

Use Authorization

In presenting this dissertation 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 dissertation 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 dissertation for financial gain shall not be allowed without my written permission.

Signature _____

Date _____

**Workspace Synthesis for Articulated
Systems
with Application to Exoskeleton Design**

by
Yimesker S Yihun

A dissertation
submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy in Engineering and Applied Science
College of Engineering, Idaho State University
Spring 2014

To the Graduate Faculty:

The members of the committee appointed to examine the dissertation of Yimesker Siraw Yihun find it satisfactory and recommend that it be accepted.

_____ Major Advisor

Dr. Alba Perez-Garcia (Associate Professor of Mechanical Engineering)

_____ Committee Member

Dr. Marco P. Schoen (Professor of Mechanical Engineering)

_____ Committee Member

Dr. Ken W. Bosworth (Professor of Computer Science and Mechanical Engineering)

_____ Committee Member

Dr. D. Subbaram Naidu (Professor of Electrical Engineering)

_____ Graduate Faculty Representative (GFR)

Dr. Steve Chiu (Associate Professor of Electrical Engineering)

Dedication

To my brother, Semere Siraw Yihun and my grand ma, Degitu Beyene, who are now sorely missed.

Acknowledgments

I would like to express my sincere gratitude to my advisor Dr. Alba Perez-Gracia, for her continuous support of my Ph.D study and research, for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also goes to Dr.D. Subbaram Naidu, Dr.Steve Chiu, Dr. Marco P. Schoen, Dr.Ken W. Bosworth, for offering me the opportunity to involve in their research groups and exciting projects; for their encouragement and insightful comments being my thesis committee member.

I also would like to acknowledge the financial support by the Telemedicine Advanced Technology Research Center (TATRC) and the National Science Foundation NRI.

I thank my fellow research mates in Measurement and Control Research Center at Idaho State University for the stimulating discussions and for all the fun we have had in the last couple of years.

Last but not the least, for my loving, supportive, encouraging, and patient wife, Yetnayet Tsegaw, whose faithful support in all my life is so appreciated. She was always there cheering me up and stood by me through the good times and bad. Thank you.

Contents

List of Figures	x
List of Tables	xiii
Nomenclature	xiv
Abstract	xv
1 Introduction	1
1.1 Background	5
1.2 Design of Robotic Systems	8
1.3 Kinematic synthesis	9
1.4 Research Contributions	12
1.5 Organization of the Dissertation	12
2 Kinematics Background	14
2.1 Kinematics Theory	14
2.1.1 Spatial Displacements	14
2.1.1.1 Homogeneous Transformations and Forward Kinematics	17
2.2 Kinematics Synthesis	23
3 Methodology for the Design of Exoskeletons	25

3.1	The Design Process	25
3.2	Human Motion Capture	27
3.3	Kinematic Synthesis	28
3.3.1	Type synthesis	29
3.3.2	Dimensional synthesis	30
3.4	Post-Synthesis Optimization and CAD Implementation	31
3.5	Exoskeleton Design Examples	34
4	Exact Workspace Synthesis of Spatial Linkages	36
4.1	Exact Workspace Synthesis for RCCR linkages	37
4.1.1	The Workspace of Finite Displacements of the RC-CR Linkage	40
4.1.2	Dimensional Synthesis for the RCCR Linkage	43
4.2	Example and Results	45
4.2.1	The absolute displacements workspace	47
4.3	Workspace of RRRR linkage	51
4.3.1	Workspace of the RR chain	51
4.3.2	Implicitization of the relative transformation of the RR chain	52
4.4	Summary	53
5	Link-based performance optimization of spatial mechanisms	54
5.1	Introduction	54
5.2	Link-Based Optimization	56
5.2.1	Optimization problem formulation	56
5.2.2	Additional Constraint Functions	61
5.2.2.1	Link length and joint offset constraints	61
5.2.2.2	Constraints for obstacle avoidance	62
5.2.2.3	Force transmission specifications	66
5.2.2.4	Planarity	67
5.2.3	Configuration-dependent constraints	68
5.3	Overall optimization strategy	70

5.4	Examples	72
5.4.1	CRR-RRR mechanism	73
5.4.1.1	Problem Definition and Formulation	73
5.4.2	The Bennett mechanism	80
5.4.2.1	Problem Definition and Formulation	80
5.4.3	Application in robotic hand design	85
5.4.4	Summary	88
6	Design of an Exoskeleton as a Finger-Joint Angular Sensor	89
6.1	Introduction	89
6.2	Design methodology	90
6.3	Collection of design input data	90
6.3.1	Image acquisition	90
6.3.2	Image processing	92
6.4	Finger Exoskeleton Design	94
6.4.1	Exoskeleton Design Equations	95
6.5	Results	99
6.6	Summary	100
7	Single Degree-of-Freedom Exoskeleton Mechanism Design for Thumb	
	Rehabilitation	104
7.1	Introduction	104
7.2	Thumb Mechanism Design	105
7.2.1	Mechanism Selection	107
7.2.2	Mechanism Design Equations	108
7.2.2.1	Implementation of the Design Equations	111
7.3	Results	112
7.4	Summary	116

8	Conclusions and Future works	117
8.1	Conclusions	117
8.2	Future works	118
	References	134
	Appendix	135

List of Figures

2.1	Pure translation and rotation of a rigid body	15
2.2	The screw axis, rotation angle and slide of a displacement	17
2.3	A serial chain with its joints[86]	19
3.1	The general design methodology	26
3.2	Markers placed in the thumb and data capture setup	27
3.3	CAD model for manual adjustment and optimization; Sliding anchor point from (a) to (b) and from (c) to (d)	33
4.1	Methodology for the exact workspace synthesis	38
4.2	RCCR linkage	39
4.3	Workspace of relative displacements for the parallel RC chain, left; for the RC-CR linkage, right.	42
4.4	Three of the six RCCR workspaces with the task points	46
4.5	The absolute work space of the RC chain with points on the surface	48
4.6	The absolute work space of the RCRC	49
4.7	RCCR linkage created with chains 2 and 3, showing one of its circuits.	50
4.8	RCCR linkage created with chains 2 and 4, showing one of its circuits.	50
4.9	Plot of the workspace of relative translations of a randomly-generated RR chain	52
5.1	Overall design strategy. Link-based optimization stages are shown inside the broken lines	58

5.2	Joint axes for two spatial mechanism topologies:(a) A closed linkage, CRR-RRR; (b) A linkage with a tree structure, 3R-(2R,2R)	59
5.3	Constraint Region Represented by a Cylindrical Surface	64
5.4	Constraint Region Represented by a spherical Surface	64
5.5	Schematic diagram for the transmission angle	67
5.6	Desired trajectory of the exoskeleton design(CRR-RRR)	69
5.7	Trajectory for the Bennett linkage	70
5.8	Initial CRR-RRR Mechanism. Black links are located at the common normal lines between joints.	74
5.9	Mechanism obtained after link length constraints used	76
5.10	Mechanism obtained after region avoidance constraint used	76
5.11	Mechanism Obtained After Optimizing Region Avoidance, Overall Length and Force Transmission	77
5.12	Motion of the mechanism obtained after optimizing region avoidance, overall length and force transmission. Five positions along the trajectory are reached by the mechanism while avoiding the obstacle.	78
5.13	The CAD model for the final optimized solution	79
5.14	The Bennett Linkage used as a hinge and a cabinet door (Courtesy of G. Lachhwani at PsiStar Solutions)	80
5.15	Initial Solution of the Bennett Linkage	82
5.16	Optimized Bennett Linkage with Link Length Constraints	83
5.17	Mechanism with obstacle avoidance and offset constraint. The blue square corresponds to the cabinet door.	84
5.18	Mechanism obtained with link length, offset length and obstacle avoidance constraints.	84
5.19	Motion of the final design for the cabinet linkage	84
5.20	Trajectory of the three fingers to rotate a door nob	85
5.21	The robotic hand before optimization	86
5.22	Optimized with link length and offset constraints	87
5.23	A three fingered robotic hand rotate a door nob	87

6.1	The methodology applied to design the exoskeleton based sensor . . .	91
6.2	The captured image of the index finger with the square blobs on it . . .	92
6.3	The index finger movement and the square blobs	93
6.4	Corner detection of one frame done by Harris corner detection	93
6.5	Schematic drawing of the exoskeleton on the index finger	96
6.6	The six- bar linkage with variables used	97
6.7	CAD model of the selected linkage and the sensor on the hand	101
6.8	Initial prototype of the selected linkage	102
6.9	The modified CAD model of the exoskeleton and the sensor on the hand	102
6.10	The experimental setup of the sensor	102
6.11	Labview program to show and record the data	103
6.12	The joint angle data	103
7.1	Thumb's proximal phalanx point path	106
7.2	Thumb's proximal phalanx path: screw surface of relative screw axes	107
7.3	A spatial 4-bar CCCC linkage	108
7.4	A spatial 6-bar CCCCCC linkage	109
7.5	One of the thumb paths (thin frames) with superimposed linkage path (thick lines)	113
7.6	Comparison between design positions (thin lines) and linkage positions (thick lines)	114
7.7	The two initial solutions selected for prototyping	114
7.8	Other solutions for different data set	115
7.9	prototype of the initial solutions	116

List of Tables

4.1	Goal points and solution RC chains	45
5.1	Plucker Coordinates of the CRR-RRR Screw axes [mm]	75
5.2	Cylinder position and size in [mm]	75
5.3	Link Lengths [mm] obtained through different optimization stages: (1) Common normal lines directly from synthesis; (2) Link-size optimization;(3) Region avoidance; (4) Link size and region avoidance	77
5.4	Plucker Coordinates of the Bennett Linkage Screw Axes [mm]	81
6.1	Exoskeleton dimensions	100

Nomenclature

$[M]$	A matrix.
\mathbf{s}	A vector; also a three-dimensional point.
\mathbf{S}	A dual vector; $\mathbf{S} = \mathbf{s} + \epsilon \mathbf{s}^0$.
\hat{Q}	A dual quaternion.
\hat{q}	A dual number.
Q	A quaternion.
ϵ	The dual unit such that $\epsilon^2 = 0$.
\times	The cross-product of three-dimensional vectors.
\cdot	The dot product of vectors.
$[Z(\theta, d)]$	A screw displacement about the Z axis, of rotation θ and translation d .

Abstract

With advances in robotics and mechatronics, exoskeletons are being widely developed to assist in, rehabilitate or augment human body motions. However, there are still many challenges that need to be overcome in the design of exoskeletons. Research is being done in four main areas: data acquisition, kinematic design, detailed design and instrumentation/actuation; a good design methodology for exoskeletons needs to incorporate all of these aspects. In this research, results are developed in three of these areas, with emphasis on the kinematic design.

The main contribution of this research is the development of a kinematic-synthesis based, multi-step methodology for the design of exoskeletons. In this method, the input is the human motion captured using vision-based or similar systems; research in this area includes the development of a joint-based sensor. The second step is finite-position kinematic synthesis that yields a set of exoskeletons that can follow the desired motion. In this second step, research has been devoted to exploring the synthesis of spatial mechanisms for a given workspace or trajectory, which ensures compatibility with the human user. The exoskeleton is then further optimized to ensure motion smoothness, obstacle avoidance and optimum physical dimensions, using a newly-developed link-based hybrid optimization routine.

Traditionally, the exoskeletons are designed with full degrees of freedom and trying to align with the human joint axes of motion, which relies on several premises such as identification of the accurate location of the axes and the simplification of their motion. The main advantage of the method developed in this research over other methods is that it does not need any assumption about the location and type of joints in the subject; the exoskeleton is going to follow the path that is selected as task regardless of the skeletal structure that generates it. This new methodology may help in the creation of new and innovative exoskeleton designs.

Chapter 1

Introduction

Robotic exoskeleton systems are one of the highly active areas of recent robotic research. In the past few years, the demand for high performance robots for daily human activities increased rapidly due to the advancement of robotics technology. New robot technologies, acting in collaboration with humans, have the potential to greatly increase both productivity and quality of life. One such evolving co-robot technology is the robotic exoskeleton, which attaches directly to a human to augment the abilities of the user. Exoskeletons are being developed for many applications, including rehabilitation after neurological injury, strenuous and repetitive work, dangerous work, and military missions. Robotic devices have been shown to be capable of automating the strenuous and repetitive nature of movement therapy after stroke or other neurological injury ([35], [104]). Additionally, robotic devices give scientists a new investigative tool for recording progress during movement training and for determining the factors that promote functional recovery [114].

For the purpose of industrial and medical application, robotic exoskeletons were

studied in the late 1960s and 1970s ([91], [24], [107]). Exoskeletons were also designed to enhance the strength of humans ([15], [63]). Currently, many exoskeleton robots are proposed/ designed for the rehabilitation, haptic interaction, and power augmentation purpose ([61], [39]). However, there are still many challenges and issues that limit the advancement of the exoskeleton design such as the availability of effective controls, higher performance actuators, and advanced power transmission technology that can provide a better power to weight ratio to the exoskeleton. An ideal exoskeleton should generate natural motions of the limb without causing vibration, or sudden motion change and without adding extra load or burden on the user. All these considerations made the design of exoskeleton devices difficult. Since there is a direct interaction with human users, designing exoskeleton robot needs special consideration. Some of the issues that need to be addressed in the mechanical design of exoskeleton are range of motion, comfort, low inertia and safety. Similarly, in the control of exoskeleton issues like controllability, smooth motion generation and flexibility are required. Of particular interest are the ability to design, implement, and test assistive control strategies. Many different control strategies have been developed [27]. The most promising approaches fall into the assist-as-needed category, where an attempt is made to vary the level of assistance to match the impairment level of the patient. For example, in [128] an assist-as-needed controller that learns a model of the patients' abilities while simultaneously reducing assistance when the patient performs well is presented.

Proper evaluation of assist-as-needed control strategies is dependent upon the abilities of the robotic device. Ideally, the device would be able to apply any force

at any speed at any location along the movement trajectory. In practice, this is impossible, but can be approached in a robotic device by keeping apparent inertia and friction low, and the controllable force bandwidth high. The result is a robotic device that can help patients complete movements, but continually challenges them to try. The efficacy of these and other control strategies has been documented ([17], [75], [87]) but it remains unclear what specific controller characteristics increases patient learning during therapy.

A special design procedure needs to be adopted to minimize the effect and advance the exoskeleton technology. Some of the human bodies follow a complex motion that with the traditional design approach is difficult to generate a similar motion as the joint. Traditionally, exoskeletons are designed so that they align with the human joint axes of motion [13], [16], [62]. This assumes that the location of the axis can be accurately known. In addition, that such a fixed axis exists for the range of motion of the joint or set of joints, which is not always the case. A clear example of complex kinematic modeling is the thumb, for which precise detection methods such as MRI segmentation [116] show that considering fixed rotational axes, especially for the CMC joint, is not a good approximation; see also [22], similarly, the human shoulder follows a complex motion that its center of rotation changes with its motion [61]. Which makes the alignment joints of the human with the exoskeleton more difficult as the location of complex human joints change. In this thesis, a systematic design approach is proposed with illustrative examples for the design of exoskeletons that can follow the complex 3D motions of a human body. With this method, it is not necessary to know the geometry of the hand or the targeted body, but rather to have

a description of its motion at the point of attachment.

In this proposed design methodology, the process is divided into two stages. The first stage uses dimensional kinematic synthesis in order to create an articulated system able to follow a specified motion. This stage defines, given the type and number of joints and the loops of the mechanism, the relative position between the joints; this specifies the workspace of the mechanism. Several methods exist for the dimensional kinematic synthesis of linkages. Geometric constraints imposed by the joints can be used to define design equations [83]; robot kinematics equations to reach a set of positions can be stated and solved for both the joint variables and the structural variables [77]. The geometry of the motion is exploited in the kinematic mapping approach as in [56]. In our research, we follow [100].

In this thesis, to maximize the accuracy of the trajectory of the exoskeleton, a new method of exact workspace kinematic synthesis is also developed. For the exact workspace synthesis, the parameterized forward kinematics equations of each serial chain is to be converted to implicit equations via elimination. The implicit description of the workspace will be a function of the structural parameters of the serial chain, making it easy to relate those parameters to a given algebraic surface. The solution strategy for the remaining system of equations is to be assessed as a function of the complexity of the equations obtained. For simple chains, algebraic solution has been obtained and published in [11], however much more complex systems are expected for exoskeletons and the solutions involve numerical solvers.

It is important to notice that any dimensional synthesis method used for the first stage can be used to provide the input data (the joint axes and their connectivity) for

the second stage. The second stage deals with the optimization of the links to satisfy a set of performance requirements. Many of these additional performances, such as motion smoothness, obstacle avoidance, force transmission, or physical dimensions to name a few, are fully or partially independent of the kinematic task. The optimization stage has been successfully developed, implemented and tested in several mechanism designs. It is a general method that can be used to optimize different topologies; such as serial chain, closed linkages, linkages with tree structure and hybrid mechanisms [137]. Preliminary results also published in [138].

The output from the optimization algorithm is used for CAD implementation. To facilitate this implementation, a macro program is used to semi-automate the process of modeling the joint axis and the linkages of the mechanism. This helps to have a 3D visualization and simulation of several candidate solutions easily. The CAD model is also used to check the response of different actuators and their placement in the mechanism.

1.1 Background

Kinematics is the analytical study of the geometry of motion with respect to a certain reference coordinate system without regard to the forces or moments that cause the motion. Hence, within the science of kinematics, one studies the position, velocity, acceleration, and all higher order derivatives of the position variables.

Manipulators are made of nearly rigid links which are connected with joints that allow relative motion of adjacent links. The most common and primitive types of joints are the revolute joint and the prismatic joint. The revolute joint allows rotary

movement between links, and prismatic joint allows linear movement (translation) between consecutive links. Combining these two types of primitive joints, we can create other joint types (Cylindrical, Universal, Spherical etc.) and many useful mechanisms for robot manipulation and locomotion. These two types of joints are simple to build and are well studied in engineering design.

The end-effector is the part mounted on the moving link to do the required job of the robot/manipulator. It is the end-effector or tool that actually performs the work. The arm and wrist assemblies of a robot are used primarily for positioning the end-effector and any tool it may carry. The workspace of a manipulator is defined as the set of all end-effector positions which can be reached by some choice of joint angles. The workspace is used when planning a task for the manipulator to execute all desired motions of the manipulator within the workspace.

In the study of robotics, description of the position and orientation of objects in three dimensional space is required. To obtain this information a coordinate system, or a frame is rigidly attached to the object. Then the position and orientation of this frame are described with respect to a reference coordinate system. Since any frame can serve as a reference system, a transformation technique or changing the description of the object from one frame to another is required.

The number of degrees of freedom of a manipulator is the number of independent position variables which would have to be specified in order to locate all parts of the mechanism. The most commonly used criteria to find the mobility of mechanisms is the *Kutzbach – Grübler* formula. For a robot with n number of links and j number

of joints with each joint allows f_i degrees of freedom, the mobility is given by

$$M = 6(n - 1) - \sum_{i=1}^j (6 - f_i) \quad (1.1.1)$$

Mechanisms that have zero or negative mobility when applying the *Kutzbach – Grübler* criterion but have full cycle mobility with a special geometry are called over-constrained mechanisms.

In order to analyze robotic systems, we need algebraic tools that relate the link dimensions and joints of the robot with the movement able to perform. Basically we need descriptions for the position and velocity of the robot at its spatial configuration. There are two kinematics approaches to describe the spatial configuration of a robot. These are forward kinematics and inverse kinematics. Forward and inverse kinematics of articulated systems studies the analytical relationship between the angular positions of joints and the positions and orientations of the end-effector. For the given joint angles for the robot, forward kinematics finds the orientation and position of the end effectors. Similarly for the given desired end-effector positions, the inverse kinematics finds the joint angles to achieve the given positions. For articulated systems, forward kinematics problems are straightforward and there is no complexity deriving the equations. Hence, there is always a solution for the forward kinematics problem. Whereas, the inverse kinematics problem is one of the most difficult to solve. Since it is comprised of a set of nonlinear simultaneous equations which might result existence of multiple solutions, non-existence of a solution and singularities etc. Such problems get worse with higher DOF mechanisms. Non existence of a solution occurs when

the target point is outside the workspace of the robot and when the target point is coincide with the unreachable point within the workspace of the manipulator due to physical constraints.

1.2 Design of Robotic Systems

Mechanism design for a robot is started by identification of problem requirements such as the desired motion, allowed space, minimization of power consumption and others. This stage is then followed by synthesis, which is comprised of topological or structural synthesis and dimensional synthesis. The topological or structural synthesis includes type synthesis that is combination of the different types of mechanisms for the desired function: cams, gear-trains, linkages, etc, and number synthesis, which is a decision about number, type and connectivity of the component parts for the required degrees-of-freedom. Dimensional synthesis is the calculation of the significant dimensions for each constituent member. For more information on the design of robotic systems, see [122], [135], [105], and [32].

The synthesis is followed by analysis, which is the study of the motion of different members constituting a mechanism, and the mechanism as a whole entity while it is being operated or run [29], [31]. This study of motion involves linear as well as angular position, velocity and acceleration of different points on members of mechanisms etc. The analysis method could be graphical or analytical. The final stage is the optimization, detailed design, test and experimentation. The first two stages of the design process belong to the conceptual design and are of great importance due to the fact that the time and design costs for the analysis, the optimization and the test

or experimentation depend strongly on the feasibility of the chosen concept.

1.3 Kinematic synthesis

Kinematic synthesis is the determination of mechanisms to fulfill certain motion specifications. Spatial mechanisms are used in a great variety of robots and devices; Several methods have been introduced for the kinematic synthesis of spatial mechanisms [29]. Robots can be characterized based on the work space geometry, degrees of freedom and based on their kinematic structures as serial, parallel, and hybrid robots. Based on the type of the workspace, the structure and the degree of freedom required, the synthesis approach and complexity may need to be different. Synthesis of parallel robots has focused mainly on type or structural synthesis, using group theory, screw theory, or geometric methods, see for instance [45], [8], [37]. Dimensional synthesis examples exist, mainly for optimizing performance indices [51], [52], [67] or for reachable workspace sizing [3], [74], [21]; see also [88] for a comprehensive approach.

The dimensional synthesis of spatial serial chains for a prescribed set of positions can be used for the design of parallel robots by synthesizing all supporting legs for the same set of positions. There are a few examples of finite-position dimensional synthesis of parallel robots in the literature, most of them do partial synthesis. Wolbrecht et al. [127] perform synthesis of 3-RRS, 4-RRS, and 5-RRS symmetric parallel manipulators. Kim and Tsai [68] and Rao [102] solve the partial kinematic synthesis of a 3-RPS parallel manipulator. This method has been successfully applied mainly to special parallel systems with imposed symmetry. In general, the method does not

allow the control of the final trajectory of the parallel system. In the most extreme cases, it may yield a system with negative mobility, that can be assembled at each task positions, but cannot be driven from task position to task position.

The kinematic mapping is used for the synthesis of planar and spherical linkages in order to state design equations and to provide a tool for visualizing the workspace and trajectories of the linkage. See Ravani and Roth [103] and more recent applications by Hayes [43], Schröcker [108] and Wu and Ge [130]. For spatial motion, Study's kinematic mapping is used to obtain simplified equations for analysis and synthesis, see Husty et al. [56] and [18]. However, the kinematic image for the spatial motion is a six-dimensional quadric and that makes the visualization of workspaces and trajectories difficult for the designer. For the ease of tracking many points along the trajectory in the workspace, a lot of work have been done using the combined method of the kinematic synthesis and Optimization techniques [58], [121], [80], [110].

Most of the current and past literature focuses on the optimization for planar mechanisms, in which the kinematic synthesis mostly approximate synthesis- and the additional requirements are solved simultaneously. The focus in most of them is in optimizing the desired trajectory or motion generation. Some commonly used optimization methods for planar mechanism design are the least-square technique with vector geometric and functional equations [28], the least-square technique with assembly constraints [90], [57] and [41], with the use of loop equation techniques, [46], [33], [95]. Other methods include geometric constraint programming [69], and genetic algorithms ([20], [2]). In [112], the focus is on optimizing performance requirements. Also see [136], [141] and [111].

For the optimized synthesis of spatial linkages, there is research being conducted on using partial dimensioning to optimize some characteristics, for instance workspace, isotropy and dexterity [117], [10], [5] or stiffness [67]. For a summary and literature review of this approach, with its challenges and shortcomings, see [89]. The definition of the *characteristic length* has also been used for the optimization of performance parameters with disparate units, see [66]. In [6], Pareto-optimal solutions are found for the optimization of kinematic and dynamic specifications. Several indices have been defined for optimizing a spatial version of the transmission index [23] and the optimization of motion/force transmissibility using screw theory [129]. The kinematic mapping can also be used for optimization. This method, developed by Ravani and Roth [103], has been used in [47], [43] to optimize dimension and type of a mechanism. Optimization of spatial mechanisms using GA has been done in [9]. The work focuses on the optimization of the link lengths to obtain a closer trajectory. In the case of parallel robots [74], the focus has been to find parameters of the manipulator, whose workspace contains the specified points. Kim and Tsai [67] present the optimization of link lengths and some link position parameters for a 3-CRR parallel manipulator in order to maximize the stiffness for a given workspace volume. In [79], kinematic optimization is applied to optimize the structure of a spatial mechanism that can be used for surgical robot.

1.4 Research Contributions

This dissertation is devoted to develop a method for the design of exoskeletons and spatial mechanisms. Its contributions are listed below.

- Introduce a systematic methodology for the design of exoskeleton devices. The method is comprised of image acquisition, kinematic synthesis, optimization and CAD modeling. The methodology is demonstrated with design examples. See, [138],[139].
- A new optimization algorithm is developed to satisfy different link based objectives and constraints in the design of spatial mechanisms. The algorithm is generic, which can be used for optimizing serial, parallel and hybrid robots. See [140],[137].
- An exact workspace synthesis methodology is introduced. The method helps to visually identify circuit defects and possible trajectories of the mechanism. See, [14]

1.5 Organization of the Dissertation

The dissertation is organized as follow. Chapter 1 discusses the theme of the thesis and literature review and back ground on mechanism design. Chapter 2 provides a mathematical background in kinematic analysis and synthesis. Chapter 3 presents a kinematic synthesis based methodology for the design of exoskeleton devices. Chapter 4 introduces an exact kinematic synthesis approach to synthesis a

parallel over-constraint mechanisms. Chapter 5 introduce a link based post synthesis optimization method to satisfy additional objectives and constraints such as link length, obstacle avoidance, force transmission and others. Chapter 6 presents a design of an exoskeleton device used for measuring a finger joint angle. Chapter 7 presents a design of a single degree of freedom spatial mechanism for thumb rehabilitation. Finally, Chapter 8 presents conclusions and future works.

Chapter 2

Kinematics Background

2.1 Kinematics Theory

Formulating the suitable kinematics models for a robot mechanism is very crucial for analyzing the behavior of the manipulator. To control and program a robot one must have knowledge of both its spatial arrangement and a means of reference to the environment.

2.1.1 Spatial Displacements

A rigid motion of an object is a motion which preserves distance between points. To describe a rigid body motion a coordinate frame is attached to it, in which the coordinates of points of the body remain fixed. Every point can be given coordinates with respect to a frame; In Figure 2.1, a point P has coordinates $[P_X \ P_Y \ P_Z]'$ when measured in frame $\{\mathbf{F}\}$ and $\mathbf{p} = [p_x \ p_y \ p_z]'$ with respect to frame $\{\mathbf{M}\}$. A displacement is modeled as the change of coordinates from points expressed in the moving frame $\{\mathbf{M}\}$ to points expressed in the fixed frame $\{\mathbf{F}\}$.

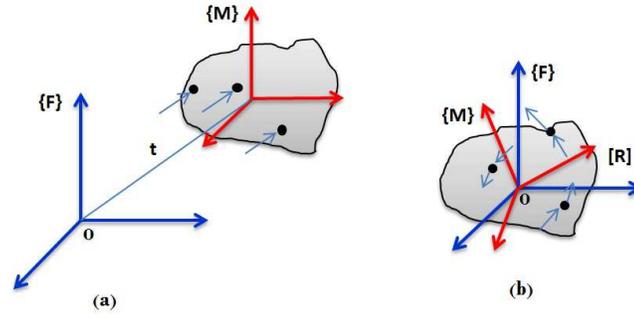


Figure 2.1: Pure translation and rotation of a rigid body

Translation

During a translation, all points of the rigid body move by the same amount in the same direction. If the direction and the magnitude of the translation are given by a vector \mathbf{t} , a point that had coordinates x in frame $\{\mathbf{F}\}$ before the translation, is transformed to a point of coordinates X in the fixed frame $\{\mathbf{F}\}$ after the translation.

$$X = x + t \quad (2.1.1)$$

Rotation

The most common way of representing rotations is by using matrices. The matrix that contains the expression of the column vectors of the moving frame $\{x, y, z\}$ expressed in the fixed frame,

$$[R] = [x, y, z] \quad (2.1.2)$$

The matrix $[R]$ defines the change in coordinates from points of the body expressed in coordinates of the moving frame, to the expression of the same point in the fixed frame. A point that had coordinates x (expressed in the fixed frame) before the

rotation, is transformed to a point of coordinates X after the rotation.

$$X = [R]x \quad (2.1.3)$$

Rotation about one of the coordinate axes of the fixed frame is called Euler angle.

It adopts a very simple expression and it is also easy to visualize. The coordinate

rotations about axes X , Y and Z are given by the matrices.

$$[\mathbf{X}(\alpha)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (2.1.4)$$

$$[\mathbf{Y}(\beta)] = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (2.1.5)$$

$$[\mathbf{Z}(\theta)] = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1.6)$$

A general finite displacement

A general finite displacement can be described as a composition of a translation and

a rotation. The expression of the general displacement for a composition of rotation

by an orthogonal matrix $[R]$ and a translation of a vector d is,

$$X = [R]x + d \quad (2.1.7)$$

A rigid body motion can be described using the tools of linear algebra and screw theory. According to Chasles' theorem [86] a general displacement in 3-dimensional space is equivalent to a screw motion consisting of a rotation of angle θ about and a translation t along a line as Figure 2.2.

