

Robot-Assisted Endodontic Treatment Based on Force-Guided Alignment and File Feedrate Control

Yi-Chan Li

Advisor: Cheng-Wei Chen

Graduate Institute of Electronics Engineering

National Taiwan University

Taipei, Taiwan

June 2021

Abstract

- 1.introduction: present tense
- 2.method and result: past tense
- 3.conclusion : present tense

Contents

Abstract	i
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Motivation	1
1.2 Previous Work and Problem Definition	2
1.3 The Proposed Method	6
1.4 Main Contributions of the Thesis	6
1.5 Organization of the Thesis	7
2 State-of-the-Art	9
3 Design and Analysis of the Dental Surgical Robot - DentiBot	11
3.1 Requirement and Specification	11
3.2 Design of the DentiBot	12

3.3 Kinematics Analysis	13
3.3.1 Coordinate Definition	14
3.3.2 Forward and Inverse Kinematics	14
3.3.3 Jacobian matrix	16
3.4 Coordinate Transformation of the Robot Arm	22
3.4.1 Translation Analysis - Tool Center Point	23
3.4.2 Rotation Analysis	26
4 Force-Guided Robot Alignment	29
4.1 Problem Definition	30
4.2 Integration of F/T sensor	31
4.3 Dragging Mode	39
4.3.1 Admittance Control based on F/T sensor	40
4.4 Self-Alignment Mode	44
4.4.1 Coordinate Transformation of F/T sensor	45
4.4.2 Motion Planning: Based on Admittance Control	47
4.5 Affections of Parameters Setting	48
5 File Feedrate Control	53
5.1 Problem Definition	53
5.2 The Proposed Method and Theorem	55
5.2.1 Feedrate Control - Inverse Rotation Control	57
5.2.2 Feedrate Control - Feed Control	59

6 Preliminary Experiment Result	61
6.1 Experimental Setup	61
6.2 Admittance Control	61
6.3 Automatically Direction Changing	62
6.4 Repetitive Experiment	62
7 Conclusions and Future works	63
7.1 Conclusions	63
7.2 Discussion and Future Works	64
8 Appendix	67
8.1 Forward Kinematics	67
8.2 Jacobian matrix	69
8.2.1 Jg0	70
8.2.2 Jg6	73
Reference	75

List of Figures

1.1	The endodontic therapy steps	3
3.1	The DentiBot and an acyclic tooth model	12
3.2	Coordinate Definition	14
3.3	Schematic diagram for Tool Center Point. The translation vector ${}^6p_{H_{org}}$ denotes the origin position relative to the frame{6}.	24
3.4	Schematic diagram for obtaining the tool vector.	26
3.5	Illustration of finding the rotation matrix	27
4.1	Data Analysis of F/T sensor.	31
4.2	Control scheme. f_d denotes the desired forces and torques vector. f_s denotes the real value detected by F/T sensor and is also a forces and torques vector. \dot{x} denotes $[\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z]$. J_g denotes the geometric Jacobian matrix. \dot{q} denotes $[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6]$. q denotes $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$.	40
4.3	Illustration of changing reference frame from {S} to {T}. f and m respectively denote a force and a moment measured at point p.	45

4.4 Illustration of changing reference frame from {S} to {T}. <i>f</i> and <i>m</i> respectively denote a force and a moment measured at point p.	46
4.5 Control scheme. M , B , K are all diagonal positive definite matrices, whose diagonal indexes are respectively related to $x, y, z, \theta_x, \theta_y, \theta_z$. M , B are related to the inertial and damping respectively, and K is a proportional gain.	48
4.6 Holding gesture in "Dragging Mode"	51
5.1 The illustration of an endodontic file	54
5.2 The first prototype of DentiBot	57
5.3 Motion planning of Inverse Rotation Control.	58
5.4 Flow chart of Feed Control	59

List of Tables

3.1	Denavit-Hartenberg parameters of Meca500	15
4.1	Arc-function Comparison	38
4.2	Moving commands in Meca500	42
4.3	Parameters setting of Admittance control.	50

Chapter 1

Introduction

The robot-assisted system for endodontic treatment - DentiBot is presented in this thesis. The definition of a robot-assisted system in the thesis refers to be a dental assistant. That means we wish DentiBot could help dentists perform better clinical results. This chapter will give brief introductions of the endodontic treatment, previous work, problem definition, and the proposed method.

1.1 Motivation

A qualified dentist with a certification can operate an endodontic treatment. They accumulate their experience, thereby increasing the success rate of surgery. With enough clinical experience, the dentist can acquire an endodontist license. According to statistics from the Ministry of Health and Welfare, R.O.C. (Taiwan) [1], the number of dentists in Taiwan is 15,178. However, according to The Academy of Endodontology, R.O.C. (Taiwan) [2], there are only 238 dentists

to acquire an endodontist license. Thus, performing an endodontic treatment requires the dentist's self expertise of endodontics. The performance of endodontic treatment depends on a dentist's long-term experience.

Besides, root canal treatment is a tedious and time-consuming surgery for a dentist due to different complicated conditions of each tooth. A patient who suffered from an infected tooth spends countless hours see a dentist. Endodontic treatment takes at least two to three rounds of treatments, even more than two months in the worst case.

Therefore, our team looks forward to designing a robot-assisted system to accomplish a root canal treatment. With the robot, we wish it can reduce times for entire treatment, increase the success rate of root canal treatment for dentists, and provide patients safer surgery.

1.2 Previous Work and Problem Definition

First, we introduce the endodontic treatment and its detailed procedure.

Endodontic treatment, also known as root canal treatment and nerve extraction, is performed to cure an infected tooth. The main procedure of endodontic treatment is divided into three parts - Opening, Cleaning, and Filling shown in Figure 1.1.

An infected tooth arises from periodontal disease, attrition, trauma, or decay. Once the dental pulp is infected, it causes an irreversible inflammation and lets patients confront a root canal treatment. Figure 1.1 shows an infected tooth and its dental pulp, which consisted of blood vessels, nerves, connective tissues, and

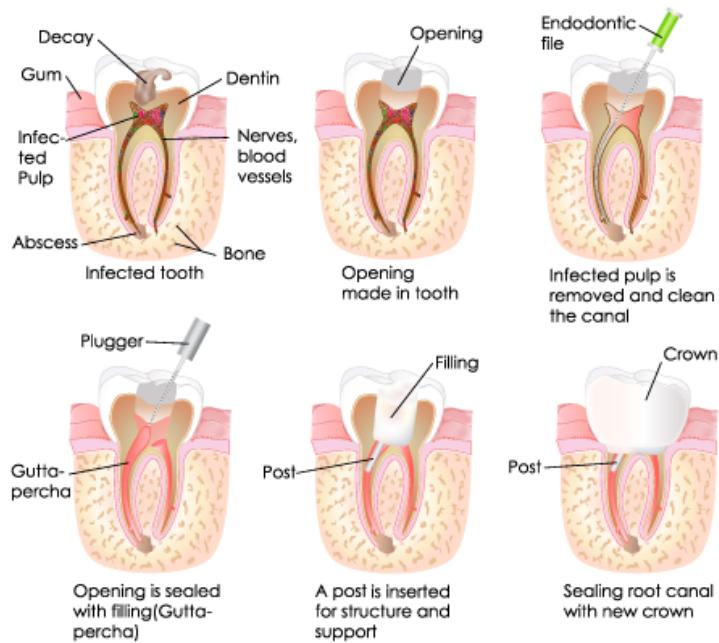


Figure 1.1: The endodontic therapy steps

lymphatics. In the "Opening" step, an experienced dentist drills the crown of the infected tooth to remove the dentin and expose the infected pulp inside the canal to the air. Next, in the "Cleaning" step, the dentist uses an endodontic file, a superelastic root reamer, to remove the infected pulp. It is necessary to ensure that there is no remained infected pulp. Then, in the "Filling" step, the dentist uses a dental plugger to fill the empty root canal with Gutta-percha, a plastic substance. "Filling" can prevent cross-infection between root canals because the cured tooth remains many invisible and inaccessible pulp tissue. Finally, the dentist seals the root canal with a new crown to protect cured root canal.

As stated above, "Cleaning" is of paramount importance in a whole treatment because cleaning improperly will result in pulp necrosis, apical abscess, periodontal

ligament inflammation, or even cellulitis. If there are many remained infected pulp after root canal treatment, the surgery should be operated on again. The "Cleaning" procedure is a big challenge in and of itself. Therefore, we should figure out how to enable the robot to assist dentists and perform the root canal treatment.

Previous Work

There are more and more robots which applied to specific surgery. In the dental field, the majority of robotic applications are in implant surgery. The researchers in Chosun University built a dental implant robot [3], a remote center of motion (RCM) mechanism. Li, J. et al. designed a robotic system using a soft bracing technique to drill teeth [4]. Also, there is the first commercial implant robot, YOMI, developed by Neosis [5]. However, there was one and only one robot for the endodontic treatment. The domestic researcher Janet Dong and his team proposed a microrobot performing root canal treatment with the assistance of a 3D computer model system. However, the study using 3D model belongs to pre-operation. If a patient or an image error causes a movement, it will not be easy to reschedule the motion planning because the root canal in a tooth is unseen by a camera from any angle. Besides, once an endodontic file enters into a root canal, it is hard to obtain the tool tip position due to its flexibility. The endodontic file will bend when it bears a force. Once we lose the position of the endodontic file, it may lead to perforation, overpreparation, and underpreparation. Therefore, a problem reveals - how the robot know the path of a root canal without visual feedback?

On the other hand, instrument fracture is the other concern during the therapy. If an endodontic file suffered excessive usage, it would unpredictably break. The leading causes of fractured files are torsional fracture and flexural fatigue, which account for 55.7% and 44.3% separately [6]. Removal of broken files is technically tricky, so it is essential to reduce the probability of the instrument fracture. Therefore, our robot should prevent the instrument fracture by some detection. In addition, the root canal treatment also requires repeatedly drilling to clean the canal thoroughly and motion planning to avoid perforation. This repetitive action of root canal treatment is tedious and time-consuming. Therefore, we decide to design an automatic endodontic robot. It can improve time efficiency and prevent instrument fracture in endodontic treatment.

Problem Definition

To sum up, there are three main problems.

1. How to assist dentists to perform the root canal treatment including motion planning, especially in the second procedure - cleaning?
2. How to overcome the problem that the root canal cannot be visually observed, and how to clean well in any complex conditions?
3. How to protect the endodontic file from fracturing during the surgery?

1.3 The Proposed Method

To assist dentists in accomplishing an endodontic treatment, we build a robot

- DentiBot to provide more precise and safer treatment. DentiBot consists of a robot arm, a force/torque sensor, and a modified end effector. They make DentiBot satisfy the requirements of the endodontic treatment, it will be discussed in Section 3.1.

Thanks to the robot arm and the modified end effector, we could make the system manifest various motions such as drilling and reciprocation. It enables the DentiBot to assist dentists in performing the second procedure - cleaning.

To overcome the problem that the root canal cannot be visually observed. We use force-guided alignment with the feedback of the F/T sensor. That means the Dentibot could align the root canal path by force feedback without vision feedback.

To protect the endodontic file from fracturing during the endodontic treatment, we use current feedback to keep track of the file's torque. With the torque feedback, we can regulate the feedrate of the endodontic file to control the file torque. It successfully prevent instrument fracture.

1.4 Main Contributions of the Thesis

Prospect

There are four parts of the robot-assisted project.

1. Dentist could move the DentiBot to the infected tooth.

2. The DentiBot could search all root canals of the infected tooth.
3. The DentiBot could do repetitive drilling and clean thoroughly.
4. The DentiBot continues drilling until detecting the apex of the root canal.

Main Contributions

The entire endodontic project is huge. The thesis is the beginning of the project and focus on the first and the third parts of the project. In conclusion, there are three main contributions of the thesis.

1. Integrate a 6-DoF robotic manipulator with 6-DoF F/T sensor for performing endodontic treatment.
2. Develop a framework for robot alignment regarding the position and orientation of root canal.
3. Protect the endodontic file from fracturing by controlling file rotation speed.

1.5 Organization of the Thesis

The structure of this thesis is as follows.

Chapter 2

State-of-the-Art

A. YOMI – commercial robot B. HK - dental implant robot C. Korean - dental implant robot D. NCTU – RCT robot It utilizes the 3D model to plan the corresponding path to accomplish an endodontic treatment. YOMI has received the clearance from FDA and has performed more than 2,700 times surgery in the USA. In the pandemic of Covid-19 in 2020, YOMI provided non-contact surgery between dentists and patients due to its automatic robotic system.

Chapter 3

Design and Analysis of the Dental Surgical Robot - DentiBot

3.1 Requirement and Specification

The dental surgical robot should enable dentists to perform delicate and complicated surgery operations because the average diameter of a root canal is $0.28(\pm 0.08)$ mm [7]. Therefore, our system should have a high resolution of movement. Next, an appropriate workspace is required. In dental anatomy, teeth are located on the maxillary (lower jaw) and the mandibular (upper jaw). To perform surgery with both sides, we should rotate the end effector at least 180 degrees. Also, there is a previous research, which shows the average range of maximum mouth opening is $50.3 \text{ mm} \pm 6.26 \text{ mm}$ [8]. Hence, the end effector of our system should be less than this range.

3.2 Design of the DentiBot

As discussed in the previous section, we decide to build a system composed of a robot arm, an F/T sensor, and a modified handpiece. Why we choose a robot arm and an F/T sensor is because we want to mimic the dentist's motion. Due to the 6-DOF robot arm, we can achieve the action with less hardware restriction. We can easily move to almost desired positions and rotate more than 180 degrees to drill tooth in both sides of mouth. Also, with the 6-DoF F/T sensor, we can take the real-time force and torque feedbacks as haptic feedback. Our system can take reaction with the F/T sensor such as a dentist touch something in the surgery and do the corresponding reaction. Besides, by modifying the existing handpiece, which is a handheld dental electric device, we do not worry about the workspace in the mouth.

