Design and Analysis of an Adjustable Wrist Rehabilitation Robot

by

Mirui Wang

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Applied Science

in

The Faculty of Engineering and Applied Science

Mechanical Engineering Program

University of Ontario Institute of Technology

July, 2014

Abstract

Numerous people lose motor capabilities of their hands every year because of the Stroke. Rehabilitation robot was used to recover the movement ability based on the plasticity theory. Many robots have been proposed but most of them are either expensive or quite complicated.

This thesis presents an adjust wrist rehabilitation robot. The proposed design fully utilizes the three rotational motions to rehabilitate the human wrist. A multi-objective optimization problem was applied to increase both stiffness and dexterity of the robot.

The CAD model is constructed and the inverse kinematic, Jacobian matrix, stiffness, dexterity formulas are derived. The workspace atlas is generated. The optimization work is conducted and it proves quick, robust and easy-to-use by the results comparisons.

Lastly, a CAD model was developed and results were verified by animation video and FEA analysis.

Acknowledgements

First and foremost, I would like to give my inmost gratitude to my supervisor, Dr. Dan Zhang, for his guidance, encouragement and continuous great support. I would also thank him for giving me this valuable opportunity to work in the domain of parallel mechanism. His broad knowledge, percipience and motivation encourage me throughout my research in Robotics and Automation Laboratory. I would like to thank him for his valuable supervision, full support and careful review of the manuscript. This thesis would and could never be what it is like right now without his contribution.

I would also like to thanks to my colleague Dr. Kailiang Zhang, his wide knowledge give me many ideas when I got problem. Dr. Zhen Gao's advice and revise on my thesis also make it much better. I couldn't have completed my thesis without their help.

Thanks will give to my parents and girlfriend in China. Without your patience, support and understanding, I would never make this possible.

Table of Contents

List of Figures

List of Tables

Nomenclature

$\boldsymbol{U}$	Matrix
$\boldsymbol{R}$	Transformation matrix
j	One row of the Jacobian matrix
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{v,w,u}$	Vector
c	Cosine function
s	Sine function
det	Determinant
$cond$	Condition number
$\dot{\boldsymbol{\theta}}$	Velocity vector of the actuators
$\boldsymbol{\omega}$	Velocity vector of the end-effector
$\boldsymbol{K}$	Stiffness matrix
$trace$	Trace of the matrix
M	Manipulability
F	Mutation rate
$rand$	Random number
j	Current element in the vector
$jrand$	Random number within the number of variables
Cr	Crossover rate
NP	Number of population

Greek Letters

θ	Angle
β	Angle that determines the shape of pyramid
λ	Angle that determines the shape of pyramid
α	Angle between two links
ϕ	Azimuth angle
σ	Torsion angle
η	Angle between two edges on the pyramid
δ	Tolerance
μ	Singularity value

Superscript

*	Vector that in the moving platform coordinate system
^{T}	Transposition of the matrix
^{g}	Number of current generation

Subscripts

i	One point in the workspace
min	Minimum
max	Maximum

Acronyms

IKP	Inverse Kinematics Problem
FKP	Forward Kinematic problem

T&T Tilt and Torsion

DE Differential Evolution

GA Genetic Algorithm

GSI Globe Stiffness Index

GCI Globe Condition Index

RRR Rotation, Rotation and Rotation

1 Introduction

1.1 Background

A stroke is the loss of brain function because of the disruption or blockage of the blood vessel. As a result, the affected part of the brain cannot work normally which leads to inability to move one or more limbs, as shown in Figure 1-1. According to statistics, about 800,000 people in the United States have a stroke each year and only 3,000,000 can survive after the stroke [1]. This number grows up to 2,000,000 in China every year [2]. Even in Canada, 50,000 people suffer the stroke each year [3].

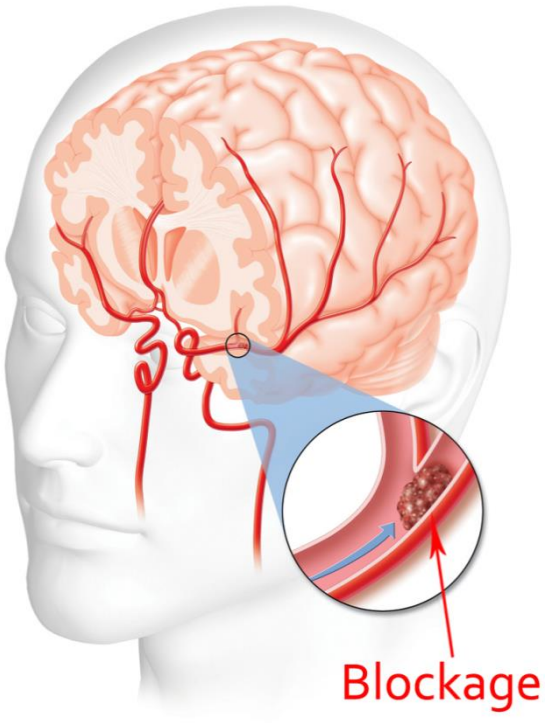

Figure 1-1: The Strokes [4]

Most of the patients have to experience long time of motor impairment. After some medical treatment, the vital signs come to stable. But the injury of the brain cannot be recovered.

Figure 1-2: The Phsical Treamment of After the Stroke [5]-[6]

Part of the movement ability can be regained if the patients keep receiving some rehabilitation training, actively or positively, under the brain plasticity theory, which also was called as neuroplasticity [4], as shown in the Figure 1-2.

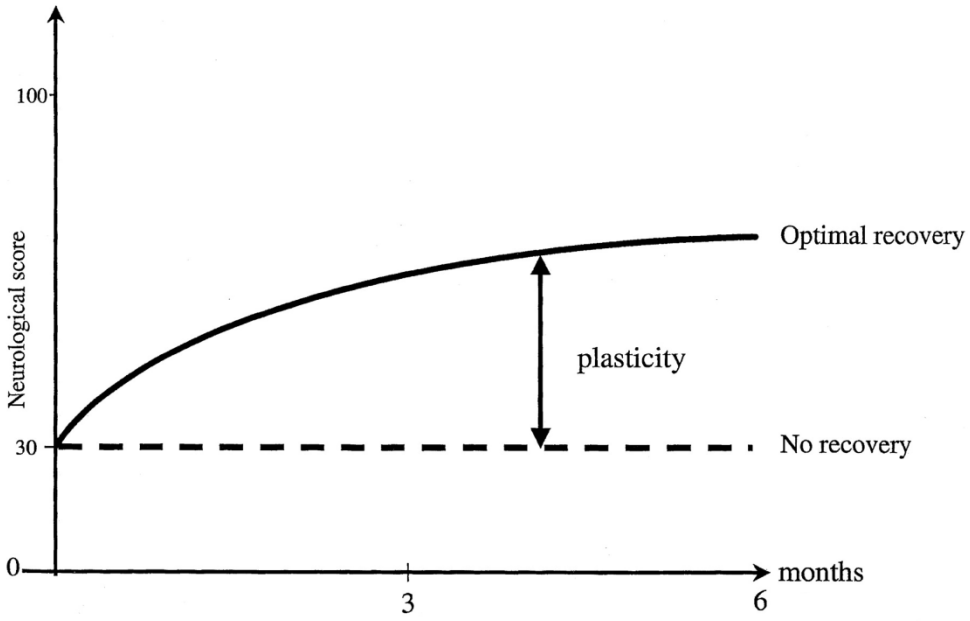

Figure 1-3: The Brain Plasticity Theory [1]

However, from the Figure 1-3, this recovery process is time-consuming, labour-intensive and it needs experienced therapists. Consequently, rehabilitation robot was generated at this right time. Since 1980, various kind of robotic therapy devices were researched in different countries [7]. A wrist rehabilitation device based on Spherical Parallel Mechanism (SPM) was proposed in this thesis to help people recover from either the stroke or physical injury.

In order to clearly understand the wrist rehabilitation robot, it is necessary to describe some background about the geometrical reference of hand. There are two reference planes in anatomy: Frontal Plane and Sagittal Plane. The plane that is coplanar with palm is called as Frontal Plane or Coronal Plane. The plane that goes through with middle finger and perpendicular with coronal plane is called as Sagittal Plane.

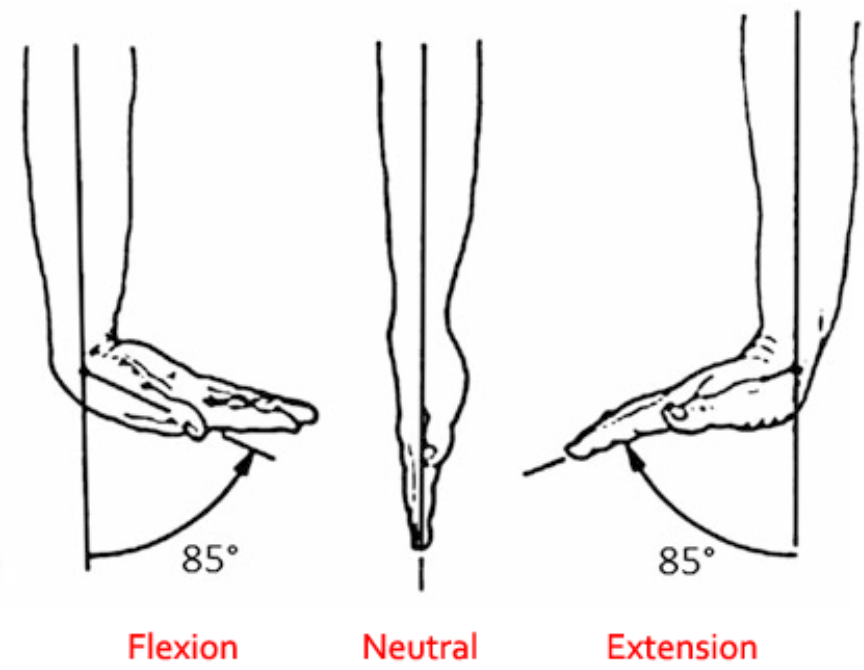

Figure 1-4: Flexion, Neutral and Extension [29]

The movement on the sagittal plane is called as flexion-extension, as shown in Figure 1-4. These two movements all start at the anatomic position. If the palm goes towards the palm side, it is calls as flexion. If the palm goes towards the dorsal surface, it names extension.

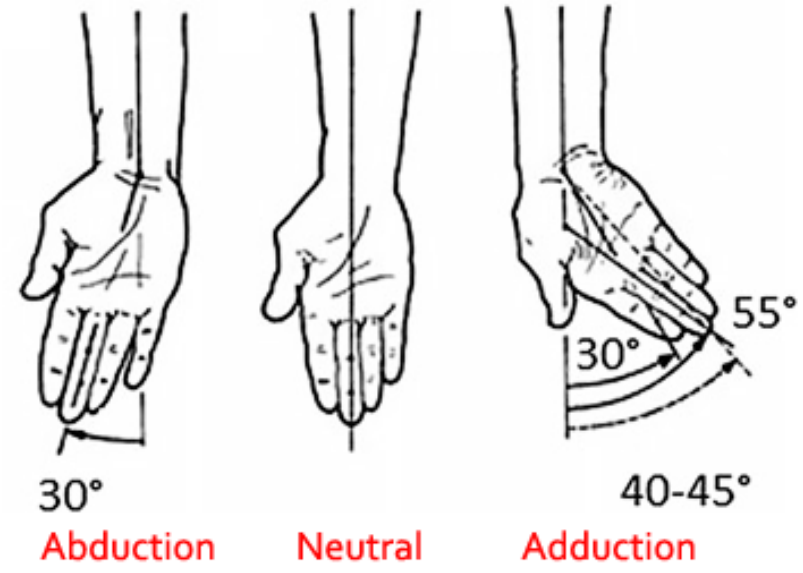

Figure 1-5: Abduction, Neutral and Adduction [29]

Similarly, the movement on the frontal plane contains abduction and adduction. From the anatomic position, as shown in Figure 1-5, if the palm goes toward the thumb finger, then this movement is called as adduction. If it goes toward little finger, it is labeled as adduction [29].

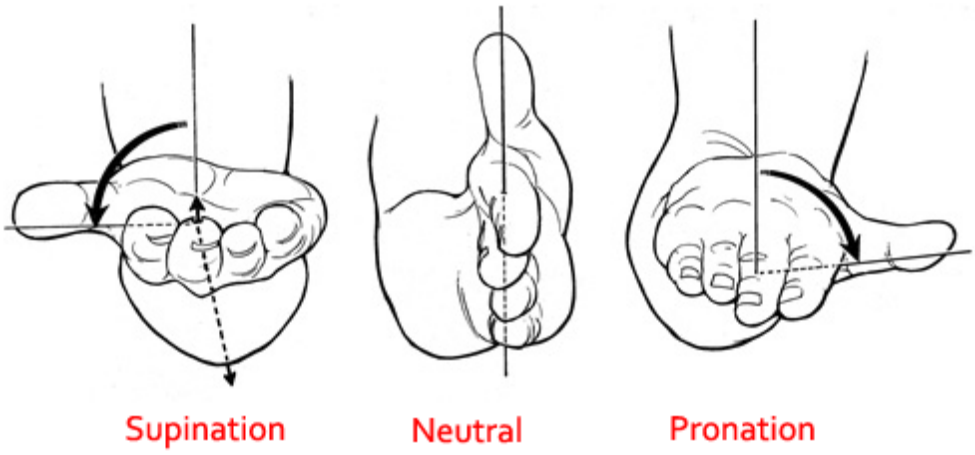

Figure 1-6: Supination, Neutral and Pronation [29]

The third movement comes from the forearm, as shown in Figure 1-6. Supination and pronation are the rotation movement that can rotate 180 degrees of the forearm. Pronation at the forearm are rotational movement where the hand and upper arm are turned inwards, while supination of the forearm occurs when the forearm or palm are rotated outwards [30].

The amplitude of flexion and extension can reach to 85 degree when there is no abduction and adduction. However, the wrist abduction can only go to 15 degree and the adduction has 30 degree.

1.2 Motivation and Objectives

In order to help millions of Stroke patients to quickly regain their motor ability, this thesis aims to design an inexpensive and compact wrist rehabilitation robot.

The specific objectives of this thesis are listed as follows:

1. Design a wrist rehabilitation robot based on 3RRR spherical parallel mechanism that can perform 3 degree-of-freedom rehabilitation therapy.
2. Describe the robot structure in an appropriate coordinate system; obtain the inverse kinematics and analyze the workspace.
3. Perform the kinematic analysis and optimization work to insure that the structure has the best kinematic performance.
4. Complete necessary mechanical static and frequency analysis.
5. Design the device that makes sure the wrist rehabilitation robot can fit for the hands of different people.

1.3 Literature review

In this sub-section, the review of Parallel mechanism will first be introduced. Then the rehabilitation robot will be discussed.

1.3.1Parallel Mechanism

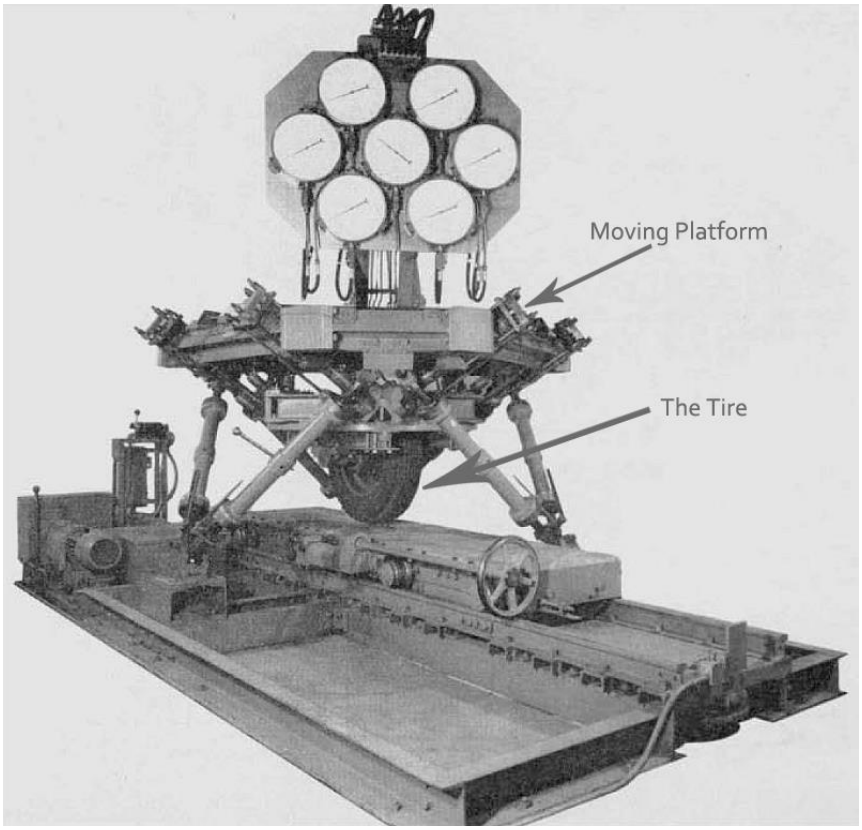

Figure 1-7: The Gough Tire Test Platform [8]

Parallel Mechanism (PM) has been researched for almost 67 years since the first parallel manipulator was used to test the tire [8] in Figure 1-7. From then on, numerous scientists kept working on that and made great contributions in this field. Compared with serious robot which has been widely used in the industry, parallel manipulator offers high payload, high stiffness and high precision [9].

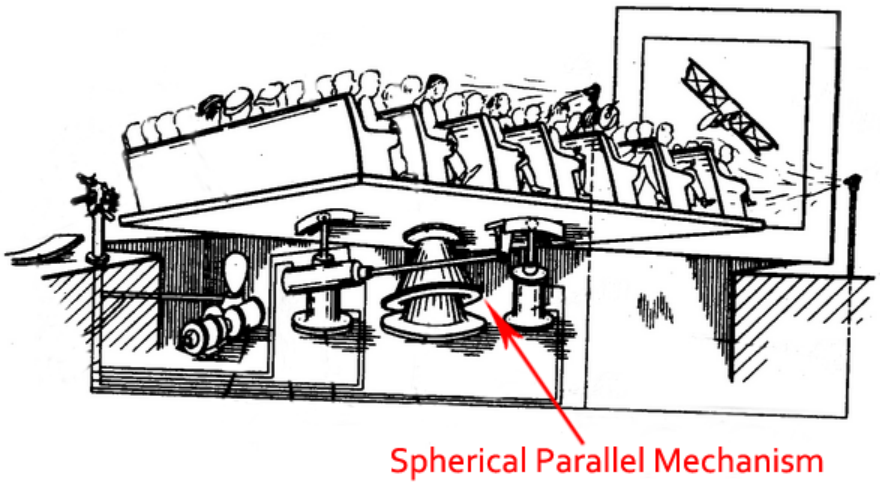

Figure 1-8: The Earliest Spherical Parallel Mechanism [10]

As an important part of parallel robot, Spherical Parallel Mechanism (SPM) always attracts researchers' attention because of its special feature. Actually, the first parallel mechanism is reported from a patent that a spherical mechanism was introduced and used in the theater in 1928 [10], as shown in Figure 1-8.

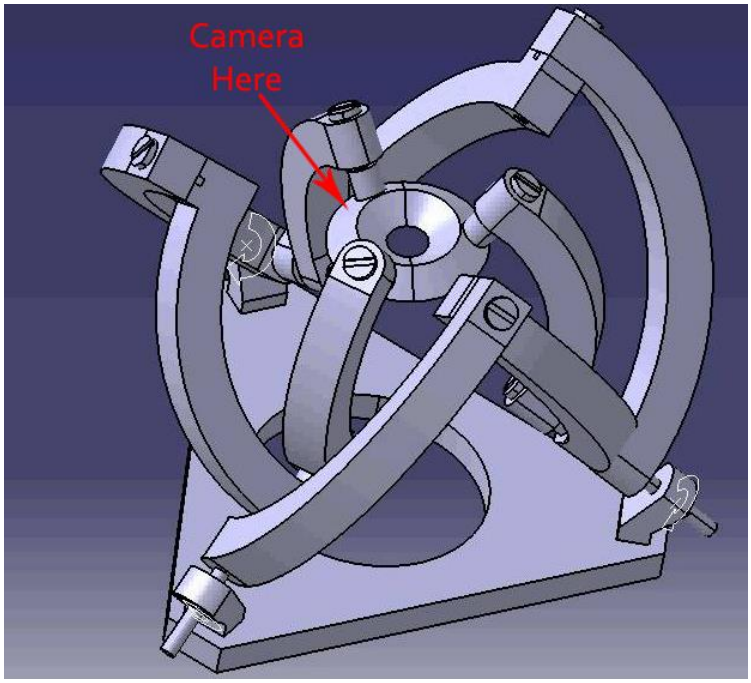

Figure 1-9: The Agile Eye [11]

Due to its pure three degrees of freedom of rotation, orientation device is the main application of spherical parallel robot, For example, the Agile Eye was the most famous application of SPM [11]. Its application in wrist joint also can be found in literature [12] and shown in Figure 1-9.

For the general model of the spherical parallel manipulator, it contains 6 revolt joints and 3 pair of legs. The most interesting point of SPM is that their 2 rotation centers are coincident with each other. This thesis is aimed to design a human wrist re-habitation device by taking advantage of this pure 3 degrees-of-freedom of rotation of SPM.

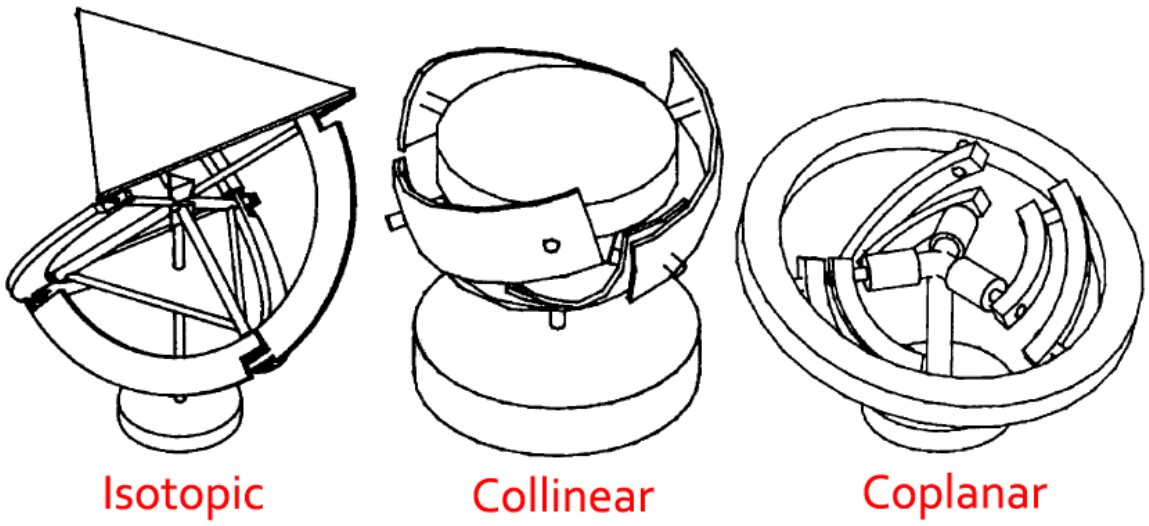

Figure 1-10: Isotopic, collinear and coplanar SPM [15]

Without any doubt, Gosselin family played the leading role in the research of SPM [13]. They did many works in this field, such as workspace, singularity and direct kinematics. This section will briefly review their work. Gosselin [13] did kinematic analysis and workspace optimization of a coplanar SPM in his doctoral dissertation. He [14] also proved that SPM possessed the maximum workspace when the two link

angles were equal to 90 degree. In his following work [15], he systematically illuminated the kinematic analysis of three special cases of SPM, namely the shoulder module, coplanar manipulator and collinear manipulator. After that, Gosselin et al. [16] gave the direct kinematics of SPM in general module and coplanar platform. Gosselin et al. [18] presented a particular geometry of SPM in which the axes of the frame perpendiculars with each other. This isotropic configuration was adopted by the renowned Agile Eye and it led to easy direct kinematic problem.