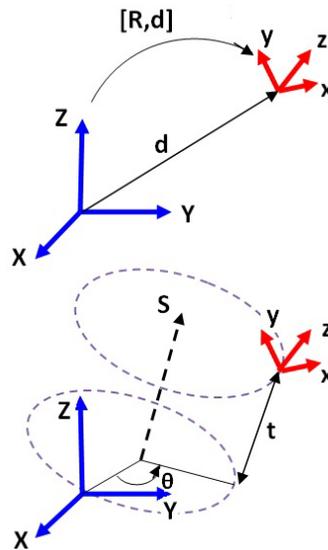


Figure 2.2: The screw axis, rotation angle and slide of a displacement

In the screw axis representation, we need to define six parameters: four to define the direction and location of the line, and two to define the rotation and slide values. The screw-angle is a very efficient way of representing a general displacement.

2.1.1.1 Homogeneous Transformations and Forward Kinematics

All standard transformations (rotation, translation, scaling) can be implemented by matrix multiplications with 4x4 matrices called homogeneous transforms. Homo-

geneous transforms are used to describe the coordinate transformation from a fixed frame F located at the base to a moving frame M located at the end-effector of the robot.

The kinematics equations define the position of the end-effector M of a robot in terms of the coordinate screw displacements along the chain. The Denavit-Hartenberg convention [86] is frequently used to assign reference frames to each link of the robot defined as a series of joint axes denoted $S_i, i = 1, \dots, n$ (Figure 2.3). In this convention, link coordinate frames are attached so that the z-axis is directed along the axis S_i and its x-axis is directed along the common normal A_{ij} . This kind of assignment of frames allows to define the 4 x 4 homogeneous transformation [D] that locates the end-link of a spatial open chain as the composition of local transformations. Considering $x = (x, y, z)^T$, to be coordinates in M and $X = (X, Y, Z)^T$ to be coordinates measured in F, the screw displacement along a joint axis, S_i , is defined by $X = [Z(\theta_i, d_i)]x$, where,

$$[Z(\theta_i, d_i)] = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.8)$$

Similarly, the screw displacement from one joint axis to another is a homogeneous transformation along the X-axis by the amounts a_{ij} and α_{ij} , which is defined by

$X = [X(\alpha_{ij}, a_{ij})]x$, where,

$$[X(\alpha_{ij}, a_{ij})] = \begin{bmatrix} 1 & 0 & 0 & a_{ij} \\ 0 & \cos \alpha_{ij} & -\sin \alpha_{ij} & 0 \\ 0 & \sin \alpha_{ij} & \cos \alpha_{ij} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.9)$$

The inverse of a screw displacement can be obtained by negating its parameter, for

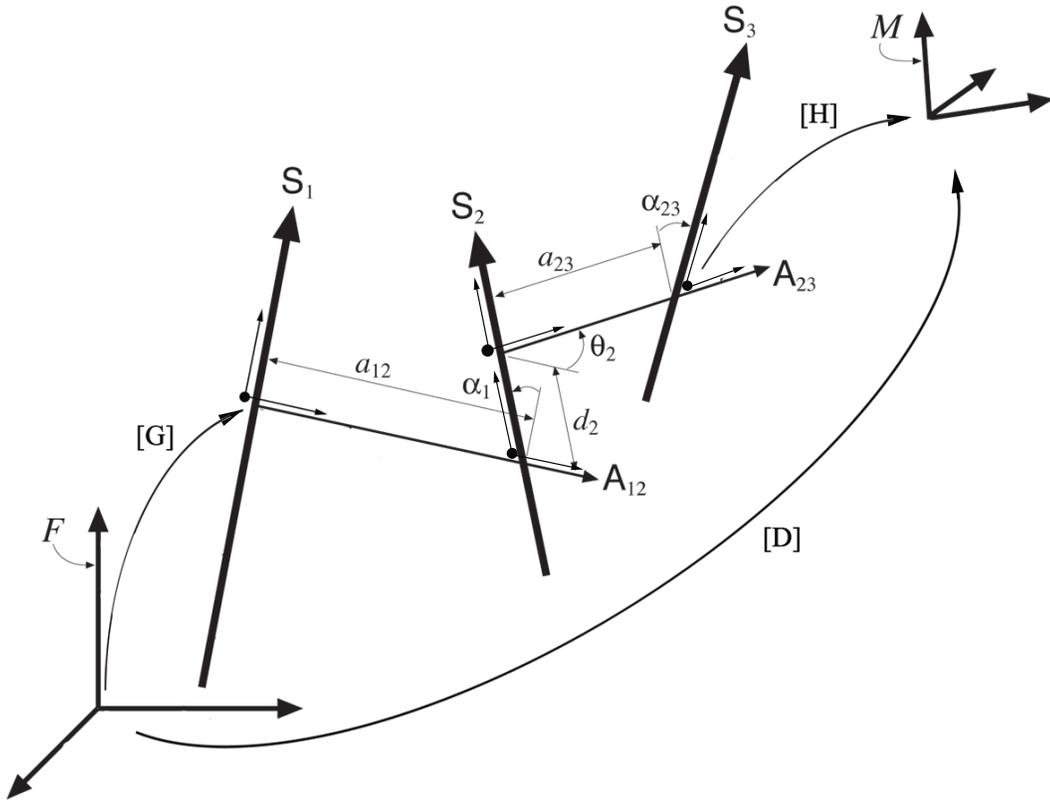


Figure 2.3: A serial chain with its joints[86]

instance, $[Z(\theta_i, d_i)]^{-1} = [Z(-\theta_i, -d_i)]$. The set of all positions reachable by the robot is defined by its kinematic equations as the set of all homogeneous transforms $[D]$

from the base frame to the end-effector frame,

$$[D] = [G][Z(\theta_1, d_1)][X(\alpha_{12}, a_{12})][Z(\theta_2, d_2)] \dots [X(\alpha_{n-1,n}, a_{n-1,n})][Z(\theta_n, d_n)][H] \quad (2.1.10)$$

where the homogeneous transformations $[G]$ and $[H]$ are the coordinate transformations from the base frame to the first joint axis and from the last joint axis to the end-effector frame, respectively. Equation (2.1.10) defines the kinematic equations of the open chain. Which is the representation of the workspace parameterized by the joint variables, (θ_i, d_i) , and the link dimensions, (α_{ij}, a_{ij}) . The workspace can also be described by using relative transformation matrices if we choose a reference position $[D_0]$. The relative motion with respect to a reference configuration $[D_0]$, is given by $[D_{0i}] = [D_i][D_0]^{-1}$.

$$[D_{0i}] = [D_i][D_0]^{-1} = ([G][Z(\theta_{1i}, d_{1i})] \dots [Z(\theta_{ni}, d_{ni})][H])([G][Z(\theta_{10}, d_{10})] \dots [Z(\theta_{n0}, d_{n0})][H])^{-1} \quad (2.1.11)$$

The relative displacement equations take the form

$$[D_{0i}(\Delta\theta_i)] = [T(\Delta\theta_i, S_1)][T(\Delta\theta_2, S_2)] \dots [T(\Delta\theta_n, S_n)] \quad (2.1.12)$$

The displacements $[T(\Delta\theta_i, S_i)]$ are the relative rotations and translations along the joint axes of the robot from the chosen reference configuration.

The matrix exponential form

The kinematic equations can also be expressed in the matrix exponential form [86].

Let the two transform matrices be $[Z(\theta_i, d_i)]$ and $[X(\alpha_{i,i+1}, a_{i,i+1})]$ and the screws defined for these two matrices be $K = (k, vk)$ and $I = (l, \lambda l)$, where $v = d_i/\theta_i$ and $\lambda = a_{i,i+1}/\alpha_{i,i+1}$ are their respective pitches. Thus, they can be related by equation (2.1.13).

$$\begin{aligned} [Z(\theta_i, d_i)] &= e^{\theta_i K} \\ [X(\alpha_{i,i+1}, a_{i,i+1})] &= e^{\alpha_{i,i+1} I} \end{aligned} \quad (2.1.13)$$

The kinematic equation (Equation 2.1.10) becomes

$$[D] = [G] e^{\theta_1 K} e^{\alpha_{12} I} e^{\theta_2 K} \dots e^{\alpha_{n-1,n} I} e^{\theta_n K} [H] \quad (2.1.14)$$

The forward kinematics of relative displacements (with respect to a reference position) can be expressed as the composition of Clifford algebra elements [99]. Let θ_0 and d_0 be the joint parameters at the reference configuration, so we have $\Delta\hat{\theta} = (\theta - \theta_0 + \epsilon(d - d_0))$.

The movement from this reference configuration is defined by

$$\hat{Q}(\hat{\theta}) = e^{\Delta\frac{\hat{\theta}_1}{2} S_1} e^{\Delta\frac{\hat{\theta}_2}{2} S_2} \dots e^{\Delta\frac{\hat{\theta}_n}{2} S_n} \quad (2.1.15)$$

Dual Quaternions

The most common methods to represent rigid body orientation and translations in 3D are Matrices, (Euler-Angles + Translation), (Quaternions + translations). However, recent study shows that dual quaternions are more efficient in terms of interpolation, computational speed, and mathematical robustness, for detail comparison and

introduction see [64]. The workspace of the robot can also be calculated by using the Clifford algebra of the dual quaternions [85]. Dual quaternions capture the basic information contained in the transformation matrix, that is, the axis of rotation and the value of the rotation angle about the axis and the translation along it. For the given screw axis of the transformation $S = s + \epsilon S^0$, where $\epsilon^2 = 0$, s and S^0 are the direction and moments of the axis respectively, the general displacement using dual quaternions is expressed as Equation 2.1.16:

$$\hat{D}(\hat{\theta}) = \begin{pmatrix} \sin(\hat{\theta}/2)S_x \\ \sin(\hat{\theta}/2)S_y \\ \sin(\hat{\theta}/2)S_z \\ \cos(\hat{\theta}/2) \end{pmatrix} \quad (2.1.16)$$

The conjugate form of the dual quaternion (equation 2.1.16) is:

$$\hat{D}^*(\hat{\theta}) = \begin{pmatrix} -\sin(\hat{\theta}/2) S_x \\ -\sin(\hat{\theta}/2) S_y \\ -\sin(\hat{\theta}/2) S_z \\ \cos(\hat{\theta}/2) \end{pmatrix} \quad (2.1.17)$$

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

2.2 Kinematics Synthesis

The synthesis equations for a spatial linkage are obtained from the kinematic equations. If the matrix exponential form is used in the kinematic equation, the unknown design parameters will be the coordinates of the joint axes $S_i, i = 1, \dots, n$.

The exponential of the screw defines a unit dual quaternion, corresponding to a relative displacement from an initial position to a final position in terms of a rotation around and a slide along axis S .

For a serial chain with n joints, with joint parameters $\Delta\hat{\Theta} = (\Theta - \Theta_0 + (\mathbf{d} - \mathbf{d}_0)\epsilon)$ around and along the axis $S_i, i = 1, \dots, k$, the product of exponentials defines the relative workspace from a reference configuration,

$$\hat{D}(\Delta\hat{\Theta}) = \cos \frac{\hat{\psi}}{2} + \sin \frac{\hat{\psi}}{2} S = e^{\frac{\Delta\hat{\theta}_1}{2} S_1} e^{\frac{\Delta\hat{\theta}_2}{2} S_2} \dots e^{\frac{\Delta\hat{\theta}_k}{2} S_k}. \quad (2.2.1)$$

It is immediate to find the screw axis S , magnitude and pitch from this expression,

$$\sin \frac{\hat{\psi}}{2} S = \left(\sin \frac{\psi}{2} + \epsilon \frac{t}{2} \cos \frac{\psi}{2} \right) S = \sin \frac{\psi}{2} \left(1 + \epsilon \frac{t/2}{\tan \frac{\psi}{2}} \right) S, \quad (2.2.2)$$

so that the finite-screw relative workspace is a set of screw axes with magnitude $\sin \frac{\psi}{2}$ and Parkin's pitch $\frac{t/2}{\tan \frac{\psi}{2}}$. The value of the magnitude is unique and can be calculated using the scalar part of the forward kinematics, see [50].

Design Equations for a Serial Chain

For a given set of task positions $([P_j], j = 1, \dots, m)$, the goal is to find the dimensions of the serial chain that can position the end-effector at the given set of task positions. In other words, for each position $[P_j]$, there is at least one joint parameter vector θ_j and a set of structural parameters such that the kinematic equations of the chain satisfy the relations.

$$[P_j] = [D(\theta_j)], i = 1, \dots, m. \quad (2.2.3)$$

Consider $[P_1]$, the first position, as the reference position and compute the relative displacements $[P_j][P_1^{-1}] = [P_{1j}], j = 2, \dots, m$ and express them as the unit dual quaternions, $\hat{P}_{1j} = \cos \frac{\hat{\phi}_{1j}}{2} + \sin \frac{\hat{\phi}_{1j}}{2} P_{1j}, j = 2, \dots, m$. The dual angle $\hat{\phi}_{1j}$ define the rotation about and slide along the axis P_{1j} that defines the the displacement from the first to the j^{th} position. The $(m - 1)$ relative displacements of the serial chain is given by

$$\hat{P}_{1j} = e^{\frac{\Delta \hat{\theta}_{1i}}{2} S_1} e^{\frac{\Delta \hat{\theta}_{2i}}{2} S_2} \dots e^{\frac{\Delta \hat{\theta}_{ni}}{2} S_n}, j = 2, \dots, m. \quad (2.2.4)$$

The problem will have $8(m - 1)$ design equations. The unknowns are the n joint axes $S_i, i = 1, \dots, n$, and the $n(m - 1)$ pairs of joint parameters $\Delta \hat{\theta}_{ij} = \Delta \theta_{ij} + \Delta d_{ij} \epsilon$.

Chapter 3

Methodology for the Design of Exoskeletons

This chapter is focused on developing a systematic approach for the design of exoskeletons that could be used as an instrumentation, rehabilitation device or other uses. The method is comprised of several stages such as vision systems for data collection, Kinematic synthesis, Post-synthesis link based optimization, CAD modeling and prototyping.

3.1 The Design Process

A new methodology is followed for the design of exoskeletons. In this method, the input is the human motion captured using vision-based or similar systems. Finite-position kinematic synthesis yields a set of exoskeletons that can follow the desired motion. Parallel robots are used as input topology due to their higher robustness, high payload, and lower degrees of freedom. Using a link-based hybrid optimiza-

tion routine, the exoskeleton is then further optimized to ensure motion smoothness, obstacle avoidance and optimum physical dimensions. The general design process scheme is shown in Figure 3.1

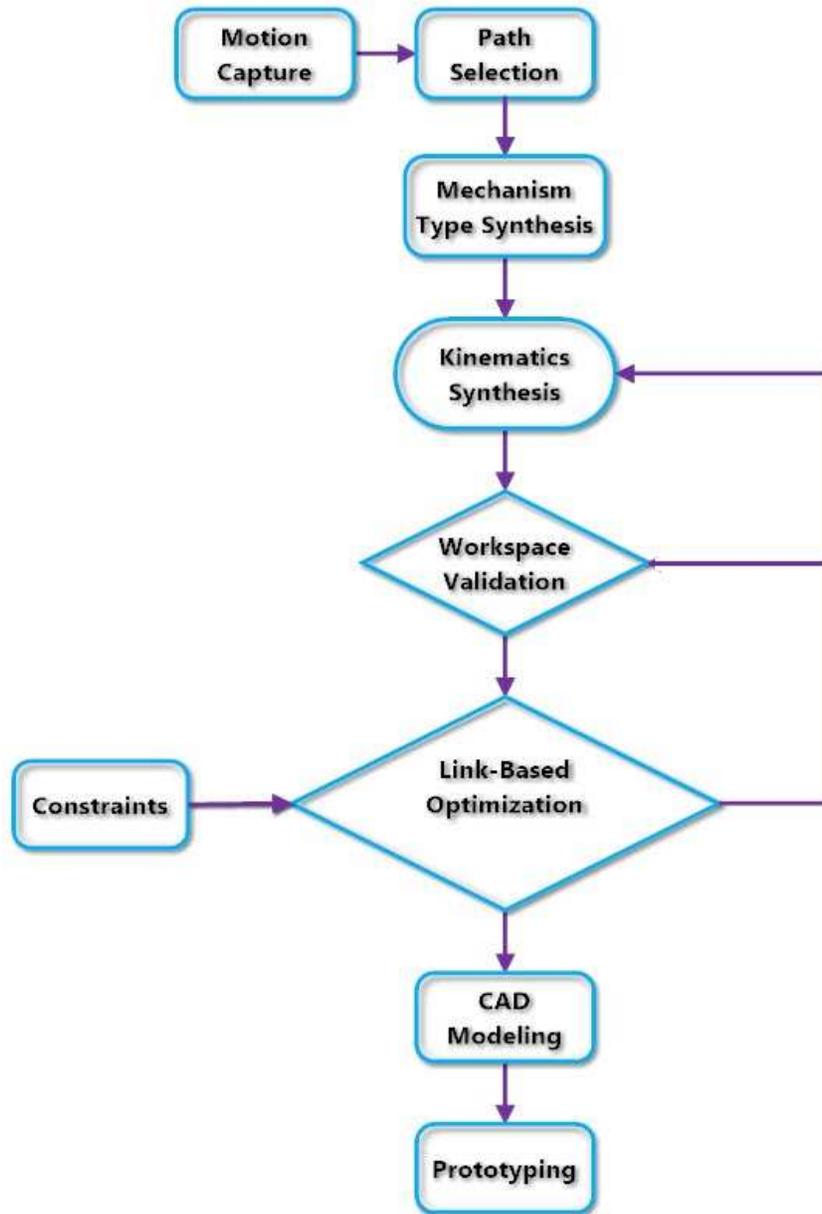


Figure 3.1: The general design methodology

3.2 Human Motion Capture

The desired task has been traditionally specified as a set of finite precision positions that the end-effector of the kinematic chain should pass through [7]. Recent research efforts are directed towards the use of full trajectories [19] or regions of the space that define the workspace. For robotic systems designed for anthropomorphic tasks, the human motion is usually captured with video cameras [4] or infrared technology, as demonstrated in [113]. In this research, the human motion data are acquired using high speed cameras and a Vicon motion tracking system.

The set up for the infrared cameras were set around the room, primarily for larger applications; however it worked very well for the hand motions too. The markers that are used in the system are small white balls that reflect the infrared light. For instance, for one of the thumb exoskeleton design example in this thesis, arrays of markers placed 1.25 inches apart are used, making it easy to collect data in the three dimensions. In order to assess the exact location of the fixed link with respect to the hand, additional sets of sensors are placed on the arm, see Figure 3.2

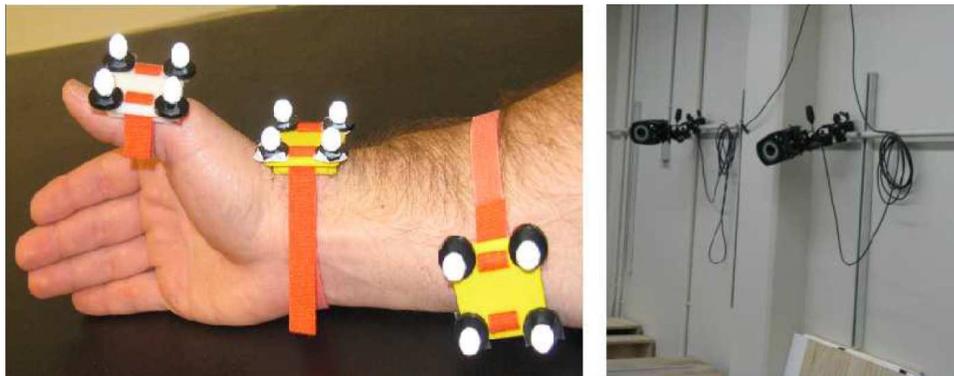


Figure 3.2: Markers placed in the thumb and data capture setup

The data obtained from the markers are used to estimate the pose, which consists

of the position and orientation of the targeted part of the human body. An important aspect of this setup is the geometry of the markers. The geometry of a marker affects directly its performance and usability in computer vision applications. The several experimental paths so obtained can be separated for clarity. The pose, obtained from this stage will be used as an input to the kinematic synthesis, in which the design equations are formulated for the selected linkage to fulfill the desired workspace of the human motion.

3.3 Kinematic Synthesis

Kinematic synthesis of articulated structures is a promising field for the design of complex systems. For instance, the biomechanics design to rehabilitate human hands is usually an important and complex task, especially when the mechanisms attempt to simulate dexterous fingers with multiple joints actuated. The approach here is that, having the finite positions of the tip of the target body using the computer vision system, a kinematics synthesis will be performed so that the new mechanism can mimic the body motion with a less degree of freedom mechanisms. Kinematic synthesis theory allows to create exoskeletons able to perform a desired motion without regard of the anatomic kinematic chain that is producing the motion. It is known that, as the number of joints of a serial kinematic chain increases, the number of real solutions, that is, of possible chains able to perform a similar task, increases very rapidly [100]. This multiplicity of solutions helps in selecting a design that can be made compatible with the anatomy of the user.

Kinematic synthesis is traditionally divided into two steps [42]: structural or type

synthesis, and dimensional synthesis. In type synthesis, the topology of the chain (that is, number and type of joints and connectivity between them) that better fits the task to be performed is selected. The dimensional synthesis step has both the mechanism topology and the desired task as inputs, and yields the dimensions and relative location of the mechanism in order to exactly (exact synthesis) or approximately (approximate synthesis) perform the task.

3.3.1 Type synthesis

The selection of the topology, determines the motion of the robot to a great extent. Systematic approach of type synthesis lead to the invention of new mechanisms. For type synthesis, many researchers were used the Grubler's equation for general mobility and connectivity in the mechanism [55], [30], [81]. However, this general mobility equation do not consider the geometric properties of the mechanisms, as a result, it fails to describe mechanisms in singular configurations in which the mobility is affected by the specific geometrical relations between the joint axes. Current techniques to relate topology to workspace use of group theory, Lie algebra and screw theory [78], [131], [53], [72], [73] See also [37]] yield good results to characterize subgroups of rigid motion. However, these methods cannot be extended to more generally-shaped subsets. In this research, a possible way of selecting the type of the mechanism is addressed but not fully developed. The method is to approximate and compare the workspaces of the required task with the candidate mechanisms.

3.3.2 Dimensional synthesis

In order to follow as closely as possible the whole task trajectory, approximate dimensional synthesis is used. In this case the obtained exoskeleton will not, in general, match the trajectory exactly, but rather the approximation can be controlled by minimizing the distance between the desired task and the trajectory of the exoskeleton. It is well known that there is no Haar measure for spatial displacements, however in the case of an exoskeleton application, we can consider that the fixed frame is anthropocentric in all cases.

Depending on the type of the mechanism i.e, serial chain or parallel, the design equations are formed, in both cases the forward kinematics equations as the set of all positions reachable by the robot is defined by a transformation from the base frame to the end- effector frame. In the parallel robots, in addition to the forward kinematics we may impose loop equations as a constraint to be satisfied.

The position and orientation of the end effector of a robot are defined in terms of its joint parameters and physical dimensions by the kinematic equations. Most researchers use Denavit-Hartenberg formulation to assign the local joint coordinate frames to define the kinematic equations [26], [119]. In this research, instead of the Denavit-Hartenberg parameters we use matrix exponentials [92] to define the 4×4 homogeneous transformations. This kind of formulation gives the coordinates of the joint axes explicitly in the kinematics equations. The joint axes are expressed as lines using the Plucker coordinates. The plucker coordinates of an axis S are given by $S = s + \epsilon S^0$ where $\epsilon^2 = 0$, the first three-dimensional vector, s , is a unit vector

defining the direction of the axis and S^0 is the moment and is obtained as the cross product of a point on the axis, C , and the direction s .

The synthesis equations for a robot are obtained from the kinematic equations. If the matrix exponential form is used in the kinematic equation, the unknown design parameters will be the coordinates of the joint axes $S_i, i = 1, \dots, n$. Equation 3.3.1 defines the workspace of the robot,

$$\hat{D}(\Delta\hat{\Theta}) = e^{\frac{\Delta\hat{\theta}_1}{2}S_1}e^{\frac{\Delta\hat{\theta}_2}{2}S_2} \dots e^{\frac{\Delta\hat{\theta}_k}{2}S_k}. \quad (3.3.1)$$

Thus, for the given set of task positions ($[P_j], j = 1, \dots, m$), the kinematic equations of the chain satisfy the relations.

$$[P_j] = [D(\theta_j)], i = 1, \dots, m. \quad (3.3.2)$$

3.4 Post-Synthesis Optimization and CAD Implementation

In the design of spatial linkages, the finite-position kinematics synthesis gives the position of the joint axes in space. Most of the tasks have additional requirements regarding motion smoothness, obstacle avoidance, force transmission, or physical dimensions, to name a few. Many of these additional performance requirements are fully or partially independent of the kinematic task and can be fulfilled using a link-based optimization after the set of joint axes has been defined.

An initial implementation of the spatial linkage can be obtained by drawing the links at the common normal lines between consecutive axes. Manual adjustment of the links can be performed by using the CAD model. This is done by plotting the screw axes obtained from the synthesis and then modifying the links by sliding the anchor points as shown in Figure 3.3. A simplified method is developed in a CAD environment to facilitate this modification. This procedure can give a mechanism which is out of a constrained region for a given configuration, or it can help reduce the length of some of the links; however, due to the high degree of nonlinearity and difficulty of visualization of spatial linkages, this process is time-consuming and does not grant an optimized solution. In this work it is shown that a better solution may be obtained for some of the requirements if the link-sliding operation as an optimization problem is performed.

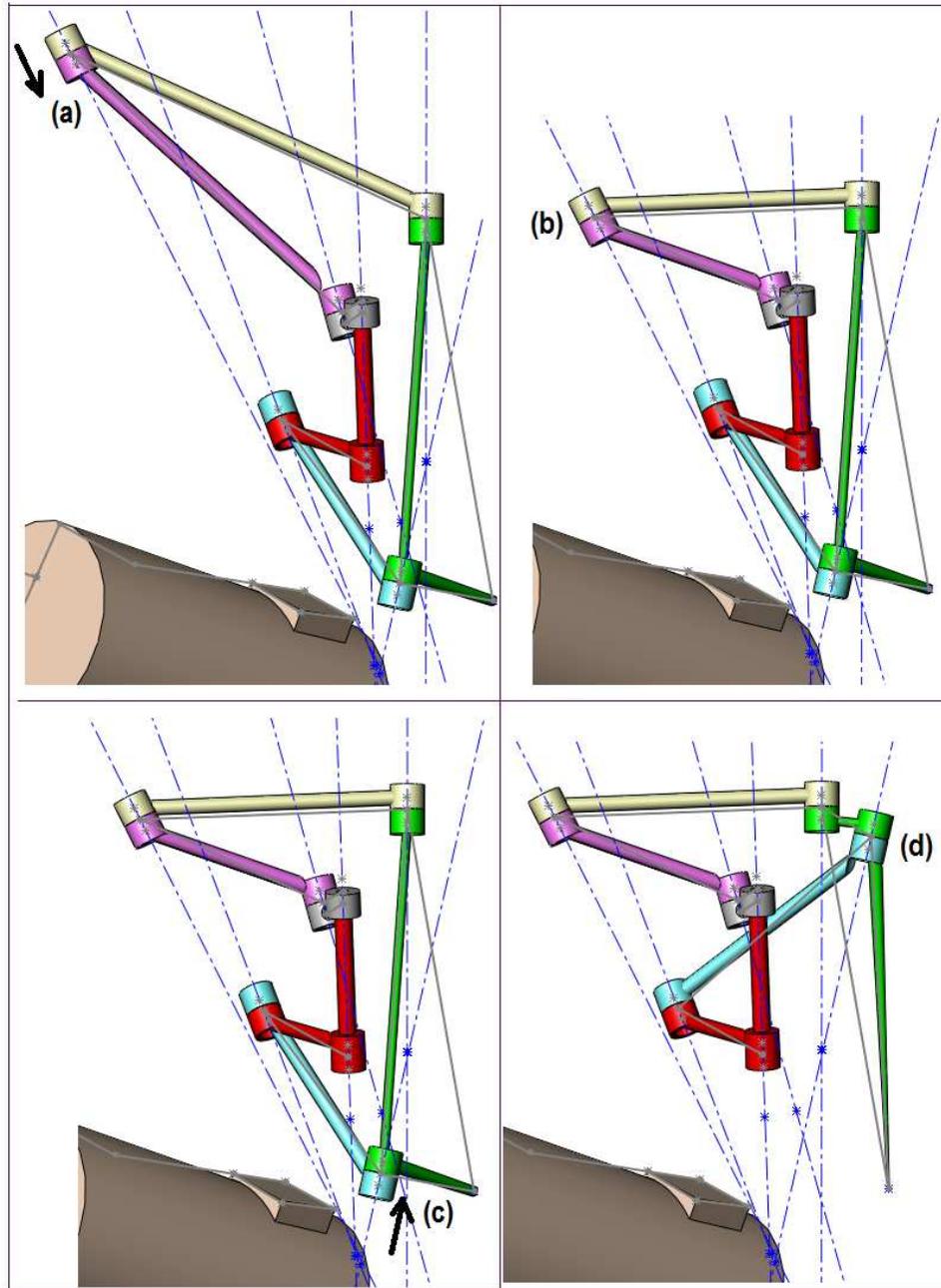


Figure 3.3: CAD model for manual adjustment and optimization; Sliding anchor point from (a) to (b) and from (c) to (d)

In this research, the optimization algorithm is developed and tested to account for different issues that arise in spatial mechanism design. The method is based on considering the links as anchored to sliding points on the set of joint axes, and making the additional requirements be a function of the location of the link relative to the two joints that it connects. The optimization of this function is performed using a hybrid algorithm, including a genetic algorithm (GA) and a gradient-based minimization solver.