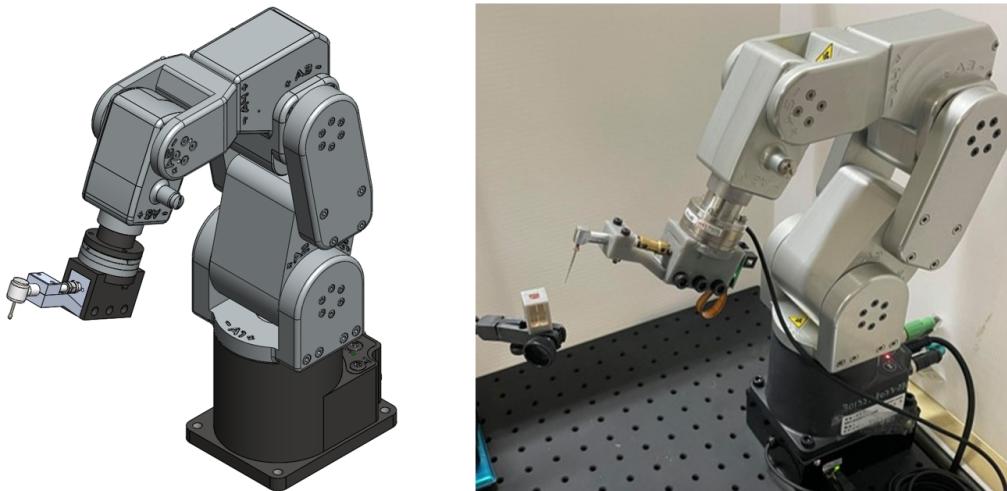


Figure 3.1: The DentiBot and an acyclic tooth model

To meet the requirement of workspace we have to select suitable devices

according to those requirements as shown in Fig 3.1. First, We choose Meca500 manufactured by Mecademic Inc. as our 6 DOF robot arm [9]. Its feature is high repeatability (precision: 5 μm), and it is equipped with zero-backlash speed reducers. In addition, it is compact and portable for laboratory investigation. Second, Mini40 manufactured by ATI Inc. is the corresponding F/T sensor with three force and three torque detections [10]. As for the end effector, we modify an existing dental handpiece that equips a tool change mechanism. The modified handpiece weighs around 139 grams. We also design adapters to assemble these devices.

Therefore, DentiBot totally has seven degrees of freedom. Six degree of freedom come from Meca500, and the other thanks to our modified handpiece. The rotation of the root canal file is driven by a servo motor whose maximum rotation speed is more than 600 rpm.

3.3 Kinematics Analysis

The purpose of this section and section 3.4 is to serve as a tutorial and provide some crucial approaches when combining a robot arm and an end effector. We derive the forward and inverse kinematics in section 3.3.2 and describe the Jacobian matrix in section 3.3.3.

3.3.1 Coordinate Definition

In Fig 3.2 , we define frame{0} to frame{6} which represent each frame of axes of the Meca500, frame{S} which represent the frame of the ATI-mini40 and frame{H} which represent the frame of the handpiece.

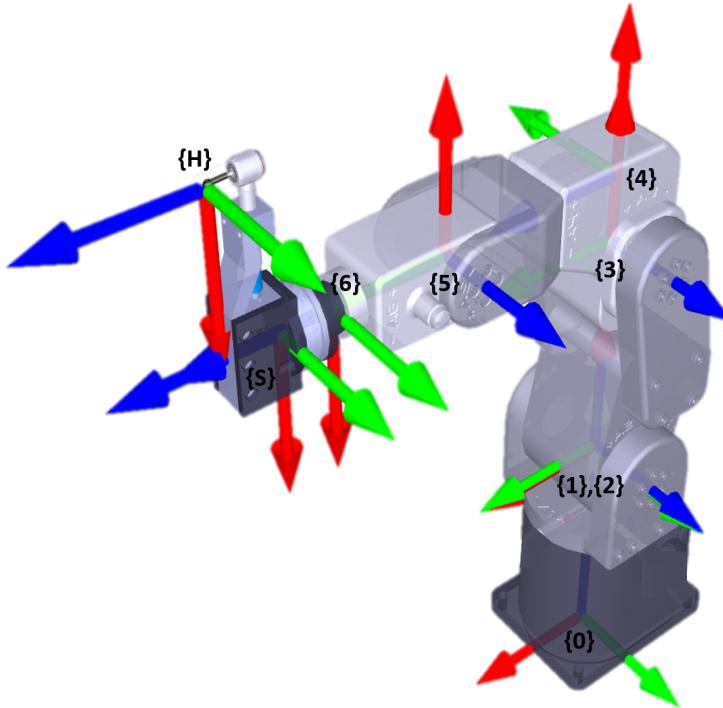


Figure 3.2: Coordinate Definition

3.3.2 Forward and Inverse Kinematics

Denavit-Hartenberg parameters are shown as Table 3.1. Then, the forward kinematics of Meca500 is derived as

$${}^0\mathbf{T} = {}^0\mathbf{T} \cdot {}^1\mathbf{T} \cdot {}^2\mathbf{T} \cdot {}^3\mathbf{T} \cdot {}^4\mathbf{T} \cdot {}^5\mathbf{T} = \begin{bmatrix} {}^0\mathbf{R} & {}^0\mathbf{p}_{6\text{org}} \\ 0 & 1 \end{bmatrix} \quad (3.1)$$

Table 3.1: Denavit-Hartenberg parameters of Meca500

i (link number)	α_{i-1} (deg)	a_{i-1} (mm)	θ_i (deg)	d_i (mm)
1	0	0	θ_1	135
2	-90	0	θ_2	0
3	0	135	θ_3	0
4	-90	38	θ_4	120
5	90	0	θ_5	0
6	-90	0	θ_6	70

where ${}^0_6\mathbf{R}$ is the rotation matrix from frame{6} to frame{0}, ${}^0\mathbf{p}_{6\text{org}}$ is the origin of the frame{6} observed from frame{0}. All detailed indexes of ${}^0_6\mathbf{T}$ are shown as

Appendix 8.1

Incidentally, there is an alternative to calculate the transformation matrix of Meca500. We can use command "GetPose" to obtain $(x,y,z,\alpha,\beta,\gamma)$. Then, we can

use this information to derive the following equation.

$$\begin{aligned}
 {}^0_6\mathbf{T} &= \begin{bmatrix} & & & x \\ \mathbf{R}_x(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_z(\gamma) & & & y \\ & & & z \\ & 0 & & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta & x \\ c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -s_\alpha c_\beta & y \\ s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma & s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}
 \end{aligned}$$

where C_* , S_* denote $\cos(\star)$, $\sin(\star)$; α, β, γ are in representation of Euler angle.

3.3.3 Jacobian matrix

Here we evaluate geometric Jacobian based on frame{0}, geometric Jacobian based on frame{6} and analytical Jacobian.

First and foremost, we should clarify the difference between geometric Jacobian and analytical Jacobian. They both use the same linear velocity but consider different angular velocity. The angular velocity which geometric Jacobian applies is relevant to the axis angles ($\theta_x, \theta_y, \theta_z$). In contrast, the angular velocity which analytical Jacobian contemplates is related to the orientation (α, β, γ) of the end effector.

Geometric Jacobian Based on Frame{0}

Before examining the geometric Jacobian matrix, it will be necessary to find the relationship between the position and joints' angles and the relationship between the axis angle and the joints' angles.

Based on the translation matrix in Equation 3.3.2, we obtain the relationship between the position and joints' angles.

$${}^0\mathbf{p}_{6\text{org}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x(\theta_1, \theta_2, \dots, \theta_6) \\ y(\theta_1, \theta_2, \dots, \theta_6) \\ z(\theta_1, \theta_2, \dots, \theta_6) \end{bmatrix} \quad (3.3)$$

Moreover, we dissect the relationship between the axis angle and joints' angles.

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = {}^0{}_1\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \theta_1 \end{bmatrix} + {}^0{}_2\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \theta_2 \end{bmatrix} + {}^0{}_3\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \theta_3 \end{bmatrix} + {}^0{}_4\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \theta_4 \end{bmatrix} + {}^0{}_5\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix} + {}^0{}_6\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \theta_6 \end{bmatrix} \quad (3.4)$$

We can differentiate Equation 3.3 and 3.4 to obtain Jacobian matrices.

$$\mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{J}_{\mathbf{gv}} \cdot \dot{\mathbf{q}}, \quad \mathbf{w} = \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \mathbf{J}_{\mathbf{gw}} \cdot \dot{\mathbf{q}} \quad (3.5)$$

where

$$\mathbf{J}_{\text{gv}} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \cdots & \frac{\partial x}{\partial \theta_6} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \cdots & \frac{\partial y}{\partial \theta_6} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \cdots & \frac{\partial z}{\partial \theta_6} \end{bmatrix}, \quad \mathbf{J}_{\text{gw}} = \begin{bmatrix} \frac{\partial \theta_x}{\partial \theta_1} & \frac{\partial \theta_x}{\partial \theta_2} & \cdots & \frac{\partial \theta_x}{\partial \theta_6} \\ \frac{\partial \theta_y}{\partial \theta_1} & \frac{\partial \theta_y}{\partial \theta_2} & \cdots & \frac{\partial \theta_y}{\partial \theta_6} \\ \frac{\partial \theta_z}{\partial \theta_1} & \frac{\partial \theta_z}{\partial \theta_2} & \cdots & \frac{\partial \theta_z}{\partial \theta_6} \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix}$$

As a result, the geometric Jacobian matrix based on frame0 ${}^0\mathbf{J}_g$ is derived as

$$\dot{\mathbf{x}} = {}^0\mathbf{J}_g \cdot \dot{\mathbf{q}}, \quad \dot{\mathbf{q}} = {}^0\mathbf{J}_g^{-1} \cdot \dot{\mathbf{x}} \quad (3.6)$$

where

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix}_{\{0\}}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix} \quad (3.7)$$

There is a further derivation. In terms of the Jacobian matrix, we can derive the

other property.

$$\boldsymbol{\tau} = {}^0\mathbf{J}_g^\top \cdot \mathbf{f} \Leftrightarrow \begin{bmatrix} \tau_{\theta_1} \\ \tau_{\theta_2} \\ \tau_{\theta_3} \\ \tau_{\theta_4} \\ \tau_{\theta_5} \\ \tau_{\theta_6} \end{bmatrix} = {}^0\mathbf{J}_g^\top \cdot \begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (3.8)$$

where $\boldsymbol{\tau}$ is the vector of joints' torques and \mathbf{f} is the vector composed of inertial forces and torques of the robot arm.

Geometric Jacobian Based on Frame{6}

Why we need geometric Jacobian based on frame{6} is that in section 4.3.1 we will apply admittance control and it will use F/T sensor mounted on frame{6} instead of frame{0} to detect forces and torques. To start with Equation 3.6

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix}_{\{0\}} = {}^0\mathbf{J}_g \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix} \quad (3.9)$$

Then, left-multiply a matrix.

$$\begin{bmatrix} {}^0\mathbf{R} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^0\mathbf{R} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix}_{\{0\}} = \begin{bmatrix} {}^0\mathbf{R} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^0\mathbf{R} \end{bmatrix} {}^0\mathbf{J}_g \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix} \quad (3.10)$$

According to the transformation coordinate relationship between frame{0} and frame{6},

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix}_{\{6\}} = \begin{bmatrix} {}^0\mathbf{R} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^0\mathbf{R} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix}_{\{0\}} \quad (3.11)$$

substitute it into Eq 3.10, we can observe an essential equation.

$${}^6\mathbf{J}_g = \begin{bmatrix} {}^0\mathbf{R} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^0\mathbf{R} \end{bmatrix} \cdot {}^0\mathbf{J}_g \quad (3.12)$$

Notably,

$$\dot{x} = {}^6\mathbf{J}_g \cdot \dot{q}, \quad \dot{q} = {}^6\mathbf{J}_g^{-1} \cdot \dot{x} \quad (3.13)$$

where

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix}_{\{6\}}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix}$$

Analytical Jacobian

The linear velocity of analytical Jacobian and geometric Jacobian is the same as shown in Eq 3.3. Nevertheless, as for angular velocity, analytical Jacobian takes the orientation (α, β, γ) of the end effector into consideration. First of all, we investigate the relationship between the axis angle and the orientation of the end effector as following.

$$\begin{aligned} \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} &= \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix} + R_x(\alpha) \begin{bmatrix} 0 \\ \beta \\ 0 \end{bmatrix} + R_x(\alpha)R_y(\beta) \begin{bmatrix} 0 \\ 0 \\ \gamma \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & S_\beta \\ 0 & C_\alpha & -S_\alpha C_\beta \\ 0 & S_\alpha & C_\alpha C_\beta \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \end{aligned} \quad (3.14)$$

Then, utilize this generalized vector to get its Jacobian matrix \mathbf{J}_{we} .

