# Cycle Time Reduction of 5-axis Laser Drilling via Time-optimal Trajectory Generation and Sequence Optimization

by

Kyongjae Woo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2019

# AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

# Abstract

Cycle time reduction is one of the crucial tasks in manufacturing that needs to be achieved to maximize productivity and profits. Laser drilling processes, depending on the size and complexity of the parts, require few hundreds to few thousands of holes to be drilled. Therefore, cycle time is directly related to in what order and manner the holes are visited. In this thesis, a method of cycle time reduction for 5-axis percussion laser drilling process is presented via generation of time-optimal trajectory and optimization of hole visiting sequence.

In percussion laser drilling, a series of laser pulses are fired to each hole while the workpiece is stationary. Once a hole is completely opened up, then drilling of the next hole continues by repositioning the workpiece with respect to the beam. This stop-and-go nature of the drilling process enables one to describe the sequence optimization problem as a well-known Traveling Salesman Problem (TSP) in combinatorial optimization. The objective of TSP is to find a minimum cost sequence of points when the point-to-point cost information for every possible pair is known. In the case of the minimum cycle time problem, the point-to-point cost is the travel time, and the objective of TSP is to find a sequence with the minimum overall travel time.

In planning of time-optimal trajectory for point-to-point motion under a specified path, industry uses CNC controller's G00 (rapid traverse) + TRAORI (5-axis transformation and tool orientation retaining tactic) commands. To be practically beneficial, time-optimal trajectory generation strategies discussed in this thesis is focused on closely estimating these CNC controller's behaviors. A total of four strategies are studied, and the most accurate strategy is chosen by comparing the results with the experimentally measured CNC trajectories. The most accurate one specifies the tool paths in Workpiece Coordinates followed by iterative velocity profiling of the tool path parameter to achieve minimum time trajectory under the machine's velocity, acceleration, and jerk limits.

With every hole-to-hole travel time calculated from the above strategy, sequence optimization can be conducted. In this thesis, two methods from the industry partner, the proposed method, and the optimal solver method are discussed. Due to licensing limitations, the proposed method is developed in-house instead of using existing non-commercial TSP algorithms. The proposed method uses local search heuristics approach inspired by famous Lin-Kernighan heuristics. The results are compared to the optimal solutions generated from the non-commercial state-of-the-art TSP solver called Concorde for benchmarking purposes.

To understand the impact of the research in a real environment, one sample part and its original drilling process information have been made available by the industry partner. Although the full experimental results are not yet acquired at the moment of writing this thesis, the simulation results show that the proposed sequencing optimization in conjunction with the proposed hole-to-hole trajectory generation strategy for correct estimation of travel time improves the overall cycle time by 26.0 %.

# Acknowledgments

This research has been sponsored by Pratt and Whitney Canada (P&WC) and NSERC CANRIMT II.

I would like to acknowledge the assistance of Dr. Serafettin Engin and Mr. Samuel Jacques from P&WC and their support in the experimental development of this thesis. I would like to thank the members of my committee, Professor Soo Jeon and Professor Michael Mayer for taking the time to read my thesis and provide valuable feedback.

I would like to thank Mr. Chiapei Wang for his valuable insights and technical support. Many thanks to all my friends and the laboratory team for their continuous support and motivation.

I sincerely acknowledge my co-supervisor William Cook for his insightful guidance and expertise. I would like to express the deepest appreciation to my supervisor, Professor Kaan Erkorkmaz, for his immeasurable support and encouragement. This thesis would not have been possible without his guidance, persistent help, and true support.

I am extremely grateful to my best friend Ami Woo for her endless inspiration, motivation, and encouragement. Lastly, I thank my family for their endless support and love.

# Dedication

*I dedicate this thesis to my family and my best friend Ami Woo.*

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1
# Introduction

Cycle time reduction is one of the crucial tasks in manufacturing that needs to be achieved to maximize productivity and profits. Laser drilling processes, depending on the size and complexity of the parts, require few hundreds to few thousands of holes to be drilled. Therefore, cycle time is directly related to in what order and manner the holes are visited. In this thesis, a method of cycle time reduction for 5-axis percussion laser drilling process is presented via generation of time-optimal trajectory and optimization of hole visiting sequence.

Laser drilling is a unique method for producing high depth-to-diameter ratio holes. Unlike the conventional drilling methods, it is a non-contact process, from which holes can be drilled in materials that are considered traditionally as hard-to-machine. In addition, it is capable of drilling holes with shallow angles to the surface, which makes it a highly productive method for producing arrays of holes with complex orientations on freeform shaped components such as film-cooling holes for gas turbine combustion chamber panels. Its industrial applications also include printed circuit boards (PCB), medical devices, fuel injection nozzles, micro holes for scientific instrumentation, and so on.

One common laser drilling method is percussion drilling. In percussion drilling, a series of laser pulses are fired to each hole while the workpiece is stationary. Each laser pulse removes a certain volume of material via ablation. Once a hole is completely opened up, then drilling of the next hole continues by repositioning the workpiece with respect to the beam. For $n$ number of holes, there are $\frac{(n-1)!}{2}$ number of hole visiting sequences available, of which total travel times vary drastically. Therefore, obtaining optimal hole visiting sequence and hole-to-hole trajectory subject to machine capabilities enables the reduction of the overall beam positioning time, which contributes to an increase in manufacturing productivity. In Chapter 2, literatures that discuss trajectory generations for industrial machines and sequence optimizations are reviewed.

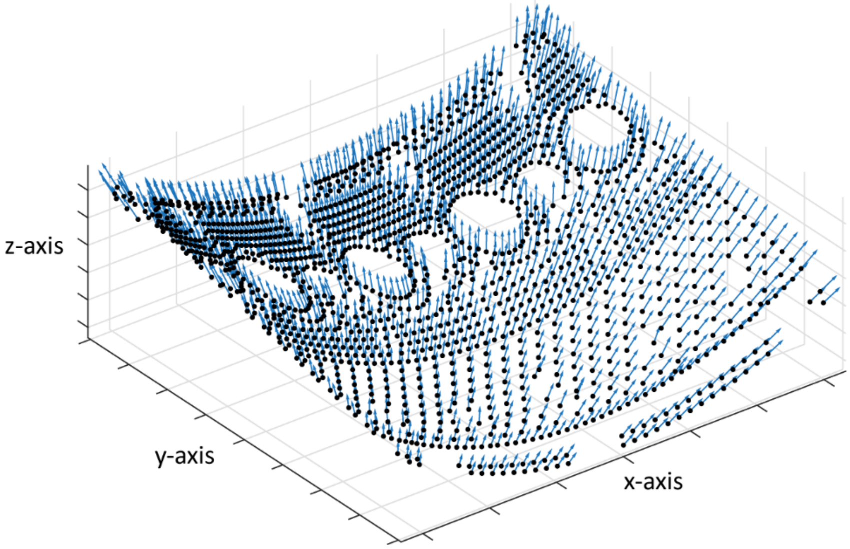


**Figure 1.1 Hole locations and orientations for a sample part**

A sample part of percussion laser drilling application of a gas turbine combustion chamber panels from a Canadian aero-engine producer, Pratt & Whitney Canada, is shown in Figure 1.1. Such hole patterns are drilled with a 5-axis laser drilling machine as shown in Figure 1.2 to comply with complex hole locations and their orientations. The machine has three translational axes $(X, Y, \& Z)$ and two rotational axes $(B \& C)$.

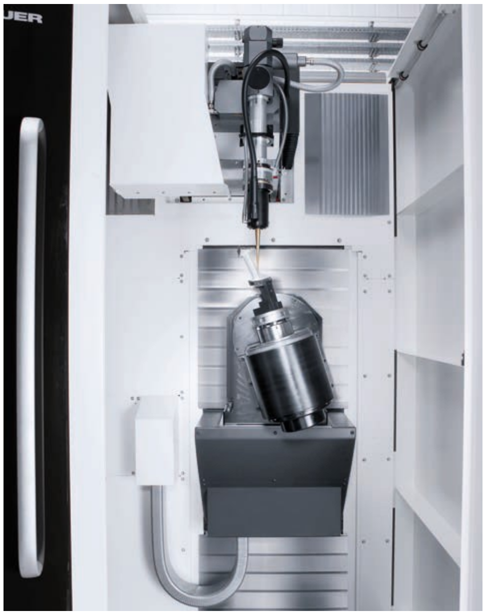


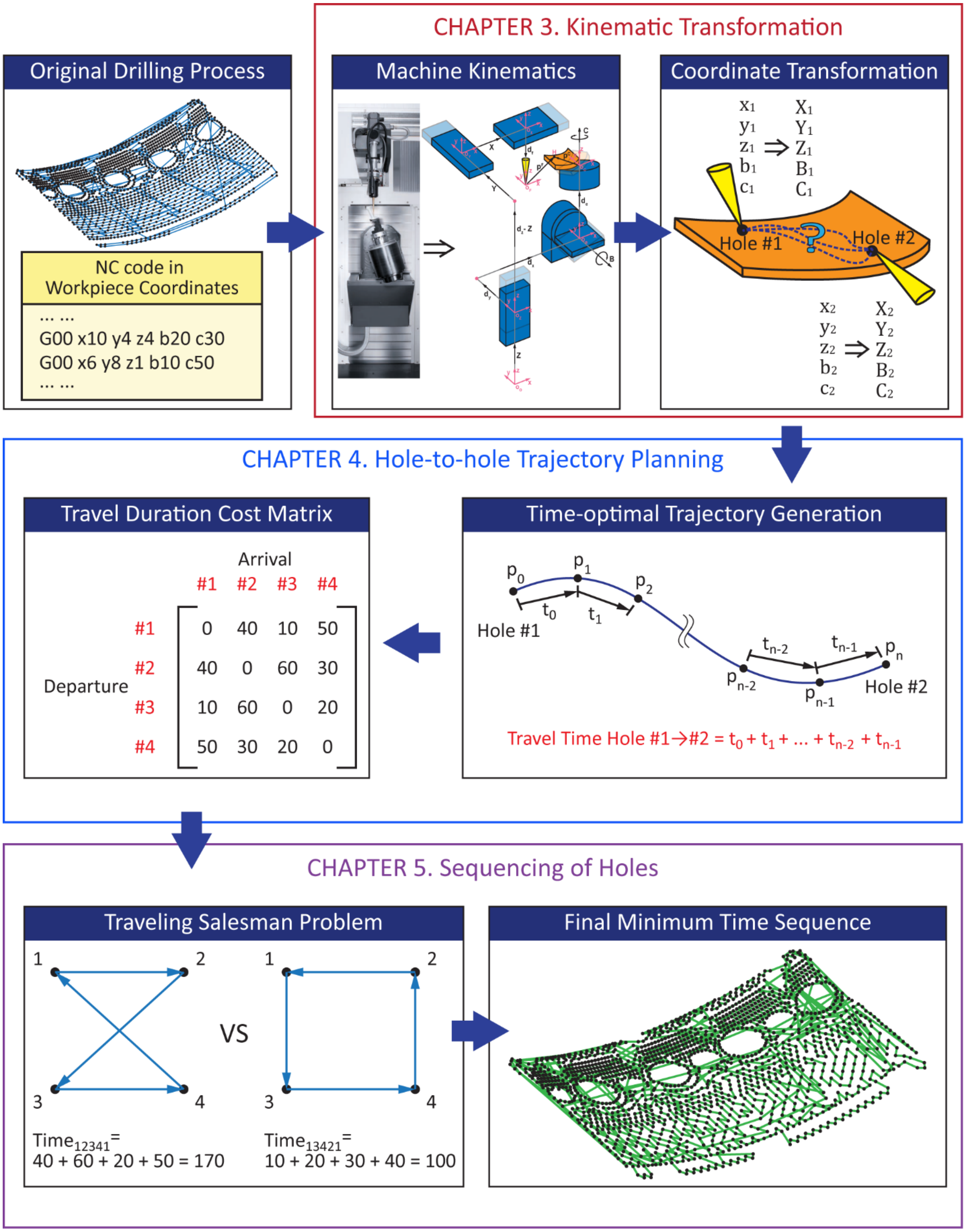
**Figure 1.2 5axis laser machine [1]**

An overview of the thesis work is illustrated in Figure 1.3. The original unoptimized drilling process is given in the form of NC code in which the positions and orientations of the drilling holes are defined with respect to the workpiece (Workpiece Coordinates). In order to optimize the process under each servo drive’s kinematic constraints, such as velocity limits, acceleration limits, and jerk limits, it is essential to represent these values with respect to the machine body (Machine Coordinates). Such transformation is referred to as the kinematic transformation and are discussed in more detail in Chapter 3.

Once the starting and the end coordinates are known, hole-to-hole trajectory planning is conducted to yield the minimum travel time. Although it is not time-optimal, in order to maximize the benefits of the industrial partner in terms of the optimality and practicality, the study is focused on delivering a trajectory generation method that closely estimates the current implementation of NC codes G00 (rapid traverse) with TRAORI (CNC controller’s 5-axis transformation with tool orientation tactic) motions exerted by the machine. With correct estimation, it is possible to obtain realistic hole-to-hole travel durations, and from this information, the optimal sequence problem can produce more precise and realistic results. The detail is discussed in Chapter 4.

In the context of a multi-point drilling application, the optimal sequence problem can be described by a well-known Traveling Salesman Problem (TSP) in combinatorial optimization. Given a set of cities and the distances between each pair of cities, the objective of a standard TSP is to find the minimum distance tour to visit all the cities only once. In this thesis, the problem of finding an optimal sequence is formulated in the TSP format so that when a set of holes and the travel durations between each pair of holes are calculated, the resultant sequence would yield minimum travel time to visit all the holes only once. In Chapter 5, the formulation to TSP and the proposed solution algorithm is presented.

Then the rest of the thesis is organized as follows: both simulation and experimental results showing the effectiveness of proposed method are presented in Chapter 6, followed by conclusions and future work in Chapter 7.


CHAPTER 3. Kinematic Transformation
Original Drilling Process
Machine Kinematics
Coordinate Transformation
NC code in
Workpiece Coordinates
x1 y1 z1 b1 c1 ⇒ X1 Y1 Z1 B1 C1
Hole #1
Hole #2
x2 y2 z2 b2 c2 ⇒ X2 Y2 Z2 B2 C2
CHAPTER 4. Hole-to-hole Trajectory Planning
Travel Duration Cost Matrix
Time-optimal Trajectory Generation
Arrival
Departure
p0
p1
p2
t0
t1
Hole #1
tn-2
tn-1
pn
pn-2
pn-1
Hole #2
CHAPTER 5. Sequencing of Holes
Traveling Salesman Problem
Final Minimum Time Sequence
1
2
3
4
VS
1
2
3
4

```
… …
G00 x10 y4 z4 b20 c30
G00 x6 y8 z1 b10 c50
… …
```

|  | #1 | #2 | #3 | #4 |
|---|---|---|---|---|
| #1 | 0 | 40 | 10 | 50 |
| #2 | 40 | 0 | 60 | 30 |
| #3 | 10 | 60 | 0 | 20 |
| #4 | 50 | 30 | 20 | 0 |

Travel Time Hole #1→#2 = $t_0 + t_1 + \ldots + t_{n-2} + t_{n-1}$

$\text{Time}_{12341}=$
$40 + 60 + 20 + 50 = 170$

$\text{Time}_{13421}=$
$10 + 20 + 30 + 40 = 100$

**Figure 1.3 An overview of the thesis work**

# Chapter 2
# Literature Review

## 2.1 Introduction

Minimum time trajectory planning for laser drilling process can be broken down into two tasks: trajectory generation and sequence optimization. It is desired to generate a trajectory from one hole to another hole so that its travel time is minimized. Such trajectories should be bounded by the machine's kinematic limits to ensure no saturation of the drives and no excessive vibration. Next is to optimize the hole visiting sequence to minimize the total travel time. With every hole-to-hole travel time known from the trajectory generation, the sequence optimization task can be formulated to a TSP to solve for an optimal or near-optimal sequence which results in the minimum total travel time.

## 2.2 Trajectory Generation

There has been a lot of research in generating minimum time trajectory for both manipulator type robots and machine tools. There are two main approaches in trajectory generation: a dynamic approach and a kinematic approach. In the dynamic approach, the system is often modeled with coupled dynamic equations, such as shown in the following equation, that combine the dynamics, joint forces/torques, or load characteristics for a given trajectory.

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) \tag{2.1}$$

Here $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of actuator torques, $\boldsymbol{q} \in \mathbb{R}^n$ is the vector of joint positions, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix of the manipulator, $\boldsymbol{C}(\boldsymbol{q}) \in \mathbb{R}^{n\times n\times n}$ contains centrifugal and Coriolis force terms, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^n$ represents the vector of gravity terms.

This approach can be found commonly in most of the robotics and manipulator type researches as the dynamic load at the end effector and the external forces such as gravity affect system performance noticeably in robotic arm structures [2], [3], [4], [5]. In generating trajectories, some of the early studies do not consider the continuity of acceleration profile, resulting in a bang-bang or bang-singular-bang type characteristics, which are hard to implement because such trajectories may cause mechanical wear, saturation, and vibration. The need for smooth trajectories is understood and Constantinescu and Croft [6] presented a smooth time-optimal trajectory method by limiting the torque rate of each joint. Bianco and Piazzi [7] obtained minimum time cubic spline trajectory subjected to torque constraints using hybrid genetic/interval algorithm based global optimization approach.

More recently, time-optimal trajectory planning is formulated into a form of convex optimization problem motivated by its advantages on its theoretical and practical strengths: optimal solution in convex optimization represents the global optimal solution and the convex optimization problems can be solved efficiently by mature methods such as interior-point methods or other. In their study, Zhang and Zhao [8] used a convex optimization technique to generate smooth minimum time trajectory while utilizing the maximum machine capabilities. Zhang et al. [9] used the cubic Hermite polynomial to generate a smooth tool path for a point-to-point motion and constructed two-level nested optimization problem that searches for minimum time trajectory for different given point-to-point tool paths. For the confined path tracking problem, Zhang et al. [10] obtained minimum time trajectory under confined jerk, rate of change of the torque, and the voltage by formulating a relaxed convex optimization problem.

On the other hand, a common industrial practice in the machine tool and manufacturing industry is to use the kinematic approach which represents machine drive capabilities with kinematic limits such as velocity, acceleration, and jerk shown in equation (2.2).

$$\begin{aligned} -\boldsymbol{v_{max}} &\leq \dot{\boldsymbol{q}} \leq \boldsymbol{v_{max}} \\ -\boldsymbol{a_{max}} &\leq \ddot{\boldsymbol{q}} \leq \boldsymbol{a_{max}} \\ -\boldsymbol{j_{max}} &\leq \dddot{\boldsymbol{q}} \leq \boldsymbol{j_{max}} \end{aligned} \tag{2.2}$$

This practice enables an easy transfer of machine specification provided by the toolmakers onto trajectory generation. In the kinematic approach, it is highly desired to have continuous acceleration and bounded jerk profiles to avoid undesirable high-frequency content in the reference trajectory that can induce excessive vibration, degrade axis tracking performance leading to poor contouring accuracy, and saturate the actuators. Therefore, different jerk bounded and jerk continuous feedrate planning strategies have been suggested in a lot of research.

Analytical functions can be used in generating such smooth motion profile. Makino and Ohde [11] generated a universal cam curve acceleration profile from combining sinusoidal curves and straight lines to ensure jerk continuous feedrate. Tomita et al. [12] used trigonometric functions to get jerk continuous trajectory. In their work, Erkorkmaz and Altintas [13] developed a quintic spline trajectory generation method with a jerk bounded smooth feedrate having trapezoidal acceleration profile. A dynamic filter such as Finite Impulse Response (FIR) filters can also be utilized to generate a smooth trajectory. By simply convolving a number of FIR filters, higher order polynomial smooth trajectories can be obtained [14], [15].

Another kinematic approach is to transform the feedrate planning problem into an optimization problem, which can be solved using mathematical optimization methods [16], [17], [18], [19]. Quintic spline is used to approximate the position profile for the given starting and ending point and up to eight control points are optimized to yield minimum time jerk bounded smooth trajectory in real time [20]. Sencer et al. [21] approximated the feed profile with cubic B-spline, and its control points are optimized to obtain a time-optimal smooth trajectory. In their research, Kyriakopoulos and Saridis [22] adopted an optimal control problem to minimize the maximum of jerk values of the joints to generate a smooth trajectory. It is worth to mention that in [23], both the total execution time and the integral of squared jerk terms are minimized to yield optimal trajectory.

## 2.3 Sequence Optimization

As mentioned previously, the total travel time for 5-axis percussion laser drilling process comes from the laser nozzle end maneuvering over all the holes in a given visiting sequence. Therefore, great attention is required on optimizing the sequence to minimize the travel time. In fact, the stop-and-go nature of the percussion drilling process resembles a well-known combinatorial optimization topic called Traveling Salesman Problem (TSP). Given a set of cities and the costs of traveling among all the cities, a TSP is to find the minimum cost tour that visits all the cities exactly once [24]. The problem was mathematically formulated in the 1800s and starting in 1950s, it became increasingly popular among mathematicians, computer scientists, and scientists from other fields (Figure 2.1) [25]. TSP falls into a category of NP-hard problem as there is no way of checking if there exists a better solution to a candidate solution in polynomial time. In solving such

NP-hard problems, two approaches are taken: exact algorithms and heuristic or metaheuristic algorithms.

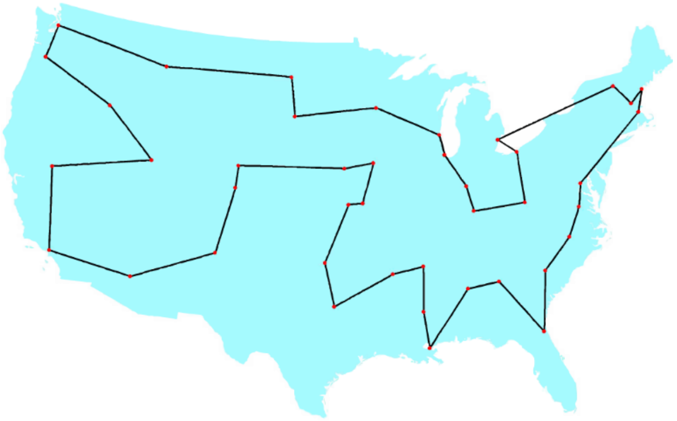


**Figure 2.1 TSP example of an optimal tour of 42 cities in the USA solved in 1954 [26]**

In computer sciences and other optimization fields, exact algorithms are algorithms that are guaranteed to generate optimal solutions. For TSP, the most direct method would be trying out all the possible permutations of the given cities and check which one is the optimal solution. For $n$ number of cities, there are $\frac{(n-1)!}{2}$ feasible solutions to check, which makes this brute force search almost impossible even for a small number of cities. In 1954, Dantzig et al. [27] developed a breakthrough cutting plane method of solving TSP with linear programming. First, relaxed linear programming that can be solved by the simplex method is formulated from the original linear programming. Then, the solution to the relaxed one is checked if it satisfies the original constraint. If not, the linear inequalities that violate, called a cutting plane, are added to the relaxed problem to make it tighter and it is solved again. This process is iterated so that eventually, the solution to the original linear programming is obtained. Although it was successful in solving the 49-city problem [28], it is still not very efficient to solve for larger instances. Applegate et al. [29] presented in 1998 a powerful computer code called Concorde that uses a hybrid method called branch-and-cut to dramatically increase the efficiency. While the problem is recursively split from branching and establishes bounds, the cutting planes are added, hence branch-and-cut, to relax the problem. In 2006, a very large 85,900-city problem is solved using the solver [30]. To date, Concorde TSP solver is considered as a state-of-the-art exact solver.

Despite the guaranteed optimality, the exact solvers still take a relatively long time to solve. Hence, researchers and practitioners tend to use heuristic or metaheuristic algorithms to find optimal or near-optimal solutions as fast as possible. These algorithms include Nearest Neighbor (NN), Lin-Kernighan Helsgaun (LKH), Genetic Algorithm (GA), Ant Colony Optimization (ACO), and so on. Nearest Neighbour algorithm is one of the first algorithms to build a TSP tour. Starting from a random city, it constructs a tour by keeping adding the nearest city to the current city until all the cities have been visited. It is easy to implement and generates a solution quickly, but it is most likely to produce suboptimal solutions due to its greedy nature. Alwis et al. [31] used NN to generate a sequence for automated PCB drilling application. However, their solutions are not optimal, and in fact, it is very much noticeable to human eyes. Oftentimes, NN is used in conjunction with local search optimization methods such as powerful Lin-Kernighan Helsgaun (LKH).

LKH is a more computationally efficient and effective version of Lin-Kernighan (LK) heuristics developed by computer scientists Keld Helsgaun. The original LK heuristics uses a local search technique called k-opt moves (or exchanges). When a feasible tour is given, a k-opt move modifies a tour by replacing k number of edges with different k number of edges in such a way that the result is a cheaper tour [32]. With effective implementation of the k-opt move and other revised criteria, LKH is one of the top heuristics that holds the record for finding the best reported tour for the 1,904,711-city large-scale World TSP [33]. Aciu and Ciocarlie successfully implemented LKH to generate minimum tool path lengths for PCB drilling application [34]. While these problem specific heuristic algorithms produce excellent solutions, a trend has been to utilize metaheuristic algorithms to address more general problem types and for a greater chance of reaching global optimum.

Genetic Algorithm (GA) is based on the evolutionary algorithm, which closely reflects Charles Darwin's natural selection of fittest individuals in reproduction for the next generation [35]. With special parameters such as fittest score assigned and undergoing selection and mating processes, GA can produce promising TSP results such as the best-known-to-date solution for 100,000-city Mona Lisa TSP instance [36]. Zhang and Zhao [37] have solved minimum time drilling problems for a 3-DOF robotic manipulator using GA.

Another popular metaheuristic algorithm is Ant Colony Optimization (ACO). It mimics the route selection strategy of an ant colony. Movement of a single ant seems rather erratic and uncoordinated, however communicating via pheromone trails, the entire group of ants can find an efficient route to food [38]. For TSP, pheromone values are assigned to each edge to indicate the probability of being selected. Ross et al. [39] presented a method of optimizing the drilling path for the CNC machine by using Parallel ACO, a high-performance implementation of ACO.

## 2.4 Conclusion

In this chapter, literature covering the topics of trajectory planning and sequence optimization has been reviewed. The main objective of the research presented in this thesis is the integration of these two topics to generate a time-optimal trajectory that considers the constraints of the drilling process, and the kinematic configuration and drive limits of the 5-axis laser drilling machine with hole visiting sequence optimized to minimize the travel time of the drilling process. However, the applicability of the findings from this research is not limited to the laser drilling process but can be expanded to any multi-point manufacturing process.

# Chapter 3

# Kinematic Model of 5-axis Laser Drilling Machine

## 3.1 Introduction

In general, planning of machining processes for CNC machines are done in Workpiece Coordinates (WCS) since machining occurs on the workpiece. Tool paths are defined relative to the workpiece and depending on the process type, the speed the tool end, known as the feedrate, is defined relative to the workpiece as well. However, actual motions are executed by individual servo drives, of which motions are defined in Machine Coordinates (MCS). Hence, to fully understand the processes, it is desired to transform the job information in WCS to MCS. Such transformation is called kinematic transformation. Depending on the machine configuration of how the individual servo drives are attached to the machine base, the transformations differ from machine to machine. In this chapter, the kinematic transformation of the 5-axs laser drilling machine from the industry partner is presented.

## 3.2 5-axis Laser Drilling Machine Diagram

Based on the product manual of the 5-axis laser drilling machine, the following kinematic diagram is constructed.