The combination of the kinematic synthesis together with the link optimization developed here allows to interactively monitor, control and adjust objectives and constraints, to yield practical solutions to realistic spatial mechanism design problems.

3.5 Exoskeleton Design Examples

Based on the outlined design methodology, a lightweight single-degree-of freedom mechanism has been designed for following the paths of the thumb during simple pinch and grasping movements. The complete exoskeleton device consists of two separate single degree-of-freedom exoskeleton mechanisms: one for finger curling motions, whose design was presented in [126], and one for thumb motions, presented in this thesis (see chapter 7). The resulting robot will be able to assist in common, naturalistic finger and thumb motions. As such, the therapy delivered should translate to a wide range of functional tasks, even if the degrees-of-freedom of the robot are minimal.

The methodology used in the design of exoskeletons for rehabilitation application is extended to design a novel exoskeleton based finger joint angular sensor. To relate the

surface EMG signals to the finger motion, and also for gesture recognition, researchers have been using data gloves and expensive infrared sensors [82]. These devices tend to be expensive and affected by noise; in the case of glove devices, the fitting greatly influences the measurement error [71]. In addition, the accuracy of the result depends on a faithful underlying hand model, which is a complex problem on its own [116]. The exoskeleton based sensor in this thesis presents a simple and cost-effective mechanism for the estimation of the angles of the carpometacarpal (CMC) joint of the index finger. The design strategy includes vision system and image processing coupled with kinematic synthesis techniques. The kinematics synthesis procedure and experimental setups are presented in chapter 6.

Chapter 4

Exact Workspace Synthesis of Spatial Linkages

This chapter presents an exact-workspace synthesis method for spatial mechanisms. Here, the workspace of the linkage is visualized as a set of finite screws corresponding to the set of finite displacements of the end-effector. One interesting question is whether the finite-screw surfaces generated by a set of task positions can give any information for the synthesis of parallel robots and, in particular, of some overconstrained closed linkages. Using Parkin's definition for pitch [96], the screws corresponding to finite displacements of some linkages can form screw systems. Huang [48] showed that the single RR chain forms a finite screw system of third order; however, the set of finite displacements of the coupler of the Bennett linkage form a cylindroid, which is a general 2-system of screws [49]. Baker [12] has also studied the motion of the Bennett linkage. Perez and McCarthy [97] used two arbitrary displacements to generate the cylindroid of finite screws associated to the Bennett linkage in

order to perform dimensional synthesis. In this thesis, the exact workspace synthesis methodology shown at Figure 4.1 is followed.

The methodology is implemented in the synthesis of RCCR linkage, an overconstrained mechanism with mobility one. The implicitization of the algebraic equations of the workspace of relative displacements yields a circular cylinder that can be shaped using a set of finite positions. This simple case, in which the workspace has a constant orientation, is a building block towards a more general methodology for the exact workspace synthesis of spatial linkages. The following subsections present the detail procedure of the RCCR mechanism synthesis.

4.1 Exact Workspace Synthesis for RCCR linkages

The closed RC-CR linkage is overconstrained and able to move with one degree of freedom [124] when the cylindrical (C) and revolute (R) joints of each pair are parallel, while both pairs are skew one to each other, see Figure 4.2.

For arbitrarily-positioned axes, the mobility of this spatial four-bar linkage is, using CKG formula, equal to zero. However it is possible to obtain a one-dof linkage for some special geometry.

The geometric features and the joint variable functions can be derived, for instance, by equating the forward kinematics of both RC serial chains at their end-effector [124]. According to the coordinate frame shown in Figure 4.2, and applying the needed condition of parallel axes, that is, $\alpha_1 = \alpha_3 = 0$, the forward kinematics of

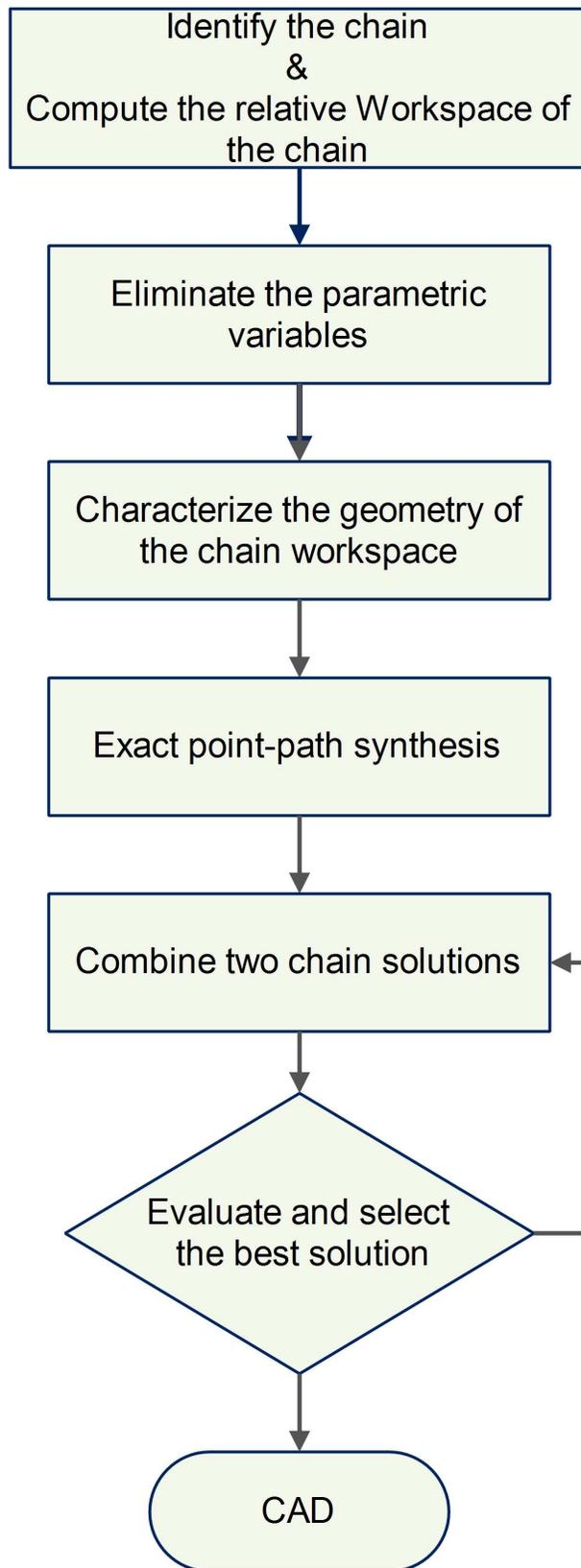


Figure 4.1: Methodology for the exact workspace synthesis

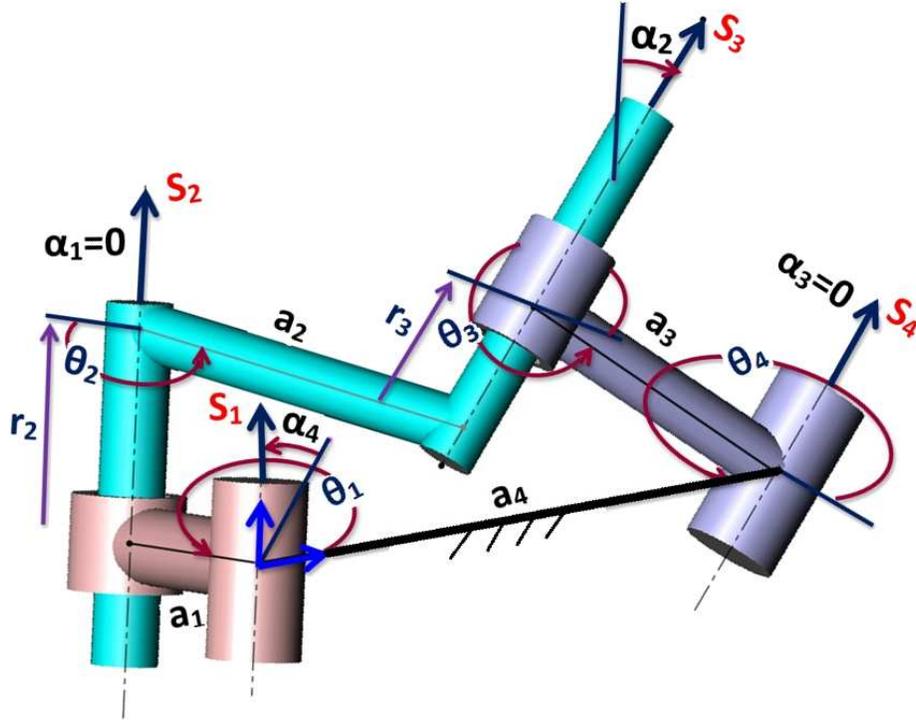


Figure 4.2: RCCR linkage

both RC chains 1-2 and 4-3 are,

$$[D_{RC1}] = \begin{bmatrix} c(\theta_1 + \theta_2) & -s(\theta_1 + \theta_2)c\alpha_2 & s(\theta_1 + \theta_2)s\alpha_2 & a_2c(\theta_1 + \theta_2) + a_1c\theta_1 \\ s(\theta_1 + \theta_2) & c(\theta_1 + \theta_2)c\alpha_2 & -c(\theta_1 + \theta_2)s\alpha_2 & a_2s(\theta_1 + \theta_2) + a_1s\theta_1 \\ 0 & s\alpha_2 & c\alpha_2 & r_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.1.1)$$

$$[D_{RC2}] = \begin{bmatrix} c(\theta_3 + \theta_4) & s(\theta_3 + \theta_4) & 0 & -a_3c\theta_4 - a_4 \\ -s(\theta_3 + \theta_4)c\alpha_4 & c(\theta_3 + \theta_4)c\alpha_4 & s\alpha_4 & -r_3s\alpha_4 + a_3s\theta_4c\alpha_4 \\ s(\theta_3 + \theta_4)s\alpha_4 & -c(\theta_3 + \theta_4)s\alpha_4 & c\alpha_4 & -r_3c\alpha_4 - a_3s\theta_4s\alpha_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.1.2)$$

where s and c stand for the *sin* and *cos* functions respectively.

Some geometrical constraints and angular relations are obtained from equating these two transformations,

$$\begin{aligned}
\cos \alpha_2 = \cos \alpha_4 &\implies \alpha_4 = \pm \alpha_2, \\
\cos(\theta_1 + \theta_2) = \pm 1, \quad \sin(\theta_1 + \theta_2) = 0 &\implies \theta_2 = n * \pi - \theta_1, \\
\cos(\theta_3 + \theta_4) = \pm 1, \quad \sin(\theta_3 + \theta_4) = 0 &\implies \theta_3 = n * \pi - \theta_4,
\end{aligned} \tag{4.1.3}$$

in which the directions of the fixed joints are parallel to the directions of the moving joints, with coupled rotation angles. We can also derive the following joint variable relations:

$$\begin{aligned}
\theta_4 &= \pm \arccos\left(\frac{\pm a_2 - a_4 - a_1 \cos \theta_1}{a_3}\right) \\
r_3 &= \frac{a_1 \sin \theta_1 - a_3 \cos \alpha_4 \sin \theta_4}{\sin \alpha_4}, \quad r_2 = \frac{a_1 \cos \alpha_4 \sin \theta_1 - a_3 \sin \theta_4}{\sin \alpha_4}
\end{aligned} \tag{4.1.4}$$

4.1.1 The Workspace of Finite Displacements of the RC-CR Linkage

We denote the RC chain with parallel axes and angles $\theta_2 = -\theta_1$ a *parallel RC chain*. For solving the design problem, it is advantageous to compute the workspace of relative displacements with respect to a reference configuration. The reference configuration can be arbitrarily selected, with $\Delta r_2 = r_2 - r_{20}$ and $\Delta \theta_i = \theta_i - \theta_{i0}$. The

workspace of relative displacements a parallel RC chain is

$$\begin{aligned}\hat{D} &= \hat{R}(\Delta\theta_1)\hat{C}(\Delta\theta_2, \Delta r_2) \\ &= 1 + \epsilon \frac{1}{2}(\Delta r_2 \mathbf{s}_1 + (\cos \Delta\theta_1 - 1)(\mathbf{c}_2 - \mathbf{c}_1) - \sin \Delta\theta_1(\mathbf{c}_2 - \mathbf{c}_1) \times \mathbf{s}_1),\end{aligned}\quad (4.1.5)$$

where $\hat{R}(\Delta\theta_1)$ is a rotation about an axis with Plucker coordinates \mathbf{S}_1 , and $\hat{C}(\Delta\theta_2, \Delta r_2)$ is a rotation and a translation about and along an axis with Plucker coordinates \mathbf{S}_2 . Both axes share the same direction \mathbf{s}_1 and their rotations are $\Delta\theta_1$ and $\Delta\theta_2 = -\Delta\theta_1$; the points \mathbf{c}_1 and \mathbf{c}_2 are any points on the axes along a common normal line. Notice that the relative displacements have no change in orientation, so that the chain has a constant-orientation workspace.

The workspace of relative translations for the RCCR linkage is given by the intersection of the workspaces of two parallel RC chains. Figure 4.3 shows the workspace of a parallel RC chain and the intersection workspace for two chains.

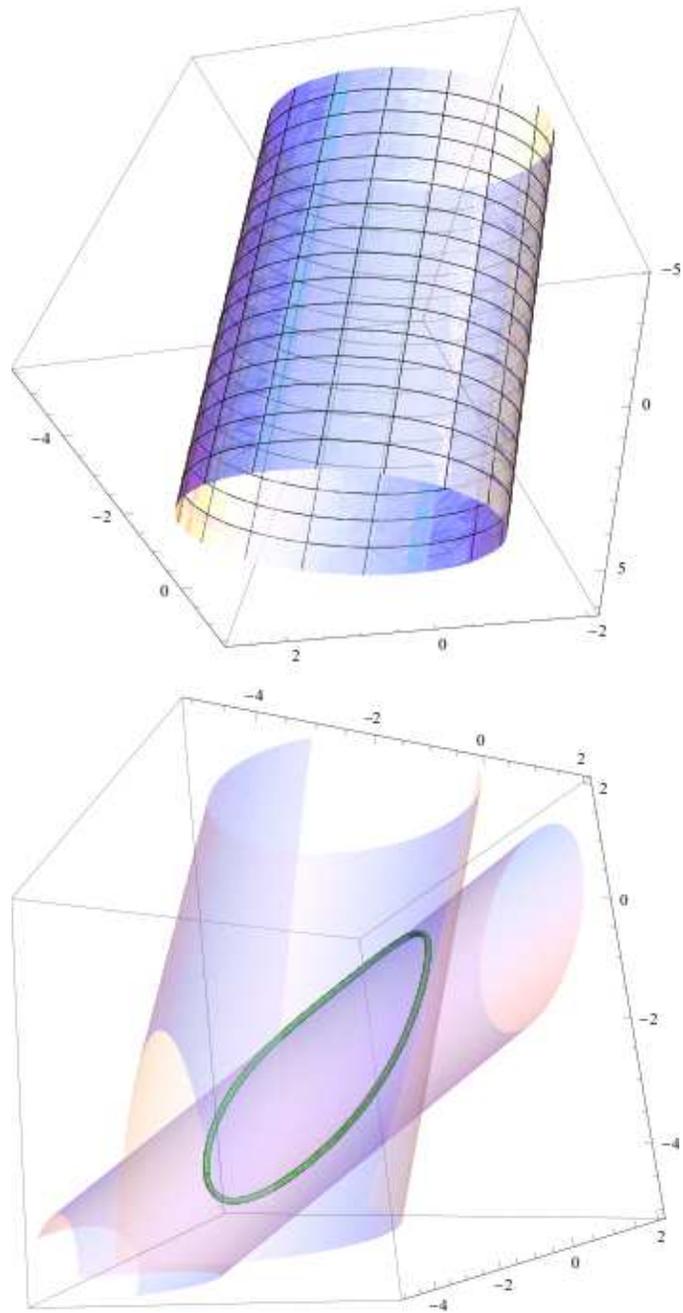


Figure 4.3: Workspace of relative displacements for the parallel RC chain, left; for the RC-CR linkage, right.

In order to characterize the workspace of the parallel RC chain, we perform implicitization in Eq.(4.1.5) to eliminate the joint variables θ_1 and r_2 . The elimination yields a quadratic surface of expression

$$\begin{aligned} \mathcal{Q}(x, y, z): \quad & (s_{1y}^2 + s_{1z}^2)x^2 + (s_{1x}^2 + s_{1z}^2)y^2 + (s_{1x}^2 + s_{1y}^2)z^2 \\ & - 2s_{1x}s_{1y}xy - 2s_{1x}s_{1z}xz - 2s_{1y}s_{1z}yz + c_{21x}x + c_{21y}y + c_{21z}z = 0, \end{aligned} \tag{4.1.6}$$

where (x, y, z) is a point of the \mathbb{R}^3 space of relative translations, $\mathbf{s}_1 = (s_{1x}, s_{1y}, s_{1z})$ is the direction for both joints, and $\mathbf{c}_{21} = \mathbf{c}_2 - \mathbf{c}_1 = (c_{21x}, c_{21y}, c_{21z})$ is the vector along the common normal between both joints.

This surface is classified as a circular cylinder, with radius $R = \sqrt{\mathbf{c}_{21} \cdot \mathbf{c}_{21}}$ and passing through the origin, which corresponds to the zero relative displacement. The intersection of two such circular cylinders yields a quartic curve which is the workspace of the RCCR linkage.

4.1.2 Dimensional Synthesis for the RCCR Linkage

The workspace of the RCCR linkage is a constant-orientation curve, and hence the synthesis problem can be reduced to a point-path synthesis problem. The point-path synthesis problem is stated as follows: given an initial point \mathbf{P}_1 (which we will use as reference configuration), relative displacements of the RC-CR chain will move this point to the rest of task points $\mathbf{P}_2, \mathbf{P}_3, \dots, \mathbf{P}_n$.

The action of the chain on this point can be calculated using one of the conjugations in the Clifford algebra. If the forward kinematics of relative displacements of

Eq.(4.1.5) is denoted by $\hat{D} = 1 + \epsilon \mathbf{d}$, then

$$\mathbf{P}_i = \hat{D} \hat{P}_1 \hat{D}^*, \quad i = 2, \dots, n, \quad (4.1.7)$$

where $\hat{P}_1 = (1 + \epsilon \mathbf{P}_1)$ is the dual quaternion expression of the point \mathbf{P}_1 , and the conjugation yields

$$(1 + \epsilon \frac{1}{2} \mathbf{d})(1 + \epsilon \mathbf{P}_1)(1 + \epsilon \frac{1}{2} \mathbf{d}) = 1 + \epsilon(\mathbf{P}_1 + \mathbf{d}) \quad (4.1.8)$$

Notice that this is equivalent to equating the relative translations, $\mathbf{d} = \mathbf{P}_i - \mathbf{P}_1$ for $i = 2, \dots, n$.

Let us consider the case of the parallel RC chain, in which the values of θ_1 and r_2 are independent. This results in $3(n - 1)$ design equations, with the structural variables \mathbf{s}_1 and $\mathbf{c}_{21} = \mathbf{c}_2 - \mathbf{c}_1$ and the joint variables r_2 and θ_1 for each point, for a total of $4 + 2(n - 1)$ unknowns. Up to $n = 5$ point-positions can be defined in order to do exact point-path synthesis.

The standard finite-position synthesis technique equates the parameterized expression of the translation workspace to the task relative translations. However in this case, the implicit equation for the workspace has a simpler expression as a function of the chain structural parameters.

The four relative translations \mathbf{P}_i , $i = 2, \dots, 5$, are used to shape the circular cylinder, and define the parallel RC chain that creates the motion. The system of

design equations consists of six quadratic equations in six unknowns,

$$\begin{aligned} \mathcal{Q}(\mathbf{P}_i) &= 0, \quad i = 2, 3, 4, 5; \\ \mathbf{s}_1 \cdot \mathbf{s}_1 &= 1, \mathbf{s}_1 \cdot \mathbf{c}_{21} = 0, \end{aligned} \tag{4.1.9}$$

which are easy enough to be solved using algebraic techniques. There are at most 6 different solutions.

4.2 Example and Results

The task points used in this example are presented in Table 4.1. The system of equations (4.1.9) yields four real solutions, presented in Table 4.1, which can be assembled in pairs in order to create RCCR linkages. The number of different workspaces obtained are six.

Table 4.1: Goal points and solution RC chains

Point	Coordinates	
P_1	(2.31, 3.84, -1.08)	
P_2	(0.34, -2.81, 0.89)	
P_3	(2.21, -3.47, 0.63)	
P_4	(2.18, 3.77, -2.66)	
P_5	(-1.22, -1.42, -2.22)	

Solution	\mathbf{s}_1	$\mathbf{c}_2 - \mathbf{c}_1$
1	(-0.09, -0.04, -0.99)	(0.03, 7.38, -0.30)
2	(-0.54, 0.42, -0.73)	(3.86, 5.19, 0.15)
3	(-0.10, 0.99, -0.08)	(4.11, 0.51, 1.22)
4	(0.54, 0.84, -0.03)	(-4.32, 2.85, 1.82)

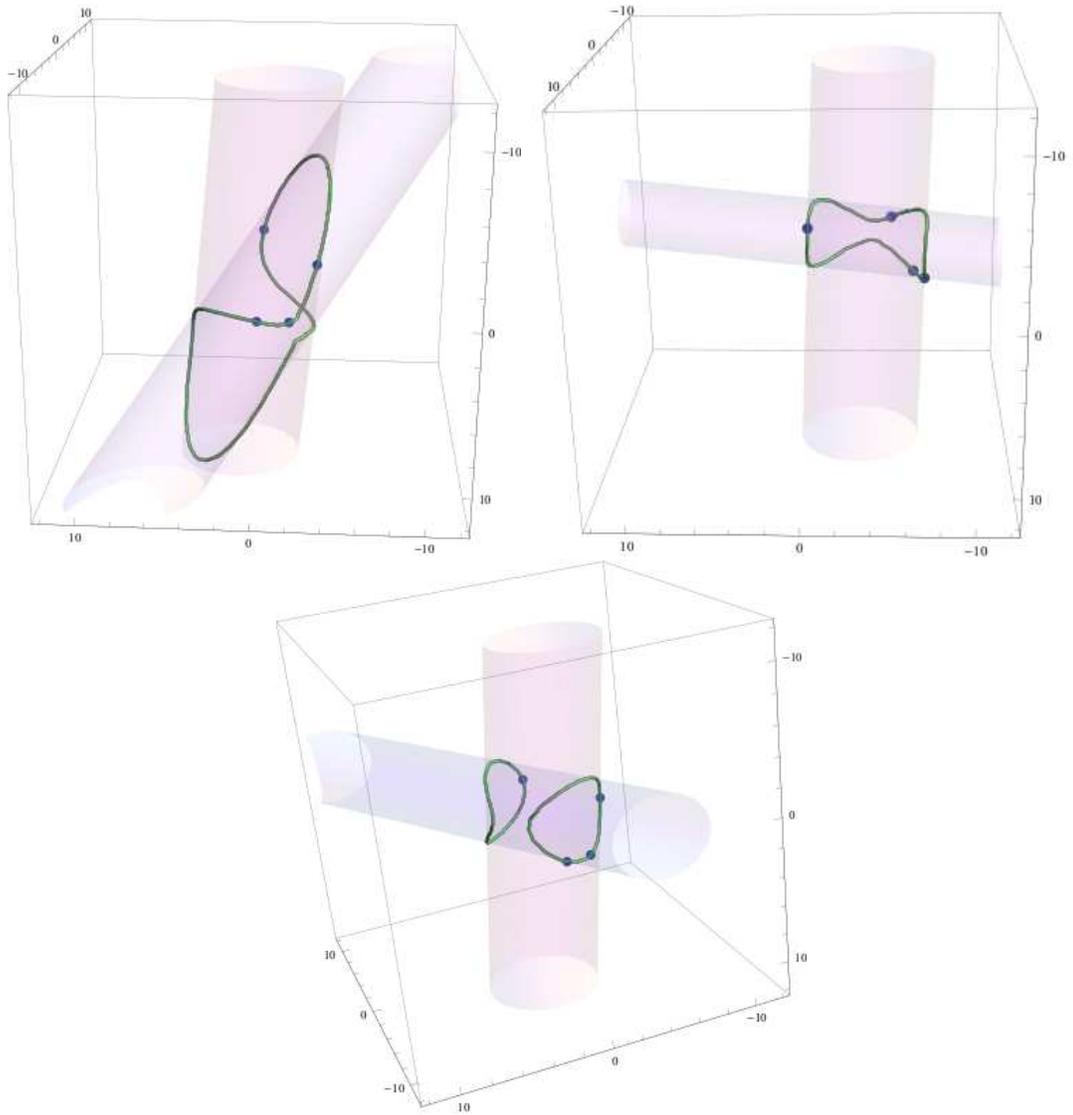


Figure 4.4: Three of the six RCCR workspaces with the task points

The solution workspaces for the parallel RC chains can also be intersected pairwise in order to create workspaces for the RCCR chains. The workspace equations can be used to visually assess the trajectory of the linkage and also to check for circuit defect. Figure 4.4 shows three of the six possible combinations for this example, the figure shows that all the four points lie on the intersection of the relative workspace of each chain, that is all the points lie on intersection of the two cylinders.

4.2.1 The absolute displacements workspace

A similar procedure is followed to characterize the workspace of the absolute workspace and it is found to be an elliptic cylinder. The reference configuration to reach point 1 is solved with arbitrary orientation. In that transformation, S is the direction and $S0$ is the moment of the screw axis of the displacement. In order to do so, there are some of the variables that will be specified, and some that will be arbitrary. Basically we have specified the origin of the end-effector frame, while the rotation is arbitrary and can be pre-selected. In any case, we are equating displacements, not points. As the change in orientation is zero, the dual part of the displacement is going to be the point/2. Figure 4.5 shows that the workspace of the absolute transformation of the RC chain, which is an elliptic cylinder and all the five points lie on the elliptic cylinder. Figure 4.6 shows the intersection of the two elliptical cylinders, and the points

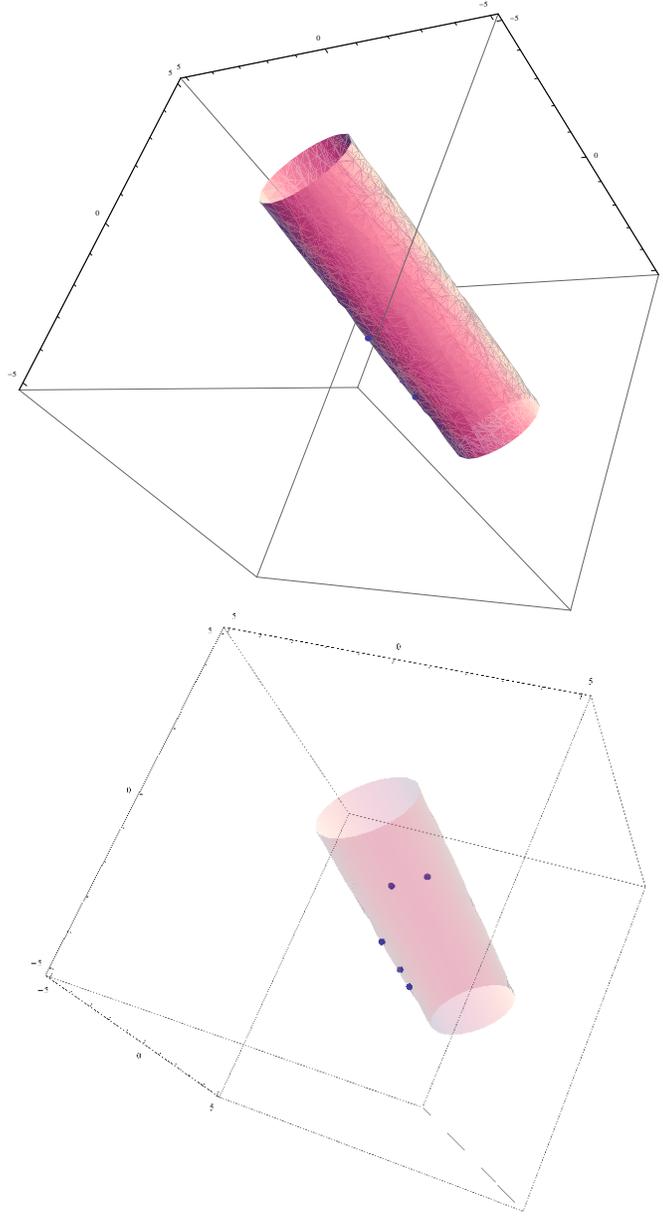


Figure 4.5: The absolute work space of the RC chain with points on the surface

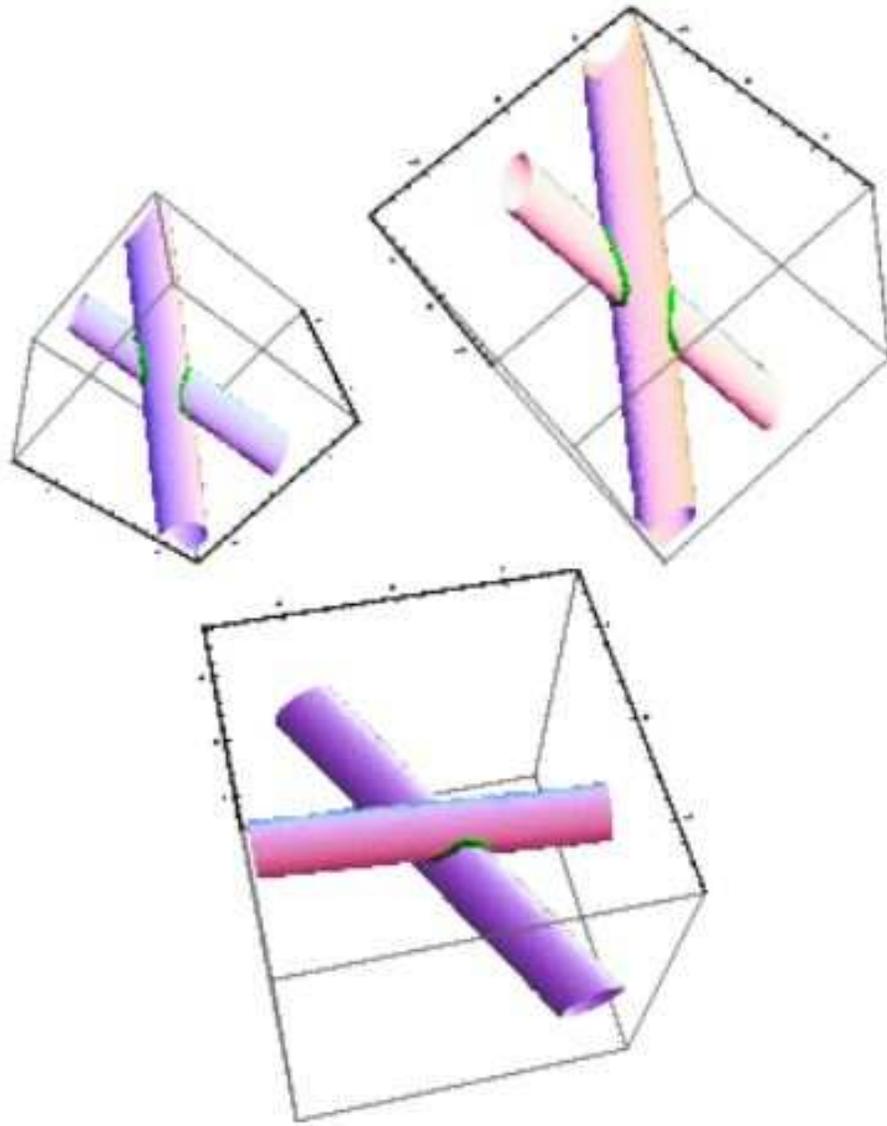


Figure 4.6: The absolute work space of the RCRC

Figure 4.7 and 4.8 show the CAD model of two exact workspace solutions of RCCR spatial mechanism. Out of these solutions, it is clear that in Figure 4.7, there is a circuit defect; that is the mechanism require different configuration to reach all the desired points.