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \mathbf{J}_{\text{we}} \cdot \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (3.15)$$

where

$$\mathbf{J}_{\text{we}} = \begin{bmatrix} 1 & \gamma C_\beta & S_\beta \\ -\beta S_\alpha - \gamma C_\alpha C_\beta & C_\alpha + \gamma S_\alpha S_\beta & -C_\beta S_\alpha \\ \beta C_\alpha - \gamma C_\beta S_\alpha & S_\alpha - \gamma C_\alpha S_\beta & C_\alpha C_\beta \end{bmatrix} \quad (3.16)$$

Therefore,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{J}_{\text{we}} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (3.17)$$

Finally, we obtain the relationship between geometric and analytical Jacobian.

$$\mathbf{J}_g = \begin{bmatrix} \mathbf{I}_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{J}_{\text{we}} \end{bmatrix} \mathbf{J}_a, \quad \mathbf{J}_g = \begin{bmatrix} \mathbf{I}_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{J}_{\text{we}}^{-1} \end{bmatrix} \mathbf{J}_a \quad (3.18)$$

3.4 Coordinate Transformation of the Robot Arm

So far, with the forward and inverse kinematics, the robot arm can translate and rotate around frame{6}. However, we do not see the origin of the frame{6}

as an operating point. Because the F/T sensor and a detachable end effector will be both mounted on the wrist, the tooltip position is what we want. That means we should let the robot arm know how to translate and rotate in frame{ T } instead of frame{6}. If we have translation and rotation information of the tooltip, there is an easy way to directly give the above information to the robot arm. SetTRF $(x,y,z,\alpha,\beta,\gamma)$ is the command of the robot arm, whose (x,y,z) is translation vector and (α,β,γ) is rotation vector in representation of Euler angle .

To obtain translation and rotation vector, we respectively introduce Tool Center Point in section 3.4.1 to find the translation vector and propose an approach in section 3.4.2 to find the rotation vector.

3.4.1 Translation Analysis - Tool Center Point

Tool Center Point (TCP) is a critical problem for robot arm control [11]. In the previous section, we have calculated the forward and inverse kinematics of the robot arm. By Calculating kinematics, we can keep track of the origin of the frame{6}, which is observed from the base frame. The robot arm has the capability to translate and rotate with the origin of the frame{6}. The overhead motions are like a remote center motion (RCM). We should find the position of the tooltip and make it be an RCM point. Nevertheless, it's inefficient to recalculate the transformation matrix via mechanism dimension when changing an end effector or a tool (root canal reamer).

In order to overcome this problem, we interpret the four-points method to

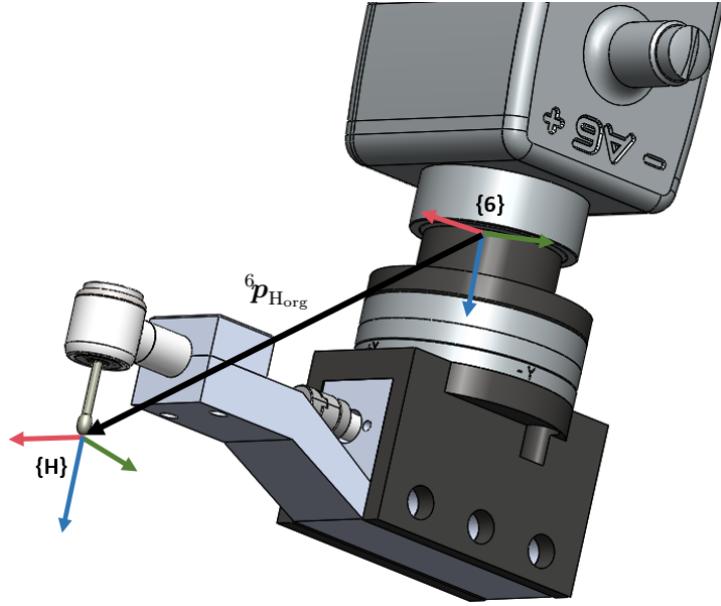


Figure 3.3: Schematic diagram for Tool Center Point. The translation vector ${}^6\mathbf{p}_{\text{H}_{\text{org}}}$ denotes the origin position relative to the frame $\{6\}$.

obtain the tooltip position, which is also the translation vector.

From Fig 3.2, we can obtain the following transformation matrix,

$${}^B_H \mathbf{T} = {}^B_6 \mathbf{T} \cdot {}^6_H \mathbf{T} \quad (3.19)$$

and it can be rewritten as

$$\begin{bmatrix} {}^B_H \mathbf{R} & {}^B \mathbf{p}_{\text{H}_{\text{org}}} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^B_6 \mathbf{R} & {}^B \mathbf{p}_{6_{\text{org}}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^6_H \mathbf{R} & {}^6 \mathbf{p}_{\text{H}_{\text{org}}} \\ 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} {}^B_6 \mathbf{R} \cdot {}^6_H \mathbf{R} & {}^B_6 \mathbf{R} \cdot {}^6 \mathbf{p}_{\text{H}_{\text{org}}} + {}^B \mathbf{p}_{6_{\text{org}}} \\ 0 & 1 \end{bmatrix} \quad (3.20)$$

Consequently, we get a crucial equation:

$${}^B \mathbf{p}_{\text{H}_{\text{org}}} = {}^B_6 \mathbf{R} \cdot {}^6 \mathbf{p}_{\text{H}_{\text{org}}} + {}^B \mathbf{p}_{6_{\text{org}}} \quad (3.21)$$

Now, we move the tooltip to a fixed point with four different poses, including position and orientation. Then, we will get four different rotation matrices and vectors in real-time.

$$\begin{aligned}
 {}^B\mathbf{p}_{H_{org}} &= {}^B\mathbf{R}^1 \cdot {}^6\mathbf{p}_{H_{org}} + {}^B\mathbf{p}_{F_{org}}^1 \\
 &= {}^B\mathbf{R}^2 \cdot {}^6\mathbf{p}_{H_{org}} + {}^B\mathbf{p}_{F_{org}}^2 \\
 &= {}^B\mathbf{R}^3 \cdot {}^6\mathbf{p}_{H_{org}} + {}^B\mathbf{p}_{F_{org}}^3 \\
 &= {}^B\mathbf{R}^4 \cdot {}^6\mathbf{p}_{H_{org}} + {}^B\mathbf{p}_{F_{org}}^4
 \end{aligned} \tag{3.22}$$

In order to extract ${}^6\mathbf{p}_{H_{org}}$ from Eq.3.22, we subtract the second to forth equation from the first equation.

$$\begin{bmatrix} {}^B\mathbf{R}^1 - {}^B\mathbf{R}^2 \\ {}^B\mathbf{R}^1 - {}^B\mathbf{R}^3 \\ {}^B\mathbf{R}^1 - {}^B\mathbf{R}^4 \end{bmatrix} \cdot {}^6\mathbf{p}_{H_{org}} = \begin{bmatrix} {}^B\mathbf{p}_{6_{org}}^2 - {}^B\mathbf{p}_{6_{org}}^1 \\ {}^B\mathbf{p}_{6_{org}}^3 - {}^B\mathbf{p}_{6_{org}}^1 \\ {}^B\mathbf{p}_{6_{org}}^4 - {}^B\mathbf{p}_{6_{org}}^1 \end{bmatrix} \tag{3.23}$$

where we define

$$\mathbf{R} = \begin{bmatrix} {}^B\mathbf{R}^1 - {}^B\mathbf{R}^2 \\ {}^B\mathbf{R}^1 - {}^B\mathbf{R}^3 \\ {}^B\mathbf{R}^1 - {}^B\mathbf{R}^4 \end{bmatrix}_{9 \times 3}, \mathbf{p} = \begin{bmatrix} {}^B\mathbf{p}_{6_{org}}^2 - {}^B\mathbf{p}_{6_{org}}^1 \\ {}^B\mathbf{p}_{6_{org}}^3 - {}^B\mathbf{p}_{6_{org}}^1 \\ {}^B\mathbf{p}_{6_{org}}^4 - {}^B\mathbf{p}_{6_{org}}^1 \end{bmatrix}_{9 \times 1}$$

Therefore,

$$\begin{aligned}
 {}^6\mathbf{p}_{H_{org}} &= \mathbf{R}^\dagger \cdot \mathbf{p} \\
 &= (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \cdot \mathbf{p}
 \end{aligned}$$

As a result, we can utilize the four-points method to obtain the translation vector.

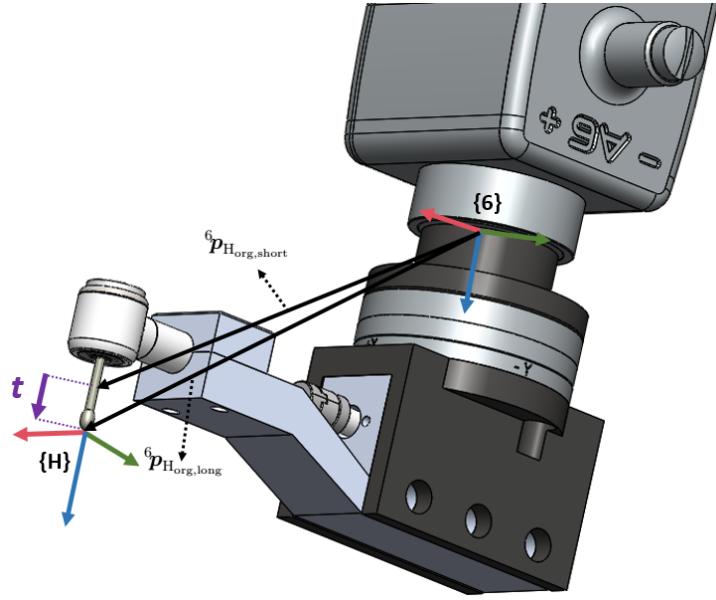


Figure 3.4: Schematic diagram for obtaining the tool vector.

3.4.2 Rotation Analysis

We are turning now to the discussion about the rotation vector. Above all, we have to find the vector of tool insertion direction t . We can obtain the translation vector from the origin of frame $\{6\}$ to the tooltip through the TCP method. Accordingly, we use two root canal files with different lengths and apply the TCP method to separately obtain two vectors illustrated as Fig 3.4. Hence,

$$\mathbf{t} = {}^6\mathbf{p}_{\mathbf{H}_{\text{org}},\text{long}} - {}^6\mathbf{p}_{\mathbf{H}_{\text{org}},\text{short}} \quad (3.24)$$

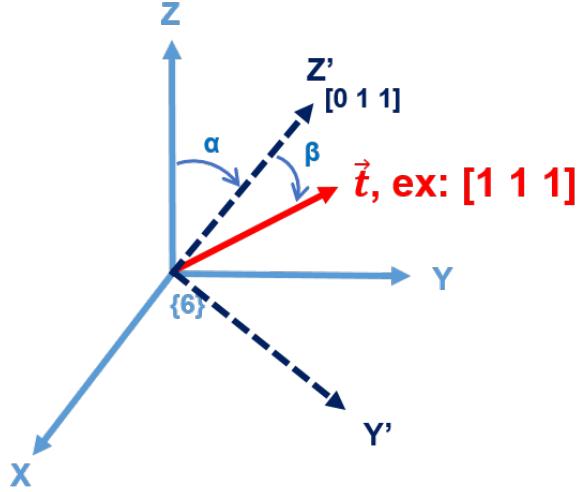


Figure 3.5: Illustration of finding the rotation matrix

For analyzing it easily, we depict it in Fig 3.5. Note that here we only discuss rotation, so we assume that we have done translation and matched the frame $\{S\}$ with frame $\{6\}$. Because we hope to send a Z-axis command to achieve tool insertion, we should align the original Z-axis to the target vector. Nevertheless, Z-axis alignment without other restrictions will produce many solutions. We choose one of the solutions to align Z-axis to the target vector. According to the figure, we assume the target vector t is $[x, y, z]$, whose projection to yz -plane $\text{proj}_{(y-z)}t$ is $[0, y, z]$. Initially, we rotate α degree around X-axis to make original Z-axis align the projection $[0, y, z]$. Next, we rotate β degree around Y' axis and finally align the original Z-axis to the target vector $[1, 1, 1]$. The following equation

$${}^T_6 \mathbf{R} = \mathbf{R}_x(\alpha) \cdot \mathbf{R}_y(\beta) \quad (3.25)$$

where

$$\begin{aligned}\alpha &= -\text{sign}(t_y) \cdot \cos^{-1} \left(\frac{\hat{k} \cdot \text{proj}_{(y-z)} \mathbf{t}}{\|\hat{k}\| \cdot \|\text{proj}_{(y-z)} \mathbf{t}\|} \right) \\ \beta &= \text{sign}(t_x) \cdot \cos^{-1} \left(\frac{\mathbf{t} \cdot \text{proj}_{(y-z)} \mathbf{t}}{\|\mathbf{t}\| \cdot \|\text{proj}_{(y-z)} \mathbf{t}\|} \right)\end{aligned}\quad (3.26)$$

Assume $\mathbf{t} = [x, y, z]$,

$$\begin{aligned}\alpha &= -\text{sign}(y) \cdot \cos^{-1} \left(\frac{z^2}{\sqrt{y^2 + z^2}} \right) \text{ rad} \\ \beta &= \text{sign}(x) \cdot \cos^{-1} \left(\frac{y^2 + z^2}{\sqrt{x^2 + y^2 + z^2} \sqrt{x^2 + y^2 + z^2}} \right) \text{ rad}\end{aligned}\quad (3.27)$$

α and β are Euler angles, which meet the command demand.

In this section, we have demonstrated two key aspects of reference frame changing of the robot arm. It's easy to input the results of section 3.4.1 and section 3.4.2 via the command setTRF, the robot arm will recognize frame{T}. Having discussed how to combine a robot arm with an end effector, the next section addresses ways of combining an F/T sensor.

Chapter 4

Force-Guided Robot Alignment

This chapter follows from the previous issue about integration. We continue to demonstrate some technical solutions for system integration. On top of that, the Forces and Torques (F/T) sensor will be included to discuss. Therefore, you can see the chapter as an operating manual when you simultaneously own a robot arm, an F/T sensor, and an end effector. First of all, we will explain why we need to use the F/T sensor in our project in section 4.1. Furthermore, we will introduce how to compensate the gravity affection while moving the robot in section 4.2. Admittance control based on F/T sensor will be described in section 4.3.1. Coordinate transformation of F/T sensor will be interpreted in section 4.4.1. Last but not least, we will discuss affection of setting admittance control parameters in section 4.5.