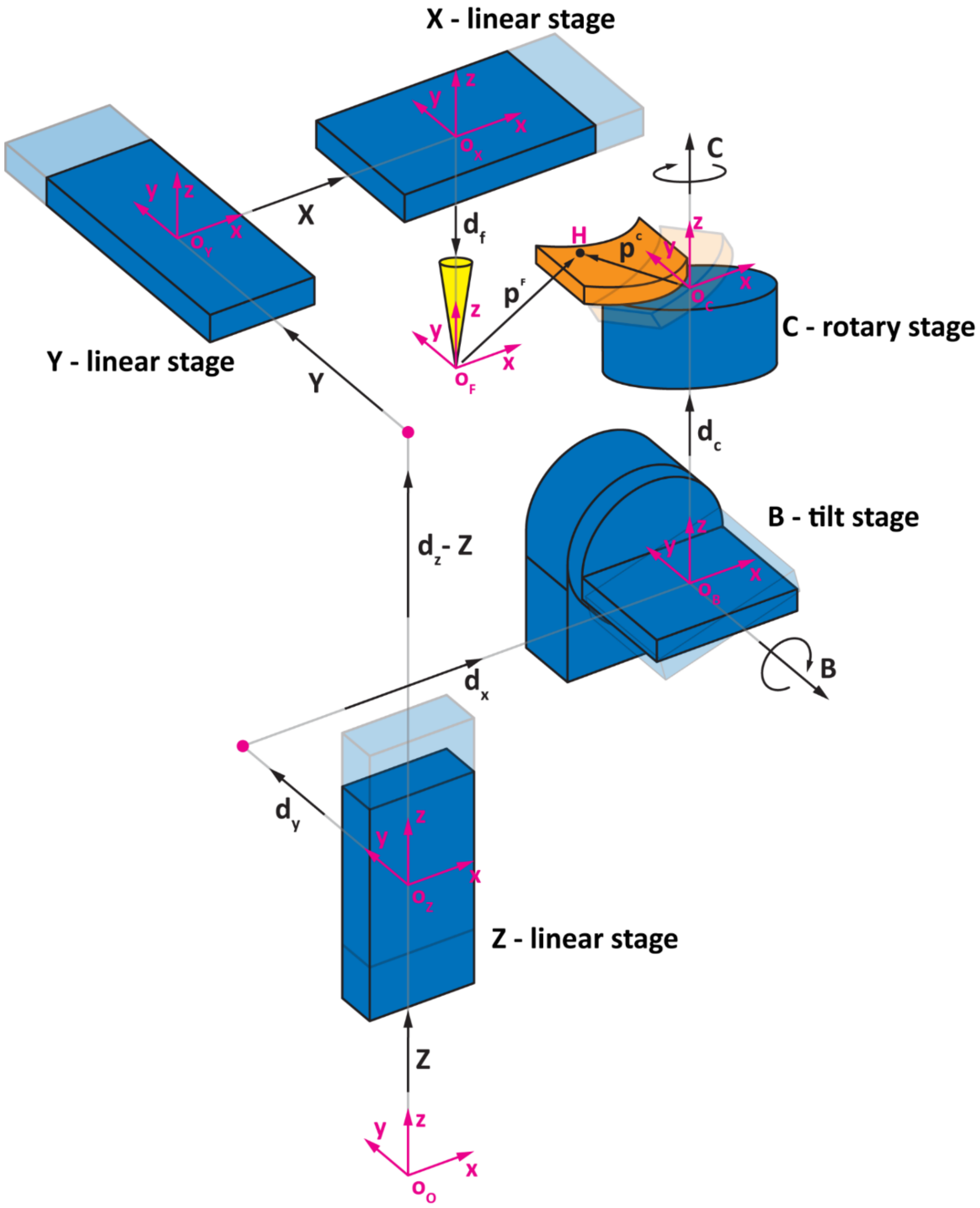


**Figure 3.1 Kinematic diagram of 5-axis laser drilling machine**

The following table lists the coordinate systems (C.S's) considered in the system:

**Table 3.1 List of the coordinate systems considered to construct the kinematic model**

| Coordinates | Description |
|---|---|
| C.S.O ($\boldsymbol{O_O x_O y_O z_O}$) | This frame is fixed to the machine base bottom. Its axes are parallel to the translating joints of the machine |
| C.S.Z ($\boldsymbol{O_Z x_Z y_Z z_Z}$) | This frame is attached to the moving Z stage of the machine. The tilt(B) stage is also attached to the Z stage. $\boldsymbol{x_Z}$ and $\boldsymbol{y_Z}$ axes are parallel to the $\boldsymbol{x_O}$ and $\boldsymbol{y_O}$ axes. |
| C.S.B ($\boldsymbol{O_B x_B y_B z_B}$) | This is the tilt (B) stage coordinate system. $\boldsymbol{y_B}$ is the axis of rotation for tilt stage. $\boldsymbol{O_B}$ is translated from $\boldsymbol{O_z}$ by $\boldsymbol{d_y}$ along $\boldsymbol{y_Z}$ axis. $\boldsymbol{d_y}$ is a constant offset by the machine structure. $\boldsymbol{O_B}$ is rotated by $\boldsymbol{\theta_B}$ along $\boldsymbol{y_z}$ axis. $\boldsymbol{\theta_B}$ is a variable dependent on the tilt motion. |
| C.S.C ($\boldsymbol{O_C x_C y_C z_C}$) | This is the workpiece base frame that sits on the rotary (C) stage. $\boldsymbol{z_C}$ is the axis of rotation for the rotary stage. $\boldsymbol{O_C}$ is translated from $\boldsymbol{O_B}$ by $\boldsymbol{d_C}$ along $\boldsymbol{z_B}$ axis. $\boldsymbol{d_C}$ is a constant offset by the machine structure. $\boldsymbol{O_C}$ is rotated by $\boldsymbol{\theta_C}$ along $\boldsymbol{z_B}$ axis. $\boldsymbol{\theta_C}$ is a variable dependent on the rotary motion. |
| C.S.Y ($\boldsymbol{O_Y x_Y y_Y z_Y}$) | This is the translating Y-stage coordinate system. $\boldsymbol{O_Y}$ is translated from $\boldsymbol{O_O}$ by $\boldsymbol{Y}$ along $\boldsymbol{y_O}$ axis and by $\boldsymbol{d_Z}$ along $\boldsymbol{z_O}$ axis. $\boldsymbol{Y}$ is a variable of Y moving motion and $\boldsymbol{d_Z}$ is a constant offset by the machine structure. |
| C.S.X ($\boldsymbol{O_X x_X y_X z_X}$) | This is the translating X-stage coordinate system. $\boldsymbol{O_X}$ is translated from $\boldsymbol{O_Y}$ by $\boldsymbol{X}$ along $\boldsymbol{x_Y}$ axis. $\boldsymbol{X}$ is a variable of X moving motion. |
| C.S.F ($\boldsymbol{O_F x_F y_F z_F}$) | This frame is attached to the laser focal point where the laser is focused and the drilling occurs. $\boldsymbol{O_F}$ is translated from $\boldsymbol{O_X}$ by $\boldsymbol{d_f}$ along $\boldsymbol{z_X}$ axis. $\boldsymbol{d_f}$ is a constant offset by the machine structure. |

## 3.3 Coordinate Transformation

With TRAORI mode (explained in more detail in Chapter 4), the position of each hole is expressed with respect to WCS, which is defined as C.S.C from the above illustration, and the orientation of each hole is directly expressed as $B$ and $C$ angles. When the NC code is read in, the CNC transforms WCS values to MCS values automatically. The measurements from the machine are in MCS values, therefore it is necessary to study the kinematic transformation of the coordinates to transform trajectory and hole pattern information between them.

In this thesis, coordinate transformation is expressed using homogeneous transformation matrices. Homogeneous transformation matrix is a $4 \times 4$ matrix containing the rotation matrix and the translation vector. For example, consider an arbitrary point in two coordinate systems shown in the below figure.

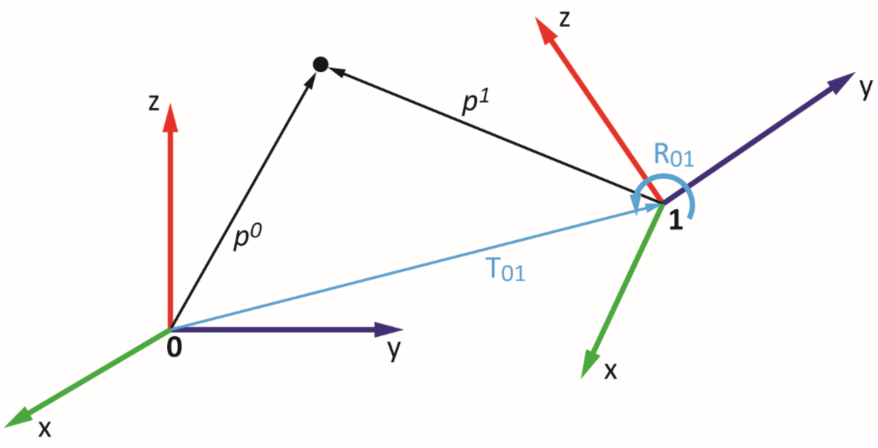


**Figure 3.2 Visualization of coordinate transformation between two coordinate systems**

The point can be expressed by two vectors, $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$, and the coordinate transformation between these vectors can be represented by using a homogeneous transformation matrix, $\boldsymbol{H}_{01}$, in the following equation.

$$\boldsymbol{p}^0 = \boldsymbol{H}_{01}\boldsymbol{p}^1 = \begin{bmatrix} & \boldsymbol{R}_{01} & & \boldsymbol{T}_{01} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} p_x^1 \\ p_y^1 \\ p_z^1 \\ 1 \end{pmatrix} \tag{3.1}$$

where $\boldsymbol{R}_{01}$ is a $3 \times 3$ rotation matrix of frame 1 with respect to frame 0 and $\boldsymbol{T}_{01}$ is a $3 \times 1$ translation vector from the origin of frame 0 to frame 1. With the same transformation matrix, the reverse can be represented as well by taking an inverse of the matrix.

$$\boldsymbol{p}^1 = (\boldsymbol{H}_{01})^{-1}\boldsymbol{p}^0 = \begin{bmatrix} & \boldsymbol{R}_{01} & & \boldsymbol{T}_{01} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{pmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \\ 1 \end{pmatrix} = \begin{bmatrix} & \boldsymbol{R}_{10} & & \boldsymbol{T}_{10} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \\ 1 \end{pmatrix} \tag{3.2}$$

One of the benefits of using the homogeneous transformation matrices is that a sequence of coordinate transformation can be represented by the products of the individual homogeneous transformation matrix.

Since $\{B\ C\}$ values are directly given in TRAORI mode, $\{X\ Y\ Z\}$ are required to understand the motion in MCS. To find the corresponding coordinate transformation between MCS and WCS, the transformation from the workpiece base frame (C.S.C) to the tool end frame, in this case at the laser focal point (C.S.F), is defined first by the following equation.

$$\boldsymbol{H}_{FC} = (\boldsymbol{H}_{OZ}\boldsymbol{H}_{ZY}\boldsymbol{H}_{YX}\boldsymbol{H}_{XF})^{-1}\boldsymbol{H}_{ZB}\boldsymbol{H}_{BC} \tag{3.3}$$

where each homogeneous transformation matrix is shown below with the following representation, $S_B = sin\theta_B, C_B = cos\theta_B, S_C = sin\theta_C, C_C = cos\theta_C$.

$$\boldsymbol{H}_{OZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{H}_{ZY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & d_z - Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H}_{YX} = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{H}_{XF} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H}_{ZB} = \begin{bmatrix} C_B & 0 & -S_B & d_x \\ 0 & 1 & 0 & d_y \\ S_B & 0 & C_B & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{H}_{BC} = \begin{bmatrix} C_C & S_C & 0 & 0 \\ -S_C & C_C & 0 & 0 \\ 0 & 0 & 1 & d_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$H_{fc}$ can be calculated by multiplying the above matrices.

$$\boldsymbol{H}_{OY} = \boldsymbol{H}_{OZ}\boldsymbol{H}_{ZY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & d_z - Z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H}_{OX} = \boldsymbol{H}_{OY}\boldsymbol{H}_{YX} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H}_{Of} = \boldsymbol{H}_{OX}\boldsymbol{H}_{XF} = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_f \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & d_z - d_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H}_{OB} = \boldsymbol{H}_{OZ}\boldsymbol{H}_{ZB} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_B & 0 & -S_B & d_x \\ 0 & 1 & 0 & d_y \\ S_B & 0 & C_B & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_B & 0 & -S_B & d_x \\ 0 & 1 & 0 & d_y \\ S_B & 0 & C_B & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H}_{OC} = \boldsymbol{H}_{OB}\boldsymbol{H}_{BC} = \begin{bmatrix} C_B & 0 & -S_B & d_x \\ 0 & 1 & 0 & d_y \\ S_B & 0 & C_B & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_C & S_C & 0 & 0 \\ -S_C & C_C & 0 & 0 \\ 0 & 0 & 1 & d_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_B C_C & C_B S_C & -S_B & -S_B d_c + d_x \\ -S_C & C_C & 0 & d_y \\ S_B C_C & S_B S_C & C_B & C_B d_c + Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H}_{FC} = (\boldsymbol{H}_{OF})^{-1}\boldsymbol{H}_{OC} = \begin{bmatrix} C_B C_C & C_B S_C & -S_B & -S_B d_c + d_x - X \\ -S_C & C_C & 0 & d_y - Y \\ S_B C_C & S_B S_C & C_B & C_B d_c + Z + d_f - d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With this transformation matrix, it is possible to express the coordinate transformation of each hole pattern from the WCS ($\boldsymbol{p}^C$) to the laser focal coordinate system as follows

$$\boldsymbol{p}^F = \boldsymbol{H}_{FC}\boldsymbol{p}^C \tag{3.4}$$

In laser drilling, the drilling occurs at the laser focal points. In other words, the hole location with respect to the laser focal coordinate system, $\boldsymbol{p}^F$, is zero. Using this knowledge, the above equation can be rearranged and expanded further.

$$\boldsymbol{p}^C = (\boldsymbol{H}_{FC})^{-1}\boldsymbol{p}^F$$

$$\begin{pmatrix} p_x^C \\ p_y^C \\ p_z^C \\ 1 \end{pmatrix} = \begin{bmatrix} C_B C_C & C_B S_C & -S_B & -S_B d_c + d_x - X \\ -S_C & C_C & 0 & d_y - Y \\ S_B C_C & S_B S_C & C_B & C_B d_c + Z + d_f - d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.5}$$

$$\begin{pmatrix} p_x^C \\ p_y^C \\ p_z^C \\ 1 \end{pmatrix} = \begin{pmatrix} C_B C_C (X - d_x) - S_C (Y - d_Y) + S_B C_C (-Z + d_z - d_f) \\ C_B S_C (X - d_x) - C_C \left(Y - d_y\right) + S_B S_C (-Z + d_z - d_f) \\ -S_B (X - d_x) + C_B \left(-Z + d_z - d_f\right) - d_C \\ 1 \end{pmatrix}$$

which is then simplified to

$$\begin{pmatrix} p_x^C \\ p_y^C \\ p_z^C \end{pmatrix} = \begin{bmatrix} C_B C_C & -S_C & S_B C_C \\ C_B S_C & C_C & S_B S_C \\ -S_B & 0 & C_B \end{bmatrix} \begin{pmatrix} X - d_x \\ Y - d_y \\ -Z + d_z - d_f \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ d_c \end{pmatrix} \tag{3.6}$$

Finally, a small adjustment is made to the equation since in TRAORI mode, the direction of Z motion is reversed, that lifting up the Z stage decreases the value of Z measured in CNC.

$$\begin{pmatrix} p_x^C \\ p_y^C \\ p_z^C \end{pmatrix} = \begin{bmatrix} C_B C_C & -S_C & S_B C_C \\ C_B S_C & C_C & S_B S_C \\ -S_B & 0 & C_B \end{bmatrix} \begin{pmatrix} X - d_x \\ Y - d_y \\ Z - (d_z - d_f) \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ d_c \end{pmatrix} \tag{3.7}$$

For WCS to MCS transformation, the above equation is simply reversed.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} C_B C_C & C_B S_C & -S_B \\ -S_C & C_C & 0 \\ S_B C_C & S_B S_C & C_B \end{bmatrix} \begin{pmatrix} p_x^C \\ p_y^C \\ p_z^C + d_c \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \\ d_z - d_f \end{pmatrix} \tag{3.8}$$

Note that the matrix is in the form of a rotation matrix. In fact, it is the inverse of $\boldsymbol{R}_{FC}$, which is the rotation matrix of $\boldsymbol{H}_{FC}$. Using the above equation, each hole location with respect to the WCS is calculated from the MCS vector $\{X\ Y\ Z\}$. The constants $\{d_x\ d_y\ d_z\ d_f\ d_c\}$ are obtained from the machine configuration.

To verify the above solution, the hole patterns constructed directly from the NC code and the hole patterns constructed from the measured MCS data using the same NC code are compared. Figure 3.3 shows the drill hole locations in MCS from servo $\{X\ Y\ Z\}$ positions executed by the machine CNC and the desired hole locations in WCS programmed in the NC code. It is clear that the hole locations do not match prior to kinematic transformation.

Once the above kinematic transformation is applied to the measured $\{X\ Y\ Z\}$ data, the expected hole pattern is produced as shown in Figure 3.4 proving that the transformation is correctly done.

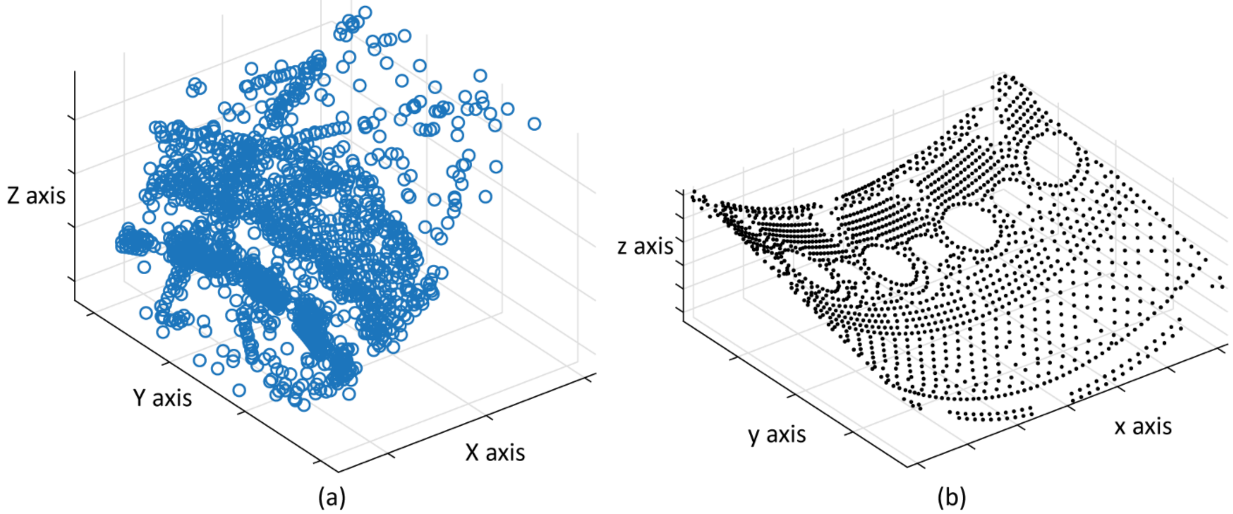


**Figure 3.3 (a) hole locations in MCS measured from the machine servo drives (b) desired hole locations in WCS programmed in NC code**

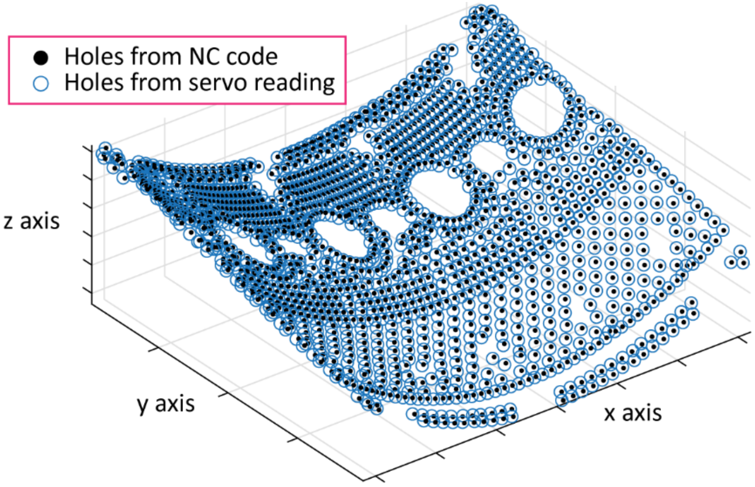


**Figure 3.4 Reconstructed hole locations from the servo reading overlaid on top of the desired hole locations**

## 3.4 Conclusion

In this chapter, a specific kinematic model of 5-axis laser drilling machine is presented to transform the given drilling process information defined in Workpiece Coordinates into Machine Coordinates. In this thesis, homogeneous transformation matrices are used to express the coordinate transformation and are validated by overlaying the experimentally measured hole locations on top of the defined hole locations from the NC code. It is necessary to acquire this transformation between WCS and MCS for trajectory planning, since the actual motion is realized by the individual axis servo drives under the given kinematic limits. The following chapter delivers the details of trajectory planning methods.

# Chapter 4
# Hole-to-hole Trajectory Planning

## 4.1 Introduction

The essences of minimum time drilling process, in the perspective of trajectory, are execution and estimation. It is required to travel from one hole to another as fast as possible and its motion time needs to be correctly estimated so that a more realistic result can be obtained when used in the sequence optimization step. Percussion laser drilling is a type of machining process where the motion type is classified as point-to-point (PTP) or positioning. Unlike contouring, where the cutter tool path is an essential process objective, the main objective of PTP is to position the tool end at the desired location and orientation, which makes the path irrelevant. To accomplish fast PTP motion, most CNCs use the motion type command called rapid traverse (G00 in G-code) which maximizes the velocity under the defined machine limits along with other kinematic limits such as acceleration and jerk limits. The common resultant motion profiles, often referred by its shape, are trapezoidal velocity profile if jerk value is not bounded, and s-curve velocity profile if jerk value is bounded, with preferences given to the s-curve velocity profile for its smoothness.

The industry partner, in their current NC code programming strategy, also uses G00 during the percussion drilling process to realize fast hole-to-hole motion. In conjunction with G00, they use the CNC controller's 5-axis transformation with tool orientation tactic called TRAORI. Typical planning of the drilling process is done in Workpiece Coordinates (task spatial coordinates) and CAM software generates corresponding G-code in WCS. It is TRAORI that performs the correct conversion of the G-code to the equivalent Machine Coordinates (joint spatial coordinates) values to perform the desired tasks.

To be practically beneficial to the industry partner and their job planning strategy, a custom trajectory is not developed in this research. Instead, CNC controller's G00 + TRAORI trajectory is closely studied and four trajectory planning strategies are discussed in this chapter to approximate the controller's strategy for the sake of correct estimation of motion time. The overview of the four strategies are illustrated in the figure below.

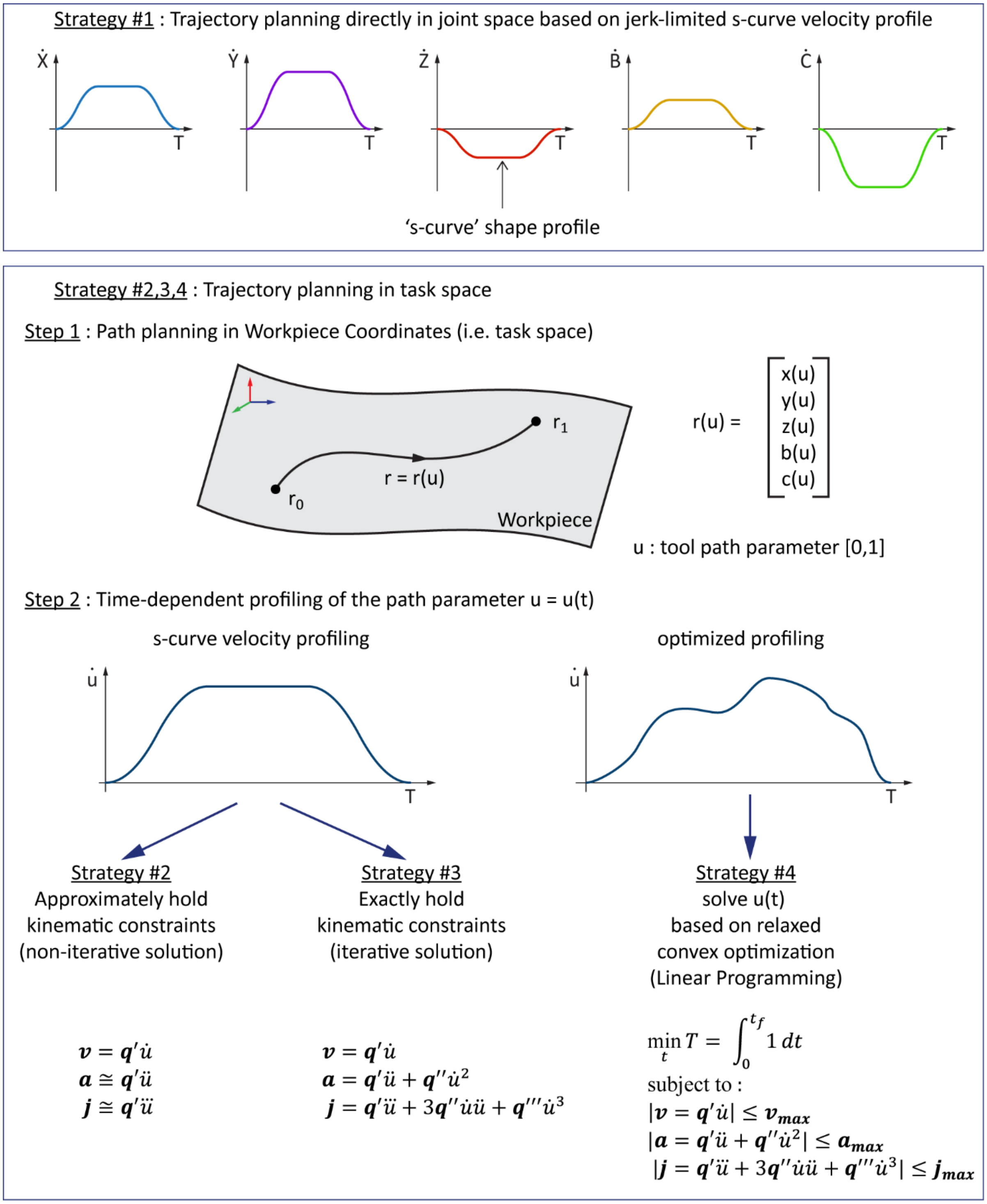

Strategy #1 : Trajectory planning directly in joint space based on jerk-limited s-curve velocity profile
$\dot{X}$
$\dot{Y}$
$\dot{Z}$
$\dot{B}$
$\dot{C}$
T
T
T
T
T
's-curve' shape profile
Strategy #2,3,4 : Trajectory planning in task space
Step 1 : Path planning in Workpiece Coordinates (i.e. task space)
$r_1$
r = r(u)
$r_0$
Workpiece
r(u) = $\begin{bmatrix} x(u) \\ y(u) \\ z(u) \\ b(u) \\ c(u) \end{bmatrix}$
u : tool path parameter [0,1]
Step 2 : Time-dependent profiling of the path parameter u = u(t)
s-curve velocity profiling
optimized profiling
$\dot{u}$
T
$\dot{u}$
T
Strategy #2
Approximately hold
kinematic constraints
(non-iterative solution)
Strategy #3
Exactly hold
kinematic constraints
(iterative solution)
Strategy #4
solve u(t)
based on relaxed
convex optimization
(Linear Programming)
$\boldsymbol{v} = \boldsymbol{q}'\dot{u}$
$\boldsymbol{a} \cong \boldsymbol{q}'\ddot{u}$
$\boldsymbol{j} \cong \boldsymbol{q}'\dddot{u}$
$\boldsymbol{v} = \boldsymbol{q}'\dot{u}$
$\boldsymbol{a} = \boldsymbol{q}'\ddot{u} + \boldsymbol{q}''\dot{u}^2$
$\boldsymbol{j} = \boldsymbol{q}'\dddot{u} + 3\boldsymbol{q}''\dot{u}\ddot{u} + \boldsymbol{q}'''\dot{u}^3$
$\min_t T = \int_0^{t_f} 1\,dt$
subject to :
$|\boldsymbol{v} = \boldsymbol{q}'\dot{u}| \leq \boldsymbol{v_{max}}$
$|\boldsymbol{a} = \boldsymbol{q}'\ddot{u} + \boldsymbol{q}''\dot{u}^2| \leq \boldsymbol{a_{max}}$
$|\boldsymbol{j} = \boldsymbol{q}'\dddot{u} + 3\boldsymbol{q}''\dot{u}\ddot{u} + \boldsymbol{q}'''\dot{u}^3| \leq \boldsymbol{j_{max}}$

**Figure 4.1 Overview of four trajectory planning strategies**

## 4.2 Time-optimal Velocity Profile

A time-optimal trajectory would fully utilize the maximum velocity during traverse resulting in a rectangular profile. However, this results in infinite spikes in acceleration, which is not feasible to implement (Figure 4.2).

