

Use Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission to download and/or print my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature_____

Date_____

DESIGN OF DEXTEROUS MULTI-FINGERED SINGLE ROBOTIC HAND FOR BIMANUAL TASKS

by

Reza Movassagh Khaniki

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Mechanical Engineering

College of Engineering, Idaho State University

Summer, 2017

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of Reza Movassagh Khaniki find it satisfactory and recommend that it be accepted.

_____**Major Advisor**

Dr. Alba Perez-Gracia

_____**Member**

Dr. Kenneth Bosworth

_____**Graduate Faculty Representative**

Dr. Arya Ebrahimpour

Dedication

This thesis is dedicated to my beloved and brilliant wife, **Neda Hassanzadeh**; who has been a ceaseless source of support, energy and encouragement during the every second of my life. If it was not for you, I would not be writing this.

Acknowledgements

I would like to express my gratitude to my advisor, Dr. Alba Perez-Gracia, for her excellent supports, guidance, and providing me with a great opportunity to peruse my education.

Special thanks goes to Dr. Bosworth and Dr. Ebrahimpour, who have always been supporting me and also willing to participate in my final defense committee.

I would like to thank all the people who contributed in some way to the work described in this thesis, in particular, Vahid Pourgharibshahi and Abhijit Makhal.

I would also like to thank my parents, my brother and my sister. They were always supporting me and encouraging me with their best wishes.

Finally, I would like to thank my wonderful wife, Neda. She has been always there when I was frustrated and she has stood by me through every moment in my life. I greatly value her contribution and deeply appreciate her belief in me.

CONTENTS

List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Literature Review.....	1
1.2 Thesis Goals.....	4
2 Background.....	5
2.1 Kinematics	6
2.1.1 Finite Displacement	7
2.1.2 Spatial Displacement	8
2.1.3 Screw Displacement.....	9
2.1.4 Dual Quaternions	10
2.2 Kinematic Synthesis.....	11
2.2.1 Tree Topologies	12
2.2.2 Task Definition	13
2.2.3 Type Synthesis.....	14
2.2.4 Solvability.....	14
2.2.5 Dimensional Synthesis.....	15
3 Robotic Hand Design Methodology.....	16
3.1 Tree Topology.....	16
3.2 Kinematic Tasks Data Gathering Methods:.....	20
3.2.1 Simulation Data	20
3.2.2 Motion Capture for Kinematic Tasks	21
3.3 Structural Synthesis	23
3.4 Dimensional Synthesis.....	24
3.4.1 Solvability	25
3.4.2 Numerical Solution	26
4 Design of A Multi-Palm Hand for Assembly Tasks.....	27
4.1 Multi-Hand Tasks	27
4.2 In-Hand Manipulation and Dexterity.....	27
4.3 Design of a Multi-Palm Robotic Hand	28
4.3.1 Motivation and Task Selection	28
4.3.2 Task-Based Mobility.....	30

4.3.3	Type Synthesis	31
4.3.4	Hand Design.....	32
5	Design of a Dexterous Hand for An Orange Peeling Task.....	52
5.1	Motivation and Task Selection	52
5.2	Task Definition	53
5.3	Structural Synthesis	55
5.4	Dimensional Synthesis.....	56
5.5	Hand Implementation.....	57
6	Conclusion	60
REFERENCES		62
Appendix A	Mathematica® Codes for Assembly Hand Using Sequential Synthesis	66
Appendix B	Mathematica® Codes for Orange Peeler hand.....	101

List of Figures

FIGURE 1-1: IN-HAND MANIPULATION WITH EXTERNAL FORCES [5].....	2
FIGURE 1-2: ROBOTIC ASSEMBLY OF EMERGENCY STOP BUTTONS [6].....	3
FIGURE 2-1: A RIGID BODY [24]	7
FIGURE 2-2: A RIGID DISPLACEMENT [24]	7
FIGURE 2-3: REPRESENTATION OF SCREW AXIS [24]	9
FIGURE 2-4: ON THE RIGHT; MOTION OF THE LEFT HAND, ON THE LEFT; MOTION OF RIGHT HAND	13
FIGURE 3-1: HUMAN HAND AND BARRETT HAND® [32].....	17
FIGURE 3-2: HUMAN HAND GRAPH [32]	17
FIGURE 3-3: BARRETT HAND® GRAPH [32]	18
FIGURE 3-4: LEFT SIDE, ORIGINAL GRAPH; RIGHT SIDE, REDUCED GRAPH [27]	18
FIGURE 3-5: TREE GRAPH FOR THE 1-(1;1;1;1) ROBOTIC HAND WITH ONE PALM [33].....	19
FIGURE 3-6: LEFT SIDE, THE NUMBERING OF THE EDGE; RIGHT SIDE, THE NUMBER OF JOINTS FOR EACH EDGE [29]	19
FIGURE 3-7: A SAMPLE OF A MOTION CAPTURE SYSTEM (VICON®).....	22
FIGURE 3-8: EXPERIMENTAL SETUP FOR MOTION CAPTURE.....	22
FIGURE 4-1: ASSEMBLY SUB-MOTIONS	29
FIGURE 4-2: THE TREE TOPOLOGY EARMARKED FOR THE HAND.	32
FIGURE 4-3 KINEMATIC SKETCH OF THE SELECTED HAND	32
FIGURE 4-4: TASK SIMULATION FRAMEWORK [38].....	34
FIGURE 4-5: ASSEMBLY TASK AND SUB-TASKS CAPTURED FRAMES FOR EACH FINGER FOR THE TASK.....	35
FIGURE 4-6: ON TOP, IMPROVED MODELLED HAND IN CAD ENVIRONMENT; ON BOTTOM, 3D PRINTED MODEL	37
FIGURE 4-7: MARKERS CONFIGURATION ON THE HAND.....	38
FIGURE 4-8: M1 FOR RIGHT HAND (GRASPING THE SWITCH).....	39
FIGURE 4-9: M1 FOR LEFT HAND (HOLDING THE BASE)	40
FIGURE 4-10: M2 FOR RIGHT HAND (INSERTING THE BUTTON; IN TWO VIEWS).....	40

FIGURE 4-11: M2 FOR LEFT HAND (HOLDING AND ROTATING THE COVER)	41
FIGURE 4-12: M3 FOR RIGHT HAND (HOLDING THE UPPER- PART)	41
FIGURE 4-13: M3 FOR RIGHT HAND (GRASPING AND SCREWING THE NUT; IN TWO VIEWS).....	42
FIGURE 4-14: M4 FOR RIGHT HAND (INSERTING UPPER PART ON THE BASE)	42
FIGURE 4-15: M4 FOR LEFT HAND (HOLDING THE BASE)	42
FIGURE 4-16 RELATIVE MOTION LEVELS.	43
FIGURE 4-17: GRAPH OF THE 2-(2,1) HAND.	44
FIGURE 4-18: KINEMATIC SKETCH OF THE 2-(2,1) SUB- HAND.....	44
FIGURE 4-19: GRAPH OF THE 2-(2,2) HAND.	47
FIGURE 4-20: KINEMATIC SKETCH OF THE 2-(2,2) HAND.	47
FIGURE 4-21: KINEMATIC SKETCH OF THE 2-(1,1) HAND.	49
FIGURE 4-22: THE HAND MODELLED IN A 3D ENVIRONMENT	50
FIGURE 5-1 THE DA VINCI® SURGICAL SYSTEM [42]	52
FIGURE 5-2: DA VINCI® SYSTEM - PEELING A GRAPE [42]	53
FIGURE 5-3: MOTION CAPTURE STEPS	54
FIGURE 5-4: GRAPHIC OF MOTION CAPTURE [43]	55
FIGURE 5-5 SELECTED HAND TOPOLOGY FOR THE TASK AND KINEMATIC SKETCH. [43]	56
FIGURE 5-6: THE FINAL HAND MODEL	58
FIGURE 5-7: PROTOTYPE MADE BY 3D PRINTER	59

List of Tables

TABLE 1 EXAMPLES OF TYPE SYNTHESIS.....	24
TABLE 2: SUB-TASKS WITH THE REQUIRED MOBILITY AND MOBILITY LEVEL.	31
TABLE 3 PLUCKER COORDINATES OF THE HAND WITH MOTION SIMULATION DATA.	36
TABLE 4 GRIPPER 1 JOINT AXES.	46
TABLE 5 GRIPPER 2 JOINT AXES.	46
TABLE 6 SECONDARY LEVEL JOINT AXES.	48
TABLE 7 WRIST LEVEL JOINT AXES DATA.	49
TABLE 8 PLUCKER COORDINATES FOR THE JOINTS OF THE HAND.	57

ABSTRACT

Vast numbers of researchers around the globe are devoted to the research topics of the robotic hands. Studying robotic hands for a particular task might led to the creation of effective and developed end-effectors for grasping and manipulation. In case of studying robotic tasks, specifically, those which have interaction between manipulated objects, working of two robotic arms or robotic hands equipped with end-effector grippers is needed.

Nowadays, many of current bimanual tasks are performed by using two robot arm manipulators with simple grippers. In these scenarios, the end-effectors are mainly used for grasping an object. On the other hand, motion itself is carried out by robotic arms. In this thesis, we propose the design of a highly dexterous multi-fingered robotic hand which is able to perform the bimanual tasks when attached to a simple arm manipulator. By considering the kinematic synthesis for tree topologies, dexterous robotic hands can be created with more than one splitting stage. The synthesis process is applied in this research to the design of a robotic hand with three palms for a bimanual task consisting of assembling an emergency stop button and single-palmed with three fingers for peeling an orange.

1 Introduction

1.1 Literature Review

The use of robotic hands in machinery and manufacturing industrial applications is becoming increasingly common. Also, we can trace their existence in experiential and laboratory environments, specifically those in which, human is not able to work freely such as nuclear laboratories or biohazard conditions.

Usually robotic tasks need simultaneous work of two or more robot manipulators, namely two robot arms working collaboratively on a single object or the combination of a positioning device with a robot manipulator, performing an operational task. We could mention many examples in this regard. For instance, delicate surgeries such as eye cornea or mixing some particular chemical materials that may not be doable for human due to dangerous condition which may cause serious damages.

To fulfill such tasks, the coordination of the robots needs to be planned, which includes path planning with collision avoidance and control strategies. These three steps are necessary to accomplish a defined task. The path planning with collision avoidance is to find a sequence of configurations that moves a robot from an initial configuration to a goal destination without colliding with obstacles or itself in the environment.

Various methods have been explored on the control of such systems with simultaneous control strategies and collision avoidance. In [1], a control architecture is presented for real-time, simultaneous sensor-based, shared control of remote, multiple-manipulator tele-robotic systems. In [2], a relative description of the task has been proposed and a very simple collision avoidance algorithm, which permits a continuous progression of the task, has been developed. The work of [3] studies the problem of modeling and control of multiple cooperative under-actuated manipulators handling a rigid object. In [4],

a new adaptive, synchronized control algorithm is proposed for coordination of multiple manipulators in assembly tasks.

These tasks are mainly performed with non-dexterous end-effectors. For instance, we can see **Figure 1-1** and **Figure 1-2**. Whereas, by exploiting dexterous end-effectors which can be designed in shape of multi-fingered robotic hand with several splitting stages and with a kinematic structure, we would capable of doing more precise and complicated task with just one robotic arm. This act will lead to the desired dexterity for the task.

From a kinematic perspective, the robotic hand or the robotic arm plus hand system, can be defined as kinematic chain with tree topology with several common joints that branch to a number of serial chains, and possibly with several branching stages, ending with several end-effectors.

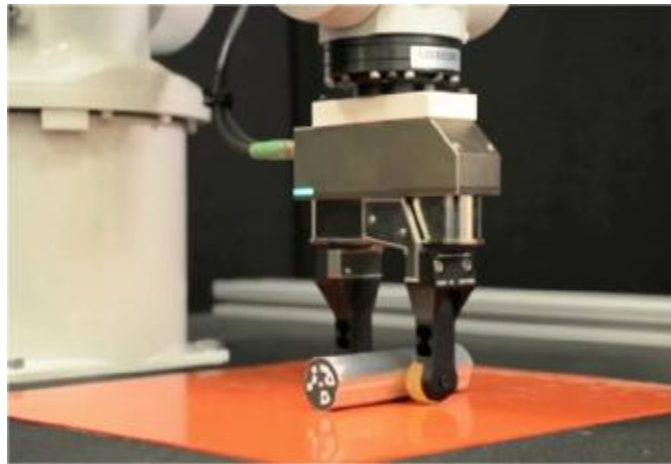


Figure 1-1: In-Hand Manipulation with External Forces[5]

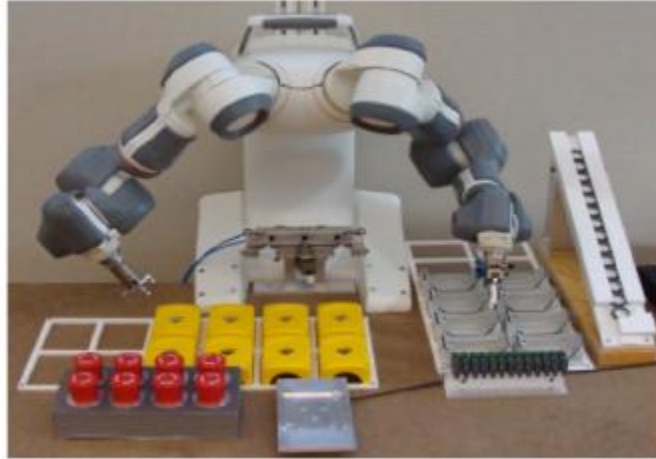


Figure 1-2: Robotic Assembly of Emergency Stop Buttons[6]

Dynamic and kinematic analysis of tree topologies is considered in [7] , [8] and [9], and type synthesis of tree topologies can be found in [10] for grasping and manipulation requirements. The dimensional synthesis of the tree topologies has been studied in [11], [12] and more extensively in [13]. As an example, we could say that, when we use the synthesis of tree topologies, by applying it to wristed multi-fingered hands, this makes it possible for us to design robotic hands for in-palm manipulation tasks, or complete arm and hand designs.

In this research, our interest is to find a way to switch from bimanual manipulation with non-dexterous grippers to high-dexterity single-hand manipulation, in which we are capable of selecting the degree of dexterity with respect to the kind of task we want to be exerted. In particular, the design of a hand for a set of tasks performed on multiple parts, such as an industrial assembly task and also orange peeling task are explored. The examples include all the steps from motion capture and synthesis to hand implementation and testing in CAD environment.

By applying this method, not only we could improve degree of accuracy during performing a task, but also, we could save energy and space when we want to implement robotic hands in different environments.

1.2 Thesis Goals

1. Study the kinematic and design issues in switching from bimanual manipulation with non-dexterous grippers to high-dexterity single-hand manipulation, where the degree of dexterity is selected according to the kind of task to be performed.
2. Design two dexterous robotic hand for two selected bimanual tasks: a) design a single three-palmed, four-fingered hand for assembling a commercial emergency stop button and b) design a single-palmed hand with three fingers to perform orange peeling task.

2 Background

Robotic hands are a type of mechanical linkages, mainly programmable in which a common set of links spans a number of serial chains. The links of these type of manipulators are connected by joints which allow the links to undergo rotational or linear motion. The end effector is located at the end of the kinematic chain. Among the variety of robotic end-effectors, those generally defined as robotic hands are considered suitable not only for grasping, but also for some dexterous manipulation.

Researches which have been focused on grasping and manipulation, has been developed from rigid manipulators and simple grippers to dexterous robotic arms. These new hands are capable of grasping and manipulating a vast spectrum of objects and able to operate several tasks. Trends on robotic arms and hands are to become more dexterous, flexible in performance, light-weight, and easy to control.

When we talk about applications in robotic grasping and manipulation of grasped objects, we have to consider many characteristics of robotics like sensing, identification, learning and planning, etc. However, one field that has not attracted so much attention is the systematic methodology for the physical incarnation of the robotic end-effector. The design of end-effector robotic tools has focused on three different strategies [14], so that each of them lead to different design. Anthropomorphism, designing for grasping tasks, and designing for dexterous manipulation have been under the radar of researches.

The Anthropomorphism implies the ability of a hand to imitating the human hand motion. These hands are designed so that they could be humanized as much as possible. These class of robotic hands have restriction in their performance, and their level of dexterity is not suitable in some tasks. They are constrained in their design, but they are considered a straightforward solution for human environment and human manipulation task mapping [15], [16].

For the second strategy, overall, 33 different grasp types are found and arranged into the GRASP taxonomy. Grasps are sorted according to 1) opposition type, 2) the virtual

finger assignments, 3) type in terms of power, precision, or intermediate grasp, and 4) the position of the thumb [17]. Hands oriented to grasping tasks are usually simpler or under actuated; new efforts are being devoted to obtain under actuated or simple hands with some degrees of dexterity [18], [19].

On the other hand, hands for in-hand manipulation tend to be more complex, especially if a wide range of manipulation actions are targeted. However, most of them are anthropomorphic in design. For a current review on design efforts of anthropomorphic hands, we can address [20], [21], [22] and [23].

With respect to the foregoing discussion, in this thesis we are trying to design a hand for dexterous manipulation which is also called in-hand manipulation. Hence, for making this possible, first we need to review some aspects of kinematic and synthesis in robotic fields.

2.1 Kinematics

The kinematic background will be necessary for the synthesis process. As it is commonly accepted, “Kinematics is defined as the study of the motion, regardless of the force causing it and caused by it” [24]. The study of motion includes the study of derivatives of it, which are mainly velocity and acceleration [24]. The subject is modeled as a rigid body, which is “a set of particles such that the distance between them remains fixed” [24]. For defining a position of a rigid body, three positions will be needed and each of them includes three parameters. It means that nine parameters are needed. Since the distances between the positions must remain constantly, these three points are dependent to each other [24]. Because of these constraints, only six parameters are needed to define the position of the rigid body at a given instant of time. Figure 2-1 [24] shows a rigid body and the position of three of its points. The six needed parameters are given by d and $[R]$ which are the position vector of a point and the rotation matrix of the rigid body.

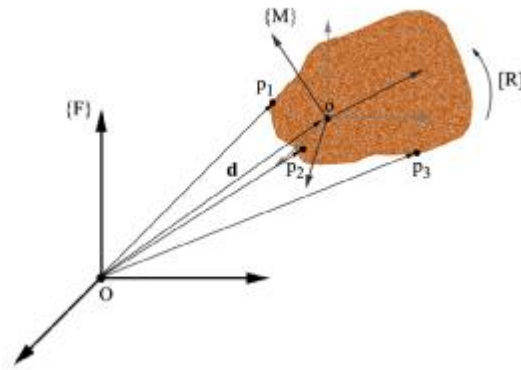


Figure 2-1: A rigid body [24]

2.1.1 Finite Displacement

In order to define the movement of a rigid body, a coordinate frame is attached to the rigid body. The coordinates of points of the rigid body are fixed in the attached coordinate frame. In Figure 2-2 [24], there are two different coordinates respect to frame F and frame M . A finite motion is called a displacement [24]. A displacement can be a translation, a rotation, or both of them.

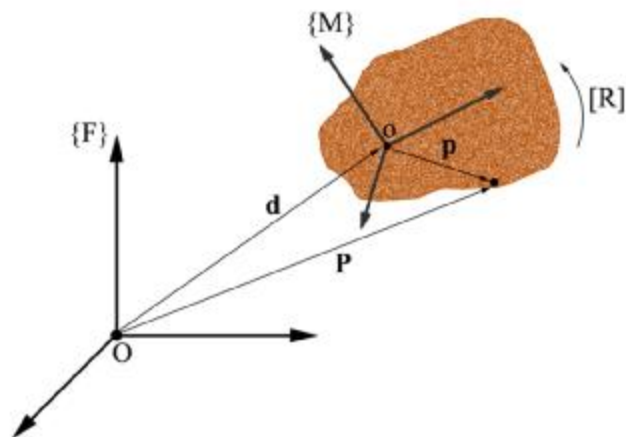


Figure 2-2: A rigid displacement [24]

2.1.2 Spatial Displacement

Spatial displacement is the combination of a rotation and a translation about an axis [24]. Equation 2.1 is a spatial displacement, which consists of rotation of θ about X-axis and a translation.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}. \quad (2.1)$$

The equation 2.1 can be written as equation 2.2

$$\mathbf{X} = [\mathbf{R}(\theta), \mathbf{d}] \mathbf{x} \quad (2.2)$$

where

\mathbf{X} is a coordination after the displacement

\mathbf{x} is a coordination before the displacement

$[\mathbf{R}(\theta), \mathbf{d}]$ is displacement matrix including translation and rotation.

Displacement matrix is expressed as equation 2.3.

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & \cos(\theta) & -\sin(\theta) & d_y \\ 0 & \sin(\theta) & \cos(\theta) & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (2.3)$$

The above matrix is the 4×4 homogeneous transformation matrix. The fourth column is added to include the translation vector in the linear operation.

2.1.3 Screw Displacement

A displacement in space can be written as a screw displacement. The screw displacement is given by rotation and translation about an axes [24]. The screw axis is represented by six parameters. For defining the direction and location of the line, four parameters are needed and two other parameters are needed for defining the rotation and slide values. Figure 2-3 [24] shows the representation of screw axis.

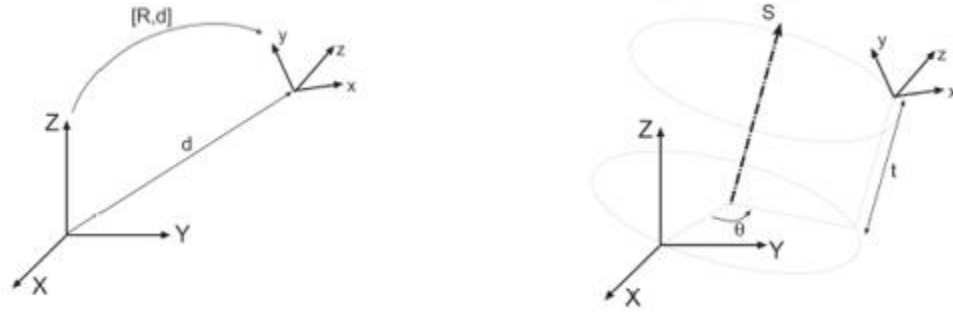


Figure 2-3: Representation of Screw axis [24]

Equation 2.4 to 2.6 show the screw displacements for X, Y and Z.

$$X(\alpha, a) = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2.4)$$

$$Y(\beta, b) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & b \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2.5)$$

$$Z(\theta, d) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.6)$$

2.1.4 Dual Quaternions

The homogeneous matrices method has been used to represent the rigid body displacement in three-dimensional space. The more efficient method in terms of computational speed, mathematical robustness and interpolation is the dual quaternions method. The main information, which is applied by the dual quaternions method, is the axis of the rotation and the translation vector and rotation angle. The Plucker coordinate of the given screw axis, S , is given by $S = s + \varepsilon s^0$, where $\varepsilon^2 = 0$ and, s is the direction of the axis and s^0 is the moment of the axis. Generally, the displacement of the dual quaternion is represented as below [24].

$$\hat{D}(\hat{\theta}) = \begin{pmatrix} \sin\left(\frac{\hat{\theta}}{2}\right) S_x \\ \sin\left(\frac{\hat{\theta}}{2}\right) S_y \\ \sin\left(\frac{\hat{\theta}}{2}\right) S_z \\ \cos\left(\frac{\hat{\theta}}{2}\right) \end{pmatrix}. \quad (2.7)$$

The conjugate of the dual quaternion is

$$\hat{D}^*(\hat{\theta}) = \begin{pmatrix} -\sin\left(\frac{\hat{\theta}}{2}\right) S_x \\ -\sin\left(\frac{\hat{\theta}}{2}\right) S_y \\ -\sin\left(\frac{\hat{\theta}}{2}\right) S_z \\ \cos\left(\frac{\hat{\theta}}{2}\right) \end{pmatrix}, \quad (2.8)$$

where, $\cos(\hat{\theta}/2) = \cos(\frac{\theta}{2}) + \epsilon (-\frac{d}{2} \sin(\frac{\theta}{2}))$ and $\sin(\frac{\hat{\theta}}{2}) = \sin(\frac{\theta}{2}) + \epsilon (\frac{d}{2} \cos(\frac{\theta}{2}))$

2.2 Kinematic Synthesis

From a kinematics point of view, a robot is a mechanical system that can perform a number of tasks involving movement under automatic control [24]. Using the combination of translations and rotations, a robot can move in a six dimensional space.

A mechanical system can be defined as a series of rigid links connected by joints. The degree of freedom of a robot is called mobility, which is the number of required independent parameters to specify the positions of a rigid body relative to a base frame. The degrees of the freedom of a robot is calculated using equation 2.9 [24].

$$M = 6(n - 1) - \sum_{i=1}^j (6 - f_i) , \quad (2.9)$$

where,

n is the number of links in the serial chain.

j is the number of joints in the serial chain.

f_i is the degree of freedom of the joint number i . i can be any number from 1 to j .

Kinematic synthesis is the process in which an articulated system created and calculated to apply a motion. The main method to design the equations for the dimensional kinematic synthesis is setting up the forward kinematics equations of the robot.

The Forward Kinematic equations in (2.10) calculate a relative displacement from initial position to final position. This displacement can be a combination of rotations and slide along the axis S for a serial kinematic chain with k revolute joints.

$$\hat{D}(\Delta\hat{\Theta}) = \cos\frac{\hat{\Psi}}{2} + \sin\frac{\hat{\Psi}}{2}S = e^{\frac{\Delta\hat{\theta}_1 S_1}{2}} e^{\frac{\Delta\hat{\theta}_2 S_2}{2}} \dots\dots\dots e^{\frac{\Delta\hat{\theta}_k S_k}{2}}. \quad (2.10)$$

2.2.1 Tree Topologies

A tree topology for a kinematic chain has a set of common joints spanning several chains, possibly in several stages, and ending in multiple end-effectors [25]. A branch of the hand is defined as a serial chain connecting the root node to one of the end-effectors, and a palm is a link that is ternary or above. The tree topology is represented as rooted a tree graph; the approach of Tsai [26] is followed, with the root vertex being fixed with respect to a reference system.

A multi-fingered hand is defined as a kinematic chain with several common joints - wrist, which is a fundamental part of the hand manipulation- spanning several branches, possibly in several stages. At the end of each branch, the end- effectors are located. They are the main elements whose motion or contact with the environment is being defined by the task; this can be generalized to consider other intermediate vertices of the topology. Open hands, that is, hands not holding an object, are represented as kinematic chains with a tree or hybrid topology.

A tree topology is represented by two arrays, which capture incidence and adjacency properties as well as information on the edges. Assume a numbering of the graph edges to define a parent-pointer array and a joint array. The length of both arrays is equal to number of edges of the tree graph after the reduction process is applied, that is, each edge, and each entry of the arrays, will correspond to a serial chain of the robotic hand.

The parent-pointer array implements the parent-pointer representation, where each element takes the value of the previous edge, the first edge being usually the one incident at the root vertex. The edges incident at the root vertex have no parent and therefore, take the value zero. Each element of the joint array contains the number and type of joints for each edge. If we are limited to revolute joints, then the joint array element will be equal to the number of joints for that edge.

Kinematic synthesis can be used in order to select and size a topology as a candidate hand design. The synthesis process for robotic hands has four main steps that are detailed below:

1. Task definition
2. Type or structural synthesis
3. Solvability calculations
4. Dimensional synthesis

After this process, we obtain a set of joints, their connectivity, and their relative position along a chain. Further steps of ranking, optimization and detailed design will be necessary to implement the candidates into functioning hands.

2.2.2 Task Definition

The task is the desired motion of the elements of the hand whose interaction with the environment is of interest. For a multi-fingered hand, a simultaneous motion of all fingertips or surface contacts, which could be any limb of the hand, is to be defined. For each fingertip or contact limb, a set of positions are defined as location plus orientation. Figure 2-4 shows a trajectory task for a right and left hand, respectively with two fingertips for each.

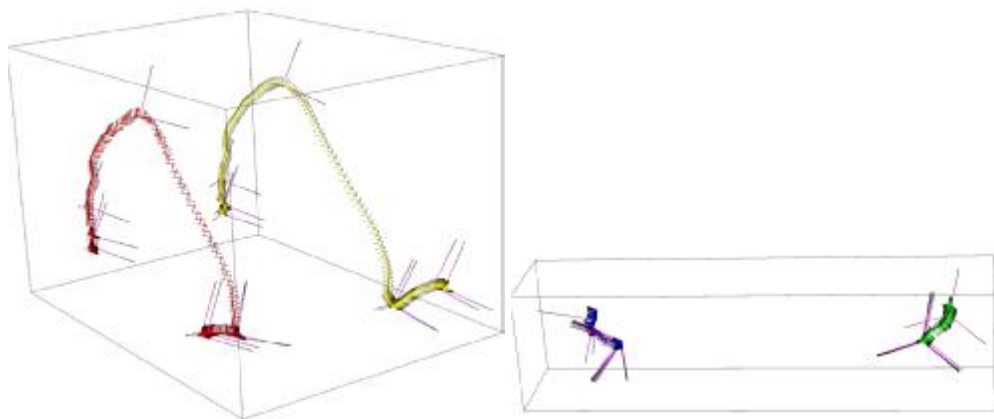


Figure 2-4: On the right; motion of the left hand, On the left; motion of right hand

2.2.3 Type Synthesis

Type synthesis, or structural synthesis, is the enumeration, selection and ranking of the kinematic chain topologies to be used as candidate designs. When applied to robotic hands, it implies the selection or calculation of the number of fingers, the number of joints at the wrist, the number of splits or branching, and the number of joints for the serial chain making each branch, as well as the type of joints to be used.

Given a simultaneous motion task for all fingertips, it is important to know how many, and what hand topologies are suited for the task. The number of candidate hand topologies of a certain type is usually very high and unbounded in some cases [27]. This number of suited topologies can be reduced if some additional constraints are added. At the end of the process, one or a few of these topologies will be selected for performing the additional design steps.

The conditions for considering a topology for the task are to have the same number of end-effectors as the task and to be solvable, according to the criteria defined in [28] and [27]. The set of suited topologies can be ranked according to other criteria, such as number of edges, number of splits, and number of joints per edge, among others.

2.2.4 Solvability

Solvability is defined as the ability of all combination of fingers to perform their relative tasks, and it is a condition that needs to be checked in order to be able to do the dimensional synthesis. It consists of checking what is the maximum number of positions that do not over constrain each root-to-end-effector(s) subgraph, and ensuring that each subgraph is less constrained than the overall graph. In this calculation, the subgraphs obtained by moving the root to each end-effector need to be included to account for relative motion between fingertips.

A hand is defined as *solvable* when it can be designed for a meaningful simultaneous task of all the fingertips or end-effectors, that is, a positive rational task with

at least two positions. Because some fingers may be over-constrained while others are under-constrained for a given topology, solvability needs to be checked systematically for all root-to-end-effector subgraphs of the hand, including those obtained when changing the root vertex to one of the end-effectors.

2.2.5 Dimensional Synthesis

In dimensional synthesis, positions of the joint axes are to be calculated, for the selected solvable topology and for the desired kinematic task. There are numerous techniques to address and solve the dimensional synthesis equations, regardless of the formulation; the output is the position of the joint axes at a reference configuration. This output is equivalent to the set of parameters defining the relative location and orientation between adjacent joints. The kinematic solution can be then used for the detailed design of the hand.

Kinematic dimensional synthesis is used to shape the new designs for robotic hands, able to grasp and/or manipulate in a given application. Dimensional synthesis has the candidate topology and the kinematic task as inputs.

3 Robotic Hand Design Methodology

As we discussed in previous chapter, a tree topology for a kinematic chain has a set of common joints splitting on several chains, ending in multiple end-effectors [25]. A typical example of such chains is wristed, multi-fingered hand. The kinematic synthesis of these topologies presents particular challenges that are different from those that appear in single serial chains or in closed-loop systems and compared to other topologies, they have not been so widely studied [29].

Kinematic analysis for applications in-modular robots and robotic hands can be found in [30] and [9], and dynamic analysis is found in [7]and [31]. Structural synthesis for multiple fingers with no wrist, considering grasping and manipulation requirements, are found in [10]. The first reference to kinematic synthesis of tree topologies is found in [11]. In particular, the kinematic synthesis of multi-fingered hands have been explored also in [12] and more extensively in [13].

In this chapter in order to present a complete design methodology for multi-fingered robotic hands, all steps of task definition, type synthesis and dimensional synthesis are explained.

3.1 Tree Topology

The tree topologies is the rooted tree graph consisting of vertices and edges, in which a vertex can be connected to one or more edges and the root vertex is fixed to a reference system [26]. In Figure 3-2 in which the root vertex is shown by double circle, shows the graph representation of a human hand and the Barrett hand[®], which we can see in Figure 3-1.



Figure 3-1: Human hand and Barrett hand®[32]

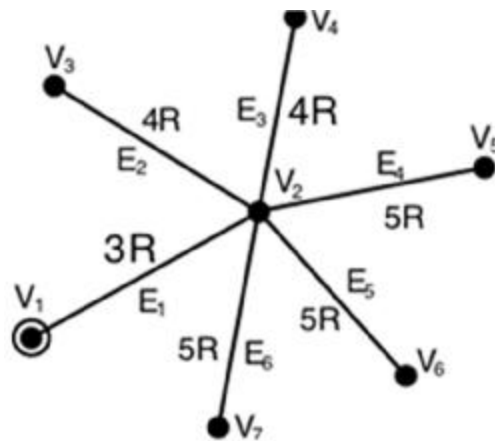


Figure 3-2: Human hand graph [32]

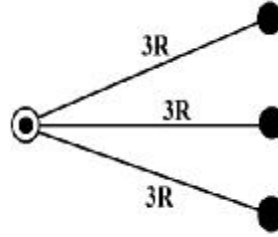


Figure 3-3: Barrett Hand® graph [32]

A process which is called reduction is used on graphs while formulating a synthesis problem. This procedure eliminates the loops and substitutes those with the most constrained path of corresponded loop so that the hand has a tree topology with links that are ternary or above [13]. For instance, see Figure 3-4.

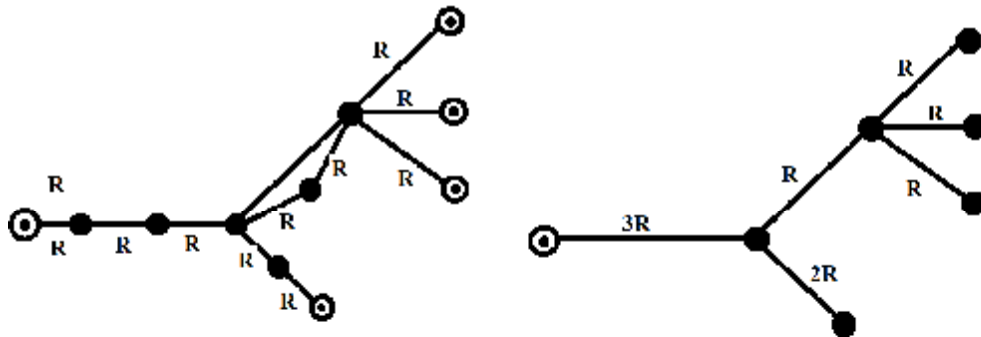


Figure 3-4: Left side, Original graph; Right side, reduced graph [27]

A common notation for tree topologies is $W-(B_1, B_2, \dots, B_b)$, with W being the common joints, the dash indicating a splitting stage and the branches inside the parenthesis. Its type and number of joints characterize each branch B_i . Usually, when revolute joints are the only type used, only the number of joints is indicated. Figure 3-5 shows the compacted graph for the $R-(R, R, R, R, R)$, that also is represented as $1-(1, 1, 1, 1, 1)$ and is a member of one-jointed hand family, with five fingers and also a fractal hand with two splitting stages and three branches per split [29].

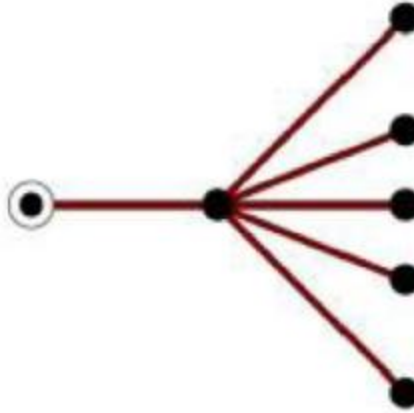


Figure 3-5: Tree graph for the 1-(1;1;1;1;1) robotic hand with one palm [33]

When we discuss about hand topology, two separate arrays are presented; *parent pointer array* and a *joint array*. Both of them have the same length equal to number of the edges. In the parent-pointer array, each edge element takes the value equal to the number of its previous edge (edges) in which the edge is connected to them. And number zero is being considered for the edges that are incident at the root vertex. On the other hand, the joint array represents the number and type of joints in each edge. Figure 3-6 represented the parent-pointer array and joint array of a five-fingered, two-palm hand topology hand indicated as; $p = \{0, 1, 1, 1, 3, 3, 3\}$ and $j = \{2, 2, 1, 2, 3, 3, 3\}$.

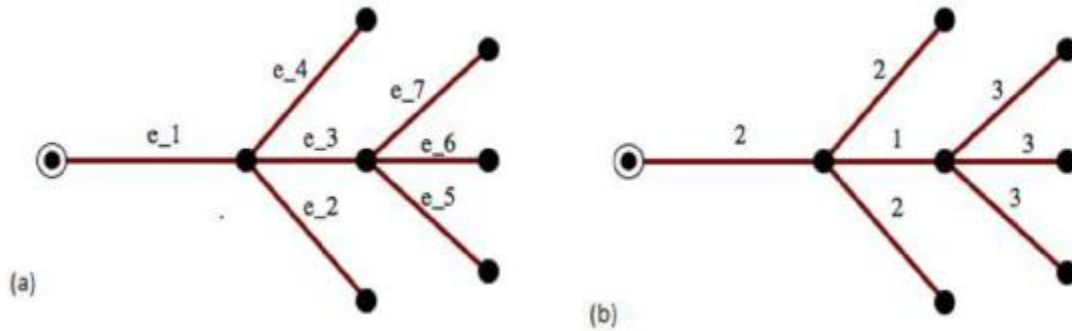


Figure 3-6: Left side, the numbering of the edge; Right side, the number of joints for each edge [29]

3.2 Kinematic Tasks Data Gathering Methods:

Kinematic tasks or simultaneous task for robotic hands indicates several positions for each fingertip to reach each other at the same time. A set of separated positions, velocities, or even accelerations can be defined for this.

The input motion data for the design can be acquired by using motion capture for a bimanual human performance. There is also an alternative approach, which is to utilize joint angles and coordinate frames from simulation data. In this thesis, we used both techniques.

3.2.1 Simulation Data

There are several technics for achieving suitable motion data in robotic field. One of them that we are going to use is Gazebo [34] in ROS[®] environment where the model of the robot is created first by URDF (Unified Robot Description Format) and the robotic system is implicated to perform the task in hand.

Gazebo is a 3D dynamic simulator with the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. While similar to game engines, Gazebo offers physics simulation at a much higher degree of fidelity, a suite of sensors, and interfaces for both users and programs.

Simulating a robot's controllers in Gazebo can be accomplished using ROS-control and a simple Gazebo plugin adapter.

Typical uses of Gazebo include:

- Testing robotics algorithms,
- Designing robots,

- Performing regression testing with realistic scenarios.

A few key features of Gazebo include:

- multiple physics engines,
- a rich library of robot models and environments,
- a wide variety of sensors,
- convenient programmatic and graphical interfaces.

In this research, model of the robot is created first by URDF (Unified Robot Description Format) and the robotic system is implicated to perform the task in hand. The coordinate frames of the robot for individual links are maintained in tree structure by TF library, which is updated through each time stamp. In addition, the individual joint angle movements are considered in each time stamp. A little later, we will talk in depth about this method.

3.2.2 Motion Capture for Kinematic Tasks

Another method used in this research is motion capture. Motion capture (Mo-cap for short) is the process of recording the movement of objects or people. This technique is used in military, entertainment, sports, medical applications, and for robotics. In this research, we used VICON[®] system.

Generally speaking, for a realistic and fast kinematic task definition, motion capture is usually used in which, the data is generated by capturing the desired motion, performed by human hand in a motion capture system. The motion of the hand is captured by infrared cameras obtaining the motion of the hand via reflective markers attached on the end-effectors.

A sample of a motion capture system is shown in **Figure 3-7**. In order to acquire the rigid-body motion a reference frame is also captured. **Figure 3-8** shows the experimental setup of markers used for the motion capturing of a task for a three-end-effectors hand.



Figure 3-7: A sample of a motion capture system (VICON®)



Figure 3-8: Experimental setup for motion capture

Nevertheless, there are some problems associated with using this technic, such as the sensors' noise or excessive richness of data [29]. Another issue we can address here is

overlapping of markers during the capturing motion, especially when hands are very close to each other. Moreover, reflection of cameras on each other limits the possible camera setups.

3.3 Structural Synthesis

In this stage of designing a robotic hand, numerous topologies of multi-fingered hands exist. Thus, we need to concentrate on the best one so that our requirement can be fulfilled. However, except for simple cases, matching a workspace shape to a topology is challenging and still far from being accomplished [30].

To continue the progress, some important factors have to be considered. We need to take care of number of positions and fingertips which are suitable for our hand design. Besides, solvability of the hand is another criterion which has to be taken into account. If a hand can be solved; then, it could be synthesized, and without over-constraining of its parts, reach the desired workspace [29]. In [9] some algorithms have been developed to find all possible topologies for a set of user-defined restrictions and a task in the SE(3) group of rigid motion [29]. By applying these extra conditions, we could bound a problem that may not be bounded otherwise [7].

The formula in (3-1) is applied to do the enumeration,

$$\sum_{i=1}^e j_i = \frac{(m-1)*6*b}{(m+3)}, \quad (3-1)$$

where m, b and e respectively are:

- the number of positions considered for the task,
- the number of end-effectors and
- the total number of edges of the graph, as user-defined inputs.

Here j_i are the joints for the i^{th} edge and $\sum_{i=1}^e j_i$ represent the total number of joints. The number of joints per edge must be between 1 and 5 for synthesis purposes, since a serial chain with length of 6 or higher does not impose any restriction on the motion [29].

All possible joint arrays of length e , satisfying (3-1), are defined, also all possible parent-pointer arrays (or structures) that meet the inputs, including the number of branches and number of edges are defined. Finally, all combinations of joint array and parent-pointer array are checked for solvability and the ones that match the criteria are outputted as candidate topologies. Table 1 represents two examples for the algorithm results [29].

Table 1 Examples of type synthesis

Example	m	b	e	Solvable topologies
Ex. 1	5	4	6	219
Ex. 2	21	5	8	34548

3.4 Dimensional Synthesis

In this stage, we need to find the position of a given topology joint axes, so that each of the end-effectors are capable to perform their designated displacements for the desired task [29]; for more information check [31].

In kinematic synthesis point of view for a hand topology, we consider b as number of branches of the hand, and n_e joint axes (S_i), and the sets of ordered indices B_j of joints belonging to the serial chain starting at the root and ending on end-effector j , for $j = 1, \dots, b$, and a set of m_p finite positions \hat{p}_{1k}^b , and m_v velocities \dot{P}_k^b , as the simultaneous task for each fingertip. By equating the forward kinematics equations of each branch to the relative

displacement of the corresponding fingertips and also, defining velocities for some of those task positions we would have,

$$F(S, \Delta\theta, \theta) = \begin{cases} \hat{P}_{1k}^j - \prod_{i \in B_j} e^{\frac{\Delta\theta_i^k}{2} S_i}, & \begin{matrix} k = 2, \dots, m_p \\ j = 1, \dots, b \end{matrix} \\ \dot{P}_k^j - \sum_{i \in B_j} \dot{\theta}_i^k S_i^k, & \begin{matrix} k = 2, \dots, m_v \\ j = 1, \dots, b \end{matrix} \end{cases}, \quad (3-1)$$

where S_i^k is the i^{th} joint axes when moved to position k . This yields a total of $6(m-1)b$ independent equations to be simultaneously solved. In most of the cases, numerical methods are used on this set to obtain a kinematic design [29].

3.4.1 Solvability

If a positive rational number of positions exists for which the exact dimensional synthesis yields a finite number of solutions, then a kinematic chain is defined as solvable. For serial chains, the solvability problem is trivial, while for tree topologies, the task sizing must be done so that the system of equations can be solved simultaneously while not over-constraining any of the branches [29].