In terms of singularity of SPM, it is quite difficult to analyze the general case. Gosselin et al. [19] studied the special class of SPM, which is the class with axes collinear with the coordinate frame. Bonev et al. [20] used another kind of coordinate system to analyze the singularity loci of spherical parallel mechanism in the isotropic configuration and in a more explicit way. Bonev et al. [21] also made in-depth study of the direct kinematic, workspace and the singularity analysis of isotropic configuration.

On the optimal design of manipulator, Gosselin et al. [13] studied the optimum design of spherical parallel robot by considering three different standards: symmetry, workspace and isotropy. He [23] proposed the concept of Global Condition Index to evaluate the robot's dexterity. This method counts all the points in the workspace and has been used as an important performance index for a long time. The drawback of this method is time consuming [24]. Then Gosselin [25] presented another dexterity index based on the condition number of the Jacobian

matrix. Hang et al. [26] employed the analytical method to perform optimal design of perpendicular spherical manipulator. The analytical result was validated the previous work and offered a new approach to do the optimal design in the future. Bai [27] optimized the maximum dexterity by analyzing the inequality which was derived from the inverse kinematic problem. This numerical method at last transformed to solve a nonlinear least squares problem which can be figured out immediately by Matlab. Wu [28] performed multiobjective optimization of 3RRR spherical parallel robot in view of both global conditional number and dynamic dexterity in terms of generalized inertia ellipsoid.

1.3.2Wrist Rehabilitation Robot

There are some classic rehabilitation robots that have been developed for many years with extensive clinical test, such as MIT-MANUS [31], ARM [33] and MIME [34]. More other early research results can be found in [7].

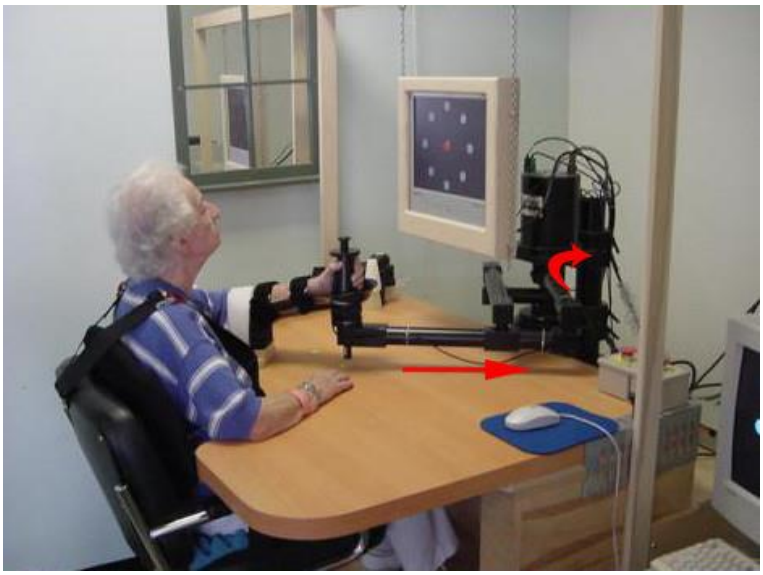

Figure 1-11: MIT-MANUS Rehabitation Robot [31]

The first robot-based therapy device is the MIT-MANUS which was developed by Hogan and Krebs in MIT [31]. This manipulator has three parts: planar module, wrist module and hand module. The wrist module offers three degrees-of-freedom which helps patient to move by virtue of forearm and wrist joint. Hand part can assist hand to do some hand grasp practice. Software interface also was developed that the patient can see the whole training process during the treatment. Many clinical researches have been done based on this robot and got some results. The robot-treated group received more motor ability than the control group without any side-effect.

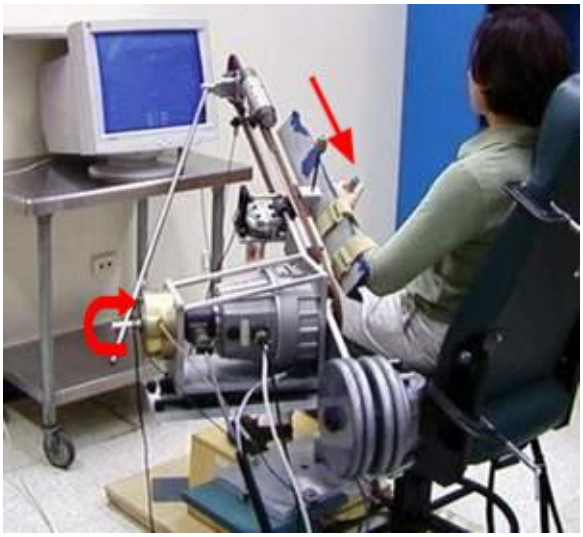

Figure 1-12: The ARM Guide [33]

David developed a therapy robot called ARM Guide (Assisted Rehabilitation and Measurement Guide) in 2000 [33]. Patient's one hand was fixed on the splint and the motor drives it move on the straight line. Both the angles of zenith and tilt can be adjusted. After training, the range of the patient's movement increased to some

extent. The top speed of the movement of their hand got improved and the total motor control ability got great enhanced.

Figure 1-13: The MIME Robot [34]

In the same year, Stanford University lunched a therapy robot named Mirror Image Movement Enhancer (MIME) system [34]. This system was based on Puma-560 robot to help the patient's arm to make the movement. Patient's one arm attaches to the splint and there is a connector to separate from system if the force is too large. Thanks to the six degree-of-freedom of Puma robot, MIME offers a more movement than MIT-MANUS. Four control modes can be used during the therapy: active assist mode, passive mode, active constrained mode and mirror-image mode. The most innovative was the last two modes. In the active constrained mode, individuals allow to move actively to a target, while the system can prevent patient's path deviate. The mirror-image mode is useful when patients' arm impairment is asymmetric.

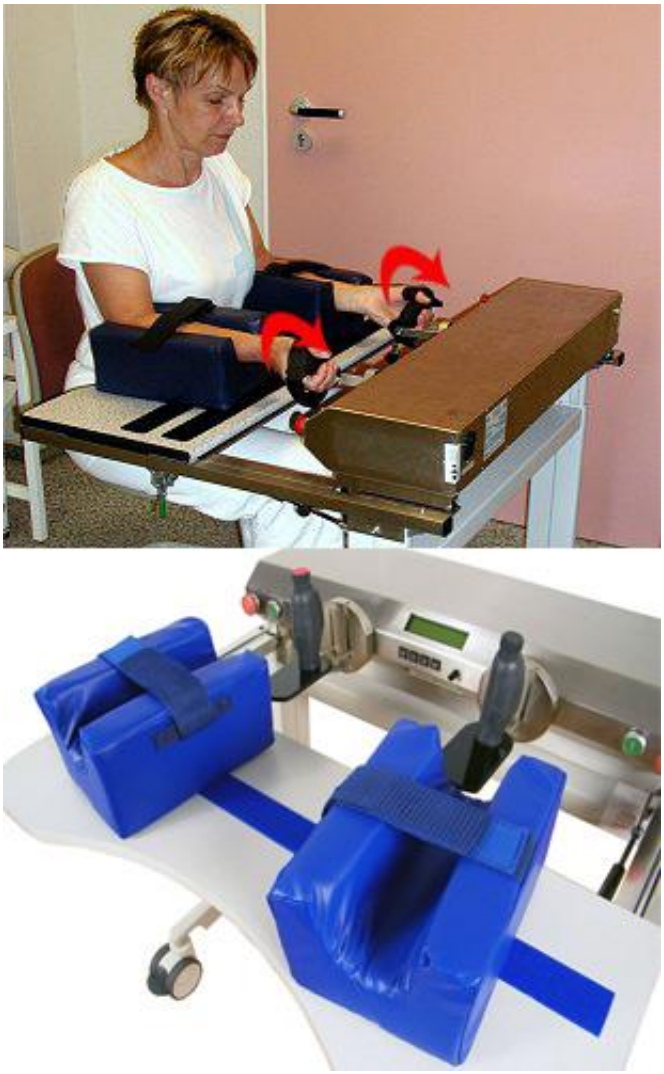

Figure 1-14: Bi-Manu-Track [32]

The Bi-Manu-Track was developed in German and only offers two degree-of-freedom, namely the pronation and supination, the flexion and extension [32]. The Bi-Manu-Track also offers two movement modes, active and passive. The whole system is simple but takes a quite large place.

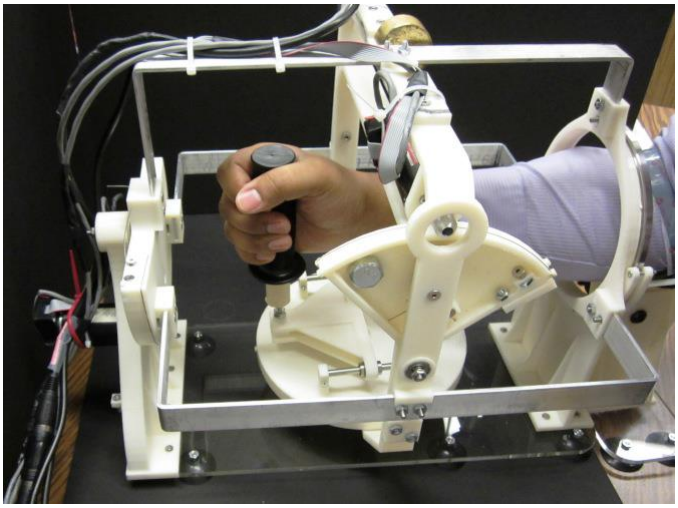

Figure 1-15: Exoskeleton Therapy Robot [32]

Martinez [35] presented a 3 DOF exoskeleton therapy robot used for stroke rehabilitation. It is called Wrist Gimbal. This device uses three revolute joints to realize 3 DOF movements in serial kinematic configuration. Two bearing supports were applied in order to increase the robustness and rigidity. Each axis has an emergency stop button. The whole structure design is similar as an exoskeleton and is quite complicated.

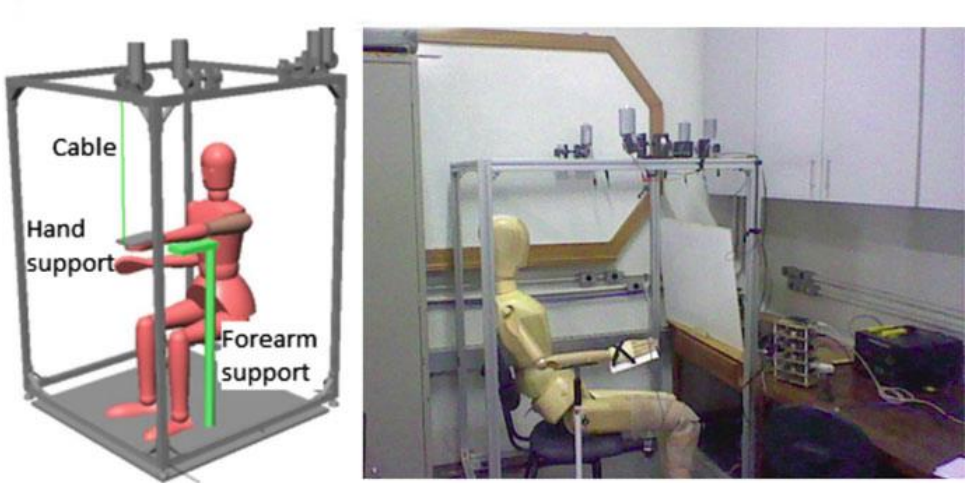

Figure 1-16: Cable-Based Wrist Rehabitation Robot [33]

Ribeiro et al. [36] developed a wrist rehabilitation system based on the cable-based parallel mechanism. It only uses two cables to achieve 2 degrees-of-freedom movements: flexion-extension and abduction-adduction. Four motors drive the cable to move the hand. The hand is fixed on a plate with all fingers unfold. The whole arm was attached on a supporter. Because of the nature of cable-drive mechanism, large operation room is necessary.

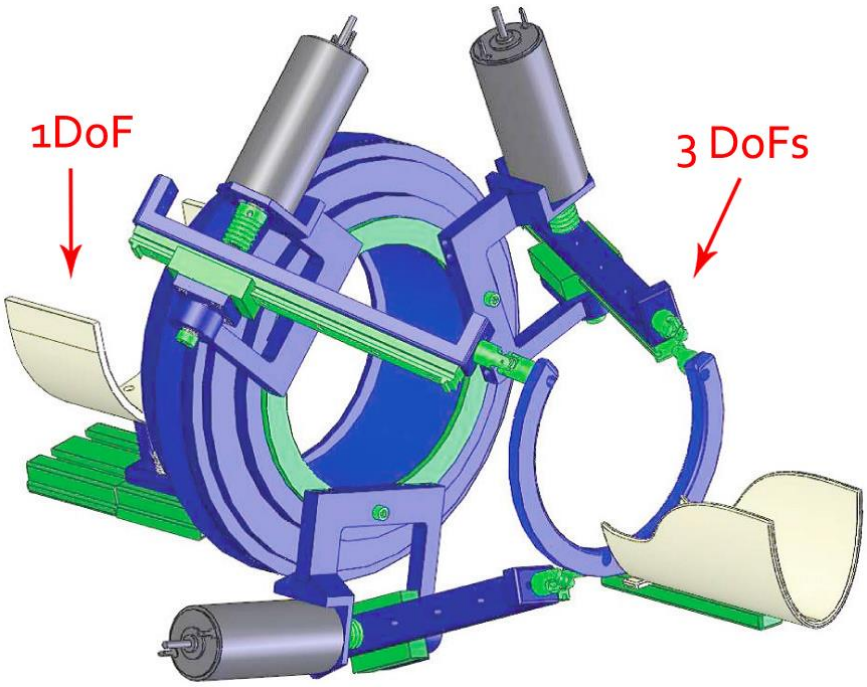

Figure 1-17: Parallel Mechanism-Based Rehabitation Robot [37]

Abhishek et al. [37] designed a Rice Wrist exoskeleton robot based on serial-in-parallel mechanism. It contains a revolute joint to afford the arm. Another end of the revolute joint connects with a 3-RPS parallel platform where the revolute joint is located at the base. The actuator is located at the prismatic joint and the spherical joint is placed at the end-effector. By controlling the length of the prismatic joint, the

parallel part can reach to 3 degrees-of-freedom, plus one degree-of-freedom on the arm.

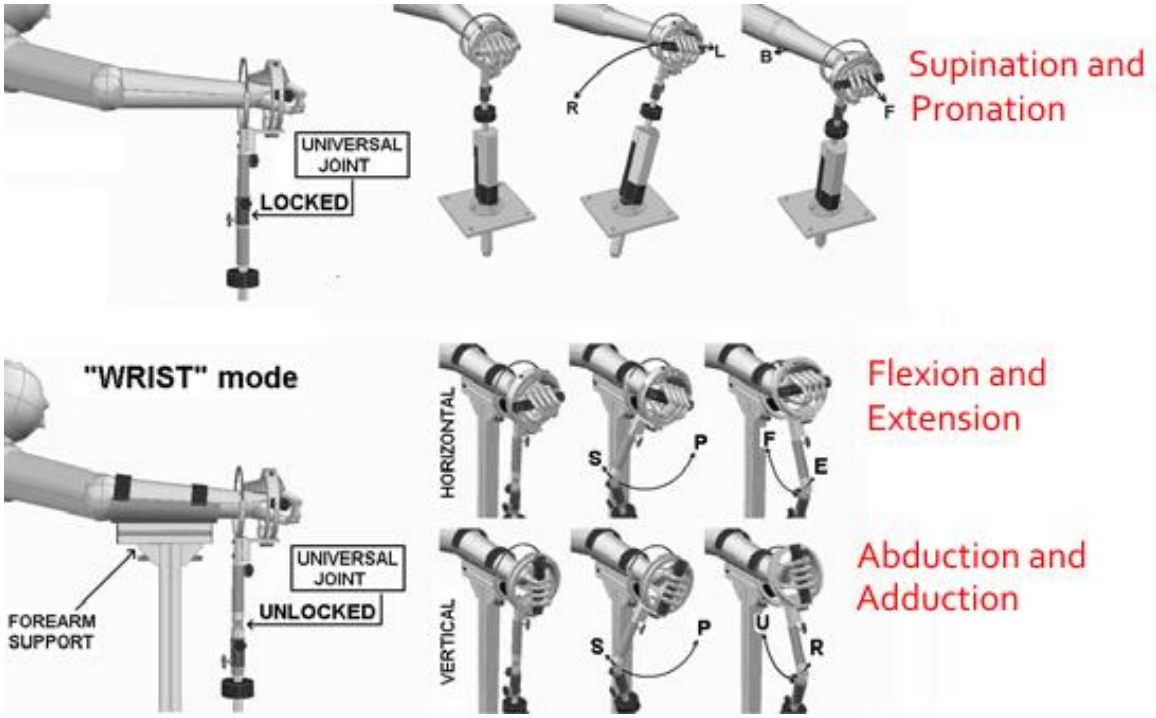

Figure 1-18: Universal Rehabitation Robot [38]

Oblak [38] presented a universal haptic drive rehabilitation device. It is called as "universal" because it can recover both wrist and arm and the main function based on a universal joint. When the universal joint is locked, it can offer 1 DOF for arm to move. When it is unlocked, it can offer 2 DOF for wrist to recover. But the total system needs another base platform to offer driving force and auxiliary equipment.

Figure 1-19: Supinator Extender [36]

Allington [39] designed a robot called Supinator Extender. It is a 2 DOF serial kinematic chain that can offer supination/pronation for the arm and flexion/extension for the wrist. Pneumatic actuators were adopted in order to remain lightweight for individuals to carry. On the other hand, pneumatic drive needs extra equipment to offer constant pressured gas.

To summary, many therapy devices have been proposed to help patients to recover their wrist movement ability. Parallel and serial structures are the mainstream in the rehabilitation device. Most of them offer 2 DOF movements: flexion-extension and adduction-abduction. Certain of them give extra arm movement: pronation-supination. In terms of power, most of them use DC motor, only one use pneumatic power.

1.4 Outline

The rest of the chapters are organized as follows,

Chapter 2 will illustrate the geometric description of the SPM, inverse kinematics and workspace analysis.

Chapter 3 will derive the kinematic indexes, namely stiffness and dexterity.

Chapter 4 will introduce the proposed Differential Evolution and some optimization scenarios will be conducted and analyzed.

Chapter 5 will describe the detailed design of rehabilitation robot and perform the mechanical static and frequencies analysis.

Chapter 6 will give out the results and discussion.

Chapter 7 will present the conclusions and recommendations for the future research.

2 Inverse Kinematics and Workspace Analysis

2.1 Introduction

Geometric modelling is the fundamental work for the kinematic analysis. It reveals that how the mathematic relation was built up. The main work is to create a proper coordinate system in the base and moving platform in order to calculate the coordinates of the critical points.

Inverse Kinematics Problem (IKP) has significant meaning for Parallel Mechanism (PM) because it offers the values that actuator is required to reach to a specific position and orientation. IKP for parallel manipulator is pretty straightforward and most of them can be solved by vector method.

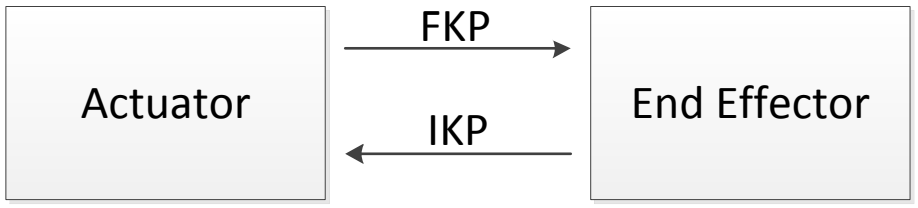

Figure 2-1: FKP and IKP

Once IKP has been solved, the workspace is at hand immediately. In literature, various methods have been suggested to calculate the workspace of the parallel manipulator. In detail, there are mainly three types of workspaces, namely, constant orientation workspace, reachable workspace and dextrous workspace. The dextrous

workspace is generally preferred for its pragmatic in the real world application. Hence it has been employed also in this thesis.

This chapter will first present how the two coordinates are constructed. Then the inverse kinematics will be derived. Finally, the workspace atlas is given to show the workspace of the spherical workspace.

2.2 Geometric Description

The general spherical parallel manipulator is shown in the Figure 2-3. There are three identical kinematic chains with 3 revolute joints on each leg. From the top to the bottom, three links connecting with moving platform have the same angles α_2, while the three links that join with base have identical angles α_1. Moreover, β and γ are the angles which define the upper and lower regular pyramids. Another way to define the regular pyramid is to use the angle between two edges [14] which leads to the same result. The joints on the basement are called as proximal joint and its axis denoted by unit vector $\boldsymbol{u}_i$. The joints on the end-effector are called as distal joint and denoted as vector $\boldsymbol{v}_i$. The last set of joints between the proximal and distal are called as intermediate joints. Unit vector $\boldsymbol{w}_i$ is applied to express its axes.

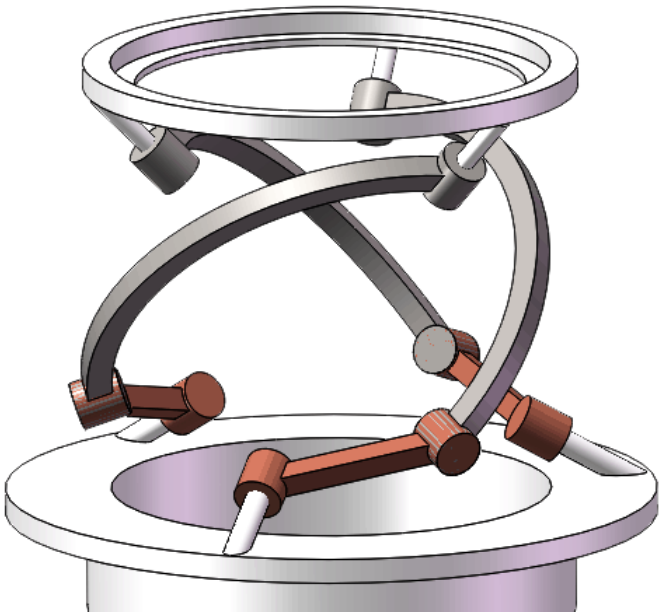

Figure 2-2: 3D model of the Spherical Parallel Mechanism

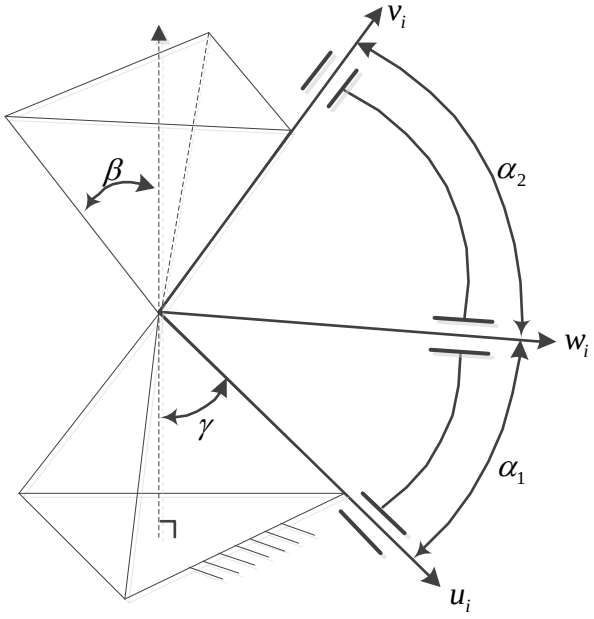

Figure 2-3: General Spherical Parallel Manipulator Model

The most important issue during the analysis of the SPM is the choice of coordinate system. For example, Gosslin et al. [17] got the simple solution of direct kinematics

of the SPM by taking advantage of isotropy of SPM configuration where the axes of basement joints and platform joints coincide with the Cartesian coordinate.