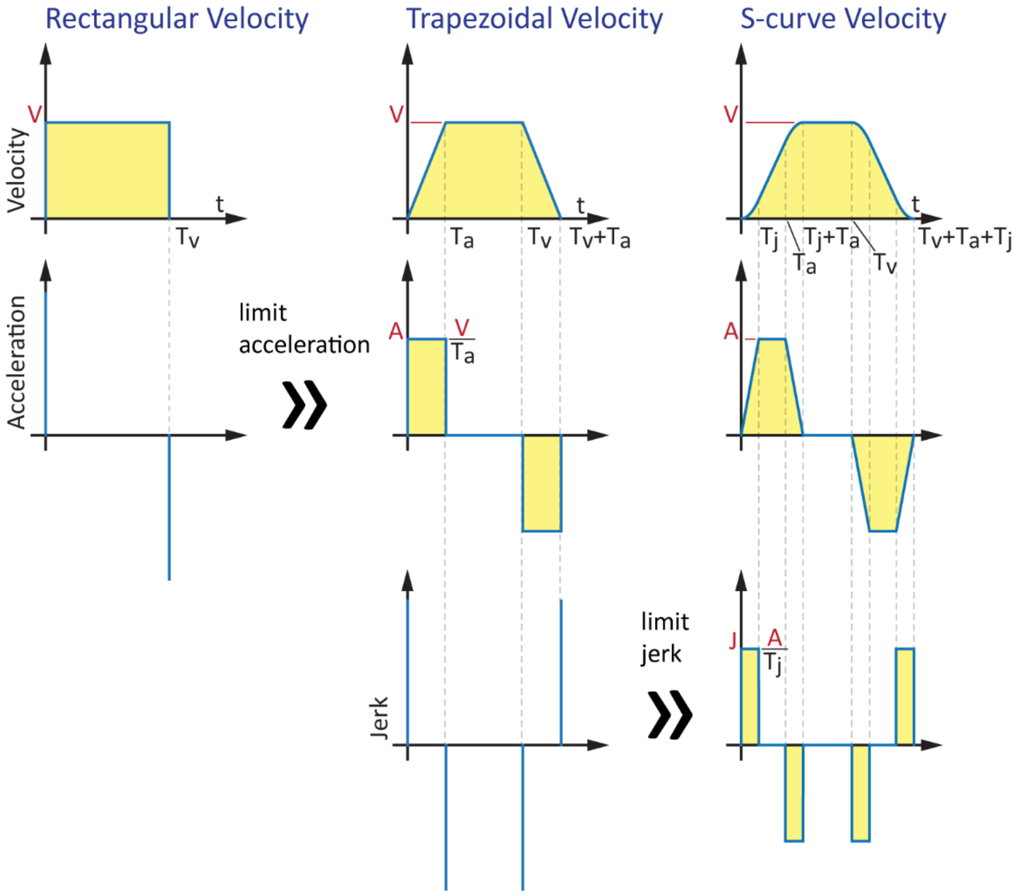


**Figure 4.2 Evolution of time-optimal velocity profile**

As a remedy, acceleration is limited to a finite value that results in a trapezoidal velocity profile. This is not favorable either since the discontinuities in acceleration cause extreme jerk, which induces high-frequency content in the reference signal that can result in excessive vibration of the machine tool structure. It can also cause actuator saturation by demanding trajectories beyond their functional limits. As a result of saturation, a deviation from the desired trajectory could happen meaning that part manufacturing tolerances could be violated. Thus, allowing the jerk to be finite, the piecewise 2nd order polynomial velocity profile, widely referred to as s-curve profile, is obtained (Figure 4.3).

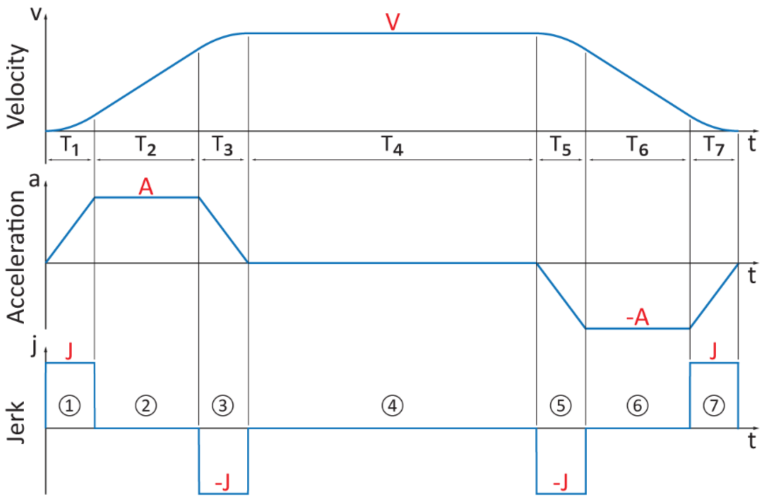


**Figure 4.3 S-curve velocity profile**

The profile is composed of seven segments. Segments 1, 3, 5, and 7 have accelerations with linear profiles whose slopes are predetermined by the jerk limit $J$. During segment 2 and 6, the velocity changes with the constant acceleration, $A$. Then, the velocity is kept constant at its maximum, $V$, during segment 4. The duration for each segment is determined by the kinematic limits. $T_1 = T_3$ ($T_5 = T_7$) is the time it takes for the machine to reach its maximum acceleration, $T_2 = T_6$ is the time required to reach the maximum velocity if the desired velocity is not reached during segment 1 and 3 (5 and 6). $T_4$ is the extra time that is required if the desired displacement is not met during segment 1, 2, 3, 5, 6, and 7 combined.

Two methods can be used to obtain the profile. By convolving the Finite Impulse Response (FIR) filter twice with a rectangular velocity profile, the s-curve velocity profile is obtained [2], [6]. Another way is to derive a set of analytical equations that represent the polynomial curve for each section of the profile. In this research, the analytic method described by Erkorkmaz and Altintas [12] and Alzaydi [40] is used to construct the s-curve velocity profile for the given machine kinematic limits.

The s-curve profile can be broken down to three phases: accelerating phase, coasting with constant velocity phase, and decelerating phase. Depending on the kinematic limits and the desired travel length, the coasting phase may not exist, leaving only the accelerating and decelerating phases to achieve the desired travel distance. Regardless of this, the accelerating/decelerating phases can have two different shape profiles based on the $V$, $A$, and $J$ relationship. Combining these two, the s-curve motion can be classified into four different cases illustrated in the below figure.

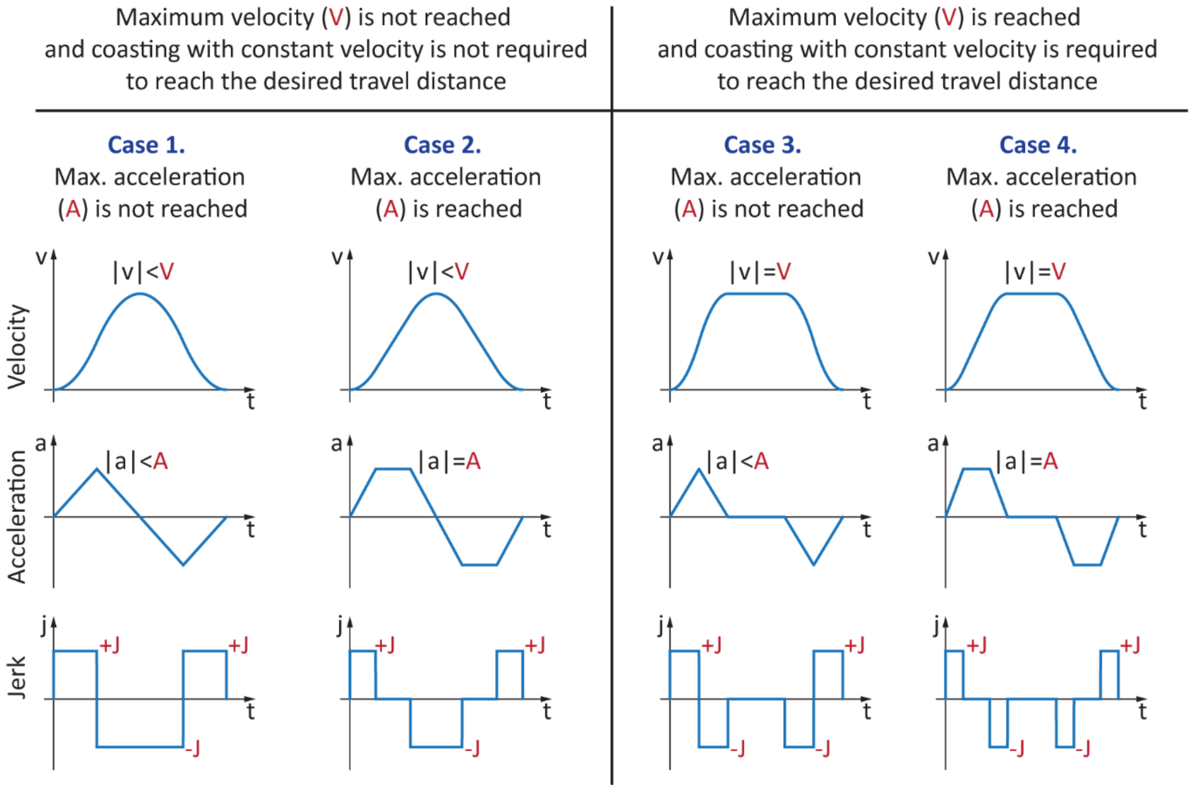


**Figure 4.4 Illustration of 4 cases of s-curve motion profiles**

CASE 1. If $V \leq \frac{A^2}{J}$, and the acceleration and deceleration distance to reach the maximum velocity is greater than the total distance, $l_{acc} > L$:

In this case, the velocity never reaches its limit and the entire motion is employed for acceleration and deceleration phases only. Therefore, the velocity profile does not have a constant region putting $T_4 = 0$. The maximum acceleration/deceleration is not reached either, creating a triangular profile. This sets $T_2 = T_6 = 0$.

CASE 2. If $V > \frac{A^2}{J}$ and $l_{acc} > L$:

Same as case 1, the maximum velocity is never reached putting $T_4 = 0$. While it is accelerating, it reaches the acceleration limit, creating the constant acceleration region. For this reason, a trapezoidal shape profile is achieved in acceleration.

CASE 3. If $V \leq \frac{A^2}{J}$ and $l_{acc} \leq L$:

In this case, the desired distance is greater than or equal to the distance it needs to fully accelerate, the extra distance is covered by coasting at the constant velocity, setting $T_4 \neq 0$. The maximum acceleration is not reached or just reached to increase the velocity to its maximum creating the triangular acceleration profiles. Hence, $T_2 = T_6 = 0$ in this case.

CASE 4. If $V > \frac{A^2}{J}$ and $l_{acc} \leq L$:

Same as case 3, it requires the coasting region to achieve the desired travel distance, setting $T_4 \neq 0$. The acceleration is allowed to reach the maximum in this case and capped by it for a non-zero period for the velocity to reach its maximum. Hence, $T_2 = T_6 \neq 0$.

In this thesis, case 1 and case 2 s-curve motions are referred to as short trajectories, and case 3 and case 4 s-curve motions as long trajectories.

## 4.3 Strategy #1: Trajectory Planning with S-curve Velocity Profiling in Joint Space

As a preliminary and fundamental study for smooth hole-to-hole motion, trajectory planning is first taken in the machine joint space. Since the actual motion control occurs on the individual joint, direct planning of smooth trajectories in the joint space is beneficial. Also, for multi-axis application, singularities that could arise from inverse kinematics can be avoided. However, it is difficult to obtain the desired tool end path due to the non-linearities introduced from the transformation between the Cartesian space and the joint space. Nonetheless, for pure PTP motion, this method would generate the fastest motion compared to the other methods presented later in this chapter. In CNC, this is equivalent to using G00 without TRAORI.

Given starting and ending points for each axis, the travel distances are calculated and with the known velocity, acceleration and jerk limits for each axis, the corresponding s-curve velocity profiles are generated. To verify if this s-curve profile is what the CNC controller uses in its rapid traverse motion, the estimated rapid single axis motion is overlaid on top of the single axis G00 commanded reference trajectory measured from the machine in Figure 4.5.

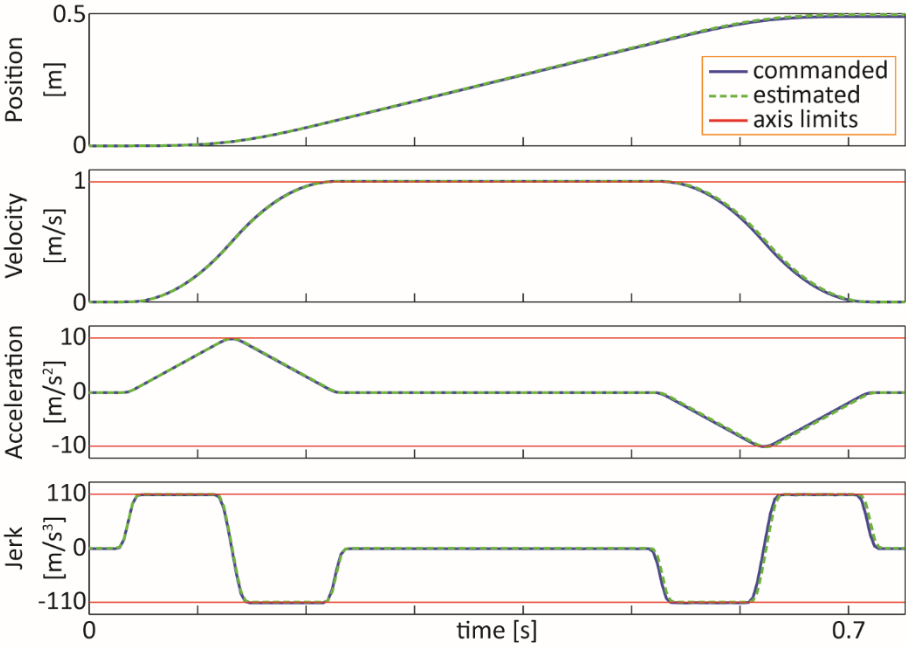


**Figure 4.5 Commanded and estimated G00 trajectories for single axis**

The commanded reference position trajectory is measured at every 4ms and numerically differentiated to generate corresponding velocity, acceleration and jerk profile. The estimated trajectory is generated at 4ms rate as well and numerically differentiated at the same rate to match the loss of shape from the numerical differentiation. It is shown that both trajectory profiles synchronize greatly indicating that the estimated trajectory planning strategy accurately reflects the controller's strategy.

Since the trajectory is planned individually, the travel time for each axis could be different depending on its travel distance and kinematic capacity. By taking the travel time from the slowest axis as an overall motion time, the kinematic limits of faster axes are reduced to yield more relaxed profiles. The result is a synchronous motion that all of the axes start and finish their traverses at the same time (Figure 4.6).

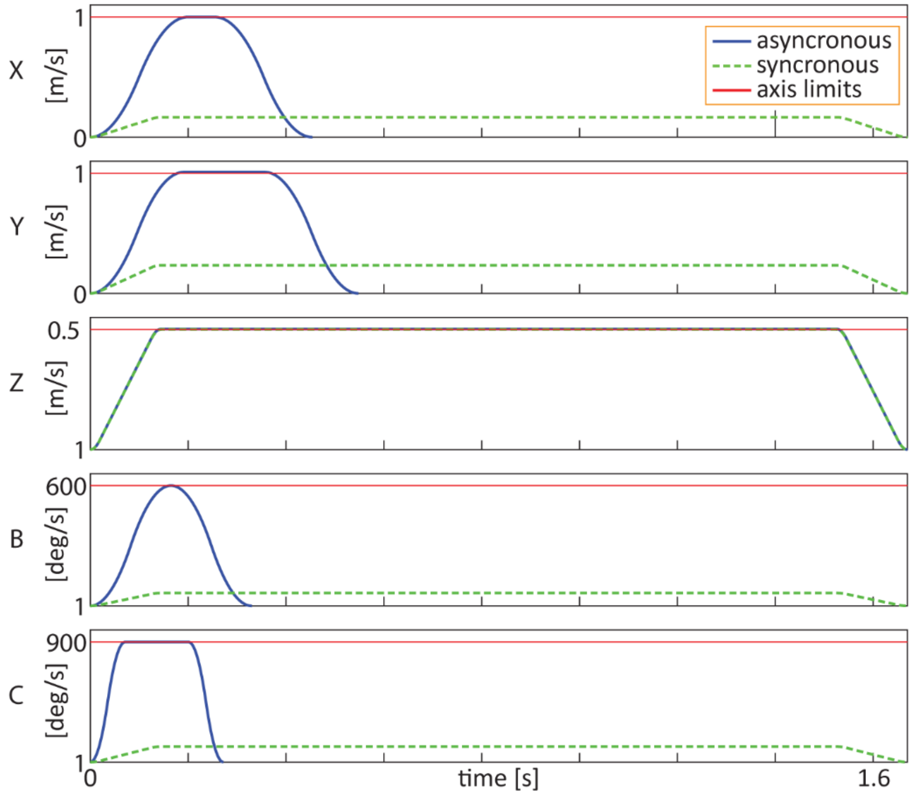
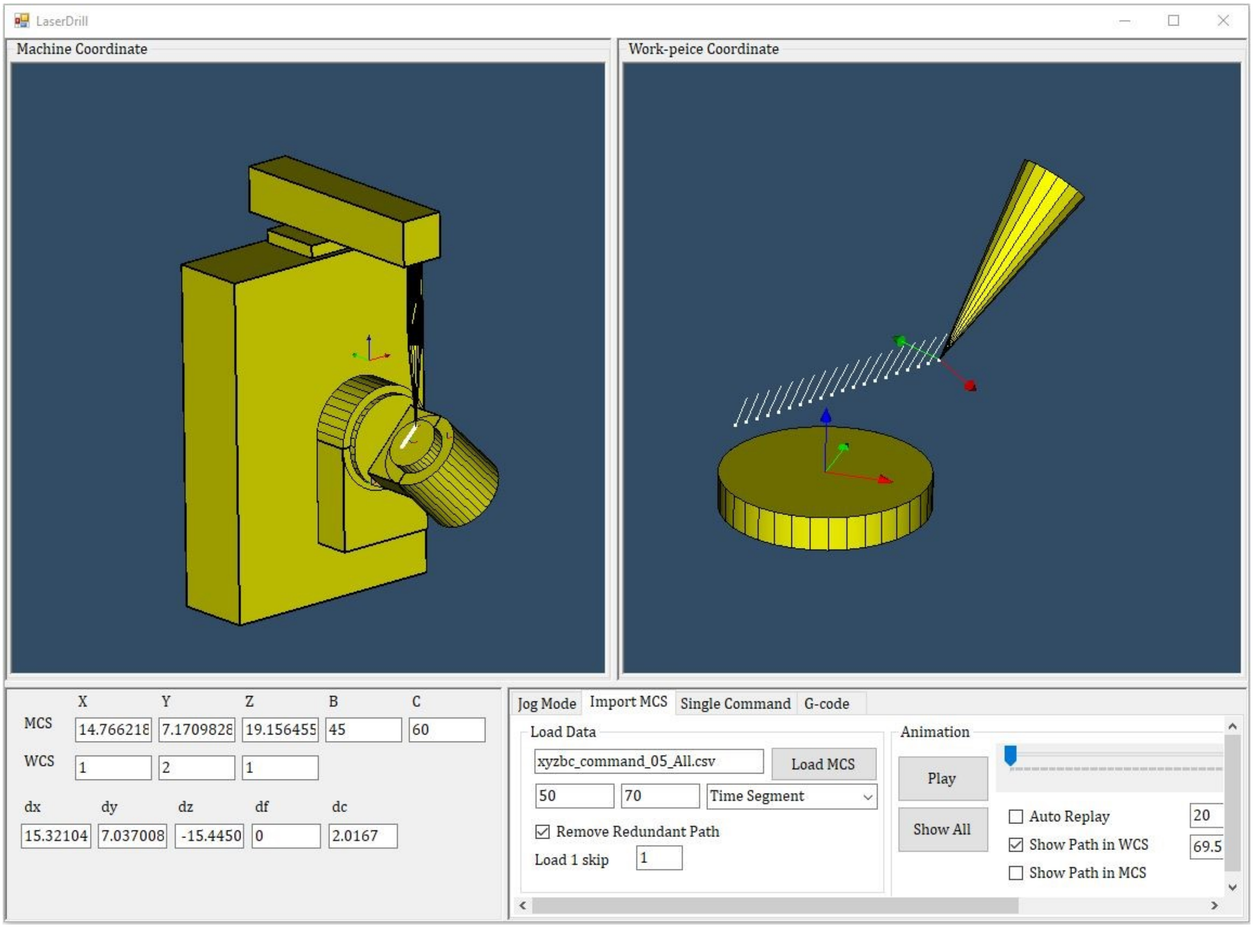


**Figure 4.6 Non-synchronized velocity profile vs. synchronized velocity profile**

In this method, since the trajectory is modeled as PTP motion, the feedrate is planned first then the corresponding tool path information is acquired later by the direct kinematic transformation of the resultant position profile of each axis. Hence, although it is time optimal within the machine's kinematic limits, it is not directly applied due to its path being not known until the end, which causes difficulties in planning ahead for collisions among the tool, the parts, and the fixture. Therefore, in practice, G00 alone is not used, but an orientation retaining function called TRAORI is used together to keep the tool path straight in WCS.

## 4.4 Trajectory Planning in Task Space

To fully address the controller's G00 + TRAORI trajectory generation strategy, some task space planning methods are presented. According to the CNC controller manual, PTP motion associated with TRAORI in task space is generated by linear interpolation, meaning that path between the starting and end points is straight. This is also verified by mapping the measured data onto the virtual workspace via in-house 3D visualization software using the Visualization Toolkit (VTK).



|     | X | Y | Z | B | C |
|-----|---|---|---|---|---|
| MCS | 14.766218 | 7.1709828 | 19.156455 | 45 | 60 |
| WCS | 1 | 2 | 1 |  |  |

| dx | dy | dz | df | dc |
|----|----|----|----|----|
| 15.32104 | 7.037008 | -15.4450 | 0 | 2.0167 |

Load Data

xyzbc_command_05_All.csv

50 70 Time Segment

☑ Remove Redundant Path

Load 1 skip 1

Animation

☐ Auto Replay 20

☑ Show Path in WCS 69.5

☐ Show Path in MCS

**Figure 4.7 5-axis laser drilling motion simulator developed in-house**

Hence the following task space planning strategies have the tool path predefined as a straight line in Workpiece Coordinates and different approaches of assigning the timing law onto it are presented. To distinguish the working coordinates, the lowercase letters $(x, y, z, b, c)$ are used for WCS and the uppercase letters $(X, Y, Z, B, C)$ for MCS. The development of task space trajectory generation strategies (Section 4.4.1.1) is the result of close collaboration with PhD student Chia-Pei Wang from the same research group.

### 4.4.1 S-curve Velocity Profiling

It is assumed that the controller applies the same s-curve generating technique for the velocity planning in task space. The difference to the previous joint space s-curve planning is that the s-curve motion profile is now applied in the task space, resulting in joint space motion that is not an s-curve. In task space planning, the tool path is defined in WCS and the timing law is designed to complete the traverse. Given the starting and ending points $\boldsymbol{p}_0$ & $\boldsymbol{p}_1$ in WCS as follows:

$$\boldsymbol{p}_0 = (x_0, y_0, z_0, \theta_{b_0}, \theta_{c_0})$$
$$\boldsymbol{p}_1 = (x_1, y_1, z_1, \theta_{b_1}, \theta_{c_1}) \tag{4.1}$$

then the tool path from $\boldsymbol{p}_0$ to $\boldsymbol{p}_1$ is defined as

$$\boldsymbol{s}(u) = \begin{Bmatrix} x(u) \\ y(u) \\ z(u) \end{Bmatrix} \tag{4.2}$$

and the orientation vector $\boldsymbol{v}(u)$, based on the machine configuration, is defined as follows:

$$\boldsymbol{v}(u) = \begin{Bmatrix} v_x(u) \\ v_y(u) \\ v_z(u) \end{Bmatrix} = \begin{Bmatrix} sin\theta_b(u)cos\theta_c(u) \\ sin\theta_b(u)sin\theta_c(u) \\ cos\theta_b(u) \end{Bmatrix} \tag{4.3}$$

where $u$ is the path parameter [0,1].

In TRAORI mode, the tooltip path is a straight line and the change of the orientation vector from the initial point to the end point is constant with respect to the tool path. This linear behavior of the tooltip position can be expressed in terms of the path parameter $u$ by the following equations:

$$\boldsymbol{s}(u) = \begin{Bmatrix} x(u) \\ y(u) \\ z(u) \end{Bmatrix} = \begin{Bmatrix} \Delta xu \\ \Delta yu \\ \Delta zu \end{Bmatrix} \tag{4.4}$$

where $\Delta x, \Delta y, \Delta z$ are the Euclidean distances for each axis in WCS and calculated by $x_1 - x_0$, $y_1 - y_0$, $z_1 - z_0$ respectively.

To express the rotation of the orientation vector, Rodrigues' rotation formula is used. This formula represents the three-dimensional rotation for a rotating vector with the axis of rotation and the corresponding rotation angle (Figure 4.8).