Total number of $2^b - 1$ possible subgraphs for a given rooted tree graph exist, which consist of overall graph. As we already discussed in previous chapter, the maximum number of positions for the subgraph is given by

$$m_i = \frac{D_s^e E_i - D_c^n B_i}{D_{ee}^n B_i - D_j^e E_i} + 1, \quad (3-2)$$

where D_j^e is a $e \times 1$ vector containing the joint degrees-of-freedom for each edge of the contracted graph. D_s^e is the $e \times 1$ vector containing the number of structural parameters (four per joint in the general case) for each edge of the contracted graph. D_{ee}^e is the $b \times 1$ vector containing the degrees-of-freedom of the space of each end-effector, and D_c^n the $b \times 1$ vector with the number of additionally imposed constraints (if there exists any)

for each of the branches. B_i is the $b_i \times 1$ vector of ones corresponding to the branches for subgraph i , and E_i is an $e_i \times 1$ vector of ones for the edges in the subgraph. Those can be easily computed using the end-effector path matrix and the incidence matrix of the graph. In addition, we need to consider all different and non-isomorphic subgraphs that appear when exchanging the root node to each of the end-effectors [7].

The topology is solvable if

$$\text{a. } m \in \mathbb{Q}^+ \quad \text{b. } m \leq m_i, \forall m_i \in \mathbb{Q}^+, i \in S,$$

where m_i is the number of positions for exact synthesis for a subgraph i with $i \in S$ the set of all possible different end-effector subgraphs up to isomorphism, and m for the overall graph, considering all root-to-end-effector subgraphs, including root and end-effector switching.

A subgraph containing c branches and solvable for $m_i = m$ positions, can be solved separately, that eliminates exactly $6c(m-1)$ equations and the same number of unknowns, so that the rest of the graph can be solved afterwards [29].

3.4.2 Numerical Solution

Structural and the dimensional synthesis have been implemented in a solver. The structural synthesis follows a straightforward enumeration process while the dimensional synthesis is solved using a hybrid solver that returns a single solution for each run. This software is freely available at the project webpage [10].

4 Design of A Multi-Palm Hand for Assembly Tasks

4.1 Multi-Hand Tasks

A multi-hand task is defined as a task that requires the collaboration of more than one end-effector in order to be accomplished. Sometimes multi-hand tasks can be performed with a single gripper and the help of fixtures, feeding devices or positioning tables, in which the hand uses the environment for creating the missing forces. In other cases, two arm manipulators are used with their grippers performing collaborative tasks, similar to the human bimanual manipulation. Replacing a multi-hand task with a single hand requires a higher dexterity of the single hand.

4.2 In-Hand Manipulation and Dexterity

A possible definition of dexterity in robotic hand manipulation is the capability of changing the position and orientation of the manipulated object from a given reference configuration to a different one, arbitrarily chosen within the hand workspace [35], see also [36]. Human hand has been used for many robotic hand designs, however the human hand is not particularly dexterous for in-palm manipulation, and so are most of the current robotic hands. The dexterity has been investigated in the design of robotic hands starting with the work of [37] by calculating different metrics related to mobility when holding an object. Difficulties in the implementation of highly dexterous robotic hands include the complexity in the design, sensing and actuation.

4.3 Design of a Multi-Palm Robotic Hand

In manufacturing environments, robotic hands are used to automate processes and decrease human labors. Assembly tasks are some of the tasks more easily suited for robotic development.

4.3.1 Motivation and Task Selection

Usually they are multi hand tasks and more than one manipulator is needed to accomplish a complete assembly task. An example of such tasks is being performed by ABB FRIDA[®], which is a dual-arm manipulator [6]. One of its application is to use it for assembling emergency stop buttons that were tailored to be assembled by humans. The same robotic task with the similar switch is targeted in this research to demonstrate the introduced procedure. The example includes all the steps from motion capture and synthesis to hand implementation and testing in CAD environment. The task is addressed in Figure 4-1.

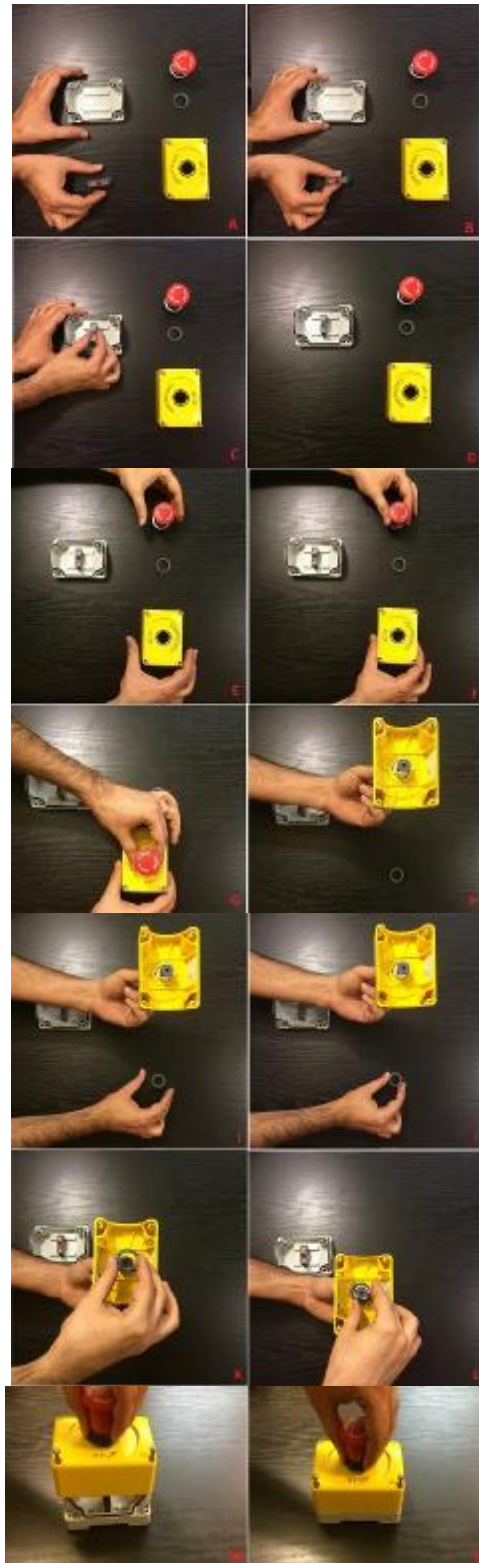


Figure 4-1: Assembly sub-Motions

4.3.2 Task-Based Mobility

Mobility of the hand is defined with a task-based strategy. To accomplish this step, the sub-tasks within the assembly task are identified and the required degrees of freedom for each sub-tasks are calculated. Fingertips and in-palm motion are associated to the relative motion of each subtask. This method allows the design of hands with dis-joint solvability, for which solvability criteria are applied to sub-trees related to designated sub-tasks.

Tasks-based mobility method is performed to define the required mobility of the hand. For this, the task is divided into five sub-tasks, please see **Table 2**. Free-hand position is a 6 DOF motion at wrist level that is the hand (not holding any object) change of position for grasping the object.

Object grasp is when the fingers already positioned for grasp perform a one DOF motion in secondary-palms-level to grasp the object. Object grasped position is also a six DOF motion in primary-palm level, for the hand change of position while it has grasped the object.

In inserting sub-task, the hand already positioned and holding one object, insert it in a single direction and is also a one DOF motion at primary-palm-level.

At the end, screw motion is the hand performing the screwing of one part which is a 2 DOF motion in primary-palm-level. In **Table 2**, required mobility for each sub-task and its level is addressed.

Table 2: Sub-tasks with the required mobility and mobility level.

Sub-task	Required mobility	Mobility level
Free-hand position	6	Wrist
Grasp	1	□ Secondary palms
Object grasped position	6	Primary palm
Insert	1	Secondary palms
Screw	2	Primary palm

4.3.3 Type Synthesis

In this section, the topology enumeration algorithm is used and depending on the design requirements of 4 fingertips, one primary and 2 secondary palms, the $p = \{0, 1, 1, 2, 2, 3, 3\}$ parent pointer array (Figure 4-2) is selected for the hand. After this, based on the required mobility at different levels which we can address in Table 2, the number of joints of the hand is chosen from the topologies provided by the structural synthesis algorithm.

Our candidate design has joint array of $j = \{2, 2, 2, 1, 2, 1, 2\}$. Figure 4-2 shows the graph of the topology and Figure 4-3 shows the kinematic sketch for this hand. After that, the hand is checked for the designated number of positions to make sure each sub-trees are solvable.

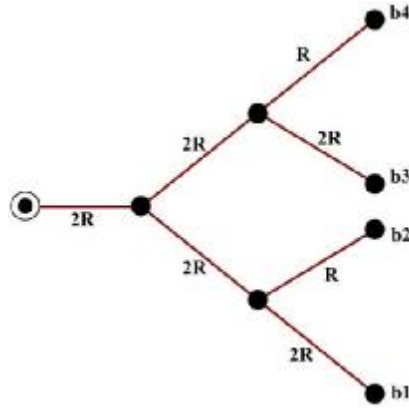


Figure 4-2: the tree topology earmarked for the hand.

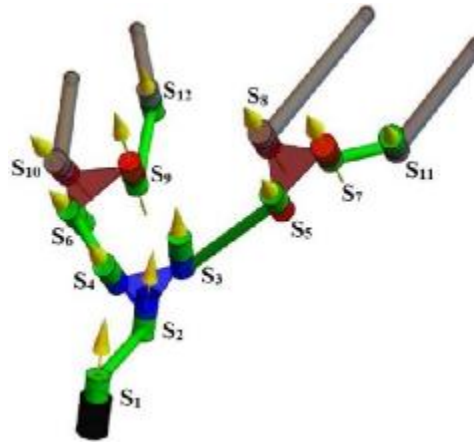


Figure 4-3 Kinematic sketch of the selected hand

4.3.4 Hand Design

For our design, we must have data as an input motion which can be achieved by using motion capture for bimanual performance. There is another approach in which we extract joint angles and coordinate frames from simulation data. In this research, both methods have been used. The motion simulation data is used to design the hand in a single synthesis step and the motion capture data is synthesized introducing a sequential procedure. Both methods are discussed in next sections.

4.3.4.1 Whole Hand Synthesis Method

Task definition (motion simulation)

In this method of data gathering, Gazebo [34] has been used where the model of the robot is created first by URDF and the robotic system is implicated to perform the task in hand. For the task of designing hands, we capture the motions in terms of frame transformations between the links in the fingertips of the robot and world. The transformations between the last frame of the robot where the hand is connected with manipulator and the fingertips of the robot are also calculated. The angles of each joints in the corresponding time stamps are also been stored as part of captured motions [38]. A URDF model of robot has been created with two *UR5* robotic arms as the manipulators and Robotic parallel grippers as the hands.

The components of assembly task namely the button, switch, base, cover and nut are also created in CAD design framework and also been imported in Gazebo simulation scenario (Figure 4-4). The dual arm robotic manipulator performs the desired task of fixing the switch in the base, button in the cover and attaching the button by bolt on the other side of the cover.

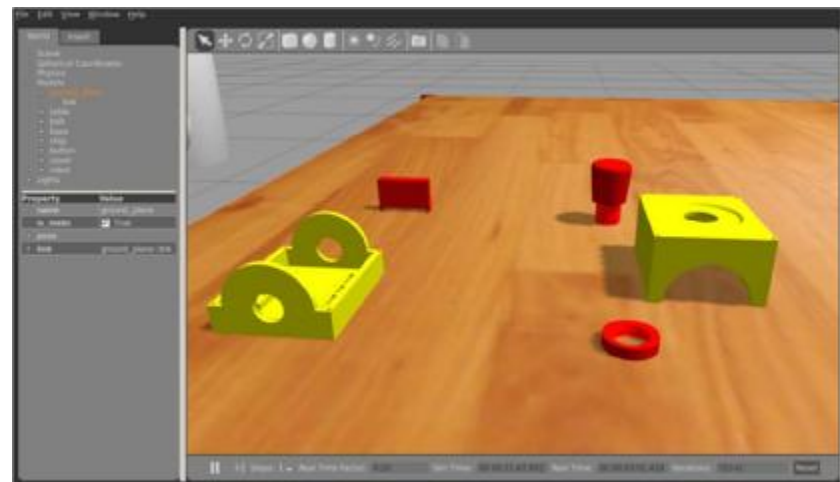
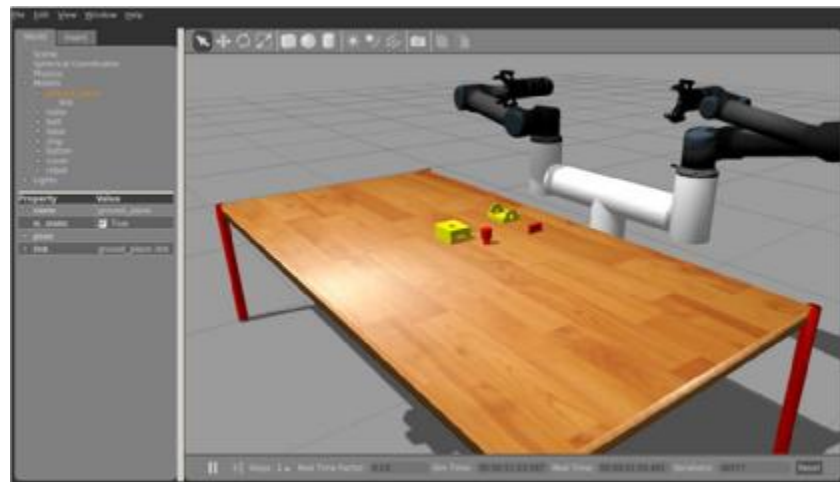
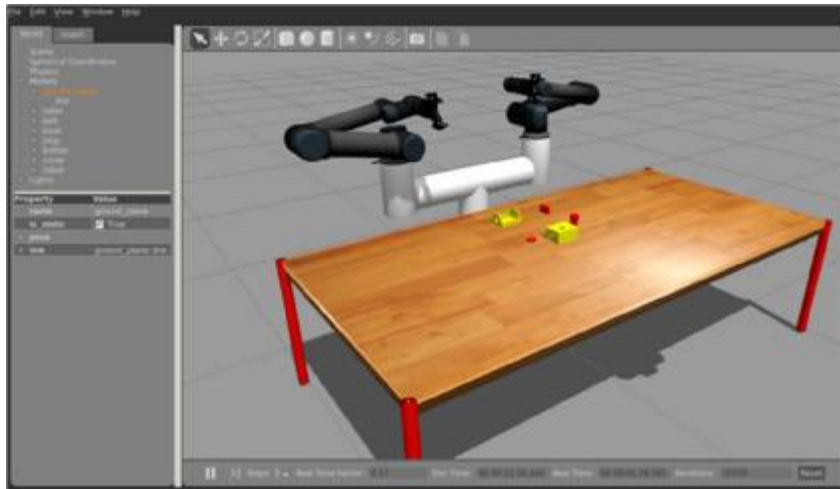


Figure 4-4: Task simulation framework [38].

Figure 4-5 shows the complete set of frames and the selected 13 frames to perform the approximate synthesis. Complete task, right, and selected synthesis frames, left. Frames corresponding to different fingertips are shown in different colors. We chose only 13 position tasks for the synthesis because the higher number of positions led to high and unacceptable synthesis error.

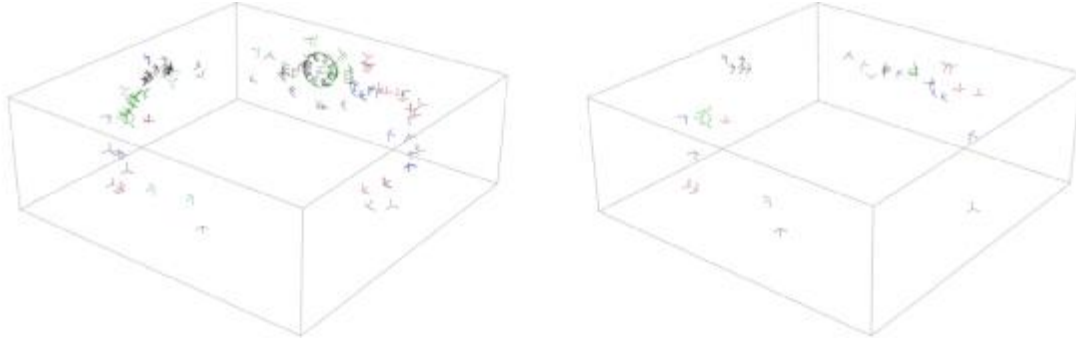


Figure 4-5: Assembly task and sub-tasks captured frames for each finger for the task

Dimensional synthesis

Design equations were created and solved using *ArtTreeKS* [28] for the approximate synthesis. Table 3 presents the resulting Plucker coordinates of the hand joints. The solution presented was obtained after 200 generations with a final error close to 19.

Table 3 Plucker coordinates of the hand with motion simulation data.

Joint Axis	Plucker coordinates
S_1	{0.718, -0.100, 0.688} , {80.674, 85.694, -71.668}
S_2	{0.957, -0.170, 0.233} , { 40.301, 104.025, -89.330}
S_3	{0.853, 0.419, -0.312} , { -78.545, 82.856, -103.278}
S_4	{0.626, -0.736, 0.258, } , {114.116, 79.806, -49.304}
S_5	{-0.404, 0.529, 0.747} , {113.737, -55.114, 100.489}
S_6	{0.612, 0.634, 0.473} , { -18.576, 60.822, -57.442}
S_7	{-0.213, -0.976, -0.048} , { -78.497, 81.901, -8.109}
S_8	{-0.253, -0.575, 0.778} , {-2.637, -27.805, -21.393}
S_9	{-0.569, -0.810, 0.145} , {0.553, -18.896, -103.575}
S_{10}	{0.516, -0.209, -0.830} , {91.832, 157.747, 17.349}
S_{11}	{-0.303, -0.683, 0.665} , {-27.398, -60.416, -74.499}
S_{12}	{-0.892, 0.297, -0.342} , {11.000, -1.296, -29.810}

Hand implementation

We utilized link-based optimization [39] on the synthesis. The hand overall link length has been considered as the objective function and the hand has been optimized for minimum overall length within upper and lower limits for each link and to avoid self-intersection during the motion.

After the optimization, the hand is created by utilizing the automatic modelling procedure developed in [40] and has been checked for all 13 positions reachability. Figure 4-6 shows the optimized hand at one of the positions and the created model of the hand, which was made using 3D printer.



Figure 4-6: On top, improved modelled hand in CAD environment; On bottom, 3D printed model

4.3.4.2 Sequential synthesis method

Task definition (Motion capture)

Motion capture system is used to capture the motion performed by both human hands. Due to the task complexity, and markers' occlusion, motion is divided into 4 sub motions;

- Sub Motion M1: positioning the switch in lower case (A-D in Figure 4-1),
- Sub Motion M2: positioning the button to the top case (E-H in Figure 4-1)
- Sub Motion M3: screwing the nut in the button (I-L in Figure 4-1),
- Sub Motion M4: positioning the top case over the lower case (M-N in Figure 4-1).

Figure 4-7 shows the configuration of the markers on one hand.



Figure 4-7: Markers configuration on the hand

We used the markers data for each finger to calculate the frames. Then, the frames were plotted and the importance priority motion selection was performed. For this, task position selection must be performed, by first dividing the motion into its sub-motions. For instance, a pick and place motions of an object can be divided into sub-motions of; open-hand motion, grasp, closed-hand motion and insert. After that the total number of positions are distributed between sub-motions and finally in each sub-motion the numbers of positions are selected based on that specific sub-motions characteristics.

In total, 40 frames were selected. Figure 4-8 to Figure 4-15 show the captured frames for each of the fingers in each sub-motion. Selected frames are also displayed in bigger frames and different color; so that, assembly sub-motions from M1 to M4 include right hand index, right hand thumb, left hand index and left-hand thumb, which are shown in yellow, red, green and blue respectively. Selected Frames are shown as bigger frames in magenta.

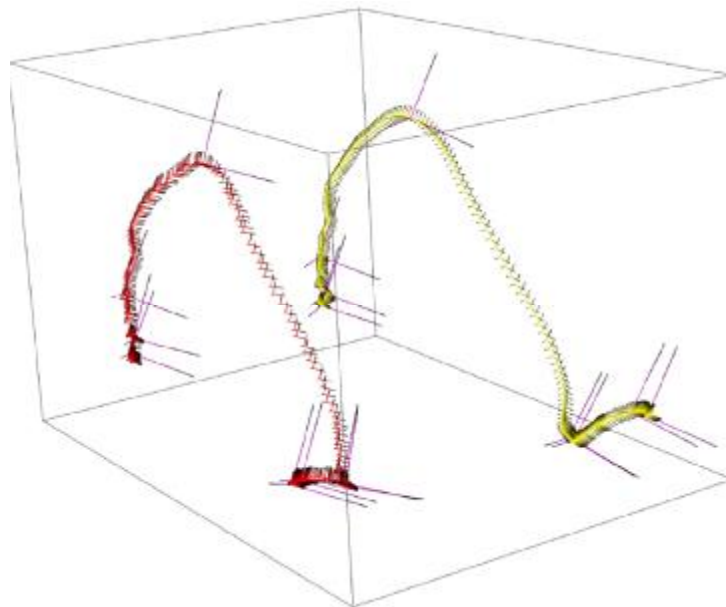


Figure 4-8: M1 for Right hand (Grasping the switch)

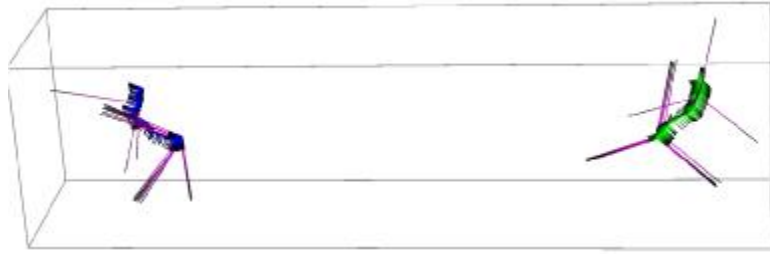


Figure 4-9: M1 for Left hand (Holding the base)

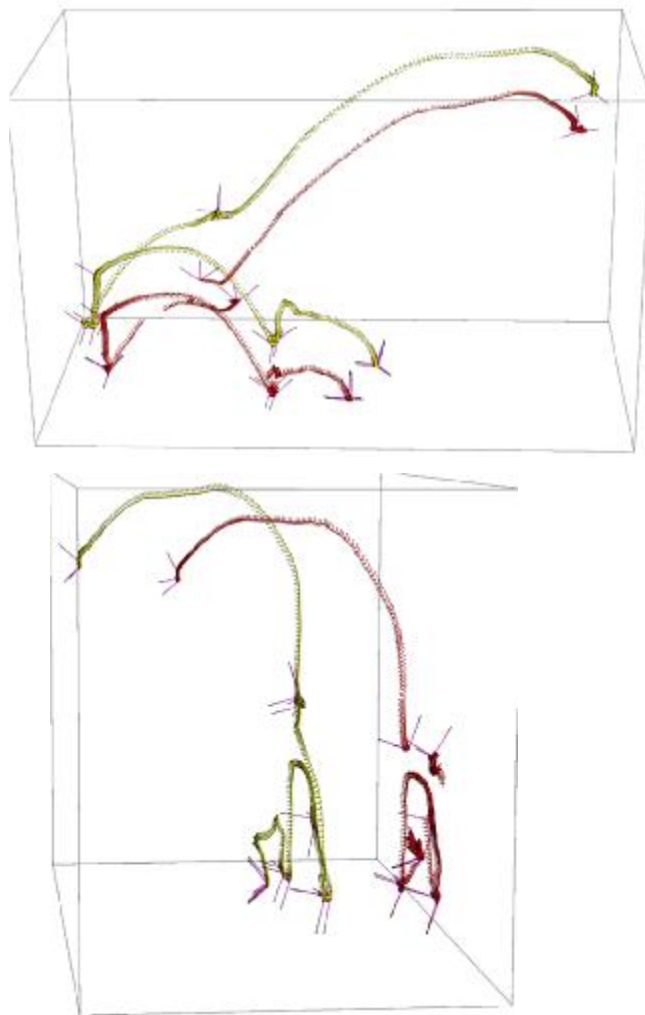


Figure 4-10: M2 for Right hand (Inserting the Button; in two views)

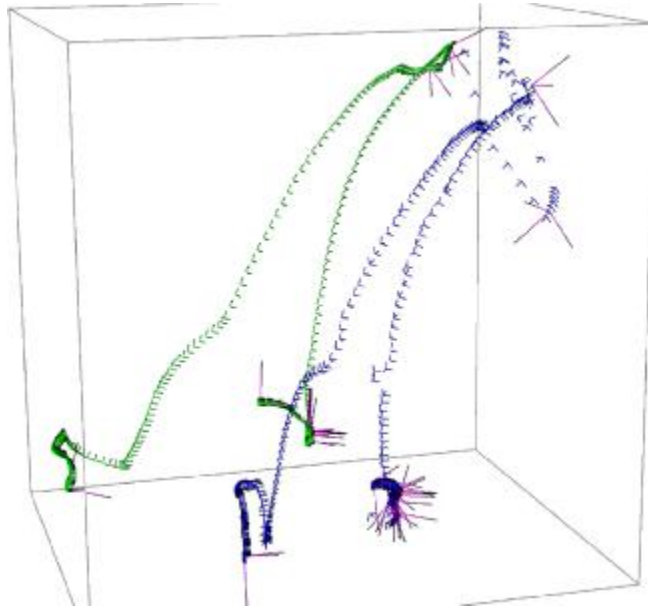


Figure 4-11: M2 for Left hand (Holding and rotating the Cover)

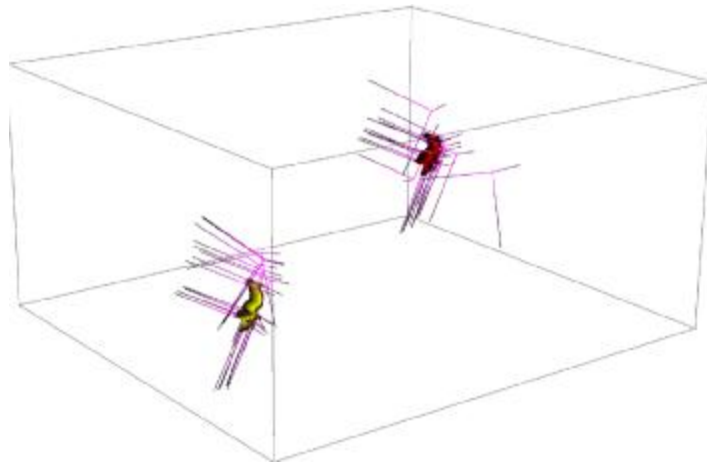


Figure 4-12: M3 for Right hand (Holding the upper- part)

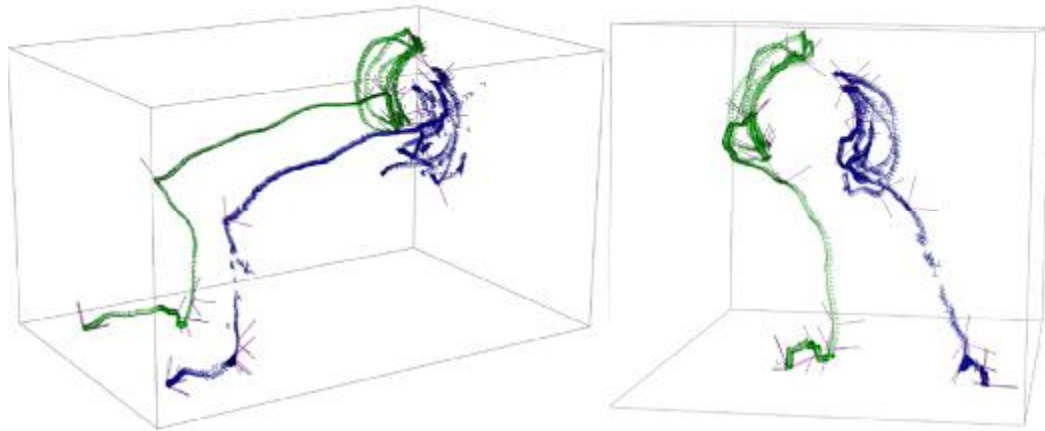


Figure 4-13: M3 for Right hand (Grasping and Screwing the nut; in two views)

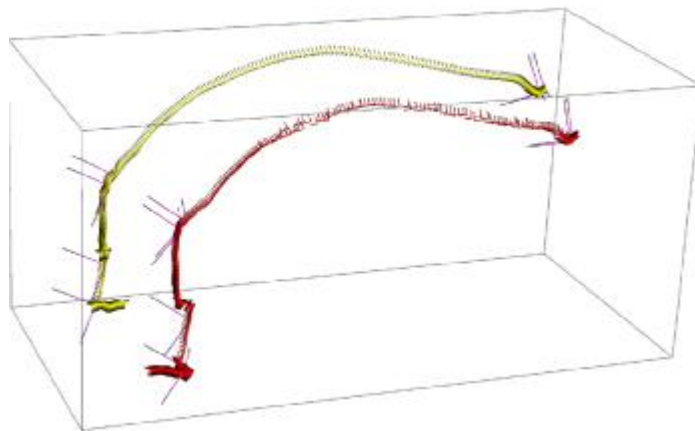


Figure 4-14: M4 for Right hand (Inserting upper part on the base)

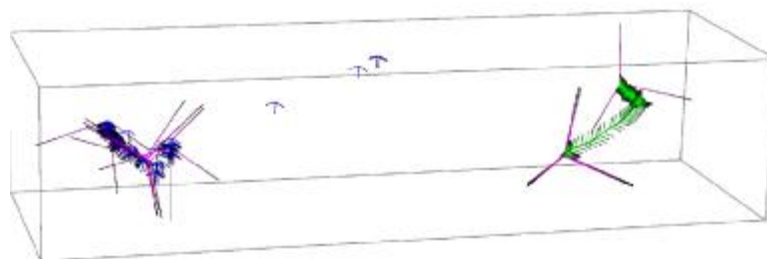


Figure 4-15: M4 for Left hand (Holding the base)

Dimensional synthesis

For dimensional synthesis, the sequential method was introduced. In this method, the hand is divided into 3 levels; grippers level, secondary palm level and wrist level. The levels are considered as simpler hands and then reduced to serial chains, and joints are solved from gripper toward the wrist, knowing the relative motions between the fingers. The results of each level are used for the next level that are solved using similar method.

Gripper Level (Depth 3 Hand design):

- The relative motions between the end effectors of each gripper were calculated.
- Gripper were synthesized as 3R chains for its axes and angles **Figure 4-16**.

Secondary Palm Level:

- The relative motions between the two grippers are used to size the joints at first palm as a 4R chain **Figure 4-16**.

Wrist Level:

- The wrist joints are dimensioned using the solution of the first two stages for any of the branches as a 2R chain.

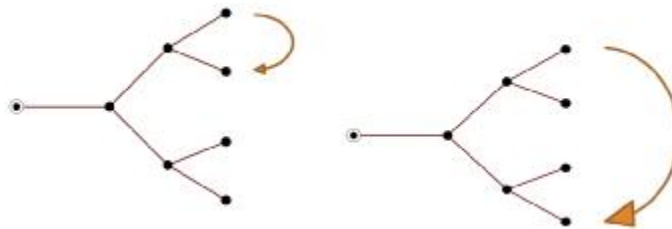


Figure 4-16 Relative motion levels.

Solution for the grippers

For solving the grippers, we consider the gripper joints and the joints involved in the kinematic chain between them before each secondary palm as a 2 – (2,1) hand that has a two-jointed wrist and two fingers, consisting of a single and double revolute joints. Figure 4-17 shows the tree graph for the hand and Figure 4-18 shows its kinematic sketch.

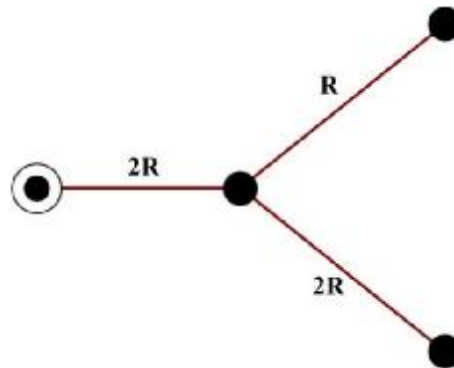


Figure 4-17: Graph of the 2-(2,1) hand.

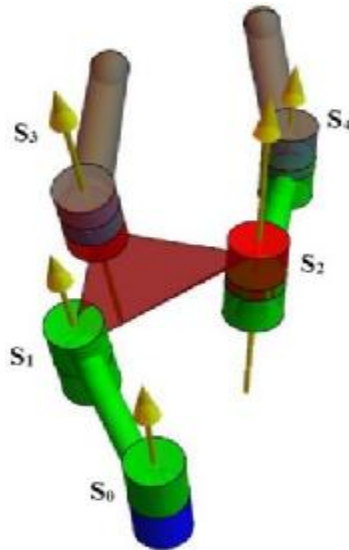


Figure 4-18: Kinematic sketch of the 2-(2,1) sub-hand.

In here, we reduce the rigid-body guidance problem for this hand to the spatial 3R chain. We come up with the design equations for each finger by level out the forward

kinematics equations for each branch to the desired position, [41]. As we proceed, j shows the screw displacement of rotation θ_i about joint axes S_j . So here, we use absolute displacements from the fixed frame associated to the root node,

$$\begin{aligned} e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} e^{\hat{S}_2\theta_2^i} e^{\hat{S}_4\theta_4^i} \hat{D}_0^1 &= \hat{P}_i^1 \\ e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} e^{\hat{S}_3\theta_3^i} \hat{D}_0^2 &= \hat{P}_i^2, \quad i = 1, \dots, n \end{aligned} \quad (4-1)$$

where \hat{D}_0^j represents the absolute displacement of the end-effector j at a reference configuration, for each branch j . Notice that $\hat{P}_i^j (\hat{D}_0^j)^{-1}$ is the relative displacement of branch j from the reference configuration up to position \hat{P}_i^j , expressed in the fixed frame, so that the set of equations can be written,

$$\begin{aligned} e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} e^{\hat{S}_2\theta_2^i} e^{\hat{S}_4\theta_4^i} &= \hat{P}_i^1 (\hat{D}_0^1)^{-1} \\ e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} e^{\hat{S}_3\theta_3^i} &= \hat{P}_i^2 (\hat{D}_0^2)^{-1}, \quad i = 1, \dots, n \end{aligned} \quad (4-2)$$

Taking the first task position as the reference configuration, and writing the forward kinematics of the common joints as a function of the rest of the chain and the task position, for each of the two branches,

$$\begin{aligned} e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} &= \hat{P}_i^1 (\hat{P}_1^1)^{-1} e^{-\hat{S}_4\theta_4^i} e^{-\hat{S}_2\theta_2^i}, \quad i = 1, \dots, n, \\ e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} &= \hat{P}_i^2 (\hat{P}_1^2)^{-1} e^{-\hat{S}_3\theta_3^i}, \quad i = 1, \dots, n; \end{aligned} \quad (4-3)$$

The following system of equations is created by imposing equality of the common joints at the wrist.

$$e^{-\hat{S}_4\theta_4^i} e^{-\hat{S}_2\theta_2^i} e^{\hat{S}_3\theta_3^i} = (\hat{P}_i^1 (\hat{P}_1^1)^{-1})^{-1} \hat{P}_i^2 (\hat{P}_1^2)^{-1}, \quad i = 1, \dots, n \quad (4-4)$$

Equation above is in agreement with the change of reference frame to the first end-effector in branch 1 which is making a serial chain who star at the first end-effector and going to the second one. The displacement corresponds to the relative displacements from the first position to position i , measured in the first end-effector frame. The set in Eq. (4-2)

corresponds to the equations for a spatial 3R chain. Thus, of both grippers, we are going to numerically solve for the axes S_2 , S_3 and S_4 .

We should notice that all the joint axes are in the first branch moving frame and have to be transformed to fixed frame for the final joint. By doing this, joints S_7 , S_8 and S_{11} of the first gripper and S_9 , S_{10} and S_{12} of the second gripper (belonging to the whole hand in Figure 4-3) are calculated. Respectively, as a result, joint axes are shown in Table 4 and Table 5.

Table 4 Gripper 1 joint axes.

Joint Axis	Plucker coordinates
S_7 □	{0.439055,0.37981,0.0936298},{1559.8,1889.81,-1333.7}
S_8	{0.417786,0.559087,-0.193594},{994.487,2069.7,-2380.89}
S_{11}	{-0.677864,0.633004,-0.337012},{-145.349,-80.4674,101.516}

Table 5 Gripper 2 joint axes.

Joint Axis	Plucker coordinates
S_9	{0.670164,-0.251773,0.386813},{-28.2967,54.9453,-79.7537}
S_{10}	{-0.518603,-0.680434,0.331417},{53.5611,-1098.28,2279.97}
S_{12}	{0.880307,0.0102963,-0.0292769},{-78.7641,11.9867,-10.8101}

Since we could be able to find the solution for our axes, now, we can be ready to proceed to the next level for solving for secondary palm.

Solution for the Secondary Palm

The wrist-to-secondary-palm-level joints are the joints that are located before the secondary palms. For solving this level, we take the grippers as rigid bodies that form the

end effectors of a 2 – (2,2) hand that has a two-jointed wrist and two fingers, consisting of double revolute joints. Figure 4-20 shows its kinematic sketch and Figure 4-19 shows the tree graph for the hand.

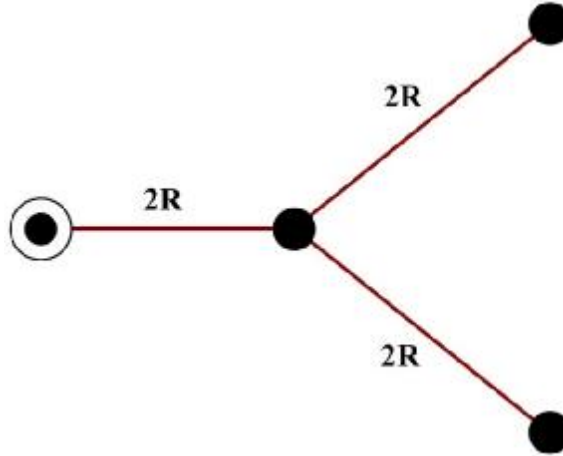


Figure 4-19: Graph of the 2-(2,2) hand.

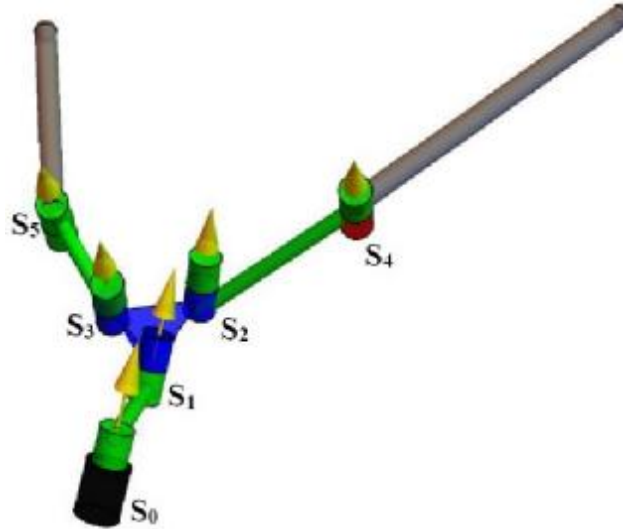


Figure 4-20: Kinematic sketch of the 2-(2,2) hand.

We are going to reduce the hand to a spatial 4R chain with the same methodology as the gripper level, so we have,

$$\begin{aligned} e^{\hat{S}_0 \theta_0^i} e^{\hat{S}_1 \theta_1^i} e^{\hat{S}_2 \theta_2^i} e^{\hat{S}_4 \theta_4^i} e^{\hat{S}_{6c} \theta_{6c}^i} e^{\hat{S}_{10c} \theta_{10c}^i} \hat{D}_0^1 &= \hat{P}_i^1, \\ e^{\hat{S}_0 \theta_0^i} e^{\hat{S}_1 \theta_1^i} e^{\hat{S}_3 \theta_3^i} e^{\hat{S}_5 \theta_5^i} e^{\hat{S}_{9c} \theta_{9c}^i} \hat{D}_0^2 &= \hat{P}_i^2, \quad i = 1, \dots, n, \end{aligned} \tag{4-5}$$

Now, by considering the same notation as the previous level, the set of equations can be written as,

$$\begin{aligned} e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} e^{\hat{S}_2\theta_2^i} e^{\hat{S}_4\theta_4^i} &= \hat{P}_i^1(\hat{D}_0^1)^{-1} e^{-\hat{S}_{10c}\theta_{10c}^i} e^{-\hat{S}_{6c}\theta_{6c}^i}, \\ e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} e^{\hat{S}_3\theta_3^i} e^{\hat{S}_5\theta_5^i} &= \hat{P}_i^2(\hat{D}_0^2)^{-1} e^{-\hat{S}_{9c}\theta_{9c}^i}, \quad i = 1, \dots, n. \end{aligned} \quad (4-6)$$

And like previous section, the forward kinematics of the common joints as a function of the rest of the chain can be written as,

$$e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} = \hat{P}_i^1(\hat{P}_1^1)^{-1} e^{-\hat{S}_{10c}\theta_{10c}^i} e^{-\hat{S}_{6c}\theta_{6c}^i} e^{-\hat{S}_4\theta_4^i} e^{-\hat{S}_2\theta_2^i}, \quad i = 1, \dots, n; \quad (4-7)$$

$$e^{\hat{S}_0\theta_0^i} e^{\hat{S}_1\theta_1^i} = \hat{P}_i^2(\hat{P}_1^2)^{-1} e^{-\hat{S}_{9c}\theta_{9c}^i} e^{-\hat{S}_5\theta_5^i} e^{-\hat{S}_3\theta_3^i}, \quad i = 1, \dots, n;$$

At the end, we set the common joints at the wrist equal, so we get,

$$e^{-\hat{S}_4\theta_4^i} e^{-\hat{S}_2\theta_2^i} e^{\hat{S}_3\theta_3^i} e^{\hat{S}_5\theta_5^i} = e^{\hat{S}_{6c}\theta_{6c}^i} e^{\hat{S}_{10c}\theta_{10c}^i} \hat{P}_1^1(\hat{P}_1^1)^{-1} \hat{P}_i^2(\hat{P}_1^2)^{-1} e^{-\hat{S}_{9c}\theta_{9c}^i}, \quad i = 1, \dots, n. \quad (4-8)$$

Equation (4-8) corresponds to the change of reference frame to the first end-effector in branch 1, which led to create a serial chain starting at the first end-effector and going to the second one. The displacement corresponds to the relative displacements from the first position to position i , measured in the first end-effector frame. The set corresponds to the equations obtained for a spatial 4R chain. We numerically solve for the axes S_2 , S_3 , S_4 and S_5 . The calculated are then moved to fixed frame for the S_3 , S_4 , S_5 and S_6 of the whole hand in Figure 4-3. Table 6 shows the resulting joint axes for the secondary palm level.

Table 6 Secondary level joint axes.

Joint axis	Plucker coordinates
S_3	{0.378453,0.32686,0.0123909},{-244.949,299.853,-428.496}
S_4	{0.192557,-0.54837,0.079682},{305.31,231.246,853.608}
S_5	{0.21006,-0.154842,0.139559},{606.315,452.786,-410.274}
S_6	{-0.206645,-0.211819,-0.00289715},{475.104,-457.7,-422.112}

Solution for the Wrist Level

We can solve the wrist axes joints as a spatial RR chain. In this step, each of two general branches after the first palm is taken as a rigid body for a $2 - (1,1)$ hand. Figure 4-21 shows the kinematic sketch of the hand.

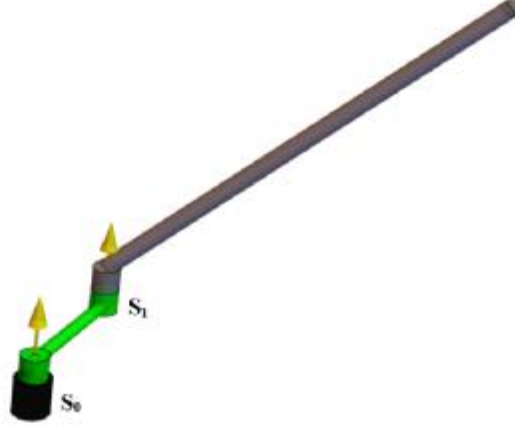


Figure 4-21: Kinematic sketch of the 2-(1,1) hand.

By using the same technic as the previous levels, we can have,

$$e^{\hat{S}_0 \theta_0^i} e^{\hat{S}_1 \theta_1^i} = \hat{P}_i^1 (\hat{P}_1^1)^{-1} e^{-\hat{S}_{10c} \theta_{10c}^i} e^{-\hat{S}_{6c} \theta_{6c}^i} e^{-\hat{S}_{4c} \theta_{4c}^i} e^{-\hat{S}_{2c} \theta_{2c}^i}, \quad i = 1, \dots, n, \quad (4-9)$$

Table 7 presents the resulting Plucker coordinates of wrist joints.