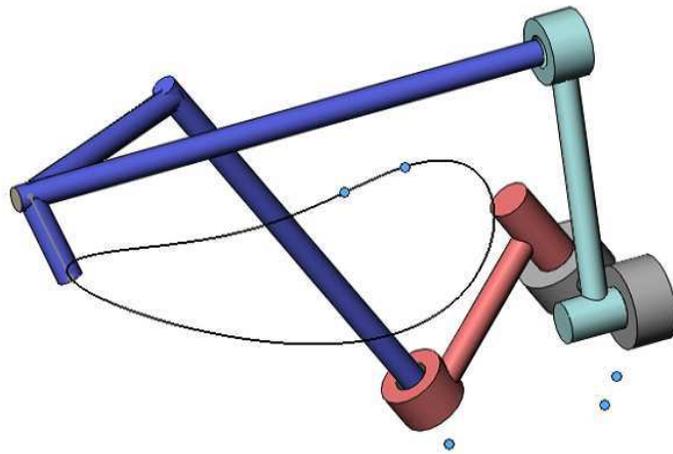


Figure 4.7: RCCR linkage created with chains 2 and 3, showing one of its circuits.

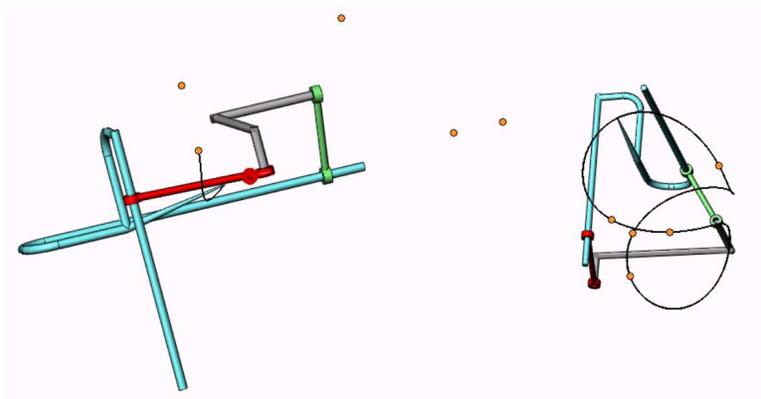


Figure 4.8: RCCR linkage created with chains 2 and 4, showing one of its circuits.

The following section addresses the preliminary results approaches of the RR-RR chain spatial mechanism.

4.3 Workspace of RRRR linkage

The mobility of this spatial four-bar linkage using CKG formula is equal to negative two, however with spatial positioning of the axis this mechanism is able to move with one degree of freedom. The workspace of the two RR chains of the mechanism are studied separately and the intersection of these two chains give us the whole workspace of the RRRR spatial mechanism.

4.3.1 Workspace of the RR chain

The RR chain is a two degree of freedom mechanism consist of two revolute axes, let these axes be S_1 and S_2 . The rotation about S_1 and S_2 are θ_1 and θ_2 respectively. The quaternion representation for the relative displacements of the chain given by the composition of displacements is given by 4.3.1

$$Q_{RR} = S1(\theta1)S2(\theta2) \tag{4.3.1}$$

In this synthesis we are focused on the point or translation workspace. The point workspace, which is parameterized by joint variables θ_1 and θ_2 for a randomly generated axis looks like Figure 4.9. It looks like a twisted hyperbolic paraboloid.

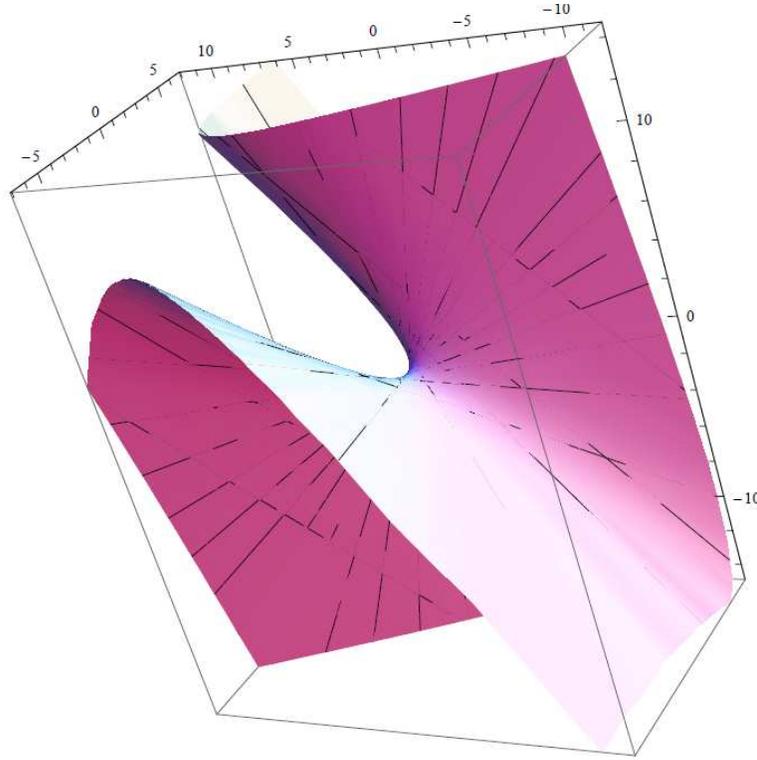


Figure 4.9: Plot of the workspace of relative translations of a randomly-generated RR chain

4.3.2 Implicitization of the relative transformation of the RR chain

In order to assess whether the parametric equation of the RR chain is a hyperbolic paraboloid or any other shape in general, we perform implicitization using Grobner bases to eliminate the parametric variables. The equations are created in 3D space $X = [x, y, z]'$ that is X is equal to the relative forward kinematics of the chain. Then the sin and cosine terms are converted in to their rational expressions. Once the parameters are homogenized, elimination is applied to get rid of other terms and to obtain the desired equation as a function of x, y and z . The expression of the implicitized version of the relative workspace obtained through the above procedure

is too long to put it here. The implicitized equation confirmed that is a quadratic surface. The independent term in this quadratic surface is zero, indicating that zero belongs to the quadric. This is correct because the zero relative displacement is always a point on the quadric. According to the counting, this quadric needs 8 points, which means there is only one relation among its coefficients. To classify the surface, the classification scheme at [125] is followed. The RR chain with general axes and angles requires $n = 9$ absolute, $n = 8$ relative points for the point-path synthesis, therefore we have equal number of equations as the number of unknowns, which we can solve it easily in any numerical solver.

4.4 Summary

The exact workspace synthesis method helps to visually assess the trajectory of the linkage and to inspect circuit defect along the trajectory. However, the method is computationally expensive to find the algebraic solution for some mechanisms. The computation becomes more challenging for higher DOF mechanisms.

Chapter 5

Link-based performance

optimization of spatial mechanisms

5.1 Introduction

The finite-position kinematic synthesis methods yield a linkage able to reach a set of specified positions. In the case of planar linkages, the links are usually located in parallel planes perpendicular to the joints, which defines their geometry to a great extent. In the case of spatial mechanisms with general relative position between joints, such as those obtained from spatial kinematic synthesis, the geometry and properties of the links can be greatly modified by just sliding the joint location along the joint axis. Notice that this operation does not modify the trajectory of the linkage.

The actual geometry and location of the links have important consequences for the performance of the linkage, including but not limited to linkage size and occupied space, self-intersections and obstacle avoidance, friction at the joints and other force

transmission issues.

In most literatures, the optimization is performed to account for additional requirements and to perform approximate synthesis simultaneously. This approach has only been applied to simplified geometries, such as symmetric or planar mechanisms; the additional requirements are mostly joint-related. No general synthesis plus link optimization method exists to our knowledge, possibly due to the very high complexity of the resulting system of equations. We claim that an optimization method for general spatial mechanisms to treat the synthesis and additional requirements independently is a good solution, both from the computational and from the user interaction and assessment point of view.

In our design methodology, the design process is divided into two stages. The first stage uses kinematic synthesis in order to create an articulated system able to follow a specified motion. The second stage, which is the focus of this chapter, deals with the optimization of the links to satisfy a set of performance requirements. The optimization is performed using a genetic algorithm (GA) together with gradient-based minimization. The GA creates a grid of iterative points and keeps only those under a certain objective function value with respect to the previous iteration, then the output from the GA is used as an input for the gradient-based minimization to get into a global minimum point. The optimization approach is illustrated on three examples: a spatial, one-degree-of-freedom CRR-RRR closed linkage, a Bennett linkage, and a three-fingered robotic hand.

The results show that the modification of the links along the joints leads to dramatic changes in the final design and performance of spatial mechanisms.

5.2 Link-Based Optimization

The input for the link-based optimization process is taken from the output of a previous kinematic synthesis process. The synthesis step yields a set of structural parameters that can be, depending on the synthesis methodology used, a set of points or vectors defining the axes if loop equations or geometric constraints are used [93], Denavit-Hartenberg parameters [94], [?] defining relative position between joint axes if forward kinematics equations are used, etc. In any case, those design parameters can be used to compute the Plücker coordinates of the joint axes at a given configuration. The selection of reference configuration is arbitrary; usually the first task position is used to define it. For instance, for an n -jointed linkage, the input data for the optimization stage is the set of joint axes

$$\{\mathbf{S}_i = \mathbf{s}_i + \epsilon \mathbf{s}_i^0\}, \quad i = 1, \dots, n, \quad (5.2.1)$$

where \mathbf{s}_i is the unit direction vector for axis \mathbf{S}_i , \mathbf{s}_i^0 is the moment of the axis, obtained as the cross product of a point on the axis and the direction \mathbf{s}_i , and ϵ is the dual unit such that $\epsilon^2 = 0$. For some methods to compute the Plücker coordinates of the joint axes from other structural data, see for instance [34].

5.2.1 Optimization problem formulation

In the optimization, the variables considered are the sliding parameters that define the anchoring points of the links on the joints. An objective function and additional constraints that can be written as a function of these sliding parameters in order

to fulfill a set of requirements are considered. Figure 5.1 shows how the optimization process fits on an overall design strategy for spatial mechanisms. An objective function is defined as the first step in the optimization process. The selection of the objective function depends on the requirements of the design, and the only condition is for it to be a continuous function of the sliding parameters. Some of the possible objective functions include minimizing the overall length of the mechanism, which may help decreasing material usage, weight, inertia and overall space occupied by the mechanism; minimize friction at the joints, which helps the overall efficiency of the linkage, etc.

The sliding parameter t_{ij} is defined on joint axis \mathbf{S}_i and it defines the anchor point of link ij along the joint axis. The formulation used to create the objective function and additional constraints using the sliding parameters is as follows: let C_i be the point on the joint axis \mathbf{S}_i at which the line from the origin intersects the joint axis \mathbf{S}_i at a right angle, see Figure 5.2. Let P_{ij} be a point on the joint axis \mathbf{S}_i as defined in Figure 5.2, and t_{ij} the corresponding distance to P_{ij} from the point C_i along the direction of the axis. For a similar formulation used to detect self-intersection, see [65]. Point C_i can be obtained using

$$\mathbf{C}_i = \frac{\mathbf{s}_i \times \mathbf{s}_i^0}{\mathbf{s}_i \cdot \mathbf{s}_i}. \quad (5.2.2)$$

Then P_{ij} is expressed with respect to C_i as follows,

$$\mathbf{P}_{ij} = \mathbf{C}_i + t_{ij}\mathbf{s}_i \quad i = 1, \dots, n. \quad (5.2.3)$$

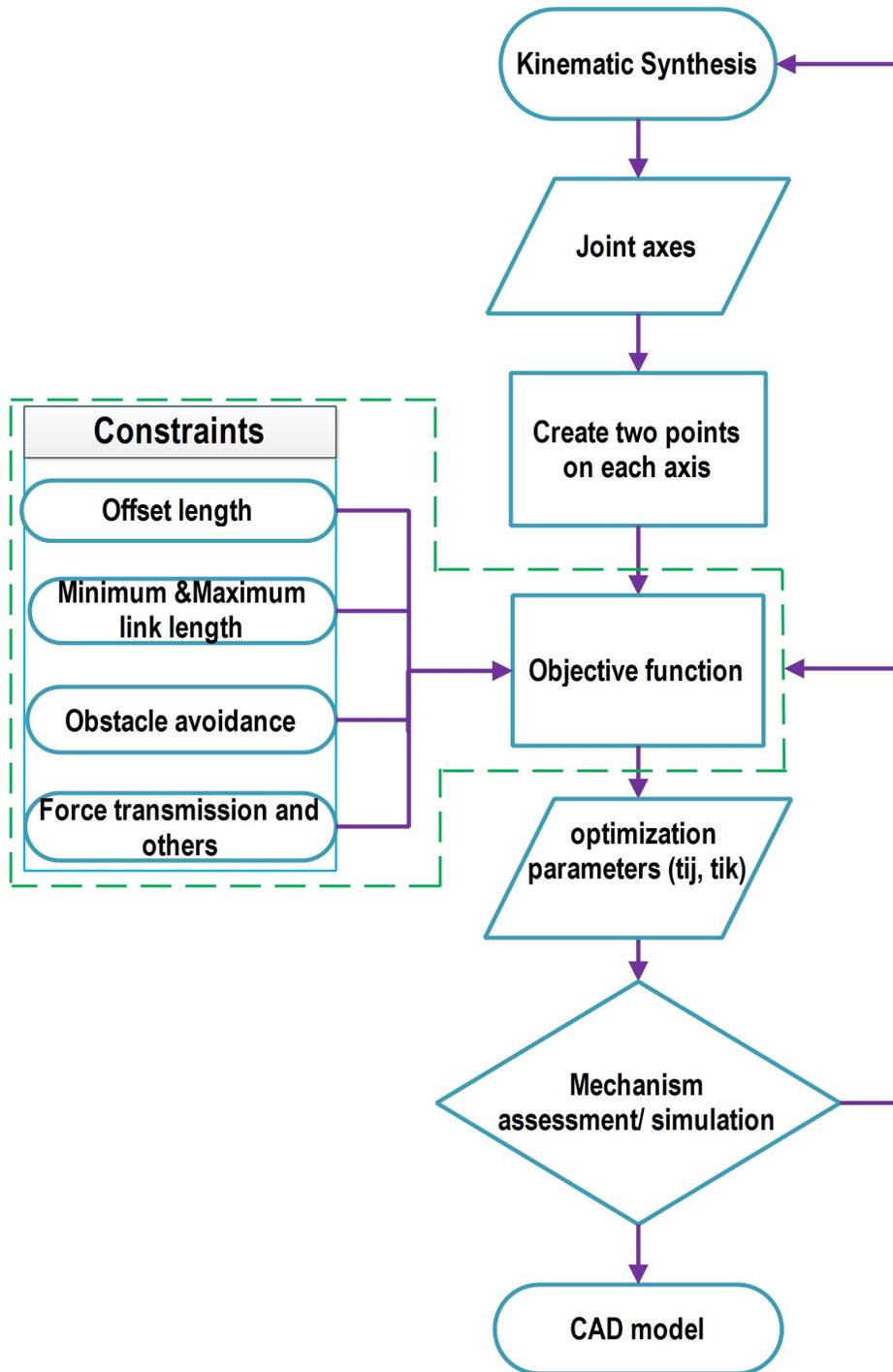


Figure 5.1: Overall design strategy. Link-based optimization stages are shown inside the broken lines

Notice that in P_{ij} , the subscript i indicates that the point lies on joint axis S_i , while the second subscript j indicates that it also belongs to the line linking joint

axis S_i with joint axis S_j . For a single-loop closed mechanism, two points need to be defined on each axis in order to specify all links, see Figure 5.2 (a). For a serial robot, two points are defined in each of the axes except for the first axis. If the distance to the reference frame is a parameter of interest, the first point C_1 could be also included in the optimization as P_{10} .

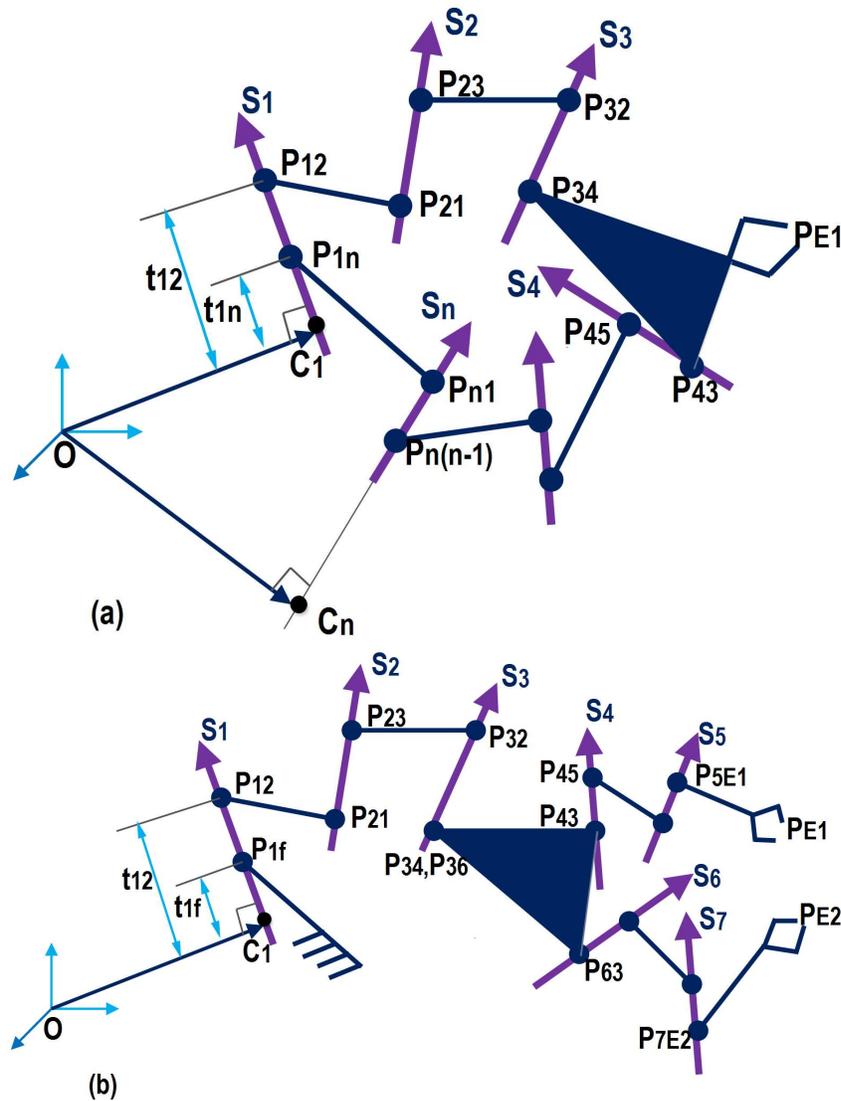


Figure 5.2: Joint axes for two spatial mechanism topologies:(a) A closed linkage, CRR-RRR; (b) A linkage with a tree structure, 3R-(2R,2R)

The objective function is formulated to minimize the overall length by considering the offset distances along the joint axes, the lengths of the linkage between each pair of connected axes, as well as the length of the links to the end effectors. For example for a closed, n -jointed mechanism, the objective function is a quadratic function with $2n$ variables (the scalar slides t_{ij}). A mechanism can have multiple end effectors, like the one in Figure 5.2 (b); in that case, distances to each of the end effectors are also included in the objective function. For instance, equation (5.2.4) is an objective function with m number of of end-effectors, where point P_{Dj} is one point in the end-effector D and point P_{Dj} is a point on axis j , and the line joining both points defines the last link on that branch. Assuming that the joints are numbered consecutively, the objective function is stated as

$$\begin{aligned}
F = & \left(\sum_{i=1(\text{ mod } n)}^n ((\mathbf{P}_{i,i+1} - \mathbf{P}_{i,i-1}) \cdot (\mathbf{P}_{i,i+1} - \mathbf{P}_{i,i-1})) \right. \\
& + (\mathbf{P}_{i+1,i} - \mathbf{P}_{i,i+1}) \cdot (\mathbf{P}_{i+1,i} - \mathbf{P}_{i,i+1}) \\
& \left. + \sum_{j=1}^m (\mathbf{P}_{Dj} - \mathbf{P}_{jD}) \cdot (\mathbf{P}_{Dj} - \mathbf{P}_{jD}) \right) \tag{5.2.4}
\end{aligned}$$

Using this notation, we consider S_0 to be the joint “previous to the first one”. For serial chains, that defines a ground link from the reference frame, while for closed linkages, the last joint axis is the one denoted as S_0 , creating a ground link between the last and first joint.

Another possible objective function can be formulated as shown in Equation (5.2.5), used to optimize the efficiency by minimizing friction for a linkage with rev-

olute joints by imposing a certain angle of incidence of the links on the joints.

$$F = \left(\sum_{i=1(\text{ mod } n)}^n \left(\frac{(\mathbf{P}_{ji} - \mathbf{P}_{ij})}{\|\mathbf{P}_{ji} - \mathbf{P}_{ij}\|} \cdot \mathbf{s}_j \right)^2 \right) \quad (5.2.5)$$

Several requirements could be optimized simultaneously by using multi-objective optimization [134],[44], however in our case adding additional constraints to create a constrained optimization solves the problem efficiently. The derivation of additional constraints is shown in next section.

5.2.2 Additional Constraint Functions

Depending on the application, the design of a mechanism may need to simultaneously meet various geometrical constraints and manufacturing criteria. The link-based optimization algorithm developed in this research allows the definition of a number of linear and nonlinear constraints to satisfy link dimensions, obstacle avoidance, reduction of friction loads and manufacturability constraints. In this section we show how to formulate the constraints as a function of the sliding parameters so that the change of the link location and dimension allows satisfying those criteria.

5.2.2.1 Link length and joint offset constraints

For many practical cases, link size constraints are required [44]. In the case of offset length, the distance between P_{ij} and P_{ik} is set to be greater than or equal to a constant value d , in order to help in the implementation of joints for ease of assembly.

This adds n equality or inequality constraints,

$$\sqrt{(\mathbf{P}_{ik} - \mathbf{P}_{ij}) \cdot (\mathbf{P}_{ik} - \mathbf{P}_{ij})} \geq d, \quad i = 1, \dots, n, \quad (5.2.6)$$

Which are linear in the slide parameters, yielding

$$t_{ik} - t_{ij} \geq d, \quad i = 1, \dots, n. \quad (5.2.7)$$

Similarly, a minimum and a maximum link size may be required for manufacturability and compactness. This is accomplished by setting the values l_{min} and l_{max} for the minimum and maximum link lengths respectively, which add $2n$ additional inequality constraints,

$$l_{min}^2 \leq (\mathbf{P}_{ij} - \mathbf{P}_{ji}) \cdot (\mathbf{P}_{ij} - \mathbf{P}_{ji}) \leq l_{max}^2, \quad i = 1, \dots, n. \quad (5.2.8)$$

These equations are quadratic in the link slide parameters t_{ij} .

5.2.2.2 Constraints for obstacle avoidance

Another common requirement is for the mechanism to avoid a certain region of space during its motion, or to establish a certain relation of closeness/separation with respect to a certain region. The links are modeled as cylinders, with radius R_{il} . The obstacle region can be modeled or represented by different geometrical shapes, for instance as a cylinder with radius R_c and axis of the cylinder pass through points P_{c1}

and P_{c2} as shown in the Figure 5.3; as a sphere with radius R_s and its center at point S_c (Figure 5.4); as a plane or set of planes, etc.

Within the minimization algorithm, this problem has been stated as a set of constraint functions, defined in such a way that the linkages of the mechanism must stay out of the constrained region (cylinder, sphere etc) or maintain a certain distance with respect to the region.

Cylinder constraint equations

Considering a point P_{ij} on axis i and belonging to link that joins axis j with axis i . The axes of the cylinder and the links are expressed using dual vectors as L_c and L_i ,

$$\begin{aligned} L_c &= \mathbf{s}_c + \epsilon(\mathbf{P}_{c1} \times \mathbf{s}_c) \\ L_i &= \mathbf{s}_{li} + \epsilon(P_{ij} \times \mathbf{s}_{li}) \quad i = 1, \dots, n, \end{aligned} \quad (5.2.9)$$

where \mathbf{s}_c and \mathbf{s}_{li} are the direction vectors for the cylinder axis and for the link respectively, and they can be found using Equation (5.2.10),

$$\begin{aligned} \mathbf{s}_c &= \frac{(\mathbf{P}_{c2} - \mathbf{P}_{c1})}{\|\mathbf{P}_{c2} - \mathbf{P}_{c1}\|} \\ \mathbf{s}_{li} &= \frac{(\mathbf{P}_{ji} - \mathbf{P}_{ij})}{\|\mathbf{P}_{ji} - \mathbf{P}_{ij}\|} \end{aligned} \quad (5.2.10)$$

The expression for the distance between L_c and L_i along the common normal line, a_i , can be found from the dual dot product of the lines L_c and L_i as shown in

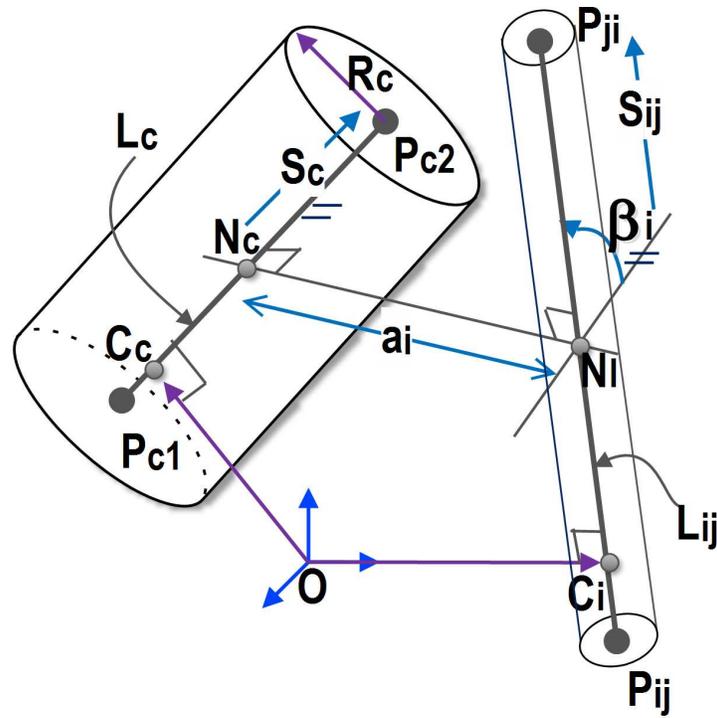


Figure 5.3: Constraint Region Represented by a Cylindrical Surface

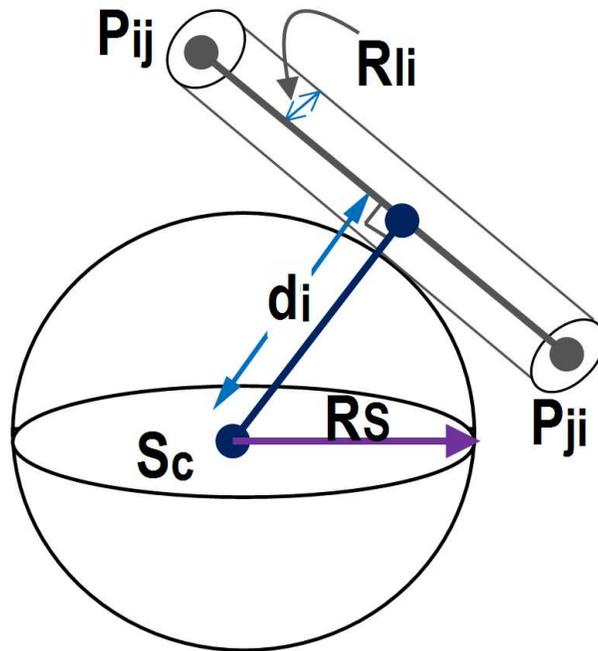


Figure 5.4: Constraint Region Represented by a spherical Surface

Equation (5.2.12). For the notation see Figure 5.3. In order to keep the links out of the restricted region, the constraints $a_i \geq (R_C + R_{il})$, where R_{il} is the radius of link i , are enforced in the optimization problem. In order to keep this obstacle avoidance constraint within the length of the cylinder, a condition is added to penalize this constraint when a link of the mechanism falls beyond the length of the cylinder. In Figure 5.3, point N_C lies on the axis of the cylinder. If it lies between P_{C1} and P_{C2} , then the constraint will be imposed, else the constraint will be penalized to be zero.

$$\mathbf{P}_{C1} + t_0 \mathbf{s}_c \leq \mathbf{N}_C \leq \mathbf{P}_{C1} + t_L \mathbf{s}_c \quad (5.2.11)$$

$$\mathbf{L}_c \cdot \mathbf{L}_i = \cos(\beta_i) - \epsilon a_i \sin(\beta_i), \quad i = 1 \dots, n. \quad (5.2.12)$$

Where t_0 and t_L are the scalars defining the limits of the cylinder and \mathbf{s}_c is the unit vector along the cylinder and defined in Equation (5.2.10).