Because the end-effector will move around with moving platform in the three dimensional space through the parallel mechanism chain, it is necessary to represent the position and orientation of moving platform, namely, the coordinate system and conventions. 12 different Euler-angles are usually be used: XYZ, XZY, YXZ, YZX, ZXY, ZYX, XYX, XZX, YXY, YZY, ZXZ and ZYZ.

However, considering our rehabilitation application, our coordinate system must be able to describe the three therapy movements in an easy way. Thus, traditional 12 set of Euler-angle conventions cannot be adopted. A novel, less-well known coordinate representation called as Tilt-and-Torsion angle (T&T) will be used to describe the orientation of mobile platform here. It was proposed by Korein [40] in 1984 in order to model the human body joints. Bonev [41] first introduced it the research of spherical parallel robot and his later work [42] had proved that T&T took full advantage of mechanism's symmetry.

The three successive rotations in the T&T convention is like this: first, the body frame tilt about a horizontal axis a, at angle θ, referred as the tilt. This axis a is determined by another angle ϕ which is the angle between the projection of the z axis onto the fixed xy plane and the fixed x axis, is called as azimuth. Secondly, the frame rotates about the new z axis at the angle σ which is called as torsion.

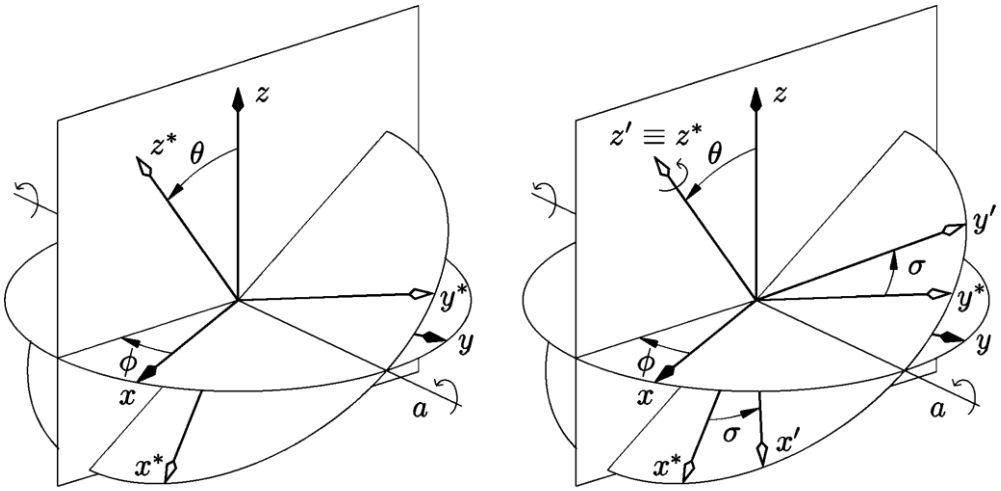

Figure 2-4: Tilt-Torsion Convention [38]

Under above definition, the rotation matrix of the T&T angle is expressed as below and the detailed derivation process can be found in [42].

$$\boldsymbol{R}(\phi,\theta,\sigma)=\begin{bmatrix} c\phi c\theta c(\sigma-\phi)-s\phi s(\sigma-\phi) & -c\phi c\theta s(\sigma-\phi)-s\phi c(\sigma-\phi) & c\phi s\theta \\ s\phi c\theta c(\sigma-\phi)+c\phi s(\sigma-\phi) & -s\phi c\theta s(\sigma-\phi)+c\phi c(\sigma-\phi) & s\phi s\theta \\ -s\theta c(\sigma-\phi) & s\theta s(\sigma-\phi) & c\theta \end{bmatrix} \quad (2\text{-}1)$$

where $c\phi=\cos\phi$, $s\phi=\sin\phi$, $c(\sigma-\phi)=\cos(\sigma-\phi)$ and $s(\sigma-\phi)=\sin(\sigma-\phi)$.

According to the selected coordinate system, the unit vector of $\boldsymbol{u}_i$ can be derived as:

$$\boldsymbol{u}_i=\begin{bmatrix} -\sin\eta_i\sin\gamma \\ \cos\eta_i\sin\gamma \\ -\cos\gamma \end{bmatrix} \quad (2\text{-}2)$$

where $\eta_i=\dfrac{2(i-1)\pi}{3}$, $i=1,2,3$.

Then $\boldsymbol{w}_i$, the unit vector of middle revolute joint axis in the i th leg is obtained as follows,

$$\boldsymbol{w_i} = \begin{bmatrix} -s\eta_i s\gamma c\alpha_1 + (c\eta_i s\theta_i - s\eta_i c\gamma c\theta_i)s\alpha_1 \\ c\eta_i s\gamma c\alpha_1 + (c\eta_i s\theta_i - s\eta_i c\gamma c\theta_i)s\alpha_1 \\ -c\gamma c\alpha_1 + s\gamma c\theta_i s\alpha_1 \end{bmatrix} \tag{2-3}$$

In the end, the unit vector of the axes of the distal link $\boldsymbol{v_i}$ is a function of the orientation of moving platform, then,

$$\boldsymbol{v_i} = \boldsymbol{R}\boldsymbol{v_i^*} \tag{2-4}$$

where $\boldsymbol{R}$ is the aforementioned rotation matrix and $\boldsymbol{v_i^*}$ is the unite vector of the axes of the top link joints expressed at home position in its own reference orientation, that is,

$$\boldsymbol{v_i^*} = \begin{bmatrix} -\sin\eta_i \sin\beta \\ \cos\eta_i \sin\beta \\ \cos\beta \end{bmatrix} \tag{2-5}$$

2.3 Inverse Kinematics

Inverse kinematics deals with problems by giving the position and orientation of the moving platform and to obtain the values of variables of actuators. Generally, inverse kinematics is pretty easy for parallel robot and forward kinematic is difficult to get the close-form solution. This feature is just opposite with serial robot.

Most of the inverse kinematic problems of parallel robot can be solved by its geometric and structure features and the vector method is one of the popular methods. For spherical parallel manipulator, there are some characteristics that different from normal parallel robot. One can write the dot product in the Eq. (2-6),

since the angle between distal link and intermediate link, that is $\boldsymbol{w}_i$ and $\boldsymbol{v}_i$, equals α_2.

$$\boldsymbol{w}_i \cdot \boldsymbol{v}_i = \cos\alpha_2 \quad \text{i=1,2,3} \tag{2-6}$$

Substitute the previous equations then leads to the inverse kinematics solution of this structure. With the help of tan-half identifies, namely,

$$\cos\theta = \frac{1-t^2}{1+t^2},\ \sin\theta = \frac{2t}{1+t^2},\ \text{t=tan}(\frac{\theta}{2}) \tag{2-7}$$

Rearrangement the new equation then produces,

$$Dt^2 + 2At + E = 0 \tag{2-8}$$

where,

$$D = v_x(s\eta_1 src\alpha_1 + s\eta_1 crc\alpha_1) + v_y(c\eta_i src\alpha_1 - c\eta_i crs\alpha_1) - v_z(crc\alpha_1 + srs\alpha_1) - c\alpha_2$$
$$E = v_x(s\eta_1 src\alpha_1 - s\eta_1 crc\alpha_1) + v_y(c\eta_i src\alpha_1 + c\eta_i crs\alpha_1) + v_z(srs\alpha_1 - crc\alpha_1) - c\alpha_2$$
$$A = v_x c\eta_1 s\alpha_1 + v_y s\eta_1 s\alpha_1$$

where $s\beta = \sin\beta$, cβ=cosβ, $\beta = \eta_i, r, \alpha_1, \alpha_2$, i=1,2,3. v_x, v_y, v_z are the x, y, z component of the $\boldsymbol{v}$. Thus, each orientation yields 8 set of solutions by the inverse kinematics. For the real world control application, decision must be made because only one set of solution can be used.

2.4 Workspace Analysis

2.4.1 Workspace without Constraints

When one directly visualizes the solutions from the inverse kinematic into the 3 dimensional spaces, it can be considered as the workspace without constraints. Though workspace without constraints is not meaningful in the real application, it can help one to have a brief landscape of the workspace. Some configurations and its workspaces will be shown and analyzed later.

2.4.2 Dextrous Workspace

In this scenario, all the real world constrains are considered, including singularity, ill-condition and link collision.

The first constraint is singularity. Considering matrix A and B, if the determinant of matrix A equals or close to 0, the mechanism is called as Type 2 singularity. If any one of the element in matrix B equals or closes to 0, it means that leg is in or close to folded or fully extended situation. These two situations should be avoided in the real world application, this leads to two inequalities,

$$\det(\boldsymbol{A}) \geq \varepsilon \tag{2-9}$$

$$\boldsymbol{B}_{ii} \neq 0 \tag{2-10}$$

The second constraint is the ill-condition of Jacobian matrix. Although we already considered type 1 and type 2 singularities, it is still necessary to consider the quality

of Jacobian matrix. Here we conform that the condition number at each point in the workspace is larger than 1.

$$cond(\boldsymbol{J}_i) > \delta \tag{2-11}$$

where δ is set to 1000, the determinant almost equals 0 that this value.

The last constraint is the collision between two contiguous links. There are two kinds of link collisions. The first kind is the collision that happens at two distal points between two adjacent links. The second kind is the collision that occurs between one link's distal with another's proximal which will be discussed in the Chapter 5. Figure 2-5 shows one collision possibility that may be happened.

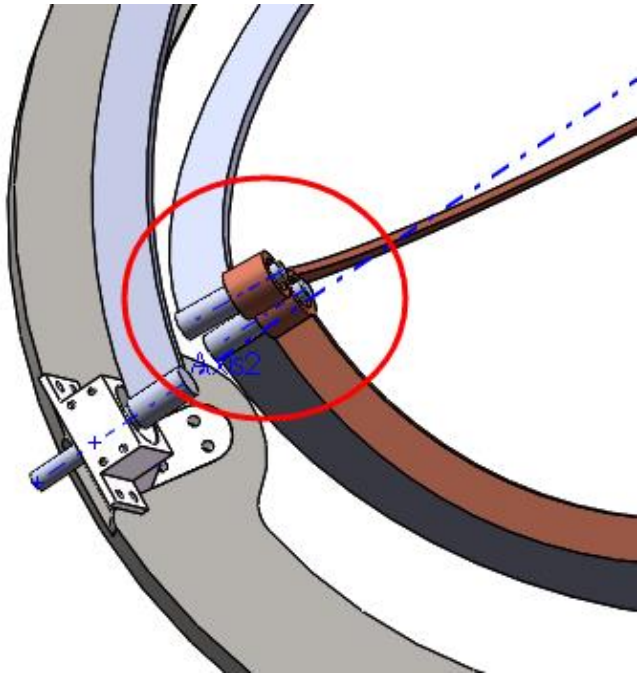

Figure 2-5: Two adjacent links collision at distal points

In order to avoid this becoming true, some constraints on the distance of two link's distal must to be add, that is, the angle between two neighbouring unit vector $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ ($i, j = 1,2,3 \ \ i \neq j$):

$$\theta_{12}, \theta_{13}, \theta_{23} \geq 10^{\circ} \tag{2-12}$$

where θ_{12} means the angle between unit vector w_1 and w_2.

2.4.3Workspace Atlas

There are 5 sets of workspace pictures with different α_1 and α_2 angles. Each set contains two pictures, the left one is the workspace without constraints and the right one is the workspace with all constraints.

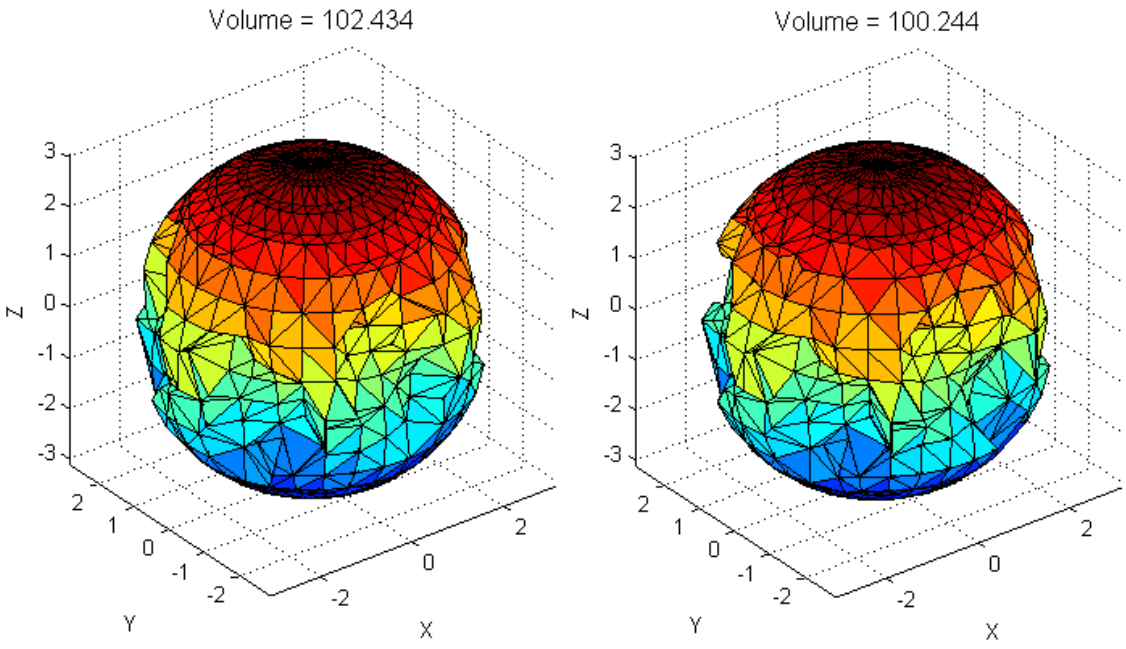

Figure 2-6: Workspace at $\alpha_1 = 90^{\circ}, \alpha_2 = 120^{\circ}$

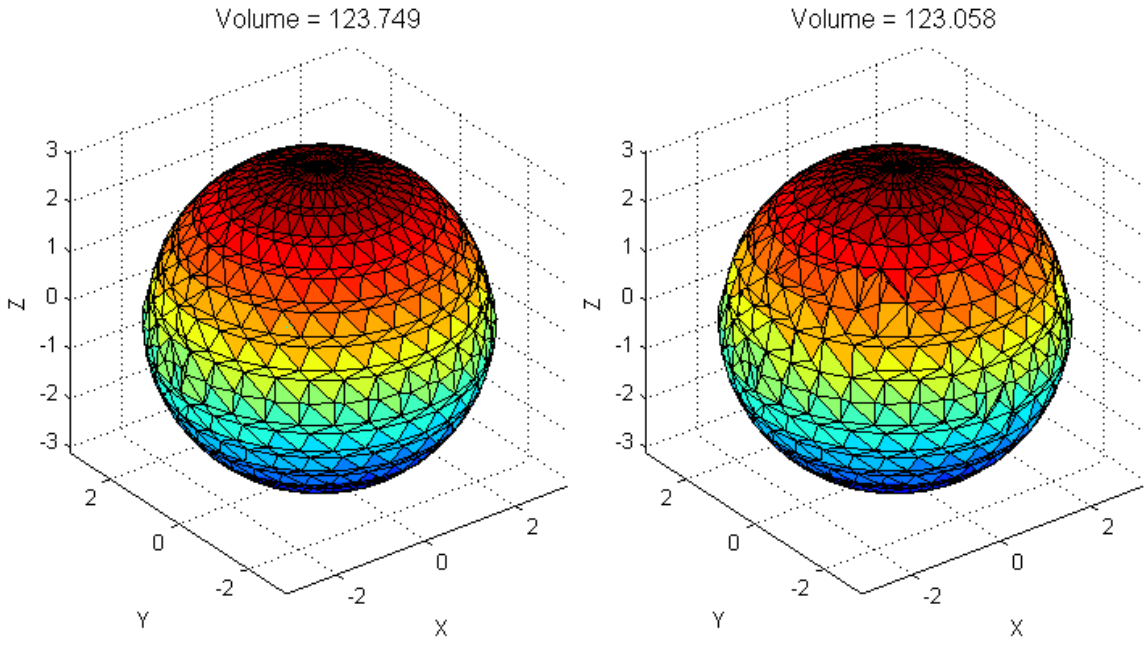

Figure 2-7: Workspace at $\alpha_1 = 90^\circ, \alpha_2 = 90^\circ$

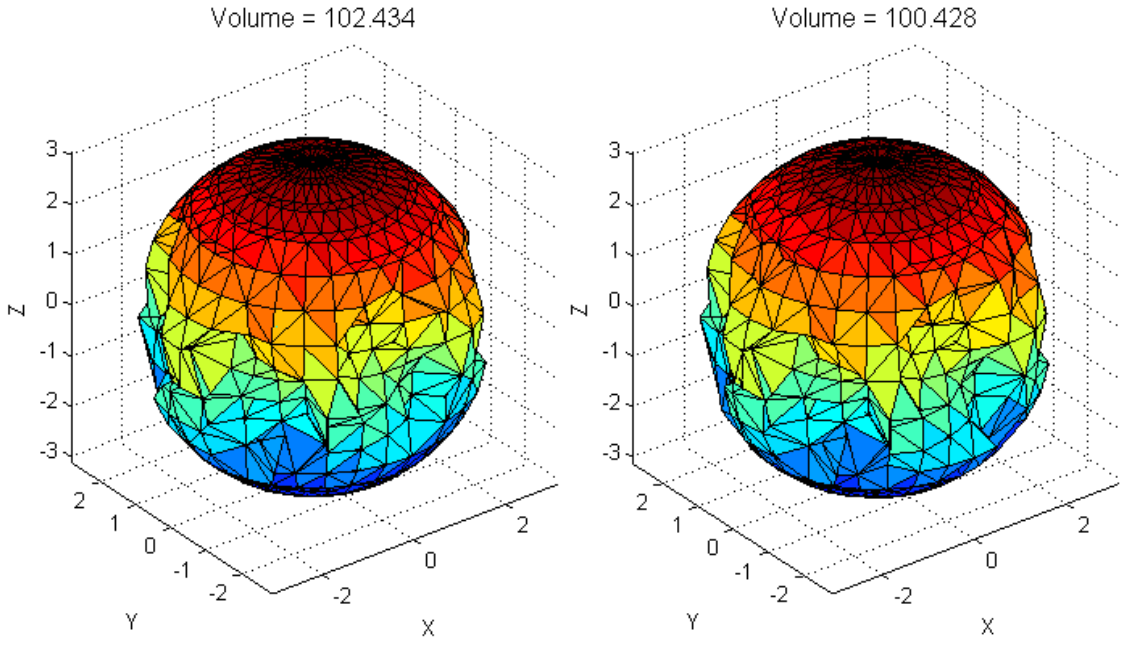

Figure 2-8: Workspace at $\alpha_1 = 90^\circ, \alpha_2 = 60^\circ$

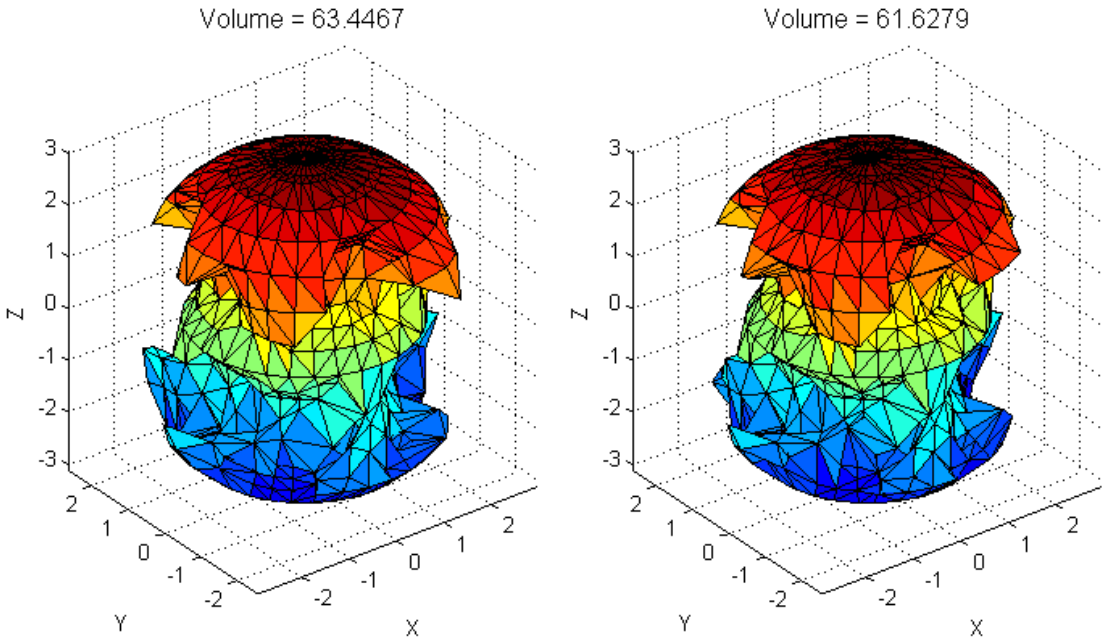

Figure 2-9: Workspace at $\alpha_1 = 90^\circ, \alpha_2 = 45^\circ$

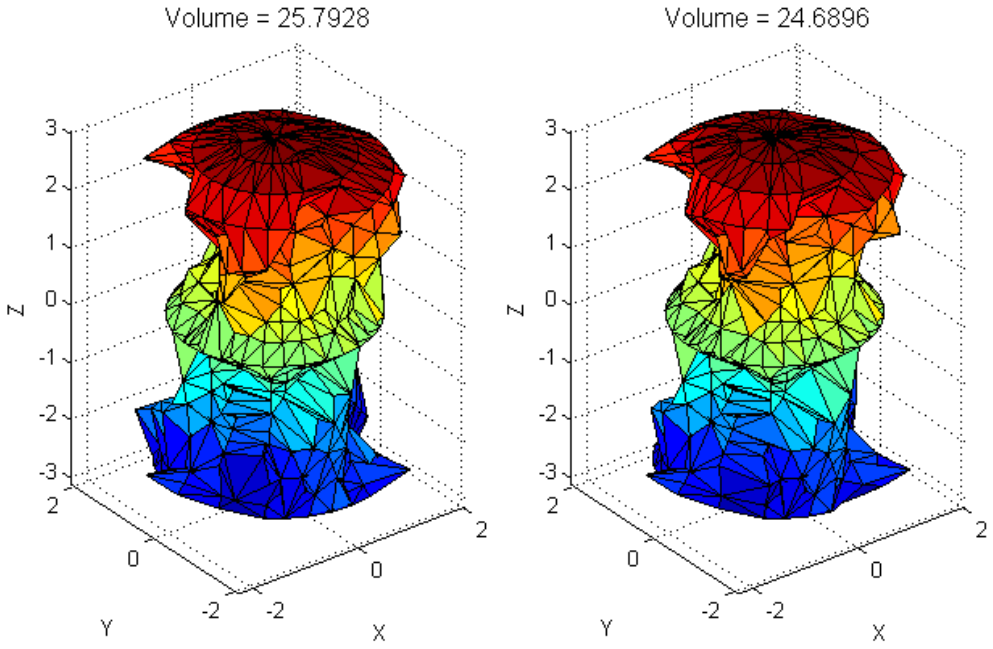

Figure 2-10: Workspace at $\alpha_1 = 90^\circ, \alpha_2 = 30^\circ$

From the above pictures, we can clearly drew the conclusion that the SPM has the largest workspace when α_1 and α_2 are equal to 90 degree, while the workspace

decreased as these two angles change. Also, the workspace declines to some extent when consider all the constraints.

The first set of pictures is when α_1 equals 90 degree and α_2 equals 120 degree. From the picture, it can be found that there are some bumps in the surface of spherical comparing with the second set of picture.

The second set of pictures is the one with both two angles equal 90. According to the literature [14], this configuration has the largest workspace and it can be proved in Figure 2-7.