4.1 Problem Definition

In modern robot-assisted surgery, to simulate the eyes of doctors, image processing is often considered. With the image's information, the surgeon can obtain real-time data that humans can not observe clearly. Surgeons use the image process to do many things, such as tracking whether the patient is moving, positioning an incision point, and navigating an insertion path. As for root canal treatment, the endodontist checks the number of root canals with computed tomography. Before starting to clean the pulp, the endodontist tries to insert root files into root canals. Then the endodontist takes the patient to have computed tomography(CT). CT scan can clearly show the tooth with the root files. With the preoperative image, the endodontist can check whether all root canals are all found and determine all root canals' lengths. However, the above image application is preoperative. The endodontist can only use the preoperative image to guess the direction of the insertion. The intraoperative image processing is absent due to the dimension restriction of a tooth. It is difficult to get a real-time image when the dentist inserts a root file into the opened tooth because it will obstacle the observation. Besides, the diameter of a root canal is 1 mm and is smaller than a root file. It is impossible to get the image from any shooting angle.

To solve the first problem - without image processing, we propose a method based on force-guided. We surveyed the peg-in-hole based on F/T sensor feedback [12] because the cleaning procedure is similar to it. On the other hand, we provide two operating modes - "Dragging Mode" and "Self-Alignment Mode" to cooperate

with a dentist. In "Dragging Mode", we can hold the handpiece to the desired position and orientation. We hope to use this mode to let the dentist move the handpiece to the injured tooth. In "Self-Alignment Mode", the handpiece can automatically align the root canal direction even though there are many complicated conditions with different teeth.

4.2 Integration of F/T sensor

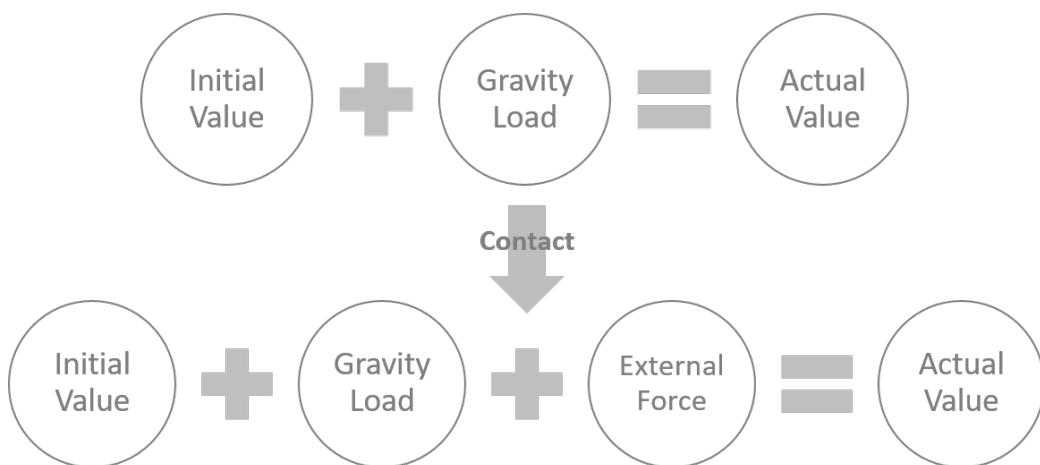


Figure 4.1: Data Analysis of F/T sensor.

Gravity compensation is a critical technical issue when combining an F/T sensor with a robot arm and an end effector [13]. Fundamentally, we should receive stable data when a static F/T sensor bears the same load or force. Nevertheless, our F/T sensor is installed on the robot arm and move with the pose of the robot arm. On account of its mobility, the gravity of the end effector will significantly affect

the actual value we received. Moreover, starting without resetting the F/T sensor to zero would lead to an initial value. If we could not analyze the actual value to an initial value and a gravity load, we would obtain an unstable actual value, not to mention the external force caused by contact force.

Therefore, we illustrate a method, which is to analyze the actual value of the F/T sensor to an initial value and gravity load in real-time. It's worth noting that with this approach, we can get the installation angle between the F/T sensor and the robot arm, including assembly error, and we no longer need to reset the F/T sensor to zero every time.

The Centroid Position of End Effector

To start with the first equation in Figure 4.1,

$$\text{Initial value} + \text{Gravity Load} = \text{Actual Value}$$

$$\Rightarrow \begin{cases} \mathbf{f}_0 + \mathbf{f}_g = \mathbf{f} \\ \boldsymbol{\tau}_0 + \boldsymbol{\tau}_g = \boldsymbol{\tau} \end{cases} \Rightarrow \begin{cases} f_{0x} + f_{gx} = f_x \\ f_{0y} + f_{gz} = f_y \\ f_{0z} + f_{gy} = f_z \\ \tau_{0x} + \tau_{gx} = \tau_x \\ \tau_{0y} + \tau_{gy} = \tau_y \\ \tau_{0z} + \tau_{gz} = \tau_z \end{cases} \quad (4.1)$$

where \mathbf{f} and $\boldsymbol{\tau}$ are force and torque vector respectively.

And, by terms of moment arm formula,

$$\therefore \boldsymbol{\tau}_g = \mathbf{r} \times \mathbf{f}_g \quad (4.2)$$

where \mathbf{r} denotes the centroid position of the end effector in the sensor frame.

$$\begin{aligned} \therefore \boldsymbol{\tau} &= \boldsymbol{\tau}_0 + \boldsymbol{\tau}_g \\ &= \boldsymbol{\tau}_0 + \mathbf{r} \times \mathbf{f}_g \end{aligned} \quad (4.3)$$

Then, Substitute the first line of Equation 4.1 into the above equation, we will obtain

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \mathbf{p} \times (\mathbf{f} - \mathbf{f}_0) \quad (4.4)$$

, which could be extended as

$$\begin{cases} \tau_x = \tau_{0x} + (f_z - f_{0z}) \cdot y - (f_y - f_{0y}) \cdot z \\ \tau_y = \tau_{0y} + (f_x - f_{0x}) \cdot z - (f_z - f_{0z}) \cdot x \\ \tau_z = \tau_{0z} + (f_y - f_{0y}) \cdot x - (f_x - f_{0x}) \cdot y \end{cases} \quad (4.5)$$

and be overwritten as

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 & f_z & -f_y & 1 & 0 & 0 \\ -f_z & 0 & f_x & 0 & 1 & 0 \\ f_y & -f_x & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (4.6)$$

where

$$\begin{cases} k_1 = \tau_{0x} - (f_{0z} \cdot y + f_{0y} \cdot z) \\ k_2 = \tau_{0y} - (f_{0x} \cdot z + f_{0z} \cdot x) \quad \text{are all constant} \\ k_3 = \tau_{0z} - (f_{0y} \cdot x + f_{0x} \cdot y) \end{cases} \quad (4.7)$$

With extracting $[x, y, z, k_1, k_2, k_3]$ in mind, we move the robot arm to n ($n \geq 3$) positions with different poses. By recording n torque vectors and corresponding n force vectors from F/T sensor, we can expand Equation 4.6 as

$$\begin{bmatrix} \tau_x^1 \\ \tau_y^1 \\ \tau_z^1 \\ \vdots \\ \tau_x^n \\ \tau_y^n \\ \tau_z^n \end{bmatrix} = \begin{bmatrix} 0 & f_z^1 & -f_y^1 & 1 & 0 & 0 \\ -f_z^1 & 0 & f_x^1 & 0 & 1 & 0 \\ f_y^1 & -f_x^1 & 0 & 0 & 0 & 1 \\ 0 & f_z^2 & -f_y^2 & 1 & 0 & 0 \\ -f_z^2 & 0 & f_x^2 & 0 & 1 & 0 \\ f_y^2 & -f_x^2 & 0 & 0 & 0 & 1 \\ \vdots & & \vdots & & & \\ 0 & f_z^3 & -f_y^3 & 1 & 0 & 0 \\ -f_z^3 & 0 & f_x^3 & 0 & 1 & 0 \\ f_y^3 & -f_x^3 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (4.8)$$

, which is defined as

$$\mathbf{m}_{(3n \times 1)} = \mathbf{F}_{(3n \times 6)} \cdot \mathbf{p}_{(6 \times 1)} \quad (4.9)$$

As a consequence of the full column rank of \mathbf{F} , we can apply Moore-Penrose pseudoinverse. Then,

$$\begin{aligned} \mathbf{p} &= \mathbf{F}^\dagger \cdot \mathbf{m} \\ &= (\mathbf{F}^\top \mathbf{F})^{-1} \mathbf{F}^\top \cdot \mathbf{m} \end{aligned}$$

From now on, we have already known the centroid position of end effector in the sensor frame and values of the constants k_1, k_2, k_3 .

Gravity Compensation and Initial Value Reset

Next, let us return to the first equation in Figure 4.1. We continue to use this formula and contemplate it from the perspective of coordinate transformation relation. Here we hypothesize that the end effector weighs g kilograms relative to the frame $\{0\}$ which is also the world frame. That means the gravity vector of the end effector ${}^0\mathbf{g}$ is $[0, 0, -g]$ in the frame $\{0\}$. Then, we can derive it as following.

$$\begin{aligned} \text{Initial value + Gravity Load} &= \text{Actual Value} \\ \Rightarrow \mathbf{f}_0 + {}^S_6\mathbf{R} \cdot {}^0_6\mathbf{R} \cdot {}^0\mathbf{g} &= \mathbf{f} \end{aligned} \quad (4.10)$$

which is relative to frame $\{6\}$. Hence, we assume an installation angle θ including assembly error.

$${}^S_6\mathbf{R} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

Besides, we can easily calculate ${}^6_0\mathbf{R}$ from Equation 3.3.2.

$$\begin{aligned} {}^6_0\mathbf{R} &= {}^0_6\mathbf{R}^\top \\ &:= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \end{aligned} \quad (4.12)$$

Therefore,

$$\begin{bmatrix} f_{0x} \\ f_{0y} \\ f_{0z} \end{bmatrix} + \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (4.13)$$

Further, we rewrite it as

$$\begin{bmatrix} -r_{13} & -r_{23} & 0 & 1 & 0 & 0 \\ -r_{23} & r_{13} & 0 & 0 & 1 & 0 \\ 0 & 0 & -r_{33} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g \cos(\theta) \\ g \sin(\theta) \\ g \\ f_{0x} \\ f_{0y} \\ f_{0z} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (4.14)$$

In the same way as Equation 4.6, we can extract $[g \cos(\theta), g \sin(\theta), g, f_{0x}, f_{0y}, f_{0z}]$ by the least square solution. Note that, we can simultaneously record these data when using the method in Equation 4.6. By recording $n(n \geq 3)$ third columns of rotation matrices and corresponding n force vectors from F/T sensor, we can expand Equation 4.14 as

$$\begin{bmatrix} -r_{13}^1 & -r_{23}^1 & 0 & 1 & 0 & 0 \\ -r_{23}^1 & r_{13}^1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -r_{33}^1 & 0 & 0 & 1 \\ -r_{13}^2 & -r_{23}^2 & 0 & 1 & 0 & 0 \\ -r_{23}^2 & r_{13}^2 & 0 & 0 & 1 & 0 \\ 0 & 0 & -r_{33}^2 & 0 & 0 & 1 \\ \vdots & & & & & \\ -r_{13}^n & -r_{23}^n & 0 & 1 & 0 & 0 \\ -r_{23}^n & r_{13}^n & 0 & 0 & 1 & 0 \\ 0 & 0 & -r_{33}^n & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g \cos(\theta) \\ g \sin(\theta) \\ g \\ f_{0x} \\ f_{0y} \\ f_{0z} \end{bmatrix} = \begin{bmatrix} f_{0x}^1 \\ f_{0y}^1 \\ f_{0z}^1 \\ f_{0x}^2 \\ f_{0y}^2 \\ f_{0z}^2 \\ \vdots \\ f_{0x}^n \\ f_{0y}^n \\ f_{0z}^n \end{bmatrix} \quad (4.15)$$

, which is defined as

$$\mathbf{M}_{(3n \times 6)} \cdot \mathbf{p}_{(6 \times 1)} = \mathbf{f}_{(3n \times 1)} \quad (4.16)$$

As a consequence of the full column rank of \mathbf{M} , we can apply Moore-Penrose pseudoinverse. Then,

$$\begin{aligned} \mathbf{p} &= \mathbf{M}^\dagger \cdot \mathbf{f} \\ &= (\mathbf{M}^\top \mathbf{M})^{-1} \mathbf{M}^\top \cdot \mathbf{f} \end{aligned}$$

Apparently, we can directly obtain $g, f_{0x}, f_{0y}, f_{0z}$. Afterwards, we substitute them into Equation 4.7 to calculate

$$\begin{cases} \tau_{0x} = k_1 + (f_{0z} \cdot y + f_{0y} \cdot z) \\ \tau_{0y} = k_2 + (f_{0x} \cdot z + f_{0z} \cdot x) \\ \tau_{0z} = k_3 + (f_{0y} \cdot x + f_{0x} \cdot y) \end{cases} \quad (4.17)$$

Finally, we successfully obtain the weight of the end effector g , the initial value of the F/T sensor $[f_{0x}, f_{0y}, f_{0z}, \tau_{0x}, \tau_{0y}, \tau_{0z}]$.