Sphere constraint equations

Let S_C be the center of the sphere and R_C be its radius. When the region of interest is defined as a sphere, the region avoidance condition is created by imposing the distance to the center of the sphere to be greater than its radius, which yields again $2n$ quadratic inequality constraints for each configuration of the mechanism.

To avoid linkage outside of the spherical region can be stated as follows: calculate the perpendicular line from the sphere center to each link, using Equation (5.2.13). In order to keep the links out of the sphere region, the constraint $d_i \geq (R_s + R_{il})$ is

used.

$$d_i = \frac{\|(S_c - \mathbf{P}_{ij}) \times (S_c - \mathbf{P}_{ji})\|}{\|\mathbf{P}_{ji} - \mathbf{P}_{ij}\|}, \quad i = 1 \dots, n. \quad (5.2.13)$$

5.2.2.3 Force transmission specifications

The concept of transmission angle, a common performance parameter for planar mechanisms, can be adopted for spatial linkages. The components of the forces as projected from link to axis are associated with friction, chatter and jamming at the joints, while the link-to-link projection is analogous to the traditional transmission angle; transmission indices have been defined by calculating the screw geometry of input and output joint axes [23], and it is independent on the realization of the links along the joints. However, problems associated with friction at the joints can be minimized by finding proper placement of the links with respect to the joint axes. This is the issue covered in the current optimization technique.

Let us consider α_j as the minimum acceptable transmission angle for force along the axis S_j versus force perpendicular to the axis, see Figure (5.5). Then we can state the constraints for for each link of the mechanism as

$$\frac{(\mathbf{P}_{ji} - \mathbf{P}_{ij})}{\|\mathbf{P}_{ji} - \mathbf{P}_{ij}\|} \cdot \mathbf{s}_j > \cos(\alpha_j), \quad (5.2.14)$$

Notice that the acceptable angle for revolute and prismatic joints needs to be different. Compared to revolute joints, prismatic joints are much more problematic in their application, they are sensitive to the direction and manner of load application. As shown in [60], If the friction force exceeds the component of the applied force along the slide direction the joint will jam. Considering such situations and assuming that

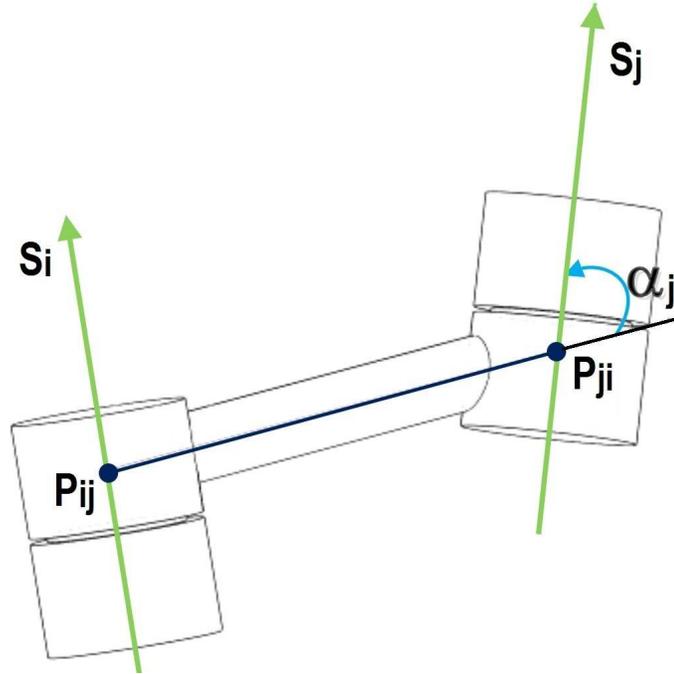


Figure 5.5: Schematic diagram for the transmission angle

the final value selected will be a function of the materials and lubrication, a generic value $\alpha = 60^\circ$ is chosen for revolute joints and $\alpha = 20^\circ$ for prismatic joints, while for cylindrical joints we use a compromise value and consider that the acceptable angle must be around $\alpha = 45^\circ$.

5.2.2.4 Planarity

For some cases, we may need spatial mechanisms to show a certain degree of planarity, at least in a given configuration, for instance at the reference configuration. This could yield quasi-foldable linkages, and it is also interesting for robotic hands used in human environments, where the fingers may be required to maintain planar shape at a reference configuration, similar to the extended human hand.

To insure the planarity of the mechanism, Equation (5.2.15) can be applied for each joint of the mechanism to be at a distance less or equal to a certain value d from

the desired reference plane. Consider the distance from a point, P , to a plane, as the smallest distance calculated from the point to any of the points on the plane, which happens along the perpendicular line. This can be defined as,

$$d = \frac{||Ax_0 + By_0 + Cz_0 + D||}{\sqrt{(A^2 + B^2 + C^2)}} \quad (5.2.15)$$

where the mechanism point is $P_{mi} = (x_0, y_0, z_0)$ and the desired reference plane is $Ax + By + Cz + D = 0$.

5.2.3 Configuration-dependent constraints

Some of the constraints on the performance of the mechanism, such as the obstacle avoidance, are a function of the configuration of the mechanism along a desired trajectory or for its whole workspace. In these cases, the constraints need to be enforced at a set of sampling points along the trajectory or workspace of the linkage. This is not too computationally costly for 1-dof mechanisms or when the motion of interest is a single trajectory, however it becomes more costly as the degrees of freedom of the mechanism increase, in which case a better strategy may be to consider the boundaries of the workspace instead. For the purpose of this article, the focus is on low-dof linkages or single one-dimensional trajectories, and the sampled trajectory strategy is used, for which a set of t equally-spaced points along the trajectory is considered.

In order to obtain the sampled trajectory, several strategies can be used. If the kinematics of the mechanism is fully defined, the trajectory can be generated using inverse or forward kinematics techniques. For those cases in which this approach is not available, a pre-defined trajectory may be given or interpolated ([36], [70]) and then

the real trajectory can be approximated [98]. For other path planning techniques, see for instance [118].

For the CRR-RRR example, Figure 5.6 shows the required trajectory. For the Bennett mechanism, the kinematics is well known and an exact trajectory can be generated, shown in Figure 5.7.

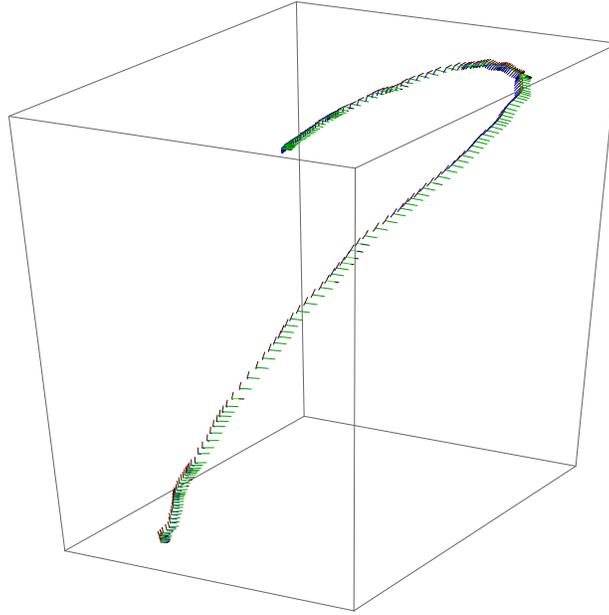


Figure 5.6: Desired trajectory of the exoskeleton design(CRR-RRR)

As an example, for the cylindrical and spherical region avoidance constraints on the anchor points, the total number of inequality constraints for space avoidance increases to $2tn$, where t is the number of positions along the trajectory.

$$\begin{aligned}
 (\mathbf{P}_{mi}^k - \mathbf{P}_{C1}) \cdot (\mathbf{P}_{mi}^k - \mathbf{P}_{C1}) - L_{mc}^2 &\geq (R_C + R_{il})^2, \\
 k = 1, \dots, t, \quad i = 1, \dots, n. & \quad (5.2.16)
 \end{aligned}$$

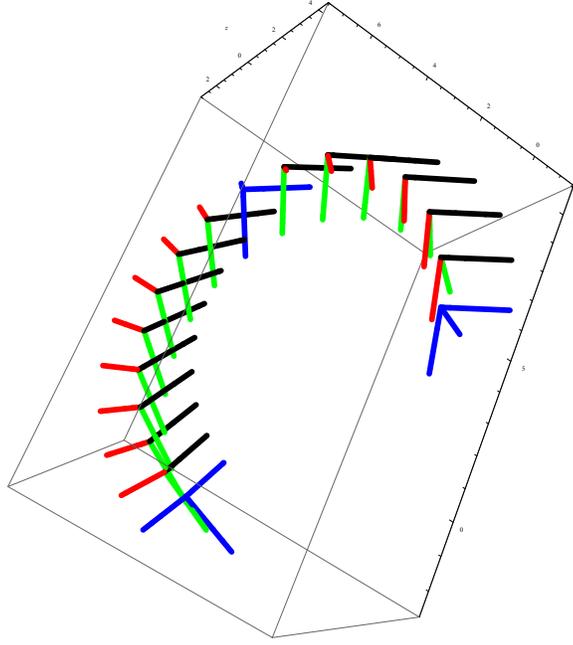


Figure 5.7: Trajectory for the Bennett linkage

$$\begin{aligned}
 (\mathbf{P}_{mi}^k - \mathbf{S}_{C1}) \cdot (\mathbf{P}_{mi}^k - \mathbf{S}_{C1}) &\geq (R_S + R_{il})^2, \\
 k = 1, \dots, t, \quad i = 1, \dots, n.
 \end{aligned}
 \tag{5.2.17}$$

Every other constraint that depends on the configuration of the mechanism must be stated similarly.

5.3 Overall optimization strategy

Considering all the constraints and the objective function, the optimization problem involves the decision variables with their lower and upper limits, nonlinear objective functions, linear inequality constraints and nonlinear constraints. The nonlinearity in the objective function and constraints provide the main difficulty in solving the

problem. A set of randomly generated solutions over those variable bounds indicate that only few solutions are feasible. Such a severe geometry of the feasible region makes the problem even more difficult to solve. For this reason the hybrid genetic optimization algorithm was found suitable to reach at a global minimum point.

For each spatial mechanism, the optimization problem is developed as shown in Equation (5.3.1). Due to the co-existence of linear and nonlinear constraints in the optimization problem, the linear constraints are mostly satisfied easily, however the nonlinear constraints may not be satisfied as easily as the linear ones. Thus, depending on the problems and the solution obtained, the Augmented Lagrangian Genetic Algorithm (ALGA)[25] is used with a penalty parameter to find the feasible region. In ALGA, a subproblem is formulated by combining the objective function and nonlinear constraint function using the Lagrangian and the penalty parameters, see Equation (5.3.2). A sequence of such optimization problems are approximately minimized using the genetic algorithm such that the linear constraints and bounds are satisfied.

min $f(x)$, such that

$$g_i(x) \leq 0, \quad i = 1, \dots, k$$

$$g_{eq_i}(x) = 0, \quad i = k + 1, \dots, kt$$

$$Ax \leq b$$

$$A_{eq}x \leq b_{eq}$$

$$lb \leq x \leq lu \tag{5.3.1}$$

$$\begin{aligned}
F(x, \lambda, s, \rho) = & f(x) - \sum_{i=1}^k \lambda_i s_i \log(s_i - g_i(x)) + \sum_{i=k+1}^{kt} \lambda_i g_{eq_i}(x) \\
& + \frac{\rho}{2} \sum_{i=k+1}^{kt} g_{eq_i}(x)^2
\end{aligned} \tag{5.3.2}$$

In Equation (5.3.2), $g(x)$ represents the nonlinear inequality constraints, $g_{eq}(x)$ the nonlinear equality constraints, k the number of nonlinear inequality constraints, and kt the total number of nonlinear constraints. A is an $m \times n$ matrix, x is an $n \times 1$ column vector of variables, and b is an $m \times 1$ column vector of constants.

The parameters λ_i are Lagrange multiplier estimates, s_i are nonnegative shifts and ρ is the positive penalty parameter. *Mathematica*[®] is used to formulate the equations and to create *Matlab*[®] executable files. Depending on the nature and complexity of the problem, the GA took from seconds to considerable amount of time to converge to optimum solution. For the illustration of the algorithm, three different examples are presented. These examples are selected to cover closed and open linkages with different applications and design requirements.

5.4 Examples

The above procedure has been applied to a spatial, closed-loop CRR-RRR, to a Bennett mechanism, and to a multi-fingered robotic hand. The *Mathematica*[®] code for the Bennett mechanism is listed in the appendix. The parameters used to run the genetic algorithm are the chromosome length that is the number of variables in the problem, the initial population, and initial value for the penalty parameter. From multiple trials, we found a probability crossover of 0.8 was good so that majority

of the population is regenerated after each generation cycle while keeping the best solutions found. A stopping criterion is set so that the algorithm stops when the population has fully converged.

5.4.1 CRR-RRR mechanism

The spatial, closed-loop CRR-RRR mechanism has one degree of freedom and with the end effector located at the intersection of the serial CRR and the serial RRR chains as shown in Figure 5.2(a). The mechanism has been designed to be used as an exoskeleton device for thumb motion [138]. Additional constraints are added to control the position, size of the mechanism, and to reduce physical interference with the user.

The kinematic synthesis of the mechanism alone yields 147 nonlinear equations in 97 variables, which makes it challenging to add more constraints in the synthesis process. The output of the synthesis ensures the trajectory of the end effector but does not provide any insight regarding size and placement of the linkage. The use of the link-based optimization as a second stage is used to adapt the linkage to those performance parameters, and illustrates the dramatic changes in the linkage that can be obtained with this method, while still targeting the same trajectory.

5.4.1.1 Problem Definition and Formulation

Consider one of the solutions obtained through the synthesis of the CRR-RRR linkage as seen in Figure 5.8. The screw axis in Plücker coordinates are given in the Table 5.1. The black lines are the links of the mechanism, drawn by connecting the common normal points of consecutive axes. The green-colored lines represent

the joint axes. The cylinder represents the user's hand and forearm, with which the mechanism should not interfere. As we can see in this initial solution, some of the joints interfere with the volume of the cylinder, and the link sizes are also not controlled, ranging from very small to very large. This initial solution is not acceptable in terms of compactness, manufacturability and assembly point of view. Therefore, a post-optimization of the design was found necessary.

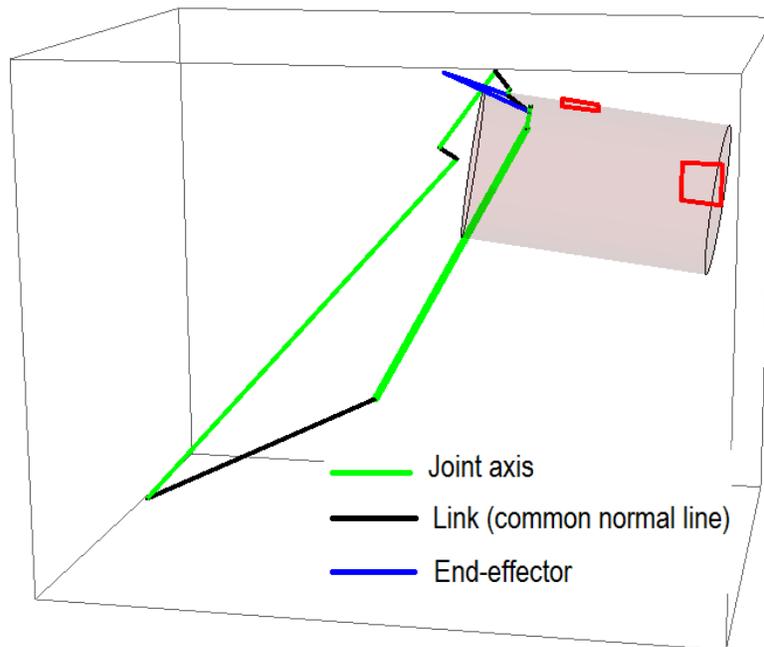


Figure 5.8: Initial CRR-RRR Mechanism. Black links are located at the common normal lines between joints.

In order to see the effect of each optimization step, the optimization of length, region avoidance and force transmission have been tested one by one and finally all at a time. In all cases, the range of the required link lengths have been set between 20mm and 150mm. The cylinder position and size is shown in Table 5.2. The optimization problem was implemented in *Matlab* code, and a total of 45 generations were used for all cases.

Table 5.1: Plucker Coordinates of the CRR-RRR Screw axes [mm]

Axis	$s_i + \epsilon(s_i^0)$
1	$\begin{Bmatrix} -0.397 \\ 0.675 \\ -0.622 \end{Bmatrix} + \epsilon \begin{Bmatrix} -295.833 \\ -191.017 \\ -18.657 \end{Bmatrix}$
2	$\begin{Bmatrix} -0.224 \\ 0.480 \\ -0.848 \end{Bmatrix} + \epsilon \begin{Bmatrix} -108.798 \\ -49.319 \\ 0.889 \end{Bmatrix}$
3	$\begin{Bmatrix} 0.253 \\ -0.523 \\ 0.814 \end{Bmatrix} + \epsilon \begin{Bmatrix} 147.218 \\ 73.695 \\ 1.551 \end{Bmatrix}$
4	$\begin{Bmatrix} 0.482 \\ -0.109 \\ 0.870 \end{Bmatrix} + \epsilon \begin{Bmatrix} -139.844 \\ 246.543 \\ 108.428 \end{Bmatrix}$
5	$\begin{Bmatrix} 0.591 \\ -0.002 \\ 0.806 \end{Bmatrix} + \epsilon \begin{Bmatrix} -179.429 \\ 349.744 \\ 132.504 \end{Bmatrix}$
6	$\begin{Bmatrix} -0.642 \\ 0.317 \\ -0.698 \end{Bmatrix} + \epsilon \begin{Bmatrix} -48.561 \\ -369.995 \\ -123.356 \end{Bmatrix}$

Table 5.2: Cylinder position and size in [mm]

P_{c1}	P_{c2}	R_C
$\begin{Bmatrix} 274.62 \\ -213.50 \\ 605.01 \end{Bmatrix}$	$\begin{Bmatrix} 71.06 \\ -239.32 \\ 634.43 \end{Bmatrix}$	63.26

When minimizing only the total link size, we obtain the link lengths shown in Table 5.3. However, as we can see in Figure 5.9, most of the mechanism joints lie inside the restricted region.

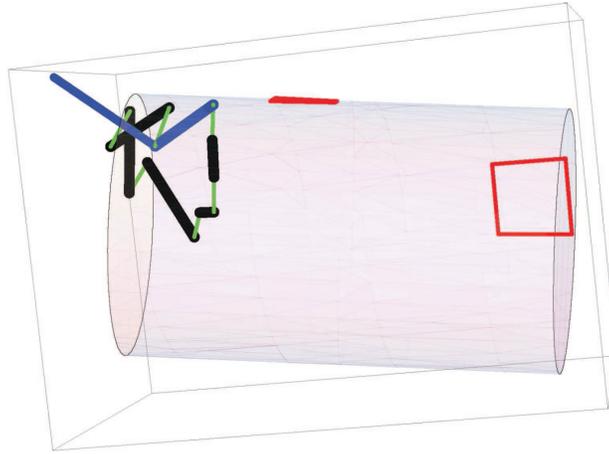


Figure 5.9: Mechanism obtained after link length constraints used

Considering the region-avoidance problem alone, the solution shows that all the links at the different task positions along the trajectory of the linkage stay out of the the restricted region, see Figure 5.10 . However, most of the link sizes are very large, see Table 5.3.

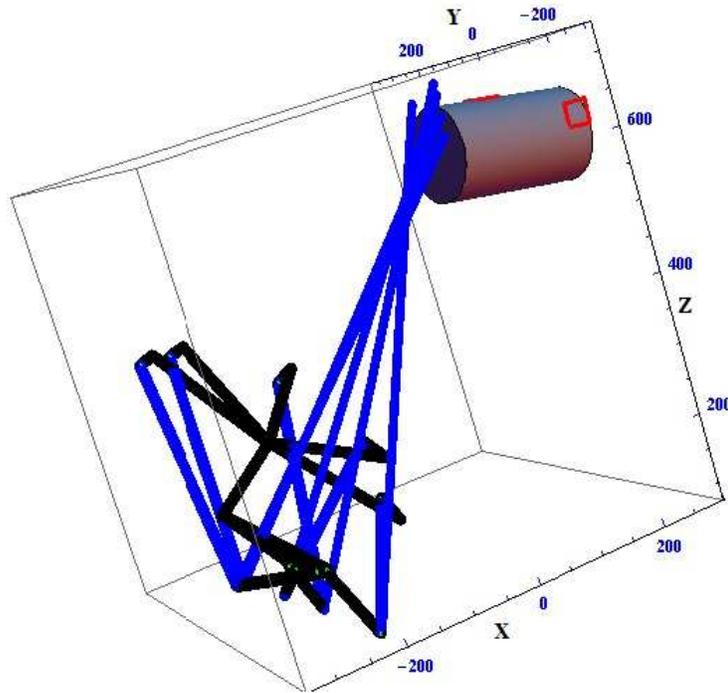


Figure 5.10: Mechanism obtained after region avoidance constraint used

Considering the link optimization, obstacle avoidance and the force transmission simultaneously gives a better result in terms of fulfilling the requirements. See Figure 5.11 for the final solution and Figure 5.12 for the motion to avoid the obstacle. The link lengths are shown in Table 5.3. Figure 5.13 shows the CAD implementation of the optimized solution.

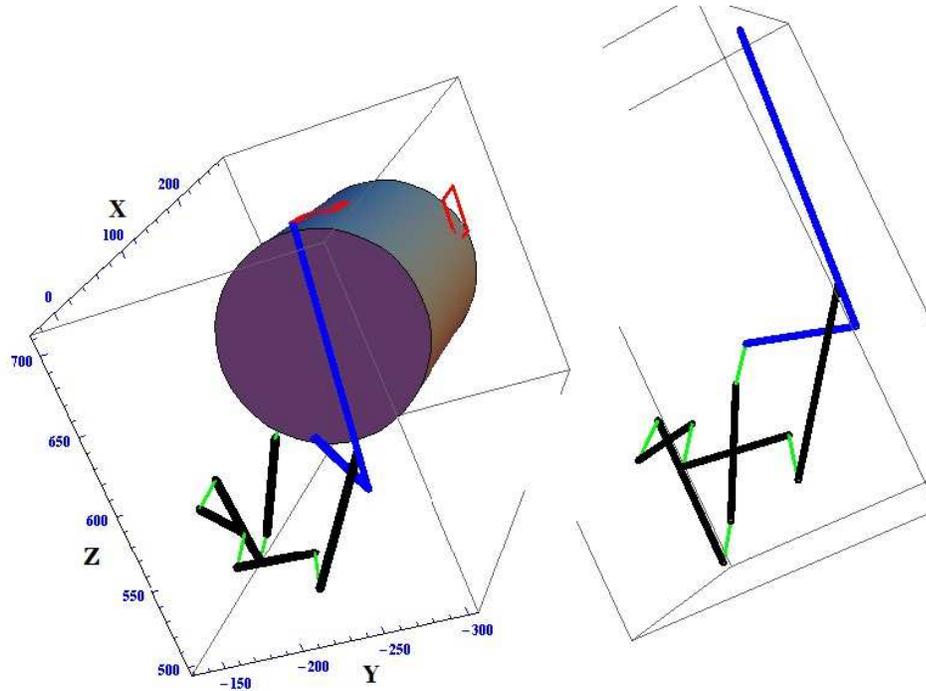


Figure 5.11: Mechanism Obtained After Optimizing Region Avoidance, Overall Length and Force Transmission

Table 5.3: Link Lengths [mm] obtained through different optimization stages: (1) Common normal lines directly from synthesis; (2) Link-size optimization;(3) Region avoidance; (4) Link size and region avoidance

Result from	L_{12}	L_{23}	L_{34}	L_{45}	L_{56}	L_{61}	L_{AEf}
(1)	0.31	0.50	25.36	31.44	18.36	26.90	55.87
(2)	20.00	57.13	76.50	46.96	40.04	51.18	20.00
(3)	244.91	46.43	543.16	125.93	184.05	277.83	671.00
(4)	150.49	57.13	138.75	46.96	109.92	150.66	151.00

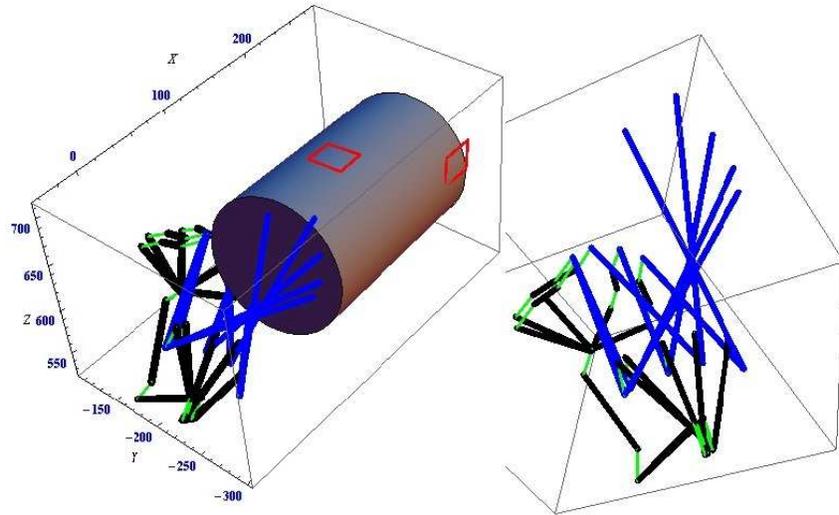


Figure 5.12: Motion of the mechanism obtained after optimizing region avoidance, overall length and force transmission. Five positions along the trajectory are reached by the mechanism while avoiding the obstacle.

The objective function (F) obtained for link-size optimization is 2.0785×10^4 , for region avoidance 1.2306×10^5 , and for link size and region avoidance is 6.6886×10^4 .

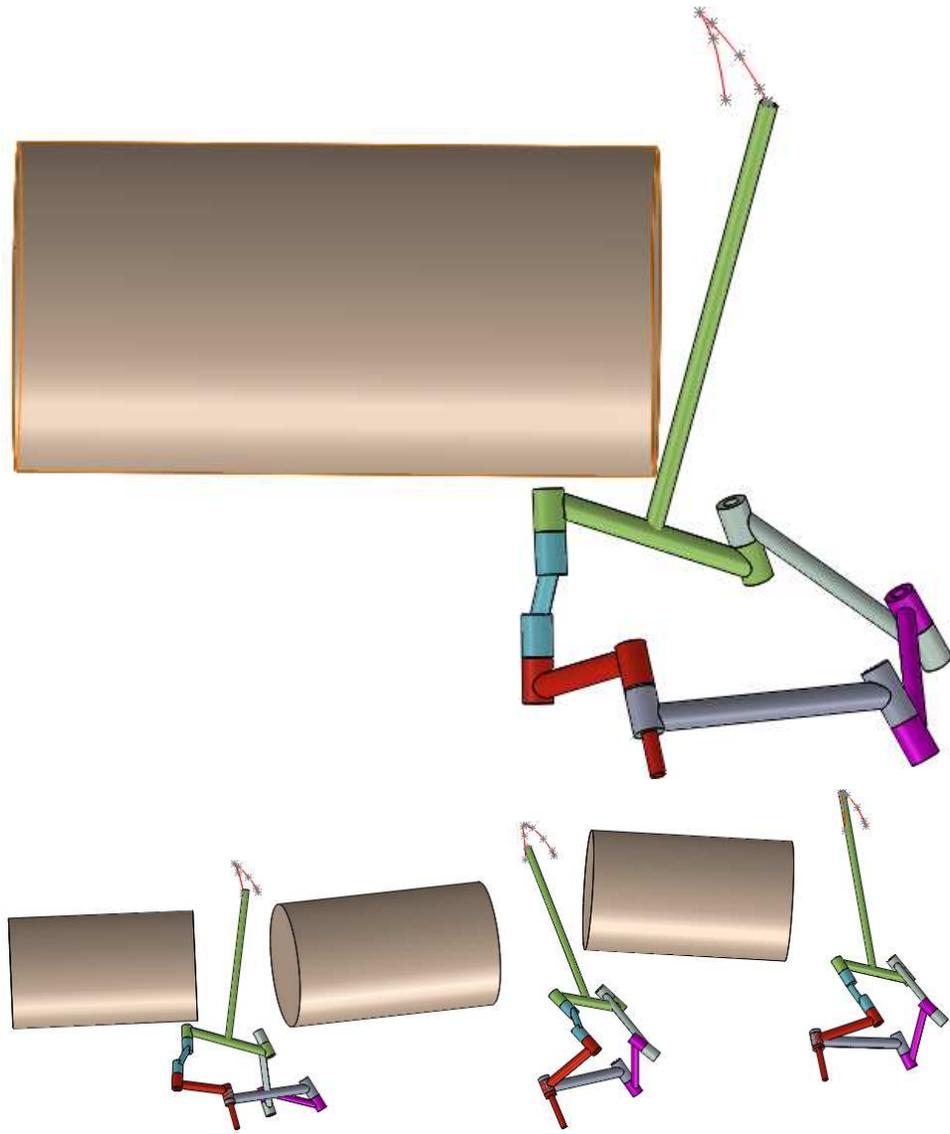


Figure 5.13: The CAD model for the final optimized solution

5.4.2 The Bennett mechanism

The Bennett Linkage is a 4R spatial closed chain. Bennett discovered the geometric relations that ensure that this chain can move with one degree of freedom. Research on the Bennett linkage has focused on its instantaneous kinematic geometry and its finite-position synthesis. In this study, the Bennett mechanism is used as a hinge for a cabinet door, Figure 5.14. The kinematic synthesis is performed to obtain the desired trajectory. To get the optimum solution considering manufacturability, obstacle avoidance and smoothness of motion, the link-based optimization technique outlined above is implemented.