From the literature [14] it can be found that when *r* equaled to 90, this configuration also is also called as “coplanar”, that is, its three actuators all locate on the same plane. And when *r* is equaled to zero, this configuration is called as “collinear”, that is, its three actuators are located on one axis. An interesting thing about their workspace is that this three cases have pretty the same volume of workspace. In other words, their workspace is only related with their link angle.

3 Kinematics Performance Analysis

3.1 Introduction

The Jacobian matrix has significant meaning for parallel robot. It is not only the fundamental part to calculate many other performance indices, such as stiffness, dexterity and manipulability, it is also important to implement the static analysis. Jacobian matrix generally can be derived by differentiating the inverse kinematic relations or the vector relation in one kinematic chain.

Kinematic performance relates to the movement performance of the parallel manipulator, such as dexterity. Dexterity is especially important in manipulator design stages. It strongly affects the kinematic performance. There are some measurements of dexterity based on the Jacobian matrix such as its determinant, condition number and minimal singular value [59]. For example, Gosselin and Angeles [23] presented a global dexterity index (GDI) based on the condition number of the Jacobian matrix.

The condition number of Jacobian matrix of the manipulator has attracted the attention of some researchers [63]. In fact, the condition number of a matrix is used in numerical analysis to estimate the error generated in the solution of a linear system of equations by the error on the data [64]. When it applied to the Jacobian matrix, the condition number will give a measure of the accuracy of the Cartesian velocity of the end effector and the static load acting on the end effector. Indeed, as

shown by Salisbury and Craig [65], the dexterity of a manipulator can be defined as the condition number of its Jacobian matrix. For instance, Gosselin applied the dexterity based on the condition number of the Jacobian matrix to the optimization of planar and spatial manipulators [25]. The condition number has already been used for the kinematic design of 3-DOF planar parallel manipulators and 3-DOF spherical parallel manipulators [28]. The global conditioning index (GCI) was defined and applied to the optimization of robotic manipulators [23]. The atlases of GCI for 2-DOF planar parallel manipulators are obtained to optimize the manipulators [59].

The manipulability ellipsoid, the most popular index, was at first successfully introduced to evaluate the static performance of a robot manipulator as an index of the relationship between the linear (angular) velocities at each joint and the linear and angular velocity at the end-effecter of the manipulator [48]. There are two kinds of manipulability ellipsoids. The manipulability force ellipsoid is a similar index that evaluates the static torque-force transmission from the joints to the end-effecter [61], while the dynamic manipulability ellipsoid was introduced as a measure of the acceleration capability of the end effector by taking into account the dynamic parameters of the mechanism [62].

Generally, if the parallel mechanism is pure rotation or pure translation, the condition number of Jacobian matrix is preferred. On the other hand, if the motion of the end-effector of the parallel mechanism contains both rotation and translation,

manipulability is usually applied as the evaluation index. This chapter will illustrate the derivation of Jacobian matrix and the dexterity index.

3.2 Jacobian Matrix

Jacobian matrix is a matrix which maps the relation between the speed at actuator and the speed at the moving platform. The Jacobian matrix of the parallel robot is defined as the matrix represents the transformation from Cartesian velocities into the joint rates in the platform. Note that this mapping relation in parallel manipulator is just opposite with that in serial robot.

Generally, the Jacobian matrix of the parallel robot can be derived from the differentiation of inverse kinematics expression. Differentiate both sides of the Eq. (2-6), which leads to,

$$\dot{\boldsymbol{w}}_i \cdot \boldsymbol{v}_i + \boldsymbol{w}_i \cdot \dot{\boldsymbol{v}}_i = 0 \tag{3-1}$$

where,

$$\dot{\boldsymbol{v}}_i = \boldsymbol{\omega} \times \boldsymbol{v}_i$$
$$\dot{\boldsymbol{w}}_i = \dot{\boldsymbol{\theta}}_i \boldsymbol{u}_i \times \boldsymbol{w}_i \quad i = 1, 2, 3$$

The $\boldsymbol{\omega}$ is angular velocity of the moving platform. Eventually, one can get Eq. (3-1) in the form of,

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}\boldsymbol{\omega} \tag{3-2}$$

where $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$, $\boldsymbol{J} = [j_1, j_2, j_3]^T$ with $\boldsymbol{j_i} = \dfrac{w_i \times v_i}{(u_i \times w_i) \cdot v_i}$.

3.3 Stiffness Evaluation

Stiffness stands for the rigidity of the objective. The modelling of stiffness is important because it can evaluate the ability of resisting deformation in response to an applied force. Lower stiffness may lead to deformation and thus reduce the precision of the parallel mechanism. In our application, the parallel manipulator is used as the orientation device, so the stiffness of the orientation should also be considered and higher orientation stiffness is desired.

The Virtual Joint Method (VJM), also called lumped modelling, is the most popular stiffness modelling method for parallel mechanism at present. Gosselin [44] first employed this method to calculate the parallel manipulator's stiffness considering the stiffness at actuators of each leg. Then Zhang [45] built up the Lumped joint models for planar system and spatial system where the deformation caused by links' flexibility took into account as virtual joints with corresponding torsional springs. The lumped modelling offers acceptable accuracy as well relative short computation time at the numeral analysis stage.

On the other hand, multidimensional lumped-parameter model was proposed by Pashkevich [46] considering translational, rotational and their coupling effect through substituting the link flexibility by localized six degrees-of-freedom virtual springs. Wu [47] adopted Pashkevich's stiffness modelling approach and built up the stiffness of 3RRR spherical parallel manipulator. The displacement of the link then was computed under the Castigliano's theorem. This thesis will adopt the Pashkevich's method since the stiffness is important for this application and the

orientation stiffness also must be considered. Here we only give the final result and the detailed derivation can be found in references [46].

$$\boldsymbol{K} = \sum_{i=1}^{3} \boldsymbol{K}_i \tag{3-3}$$

where $\boldsymbol{K}_i$ is a 6 by 6 matrix and it can be decomposed as:

$$\boldsymbol{K}_i = \begin{bmatrix} \boldsymbol{K}_{rr} & \boldsymbol{K}_{rt} \\ \boldsymbol{K}_{rt}^T & \boldsymbol{K}_{tt} \end{bmatrix} \tag{3-4}$$

where $\boldsymbol{K}_{rr}$ stands for rotation matrix; $\boldsymbol{K}_{tt}$ means translation and $\boldsymbol{K}_{rt}$ is the coupling of the two. One sample of this stiffness in the workspace would like:

$$\boldsymbol{K} = \left[\begin{array}{ccc|ccc} 0.035 & 0 & 0 & 0.361 & 0.362 & 0 \\ 0 & 0.035 & 0 & 0.362 & -0.625 & 0 \\ 0 & 0 & 0.057 & 0 & 0 & 0.398 \\ \hline 0.361 & 0.362 & 0 & 7.568 & 0 & 0 \\ 0.362 & -0.625 & 0 & 0 & 7.568 & 0 \\ 0 & 0 & 0.398 & 0 & 0 & 5.635 \end{array}\right] \cdot 10^5 \tag{3-5}$$

In our application, the mechanism needs to drive the human hand moving around, so both rotational and translational stiffness have a significant impact in terms of the deformation. Thus this requirement necessitates higher stiffness to perform well. Here we consider the diagonal element of the stiffness which stands for three pure rotations and three pure translations. Considering the trace of the stiffness at one specific orientation:

$$Stiff_i = trace(\boldsymbol{K}_i) \tag{3-6}$$

where i stands for one orientation in the whole workspace. In order to get a measurement of the stiffness over the whole workspace, the concept of Global Stiffness Index was implemented here.

$$GSI = \frac{\int_{\Omega} Stiff_i dw}{\int_{\Omega} dw} \tag{3-7}$$

where $Stiff_i$ is the stiffness at one specific orientation. In practice, the GSI of SPM is computed by the discrete approach which can get a compromise between accuracy and computation time.

$$GSI = \frac{1}{n}\sum_{i=1}^{n} Stiff_i \tag{3-8}$$

Thus the first objective function of the optimization problem can be written as:

$$f_1 = \max(GSI) \tag{3-9}$$

3.4 Dexterity Evaluation

Manipulability measures the manipulation ability of the parallel system and has been researched for many years. It can be regarded as a numerical method to evaluate the error of position and orientation of the moving platform. Yoshikawa [48] first proposed the manipulability measure which is given by,

$$M = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)} \tag{3-10}$$

Because the Jacobian matrix is not an invariant value and change as the position and orientation varied. The second problem is that different dimension makes it is hard

to understand the physic meaning of the manipulability. To solve this problem, many methods have been presented. Kim and Khosla [49] made another way to compute the manipulability,

$$M = \sqrt[m]{\det(\boldsymbol{J}\boldsymbol{J}^T)} \tag{3-11}$$

where m is the number of degrees-of-freedom of the robot.

On the other hand, Gosselin suggested using the condition number of the Jacobian matrix to measure the dexterity of the parallel robot. Global Conditioning Index [23] was used to evaluate the manipulability capability on the whole workspace. The disadvantage of this method is that it is time consuming because one has to check every point in the workspace. Another short point is that it is not accuracy because of its work principle. Monte Carlo can greatly improve the accuracy and the error can be decreased to $1/\sqrt{n}$ [24].

$$GCI = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\kappa_i(\boldsymbol{J})} \tag{3-12}$$

where the n is the number of discrete workspace point, each point represents one orientation. $\kappa_i(\boldsymbol{J})$ is the condition number of Jacobian matrix at point i.

In order to avoid the affection of the scaling of the manipulator, Gosselin [23] proposed another index called dexterity indices which is defined as,

$$v = \kappa(\boldsymbol{J}) \tag{3-13}$$

where $\kappa(\boldsymbol{J})$ is the condition number of Jacobian matrix and is defined as,

$$\kappa(\boldsymbol{J}) = \| \boldsymbol{J} \| \cdot \| \boldsymbol{J}^{-1} \| \tag{3-14}$$

With $\|\boldsymbol{J}\| = \sqrt{tr(\boldsymbol{J}W\boldsymbol{J}^T)}$ and $W = 1/n$, where n is the dimension of the matrix $\boldsymbol{J}$. When $\boldsymbol{J}$ is not square matrix, a substitute definition of condition number is defined as,

$$\kappa(\boldsymbol{A}) = \frac{\mu_{\max}}{\mu_{\min}} \tag{3-15}$$

where $\mu_{\max}$ and $\mu_{\min}$ stand for the largest and smallest singular values of $\boldsymbol{A}$.

Similarly, Pham [50] defined a Uniformity of Manipulability over the whole workspace,

$$U = \frac{M_{\min}}{M_{\max}} \tag{3-16}$$

where $M_{\min}$ and $M_{\max}$ stand for the minimum value and the maximum value of the manipulability over the whole workspace.

The Global Conditioning Indices [23] is applied in this research considering our real world application that each rehabilitation movement should have good dexterity performance.

4 Kinematics Performance Optimization

4.1 Introduction

As the development of modern society, people's requirement on the product's quality also increased. To make sure the design satisfies one or more performance indices becomes more and more popular. Traditional optimization method demand the objective function can be differentiated, or has gradient. However, many objective functions in real world application are high non-linear, multi-variable with high constraints. Thanks to the development of artificial intelligent algorithm, such as the Genetic Algorithm, Differential Evolution and Particle Swarm Optimization, many complicated optimization problem can be solved right now. These evolution based stochastic algorithms have been widely used in many fields. This Chapter will first introduce some basic concepts of the Differential Evolution (DE) algorithm. Then the proposed DE will be given and its advantages over Genetic Algorithm will be explained. Finally, the optimization result will be discussed.

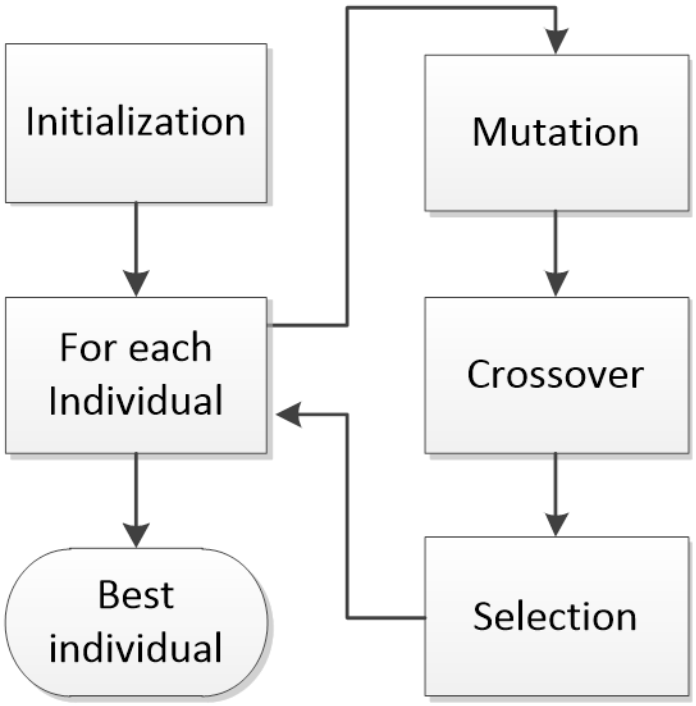

Figure 4-1: Flowchart of classic DE algorithm

4.2 Differential Evolution

DE is a powerful, effective stochastic population-based search algorithm for continuous optimization problem [51]. Some dimensional synthesis works have been done by DE [52], including single objective optimization of stiffness or dexterity and multi-objective optimization combining two or three of them. Here we only discuss the DE algorithm in the applications of multi-objective optimizations.

Many methods and theories have been proposed to deal with the multi-objective optimization problems. There are three most important aspects for designing the multi-objective optimization algorithms, namely: the fitness assignment, diversity

and elitism. In this thesis, the proposed DE is presented to optimize the multi-objective optimization problems.

4.2.1Mutation method

Various methods have been presented during the mutation phase. Just for convenience, the notation ”DE/a/b/c” is used, where ”a” stands for the vector be mutated; ”b” means the number of used difference vector and ”c” is the combine scheme, it can be binomial or exponential [54]. The selected well-known mutation methods are as follows,

DE/rand/1/bin:

$$\boldsymbol{v}_i^{g+1} = \boldsymbol{v}_{r1}^{g} + f(\boldsymbol{v}_{r2}^{g} + \boldsymbol{v}_{r3}^{g}) \tag{4-1}$$

DE/best/1/bin:

$$\boldsymbol{v}_i^{g+1} = \boldsymbol{v}_{best}^{g} + f(\boldsymbol{v}_{r1}^{g} + \boldsymbol{v}_{r2}^{g}) \tag{4-2}$$

DE/rand-to-best/1/bin:

$$\boldsymbol{v}_i^{g+1} = \boldsymbol{v}_{r1}^{g} + f_1(\boldsymbol{v}_{r2}^{g} + \boldsymbol{v}_{r3}^{g}) + f_2(\boldsymbol{v}_{r4}^{g} + \boldsymbol{v}_{r5}^{g}) \tag{4-3}$$

Eq. (4-1) is the standard and also most popular random mutant scheme which leads to global search or exploration. Eq. (4-2) introduces the best individual in to mutant vector which favours local search or exploitation. In this thesis, at early stage, when few ”good solutions” are found during the iteration, Eq. (4-1) will be used to make sure that DE can explore large areas. Later on, Eq. (4-2) will be adapted to exploit

the global optimal point after more “good solutions” are found and stored in the extra archive.

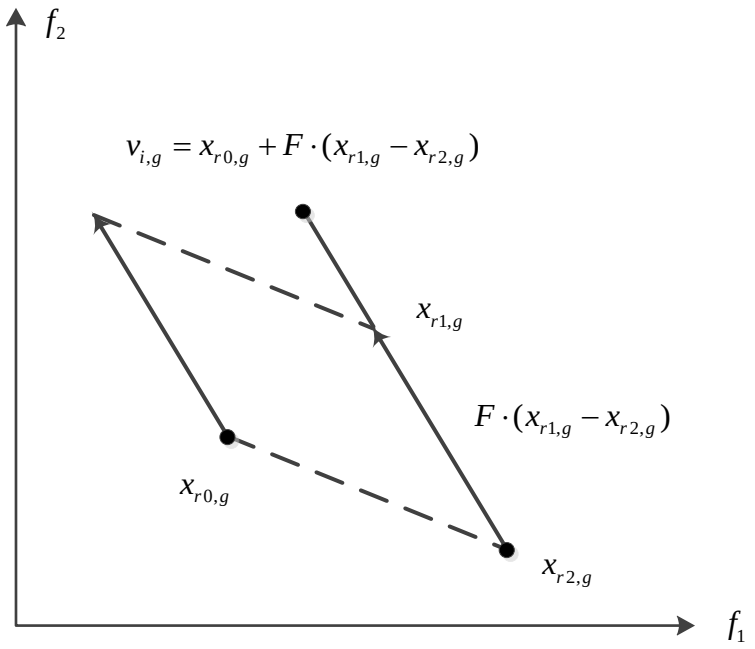

Figure 4-2: The Vector Expression of the Mutation Method

The mathematic meaning of mutation can be expressed as the vector computation, as shown in the Figure 4-2. The vector can be large or small depending on the three randomly selected vectors during the mutation. In this way, the algorithm simulates the natural mutation process.

4.2.2Crossover Method

The crossover is important because it provide the algorithm with infinite possibilities which may lead the algorithm jump out of the local optimal points. Of course, these possibilities can be bad.

$$U_{G,i,j} = \begin{cases} V_{G,i,j} & rand_j(0,1) \le CR \text{ or } j = jrand \\ X_{G,i,j}, & otherwise \end{cases} \qquad (4\text{-}4)$$
$$i = 1,2,\ldots,NP \; j = 1,2,\ldots,D$$

The Eq. (4-4) gives the mathematic expression of the crossover of Proposed DE. The Figure 4-3 shows detailed process of crossover. When the random number smaller than crossover rate, the corresponding element in the U vector will be replaced by the element in the V vector. The U vector also is called as trail vector.

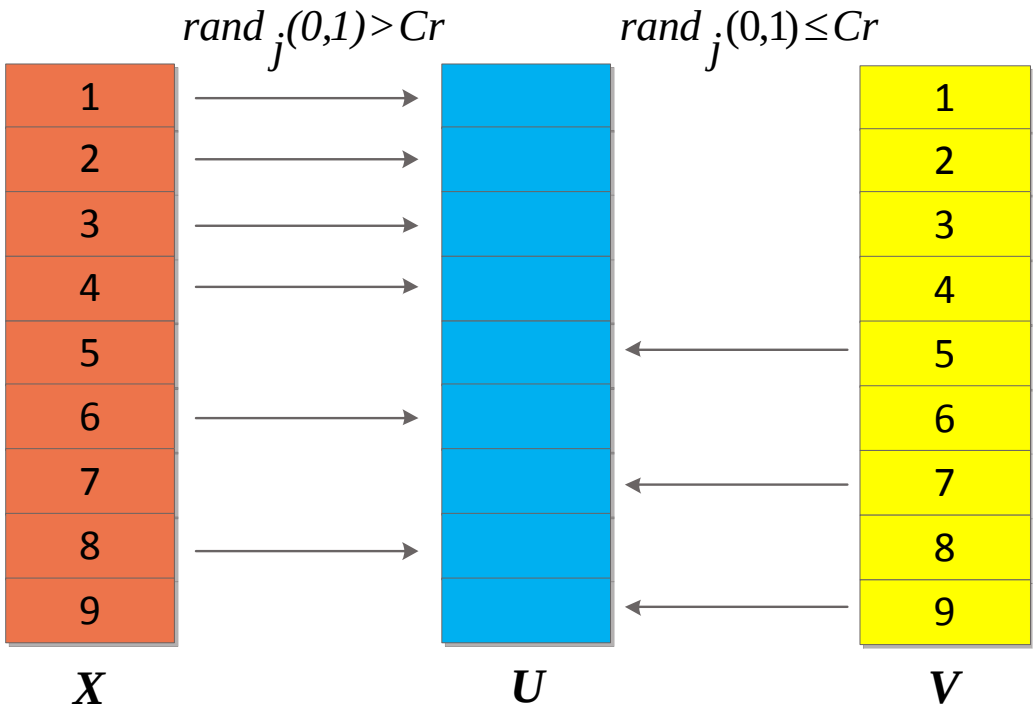

Figure 4-3: The Crossover Method of the Proposed DE

4.2.3Dominance Rank

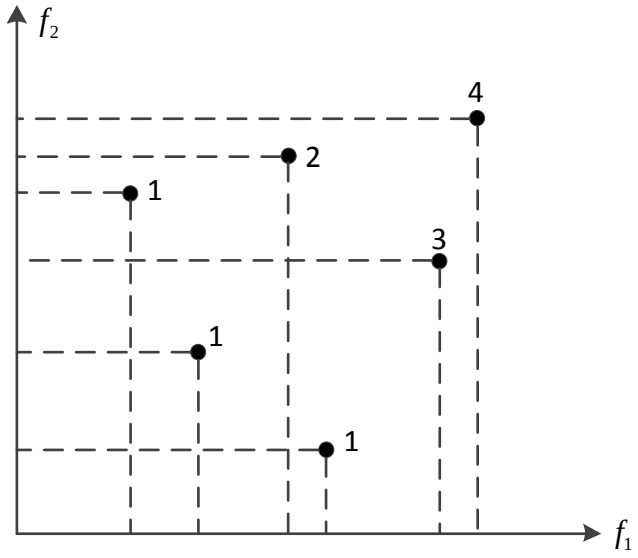

Figure 4-4: Donminance Rank

The dominance concept was proposed initially by Edgeworth in 1881 [55]. The dominance relation deeply shows the performance between two solutions in the multi-objective situation. For example, in Figure 4-4, rank 1 is a set of solutions who dominate all other solutions. After the optimization, rank 1 becomes the Pareto Front. In this thesis, we apply this dominance depth strategy to cope with fitness assignment problem.

4.2.4Diversity method

The loss of diversity had been observed in many P-Metaheuristics [56]. So it is significant to keep diversity in the population. Several methods have been proposed to preserve diversity in the population.

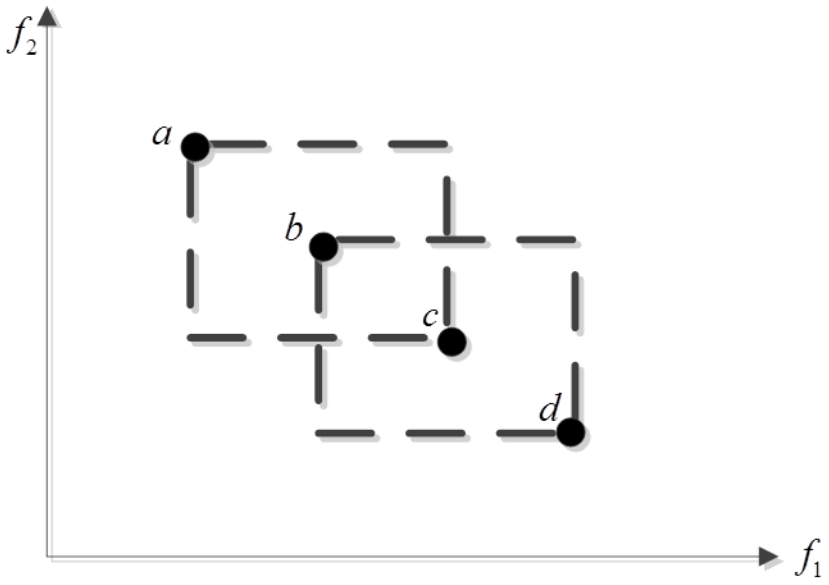

Figure 4-5: Crowding distance ranking

Among them, crowding distance method was selected because it is simple and does not need additional parameters which experience may be required to tune them. The crowding distance is defined as follow, for the solution which is located at the first and the last one in the dominance level rank 1, it has an endless distance, as shown in the Figure 4-5. For the other solutions, it estimates by the distance of its nearest right and left neighbours. It has been proved that crowding distance is highly competitive for matching with other algorithms [57].