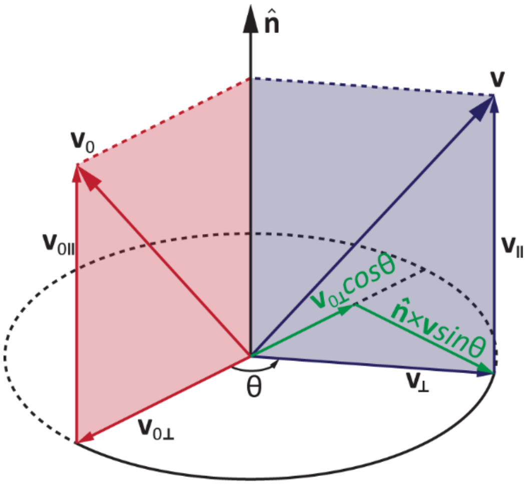


**Figure 4.8 Rodrigues' rotation formula visualization of a rotating vector $\boldsymbol{v}$ by an angle $\boldsymbol{\theta}$ around vector $\hat{\boldsymbol{n}}$ as the axis of rotation**

The orientation vectors for the starting and end points, $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$, can be expressed as follows:

$$\begin{aligned} \boldsymbol{v}_0 &= \left(sin\theta_{b_0}cos\theta_{c_0}\ ,\ sin\theta_{b_0}sin\theta_{c_0}\ ,\ cos\theta_{b_0}\right) \\ \boldsymbol{v}_1 &= \left(sin\theta_{b_1}cos\theta_{c_1}\ ,\ sin\theta_{b_1}sin\theta_{c_1}\ ,\ cos\theta_{b_1}\right) \end{aligned} \qquad \textbf{(4.5)}$$

The unit vector, $\hat{\boldsymbol{n}}$, that is an axis of rotation about which the intermediate vector $\boldsymbol{v}$ rotates as the same normal vector to $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ is defined as follows

$$\hat{\boldsymbol{n}} = \hat{\boldsymbol{n}}\left(\theta_{b_0},\theta_{c_0},\theta_{b_1},\theta_{c_1}\right) = \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} = \frac{\boldsymbol{v}_0 \times \boldsymbol{v}_1}{|\boldsymbol{v}_0 \times \boldsymbol{v}_1|}$$

$$\hat{\boldsymbol{n}} = \begin{Bmatrix} sin\theta_{b_0}sin\theta_{c_0}cos\theta_{b_1} - cos\theta_{b_0}sin\theta_{b_1}sin\theta_{c_1} \\ cos\theta_{b_0}sin\theta_{b_1}cos\theta_{c_1} - sin\theta_{b_0}cos\theta_{c_0}cos\theta_{b_1} \\ sin\theta_{b_0}cos\theta_{c_0}sin\theta_{b_1}sin\theta_{c_1} - sin\theta_{b_0}sin\theta_{c_0}sin\theta_{b_1}cos\theta_{c_1} \end{Bmatrix} / |\boldsymbol{v}_0 \times \boldsymbol{v}_1| \qquad \textbf{(4.6)}$$

With the overall rotation angle from $\boldsymbol{v}_0$ to $\boldsymbol{v}_1$ as $\theta_{01}$, the angle of rotation for the vector $\boldsymbol{v}$, $\theta$, is defined as follows

$$\theta(u) = \theta_{01}u = cos^{-1}(\boldsymbol{v}_0 \cdot \boldsymbol{v}_1)u \qquad \textbf{(4.7)}$$

Then the vector $\boldsymbol{v}$ is expressed as the following equation according to Rodrigues' formula.

$$\begin{aligned} &\boldsymbol{v}(u) = \boldsymbol{v}_{\parallel} + \boldsymbol{v}_{\perp} \\ &\boldsymbol{v}(u) = \boldsymbol{v}_0 cos\theta(u) + (\hat{\boldsymbol{n}} \times \boldsymbol{v}_0) sin\theta(u) + \hat{\boldsymbol{n}}(\hat{\boldsymbol{n}} \cdot \boldsymbol{v}_0)(1 - cos\theta(u)) \end{aligned} \tag{4.8}$$

Since the $\hat{\boldsymbol{n}}$ is normal to $\boldsymbol{v}_0$, the third term disappears. In addition, by defining an arbitrary vector $\boldsymbol{T}$ for $(\hat{\boldsymbol{n}} \times \boldsymbol{v}_0)$, the equation is simplified into the following:

$$\begin{gathered} \boldsymbol{v} = \boldsymbol{v}_0 cos\theta(u) + \boldsymbol{T} sin\theta(u) \\ \boldsymbol{v} = \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} = \begin{bmatrix} v_{0_x} cos\theta(u) + T_x sin\theta(u) \\ v_{0_y} cos\theta(u) + T_y sin\theta(u) \\ v_{0_z} cos\theta(u) + T_z sin\theta(u) \end{bmatrix} \end{gathered} \tag{4.9}$$

Applying trigonometric rules to equation (4.3), above expression can be rearranged for WCS variables $\theta_b$ and $\theta_c$.

$$\begin{Bmatrix} \theta_b(u) \\ \theta_c(u) \end{Bmatrix} = \begin{Bmatrix} cos^{-1}(v_z) \\ tan^{-1}(v_y/v_x) \end{Bmatrix} = \begin{Bmatrix} cos^{-1}(v_{0_z} cos\theta(u) + T_z sin\theta(u)) \\ tan^{-1}\left(\dfrac{v_{0_y} cos\theta(u) + T_y sin\theta(u)}{v_{0_x} cos\theta(u) + T_x sin\theta(u)}\right) \end{Bmatrix} \tag{4.10}$$

Using equations (4.4) and (4.10), the tool path and its orientation in WCS can be express as the following vector:

$$\boldsymbol{r}(u) = \begin{Bmatrix} x(u) \\ y(u) \\ z(u) \\ b(u) \\ c(u) \end{Bmatrix} = \begin{Bmatrix} \Delta xu \\ \Delta yu \\ \Delta zu \\ cos^{-1}(v_{0_z} cos\theta(u) + T_z sin\theta(u)) \\ tan^{-1}\left(\dfrac{v_{0_y} cos\theta(u) + T_y sin\theta(u)}{v_{0_x} cos\theta(u) + T_x sin\theta(u)}\right) \end{Bmatrix} \tag{4.11}$$

The progression of the tool path and its orientation in WCS is illustrated in the below figure.

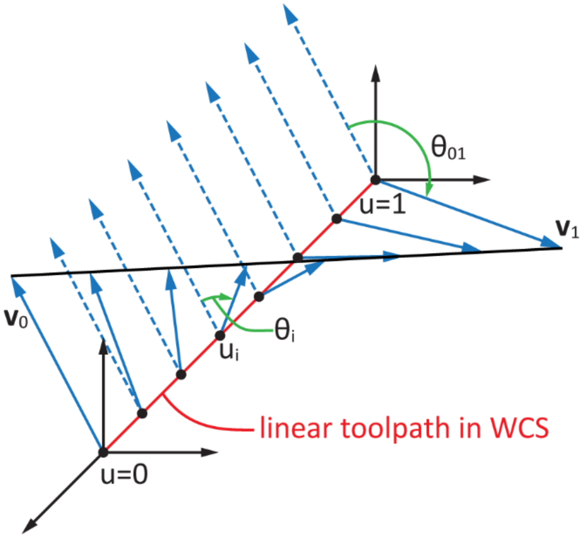


**Figure 4.9 Visualization of tool path and orientation in WCS**

Since the machine kinematic limits are defined in the joint spaces, this WCS vector is transformed into MCS vector using the transformation matrix, (3.8), defined in the previous chapter.

$$\boldsymbol{q}(u) = \begin{Bmatrix} X(u) \\ Y(u) \\ Z(u) \\ B(u) \\ C(u) \end{Bmatrix} = \boldsymbol{H} \times \boldsymbol{r}(u) + d$$

$$= \begin{bmatrix} cos\big(b(u)\big)\, cos(c(u)) & cos(b(u))\, sin(c(u)) & -sin(b(u)) & 0 & 0 \\ -sin(c(u)) & cos(c(u)) & 0 & 0 & 0 \\ sin(b(u))\, cos(c(u)) & sin(b(u))\, sin(c(u)) & cos(b(u)) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

$$\times \begin{Bmatrix} x(u) \\ y(u) \\ z(u) \\ b(u) \\ c(u) \end{Bmatrix} + d = \begin{Bmatrix} X(u) \\ Y(u) \\ Z(u) \\ B(u) \\ C(u) \end{Bmatrix}$$

Then the expression for the joint space velocity, acceleration, and jerk that the machine experiences during the traverse can be found by taking the time derivative of the above vector.

$$\begin{aligned}
&\boldsymbol{v}(t) = \dot{\boldsymbol{q}}(t) = \boldsymbol{q}'(u)\dot{u}(t) \\
&\boldsymbol{a}(t) = \ddot{\boldsymbol{q}}(t) = \boldsymbol{q}'(u)\ddot{u}(t) + \boldsymbol{q}''(u)\dot{u}^2(t) \\
&\boldsymbol{j}(t) = \dddot{\boldsymbol{q}}(t) = \boldsymbol{q}'(u)\dddot{u}(t) + 3\boldsymbol{q}''(u)\dot{u}(t)\ddot{u}(t) + \boldsymbol{q}'''(u)\dot{u}^3(t)
\end{aligned} \tag{4.13}$$

Here, primes denote a geometric derivative with respect to the path parameter u, and the overhead dots denote a time derivative with respect to time $t$. Then the objective of the task space planning for the s-curve velocity trajectory is to find the maximum parametric velocity, acceleration, and jerk values ($\dot{u}_{max}$, $\ddot{u}_{max}$, and $\dddot{u}_{max}$ respectively) that are used to construct a corresponding s-curve profile so that at anytime $t = t_i$ during this s-curve motion, the following constraints are satisfied for all five axes.

$$\begin{aligned}
-\boldsymbol{v_{max}} \le \boldsymbol{v}(t_i) \le \boldsymbol{v_{max}} \\
-\boldsymbol{a_{max}} \le \boldsymbol{a}(t_i) \le \boldsymbol{a_{max}} \\
-\boldsymbol{j_{max}} \le \boldsymbol{j}(t_i) \le \boldsymbol{j_{max}}
\end{aligned} \tag{4.14}$$

Where $\boldsymbol{v_{max}}$, $\boldsymbol{a_{max}}$, and $\boldsymbol{j_{max}}$ are the vectors that represent the axis level kinematic limits for the machine.

Since the velocity at any point along the path should meet the above constraints, the above bounded problem for velocity can be rewritten as follows:

$$|\dot{\boldsymbol{q}}_{\boldsymbol{max}}| = |\boldsymbol{q}'(u)|\dot{u}_{max} = \boldsymbol{v_{max}} \tag{4.15}$$

$\boldsymbol{q}'(u)$ is the geometric derivative which can be pre-computed for the sampled path parameter $u$, and $\boldsymbol{v_{max}}$ is the known machine parameter. Hence, the maximum allowable parametric velocity, $\dot{u}_{max}$, can be explicitly calculated by the following equation.

$$\dot{u}_{max} = \min(\frac{v_{X_{\max}}}{|q_X'(u)|}, \frac{v_{Y_{\max}}}{|q_Y'(u)|}, \frac{v_{Z_{\max}}}{|q_Z'(u)|}, \frac{v_{B_{\max}}}{|q_B'(u)|}, \frac{v_{C_{\max}}}{|q_C'(u)|}) \tag{4.16}$$

However, for the higher order derivatives of parametric kinematic limits, it is challenging to solve explicitly because $t$ is unknown. Thus, the time dependent elements $\dot{u}(t)$, $\ddot{u}(t)$, and $\dddot{u}(t)$ cannot be computed beforehand.

The following sections present the different methods proposed to resolve this difficulty.

#### 4.4.1.1 Strategy #2: Specified Path in WCS with Non-iterative Approximated S-curve Velocity Profiling in Task Space

The method in this section simplifies the above equation (4.13) by only considering the highest order parametric derivative terms and neglecting the exponential and coupled terms.

$$\begin{aligned} \boldsymbol{a}(t) &= \ddot{\boldsymbol{q}}(t) \cong \boldsymbol{q}'(u)\ddot{u}(t) \\ \boldsymbol{j}(t) &= \dddot{\boldsymbol{q}}(t) \cong \boldsymbol{q}'(u)\dddot{u}(t) \end{aligned} \qquad \textbf{(4.17)}$$

This assumption is valid at low velocity regions near the beginning and the ending of the trajectory where it is at its acceleration and deceleration phases. In particular, this assumption holds better in short trajectories where the velocity profile is not mature, and the constant velocity region has not developed. Based on the typical workpiece sizes, their hole patterns, and experimentally measured data, these short trajectories make up about 90% of the entire trajectory.

Using the same analogy for the equation (4.16), the bounded problem for the acceleration and the jerk are rewritten as follows:

$$\begin{aligned} |\ddot{\boldsymbol{q}}_{\boldsymbol{max}}| &= |\boldsymbol{q}'(u)|\ddot{u}_{max} = \boldsymbol{a}_{\boldsymbol{max}} \\ |\dddot{\boldsymbol{q}}_{\boldsymbol{max}}| &= |\boldsymbol{q}'(u)|\dddot{u}_{max} = \boldsymbol{j}_{\boldsymbol{max}} \end{aligned} \qquad \textbf{(4.18)}$$

Note that $\boldsymbol{q}'(u)$ can be pre-computed as mentioned earlier, the maximum allowable parametric acceleration and jerk can be explicitly found by the following equations.

$$\begin{aligned} \ddot{u}_{max} &= \min\left(\frac{a_{X_{\max}}}{|q_X'(u)|}, \frac{a_{Y_{\max}}}{|q_Y'(u)|}, \frac{a_{Z_{\max}}}{|q_Z'(u)|}, \frac{a_{B_{\max}}}{|q_B'(u)|}, \frac{a_{C_{\max}}}{|q_C'(u)|}\right) \\ \dddot{u}_{max} &= \min\left(\frac{j_{X_{\max}}}{|q_X'(u)|}, \frac{j_{Y_{\max}}}{|q_Y'(u)|}, \frac{j_{Z_{\max}}}{|q_Z'(u)|}, \frac{j_{B_{\max}}}{|q_B'(u)|}, \frac{j_{C_{\max}}}{|q_C'(u)|}\right) \end{aligned} \qquad \textbf{(4.19)}$$

With $\dot{u}_{max}$, $\ddot{u}_{max}$, and $\dddot{u}_{max}$ found, the corresponding s-curve is generated to populate parametric position, velocity, acceleration and jerk profiles ($u(t)$, $\dot{u}(t)$, $\ddot{u}(t)$, and $\dddot{u}(t)$). Then the kinematic profiles in WCS and MCS are found from these parametric profiles using the equations (4.11) and (4.12).

According to equation (4.11), where the translational axes are linearly related to the path parameter and the rotational axes are not, it is expected that the s-curve profiles are preserved for the $x, y, z$ in WCS but not for the $b, c$ in WCS. From the formulation, it is also expected that the jerk values in MCS exceed the limits in some regions where the product of the path geometric derivatives and the parametric velocity, and/or the acceleration, sufficiently increases. Due to the complex non-linearities, it is challenging to precisely predict under which condition that the limits are exceeded, but in general, it is during the long trajectories in which the violations are observed more frequently.

#### 4.4.1.2 Strategy #3: Specified Path in WCS with Iterative Exact S-curve Velocity Profiling in Task Space

The method in this section utilizes complete kinematic equations (4.13) derived in section 4.4.1 and adds an iterative process to the previous method. Initial kinematic profiles in MCS are generated by $\dot{u}_{max}$, $\ddot{u}_{max}$, and $\dddot{u}_{max}$ found from the method in section 4.4.1.1. Then the profiles are checked if they exceed the MCS kinematic limits. If exceeded, $\ddot{u}_{max}$, and $\dddot{u}_{max}$ are adjusted iteratively by using a bisection search method as illustrated in the below figure. The kinematic profiles are then updated using the full equations and checked with the limits again. This process is iterated until the limits are not exceeded and both $\ddot{u}_{max}$, and $\dddot{u}_{max}$ have converged.

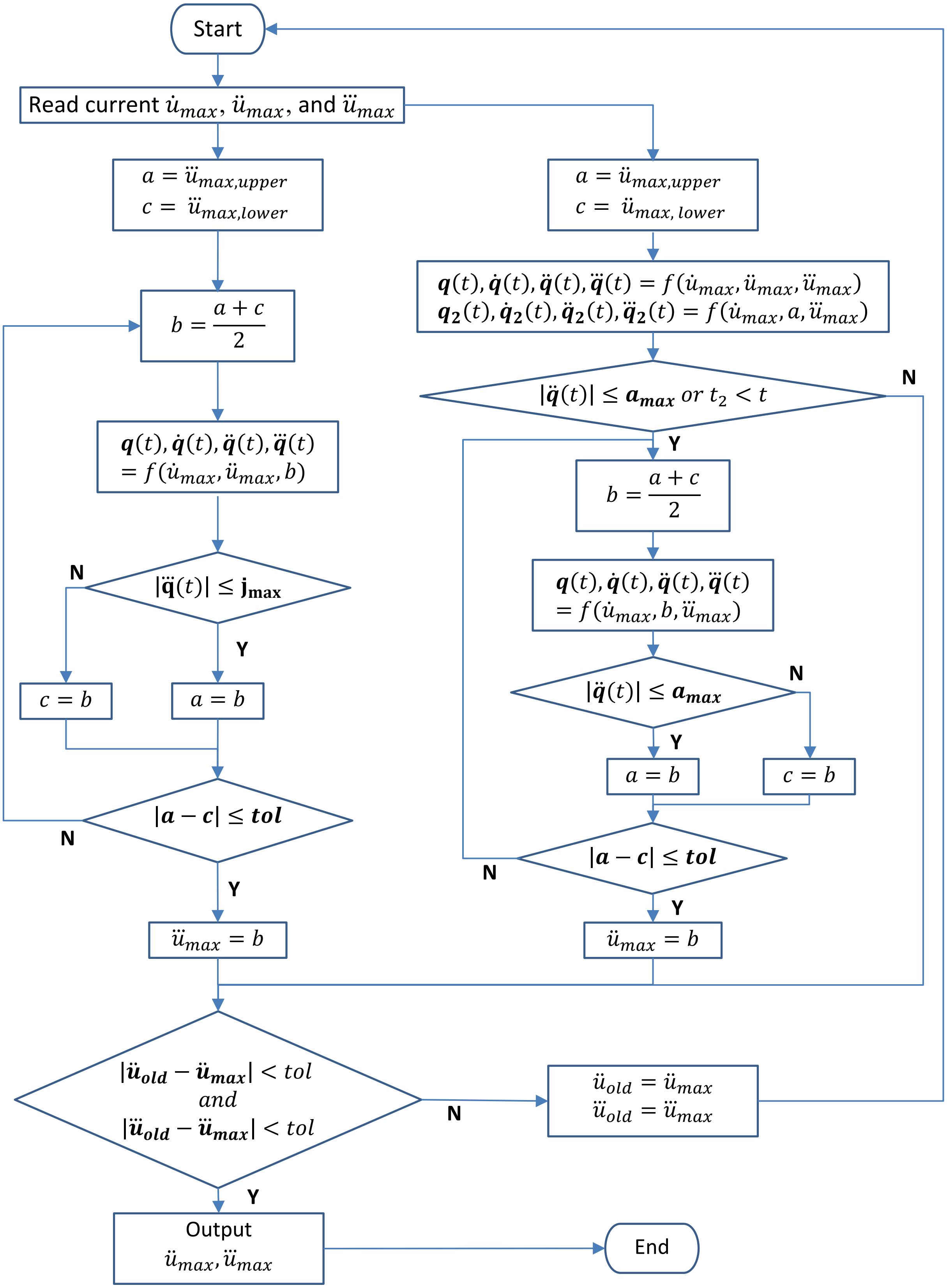

Start
Read current $\dot{u}_{max}$, $\ddot{u}_{max}$, and $\dddot{u}_{max}$
$a = \ddot{u}_{max,upper}$
$c = \dddot{u}_{max,lower}$
$a = \ddot{u}_{max,upper}$
$c = \ddot{u}_{max,\,lower}$
$\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t), \dddot{\boldsymbol{q}}(t) = f(\dot{u}_{max}, \ddot{u}_{max}, \dddot{u}_{max})$
$\boldsymbol{q_2}(t), \dot{\boldsymbol{q}}_{\boldsymbol{2}}(t), \ddot{\boldsymbol{q}}_{\boldsymbol{2}}(t), \dddot{\boldsymbol{q}}_{\boldsymbol{2}}(t) = f(\dot{u}_{max}, a, \dddot{u}_{max})$
$b = \dfrac{a+c}{2}$
$|\ddot{\boldsymbol{q}}(t)| \leq \boldsymbol{a_{max}}\ or\ t_2 < t$
N
Y
$\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t), \dddot{\boldsymbol{q}}(t)$
$= f(\dot{u}_{max}, \ddot{u}_{max}, b)$
$b = \dfrac{a+c}{2}$
$\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t), \dddot{\boldsymbol{q}}(t)$
$= f(\dot{u}_{max}, b, \dddot{u}_{max})$
N
$|\dddot{\mathbf{q}}(t)| \leq \mathbf{j_{max}}$
Y
$c = b$
$a = b$
$|\ddot{\boldsymbol{q}}(t)| \leq \boldsymbol{a_{max}}$
N
Y
$a = b$
$c = b$
$|\boldsymbol{a} - \boldsymbol{c}| \leq \boldsymbol{tol}$
N
Y
$|\boldsymbol{a} - \boldsymbol{c}| \leq \boldsymbol{tol}$
N
Y
$\dddot{u}_{max} = b$
$\ddot{u}_{max} = b$
$|\ddot{\boldsymbol{u}}_{\boldsymbol{old}} - \ddot{\boldsymbol{u}}_{\boldsymbol{max}}| < tol$
$and$
$|\dddot{\boldsymbol{u}}_{\boldsymbol{old}} - \dddot{\boldsymbol{u}}_{\boldsymbol{max}}| < tol$
N
$\ddot{u}_{old} = \ddot{u}_{max}$
$\dddot{u}_{old} = \dddot{u}_{max}$
Y
Output
$\ddot{u}_{max}, \dddot{u}_{max}$
End

**Figure 4.10 Flow chart of the iterative process of strategy #3**

Two separate bisection searches are done for each $\ddot{u}_{max}$, and $\dddot{u}_{max}$ since they are independent variables to control. However, they are dependent variables in the perspective of s-curve generation and therefore, the two searches are enclosed by one outer loop. To relax the jerk profiles in MCS, adjustment to $\dddot{u}_{max}$ is made since it is the most prominent term in the jerk equation. For the acceleration profiles, an additional evaluation is made by adjusting $\ddot{u}_{max}$. It checks if the motion time resulted from the higher $\ddot{u}_{max}$ value is less than the motion time resulted from the current $\ddot{u}_{max}$ value. This check is necessary for a more thorough search as the adjustment in $\dddot{u}_{max}$ value could create scenarios in which higher accelerations can be achieved without violating the limits. With newly obtained $\ddot{u}_{max}$ and $\dddot{u}_{max}$, the corresponding parametric s-curve velocity profile is obtained and then the kinematic profiles in WCS and MCS are acquired. Since the MCS kinematic values are directly checked via the iterative process in this method, it is expected that none of the limits are violated.

### 4.4.2 Strategy #4: Specified Path in WCS with Relaxed Convex Optimization Method via Linear Programming Formulation

The aforementioned task space S-curve generation strategy is a simple and robust method to generate trajectories under servo kinematic constraints. However, complying with the s-curve profile in task space results in trajectories in joint space that are not time-optimal. With CNC's exact trajectory generation method not known, an optimization method is suggested as another approach to estimate the machine's trajectory and to address the time-optimality at the same time. The objective function of the problem is to minimize the hole-to-hole trajectory travel time, and the constraints are the axis actuator kinematic limits expressed in task space (4.14) to satisfy the desired tool path.

$$
\begin{aligned}
&\min_{t} T = \int_0^{t_f} 1\,dt \\
&\text{subject to :} \\
&|\boldsymbol{v}(t) = \boldsymbol{q}'(u)\dot{u}(t)| \leq \boldsymbol{v_{max}} \\
&|\boldsymbol{a}(t) = \boldsymbol{q}'(u)\ddot{u}(t) + \boldsymbol{q}''(u)\dot{u}^2(t)| \leq \boldsymbol{a_{max}} \\
&|\boldsymbol{j}(t) = \boldsymbol{q}'(u)\dddot{u}(t) + 3\boldsymbol{q}''(u)\dot{u}(t)\ddot{u}(t) + \boldsymbol{q}'''(u)\dot{u}^3(t)| \leq \boldsymbol{j_{max}}
\end{aligned}
\tag{4.20}
$$

The above optimization problem is challenging to be solved directly as it is time dependent. Therefore, it is converted to an equivalent time independent problem. Using the fact that the tool path parameter $u(t)$ is strictly increasing and one-to-one function to be time-optimal, the objective function can be rewritten as follows:

$$\min_{t} T = \int_0^{t_f} 1\, dt = \int_0^1 \left(\frac{du}{dt}\right)^{-1} du = \int_0^1 \frac{1}{\dot{u}(t)} du \tag{4.21}$$

Furthermore, the optimization variable $t$ is replaced by path parameter $u$ creating the pseudo-kinematic terms: pseudo-velocity, $\dot{u}(u)$, pseudo-acceleration, $\ddot{u}(u)$, and pseudo-jerk, $\dddot{u}(u)$.

$$\begin{aligned}
&\min_{u} T = \int_0^1 \frac{1}{\dot{u}(u)} du \\
&\text{subject to:} \\
&|\boldsymbol{q}'(u)\dot{u}(u)| \leq \boldsymbol{v_{max}} \\
&|\boldsymbol{q}'(u)\ddot{u}(u) + \boldsymbol{q}''(u)\dot{u}^2(u)| \leq \boldsymbol{a_{max}} \\
&|\boldsymbol{q}'(u)\dddot{u}(u) + 3\boldsymbol{q}''(u)\dot{u}(u)\ddot{u}(u) + \boldsymbol{q}'''(u)\dot{u}^3(u)| \leq \boldsymbol{j_{max}}
\end{aligned} \tag{4.22}$$

The above nonlinear programming problem (NLP) is equivalent to the original optimization problem in (4.20). NLPs are difficult to solve and there are no effective methods or solvers for solving them [41]. Also, there is no guarantee that the solutions found from such solvers are globally optimal. In this thesis, the above NLP is approximated as a linear programming problem using the methods explained in the following section.

#### 4.4.2.1 Linear Programming Formulation

The optimization problem is linearized to yield a convex linear programming problem. The benefit of this convex relaxation is that the optimal solution to the convex problem is the global minimum. In addition, solving linear programming is less computationally expensive compared to solving nonlinear programming.

A linear program (LP) is a type of convex optimization problem where the objective and the constraint functions are all affine and have the following general form.

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \boldsymbol{G}\boldsymbol{x} \leq \boldsymbol{h} \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{array} \tag{4.23}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of variables to be optimized, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{h} \in \mathbb{R}^m$ and $\boldsymbol{b} \in \mathbb{R}^p$ are the vectors of known coefficients, and $\boldsymbol{G} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{A} \in \mathbb{R}^{p\times n}$ are the matrices of known coefficients.

To linearize the problem and convert the equation into the above LP form, first a new parameter, $w$, is defined as the square of pseudo-velocity, $\dot{u}^2(u)$. Using the chain rule, the following derivatives of $w$ with respect to $u$ are found.

$$\begin{aligned} w(u) &= \dot{u}^2(u) \\ w'(u) &= \frac{d\dot{u}^2(u)}{du} = 2\dot{u}(u)\frac{d\dot{u}(u)}{du} = 2\dot{u}(u)\frac{d\dot{u}(u)}{dt}\frac{dt}{du(u)} = \frac{2\dot{u}(u)\ddot{u}(u)}{\dot{u}(u)} = 2\ddot{u}(u) \\ w''(u) &= 2\frac{d\ddot{u}(u)}{du} = 2\frac{d\ddot{u}(u)}{dt}\frac{dt}{du(u)} = 2\frac{\dddot{u}(u)}{\dot{u}(u)} \end{aligned} \tag{4.24}$$

These can then be used to relate the pseudo-terms from the constraint functions in (4.22).

$$\begin{aligned} \dot{u}(u) &= \sqrt{w(u)} \\ \dot{u}^2(u) &= w(u) \\ \dot{u}^3(u) &= w(u)\sqrt{w(u)} \\ \ddot{u}(u) &= \frac{1}{2}w'(u) \\ \dddot{u}(u) &= \frac{1}{2}w''(u)\dot{u}(u) = \frac{1}{2}w''(u)\sqrt{w(u)} \end{aligned} \tag{4.25}$$

Substituting back with the above terms, the optimization problem in (4.22) can be rewritten without any time derivative term:

$$\min_{u} \int_0^1 \frac{1}{\sqrt{w(u)}} du$$

subject to :

$$|\boldsymbol{q}'(u)|\sqrt{w(u)} \le \boldsymbol{v_{max}} \tag{4.26}$$

$$\left|\frac{1}{2}\boldsymbol{q}'(u)w'(u) + \boldsymbol{q}''(u)w(u)\right| \le \boldsymbol{a_{max}}$$

$$\left|\frac{1}{2}\boldsymbol{q}'(u)w''(u) + \frac{3}{2}\boldsymbol{q}''(u)w'(u) + \boldsymbol{q}'''(u)w(u)\right| \sqrt{w(u)} \le \boldsymbol{j_{max}}$$

Although the objective function is a convex function, it is not in a linear form. Therefore, it is transformed to the equivalent linear programming problem using the convex operations.