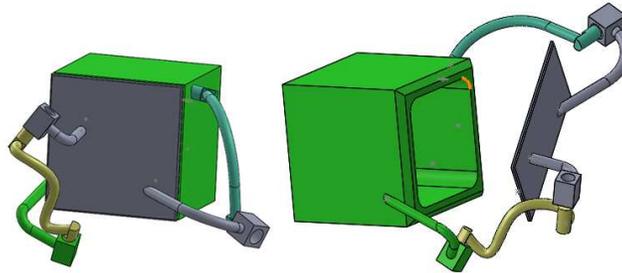


Figure 5.14: The Bennett Linkage used as a hinge and a cabinet door (Courtesy of G. Lachhwani at PsiStar Solutions)

5.4.2.1 Problem Definition and Formulation

The initial implementation from the synthesis stage uses the screw axes in Plücker coordinates as shown in Table 5.4. In this particular example the mechanism created by connecting the common normal lines is fairly compact (Figure 5.15), however the dimensions are too small from a manufacturing and assembly point of view; for instance, the smallest length is 5mm and the placement of the joints along the axes is almost zero. In addition to this, part of the mechanism lies inside the cabinet, that

is, inside of the restricted region. Considering minimum and maximum lengths, and obstacle avoidance constraints, the solution may be a less compact but applicable mechanism.

Table 5.4: Plucker Coordinates of the Bennett Linkage Screw Axes [mm]

Axis	$s_i + \epsilon(s_i^0)$
1	$\begin{Bmatrix} 0.301 \\ 0.827 \\ 0.475 \end{Bmatrix} + \epsilon \begin{Bmatrix} -5.12 \\ 1.901 \\ -0.068 \end{Bmatrix}$
2	$\begin{Bmatrix} 0.588 \\ 0.809 \\ 0.009 \end{Bmatrix} + \epsilon \begin{Bmatrix} -2.970 \\ 2.210 \\ -4.896 \end{Bmatrix}$
3	$\begin{Bmatrix} 0.798 \\ -0.372 \\ 0.475 \end{Bmatrix} + \epsilon \begin{Bmatrix} -1.901 \\ -4.164 \\ -0.068 \end{Bmatrix}$
4	$\begin{Bmatrix} 0.988 \\ -0.156 \\ 0.009 \end{Bmatrix} + \epsilon \begin{Bmatrix} -0.381 \\ -2.675 \\ -4.896 \end{Bmatrix}$

Figure 5.16 shows the optimized Bennett mechanism with imposed constraints of minimum link length of 30mm, maximum link length of 60mm and an offset length of 15mm. As we can see, the joints have now enough offset for assembly, and the link lengths also fall within the range of the specified constraints. However, part of the mechanism is still inside of the restricted region, which is the cabinet; therefore, additional constraints to avoid this problem are required.

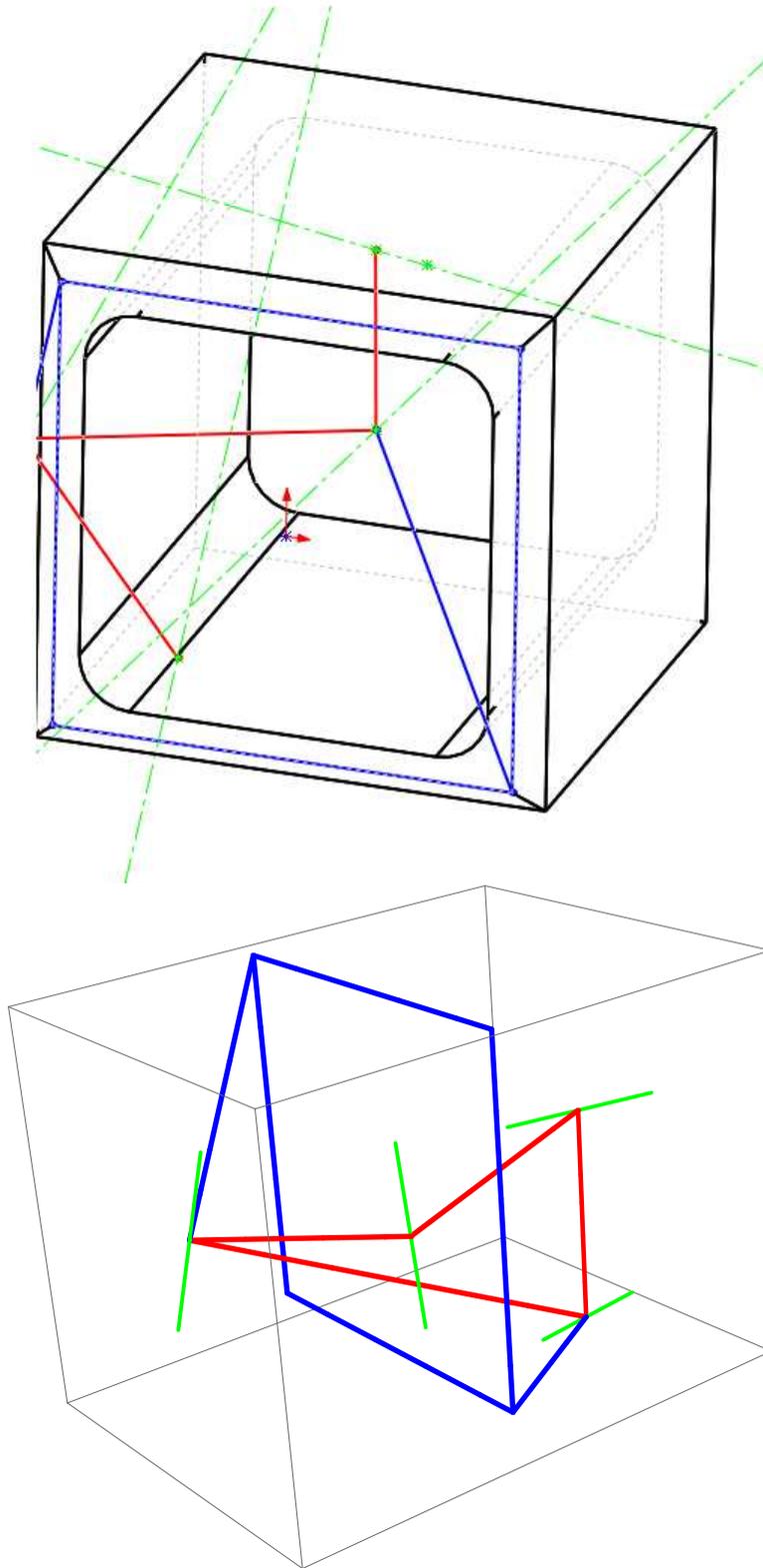


Figure 5.15: Initial Solution of the Bennett Linkage

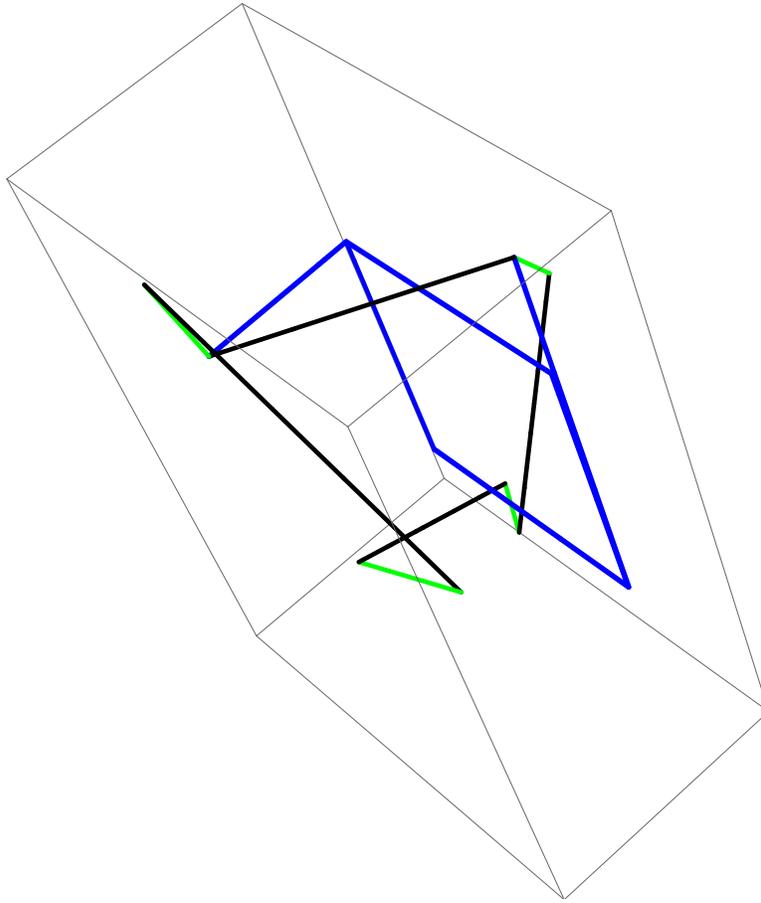


Figure 5.16: Optimized Bennett Linkage with Link Length Constraints

Obstacle avoidance

The cabinet has the shape of a cube. To simplify the problem, an inscribed sphere is considered to define the region avoidance constraint. An increment δ for the radius is used to take into consideration the thickness of the links, to yield a sphere radius $S_R + \delta$. The obstacle avoidance constraint keeps the mechanism on one side of the door and out of the cube, Figure 5.17. However, the link sizes are much longer than the perceived optimal solution. Finally, by incorporating all the constraints including link length, offset length and obstacle avoidance, a better solution is found and shown in Figure 5.18. The motion of the linkage can be seen in Figure 5.19.

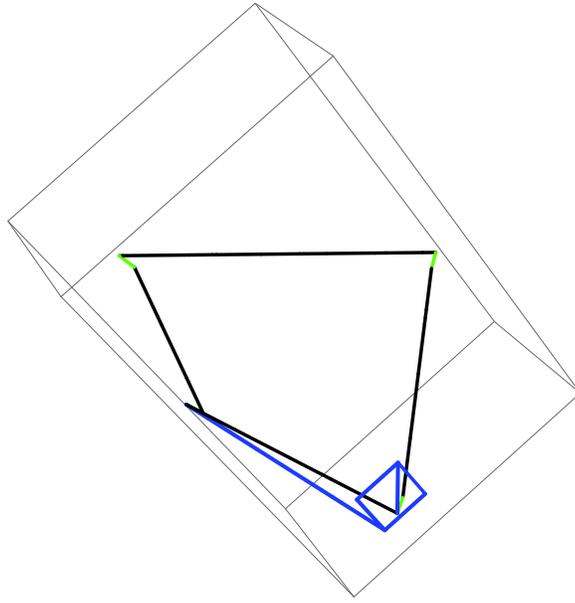


Figure 5.17: Mechanism with obstacle avoidance and offset constraint. The blue square corresponds to the cabinet door.

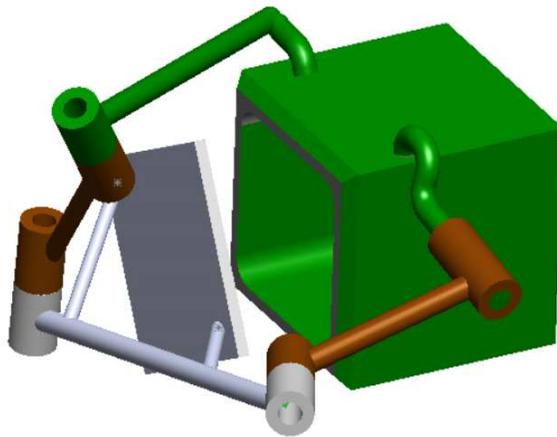


Figure 5.18: Mechanism obtained with link length, offset length and obstacle avoidance constraints.

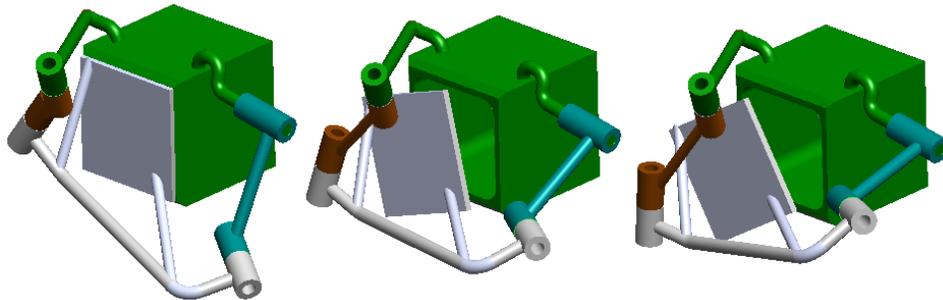


Figure 5.19: Motion of the final design for the cabinet linkage

5.4.3 Application in robotic hand design

The link-based optimization method is a useful tool in the design of robotics hands. For robotic hands and grippers that need not be anatomically similar to the human hand, and depending on the robotic hand task, we can have different sized fingers, bigger or smaller palm, smaller wrist and so on. With the use of the optimization method, these design preferences can be incorporated either in the objective function or as a constraint. In this example, a three-fingered robot hand is considered for the task of rotating a door nob. The action is recorded using a Vicon camera system and the trajectory is shown in Figure 5.20.

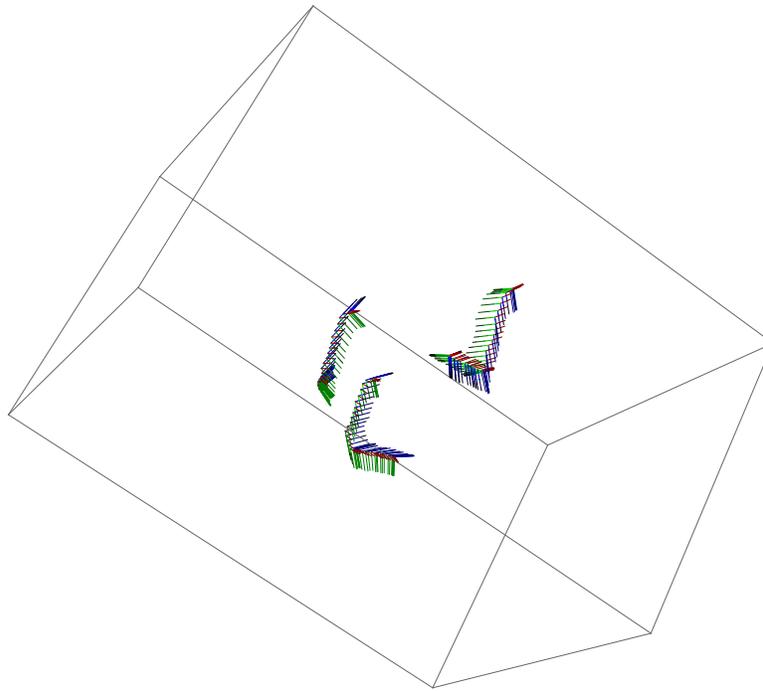


Figure 5.20: Trajectory of the three fingers to rotate a door nob

Considering one of the many synthesis solutions for this problem, and creating the links at the common normals, the hand shown in Figure 5.21 is obtained. Here, it is clear that some of the links are too small to be manufactured and assembled. In a

first run, the link-based optimization with link length and offset constraints improves the result, see Figure 5.22. Another option is found by providing separate ranges of the link length constraints for the wrist alone, in which case a very compact solution is achieved, see Figure 5.22.

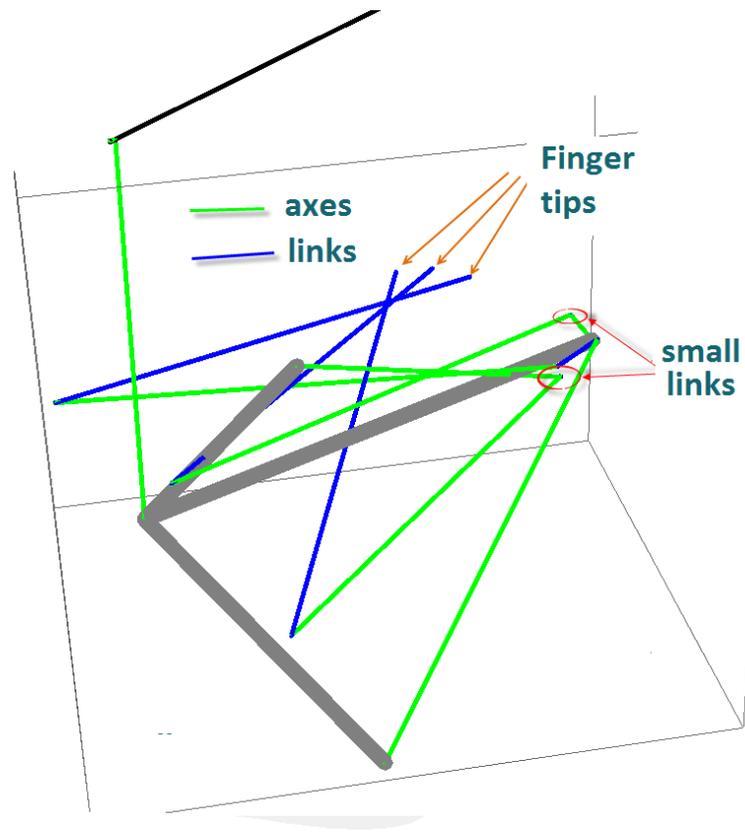


Figure 5.21: The robotic hand before optimization

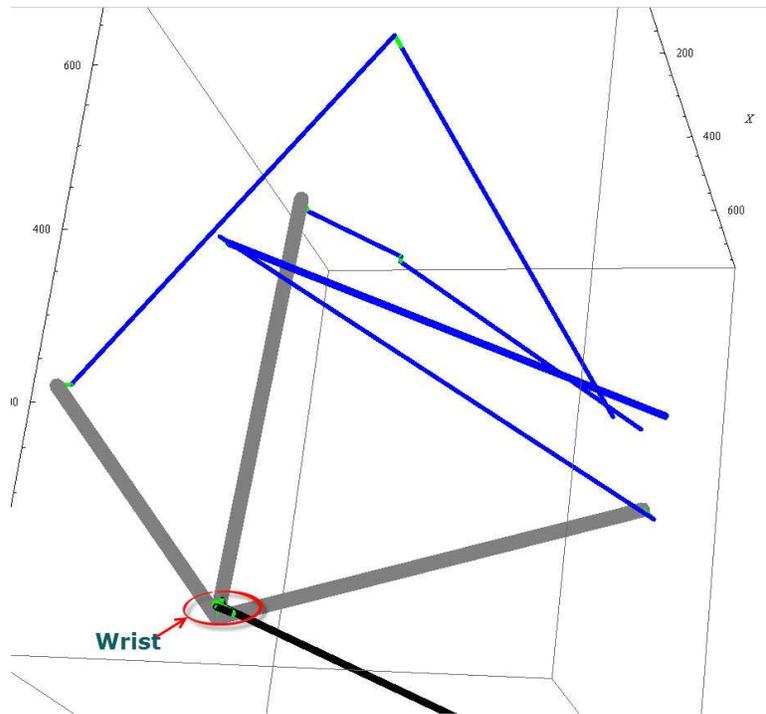


Figure 5.22: Optimized with link length and offset constraints

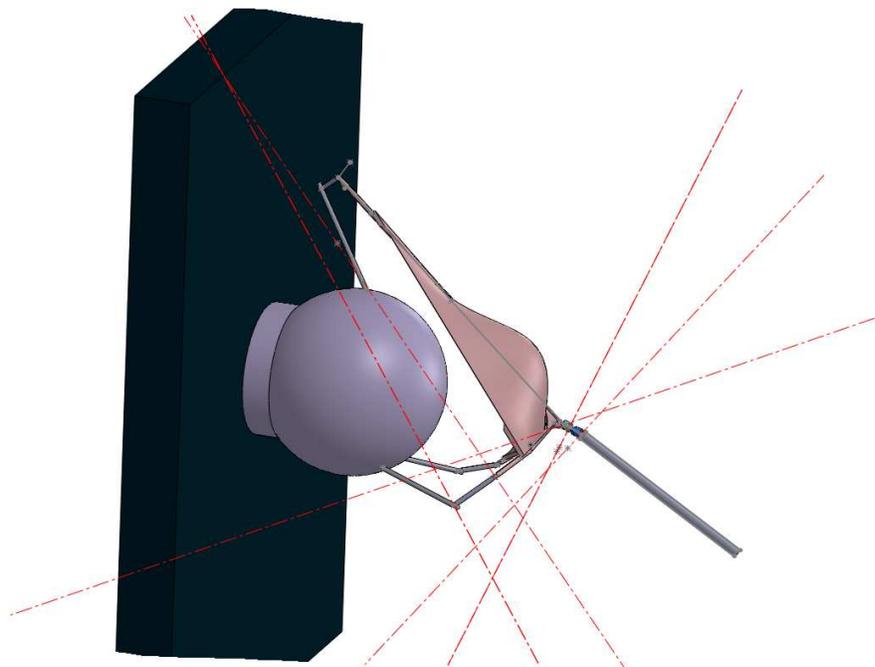


Figure 5.23: A three fingered robotic hand rotate a door nob

5.4.4 Summary

In the design of spatial linkages for a desired motion, a great deal of flexibility is allowed in the choice of the links. The optimization presented in this thesis is applied to the linkage in order to fulfill additional performance requirements, such as total length, force transmission, obstacle avoidance or geometry at a given configuration. It is our experience that trying to fulfill these requirements with a manual manipulation of the links is tedious and overall difficult. The results obtained using this optimization show that it could be a useful tool for the designer of spatial mechanisms with arbitrarily-located axes. It is straightforward to define the links in a CAD environment as anchored in points sliding along the joint axes of the linkage, making the application of the optimization results automatic.

The methodology is general enough that any link-based requirement can be added to the optimization, either within the objective function or as additional constraints. The algorithm is also flexible so that it can be applied to the overall mechanism or just to a part of it. The presented examples show that dramatic changes in the implementation of the mechanism can be obtained by using this method. The links are defined as straight lines between anchored points at the joints; in a future extension, the links will be allowed to have a curved shape in order to increase the solution space.

Chapter 6

Design of an Exoskeleton as a Finger-Joint Angular Sensor

6.1 Introduction

Accurate hand pose tracking and finger joint measurements are important research topics in bioengineering fields for different applications; the design of exoskeleton and prosthetic devices and the implementation of control algorithms are a few of those. Most of the available prosthetics using EMG or sEMG sensors compute some threshold value for the corresponding finger joint positions [106] [132] [54] [101] [38]. In most cases, models are obtained with the smoothed sEMG data as input and the respective smoothed finger angle data as output. The dynamic model obtained allows the instantaneous control of the finger motions.

Most of the existing angular joint sensors rely on the assumption of the knowledge of the type of motion and location of the joint. The design presented here consists of

an exoskeleton, designed to fit the finger motion, in which we can relate the angular displacement of its links to the change in orientation of the phalanx under consideration. Unlike other designs, the exoskeleton does not need any information about the actual anatomy and dimensions of the hand in order to provide with the angular information.

6.2 Design methodology

The methodology shown in Figure 6.1 has been followed to design the sensor.

6.3 Collection of design input data

A camera vision system is used to track the hand motion. Non-contact technologies are mostly vision-based or use infrared or magnetic technologies. Most of the focus in single-camera tracking has been with detecting the region of interest (ROI) of the hand. Some research has attempted to create simplified hand models through markerless detection [40], [76]. Work has also been done using markers [76] and multi camera systems [59]. A marker-based detection technique is used to get information about the finger motion.

6.3.1 Image acquisition

The experimental approach to get the data is based on the principle that the pose of a calibrated camera can be uniquely determined from a minimum of four coplanar but non-collinear points. Thus, two squares attached to the proximal phalanges and to the dorsal part of the carpal are used as shown in Figure 6.2. Then the index finger is moved to its different positions/reaches. While moving the index finger,

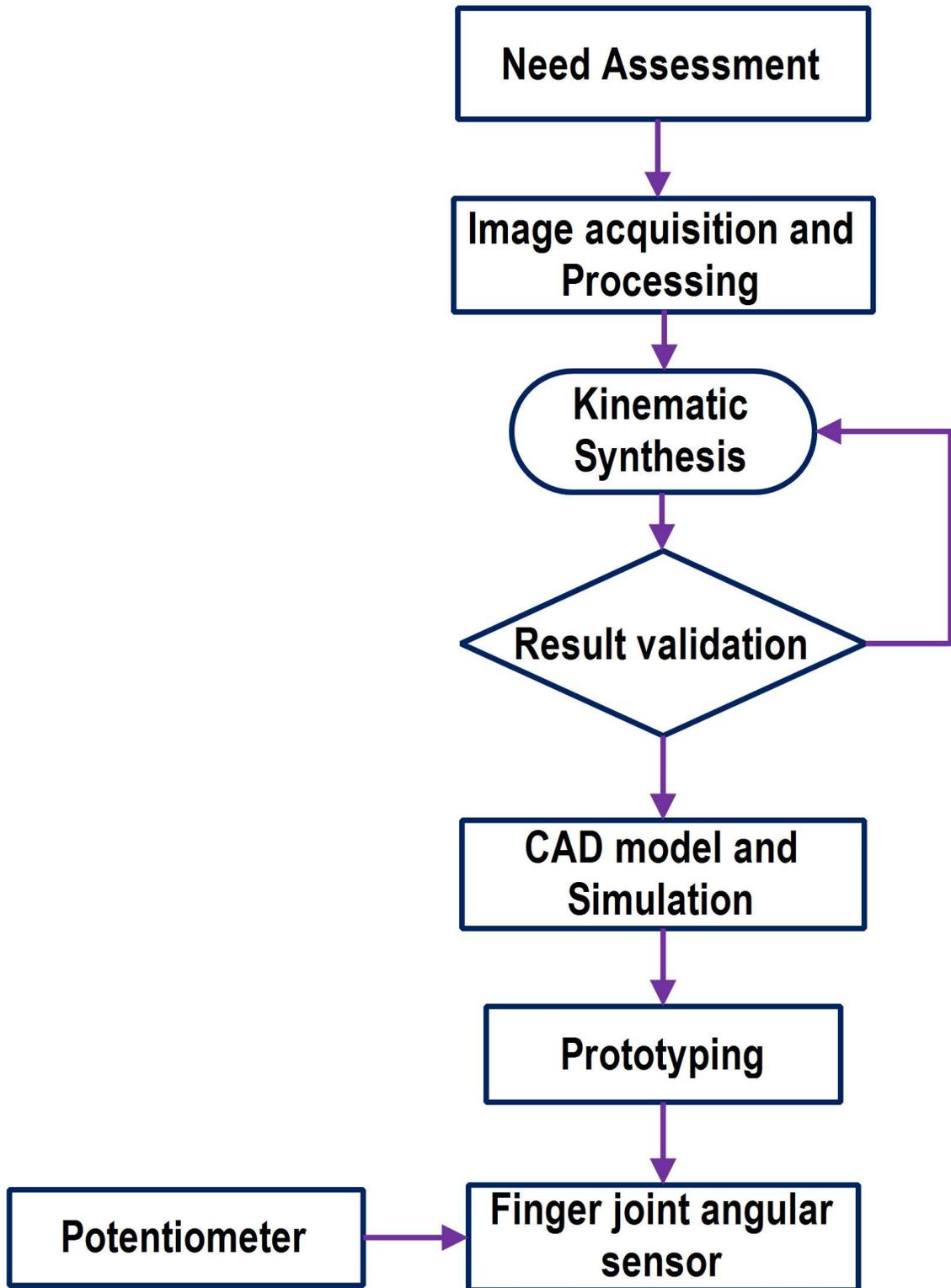


Figure 6.1: The methodology applied to design the exoskeleton based sensor

different frames are captured using a Dragonfly 2 camera from GreyPoint, with a Fujinon 1:1.4/9mm lens, interfaced with a computer using LabVIEW. The LabVIEW program captures and saves multiple images into the hard disk [4]. Once the frames are saved, sample frames from different orientation of the index finger are selected for processing.



Figure 6.2: The captured image of the index finger with the square blobs on it

6.3.2 Image processing

In order to compute pose for 3D to 2D correspondences of a planar target, the algorithm used is based on Hager and Schweighofer [109]. The algorithm is customized to obtain a set of task positions from a video stream of a hand moving with markers. As it is clearly visible on the figure, the square on the dorsal carpal is used as a reference position, as the relative position between this square and the fixed joint of the mechanism is always constant while the index finger is moving.

The markers are used to estimate the pose, which consists of the position and orientation of the finger. An important aspect of this setup is the geometry of the markers. The geometry of a marker affects directly its performance and usability in computer vision applications. The design used by this project is a simple white

square with a smaller black square inside as shown in Figure 6.2. It gives four sharp visible corners that form a perfect square to be used to find the 3D pose of the marker. Figure 6.4 shows a properly detected marker. It is important for these markers to be completely rigid for accurate pose estimation. For detection of the candidate points the Harris corner and edge detector algorithm [115] is used.