As for installation angle θ including assembly error, there is an further discussion. Undoubtedly, we can derive it as

$$\theta = \cos^{-1} \left(\frac{g \cos(\theta)}{g} \right) \text{ or } \sin^{-1} \left(\frac{g \sin(\theta)}{g} \right) \quad (4.18)$$

In theory, By either the arccos function or arcsin function, we can derive the same value. However, we estimate $[g \cos(\theta), g \sin(\theta), g, f_{0x}, f_{0y}, f_{0z}]$ by using the least square solution which produce a approximated answer rather than the correct answer. A subtle bias caused by the least square solution will be enlarged through

the arc function. Thankfully, we originally designed an adapter to connect the robot arm and F/T sensor, and the installation angle is exactly zero degrees. Therefore, here we only need to concern about the subtle bias result from assembly errors.

Here we take the same bias and compare arccos to arcsin. Obviously, if we

Table 4.1: Arc-function Comparison

bias n (rad)	$\sin^{-1}(n)$ (degree)	$\cos^{-1}(1 - n)$ (degree)
0	0	0
0.001	0.057	2.56
0.01	0.57	8.11
0.1	5.7	25.84

separately give the same bias (0.1 rad) into the arccos and arcsin function, the arccos function will enlarge the bias significantly larger than the arcsin function.

Hence we should use

$$\theta = \sin^{-1} \left(\frac{g \sin(\theta)}{g} \right) \quad (4.19)$$

Notably, as a result of the zero installation angle we initially designed from Table 4.1, we could infer using the arcsin function. In contrast, if the installation angle is not zero, the above assumption will be invalid.

Ultimately, we have successfully dissected the actual value of the F/T sensor into the initial value of the F/T sensor and the gravity of the end effector.

External Force Evaluation

Next, we continue to review the second equation in Figure 4.1.

$$\text{Initial value} + \text{Gravity Load} + \text{External Force} = \text{Actual Value}$$

$$\Rightarrow \begin{cases} f_0 + f_g + f_e = f \\ \tau_0 + \tau_g + \tau_e = \tau \end{cases} \quad (4.20)$$

Therefore,

$$\begin{aligned} f_e &= f - f_0 - f_g \\ &= f - f_0 - {}^S_6 \mathbf{R} \cdot {}^6_0 \mathbf{R} \cdot {}^0 g \end{aligned} \quad (4.21)$$

$$\tau_e = \tau - \tau_0 - \tau_g$$

$$= \tau - \tau_0 - \mathbf{r} \times f_g$$

$$\Rightarrow \begin{cases} f_{ex} = f_x - f_{0x} + g \cos(\theta) r_{13} + g \sin(\theta) r_{23} \\ f_{ey} = f_y - f_{0y} - g \sin(\theta) r_{13} + g \cos(\theta) r_{23} \\ f_{ez} = f_z - f_{0z} + g r_{33} \\ \tau_{ex} = f_x - \tau_{0x} - (g_z \cdot y - g_y \cdot z) \\ \tau_{ey} = f_y - \tau_{0y} - (g_x \cdot z - g_z \cdot x) \\ \tau_{ez} = f_z - \tau_{0z} - (g_y \cdot x - g_x \cdot y) \end{cases}$$

4.3 Dragging Mode

In the hope that dentist could drag our system to an infected tooth by holding the end effector, we usher in admittance control based on F/T sensor.

4.3.1 Admittance Control based on F/T sensor

Control Scheme

Admittance control makes the robot move like a spring-mass-damper system. Forces and torques can be mapped into the movements such as position or velocity. Most importantly, admittance control enables a robot arm to cooperate with humans in a safe work environment. Since Meca500 is an industrial robot arm without admittance control, we subsequently combine the robot arm with the F/T sensor to adopt admittance control. With force and torque feedback, the F/T sensor makes Meca500 be like a collaborative robot arm. Therefore, we propose a control scheme depicted as Figure 4.2. It's worth noting that in this approach, the admittance control function is triggered by the end effector mounted on the F/T sensor instead of detecting each wrist torque of the robot arm.

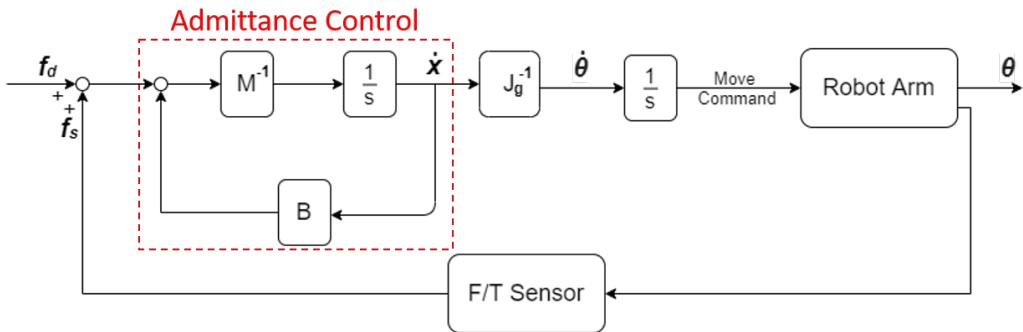


Figure 4.2: Control scheme. f_d denotes the desired forces and torques vector.

f_s denotes the real value detected by F/T sensor and is also a forces and torques vector. \dot{x} denotes $[\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z]$. J_g denotes the geometric Jacobian matrix. \dot{q} denotes $[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6]$. q denotes $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$.

A standard equation of admittance control is represented as Equation 4.22.

The values we obtain from the F/T sensor are $[f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]$, whose forces $[f_x, f_y, f_z]$ are related to the translations $[x, y, z]$ and torques $[\tau_x, \tau_y, \tau_z]$ are related to the axis angle $[\theta_x, \theta_y, \theta_z]$.

$$\begin{bmatrix} x \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \frac{1}{\mathbf{M}s^2 + \mathbf{B}s + \mathbf{K}} \begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (4.22)$$

Selection of Admittance Control Model

In our proposed approach, we omit parameter \mathbf{K} which is relevant to spring stiffness, considering that it's not necessary to bounce such as a spring. Therefore, our system should behave like a mass-damper system as following.

$$\begin{bmatrix} x \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \frac{1}{\mathbf{M}s^2 + \mathbf{B}s} \begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (4.23)$$

where $\mathbf{M}, \mathbf{B}, \mathbf{K}$ are diagonal positive definite matrices. Affections of these parameters will be discussed in section 4.5.

Robot Command Decision

After determining our admittance control model, we should select a corresponding command to move the robot. However, there are many commands of moving the robot arm, as shown in Table 4.2.

Table 4.2: Moving commands in Meca500

Types	Commands	Input parameters
Position	MoveJoints	$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$
	MovePose	$x, y, z, \alpha, \beta, \gamma$
Velocity	MoveJointsVel	$\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6$
	MoveLinVelTRF	$\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z$

Considering that the singularity problem is an imperative subject in robotics, we intend to use MoveJoints or MoveJointsVel to set the axes' angles directly. Despite that it's easier to implement admittance control via other commands, the system would touch the singularity point at any time. It undoubtedly exposes patients to danger because of the uncertainty of the system. In our case, we could detect whether it is a singularity or not before commanding the robot. Therefore, position command - MoveJoints and velocity command - MoveJointsVel is our option. Why we choose the command MoveJoints is that Meca500 has a default time-out value to ensure its safety. Even though we could set this value from 0.001 to 2 seconds, Meca500 is still restricted to move with this value. For example, the value of time-out is set 0.1 sec. While the robot receives a command, the

robot move for 0.1 sec then immediately stops. It's not easy to control via velocity command due to this default property.

Finally, we designate $\text{MoveJointsVel}(\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_6)$ as our main command. Thanks to the property of Jacobian matrix shown in Equation 3.13, we can transform $\dot{\mathbf{x}}$ into $\dot{\mathbf{q}}$. Then we multiply it an integrator $\frac{1}{S}$ to obtain \mathbf{q} . Note that, As F/T sensor is mounted on the frame{6}, we should use Equation 3.13 rather than Equation 3.6.

Implementation in Discrete Time

To implement it in practice, we use the bilinear transformation, also known as Tustin method, to control it in discrete time.

$$\begin{aligned} \frac{\mathbf{v}(s)}{\mathbf{f}(s)} &= (\mathbf{M}s + \mathbf{B})^{-1} \\ \Rightarrow \frac{\mathbf{v}(z)}{\mathbf{f}(z)} &= \frac{T(1 + z^{-1})}{(T \cdot \mathbf{B} + 2\mathbf{M}) + (TB - 2M) \cdot z^{-1}} \end{aligned} \quad (4.24)$$

and,

$$\begin{aligned} \frac{\boldsymbol{\theta}(s)}{\mathbf{w}(s)} &= \frac{\mathbf{K}}{s} \\ \Rightarrow \frac{\boldsymbol{\theta}(z)}{\mathbf{w}(z)} &= \frac{TK}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \end{aligned} \quad (4.25)$$

Therefore,

$$\begin{aligned} \mathbf{v}[k+1] &= \frac{T}{TB + 2M} \cdot (\mathbf{f}[k+1] + \mathbf{f}[k]) - \frac{TB - 2M}{TB + 2M} \cdot \mathbf{v}[k] \\ \boldsymbol{\theta}[k+1] &= \boldsymbol{\theta}[k] + \frac{TK}{2} (\mathbf{w}[k+1] + \mathbf{w}[k]) \end{aligned} \quad (4.26)$$

where T is the sampling time.

4.4 Self-Alignment Mode

This section aims to develop a framework for robot self-alignment regarding the position and orientation of the root canal, which is one of the main contributions of the thesis. This function assists endodontists in operating the cleaning procedure automatically and ensures a postoperative recovery for patients. The main idea is to amend the direction of insertion by itself to lower the contact resistance.

We propose a complete robotic procedure. First and foremost, the prominent role of this procedure is the root canal reamer. We rotate the root canal reamer to clean a pulp, move it to do reciprocation and self-alignment. So, we care about the root canal, such as its position, orientation, rotation speed, and contact force. Nevertheless, the robot arm and the F/T sensor have their coordinates initially. They originally recognize its own coordinates rather than the root canal reamer's frame, so they do their work respectively. Therefore, we need to let them understand the tool frame. Thanks to section 3.4, we have demonstrated how to change the reference frame of the robot arm. After that, we can identify the translation and rotation information of the root canal reamer and subsequently obtain the transformation matrix from the robot arm to the tool. As for the F/T sensor, we have explained how to do gravity compensation in section 4.2, we will interpret how to change its fiducial to the tooltip in section 4.4.1. Besides, we will advance motion planning in accordance with the dentist's motion and standard surgical solution in section 4.4.2

4.4.1 Coordinate Transformation of F/T sensor

Similar to Coordinate transformation of F/T sensor in section 3.4. The F/T sensor originally recognizes its coordinate rather than the root canal reamer's frame. Namely, the F/T sensor receives unmatched data when the tool contacts an obstacle due to the wrong reference frame. Hence, here we illustrate how to obtain the coordinate transformation of the F/T sensor to decouple the force and torque from frame{S} to the frame{T}. We analyze it from two perspectives - reference frame changing and measurement point changing.

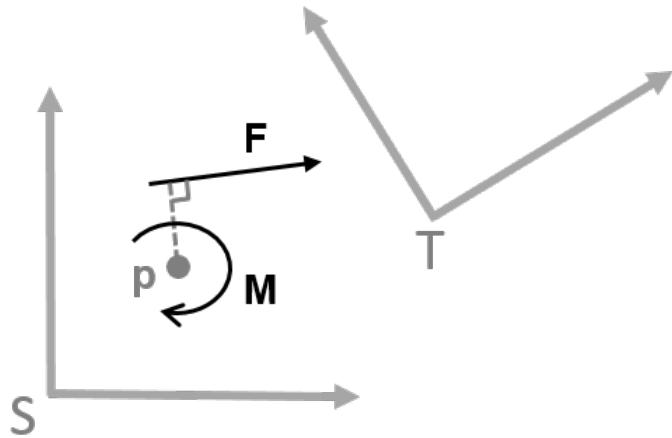


Figure 4.3: Illustration of changing reference frame from {S} to {T}. f and m respectively denote a force and a moment measured at point p.

In view of reference frame changing in Figure 4.3, we can derive the following

equation.

$$\begin{bmatrix} \mathbf{f}_p \\ \mathbf{m}_p \end{bmatrix}_{\{T\}} = \begin{bmatrix} {}^T_S \mathbf{R} & \mathbf{0} \\ \mathbf{0} & {}^T_S \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{f}_p \\ \mathbf{m}_p \end{bmatrix}_{\{S\}} \quad (4.27)$$

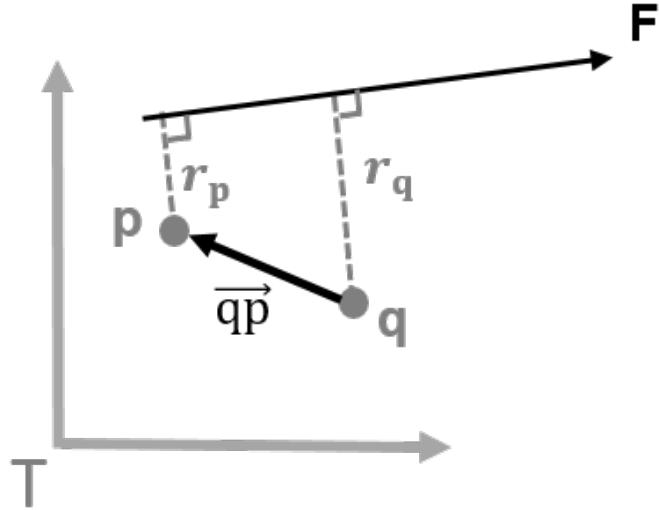


Figure 4.4: Illustration of changing reference frame from $\{S\}$ to $\{T\}$. \mathbf{f} and \mathbf{m} respectively denote a force and a moment measured at point p.