Table 7 Wrist level joint axes data.

Joint axis	Plucker coordinates
$S_1 \square$	$\{-0.0618612, 0.624443, -0.803186\}, \{-315.674, -288.089, -199.599\}$
S_2	$\{-0.0704628, -0.176724, -0.170489\}, \{92.1352, 330.7, -380.947\}$

All codes for this methodology in Mathematica[®] are available in Appendix A. The codes are partially imported to Matlab[®] and solutions were obtained there due to each software capabilities for solutions. Nonlinear least-squares solver was used with the maximum iteration of 5000 and tolerance of e^{-12} .

Hand Implementation

Then, the final hand is modelled using the automatic modelling procedure developed in [40] and checked for all 40 positions for reachability. Figure 4-22 shows the hand at some positions of the motion.



Figure 4-22: The hand Modelled in a 3D environment

Summery

In this chapter, a new design methodology of sequential synthesis is introduced and implemented. Then, the design was compared with the conventional synthesis method hand design. Sequential synthesis is better for design of highly dexterous hands because they can be designed to reach more positions. However, the design and the after-synthesis process of the hand design is an exhaustive task.

5 Design of a Dexterous Hand for An Orange Peeling Task

5.1 Motivation and Task Selection

Holding and operating on an object has been one of the desired tasks for many manipulators, like operational tasks in aerospace and medical science. The desired operation is often a very delicate and needs to exert precisely and so, requires a high level of accuracy and precision. As a good and familiar example, we could address the robots designed for precise surgical operations. One of the very well-known robots in this field is the Da Vinci Surgical System[®]. This system is powered by robotic technology that allows the surgeon's hand movements to be translated into smaller, precise movements of tiny instruments inside the patient's body [42]. See Figure 5-1



Figure 5-1 The da Vinci[®] Surgical System [42]

One of the most significant works of this system is the peeling of a grape placed on a table is demonstrated for this robot as an indication of its precision. Figure 5-2 shows the procedure.



Figure 5-2: da Vinci@System - Peeling A Grape [42]

This advanced surgical system became a motivation to concentrate on the similar task which is targeted peeling an orange while holding it with another hand.

5.2 Task Definition

In this research, motion capture system is used for data gathering from the motion performed by both human hands so that a knife is held by one hand and an orange had held by another hand. Our task requires implementation of two hands so that one holds the orange and the other hand performs the cutting with a knife. Two fingertips are used to hold the object, while the knife cuts through the orange peel. **Figure 5-3** shows the configuration of the markers on thumb and index fingers and on the knife holding by the other hand. Orange body and peel were also marked for the hand implementation and for their intersection with the hand to be avoided. The motion capture system screen monitoring the markers is also shown in **Figure 5-4** [43].

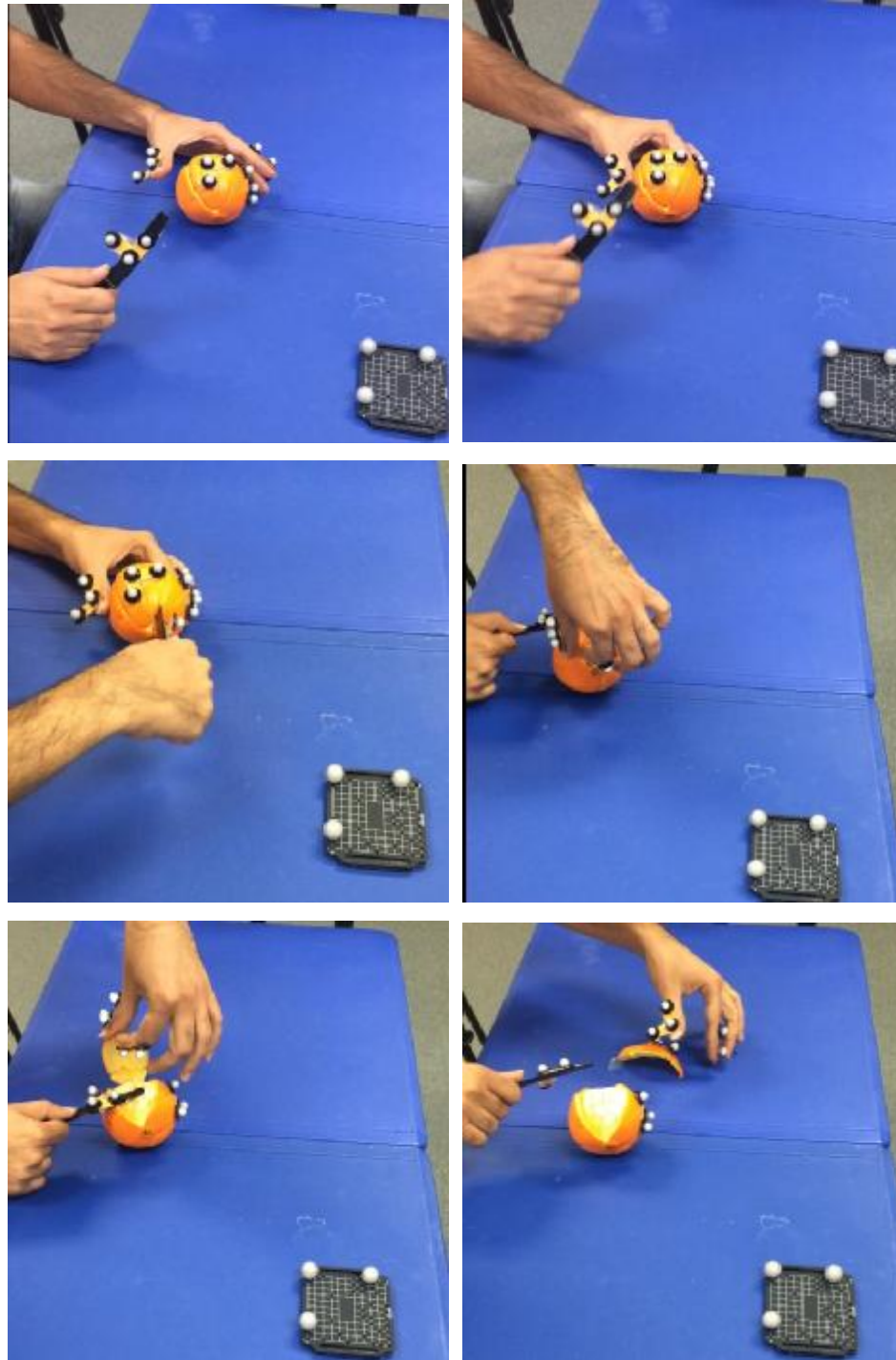


Figure 5-3: Motion capture steps

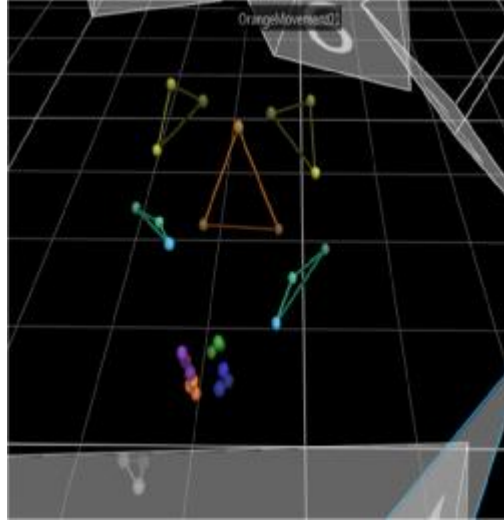


Figure 5-4: Graphic of motion capture [43]

Nearly 5000 frames were captured in 34 seconds of motion, however, in many of the captured frames, data were partially lost due to occlusion of the markers and in total 3154 remained.

5.3 Structural Synthesis

For structural synthesis, the final topology was selected with the parent pointer array of $\{0, 1, 1, 1\}$ and joint array $\{1, 3, 3, 3\}$, that is a hand with one joint in the wrist and three fingers, each consisting three joints. Figure 5-5 shows the graph representation of the hand and its kinematic sketch.

Two fingers are designated for holding and positioning the orange while the third branch is considered as the cutter branch to peel through the orange.

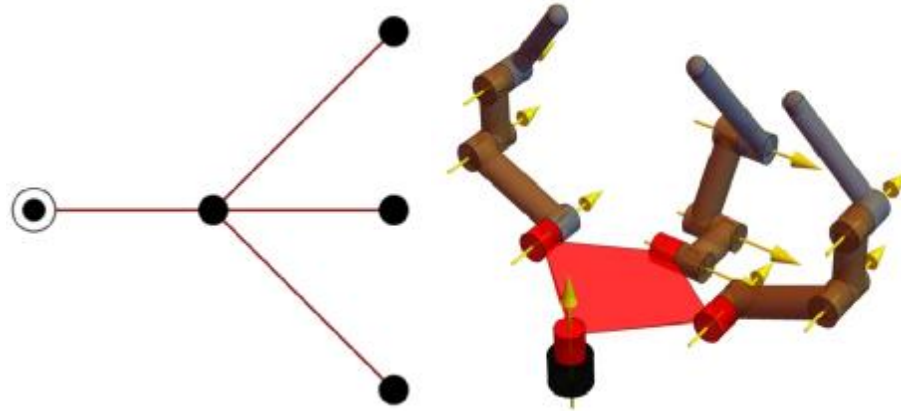


Figure 5-5 Selected hand topology for the task and kinematic sketch. [43]

By assuming that we grasp an object with the appropriate level of stability with soft fingers, the whole hand and object as a one system, has mobility of 4. This topology is solvable for,

$$m = 6 \text{ positions [27].}$$

One of the degrees of freedom belongs to the wrist rotation, and the other three are in-palm manipulation degrees of freedom.

5.4 Dimensional Synthesis

Design equations performed by utilizing ArtTeeKS [44] with respect to 6 positions we had calculated. Figure 5-6 shows the final hand model which has been tested to make sure it can reach the positions we wanted.

By looking at Table 8 we can see the Plucker coordinates of the hand joints.

Table 8 Plucker coordinates for the joints of the hand.

Joint Axis	Plucker coordinates
S_1	$\{0.52, -0.81, -0.27\}$, $\{46.72, 29.88, -180.89\}$
S_2	$\{0.53, -0.34, -0.78\}$, $\{196.78, 19.13, -143.00\}$
S_3	$\{-0.27, 0.96, -0.04\}$, $\{20.15, -0.14, 132.52\}$
S_4	$\{0.72, -0.60, 0.34\}$, $\{66.52, -375.97, -512.76\}$
S_5	$\{-0.71, -0.57, -0.42\}$, $\{50.40, 82.83, -28.12\}$
S_6	$\{-0.98, 0.21, 0.02\}$, $\{26.32, 90.24, 339.39\}$
S_7	$\{0.48, -0.86, -0.17\}$, $\{51.15, -1.03, -136.98\}$
S_8	$\{-0.22, 0.97, -0.11\}$, $\{62.32, 6.13, 173.44\}$
S_9	$\{-0.14, -0.93, 0.34\}$, $\{95.94, -43.65, -78.43\}$
S_{10}	$\{0.52, -0.85, 0.07\}$, $\{32.05, -29.03, -113.25\}$

5.5 Hand Implementation

For optimizing the hand, we used Link-based technic [39]. This method has been accomplished on the synthesis results with respect to this fact that we have to take overall shape of the hand and also collision avoidance into account.

Final hand is modelled following the automatic modeling procedure. The model of the optimized hand is presented in Figure 5-6.

For analyzing the hand movability including two fingers which hold the orange as an object, knife finger and also the orange system, we consider the hand as pointy fingertips with friction. Then, the hand is modelled using [45].



Figure 5-6: The final hand model

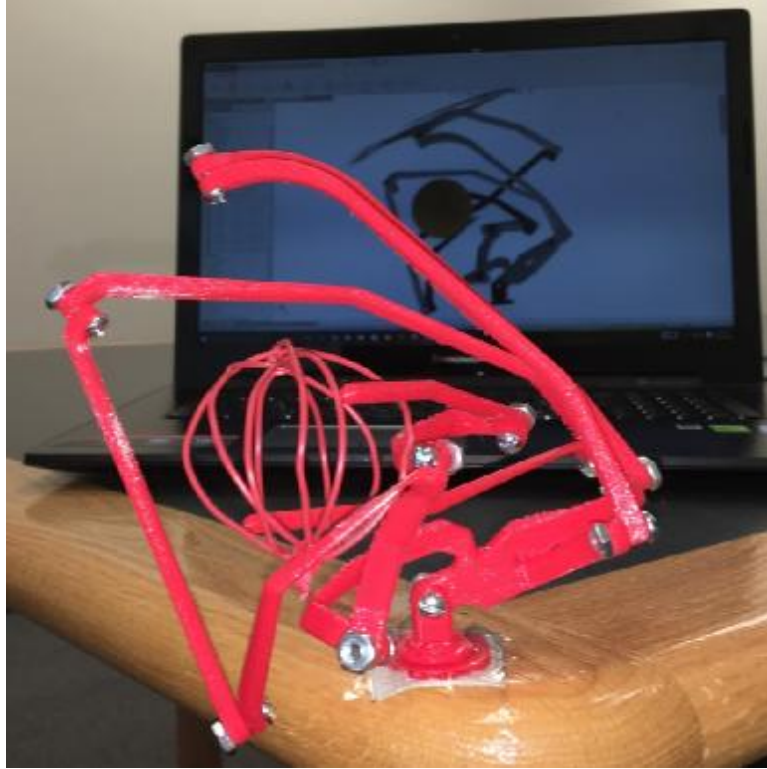


Figure 5-7: Prototype made by 3D printer

Summery

In this chapter, a single, dexterous hand has been created for an orange-peeling task which is usually required two robotic hands. The motion capture, synthesis, optimization and implementation of the hand were presented in this chapter as an indication that by kinematic synthesis method, a single dexterous hand can be designed and used for the tasks that are traditionally performed by collaborative manipulators.

6 Conclusion

In this research, our goal was to design highly dexterous multi-fingered robotic hands for performing multi-hand tasks that are accomplished usually by two or more robotic arm manipulators. We initiated our work with capturing the desirable motion by performing the task, bimanually with human hands and capturing the data by motion capture systems. Also, motion simulation has been utilized as an alternative approach of data gathering, for one of the hands [39]. We have focused on two hands for our designing target, assembly hand and orange peeler hand.

Process of designing the assembly hand consist of two distinct methods, a) Whole hand synthesis method and b) Sequential synthesis method. In the first approach, we used simulation data generated by Gazebo. 13 task positions have been selected for the synthesis, because the higher number of positions were resulting in high and inappropriate synthesis error. Subsequently, related equations were created and then solved using *ArtTreeKS*. Link-based optimization was performed on the synthesis results and the final hand was modelled and checked for positions' reachability and finally was prototyped with 3D Printer. In the second method, 40 tasks' positions have been selected and synthesis was performed with a new sequential method. The final hand was also optimized, modelled and checked for positions reachability. Obviously, the sequential synthesis is a better procedure for designing highly dexterous hands, because by utilizing this approach, the hand motion can be better adjusted to the task.

For designing the orange peeler hand, we used 6 positions of the data, gathered by motion capture. Design equations were created and solved by *ArtTeeKS* and the results were optimized with link based optimization method. The final hand was modeled, checked for reachability and was also prototyped with 3D Printer.

The designed hands were all capable to reach the positions they were designed for, however, at the moment they are not completely functional for the complete task due to,

- The forces were not contributed to the design.
- The motion of the designed hand is not defined to move between consequent positions, and usually the motion is not smooth. This will result in hands possible failure in grasp and manipulation of the object.
- The hand is guaranteed to reach the selected positions within the solver error limits, however, it is not guaranteed to reach other positions of the tasks, and there is a limit to the number of selected positions, due to increasing error of the solvers.
- Although post synthesis methods were applied, most of the hands final shape characteristics were already determined by the synthesis result. Usually, the joint axes are not in appropriate location and directions with respect to other axes and the object, and designing a functional and manufacturable hand between them is sometimes impossible.

In order to address these current short comings, the following is proposed as possible future research efforts,

- To ensure the grasp and manipulation of the object, contact forces can be contributed to the design, both to balance the external forces and to manipulate the object.
- It will be very effective to dictate some constraints for the motion between consecutive positions to the design equation, to ensure the motion smoothness.
- When the smoothness of motion is guaranteed, hands actuation can be performed for the next step of completely functional hand.
- For the design of manufacturable hands, it is necessary to dictate some constraints to the design in the synthesis level for the application optimization purpose. For instance, parallel axes will help the design modeling and manufacturability.
- In order to optimize reachability of the designed hand to the task positions, statistical definition of positions can be performed instead of the current method of exact position definitions.

REFERENCES

- [1] R. William, W. Harrison and D. Soloway, "Shared control of multiple-manipulator, sensor-based telerobotic systems," in *IEEE International Conference on Robotics and Automation*, 1997.
- [2] P. Dauchez, X. Delebarre, Y. Bouffard and E. Degoulange, "Task description for two cooperative manipulators," in *American Control Conference*, 1991.
- [3] Y. Liu, Y. Xu and B. M., "Cooperation control of multiple manipulators with passive joints," in *IEEE Transactions on Robotics*, 1999.
- [4] D. Sun and J. Mills, "Adaptive synchronized control for coordination of multi-robot assembly tasks," *IEEE Transactions on Robotics*, 2002.
- [5] A. R. e. a. N.C. Dafle, "Extrinsic Dexterity: In-Hand Manipulation with External Forces," in *ICRA*, Hong Kong, China., 2014.
- [6] A. Stolt, M. Linderöth, A. Robertsson and R. Johansson, "Robotic Assembly of Emergency Stop Buttons," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, 2013.
- [7] J. a. B. E. Garcia de Jalon, "Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge.," *Springer-Verlag*, 1994.
- [8] A. Jain, "Graph-theory roots of spatial operators for kinematics and dynamics," in *In Proc. of the 2010 International Conference on Robotics and Automation*, 2010.
- [9] C. S. A. H. K. Tischler, "Kinematic chains for robot hands - 1. orderly number synthesis," in *Mechanism and Machine Theory*, 1995, p. 1193–1215.
- [10] J. a. T. L. Lee, "Structural synthesis of multifingered hands," *Journal of Mechanical Design*, p. 272–276, 2002.
- [11] E. M.-N. F. P.-G. A. Smo-Serra, "Design of non-anthropomorphic robotic hands for anthropomorphic tasks," in *In Proc. of the 2011 ASME International Design Engineering Technical Conferences, IDETC-CIE*, 2011.
- [12] E. P.-G. A. M. H. a. R. N. Smo-Serra, "Design of multi fingered robotic hands for finite and infinitesimal tasks using kinematic synthesis," *In Advances in Robot Kinematics*, 2012.
- [13] E. a. P.-G. A. Smo-Serra, "Kinematic synthesis using tree topologies," *Mechanism and Machine Theory*, p. pp. 94–113., 2014.

- [14] S. S. A. V. a. A. R.-. d. M. T. Mason, "Generality and simple hands," *International Journal of Robotics Research*, 2010.
- [15] J. G. M. Martin, "Design of a fully modular and back drivable dexterous hand.," *The International Journal of Robotics Research*, p. 783–798, 2014.
- [16] E. Mattar, "A survey of bio-inspired robotics hands implementation: New directions in dexterous manipulation.," *Robotics and Autonomous Systems*, 2013.
- [17] T. Feix, J. Romero, H. Schmiedmayer, A. M. Dollar and D. Kragic, "The GRASP Taxonomy of Human Grasp Types," in *IEEE*, 2015.
- [18] L. e. a. Odhner, "A compliant, underactuated hand for robust manipulation.," *The International Journal of Robotics Research*, p. 736–752, 2014.
- [19] M. S. C. N. A. S. J. Quigley, "Mechatronic design of an integrated robotic hand.," *The International Journal of Robotics Research*, p. 706– 720, 2014.
- [20] V. B. R. S. V. J. J. Santos, "Design of Artificial Hands: A Review.," in *The Human Hand as an Inspiration for Robot Hand Development*, Switzerland, Springer, 2014., p. 219–244.
- [21] J. N. R. N. C.-r. M. Parada Puig, "A methodology for the design of robotic hands with multiple fingers," *Advanced Robotic Systems*, p. 177–184, 2008.
- [22] D. Sterusand and C. Turner, "A design methodology based process for robotic gripper design," in *In Proc. of the 2011 ASME International Design Engineering Technical Conferences, IDETC-CIE, Washington D.C*, 2011.
- [23] A. Goccarlie and P. Allen, "Data-driven optimization for underactuated robotic hands.," in *In Proc. of the 2010 International Conference on Robotics and Automation*, Anchorage, Alaska, 2010.
- [24] A. Perez Garcia, *Kinematics of Robotics*.
- [25] J. M. Selig, *Geometric Fundamentals of Robotics*, New York: Springer Verlag, 2004.
- [26] L. W. Tsai, *Mechanism Design: Enumeration of Kinematic Structures According to Function*, Boca Raton: CRC Press, 2001.
- [27] A. Makhal and A. Perez-Gracia, "Solvable multi-fingered hands for exact kinematic synthesis," in *Advances in Robot Kinematics*, Ljubljana, Slovenia, June 2014.
- [28] E. Smo-Serra and A. Perez-Gracia, "Kinematic synthesis using tree topologies," *Mechanism and Machine Theory*, vol. 72 C, p. 2014, 94–113.
- [29] A. Perez-Gracia, "SYNTHESIS OF MULTI-FINGERED ROBOTIC HANDS," in *Pan-American Congress on Computational Mechanics - PANAQM 2015*, Buenos Aires, 2015.

- [30] S. Stramigioli, Modeling and IPC control of interactive mechanical systems – A coordinate-free approach, London: Springer-Verlag, 2001.
- [31] J. Abhinandan, "Graph-theory roots of spatial operators for kinematics and dynamics," in *International Conference on Robotics and Automation*, Anchorage, Alaska, USA, 2010.
- [32] N. Hassanzadeh and A. Perez-Gracia, "Dimensional Synthesis of Wristed Binary Hands," *ASME Journal of Mechanisms and Robotics*, 2015.
- [33] A. Perez-Gracia, "Dimensional synthesis of one-jointed, multi-fingered hands," in *3rd IFToMM Symposium on Mechanism Design for Robotics*, Aalborg, Denmark, 2-4 June, 2015.
- [34] IsaacSaito, "ROS.org," last edited 2015-07-19 14:11:05 by IsaacSaito. [Online]. Available: <http://wiki.ros.org/tf>.
- [35] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity," *IEEE Transactions on Robotics*, vol. 16(6), 2000.
- [36] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *The 15th International Conference on Advanced Robotics*, 2011.
- [37] J. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands," *Journal of Mechanical Design*, vol. 105(1), p. 35–41, 1983.
- [38] N. Hassanzadeh, R. Movassagh Khaniki, A. Makhal and A. Perez-Gracia, "Design of a multi-palm robotic hand for assembly tasks," in *Proceedings of the ASME 2016 International Design Engineering Technical Conferences &*, Charlotte,, 2016.
- [39] Y. Yihun, K. Bosworth and A. and Perez-Gracia, "Link-based performance optimization of spatial mechanisms," *Journal of Mechanical Design*, vol. 136(12).
- [40] N. Hassanzadeh, X. He and A. Perez-Gracia, "A design implementation process for robotic hand synthesis".
- [41] A. Perez-Gracia and J. M. McCarthy, "Kinematic synthesis of spatial serial chains using clifford algebra exponentials," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 220(7), p. 953–968.
- [42] "<http://www.davincisurgery.com/da-vinci-surgery/da-vinci-surgical-system/>," The da Vinci®Surgical System, [Online]. Available: <http://www.davincisurgery.com/>.
- [43] N. Hassanzadeh, R. Movassagh khaniki and A. Perez-Gracia, "Design of a dexterous hand for a multi-hand task," in *ROMANSY21-Robot Design, Dynamics and Control*, Springer International Publishing, 2016.

- [44] A. Tamimi and A. Perez-Gracia, "A robotic hand synthesis process for general tree topologies," in *submitted to: Robotic Science and Systems*, Rome, Italy, July 13-17, 2015.
- [45] E. Staffetti and F. Thomas, "Analysis of rigid body interactions for compliant motion tasks using the grassmann-cayley algebra," in *In Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.
- [46] B. Batbold, Y. Yihun, J. S. Wolper and A. Pérez-Gracia, "Exact workspace synthesis for rocr linkages," 2013 IFToMM Computational Kinematics Workshop, , 2013.
- [47] "Design of robotic hands webpage.," 15 01 2015. [Online]. Available: <http://robotics.engr.isu.edu/indexRH.html>. .
- [48] L. Tsai and B. Roth, "A note on the design of revolute-revolute cranks," *Mechanism and Machine Theory* 8, p. 23–31, 1973.
- [49] A. Makhal and A. Perez-Gracia., "Task data association for robot design," in *submitted to Robotic Science and Systems*, Rome, Italy, 2015.
- [50] P. Dauchez, X. Delebarre, Y. Bouffard and E. Degoulange, "Ask description for two cooperative manipulators," in *In Proc. of the American Control Conference*, 1991.
- [51] Y. Liu, Y. Xu and M. Bergerman, "Cooperation control of multiple manipulators with passive joints," *IEEE Transactions on Robotics*, vol. 5(2), 1999.
- [52] D. Sun and J. K. Mills, "Adaptive synchronized control for coordination of multirobot assembly tasks," *IEEE Transactions on Robotics*, vol. 18(4), 2002.

Appendix A

Mathematica[®] Codes for Assembly Hand Using Sequential Synthesis

```

SetDirectory[NotebookDirectory[]];
<<Matrix.m;
<<ToMatlab.m;
Data Entry
M1. 4,556 Grasp Base & Insert switch
M2. 557,1383 Grasp Cover & Insert Button
M3. 1384,2818 Grasp Nut and scrow it on Button
M4 2819,3584 Assamble the cover on the base
ImCord=Flatten[Import["CapturedMotion31.xlsx"],1];
ImCord2=Flatten[Import["CapturedMotion32.xlsx"],1];
PositionsRL1=Table[i,{i,4,556}];
PositionsRL2=Table[i,{i,557,1383}];
PositionsRL3=Table[i,{i,4,1438}];PositionsRL4=Table[i,{i,1439,2204}];

```

M1

Right Hand-M1

```

RHThQ1M1=Table[ImCord[[PositionsRL1[[i]],{3,4,5}]],{i,1,Length[PositionsRL1]};
RHThQ2M1=Table[ImCord[[PositionsRL1[[i]],{6,7,8}]],{i,1,Length[PositionsRL1]};
RHThQ3M1=Table[ImCord[[PositionsRL1[[i]],{9,10,11}]],{i,1,Length[PositionsRL1]};
;
RHThQ4M1=Table[ImCord[[PositionsRL1[[i]],{12,13,14}]],{i,1,Length[PositionsRL1]};
];

RHInQ1M1=Table[ImCord[[PositionsRL1[[i]],{15,16,17}]],{i,1,Length[PositionsRL1]};
];
RHInQ2M1=Table[ImCord[[PositionsRL1[[i]],{18,19,20}]],{i,1,Length[PositionsRL1]};
];
RHInQ3M1=Table[ImCord[[PositionsRL1[[i]],{21,22,23}]],{i,1,Length[PositionsRL1]};
];
RHInQ4M1=Table[ImCord[[PositionsRL1[[i]],{24,25,26}]],{i,1,Length[PositionsRL1]};
];
RHInQ5M1=Table[ImCord[[PositionsRL1[[i]],{27,28,29}]],{i,1,Length[PositionsRL1]};
];
XRHInM1=Table[Normalize[RHInQ3M1[[i]]-
RHInQ4M1[[i]]],{i,1,Length[PositionsRL1]};
ZRHInM1=Table[Normalize[(RHInQ3M1[[i]]-RHInQ5M1[[i]])/(RHInQ4M1[[i]]-
RHInQ5M1[[i]])],{i,1,Length[PositionsRL1]};
YRHInM1=Table[-Normalize[XRHInM1[[i]]/(ZRHInM1[[i]]),{i,1,Length[PositionsRL1]};
ORHInM1=Table[Orthogonalize[{XRHInM1[[i]],YRHInM1[[i]],ZRHInM1[[i]]}],{i,1,Length[PositionsRL1]};
DRHInM1=Table[Append[Transpose[{ORHInM1[[i,1]],ORHInM1[[i,2]],ORHInM1[[i,3]],RHInQ1M1[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL1]};
dqERHInM1=Table[HM2dq[DRHInM1[[i]]],{i,1,Length[PositionsRL1]};
plotRHInM1=generatePlot[dqERHInM1,Black,3];
RHThM1=Table[Normalize[RHThQ2M1[[i]]-
RHThQ3M1[[i]]],{i,1,Length[PositionsRL1]};
ZRHThM1=Table[Normalize[(RHThQ2M1[[i]]-RHThQ4M1[[i]])/(RHThQ3M1[[i]]-
RHThQ4M1[[i]])],{i,1,Length[PositionsRL1]};
YRHThM1=Table[-Normalize[XRHThM1[[i]]/(ZRHThM1[[i]]),{i,1,Length[PositionsRL1]};
ORHThM1=Table[Orthogonalize[{XRHThM1[[i]],YRHThM1[[i]],ZRHThM1[[i]]}],{i,1,Length[PositionsRL1]};
DRHThM1=Table[Append[Transpose[{ORHThM1[[i,1]],ORHThM1[[i,2]],ORHThM1[[i,3]],RHThQ1M1[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL1]};
dqERHThM1=Table[HM2dq[DRHThM1[[i]]],{i,1,Length[PositionsRL1]};
plotRHThM1=generatePlot[dqERHThM1,Blue,3];
Show[plotRHInM1,plotRHThM1]

```

Left Hand-M1

```

LHThQ1M1=Table[ImCord[[PositionsRL1[[i]],{30,31,32}]],{i,1,Length[PositionsRL1]};
];
LHThQ2M1=Table[ImCord[[PositionsRL1[[i]],{33,34,35}]],{i,1,Length[PositionsRL1]};

```

```

];
LHThQ3M1=Table[ImCord[[PositionsRL1[[i]],{36,37,38}]],{i,1,Length[PositionsRL1]}
];
LHThQ4M1=Table[ImCord[[PositionsRL1[[i]],{39,40,41}]],{i,1,Length[PositionsRL1]}
];

LHInQ1M1=Table[ImCord[[PositionsRL1[[i]],{42,43,44}]],{i,1,Length[PositionsRL1]}
];
LHInQ2M1=Table[ImCord[[PositionsRL1[[i]],{45,46,47}]],{i,1,Length[PositionsRL1]}
];
LHInQ3M1=Table[ImCord[[PositionsRL1[[i]],{48,49,50}]],{i,1,Length[PositionsRL1]}
];
LHInQ4M1=Table[ImCord[[PositionsRL1[[i]],{51,52,53}]],{i,1,Length[PositionsRL1]}
];
LHInQ5M1=Table[ImCord[[PositionsRL1[[i]],{54,55,56}]],{i,1,Length[PositionsRL1]}
];
XLHInM1=Table[Normalize[LHInQ3M1[[i]]-
LHInQ4M1[[i]]],{i,1,Length[PositionsRL1]}];
ZLHInM1=Table[Normalize[(LHInQ3M1[[i]]-LHInQ5M1[[i]])⊗(LHInQ4M1[[i]]-
LHInQ5M1[[i]])],{i,1,Length[PositionsRL1]}];
YLHInM1=Table[-Normalize[XLHInM1[[i]]⊗ZLHInM1[[i]]],{i,1,Length[PositionsRL1]}];
OLHInM1=Table[Orthogonalize[{XLHInM1[[i]],YLHInM1[[i]],ZLHInM1[[i]]}],{i,1,Length[PositionsRL1]}];
DLHInM1=Table[Append[Transpose[{OLHInM1[[i,1]],OLHInM1[[i,2]],OLHInM1[[i,3]],LHInQ1M1[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL1]}];
dqELHInM1=Table[HM2dq[DLHInM1[[i]]],{i,1,Length[PositionsRL1]}];
plotLHInM1=generatePlot[dqELHInM1,Black,3];
XLHThM1=Table[Normalize[LHThQ2M1[[i]]-
LHThQ3M1[[i]]],{i,1,Length[PositionsRL1]}];
ZLHThM1=Table[Normalize[(LHThQ2M1[[i]]-LHThQ4M1[[i]])⊗(LHThQ3M1[[i]]-
LHThQ4M1[[i]])],{i,1,Length[PositionsRL1]}];
YLHThM1=Table[-Normalize[XLHThM1[[i]]⊗ZLHThM1[[i]]],{i,1,Length[PositionsRL1]}];
OLHThM1=Table[Orthogonalize[{XLHThM1[[i]],YLHThM1[[i]],ZLHThM1[[i]]}],{i,1,Length[PositionsRL1]}];
DLHThM1=Table[Append[Transpose[{OLHThM1[[i,1]],OLHThM1[[i,2]],OLHThM1[[i,3]],LHThQ1M1[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL1]}];

dqELHThM1=Table[HM2dq[DLHThM1[[i]]],{i,1,Length[PositionsRL1]}];
plotLHThM1=generatePlot[dqELHThM1,Blue,3];
Show[plotLHInM1,plotLHThM1]

```

To choose

```

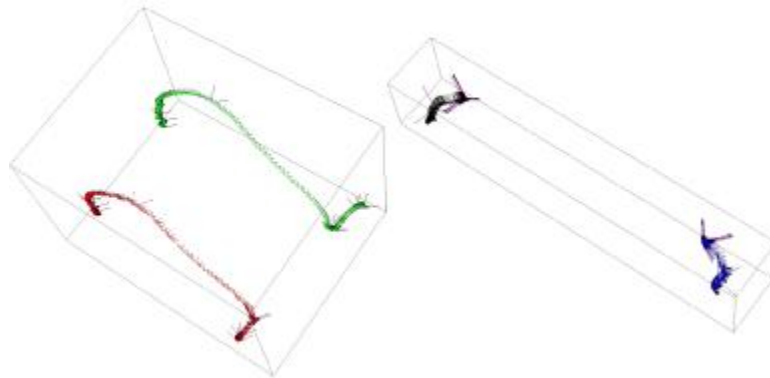
TL=Length[DRHInM1];
M1ch1RH={60,160,200,240,300,370,400,500};
M1ch1LH={60,160,200,240,300,370,400,500};
SL=1;EL=TL;
dqERHInM1c=Table[HM2dq[DRHInM1[[i]]],{i,SL,EL}];
dqERHThM1c=Table[HM2dq[DRHThM1[[i]]],{i,SL,EL}];
plotRHInM1c=generatePlot[dqERHInM1c,Green,3];
plotRHThM1c=generatePlot[dqERHThM1c,Red,3];
Show[plotRHInM1c,plotRHThM1c];
dqELHInM1c=Table[HM2dq[DLHInM1[[i]]],{i,SL,EL}];
dqELHThM1c=Table[HM2dq[DLHThM1[[i]]],{i,SL,EL}];
plotLHInM1c=generatePlot[dqELHInM1c,Black,3];
plotLHThM1c=generatePlot[dqELHThM1c,Blue,3];
Show[plotLHInM1c,plotLHThM1c];
dqERHInM1cS=Table[HM2dq[DRHInM1[[M1ch1RH[[i]]]]],{i,1,Length[M1ch1RH]}];
dqERHThM1cS=Table[HM2dq[DRHThM1[[M1ch1RH[[i]]]]],{i,1,Length[M1ch1RH]}];
plotRHInM1cS=generatePlot[dqERHInM1cS,Magenta,10];
plotRHThM1cS=generatePlot[dqERHThM1cS,Magenta,10];
Show[plotRHInM1cS,plotRHThM1cS,plotRHInM1c,plotRHThM1c];
dqELHInM1cS=Table[HM2dq[DLHInM1[[M1ch1RH[[i]]]]],{i,1,Length[M1ch1RH]}];

```

```

dqELHThM1c=Table[HM2dq[DLHThM1[[M1ch1RH[[i]]]],{i,SL,Length[M1ch1RH]}];
plotLHInM1cS=generatePlot[dqELHInM1c,Magenta,10];
plotLHThM1cS=generatePlot[dqELHThM1c,Magenta,10];
Show[plotLHInM1cS,plotLHThM1cS,plotLHInM1c,plotLHThM1c]

```



M2

Right Hand-M2

```

RHThQ1M2=Table[ImCord[[PositionsRL2[[i]],{3,4,5}]],{i,1,Length[PositionsRL2]};
RHThQ2M2=Table[ImCord[[PositionsRL2[[i]],{6,7,8}]],{i,1,Length[PositionsRL2]};
RHThQ3M2=Table[ImCord[[PositionsRL2[[i]],{9,10,11}]],{i,1,Length[PositionsRL2]};
;
RHThQ4M2=Table[ImCord[[PositionsRL2[[i]],{12,13,14}]],{i,1,Length[PositionsRL2]};
];

RHInQ1M2=Table[ImCord[[PositionsRL2[[i]],{15,16,17}]],{i,1,Length[PositionsRL2]};
];
RHInQ2M2=Table[ImCord[[PositionsRL2[[i]],{18,19,20}]],{i,1,Length[PositionsRL2]};
];
RHInQ3M2=Table[ImCord[[PositionsRL2[[i]],{21,22,23}]],{i,1,Length[PositionsRL2]};
];
RHInQ4M2=Table[ImCord[[PositionsRL2[[i]],{24,25,26}]],{i,1,Length[PositionsRL2]};
];
RHInQ5M2=Table[ImCord[[PositionsRL2[[i]],{27,28,29}]],{i,1,Length[PositionsRL2]};
];
XRHInM2=Table[Normalize[RHInQ3M2[[i]]-
RHInQ4M2[[i]]],{i,1,Length[PositionsRL2]};
ZRHInM2=Table[Normalize[(RHInQ3M2[[i]]-RHInQ5M2[[i]])⊗(RHInQ4M2[[i]]-
RHInQ5M2[[i]])],{i,1,Length[PositionsRL2]};
YRHInM2=Table[-Normalize[XRHInM2[[i]]⊗ZRHInM2[[i]]],{i,1,Length[PositionsRL2]};
ORHInM2=Table[Orthogonalize[{XRHInM2[[i]],YRHInM2[[i]],ZRHInM2[[i]]}],{i,1,Length[PositionsRL2]};
DRHInM2=Table[Append[Transpose[{ORHInM2[[i,1]],ORHInM2[[i,2]],ORHInM2[[i,3]],RHInQ1M2[[i]]},{0,0,0,1}],{i,1,Length[PositionsRL2]};
dqERHInM2=Table[HM2dq[DRHInM2[[i]]],{i,1,Length[PositionsRL2]};
plotRHInM2=generatePlot[dqERHInM2,Black,3];
XRHThM2=Table[Normalize[RHThQ2M2[[i]]-
RHThQ3M2[[i]]],{i,1,Length[PositionsRL2]};
ZRHThM2=Table[Normalize[(RHThQ2M2[[i]]-RHThQ4M2[[i]])⊗(RHThQ3M2[[i]]-
RHThQ4M2[[i]])],{i,1,Length[PositionsRL2]};
YRHThM2=Table[-Normalize[XRHThM2[[i]]⊗ZRHThM2[[i]]],{i,1,Length[PositionsRL2]};
ORHThM2=Table[Orthogonalize[{XRHThM2[[i]],YRHThM2[[i]],ZRHThM2[[i]]}],{i,1,Length[PositionsRL2]};
DRHThM2=Table[Append[Transpose[{ORHThM2[[i,1]],ORHThM2[[i,2]],ORHThM2[[i,3]],RHThQ1M2[[i]]},{0,0,0,1}],{i,1,Length[PositionsRL2]};
dqERHThM2=Table[HM2dq[DRHThM2[[i]]],{i,1,Length[PositionsRL2]};

```



```
plotRHThM2=generatePlot[dqERHThM2,Blue,3];
Show[plotRHInM2,plotRHThM2]
```

Left Hand-M2

```
LHThQ1M2=Table[ImCord[[PositionsRL2[[i]],{30,31,32}]],{i,1,Length[PositionsRL2]}];
LHThQ2M2=Table[ImCord[[PositionsRL2[[i]],{33,34,35}]],{i,1,Length[PositionsRL2]}];
LHThQ3M2=Table[ImCord[[PositionsRL2[[i]],{36,37,38}]],{i,1,Length[PositionsRL2]}];
LHThQ4M2=Table[ImCord[[PositionsRL2[[i]],{39,40,41}]],{i,1,Length[PositionsRL2]}];

LHInQ1M2=Table[ImCord[[PositionsRL2[[i]],{42,43,44}]],{i,1,Length[PositionsRL2]}];
LHInQ2M2=Table[ImCord[[PositionsRL2[[i]],{45,46,47}]],{i,1,Length[PositionsRL2]}];
LHInQ3M2=Table[ImCord[[PositionsRL2[[i]],{48,49,50}]],{i,1,Length[PositionsRL2]}];
LHInQ4M2=Table[ImCord[[PositionsRL2[[i]],{51,52,53}]],{i,1,Length[PositionsRL2]}];
LHInQ5M2=Table[ImCord[[PositionsRL2[[i]],{54,55,56}]],{i,1,Length[PositionsRL2]}];

XLHInM2=Table[Normalize[LHInQ3M2[[i]]-
LHInQ4M2[[i]]],{i,1,Length[PositionsRL2]}];
ZLHInM2=Table[Normalize[(LHInQ3M2[[i]]-LHInQ5M2[[i]])⊗(LHInQ4M2[[i]]-
LHInQ5M2[[i]])],{i,1,Length[PositionsRL2]}];
YLHInM2=Table[-Normalize[XLHInM2[[i]]⊗ZLHInM2[[i]]],{i,1,Length[PositionsRL2]}];
OLHInM2=Table[Orthogonalize[{XLHInM2[[i]],YLHInM2[[i]],ZLHInM2[[i]]}],{i,1,Length[PositionsRL2]}];
DLHInM2=Table[Append[Transpose[{OLHInM2[[i,1]],OLHInM2[[i,2]],OLHInM2[[i,3]],LHInQ1M2[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL2]}];
dqELHInM2=Table[HM2dq[DLHInM2[[i]]],{i,1,Length[PositionsRL2]}];
plotLHInM2=generatePlot[dqELHInM2,Black,3];
XLHThM2=Table[Normalize[LHThQ2M2[[i]]-
LHThQ3M2[[i]]],{i,1,Length[PositionsRL2]}];
ZLHThM2=Table[Normalize[(LHThQ2M2[[i]]-LHThQ4M2[[i]])⊗(LHThQ3M2[[i]]-
LHThQ4M2[[i]])],{i,1,Length[PositionsRL2]}];
YLHThM2=Table[-Normalize[XLHThM2[[i]]⊗ZLHThM2[[i]]],{i,1,Length[PositionsRL2]}];
OLHThM2=Table[Orthogonalize[{XLHThM2[[i]],YLHThM2[[i]],ZLHThM2[[i]]}],{i,1,Length[PositionsRL2]}];
DLHThM2=Table[Append[Transpose[{OLHThM2[[i,1]],OLHThM2[[i,2]],OLHThM2[[i,3]],LHThQ1M2[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL2]}];

dqELHThM2=Table[HM2dq[DLHThM2[[i]]],{i,1,Length[PositionsRL2]}];
plotLHThM2=generatePlot[dqELHThM2,Blue,3];
Show[plotLHInM2,plotLHThM2]
```