For a convex minimization problem whose objective is to minimize a convex function, $f$, maximizing a concave function, $-f$, is the equivalent problem. Therefore, the objective function can be rewritten as follows:

$$\max_{u} - \int_0^1 \frac{1}{\sqrt{w(u)}} du \tag{4.27}$$

Then, consider the following convex optimization problem

$$\text{minimize } f(x) \tag{4.28}$$

For a monotone increasing function $\psi: \mathbb{R} \to \mathbb{R}$, the transformation of the objective function via composition can be made, $\tilde{f}(x) = \psi(f(x))$, and the following corresponding optimization problem is equivalent to the above original problem.

$$\text{minimize } \tilde{f}(x) \tag{4.29}$$

Therefore, the objective function in (4.27) can be written as a transformed function via the following composition,

$$\frac{-1}{\sqrt{w(u)}} = \tilde{f}(w) = \psi_1\left(\psi_2(f(w))\right) \tag{4.30}$$

where $\psi_1(x) = -\frac{1}{x}$ and $\psi_2(x) = \sqrt{x}$ are both monotone increasing $\mathbb{R} \rightarrow \mathbb{R}$ functions for $x > 0$. Rewriting (4.27) using the above rule, it is true to state that the following LP problem is equivalent to the original optimization problem.

$$\max_{u} \int_0^1 w(u)\, du = \min_{u} - \int_0^1 w(u)\, du \tag{4.31}$$

Next, linearization of the constraint functions is made as there still present nonlinear terms. Using the same concept of composition rule, the velocity equation is linearized by squaring both sides as this operation preserves the convexity.

$$\boldsymbol{q'}^2(u) w(u) \leq \boldsymbol{v}_{\boldsymbol{max}}^2 \tag{4.32}$$

For the acceleration and the jerk equations, B-spline approximation is used to linearize $w'(u)$ and $w''(u)$ terms, as suggested in [42]. Since the jerk values are bounded by a finite number, the third order derivatives should exist for the path parameter, $u(t)$. Hence, for $w(u) = \dot{u}^2(u)$, its profile is approximated by a 5th order B-spline. A B-spline of order $p$ is a spline function that creates a smooth curve by joining several pieces of polynomials of degree $k = p - 1$ end to end, which follows the control points. The below figure illustrates the 3rd order B-spline curve that follows 12 control points.

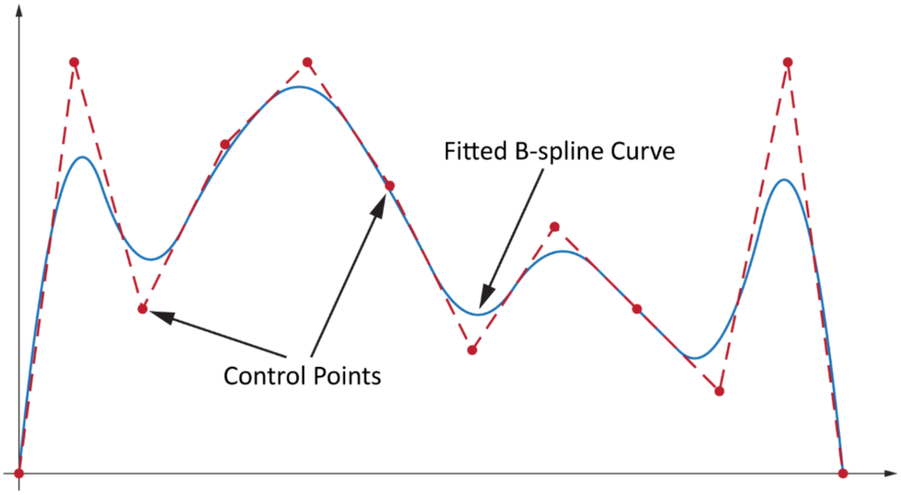


**Figure 4.11 Illustration of 3rd order B-spline curve following 10 control points**

The benefit of this B-spline approximation is that the equation for B-spline curve is expressed by a linear combination of the control points and the basis function as follows:

$$w(u) \cong \widetilde{w}(u) = \sum_{i=0}^{n} N_{i,k}(u)\alpha_i \tag{4.33}$$

Here, $\boldsymbol{\alpha}$ is the control point vector of size $n + 1$ and $N_{i,k}(u)$ is the basis function defined on a knot vector $\boldsymbol{U}$, which spans the path parameter $u \in [0,1]$. The size of knot vector $m + 1$ is determined by the number of control points and the degree of the polynomials $m = n + k + 1$. The value of the B-spline curve at any point $u_j$ is obtained by the summation of the product of the adjacent $k + 1$ control points and their basis function affected by the knot values. Therefore, for the 5th order B-spline curve to strictly pass the initial and the final control points, the initial and final velocity conditions, $\alpha_0$ and $\alpha_n$ are set to zero and the knot vector is defined as follows:

$$\boldsymbol{U} = \left[\underbrace{0,0,0,0,0}_{k+1}, u_1, u_2, \dots, u_{n-5}, \underbrace{1,1,1,1,1}_{k+1}\right] \tag{4.34}$$

Using the properties of B-spline, the derivative terms can also be represented by the linear combination of the control points and the basis function as follows:

$$\begin{gathered} w'(u) \cong \widetilde{w}'^{(u)} = \sum_{i=0}^{n} N'_{i,k}(u)\alpha_i \\ w''(u) \cong \widetilde{w}''(u) = \sum_{i=0}^{n} N''_{i,k}(u)\alpha_i \end{gathered} \tag{4.35}$$

The values of the basis functions and its derivatives are calculated using the following Cox-de Boor recursion formula:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & if\ \boldsymbol{U}(i) \le u < \boldsymbol{U}(i+1) \\ 0 & \text{otherwise} \end{cases}, \quad i = 0, \dots n + 5 \\ N_{i,k}(u) &= \frac{u - \boldsymbol{U}(i)}{\boldsymbol{U}(i+k) - \boldsymbol{U}(i)} N_{i,k-1}(u) + \frac{\boldsymbol{U}(i+k+1) - u}{\boldsymbol{U}(i+k+1) - \boldsymbol{U}(i+1)} N_{i+1,k-1}(u) \\ N'_{i,k}(u) &= \frac{k}{\boldsymbol{U}(i+k) - \boldsymbol{U}(i)} N_{i,k-1}(u) + \frac{k}{\boldsymbol{U}(i+k+1) - \boldsymbol{U}(i+1)} N_{i+1,k-1}(u) \\ N''_{i,k}(u) &= \frac{k}{\boldsymbol{U}(i+k) - \boldsymbol{U}(i)} N'_{i,k-1}(u) + \frac{k}{\boldsymbol{U}(i+k+1) - \boldsymbol{U}(i+1)} N'_{i+1,k-1}(u) \end{aligned} \tag{4.36}$$

Since these functions depend on the knot vector $\boldsymbol{U}$, and the current path parameter value $u_j$ for all $u$, they can be precomputed leaving the control points $\alpha$ as an optimization variable. Hence, the expressions for $\mathrm{w}(u)$, $w'(u)$, and $w''(u)$ in the equation becomes the linear form in terms of $\alpha$. The only nonlinear term left in the equation is $\sqrt{w(u)}$ within the jerk expression. To overcome this, the optimization is divided into two steps. First, the problem is solved without the jerk constraint to yield the optimal solution, $w^*$, then it is solved again with the jerk constraint that is now linearized using the following property.

$$\sqrt{\frac{w^*}{w}} \geq \frac{3}{2} - \frac{w}{2w^*} \geq 1 \tag{4.37}$$

Applying the above property to the jerk equation, the following relaxed jerk equation is obtained:

$$\left|\frac{1}{2}\boldsymbol{q}'(u)w''(u) + \frac{3}{2}\boldsymbol{q}''(u)w'(u) + \boldsymbol{q}'''(u)w(u)\right|\sqrt{w^*(u)} + \frac{\boldsymbol{j_{max}}}{2w^*(u)}w(u) \leq \frac{3}{2}\boldsymbol{j_{max}} \tag{4.38}$$

Although the relaxation yields suboptimal results, the performance is not deteriorated significantly. The below shows this two-step optimization problem.

Initial optimization problem:

$$\begin{aligned}
&\min_{\alpha} - \sum_{i=0}^{n} N_{i,4}(u)\alpha_i \\
&\text{subject to:} \\
&\left|\boldsymbol{q}'^{2}(u)\sum_{i=0}^{n} N_{i,4}(u)\alpha_i\right| \leq \boldsymbol{v}_{\boldsymbol{max}}^2 \\
&\left|\frac{1}{2}\boldsymbol{q}'(u)\sum_{i=0}^{n} N'_{i,4}(u)\alpha_i + \boldsymbol{q}''(u)\sum_{i=0}^{n} N_{i,4}(u)\alpha_i\right| \leq \boldsymbol{a_{max}}
\end{aligned} \tag{4.39}$$

Final optimization problem:

$$
\begin{aligned}
&\min_{\alpha} -\sum_{i=0}^{n} N_{i,4}(u)\alpha_i \\
&\text{subject to:} \\
&\left| \boldsymbol{q'}^2(u) \sum_{i=0}^{n} N_{i,4}(u)\alpha_i \right| \leq \boldsymbol{v}_{\boldsymbol{max}}^2 \\
&\left| \frac{1}{2}\boldsymbol{q}'(u) \sum_{i=0}^{n} N'_{i,4}(u)\alpha_i + \boldsymbol{q}''(u) \sum_{i=0}^{n} N_{i,4}(u)\alpha_i \right| \leq \boldsymbol{a}_{\boldsymbol{max}} \\
&\left| \frac{1}{2}\boldsymbol{q}'(u) \sum_{i=0}^{n} N''_{i,4}(u)\alpha_i + \frac{3}{2}\boldsymbol{q}''(u) \sum_{i=0}^{n} N'_{i,4}(u)\alpha_i + \boldsymbol{q}'''(u) \sum_{i=0}^{n} N_{i,4}(u)\alpha_i \right| \sqrt{w^*(u)} \\
&\qquad\qquad + \frac{\boldsymbol{j}_{\boldsymbol{max}}}{2w^*(u)} \sum_{i=0}^{n} N_{i,4}(u)\alpha_i \leq \frac{3}{2}\boldsymbol{j}_{\boldsymbol{max}}
\end{aligned}
\tag{4.40}
$$

The problem is solved using CVX, a MATLAB based modeling system for convex optimization. The solution to the optimization problem is strictly in the path parameter domain $(u)$. To retrieve the time domain $(t)$ trajectory, the following integration is evaluated.

$$
\text{total time } T = \int_{0}^{1} \frac{1}{\sqrt{w(u)}} du = \sum_{k=1}^{N-1} \int_{u_k}^{u_{k+1}} \frac{1}{\sqrt{w(u)}} du \tag{4.41}
$$

Since $w(u)$ profile is designed to have zero initial and final condition, the beginning and ending of the above integration at $\frac{1}{\sqrt{w(u_1=0)}}$ and $\frac{1}{\sqrt{w(u_N=1)}}$ approach near infinity. In remedy, the following approximation is made:

$$
\sum_{k=1}^{N-1} \int_{u_k}^{u_{k+1}} \frac{1}{\sqrt{w(u)}} du = T(u_1 = 0) + \sum_{k=2}^{N-1} \int_{u_k}^{u_{k+1}} \frac{1}{\sqrt{w(u)}} du \tag{4.42}
$$

where,

$$
\begin{aligned}
&T(0) = 0 \\
&T(u_2) = (u_2 - u_1)\left(\frac{1}{\sqrt{w(u_2)}}\right) \\
&T(u_{k+1}) = T(u_k) + \frac{1}{2}(u_{k+1} - u_k)\left(\frac{1}{\sqrt{w(u_{k+1})}} + \frac{1}{\sqrt{w(u_k)}}\right) \\
&T(1) = T(u_{N-1}) + (u_N - u_{N-1})(\frac{1}{\sqrt{w(u_{N-1})}})
\end{aligned}
\qquad \textbf{(4.43)}
$$

Therefore, parameter domain time $T(u)$ is retrieved from the above formulation and the time domain path parameter $u(t)$ is generated by simply reversing $T(u)$. Using equation (4.11), tool paths are generated from the path parameter. No specific shapes or profiles are expected to appear in this strategy. However, it is expected from the optimality criterion that one or more of the resultant profiles in MCS should have regions where the motions are at their limits.

## 4.5 Estimation Performance Comparisons of Four Strategies

The purpose of minimum time trajectory generation in this research is to closely estimate the CNC's G00 + TRAORI trajectory. During the actual drilling process, a total of 2000 reference trajectories are collected from the machine. Although the CNC generates trajectories with 2ms sampling time, due to difficulties in computational power in the logging setup, data was recorded at every 4ms. For each starting and end point pair, trajectories proposed in this chapter are generated with the same 2ms sampling time and compared against the measured hole-to-hole trajectory.

While comparing the strategies, it was found that the optimization approach is much more computationally expensive than the other approaches. The average CPU time measured from MATLAB for 10 randomly selected sampled trajectories is shown in the below table.

**Table 4.1 Average CPU time used in MATLAB for each strategy**

| Strategy | #1 | #2 | #3 | #4 |
|---|---|---|---|---|
| Time [s] | 0.0014 | 0.0122 | 0.1438 | 84.6842 |

It would take approximately 50 hours to estimate 2000 trajectories using the optimization approach and considering the actual planning processes, where the number of hole-to-hole permutations is more than 1,000,000, it is decided to first conduct a preliminary abridged analysis that compares the strategies for only the selected number of trajectories.

### 4.5.1 Preliminary Abridged Analysis

The selection includes the 50 fastest, 50 slowest, and 50 median trajectories from the 2000 samples. For each hole-to-hole trajectory, the errors are calculated by subtracting the estimated time from the experimentally measured time. Figure 4.12 illustrates the timing error probability density function of the 150 trajectory samples.

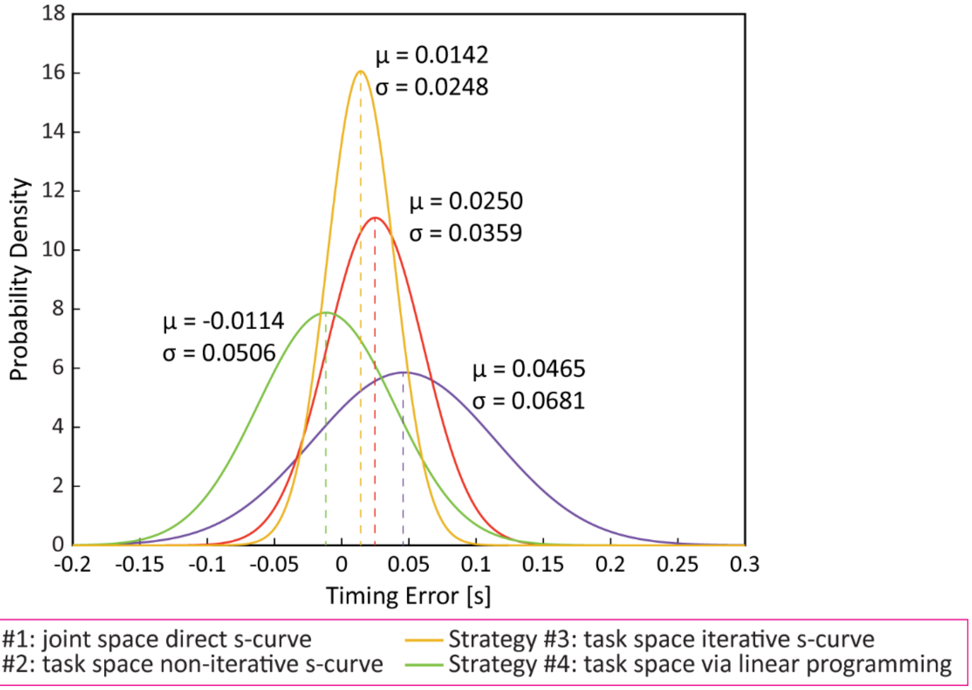


**Figure 4.12 Timing error probability density function of four strategies for 150 trajectory samples**

Strategy #4 result has the closest mean value to zero, which indicates that it has the most accurate estimations. However, strategy #3 has the smallest standard deviation, indicating that it has the most robust estimations. Nonetheless, further analysis is conducted on each comparison case to fully understand the performance of each individual strategy. For each strategy, the mean and standard deviation values for different comparison cases are listed in Table 4.2. The actual values for the kinematic limits or hole-to-hole trajectory travel times are not presented in this thesis, so as to not disclose any process related information.

**Table 4.2 Average timing error and standard deviation for each strategy in different comparison cases**

| Strategy | Slow [s] | Median [s] | Fast [s] |
|---|---|---|---|
| #1 | $0.1371 \pm 0.0425$ | $0.0034 \pm 0.0031$ | $0.0008 \pm 0.0018$ |
| #2 | $0.0728 \pm 0.0223$ | $0.0026 \pm 0.0031$ | $0.0006 \pm 0.0016$ |
| #3 | $0.0394 \pm 0.0299$ | $0.0023 \pm 0.0060$ | $0.0013 \pm 0.0019$ |
| #4 | $0.0529 \pm 0.0284$ | $-0.0568 \pm 0.0179$ | $-0.0291 \pm 0.0081$ |

### 4.5.1.1 Slow trajectories

All of the strategies have positive errors meaning that they overestimate trajectories resulting in faster travel time than the experimentally measured trajectories. Among all, strategy #1 overestimates the most, this is expected since strategy #1 directly plans the trajectories in the joint space without any coordinate transformation involved. In fact, the gaps increase more as it gets more nonlinear between WCS and MCS from larger $B$ and $C$ axis actions. To illustrate this, one of the slowest trajectories is selected and plotted in Figure 4.13.

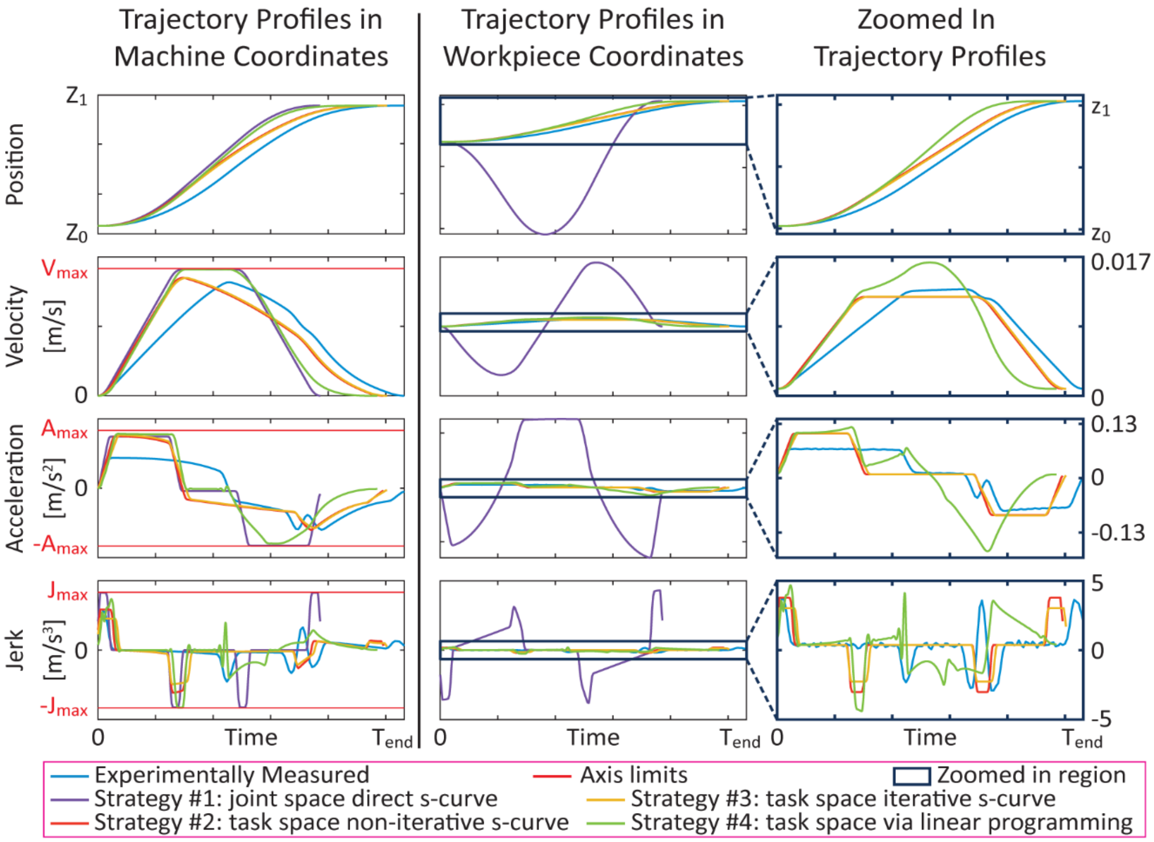


**Figure 4.13 Z-axis motion in MCS and WCS for one of the slow trajectories**

For visualization purposes, only one axis is presented instead of all five axes. Strategy #1 exhibits the s-curve profile in MCS, fully utilizing the kinematic limits. However, as mentioned previously, its WCS motion is very much different from the other strategies and the measured trajectory. It is noteworthy that in this trajectory, strategy #4 is faster than strategy #2 as opposed to shown in table 2 (greater timing error for strategy #2 than for strategy #4). In fact, this makes sense since the optimization method in strategy #4 is not bounded to s-curve profiles in task space, hence is granted with more possibilities to reach the optimal trajectory. The reason that strategy #2 is shown to be faster in table 2 is that it sometimes generates trajectories that exceed kinematic limits causing the motion time to be faster than if it were kept under the limits. Figure 4.14 shows the case in which the jerk limits for *X*-axis is violated from strategy #1.

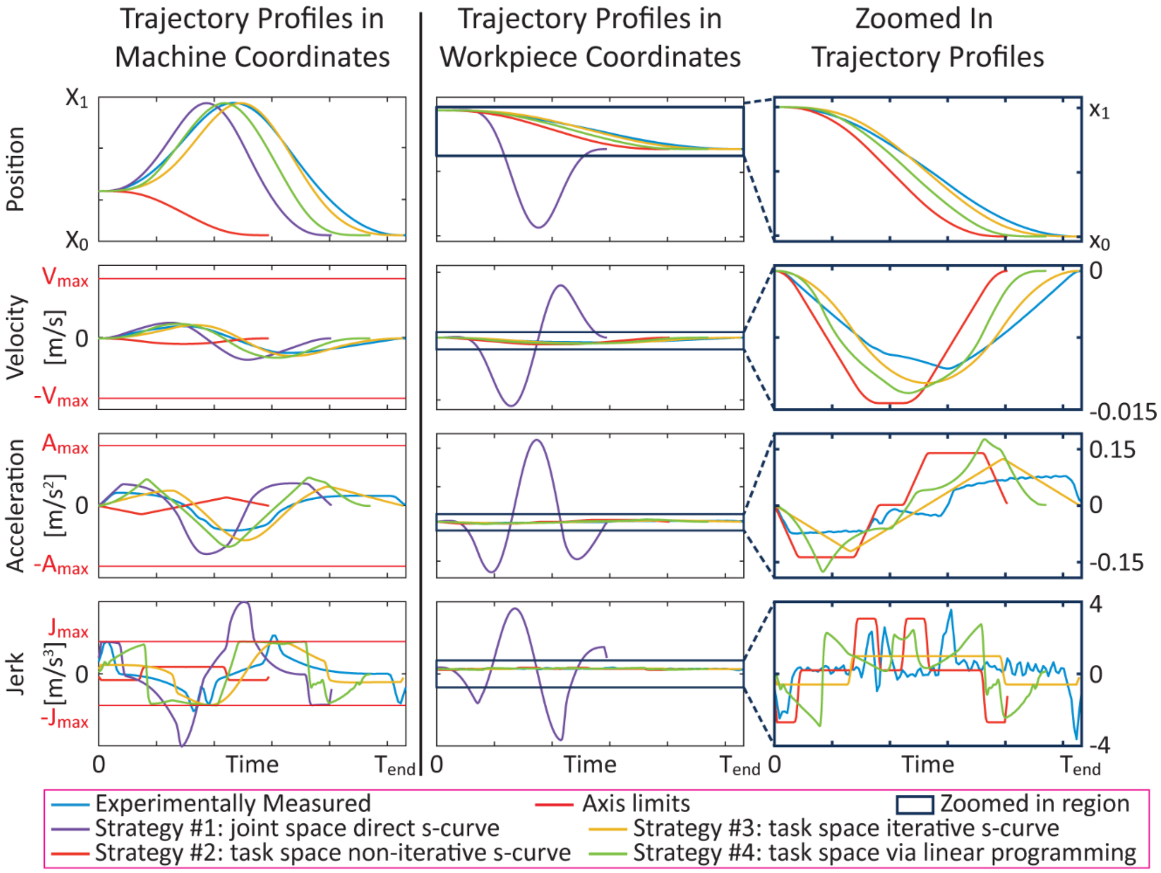


**Figure 4.14 X-axis motion in MCS and WCS for one of the slow trajectories**

As mentioned, the trajectory from strategy #2 exceeds the kinematic limit causing the motion to be faster than strategy #4. However, compared to strategy #3, which is the other s-curve profile planning in task space, strategy #4 is shown to be faster as expected.

#### 4.5.1.2 Median trajectories

Strategy #1, #2, and #3 perform similarly in most of the trajectories in the group of median trajectories. Strategy #4, contradicting to the slow cases, underestimates the motion time generating much slower trajectories than the others (Figure 4.15).