4.2.5Elitism

Elitism is a subordinate population compose high-quality solutions and it is very vital for multiobjective algorithms [58].

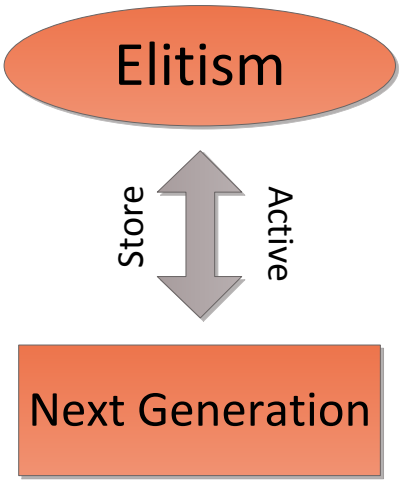

Figure 4-6: The Interaction Between Extra Archive and New Generation

If it takes part in the process of generate new solutions, then it is named as active elitism, otherwise, it is called as passive elitism. Different from other elitisms, proposed method preserves NP number of best elite solutions [60]. These solutions are selected from the rank 1 solution according to the dominance depth principle at each generation. The size always keeps at NP number by the standards of both dominance and crowding distance as the number of elite solutions increase at each generation. This method insures that all NP solutions are always best as the iteration continuous.

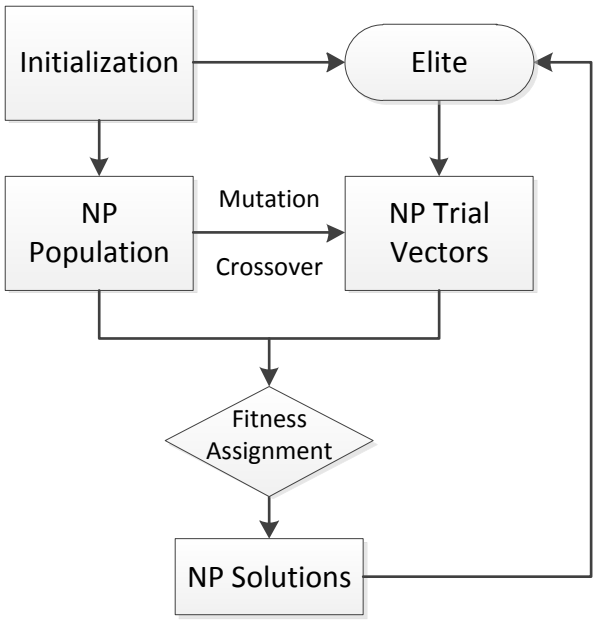

Figure 4-7 Proposed DE procedure

There are two innovative points of the proposed DE. Firstly, instead of generating one trial vector in each generation, this algorithm generates NP number trial vectors once a time (Assuming that NP is the number of population size). Then this NP number of trail vectors and the initialized population is combined together to form 2NP number of populations. After sorting by the dominance and distance principle, the best NP individuals will become the new generation. The second important part is that the rank 1 individual at each generation will be stored as "elite". In order to guarantee that the elites are always in the rank 1 group, once the number of elite is larger than NP, it will be re-evaluated and only the best NP number of individual will be chosen. By this way, the algorithm can always lead the individual head to the global optimal point.

4.3 Optimization Result

In this subsection, some scenarios will be considered with different objective functions.

4.3.1Objective functions

Objective function is the mathematic expression of the optimization goal. In this application, there are many objectives can be considered. For example, workspace of parallel robot is habitually lower than that of the serial robot. So workspace is the regular index of the optimization objective function. Furthermore, larger workspace can contain larger size of the human wrist, thus bigger workspace is preferred. Stiffness means the deformation resistant ability of the structure, it is important since large deformation leads to the decrease of accuracy. Thus higher stiffness is favored. Dexterity reflects the dextrous moving ability of the end-effector as well as accuracy of spherical manipulator in the workspace.

4.3.2Parallel Computation

Because the algorithm has to exhaust all eligible points in the whole workspace, it would take lots of the computation time in the computation of GCI and GSI.

As the development of the computer hardware, computers with multi-core CPU have become very popular. However, most computation software cannot fully take advantage of the multi-core resource in the computer. Matlab is known as good computation software packages in the scientific field. It contains many useful

toolboxes such as Parallel Computation Toolbox, Control System Toolbox, Optimization Toolbox and Neural Network Toolbox.

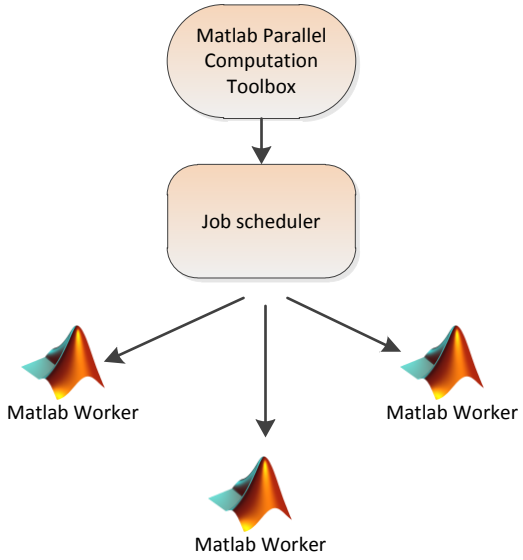

Figure 4-8: The Matlab Parallel Computation Toolbox Work Mode

There are two kinds of parallel computation modes. Both of them are easy to apply. The command "*parfor*" is coming from the "parallel + for". For the "*for*" loop in the original loop, the only job needed to do is to replace "*for*" with "*parfor*". But there are some aspects need to pay attention:

1. *"Parfor"* cannot use nested function except function handle;
2. *"Parfor"* cannot contain other parfor loop;
3. The *"break"* and *"return"* command cannot be used in the "*parfor"* process;
4. Iteration cannot be used in the "*parfor*" loop.

The command "*matlabpool open*" will automatically initialize the parallel computation of Matlab, namely the number of "workers" according to the hardware

of the computer. However, if your computer is dual core, but you set up 4 "works", the software will give out warning.

```
matlabpool local 2

% Initialize the population
for i=1:M
    for j=1:N
        X(i,j)=X_min(j)+rand*(X_max(j)-X_min(j));
    end
end

% Initialize the Gbest
X2=X;
parfor i=1:M
    X22(i,:)=feval(@OptFun,X2(i,:));
end
```

Figure 4-9: The implement of Parallel Computation in the Proposed DE

As shown in the Figure 4-9, 2 works are setup to do the parallel computation work. In the initialization of the globe best solution, "*parfor*" is used to do the parallel computation. Here the function handle was used to recall the evaluation function.

4.3.3Stiffness and the Workspace Optimization

The first set of optimization considered both workspace and stiffness, as Eq. (4-6) shown,

$$\begin{cases} f_1 = \max(WorkSpace) \\ f_2 = \max(GSI) \end{cases} \tag{4-5}$$

For illustration, both Genetic Algorithm and Differential Evolution will be compared. They will run independently with the same parameter setting, such as iteration time and the variables boundary settings. The iteration is set as 1000 times and other parameters are shown in the Table 4-1,

Table 4-1: Boundaries of Parameters

	$\beta/^\circ$	$\gamma/^\circ$	$\alpha_1/^\circ$	$\alpha_2/^\circ$	R_1/m	R_2/m
Minimum	45	45	45	45	0.1	0.1
Maximum	90	90	135	135	0.5	0.5

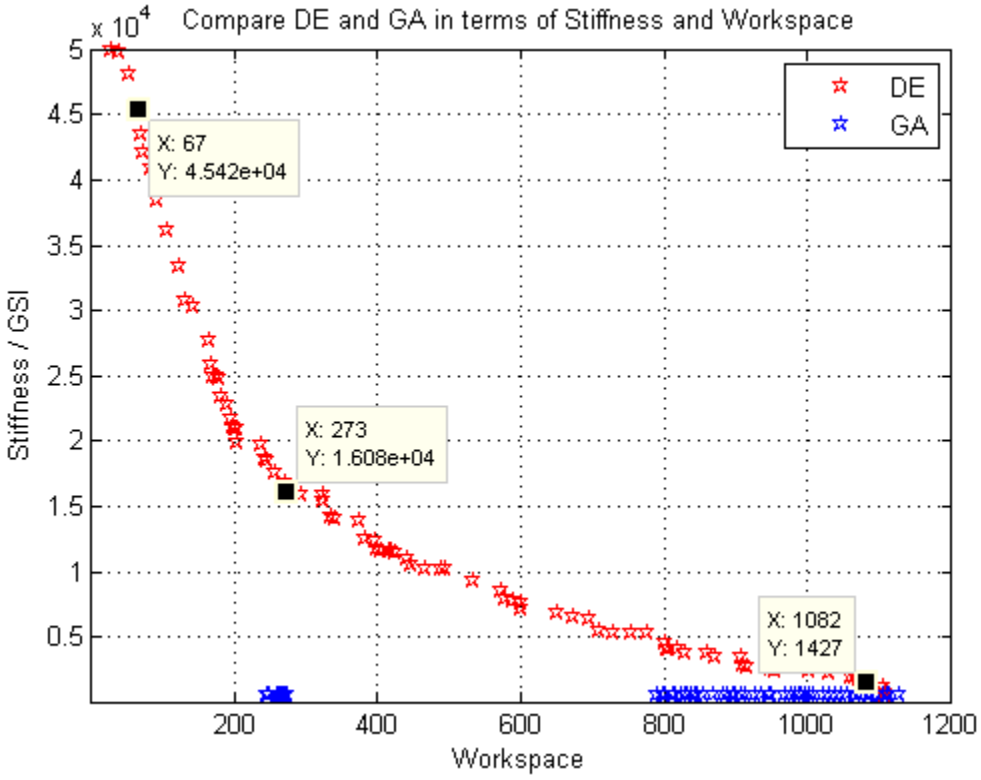

Figure 4-10: Result of Optimization Workspace and Stiffness

The result is shown in the Figure 4-10: Result of Optimization Workspace and Stiffness. From the optimization point of view, the proposed DE outweighs the GA in

both diversity and convergence. The proposed DE offers much more solutions than GA thus have more potential options compare with GA. GA, instead, outnumbers DE in terms of workspace optimization solutions.

Table 4-2: Optimization Result

	$\beta/^\circ$	$\gamma/^\circ$	$\alpha_1/^\circ$	$\alpha_2/^\circ$	R_1/m	R_2/m	Workspace$/rad^3$	GSI
1	4.545	98.132	87.991	76.482	0.016	0.164	67	45419
2	9.231	90.245	103.181	71.213	0.153	0.053	273	16077
3	88.293	78.717	90.882	86.678	0.055	0.354	1082	1427

4.3.4Stiffness and Dexterity Optimization

In this scenario, two objective functions switch to Global Stiffness Index and Global Condition Index, as Eq. (4-7) shows, while the parameter settings still the same as last subsection.

$$\begin{cases} f_1 = \max(GSI) \\ f_2 = \min(GCI) \end{cases} \tag{4-6}$$

Both GA and proposed DE are performed separately once with 1000 iterations. Figure 4-11 shows the optimization result.

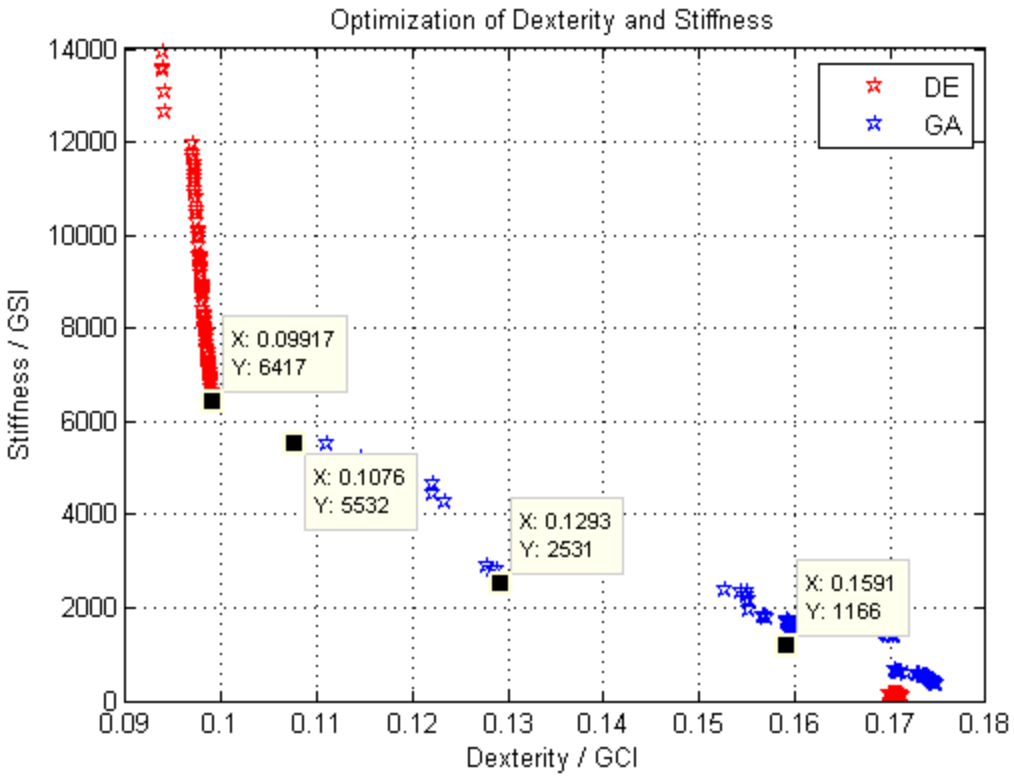

Figure 4-11: Optimization of Dexterity and Stiffness

In terms of stiffness, the proposed DE covers most of the range and the corresponding dexterity value is much smaller than GA, while GA covers most of the dexterity range. However, in terms of Global Condition Index, their result is pretty larger than that of proposed DE. Four typical points are picked out and shown in the Table 4-3.

Table 4-3: Typical Points in the Dexterity and Stiffness Optimization

	$\beta/^\circ$	$\gamma/^\circ$	$\alpha_1/^\circ$	$\alpha_2/^\circ$	R_1/m	R_2/m	GCI	GSI
1	45.000	57.268	45.336	135.000	0.103	0.100	0.099	6416.520
2	45.000	57.302	45.337	135.000	0.117	0.100	0.159	1166.364
3	50.380	62.750	133.500	53.184	0.152	0.129	0.108	5531.790
4	61.356	64.837	118.971	52.675	0.157	0.151	0.129	2530.640

4.3.5Stiffness and Dexterity Optimization at Isotopic Configuration

Here we consider the situation that when α_1, α_2 all equal to 90 degree, so called isotopic configuration. Literature [14] claimed that the SPM possesses the largest workspace when two angles of the link equal to 90°. How about the stiffness and the dexterity? Note that all other parameters are the same as previous optimizations and the result shows in the Figure 4-12.

$$\begin{cases} f_1 = \max(GSI) \\ f_2 = \min(GCI) \end{cases} \quad \text{with } \alpha_1, \alpha_2 = 90^\circ \tag{4-7}$$

Table 4-4: The Parameters of the Stiffness and Dexterity Optimization

	$\beta/^\circ$	$\gamma/^\circ$	$\alpha_1/^\circ$	$\alpha_2/^\circ$	R_1/m	R_2/m
Minimum	45°	45°	90°	90°	0.1	0.1
Maximum	90°	90°	90°	90°	0.5	0.5

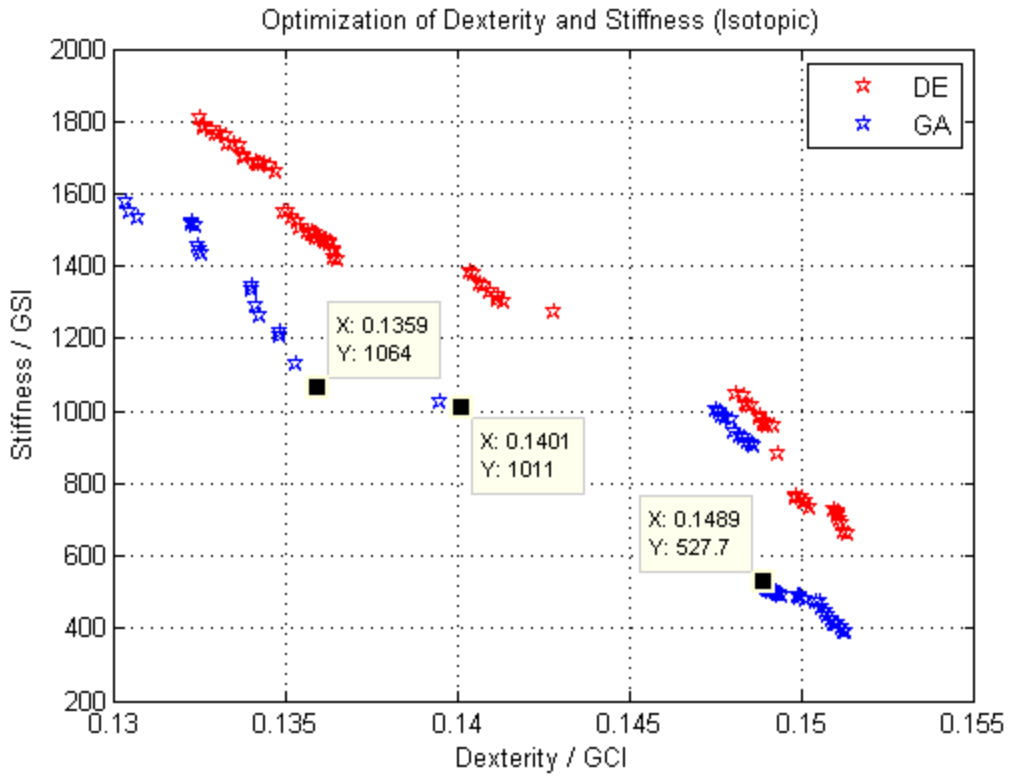

Figure 4-12: Optimization of Stiffness and Dexterity when $\alpha_1, \alpha_2 = 90^\circ$

Compared with the result in last section, the significant change in Figure 4-12 is the huge drop of GSI (stiffness) and little increase of the GCI (dexterity). In the previous subsection, one set of the solutions is (0.099, 6416), but in Fig. 20, the best GSI value is only 1064.

Table 4-5: Typical Points in the Dexterity and Stiffness Optimization (Isotopic)

	β / °	γ / °	R_1 / m	R_2 / m	GCI	GSI
1	56.890	73.571	0.101	0.122	0.136	1063.900
2	56.662	79.403	0.101	0.102	0.140	1010.795
3	56.712	79.743	0.104	0.106	0.149	527.654

4.3.6Stiffness, Dexterity and Workspace Optimization

Finally, the optimization of three objectives optimization is conducted. Generally for optimization, three objectives optimization is pretty difficult for optimization algorithms due to its complexity. For the three most important factors, stiffness, dexterity and workspace are chosen as the objectives, as shown in Eq. (4-9),

$$\begin{cases} f_1 = \min(GCI) \\ f_2 = \max(GSI) \\ f_3 = \max(Workspace) \end{cases} \tag{4-8}$$

All parameters are set as the same for both proposed DE and GA algorithms, as it is shown in the Table 4-1. The optimization result is shown in the Figure 4-13.

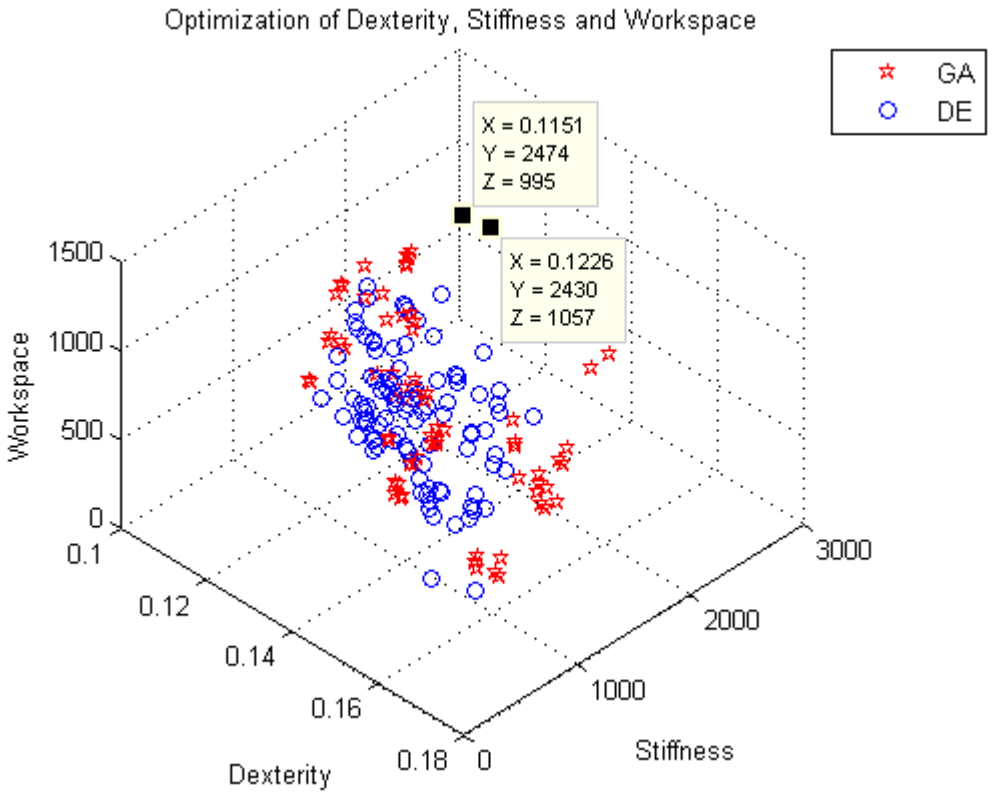

Figure 4-13: Optimization of Stiffness, Dexterity and Workspace

Because it is three objectives optimization, so the result is visualized in three dimension picture. Since strong stiffness, smaller dexterity and larger working volume is preferred, so the relative good solution is located at the right up corner of the cube, as two optimal solutions has been tagged.

As shown in the Table 4-6, because the workspace of the SPM is targeted to be minimized, so the two angles reaches to 90 degree, and the stiffness is decreased to some extent. However, the dexterity also does not change too much.

Table 4-6: Typical Points of the Optimization result

	$\beta/^\circ$	$\gamma/^\circ$	$\alpha_1/^\circ$	$\alpha_2/^\circ$	R_1/m	R_2/m	GCI	GSI	Workspace
1	90.000	83.041	89.559	89.709	0.500	0.100	0.115	2473.722	995
2	84.747	80.213	89.446	91.298	0.184	0.115	0.123	2429.778	1057

After these optimization works, some promising solution were obtained. Some decision must be made between them. Here we consider the stiffness is the first priority among all those indexes because lower stiffness may lead to inaccuracy of the rotation center and side effect may be generated as a result. Secondly, on the basis of first point, the design prefers the larger workspace. Finally, the point 1 in the Table 4-3 was chosen and implement in Chapter 5.

5 Structure Design and Analysis

This human wrist rehabilitation manipulator takes full advantage of the 3 pure rotation degrees of SPM. In the aforementioned coordinate, the 3 DOFs movement is the three therapy motions: flexion-extension, abduction-adduction and supination-pronation.

5.1 Adjustable Hand holding device design

Since different people own different sizes of hand, an adjustable device becomes necessary. Another reason for designing this device is that the rehabilitation movement, namely the movement should rotate around the wrist joint, which is the centre point of the SPM.