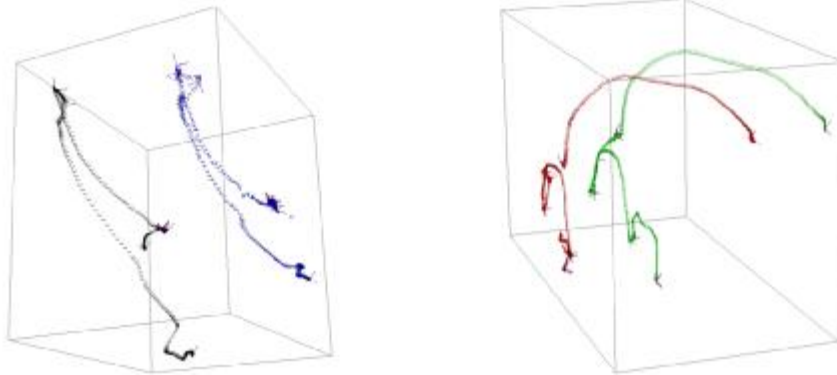
To choose

```
TL=Length[DRHInM2];
M2ch1RH={5,60,90,200,250,350,445,470,545,565,780};
M2ch1LH={5,50,200,250,350,445,470,545,550,700,820};
SL=1;EL=TL;
dqERHInM2c=Table[HM2dq[DRHInM2[[i]]],{i,SL,EL}];
dqERHThM2c=Table[HM2dq[DRHThM2[[i]]],{i,SL,EL}];
plotRHInM2c=generatePlot[dqERHInM2c,Green,3];
plotRHThM2c=generatePlot[dqERHThM2c,Red,3];
Show[plotRHInM2c,plotRHThM2c];
dqELHInM2c=Table[HM2dq[DLHInM2[[i]]],{i,SL,EL}];
dqELHThM2c=Table[HM2dq[DLHThM2[[i]]],{i,SL,EL}];
```

```

plotLHInM2c=generatePlot[dqELHInM2c,Black,3];
plotLHThM2c=generatePlot[dqELHThM2c,Blue,3];
Show[plotLHInM2c,plotLHThM2c];
dqERHInM2c=Table[HM2dq[DRHInM2[[M2ch1RH[[i]]]]],{i,1,Length[M2ch1RH]}];
dqERHThM2c=Table[HM2dq[DRHThM2[[M2ch1RH[[i]]]]],{i,SL,Length[M2ch1RH]}];
plotRHInM2cS=generatePlot[dqERHInM2c,Magenta,10];
plotRHThM2cS=generatePlot[dqERHThM2c,Magenta,10];
Show[plotRHInM2cS,plotRHThM2cS,plotRHInM2c,plotRHThM2c]
dqELHInM2c=Table[HM2dq[DLHInM2[[M2ch1RH[[i]]]]],{i,SL,Length[M2ch1RH]}];
dqELHThM2c=Table[HM2dq[DLHThM2[[M2ch1RH[[i]]]]],{i,SL,Length[M2ch1RH]}];
plotLHInM2cS=generatePlot[dqELHInM2c,Magenta,10];
plotLHThM2cS=generatePlot[dqELHThM2c,Magenta,10];
Show[plotLHInM2cS,plotLHThM2cS,plotLHInM2c,plotLHThM2c]

```



M3 Right Hand-M3

```

RHThQ1M3=Table[ImCord2[[PositionsRL3[[i]],{3,4,5}]],{i,1,Length[PositionsRL3]}];
RHThQ2M3=Table[ImCord2[[PositionsRL3[[i]],{6,7,8}]],{i,1,Length[PositionsRL3]}];
RHThQ3M3=Table[ImCord2[[PositionsRL3[[i]],{9,10,11}]],{i,1,Length[PositionsRL3]}];
];
RHThQ4M3=Table[ImCord2[[PositionsRL3[[i]],{12,13,14}]],{i,1,Length[PositionsRL3]}];
];

RHInQ1M3=Table[ImCord2[[PositionsRL3[[i]],{15,16,17}]],{i,1,Length[PositionsRL3]}];
];
RHInQ2M3=Table[ImCord2[[PositionsRL3[[i]],{18,19,20}]],{i,1,Length[PositionsRL3]}];
];
RHInQ3M3=Table[ImCord2[[PositionsRL3[[i]],{21,22,23}]],{i,1,Length[PositionsRL3]}];
];
RHInQ4M3=Table[ImCord2[[PositionsRL3[[i]],{24,25,26}]],{i,1,Length[PositionsRL3]}];
];
RHInQ5M3=Table[ImCord2[[PositionsRL3[[i]],{27,28,29}]],{i,1,Length[PositionsRL3]}];
];
XRHInM3=Table[Normalize[RHInQ3M3[[i]]-
RHInQ4M3[[i]]],{i,1,Length[PositionsRL3]}];
ZRHInM3=Table[Normalize[(RHInQ3M3[[i]]-RHInQ5M3[[i]])/(RHInQ4M3[[i]]-
RHInQ5M3[[i]])],{i,1,Length[PositionsRL3]}];
YRHInM3=Table[-Normalize[XRHInM3[[i]]/(ZRHInM3[[i]]),{i,1,Length[PositionsRL3]}];
ORHInM3=Table[Orthogonalize[{XRHInM3[[i]],YRHInM3[[i]],ZRHInM3[[i]]}],{i,1,Length[PositionsRL3]}];
DRHInM3=Table[Append[Transpose[{ORHInM3[[i,1]],ORHInM3[[i,2]],ORHInM3[[i,3]],RHInQ1M3[[i]]},{0,0,0,1}],{i,1,Length[PositionsRL3]}];
dqERHInM3=Table[HM2dq[DRHInM3[[i]]],{i,1,Length[PositionsRL3]}];
plotRHInM3=generatePlot[dqERHInM3,Black,3];
XRHThM3=Table[Normalize[RHThQ2M3[[i]]-
RHThQ3M3[[i]]],{i,1,Length[PositionsRL3]}];
ZRHThM3=Table[Normalize[(RHThQ2M3[[i]]-RHThQ4M3[[i]])/(RHThQ3M3[[i]]-
RHThQ4M3[[i]])],{i,1,Length[PositionsRL3]}];

```

```

YRHTM3=Table[-Normalize[XRHTM3[[i]]⊗ZRHTM3[[i]]],{i,1,Length[PositionsRL3]}};
ORHTM3=Table[Orthogonalize[{XRHTM3[[i]],YRHTM3[[i]],ZRHTM3[[i]]}],{i,1,Length[PositionsRL3]}};
DRHTM3=Table[Append[Transpose[{ORHTM3[[i,1]],ORHTM3[[i,2]],ORHTM3[[i,3]],RHTM3[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL3]}};

dqERHTM3=Table[HM2dq[DRHTM3[[i]]],{i,1,Length[PositionsRL3]}};
plotRHThM3=generatePlot[dqERHTM3,Blue,3];
Show[plotRHInM3,plotRHThM3]

```

Left Hand-M3

```

LHTHq1M3=Table[ImCord2[PositionsRL3[[i]],{30,31,32}],{i,1,Length[PositionsRL3]}};
LHTHq2M3=Table[ImCord2[PositionsRL3[[i]],{33,34,35}],{i,1,Length[PositionsRL3]}};
LHTHq3M3=Table[ImCord2[PositionsRL3[[i]],{36,37,38}],{i,1,Length[PositionsRL3]}};
LHTHq4M3=Table[ImCord2[PositionsRL3[[i]],{39,40,41}],{i,1,Length[PositionsRL3]}};

LHInq1M3=Table[ImCord2[PositionsRL3[[i]],{42,43,44}],{i,1,Length[PositionsRL3]}};
LHInq2M3=Table[ImCord2[PositionsRL3[[i]],{45,46,47}],{i,1,Length[PositionsRL3]}};
LHInq3M3=Table[ImCord2[PositionsRL3[[i]],{48,49,50}],{i,1,Length[PositionsRL3]}};
LHInq4M3=Table[ImCord2[PositionsRL3[[i]],{51,52,53}],{i,1,Length[PositionsRL3]}};
LHInq5M3=Table[ImCord2[PositionsRL3[[i]],{54,55,56}],{i,1,Length[PositionsRL3]}};

XLHInM3=Table[Normalize[LHInq3M3[[i]]-LHInq4M3[[i]]],{i,1,Length[PositionsRL3]}};
ZLHInM3=Table[Normalize[(LHInq3M3[[i]]-LHInq5M3[[i]])⊗(LHInq4M3[[i]]-LHInq5M3[[i]])],{i,1,Length[PositionsRL3]}};
YLHInM3=Table[-Normalize[XLHInM3[[i]]⊗ZLHInM3[[i]]],{i,1,Length[PositionsRL3]}};
OLHInM3=Table[Orthogonalize[{XLHInM3[[i]],YLHInM3[[i]],ZLHInM3[[i]]}],{i,1,Length[PositionsRL3]}};
DLHInM3=Table[Append[Transpose[{OLHInM3[[i,1]],OLHInM3[[i,2]],OLHInM3[[i,3]],LHInq1M3[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL3]}};
dqELHInM3=Table[HM2dq[DLHInM3[[i]]],{i,1,Length[PositionsRL3]}};
plotLHInM3=generatePlot[dqELHInM3,Black,3];
XLHTM3=Table[Normalize[LHTHq2M3[[i]]-LHTHq3M3[[i]]],{i,1,Length[PositionsRL3]}};
ZLHTM3=Table[Normalize[(LHTHq2M3[[i]]-LHTHq4M3[[i]])⊗(LHTHq3M3[[i]]-LHTHq4M3[[i]])],{i,1,Length[PositionsRL3]}};
YLHTM3=Table[-Normalize[XLHTM3[[i]]⊗ZLHTM3[[i]]],{i,1,Length[PositionsRL3]}};
OLHTM3=Table[Orthogonalize[{XLHTM3[[i]],YLHTM3[[i]],ZLHTM3[[i]]}],{i,1,Length[PositionsRL3]}};
DLHTM3=Table[Append[Transpose[{OLHTM3[[i,1]],OLHTM3[[i,2]],OLHTM3[[i,3]],LHTHq1M3[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL3]}};

dqELHTM3=Table[HM2dq[DLHTM3[[i]]],{i,1,Length[PositionsRL3]}};
plotLHTM3=generatePlot[dqELHTM3,Blue,3];
Show[plotLHInM3,plotLHTM3]

ZLHInM3=Table[XLHInM3[[i]]⊗YLHInM3[[i]],{i,1,Length[XLHInM3]}};
OLHInM3=Table[Orthogonalize[{XLHInM3[[i]],YLHInM3[[i]],ZLHInM3[[i]]}],{i,1,Length[XLHInM3]}};
DLHInM3=Table[Append[Transpose[{OLHInM3[[i,1]],OLHInM3[[i,2]],OLHInM3[[i,3]],LHInq2M3[[i]]}],{0,0,0,1}],{i,1,Length[XLHInM3]}};

```

```

dqELHInM3=Table[HM2dq[DLHInM3[[i]]],{i,1,Length[DLHInM3]}];
plotLHInM3=generatePlot[dqELHInM3,Black,3];
XLHThM3=Table[(LHThQ1M3[[i]]-LHInQ2M3[[i]])/Sqrt[(LHThQ1M3[[i]]-
LHInQ2M3[[i]])*(LHThQ1M3[[i]]-LHInQ2M3[[i]])],{i,1,Length[LHThQ1M3]}];
YLHThM3=Table[(LHThQ1M3[[i]]-LHThQ2M3[[i]])/Sqrt[(LHThQ1M3[[i]]-
LHThQ2M3[[i]])*(LHThQ1M3[[i]]-LHThQ2M3[[i]])],{i,1,Length[LHThQ1M3]}];
ZLHThM3=Table[XLHThM3[[i]]*YLHThM3[[i]],{i,1,Length[XLHThM3]}];
OLHThM3=Table[Orthogonalize[{XLHThM3[[i]],YLHThM3[[i]],ZLHThM3[[i]]}],{i,1,Length[XLHThM3]}];
DLHThM3=Table[Append[Transpose[{OLHThM3[[i,1]],OLHThM3[[i,2]],OLHThM3[[i,3]],LHThQ2M3[[i]]}],{0,0,0,1}],{i,1,Length[XLHThM3]}];

dqELHThM3=Table[HM2dq[DLHThM3[[i]]],{i,1,Length[DLHThM3]}];
plotLHThM3=generatePlot[dqELHThM3,Blue,3];
Show[plotLHInM3,plotLHThM3]

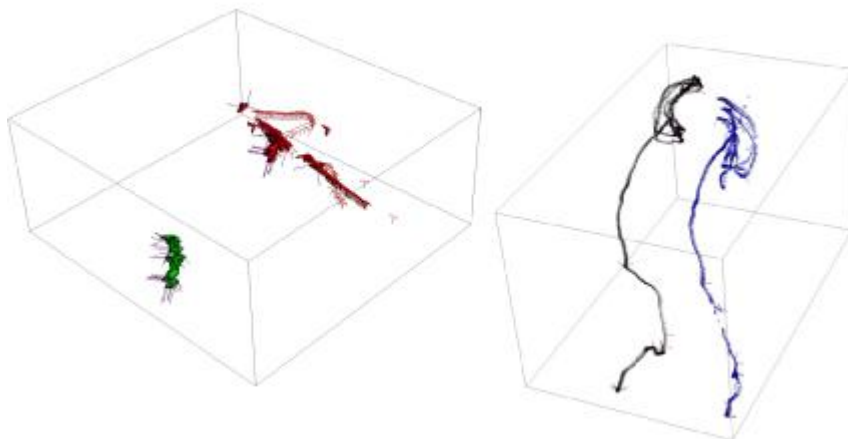
```

To choose

```

TL=Length[DRHInM3];
M3ch1RH={5,80,200,250,260,330,550,600,650,710,840,890,920,950,1250};
M3ch1LH=M3ch1RH;
SL=1;EL=TL;
dqERHInM3c=Table[HM2dq[DRHInM3[[i]]],{i,SL,EL}];
dqERHThM3c=Table[HM2dq[DRHThM3[[i]]],{i,SL,EL}];
plotRHInM3c=generatePlot[dqERHInM3c,Green,3];
plotRHThM3c=generatePlot[dqERHThM3c,Red,3];
Show[plotRHInM3c,plotRHThM3c];
dqELHInM3c=Table[HM2dq[DLHInM3[[i]]],{i,SL,EL}];
dqELHThM3c=Table[HM2dq[DLHThM3[[i]]],{i,SL,EL}];
plotLHInM3c=generatePlot[dqELHInM3c,Black,3];
plotLHThM3c=generatePlot[dqELHThM3c,Blue,3];
Show[plotLHInM3c,plotLHThM3c];
dqERHInM3cS=Table[HM2dq[DRHInM3[[M3ch1RH[[i]]]]],{i,1,Length[M3ch1RH]}];
dqERHThM3cS=Table[HM2dq[DRHThM3[[M3ch1RH[[i]]]]],{i,1,Length[M3ch1RH]}];
plotRHInM3cS=generatePlot[dqERHInM3cS,Magenta,10];
plotRHThM3cS=generatePlot[dqERHThM3cS,Magenta,10];
Show[plotRHInM3cS,plotRHThM3cS,plotRHInM3c,plotRHThM3c];
dqELHInM3cS=Table[HM2dq[DLHInM3[[M3ch1RH[[i]]]]],{i,1,Length[M3ch1RH]}];
dqELHThM3cS=Table[HM2dq[DLHThM3[[M3ch1RH[[i]]]]],{i,1,Length[M3ch1RH]}];
plotLHInM3cS=generatePlot[dqELHInM3cS,Magenta,10];
plotLHThM3cS=generatePlot[dqELHThM3cS,Magenta,10];
Show[plotLHInM3cS,plotLHThM3cS,plotLHInM3c,plotLHThM3c]

```



M4

Right Hand-M4

```
RHThQ1M4=Table[ImCord2[[PositionsRL4[[i]],{3,4,5}]],{i,1,Length[PositionsRL4]};
RHThQ2M4=Table[ImCord2[[PositionsRL4[[i]],{6,7,8}]],{i,1,Length[PositionsRL4]};
RHThQ3M4=Table[ImCord2[[PositionsRL4[[i]],{9,10,11}]],{i,1,Length[PositionsRL4]};
];
RHThQ4M4=Table[ImCord2[[PositionsRL4[[i]],{12,13,14}]],{i,1,Length[PositionsRL4]};
];

RHInQ1M4=Table[ImCord2[[PositionsRL4[[i]],{15,16,17}]],{i,1,Length[PositionsRL4]};
];
RHInQ2M4=Table[ImCord2[[PositionsRL4[[i]],{18,19,20}]],{i,1,Length[PositionsRL4]};
];
RHInQ3M4=Table[ImCord2[[PositionsRL4[[i]],{21,22,23}]],{i,1,Length[PositionsRL4]};
];
RHInQ4M4=Table[ImCord2[[PositionsRL4[[i]],{24,25,26}]],{i,1,Length[PositionsRL4]};
];
RHInQ5M4=Table[ImCord2[[PositionsRL4[[i]],{27,28,29}]],{i,1,Length[PositionsRL4]};
];
XRHInM4=Table[Normalize[RHInQ3M4[[i]]-
RHInQ4M4[[i]]],{i,1,Length[PositionsRL4]};
ZRHInM4=Table[Normalize[(RHInQ3M4[[i]]-RHInQ5M4[[i]])⊗(RHInQ4M4[[i]]-
RHInQ5M4[[i]])],{i,1,Length[PositionsRL4]};
YRHInM4=Table[-Normalize[XRHInM4[[i]]⊗ZRHInM4[[i]]],{i,1,Length[PositionsRL4]};
ORHInM4=Table[Orthogonalize[{XRHInM4[[i]],YRHInM4[[i]],ZRHInM4[[i]]}],{i,1,Length[PositionsRL4]};
DRHInM4=Table[Append[Transpose[{ORHInM4[[i,1]],ORHInM4[[i,2]],ORHInM4[[i,3]],RHInQ1M4[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL4]};
dqERHInM4=Table[HM2dq[DRHInM4[[i]]],{i,1,Length[PositionsRL4]};
plotRHInM4=generatePlot[dqERHInM4,Black,3];
XRHThM4=Table[Normalize[RHThQ2M4[[i]]-
RHThQ3M4[[i]]],{i,1,Length[PositionsRL4]};
ZRHThM4=Table[Normalize[(RHThQ2M4[[i]]-RHThQ4M4[[i]])⊗(RHThQ3M4[[i]]-
RHThQ4M4[[i]])],{i,1,Length[PositionsRL4]};
YRHThM4=Table[-Normalize[XRHThM4[[i]]⊗ZRHThM4[[i]]],{i,1,Length[PositionsRL4]};
ORHThM4=Table[Orthogonalize[{XRHThM4[[i]],YRHThM4[[i]],ZRHThM4[[i]]}],{i,1,Length[PositionsRL4]};
DRHThM4=Table[Append[Transpose[{ORHThM4[[i,1]],ORHThM4[[i,2]],ORHThM4[[i,3]],RHThQ1M4[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL4]};

dqERHThM4=Table[HM2dq[DRHThM4[[i]]],{i,1,Length[PositionsRL4]};
plotRHThM4=generatePlot[dqERHThM4,Blue,3];
Show[plotRHInM4,plotRHThM4]
```

Left Hand-M4

```
LHThQ1M4=Table[ImCord2[[PositionsRL4[[i]],{30,31,32}]],{i,1,Length[PositionsRL4]};
];
LHThQ2M4=Table[ImCord2[[PositionsRL4[[i]],{33,34,35}]],{i,1,Length[PositionsRL4]};
];
LHThQ3M4=Table[ImCord2[[PositionsRL4[[i]],{36,37,38}]],{i,1,Length[PositionsRL4]};
];
LHThQ4M4=Table[ImCord2[[PositionsRL4[[i]],{39,40,41}]],{i,1,Length[PositionsRL4]};
];

LHInQ1M4=Table[ImCord2[[PositionsRL4[[i]],{42,43,44}]],{i,1,Length[PositionsRL4]};
];
LHInQ2M4=Table[ImCord2[[PositionsRL4[[i]],{45,46,47}]],{i,1,Length[PositionsRL4]};
];
LHInQ3M4=Table[ImCord2[[PositionsRL4[[i]],{48,49,50}]],{i,1,Length[PositionsRL4]};
];
```

```

LHInQ4M4=Table[ImCord2[PositionsRL4[[i]],{51,52,53}]],{i,1,Length[PositionsRL4]
}];
LHInQ5M4=Table[ImCord2[PositionsRL4[[i]],{54,55,56}]],{i,1,Length[PositionsRL4]
}];

XLHInM4=Table[Normalize[LHInQ3M4[[i]]-
LHInQ4M4[[i]]],{i,1,Length[PositionsRL4]}};
ZLHInM4=Table[Normalize[(LHInQ3M4[[i]]-LHInQ5M4[[i]])⊗(LHInQ4M4[[i]]-
LHInQ5M4[[i]])],{i,1,Length[PositionsRL4]}};
YLHInM4=Table[-Normalize[XLHInM4[[i]]⊗ZLHInM4[[i]]],{i,1,Length[PositionsRL4]}};
OLHInM4=Table[Orthogonalize[{XLHInM4[[i]],YLHInM4[[i]],ZLHInM4[[i]]}],{i,1,Length[PositionsRL4]}};
DLHInM4=Table[Append[Transpose[{OLHInM4[[i,1]],OLHInM4[[i,2]],OLHInM4[[i,3]],LHInQ1M4[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL4]}};
dqELHInM4=Table[HM2dq[DLHInM4[[i]]],{i,1,Length[PositionsRL4]}};
plotLHInM4=generatePlot[dqELHInM4,Black,3];
XLHThM4=Table[Normalize[LHThQ2M4[[i]]-
LHThQ3M4[[i]]],{i,1,Length[PositionsRL4]}};
ZLHThM4=Table[Normalize[(LHThQ2M4[[i]]-LHThQ4M4[[i]])⊗(LHThQ3M4[[i]]-
LHThQ4M4[[i]])],{i,1,Length[PositionsRL4]}};
YLHThM4=Table[-Normalize[XLHThM4[[i]]⊗ZLHThM4[[i]]],{i,1,Length[PositionsRL4]}};
OLHThM4=Table[Orthogonalize[{XLHThM4[[i]],YLHThM4[[i]],ZLHThM4[[i]]}],{i,1,Length[PositionsRL4]}};
DLHThM4=Table[Append[Transpose[{OLHThM4[[i,1]],OLHThM4[[i,2]],OLHThM4[[i,3]],LHThQ1M4[[i]]}],{0,0,0,1}],{i,1,Length[PositionsRL4]}};

dqELHThM4=Table[HM2dq[DLHThM4[[i]]],{i,1,Length[PositionsRL4]}};
plotLHThM4=generatePlot[dqELHThM4,Blue,3];
Show[plotLHInM4,plotLHThM4]

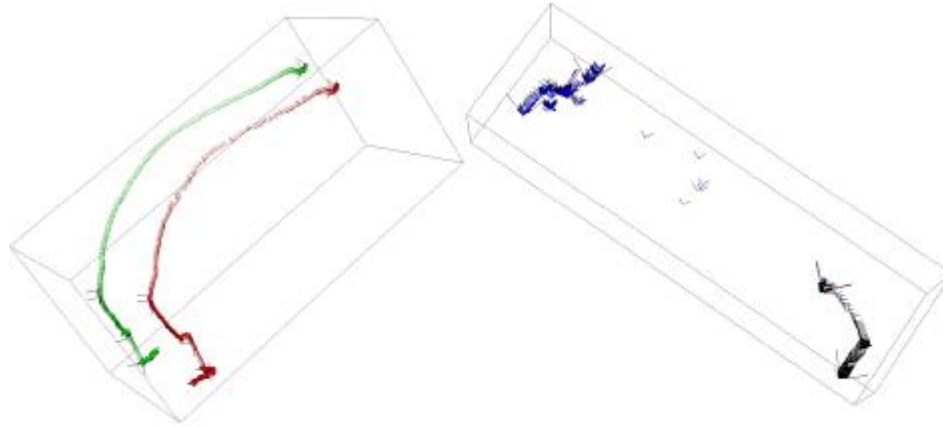
```

To choose

```

TL=Length[DRHInM4]
766
M4ch1RH={5,210,370,380,540,600};
M4ch1LH=M4ch1RH;
SL=1;EL=TL;
dqERHInM4c=Table[HM2dq[DRHInM4[[i]]],{i,SL,EL}];
dqERHThM4c=Table[HM2dq[DRHThM4[[i]]],{i,SL,EL}];
plotRHInM4c=generatePlot[dqERHInM4c,Green,3];
plotRHThM4c=generatePlot[dqERHThM4c,Red,3];
Show[plotRHInM4c,plotRHThM4c];
dqELHInM4c=Table[HM2dq[DLHInM4[[i]]],{i,SL,EL}];
dqELHThM4c=Table[HM2dq[DLHThM4[[i]]],{i,SL,EL}];
plotLHInM4c=generatePlot[dqELHInM4c,Black,3];
plotLHThM4c=generatePlot[dqELHThM4c,Blue,3];
Show[plotLHInM4c,plotLHThM4c];
dqERHInM4cS=Table[HM2dq[DRHInM4[[M4ch1RH[[i]]]]],{i,1,Length[M4ch1RH]}};
dqERHThM4cS=Table[HM2dq[DRHThM4[[M4ch1RH[[i]]]]],{i,SL,Length[M4ch1RH]}};
plotRHInM4cS=generatePlot[dqERHInM4cS,Magenta,10];
plotRHThM4cS=generatePlot[dqERHThM4cS,Magenta,10];
Show[plotRHInM4cS,plotRHThM4cS,plotRHInM4c,plotRHThM4c];
dqELHInM4cS=Table[HM2dq[DLHInM4[[M4ch1RH[[i]]]]],{i,SL,Length[M4ch1RH]}};
dqELHThM4cS=Table[HM2dq[DLHThM4[[M4ch1RH[[i]]]]],{i,SL,Length[M4ch1RH]}};
plotLHInM4cS=generatePlot[dqELHInM4cS,Magenta,10];
plotLHThM4cS=generatePlot[dqELHThM4cS,Magenta,10];
Show[plotLHInM4cS,plotLHThM4cS,plotLHInM4c,plotLHThM4c]

```



Data Sorting

```
SeleM1RHIn1=dqERHInM1[[M1ch1RH]];
SeleM1RHTh1=dqERHThM1[[M1ch1RH]];
SeleM1LHIn1=dqELHInM1[[M1ch1LH]];
SeleM1LHTh1=dqELHThM1[[M1ch1LH]];
SeleM2RHIn1=dqERHInM2[[M2ch1RH]];
SeleM2RHTh1=dqERHThM2[[M2ch1RH]];
SeleM2LHIn1=dqELHInM2[[M2ch1LH]];
SeleM2LHTh1=dqELHThM2[[M2ch1LH]];
SeleM3RHIn1=dqERHInM3[[M3ch1RH]];
SeleM3RHTh1=dqERHThM3[[M3ch1RH]];
SeleM3LHIn1=dqELHInM3[[M3ch1LH]];
SeleM3LHTh1=dqELHThM3[[M3ch1LH]];
SeleM4RHIn1=dqERHInM4[[M4ch1RH]];
SeleM4RHTh1=dqERHThM4[[M4ch1RH]];
SeleM4LHIn1=dqELHInM4[[M4ch1LH]];
SeleM4LHTh1=dqELHThM4[[M4ch1LH]];
```

lets name the RHIn fin1, RHTh fin2, LHIn fin3 and LHTh fin4. here the fin1 means finger 1 fin11 means first level positions chosen for finger 1

fin1

```
Fin1L1=Flatten[{SeleM1RHIn1,SeleM2RHIn1,SeleM3RHIn1,SeleM4RHIn1},1];
```

fin2

```
Fin2L1=Flatten[{SeleM1RHTh1,SeleM2RHTh1,SeleM3RHTh1,SeleM4RHTh1},1];
```

fin3

```
Fin3L1=Flatten[{SeleM1LHIn1,SeleM2LHIn1,SeleM3LHIn1,SeleM4LHIn1},1]
```

fin4

```
Fin4L1=Flatten[{SeleM1LHTh1,SeleM2LHTh1,SeleM3LHTh1,SeleM4LHTh1},1];
```

Forming the relative displacements

```
Fin1L1=Flatten[{SeleM1RHIn1,SeleM2RHIn1,SeleM3RHIn1,SeleM4RHIn1},1];
Fin2L1=Flatten[{SeleM1RHTh1,SeleM2RHTh1,SeleM3RHTh1,SeleM4RHTh1},1];
Fin3L1=Flatten[{SeleM1LHIn1,SeleM2LHIn1,SeleM3LHIn1,SeleM4LHIn1},1];
Fin4L1=Flatten[{SeleM1LHTh1,SeleM2LHTh1,SeleM3LHTh1,SeleM4LHTh1},1];
Fin1L2=Fin1L1;
Fin4L2=Fin4L1;
Fin1L3=Fin1L1;
Db1b2=Table[quatmult[quatconj[Fin1L1[[i]]],Fin2L1[[i]]],{i,1,Length[Fin2L1]};
Db3b4=Table[quatmult[quatconj[Fin3L1[[i]]],Fin4L1[[i]]],{i,1,Length[Fin4L1]};
RelativaMotion12=Table[quatmult[Db1b2[[i]],quatconj[Db1b2[[1]]]],{i,2,Length[Db1b2]};
RelativaMotion34=Table[quatmult[Db3b4[[i]],quatconj[Db3b4[[1]]]],{i,2,Length[Db3b4]};
```

Solving Level 1- Gripper Level

First

Calculations

Here S0 is S11, S1 is S7, S2 is S8

```
t0Solv={t01,t02,t03,t04,t05,t06,t07,t08,t09,t010,t011,t012,t013,t014,t015,t016,t017,t018,t019,t020,t021,t022,t023,t024,t025,t026,t027,t028,t029,t030,t031,t032,t033,t034,t035,t036,t037,t038,t039};t1Solv={t11,t12,t13,t14,t15,t16,t17,t18,t19,t110,t111,t112,t113,t114,t115,t116,t117,t118,t119,t120,t121,t122,t123,t124,t125,t126,t127,t128,t129,t130,t131,t132,t133,t134,t135,t136,t137,t138,t139};t2Solv={t21,t22,t23,t24,t25,t26,t27,t28,t29,t210,t211,t212,t213,t214,t215,t216,t217,t218,t219,t220,t221,t222,t223,t224,t225,t226,t227,t228,t229,t230,t231,t232,t233,t234,t235,t236,t237,t238,t239};
tSolve=Flatten[{t0Solv,t1Solv,t2Solv}];
SSolv={{S0x,S0y,S0z},{Sm0x,Sm0y,Sm0z}},{S1x,S1y,S1z},{Smlx,Smly,Sm1z}},{S2x,S2y,S2z},{Sm2x,Sm2y,Sm2z}};
S0dqSolv=Table[makequat[SSolv[[1]],t0Solv[[i]],0},{i,1,39}];
S1dqSolv=Table[makequat[SSolv[[2]],t1Solv[[i]],0},{i,1,39}];
S2dqSolv=Table[makequat[SSolv[[3]],t2Solv[[i]],0},{i,1,39}];
FK=Table[quatmult[quatmult[quatconj[S0dqSolv[[i]]],quatconj[S1dqSolv[[i]]]],S2dqSolv[[i]]},{i,1,39}];
var=Flatten[{t0Solv,t1Solv,t2Solv,SSolv}];
eq1=Flatten[Table[(FK[[i]]-RelativaMotion12[[i]]),{i,1,39}]];

eq2={SSolv[[1,1]].SSolv[[1,2]],SSolv[[2,1]].SSolv[[2,2]],SSolv[[3,1]].SSolv[[3,2]]};
eq3=10*{SSolv[[1,1]].SSolv[[1,1]]-1,SSolv[[2,1]].SSolv[[2,1]]-1,SSolv[[3,1]].SSolv[[3,1]]-1};
eq=Flatten[{eq1,eq2,eq3}];
Fun=Table[F[k],{k,1,Length[eq]}];
Length[eq];
Length[var];
LB=Flatten[Append[Table[-6.28,{i,1,Length[tSolve]}],Table[{-1,-1,-1,-1000,-1000,-1000},{i,1,Length[SSolv]}]]];
UB=Flatten[Append[Table[6.28,{i,1,Length[tSolve]}],Table[{1,1,1,1000,1000,1000},{i,1,Length[SSolv]}]]];
EqAdres= FileNameJoin[{"Equation.m"}];
Equation=OpenWrite[EqAdres,PageWidth->300];
WriteString[Equation,"function F=Equation(x) \n"]
WriteString[Equation,"t01=x(1),t02=x(2),t03=x(3),t04=x(4),t05=x(5),t06=x(6),t07=x(7),t08=x(8),t09=x(9),t010=x(10),t011=x(11),t012=x(12),t013=x(13),t014=x(14),t015=x(15),t016=x(16),t017=x(17),t018=x(18),t019=x(19),t020=x(20),t021=x(21),t022=x(22),t023=x(23),t024=x(24),t025=x(25),t026=x(26),t027=x(27),t028=x(28),t029=x(29),t030=x(30),t031=x(31),t032=x(32),t033=x(33),t034=x(34),t035=x(35),t036=x(36),t037=x(37),t038=x(38),t039=x(39),t11=x(40),t12=x(41),t13=x(42),t14=x(43),t15=x(44),t16=x(45),t17=x(46),t18=x(47),t19=x(48),t110=x(49),t111=x(50),t112=x(51),t113=x(52),t114=x(53),t115=x(54),t116=x(55),t117=x(56),t118=x(57),t119=x(58),t120=x(59),t121=x(60),t122=x(61),t123=x(62),t124=x(63),t125=x(64),t126=x(65),t127=x(66),t128=x(67),t129=x(68),t130=x(69),t131=x(70),t132=x(71),t133=x(72),t134=x(73),t135=x(74),t136=x(75),t137=x(76),t138=x(77),t139=x(78),t21=x(79),t22=x(80),t23=x(81),t24=x(82),t25=x(83),t26=x(84),t27=x(85),t28=x(86),t29=x(87),t210=x(88),t211=x(89),t212=x(90),t213=x(91),t214=x(92),t215=x(93),t216=x(94),t217=x(95),t218=x(96),t219=x(97),t220=x(98),t221=x(99),t222=x(100),t223=x(101),t224=x(102),t225=x(103),t226=x(104),t227=x(105),t228=x(106),t229=x(107),t230=x(108),t231=x(109),t232=x(110),t233=x(111),t234=x(112),t235=x(113),t236=x(114),t237=x(115),t238=x(116),t239=x(117),S0x=x(118),S0y=x(119),S0z=x(120),Sm0x=x(121),Sm0y=x(122),Sm0z=x(123),S1x=x(124),S1y=x(125),S1z=x(126),Smlx=x(127),Smly=x(128),Sm1z=x(129),S2x=x(130),S2y=x(131),S2z=x(132),Sm2x=x(133),Sm2y=x(134),Sm2z=x(135);\n\n"];
Table[WriteMatlab[eq[[i]],Equation, Fun[[i]]],{i,1,Length[eq]}];
Close[Equation];
xinitial=Table[2*Random[]-1,{i,1,Length[var]}];
(*xinitial=xc*)
AddressRun= FileNameJoin[{"run.m"}];
```



```

run=OpenWrite[AddressRun,PageWidth->300];
OutputStream[AddressRun,125];
WriteString[run,"options=optioptions('lsqnonlin','Display', 'iter',
'MaxIter',5000,'TolFun',1.0000e-12,
'MaxFunEvals',150000,'PlotFcns',{@optimplotfirstorderopt})\n\n"];
WriteMatlab[xinitial,run, x0];
WriteMatlab[LB,run, lb];
WriteMatlab[UB,run, ub];
WriteString[run," x=lsqnonlin(@Equation,x0, lb, ub,options);\n
save('dataxb','x') \n"]
Close[run];

```

results

```

x=Flatten[Import["dataxb.mat"]];

```

```

t01out=x[[1]];t02out=x[[2]];t03out=x[[3]];t04out=x[[4]];t05out=x[[5]];t06out=x[[
6]];t07out=x[[7]];t08out=x[[8]];t09out=x[[9]];t010out=x[[10]];t011out=x[[11]];t0
12out=x[[12]];t013out=x[[13]];t014out=x[[14]];t015out=x[[15]];t016out=x[[16]];t0
17out=x[[17]];t018out=x[[18]];t019out=x[[19]];t020out=x[[20]];t021out=x[[21]];t0
22out=x[[22]];t023out=x[[23]];t024out=x[[24]];t025out=x[[25]];t026out=x[[26]];t0
27out=x[[27]];t028out=x[[28]];t029out=x[[29]];t030out=x[[30]];t031out=x[[31]];t0
32out=x[[32]];t033out=x[[33]];t034out=x[[34]];t035out=x[[35]];t036out=x[[36]];t0
37out=x[[37]];t038out=x[[38]];t039out=x[[39]];t11out=x[[40]];t12out=x[[41]];t13o
ut=x[[42]];t14out=x[[43]];t15out=x[[44]];t16out=x[[45]];t17out=x[[46]];t18out=x[
[47]];t19out=x[[48]];t110out=x[[49]];t111out=x[[50]];t112out=x[[51]];t113out=x[[
52]];t114out=x[[53]];t115out=x[[54]];t116out=x[[55]];t117out=x[[56]];t118out=x[[
57]];t119out=x[[58]];t120out=x[[59]];t121out=x[[60]];t122out=x[[61]];t123out=x[[
62]];t124out=x[[63]];t125out=x[[64]];t126out=x[[65]];t127out=x[[66]];t128out=x[[
67]];t129out=x[[68]];t130out=x[[69]];t131out=x[[70]];t132out=x[[71]];t133out=x[[
72]];t134out=x[[73]];t135out=x[[74]];t136out=x[[75]];t137out=x[[76]];t138out=x[[
77]];t139out=x[[78]];t21out=x[[79]];t22out=x[[80]];t23out=x[[81]];t24out=x[[82]]
;t25out=x[[83]];t26out=x[[84]];t27out=x[[85]];t28out=x[[86]];t29out=x[[87]];t210
out=x[[88]];t211out=x[[89]];t212out=x[[90]];t213out=x[[91]];t214out=x[[92]];t215
out=x[[93]];t216out=x[[94]];t217out=x[[95]];t218out=x[[96]];t219out=x[[97]];t220
out=x[[98]];t221out=x[[99]];t222out=x[[100]];t223out=x[[101]];t224out=x[[102]];t
225out=x[[103]];t226out=x[[104]];t227out=x[[105]];t228out=x[[106]];t229out=x[[10
7]];t230out=x[[108]];t231out=x[[109]];t232out=x[[110]];t233out=x[[111]];t234out=
x[[112]];t235out=x[[113]];t236out=x[[114]];t237out=x[[115]];t238out=x[[116]];t23
9out=x[[117]];S0xout=x[[118]];S0yout=x[[119]];S0zout=x[[120]];Sm0xout=x[[121]];S
m0yout=x[[122]];Sm0zout=x[[123]];S1xout=x[[124]];S1yout=x[[125]];S1zout=x[[126]]
;Smlxout=x[[127]];Smlyout=x[[128]];Smlzout=x[[129]];S2xout=x[[130]];S2yout=x[[13
1]];S2zout=x[[132]];Sm2xout=x[[133]];Sm2yout=x[[134]];Sm2zout=x[[135]];
t0Solvout1={t01out, t02out, t03out, t04out, t05out, t06out, t07out, t08out,
t09out, t010out, t011out, t012out, t013out, t014out, t015out, t016out, t017out,
t018out, t019out, t020out, t021out, t022out, t023out, t024out, t025out, t026out,
t027out, t028out, t029out, t030out, t031out, t032out, t033out, t034out, t035out,
t036out, t037out, t038out, t039out};t1Solvout1={t11out, t12out, t13out, t14out,
t15out, t16out, t17out, t18out, t19out, t110out, t111out, t112out, t113out,
t114out, t115out, t116out, t117out, t118out, t119out, t120out, t121out, t122out,
t123out, t124out, t125out, t126out, t127out, t128out, t129out, t130out, t131out,
t132out, t133out, t134out, t135out, t136out, t137out, t138out,
t139out};t2Solvout1={t21out, t22out, t23out, t24out, t25out, t26out, t27out,
t28out, t29out, t210out, t211out, t212out, t213out, t214out, t215out, t216out,
t217out, t218out, t219out, t220out, t221out, t222out, t223out, t224out, t225out,
t226out, t227out, t228out, t229out, t230out, t231out, t232out, t233out, t234out,
t235out, t236out, t237out, t238out, t239out};
tSolveout1={t0Solvout1, t1Solvout1, t2Solvout1};
SSolvout1={{S0xout, S0yout, S0zout},{Sm0xout, Sm0yout, Sm0zout}},{S1xout, S1yout, S1
zout},{Smlxout, Smlyout, Smlzout}},{S2xout, S2yout, S2zout},{Sm2xout, Sm2yout, Sm2zou
t}};

```

Second

Calculations

Here S0 is S12, S1 is S9, S2 is S10

```

t0Solv={t01,t02,t03,t04,t05,t06,t07,t08,t09,t010,t011,t012,t013,t014,t015,t016,t
017,t018,t019,t020,t021,t022,t023,t024,t025,t026,t027,t028,t029,t030,t031,t032,t
033,t034,t035,t036,t037,t038,t039};t1Solv={t11,t12,t13,t14,t15,t16,t17,t18,t19,t
110,t111,t112,t113,t114,t115,t116,t117,t118,t119,t120,t121,t122,t123,t124,t125,t
126,t127,t128,t129,t130,t131,t132,t133,t134,t135,t136,t137,t138,t139};t2Solv={t2
1,t22,t23,t24,t25,t26,t27,t28,t29,t210,t211,t212,t213,t214,t215,t216,t217,t218,t
219,t220,t221,t222,t223,t224,t225,t226,t227,t228,t229,t230,t231,t232,t233,t234,t
235,t236,t237,t238,t239};
tSolve=Flatten[{t0Solv,t1Solv,t2Solv}];
SSolv={{S0x,S0y,S0z},{Sm0x,Sm0y,Sm0z}},{S1x,S1y,S1z},{Smlx,Smly,Sm1z}},{S2x,S
2y,S2z},{Sm2x,Sm2y,Sm2z}}};
S0dqSolv=Table[makequat[SSolv[[1]],t0Solv[[i]],0},{i,1,39}];
S1dqSolv=Table[makequat[SSolv[[2]],t1Solv[[i]],0},{i,1,39}];
S2dqSolv=Table[makequat[SSolv[[3]],t2Solv[[i]],0},{i,1,39}];
FK=Table[quatmult[quatmult[quatconj[S0dqSolv[[i]]],quatconj[S1dqSolv[[i]]]],S2dq
Solv[[i]],{i,1,39}];
var=Flatten[{t0Solv,t1Solv,t2Solv,SSolv}];
eq1=Flatten[Table[(FK[[i]]-RelativaMotion34[[i]]),{i,1,39}]];
eq2={SSolv[[1,1]].SSolv[[1,2]],SSolv[[2,1]].SSolv[[2,2]],SSolv[[3,1]].SSolv[[3,2
]]};
eq3={SSolv[[1,1]].SSolv[[1,1]]-1,SSolv[[2,1]].SSolv[[2,1]]-
1,SSolv[[3,1]].SSolv[[3,1]]-1};
eq=Flatten[{eq1,eq2,eq3}];
Fun=Table[F[k],{k,1,Length[eq]}];
Length[eq];
Length[var];
LB=Flatten[Append[Table[-6.28,{i,1,Length[tSolve]}],Table[{-1,-1,-1,-1000,-
1000,-1000},{i,1,Length[SSolv]}]]];
UB=Flatten[Append[Table[6.28,
{i,1,Length[tSolve]}],Table[{1,1,1,1000,1000,1000},{i,1,Length[SSolv]}]]];
EqAdres2= FileNameJoin[{"Equation2.m"}];
Equation2=OpenWrite[EqAdres2,PageWidth->300];
WriteString[Equation2,"function F=Equation2(x) \n"]
WriteString[Equation2,"t01=x(1),t02=x(2),t03=x(3),t04=x(4),t05=x(5),t06=x(6),t07
=x(7),t08=x(8),t09=x(9),t010=x(10),t011=x(11),t012=x(12),t013=x(13),t014=x(14),t
015=x(15),t016=x(16),t017=x(17),t018=x(18),t019=x(19),t020=x(20),t021=x(21),t022
=x(22),t023=x(23),t024=x(24),t025=x(25),t026=x(26),t027=x(27),t028=x(28),t029=x(
29),t030=x(30),t031=x(31),t032=x(32),t033=x(33),t034=x(34),t035=x(35),t036=x(36)
,t037=x(37),t038=x(38),t039=x(39),t11=x(40),t12=x(41),t13=x(42),t14=x(43),t15=x(
44),t16=x(45),t17=x(46),t18=x(47),t19=x(48),t110=x(49),t111=x(50),t112=x(51),t11
3=x(52),t114=x(53),t115=x(54),t116=x(55),t117=x(56),t118=x(57),t119=x(58),t120=x
(59),t121=x(60),t122=x(61),t123=x(62),t124=x(63),t125=x(64),t126=x(65),t127=x(66)
,t128=x(67),t129=x(68),t130=x(69),t131=x(70),t132=x(71),t133=x(72),t134=x(73),t
135=x(74),t136=x(75),t137=x(76),t138=x(77),t139=x(78),t21=x(79),t22=x(80),t23=x(
81),t24=x(82),t25=x(83),t26=x(84),t27=x(85),t28=x(86),t29=x(87),t210=x(88),t211=
x(89),t212=x(90),t213=x(91),t214=x(92),t215=x(93),t216=x(94),t217=x(95),t218=x(9
6),t219=x(97),t220=x(98),t221=x(99),t222=x(100),t223=x(101),t224=x(102),t225=x(1
03),t226=x(104),t227=x(105),t228=x(106),t229=x(107),t230=x(108),t231=x(109),t232
=x(110),t233=x(111),t234=x(112),t235=x(113),t236=x(114),t237=x(115),t238=x(116),
t239=x(117),S0x=x(118),S0y=x(119),S0z=x(120),Sm0x=x(121),Sm0y=x(122),Sm0z=x(123)
,S1x=x(124),S1y=x(125),S1z=x(126),Smlx=x(127),Smly=x(128),Sm1z=x(129),S2x=x(130)
,S2y=x(131),S2z=x(132),Sm2x=x(133),Sm2y=x(134),Sm2z=x(135);\n\n"];
Table[WriteMatlab[eq[[i]],Equation2, Fun[[i]],{i,1,Length[eq]}];
Close[Equation2];
xinitial=Table[2*Random[]-1,{i,1,Length[var]}];
(*xinitial=xc*)
AddressRun2= FileNameJoin[{"run2.m"}];
run2=OpenWrite[AddressRun2,PageWidth->300];
OutputStream[AddressRun2,125];
WriteString[run2,"options=optimoptions('lsqnonlin','Display','iter',
'MaxIter',10000,'TolFun',1.0000e-12,
'MaxFunEvals',1500000,'PlotFcns',{optimplotfirstorderopt})\n\n"];
WriteMatlab[xinitial,run2, x0];