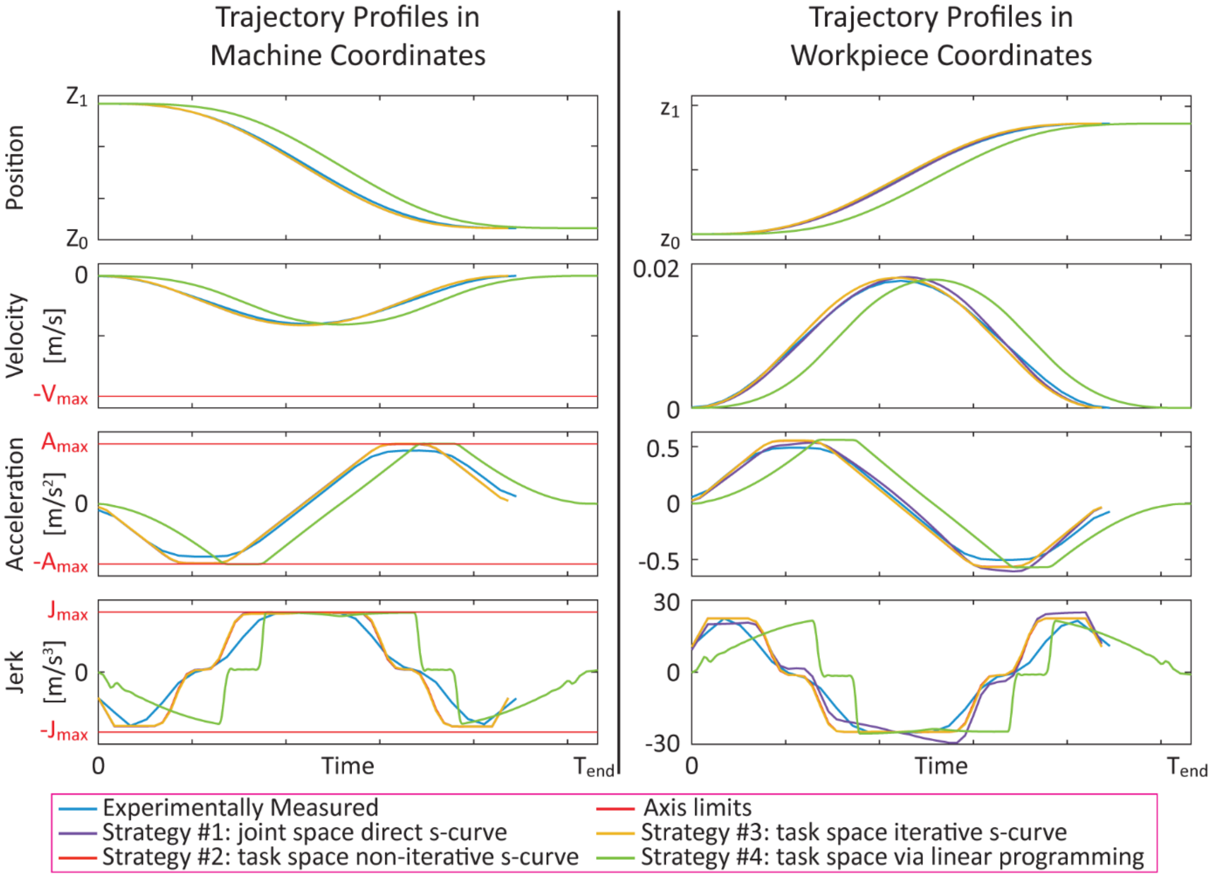


**Figure 4.15 Z-axis motion in MCS and WCS for one of the median trajectories**

It is shown that all the strategies utilize full limits for the *Z*-axis. It is noted that this motion preserves linearity between MCS and WCS well that the motion from strategy #1 is not much different from the motions from strategy #2 and #3. The strategy #4 motion is the only one that is longer than the measured one. Although utilizing the limit, it has more moderate slopes at the beginning and the end of the trajectories compared to the others causing more time while accelerating and decelerating. This can be explained by two things. Firstly, it can be explained by the assumption that is made to linearize the optimization problem. B-spline and its derivatives are used to linearize the original optimization problem, in which the initial and final conditions are strictly set to zero. When numerical integration is done to obtain the time domain trajectories from the path parameter domain, the approximation is made at the initial stage to avoid an unbounded answer. These two factors contribute to slower development of trajectories. Compared to the slow cases, this effect is more crucial since the majority of motion is used up in acceleration and deceleration. Hence, the strategy yields slower trajectories as the trajectory gets shorter and shorter.

### 4.5.1.3 Fast trajectories

It is found that the less $B$ and $C$ axes motions are involved, the faster the trajectories become. Therefore, strategy #1, #2, and #3 perform very much alike in this group (Figure 4.16). The mean timing errors for these strategies are all less than the sampling rate. Meanwhile, strategy #4, for the same reason mentioned previously, estimates much slower trajectories of which the mean errors are about 10 times larger than the sampling rate.

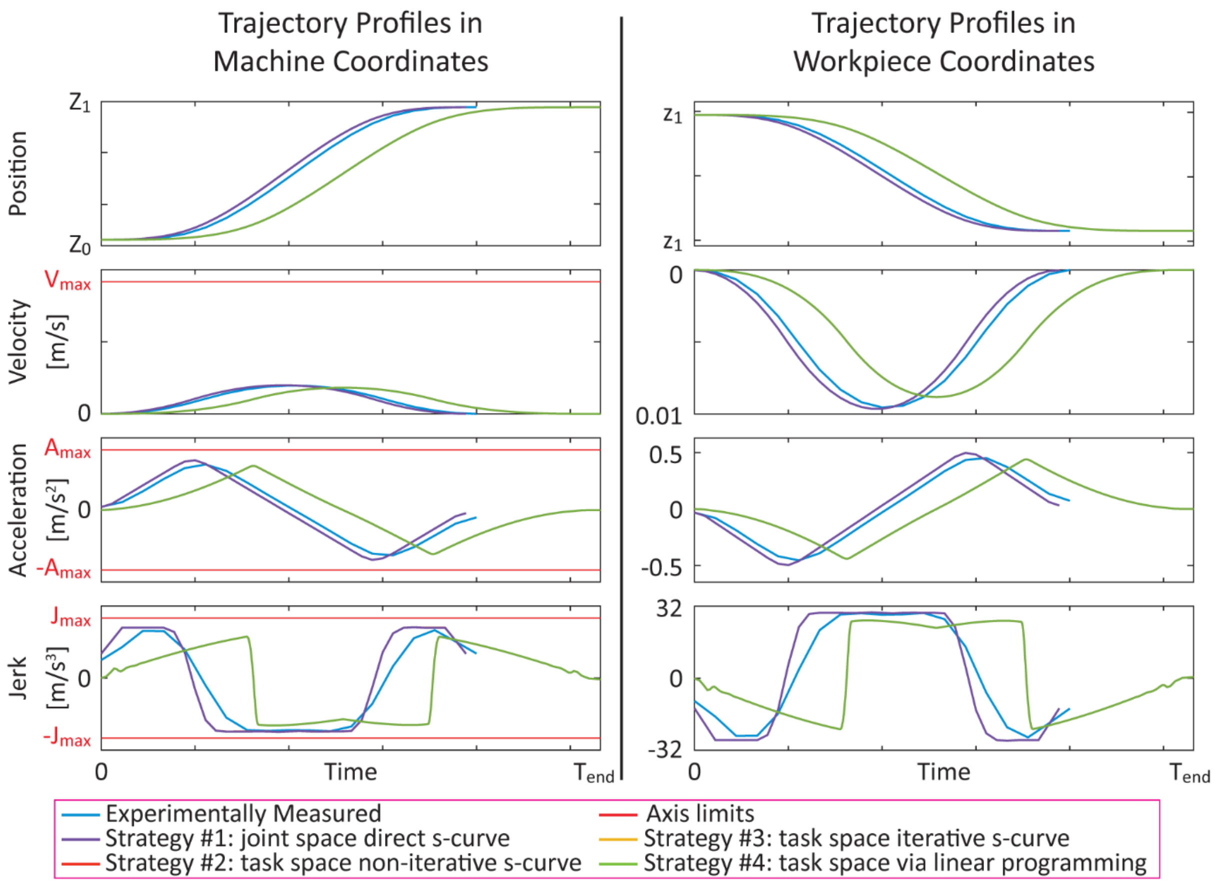


**Figure 4.16 Z-axis motion in MCS and WCS for one of the fast trajectories**

### 4.5.2 Preliminary Abridged Analysis Conclusion

Based on the analysis of the selected trajectories from different groups, it is decided to disregard strategy #1 and strategy #4 for further analysis. Strategy #1 yields similar estimation to strategy #2 and #3 for median and fast trajectories, in which the linearity between MCS and WCS is well preserved due to little to no rotary axis ($B$ and $C$) movement. However, the more it gets nonlinear, strategy #4 tends to overestimate the trajectories and yields much faster trajectories than the experimentally measured ones. For strategy #4, it also overestimates the trajectories when there is enough period to fully utilize kinematic limits. However, it underestimates significantly in median and fast trajectory cases due to its limitations from the assumptions and approximations used in the method. Strategy #3 estimates the trajectories more consistently than strategy #2, since trajectories from strategy #2 occasionally violate the kinematic limits, which produces faster motion time. Nonetheless, a full analysis is conducted for strategy #2 and #3 for a more thorough comparison.

### 4.5.3 Full Analysis

Among more than 2000 sampled trajectories, about 85% of the trajectories behave as assumed: they have straight paths with s-curve kinematic profiles in WCS and one of the axes hit the kinematic limits for time optimality. The other 15% of the trajectories behave slightly differently, that the trajectories are matured before hitting the limits. Figure 4.17 demonstrates this case in which the measured trajectories do not develop further to fully utilize the axis limits.

By looking at the MCS acceleration profile, it can be seen that the measured trajectory does not hit the limit, whereas strategy #2 and #3 motions do. It is also noteworthy that the WCS motion profiles are not the theoretical s-curve shape. It is not symmetric, and it has an unexpected dip at the beginning of the decelerating phase. The reason for this behavior is uncertain yet, since to the best of author's knowledge, the settings for the CNC do not promote such behaviors. It is possible that there are some settings inaccessible to end users that produce these behaviors. Further investigation is recommended to fully understand this phenomenon. Excluding these outliers, Figure 4.18 illustrates the findings for the 1837 trajectories.

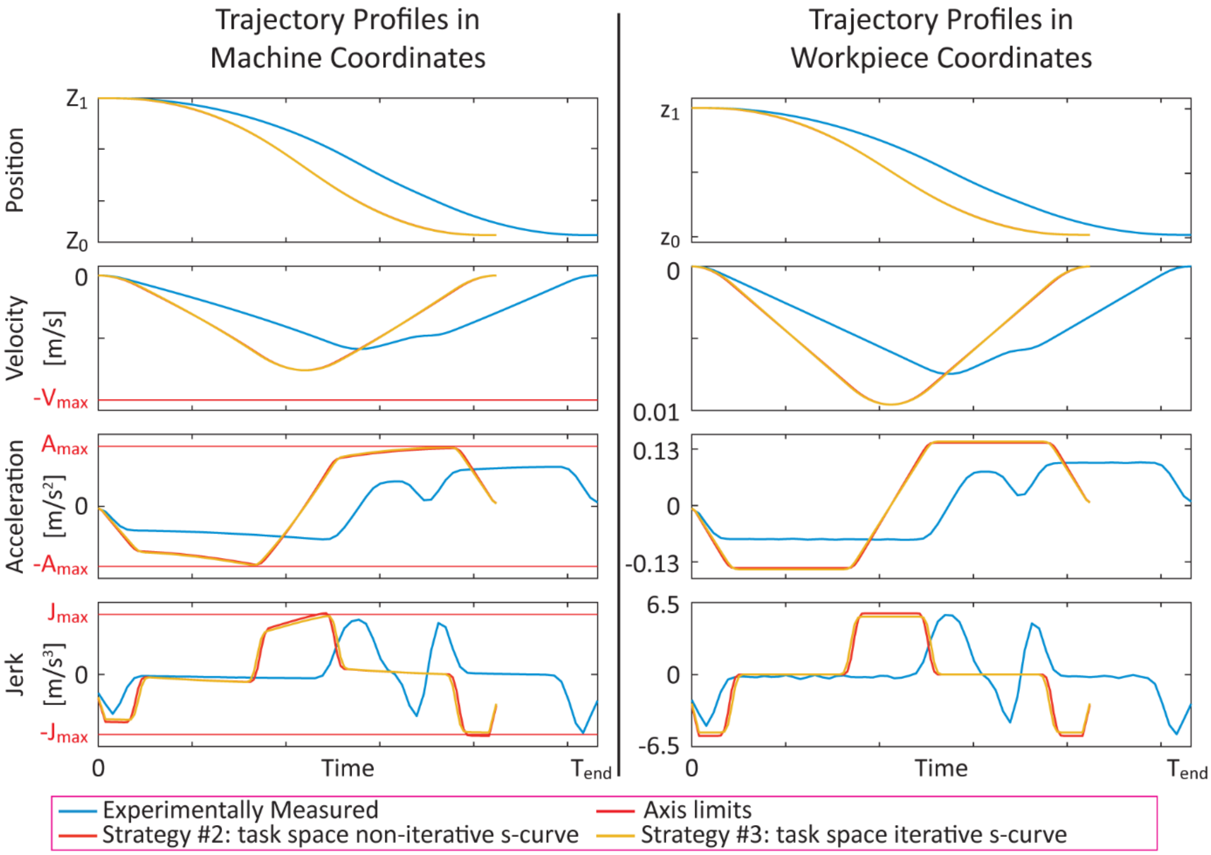


**Figure 4.17 Z-axis motion in MCS and WCS for one of the unexpected trajectories**

It is shown that strategy #3 has a mean error closer to zero and has smaller deviation than strategy #2. This suggests that strategy #3 performs better than strategy #2 for being able to more correctly estimate the CNC's trajectories with higher robustness. Furthermore, considering that strategy #2 sometimes produces trajectories that violate kinematic limits, it is decided to use strategy #3 as the main trajectory generation method for this research.

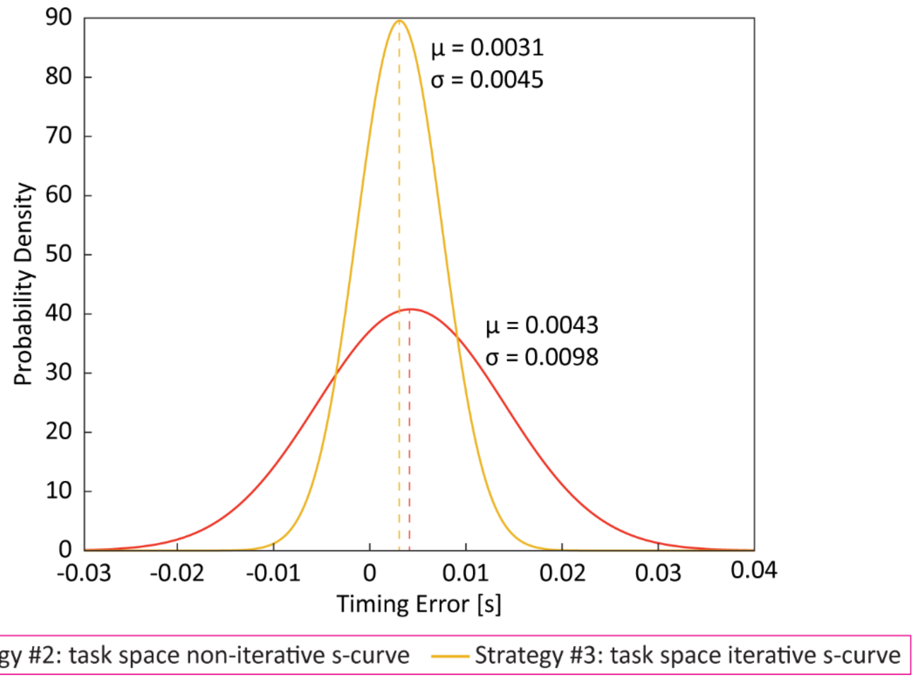


**Figure 4.18 Timing error probability density function of strategy #2 and #3 for 1837 trajectory samples**

## 4.6 Conclusion

The need for a trajectory generation method that closely estimates the trajectory generation strategy of the CNC machine is clear to obtain correct hole-to-hole motion time. Having realistic motion time makes it possible to get more realistic sequencing optimization results. In this chapter, four trajectory generation strategies are presented. Strategy #1 plans trajectory strictly in joint space to directly consider the machine's kinematic limits. Strategy #2, #3, and #4 plan trajectories in task space to address the confined path problems in the workpiece coordinate system. Both strategy #2 and #3 use the s-curve velocity profile to design the feedrate, and strategy #4 uses convex optimization to generate time optimal feedrate. More than 2000 trajectories are collected from the actual drilling processes and the performances of the four strategies are compared to the measured trajectories. As a result, strategy #3 is decided to be the main trajectory generation method for this research as it was shown to have the best estimation consistency and accuracy.

# Chapter 5
# Sequencing Optimization of Holes

## 5.1 Introduction

Given the hole-to-hole trajectory information, sequencing of the entire hole pattern can be formulated as a Traveling Salesman Problem (TSP). A TSP is to find the minimum distance tour to visit all the cities only once when a set of cities and the distances between each pair of cities are known. Referring to the set of points (cities) to $\boldsymbol{V}$, and the number of points in the set to $n_p$, a standard TSP can be expressed as an integer linear programming problem (IP) as follows:

$$
\begin{aligned}
&\min \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} C_{ij} W_{ij} \\
&\text{subject to:} \\
&\sum_{i=1, i \neq j}^{n_p} W_{ij} = 1, j = 1,2, \dots, n_p \\
&\sum_{j=1, j \neq i}^{n_p} W_{ij} = 1, j = 1,2, \dots, n_p \\
&\sum_{i,j \in S, i \neq j} W_{ij} \leq |\boldsymbol{S}| - 1, \forall \boldsymbol{S} \subset \boldsymbol{V}, \boldsymbol{S} \neq \emptyset \\
&W_{ij} \in \{0,1\}
\end{aligned}
\tag{5.1}
$$

Here $\boldsymbol{C}$ refers to the cost matrix for the problem and $C_{ij}$ represents the cost of traveling from point $i$ to $j$. $\boldsymbol{W}$ denotes the selection matrix of a certain tour, where $W_{ij} = 0$ means that the connection from point $i$ to $j$ is not selected, and $W_{ij} = 1$ means that the connection is selected. The first two equality constraints enforce a condition of departing and arriving each point exactly once. The inequality constraint requires that the number of connections between the points in the proper subset $\boldsymbol{S}$ of the set $\boldsymbol{V}$ should not exceed the number of points in the subset $\boldsymbol{S} - 1$. This constraint ensures that the solution will have only a single tour than multiple small tours which satisfy the other equality constraints.

The solution to the above IP is a single tour, represented by the selection matrix $\boldsymbol{W}$, of which its cost is the minimum among all other possible tour configurations. In order to address the minimum time sequencing problem for the drilling application, hole-to-hole travel time estimated from the methods described in the previous chapter is chosen as the cost. Since the trajectory is generated using only the kinematic limits, the motion durations for both directions are identical for the same hole pair, meaning that $T_{ij} = T_{ji}$ which creates the temporal cost matrix symmetric. In addition, the travel times between the same hole (i.e. $T_{ii}$, the diagonal elements in the matrix) are set to zero to be excluded from the sequencing calculation.

## 5.2 Solving TSP

Mathematicians and scientists have worked extensively to develop methods to solve TSP. TSP falls into a category of NP-hard problem and there are two main approaches in solving such problems: heuristic methods and exact methods. Exact methods guarantee the optimal solution; however, it often suffers from its high computation time. On the other hand, heuristics methods can solve the problems in much faster time with the optimality compromised. In this section, methods currently adopted by the industry partner, the proposed heuristic method developed by the author, and the state-of-the-art exact method used for benchmarking purpose are described.

### 5.2.1 Industry Approach #1: Zig-zag

One simple method that the industry partner uses is to construct the hole visiting sequence in a zig-zag manner. The term zig-zag is used because the shape of visiting order resembles a zig-zag pattern, which visits the holes in a row in the same direction and once at the end of the row it visits holes in the next row in the opposite direction and repeats the pattern. While its dominant pattern remains as a zig-zag, a planner, based on his/her experience, adjusts some sections to adapt for certain part geometry or to avoid contacts between the tool and the fixture. An exemplary sequencing pattern is shown in Figure 5.1.

This approach has its advantage in simplicity. For the method relies on the geometrical patterns only, the planners do not need to conduct any in-depth analysis related to the machine dynamics, the travel time calculation, or minimum time optimization problem. Therefore, they can easily produce feasible solutions. However, the downside of this method is that these solutions are far from time optimal.

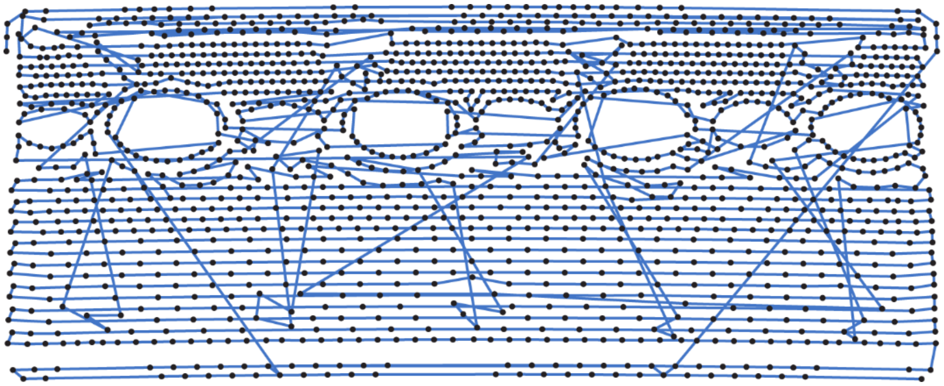


**Figure 5.1 Exemplary zig-zag sequencing pattern for the sample part**

### 5.2.2 Industry Approach #2: Modified Nearest Neighbor

A more advanced and yet still simpler method than the methods to be discussed later in this thesis has been developed by the industry partner. Due to the complex 5-axis kinematic transformation between the machine coordinates and the workpiece coordinates, the displacement in the workpiece coordinates does not have a linear relationship with the displacement in the machine coordinates. This means that what appeared to be the shorter displacement compared to others in the workpiece coordinates may not be shorter displacement in the machine coordinates after undergoing the kinematic transformation. Therefore, the previous method which relies on the geometrical patterns in the workpiece coordinates could not avoid choosing a slower path. In order to solve this issue, the displacements in the individual joints in the machine coordinates are considered in this more advanced method.

It uses a heuristic algorithm called ‘Nearest Neighbor’ (NN) to construct the hole visiting sequences (Figure 5.2). In TSP, NN is one of the simplest methods to build a tour by selecting the next unvisited point that is the closest to the current point. When there is no point to visit anymore, it simply returns to the original starting point to complete a tour. Because of this forced returning to the starting point regardless of the traveling cost, the algorithm often produces tours with poor performances. However, in the percussion drilling application, the returning to the starting point is not necessary since all the holes would be drilled completely by then. Therefore, this NN based method can produce sequences with less total travel time than the previous zig-zag method. Table 5.1 describes how the algorithm works.

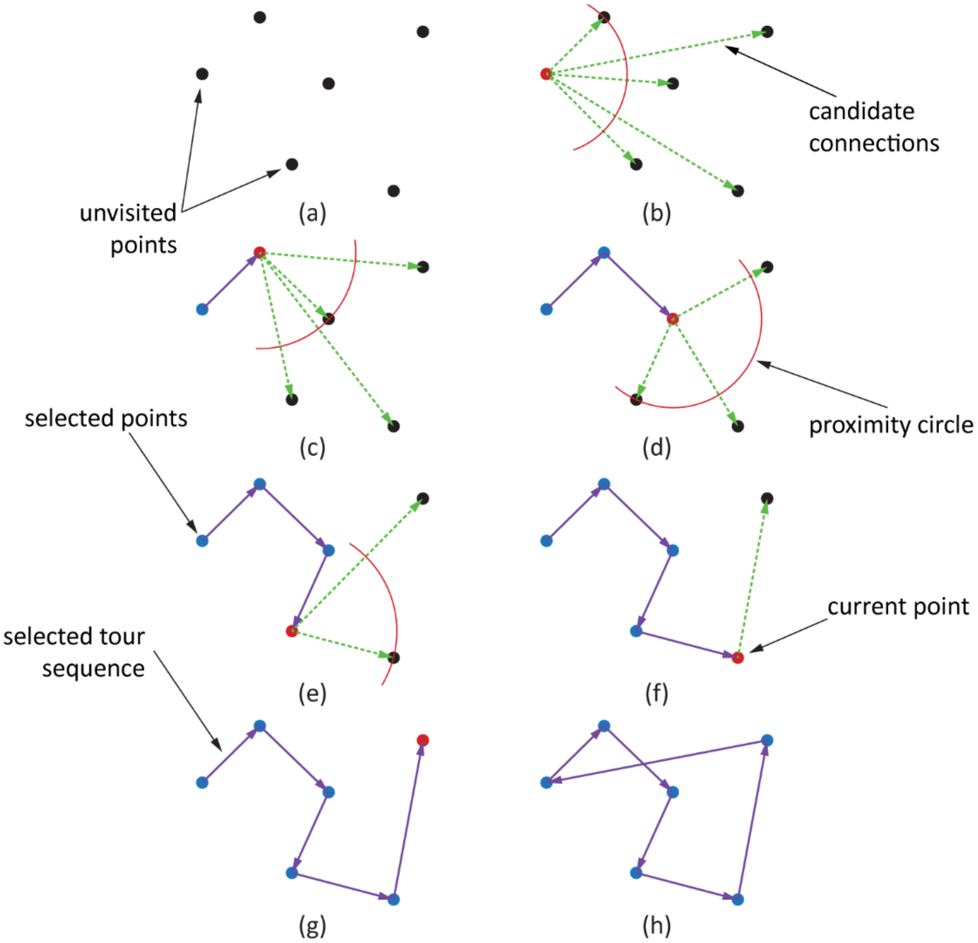


**Figure 5.2 Sequence construction illustration of Nearest Neighbor algorithm**

**Table 5.1 MCS minimum distance Nearest Neighbor approach algorithm**

```
Nearest neighbor approach on MCS distance
Input: Hole location and orientation information in WCS
Output: Nearest Neighbor optimized minimum machine coordinates distance tour
Procedure:
  Initialization:
    Hole location and orientation information in MCS are obtained via kinematic transformation;
    Distances as weighted Euclidean norm for each hole pairs are calculated, and cost matrix is
    constructed from this;
  Nearest Neighbor:
    Initial departure point is selected randomly and currentSequence is initiated;
    while (there exists unvisited point)
    next shortest distance to the current departure point is added to currentSequence;
    end
    return currentSequence as the nearest neighbor optimized tour
End Procedure.
```

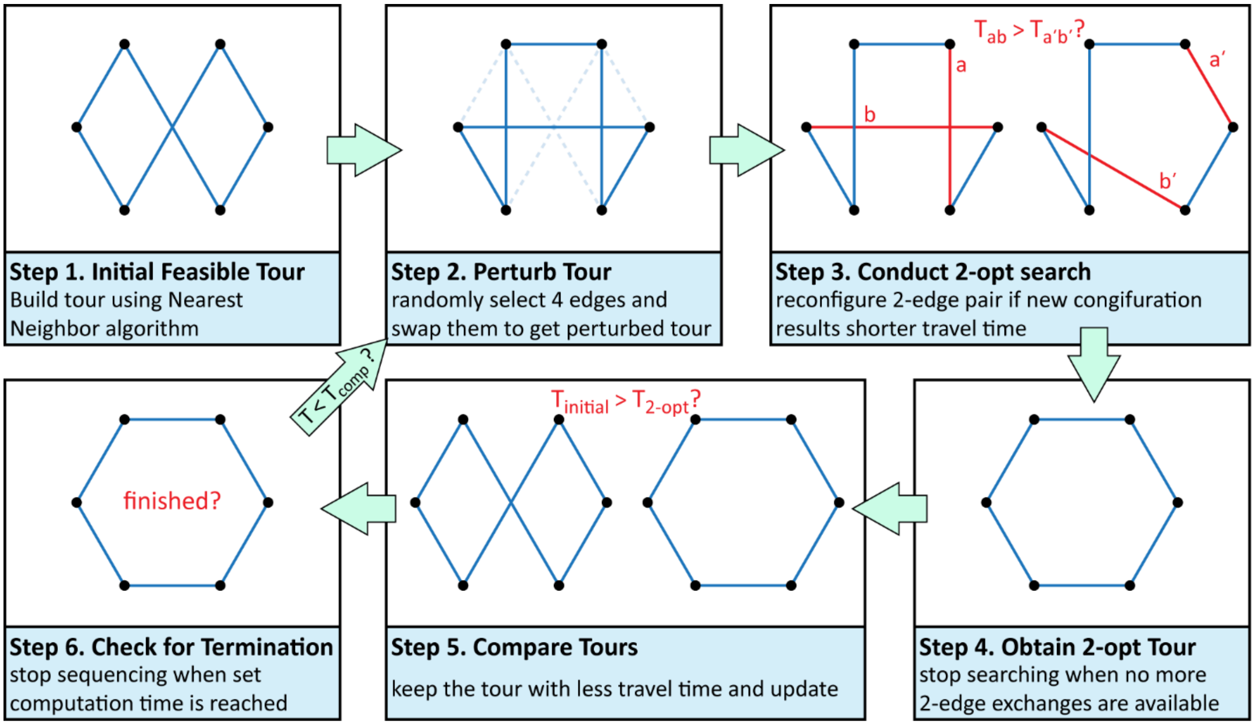
To combine translational and rotational distances into a single measurement, weighted Euclidean norm approach is taken by the following equation.

$$R = \sqrt{(\alpha(\Delta X)^2 + \alpha(\Delta Y)^2 + \alpha(\Delta Z)^2 + \beta(\Delta B)^2 + \beta(\Delta C)^2} \qquad \textbf{(5.2)}$$

Although it is more thorough than the zig-zag method and yields shorter cycle time, the NN based minimum distance method has some limitations. Firstly, weighed Euclidean norm based single distance approach to combine translational and rotational distances needs fine tuning of weighting variables, $\alpha$ and $\beta$. Secondly, NN is not powerful enough to deliver the optimal or a near optimal solution. Lastly, the travel time is still not the essence of consideration in this method.