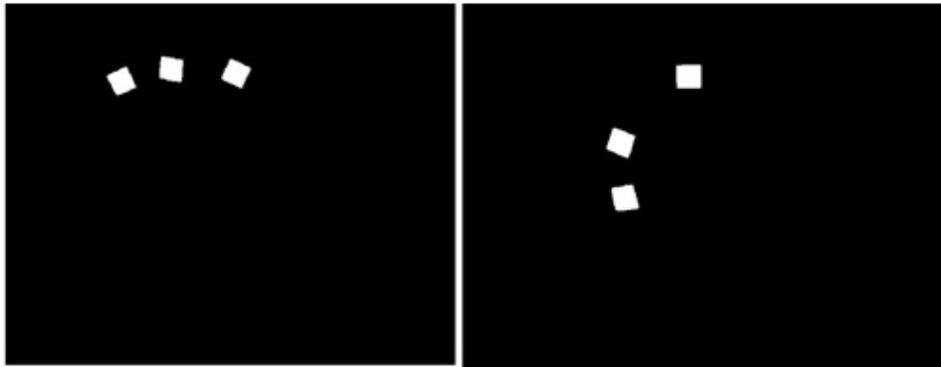


Figure 6.3: The index finger movement and the square blobs

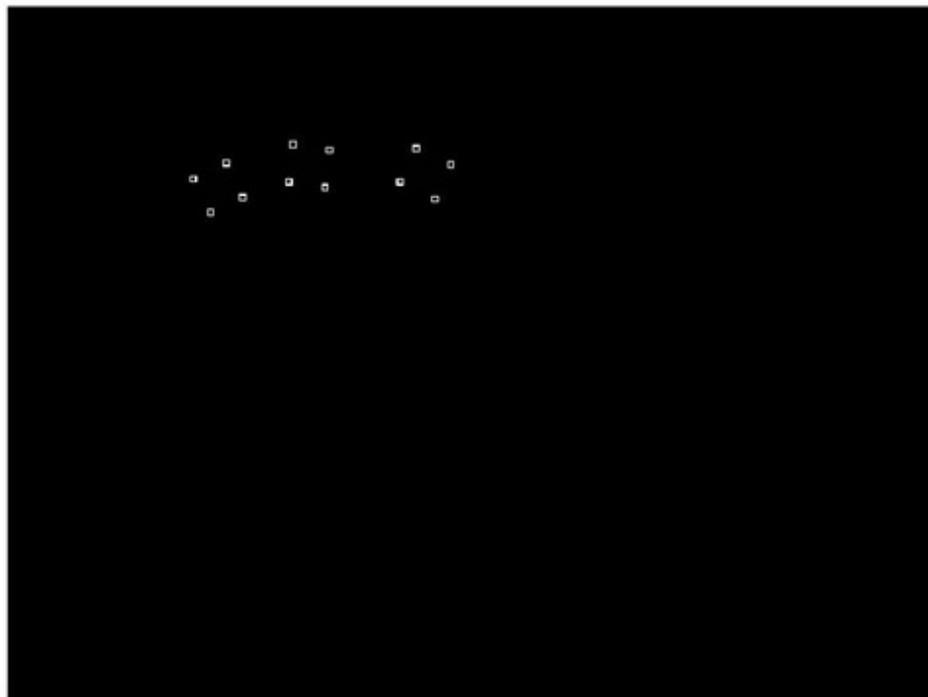


Figure 6.4: Corner detection of one frame done by Harris corner detection

The 3D pose recovery algorithm can map 3D reference points from the 2D image coordinates. It takes a set of non-collinear 3D coordinates of reference points $P_i = (x_i, y_i, z_i)^t, i = 1, \dots, n, n \geq 3$. These points can be expressed in an object-centered reference frame. The corresponding camera-space coordinates are $q_i = (x'_i, y'_i, z'_i)^t$. These two points are related by the rigid transformation

$$q_i = RP_i + t, \tag{6.3.1}$$

where R and t are the rotation matrix and translation vector respectively. Using this approach we obtain a transformation matrix T_{pi} for each position of the target point (the point from the square at the proximal phalanx) with respect to the camera and we also obtained T_{ci} (the point from the square at the carpal) with respect to the camera. In order to reference locally all movements of the target point, we use coordinate transformation as $T_{cp} = T_{ci}^{-1}T_{pi}$. After getting T_{cp} , the transformation from the reference frame at the carpal to the proximal phalanges, the design equations are formulated based on the candidate mechanism .

6.4 Finger Exoskeleton Design

It is important to notice that the internal structure of the hand -the skeleton- is not a part of the exoskeleton mechanism. The mid phalanx of the finger is attached to the coupler of the linkage, as indicated in Figure 6.5, and the whole mechanism is placed on the dorsal carpal of the hand. The exoskeleton is designed for the coupler to follow a task motion in this case, the motion of the phalanx. The exoskeleton

mechanism is designed to be light and have a smooth motion over the range of the finger, so to minimize dynamic loading effects on the signal under study. An angular potentiometer is to be attached to one of the joints of the linkage, and the sensed angle is to be related to the angle of the coupler with respect to the fixed link.

Type synthesis, the selection of the topology of the mechanism to be used as exoskeleton, is a first step in the design process. For this sensor, the approach is to use the simplest mechanism possible, with decreased complexity, weight and inertia on the finger. The planar four-bar mechanism was tried first, but it was not successful due to the interference of the links with the finger and the dorsal part of the hand in all designs. The six-bar linkage is the next simplest one. It allows positioning the links away from the finger so that no interference appears, while matching the planar 1-dof motion of the single finger joint. Thus, a planar, single-dof six-bar linkage (see Figure 6.6) has been selected as the exoskeleton topology.

6.4.1 Exoskeleton Design Equations

Using the vision information as an input, the kinematic synthesis of the mechanism is performed to obtain the exoskeleton-based joint angle sensor. For this application, the desired mechanism is required to do two things: it should follow the trajectories described by the collected data; and there should be a one-to-one correspondence between the orientation of the MIP joint of the finger to one of the joints of the mechanism.

The six-bar linkage is the simplest closed, 1-dof planar linkage able to follow the collected trajectories accurately. On the other hand, being a simple, one-dof linkage,

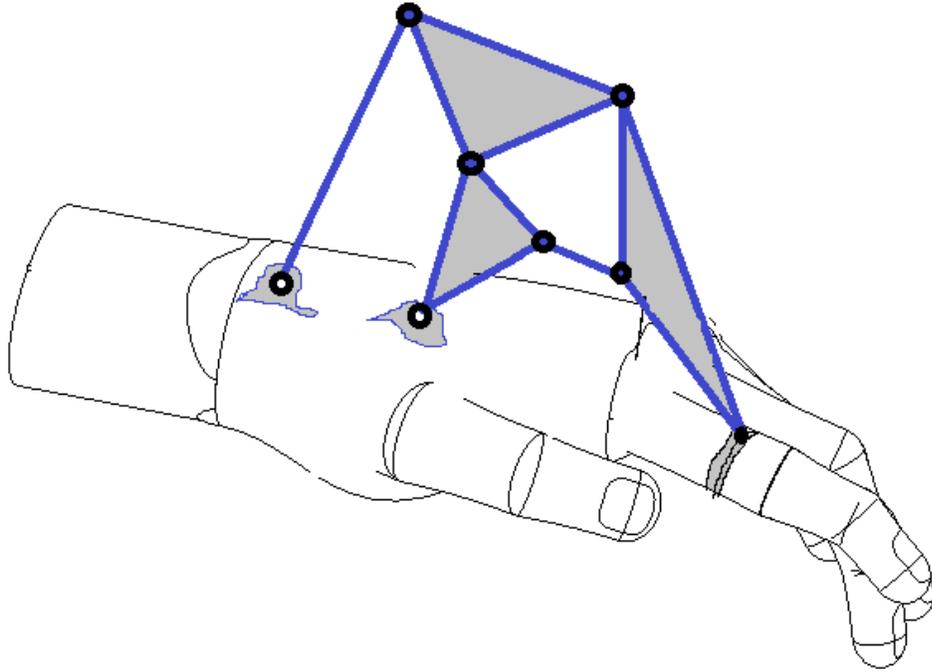


Figure 6.5: Schematic drawing of the exoskeleton on the index finger

the relationships between all its angles and the driving joint angle are well known and can be related to the angle of the coupler and the MIP joint.

The variables defined for the six-bar mechanism are shown in Figure 6.6, where θ_1 , θ_2 , θ_3 , θ_4 and θ_5 are the joint angles and the remaining parameters (i.e., s_{1x} , s_{1y} , s_{4x} , s_{4y} , a , l_1 , l_2 , l_3 , l_4 , l_5 , b_1 , b_2 , α , γ and δ) are the structural variables of the mechanism. Compared with the more common four-bar mechanisms, six-bar mechanisms have more design variables, therefore with an appropriate design, six-bar mechanisms can adapt to a wider number of motions.

Using the variables defined in Figure 6.6, the forward kinematic equations for a planar six-bar mechanism are defined as [84], [133].

the mechanism. The axis shown in Figure 6.6 indicates the angle φ is the same as the one of the MIP joint of the finger.

Nine positions are selected from the index finger trajectory. The end-effector location at the coupler link yields two equations, and the loop equations account for four equations for each position. This gives a total of 52 nonlinear equations. In addition, we have 9 angular equations; overall, we have 63 equations. The total variables to be found are 61. Some auxiliary equations are added to limit the size of some key links in the linkage.

The equations are solved using a Levenberg-Marquardt nonlinear, unconstrained solver implemented in Java. This is based on public domain MINPACK routines, translated from FORTRAN to Java by Steve Verrill [1].

The process yields many solutions. Each solution took an average of 7.5 minutes on a 2.2GHz Intel Core i7. It has been solved several times, in groups of 50 runs, out of which approximately 3 solutions were acceptable each time. The acceptability was defined in terms of position on the hand, overall dimensions and hand interference. The accepted candidates were modeled using CAD software in order to select the final design.

Angular Measurement

From the design of the mechanism it is shown that the angle φ corresponds to the orientation of the MIP joint of the finger. Thus, using Equation (6.4.2) the value of φ for every finite displacement of the finger in terms of the angles θ_1 , θ_2 , γ , θ_3 , α and β is obtained. Here α , γ and β are constants, and θ_1 , θ_2 , θ_3 are the joint variables,

which can be expressed in terms of one of them (for instance θ_1),

$$\varphi^i = \theta_1^i + \theta_2^i + \theta_3^i + \alpha + \beta + \gamma, \quad (6.4.3)$$

where i corresponds to any single measurement of the angle.

We use Equation 6.4.3, together with the constraint relations between the joint angles [83] that give $\theta_2^i = f(\theta_1^i)$ and $\theta_3^i = f(\theta_1^i)$.

Therefore, the only variable that we need to measure is θ_1 . Since one of our objectives is to come up with a cost-effective sensing device, a resistive potentiometer is selected to be mounted on that joint of the six-bar mechanism to measure θ_1 . Similarly, relations can be found to place the sensor at any other joint.

6.5 Results

The candidate designs were ranked considering size, mechanism placement and overall structure. The selected design is optimized and modeled as shown in Figure 6.8. The design parameters of the selected six-bar linkage are shown in Table 6.5 (angles in radians and lengths in millimeters).

A rapid prototype has been built in order to further assess the performance of the sensor and it is shown in Figure 6.8. The final product will be made from aluminum and the estimated total cost for several joints, including machining, is within hundreds of dollars. We believe that this sensor will be cheaper than other sensors such as data gloves, magnetic, and infrared sensors, which cost in the order of thousands of dollars.

To avoid interference and increase comfort to the user the first model has been

Table 6.1: Exoskeleton dimensions

s_1	(46.34, -94.63)
s_4	(-30.49, -57.91)
a	-77.66
l_1	-40.01
l_2	-68.46
l_3	20.00
l_4	-68.46
l_5	20.00
b_1	-122.58
b_2	-122.57
γ	-0.27
α	3.04
δ	0.28
β	2.27

modified, optimized and modeled as shown in Figure 6.9. The modified sensor has been prototyped and tested. Figure 6.10 shows the experimental setup and the prototyped sensor. To increase the sensitivity of the sensor a Wheatstone bridge has been built; the potentiometer on the exoskeleton is used as the variable resistor in the bridge. For the convenience of recording and monitoring the sensor value, the microcontroller and labview have been used as shown in Figure 6.11. Using this labview model we can monitor the analog signal change due to the position change of the potentiometer on the sensor. Once we have recorded the data, the angle relation obtained from the kinematic analysis will be applied to relate the real change of the MCP joint angle.

6.6 Summary

The development of a simple and cost-effective mechanism for the estimation of the angles of the Metacarpal interphalangeal (MIP) joint of the index finger is shown. The

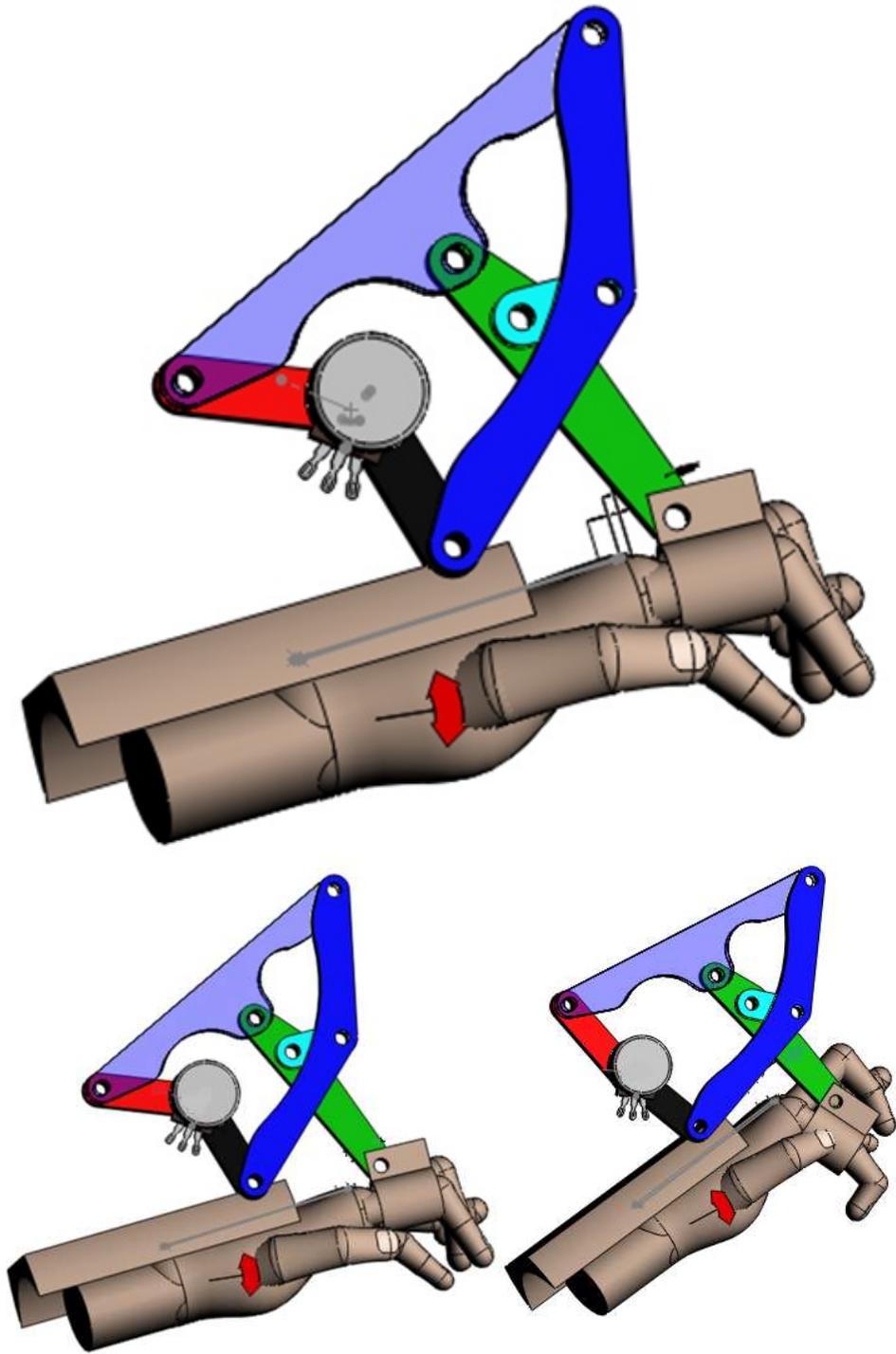


Figure 6.7: CAD model of the selected linkage and the sensor on the hand

design strategy includes vision system and image processing coupled with kinematic synthesis techniques. The main advantage of this method is that it does not need any

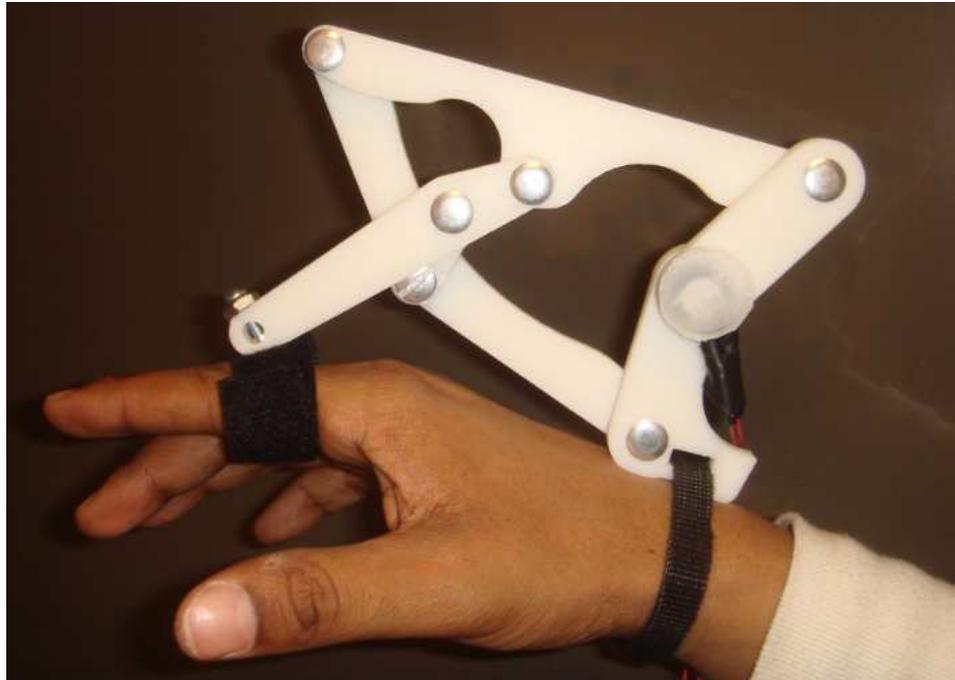


Figure 6.8: Initial prototype of the selected linkage

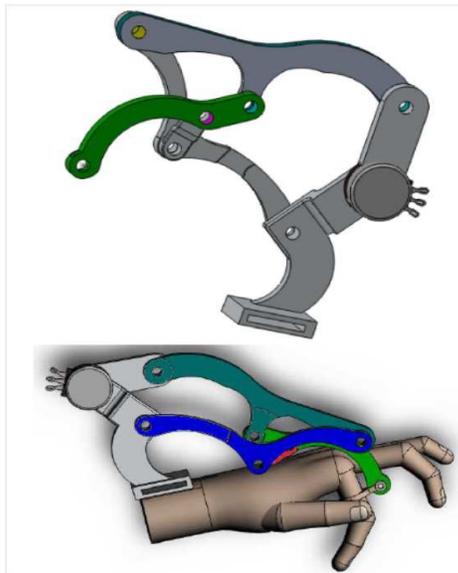


Figure 6.9: The modified CAD model of the exoskeleton and the sensor on the hand

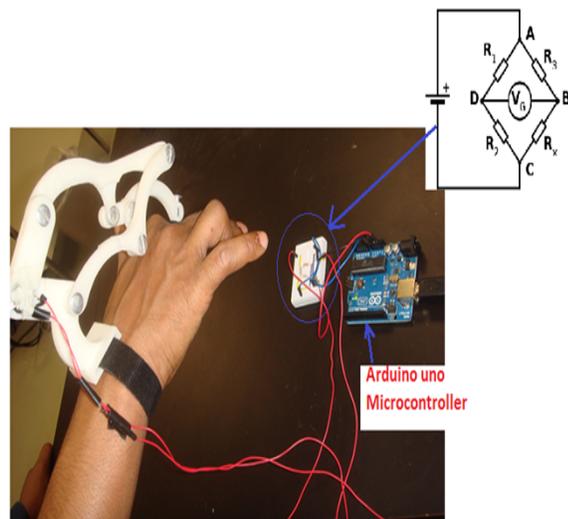


Figure 6.10: The experimental setup of the sensor

assumption about location and type of joints in the subject; the exoskeleton is going to follow the trajectory that is selected as task regardless of the skeleton structure

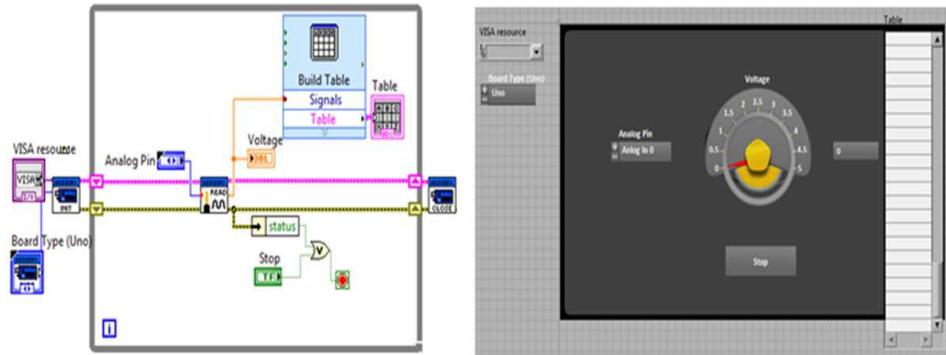


Figure 6.11: Labview program to show and record the data

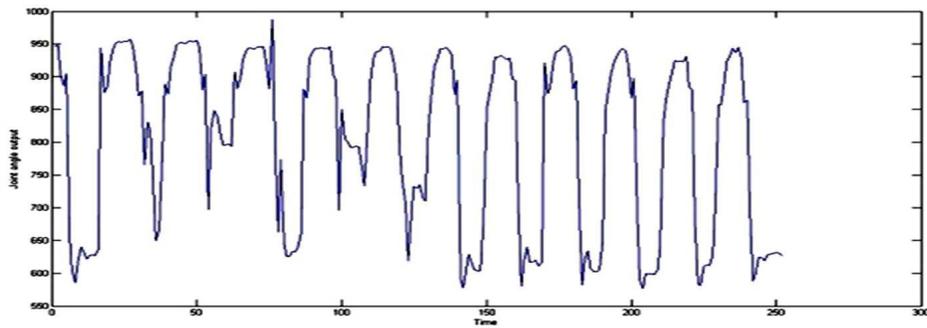


Figure 6.12: The joint angle data

that generates it. This allows for the creation of new and innovative exoskeleton-based position sensors which can help for EMG and position modeling in the grasp and control study. The application of image-processing techniques and use of a six-bar mechanism and a simple potentiometer grants a cheap, effective sensing device. The high number of solutions obtained means more choices for the designer.

Chapter 7

Single Degree-of-Freedom

Exoskeleton Mechanism Design for

Thumb Rehabilitation

7.1 Introduction

This research presents a task-oriented design methodology for exoskeletons used in spatial motion. In particular, the methodology is applied to develop a 1-dof thumb exoskeleton for rehabilitation. The exoskeleton is required to generate the desired grasping and pinching path of the thumb. The human thumb presents a complex 3D motion that can be modeled, depending on the needed accuracy, with three to four degrees of freedom, and using variable joint axes. We postulate that it is still possible to use simplified, low-dof linkages for assisting in this motion. We focus on a set of closed, spatial overconstrained and non-overconstrained four-bar to six-bar

linkages with low mobility that present the desired characteristics for this application, see [124] and [120]. The spatial mechanism is to be attached to the proximal phalanx of the thumb. In addition, the designed mechanism is confined to the back of the hand, so as to minimize sensory feedback interference, and to allow the mechanism to be manufactured with minimal size. This, combined with the intended location of the actuators, will allow the device to be constructed with low apparent inertia. The following subsections present the design methodology and results.

7.2 Thumb Mechanism Design

The method is based on synthesizing a linkage to follow as closely as possible experimental paths of the human thumb. The mechanism needs to have a single degree of freedom, which can be listed as one of the advantages of the mechanism which requires a single actuator yet gives 3D complex motion. The overall outline or design approach of the mechanism is shown in Figure 3.1.

The thumb data were acquired using a Vicon motion tracking system as shown in Figure 3.2.

The several experimental paths obtained were separated for clarity. Figure 7.1 shows one typical point path, seen from the reference frame of the motion capture system. For the design of spatial motion, it is sometimes advantageous to work with relative displacements. Each relative displacement expresses a motion of the thumb from a reference configuration, taken as the thumb position at the first frame. Each displacement can be modeled as an axis, plus a rotation about and a translation along the axis. This information is encoded as a *screw*, where the screw axis is the

axis of the displacement and the pitch is the ratio of translation to rotation for that displacement.

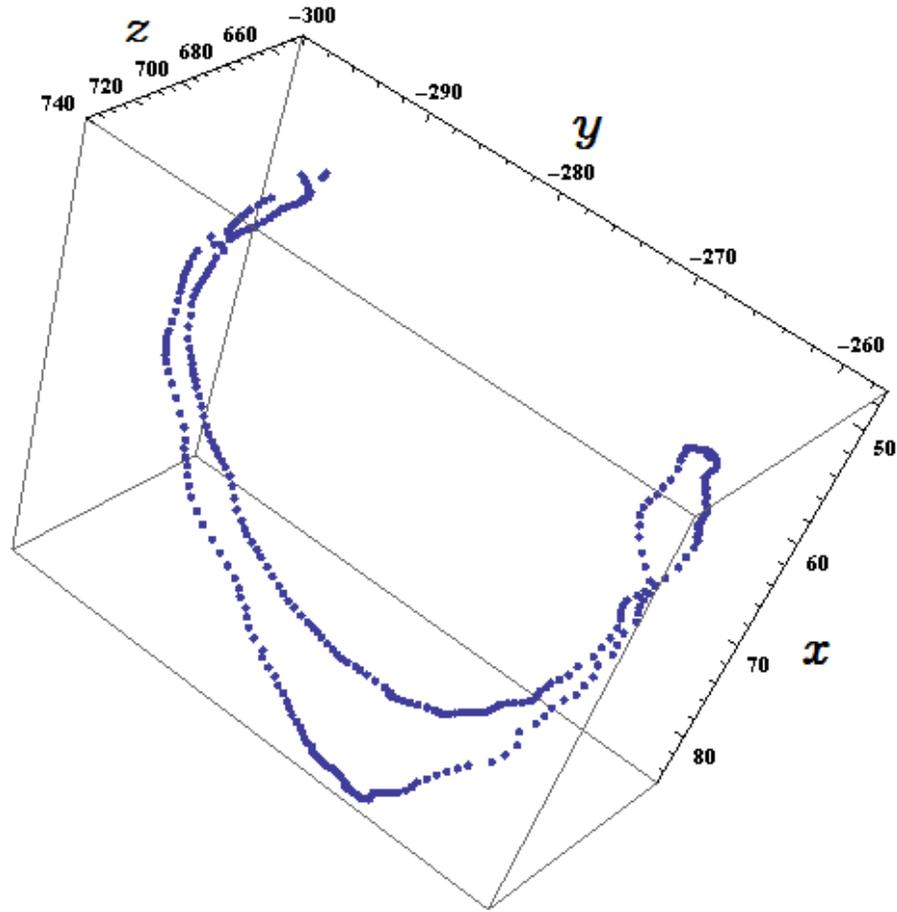


Figure 7.1: Thumb's proximal phalanx point path

Figure 7.2 shows the displacements of the thumb's proximal phalanx path as screw axes with a pitch, where the screw lengths are proportional to the pitch. The screw axes of the displacements with their pitches generate a *screw hypersurface*. This representation has all the information of the motion; except for the value of the rotation, which can be calculated independently.

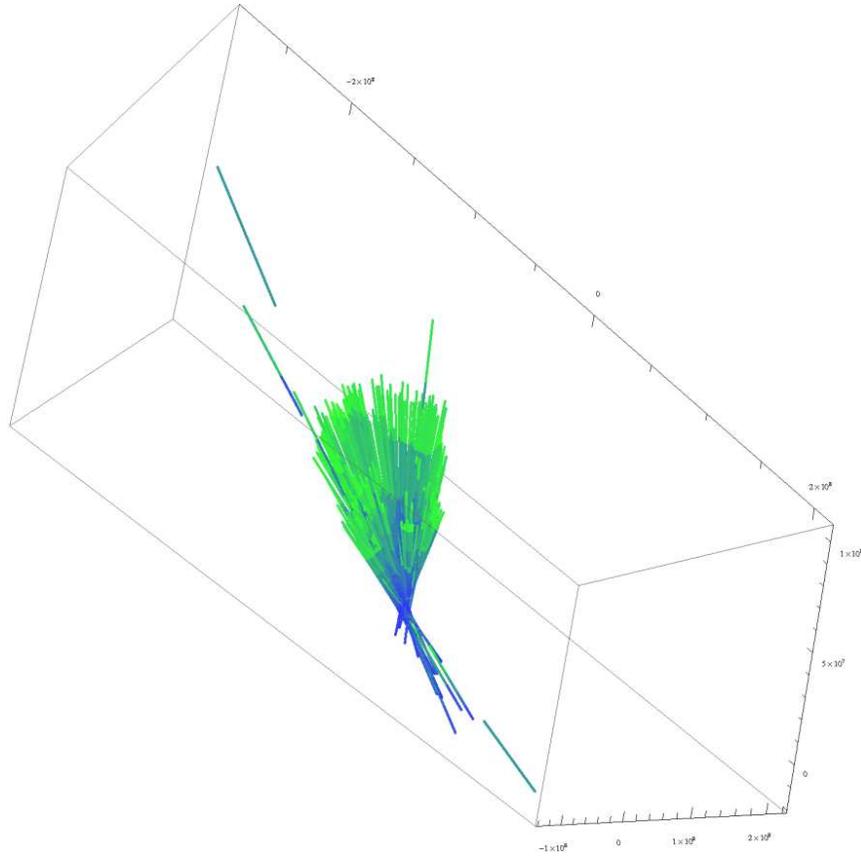


Figure 7.2: Thumb's proximal phalanx path: screw surface of relative screw axes

7.2.1 Mechanism Selection

In order to accomplish simplicity together with spatial motion under a one-degree-of-freedom system, an initial set of closed spatial linkages with four to six links and standard revolute (R), prismatic (P) and cylindrical (C) joints have been selected. Some of these linkages are overconstrained, while others are trivial; all of them with mobility equal to one [124], [120]. Figure 7.3 shows the topology of the spatial CCCC linkage (a linkage with four cylindrical joints); candidate linkages with four links are particular cases of this one, obtained by making some of the joint variables (θ_i, r_i) constant.