```

```

WriteMatlab[LB,run2, lb];
WriteMatlab[UB,run2, ub];
WriteString[run2," x=lsqnonlin(@Equation2,x0, lb, ub,options);\n
save('datax2','x') \n"]
Close[run2];

```

results

```

x2=Flatten[Import["datax2.mat"]];

t01out=x2[[1]];t02out=x2[[2]];t03out=x2[[3]];t04out=x2[[4]];t05out=x2[[5]];t06ou
t=x2[[6]];t07out=x2[[7]];t08out=x2[[8]];t09out=x2[[9]];t010out=x2[[10]];t011out=
x2[[11]];t012out=x2[[12]];t013out=x2[[13]];t014out=x2[[14]];t015out=x2[[15]];t01
6out=x2[[16]];t017out=x2[[17]];t018out=x2[[18]];t019out=x2[[19]];t020out=x2[[20]
];t021out=x2[[21]];t022out=x2[[22]];t023out=x2[[23]];t024out=x2[[24]];t025out=x2
[[25]];t026out=x2[[26]];t027out=x2[[27]];t028out=x2[[28]];t029out=x2[[29]];t030o
ut=x2[[30]];t031out=x2[[31]];t032out=x2[[32]];t033out=x2[[33]];t034out=x2[[34]];
t035out=x2[[35]];t036out=x2[[36]];t037out=x2[[37]];t038out=x2[[38]];t039out=x2[[
39]];t11out=x2[[40]];t12out=x2[[41]];t13out=x2[[42]];t14out=x2[[43]];t15out=x2[[
44]];t16out=x2[[45]];t17out=x2[[46]];t18out=x2[[47]];t19out=x2[[48]];t110out=x2[
[49]];t111out=x2[[50]];t112out=x2[[51]];t113out=x2[[52]];t114out=x2[[53]];t115ou
t=x2[[54]];t116out=x2[[55]];t117out=x2[[56]];t118out=x2[[57]];t119out=x2[[58]];t
120out=x2[[59]];t121out=x2[[60]];t122out=x2[[61]];t123out=x2[[62]];t124out=x2[[6
3]];t125out=x2[[64]];t126out=x2[[65]];t127out=x2[[66]];t128out=x2[[67]];t129out=
x2[[68]];t130out=x2[[69]];t131out=x2[[70]];t132out=x2[[71]];t133out=x2[[72]];t13
4out=x2[[73]];t135out=x2[[74]];t136out=x2[[75]];t137out=x2[[76]];t138out=x2[[77]
];t139out=x2[[78]];t21out=x2[[79]];t22out=x2[[80]];t23out=x2[[81]];t24out=x2[[82
]];t25out=x2[[83]];t26out=x2[[84]];t27out=x2[[85]];t28out=x2[[86]];t29out=x2[[87
]];t210out=x2[[88]];t211out=x2[[89]];t212out=x2[[90]];t213out=x2[[91]];t214out=x
2[[92]];t215out=x2[[93]];t216out=x2[[94]];t217out=x2[[95]];t218out=x2[[96]];t219
out=x2[[97]];t220out=x2[[98]];t221out=x2[[99]];t222out=x2[[100]];t223out=x2[[101
]];t224out=x2[[102]];t225out=x2[[103]];t226out=x2[[104]];t227out=x2[[105]];t228o
ut=x2[[106]];t229out=x2[[107]];t230out=x2[[108]];t231out=x2[[109]];t232out=x2[[1
10]];t233out=x2[[111]];t234out=x2[[112]];t235out=x2[[113]];t236out=x2[[114]];t23
7out=x2[[115]];t238out=x2[[116]];t239out=x2[[117]];S0xout=x2[[118]];S0yout=x2[[1
19]];S0zout=x2[[120]];Sm0xout=x2[[121]];Sm0yout=x2[[122]];Sm0zout=x2[[123]];S1xo
ut=x2[[124]];S1yout=x2[[125]];S1zout=x2[[126]];Sm1xout=x2[[127]];Sm1yout=x2[[128
]];Sm1zout=x2[[129]];S2xout=x2[[130]];S2yout=x2[[131]];S2zout=x2[[132]];Sm2xout=
x2[[133]];Sm2yout=x2[[134]];Sm2zout=x2[[135]];

t0Solvout2={t01out, t02out, t03out, t04out, t05out, t06out, t07out, t08out,
t09out, t010out, t011out, t012out, t013out, t014out, t015out, t016out, t017out,
t018out, t019out, t020out, t021out, t022out, t023out, t024out, t025out, t026out,
t027out, t028out, t029out, t030out, t031out, t032out, t033out, t034out, t035out,
t036out, t037out, t038out, t039out};t1Solvout2={t11out, t12out, t13out, t14out,
t15out, t16out, t17out, t18out, t19out, t110out, t111out, t112out, t113out,
t114out, t115out, t116out, t117out, t118out, t119out, t120out, t121out, t122out,
t123out, t124out, t125out, t126out, t127out, t128out, t129out, t130out, t131out,
t132out, t133out, t134out, t135out, t136out, t137out, t138out,
t139out};t2Solvout2={t21out, t22out, t23out, t24out, t25out, t26out, t27out,
t28out, t29out, t210out, t211out, t212out, t213out, t214out, t215out, t216out,
t217out, t218out, t219out, t220out, t221out, t222out, t223out, t224out, t225out,
t226out, t227out, t228out, t229out, t230out, t231out, t232out, t233out, t234out,
t235out, t236out, t237out, t238out, t239out};
tSolveout2={t0Solvout2, t1Solvout2, t2Solvout2};
SSolvout2={{S0xout,S0yout,S0zout},{Sm0xout,Sm0yout,Sm0zout}},{S1xout,S1yout,S1
zout},{Sm1xout,Sm1yout,Sm1zout}},{S2xout,S2yout,S2zout},{Sm2xout,Sm2yout,Sm2zou
t}};

```

Solving Level 2- Secondary Level Calculations

We had SSolv0 is S11, SSolv1 is S7, SSolv2 is S8, also in second gripper SSolv0 is S12, SSolv1 is S9, SSolv2 is S10

Here we have SSolv0 is S5, SSolv1 is S3, SSolv2 is S4 and SSSolv3 is S6

```

S11f=SSolvout1[[1]];
S7f=SSolvout1[[2]];
S8f=SSolvout1[[3]];
S12f=SSolvout2[[1]];
S9f=SSolvout2[[2]];
S10f=SSolvout2[[3]];
t11f=-tSolveout1[[1]];
t7f=-tSolveout1[[2]];
t8f=tSolveout1[[3]];
t12f=-tSolveout2[[1]];
t9f=-tSolveout2[[2]];
t10f=tSolveout2[[3]];
SL2s={S11f,S7f,S8f,S12f,S9f,S10f};
cL2s=Table[(SL2s[[i,1]]SL2s[[i,2]])/(SL2s[[i,1]].SL2s[[i,1]]),{i,1,6}];
JL2s=Table[{SL2s[[i,1]]/Norm[SL2s[[i,1]]],cL2s[[i]]SL2s[[i,1]]/Norm[SL2s[[i,1]]]},{i,1,6}];
SL2d=Table[{Append[JL2s[[i,1]],0],Append[JL2s[[i,2]],0]},{i,1,6}];
SL2m=Flatten[Append[{Table[quatmultSep[Fin1L1[[1]],SL2d[[i]]][{1,2},{1,2,3}]],{i,1,3}]],Table[quatmult[Fin3L1[[1]],SL2d[[i]]][{1,2},{1,2,3}]],{i,4,6}]],1];
S11f2=SL2m[[1]];
S7f2=SL2m[[2]];
S8f2=SL2m[[3]];
S12f2=SL2m[[4]];
S9f2=SL2m[[5]];
S10f2=SL2m[[6]];
JL2sN={S12f2,S9f2,S10f2};
SL2dN=Table[{Append[JL2sN[[i,1]],0],Append[JL2sN[[i,2]],0]},{i,1,3}];

SL2m3=Table[quatmultSep[quatconj[Fin1L1[[1]]],SL2dN[[i]]][{1,2},{1,2,3}]],{i,1,3}];
S12f3=SL2m3[[1]];
S9f3=SL2m3[[2]];
S10f3=SL2m3[[3]];

t0Solv={t01,t02,t03,t04,t05,t06,t07,t08,t09,t010,t011,t012,t013,t014,t015,t016,t017,t018,t019,t020,t021,t022,t023,t024,t025,t026,t027,t028,t029,t030,t031,t032,t033,t034,t035,t036,t037,t038,t039};t1Solv={t11,t12,t13,t14,t15,t16,t17,t18,t19,t110,t111,t112,t113,t114,t115,t116,t117,t118,t119,t120,t121,t122,t123,t124,t125,t126,t127,t128,t129,t130,t131,t132,t133,t134,t135,t136,t137,t138,t139};t2Solv={t21,t22,t23,t24,t25,t26,t27,t28,t29,t210,t211,t212,t213,t214,t215,t216,t217,t218,t219,t220,t221,t222,t223,t224,t225,t226,t227,t228,t229,t230,t231,t232,t233,t234,t235,t236,t237,t238,t239};
t3Solv={t31,t32,t33,t34,t35,t36,t37,t38,t39,t310,t311,t312,t313,t314,t315,t316,t317,t318,t319,t320,t321,t322,t323,t324,t325,t326,t327,t328,t329,t330,t331,t332,t333,t334,t335,t336,t337,t338,t339};
tSolve=Flatten[{t0Solv,t1Solv,t2Solv,t3Solv}];
SSolv={{S0x,S0y,S0z},{Sm0x,Sm0y,Sm0z}},{S1x,S1y,S1z},{Smlx,Smly,Sm1z}},{S2x,S2y,S2z},{Sm2x,Sm2y,Sm2z}},{S3x,S3y,S3z},{Sm3x,Sm3y,Sm3z}};
S0dqSolv=Table[makequat[SSolv[[1]],t0Solv[[i]],0],{i,1,39}];
S1dqSolv=Table[makequat[SSolv[[2]],t1Solv[[i]],0],{i,1,39}];
S2dqSolv=Table[makequat[SSolv[[3]],t2Solv[[i]],0],{i,1,39}];
S3dqSolv=Table[makequat[SSolv[[4]],t3Solv[[i]],0],{i,1,39}];
S11dqSolv=Table[makequat[S11f,t11f[[i]],0],{i,1,39}];
S7dqSolv=Table[makequat[S7f,t7f[[i]],0],{i,1,39}];
S10dqSolv=Table[makequat[S10f3,t10f[[i]],0],{i,1,39}];

LeftHandSide=Table[quatmult[quatmult[quatmult[quatconj[S0dqSolv[[i]]],quatconj[S1dqSolv[[i]]],S2dqSolv[[i]]],S3dqSolv[[i]]],{i,1,39}];
RightHandSide=Table[quatmultSep[quatmultSep[quatmultSep[S7dqSolv[[i]],S11dqSolv[[i]]],RelativaMotion14[[i]]],quatconj[S10dqSolv[[i]]],{i,1,39}];
var=Flatten[{t0Solv,t1Solv,t2Solv,t3Solv,SSolv}];

```

```

eq1a=Table[LeftHandSide[[i,1]]-RightHandSide[[i,1]],{i,1,39}];
eq1b=Table[LeftHandSide[[i,2]]-RightHandSide[[i,2]],{i,1,39}];
eq2={SSolv[[1,1]].SSolv[[1,2]],SSolv[[2,1]].SSolv[[2,2]],SSolv[[3,1]].SSolv[[3,2
]],SSolv[[4,1]].SSolv[[4,2]]];
eq3={SSolv[[1,1]].SSolv[[1,1]]-1,SSolv[[2,1]].SSolv[[2,1]]-
1,SSolv[[3,1]].SSolv[[3,1]]-1,SSolv[[4,1]].SSolv[[4,1]]-1};
eq=Flatten[{eq1a, eq1b,eq2,eq3}];

Fun=Table[F[k],{k,1,Length[eq]}];
Length[eq];
Length[var];
LB=Flatten[Append[Table[-6.28,{i,1,Length[tSolve]}],Table[{-1,-1,-1,-1000,-
1000,-1000},{i,1,Length[SSolv]}]]];
UB=Flatten[Append[Table[6.28,
{i,1,Length[tSolve]}],Table[{1,1,1,1000,1000,1000},{i,1,Length[SSolv]}]]];
EqAddress3= FileNameJoin[{"Equation3.m"}];
Equation3=OpenWrite[EqAddress3,PageWidth->300];
WriteString[Equation3,"function F=Equation3(x) \n"]
WriteString[Equation3,"t01=x(1),t02=x(2),t03=x(3),t04=x(4),t05=x(5),t06=x(6),t07
=x(7),t08=x(8),t09=x(9),t010=x(10),t011=x(11),t012=x(12),t013=x(13),t014=x(14),t
015=x(15),t016=x(16),t017=x(17),t018=x(18),t019=x(19),t020=x(20),t021=x(21),t022
=x(22),t023=x(23),t024=x(24),t025=x(25),t026=x(26),t027=x(27),t028=x(28),t029=x(
29),t030=x(30),t031=x(31),t032=x(32),t033=x(33),t034=x(34),t035=x(35),t036=x(36)
,t037=x(37),t038=x(38),t039=x(39),t11=x(40),t12=x(41),t13=x(42),t14=x(43),t15=x(
44),t16=x(45),t17=x(46),t18=x(47),t19=x(48),t110=x(49),t111=x(50),t112=x(51),t11
3=x(52),t114=x(53),t115=x(54),t116=x(55),t117=x(56),t118=x(57),t119=x(58),t120=x
(59),t121=x(60),t122=x(61),t123=x(62),t124=x(63),t125=x(64),t126=x(65),t127=x(66)
,t128=x(67),t129=x(68),t130=x(69),t131=x(70),t132=x(71),t133=x(72),t134=x(73),t
135=x(74),t136=x(75),t137=x(76),t138=x(77),t139=x(78),t21=x(79),t22=x(80),t23=x(
81),t24=x(82),t25=x(83),t26=x(84),t27=x(85),t28=x(86),t29=x(87),t210=x(88),t211=
x(89),t212=x(90),t213=x(91),t214=x(92),t215=x(93),t216=x(94),t217=x(95),t218=x(
96),t219=x(97),t220=x(98),t221=x(99),t222=x(100),t223=x(101),t224=x(102),t225=x(
103),t226=x(104),t227=x(105),t228=x(106),t229=x(107),t230=x(108),t231=x(109),t232
=x(110),t233=x(111),t234=x(112),t235=x(113),t236=x(114),t237=x(115),t238=x(116),
t239=x(117),t31=x(118),t32=x(119),t33=x(120),t34=x(121),t35=x(122),t36=x(123),t3
7=x(124),t38=x(125),t39=x(126),t310=x(127),t311=x(128),t312=x(129),t313=x(130),t
314=x(131),t315=x(132),t316=x(133),t317=x(134),t318=x(135),t319=x(136),t320=x(13
7),t321=x(138),t322=x(139),t323=x(140),t324=x(141),t325=x(142),t326=x(143),t327=
x(144),t328=x(145),t329=x(146),t330=x(147),t331=x(148),t332=x(149),t333=x(150),t
334=x(151),t335=x(152),t336=x(153),t337=x(154),t338=x(155),t339=x(156),S0x=x(157)
,S0y=x(158),S0z=x(159),Sm0x=x(160),Sm0y=x(161),Sm0z=x(162),S1x=x(163),S1y=x(164)
,S1z=x(165),Sm1x=x(166),Sm1y=x(167),Sm1z=x(168),S2x=x(169),S2y=x(170),S2z=x(171)
,Sm2x=x(172),Sm2y=x(173),Sm2z=x(174),S3x=x(175),S3y=x(176),S3z=x(177),Sm3x=x(17
8),Sm3y=x(179),Sm3z=x(180);\n\n"];
Table[WriteMatlab[eq[[i]],Equation3, Fun[[i]],{i,1,Length[eq]}];
Close[Equation3];
xinitial=Table[2*Random[]-1,{i,1,Length[var]}];
(*xinitial=xc*)
AddressRun3= FileNameJoin[{"run3.m"}];
run3=OpenWrite[AddressRun3,PageWidth->300];
OutputStream[AddressRun3,125];
WriteString[run3,"options=optimoptions('lsqnonlin','Display','iter',
'MaxIter',500,'TolFun',1.0000e-12,
'MaxFunEvals',1500000,'PlotFcns',{optimplotfirstorderopt})\n\n"];
WriteMatlab[xinitial,run3, x0];
WriteMatlab[LB,run3, lb];
WriteMatlab[UB,run3, ub];
WriteString[run3," x=lsqnonlin(@Equation3,x0, lb, ub,options);\n
save('datax3b','x') \n"]
Close[run3];

```

Results

```

x3=Flatten[Import["datax3b.mat"]];

t01out=x3[[1]];t02out=x3[[2]];t03out=x3[[3]];t04out=x3[[4]];t05out=x3[[5]];t06out=x3[[6]];t07out=x3[[7]];t08out=x3[[8]];t09out=x3[[9]];t010out=x3[[10]];t011out=x3[[11]];t012out=x3[[12]];t013out=x3[[13]];t014out=x3[[14]];t015out=x3[[15]];t016out=x3[[16]];t017out=x3[[17]];t018out=x3[[18]];t019out=x3[[19]];t020out=x3[[20]];t021out=x3[[21]];t022out=x3[[22]];t023out=x3[[23]];t024out=x3[[24]];t025out=x3[[25]];t026out=x3[[26]];t027out=x3[[27]];t028out=x3[[28]];t029out=x3[[29]];t030out=x3[[30]];t031out=x3[[31]];t032out=x3[[32]];t033out=x3[[33]];t034out=x3[[34]];t035out=x3[[35]];t036out=x3[[36]];t037out=x3[[37]];t038out=x3[[38]];t039out=x3[[39]];t11out=x3[[40]];t12out=x3[[41]];t13out=x3[[42]];t14out=x3[[43]];t15out=x3[[44]];t16out=x3[[45]];t17out=x3[[46]];t18out=x3[[47]];t19out=x3[[48]];t110out=x3[[49]];t111out=x3[[50]];t112out=x3[[51]];t113out=x3[[52]];t114out=x3[[53]];t115out=x3[[54]];t116out=x3[[55]];t117out=x3[[56]];t118out=x3[[57]];t119out=x3[[58]];t120out=x3[[59]];t121out=x3[[60]];t122out=x3[[61]];t123out=x3[[62]];t124out=x3[[63]];t125out=x3[[64]];t126out=x3[[65]];t127out=x3[[66]];t128out=x3[[67]];t129out=x3[[68]];t130out=x3[[69]];t131out=x3[[70]];t132out=x3[[71]];t133out=x3[[72]];t134out=x3[[73]];t135out=x3[[74]];t136out=x3[[75]];t137out=x3[[76]];t138out=x3[[77]];t139out=x3[[78]];t21out=x3[[79]];t22out=x3[[80]];t23out=x3[[81]];t24out=x3[[82]];t25out=x3[[83]];t26out=x3[[84]];t27out=x3[[85]];t28out=x3[[86]];t29out=x3[[87]];t210out=x3[[88]];t211out=x3[[89]];t212out=x3[[90]];t213out=x3[[91]];t214out=x3[[92]];t215out=x3[[93]];t216out=x3[[94]];t217out=x3[[95]];t218out=x3[[96]];t219out=x3[[97]];t220out=x3[[98]];t221out=x3[[99]];t222out=x3[[100]];t223out=x3[[101]];t224out=x3[[102]];t225out=x3[[103]];t226out=x3[[104]];t227out=x3[[105]];t228out=x3[[106]];t229out=x3[[107]];t230out=x3[[108]];t231out=x3[[109]];t232out=x3[[110]];t233out=x3[[111]];t234out=x3[[112]];t235out=x3[[113]];t236out=x3[[114]];t237out=x3[[115]];t238out=x3[[116]];t239out=x3[[117]];t31out=x3[[118]];t32out=x3[[119]];t33out=x3[[120]];t34out=x3[[121]];t35out=x3[[122]];t36out=x3[[123]];t37out=x3[[124]];t38out=x3[[125]];t39out=x3[[126]];t310out=x3[[127]];t311out=x3[[128]];t312out=x3[[129]];t313out=x3[[130]];t314out=x3[[131]];t315out=x3[[132]];t316out=x3[[133]];t317out=x3[[134]];t318out=x3[[135]];t319out=x3[[136]];t320out=x3[[137]];t321out=x3[[138]];t322out=x3[[139]];t323out=x3[[140]];t324out=x3[[141]];t325out=x3[[142]];t326out=x3[[143]];t327out=x3[[144]];t328out=x3[[145]];t329out=x3[[146]];t330out=x3[[147]];t331out=x3[[148]];t332out=x3[[149]];t333out=x3[[150]];t334out=x3[[151]];t335out=x3[[152]];t336out=x3[[153]];t337out=x3[[154]];t338out=x3[[155]];t339out=x3[[156]];S0xout=x3[[157]];S0yout=x3[[158]];S0zout=x3[[159]];Sm0xout=x3[[160]];Sm0yout=x3[[161]];Sm0zout=x3[[162]];S1xout=x3[[163]];S1yout=x3[[164]];S1zout=x3[[165]];Smlxout=x3[[166]];Smlyout=x3[[167]];Smlzout=x3[[168]];S2xout=x3[[169]];S2yout=x3[[170]];S2zout=x3[[171]];Sm2xout=x3[[172]];Sm2yout=x3[[173]];Sm2zout=x3[[174]];S3xout=x3[[175]];S3yout=x3[[176]];S3zout=x3[[177]];Sm3xout=x3[[178]];Sm3yout=x3[[179]];Sm3zout=x3[[180]];

t0Solvout3={t01out, t02out, t03out, t04out, t05out, t06out, t07out, t08out, t09out, t010out, t011out, t012out, t013out, t014out, t015out, t016out, t017out, t018out, t019out, t020out, t021out, t022out, t023out, t024out, t025out, t026out, t027out, t028out, t029out, t030out, t031out, t032out, t033out, t034out, t035out, t036out, t037out, t038out, t039out};t1Solvout3={t11out, t12out, t13out, t14out, t15out, t16out, t17out, t18out, t19out, t110out, t111out, t112out, t113out, t114out, t115out, t116out, t117out, t118out, t119out, t120out, t121out, t122out, t123out, t124out, t125out, t126out, t127out, t128out, t129out, t130out, t131out, t132out, t133out, t134out, t135out, t136out, t137out, t138out, t139out};t2Solvout3={t21out, t22out, t23out, t24out, t25out, t26out, t27out, t28out, t29out, t210out, t211out, t212out, t213out, t214out, t215out, t216out, t217out, t218out, t219out, t220out, t221out, t222out, t223out, t224out, t225out, t226out, t227out, t228out, t229out, t230out, t231out, t232out, t233out, t234out, t235out, t236out, t237out, t238out, t239out};

t3Solvout3={t31out, t32out, t33out, t34out, t35out, t36out, t37out, t38out, t39out, t310out, t311out, t312out, t313out, t314out, t315out, t316out, t317out, t318out, t319out, t320out, t321out, t322out, t323out, t324out, t325out, t326out, t327out, t328out, t329out, t330out, t331out, t332out, t333out, t334out, t335out, t336out, t337out, t338out, t339out};

tSolveout3={t0Solvout3, t1Solvout3, t2Solvout3, t3Solvout3};
SSolvout3={{S0xout,S0yout,S0zout},{Sm0xout,Sm0yout,Sm0zout}},{S1xout,S1yout,S1zout},{Smlxout,Smlyout,Smlzout}},{S2xout,S2yout,S2zout},{Sm2xout,Sm2yout,Sm2zout}

```

```
t}}, {{S3xout, S3yout, S3zout}, {Sm3xout, Sm3yout, Sm3zout}}};
```

Solving Level 3-With Arm

Calculations

We had SSolv0 is S11, SSolv1 is S7, SSolv2 is S8, also in second gripper SSolv0 is S12, SSolv1 is S9, SSolv2 is S10

Here we have SSolv0 is S5, SSolv1 is S3, SSolv2 is S4 and SSSolv3 is S6

```
S5f=SSolvout3[[1]];
S3f=SSolvout3[[2]];
S4f=SSolvout3[[3]];
S6f=SSolvout3[[4]];
t5f=-tSolveout3[[1]];
t3f=-tSolveout3[[2]];
t4f=tSolveout3[[3]];
t6f=tSolveout3[[4]];

SL3s={S5f, S3f, S4f, S6f};
cL3s=Table[(SL3s[[i, 1]]SL3s[[i, 2]])/(SL3s[[i, 1]].SL3s[[i, 1]]), {i, 1, 4}];
JL3s=Table[{SL3s[[i, 1]]/Norm[SL3s[[i, 1]]], cL3s[[i]]SL3s[[i, 1]]/Norm[SL3s[[i, 1]]]}, {i, 1, 4}];
SL3d=Table[{Append[JL3s[[i, 1]], 0], Append[JL3s[[i, 2]], 0]}, {i, 1, 4}];
SL3m=Table[quatmultSep[Fin1L1[[1]], SL3d[[i]]][{{1, 2}, {1, 2, 3}}], {i, 1, 4}];
S5f2=SL3m[[1]];
S3f2=SL3m[[2]];
S4f2=SL3m[[3]];
S6f2=SL3m[[4]];

JL3sN={S5f2, S3f2, S4f2, S6f2};

SL3dN=Table[{Append[JL3sN[[i, 1]], 0], Append[JL3sN[[i, 2]], 0]}, {i, 1, 4}];
SL2m3=Table[quatmultSep[quatconj[Fin1L1[[1]]], SL3dN[[i]]][{{1, 2}, {1, 2, 3}}], {i, 1, 4}];
S5f3=SL2m3[[1]];
S3f3=SL2m3[[2]];
S4f3=SL2m3[[3]];
S6f3=SL2m3[[4]];
```

```
JL3sN={S5f2, S3f2, S4f2, S6f2};
```

```
SL3dN=Table[{Append[JL3sN[[i, 1]], 0], Append[JL3sN[[i, 2]], 0]}, {i, 1, 4}];
SL2m3=Table[quatmultSep[quatconj[Fin1L1[[1]]], SL3dN[[i]]][{{1, 2}, {1, 2, 3}}], {i, 1, 4}];
S5f3=SL2m3[[1]];
S3f3=SL2m3[[2]];
S4f3=SL2m3[[3]];
S6f3=SL2m3[[4]];
```

Here we take two three axes in direction of z and two in Y in unknown locations.

```
JR0d={0, 0, 1}; JR1d={0, 1, 0}; JR2d={0, 1, 0}; CR0={C0x, C0y, C0z}; CR1={C1x, C1y, C1z}; CR2={C2x, C2y, C2z};
CR=Flatten[{CR0, CR1, CR2}];
JR0={JR0d, CR0JR0d}; JR1={JR1d, CR1JR1d}; JR2={JR2d, CR2JR2d};
tR0Solv={tR01, tR02, tR03, tR04, tR05, tR06, tR07, tR08, tR09, tR010, tR011, tR012, tR013, tR014, tR015, tR016, tR017, tR018, tR019, tR020, tR021, tR022, tR023, tR024, tR025, tR026, tR027, tR028, tR029, tR030, tR031, tR032, tR033, tR034, tR035, tR036, tR037, tR038, tR039}; tR1Solv={tR11, tR12, tR13, tR14, tR15, tR16, tR17, tR18, tR19, tR110, tR111, tR112, tR113, tR114, tR115, tR116, tR117, tR118, tR119, tR120, tR121, tR122, tR123, tR124, tR125, tR126, tR127, tR128, tR129, tR130, tR131, tR132, tR133, tR134, tR135, tR136, tR137, tR138, tR139}; tR2Solv={tR21, tR22, tR23, tR24, tR25, tR26, tR27, tR28, tR29, tR210, tR211, tR212, tR213, tR214, tR215, tR216, tR217, tR218, tR219, tR220, tR221, tR222, tR223, tR224, tR225, tR226, tR227, tR228, tR229, tR230, tR231, tR232, tR233, tR234, tR235, tR236, tR237, tR238, tR239};

tRSolv=Flatten[{tR0Solv, tR1Solv, tR2Solv}];
SR0=Table[makequat[JR0, tR0Solv[[i]], 0], {i, 1, 39}];
SR1=Table[makequat[JR1, tR1Solv[[i]], 0], {i, 1, 39}]; SR2=Table[makequat[JR2, tR2Solv[[i]], 0], {i, 1, 39}];

t0Solv={t01, t02, t03, t04, t05, t06, t07, t08, t09, t010, t011, t012, t013, t014, t015, t016, t017, t018, t019, t020, t021, t022, t023, t024, t025, t026, t027, t028, t029, t030, t031, t032, t033, t034, t035, t036, t037, t038, t039}; t1Solv={t11, t12, t13, t14, t15, t16, t17, t18, t19, t110, t111, t112, t113, t114, t115, t116, t117, t118, t119, t120, t121, t122, t123, t124, t125, t126, t127, t128, t129, t130, t131, t132, t133, t134, t135, t136, t137, t138, t139};
```

```

tSolve=Flatten[{t0Solv,t1Solv}];
SSolv={{S0x,S0y,S0z},{Sm0x,Sm0y,Sm0z},{S1x,S1y,S1z},{Smlx,Smly,Smlz}}};
S0dqSolv=Table[makequat[SSolv[[1]],t0Solv[[i]],0],{i,1,39}];
S1dqSolv=Table[makequat[SSolv[[2]],t1Solv[[i]],0],{i,1,39}];

S3fdqSolv=Table[makequat[S3f,t3f[[i]],0],{i,1,39}];
S4fdqSolv=Table[makequat[S4f,t4f[[i]],0],{i,1,39}];
S5dqSolv=Table[makequat[S5f,t5f[[i]],0],{i,1,39}];
S6dqSolv=Table[makequat[S6f,t6f[[i]],0],{i,1,39}];
S7dqSolv=Table[makequat[S7f,t7f[[i]],0],{i,1,39}];
S10dqSolv=Table[makequat[S10f3,t10f[[i]],0],{i,1,39}];
S11dqSolv=Table[makequat[S11f,t11f[[i]],0],{i,1,39}];
LeftHandSide=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[SR0[[i]],SR1[
[i]],SR2[[i]],S0dqSolv[[i]],S1dqSolv[[i]],{i,1,39}];

RightHandSide=Table[quatmultSep[quatmult[quatmultSep[quatmultSep[RelativaMotion1
1[[i]],quatconj[S11dqSolv[[i]]],quatconj[S7dqSolv[[i]]],quatconj[S5dqSolv[[i]]
]],quatconj[S3fdqSolv[[i]]]],{i,1,39}];

var=Flatten[{t0Solv,t1Solv,SSolv,tRSolv,CR}];

eq1a=Table[LeftHandSide[[i,1]]-RightHandSide[[i,1]],{i,1,39}];
eq1b=Table[LeftHandSide[[i,2]]-RightHandSide[[i,2]],{i,1,39}];
eq2={SSolv[[1,1]].SSolv[[1,2]],SSolv[[2,1]].SSolv[[2,2]]};
eq3={SSolv[[1,1]].SSolv[[1,1]]-1,SSolv[[2,1]].SSolv[[2,1]]-1};
eq=Flatten[{eq1a,eq1b,eq2,eq3}];

Fun=Table[F[k],{k,1,Length[eq]}];
Length[eq];
Length[var];
LB=Flatten[Append[Table[-6.28,{i,1,Length[tSolve]}],Table[{-1,-1,-1,-1000,-
1000,-1000},{i,1,Length[SSolv]}]]];
UB=Flatten[Append[Table[6.28,
{i,1,Length[tSolve]}],Table[{1,1,1,1000,1000,1000},{i,1,Length[SSolv]}]]];
EqAddress4=FileNameJoin[{"Equation4.m"}];
Equation4=OpenWrite[EqAddress4,PageWidth->300];
WriteString[Equation4,"function F=Equation4(x) \n"]
WriteString[Equation4,"t01=x(1),t02=x(2),t03=x(3),t04=x(4),t05=x(5),t06=x(6),t07
=x(7),t08=x(8),t09=x(9),t010=x(10),t011=x(11),t012=x(12),t013=x(13),t014=x(14),t
015=x(15),t016=x(16),t017=x(17),t018=x(18),t019=x(19),t020=x(20),t021=x(21),t022
=x(22),t023=x(23),t024=x(24),t025=x(25),t026=x(26),t027=x(27),t028=x(28),t029=x(
29),t030=x(30),t031=x(31),t032=x(32),t033=x(33),t034=x(34),t035=x(35),t036=x(36)
,t037=x(37),t038=x(38),t039=x(39),t11=x(40),t12=x(41),t13=x(42),t14=x(43),t15=x(
44),t16=x(45),t17=x(46),t18=x(47),t19=x(48),t110=x(49),t111=x(50),t112=x(51),t11
3=x(52),t114=x(53),t115=x(54),t116=x(55),t117=x(56),t118=x(57),t119=x(58),t120=x
(59),t121=x(60),t122=x(61),t123=x(62),t124=x(63),t125=x(64),t126=x(65),t127=x(66)
,t128=x(67),t129=x(68),t130=x(69),t131=x(70),t132=x(71),t133=x(72),t134=x(73),t
135=x(74),t136=x(75),t137=x(76),t138=x(77),t139=x(78),S0x=x(79),S0y=x(80),S0z=x(
81),Sm0x=x(82),Sm0y=x(83),Sm0z=x(84),S1x=x(85),S1y=x(86),S1z=x(87),Smlx=x(88),Sm
ly=x(89),Smlz=x(90),tR01=x(91),tR02=x(92),tR03=x(93),tR04=x(94),tR05=x(95),tR06=
x(96),tR07=x(97),tR08=x(98),tR09=x(99),tR010=x(100),tR011=x(101),tR012=x(102),tR
013=x(103),tR014=x(104),tR015=x(105),tR016=x(106),tR017=x(107),tR018=x(108),tR01
9=x(109),tR020=x(110),tR021=x(111),tR022=x(112),tR023=x(113),tR024=x(114),tR025=
x(115),tR026=x(116),tR027=x(117),tR028=x(118),tR029=x(119),tR030=x(120),tR031=x(
121),tR032=x(122),tR033=x(123),tR034=x(124),tR035=x(125),tR036=x(126),tR037=x(12
7),tR038=x(128),tR039=x(129),tR11=x(130),tR12=x(131),tR13=x(132),tR14=x(133),tR1
5=x(134),tR16=x(135),tR17=x(136),tR18=x(137),tR19=x(138),tR110=x(139),tR111=x(14
0),tR112=x(141),tR113=x(142),tR114=x(143),tR115=x(144),tR116=x(145),tR117=x(146)
,tR118=x(147),tR119=x(148),tR120=x(149),tR121=x(150),tR122=x(151),tR123=x(152),t
R124=x(153),tR125=x(154),tR126=x(155),tR127=x(156),tR128=x(157),tR129=x(158),tR1
30=x(159),tR131=x(160),tR132=x(161),tR133=x(162),tR134=x(163),tR135=x(164),tR136
=x(165),tR137=x(166),tR138=x(167),tR139=x(168),tR21=x(169),tR22=x(170),tR23=x(17

```



```

1), tR24=x(172), tR25=x(173), tR26=x(174), tR27=x(175), tR28=x(176), tR29=x(177), tR210
=x(178), tR211=x(179), tR212=x(180), tR213=x(181), tR214=x(182), tR215=x(183), tR216=x
(184), tR217=x(185), tR218=x(186), tR219=x(187), tR220=x(188), tR221=x(189), tR222=x(1
90), tR223=x(191), tR224=x(192), tR225=x(193), tR226=x(194), tR227=x(195), tR228=x(196
), tR229=x(197), tR230=x(198), tR231=x(199), tR232=x(200), tR233=x(201), tR234=x(202),
tR235=x(203), tR236=x(204), tR237=x(205), tR238=x(206), tR239=x(207), C0x=x(208), C0y=
x(209), C0z=x(210), C1x=x(211), C1y=x(212), C1z=x(213), C2x=x(214), C2y=x(215), C2z=x(2
16);\n\n";
Table[WriteMatlab[eq[[i]],Equation4, Fun[[i]]],{i,1,Length[eq]}};
Close[Equation4];
xinitial=Table[2*Random[]-1,{i,1,Length[var]}};
(*xinitial=xc*)
AddressRun4= FileNameJoin[{"run4.m"}];
run4=OpenWrite[AddressRun4,PageWidth->300];
OutputStream[AddressRun4,125];
WriteString[run4,"options=optimoptions('lsqnonlin','Display','iter',
'MaxIter',700,'TolFun',1.0000e-12,
'MaxFunEvals',1500000,'PlotFcns',{@optimplotfirstorderopt})\n\n"];
WriteMatlab[xinitial,run4, x0];
WriteMatlab[LB,run4, lb];
WriteMatlab[UB,run4, ub];
WriteString[run4," x=lsqnonlin(@Equation4,x0, lb, ub,options);\n
save('datax4b','x') \n"]
Close[run4];

```

results

```

x4=Flatten[Import["datax4b.mat"]];

```

```

t01out=x4[[1]];t02out=x4[[2]];t03out=x4[[3]];t04out=x4[[4]];t05out=x4[[5]];t06ou
t=x4[[6]];t07out=x4[[7]];t08out=x4[[8]];t09out=x4[[9]];t010out=x4[[10]];t011out=
x4[[11]];t012out=x4[[12]];t013out=x4[[13]];t014out=x4[[14]];t015out=x4[[15]];t01
6out=x4[[16]];t017out=x4[[17]];t018out=x4[[18]];t019out=x4[[19]];t020out=x4[[20]
];t021out=x4[[21]];t022out=x4[[22]];t023out=x4[[23]];t024out=x4[[24]];t025out=x4
[[25]];t026out=x4[[26]];t027out=x4[[27]];t028out=x4[[28]];t029out=x4[[29]];t030o
ut=x4[[30]];t031out=x4[[31]];t032out=x4[[32]];t033out=x4[[33]];t034out=x4[[34]];
t035out=x4[[35]];t036out=x4[[36]];t037out=x4[[37]];t038out=x4[[38]];t039out=x4[[
39]];t11out=x4[[40]];t12out=x4[[41]];t13out=x4[[42]];t14out=x4[[43]];t15out=x4[[
44]];t16out=x4[[45]];t17out=x4[[46]];t18out=x4[[47]];t19out=x4[[48]];t110out=x4[[
49]];t111out=x4[[50]];t112out=x4[[51]];t113out=x4[[52]];t114out=x4[[53]];t115ou
t=x4[[54]];t116out=x4[[55]];t117out=x4[[56]];t118out=x4[[57]];t119out=x4[[58]];t
120out=x4[[59]];t121out=x4[[60]];t122out=x4[[61]];t123out=x4[[62]];t124out=x4[[6
3]];t125out=x4[[64]];t126out=x4[[65]];t127out=x4[[66]];t128out=x4[[67]];t129out=
x4[[68]];t130out=x4[[69]];t131out=x4[[70]];t132out=x4[[71]];t133out=x4[[72]];t13
4out=x4[[73]];t135out=x4[[74]];t136out=x4[[75]];t137out=x4[[76]];t138out=x4[[77]
];t139out=x4[[78]];S0xout=x4[[79]];S0yout=x4[[80]];S0zout=x4[[81]];Sm0xout=x4[[8
2]];Sm0yout=x4[[83]];Sm0zout=x4[[84]];S1xout=x4[[85]];S1yout=x4[[86]];S1zout=x4[[
87]];Smlxout=x4[[88]];Smlyout=x4[[89]];Smlzout=x4[[90]];tR01out=x4[[91]];tR02ou
t=x4[[92]];tR03out=x4[[93]];tR04out=x4[[94]];tR05out=x4[[95]];tR06out=x4[[96]];t
R07out=x4[[97]];tR08out=x4[[98]];tR09out=x4[[99]];tR010out=x4[[100]];tR011out=x4
[[101]];tR012out=x4[[102]];tR013out=x4[[103]];tR014out=x4[[104]];tR015out=x4[[10
5]];tR016out=x4[[106]];tR017out=x4[[107]];tR018out=x4[[108]];tR019out=x4[[109]];
tR020out=x4[[110]];tR021out=x4[[111]];tR022out=x4[[112]];tR023out=x4[[113]];tR02
4out=x4[[114]];tR025out=x4[[115]];tR026out=x4[[116]];tR027out=x4[[117]];tR028ou
t=x4[[118]];tR029out=x4[[119]];tR030out=x4[[120]];tR031out=x4[[121]];tR032out=x4[[
122]];tR033out=x4[[123]];tR034out=x4[[124]];tR035out=x4[[125]];tR036out=x4[[126]
];tR037out=x4[[127]];tR038out=x4[[128]];tR039out=x4[[129]];tR11out=x4[[130]];tR
12out=x4[[131]];tR13out=x4[[132]];tR14out=x4[[133]];tR15out=x4[[134]];tR16out=x4
[[135]];tR17out=x4[[136]];tR18out=x4[[137]];tR19out=x4[[138]];tR110out=x4[[139]]
;tR111out=x4[[140]];tR112out=x4[[141]];tR113out=x4[[142]];tR114out=x4[[143]];tR1
15out=x4[[144]];tR116out=x4[[145]];tR117out=x4[[146]];tR118out=x4[[147]];tR119ou
t=x4[[148]];tR120out=x4[[149]];tR121out=x4[[150]];tR122out=x4[[151]];tR123out=x4
[[152]];tR124out=x4[[153]];tR125out=x4[[154]];tR126out=x4[[155]];tR127out=x4[[15
6]];tR128out=x4[[157]];tR129out=x4[[158]];tR130out=x4[[159]];tR131out=x4[[160]];
tR132out=x4[[161]];tR133out=x4[[162]];tR134out=x4[[163]];tR135out=x4[[164]];tR13