On the other hand, in Figure 4.4 we describe a force and a torque observed in the same frame but at a different point.

$$\begin{aligned} \mathbf{f} &= \mathbf{f}_p = \mathbf{f}_q \\ \mathbf{m}_q &= \mathbf{r}_q \cdot \mathbf{f} \\ &= \mathbf{r}_p \cdot \mathbf{f} + \overrightarrow{qp} \times \mathbf{f}_p \\ &= \mathbf{m}_p + \overrightarrow{qp} \times \mathbf{f}_p \end{aligned} \quad (4.28)$$

Assume

$$\vec{qp} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

then

$$\vec{qp} \times \mathbf{f}_p = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad (4.29)$$

As a result, we consider reference frame changing and measurement point changing,

and then we can obtain the following essential equation.

$$\begin{bmatrix} \mathbf{f}_q \\ \mathbf{m}_q \end{bmatrix}_{\{T\}} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0} \\ 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 & \mathbf{I}_{3 \times 3} \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} {}^T_S \mathbf{R} & \mathbf{0} \\ \mathbf{0} & {}^T_S \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{f}_p \\ \mathbf{m}_p \end{bmatrix}_{\{S\}} \quad (4.30)$$

4.4.2 Motion Planning: Based on Admittance Control

Here we plan a series of motions to simulate the cleaning procedure of the root canal treatment. Endodontist uses their professional experience to determine how many force to apply and when to reverse the file to release the torque caused by the contact force. Because the root canal resembles a cone. The profundity of the file is larger, then the resultant force composed of contact forces is more significant. Namely, the endodontist will apply more and more pressure in the insertion direction during the surgery. To simulate this series of motions, we could

directly command the robot arm to move with the series of actions. However, commanding the robot arm is a dilemma because we have already use admittance control which also commands the robot arm to move. Therefore, to solve the conflict, we continue to utilize admittance control and propose a method based on admittance control to simulate the series of motions. Because we know the tool frame from section 3.4, we can regulate F_{des} to make the robot move. That means if we want to insert along with the tool direction, we could set [00F_z000]. While F_z is larger, the robot moves faster.

4.5 Affections of Parameters Setting

Because our system is similar to a mass-damper system, we focus on the performance of the velocity \dot{x} whereby the system could be considered as

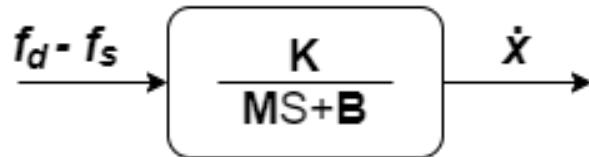


Figure 4.5: Control scheme. M, B, K are all diagonal positive definite matrices, whose diagonal indexes are respectively related to $x, y, z, \theta_x, \theta_y, \theta_z$. M, B are related to the inertial and damping respectively, and K is a proportional gain.

It is a first-order control system and its step response is

$$\begin{aligned}\mathcal{L}^{-1} \left[\frac{1}{S} \cdot \frac{K}{MS + B} \right] &= \mathcal{L}^{-1} \left[\frac{K}{B} \left(\frac{1}{S} - \frac{1}{S + \frac{B}{M}} \right) \right] \\ &= \frac{K}{B} \left(1 - e^{-\frac{B}{M}t} \right)\end{aligned}\quad (4.31)$$

From the above derivation, we can know

$$\begin{aligned}\text{time constant } \tau &= \frac{M}{B} \\ \text{gain} &= \frac{K}{B}\end{aligned}\quad (4.32)$$

Hence, the transient response can be derived as

$$\begin{aligned}\text{Rise time} &= 2.3\tau = \frac{2.3M}{B} \\ \text{Settling time} &= 4\tau = \frac{4M}{B}\end{aligned}\quad (4.33)$$

In theory, when τ is larger, the pole is farther from the origin in S domain, the system is more stable. Besides, when τ is larger, the response speed is faster. Furthermore, when the gain $\frac{K}{B}$ is larger, the robot arm moves more. Therefore, we coarse-tune parameter K to adjust the whole gain of the system and determine the mode of the system is "Dragging Mode" or "Self-Alignment Mode". Last but not least, we fine-tune diagonal parameters of B separately because the inertial and spring properties of each axis are discrepant.

In practice, we implement it in discrete time. From Equation 4.24, we can further derive the final value in discrete time shown as

$$\text{gain} = \lim_{z \rightarrow 1} (z - 1) \cdot \frac{TK(z + 1)}{(TB + 2M)z + TB - 2M} \cdot \frac{1}{\frac{2}{T} \frac{z-1}{z+1}} = \frac{KT}{B} \quad (4.34)$$

Compared to the gain in continuous time, the sampling time in discrete time also effect the gain. Because the sampling time we set is 0.002s, we set $(m_i : b_i = 1 :$

500) to make ($\tau_i = \frac{m_i}{b_i} = 0.002$), then the system will rise up to 98% after 0.004 second. Further, we can alter \mathbf{K} to easily regulate the gain. In Table 4.3, we show detailed parameters with "Dragging Mode" and "Self-Alignment Mode".

Table 4.3: Parameters setting of Admittance control.

Parameter	Dragging Mode	Self-Alignment Mode
k	60	10
b_1	80	80
b_2	80	80
b_3	80	80
b_4	160	80
b_5	160	80
b_6	160	80

$$\mathbf{K} = \text{diag}(k, k, k, k, k, k), \quad \mathbf{B} = \text{diag}(b_1, b_2, b_3, b_4, b_5, b_6)$$

Note that we change the different sensor frames with "Dragging Mode" and "Self-Alignment Mode". As shown in Fig 4.6, Point A and B are two origins of the sensor frame. In "Dragging Mode", we transform the sensor frame to the point A, and the operator can hold the point G_1 and G_2 to move the entire handpiece. On the other hand, in "Self-Alignment Mode" we set the point B as the new sensor frame because we focus on forces and torques the file bears. Due to different level arms, the forces we apply in each axis will be different. There are same level arms (\overline{AO} and \overline{OG}_2 both are 13.2 mm)if we set the point A as the sensor frame

and grasp G_1 and G_2 to move the entire handpiece. Therefore, to provide users a good experience, we should regulate parameters of B in "Dragging mode" with suitable values. With these values, the users can smoothly move the robot arm to the desired position and orientation. We set the first three variables relative to the linear velocity as the same values. Besides, We set the last three variables relative to the angular velocity as the other same values.

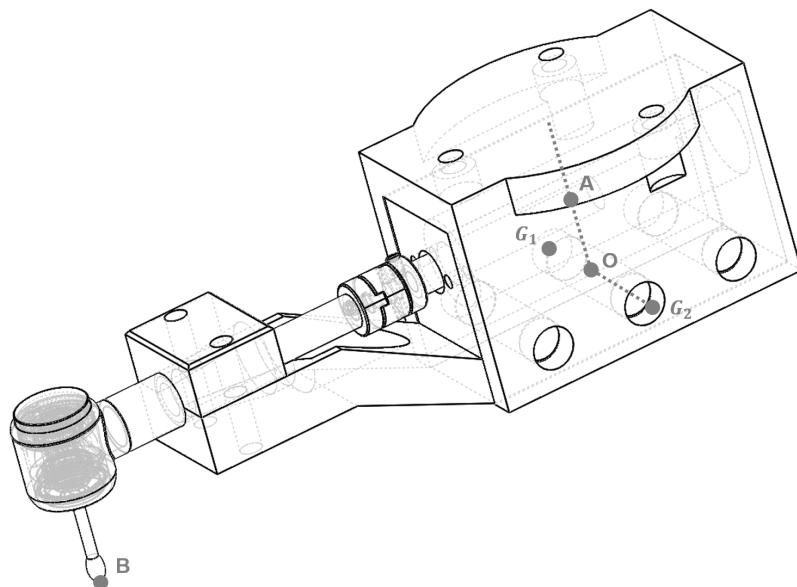


Figure 4.6: Holding gesture in "Dragging Mode"

Chapter 5

File Feedrate Control

5.1 Problem Definition

Instrument fracture is a potential crisis for endodontic treatment. If a dentist improperly operates a endodontic file, it will increase the possibility of instrument fracture. Most largely, the instrument fracture will happen while a file is inserted into the one-third place far from the root apex. It is barely possible to take out in a small root canal. Removal of broken files is technically tricky, so it is critical to reduce the probability of the instrument fracture. Therefore, to protect the endodontic file from fracturing, we need to be familiar with its physical property and its correct operations.

We delve into the physical properties of an endodontic file. A file is made of alloys of Nickel and Titanium, which is a superelastic material. The Ni-Ti file has significantly more elastic and incorruptible property. This feature allows it to bend

when inserted into a curve root canal and thereby reduce the possibility of stuck.

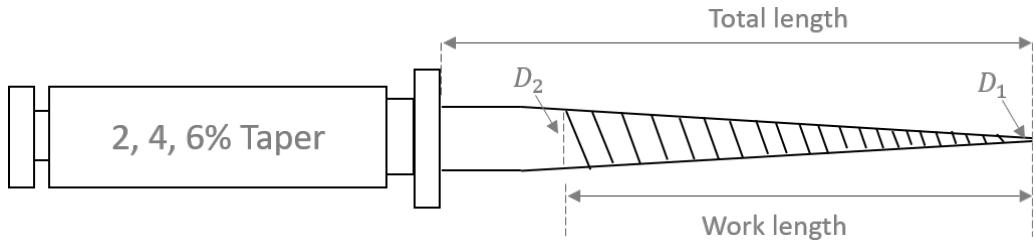


Figure 5.1: The illustration of an endodontic file

In Fig 5.1, an endodontic file is illustrated. There are four types of total length - 21, 25, 28, and 31 mm. They all have the same work length - 16 mm. Each file has their own number such as #15, #20, #25, … , #40, which represent the diameter of files. Take a #15 file with 2% taper for example, the diameter of the tooltip D_1 is

$$15/100 = 0.15 \text{ (mm)}$$

and the diameter at the end of the work length D_2 is

$$0.15 + 16 \cdot 6\% = 1.11 \text{ (mm)}$$

There is a technical solution to detect a status of a file. When a file suffers a force, it will lead to the crystalline phase transformation. However, it can only be observed from a microscope. It is impracticable to usually monitor it during the surgery. A previous literature shows there are two main causes of fractured files are torsional fracture and flexural fatigue, account for 55.7% and 44.3% separately[6]. Therefore, we can reduce the torque a file bears to prevent the instrument fracture.

5.2 The Proposed Method and Theorem

In order to protect an endodontic file from fracturing, we need to reduce the file torque. The easiest and most effective way to solve this issue would be monitoring the torque which the file is enduring. Therefore, we implement a torque monitoring system on DentiBot. We utilize current feedback to keep track of the torque. A increase in the current of the motor is indicative of contact resistance of the file. Also, We propose two approaches both based on feedrate control to release the file torque. One is inverse rotation control discussed in section 5.2.1 and the other is feed control discussed in section 5.2.2, they are both efficient way to release the file torque. Once the current of the file is in excess of a threshold, it will inversely rotate or decline the feedrate to release torque.

Theorem

A motor can be modeled as an RL circuit. In this way, we can get the following formula

$$V = R \cdot I + L \cdot \frac{dI}{dt} + \varepsilon \quad (5.1)$$

where V represents the input voltage, I represents the input current, and ε represents the back EMF produced by the motor motion.

Since the value of inductance of DC motor is small, we ignore the inductance term L . After transposition, we can get the following relation between current and

the back EMF.

$$I = \frac{I - \varepsilon}{R} \quad (5.2)$$

The back EMF produced by the motor is dependent upon the motor constant K , shown as the following equation.

$$\varepsilon = K \cdot \frac{d\theta(t)}{dt} = K \cdot \omega(t) \quad (5.3)$$

When the file encounters resistance during the drilling procedure, the rotating speed of the motor would slow down. With the Equation 5.2 and 5.3 we can infer that the back EMF would decrease, and the current would increase. Also, we have the relation between the current and torque

$$T_m = K_m \cdot I \quad (5.4)$$

where k_m represent the torque constant of a motor.

The maximum torque a root canal file can bear is 6.20 Ncm (62.0 mNm) [14]. Therefore, we can derive the following inequality.

$$T_m \cdot GR < 62 \text{ (mNm)} \quad (5.5)$$

where GR is gear ratio of gearbox mounted on a motor.

Hence, we can set a current threshold based on the following condition.

$$I_{max} < \frac{62}{GR \cdot K_m} \quad (5.6)$$

In reality, GR is 67 and K_m is 16mNm/A

5.2.1 Feedrate Control - Inverse Rotation Control

Once the current of the file exceed the specific threshold, it will inversely rotate for 90 degrees to release torque and then continue drilling in the original direction. Before designing the DentiBot, we have validated this function with the first prototype of DentiBot shown as Fig 5.2.