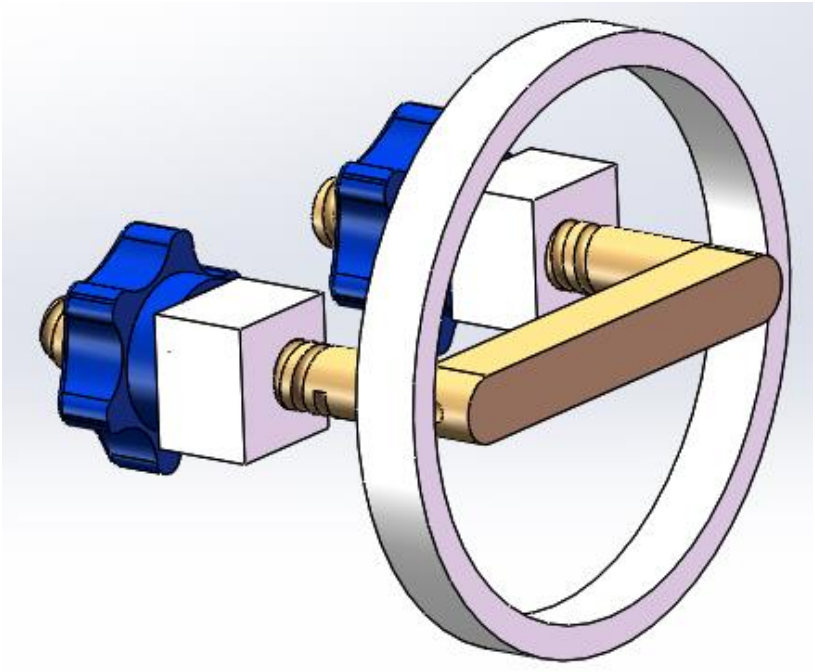

Figure 5-1: Hand Hold Device

This holding device has two main parts, as shown in Figure 5-1. The first part is the rod where the patient's hand will grasp on it. The rod on a sliding rail changes the position of the hand. The start knob used to fix the position so that it will not move during the rehabilitation movement.

5.2 Forearm fix device design

This rehabilitation offers three degrees-of-freedom: supination-pronation, flexion-extension and abduction-adduction. Among them, the supination-pronation movement realize by this device.

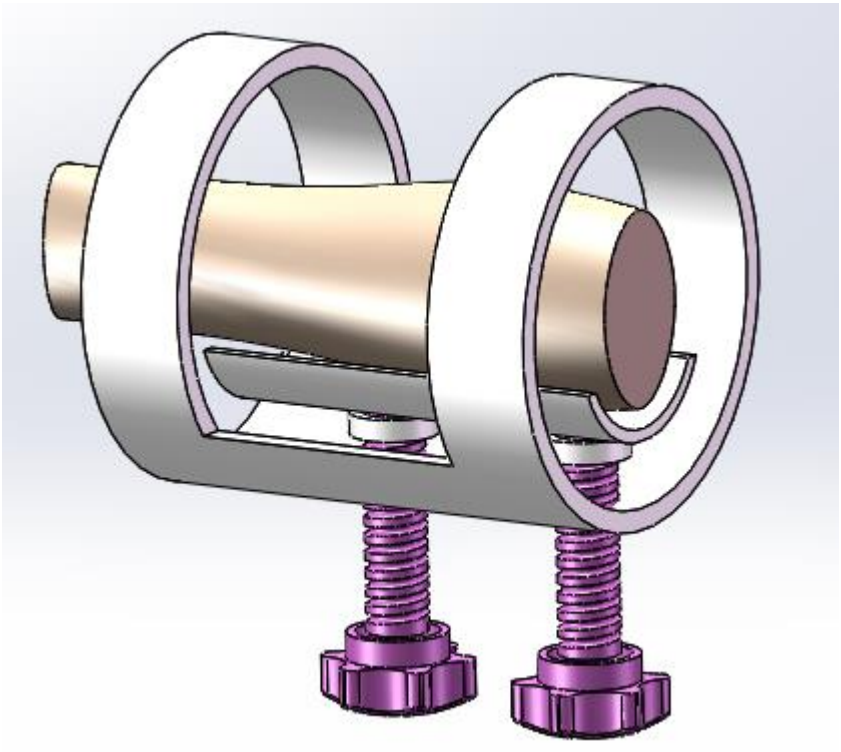

Figure 5-2: Forearm support device

It is actually rotated around the axis of the forearm. Thus there must be some device to guide its rotation as well as fixing the rear part of the forearm. As shown in Figure 5-2, the holder here is used to fix the forearm.

5.3 Anti-collision Design

As aforementioned, there are two kinds of link collisions. Chapter 2 already illustrated the first kind of collision. This subsection will discuss the last one that the collision between one's distal and another's proximal.

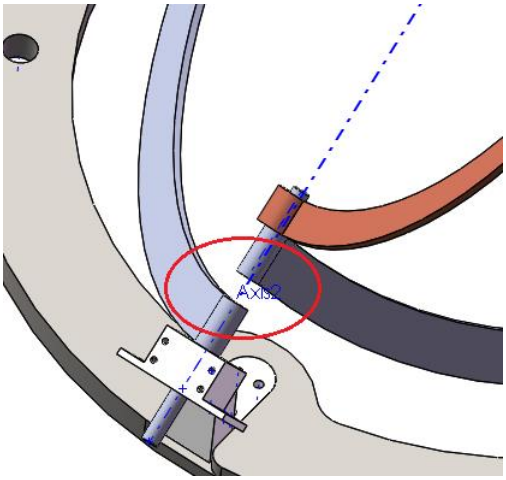

Figure 5-3: Second Kind of Link Collision and the Link Design

Since each link at each level has the same radio, there are some opportunities that two adjacent link would collision together. The second kind of collision is shown in the Figure 5-3.

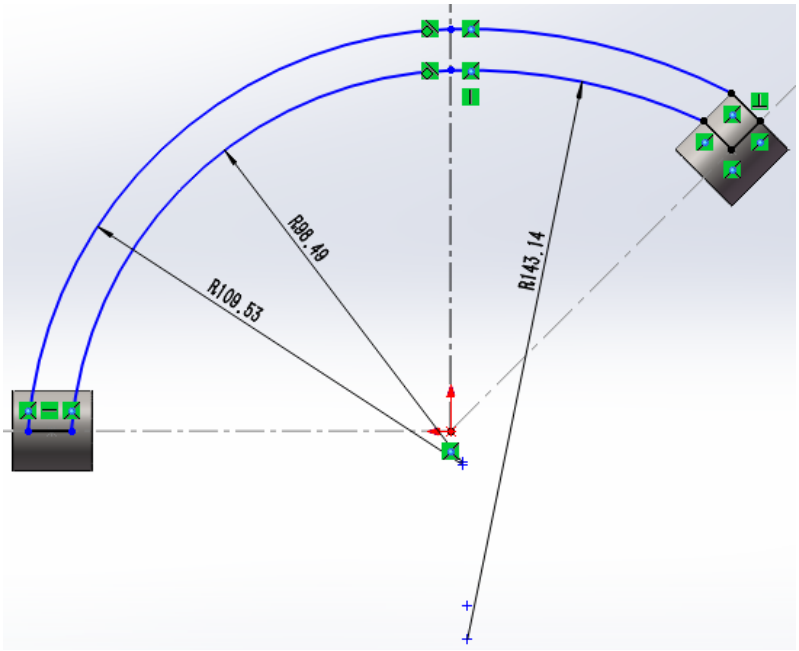

Figure 5-4: The vary-radius design of the link

This kind of collision can be solved by the special designed, non-uniform link, as shown in Figure 5-4. It is featured by different radii on the right-side and left-side of the link. The left-side is the nominal radius which is one of our design variables. The right-side radius is different from left-side to avoid the collision. By this way, the collision can be prevented when two contiguous distal and proximal link meet together.

5.4 3D CAD model of the Rehabilitation robot

After all the parts design, the whole robot is assembled in the SolidWorks, as shown in Figure 5-5.

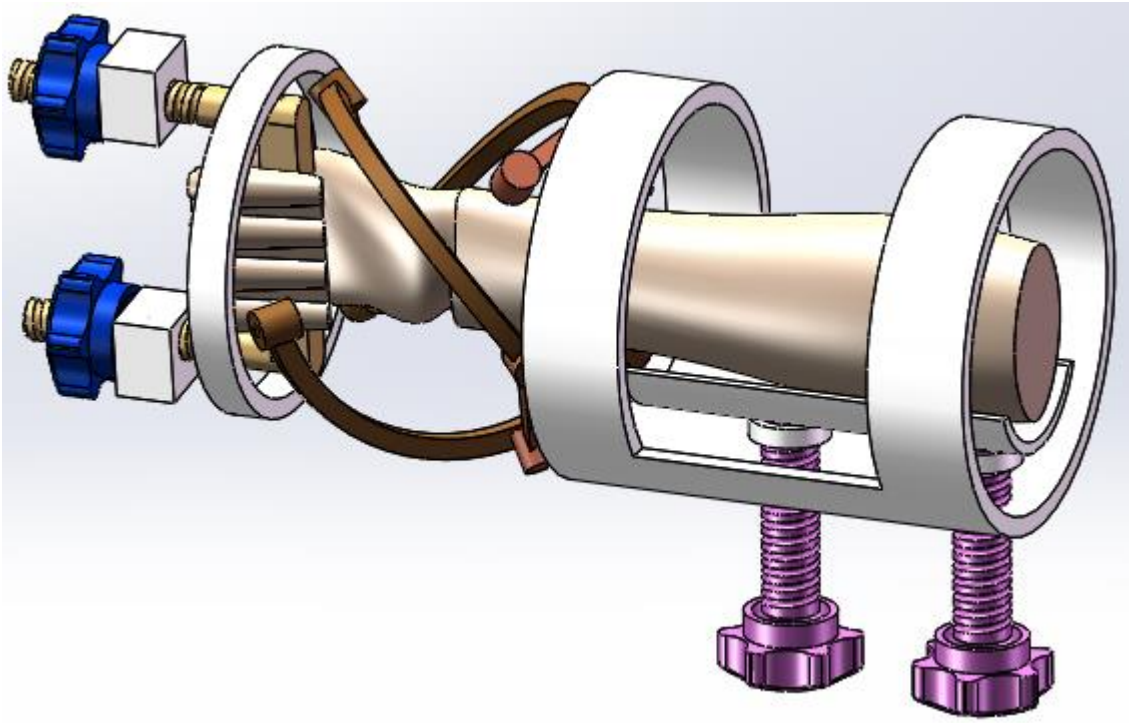

Figure 5-5: The Assemble of Wrist Rehabitation Robot in Solidworks

The entire robot includes three parts: the base, the SPM part and the handle part. The base part supports the patient's forearms and fixes it on the supporter. The supporter can be adjusted to the position that allows different people's forearms can fit in it. The SPM part is the core part of the whole system. It offers rehabilitation movement to the patient's wrists. Its design and detailed analysis will be illustrated in the following chapters. The handle part has the same function as the base part that it fixes the hand and adjusts the distant between the hand centers to the rotation center.

5.5 Static Analysis

Static analysis was conducted in the SolidWorks Simulation software. Static analysis is important because it can give designers the detailed information about the stress and strain of the structure which helps engineers to revise the virtual model in the future.

SolidWorks Simulation offers a humanization interface for the customers with précised results. First, the Simulation needs to be activated in the SolidWorks add-in console. After assembling of all models, click the Simulation tab and switch to Simulation interface. Click the new study to build an analysis example and choose the static analysis.

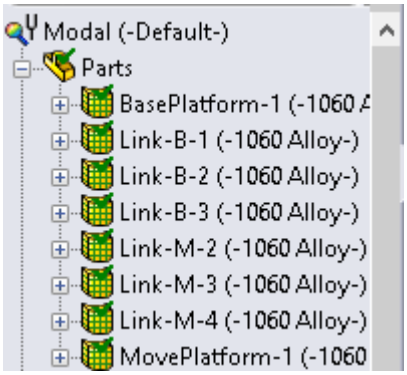

Figure 5-6: Apply material to the parts

The first work needs to do is to apply material to all the parts. One can add different parts with different materials or just apply one material to the whole assembly. Here the 1060 Alloy was adapted to all the parts because it is economical and easy to fabricate.

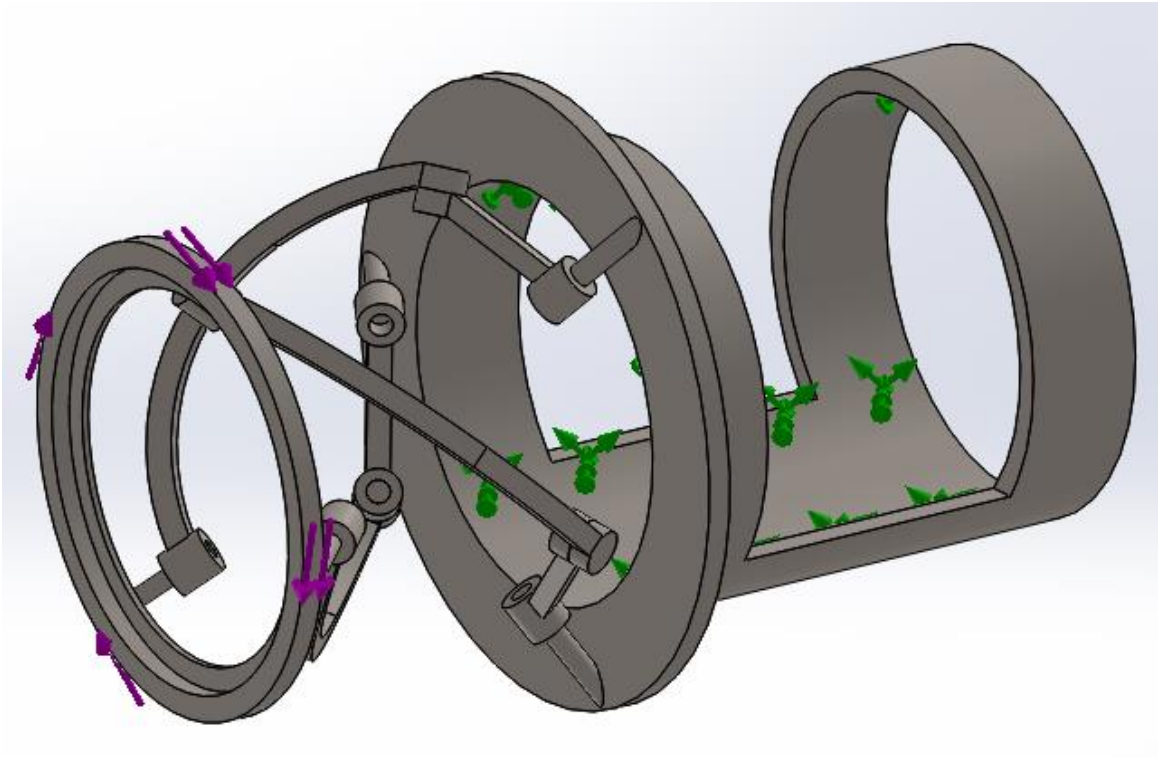

Figure 5-7: The load and the Constraints

For the loads and constraints, a toque of 100 N.m was applied at the moving platform to test the deformation, stress and strain. The forearm holding part was fixed to simulate the real world situation. After all these setting, the modeling is shown in the Figure 5-7. The green arrows mean it was fixed in six degree of freedom. The purple arrows means the torque was applied.

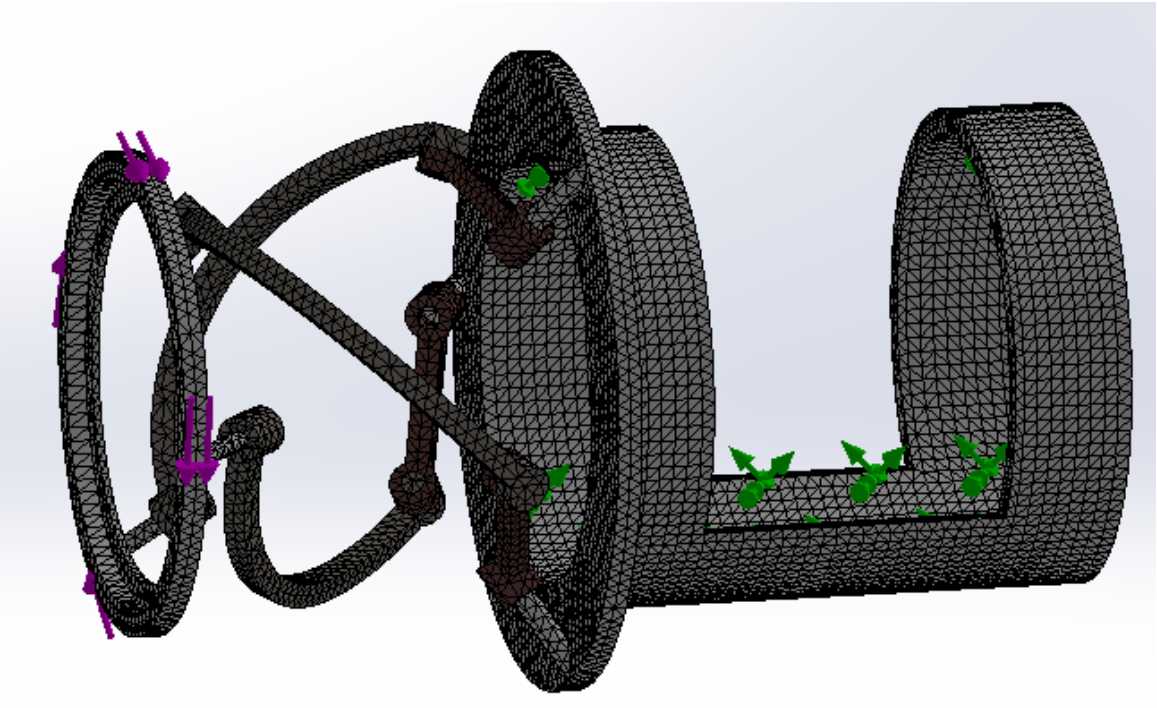

Figure 5-8: Mesh the assembly

Mesh is means separating the whole part into finite number of small parts. It is important because the quality of mesh decide the precision of the result. Fortunately, SolidWorks Simulation offers smart mesh tools that can mesh different part by different method and the whole process is been done automatically. Here the highest mesh quality was applied to the assembly. The mesh result was shown in Figure 5-8.

The static analysis results are shown in three parts, namely, stress, displacement and strain.

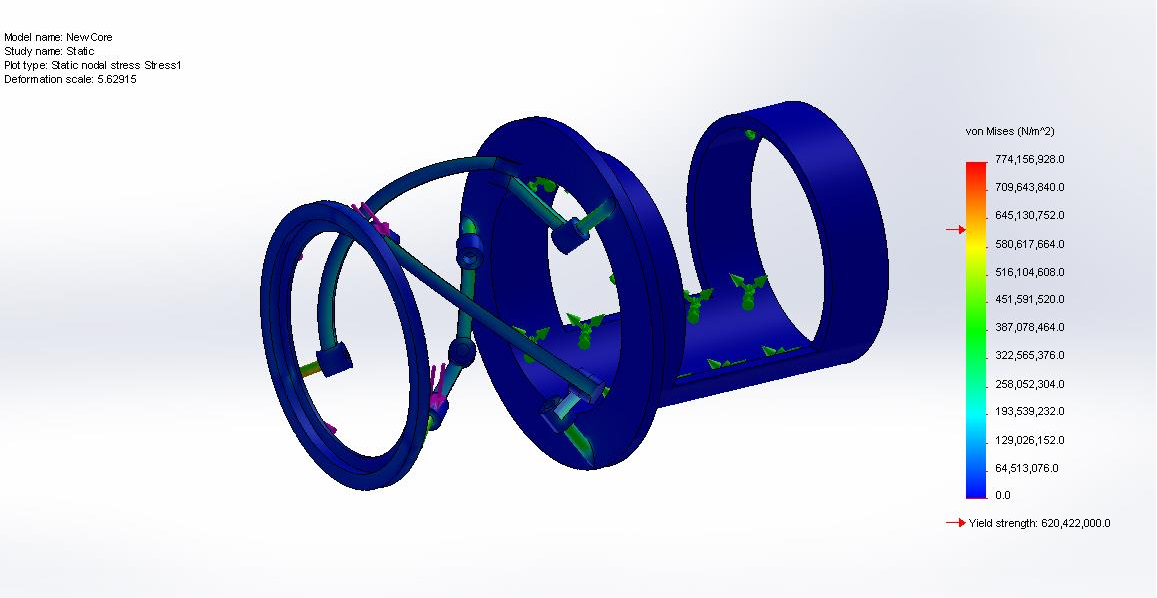

Figure 5-9: The Stress Distribution

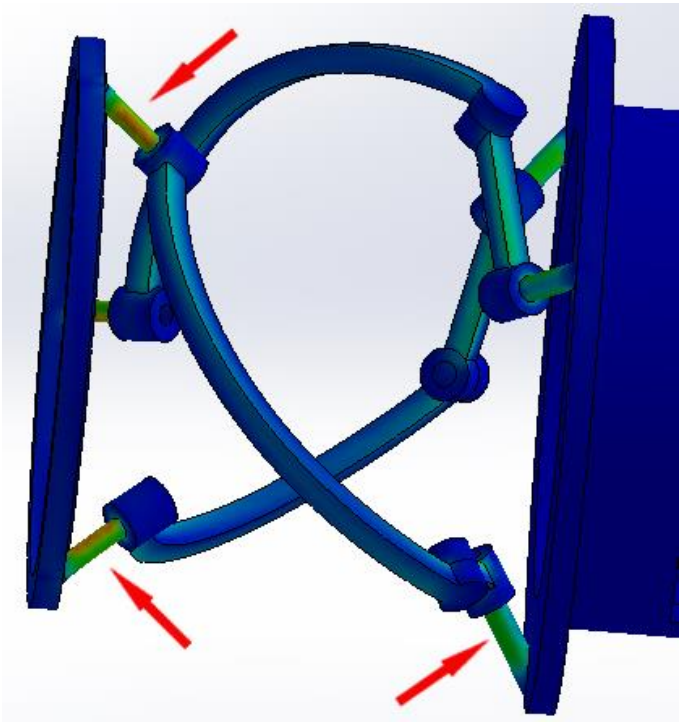

Figure 5-10: The stress concentration

From the Figure 5-10 one can know that the high stress happens at the axes of moving and the base platform.