```

```

6out=x4[[165]];tR137out=x4[[166]];tR138out=x4[[167]];tR139out=x4[[168]];tR21out=
x4[[169]];tR22out=x4[[170]];tR23out=x4[[171]];tR24out=x4[[172]];tR25out=x4[[173]
];tR26out=x4[[174]];tR27out=x4[[175]];tR28out=x4[[176]];tR29out=x4[[177]];tR210o
ut=x4[[178]];tR211out=x4[[179]];tR212out=x4[[180]];tR213out=x4[[181]];tR214out=x
4[[182]];tR215out=x4[[183]];tR216out=x4[[184]];tR217out=x4[[185]];tR218out=x4[[1
86]];tR219out=x4[[187]];tR220out=x4[[188]];tR221out=x4[[189]];tR222out=x4[[190]]
;tR223out=x4[[191]];tR224out=x4[[192]];tR225out=x4[[193]];tR226out=x4[[194]];tR2
27out=x4[[195]];tR228out=x4[[196]];tR229out=x4[[197]];tR230out=x4[[198]];tR231ou
t=x4[[199]];tR232out=x4[[200]];tR233out=x4[[201]];tR234out=x4[[202]];tR235out=x4
[[203]];tR236out=x4[[204]];tR237out=x4[[205]];tR238out=x4[[206]];tR239out=x4[[20
7]];C0xout=x4[[208]];C0yout=x4[[209]];C0zout=x4[[210]];C1xout=x4[[211]];C1yout=x
4[[212]];C1zout=x4[[213]];C2xout=x4[[214]];C2yout=x4[[215]];C2zout=x4[[216]];

CR0out={C0xout,C0yout,C0zout};CR1out={C1xout,C1yout,C1zout};CR2out={C2xout,C2you
t,C2zout};
CRout=Flatten[{CR0out,CR1out,CR2out}];
JR0out={JR0d,CR0out-JR0d};JR1out={JR1d,CR1out-JR1d};JR2out={JR2d,CR2out-JR2d};
tR0Solvout={tR01out,tR02out,tR03out,tR04out,tR05out,tR06out,tR07out,tR08out,tR09
out,tR010out,tR011out,tR012out,tR013out,tR014out,tR015out,tR016out,tR017out,tR01
8out,tR019out,tR020out,tR021out,tR022out,tR023out,tR024out,tR025out,tR026out,tR0
27out,tR028out,tR029out,tR030out,tR031out,tR032out,tR033out,tR034out,tR035out,tR
036out,tR037out,tR038out,tR039out};tR1Solvout={tR11out,tR12out,tR13out,tR14out,t
R15out,tR16out,tR17out,tR18out,tR19out,tR110out,tR111out,tR112out,tR113out,tR114
out,tR115out,tR116out,tR117out,tR118out,tR119out,tR120out,tR121out,tR122out,tR12
3out,tR124out,tR125out,tR126out,tR127out,tR128out,tR129out,tR130out,tR131out,tR1
32out,tR133out,tR134out,tR135out,tR136out,tR137out,tR138out,tR139out};tR2Solvout
={tR21out,tR22out,tR23out,tR24out,tR25out,tR26out,tR27out,tR28out,tR29out,tR210o
ut,tR211out,tR212out,tR213out,tR214out,tR215out,tR216out,tR217out,tR218out,tR219
out,tR220out,tR221out,tR222out,tR223out,tR224out,tR225out,tR226out,tR227out,tR22
8out,tR229out,tR230out,tR231out,tR232out,tR233out,tR234out,tR235out,tR236out,tR2
37out,tR238out,tR239out};
tR0Solvout=Flatten[{tR0Solvout,tR1Solvout,tR2Solvout}];
SR0out=Table[makequat[JR0out,tR0Solvout[[i]],0],{i,1,39}];
SR1out=Table[makequat[JR1out,tR1Solvout[[i]],0],{i,1,39}];SR2out=Table[makequat[
JR2out,tR2Solvout[[i]],0],{i,1,39}];
t0Solvout4={t01out,t02out,t03out,t04out,t05out,t06out,t07out,t08out,
t09out,t010out,t011out,t012out,t013out,t014out,t015out,t016out,t017out,
t018out,t019out,t020out,t021out,t022out,t023out,t024out,t025out,t026out,
t027out,t028out,t029out,t030out,t031out,t032out,t033out,t034out,t035out,
t036out,t037out,t038out,t039out};t1Solvout4={t11out,t12out,t13out,t14out,
t15out,t16out,t17out,t18out,t19out,t110out,t111out,t112out,t113out,
t114out,t115out,t116out,t117out,t118out,t119out,t120out,t121out,t122out,
t123out,t124out,t125out,t126out,t127out,t128out,t129out,t130out,t131out,
t132out,t133out,t134out,t135out,t136out,t137out,t138out,t139out};
tSolveout4={t0Solvout4,t1Solvout4};S1f2=SSolvout4[[1]];S2f2=SSolvout4[[2]];
t1f=tSolveout4[[1]];t2f=tSolveout4[[2]];t2f=tSolveout4[[2]];

```

Optimization

Kinematics

Forward Kinematics and Anchoring Points Calculation

Calculations

```

jointsAll={S1f2,S2f2,S3f2,S4f2,S5f2,S6f2,S7f2,S8f2,S9f2,S10f2,S11f2,S12f2};
θ={t1f,t2f,t3f,t4f,t5f,t6f,t7f,t8f,t9f,t10f,t11f,t12f};
Frame1HM=Table[dq2HM[Fin1L1[[i]]],{i,1,Length[Fin1L1]};
Frame2HM=Table[dq2HM[Fin2L1[[i]]],{i,1,Length[Fin1L1]};
Frame3HM=Table[dq2HM[Fin3L1[[i]]],{i,1,Length[Fin1L1]};
Frame4HM=Table[dq2HM[Fin4L1[[i]]],{i,1,Length[Fin1L1]};
CEff1=Table[Frame1HM[[i,{1,2,3},4]],{i,1,Length[Frame1HM]};
CEff2=Table[Frame2HM[[i,{1,2,3},4]],{i,1,Length[Frame1HM]};
CEff3=Table[Frame3HM[[i,{1,2,3},4]],{i,1,Length[Frame1HM]};

```

```

CEff4=Table[Frame4HM[[i,{1,2,3},4]],{i,1,Length[Frame1HM]};
Frame1dq=Table[HM2dq[Frame1HM[[i]]],{i,1,Length[Frame1HM]};
Frame2dq=Table[HM2dq[Frame2HM[[i]]],{i,1,Length[Frame1HM]};
Frame3dq=Table[HM2dq[Frame3HM[[i]]],{i,1,Length[Frame1HM]};
Frame4dq=Table[HM2dq[Frame4HM[[i]]],{i,1,Length[Frame1HM]};

```

```

j1={jointsAll[[1,1]],jointsAll[[1,2]]};
j2={jointsAll[[2,1]],jointsAll[[2,2]]};
j11=j12={jointsAll[[3,1]],jointsAll[[3,2]]};
j21=j22={jointsAll[[5,1]],jointsAll[[5,2]]};
j31={jointsAll[[7,1]],jointsAll[[7,2]]};
j32={jointsAll[[8,1]],jointsAll[[8,2]]};
j41={jointsAll[[11,1]],jointsAll[[11,2]]};
j13=j14={jointsAll[[4,1]],jointsAll[[4,2]]};
j23=j24={jointsAll[[6,1]],jointsAll[[6,2]]};
j33={jointsAll[[9,1]],jointsAll[[9,2]]};
j34={jointsAll[[10,1]],jointsAll[[10,2]]};
j43={jointsAll[[12,1]],jointsAll[[12,2]]};
c1=(j1[[1]]⊗j1[[2]])/(j1[[1]].j1[[1]]);
c2=(j2[[1]]⊗j2[[2]])/(j2[[1]].j2[[1]]);
c11=c12=(j11[[1]]⊗j11[[2]])/(j11[[1]].j11[[1]]);
c21=c22=(j21[[1]]⊗j21[[2]])/(j21[[1]].j21[[1]]);
c31=(j31[[1]]⊗j31[[2]])/(j31[[1]].j31[[1]]);
c32=(j32[[1]]⊗j32[[2]])/(j32[[1]].j32[[1]]);
c41=(j41[[1]]⊗j41[[2]])/(j41[[1]].j41[[1]]);
c13=c14=(j13[[1]]⊗j13[[2]])/(j13[[1]].j13[[1]]);
c23=c24=(j23[[1]]⊗j23[[2]])/(j23[[1]].j23[[1]]);
c33=(j33[[1]]⊗j33[[2]])/(j33[[1]].j33[[1]]);
c34=(j34[[1]]⊗j34[[2]])/(j34[[1]].j34[[1]]);
c43=(j43[[1]]⊗j43[[2]])/(j43[[1]].j43[[1]]);
CEff1=CEff1[[1]];
CEff2=CEff2[[1]];
CEff3=CEff3[[1]];

```

Now we normalize the axes

```

J1={j1[[1]]/Norm[j1[[1]]],c1⊗(j1[[1]]/Norm[j1[[1]]])};
J2={j2[[1]]/Norm[j2[[1]]],c2⊗(j2[[1]]/Norm[j2[[1]]])};
J11=J12={j11[[1]]/Norm[j11[[1]]],c11⊗(j11[[1]]/Norm[j11[[1]]])};
J21=J22={j21[[1]]/Norm[j21[[1]]],c21⊗(j21[[1]]/Norm[j21[[1]]])};
J31={j31[[1]]/Norm[j31[[1]]],c31⊗(j31[[1]]/Norm[j31[[1]]])};
J32={j32[[1]]/Norm[j32[[1]]],c32⊗(j32[[1]]/Norm[j32[[1]]])};
J41={j41[[1]]/Norm[j41[[1]]],c41⊗(j41[[1]]/Norm[j41[[1]]])};
J13=J14={j13[[1]]/Norm[j13[[1]]],c13⊗(j13[[1]]/Norm[j13[[1]]])};
J23=J24={j23[[1]]/Norm[j23[[1]]],c23⊗(j23[[1]]/Norm[j23[[1]]])};
J33={j33[[1]]/Norm[j33[[1]]],c33⊗(j33[[1]]/Norm[j33[[1]]])};J34={j34[[1]]/Norm[j34[[1]]],c34⊗(j34[[1]]/Norm[j34[[1]]])};
J43={j43[[1]]/Norm[j43[[1]]],c43⊗(j43[[1]]/Norm[j43[[1]]])};
S1q=Table[makequat[J1,θ[[1,i]],0],{i,1,39}];
S2q=Table[makequat[J2,θ[[2,i]],0],{i,1,39}];
S11q=S12q=Table[makequat[J11,θ[[3,i]],0],{i,1,39}];
S21q=S22q=Table[makequat[J21,θ[[5,i]],0],{i,1,39}];
S31q=Table[makequat[J31,θ[[7,i]],0],{i,1,39}];
S41q=Table[makequat[J31,θ[[5,i]],0],{i,1,39}];
S32q=Table[makequat[J32,θ[[11,i]],0],{i,1,39}];
S13q=S14q=Table[makequat[J13,θ[[4,i]],0],{i,1,39}];
S23q=S24q=Table[makequat[J23,θ[[6,i]],0],{i,1,39}];
S33q=Table[makequat[J33,θ[[9,i]],0],{i,1,39}];
S43q=Table[makequat[J33,θ[[10,i]],0],{i,1,39}];
S34q=Table[makequat[J34,θ[[12,i]],0],{i,1,39}];

```

At this step the Forward Kinematic of each branch from the wrist joint up to the last joint has been calculated:

```

FK1=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[SR0out[[i]],SR1out[[i]],SR2out[[i]],S1q[[i]],S2q[[i]],S11q[[i]],S21q[[i]],S31q[[i]],S41q[[i]],{i,1,39}]];
FK2=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[SR0out[[i]],SR1out[[i]],SR2out[[i]],S1q[[i]],S2q[[i]],S12q[[i]],S22q[[i]],S32q[[i]],{i,1,39}]];FK3=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[SR0out[[i]],SR1out[[i]],SR2out[[i]],S1q[[i]],S2q[[i]],S13q[[i]],S23q[[i]],S33q[[i]],S43q[[i]],{i,1,39}]];FK4=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[SR0out[[i]],SR1out[[i]],SR2out[[i]],S1q[[i]],S2q[[i]],S14q[[i]],S24q[[i]],S34q[[i]],{i,1,39}]];
FKrel1=Table[S1q[[i]],{i,1,39}]];
FKrel2=Table[quatmultSep[S1q[[i]],S2q[[i]],{i,1,39}]];
FKrel11=Table[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S11q[[i]],{i,1,39}]];
FKrel21=Table[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S11q[[i]],S21q[[i]],{i,1,39}]];
FKrel31=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S11q[[i]],S21q[[i]],S31q[[i]],{i,1,39}]];
FKrel41=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S11q[[i]],S21q[[i]],S31q[[i]],S41q[[i]],{i,1,39}]];

FKrel12=Table[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S12q[[i]],{i,1,39}]];
FKrel22=Table[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S12q[[i]],S22q[[i]],{i,1,39}]];
FKrel32=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S12q[[i]],S22q[[i]],S32q[[i]],{i,1,39}]];

FKrel13=Table[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S13q[[i]],{i,1,39}]];
FKrel23=Table[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S13q[[i]],S23q[[i]],{i,1,39}]];
FKrel33=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S13q[[i]],S23q[[i]],S33q[[i]],{i,1,39}]];
FKrel43=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S13q[[i]],S23q[[i]],S33q[[i]],S43q[[i]],{i,1,39}]];

FKrel14=Table[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S14q[[i]],{i,1,39}]];
FKrel24=Table[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S14q[[i]],S24q[[i]],{i,1,39}]];
FKrel34=Table[quatmultSep[quatmultSep[quatmultSep[quatmultSep[S1q[[i]],S2q[[i]],S14q[[i]],S24q[[i]],S34q[[i]],{i,1,39}]];

```

In order to find the anchoring point along each axes a reference point is needed for each axes, this point on each axes has been calculated as follows as the intersection point of the normal line from the origin to that line as C_i being the i th axes of the wrist, and C_{ij} being the point on the i th axes of the j th branch. Also the end effector first point has been mentioned for calculations. Similar naming method has been used for the point from now at each calculation step:

```

C1=(J1[[1]]J1[[2]])/(J1[[1]].J1[[1]]);
C2=(J2[[1]]J2[[2]])/(J2[[1]].J2[[1]]);
C11=(J11[[1]]J11[[2]])/(J11[[1]].J11[[1]]);
C21=(J21[[1]]J21[[2]])/(J21[[1]].J21[[1]]);
C31=(J31[[1]]J31[[2]])/(J31[[1]].J31[[1]]);
C12=(J12[[1]]J12[[2]])/(J12[[1]].J12[[1]]);
C22=(J22[[1]]J22[[2]])/(J22[[1]].J22[[1]]);
C32=(J32[[1]]J32[[2]])/(J32[[1]].J32[[1]]);
C41=(J41[[1]]J41[[2]])/(J41[[1]].J41[[1]]);
C13=(J13[[1]]J13[[2]])/(J13[[1]].J13[[1]]);
C23=(J23[[1]]J23[[2]])/(J23[[1]].J23[[1]]);
C33=(J33[[1]]J33[[2]])/(J33[[1]].J33[[1]]);
C14=(J14[[1]]J14[[2]])/(J14[[1]].J14[[1]]);
C24=(J24[[1]]J24[[2]])/(J24[[1]].J24[[1]]);

```

```

C34=(J34[[1]]J34[[2]])/(J34[[1]].J34[[1]]);
C43=(J43[[1]]J43[[2]])/(J43[[1]].J43[[1]]);
CEF1=CEff1[[1]];
CEF2=CEff2[[1]];
CEF3=CEff3[[1]];
CEF4=CEff4[[1]];

```

Now the pure quaternoin of the aforesaid points has been calculated as follows:

```

C1q={{0,0,0,1},Append[C1,0]};
C2q={{0,0,0,1},Append[C2,0]};
C11q={{0,0,0,1},Append[C11,0]};
C21q={{0,0,0,1},Append[C21,0]};
C31q={{0,0,0,1},Append[C31,0]};
CEF1q={{0,0,0,1},Append[CEF1,0]};
C12q={{0,0,0,1},Append[C12,0]};
C22q={{0,0,0,1},Append[C22,0]};
C32q={{0,0,0,1},Append[C32,0]};
C41q={{0,0,0,1},Append[C41,0]};
CEF2q={{0,0,0,1},Append[CEF2,0]};
C13q={{0,0,0,1},Append[C13,0]};
C23q={{0,0,0,1},Append[C23,0]};
C33q={{0,0,0,1},Append[C33,0]};
CEF3q={{0,0,0,1},Append[CEF3,0]};
C14q={{0,0,0,1},Append[C14,0]};
C24q={{0,0,0,1},Append[C24,0]};
C34q={{0,0,0,1},Append[C34,0]};
C43q={{0,0,0,1},Append[C43,0]};
CEF4q={{0,0,0,1},Append[CEF4,0]};

```

And at this step the forward kinematics of each axes are used to calculate the the moved dual quartenion of points on each axis:

```

C2mq=Table[quatmultSep[quatmultSep[FKrel1[[i]],C2q],quatconj4[FKrel1[[i]]]],{i,1,39}];
C11mq=Table[quatmultSep[quatmultSep[FKrel2[[i]],C11q],quatconj4[FKrel2[[i]]]],{i,1,39}];
C21mq=Table[quatmultSep[quatmultSep[FKrel11[[i]],C21q],quatconj4[FKrel11[[i]]]],{i,1,39}];
C31mq=Table[quatmultSep[quatmultSep[FKrel21[[i]],C31q],quatconj4[FKrel21[[i]]]],{i,1,39}];
CEF1mq=Table[quatmultSep[quatmultSep[FKrel31[[i]],CEF1q],quatconj4[FKrel31[[i]]]],{i,1,39}];
C12mq=Table[quatmultSep[quatmultSep[FKrel2[[i]],C12q],quatconj4[FKrel2[[i]]]],{i,1,39}];
C22mq=Table[quatmultSep[quatmultSep[FKrel12[[i]],C22q],quatconj4[FKrel12[[i]]]],{i,1,39}];
C32mq=Table[quatmultSep[quatmultSep[FKrel22[[i]],C32q],quatconj4[FKrel22[[i]]]],{i,1,39}];
C41mq=Table[quatmultSep[quatmultSep[FKrel31[[i]],C41q],quatconj4[FKrel31[[i]]]],{i,1,39}];
CEF2mq=Table[quatmultSep[quatmultSep[FKrel41[[i]],CEF2q],quatconj4[FKrel41[[i]]]],{i,1,39}];
C13mq=Table[quatmultSep[quatmultSep[FKrel2[[i]],C13q],quatconj4[FKrel2[[i]]]],{i,1,39}];
C23mq=Table[quatmultSep[quatmultSep[FKrel13[[i]],C23q],quatconj4[FKrel13[[i]]]],{i,1,39}];
C33mq=Table[quatmultSep[quatmultSep[FKrel23[[i]],C33q],quatconj4[FKrel23[[i]]]],{i,1,39}];
CEF3mq=Table[quatmultSep[quatmultSep[FKrel33[[i]],CEF3q],quatconj4[FKrel33[[i]]]],{i,1,39}];
C14mq=Table[quatmultSep[quatmultSep[FKrel2[[i]],C14q],quatconj4[FKrel2[[i]]]],{i,1,39}];
C24mq=Table[quatmultSep[quatmultSep[FKrel14[[i]],C24q],quatconj4[FKrel14[[i]]]],{i,1,39}];
C34mq=Table[quatmultSep[quatmultSep[FKrel24[[i]],C34q],quatconj4[FKrel24[[i]]]],{i,1,39}];

```

```

{i,1,39}};
C43mq=Table[quatmultSep[quatmultSep[FKrel133[[i]],C43q],quatconj4[FKrel133[[i]]],
{i,1,39}};
CEF4mq=Table[quatmultSep[quatmultSep[FKrel143[[i]],CEF4q],quatconj4[FKrel143[[i]]]
],{i,1,39}};

```

Here the position part for each of the calculated points ([[2,{1,2,3}]] part of each quaternion) have been gathered and added with the initial point of that set:

```

Cn2m=Prepend[Table[C2mq[[i,2,{1,2,3}]],{i,1,39}],C2];
Cn11m=Prepend[Table[C11mq[[i,2,{1,2,3}]],{i,1,39}],C11];
Cn21m=Prepend[Table[C21mq[[i,2,{1,2,3}]],{i,1,39}],C21];
Cn31m=Prepend[Table[C31mq[[i,2,{1,2,3}]],{i,1,39}],C31];
CnEF1m=Prepend[Table[CEF1mq[[i,2,{1,2,3}]],{i,1,39}],CEF1];
Cn12m=Prepend[Table[C12mq[[i,2,{1,2,3}]],{i,1,39}],C12];
Cn22m=Prepend[Table[C22mq[[i,2,{1,2,3}]],{i,1,39}],C22];
Cn32m=Prepend[Table[C32mq[[i,2,{1,2,3}]],{i,1,39}],C32];
Cn41m=Prepend[Table[C41mq[[i,2,{1,2,3}]],{i,1,39}],C41];
CnEF2m=Prepend[Table[CEF2mq[[i,2,{1,2,3}]],{i,1,39}],CEF2];
Cn13m=Prepend[Table[C13mq[[i,2,{1,2,3}]],{i,1,39}],C13];
Cn23m=Prepend[Table[C23mq[[i,2,{1,2,3}]],{i,1,39}],C23];
Cn33m=Prepend[Table[C33mq[[i,2,{1,2,3}]],{i,1,39}],C33];
CnEF3m=Prepend[Table[CEF3mq[[i,2,{1,2,3}]],{i,1,39}],CEF3];
Cn14m=Prepend[Table[C14mq[[i,2,{1,2,3}]],{i,1,39}],C14];
Cn24m=Prepend[Table[C24mq[[i,2,{1,2,3}]],{i,1,39}],C24];
Cn34m=Prepend[Table[C34mq[[i,2,{1,2,3}]],{i,1,39}],C34];
Cn43m=Prepend[Table[C43mq[[i,2,{1,2,3}]],{i,1,39}],C43];
CnEF4m=Prepend[Table[CEF4mq[[i,2,{1,2,3}]],{i,1,39}],CEF4];

```

Also to find the transformed axes their pure quaternions have been calculated as follows with Si being the ith joint of the wrist and Sij being the ith joint of the branch j:

```

S1d={Append[J1[[1]],0],Append[J1[[2]],0]};
S2d={Append[J2[[1]],0],Append[J2[[2]],0]};
S11d={Append[J11[[1]],0],Append[J11[[2]],0]};
S21d={Append[J21[[1]],0],Append[J21[[2]],0]};
S31d={Append[J31[[1]],0],Append[J31[[2]],0]};
S12d={Append[J12[[1]],0],Append[J12[[2]],0]};
S22d={Append[J22[[1]],0],Append[J22[[2]],0]};
S32d={Append[J32[[1]],0],Append[J32[[2]],0]};
S41d={Append[J41[[1]],0],Append[J41[[2]],0]};
S13d={Append[J13[[1]],0],Append[J13[[2]],0]};
S23d={Append[J23[[1]],0],Append[J23[[2]],0]};
S33d={Append[J33[[1]],0],Append[J33[[2]],0]};
S14d={Append[J14[[1]],0],Append[J14[[2]],0]};
S24d={Append[J24[[1]],0],Append[J24[[2]],0]};
S34d={Append[J34[[1]],0],Append[J34[[2]],0]};
S43d={Append[J43[[1]],0],Append[J43[[2]],0]};

```

So the forward kinematics are used to find the transformed dual quaternion of each axes:

```

S2m=Prepend[Table[quatmultSep[quatmultSep[FKrel1[[i]],S2d],quatconj[FKrel1[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S11d[{1,2},{1,2,3}]]];
S11m=Prepend[Table[quatmultSep[quatmultSep[FKrel12[[i]],S11d],quatconj[FKrel12[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S11d[{1,2},{1,2,3}]]];
S21m=Prepend[Table[quatmultSep[quatmultSep[FKrel11[[i]],S21d],quatconj[FKrel11[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S21d[{1,2},{1,2,3}]]];
S31m=Prepend[Table[quatmultSep[quatmultSep[FKrel21[[i]],S31d],quatconj[FKrel21[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S31d[{1,2},{1,2,3}]]];
S12m=Prepend[Table[quatmultSep[quatmultSep[FKrel12[[i]],S12d],quatconj[FKrel12[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S12d[{1,2},{1,2,3}]]];
S22m=Prepend[Table[quatmultSep[quatmultSep[FKrel12[[i]],S22d],quatconj[FKrel12[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S22d[{1,2},{1,2,3}]]];
S32m=Prepend[Table[quatmultSep[quatmultSep[FKrel22[[i]],S32d],quatconj[FKrel22[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S32d[{1,2},{1,2,3}]]];
S41m=Prepend[Table[quatmultSep[quatmultSep[FKrel31[[i]],S41d],quatconj[FKrel31[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S41d[{1,2},{1,2,3}]]];
S13m=Prepend[Table[quatmultSep[quatmultSep[FKrel12[[i]],S13d],quatconj[FKrel12[[i]]]
][{1,2},{1,2,3}]],{i,1,39}],S13d[{1,2},{1,2,3}]]];

```

```

]]][{{1,2},{1,2,3}}],{i,1,39}],S13d[{{1,2},{1,2,3}}]]];
S23m=Prepend[Table[quatmultSep[quatmultSep[FKrel13[[i]],S23d],quatconj[FKrel13[[i]]]][{{1,2},{1,2,3}}],{i,1,39}],S23d[{{1,2},{1,2,3}}]]];
S33m=Prepend[Table[quatmultSep[quatmultSep[FKrel23[[i]],S33d],quatconj[FKrel23[[i]]]][{{1,2},{1,2,3}}],{i,1,39}],S33d[{{1,2},{1,2,3}}]]];
S14m=Prepend[Table[quatmultSep[quatmultSep[FKrel2[[i]],S14d],quatconj[FKrel2[[i]]]][{{1,2},{1,2,3}}],{i,1,39}],S14d[{{1,2},{1,2,3}}]]];
S24m=Prepend[Table[quatmultSep[quatmultSep[FKrel13[[i]],S24d],quatconj[FKrel13[[i]]]][{{1,2},{1,2,3}}],{i,1,39}],S24d[{{1,2},{1,2,3}}]]];
S34m=Prepend[Table[quatmultSep[quatmultSep[FKrel23[[i]],S34d],quatconj[FKrel23[[i]]]][{{1,2},{1,2,3}}],{i,1,39}],S34d[{{1,2},{1,2,3}}]]];
S43m=Prepend[Table[quatmultSep[quatmultSep[FKrel33[[i]],S43d],quatconj[FKrel33[[i]]]][{{1,2},{1,2,3}}],{i,1,39}],S43d[{{1,2},{1,2,3}}]]];

```

At this step transformed points and transformed axes direction have been used to write each anchoring point as $P_{ijm}=t_{ij}*s_{ijm}+C_{ij}$; with the t_{ij} , C_{nijm} , s_{ijm} being parameter, reference point and the axis direction of the i th elements of the j th branch respectively. Also the P_i being the i th point of the wrist that was not moved in this topology because there is just one joint in the wrist:

```

P1=C1+t1*J1[[1]];
P2m=Table[S2m[[i,1,{1,2,3}]]*t2+Cn2m[[i]],{i,1,40}];
P11m=Table[S11m[[i,1,{1,2,3}]]*t11+Cn11m[[i]],{i,1,40}];
P21m=Table[S21m[[i,1,{1,2,3}]]*t21+Cn21m[[i]],{i,1,40}];
P31m=Table[S31m[[i,1,{1,2,3}]]*t31+Cn31m[[i]],{i,1,40}];
P41m=Table[S41m[[i,1,{1,2,3}]]*t41+Cn41m[[i]],{i,1,40}];
PEF1m=CnEF1m;

```

```

P22m=Table[S22m[[i,1,{1,2,3}]]*t22+Cn22m[[i]],{i,1,40}];
P32m=Table[S32m[[i,1,{1,2,3}]]*t32+Cn32m[[i]],{i,1,40}];

```

```

PEF2m=CnEF2m;
P13m=Table[S13m[[i,1,{1,2,3}]]*t13+Cn13m[[i]],{i,1,40}];
P23m=Table[S23m[[i,1,{1,2,3}]]*t23+Cn23m[[i]],{i,1,40}];
P33m=Table[S33m[[i,1,{1,2,3}]]*t33+Cn33m[[i]],{i,1,40}];
P43m=Table[S43m[[i,1,{1,2,3}]]*t43+Cn43m[[i]],{i,1,40}];
PEF3m=CnEF3m;

```

```

P24m=Table[S24m[[i,1,{1,2,3}]]*t24+Cn24m[[i]],{i,1,40}];
P34m=Table[S34m[[i,1,{1,2,3}]]*t34+Cn34m[[i]],{i,1,40}];

```

```

PEF4m=CnEF4m;

```

At this step as a test for the kinematic of the hand and points to have been calculated correctly and every part to be addressed correctly each link or end-effector lengths has been calculated to check that it remains unchanged during the rotations, for that at this step the parameters has been given the zero value in the points calculated in the previous step. As it is obvious it has been found that all links and effectors have the same lengths at each of the positions of the hand that confirms the anchoring points correct calculation also it's a second check for the kinematics and synthesis results:

plots

```

P1n=P1/.t1-> 0;
P2n=Table[P2m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0, t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41->0},{i,1,40}];
P11n=Table[P11m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0, t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41->0},{i,1,40}];
P21n=Table[P21m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0, t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41->0},{i,1,40}];
P31n=Table[P31m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0, t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41->0},{i,1,40}];
PEF1n=Table[PEF1m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0, t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41->0},{i,1,40}];
P12n=Table[P11m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0, t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41->0},{i,1,40}];

```

```

>0},{i,1,40}];
P22n=Table[P22m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P32n=Table[P32m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P41n=Table[P41m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
PEF2n=Table[PEF2m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P13n=Table[P13m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P23n=Table[P23m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P33n=Table[P33m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
PEF3n=Table[PEF3m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P14n=Table[P13m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P24n=Table[P24m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P34n=Table[P34m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
P43n=Table[P43m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];
PEF4n=Table[PEF4m[[i]]/.{t1-> 0,t2-> 0,t11-> 0,t21->0, t31->0, t12-> 0,t22->0,
t32->0, t13-> 0,t23->0, t33->0, t14-> 0,t24->0, t34->0, t43->0, t41-
>0},{i,1,40}];

```

```

length1=Table[Sqrt[((P1n-P2n[[i]])).(P1n-P2n[[i]])]),{i,1,40}]
length2=Table[Sqrt[((P2n[[i]]-P11n[[i]])).(P2n[[i]]-P11n[[i]])]),{i,1,40}]
length3=Table[Sqrt[((P11n[[i]]-P21n[[i]])).(P11n[[i]]-P21n[[i]])]),{i,1,40}]
length4=Table[Sqrt[((P21n[[i]]-P31n[[i]])).(P21n[[i]]-P31n[[i]])]),{i,1,40}]
length5=Table[Sqrt[((P31n[[i]]-P41n[[i]])).(P31n[[i]]-P41n[[i]])]),{i,1,40}]
length6=Table[Sqrt[((P41n[[i]]-CEff1[[i]])).(P41n[[i]]-CEff1[[i]])]),{i,1,40}]
length7=Table[Sqrt[((P2n[[i]]-P12n[[i]])).(P2n[[i]]-P12n[[i]])]),{i,1,40}]
length8=Table[Sqrt[((P12n[[i]]-P22n[[i]])).(P12n[[i]]-P22n[[i]])]),{i,1,40}]
length9=Table[Sqrt[((P22n[[i]]-P32n[[i]])).(P22n[[i]]-P32n[[i]])]),{i,1,40}]
length10=Table[Sqrt[((P32n[[i]]-CEff2[[i]])).(P32n[[i]]-CEff2[[i]])]),{i,1,40}]
length11=Table[Sqrt[((P13n[[i]]-P23n[[i]])).(P13n[[i]]-P23n[[i]])]),{i,1,40}]
length12=Table[Sqrt[((P23n[[i]]-P33n[[i]])).(P23n[[i]]-P33n[[i]])]),{i,1,40}]
length13=Table[Sqrt[((P33n[[i]]-P43n[[i]])).(P33n[[i]]-P43n[[i]])]),{i,1,40}]
length14=Table[Sqrt[((P43n[[i]]-CEff3[[i]])).(P43n[[i]]-CEff3[[i]])]),{i,1,40}]
length15=Table[Sqrt[((P2n[[i]]-P14n[[i]])).(P2n[[i]]-P14n[[i]])]),{i,1,40}]
length16=Table[Sqrt[((P14n[[i]]-P24n[[i]])).(P14n[[i]]-P24n[[i]])]),{i,1,40}]
length17=Table[Sqrt[((P24n[[i]]-P34n[[i]])).(P24n[[i]]-P34n[[i]])]),{i,1,40}]
length18=Table[Sqrt[((P34n[[i]]-CEff4[[i]])).(P34n[[i]]-CEff4[[i]])]),{i,1,40}]

```

```

{2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94
,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94
,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94,2426.94

```



```

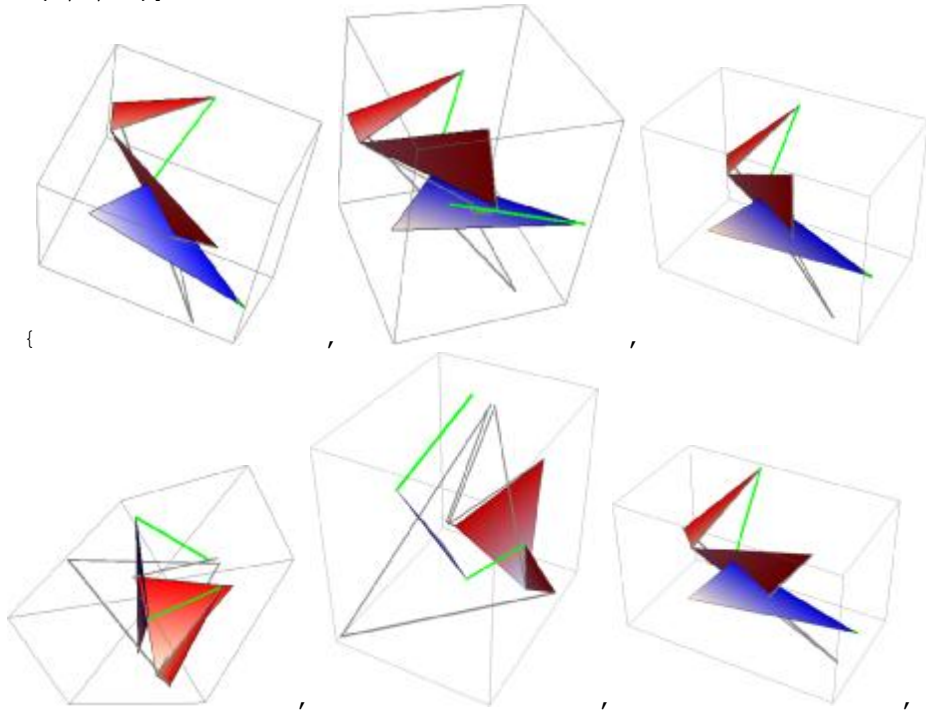
Line[{P1n,P2n[[i]]},VertexColors->VertexColor[[4]]],Thickness[.01],

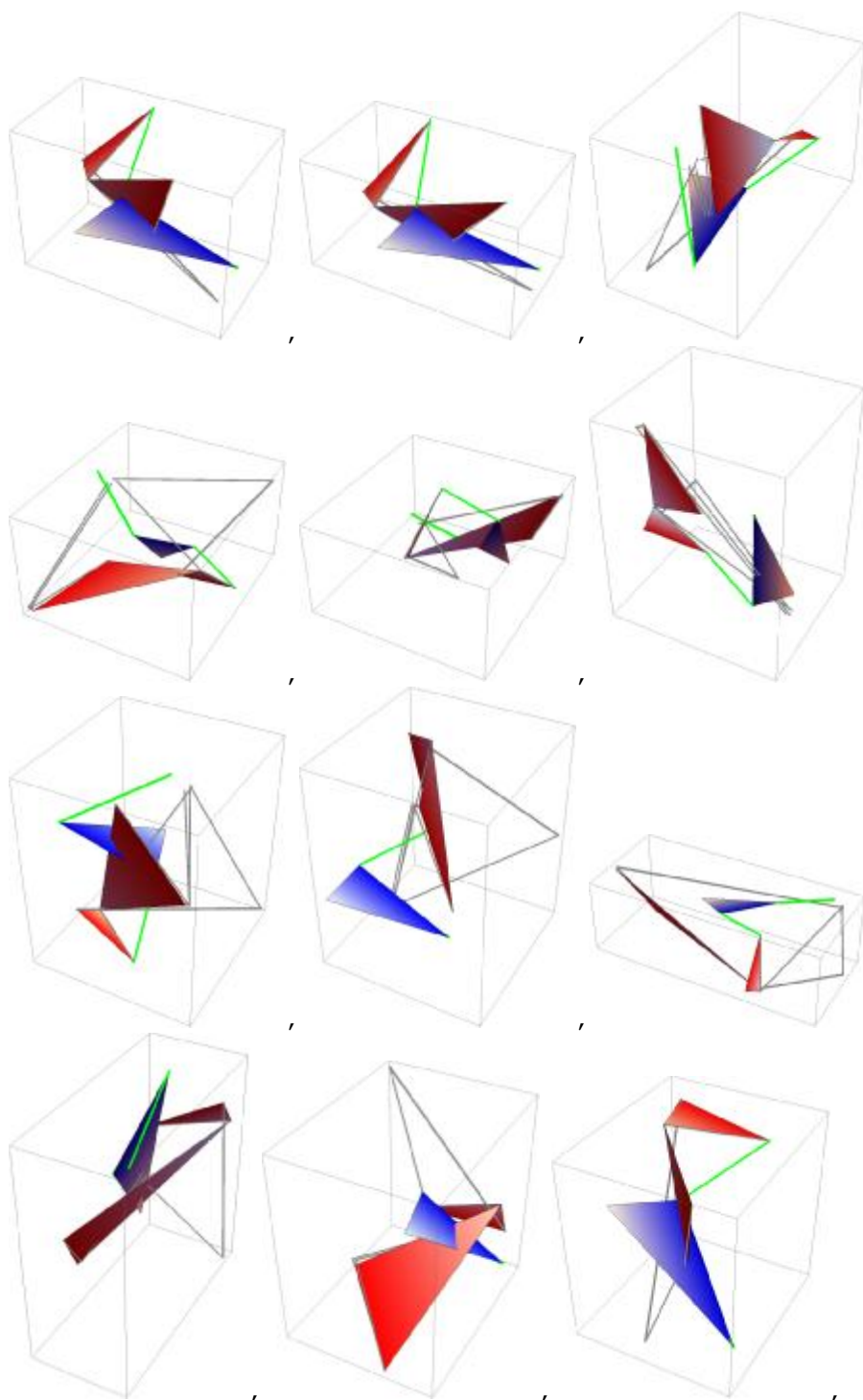
Triangle[{P2n[[i]],P11n[[i]],P13n[[i]]},VertexColors-
>VertexColor[[2]]],Thickness[0.008],

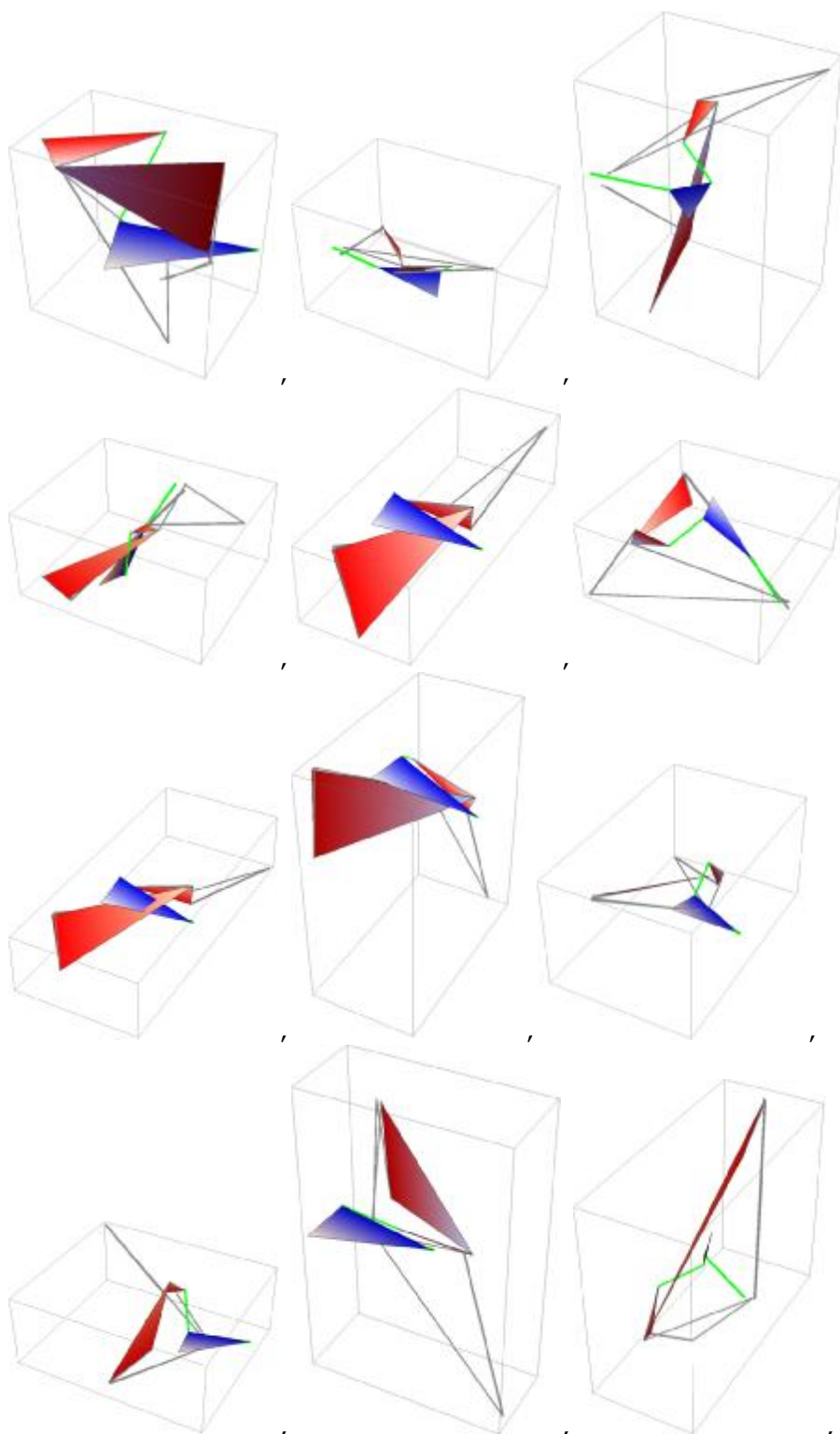
Line[{P11n[[i]],P21n[[i]]},VertexColors->VertexColor[[4]]],Thickness[0.006],

Triangle[{P31n[[i]],P21n[[i]],P32n[[i]]},VertexColors->{Red,
Red}],Thickness[0.008],
Line[{P31n[[i]],P41n[[i]]},VertexColors->VertexColor[[5]]],Thickness[0.006],
Line[{P41n[[i]],Frame1HM[[i]][{1,2,3},4]}],VertexColors-
>VertexColor[[5]]],Thickness[0.006],
Line[{P22n[[i]],P32n[[i]]},VertexColors->VertexColor[[5]]],Thickness[0.006],
Line[{P32n[[i]],Frame2HM[[i]][{1,2,3},4]}],VertexColors-
>VertexColor[[5]]],Thickness[0.006],
Triangle[{P33n[[i]],P23n[[i]],P32n[[i]]},VertexColors->{Red,
Red}],Thickness[0.008],
Line[{P33n[[i]],P43n[[i]]},VertexColors->VertexColor[[5]]],Thickness[0.006],
Line[{P43n[[i]],Frame3HM[[i]][{1,2,3},4]}],VertexColors-
>VertexColor[[5]]],Thickness[0.006],
Line[{P24n[[i]],P34n[[i]]},VertexColors->VertexColor[[5]]],Thickness[0.006],
Line[{P34n[[i]],Frame4HM[[i]][{1,2,3},4]}],VertexColors-
>VertexColor[[5]]],Thickness[0.006]]],
{i,1,40}}

```








```

PJ31=MakePoint[J31];
PJ12=MakePoint[J12];
PJ22=MakePoint[J22];
PJ32=MakePoint[J32];
PJ41=MakePoint[J41];
PJ13=MakePoint[J13];
PJ23=MakePoint[J23];
PJ33=MakePoint[J33];
PJ14=MakePoint[J14];
PJ24=MakePoint[J24];
PJ34=MakePoint[J34];
PJ43=MakePoint[J43];
P1f=C1;
P2f=C2;
P11f=C11;
P21f=C21;
P31f=C31;
P12f=C12;
P22f=C22;
P32f=C32;
P41f=C41;
P13f=C13;
P23f=C23;
P33f=C33;
P14f=C14;
P24f=C24;
P34f=C34;
P43f=C43;
the hand is :0,1,2,2,3,5,,5,7,4,9,9,11

PD0=Flatten[{Table[Frame1HM[{i,{1,2,3},4}],{i,1,40}],Table[Frame2HM[{i,{1,2,3},4}],{i,1,40}],Table[Frame3HM[{i,{1,2,3},4}],{i,1,40}],Table[Frame4HM[{i,{1,2,3},4}],{i,1,40}]],1];

PointsUnoptomized=Flatten[{{PJ1,CommonNormalGen[J1,J2][[2,1]],PJ2,CommonNormalGen[J1,J2][[2,2]],PJ11,CommonNormalGen[J2,J11][[2,2]],PJ13,CommonNormalGen[J2,J13][[2,2]],PJ21,CommonNormalGen[J11,J21][[2,2]],PJ23,CommonNormalGen[J13,J23][[2,2]],PJ31,CommonNormalGen[J21,J31][[2,2]],PJ32,CommonNormalGen[J22,J32][[2,2]],PJ33,CommonNormalGen[J23,J33][[2,2]],PJ34,CommonNormalGen[J24,J34][[2,2]],PJ41,CommonNormalGen[J31,J41][[2,2]],PJ43,CommonNormalGen[J33,J43][[2,2]],P1f,P2f,P11f,P13f,P21f,P23f,P31f,P32f,P33f,P34f,P41f,P43f},PD0},1];
Export["PointsUnoptomizedc.xls",PointsUnoptomized,"XLS"];

```

Appendix B

Mathematica[®] Codes for Orange Peeler hand


```

Qt=Import["F:\\Romansy-2016\\Orange_Peeling - reformed.csv"];
InputAddress="F:\\Romansy-2016\\out10.lua";
add="C:\\reza\\Numerical\\analysis";
add2="C:\\reza\\Numerical\\Numerical\\";
<<"Desktop/Matrix.m";
<<"Desktop/ToMatlab.m";
Qt=Import["/Volumes/FEZA/Romansy-2016/Orange_Peeling - reformed.csv"];
InputAddress="/Volumes/FEZA/Romansy-2016/out.lua";
add="/Users/reza/Desktop/Numerical/analysis";
add2="/Users/reza/Desktop/Numerical/";

```

1.Synthesis and optimization of peeling Pobot

In this file the optimization of the robotic hand for orange peeling task has been done. The hand is a 1-3,3,3 hand with the end effectors one as the knife and one as the thumb and last one as the index finger. the task if for the thumb and finger to hold the orange while knife cuts a slice of the peel and then knife is place on the cut slice so that the two thumb and finger hold the slice and remove it from the orange, and the rotate the orange to be ready for the next cut.