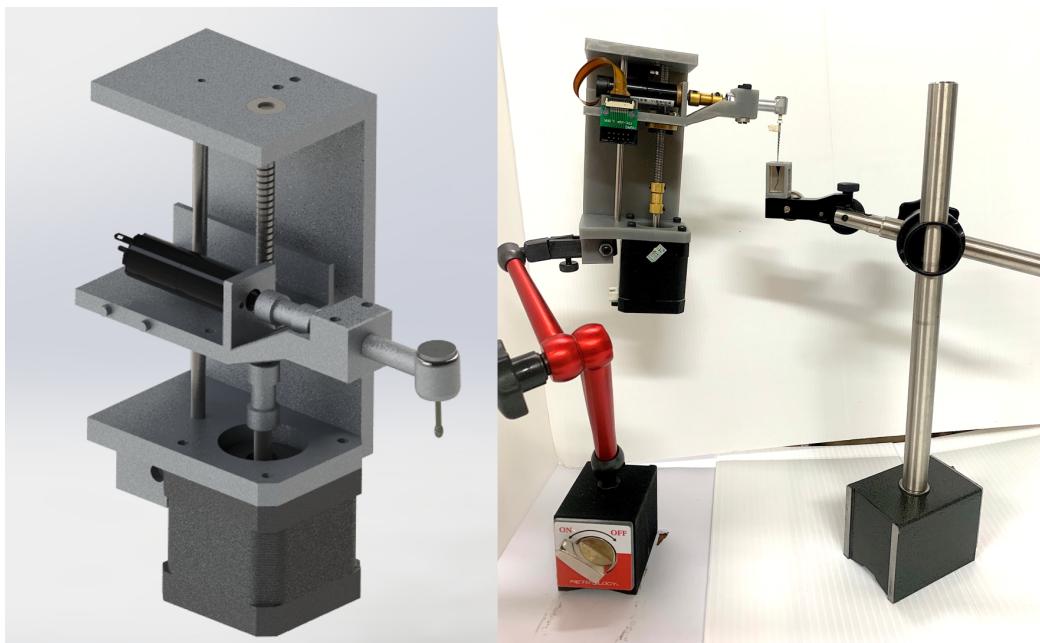


Figure 5.2: The first prototype of DentiBot

It is a drilling system mounted on a magnetic stand. The modified handpiece, a hand-held dental electric device, with a single axis mechanism can validate the inverse rotation control. We designed a motion planning to perform the drilling procedure. The root canal is divided into several sections. In Fig 5.3, we propose a motion planning to validate the function. In the beginning, the robot move to the start point and then repetitively drill one section back and forth at least ten times

until the current feedback decrease under the specific threshold. If the current exceed the threshold, the endodontic file will inversely rotate for 90° to release the torque and then drill the same section again. After one section is cleaned thoroughly, the drilling file will move to the next section. Every section will be cleaned for at least 10 rounds. If there is reverse rotation, it will be more than 10 rounds. When to be finished is contingent on whether the endodontic file reaches the final section.

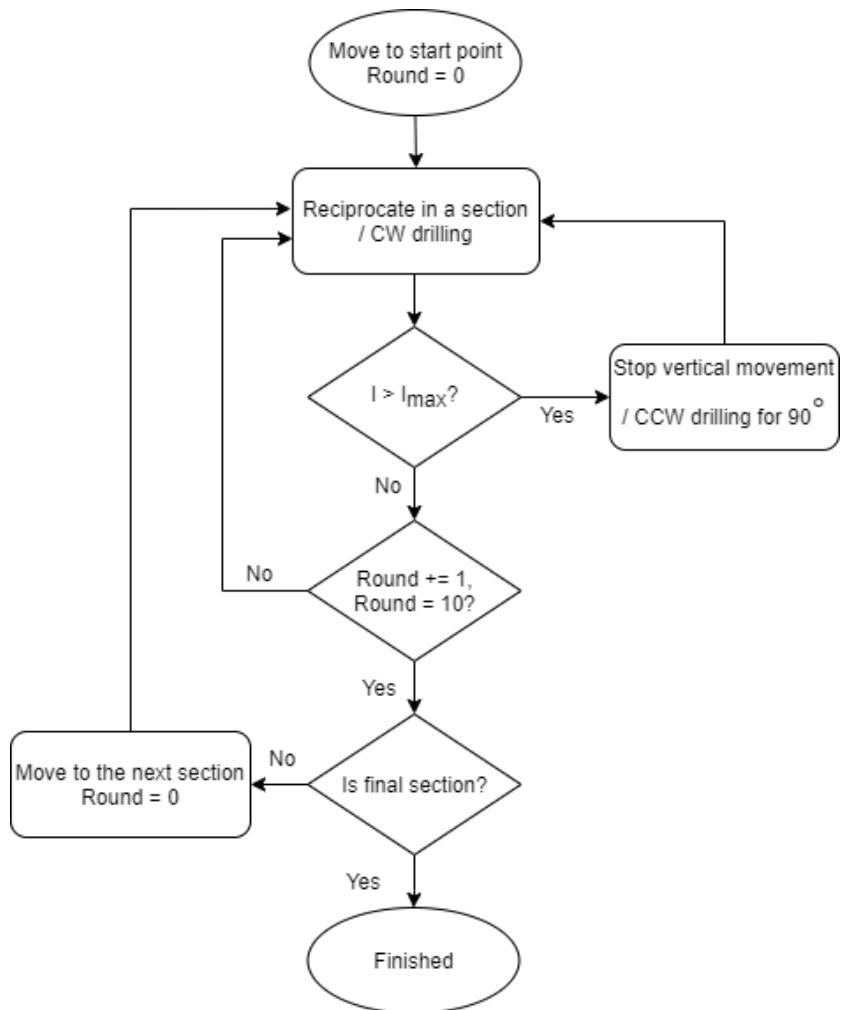


Figure 5.3: Motion planning of Inverse Rotation Control.

5.2.2 Feedrate Control - Feed Control

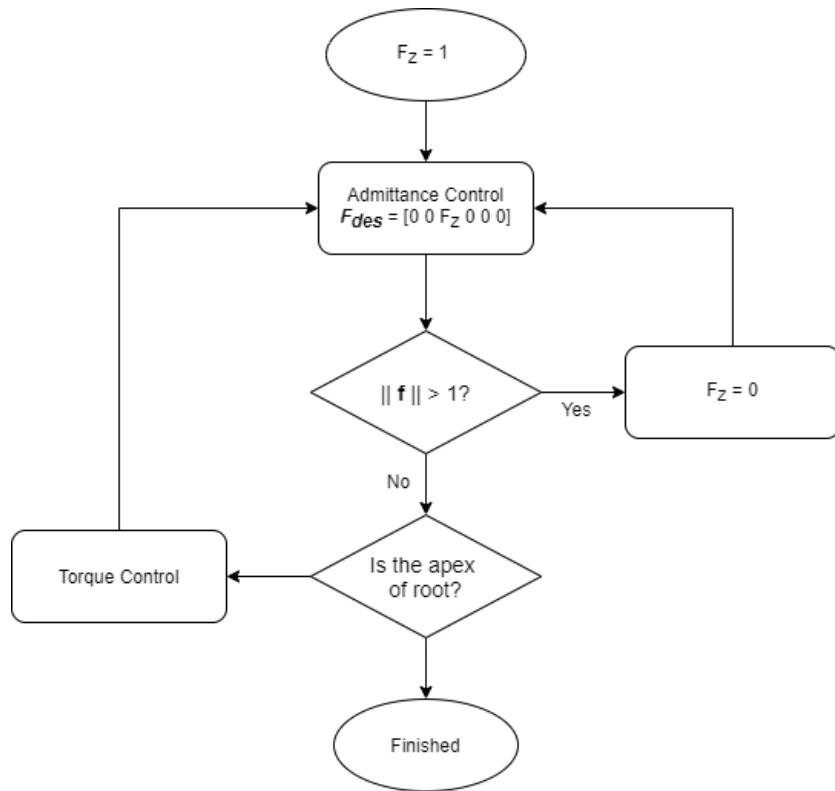


Figure 5.4: Flow chart of Feed Control

Chapter 6

Preliminary Experiment Result

6.1 Experimental Setup

Use present tense.

(Communication protocol – EtherCAT, RTOS – NI target)

For 6.2 experiment: (Stewart-Platform + PhaseSpace + markers)

For 6.3 6.4 experiments: (Acrylic root canal model + truth tooth)

6.2 Admittance Control

In order to prove the validation of admittance control, we set up this experiment.

Here we built a Stewart platform, which had six degrees of freedom and provided a slight movement. We used a Stewart platform to simulate a motion of a patient.

Basically, when the patient moves to a position, our system should move to the same place. Therefore, we planned to observe the target's and the handpiece's position

to validate that our system can track the patient. Besides, we use PhaseSpace to obtain their motion in real-time. PhaseSpace is a motion capture device whose resolution is around 1 mm. Before starting this experiment, we should fix a relative position between the file and the acrylic tooth. We made the file rotate and get stuck in the root canal of an acrylic model. Then we used "Doctor dragging" mode to install the acrylic model on the Stewart platform. Therefore, we can guarantee that the relative position between the file and the acrylic tooth.

We moved the Stewart platform in horizontal and vertical directions separately. The motion planning in the horizontal direction is a square which is 10×10 mm and in the vertical direction is a linear motion from 0 to 40 mm.

6.3 Automatically Direction Changing

Validation of Self-alignment Mode

(Metrics: time, completeness and file breakage)

(Completeness definition: comparison of pixel area before and after experiment via image)

6.4 Repetitive Experiment

validation of repetitive experiment

(Metrics: file breakage, compare with and without reverse)

Chapter 7

Conclusions and Future works

An endodontic treatment is a challenging surgery for dentists due to complex conditions of teeth. Therefore, building a robot-assisted system for the endodontic robot requires comprehensive consideration on whole surgery procedures. Review the section 1.4, we advocated the four parts of the project prospect. We highlight two main parts of the prospect and have proved the feasibility of our proposed approaches. The whole work of the thesis is concluded in this chapter.

7.1 Conclusions

Nowadays, despite that there are more and more dental robot springing up, there are still few teams specific to the endodontic treatment. In view of this, our team have built the robot-assisted system, Dentibot, composed of a 6-DoF robot arm, a 6-DoF F/T sensor, and a modified handpiece. Integration issues between the above devices are reviewed and solved. The DentiBot can assist dentists performing the

endodontic treatment as a result of the following functions.

Admittance control is ushered in to enable the dentist to move the DentiBot above an infected tooth. Also, a framework based on admittance control for robot alignment regarding the position and orientation of root canal is presented. "Dragging Mode" and "Self-Alignment Mode" are separately implemented for the above functions.

Last but not least, instrument fracture is a serious problem for dentists. It will even lead to medical dispute. With the torque monitoring system, file feedrate control is applied to reduce the possibility of instrument fracture .

Experiments indicated that the above functions had good performances on force-guided alignment and file feedrate control. It has proven the feasibility in technical and clinical perspectives. Undoubtedly, the DentiBot can help dentists perform better clinical results.

7.2 Discussion and Future Works

The thesis is the pioneer of the endodontic project. Despite that the thesis develop the DentiBot and present the above functions, there are undoubtedly some works remained and rooms for improvement on the modified hardware and functions. The modified handpiece made by 3D-print is not durable for time consuming endodontic treatment. It is necessary to do machining for more stable results.

In the thesis, we assume that a dentist move the DentiBot above the root, then

the DentiBot do "Cleaning" procedure. However, sometimes there is not only one root in the tooth. For instance, There are three to four roots in molars. Therefore, we wish that the DentiBot will have the ability to search all root canals after the dentist move it above the infected tooth.

On top of that, we proposed the alignment method while the DentiBot do drilling. It not only align the root path but also patient tracking. However, the patient tracking belongs to a minor range movement. Therefore, the DentiBot is expected to be applied to patient tracking with large movement via string potentiometers in the future.

Chapter 8

Appendix

8.1 Forward Kinematics

$${}^0_6 \mathbf{T} = {}^0_1 \mathbf{T} \cdot {}^1_2 \mathbf{T} \cdot {}^2_3 \mathbf{T} \cdot {}^3_4 \mathbf{T} \cdot {}^4_5 \mathbf{T} \cdot {}^5_6 \mathbf{T} = \begin{bmatrix} {}^0_6 \mathbf{R}_{3 \times 3} & {}^0 \mathbf{p}_{6_{\text{org}}} \\ 0_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$t_{11} = -S_6(C_4S_1 + S_4(C_1S_3 - C_2 - C_1C_3S_2)) - C_6(C_5(S_1S_4 - C_4(C_1S_3 - C_2 - C_1C_3S_2)) - S_5(C_1C_3 - C_2 + C_1S_2S_3))$$

$$t_{12} = S_6(C_5(S_1S_4 - C_4(C_1S_3 - C_2 - C_1C_3S_2)) - S_5(C_1C_3 - C_2 + C_1S_2S_3)) - C_6(C_4S_1 + S_4(C_1S_3 - C_2 - C_1C_3S_2))$$

$$t_{13} = -S_5(S_1S_4 - C_4(C_1S_3 - C_2 - C_1C_3S_2)) - C_5(C_1C_3 - C_2 + C_1S_2S_3)$$

$$t_{14} = 135C_1S_2 - 70S_5(S_1S_4 - C_4(C_1S_3 - C_2 - C_1C_3S_2)) - 70C_5(C_1C_3 - C_2 + C_1S_2S_3) - 120C_1C_3 - C_2 - 120C_1S_2S_3 - 38C_1S_3 - C_2 + 38C_1C_3S_2$$

$$t_{21} = S_6(C_1C_4 + S_4(C_3S_2S_1 - S_1S_3 - C_2)) + C_6(C_5(C_1S_4 - C_4(C_3S_2S_1 - S_1S_3 - C_2)) + S_5(C_3S_1 - C_2 + S_2S_1S_3))$$

$$t_{22} = C_6(C_1C_4 + S_4(C_3S_2S_1 - S_1S_3 - C_2)) - S_6(C_5(C_1S_4 - C_4(C_3S_2S_1 - S_1S_3 - C_2)) + S_5(C_3S_1 - C_2 + S_2S_1S_3))$$

$$t_{23} = S_5(C_1S_4 - C_4(C_3S_2S_1 - S_1S_3 - C_2)) - C_5(C_3S_1 - C_2 + S_2S_1S_3)$$

$$\begin{aligned} t_{24} = & 135S_2S_1 + 70S_5(C_1S_4 - C_4(C_3S_2S_1 - S_1S_3 - C_2)) - 70C_5(C_3S_1 - C_2 \\ & + S_2S_1S_3) + 38C_3S_2S_1 - 120C_3S_1 - C_2 - 120S_2S_1S_3 - 38S_1S_3 - C_2 \end{aligned}$$

$$\begin{aligned} t_{31} = & C_6(S_5(C_3S_2 - S_3 - C_2) + C_4C_5(C_3 - C_2 + S_2S_3)) - S_4S_6(C_3 - C_2 \\ & + S_2S_3) \end{aligned}$$

$$\begin{aligned} t_{32} = & -S_6(S_5(C_3S_2 - S_3 - C_2) + C_4C_5(C_3 - C_2 + S_2S_3)) - C_6S_4(C_3 - C_2 \\ & + S_2S_3) \end{aligned}$$

$$t_{33} = C_4S_5(C_3 - C_2 + S_2S_3) - C_5(C_3S_2 - S_3 - C_2)$$

$$\begin{aligned} t_{34} = & 120S_3 - C_2 - 120C_3S_2 - 38C_3 - C_2 - 38S_2S_3 - 135 - C_2 \\ & - 70C_5(C_3S_2 - S_3 - C_2) + 70C_4S_5(C_3 - C_2 + S_2S_3) + 135 \end{aligned}$$

