1)

    xdash_c(i,1)=cos(t);
    ydash_c(i,1)=sin(t);

    t=t+tss/1000;

end

    [rows columns]=size(xdash_c);
    plot(xdash_c(2:rows,1),ydash_c(2:rows,1), 'r');
    axis equal

end
```

MATLAB function for finding the chord error is as follows:

```
function [ chord_error, tp_point, curve_point] = chorderror(xt, yt, xc, yc,
subparts)

    i=zeros(subparts);
    j=zeros(subparts);

    for i=1:subparts-3
        for j=1:subparts-3
           distance(i,j)=sqrt(((xt(i+2)-xc(j+2))^2)+((yt(i+2)-yc(j+2))^2));
        end

        [error(i),c_point(i)]=min(distance(i,:));

%         x=[xt(i), xc(c_point(i))];
%         y=[yt(i), yc(c_point(i))]
%         plot(x,y, 'k')

    end

    [chord_error, tp_point]=max(error);
    curve_point=c_point(tp_point);


    x=[xt(tp_point) xc(curve_point)];
    y=[yt(tp_point) yc(curve_point)];
    plot(x,y, 'k', 'Linewidth', 2)
    hold on

end
```

MATLAB function for optimizing the chord error is as follows:

```
function [error tp_point curve_point iteration] = optimization(x1, y1, x2,
y2, xdash, ydash, xdash_c, ydash_c, ts, tss, th1, th2, thk, max_error)

    subparts=ts/tss;

    [error, tp_point, curve_point]=chorderror(xdash, ydash, xdash_c, ydash_c,
subparts+1);

    delta=(sqrt((xdash(tp_point))^2+(ydash(tp_point))^2))-
(sqrt((xdash_c(curve_point))^2+(ydash_c(curve_point))^2));

    delta_sign=delta/abs(delta);

    iteration=0;

    while(error>max_error)

       thk=thk-0.01;
       [xdash, ydash, xdot, ydot, thdot]=toolpath(x1, y1, x2, y2, ts, tss,
thk);
       [error, tp_point, curve_point]=chorderror(xdash, ydash, xdash_c,
ydash_c, subparts+1);

       iteration=iteration+1;

    end

end
```