Task Motion capture data recover

```
positions={40,436,713,905,1052,1393,1750,2200,2452,2751,2930};
```

First we need to generate one position

```
position={40};
```

Now the positions we want

```

QtfingerM1=Table[Qt[[positions[[i]],{3,4,9}]],{i,1,Length[positions]};
QtfingerM2=Table[Qt[[positions[[i]],{6,7,8}]],{i,1,Length[positions]};
QtfingerM3=Table[Qt[[positions[[i]],{9,10,11}]],{i,1,Length[positions]};

```

```

Xfinger=Table[(QtfingerM1[[i]]-QtfingerM2[[i]])/Sqrt[(QtfingerM1[[i]]-QtfingerM2[[i]])*(QtfingerM1[[i]]-QtfingerM2[[i]])],{i,1,Length[QtfingerM1]};
Yfinger=Table[(QtfingerM3[[i]]-QtfingerM2[[i]])/Sqrt[(QtfingerM3[[i]]-QtfingerM2[[i]])*(QtfingerM3[[i]]-QtfingerM2[[i]])],{i,1,Length[QtfingerM1]};
Zfinger=Table[Xfinger[[i]]@Yfinger[[i]],{i,1,Length[QtfingerM1]};
Ofinger=Table[Orthogonalize[{Xfinger[[i]],Yfinger[[i]],Zfinger[[i]]}],{i,1,Length[Xfinger]};
Dfinger=Table[Append[Transpose[{Ofinger[[i,1]],Ofinger[[i,2]],Ofinger[[i,3]],QtfingerM2[[i]]},{0,0,0,1}],{i,1,Length[Xfinger]};
dqEfinger=Table[HM2dq[Dfinger[[i]]],{i,1,Length[Dfinger]};
plotfinger=generatePlot[dqEfinger,Green,3];
QtthumbM1=Table[Qt[[positions[[i]],{12,13,14}]],{i,1,Length[positions]};
QtthumbM2=Table[Qt[[positions[[i]],{15,16,17}]],{i,1,Length[positions]};
QtthumbM3=Table[Qt[[positions[[i]],{18,19,20}]],{i,1,Length[positions]};
Length[QtthumbM1];
Xthumb=Table[(QtthumbM1[[i]]-QtthumbM2[[i]])/Sqrt[(QtthumbM1[[i]]-QtthumbM2[[i]])*(QtthumbM1[[i]]-QtthumbM2[[i]])],{i,1,Length[QtthumbM1]};
Ythumb=Table[(QtthumbM3[[i]]-QtthumbM2[[i]])/Sqrt[(QtthumbM3[[i]]-QtthumbM2[[i]])*(QtthumbM3[[i]]-QtthumbM2[[i]])],{i,1,Length[QtthumbM1]};
Zthumb=Table[Xthumb[[i]]@Ythumb[[i]],{i,1,Length[QtthumbM1]};
Length[Xthumb];Othumb=Table[Orthogonalize[{Xthumb[[i]],Ythumb[[i]],Zthumb[[i]]}],{i,1,Length[Xthumb]};
Dthumb=Table[Append[Transpose[{Othumb[[i,1]],Othumb[[i,2]],Othumb[[i,3]],QtthumbM2[[i]]},{0,0,0,1}],{i,1,Length[Xthumb]};
dqEthumb=Table[HM2dq[Dthumb[[i]]],{i,1,Length[Dthumb]};
plotthumb=generatePlot[dqEthumb,Blue,3];

```

```

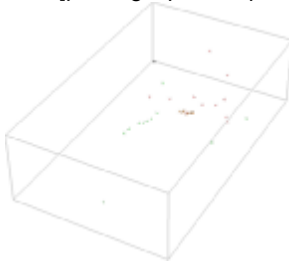
{{145.235,203.38,72.8955},{145.235,203.38,72.8955},{145.235,203.38,72.8955},{145.235,203.38,72.8955},{145.235,203.38,72.8955},{145.235,203.38,72.8955},{146.09,199.423,73.7989},{146.09,199.423,73.7989},{146.09,199.423,73.7989},{225.429,227.982,74.9037},{223.211,143.29,89.65}}
QtknifM1=Table[Qt[[positions[[i]],{21,22,23}]],{i,1,Length[positions]};
QtknifM2=Table[Qt[[positions[[i]],{24,25,26}]],{i,1,Length[positions]};
QtknifM3=Table[Qt[[positions[[i]],{27,28,29}]],{i,1,Length[positions]};

```

```

Xknife=Table[(QtknifeM1[[i]]-QtknifeM2[[i]])/Sqrt[(QtknifeM1[[i]]-QtknifeM2[[i]])*(QtknifeM1[[i]]-QtknifeM2[[i]])],{i,1,Length[QtknifeM1]};
Yknife=Table[(QtknifeM3[[i]]-QtknifeM2[[i]])/Sqrt[(QtknifeM3[[i]]-QtknifeM2[[i]])*(QtknifeM3[[i]]-QtknifeM2[[i]])],{i,1,Length[QtknifeM1]};
Zknife=Table[Xknife[[i]]*Yknife[[i]],{i,1,Length[QtknifeM1]};
Length[Xknife];
Oknif=Table[Orthogonalize[{Xknife[[i]],Yknife[[i]],Zknife[[i]]}],{i,1,Length[Xknife]};
Dknife=Table[Append[Transpose[{Oknif[[i,1]],Oknif[[i,2]],Oknif[[i,3]],QtknifeM2[[i]]}],{0,0,0,1}],{i,1,Length[Xknife]};
dqEknife=Table[HM2dq[Dknife[[i]]],{i,1,Length[Dknife]};
plotknife=generatePlot[dqEknife,Red,3];
QttableM1=Table[Qt[[positions[[i]],{30,31,32}]],{i,1,Length[positions]};
QttableM2=Table[Qt[[positions[[i]],{33,34,35}]],{i,1,Length[positions]};
QttableM3=Table[Qt[[positions[[i]],{36,37,38}]],{i,1,Length[positions]};
Xtable=Table[(QttableM1[[i]]-QttableM2[[i]])/Sqrt[(QttableM1[[i]]-QttableM2[[i]])*(QttableM1[[i]]-QttableM2[[i]])],{i,1,Length[QttableM1]};
Ytable=Table[(QttableM3[[i]]-QttableM2[[i]])/Sqrt[(QttableM3[[i]]-QttableM2[[i]])*(QttableM3[[i]]-QttableM2[[i]])],{i,1,Length[QttableM1]};
Ztable=Table[Xtable[[i]]*Ytable[[i]],{i,1,Length[QttableM1]};
Length[Xtable];
Otable=Table[Orthogonalize[{Xtable[[i]],Ytable[[i]],Ztable[[i]]}],{i,1,Length[Xtable]};
Dtable=Table[Append[Transpose[{Otable[[i,1]],Otable[[i,2]],Otable[[i,3]],QttableM2[[i]]}],{0,0,0,1}],{i,1,Length[Xtable]};
dqEtable=Table[HM2dq[Dtable[[i]]],{i,1,Length[Dtable]};
plottable=generatePlot[dqEtable,Black,3];
QtorangeM1=Table[Qt[[positions[[i]],{39,40,41}]],{i,1,Length[positions]};
QtorangeM2=Table[Qt[[positions[[i]],{42,43,44}]],{i,1,Length[positions]};
QtorangeM3=Table[Qt[[positions[[i]],{45,46,47}]],{i,1,Length[positions]};
Xorange=Table[(QtorangeM1[[i]]-QtorangeM2[[i]])/Sqrt[(QtorangeM1[[i]]-QtorangeM2[[i]])*(QtorangeM1[[i]]-QtorangeM2[[i]])],{i,1,Length[QtorangeM1]};
Yorange=Table[(QtorangeM3[[i]]-QtorangeM2[[i]])/Sqrt[(QtorangeM3[[i]]-QtorangeM2[[i]])*(QtorangeM3[[i]]-QtorangeM2[[i]])],{i,1,Length[QtorangeM1]};
Zorange=Table[Xorange[[i]]*Yorange[[i]],{i,1,Length[QtorangeM1]};
Length[Xorange];
Oorange=Table[Orthogonalize[{Xorange[[i]],Yorange[[i]],Zorange[[i]]}],{i,1,Length[Xorange]};
Dorange=Table[Append[Transpose[{Oorange[[i,1]],Oorange[[i,2]],Oorange[[i,3]],QtorangeM2[[i]]}],{0,0,0,1}],{i,1,Length[Xorange]};
dqEorange=Table[HM2dq[Dorange[[i]]],{i,1,Length[Dorange]};
plotorange=generatePlot[dqEorange,Orange,3];
Show[plotfinger,plotknife,plottable,plotorange]

```



Data Entry to the Artreeks

Now we need the poses of the finger, thumb and knife to be inserted to the Artreeks

```

InputKnife=Table[Flatten[dqEknife[[i]]],{i,1,Length[Dknife]};
ArtreeksInputKnife=Table[Flatten[{InputKnife[[i,4]],InputKnife[[i,{1,2,3,5,6,7,8}]]}],{i,1,Length[Dknife]}
{{0.641038,0.727999,0.225857,-0.0898652,-4.38368,151.457,-105.31,-63.1475},{0.667658,0.606563,-0.114755,0.416107,76.6814,86.1141,-52.2766,-22.283},{0.731116,0.652492,0.0763743,0.184093,57.7263,106.7,-47.2771,-50.7604},{0.758121,0.620679,0.157419,0.123403,70.7898,102.605,-52.7047,-70.6825},{0.42517,0.692357,-0.425693,0.398319,71.4553,63.9876,-64.2631,7.91158},{0.359202,0.899499,0.0159434,0.248237,44.9243,60.4482,-73.2601,-64.5521},{0.398967,0.803618,0.417001,0.145375,39.3495,61.2091,-67.8812,-118.501},{0.154475,-0.155718,-0.0748718,-0.972771,-110.047,80.9222,20.1656,55.2769},{0.140498,-0.100778,-0.0362321,-0.984272,-

```

```

115.337,87.1154,16.1873,53.1362},{0.125938,-0.143233,-0.0213434,-0.981412,-
109.708,69.5863,20.7387,48.6323},{0.0493915,-0.0912675,-0.0531203,-0.993181,-
146.604,68.3275,12.7856,59.6822}}
InputThumb=Table[Flatten[dqBThumb[[i]]],{i,1,Length[DThumb]]];
ArtreeksInputThumb=Table[Flatten[{InputThumb[[i,4]],InputThumb[[i,{1,2,3,5,6,7,8}]]}],{i,1,Length[DThumb]]}
{{0.960141,0.0596016,-0.080778,-0.260868,78.3309,55.328,23.7796,6.25322},{0.485865,0.713474,-0.504106,-
0.0276848,43.9748,67.3578,-113.263,-1.14248},{0.364834,0.775746,-0.514887,-0.00237181,39.8467,61.0076,-
119.957,0.593629},{0.363792,0.769882,-0.522992,0.0376319,44.4489,57.3716,-
119.959,0.821112},{0.397114,0.727299,-0.555824,-0.0663016,34.4128,65.9474,-
121.974,8.91359},{0.401881,0.683768,-0.558436,-0.243109,16.7491,81.8918,-
110.699,18.3314},{0.443203,0.634554,-0.580235,-0.253454,21.1366,83.5895,-102.431,20.5948},{0.699392,-
0.235202,-0.071552,-0.671127,-37.3489,90.3242,51.2737,45.8819},{0.629625,-0.507514,0.14542,-0.56996,-
36.0767,58.3731,103.913,51.5039},{0.318928,0.821316,-0.146849,-0.449621,-13.5934,122.01,-100.699,-
50.7793},{0.536636,0.637776,-0.446267,-0.325746,46.946,101.064,-83.5148,-22.444}}
InputFinger=Table[Flatten[dqBFinger[[i]]],{i,1,Length[DFinger]]];
ArtreeksInputFinger=Table[Flatten[{InputFinger[[i,4]],InputFinger[[i,{1,2,3,5,6,7,8}]]}],{i,1,Length[DFinger]]}
{{0.656361,-0.467488,-0.375172,-0.45814,40.5228,9.65105,2.96986,36.4515},{0.77611,0.102886,-0.468747,-
0.409076,1.01036,95.228,-0.133068,57.3107},{0.708241,0.318153,-0.52223,-0.352774,-0.209565,86.8056,-
26.6021,50.8507},{0.682488,0.419826,-0.493608,-0.338093,-1.72169,84.691,-35.4665,44.7421},{0.667134,0.643328,-
0.323645,-0.190566,8.83015,83.2242,-47.2117,18.3733},{0.628565,0.613384,-0.301696,-0.371005,-
10.2939,76.0052,-40.9972,22.3277},{0.577834,0.673802,-0.284166,-0.36242,-13.3335,68.7373,-
47.1604,19.7723},{0.157441,0.578703,-0.586417,-0.544454,-38.0563,45.4667,-72.5337,58.3993},{0.159243,-
0.637017,0.380627,0.651133,50.4581,-46.3077,89.768,-54.5219},{0.449649,-0.543626,-0.375129,-0.601303,-
19.6898,93.5254,47.9593,118.355},{0.171303,-0.695728,-0.0364035,-0.69663,-39.2986,46.348,59.5152,92.2698}}

```

2.Forward Kinematic

Data Import-new

After the synthesis has been done the results that consist of the joint axes, and rotation angles as well as the calculated positions are imported back here for the calculation and optimization as follows:

The axes are imported with automatic importing commands as follows, this functions read all axes at a time, it only requires the file path and the number of axes:

```

ReadAxes=ReadAllAxes[InputAddress,10];
jointsAll=Table[ReadAxes[[i]],{i,1,10}];
Tests=Table[{jointsAll[[i,1]],jointsAll[[i,2]],jointsAll[[i,1]],jointsAll[[i,1]]},{i,1,10}];
ReadFrames=ReadAllPoses[InputAddress,1,10];
FrameAll=ReadFrames[1];

ReadAngles=ReadAllAngles[InputAddress,10];
jointsAngles=Table[ReadAngles[[i]],{i,1,10}];
θ=Table[jointsAngles[[i,1]],{i,1,10}];
θs=Table[θ[[i,j]],{i,1,10},{j,1,9}];
thetasM=θs

```

FW-test

```

S0qOn=Table[makequat[J1,thetasM[[1,i]],0],{i,1,9}];
StqOn=Table[makequat[J11,thetasM[[2,i]],0],{i,1,9}];
S2qOn=Table[makequat[J21,thetasM[[3,i]],0],{i,1,9}];
S3qOn=Table[makequat[J31,thetasM[[4,i]],0],{i,1,9}];
S4qOn=Table[makequat[J12,thetasM[[5,i]],0],{i,1,9}];
S5qOn=Table[makequat[J22,thetasM[[6,i]],0],{i,1,9}];
S6qOn=Table[makequat[J32,thetasM[[7,i]],0],{i,1,9}];
S7qOn=Table[makequat[J13,thetasM[[8,i]],0],{i,1,9}];
S8qOn=Table[makequat[J23,thetasM[[9,i]],0],{i,1,9}];
S9qOn=Table[makequat[J33,thetasM[[10,i]],0],{i,1,9}];
FK1On=Table[quatmultSep[quatmultSep[quatmultSep[S0qOn[[i]],StqOn[[i]],S2qOn[[i]],S3qOn[[i]],{i,1,9}];
FK2On=Table[quatmultSep[quatmultSep[quatmultSep[S0qOn[[i]],S4qOn[[i]],S5qOn[[i]],S6qOn[[i]],{i,1,9}];
FK3On=Table[quatmultSep[quatmultSep[quatmultSep[S0qOn[[i]],S7qOn[[i]],S8qOn[[i]],S9qOn[[i]],{i,1,9}];
Table[Abs[FK1On[[i]]-quatmultSep[dqBknif2[[i+1]],quatconj[dqBknif2[[1]]]],{i,1,9}]]

```

```
Table[AbsFK2On[[i]]-quatmultSep[dqEhumb2[[i+1]],quatconj[dqEhumb2[[1]]]],{i,1,9}]
Table[AbsFK3On[[i]]-quatmultSep[dqEfinger2[[i+1]],quatconj[dqEfinger2[[1]]]],{i,1,9}]
```

```
{{{0.0749289,0.0194394,0.0363485,0.00921011},{0.05194,0.00535744,0.14417,0.00408047}},{{0.0211426,0.0104919,0.0016429,0.00764598},{0.0819096,0.0733259,0.609965,0.0134379}},{{0.00362155,0.00893511,0.00179137,0.000741714},{0.229994,0.171121,0.679742,0.0317093}},{{0.0000779694,0.0173766,0.000186569,0.00559064},{0.377475,0.097225,0.460393,0.0378283}},{{0.00193617,0.0110613,0.00230983,0.0077885},{0.609951,0.138116,0.304056,0.0769336}},{{0.0154228,0.0318528,0.0197068,0.000940747},{0.893154,0.252592,0.325066,0.00673803}},{{0.0228043,0.0510254,0.0278836,0.0119196},{0.905056,0.269964,0.354113,0.0278265}},{{0.0181535,0.0389498,0.0241038,0.00841564},{0.961705,0.256791,0.274205,0.0281051}},{{0.00997985,0.0206799,0.0127939,0.00432937},{2.05602,0.524569,0.745258,0.221906}}}
```

```
{{{0.0177337,0.033844,0.00739148,0.0058197},{1.4949,0.909843,0.847112,0.0556588}},{{0.122947,0.223456,0.0747284,1.98178},{19.273,0.385238,57.3374,0.929114}},{{0.0560093,0.0159328,0.0451882,0.0107649},{0.275807,0.00700261,3.20228,0.184811}},{{0.30108,0.172055,0.332206,1.94146},{90.9631,69.2763,21.0341,4.37343}},{{0.0106535,0.0110767,0.167392,0.0514881},{1.21589,0.582322,1.06871,0.22212}},{{0.517521,1.63688,0.902705,0.482005},{316.786,79.5616,335.692,17.8098}},{{0.0222082,0.0273295,0.00840935,0.0979073},{3.5211,0.925189,8.08527,0.719161}},{{0.0187273,0.0035557,0.0330795,0.00751656},{3.16373,0.22202,6.55978,1.67427}},{{0.178894,0.537492,0.283866,1.89685},{26.9256,6.711,100.292,15.6429}}}
```

```
{{{0.0786459,0.0873789,0.0622134,0.0229717},{14.155,2.52964,22.4837,0.930375}},{{0.130192,0.120698,0.0670077,0.0462707},{13.1779,5.54274,30.6135,1.51353}},{{0.122705,0.0130308,0.0379858,0.0667476},{7.77161,6.29006,23.8181,1.44228}},{{0.599246,0.379696,0.438661,1.78437},{79.8396,16.5722,149.727,6.5353}},{{0.316178,0.0828649,0.0357724,0.100329},{9.32076,20.8189,50.6473,5.66183}},{{0.547298,1.14325,0.179463,1.4864},{16.7792,46.7291,187.034,5.73666}},{{0.556654,0.117836,0.193335,0.494548},{25.0031,62.6741,133.805,2.03663}},{{0.269951,0.239094,0.409273,0.268006},{128.719,30.2501,89.332,10.2921}},{{0.19864,0.34334,0.840935,1.35935},{105.725,60.5233,87.9926,23.8113}}}
```

Error Checking

Now for future better addressing the positions of each end effectors, all calculated positions (that are the output of the synthesis) for each end effector has been put together with that end effector name also they have been put all together as the PD set as follows:

```
npos=10;
FrameKnifeHM=Table[FrameAll[[i]],{i,1,npos}];
FrameThumbHM=Table[FrameAll[[i]],{i,npos+1,2*npos}];
FrameFingerHM=Table[FrameAll[[i]],{i,2*npos+1,3*npos}];
```

```
PD={Table[FrameKnifeHM[[i]],{i,1,9}],Table[FrameThumbHM[[i]],{i,1,9}],Table[FrameFingerHM[[i]],{i,1,9}];
ins={2,3,4,5,6,7,8,9,10,11}
{2,3,4,5,6,7,8,9,10,11}
dqEknif2=dqEknif[[ins]];
dqEhumb2=dqEhumb[[ins]];
dqEfinger2=dqEfinger[[ins]];
```

```
InputKnifHM=Table[dq2HM[dqEknif2[[i]]],{i,1,Length[ins]}];
InputThumbHM=Table[dq2HM[dqEhumb2[[i]]],{i,1,Length[ins]}];
InputFingerHM=Table[dq2HM[dqEfinger2[[i]]],{i,1,Length[ins]}];
```

Here the synthesis error has been calculated by finding the difference between the calculated positions and the input data, and as it is clear, the differences are very small and so the results are reliable:

```
SynthesisErrorKnife=Max[Table[FrameKnifeHM[[i]]-InputKnifHM[[i]],{i,1,9}]]
SynthesisErrorThumb=Max[Table[FrameThumbHM[[i]]-InputThumbHM[[i]],{i,1,9}]]
SynthesisErrorFinger=Max[Table[FrameFingerHM[[i]]-InputFingerHM[[i]],{i,1,9}]]
0.000608994
0.000498093
1.13291
```

Here the only the positions of each end effectors are calculated by taking [[{1,2,3},4]] part of each Frame and put in different with new names for easier addressing:

```
CEFKnife=Table[FrameKnifeHM[[i,{1,2,3},4]],{i,1,Length[FrameKnifeHM]}];
```

```

CEFFThumb=Table[FrameThumbHM[[i,{1,2,3,4}],{i,1,Length[FrameKnifeHM]}];
CEFFinger=Table[FrameFingerHM[[i,{1,2,3,4}],{i,1,Length[FrameKnifeHM]}];
Now the positions has been transformed to dual quaternions:
FrameKnifedq=Table[HM2dq[FrameKnifeHM[[i]]],{i,1,Length[FrameKnifeHM]}];
FrameThumbdq=Table[HM2dq[FrameThumbHM[[i]]],{i,1,Length[FrameThumbHM]}];
FrameFingerdq=Table[HM2dq[FrameFingerHM[[i]]],{i,1,Length[FrameFingerHM]}];

```

Forward Kinematics and Anchoring Points Calculation

Here joints have been named as J_i for the wrist joints, joint number i of wrist and J_{ij} as the joint number i of the branch j :

```

j1={jointsAll[[1,1]],jointsAll[[1,2]]};
j11={jointsAll[[2,1]],jointsAll[[2,2]]};
j21={jointsAll[[3,1]],jointsAll[[3,2]]};
j31={jointsAll[[4,1]],jointsAll[[4,2]]};
j12={jointsAll[[5,1]],jointsAll[[5,2]]};
j22={jointsAll[[6,1]],jointsAll[[6,2]]};
j32={jointsAll[[7,1]],jointsAll[[7,2]]};
j13={jointsAll[[8,1]],jointsAll[[8,2]]};
j23={jointsAll[[9,1]],jointsAll[[9,2]]};
j33={jointsAll[[10,1]],jointsAll[[10,2]]};
c1=(j1[[1]]⊗j1[[2]])/(j1[[1]].j1[[1]]);
c11=(j11[[1]]⊗j11[[2]])/(j11[[1]].j11[[1]]);
c21=(j21[[1]]⊗j21[[2]])/(j21[[1]].j21[[1]]);
c31=(j31[[1]]⊗j31[[2]])/(j31[[1]].j31[[1]]);
c12=(j12[[1]]⊗j12[[2]])/(j12[[1]].j12[[1]]);
c22=(j22[[1]]⊗j22[[2]])/(j22[[1]].j22[[1]]);
c32=(j32[[1]]⊗j32[[2]])/(j32[[1]].j32[[1]]);
c13=(j13[[1]]⊗j13[[2]])/(j13[[1]].j13[[1]]);
c23=(j23[[1]]⊗j23[[2]])/(j23[[1]].j23[[1]]);
c33=(j33[[1]]⊗j33[[2]])/(j33[[1]].j33[[1]]);
CEF1=CEFFKnife[[1]];
CEF2=CEFFThumb[[1]];
CEF3=CEFFinger[[1]];
{69.7586298290000002,237.108801496799998,70.8048459469999756}

```

Now we normalize the axes

```

J1={j1[[1]]/Norm[j1[[1]]],c1⊗(j1[[1]]/Norm[j1[[1]]])};
J11={j11[[1]]/Norm[j11[[1]]],c11⊗(j11[[1]]/Norm[j11[[1]]])};
J21={j21[[1]]/Norm[j21[[1]]],c21⊗(j21[[1]]/Norm[j21[[1]]])};
J31={j31[[1]]/Norm[j31[[1]]],c31⊗(j31[[1]]/Norm[j31[[1]]])};
J12={j12[[1]]/Norm[j12[[1]]],c12⊗(j12[[1]]/Norm[j12[[1]]])};
J22={j22[[1]]/Norm[j22[[1]]],c22⊗(j22[[1]]/Norm[j22[[1]]])};
J32={j32[[1]]/Norm[j32[[1]]],c32⊗(j32[[1]]/Norm[j32[[1]]])};
J13={j13[[1]]/Norm[j13[[1]]],c13⊗(j13[[1]]/Norm[j13[[1]]])};
J23={j23[[1]]/Norm[j23[[1]]],c23⊗(j23[[1]]/Norm[j23[[1]]])};
J33={j33[[1]]/Norm[j33[[1]]],c33⊗(j33[[1]]/Norm[j33[[1]]])};

```

Also angles are re mentioned to fit the formulas:

```

θs=Table[θ[[i,j]],{i,1,10},{j,1,9}];

```

Here the quaternions of each axis has been made having that joint and its rotation angles:

```

S1q=Table[makequat[J1,θs[[1,i]],0],{i,1,9}];
S11q=Table[makequat[J11,θs[[2,i]],0],{i,1,9}];
S21q=Table[makequat[J21,θs[[3,i]],0],{i,1,9}];
S31q=Table[makequat[J31,θs[[4,i]],0],{i,1,9}];
S12q=Table[makequat[J12,θs[[5,i]],0],{i,1,9}];
S22q=Table[makequat[J22,θs[[6,i]],0],{i,1,9}];
S32q=Table[makequat[J32,θs[[7,i]],0],{i,1,9}];
S13q=Table[makequat[J13,θs[[8,i]],0],{i,1,9}];
S23q=Table[makequat[J23,θs[[9,i]],0],{i,1,9}];
S33q=Table[makequat[J33,θs[[10,i]],0],{i,1,9}];

```

At this step the Forward Kinematic of each branch from the wrist joint up to the last joint has been calculated:

$FK1 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S1q[[i]], S21q[[i]], S31q[[i]], \{i, 1, 9\}];$
 $FK2 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S12q[[i]], S22q[[i]], S32q[[i]], \{i, 1, 9\}];$
 $FK3 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S13q[[i]], S23q[[i]], S33q[[i]], \{i, 1, 9\}];$
 In the following calculations the forward kinematic are used to check the joints and rotation angles by finding the difference between them and the relative displacements and the results showed good percision for all of them:

$\text{Max}[\text{Table}[\text{dq2HM}[FK1[[i]] - \text{FrameKnifeHM}[[i+1]].\text{Inverse}[\text{FrameKnifeHM}[[1]], \{i, 1, 9\}]]$
 $\text{Max}[\text{Table}[\text{dq2HM}[FK2[[i]] - \text{FrameThumbHM}[[i+1]].\text{Inverse}[\text{FrameThumbHM}[[1]], \{i, 1, 9\}]]$
 $\text{Max}[\text{Table}[\text{dq2HM}[FK3[[i]] - \text{FrameFingerHM}[[i+1]].\text{Inverse}[\text{FrameFingerHM}[[1]], \{i, 1, 9\}]]$

1139.2181421496458

358.2934513167650

395.0954473592575

At this step the forward kinematic for each of the joints has been calculated as follows:

$FKrel11 = \text{Table}[S1q[[i]], \{i, 1, 9\}];$
 $FKrel21 = \text{Table}[\text{quatmultSep}[S1q[[i]], S11q[[i]], \{i, 1, 9\}];$
 $FKrel31 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S11q[[i]], S21q[[i]], \{i, 1, 9\}];$
 $FKrel41 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S11q[[i]], S21q[[i]], S31q[[i]], \{i, 1, 9\}];$
 $FKrel12 = \text{Table}[S1q[[i]], \{i, 1, 9\}];$
 $FKrel22 = \text{Table}[\text{quatmultSep}[S1q[[i]], S12q[[i]], \{i, 1, 9\}];$
 $FKrel32 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S12q[[i]], S22q[[i]], \{i, 1, 9\}];$
 $FKrel42 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S12q[[i]], S22q[[i]], S32q[[i]], \{i, 1, 9\}];$
 $FKrel13 = \text{Table}[S1q[[i]], \{i, 1, 9\}];$
 $FKrel23 = \text{Table}[\text{quatmultSep}[S1q[[i]], S13q[[i]], \{i, 1, 9\}];$
 $FKrel33 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S13q[[i]], S23q[[i]], \{i, 1, 9\}];$
 $FKrel43 = \text{Table}[\text{quatmultSep}[\text{quatmultSep}[\text{quatmultSep}[S1q[[i]], S13q[[i]], S23q[[i]], S33q[[i]], \{i, 1, 9\}];$

Inorder to find the anchoring point along each axes a reference point is needed for each axes, this point on each axes has been calculated as follows as the intersection point of the normal line from the origin to that line as C_i being the i th axes of the wrist, and C_{ij} being the point on the i th axes of the j th branch. Also the end effector first point has been mentioned for calculations. Similar naming method has been used for the point from now at each calculation step:

$C1 = (J1[[1]] \otimes J1[[2]]) / (J1[[1]] \cdot J1[[1]]);$
 $C11 = (J11[[1]] \otimes J11[[2]]) / (J11[[1]] \cdot J11[[1]]);$
 $C21 = (J21[[1]] \otimes J21[[2]]) / (J21[[1]] \cdot J21[[1]]);$
 $C31 = (J31[[1]] \otimes J31[[2]]) / (J31[[1]] \cdot J31[[1]]);$
 $C12 = (J12[[1]] \otimes J12[[2]]) / (J12[[1]] \cdot J12[[1]]);$
 $C22 = (J22[[1]] \otimes J22[[2]]) / (J22[[1]] \cdot J22[[1]]);$
 $C32 = (J32[[1]] \otimes J32[[2]]) / (J32[[1]] \cdot J32[[1]]);$
 $C13 = (J13[[1]] \otimes J13[[2]]) / (J13[[1]] \cdot J13[[1]]);$
 $C23 = (J23[[1]] \otimes J23[[2]]) / (J23[[1]] \cdot J23[[1]]);$
 $C33 = (J33[[1]] \otimes J33[[2]]) / (J33[[1]] \cdot J33[[1]]);$
 $CEF1 = \text{CEFFKnife}[[1]];$
 $CEF2 = \text{CEFFThumb}[[1]];$
 $CEF3 = \text{CEFFinger}[[1]];$

Now the pure quaternoin of the aforesaid points has been calculated as follows:

$C1q = \{(0, 0, 0, 1), \text{Append}[C1, 0]\};$
 $C11q = \{(0, 0, 0, 1), \text{Append}[C11, 0]\};$
 $C21q = \{(0, 0, 0, 1), \text{Append}[C21, 0]\};$
 $C31q = \{(0, 0, 0, 1), \text{Append}[C31, 0]\};$
 $CEF1q = \{(0, 0, 0, 1), \text{Append}[CEF1, 0]\};$
 $C12q = \{(0, 0, 0, 1), \text{Append}[C12, 0]\};$
 $C22q = \{(0, 0, 0, 1), \text{Append}[C22, 0]\};$
 $C32q = \{(0, 0, 0, 1), \text{Append}[C32, 0]\};$
 $CEF2q = \{(0, 0, 0, 1), \text{Append}[CEF2, 0]\};$
 $C13q = \{(0, 0, 0, 1), \text{Append}[C13, 0]\};$
 $C23q = \{(0, 0, 0, 1), \text{Append}[C23, 0]\};$
 $C33q = \{(0, 0, 0, 1), \text{Append}[C33, 0]\};$
 $CEF3q = \{(0, 0, 0, 1), \text{Append}[CEF3, 0]\};$

And at this step the forward kinematics of each axes are used to calculate the the moved dual quartenion of points on each axis:

```
C11mq=Table[quatmultSep[quatmultSep[FKrel11[[i]],C11q],quatconj4[FKrel11[[i]]],{i,1,9}];
C21mq=Table[quatmultSep[quatmultSep[FKrel21[[i]],C21q],quatconj4[FKrel21[[i]]],{i,1,9}];
C31mq=Table[quatmultSep[quatmultSep[FKrel31[[i]],C31q],quatconj4[FKrel31[[i]]],{i,1,9}];
CEF1mq=Table[quatmultSep[quatmultSep[FKrel41[[i]],CEF1q],quatconj4[FKrel41[[i]]],{i,1,9}];
C12mq=Table[quatmultSep[quatmultSep[FKrel12[[i]],C12q],quatconj4[FKrel12[[i]]],{i,1,9}];
C22mq=Table[quatmultSep[quatmultSep[FKrel22[[i]],C22q],quatconj4[FKrel22[[i]]],{i,1,9}];
C32mq=Table[quatmultSep[quatmultSep[FKrel32[[i]],C32q],quatconj4[FKrel32[[i]]],{i,1,9}];
CEF2mq=Table[quatmultSep[quatmultSep[FKrel42[[i]],CEF2q],quatconj4[FKrel42[[i]]],{i,1,9}];
C13mq=Table[quatmultSep[quatmultSep[FKrel13[[i]],C13q],quatconj4[FKrel13[[i]]],{i,1,9}];
C23mq=Table[quatmultSep[quatmultSep[FKrel23[[i]],C23q],quatconj4[FKrel23[[i]]],{i,1,9}];
C33mq=Table[quatmultSep[quatmultSep[FKrel33[[i]],C33q],quatconj4[FKrel33[[i]]],{i,1,9}];
CEF3mq=Table[quatmultSep[quatmultSep[FKrel43[[i]],CEF3q],quatconj4[FKrel43[[i]]],{i,1,9}];
```

Here the position part for each of the calculated points $\{[2,\{1,2,3\}]\}$ part of each quaternion) have been gathered and added with the initial point of that set:

```
On11m=Prepend[Table[C11mq[[i,2,{1,2,3}]],{i,1,9},C11];
On21m=Prepend[Table[C21mq[[i,2,{1,2,3}]],{i,1,9},C21];
On31m=Prepend[Table[C31mq[[i,2,{1,2,3}]],{i,1,9},C31];
OnEF1m=Prepend[Table[CEF1mq[[i,2,{1,2,3}]],{i,1,9},CEF1];
On12m=Prepend[Table[C12mq[[i,2,{1,2,3}]],{i,1,9},C12];
On22m=Prepend[Table[C22mq[[i,2,{1,2,3}]],{i,1,9},C22];
On32m=Prepend[Table[C32mq[[i,2,{1,2,3}]],{i,1,9},C32];
OnEF2m=Prepend[Table[CEF2mq[[i,2,{1,2,3}]],{i,1,9},CEF2];
On13m=Prepend[Table[C13mq[[i,2,{1,2,3}]],{i,1,9},C13];
On23m=Prepend[Table[C23mq[[i,2,{1,2,3}]],{i,1,9},C23];
On33m=Prepend[Table[C33mq[[i,2,{1,2,3}]],{i,1,9},C33];
OnEF3m=Prepend[Table[CEF3mq[[i,2,{1,2,3}]],{i,1,9},CEF3];
```

Also to find the transformed axes their pure quartenions have been calculated as follows with S_i being the i th joint of the wrist and S_{ij} being the i th joint of the branch j :

```
S1d={Append[J1[[1]],0],Append[J1[[2]],0]};
S11d={Append[J11[[1]],0],Append[J11[[2]],0]};
S21d={Append[J21[[1]],0],Append[J21[[2]],0]};
S31d={Append[J31[[1]],0],Append[J31[[2]],0]};
S12d={Append[J12[[1]],0],Append[J12[[2]],0]};
S22d={Append[J22[[1]],0],Append[J22[[2]],0]};
S32d={Append[J32[[1]],0],Append[J32[[2]],0]};
S13d={Append[J13[[1]],0],Append[J13[[2]],0]};
S23d={Append[J23[[1]],0],Append[J23[[2]],0]};
S33d={Append[J33[[1]],0],Append[J33[[2]],0]};
```

So the forward kinematics are used to find the transformed dual quaternion of each axes:

```
S11m=Prepend[Table[quatmultSep[quatmultSep[FKrel11[[i]],S11d],quatconj[FKrel11[[i]]][{1,2},{1,2,3}]],{i,1,9},S11d[
[{1,2},{1,2,3}]]];
S21m=Prepend[Table[quatmultSep[quatmultSep[FKrel21[[i]],S21d],quatconj[FKrel21[[i]]][{1,2},{1,2,3}]],{i,1,9},S21d[
[{1,2},{1,2,3}]]];
S31m=Prepend[Table[quatmultSep[quatmultSep[FKrel31[[i]],S31d],quatconj[FKrel31[[i]]][{1,2},{1,2,3}]],{i,1,9},S31d[
[{1,2},{1,2,3}]]];
S12m=Prepend[Table[quatmultSep[quatmultSep[FKrel12[[i]],S12d],quatconj[FKrel12[[i]]][{1,2},{1,2,3}]],{i,1,9},S12d[
[{1,2},{1,2,3}]]];
S22m=Prepend[Table[quatmultSep[quatmultSep[FKrel22[[i]],S22d],quatconj[FKrel22[[i]]][{1,2},{1,2,3}]],{i,1,9},S22d[
[{1,2},{1,2,3}]]];
S32m=Prepend[Table[quatmultSep[quatmultSep[FKrel32[[i]],S32d],quatconj[FKrel32[[i]]][{1,2},{1,2,3}]],{i,1,9},S32d[
[{1,2},{1,2,3}]]];
S13m=Prepend[Table[quatmultSep[quatmultSep[FKrel13[[i]],S13d],quatconj[FKrel13[[i]]][{1,2},{1,2,3}]],{i,1,9},S13d[
[{1,2},{1,2,3}]]];
S23m=Prepend[Table[quatmultSep[quatmultSep[FKrel23[[i]],S23d],quatconj[FKrel23[[i]]][{1,2},{1,2,3}]],{i,1,9},S23d[
[{1,2},{1,2,3}]]];
```

S33m=Prepend[Table[quatmultSep[quatmultSep[Fkrel33[[i]],S33d],quatconj[Fkrel33[[i]]][{{1,2},{1,2,3}},{i,1,9}],S33d[{{1,2},{1,2,3}}]]];

At this step transformed points and transformed axes direction have been used to write each anchoring point as $P_{ijm}=t_{ij}*s_{ijm}+C_{ij}$; with the t_{ij} , C_{nijm} , s_{ijm} being parameter, reference point and the axis direction of the i th elements of the j th branch respectively. Also the P_i being the i th point of the wrist that was not moved in this typology because there is just one joint in the wrist:

$P1=C1+t1*J1[[1]]$;
 $P11m=Table[S11m[[i,1,{1,2,3}]]*t11+Qn11m[[i]],{i,1,6}]$;
 $P21m=Table[S21m[[i,1,{1,2,3}]]*t21+Qn21m[[i]],{i,1,6}]$;
 $P31m=Table[S31m[[i,1,{1,2,3}]]*t31+Qn31m[[i]],{i,1,6}]$;
 $PEF1m=QnEF1m$;
 $P12m=Table[S12m[[i,1,{1,2,3}]]*t12+Qn12m[[i]],{i,1,6}]$;
 $P22m=Table[S22m[[i,1,{1,2,3}]]*t22+Qn22m[[i]],{i,1,6}]$;
 $P32m=Table[S32m[[i,1,{1,2,3}]]*t32+Qn32m[[i]],{i,1,6}]$;
 $PEF2m=QnEF2m$;
 $P13m=Table[S13m[[i,1,{1,2,3}]]*t13+Qn13m[[i]],{i,1,6}]$;
 $P23m=Table[S23m[[i,1,{1,2,3}]]*t23+Qn23m[[i]],{i,1,6}]$;
 $P33m=Table[S33m[[i,1,{1,2,3}]]*t33+Qn33m[[i]],{i,1,6}]$;
 $PEF3m=QnEF3m$;

At this step as a test for the kinematic of the hand and points to have been calculated correctly and every part to be addressed correctly each link or end effector lengths has been calculated to check that it remains unchanged during the rotations, for that at this step the parameters has been given the zero value in the points calculated in the previous step. As it is obvious it has been found that all links and effectors have the same lengths at each of the positions of the hand that confirms the anchoring points correct calculation also it's a second check for the kinematics and synthesis results:

$P1n=P1/t1->0$;
 $P11n=Table[P11m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P21n=Table[P21m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P31n=Table[P31m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $PEF1n=Table[PEF1m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P12n=Table[P12m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P22n=Table[P22m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P32n=Table[P32m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $PEF2n=Table[PEF2m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P13n=Table[P13m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P23n=Table[P23m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $P33n=Table[P33m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;
 $PEF3n=Table[PEF3m[[i]]/.{t1->0,t11->0,t21->0,t31->0,t12->0,t22->0,t32->0,t13->0,t23->0,t33->0},{i,1,6}]$;

$length1=Table[Sqrt[(((P1n-P11n[[i]]).(P1n-P11n[[i]]))),(i,1,6)]$
 $length2=Table[Sqrt[(((P11n[[i]]-P21n[[i]]).(P11n[[i]]-P21n[[i]]))),(i,1,6)]$
 $length3=Table[Sqrt[(((P21n[[i]]-P31n[[i]]).(P21n[[i]]-P31n[[i]]))),(i,1,6)]$
 $length4=Table[Sqrt[(((P31n[[i]]-PEF1n[[i]]).(P31n[[i]]-PEF1n[[i]]))),(i,1,6)]$
 $length5=Table[Sqrt[(((P1n-P12n[[i]]).(P1n-P12n[[i]]))),(i,1,6)]$
 $length6=Table[Sqrt[(((P12n[[i]]-P22n[[i]]).(P12n[[i]]-P22n[[i]]))),(i,1,6)]$
 $length7=Table[Sqrt[(((P22n[[i]]-P32n[[i]]).(P22n[[i]]-P32n[[i]]))),(i,1,6)]$
 $length8=Table[Sqrt[(((P32n[[i]]-PEF2n[[i]]).(P32n[[i]]-PEF2n[[i]]))),(i,1,6)]$
 $length9=Table[Sqrt[(((P1n-P13n[[i]]).(P1n-P13n[[i]]))),(i,1,6)]$
 $length10=Table[Sqrt[(((P13n[[i]]-P23n[[i]]).(P13n[[i]]-P23n[[i]]))),(i,1,6)]$
 $length11=Table[Sqrt[(((P23n[[i]]-P33n[[i]]).(P23n[[i]]-P33n[[i]]))),(i,1,6)]$
 $length12=Table[Sqrt[(((P33n[[i]]-PEF3n[[i]]).(P33n[[i]]-PEF3n[[i]]))),(i,1,6)]$

{141.649437067339056,141.649437067339056,141.649437067339056,141.64943706733906,141.64943706733906,141.64943706733906}

{341.705932096167719,341.70593209616772,341.70593209616772,341.70593209616772,341.70593209616772,341.70593209616772}

{304.66174041316263,304.66174041316263,304.6617404131626,304.6617404131626,304.6617404131626,304.6617404131626}

{146.791546412234118,146.79154641223412,146.7915464122341,146.7915464122341,146.7915464122341,146.7915464122341}