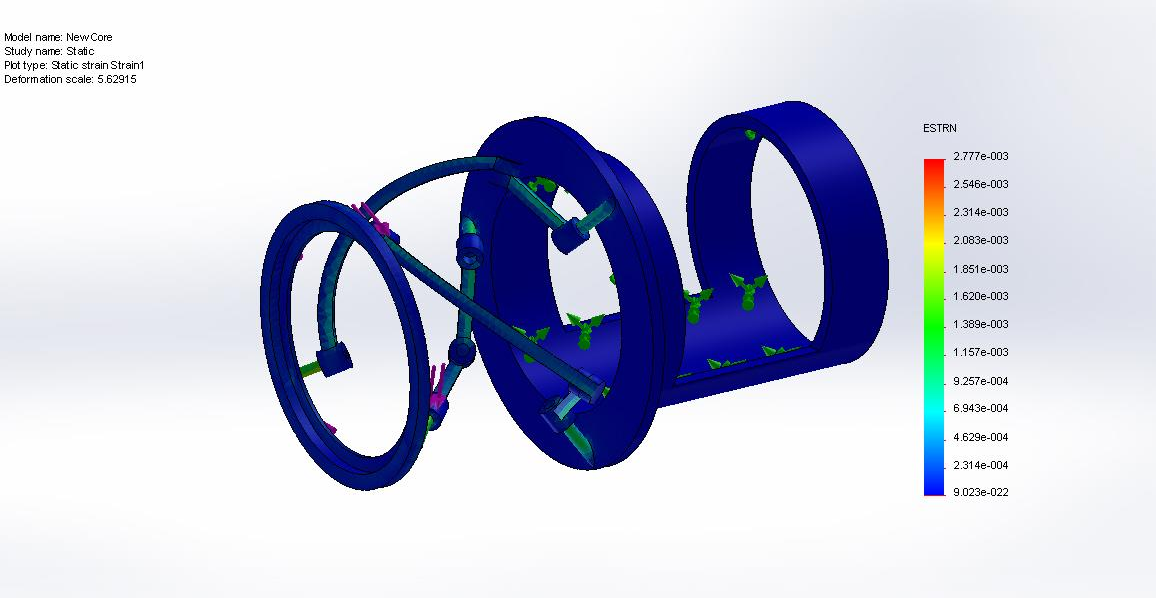

Figure 5-11: The Strain Result

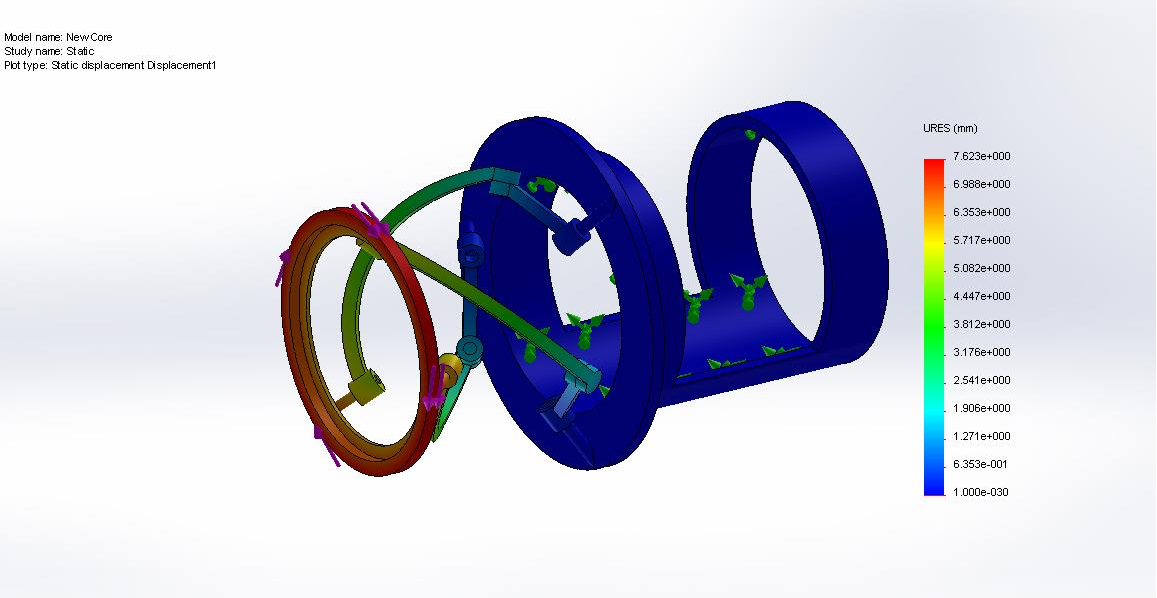

Figure 5-12: The Displacement

5.6 Modal Analysis

Modal analysis studies the frequency of the structure in the nature situation or under payloads. It calls natural modal if there is no load was applied. Otherwise it is

called work modal. Modal analysis is important because it can predict the nature frequency of the structure which can help designer to avoid their modal frequencies in the real work application.

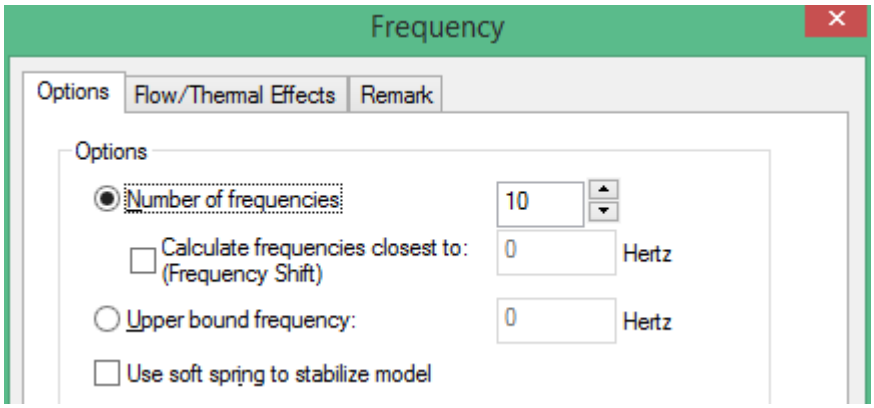

Figure 5-13: Extract 10 frequencies

The steps of modal analysis are similar with static analysis in mesh and apply material, load and constraints. Here 10 modal frequencies are set to be extracted.

The 9 modal shapes are shown from Figure 5-14 to Figure 5-16.

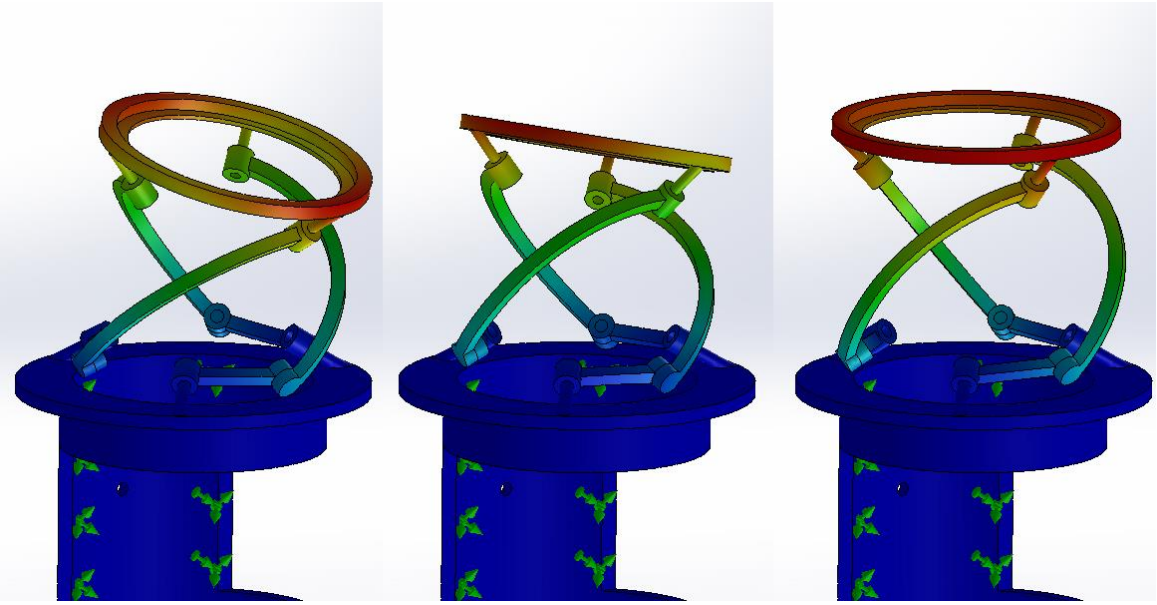

Figure 5-14: The Model Shape from 1-3 Order Modal Frequencies

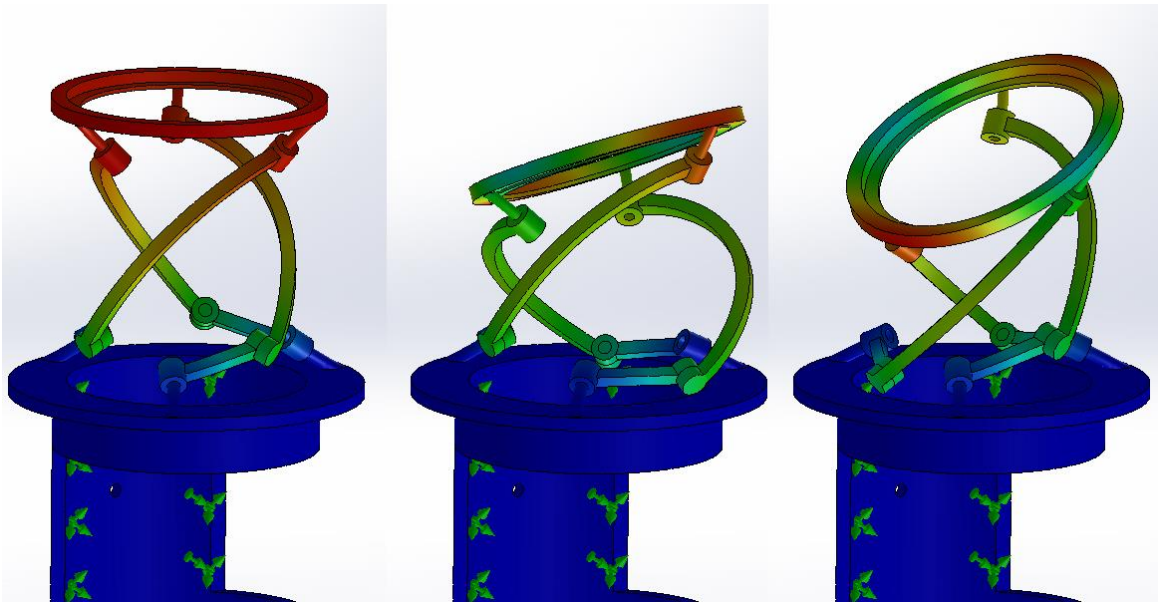

Figure 5-15: The Modal Shapes from 4-6 Order Modal Frequencies

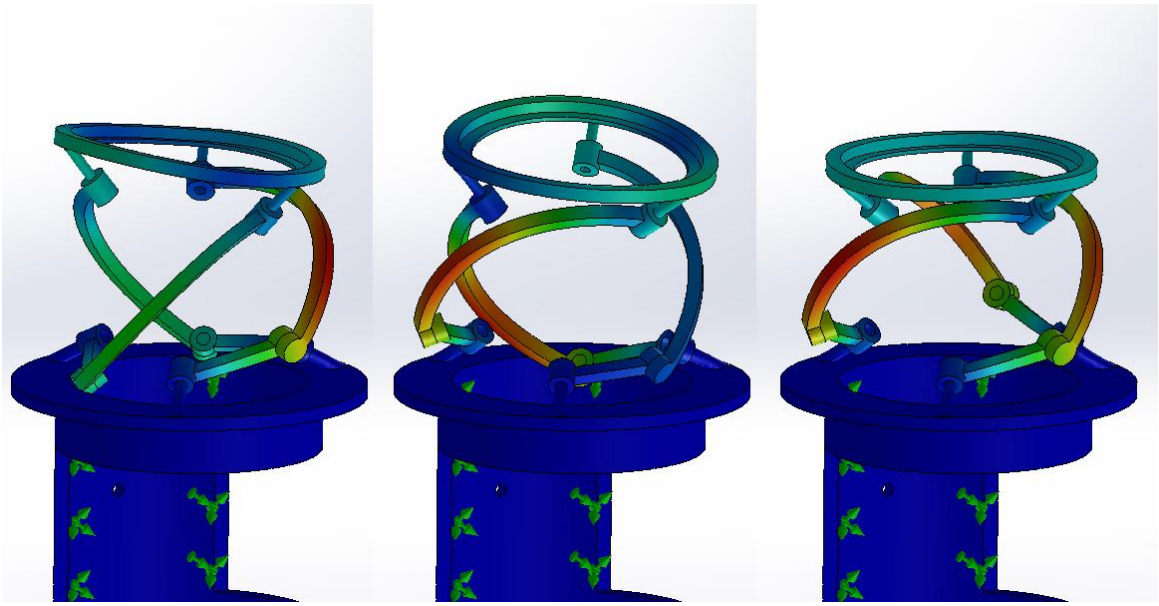

Figure 5-16: The Modal Shapes from 7-9 Order Modal Frequencies

6 Results and Discussion

6.1 Result

6.1.1 Workspace

The workspace analysis was conducted in the Chapter 2. The results verified Gosselin's conclusion that the largest workspace can be get when two angles equal to 90 degrees. The volume of workspace varies as the two angles change.

6.1.2 Kinematics Optimization

The kinematics optimizations were performed in the Chapter 4. Proposed DE algorithm was first introduced and proved efficient even faster than Genetic Algorithm sometimes. Four separate optimization scenarios were done under the Proposed DE and GA. At last, consider the stiffness, workspace and dexterity, one result was taken as the design configuration and implemented into 3D model.

6.1.3 Static and Modal analysis

Modal analysis was conduct in the Chapter 5. Static analysis reveals the stress concentration location which can help designer to improve the virtual 3D model in the future. Modal analysis gives modal frequencies of the structure. This is extremely important which can assistance engineer to escape their resonance frequencies. The Table 6-1 shows 10 order modal frequencies in details and Table 6-2 is the mass participant of the structure.

Table 6-1: 10 Order of Modal Frequencies

Mode No.	Frequency(Hertz)	Period(Seconds)
1	45.661	0.021901
2	45.671	0.021896
3	57.601	0.017361
4	101.8	0.0098235
5	146.37	0.006832
6	146.4	0.0068305
7	215.03	0.0046505
8	215.15	0.004648
9	217.53	0.004597
10	586.09	0.0017062

Table 6-2: The Mass Participations

Mode No.	Freq (Hertz)	X direction	Y direction	Z direction
1	45.661	0.0052562	0.036412	0.049144
2	45.671	0.062648	0.025166	0.0029964
3	57.601	0.0077934	0.0098999	0.013105
4	101.8	0.027048	0.034587	0.045703
5	146.37	0.0058093	0.0082363	0.018973
6	146.4	0.018881	0.014129	5.10E-06
7	215.03	0.0045955	0.00095047	0.00091309
8	215.15	8.02E-06	0.0035287	0.0029221
9	217.53	0.0024598	0.002755	0.00364
10	586.09	0.00049075	0.00040634	2.15E-07

6.2 Contribution

This thesis aims to design a rehabilitation robot based on the Spherical Parallel Mechanism. The objective of this thesis is to take full advantage of the pure rotation of 3-RRR SPM to work for human wrist rehabilitation. The contributions are as follows.

1. According to the characteristics of 3RRR SPM, this thesis present 3 degrees-of-freedom wrist rehabilitation robot with three pure rotations. Compared with previous literature review, 3 degrees-of-freedom has more advantages than 2 degrees-of-freedom robots and it is more compact and reliable than other 3 degrees-of-freedom rehabilitation robots.

2. Based on the spherical parallel mechanism 3RRR, all links in a totally different way is assembled to produce a novel revised 3RRR SPM. It means that they share the same number of links but not the same mechanism.

3. Given the specific application in the human wrist re-habitation, new coordinate was set up in order to describe the movement of platform easily and plan the trajectory in the future.

4. In order to adapt the different size of human wrist, an adjustable devise is developed for this wrist recovery mechanism. It is easy to use and economical, which differentiates this work from others.

5. Considering the mechanism itself and the application, three different objectives are chosen as the optimization goal: dexterity, workspace and manipulability. Proposed Differential Evolution and Genetic Algorithm are introduced as optimization algorithms. Two algorithms and its results are compared.

7 Conclusions and Recommendations

7.1 Conclusions

This thesis designs and analyzes a wrist rehabilitation robot based on 3RRR spherical parallel mechanism. The mathematic model was built in the T&T coordinate system. Inverse kinematic was performed and workspace atlases were obtained. Then the kinematic optimization works were conducted by proposed Differential Evolution algorithm. Static and modal analyses were completed in the SolidWorks Simulation software. 10 order modal frequencies were acquired.

Based on previous work, this thesis archives all the objectives in the Chapter 1. It is reasonable to design a three pure rotation degree-of-freedom parallel device that helps patients to regain the motor ability after the Stroke. At the same time, the designed structure is reliable, convenient, and state-of-the-art. The advancement of this wrist rehabilitation robot can contribute to the development of after-stroke rehabilitation science and device in the future.

7.2 Recommendations

The above work has illustrated the kinematic analysis in detail. However, there are some other future work will include but not limit as follows.

7.2.1 Dynamic Analysis

Dynamics are concentrated on the force and its effect on the objects. Dynamics help the control system to be more precisely. Dynamic analysis can determine the power of DC motor needed to drive the human's wrist. Dynamics analysis would include force, velocity and acceleration analysis. Newton-Euler and Lagrange equation is the popular analysis method.

7.2.2 Control System

After the dynamic analysis, the control system can be obtained immediately. The control system would like to contain following parts, namely, the microcontroller, the actuator and feedback system. The microcontroller will transfer the rehabilitation movement into control signal and send it to the motor. The actuator drives the link move and the feedback system give their real position back to the microcontroller in order to be more precisely. PID and other control law can be implemented.

7.2.3 Prototype and Test

Then the prototype can be made base on the aforementioned future work. Although lots of theoretic analysis about this rehabilitation robot has been done throughout this thesis, the prototype is necessary to validate its correctness and find out potential problems. After that, real wrist test can go into action under full protection. The feedbacks from the patients can improve the development of wrist rehabilitation robot in the future.

8 References

[1] Heart Disease and Stroke Statistics—2013 Update [Online] American Heart Association, Inc. http://circ.ahajournals.org/content/127/1/e6.full

[2] Two Million People Suffer Stroke Every Year in China [Online] Sohu.com http://health.sohu.com/20120816/n350816585.shtml

[3] STATISTICS [Online] Heart and Stroke Foundation of Canada http://www.heartandstroke.com/site/c.ikIQLcMWJtE/b.3483991/k.34A8/Statistics.htm

[4] Neuroplasticity [Online] The Wikimedia Foundation, Inc. http://en.wikipedia.org/wiki/Neuroplasticity

[5] Occupational Therapy [Online] Atlantic Health System http://www.atlantichealth.org/overlook/our+services/physical+rehabilitation/overlook+medical+center/occupational+therapy

[6] Stroke New Treatment Breakthrough New Hope for Survivors [Online] Guardian Liberty Voice http://guardianlv.com/2013/06/stroke-new-treatment-breakthrough-new-hope-for-survivors/

[7] B. Siciliano, O. Khatib. *Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2008.

[8] V.E. Gough. *Contribution to discussion of papers on research in automobile stability, control and tyre performance*. Proceedings Automobile Division Institution Mechanical Engineering, pages 407-416, 1956/57.

[9] J.P. Merlet. *Parallel Robots*. Springer, 2006.

[10] J.E. Gwinnett. *Amusement Device*. Patent 1,789,680. 20 January, 1931.

[11] C.M. Gosselin, J.F. Hamel. The agile eye: a high-performance three-degree-of-freedom camera-orienting device. *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on* ,1:781-786, 8-13 May 1994.

[12] H. Asada, J.A C. Granito. Kinematic and static characterization of wrist joints and their optimal design. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on* , 2:244-250, Mar. 1985.

[13] C.M. Gosselin. *Kinematic Analysis, Optimization and Programming of Parallel Robotic Manipulators*. McGill University, 1988.

[14] C.M. Gosselin, J. Angeles. The optimum kinematic design of spherical three-degree-of-freedom parallel manipulator. *Journal of Mechanical Design*, 111(2):202-207, 1989.

[15] C.M. Gosselin, E. Lavoie. On the kinematic design of spherical three-degree-of-freedom parallel manipulators. *The International Journal of Robotics Research*, 12(4):394-402, 1993.

[16] C.M. Gosselin, J. Sefrioui and M.J. Richard. On the direct kinematics of spherical three-degree-of-freedom parallel manipulators with a coplanar Platform. *Journal of Mechanical Design*, 116(2):587-593, 1994.

[17] C.M. Gosselin, J. Sefrioui and M.J. Richard. On the direct kinematics of spherical three-degree-of-freedom parallel manipulators of general architecture. *Journal of Mechanical Design*, 116(2):594-598, 1994.

[18] C.M. Gosselin, M. Gagne. A Closed-form solution for the direct kinematics of a special class of spherical three-degree-of-freedom parallel manipulators. *Computation Kinematics*, Kluwer Academic Publishers, 231-240, 1995.

[19] C.M. Gosselin, J. Wang. Singularity loci of a special class of spherical three-degree-of-freedom parallel mechanisms with revolute actuators. *The International Journal of Robotics Research*, 21(7):649-659, 2002.

[20] I.A. Bonev, C.M. Gosselin. Singularity loci of spherical parallel mechanisms. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* , pages 2957-2962, 18-22 Apr. 2005.

[21] I.A. Bonev, D. Chablat, P. Wenger. Working and assembly modes of the agile eye. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on* , pages 2317-2322, 2006.

[22] I.A. Bonev, C.M. Gosselin. Analytical determination of the workspace of symmetrical spherical parallel mechanisms," *Robotics, IEEE Transactions on* , 22(5):1011-1017, Oct. 2006

[23] C.M. Gosselin, J. Angeles. A global performance index for the kinematic optimization of robotic manipulators. *Journal of Mechanical Design*, 113(3):220-226, 1991.

[24] J.P. Merlet. Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots. Journal of Mechanical Design, 128(1):199-206, 2005.

[25] C.M. Gosselin. The optimum design of robotic manipulators using dexterity indices. *Robotics and Autonomous Systems*, 9(4):213-226, 1992.

[26] T. Huang, C.M. Gosselin, D.J. Whitehouse and D.G. Chetwynd. Analytical approach for optimal Design of a Type of Spherical Parallel Manipulator Using Dexterous Performance Indices. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 217(4):447-455, 2003.

[27] S.P. Bai. Optimum design of spherical parallel manipulators for a prescribed workspace. *Mechanism and Machine Theory*, 45(2):200-211, 2010.

[28] G.L. Wu. Multiobjective optimum design of a 3-RRR spherical parallel manipulator with kinematic and dynamic dexterities. *Modeling, Identification and Control*, 33(3):111-121, 2012.

[29] A.I. Kapandji. *The Physiology of the Joints, Volume 1: Upper Limb*. Churchill Livingstone, 6 edition, 2007.

[30] Pronation and Supination of The Hand [Online] Cea1.com http://www.cea1.com/anatomy-sistems/pronation-and-supination-of-the-hand/

[31] H.I. Krebs, N. Hogan, M.L. Aisen and B.T. Volpe. *Robotaided Neurorehabilitation. IEEE Transactions. Rehabilitation Engineering*, 6:75–87, 1998.

[32] P.S. Lum, D.J. Reinkensmeyer and S.L. Lehman. Robotic assist devices for bimanual physical therapy: preliminary experiments. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 1:185–191, 1993.

[33] L.E. Kahn, M.L. Zygman, W.Z. Rymer and D.J. Reinkensmeyer. Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic

stroke: A randomized controlled pilot study. *Journal of NeuroEngineering and Rehabilitation*, 3(12), 2006.

[34] P.S. Lum, C.G. Burgar, P.C. Shor, M. Majmundar and H.F.M. Van der Loos. Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper limb motor function following stroke. *Archives of Physical Medicine and Rehabilitation*, 83:952–959, 2002.

[35] J.A. Martinez, P. Ng, S. Lu, M.S. Campagna and O. Celik. Design of Wrist Gimbal. A forearm and wrist exoskeleton for stroke rehabilitation. *2013 IEEE International Conference on Rehabilitation Robotics*, 2013.

[36] J.F. Ribeiro, J.C.M. Carvalho, L.P. Oliveira, L.A.O. Rodrigues and R.S. Gonc¸alves. Robot for wrist rehabilitation. *New Trends in Mechanism and Machine Science, Mechanisms and Machine Science*. Springer, 451-458, 2013

[37] Gupta, M. K. O'Malley, V. Patoglu and C. Burgar. Design, Control and performance of RiceWrist: A force feedbackWrist exoskeleton for rehabilitation and training. *The International Journal of Robotics Research*, 27(2):233-251, 2008.

[38] J. Oblak, I. Cikajlo, Z. Matjačić. Universal haptic drive: A robot for arm and wrist rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(3):293-302, 2010.

[39] J. Allington, S.J. Spencer, J. Klein, M. Buell, D.J. Reinkensmeyer and J. Bobrow. Supinator extender (SUE): A pneumatically actuated robot for forearm/wrist rehabilitation after stroke. *Engineering in Medicine and Biology Society*, 1579-1582, 2011.

[40] J.U. Korein. *A Geometric Investigation of Reach*. MIT Press, 1984.

[41] I.A. Bonev, J. Ryu. A new approach to orientation workspace analysis of 6-DOF parallel manipulators. *Mechanism and Machine Theory*, 36(1):15-28, 2001.