### 5.2.3 Proposed Method: Chained 2-opt local search heuristics

A heuristic based method is proposed in this thesis in order to truly address the time optimality. Addressing the time optimality requires two parts to be solved separately: the time part and the optimality part. Unlike the previous methods, in which the non-temporal information was the main variable to solve with, the proposed method directly considers the travel time among the points as the objective variable. As presented in the previous chapter, strategy #3 for estimation of point-to-point motion time of the machine tool will be used to generate the travel duration cost matrix to be used in the optimization. Given this temporal cost matrix, optimization is conducted using the algorithm developed in this thesis. The proposed algorithm borrows the concepts from one of the most powerful TSP solving algorithms called Chained Lin-Kernighan [43], which will be described later in more detail. The need for an in-house algorithm arises from a commercial licensing issue. Since ultimately, it is desired for the work in this thesis to be used by the industry partner, a commercially available algorithm is needed. However, the top TSP solving algorithms are non-commercial and therefore, the proposed algorithm is developed to mitigate this issue. Figure 5.3 illustrates the overview of the proposed chained 2-opt local search heuristics of which the details are explained throughout this section.



**Step 1. Initial Feasible Tour**
Build tour using Nearest Neighbor algorithm

**Step 2. Perturb Tour**
randomly select 4 edges and swap them to get perturbed tour

**Step 3. Conduct 2-opt search**
reconfigure 2-edge pair if new congifuration results shorter travel time

**Step 4. Obtain 2-opt Tour**
stop searching when no more 2-edge exchanges are available

**Step 5. Compare Tours**
keep the tour with less travel time and update

**Step 6. Check for Termination**
stop sequencing when set computation time is reached

**Figure 5.3 Overview of the proposed chained 2-opt local search heuristics**

Before the Chained Lin-Kernighan algorithm is described, the Lin-Kernighan is first described. Lin-Kernighan is a tour improvement method which takes an existing tour and attempts to modify it to yield a tour with better cost. It does this by conducting a series of local searches called k-opt. A local search is a heuristic method that finds the best solution when several candidate solutions are given that are generated by applying local changes to the given tour. In TSP, these local changes refer to different connections among some points. Depending on the number of connections to be considered at a time, it can be referred to as 2-opt, 3-opt, or k-opt. An example of 2-opt local search is illustrated in the below figure.

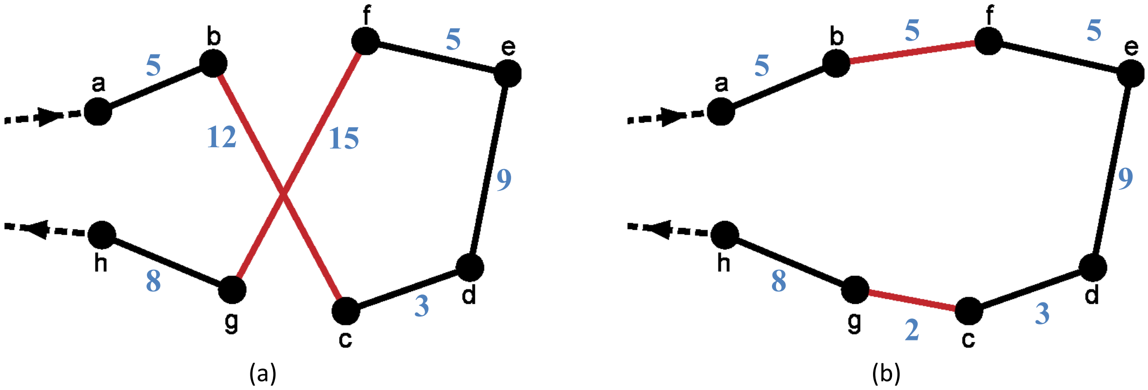


**Figure 5.4 Illustration of 2-opt local search: (a) initial tour (b) 2-opt modified tour**

Blue numbers represent the cost of each connection and the alphabetical letters express the point indices in the above illustration. The initial sequence of the tour is a-b-c-d-e-f-g-h with a total cost of 57. The local configuration of two connections of interest, expressed with red lines, has a local cost of 27. This cost is compared with its alternative configuration. For 2-opt, there is only one alternative local solution that still satisfies a tour. The alternative configuration, shown as b-f and c-g connections, has a local cost of 7. Since the alternative configuration yields better results, 2-opt adapts this alternative configuration and the sequence becomes a-b-f-e-d-c-g-h with a total cost of 37. 2-opt is repeatedly conducted for all connection pairs until there is no segment in the tour that can be improved. Then the tour is called the 2-opt optimal tour.

By exchanging more connections at a time, there are more opportunities for the modified tour to be improved: a 3-opt optimal tour could have less cost than a 2-opt optimal tour, a 4-opt optimal less than a 3-opt optimal, and so on. However, directly searching all the possible alternative configurations for a k-opt move with k greater than 2 or 3 is computationally expensive. For this reason, the proposed method uses 2-opt local search and hence, it is called chained 2-opt.

The term chained is suggested by Apple et al [43] to refer to an additional technique beyond the Lin-Kernighan's k-opt method. Instead of stopping the search when the k-opt optimal tour is found, the chained k-opt continues this search by perturbing the k-opt optimal tour slightly, referred to as kick, and applying the k-opt algorithm again. If the newly obtained k-opt optimal tour is better than the previously generated one, the old one is discarded and the new one is selected. These procedures are repeated as long as computation time is available, or the desired convergence criterion is reached. Compared to the original k-opt method, the chained k-opt method is more likely to yield better tours from this repetitive perturbation(kick)-to-search cycle [44].

Essentially, chained k-opt can be broken down to the three elements: initial tour building, perturbation strategy, and k-opt. Apple et al. [43] have extensively studied impacts of these elements on the final optimality of the modified tours. Based on their study results, Nearest Neighbor method is chosen as the initial tour building strategy and the random double bridge method is chosen as the tour perturbation strategy.

Nearest Neighbor method has been explained in the previous section. To recall, it is the method of building a tour by adding a next point that is the most cost effective to the last point in the current tour. The only difference between the previously described NN and the NN strategy in the proposed method is that the proposed method NN takes the temporal information as the cost to consider.

Random double bridge is one of the mechanisms of perturbing a tour. It exchanges four edges for four other edges. Selection of these edges is done in a random manner, hence called the random double bridge. The below figure illustrates the four-edge (in red dotted lines) exchange concept.

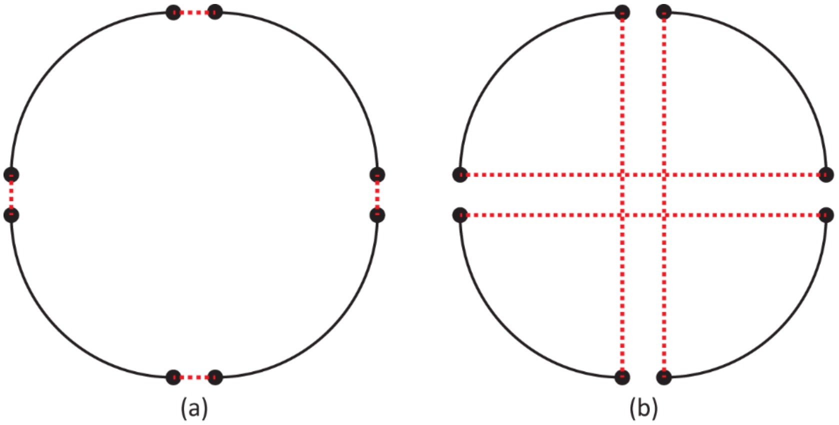


**Figure 5.5 Illustration of double bridge exchange concept: (a) initial configuration (b) after double bridge move**

The initial configuration of four edges, expressed with dotted lines, is changed with a four-edge exchange called double bridge move. The random double bridge can transform the global configuration of the tour that 2-opt cannot easily revert back to the original tour. This serves a useful purpose of perturbation which grants an opportunity for 2-opt to find another tour that is possibly more cost effective than the original one. The overall illustration of how the algorithm works is shown below.

**Table 5.2 Proposed chained 2-opt local search sequence optimization algorithm**

**Chained 2-opt local search algorithm procedure**

```
Input: The number of points $n_p$, the travel time duration matrix $\boldsymbol{T}$
Output: Optimized tour and the minimum total travel time
Procedure:
  Initialization:
    set maximum computation time compTmax;
  Initial tour using Nearest Neighbor:
    pick an initial departure point and start updating initialSequence by repeatedly adding the least time consuming arrival point;
    once initialSequence contains all the points, evaluate the total travel time initialTime;
  Loop while current computation time is less than the maximum allowable:
    twoOptSequence, bestSequence = initialSequence;
    bestTime = initialTime;
    while (compT < compTmax)
      perturb twoOptSequence using random double bridge perturbation method [6];
      while (not 2-opt optimal)
        select 2 edges (connection of 2 consecutive points) and change their configuration;
        if new configuration results shorter travel time
          change to new configuration and proceed to next 2 edges;
        else
          keep the old configuration and proceed to next 2 edges;
      end "while (not 2-opt optimal)" when all the edges are considered (2-opt optimal reached);
      twoOptTime = total travel time for the modified sequence;
      if twoOptTime < bestTime
        bestSequence = twoOptSequence;
        bestTime = twoOptTime;
      update current computation time compT;
    end "while (compT < compTmax)" when maximum allowable computation time is passed;
    return the optimal tour bestSequence and the optimal time bestTime;
End Procedure.
```

### 5.2.4 Optimal Method: Concorde Cutting Plane

For strict benchmarking purposes, a non-commercial state-of-the-art TSP solver called Concorde is used in this thesis. Concorde is an exact TSP solver that produces optimal solutions by utilizing the cutting plane method [29]. The main idea of solving an exact solution using the cutting plane method is based on the duality of linear programming. At each stage, TSP inequalities those are violated by the fractional tour are identified. A new linear programming is constructed by adding these inequalities, hence called cutting plane, and solved with the simplex algorithm. The solution to this new linear programming is verified as proper and optimal by duality. For further information about Concorde solver and the cutting plane method, reader is recommended to refer to [28].

In this research, Concorde solver is used to demonstrate the effectiveness of the proposed method as compared to the optimal solution. The solver takes the same temporal cost matrix as an input and outputs the optimal solution that results in minimum total travel time.

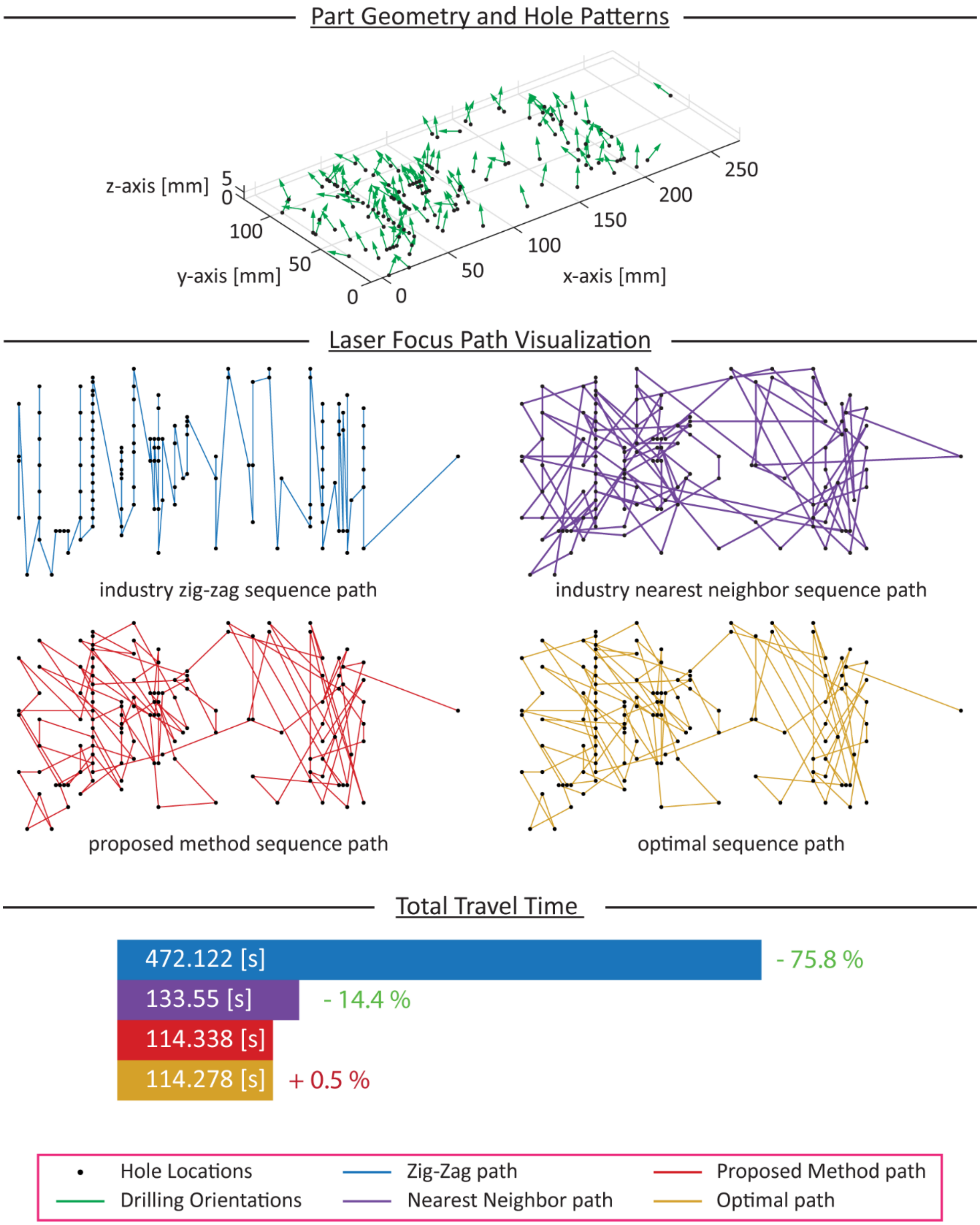
## 5.3 Comparisons

The proposed strategy was tested on three modified TSPLIB problems. TSPLIB is an open library of sample instances from various sources and types. Since the instances are 2-dimensional, additional $Z, B,$ and $C$ values are randomly generated to reflect the 5-axis laser drilling environment. The sample parts consist of 136 holes, 237 holes, and 436 holes. The results are compared with the industrial approach and the optimal approach obtained from Concorde TSP solver. For each individual sample part, hole-to-hole motion times are calculated by the proposed method described in the previous chapter. The same 5-axis laser drilling machine configuration is used in the simulations except for its kinematic values, which are chosen arbitrarily to not disclose specific machine capabilities of the industry partner and for generalization purpose. Table 5.3 lists the kinematic values used in the simulations.

**Table 5.3 Kinematic limits for 5-axis laser drilling machine used in the sequencing optimization comparisons**

| | X-axis | Y-axis | Z-axis | B-axis | C-axis |
|---|---|---|---|---|---|
| Velocity [mm/s, deg/s] | 50 | 60 | 30 | 500 | 1000 |
| Acceleration [mm/s$^2$, deg/s$^2$] | 500 | 600 | 200 | 10000 | 30000 |
| Jerk [mm/s$^3$, deg/s$^3$] | 5000 | 7000 | 10000 | 50000 | 500000 |

All four sequence optimization approaches discussed in this chapter are tested and their resultant tool paths are shown in Figure 5.6 ~ Figure 5.8.


Part Geometry and Hole Patterns

Laser Focus Path Visualization

industry zig-zag sequence path

industry nearest neighbor sequence path

proposed method sequence path

optimal sequence path

Total Travel Time

**Figure 5.6 Sample #1 tool paths for different sequencing strategies**

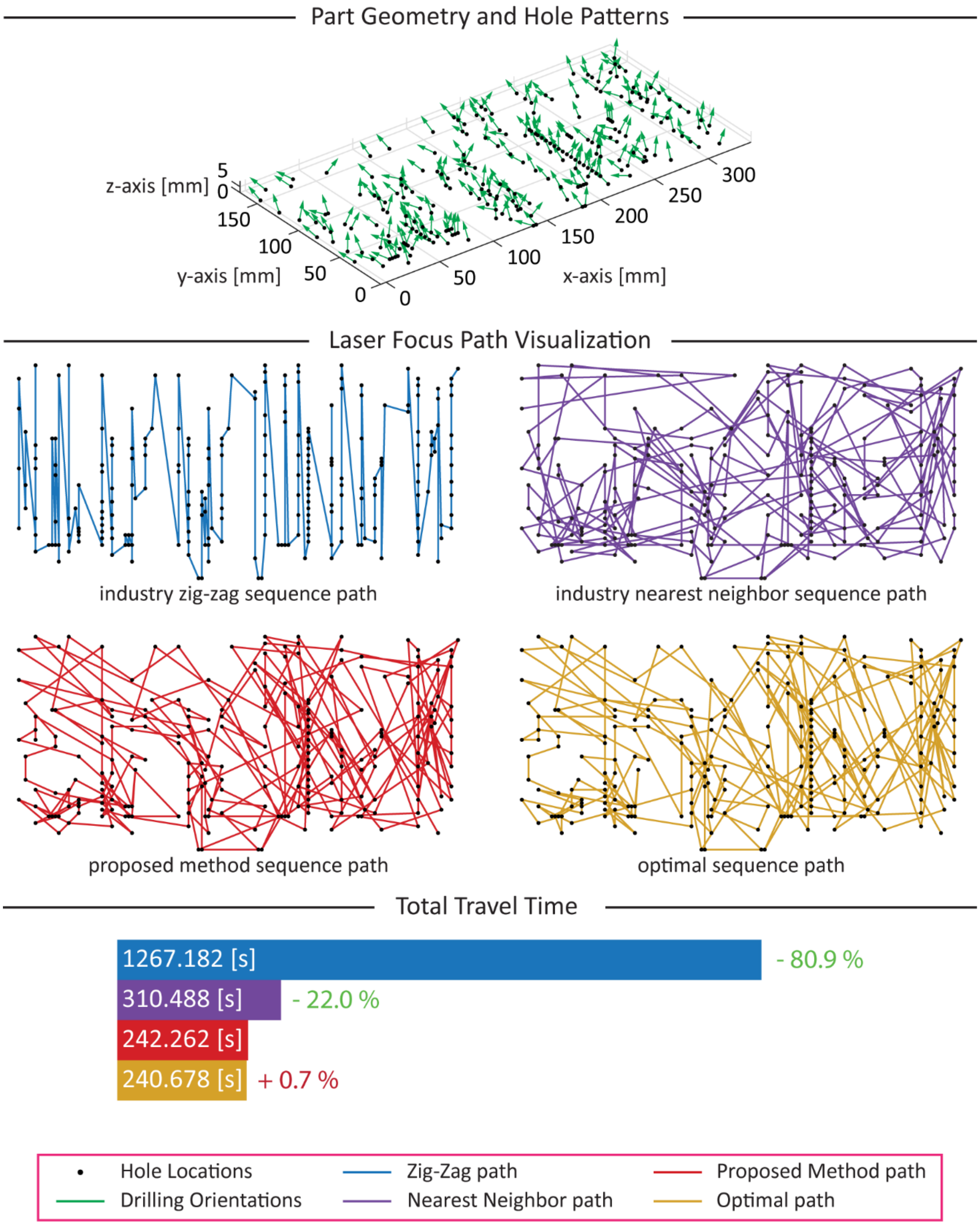


**Figure 5.7 Sample #2 tool paths for different sequencing strategies**

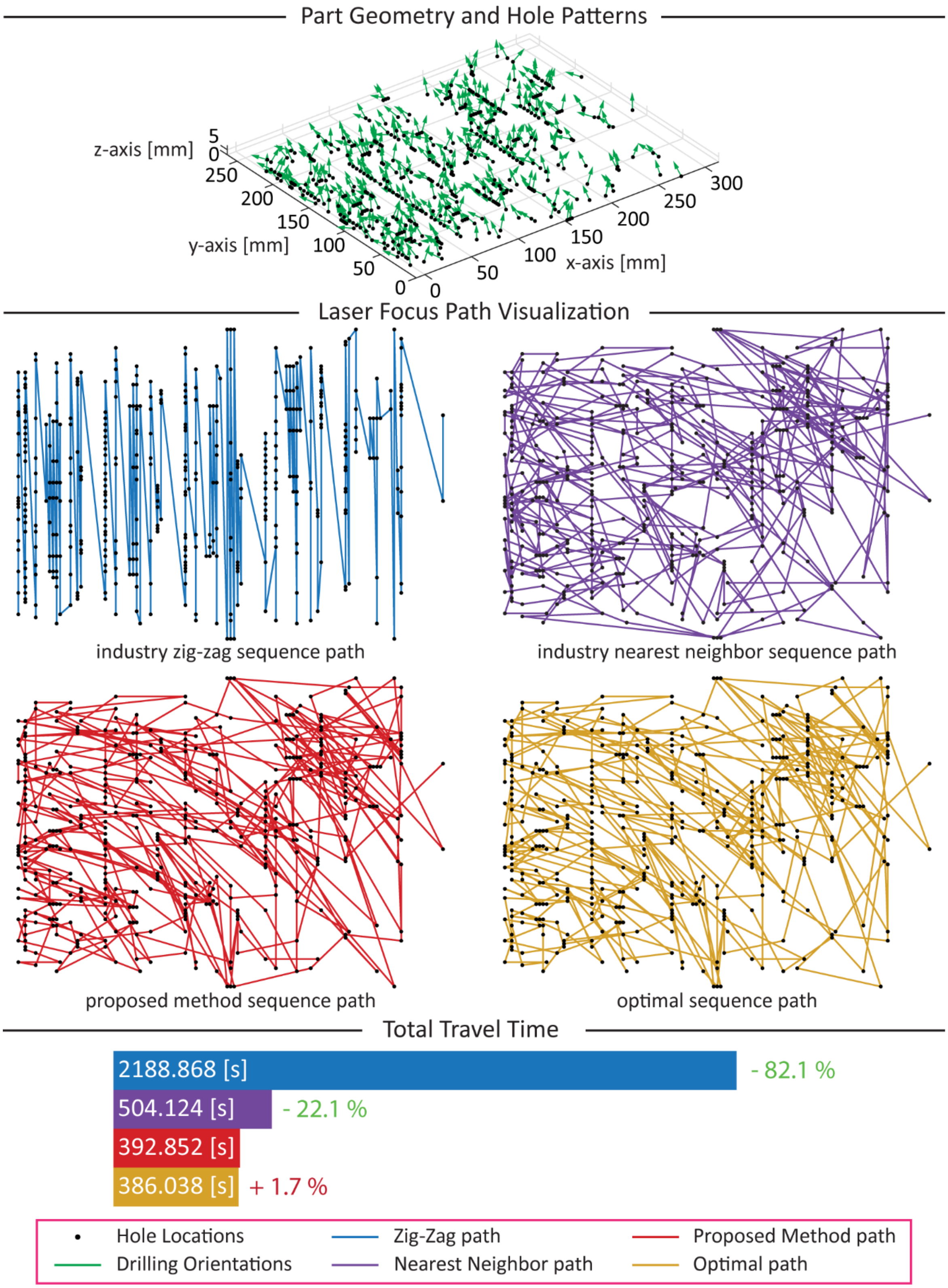


**Figure 5.8 Sample #3 tool paths for different sequencing strategies**

The quantitative results summarized in Table 5.4 show the total travel time for each sequencing strategy and the percent difference of the proposed sequencing method from the other methods. The percent difference is calculated by the following equation:

$$\frac{time_{proposed\ method} - time_{other\ methods}}{time_{other\ methods}} \times 100\% \tag{5.3}$$

**Table 5.4 total travel time for different sequencing strategies and their percent differences**

| Part Number | Travel Times for Different Sequences [s] | | | | % Difference from [%] | | |
|---|---|---|---|---|---|---|---|
| | Zig-zag | Nearest Neighbor | Proposed | Optimal | Zig-zag | Nearest Neighbor | Optimal |
| 1 | 472.122 | 133.55 | 114.338 | 114.278 | -75.8 | -14.4 | 0.5 |
| 2 | 1267.182 | 310.488 | 242.262 | 240.678 | -80.9 | -22.0 | 0.7 |
| 3 | 2188.868 | 504.124 | 392.852 | 386.038 | -82.1 | -22.1 | 1.7 |

Depending on the part geometries and the hole patterns, the proposed method can reduce the total travel time by 70~80 % compared to the zig-zag method, and 14~25 % compared to the nearest neighbor method. It is also shown that the proposed method can produce near-optimal solutions that are within 2 % of the optimal solutions.

## 5.4 Conclusion

In this chapter, it is demonstrated how hole visiting sequence optimization can be expressed as the well-known Traveling Salesman Problem of combinatorial mathematics. In solving TSP, two industrial approaches are described. Zig-zag approach is what the industry partner currently uses for its simplicity and Nearest Neighbor based minimum distance approach is what has been developed by the industry partner. Both approaches do not directly consider time optimality, and therefore in this research, chained 2-opt based minimum time approach is proposed. The effectiveness of the proposed approach is compared to the above industrial approaches and to the state-of-the-art exact TSP solver called Concorde. The results show that the proposed method can generate near-optimal sequences that successfully reduce the travel time compared to the two industrial approaches.

# Chapter 6
# Experimental Implementation on a Sample Part

In this chapter, overall cycle time reduction results achieved from the proposed sequence optimization method combined with the estimated hole-to-hole trajectory generation method are presented in both simulations and experiments. The superiority of the proposed sequencing strategy and hole-to-hole trajectory estimation strategy has already been demonstrated throughout the previous chapter. Therefore, the purpose of this chapter is to verify the effectiveness of the proposed work in a real environment and the reliability of the work by showing how well the simulated results match the experimental results. The chapter is divided into two sections. The first section discusses the results from the preliminary study that was conducted at the early stage of the research prior to the development of the proposed trajectory generation strategy. The second section discusses the results from the final study using the proposed trajectory generation strategy. Due to difficulties in acquiring experimental machine time for the second section, only simulation results are presented here.

For both sections, the kinematic limits used in the experimental setups are used in the simulations as well to bring more reflective analysis. **However, to avoid disclosing the full capabilities of the machine tools used at P&WC, the actual velocity, acceleration, and jerk limits are not presented in this thesis. Furthermore, figures do not show the dimensions and the travel times are normalized to the experimentally measured total motion time, $\boldsymbol{T_{total}}$, which is explained later in this chapter.**

One sample gas turbine chamber panel part is used for both simulations and experiments.  The part geometry and drilling orientations for 1498 hole locations are shown in the below figure.