{111.664418427440537,111.664418427440537,111.664418427440537,111.66441842744054,111.66441842744054,111.66441842744054}

{267.623597616071051,267.62359761607105,267.62359761607105,267.62359761607105,267.6235976160711,267.6235976160711}

{311.390828968789114,311.39082896878911,311.39082896878911,311.39082896878911,311.39082896878911,311.3908289687891,311.3908289687891}

{293.170635648824286,293.17063564882429,293.17063564882429,293.17063564882429,293.1706356488243,293.1706356488243}

{109.525914864645088,109.525914864645088,109.525914864645088,109.52591486464509,109.52591486464509,109.52591486464509}

{100.10271791048406,100.10271791048406,100.10271791048406,100.10271791048406,100.1027179104841,100.1027179104841}

{141.034136063109876,141.03413606310988,141.03413606310988,141.03413606310988,141.0341360631099,141.0341360631099}

{189.848482603615982,189.84848260361598,189.84848260361598,189.8484826036160,189.848482603616,189.8484826036160}

Here the lengths are gathered and total length of the hand parts has been calculated to be checked after optimization:

nonconUnoptimizaed={length1,length2,length3,length4,length5,length6,length7,length8,length9,length10,length11,length12};

Sum[nonconUnoptimizaed[[i]],{i,1,12}]

The un-optimized hand at each of its position has been plotted for overall inspections. Palm links, knife branch, thumb branch and finger branch as explained at the beginning of this file (branched no. 1 -3) has been drawn in green, blue, gray and magenta colors respectively:

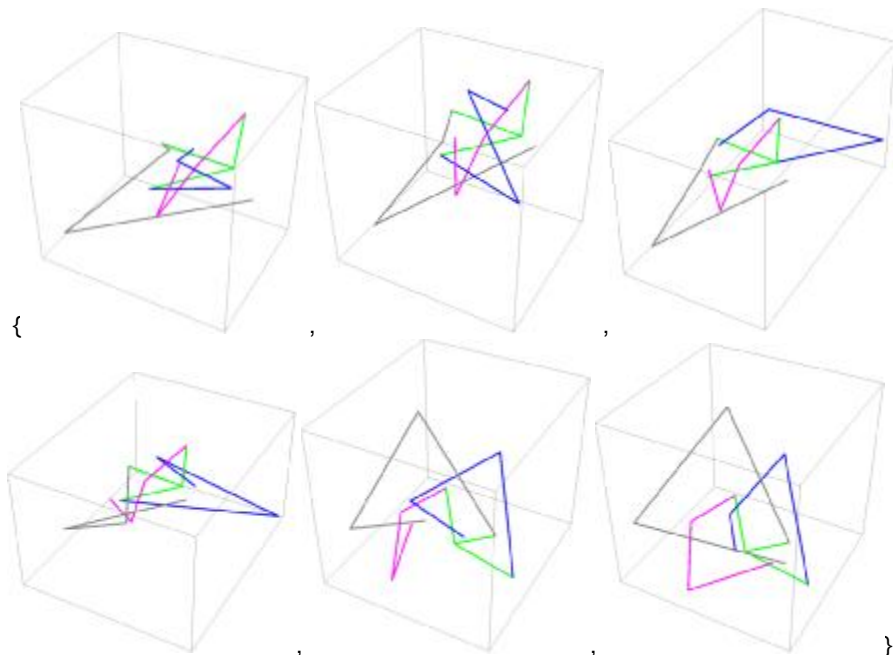
VertexColor={ {Black,Black},{Blue,Blue},{Red,Red},{Green,Green},{Blue,Blue},{Gray,Gray},{Magenta,Magenta},{Yellow,Yellow},{Magenta,Magenta},{Black,Black},{Blue,Blue},{Gray,Red}};

linOkagemO=Table[Graphics3D[Line[{P1n,P11n[[i]]},VertexColors->VertexColor[[4]],Thickness[0.006],
Line[{P1n,P12n[[i]]},VertexColors->VertexColor[[4]],Thickness[0.006],
Line[{P1n,P13n[[i]]},VertexColors->VertexColor[[4]],Thickness[0.006],
Line[{P1n,P11n[[i]]},VertexColors->VertexColor[[4]],Thickness[0.006],

Line[{P11n[[i]],P21n[[i]]},VertexColors->VertexColor[[5]],Thickness[0.006],
Line[{P21n[[i]],P31n[[i]]},VertexColors->VertexColor[[5]],Thickness[0.006],
Line[{P31n[[i]],FrameKnifeHM[[i]][{1,2,3,4}],VertexColors->VertexColor[[5]],Thickness[0.006],

Line[{P12n[[i]],P22n[[i]]},VertexColors->VertexColor[[6]],Thickness[0.006],
Line[{P22n[[i]],P32n[[i]]},VertexColors->VertexColor[[6]],Thickness[0.006],
Line[{P32n[[i]],FrameThumbHM[[i]][{1,2,3,4}],VertexColors->VertexColor[[6]],Thickness[0.006],

Line[{P13n[[i]],P23n[[i]]},VertexColors->VertexColor[[7]],Thickness[0.006],
Line[{P23n[[i]],P33n[[i]]},VertexColors->VertexColor[[7]],Thickness[0.006],
Line[{P33n[[i]],FrameFingerHM[[i]][{1,2,3,4}],VertexColors->VertexColor[[7]],Thickness[0.006]]],{i,1,6}]



Also the data at this step might be required to model the un-optimized hand, this data has been gathered as follows and exported to a CSV file named PointsUnoptomized. The format is two point of each joint as they are going to be addressed in the joint array, followed by the anchoring points of each axes and finally each end effectors positions. Better explanation of the saved data is shown in the picture. This data can be used by Solidworks automatic modeling macro for automatic modeling of the hand:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80																				

```

PJ1=MakePoint[J1];
PJ11=MakePoint[J11];
PJ21=MakePoint[J21];
PJ31=MakePoint[J31];
PJ12=MakePoint[J12];
PJ22=MakePoint[J22];
PJ32=MakePoint[J32];
PJ13=MakePoint[J13];
PJ23=MakePoint[J23];
PJ33=MakePoint[J33];
P1f=C1;
P11f=C11;
P21f=C21;
P31f=C31;
P12f=C12;
P22f=C22;
P32f=C32;
P13f=C13;
P23f=C23;

```

```
P33f=C33;
Ax=Table[PD[[i,j]][[{1,2,3},4]]+Normalize[PD[[i,j]][[{1,2,3},3]]],{i,1,3},{j,1,6}];
```

```
PointsUnoptomized={PJ1,CommonNormalGen[J1,J11][[2,1]],PJ11,CommonNormalGen[J1,J11][[2,2]],PJ21,CommonNormalGen[J1,J21][[2,2]],PJ31,CommonNormalGen[J21,J31][[2,2]],PJ12,CommonNormalGen[J1,J12][[2,2]],PJ22,CommonNormalGen[J12,J22][[2,2]],PJ32,CommonNormalGen[J22,J32][[2,2]],PJ13,CommonNormalGen[J1,J13][[2,2]],PJ23,CommonNormalGen[J13,J23][[2,2]],PJ33,CommonNormalGen[J23,J33][[2,2]],P1f,P11f,P21f,P31f,P12f,P22f,P32f,P13f,P23f,P33f,PEF1n[[1]],Ax[[1,1]],PEF1n[[2]],Ax[[1,2]],PEF1n[[3]],Ax[[1,3]],PEF1n[[4]],Ax[[1,4]],PEF1n[[5]],Ax[[1,5]],PEF1n[[6]],Ax[[1,6]],PEF2n[[1]],Ax[[2,1]],PEF2n[[2]],Ax[[2,2]],PEF2n[[3]],Ax[[2,3]],PEF2n[[4]],Ax[[2,4]],PEF2n[[5]],Ax[[2,5]],PEF2n[[6]],Ax[[2,6]],PEF3n[[1]],Ax[[3,1]],PEF3n[[2]],Ax[[3,2]],PEF3n[[3]],Ax[[3,3]],PEF3n[[4]],Ax[[3,4]],PEF3n[[5]],Ax[[3,5]],PEF3n[[6]],Ax[[3,6]]};
```

Input Address

F:\Bomansy-2016\out10.lua

```
Export["E\\Bomansy-2016\\OptimizationFiles\\PointsUnoptomized.xls",PointsUnoptomized,"XLS"];
```

4. Analysis-As a closed chain

Drawing the contacts

```
FingerRadius=12;
```

```
OrangeSmallRadius=55;
```

```
OrangeCenterF3=FrameFingerHM[[1]].Ztrans[0,-OrangeSmallRadius];
```

```
Sd0=FrameFingerHM[[1]][[{1,2,3},4]]-FrameThumbHM[[1]][[{1,2,3},4]];
```

```
Sd1=OrangeCenterF3[[{1,2,3},4]]-FrameThumbHM[[1]][[{1,2,3},4]];
```

```
SD0={Sd0,FrameThumbHM[[1]][[{1,2,3},4]]@Sd0};
```

```
SD1={Sd1,FrameThumbHM[[1]][[{1,2,3},4]]@Sd1};
```

```
SD2=CommonNormal[SD0,SD1][[1,1]];
```

```
SD3=CommonNormal[SD1,SD2][[1,1]];
```

```
ThumbFrame=Orthogonalize[{SD1[[1]]/Sqrt[SD1[[1]].SD1[[1]]],SD2[[1]]/Sqrt[SD2[[1]].SD2[[1]]],SD3[[1]]/Sqrt[SD3[[1]].SD3[[1]]]}];
```

```
FrameThumbHMNew=Append[Transpose[{ThumbFrame[[1]],ThumbFrame[[2]],ThumbFrame[[3]],FrameThumbHM[[1]][[{1,2,3},4]]},{0,0,0,1}].Xtrans[Pi/2,0];
```

```
plottthumbNew=generatePlot[{HM2dq[FrameThumbHMNew]},Green,3];
```

```
ContactCenter2=FrameThumbHMNew.Xtrans[0,-FingerRadius];
```

```
ContactCenter3=FrameFingerHM[[1]].Ztrans[0,FingerRadius];
```

```
PlotContact2=generatePlot[{HM2dq[FrameThumbHMNew]},Green,5];
```

```
PlotContact3=generatePlot[{HM2dq[FrameFingerHM[[1]]]},Blue,5];
```

```
PlotContactCenter2=generatePlot[{HM2dq[ContactCenter2]},Green,3];
```

```
PlotContactCenter3=generatePlot[{HM2dq[ContactCenter3]},Blue,3];
```

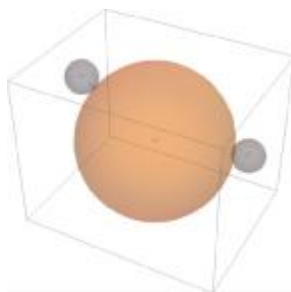
```
PlotCenter3=generatePlot[{HM2dq[OrangeCenterF3]},Blue,3];
```

```
OrangeShape=Graphics3D[{Orange,Opacity[.3],Sphere[OrangeCenterF3[[{1,2,3},4]],OrangeSmallRadius]}];
```

```
ThumbShape=Graphics3D[{Gray,Opacity[.3],Sphere[ContactCenter2[[{1,2,3},4]],FingerRadius]}];
```

```
FingerShape=Graphics3D[{Gray,Opacity[.3],Sphere[ContactCenter3[[{1,2,3},4]],FingerRadius]}];
```

```
Show[OrangeShape,ThumbShape,FingerShape,PlotContact3,PlotCenter3,PlotContactCenter3,PlotContact2,PlotContactCenter2]
```



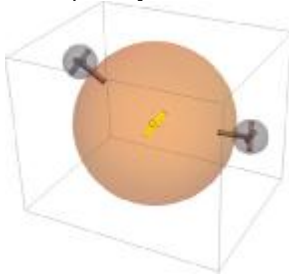
Images for modeling method

```

JointLength=10;
JointshapFC2X=Graphics3D[{Gray,Opacity[.5],Cylinder[{ContactCenter2[{1,2,3},4]]+
JointLength*ContactCenter2[{1,2,3},2]],ContactCenter2[{1,2,3},4]]+JointLength*ContactCenter2[{1,2,3},2]],2}}];
JointshapFC2Y=Graphics3D[{Gray,Opacity[.1],Cylinder[{FrameThumbHMNew[{1,2,3},4]]+
JointLength*ContactCenter2[{1,2,3},1]],FrameThumbHMNew[{1,2,3},4]]+JointLength*ContactCenter2[{1,2,3},1]],2}}];
JointshapFC2Z=Graphics3D[{Gray,Opacity[.5],Cylinder[{ContactCenter2[{1,2,3},4]]+
JointLength*FrameThumbHMNew[{1,2,3},3]],ContactCenter2[{1,2,3},4]]+JointLength*FrameThumbHMNew[{1,2,3},3]],2}}];
JointshapFC3X=Graphics3D[{Gray,Opacity[.5],Cylinder[{ContactCenter3[{1,2,3},4]]+
JointLength*ContactCenter3[{1,2,3},1]],ContactCenter3[{1,2,3},4]]+JointLength*ContactCenter3[{1,2,3},1]],2}}];
JointshapFC3Y=Graphics3D[{Gray,Opacity[.5],Cylinder[{ContactCenter3[{1,2,3},4]]+
JointLength*ContactCenter3[{1,2,3},2]],ContactCenter3[{1,2,3},4]]+JointLength*ContactCenter3[{1,2,3},2]],2}}];
JointshapFC3Z=Graphics3D[{Gray,Opacity[.1],Cylinder[{FrameFingerHM[[1]][{1,2,3},4]]+
JointLength*FrameFingerHM[[1]][{1,2,3},3]],FrameFingerHM[[1]][{1,2,3},4]]+JointLength*FrameFingerHM[[1]][{1,2,3},3]],2}}];
(* JointshapFO2X=Graphics3D[{Yellow,Opacity[3],Cylinder[{OrangeCenterF2[{1,2,3},4]]+
JointLength*OrangeCenterF2[{1,2,3},1]],OrangeCenterF2[{1,2,3},4]]+JointLength*OrangeCenterF2[{1,2,3},1]],3}}];
JointshapFO2Y=Graphics3D[{Blue,Opacity[3],Cylinder[{FrameThumbHM[[1]][{1,2,3},4]]+
JointLength*FrameThumbHM[[1]][{1,2,3},2]],FrameThumbHM[[1]][{1,2,3},4]]+JointLength*FrameThumbHM[[1]][{1,2,3},2]],2}}];
JointshapFO2Z=Graphics3D[{Yellow,Opacity[3],Cylinder[{OrangeCenterF2[{1,2,3},4]]+
JointLength*OrangeCenterF2[{1,2,3},3]],OrangeCenterF2[{1,2,3},4]]+JointLength*OrangeCenterF2[{1,2,3},3]],3}}];*)
JointshapFO3X=Graphics3D[{Yellow,Opacity[3],Cylinder[{OrangeCenterF3[{1,2,3},4]]+
JointLength*OrangeCenterF3[{1,2,3},1]],OrangeCenterF3[{1,2,3},4]]+JointLength*OrangeCenterF3[{1,2,3},1]],2}}];
JointshapFO3Y=Graphics3D[{Yellow,Opacity[3],Cylinder[{OrangeCenterF3[{1,2,3},4]]+
JointLength*OrangeCenterF3[{1,2,3},2]],OrangeCenterF3[{1,2,3},4]]+JointLength*OrangeCenterF3[{1,2,3},2]],2}}];
(* JointshapFO3Z=Graphics3D[{Blue,Opacity[3],Cylinder[{FrameFingerHM[[1]][{1,2,3},4]]+
JointLength*FrameFingerHM[[1]][{1,2,3},3]],FrameFingerHM[[1]][{1,2,3},4]]+JointLength*FrameFingerHM[[1]][{1,2,3},3]],2}}];*)

```

Show[OrangeShape,ThumbShape,FingerShape,PlotContact2,PlotContact3,PlotCenter3,PlotContactCenter2,PlotContactCenter3,JointshapFC2X,JointshapFC2Y,JointshapFC2Z,JointshapFC3X,JointshapFC3Y,JointshapFC3Z,JointshapFO3X,JointshapFO3Y]



here are the joints for the contacts

```

SFx2={ContactCenter2[{1,2,3},1]]/Norm[ContactCenter2[{1,2,3},1]],ContactCenter2[{1,2,3},4]]/(ContactCenter2[{1,2,3},1]]/Norm[ContactCenter2[{1,2,3},1]]));
SFy2={ContactCenter2[{1,2,3},2]]/Norm[ContactCenter2[{1,2,3},2]],ContactCenter2[{1,2,3},4]]/(ContactCenter2[{1,2,3},2]]/Norm[ContactCenter2[{1,2,3},2]]));
SFz2={ContactCenter2[{1,2,3},3]]/Norm[ContactCenter2[{1,2,3},3]],ContactCenter2[{1,2,3},4]]/(ContactCenter2[{1,2,3},3]]/Norm[ContactCenter2[{1,2,3},3]]));
SFx3={ContactCenter3[{1,2,3},1]]/Norm[ContactCenter3[{1,2,3},1]],ContactCenter3[{1,2,3},4]]/(ContactCenter3[{1,2,3},1]]/Norm[ContactCenter3[{1,2,3},1]]));
SFy3={ContactCenter3[{1,2,3},2]]/Norm[ContactCenter3[{1,2,3},2]],ContactCenter3[{1,2,3},4]]/(ContactCenter3[{1,2,3},2]]/Norm[ContactCenter3[{1,2,3},2]]));
SFz3={ContactCenter3[{1,2,3},3]]/Norm[ContactCenter3[{1,2,3},3]],ContactCenter3[{1,2,3},4]]/(ContactCenter3[{1,2,3},3]]/Norm[ContactCenter3[{1,2,3},3]]));

```

Orange Contact axes for the second and third contacts are

$$\begin{aligned} \text{SFxO3} &= (\text{OrangeCenterF3}[\{1,2,3,1\}] / \text{Norm}[\text{OrangeCenterF3}[\{1,2,3,1\}], \text{OrangeCenterF3}[\{1,2,3,4\}]] \otimes (\text{OrangeCenterF3}[\{1,2,3,1\}] / \text{Norm}[\text{OrangeCenterF3}[\{1,2,3,1\}]])); \\ \text{SFyO3} &= (\text{OrangeCenterF3}[\{1,2,3,2\}] / \text{Norm}[\text{OrangeCenterF3}[\{1,2,3,2\}], \text{OrangeCenterF3}[\{1,2,3,4\}]] \otimes (\text{OrangeCenterF3}[\{1,2,3,2\}] / \text{Norm}[\text{OrangeCenterF3}[\{1,2,3,2\}]])); \\ \text{SFzO3} &= (\text{OrangeCenterF3}[\{1,2,3,3\}] / \text{Norm}[\text{OrangeCenterF3}[\{1,2,3,3\}], \text{OrangeCenterF3}[\{1,2,3,4\}]] \otimes (\text{OrangeCenterF3}[\{1,2,3,3\}] / \text{Norm}[\text{OrangeCenterF3}[\{1,2,3,3\}]])); \end{aligned}$$

here are all joint DQs, for the most general case (in future). also for the case of pointy finger joints as S joints are also calculated. at the moment we just take the soft finger with one R joints the contact z direction, so most of these are not used.

```

S1qP=makequat[J1,θP1,0];
S11qP=makequat[J11,θP11,0];
S21qP=makequat[J21,θP21,0];
S31qP=makequat[J31,θP31,0];
S12qP=makequat[J12,θP12,0];
S22qP=makequat[J22,θP22,0];
S32qP=makequat[J32,θP32,0];
S13qP=makequat[J13,θP13,0];
S23qP=makequat[J23,θP23,0];
S33qP=makequat[J33,θP33,0];
SxC2qP=makequat[SFC2,θPXC2,0];
SyC2qP=makequat[SFYC2,θPYC2,0];
SzC2qP=makequat[SFC2,θPZC2,0];
SxC3qP=makequat[SFC3,θPXC3,0];
SyC3qP=makequat[SFYC3,θPYC3,0];
SzC3qP=makequat[SFC3,θPZC3,0];
SxOqP=makequat[SFO3,θPXO,0];
SyOqP=makequat[SFO3,θPYO,0];
SzOqP=makequat[SFO3,θPZO,0];

```

Drawing the chain

codes

```

plotLine[S_,len_]:=Module[{Sh,c1,lS},
  Sh=Sqrt[S[[1]],S[[1]]];
  c1=(S[[1]]⊗S[[2]])/(S[[1]].S[[1]]);
  lS=Line[{c1-len*Sh[[1]],c1+len*Sh[[1]]},VertexColors->{Blue,Green}];
  graT=Graphics3D[{Red,Thick,Opacity[.8],lS,AspectRatio->1,Axes->True}
]
plotLineArrow[S_,len_,arrowHead_,tubeWidth_,Shame_]:=Module[{Sh,c1,lS,graT},
  Sh=Sqrt[S[[1]],S[[1]]];
  c1=(S[[1]]⊗S[[2]])/(S[[1]].S[[1]]);
  lS={Hue[1/6],Arrowheads[arrowHead],Arrow[Tube[{c1,c1+len*Sh[[1]]},tubeWidth]]];
  graT=Graphics3D[{Red,Thick,Opacity[.8],lS,Text[Shame,c1+S[[1]]*10],AspectRatio->1,Axes->True}
]
plotLineArrow2[S_,start_,len_,arrowHead_,tubeWidth_,hue_]:=Module[{Sh,c1,lS,graT},
  Sh=Sqrt[S[[1]],S[[1]]];
  c1=(S[[1]]⊗S[[2]])/(S[[1]].S[[1]]);
  lS={Hue[1/hue],Arrowheads[arrowHead],Arrow[Tube[{c1+start*Sh[[1]],c1+(start+len)*Sh[[1]]},tubeWidth]]];
  graT=Graphics3D[{Red,Thick,Opacity[.8],lS,AspectRatio->1,Axes->True}
]
plotScrew2Arrow[S_,arrowHead_,tubeWidth_]:=Module[{Sh,l,c1,c2,ctot,n,pit,lS},
  l=Length[S];
  Sh=Table[S[[i]]/Sqrt[S[[i,1]].S[[i,1]]],{i,1,l}];

```

```

n=Table[CommonNormal2[S[[i]],S[[i+1]]][[1,1]],{i,1,l-1}];
c1=Table[CommonNormal2[S[[i]],S[[i+1]]][[2,1]],{i,1,l-1}];
c2=Table[CommonNormal2[S[[i]],S[[i+1]]][[2,2]],{i,1,l-1}];
ctot=Append[c1,c2[[l-1]]];
pit=Table[S[[i,1]].S[[i,2]]/(S[[i,1]].S[[i,1]]),{i,1,l}];

IS=Table[{Hue[1/6],Arrowheads[arrowHead*Sgn[pit[[i]]]*pit[[i]]/10],Arrow[Tube[{ctot[[i]],ctot[[i]]+pit[[i]]*Sh[[i,1]]},t
ubeWidth]]},{i,1,l}];
Show[Graphics3D[{Thick,Opacity[.8],IS,AspectRatio->1,Axes->True,Boxed->True,
ViewPoint->{1,1,1},ImageSize->600,PlotRange->{{-25,25},{-25,25},{-25,25}}},DisplayFunction-
->{PopupWindow[Button["Click here"],#,
WindowSize->{650,650}&}]
]
plotCylinder[S_,len_,r_,off_,col_]:=Module[{Sh,c1,IS},
Sh=S/Sqrt[S[[1]].S[[1]]];
c1=(S[[1]]#S[[2]])/(S[[1]].S[[1]]);
IS=Cylinder[{c1+(off)*len*Sh[[1]],c1+(off+2)*len*Sh[[1]],r};
graT=Graphics3D[{col,Thick,Opacity[.8],IS,AspectRatio->1,Axes->True}
]
plotLink[S1_,S2_,len_,r_,off1_,off2_,col_]:=Module[{Sh1,Sh2,c1,c2,IS},
Sh1=S1/Sqrt[S1[[1]].S1[[1]]];
c1=(S1[[1]]#S1[[2]])/(S1[[1]].S1[[1]]);
Sh2=S2/Sqrt[S2[[1]].S2[[1]]];
c2=(S2[[1]]#S2[[2]])/(S2[[1]].S2[[1]]);
IS=Cylinder[{c1+off1*len*Sh1[[1]],c2+off2*len*Sh2[[1]],r};
graT=Graphics3D[{col,Thick,Opacity[.8],IS,AspectRatio->1,Axes->True}
]

plotPalm[S1_,S2_,S3_,S4_,S5_,S6_,len_,r_]:=Module[{Sh1,Sh2,Sh3,Sh4,Sh5,Sh6,c1,c2,c3,c4,c5,c6,IS},
Sh1=S1/Sqrt[S1[[1]].S1[[1]]];
c1=(S1[[1]]#S1[[2]])/(S1[[1]].S1[[1]]);
Sh2=S2/Sqrt[S2[[1]].S2[[1]]];
c2=(S2[[1]]#S2[[2]])/(S2[[1]].S2[[1]]);
Sh3=S3/Sqrt[S3[[1]].S3[[1]]];
c3=(S3[[1]]#S3[[2]])/(S3[[1]].S3[[1]]);
Sh4=S4/Sqrt[S4[[1]].S4[[1]]];
c4=(S4[[1]]#S4[[2]])/(S4[[1]].S4[[1]]);
Sh5=S5/Sqrt[S5[[1]].S5[[1]]];
c5=(S5[[1]]#S5[[2]])/(S5[[1]].S5[[1]]);
Sh6=S6/Sqrt[S6[[1]].S6[[1]]];
c6=(S6[[1]]#S6[[2]])/(S6[[1]].S6[[1]]);

IS=Polygon[{c1+0*len*Sh1[[1]],c5+1*len*Sh5[[1]],c2+1*len*Sh2[[1]],c3+0*len*Sh3[[1]],c4+0*len*Sh4[[1]],c6+0.5*len
*Sh6[[1]],c5+1*len*Sh5[[1]]};
graT=Graphics3D[{Blue,Thick,Opacity[.8],IS,AspectRatio->1,Axes->True}
]
plotPalm2[S1_,S2_,S3_,len_,r_,w_]:=Module[{Sh1,Sh2,Sh3,Sh4,Sh5,Sh6,c1,c2,c3,c4,c5,c6,IS},
Sh1=S1/Sqrt[S1[[1]].S1[[1]]];
c1=(S1[[1]]#S1[[2]])/(S1[[1]].S1[[1]]);
Sh2=S2/Sqrt[S2[[1]].S2[[1]]];
c2=(S2[[1]]#S2[[2]])/(S2[[1]].S2[[1]]);
Sh3=S3/Sqrt[S3[[1]].S3[[1]]];
c3=(S3[[1]]#S3[[2]])/(S3[[1]].S3[[1]]);
IS=Polygon[{c1+w[[1]]*len*Sh1[[1]],c2+w[[2]]*len*Sh2[[1]],c2+(w[[2]]-
0.2)*len*Sh2[[1]],c3+w[[3]]*len*Sh3[[1]],c1+(w[[1]]+1)*len*Sh1[[1]]};
graT=Graphics3D[{Red,Thick,Opacity[.8],IS,AspectRatio->1,Axes->True}
]
plotPalm3[S1_,S2_,S3_,len_,r_,w_]:=Module[{Sh1,Sh2,Sh3,Sh4,Sh5,Sh6,c1,c2,c3,c4,c5,c6,IS},

```

```

Sn1=S1/Sqrt[S1[[1]].S1[[1]]];
c1=(S1[[1]]⊗S1[[2]])/(S1[[1]].S1[[1]]);
Sn2=S2/Sqrt[S2[[1]].S2[[1]]];
c2=(S2[[1]]⊗S2[[2]])/(S2[[1]].S2[[1]]);
Sn3=S3/Sqrt[S3[[1]].S3[[1]]];
c3=(S3[[1]]⊗S3[[2]])/(S3[[1]].S3[[1]]);
IS=Polygon[(c1+w[[1]]*len*Sn1[[1]],c2+w[[2]]*len*Sn2[[1]],c2+(w[[2]]-
0.2)*len*Sn2[[1]],c3+w[[3]]*len*Sn3[[1]],c1+(w[[1]]+1)*len*Sn1[[1]]];
graT=Graphics3D[{Blue,Thick,Opacity[.8],IS,AspectRatio->1,Axes->True]
]
plotFTP[Sl_,p_,len_,r_,col_]:=Module[{Sn1,Sn2,Sn3,Sn4,c1,c2,c3,c4,ISI,S2},
Sn1=S1/Sqrt[S1[[1]].S1[[1]]];
c1=(S1[[1]]⊗S1[[2]])/(S1[[1]].S1[[1]]);
IS=Oylinder[(c1-5*len*Sn1[[1]],p),r];
IS2=Sphere[p,r];
graT=Graphics3D[{col,Thick,Opacity[.8],ISI,S2},AspectRatio->1,Axes->True]
]

```

draw

```

A12g={0,Cos[1.2],Sin[1.2]};A22g={0,Cos[.98],Sin[.98]};A32g={0,Cos[1.2],Sin[1.2]};A42g={0,Cos[1.3],Sin[1.3]};
P12g={-20,0,0};P22g={-15,17,17};P32g={-11,22,22};P42g={-5,25,25};
A13g={0,Cos[1],Sin[1]};A23g={0,Cos[.9],Sin[.9]};A33g={0,Cos[.9],Sin[.9]};A43g={0,Cos[1.1],Sin[1.1]};
P13g={30,30,0};P23g={20,22,-17};P33g={11,22,-22};P43g={5,25,-25};

```

```

SI2g={A12g,P12g⊗A12g};
S22g={A22g,P22g⊗A22g};
S32g={A32g,P32g⊗A32g};
S42g={A42g,P42g⊗A42g};
SI3g={A13g,P13g⊗A13g};
S23g={A23g,P23g⊗A23g};
S33g={A33g,P33g⊗A33g};
S43g={A43g,P43g⊗A43g};

```

```

ax12g=plotLineArrow[SI2g,7,0.05,0.1,"SI2"];
j12ag=plotOylinder[SI2g,1.5,2.5,-3,Black];
j12bg=plotOylinder[SI2g,1,2,-1.5,Green];
l12to22g=plotLink[SI2g,S22g,1,.8,-.5,.5,Green];

```

```

ax22g=plotLineArrow[S22g,9,0.05,0.1,"S22"];
j22g=plotOylinder[S22g,1,2,1,Green];
j22g=plotOylinder[S22g,1,2,-1,Green];
l21to31g=plotLink[S22g,S32g,1,.8,-.5,.5,Green];

```

```

ax32g=plotLineArrow[S32g,9,0.05,0.1,"S32"];
j32g=plotOylinder[S32g,1,2,-1.5,Green];
j32g=plotOylinder[S32g,1,2,0.5,Green];
l32to42g=plotLink[S32g,S42g,1,.8,1,1,Green];

```

```

ax42g=plotLineArrow[S42g,10,0.05,0.1,"S42"];
j42g=plotOylinder[S42g,1,2,-1.5,Green];
j42g=plotOylinder[S42g,1,2,1,Green];

```

```

ax13g=plotLineArrow[SI3g,7,0.05,0.1,"SI3"];
j13ag=plotOylinder[SI3g,1.5,2.5,-3,Black];

```

```

j13bg=plotCylinder[S13g,1,2,-1.5,Blue];
l13to23g=plotLink[S13g,S23g,1,.8,0.1,1,Blue];

ax23g=plotLineArrow[S23g,9,0.05,0.1,"S23"];
j23g=plotCylinder[S23g,1,2,-1.5,Blue];
j23g=plotCylinder[S23g,1,2,0.5,Blue];
l23to33g=plotLink[S23g,S33g,1,.8,2,1,Blue];

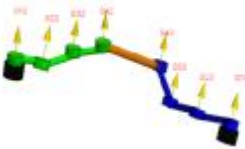
ax33g=plotLineArrow[S33g,9,0.05,0.1,"S33"];
j33g=plotCylinder[S33g,1,2,-1.5,Blue];
j33g=plotCylinder[S33g,1,2,0.5,Blue];
l33to43g=plotLink[S33g,S43g,1,.8,2,1,Blue];

ax43g=plotLineArrow[S43g,10,0.05,0.1,"S43"];
j43g=plotCylinder[S43g,1,2,-1.5,Blue];
j43g=plotCylinder[S43g,1,2,1,Blue];

(*rd=((S43g[[1]]S43g[[2]])/(S43g[[1]].S43g[[1]]))-((S42g[[1]]S42g[[2]])/(S42g[[1]].S42g[[1]]))/2
od=((S43g[[1]]S43g[[2]])/(S43g[[1]].S43g[[1]]))+((S42g[[1]]S42g[[2]])/(S42g[[1]].S42g[[1]]))/2
OrangeG=Graphics3D[{Orange,Opacity[.3],Sphere[{od,rd}]]};*)
OrangeG=plotLink[S42g,S43g,1,1,2,1.7,Orange];

Show[{ax12g,j12ag,j12bg,l12to22g,ax22g,j22g,j22g,l21to31g,ax32g,j32g,j32g,l32to42,
ax42g,j42g,j42g,ax13g,j13ag,j13bg,l13to23g,ax23g,j23g,j23g,l23to33g,ax33g,j33g,j33g,l33to43,ax43g,j43g,j43g,Orange
G},Boxed->False,Axes->None]

```



FK-Numerical

Equations

```
FK2P=Flatten[quatmultSep[quatmultSep[quatmultSep[S12qP,S22qP],S32qP],SzC2qP]];
```

```
FK3P=Flatten[quatmultSep[quatmultSep[quatmultSep[S13qP,S23qP],S33qP],SzC3qP]];
```

here we solve the equation for the two branches to be closed at orange center. I have put the equations in different formats but it never finishes solving in any case.

At first I get the first the angle of joints ($\theta_{P12}, \theta_{P13}$) in both chains as variable then i put equations again and I gave them cos of the first axes angles some value

```
CommonNormal[J12,J22][[1,{1,2}]]
```

```
CommonNormal[J22,J32][[1,{1,2}]]
```

```
CommonNormal[J32,SzC2][[1,{1,2}]]
```

```
CommonNormal[J13,J23][[1,{1,2}]]
```

```
CommonNormal[J23,J33][[1,{1,2}]]
```

```
CommonNormal[J33,SzC3][[1,{1,2}]]
```

```
{{{0.091105168477990181,0.51314285940092996,-0.85345430699155444},{-
```

```
190.74131862900034,248.98342102393362,129.34089585575223}},177.08920532546745}
```

```
{{{-0.0242590206323236308,-
```

```
0.210880391675928140,0.97721080649190921},{324.65805215005166,25.190013179602260,13.495518207108172}
```

```
},101.29847748994626}
```

```
{{{-0.85803899095429,-0.502518014759976,0.106041194089068},{-
```

```
10.8536448737069,23.7293424531612,24.6278966202226}},143.133402776594}
```



```
{{{-0.53303836303606777,-0.21862483807841029,-0.81735750055102456},{-
173.52930758533280,84.32744581262408,90.61115095632972}},18.63865377029934}
{{{-0.33520242251902626,-0.27944115427906569,-0.89975106403524725},{-
108.45005321189721,77.24170413826473,16.413661718646139}},110.54242951231010}
{{{0.47906650194125207,0.35824911183886223,0.80134440821936152},{142.81193494079590,-
55.20356730216650,-60.69772828301313}},124.77544308637247}
```

Matlab- Numerical

1. third case

```
m1=360;
n1=360;
thetaPin1=N[Table[n*2*Pi/m1,{n,0,m1}]];
thetaPin2=N[Table[n*2*Pi/n1,{n,0,m1}]];

eq=Fk2F[{1,2,3,5,6,7}]-Fk3F[{1,2,3,5,6,7}];
eqsfk2a=Table[Table[Chop[Flatten[eq/.{thetaPin2->thetaPin1[[i]],thetaPin3->thetaPin2[[j]],thetaP22->tP22,thetaP32->tP32,thetaPZC2-
->tP42,thetaP13->tP13,thetaP23->tP23,thetaP33->tP33,thetaPZC3->tP43}]],{i,1,Length[thetaPin1]},{j,1,Length[thetaPin2]}];
```

Matlab

nbegin=2

nend=360

Names

```
Iname=Range[1,Length[thetaPin1]];
Jname=Range[1,Length[thetaPin2]];
FileAddress=Table[Table[StringJoin[add,"a",ToString[Iname[[i]]], "b",ToString[Iname[[j]]], ".m"],{j,1,Length[thetaPin2]},{i,1,
Length[thetaPin1]}];
Analysisname=Table[Table[StringJoin["analysis", "a",ToString[Iname[[i]]], "b",ToString[Iname[[j]]], ".m"],{j,1,Length[thetaPin
2]},{i,1,Length[thetaPin1]}];
Analysis=Table[Table[StringJoin["analysis", "a",ToString[Iname[[i]]], "b",ToString[Iname[[j]]],{j,1,Length[thetaPin2]},{i,1,Le
ngth[thetaPin1]}];
DataAddress=Table[StringJoin[add2,ToString[i], ".mat"],{i,1,(Length[thetaPin1]*Length[thetaPin2])};
```

Writing to Matlab

```
hh=Table[F[k],{k,1,6}];
```

```
var={tP22,tP32,tP42,tP23,tP33,tP43};
```

```
Table[Table[Analysisname[[i,j]]=OpenWrite[FileAddress[[i,j]],PageWidth->300],{j,1,Length[thetaPin2]},{i,nbegin,nend}];
Table[Table[OutputStream[FileAddress[[i,j]],125],{j,1,Length[thetaPin2]},{i,nbegin,nend}];
Table[Table[WriteString[Analysisname[[i,j]],StringJoin["function F=",
ToString[Analysis[[i,j]]], "(x)"]],{j,1,Length[thetaPin2]},{i,nbegin,nend}];
Table[Table[WriteString[Analysisname[[i,j]],"\n\n"],{j,1,Length[thetaPin2]},{i,nbegin,nend}];
Table[Table[WriteString[Analysisname[[i,j]],
"tP22=x(1),tP32=x(2),tP42=x(3),tP23=x(4),tP33=x(5),tP43=x(6);\n\n"],{j,1,Length[thetaPin2]},{i,nbegin,nend}];
```

```
Table[Table[Table[WriteMatlab[eqsfk2a[[i,j,k]],Analysisname[[i,j]],
hh[[k]],{k,1,6}],{j,1,Length[thetaPin2]},{i,nbegin,nend}];
Table[Table[Close[Analysisname[[i,j]]],{j,1,Length[thetaPin2]},{i,nbegin,nend}];
```

5.OutPut Plots

```
thetaOut=Table[theta,{j,1,Length[thetaPin2]},{i,1,Length[thetaPin1]}];
Table[Table[thetaOut[[i,j]]=Table[x[k],{k,(Length[thetaPin2](int)(i-1)+(j-1)6+1),(Length[thetaPin2](int)(i-1)+(j-
1)6+int)}],{j,1,Length[thetaPin2]},{i,1,Length[thetaPin2]}];
theta22=Flatten[Table[Table[thetaOut[[i,j,1]],{j,1,Length[thetaPin2]},{i,1,Length[thetaPin2]}],1];
theta32=Flatten[Table[Table[thetaOut[[i,j,2]],{j,1,Length[thetaPin2]},{i,1,Length[thetaPin2]}],1];
theta42=Flatten[Table[Table[thetaOut[[i,j,3]],{j,1,Length[thetaPin2]},{i,1,Length[thetaPin2]}],1];
theta23=Flatten[Table[Table[thetaOut[[i,j,4]],{j,1,Length[thetaPin2]},{i,1,Length[thetaPin2]}],1];
theta33=Flatten[Table[Table[thetaOut[[i,j,5]],{j,1,Length[thetaPin2]},{i,1,Length[thetaPin2]}],1];
theta43=Flatten[Table[Table[thetaOut[[i,j,6]],{j,1,Length[thetaPin2]},{i,1,Length[thetaPin2]}],1];
```

```

theta22Points=Flatten[Table[Table[{θPin1[[i]],θPin2[[j]],θout[[i,j,1]]},{j,1,Length[θPin2]}},{i,1,Length[θPin2]}],1];
theta32Points=Flatten[Table[Table[{θPin1[[i]],θPin2[[j]],θout[[i,j,2]]},{j,1,Length[θPin2]}},{i,1,Length[θPin2]}],1];
theta42Points=Flatten[Table[Table[{θPin1[[i]],θPin2[[j]],θout[[i,j,3]]},{j,1,Length[θPin2]}},{i,1,Length[θPin2]}],1];
theta23Points=Flatten[Table[Table[{θPin1[[i]],θPin2[[j]],θout[[i,j,4]]},{j,1,Length[θPin2]}},{i,1,Length[θPin2]}],1];
theta33Points=Flatten[Table[Table[{θPin1[[i]],θPin2[[j]],θout[[i,j,5]]},{j,1,Length[θPin2]}},{i,1,Length[θPin2]}],1];
theta43Points=Flatten[Table[Table[{θPin1[[i]],θPin2[[j]],θout[[i,j,6]]},{j,1,Length[θPin2]}},{i,1,Length[θPin2]}],1];

```

```

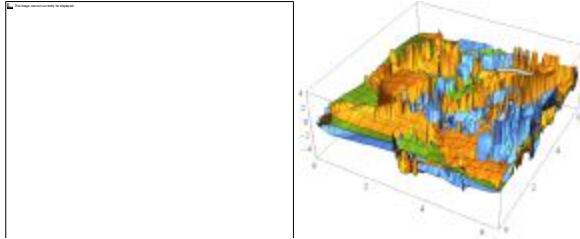
thetaAll=Flatten[Table[Table[Flatten[{θPin1[[i]],θPin2[[j]]},Table[θout[[i,j,k]],{k,1,6}]],{j,1,Length[θPin2]}},{i,1,Length[θPin2]}],1];

```

```

thetaAll[[1]]
ListPlot3D[{theta22Points,theta32Points,theta42Points}]
ListPlot3D[{theta23Points,theta33Points,theta43Points}]

```



6. Analysis- Workspace

Workspace of each finger with wrist:

```

S12qP=N[Table[makequat[J12,thetaAll[[i,1]],0],{i,1,Length[thetaAll]}]];
S22qP=N[Table[makequat[J22,thetaAll[[i,3]],0],{i,1,Length[thetaAll]}]];
S32qP=N[Table[makequat[J32,thetaAll[[i,4]],0],{i,1,Length[thetaAll]}]];
S42qP=N[Table[makequat[SPzC2,thetaAll[[i,5]],0],{i,1,Length[thetaAll]}]];
S13qP=N[Table[makequat[J13,thetaAll[[i,2]],0],{i,1,Length[thetaAll]}]];
S23qP=N[Table[makequat[J23,thetaAll[[i,6]],0],{i,1,Length[thetaAll]}]];
S33qP=N[Table[makequat[J33,thetaAll[[i,7]],0],{i,1,Length[thetaAll]}]];
S43qP=N[Table[makequat[SPzC3,thetaAll[[i,8]],0],{i,1,Length[thetaAll]}]];

```

```

FK2P=Table[quatmultSep[quatmultSep[quatmultSep[S12qP[[i]],S22qP[[i]],S32qP[[i]],S42qP[[i]]],{i,1,Length[thetaAll]}]];
FK3P=Table[quatmultSep[quatmultSep[quatmultSep[S13qP[[i]],S23qP[[i]],S33qP[[i]],S43qP[[i]]],{i,1,Length[thetaAll]}]];

```

```

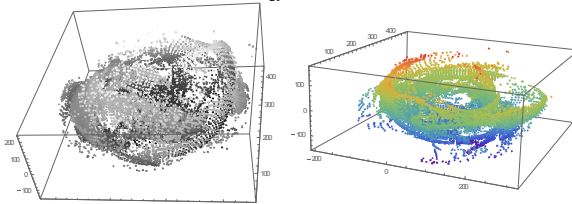
point2=Table[quatmultSep[quatmultSep[FK2P[[i]],HM2dq[OrangeCenterF3]],quatconj4[FK2P[[i]]][[2,{1,2,3}]],{i,1,Length[FK2P]}]];
point3=Table[quatmultSep[quatmultSep[FK3P[[i]],HM2dq[OrangeCenterF3]],quatconj4[FK3P[[i]]][[2,{1,2,3}]],{i,1,Length[FK3P]}]];

```

```

P2PointPlot=ListPointPlot3D[point2,ColorFunction->GrayLevel]
P3PointPlot=ListPointPlot3D[point3,ColorFunction->"Rainbow"]

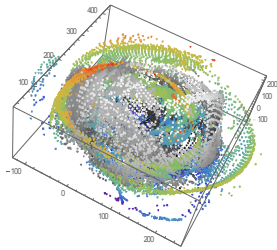
```



```

Show[P2PointPlot,P3PointPlot]

```



Show[{P2PointPlot,OrangeShape},Boxed->False]