[42] I.A.Bonev, D. Zlatanov and C.M. Gosselin. Advantages of the modified Euler angles in the design and control of PKMs. *2002 Parallel Kinematic Machines International Conference*, Chemnitz, Germany, pages 171–188, April 23–25, 2002.

[43] I.A. Bonev. Direct kinematics of zero-torsion parallel mechanisms. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* , pages 3851-3856, 19-23 May 2008.

[44] C.M. Gosselin. Stiffness mapping for parallel manipulators. *IEEE Transactions on Robotics and Automation*, 6(3):377-382, 1990.

[45] C.M. Gosselin, D. Zhang. Stiffness analysis of parallel mechanisms using a lumped model. *International Journal of Robotics and Automation*. 17(1):17–27, 2003.

[46] A. Pashkevich, D. Chablat and P. Wenger. Stiffness analysis of overconstrained parallel manipulators. *Mechanism and Machine Theory*, 44(5):966–982, 2009.

[47] G.L. Wu, S.P. Bai and J. Kepler. Mobile platform center shift in spherical parallel manipulators with flexible limbs. *Mechanism and Machine Theory*, 75:12-26, May 2014.

[48] T. Yoshikawa. Manipulability of robotic mechanisms. *International Journal of Robotics Research*, 4(2):3-9, 1985.

[49] J. Kim, P. Khosla. Dexterity measures for design and control of manipulators. *IEEE International Workshop on Robots and Systems IROS'91*, pages 403-415, 1991.

[50] H.H. Pham, I.M. Chen. Optimal synthesis for workspace and manipulability of parallel flexure mechanism. *11th World Congress on Theory of Machines and Mechanisms*, 2004.

[51] S. Rainer, P. Kenneth. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report*, Berkeley, CA, TR-95-012, 1995.

[52] M. Toz, S. Kucuk. Dexterous workspace optimization of an asymmetric six-degree of freedom Stewart–Gough platform type manipulator. *Robotics and Autonomous Systems*, 61(12):1516-1528, 2013.

[53] L. Weihmann, D. Martins and L.S. Coelho. Modified differential evolution approach for optimization of planar parallel manipulators force capabilities. *Expert Systems with Applications*, 39(6):6150-6156, 2012.

[54] S. Das. Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4-31, 2011.

[55] F.Y. Edgeworth. *Mathematical psychics: an essay on the application of mathematics to the moral Sciences*, C.K. Paul co., 1881.

[56] E.G. Talbi. *Metaheuristics: from design to implementation*, Wiley, 2009.

[57] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[58] Z. Eckart, D. Kalyanmoy and T. Lothar. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8(2):173-195, 2000.

[59] C.A. Klein, B.E. Blaho. Dexterity measures for the design and control of kinematically redundant manipulators. *International Journal of Robotics Research*, 6(2):72–83, 1987.

[60] J.Q. Zhang, A.C. Sanderson. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945-958, 2009.

[61] K.L. Doty, C. Melchiori, E.M. Schwartz, E.M. Bonivento. Robot manipulability. *IEEE Transactions on Robotics and Automation*, 11(3):462–468, 1995.

[62] T. Yoshikawa. Dynamic manipulability of robot manipulators. *Journal of Robotic Systems*, 2: 113–124, 1985.

[63] J. Angeles, C. López-Cajún. Kinematic isotropy and the conditioning index of serial robotic manipulators. *International Journal of Robotics Research*, 11(6): 560–571, 1992.

[64] G. Strang. *Linear Algebra and its Application*, Academic Press, New York, 1976.

[65] J.K. Salisbury, J.J. Graig. Articulated hands: force control and kinematic issues. *The International Journal of Robotics Research*, 1(1): 4–17, 1982.

9 Appendix A: Matlab Code

9.1 Inverse Kinematics

```
clear all;
clc;

ph = 0;
th = 0;
si = 0;
b = 54.74*pi/180;
r = 54.74*pi/180;
a1 = pi/2;
a2 = pi/2;

% Roattion Matrix
R = [cos(ph)*cos(th)*cos(si-ph)-sin(ph)*sin(si-ph), -
cos(ph)*cos(th)*sin(si-ph)-sin(ph)*cos(si-ph),
cos(ph)*sin(th);
     sin(ph)*cos(th)*cos(si-ph)+cos(ph)*sin(si-ph), -
sin(ph)*cos(th)*sin(si-ph)+cos(ph)*cos(si-ph),
sin(ph)*sin(th);
    -sin(th)*cos(si-ph),
sin(th)*sin(si-ph),                                cos(th);];

et1 = 0;
et2 = 2*pi/3;
et3 = 4*pi/3;

% v vector in moving platform
v11 = [-sin(et1)*sin(b);
        cos(et1)*sin(b);
        cos(b)];
v22 = [-sin(et2)*sin(b);
        cos(et2)*sin(b);
        cos(b)];
v33 = [-sin(et3)*sin(b);
        cos(et3)*sin(b);
        cos(b)];

% v vector in basement
v1 = R*v11;
```

```
v2 = R*v22;
v3 = R*v33;

% Equation
% Theta1
D = v1(1)*(sin(et1)*sin(r)*cos(a1)+sin(et1)*cos(r)*sin(a1))
+ v1(2)*(cos(et1)*sin(r)*cos(a1)-cos(et1)*cos(r)*sin(a1)) -
v1(3)*(cos(r)*cos(a1)+sin(r)*sin(a1)) - cos(a2);
E = v1(1)*(sin(et1)*sin(r)*cos(a1)-sin(et1)*cos(r)*sin(a1))
+ v1(2)*(cos(et1)*sin(r)*cos(a1)+cos(et1)*cos(r)*sin(a1)) +
v1(3)*(sin(r)*sin(a1)-cos(r)*cos(a1)) - cos(a2);
A = v1(1)*cos(et1)*sin(a1) + v1(2)*sin(et1)*sin(a1);
if 4*A^2-4*D*E == 0
    e1 = (2*atan(-A/D))*180/pi;
elseif 4*A^2-4*D*E > 0
    theta1 = (2*atan((-2*A+(4*A^2-4*D*E)^0.5)/2*D))*180/pi;
    theta2 = (2*atan((-2*A-(4*A^2-4*D*E)^0.5)/2*D))*180/pi;
    if abs(theta1) < abs(theta2)
        e1 = theta1;
    else
        e1 = theta2;
    end
end

% Theta2
D = v2(1)*(sin(et2)*sin(r)*cos(a1)+sin(et2)*cos(r)*sin(a1))
+ v2(2)*(cos(et2)*sin(r)*cos(a1)-cos(et2)*cos(r)*sin(a1)) -
v2(3)*(cos(r)*cos(a1)+sin(r)*sin(a1)) - cos(a2);
E = v2(1)*(sin(et2)*sin(r)*cos(a1)-sin(et2)*cos(r)*sin(a1))
+ v2(2)*(cos(et2)*sin(r)*cos(a1)+cos(et2)*cos(r)*sin(a1)) +
v2(3)*(sin(r)*sin(a1)-cos(r)*cos(a1)) - cos(a2);
A = v2(1)*cos(et2)*sin(a1) + v2(2)*sin(et2)*sin(a1);
if 4*A^2-4*D*E == 0
    e2 = (2*atan(-A/D))*180/pi;
elseif 4*A^2-4*D*E > 0
    theta1 = (2*atan((-2*A+(4*A^2-4*D*E)^0.5)/2*D))*180/pi;
    theta2 = (2*atan((-2*A-(4*A^2-4*D*E)^0.5)/2*D))*180/pi;
    if abs(theta1) < abs(theta2)
        e2 = theta1;
    else
        e2 = theta2;
    end
end

% Theta3
```

```
D = v3(1)*(sin(et3)*sin(r)*cos(a1)+sin(et3)*cos(r)*sin(a1))
+ v3(2)*(cos(et3)*sin(r)*cos(a1)-cos(et3)*cos(r)*sin(a1)) -
v3(3)*(cos(r)*cos(a1)+sin(r)*sin(a1)) - cos(a2);
E = v3(1)*(sin(et3)*sin(r)*cos(a1)-sin(et3)*cos(r)*sin(a1))
+ v3(2)*(cos(et3)*sin(r)*cos(a1)+cos(et3)*cos(r)*sin(a1)) +
v3(3)*(sin(r)*sin(a1)-cos(r)*cos(a1)) - cos(a2);
A = v3(1)*cos(et3)*sin(a1) + v3(2)*sin(et3)*sin(a1);
if 4*A^2-4*D*E == 0
    e3 = (2*atan(-A/D))*180/pi;
elseif 4*A^2-4*D*E > 0
    theta1 = (2*atan((-2*A+(4*A^2-4*D*E)^0.5)/2*D))*180/pi;
    theta2 = (2*atan((-2*A-(4*A^2-4*D*E)^0.5)/2*D))*180/pi;
    if abs(theta1) < abs(theta2)
        e3 = theta1;
    else
        e3 = theta2;
    end
end
```

9.2 Jacobian Matrix

```
clear all;
clc;

%% Parameters
et1 = 0;
et2 = 2*pi/3;
et3 = 4*pi/3;
% Orientation
ph = 0.2;
th = 0.1;
si = 0;
% Top and bottom pyramid
b = pi/3;
r = pi/3;
% Link angle
a1 = pi/2;
a2 = pi/2;
% Actuator angle
e1 = 0;
e2 = 0;
e3 = 0;
% Link radios
```

```
R1 = 5;
R2 = 3;

%% Roattion Matrix
R = [cos(ph)*cos(th)*cos(si-ph)-sin(ph)*sin(si-ph), -
cos(ph)*cos(th)*sin(si-ph)-sin(ph)*cos(si-ph),
cos(ph)*sin(th);
     sin(ph)*cos(th)*cos(si-ph)+cos(ph)*sin(si-ph), -
sin(ph)*cos(th)*sin(si-ph)+cos(ph)*cos(si-ph),
sin(ph)*sin(th);
    -sin(th)*cos(si-ph),
sin(th)*sin(si-ph),                              cos(th);];

%% V vector in moving platform
V11 = [-sin(et1)*sin(b);
        cos(et1)*sin(b);
        cos(b)];
V22 = [-sin(et2)*sin(b);
        cos(et2)*sin(b);
        cos(b)];
V33 = [-sin(et3)*sin(b);
        cos(et3)*sin(b);
        cos(b)];

% v vector in basement
v1 = R*V11;
v2 = R*V22;
v3 = R*V33;

%% U vector
u1 = [-sin(et1)*sin(r);
       cos(et1)*sin(r);
      -cos(r)];
u2 = [-sin(et2)*sin(r);
       cos(et2)*sin(r);
       -cos(r)];
u3 = [-sin(et3)*sin(r);
       cos(et3)*sin(r);
      -cos(r)];

%% W vector
w1 = [-sin(et1)*sin(r)*cos(a1)+(cos(et1)*sin(e1)-
sin(et1)*cos(r)*cos(e1))*sin(a1);
       cos(et1)*sin(r)*cos(a1)+(sin(et1)*sin(e1)-
cos(et1)*cos(r)*cos(e1))*sin(a1);
      -cos(r)*cos(a1) + sin(r)*cos(e1)*sin(a1);];
```

```
w2 = [-sin(et2)*sin(r)*cos(a1)+(cos(et2)*sin(e2)-
sin(et2)*cos(r)*cos(e2))*sin(a1);
       cos(et2)*sin(r)*cos(a1)+(sin(et2)*sin(e2)-
cos(et2)*cos(r)*cos(e2))*sin(a1);
      -cos(r)*cos(a1) + sin(r)*cos(e2)*sin(a1);];
w3 = [-sin(et3)*sin(r)*cos(a1)+(cos(et3)*sin(e3)-
sin(et3)*cos(r)*cos(e3))*sin(a1);
       cos(et3)*sin(r)*cos(a1)+(sin(et3)*sin(e3)-
cos(et3)*cos(r)*cos(e3))*sin(a1);
      -cos(r)*cos(a1) + sin(r)*cos(e3)*sin(a1);];

%% Jacobian
J1 = cross(w1,v1)/dot(cross(u1,w1),v1);
J2 = cross(w2,v2)/dot(cross(u2,w2),v2);
J3 = cross(w3,v3)/dot(cross(u3,w3),v3);
J = [J1,J2,J3].';
```

9.3 Stiffness Calculation

```
function f = stiffnessF (ph,th,si,b,r,a1,e1,e2,e3,R1,R2)
    %% Parameters
    et1 = 0;
    et2 = 2*pi/3;
    et3 = 4*pi/3;

    %% Roattion Matrix
    R = [cos(ph)*cos(th)*cos(si-ph)-sin(ph)*sin(si-ph), -
cos(ph)*cos(th)*sin(si-ph)-sin(ph)*cos(si-ph),
cos(ph)*sin(th);
         sin(ph)*cos(th)*cos(si-ph)+cos(ph)*sin(si-ph), -
sin(ph)*cos(th)*sin(si-ph)+cos(ph)*cos(si-ph),
sin(ph)*sin(th);
        -sin(th)*cos(si-ph),
sin(th)*sin(si-ph),                              cos(th);];

    %% V vector in moving platform
    V11 = [-sin(et1)*sin(b);
            cos(et1)*sin(b);
            cos(b)];
    V22 = [-sin(et2)*sin(b);
            cos(et2)*sin(b);
            cos(b)];
    V33 = [-sin(et3)*sin(b);
```

```
            cos(et3)*sin(b);
            cos(b)];

    % V vector in basement
    v1 = R*V11;
    v2 = R*V22;
    v3 = R*V33;

    %% U vector
    u1 = [-sin(et1)*sin(r);
           cos(et1)*sin(r);
          -cos(r)];
    u2 = [-sin(et2)*sin(r);
           cos(et2)*sin(r);
           -cos(r)];
    u3 = [-sin(et3)*sin(r);
           cos(et3)*sin(r);
          -cos(r)];

    %% W vector
    w1 = [-sin(et1)*sin(r)*cos(a1)+(cos(et1)*sin(e1)-
sin(et1)*cos(r)*cos(e1))*sin(a1);
           cos(et1)*sin(r)*cos(a1)+(sin(et1)*sin(e1)-
cos(et1)*cos(r)*cos(e1))*sin(a1);
          -cos(r)*cos(a1) + sin(r)*cos(e1)*sin(a1);];
    w2 = [-sin(et2)*sin(r)*cos(a1)+(cos(et2)*sin(e2)-
sin(et2)*cos(r)*cos(e2))*sin(a1);
           cos(et2)*sin(r)*cos(a1)+(sin(et2)*sin(e2)-
cos(et2)*cos(r)*cos(e2))*sin(a1);
          -cos(r)*cos(a1) + sin(r)*cos(e2)*sin(a1);];
    w3 = [-sin(et3)*sin(r)*cos(a1)+(cos(et3)*sin(e3)-
sin(et3)*cos(r)*cos(e3))*sin(a1);
           cos(et3)*sin(r)*cos(a1)+(sin(et3)*sin(e3)-
cos(et3)*cos(r)*cos(e3))*sin(a1);
          -cos(r)*cos(a1) + sin(r)*cos(e3)*sin(a1);];

    %% Stiffness
    B1 = R1*w1; B2 = R1*w2; B3 = R1*w3;
    C1 = R2*v1; C2 = R2*v2; C3 = R2*v3;

    SA1 = [u1;0;0;0]; SA2 = [u2;0;0;0]; SA3 = [u3;0;0;0];
    SB1 = [w1;0;0;0]; SB2 = [w2;0;0;0]; SB3 = [w3;0;0;0];
    SC1 = [v1;0;0;0]; SC2 = [v2;0;0;0]; SC3 = [v3;0;0;0];

    n11 = cross(u1,w1)/norm(cross(u1,w1));
    n12 = cross(u2,w2)/norm(cross(u2,w2));
    n13 = cross(u3,w3)/norm(cross(u3,w3));
```

```
n21 = cross(w1,v1)/norm(cross(w1,v1));
n22 = cross(w2,v2)/norm(cross(w2,v2));
n23 = cross(w3,v3)/norm(cross(w3,v3));

r11 = cross(w1,n11)/norm(cross(w1,n11));
r12 = cross(w2,n12)/norm(cross(w2,n12));
r13 = cross(w3,n13)/norm(cross(w3,n13));

r21 = cross(v1,n21)/norm(cross(v1,n21));
r22 = cross(v2,n22)/norm(cross(v2,n22));
r23 = cross(v3,n23)/norm(cross(v3,n23));

Su11 = [r11;cross(B1,r11)];
Su12 = [r12;cross(B2,r12)];
Su13 = [r13;cross(B3,r13)];

Su21 = SB1;
Su22 = SB2;
Su23 = SB3;

Su31 = [n11;cross(B1,n11)];
Su32 = [n12;cross(B2,n12)];
Su33 = [n13;cross(B3,n13)];

Su41 = [0;0;0;r11];
Su42 = [0;0;0;r12];
Su43 = [0;0;0;r13];

Su51 = [0;0;0;w1]; %
Su52 = [0;0;0;w2];
Su53 = [0;0;0;w3];

Su61 = [0;0;0;n11];
Su62 = [0;0;0;n12];
Su63 = [0;0;0;n13];

Su71 = [r21;cross(C1,r21)];
Su72 = [r22;cross(C2,r22)];
Su73 = [r23;cross(C3,r23)];

Su81 = SC1;
Su82 = SC2;
Su83 = SC3;

Su91 = [n21;cross(C1,n21)];
Su92 = [n22;cross(C2,n22)];
```

```
    Su93 = [n23;cross(C3,n23)];

    Su101 = [0;0;0;r21];
    Su102 = [0;0;0;r22];
    Su103 = [0;0;0;r23];

    Su111 = [0;0;0;v1];
    Su112 = [0;0;0;v2];
    Su113 = [0;0;0;v3];

    Su121 = [0;0;0;n21];
    Su122 = [0;0;0;n22];
    Su123 = [0;0;0;n23];

    Je1 =
[SA1,Su11,Su21,Su31,Su41,Su51,Su61,Su71,Su81,Su91,Su101,Su11
1,Su121];
    Je2 =
[SA2,Su12,Su22,Su32,Su42,Su52,Su62,Su72,Su82,Su92,Su102,Su11
2,Su122];
    Je3 =
[SA3,Su13,Su23,Su33,Su43,Su53,Su63,Su73,Su83,Su93,Su103,Su11
3,Su123];

    Jq1 = [SB1,SC1];
    Jq2 = [SB2,SC2];
    Jq3 = [SB3,SC3];

    % Stiffness begin
    Kact = 100;
    O16 = zeros(1,6);
    O61 = zeros(6,1);
    O66 = zeros(6,6);
    KL1 = diag([100,100,100,100,100,100]);
    KL2 = diag([100,100,100,100,100,100]);
    % stiffness end

    Ke1 = [Kact,O16,O16;
           O61,KL1,O66;
           O61,O66,KL2;];
    Ke2 = Ke1;
    Ke3 = Ke1;

    Se1 = Je1*(inv(Ke1))*(Je1.');
    Se2 = Je2*(inv(Ke2))*(Je2.');
    Se3 = Je3*(inv(Ke3))*(Je3.');
```

```
    kk1 = [Se1, Jq1;
         Jq1.', zeros(2,2)];
    kk2 = [Se2, Jq2;
         Jq2.', zeros(2,2)];
    kk3 = [Se3, Jq3;
         Jq3.', zeros(2,2)];

    k1 = inv(kk1);
    k2 = inv(kk2);
    k3 = inv(kk3);

    K1 = k1(1:6,1:6);
    K2 = k2(1:6,1:6);
    K3 = k3(1:6,1:6);

    S = K1+K2+K3;

    f = trace(S);

end
```

9.4 Constraints Check

```
function y = ckCnst (ph,th,si,e1,e2,e3,b,r,a1)

    %% Parameters
    et1 = 0;
    et2 = 2*pi/3;
    et3 = 4*pi/3;
    tol = 0.01;

    %% Roattion Matrix
    R = [cos(ph)*cos(th)*cos(si-ph)-sin(ph)*sin(si-ph), -
cos(ph)*cos(th)*sin(si-ph)-sin(ph)*cos(si-ph),
cos(ph)*sin(th);
         sin(ph)*cos(th)*cos(si-ph)+cos(ph)*sin(si-ph), -
sin(ph)*cos(th)*sin(si-ph)+cos(ph)*cos(si-ph),
sin(ph)*sin(th);
        -sin(th)*cos(si-ph),
sin(th)*sin(si-ph),                              cos(th);];

    %% V vector in moving platform
    V11 = [-sin(et1)*sin(b);
```

```
            cos(et1)*sin(b);
            cos(b)];
    V22 = [-sin(et2)*sin(b);
            cos(et2)*sin(b);
            cos(b)];
    V33 = [-sin(et3)*sin(b);
            cos(et3)*sin(b);
            cos(b)];

    % v vector in basement
    v1 = R*V11;
    v2 = R*V22;
    v3 = R*V33;

    %% U vector
    u1 = [-sin(et1)*sin(r);
           cos(et1)*sin(r);
          -cos(r)];
    u2 = [-sin(et2)*sin(r);
           cos(et2)*sin(r);
           -cos(r)];
    u3 = [-sin(et3)*sin(r);
           cos(et3)*sin(r);
          -cos(r)];

    %% W vector
    w1 = [-sin(et1)*sin(r)*cos(a1)+(cos(et1)*sin(e1)-
sin(et1)*cos(r)*cos(e1))*sin(a1);
           cos(et1)*sin(r)*cos(a1)+(sin(et1)*sin(e1)-
cos(et1)*cos(r)*cos(e1))*sin(a1);
          -cos(r)*cos(a1) + sin(r)*cos(e1)*sin(a1);];
    w2 = [-sin(et2)*sin(r)*cos(a1)+(cos(et2)*sin(e2)-
sin(et2)*cos(r)*cos(e2))*sin(a1);
           cos(et2)*sin(r)*cos(a1)+(sin(et2)*sin(e2)-
cos(et2)*cos(r)*cos(e2))*sin(a1);
          -cos(r)*cos(a1) + sin(r)*cos(e2)*sin(a1);];
    w3 = [-sin(et3)*sin(r)*cos(a1)+(cos(et3)*sin(e3)-
sin(et3)*cos(r)*cos(e3))*sin(a1);
           cos(et3)*sin(r)*cos(a1)+(sin(et3)*sin(e3)-
cos(et3)*cos(r)*cos(e3))*sin(a1);
          -cos(r)*cos(a1) + sin(r)*cos(e3)*sin(a1);];

    % 1. Check angle, if the angle between w1-w2, w1-w3, w2-
w3 smaller than 10 degree:
    w12 = acosd(dot(w1,w2)/(norm(w1)*norm(w2)));
    w13 = acosd(dot(w1,w3)/(norm(w1)*norm(w3)));
    w23 = acosd(dot(w2,w3)/(norm(w2)*norm(w3)));
```

```
    if w12 > 10 && w13 > 10 && w23 >10
        Angle = 1;
    else
        Angle = 0;
    end
    % 2. Check matrix A and B
    A = [(cross(w1,v1)).';
         (cross(w2,v2)).';
         (cross(w3,v3)).';];
    B11 = dot(cross(w1,v1),u1);
    B22 = dot(cross(w2,v2),u2);
    B33 = dot(cross(w3,v3),u3);
    if cond(A) < (1/tol) && abs(B11) > tol && abs(B22) > tol
&& abs(B33) > tol
        mAB = 1;
    else
        mAB = 0;
    end

    % 3. Check condition number
    J1 = cross(w1,v1)/dot(cross(u1,w1),v1);
    J2 = cross(w2,v2)/dot(cross(u2,w2),v2);
    J3 = cross(w3,v3)/dot(cross(u3,w3),v3);
    J = [J1,J2,J3].';
    if cond(J) < (1/tol)
        ConJ = 1;
    else
        ConJ = 0;
    end

    if Angle && mAB && ConJ
        y = 1;
    else
        y = 0;
    end

end
```