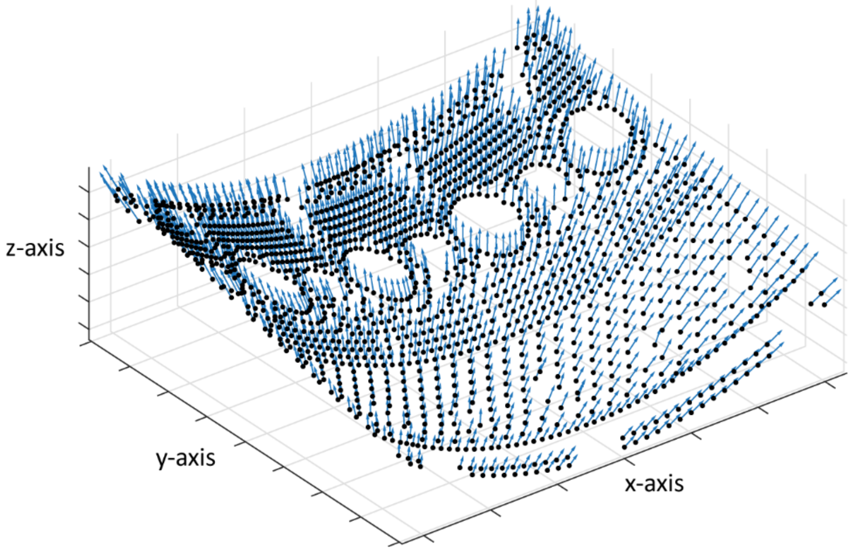


**Figure 6.1 Sample part geometry and its hole locations and drilling orientations**

## 6.1 Preliminary Study Results

As a preliminary study before in-depth analysis on the CNC controller's G00 + TRAORI trajectories was conducted, hole-to-hole travel time is estimated using the joint space s-curve planning method (strategy #1 mentioned in Chapter 4.3). Then the proposed sequence optimization method is implemented in both simulations and experiments. Experiments done in this section consisted of four separate drilling cycles in which only a subset of the entire holes is drilled at each cycle. In order to reflect this, the simulation is also divided into the same four cycles.

From the experiments, the total motion time, $T_{total}$, is calculated by adding the motion times from each cycle. Although, this does not include travel times from the previous cycle end hole to the next cycle starting hole, it still realistically represents the total motion time considering the total number of hole-to-hole trajectories.

### 6.1.1 Simulation Results

The four cycles included 561, 396, 503, and 38 holes respectively and their resultant sequence paths are shown in Figure 6.2. The actual drilling sequences used on the sample part have been made available and the paths from these sequences are shown in blue lines, whereas the paths from the proposed method sequences are shown in purple lines. The travel times for individual cycle are normalized to the total motion, $T_{total}$.

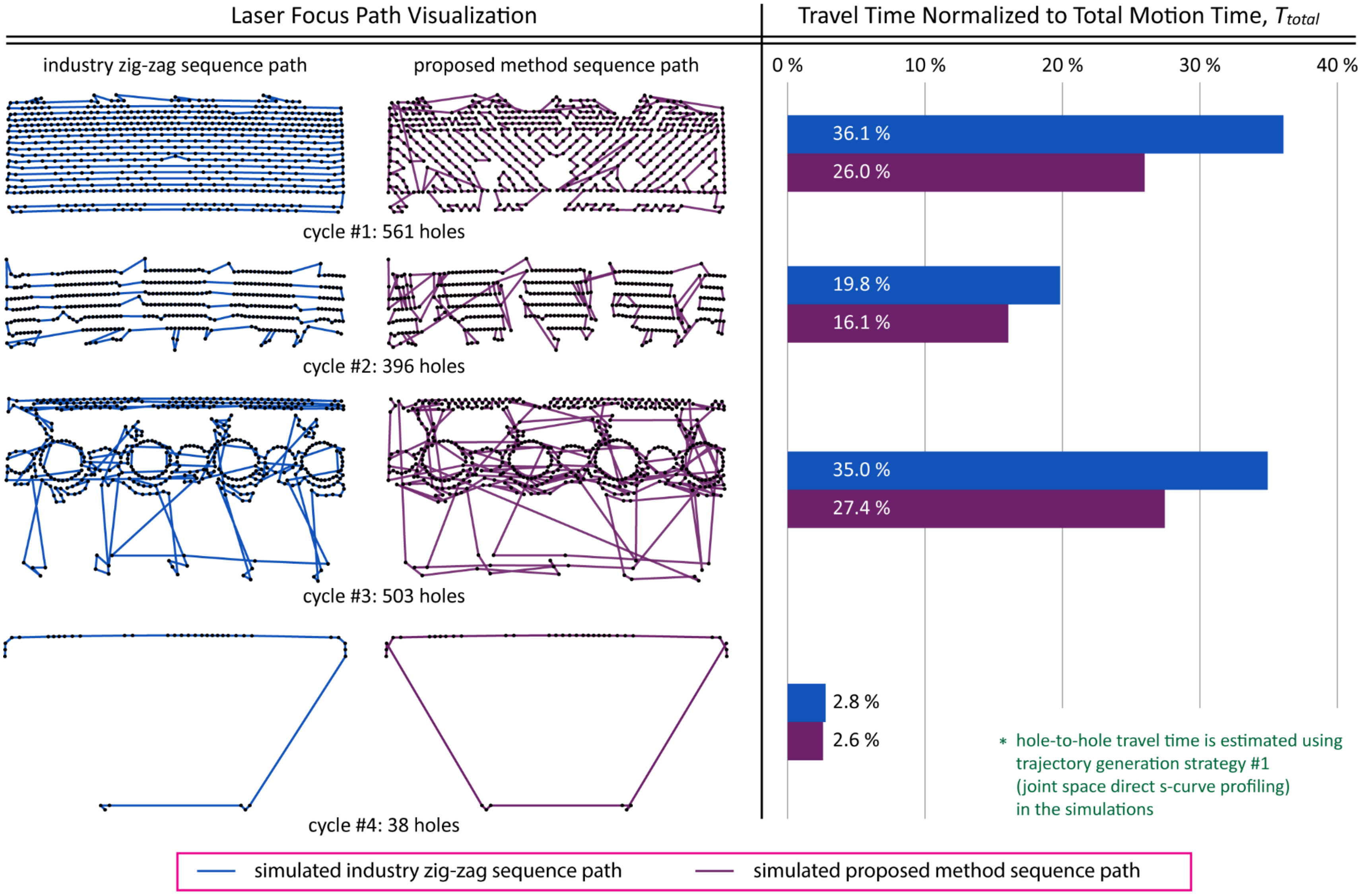


**Figure 6.2 Simulated sequence paths generated from the industry method sequence and the proposed method sequence: cycle #1 with 561 holes, cycle #2 with 396 holes, cycle #3 with 503 holes, and cycle #4 with 38 holes**

It can be shown from Figure 6.2 that the sequence paths generated from the industry strategy certainly resemble the zig-zag pattern. The percent reductions of the travel time achieved from the proposed work are calculated by the following equation and the values are listed in Table 6.1.

$$\frac{time_{industry} - time_{proposed}}{time_{industry}} \times 100\% \tag{6.1}$$

**Table 6.1 Simulated total travel time resulted from the industry sequences and the proposed sequences**

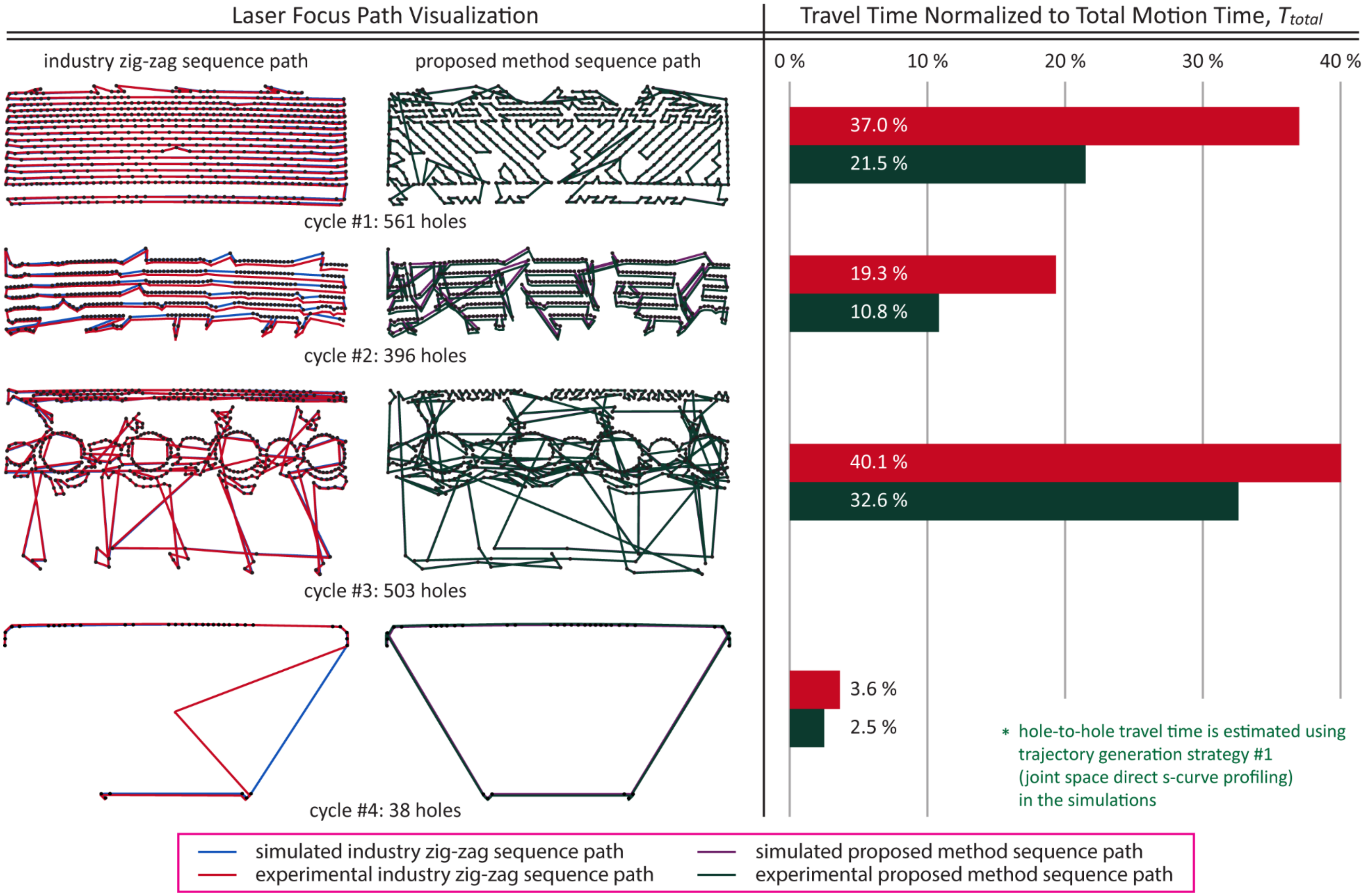
| | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 | Overall |
|---|---|---|---|---|---|
| % reduction in travel time [%] | 28.0 | 19.0 | 21.4 | 6.6 | 23.0 |

As expected, the proposed method successfully reduces the total travel time. Moreover, depending on the hole locations and their drilling patterns, the proposed method can improve the results by 6~28 % for each cycle and 23.0 % in overall.

### 6.1.2 Experimental Results

From the industry sequence and the proposed sequence, relevant NC code is programmed and the commanded trajectories are recorded at 4ms sampling rate. Figure 6.3 illustrates the measured trajectories and the simulated trajectories. The travel times for individual cycles are normalized to the total motion, $T_{total}$.

It can be seen that the measured tool paths and the expected tool paths from the simulations match well, except for the small drifts observed due to the corrections that the CNC controller makes onto the commanded trajectories. One major misalignment observed in the cycle #4 is by a detour, which is programmed directly in the NC code, and hence can be ignored. The percent reductions of the travel time are calculated in the same manner using equation (6.1) and the values are shown in Table 6.2.



* hole-to-hole travel time is estimated using trajectory generation strategy #1 (joint space direct s-curve profiling) in the simulations

**Figure 6.3 Experimentally measured paths generated from the industry method sequence and the proposed method sequence: cycle #1 with 561 holes, cycle #2 with 396 holes, and cycle #3 with 503 holes**

**Table 6.2 Measured total travel time resulted from the industry sequences and the proposed sequences**

| | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 | Overall |
|---|---|---|---|---|---|
| % reduction in travel time [%] | 41.9 | 44.0 | 18.6 | 31.2 | 32.6 |

For each cycle, the total travel time using the proposed work is reduced by 18~44 %. Therefore, it is successfully verified that the proposed work effectively reduces the total travel time in a real environment. However, it is worth to mention that the percent reductions expected from the simulations and measured from the experiments are quite different in terms of the magnitudes and the trend, which can be explained with two reasons.

In this preliminary result section, the motion duration for hole-to-hole trajectory is estimated using the joint space s-curve planning strategy (strategy #1), which is proven to not be the best estimation strategy due to its tendency of overestimation (shorter travel time than the actual). The effect of this is well shown in the cycle #3 result, in which both the normalized travel time for the industry (35.0 %) and the proposed (27.4 %) sequences are smaller than the normalized experimentally measured travel times (40.1 %).

Another crucial reason for the inconsistency comes from the errors in the experiment set up. The NC codes constructed for the experiments have the laser firing commands commented out resulting in G00 commands for the hole locations and orientations stacked one after another. When this NC code is read, the CNC controller automatically blends G00 commands so that there are no stopping motions in between the G00 commands. This phenomenon violates the very stop-and-go nature of multi-point drilling causing the hole-to-hole motion duration to be much lower than expected since there is no need to decelerate for stopping. The effect of this blended G00 commands is well shown in cycle #2. It is expected to have less travel time in the simulation for the reason explained above, however for cycle #2, the travel times are less in the experiments. The industry path from 19.8 % to 19.3 % and the proposed path from 16.1 % to 10.8 %.

Therefore, the reliability of the proposed work is not fully demonstrated by the results from the preliminary section. This leads to a new set of simulations and experiments designed to resolve the issues.

## 6.2 Final Study Results

In this section, the motion durations for hole-to-hole trajectories are estimated by the main estimation strategy, task space s-curve velocity planning via iterative process (strategy #3), to yield more accurate predictions. In addition, the stop-and-go action is enforced by adding the dwelling command, G4, in between G00 commands. Furthermore, the test is designed without dividing up the cycles, by visiting the entire 1498 holes in one run.

Unfortunately, due to the heavy demand of production, acquiring machine time for the experiments has not been successful for this final study at the moment of writing this thesis. Therefore, this section only illustrates the results from the simulations and the experiments are left for future work.

### 6.2.1 Simulation Results

Similar to the preliminary result section, the zig-zag patterned sequence is used to represent the industry approach and the minimum time chained 2-opt optimized sequence is used to represent the proposed approach on sequencing of 1498 holes. The resultant tool paths are shown in Figure 6.4. Again, the travel times are normalized to the total motion, $T_{total}$ and only the relative percentage values are presented.

Note that results from Figure 6.4 are different from Figure 6.3 in that all the holes are visited in one cycle. Also, it is interesting to observe that the proposed sequence path from Figure 6.4 is different from Figure 6.3 since now there are more candidate routes to be considered in the optimization step. Using the same equation (6.1), the percent reduction achieved from the proposed method work is 26.0 %, successfully demonstrating the improvement in cycle time.

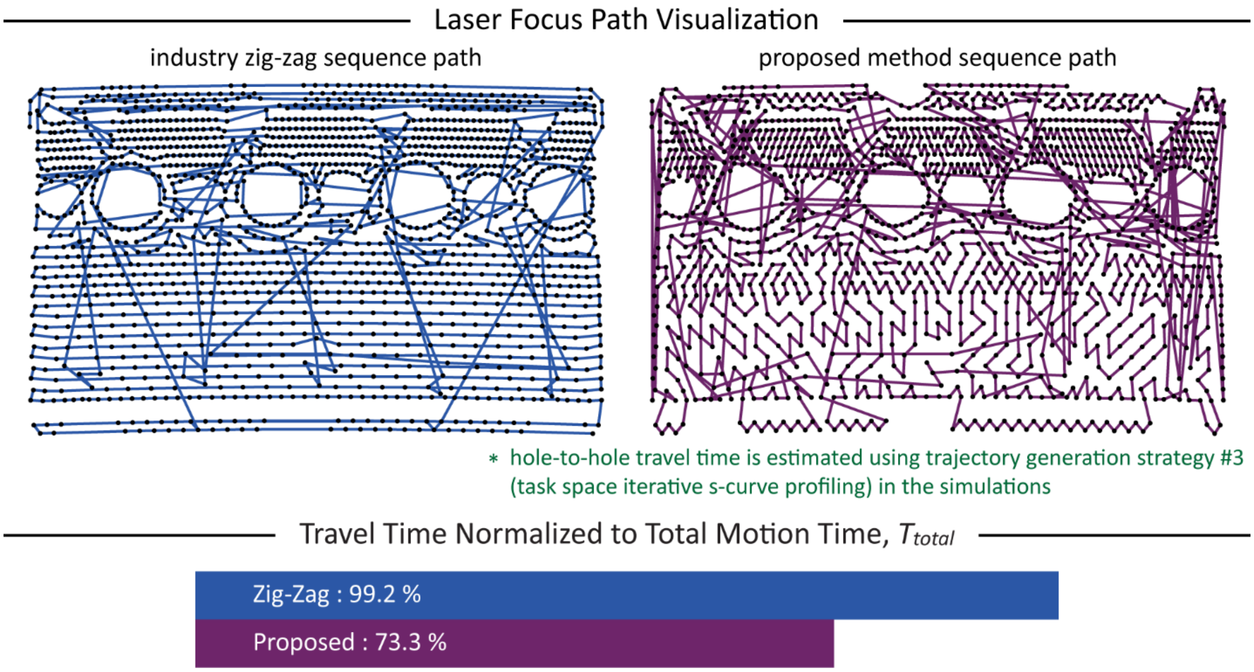


**Figure 6.4 Simulated sequence paths generated from the industry and the proposed method sequences for 1498 hole sample part**

While it is impossible to fully measure the predictability of the proposed work without the real environment experiments, it can be stated that the proposed work in this final study would produce more reflective results than the results from the preliminary study. The normalized total travel time of the industry sequence path estimated using the preliminary study method for this simulation is 94.0 %. Considering that this is an overestimated result and having 99.2 % for the estimated total travel time for the same sequence path, it is likely that the gap between the simulated and experimental results are tighter in this final study.

# Chapter 7
# Conclusions and Future Work

## 7.1 Conclusions

This thesis has presented a method of cycle time reduction for 5-axis percussion laser drilling processes. The reduction of cycle time is achieved by two aspects: time-optimal hole-to-hole trajectory generation and minimum time sequencing of drilling operations. The proposed method combines these two by first calculating the travel times for each hole pair from hole-to-hole trajectory planning and utilizing this temporal information in optimizing the hole visiting sequence to yield overall minimum travel time.

In order to be practically beneficial, time-optimal hole-to-hole trajectory generation in this thesis models the CNC controller's G00 + TRAORI command, which is the minimum time trajectory strategy used by industries. Four strategies have been presented and their performances for estimation accuracy are compared with over 2000 CNC's trajectory samples. As a result, the task space trajectory generation strategy that utilizes the complete jerk bounded s-curve velocity profile for the path parameter is selected as the proposed trajectory generation method to calculate hole-to-hole travel times for each hole pair.

Given the temporal cost information, sequencing optimization problem is formulated into the well-known Traveling Salesman Problem (TSP). In solving the TSP, an algorithm has been developed inspired by the powerful heuristics method called chained Lin-Kernighan. The proposed algorithm is compared to the two existing industrial approaches and is proven to be superior in generating the near-optimal solutions that are within 1 % from the optimal solutions obtained by the state-of-the-art non-commercial TSP solver called Concorde.

Combining the hole-to-hole trajectory generation and the sequence optimization, the proposed method results in about 26 % reduction in cycle time for the sample part in the simulations demonstrating the effectiveness of the proposed work. In addition, the reliability of the proposed method has been partially addressed by having less than 2 % difference in the simulated and experimentally measured total motion time.

## 7.2 Future Work

Future work of this research includes developing a custom time-optimal trajectory generator that performs better than the CNC controller's G00 + TRAORI trajectories. The current strategy is to impose the jerk limited s-curve velocity profiles to the tool path parameter in the task spatial domain, of which the s-curve limits are transformed from the joint spatial domain kinematic limits. The limitation to this is that the resultant motion, when transformed back to the joint space, might not be time-optimal. This is demonstrated in Chapter 4 where the optimization approach trajectory under the same kinematic limits and the same constrained tool path results in shorter time. This shows the possibilities of developing custom hole-to-hole trajectory planning methods that result in shorter travel time than the CNC's G00 + TRAORI motions.

The sequencing algorithm can also be improved further by integrating collision avoidance function. The current algorithm assumes that the maneuver between every hole pair is feasible, however, this is not the case in the real environment due to collisions. These collisions could arise between the tool tip and the tool fixture including the clamps. Therefore, resultant optimized sequence may not be feasible if any hole-to-hole path induces a collision. For future improvement, a collision detection step can be implemented before optimizing sequences to detect the hole pairs causing collisions and the improved algorithm would force these pairs from being selected during the optimization phase.

# References

[1] DMGMORI, "LASERTEC Series," 2019. [Online]. Available: https://ca-en.dmgmori.com/resource/blob/45194/e0b0d41f020241497e1d63b1e9c2154b/pl0uk-lasertec-series-pdf-data.pdf. [Accessed Feb. 5, 2019].

[2] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal on Robotics and Automation,* vol. 3, *(2),* pp. 115-123, 1987.

[3] J. E. Bobrow, S. Dubowsky and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research,* vol. 4, *(3),* pp. 3-17, 1985.

[4] D. Verscheure *et al*, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control,* vol. 54, *(10),* pp. 2318-2327, 2009.

[5] V. YEN, "An inverse dynamic-based dynamic programming method for optimal point-to-point trajectory planning of robotic manipulators," *Int. J. Syst. Sci.,* vol. 26, *(2),* pp. 181-195, 1995.

[6] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *J. Robot. Syst.,* vol. 17, *(5),* pp. 233-249, 2000.

[7] C. G. Lo Bianco and A. Piazzi, "Minimum-time trajectory planning of mechanical manipulators under dynamic constraints," *Int J Control,* vol. 75, *(13),* pp. 967-980, 2002.

[8] Q. Zhang, S. Li and X. Gao, "Practical smooth minimum time trajectory planning for path following robotic manipulators," in *American Control Conference (ACC), 2013,* Wasington, DC, USA, 2013, pp. 2778-2783.

[9] Q. Zhang and M. Zhao, "Minimum time path planning of robotic manipulator in drilling/spot welding tasks," *Journal of Computational Design and Engineering,* vol. 3, *(2),* pp. 132-139, 2016.

[10] Q. Zhang *et al*, "Time-optimal path tracking for robots under dynamics constraints based on convex optimization," *Robotica,* vol. 34, *(9),* pp. 2116-2139, 2016.

[11] H. Makino and T. Ohde, "Motion control of the direct drive actuator," *CIRP Annals-Manufacturing Technology,* vol. 40, *(1),* pp. 375-378, 1991.

[12] Y. Tomita *et al*, "High-response XY stage system driven by in-parallel linear motors," *CIRP Annals-Manufacturing Technology,* vol. 45, *(1),* pp. 359-362, 1996.

[13] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation," *Int. J. Mach. Tools Manuf.,* vol. 41, *(9),* pp. 1323-1345, 2001.

[14] L. Biagiotti and C. Melchiorri, "FIR filters for online trajectory planning with time-and frequency-domain specifications," *Control Eng. Pract.,* vol. 20, *(12),* pp. 1385-1399, 2012.

[15] S. Tajima, B. Sencer and E. Shamoto, "Accurate interpolation of machining tool-paths based on FIR filtering," *Precis Eng,* vol. 52, pp. 332-344, 2018.

[16] W. Fan *et al*, "Time-optimal interpolation for five-axis CNC machining along parametric tool path based on linear programming," *The International Journal of Advanced Manufacturing Technology,* vol. 69, *(5-8),* pp. 1373-1388, 2013.

[17] Y. Altintas and K. Erkorkmaz, "Feedrate optimization for spline interpolation in high speed machine tools," *CIRP Annals-Manufacturing Technology,* vol. 52, *(1),* pp. 297-302, 2003.

[18] H. Liu, X. Lai and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robot. Comput. Integrated Manuf.,* vol. 29, *(2),* pp. 309-317, 2013.

[19] J. Dong, P. M. Ferreira and J. A. Stori, "Feed-rate optimization with jerk constraints for generating minimum-time trajectories," *Int. J. Mach. Tools Manuf.,* vol. 47, *(12-13),* pp. 1941-1955, 2007.

[20] S. Macfarlane and E. A. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Trans. Rob. Autom.,* vol. 19, *(1),* pp. 42-52, 2003.

[21] B. Sencer, Y. Altintas and E. Croft, "Feed optimization for five-axis CNC machine tools with drive constraints," *Int. J. Mach. Tools Manuf.,* vol. 48, *(7-8),* pp. 733-745, 2008.

[22] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *1988 IEEE International Conference on Robotics and Automation,* 1988, pp. 364-369.

[23] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robot. Comput. Integrated Manuf.,* vol. 24, *(3),* pp. 415-426, 2008.

[24] D. L. Applegate *et al*, *The Traveling Salesman Problem: A Computational Study.* Princeton university press, 2006.

[25] E. L. Lawler *et al*, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* New York: Wiley-Interscience Series in Discrete Mathematics, 1985.

[26] W. J. Cook. “TSP History,” 2005. [Online]. Available: http://www.math.uwaterloo.ca/tsp/history/pictorial/dfj.html. [Accessed Jan. 27, 2019].

[27] G. Dantzig, R. Fulkerson and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the Operations Research Society of America,* vol. 2, *(4),* pp. 393-410, 1954.

[28] W. J. Cook, *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation.* Princeton university press, 2011.

[29] D. Applegate *et al*, "On the solution of traveling salesman problems," in *1988 International Congress of Mathematicians (ICM)*, 1988, vol. 3, pp. 645-656.

[30] D. L. Applegate *et al*, "Certification of an optimal TSP tour through 85,900 cities," *Oper. Res. Lett.,* vol. 37, *(1),* pp. 11-15, 2009.

[31] P. Alwis *et al*, "Automated printed circuit board (PCB) drilling machine with efficient path planning," in *SAITM Research Symposium on Engineering Advancements,* 2014, pp. 159-162.

[32] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Oper. Res.,* vol. 21, *(2),* pp. 498-516, 1973.

[33] W. J. Cook, "World TSP," 2018. [Online]. Available: http://www.math.uwaterloo.ca/tsp/world/index.html. [Accessed Jan. 27. 2019].

[34] R. Aciu and H. Ciocarlie, "G-code optimization algorithm and its application on printed circuit board drilling," in *2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI),* 2014, pp. 43-47.

[35] L. Qu and R. Sun, "A synergetic approach to genetic algorithms for solving traveling salesman problem," *Inf. Sci.,* vol. 117, *(3-4),* pp. 267-283, 1999.

[36] K. Honda, Y. Nagata and I. Ono, "A parallel genetic algorithm with edge assembly crossover for 100,000-city scale TSPs," in *2013 IEEE Congress on Evolutionary Computation (CEC),* 2013, pp. 1278-1285.

[37] Q. Zhang and M. Zhao, "Minimum time path planning of robotic manipulator in drilling/spot welding tasks," *Journal of Computational Design and Engineering,* vol. 3, *(2),* pp. 132-139, 2016.

[38] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics,* Springer 2019, pp.311-351.

[39] O. Montiel-Ross *et al*, "Methodology to optimize manufacturing time for a CNC using a high performance implementation of ACO," *International Journal of Advanced Robotic Systems,* vol. 9, *(4),* pp. 121, 2012.

[40] A. Alzaydi, "Time-optimal trajectory generation and way-point sequencing for 5-axis laser drilling," Ph.D. dissertation, Dept. Mechanical and Mechatronics Eng., Univ. Waterloo, Waterloo, ON, Canada, 2016.

[41] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge university press, 2004.

[42] Q. Zhang *et al*, "Time-optimal path tracking for robots under dynamics constraints based on convex optimization," *Robotica,* vol. 34, *(9),* pp. 2116-2139, 2016.

[43] D. Applegate, W. Cook and A. Rohe, "Chained Lin-Kernighan for large traveling salesman problems," *INFORMS Journal on Computing,* vol. 15, *(1),* pp. 82-92, 2003.

[44] D. S. Johnson, "Local optimization and the traveling salesman problem," in *International Colloquium on Automata, Languages, and Programming,* 1990, pp. 446-461.