8.2 Jacobian matrix

8.2.1 Jg0

$$j_{g0,21} = 135C_1S_2 - 120C_1S_2S_3 - 70S_1S_4S_5 + 120C_1C_2C_3 + 38C_1C_2S_3$$

$$+ 38C_1C_3S_2 + 70C_1C_2C_3C_5 - 70C_1C_5S_2S_3 - 70C_1C_2C_4S_3S_5$$

$$- 70C_1C_3C_4S_2S_5$$

$$j_{g0,22} = - S_1(120C_2S_3 - 38C_2C_3 - 135C_2 + 120C_3S_2 + 38S_2S_3 + 70C_2C_5S_3$$

$$+ 70C_3C_5S_2 + 70C_2C_3C_4S_5 - 70C_4S_2S_3S_5)$$

$$j_{g0,23} = - 2S_1(60C_2S_3 - 19C_2C_3 + 60C_3S_2 + 19S_2S_3 + 35C_2C_5S_3$$

$$+ 35C_3C_5S_2 + 35C_2C_3C_4S_5 - 35C_4S_2S_3S_5)$$

$$j_{g0,24} = 70S_5(C_1C_4 + C_2S_1S_3S_4 + C_3S_1S_2S_4)$$

$$j_{g0,25} = - 70C_5(C_2C_4S_1S_3 - C_1S_4 + C_3C_4S_1S_2) - 70C_3S_1S_5$$

$$j_{g0,26} = 0$$

$$j_{g0,31} = 0$$

$$j_{g0,32} = 120S_2S_3 - 120C_2C_3 - 38C_2S_3 - 38C_3S_2 - 135S_2 + 70C_5S_2S_3$$

$$- 70C_2C_3C_5 + 70C_2C_4S_3S_5 + 70C_3C_4S_2S_5$$

$$j_{g0,33} = 120S_2S_3 - 38C_2S_3 - 38C_3S_2 - 120C_2C_3 + 70C_5S_2S_3 - 70C_2C_3C_5$$

$$+ 70C_2C_4S_3S_5 + 70C_3C_4S_2S_5$$

$$j_{g0,34} = 70C_{34}S_4S_5$$

$$j_{g0,35} = 70S_{34}S_5 - 70C_{34}C_4C_5$$

$$j_{g0,36} = 0$$

$$j_{g0,41} = \theta_4S_1S_2S_3 - \theta_3C_1 - \theta_5C_1C_4 - \theta_4C_2C_3S_1 - \theta_6C_1S_4S_5 - \theta_2C_1$$

$$- \theta_6C_2C_3C_5S_1 - \theta_5C_2S_1S_3S_4 - \theta_5C_3S_1S_2S_4 + \theta_6C_5S_1S_2S_3$$

$$+ \theta_6C_2C_4S_1S_3S_5 + \theta_6C_3C_4S_1S_2S_5$$

$$j_{g0,42} = \theta_5C_1C_2C_3S_4 - \theta_4C_1C_2S_3 - \theta_4C_1C_3S_2 - S_1 - \theta_6C_1C_2C_5S_3$$

$$- \theta_6C_1C_3C_5S_2 - \theta_5C_1S_2S_3S_4 - \theta_6C_1C_2C_3C_4S_5 + \theta_6C_1C_4S_2S_3S_5$$

$$j_{g0,43} = \theta_5C_1C_2C_3S_4 - \theta_4C_1C_2S_3 - \theta_4C_1C_3S_2 - S_1 - \theta_6C_1C_2C_5S_3$$

$$- \theta_6C_1C_3C_5S_2 - \theta_5C_1S_2S_3S_4 - \theta_6C_1C_2C_3C_4S_5 + \theta_6C_1C_4S_2S_3S_5$$

$$j_{g0,44} = \theta_5(S_1S_4 + C_1C_2C_4S_3 + C_1C_3C_4S_2) - C_1S_2S_3 + C_1C_2C_3$$

$$+ \theta_6S_5(C_1C_2S_3S_4 - C_4S_1 + C_1C_3S_2S_4)$$

$$j_{g0,45} = - C_4S_1 - \theta_6(C_5(S_1S_4 - C_4(C_1S_3 - C_2 - C_1C_3S_2)) - S_5(C_1C_3$$

$$- C_2 + C_1S_2S_3)) - S_4(C_1S_3 - C_2 - C_1C_3S_2)$$

$$j_{g0,46} = C_{34}C_1C_5 - S_5(S_1S_4 + C_1C_2C_4S_3 + C_1C_3C_4S_2)$$

$$j_{g0,51} = \theta_4C_1C_2C_3 - \theta_3S_1 - \theta_5C_4S_1 - \text{theta2}S_1 - \theta_4C_1S_2S_3 - \theta_6S_1S_4S_5$$

$$+ \theta_6C_1C_2C_3C_5 + \theta_5C_1C_2S_3S_4 + \theta_5C_1C_3S_2S_4 - \theta_6C_1C_5S_2S_3$$

$$- \theta_6C_1C_2C_4S_3S_5 - \theta_6C_1C_3C_4S_2S_5$$

$$j_{g0,52} = C_1 - \theta_4C_2S_1S_3 - \theta_4C_3S_1S_2 + \theta_5C_2C_3S_1S_4 - \theta_6C_2C_5S_1S_3$$

$$- \theta_6C_3C_5S_1S_2 - \theta_5S_1S_2S_3S_4 - \theta_6C_2C_3C_4S_1S_5 + \theta_6C_4S_1S_2S_3S_5$$

$$j_{g0,53} = C_1 - \theta_4C_2S_1S_3 - \theta_4C_3S_1S_2 + \theta_5C_2C_3S_1S_4 - \theta_6C_2C_5S_1S_3$$

$$j_{g0,55} = C_1 C_4 + S_{34} S_1 S_4 - \theta_6 C_{34} S_1 S_5 + \theta_6 C_1 C_5 S_4 - \theta_6 C_2 C_4 C_5 S_1 S_3$$

$$- \theta_6 C_3 C_4 C_5 S_1 S_2$$

$$j_{g0,56} = C_{34} C_5 S_1 - S_5 (C_2 C_4 S_1 S_3 - C_1 S_4 + C_3 C_4 S_1 S_2)$$

$$j_{g0,61} = 1$$

$$j_{g0,62} = \theta_6 S_{34} C_4 S_5 - \theta_6 C_{34} C_5 - \theta_5 S_{34} S_4 - \theta_4 C_{34}$$

$$j_{g0,63} = \theta_6 S_{34} C_4 S_5 - \theta_6 C_{34} C_5 - \theta_5 S_{34} S_4 - \theta_4 C_{34}$$

$$j_{g0,64} = \theta_5 C_2 C_3 C_4 - C_3 S_2 - C_2 S_3 - \theta_5 C_4 S_2 S_3 + \theta_6 C_2 C_3 S_4 S_5 - \theta_6 S_2 S_3 S_4 S_5$$

$$j_{g0,65} = \theta_6 (S_{34} S_5 - C_{34} C_4 C_5) + C_{34} S_4$$

$$j_{g0,66} = - S_{34} C_5 - C_{34} C_4 S_5$$

8.2.2 Jg6

$$j_{g6,11} = 135C_4S_2S_6 + 120C_2C_3C_4S_6 + 70C_2C_3C_6S_4 + 38C_2C_4S_3S_6$$

$$+ 38C_3C_4S_2S_6 + 135C_5C_6S_2S_4 - 120C_4S_2S_3S_6 - 70C_6S_2S_3S_4$$

$$- 70C_2S_3S_5S_6 - 70C_3S_2S_5S_6 + 70C_2C_3C_4C_5S_6 + 120C_2C_3C_5C_6S_4$$

$$+ 38C_2C_5C_6S_3S_4 + 38C_3C_5C_6S_2S_4 - 70C_4C_5S_2S_3S_6$$

$$- 120C_5C_6S_2S_3S_4$$

$$j_{g6,12} = 70C_4C_6 - 38C_6S_5 - 120S_4S_6 - 70C_5S_4S_6 + 135S_3S_4S_6 + 120C_4C_5C_6$$

$$- 135C_3C_6S_5 - 135C_4C_5C_6S_3$$

$$j_{g6,13} = 70C_4C_6 - 38C_6S_5 - 120S_4S_6 - 70C_5S_4S_6 + 120C_4C_5C_6$$

$$j_{g6,14} = 70S_5S_6$$

$$j_{g6,15} = 70C_6$$

$$j_{g6,16} = 0$$

$$j_{g6,21} = 135C_4C_6S_2 + 120C_2C_3C_4C_6 + 38C_2C_4C_6S_3 + 38C_3C_4C_6S_2$$

$$- 70C_2C_3S_4S_6 - 120C_4C_6S_2S_3 - 70C_2C_6S_3S_5 - 70C_3C_6S_2S_5$$

$$- 135C_5S_2S_4S_6 + 70S_2S_3S_4S_6 + 70C_2C_3C_4C_5C_6 - 120C_2C_3C_5S_4S_6$$

$$- 70C_4C_5C_6S_2S_3 - 38C_2C_5S_3S_4S_6 - 38C_3C_5S_2S_4S_6$$

$$+ 120C_5S_2S_3S_4S_6$$

$$j_{g6,22} = 38S_5S_6 - 120C_6S_4 - 70C_4S_6 + 135C_6S_3S_4 + 135C_3S_5S_6$$

$$- 120C_4C_5S_6 - 70C_5C_6S_4 + 135C_4C_5S_3S_6$$

$$j_{g6,23} = 38S_5S_6 - 120C_6S_4 - 70C_4S_6 - 120C_4C_5S_6 - 70C_5C_6S_4$$

$$j_{g6,24} = 70C_6S_5$$

$$j_{g6,25} = - 70S_6$$

$$j_{g6,26} = 0$$

Reference

- [1] “the ministry of health and welfare,” 2018, <https://dep.mohw.gov.tw/dos/cp-1735-3245-113.html>. 1
- [2] “the academy of endodontology,” 2021, <http://www.aeroc.org.tw/about/>. 1
- [3] G. Kim, H. Seo, S. Im, D. Kang, and S. Jeong, “A study on simulator of human-robot cooperative manipulator for dental implant surgery,” *2009 IEEE International Symposium on Industrial Electronics*, pp. 2159–2164, 2009. 4
- [4] J. Li, Z. Shen, W. Xu, W. Y. Lam, R. T. C. Hsung, E. H. Pow, K. Kosuge, and Z. Wang, “A compact dental robotic system using soft bracing technique,” *IEEE Robotics and Automation Letters*, vol. 4, pp. 1271–1278, 2019. 4
- [5] “Yomi, neocis,” 2017, <https://www.neocis.com/>. 4
- [6] B. Sattapan, G. J. Nervo, J. E. Palamara, and H. H. Messer, “Defects in rotary nickel-titanium files after clinical use,” *Journal of Endodontics*, vol. 26, no. 3, pp. 161–165, 2000. 5, 54

- [7] M.-K. Wu, D. Barkis, A. Roris, and P. Wesselink, “Does the first file to bind correspond to the diameter of the canal in the apical region?” *International Endodontic Journal*, vol. 35, no. 3, pp. 264–267, 2002. 11
- [8] J. Agrawal, P. K. Shenai, L. Chatra, P. Y. Kumar *et al.*, “Evaluation of normal range of mouth opening using three finger index: South india perspective study,” *Indian Journal of Dental Research*, vol. 26, no. 4, p. 361, 2015. 11
- [9] “Meca500, mecademic,” 2021, <https://www.mecademic.com/>. 13
- [10] “Mini40, ati industrial automation, inc.” 2021, <https://www.ati-ia.com/index.aspx>. 13
- [11] C. Yang, J. Wang, L. Mi, X. Liu, Y. Xia, Y. Li, S. Ma, and Q. Teng, “A four-point measurement model for evaluating the pose of industrial robot and its influence factor analysis,” *Industrial Robot: An International Journal*, 2017. 23
- [12] T. Tang, H. Lin, Yu Zhao, Wenjie Chen, and M. Tomizuka, “Autonomous alignment of peg and hole by force/torque measurement for robotic assembly,” in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 162–167. 30
- [13] P. Zou, Q. Zhu, J. Wu, and J. Jin, “An approach for peg-in-hole assembling based on force feedback control,” in *2019 Chinese Automation Congress (CAC)*, 2019, pp. 3269–3273. 31

- [14] C. Boessler, F. Paque, and O. A. Peters, “The effect of electropolishing on torque and force during simulated root canal preparation with protaper shaping files,” *Journal of Endodontics*, vol. 35, no. 1, pp. 102–106, 2009. [56](#)