Similarly, the closed, spatial CCC-CCC linkage can be seen as the general case

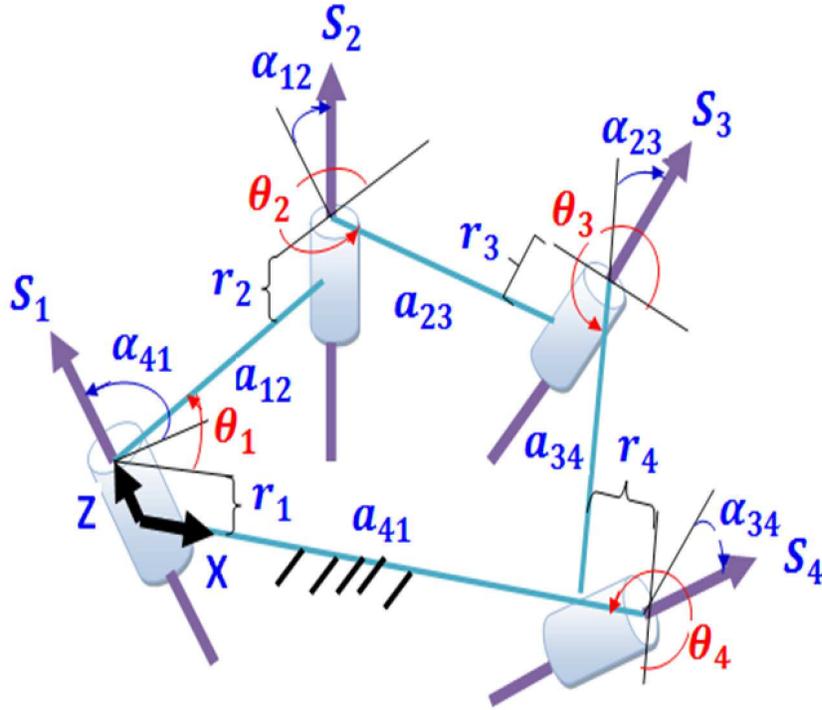


Figure 7.3: A spatial 4-bar CCCC linkage

for the six-bar candidate linkages, see Figure 7.4. In particular, the following four-bar linkages: RC-CC, RP-RP, RR-RR, and the following six-bar linkage: CRR-RRR were selected as candidates. Here, the dash separating joints indicates where the end-effector, or attachment to the thumb, is being placed.

Among the properties of these linkages that are useful for our application we can cite the 1-dof motion, requiring only one actuator, and topological simplicity while creating a complex motion. In addition, overconstrained linkages have other advantages; such as inherent structural rigidity.

7.2.2 Mechanism Design Equations

In this section, the design equations corresponding to the CRR-RRR mechanism are presented. The reason to do so is that it turned out to give the most fitted

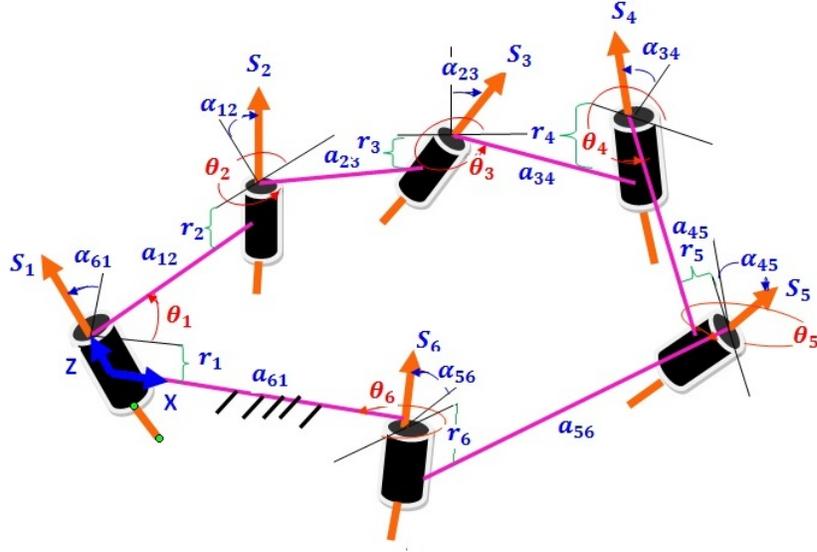


Figure 7.4: A spatial 6-bar CCCC linkage

mechanisms for the task. Let us consider the closed CRR-RRR linkage as two serial chains, CRR and RRR, joined at their end-effectors. The axes are labeled as shown in Figure 7.4, starting at the fixed C joint and going around up to the final fixed R joint. For every joint i , let $S_i = s_i + \epsilon s_i^0$ be the joint axis, with rotation θ_i , and slide (for the C joint only) d_i . The forward kinematics equations of the CRR and RRR chains are expressed using dual quaternions [100],

$$\begin{aligned} \hat{Q}_{CRR}(\Delta\hat{\theta}_1, \Delta\theta_2, \Delta\theta_3) &= \prod_{i=1}^3 \left(\cos \frac{\Delta\hat{\theta}_i}{2} + \sin \frac{\Delta\hat{\theta}_i}{2} s_i S_i \right) \\ \hat{Q}_{RRR}(\Delta\theta_6, \Delta\theta_5, \Delta\theta_4) &= \prod_{i \in \{6,5,4\}} \left(\cos \frac{\Delta\hat{\theta}_i}{2} + \sin \frac{\Delta\hat{\theta}_i}{2} s_i S_i \right) \end{aligned} \quad (7.2.1)$$

where $\Delta\hat{\theta}_i = \Delta\theta_i + \epsilon\Delta d_i$ is the dual angle, and all $d_i = 0$ except d_1 corresponding to the cylindrical joint. The forward kinematics so expressed represent the set of relative displacements of the chain with respect to a reference configuration.

In order to create the design equations, the distance between the displacements captured in Section II.B. and the displacements of the candidate chain is minimized. Then dimensional synthesis is performed, with a goal to find the location and dimensions of the mechanism that approximately performs the task.

The design equations are created by equating the forward kinematics of the mechanism to each of the discrete positions obtained from the motion capture. If we denote each finite displacement of the thumb as \hat{P}^i , we can create the relative displacements with respect to the first position of the thumb, $\hat{P}^{1i} = \hat{P}^i(\hat{P}^1)^{-1}$, to yield design equations.

$$\begin{aligned}\hat{Q}_{CRR}(\Delta\hat{\theta}_1^i, \Delta\theta_2^i, \Delta\theta_3^i) &= \hat{P}^{1i}, \\ \hat{Q}_{RRR}(\Delta\theta_6^i, \Delta\theta_5^i, \Delta\theta_4^i) &= \hat{P}^{1i}, \quad i = 2, \dots, m.\end{aligned}\tag{7.2.2}$$

In these equations, the variables we are interested in are what we call the *structural variables*, which are the Plucker coordinates of the joint axes $\mathbf{S}_i = \mathbf{s}_i + \epsilon\mathbf{s}_i^0$ at the reference configuration. In addition, the optimization process outputs the angles of the chains in order to reach the thumb displacements.

To complete the system of Equations in (7.2.2), size constraints were imposed on the mechanism so that it can be attached to the lower arm and with reasonable dimensions. In particular, for the six-link CRR-RRR mechanism, the constraints of distance between both fixed axes and also between the fixed axes and the thumb were

added,

$$\begin{aligned} S_1 \cdot S_6 &= \cos \alpha + \epsilon a \sin \alpha \\ S_1 \cdot P^1 &= \cos \beta + \epsilon b \sin \beta \end{aligned} \tag{7.2.3}$$

where P^1 is the screw axis of the first thumb position, and we fix the distance between the axes along the common normal, a , to a value between 50mm and 150mm, and the distance between the thumb attachment and the coupler axes, b , to similar values.

7.2.2.1 Implementation of the Design Equations

Ten positions were selected from the thumb path, and the first frame was taken as the reference configuration. Each forward kinematics equality is composed of 8 equations, and forward kinematics are written for both serial chains composing the mechanism. This gives a total of 144 nonlinear equations. In addition, we have the constraints of Equation(7.2.3). In sum, we have 147 equations.

The variables to solve for are the Plucker coordinates of the axes, that is, six parameters per axis, and the joint variables to reach each thumb position. The total is 97 unknowns.

The equations were solved using a Levenberg-Marquardt nonlinear, unconstrained solver implemented in Java. This is based on public domain MINPACK routines, translated from FORTRAN to Java by Steve Verrill [123].

7.3 Results

The best results were obtained for the CRR-RRR mechanism. One of the sets of ten equally-spaced positions selected from the thumb data can be seen in Figure 7.6. The equations were run 14 times for three different sets of positions chosen from the thumb frames. The distance to the desired path has been optimized by minimizing the distance at each step. The overall error of the function was smaller than 0.03, and it took a variable amount of time, from a few minutes to a few hours, to find solutions. For these 14 runs, 14 considerably different solutions were found.

Out of these 14 solutions, 2 linkages were selected because of their overall dimensions and placement on the hand. Figure 7.7 shows the SolidWorks model of those solutions, named candidate I and II.

The mechanism with the best combination of fit to the path, dimensions and placement is selected as a solution. Due to the potentially very large number of solutions for this problem, not all the solution space has been searched and hence we cannot assume that the selected candidate is the optimal one, but rather an acceptable one. Figures 7.5 and 7.6 present the actual motion of the linkage as compared to the thumb path and design poses. Even though there is some small divergence in the paths, we must point out that it is of the order of the variability of the several paths observed in the motion capture data, rendering an overall motion that is within the normal thumb actuation. The other possible option to minimize the deviation is to use higher degree of freedom mechanism, but with the cost of increasing actuation and control complexity. In this thesis, to avail the advantages of the multi-degree

of freedom mechanisms while keeping the control complexity issue minimum, a cable driven RRC-CRR spatial mechanism for the thumb rehabilitation device is used.

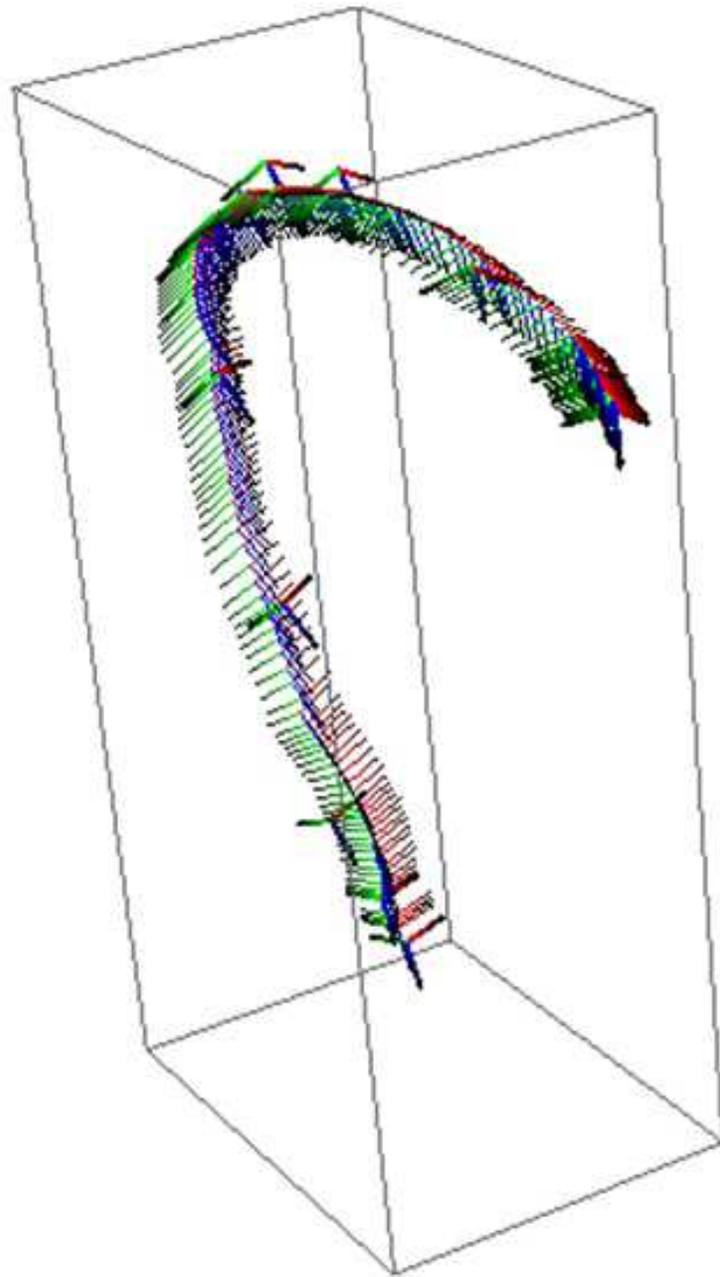


Figure 7.5: One of the thumb paths (thin frames) with superimposed linkage path (thick lines)

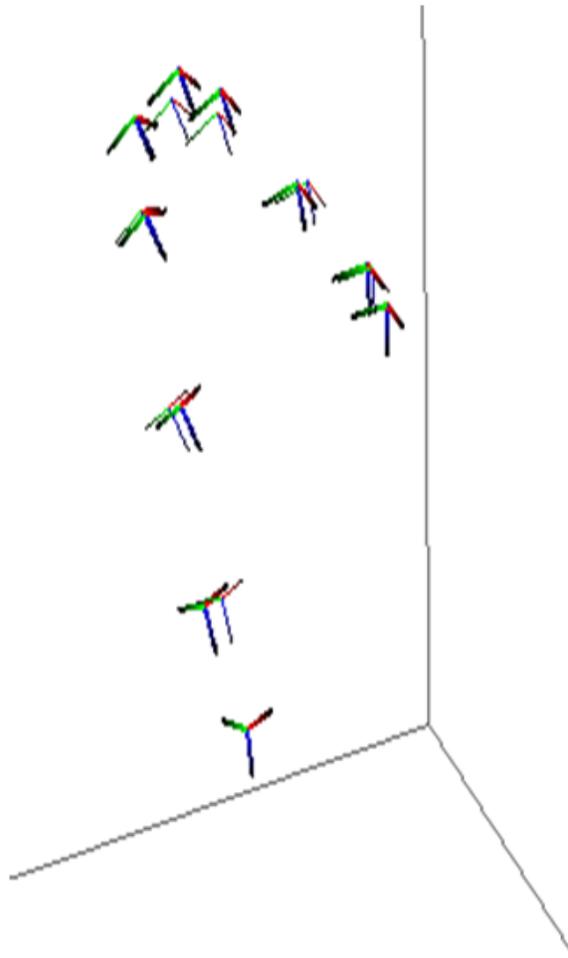


Figure 7.6: Comparison between design positions (thin lines) and linkage positions (thick lines)



Figure 7.7: The two initial solutions selected for prototyping

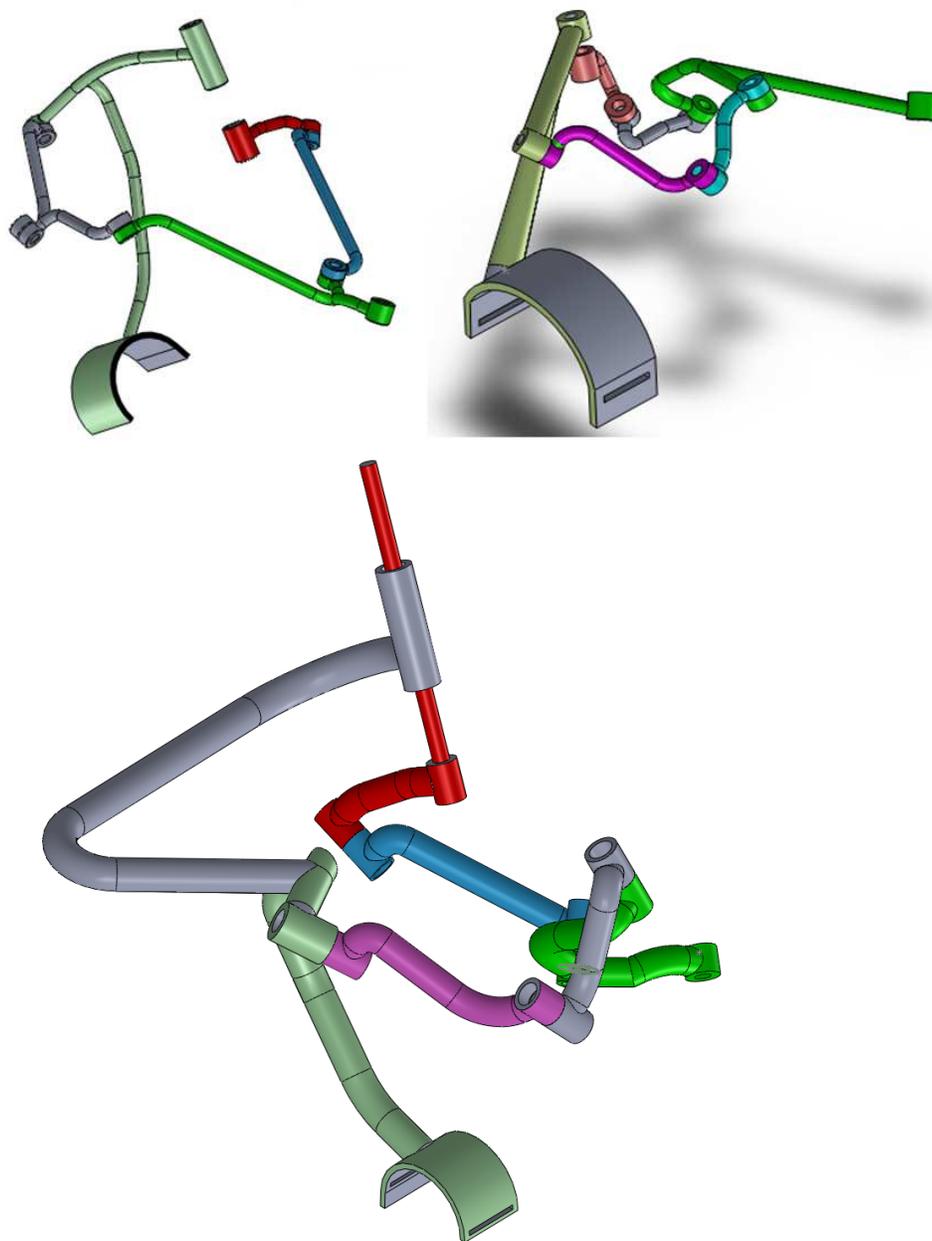


Figure 7.8: Other solutions for different data set

Rapid prototypes have been built in order to assess the motion and to better design the hand attachment and compatibility with the finger exoskeleton. Figure 7.9 shows the prototype linkages mounted on the hand.

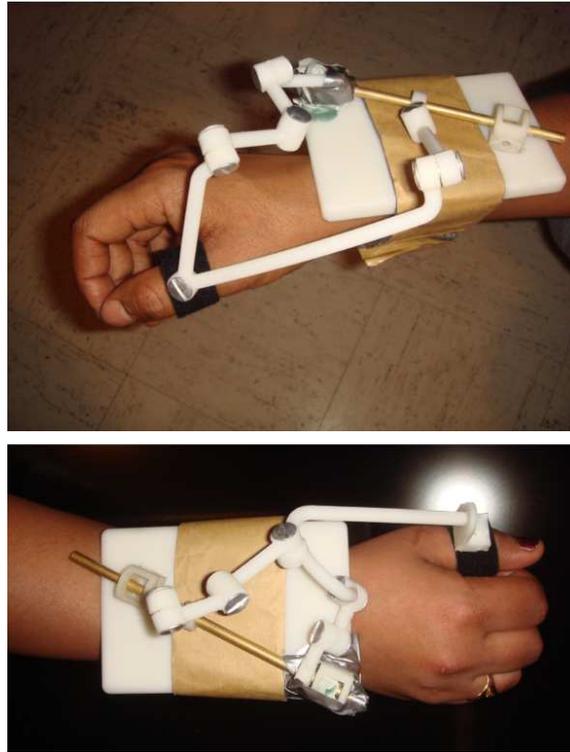


Figure 7.9: prototype of the initial solutions

7.4 Summary

This chapter presented a task-oriented design methodology to develop a 1-dof thumb exoskeleton for rehabilitation. The kinematic design is followed by a CAD modeling and prototyping.

Chapter 8

Conclusions and Future works

8.1 Conclusions

In this research, a comprehensive methodology for the kinematic design of exoskeleton devices, including input data acquisition, kinematic synthesis and post-synthesis optimization is presented. The algorithms and the methodologies developed in this research are tested on exoskeleton applications. Results are showing that with a lower degree of freedom mechanisms, it is possible to build an exoskeleton which can follow a complex 3D motion without having any assumption about the location and type of joints in the subject. The exoskeleton follows the trajectory that is selected as task regardless of the skeletal structure that generates it. This allows for the creation of new and innovative exoskeleton designs. Results have been published in conference articles [138], [139], [140], book chapters [14], and submitted to journals [137].

8.2 Future works

In this research, the following ones are proposed as a future research. The focus is being directed towards the kinematic synthesis and post-synthesis optimization stages.

- In order to ensure the motion along a full trajectory, the new direction of research on exact workspace synthesis seems to be an appropriate solution. This research will extend the exact synthesis methodology already developed for the RCCR chain to other closed, spatial mechanisms such as RRRR and RCRC. This requires characterizing their workspace using the robot kinematics equations. The challenges include finding a suitable mathematical formulation to computing the intersection of each serial chain that creates the parallel linkage, either in parameterized or in implicit form.
- Post-synthesis, link-based optimization has yielded good results for the detailed design of the exoskeleton. Current results can be extended to include several other factors and applications. For instance, based on the result shown in [65], self-intersection of the spatial linkages can be incorporated in the link based optimization formulation.

References

- [1] *Optimization java package*, 2000. Program library long writeup W5013[Online]. Available <http://www1.fpl.fs.fed.us/optimization.html>.
- [2] S.K. Acharyya and M. Mandal. Performance of eas for four-bar linkage synthesis. *Mechanism and Machine Theory*, 44:17841794, 2009.
- [3] Z. Affi, L. Romdhane, and A. Maalej. Dimensional synthesis of a 3-translational-dof in-parallel manipulator for a desired workspace. *European Journal of Mechanics, A / Solids*, 23:311–324, 2004.
- [4] H. Ahsan. *3D Computer Vision System for Hand Joint Motion Calculation*. PhD thesis, College of Engineering, Idaho State University, Pocatello, ID, USA, 2008.
- [5] O. Altuzarra, A. Hernandez, O. Salgado, and J. Angeles. Multiobjective optimum design of a symmetric parallel schnflies-motion generator. *ASME Journal of Mechanical Design*, 131:0310021–03100211, 2009.
- [6] O. Altuzarra, C. Pinto, B. Sandru, and A. Hernandez. Optimal dimensioning for parallel manipulators: Workspace, dexterity, and energy. *ASME Journal of Mechanical Design*, 133:0410071–0410077, 2011.
- [7] J. Angeles. *Spatial kinematic chains: analysis, synthesis, optimization*. Springer-Verlag, New York, first edition, 1982.

- [8] J. Angeles. The qualitative synthesis of parallel manipulators. In C. Gosselin and I. Ebert-Uphoff, editors, *Proceedings of the WORKSHOP on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, October 3-4, 2002, Quebec City, Quebec, Canada, 2002.
- [9] M.R. Anoop and A. Samson. Optimal synthesis of spatial mechanism using genetic algorithm. In *10th National Conference on Technological Trends (NCTT09)*, pages 26–32, Trivandrum, Kerala, India , Nov 6 - Nov 7, 2009.
- [10] M. Arsenault and R. Boudreau. The synthesis of three- degree-of-freedom planar parallel mechanisms with revolute joints (3-rrr) for an optimal singularity-free workspace. *Journal of Robotic Systems*, 21(5):259274, 2004.
- [11] J. S. Wolper B. Batbold, Y. Yihun and A. Prez-Gracia. Exact workspace synthesis for rccr linkages. *Computational Kinematicsed: Springer*,, pages 349–357, 2013.
- [12] J.E. Baker. On the motion geometry of the bennett linkage. In *Proc. 8th Internat. Conf. on Engineering Computer Graphics and Descriptive Geometry*, pages 433–437, Austin, Texas, USA,, 1998.
- [13] S. Balasubramanian, J. Klein, and E. Burdet. Robot-assisted rehabilitation of hand function. *Curr. Opin. Neurol.*, 23(6):661–670, 2010.
- [14] Batchimeg Batbold, Yimesker Yihun, James S Wolper, and Alba Pérez-Gracia. Exact workspace synthesis for rccr linkages. In *Computational Kinematics*, pages 349–357. Springer, 2014.
- [15] N. Benjuya and S. B. Kenney. Hybrid arm orthosis. *Prosthetics Orthotics*, 2:155–163, 1990.
- [16] Herr H Blaya JA. Control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *IEEE Trans Neural System Rehabilitation Engineering*, 12:24–31, 2004.

- [17] B. R. Brewer, S. K. McDowell, and L. C. Worthen-Chaudhari. Poststroke upper extremity rehabilitation: a review of robotic systems and clinical results. *Top. Stroke Rehabil.*, 14:22–44, 2007.
- [18] K. Brunnthaler. *Synthesis of 4R Linkages Using Kinematic Mapping*. PhD thesis, Institute for Basic Sciences and Engineering, University of Innsbruck, Innsbruck, Austria, 2006.
- [19] H.-J. Su C. Yue and Q. Ge. A hybrid computer-aided linkage design system for tracing open and closed planar curves. *Computer-Aided Design*, 44:1141–1150, 2012.
- [20] J.A. Cabrera, A. Simon, and M. Prado. Optimal synthesis of mechanisms with genetic algorithms. *Mechanism and Machine Theory*, 37:1165–1175, 2002.
- [21] D. Chablat and P. Wenger. Architecture optimization of a 3-dof parallel mechanism for machining applications, the orthoglide. *IEEE Transactions on Robotics and Automation*, 19(3):403–410, 2003.
- [22] M. Chalon, M. Grebenstein, T. Wimbock, and G. Hirzinger. The thumb: Guidelines for a robotic design. In *Proc. of the 2010 Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22, 2010.
- [23] C. Chen and J. Angeles. Generalized transmission index and transmission quality for spatial linkages. *Mechanism and Machine Theory*, 42:1225–1237, 2006.
- [24] W. Cloud. Man amplifiers: Machines that let you carry a ton. *Popular Science*, 187:7073, 1965.
- [25] N. I. M. Gould Conn, A. R. and Ph. L. Toint. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28:545–572, 1991.
- [26] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Prentice Hall, second edition, 1989.

- [27] L. Marchal Crespo and D. J. Reinkensmeyer. Review of control strategies for robotic movement training after neurologic injury. *J Neur Eng Reh*, submitted, 2008.
- [28] Garcia de Jalon and Bayo. *Kinematic and Dynamic Simulation of Multibody Systems*. Springer-Verlag, 1994.
- [29] Rao V Dukkupati. *Spatial mechanisms: analysis and synthesis*. CRC Press, 2001.
- [30] CF Earl and J Rooney. Some kinematic structures for robot manipulator designs. *Journal of Mechanisms, Transmissions and Automation in Design*, 105(1):15–22, 1983.
- [31] Homer D Eckhardt. *Kinematic design of machines and mechanisms*. McGraw-Hill New York, 1998.
- [32] Arthur G Erdman and George N Sandor. *Mechanism design: analysis and synthesis (Vol. 1)*. Prentice-Hall, Inc., 1997.
- [33] Jin et al. Kinematic and dynamic performance of prosthetic knee joint using six-bar mechanism. *Journal of Rehabilitation Research and Development*, 40(1):39–48, 2003.
- [34] Ian Fischer. *Dual-number methods in kinematics, statics and dynamics*. CRC Press, 1998.
- [35] E. M. Frick and J. L. Alberts. Combined use of repetitive task practice and an assistive robotic device in a patient with subacute stroke. *Physical Therapy*, 86(10):1378–1386, 2006.
- [36] Anton Gferrer. Study’s kinematic mapping - a tool for motion design. *Recent Advances in Robot Kinematics*, pages 7–16, 2000.
- [37] G. Gogu. *Structural Synthesis of Parallel Robots. Part 1: Methodology*. Springer, first edition, 2007.

- [38] D. Graupe and William K. Cline. Functional separation of emg signals via arma identification methods for prosthesis control purposes. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-5(2):252–259, March 1975.
- [39] A. Gupta and M. K. OMalley. Design of a haptic arm exoskeleton for training and rehabilitation. *IEEE/ASME Trans. on Mechatronics*, 11:280289, 2006.
- [40] Zhou Hang, Ruan Qiuqi, and Chen Houjin. A new approach of hand tracking based on integrated optical low analyse. In *Signal processing (ICSP), 2010 IEEE 10th International Conference on , Oct 2010*, pages 1194–1197, 2010.
- [41] Hansen. Synthesis of mechanisms using time-varying dimensions. *Multibody System Dynamics*, 7:127–144, 2002.
- [42] Richard S Hartenberg and Jacques Denavit. *Kinematic synthesis of linkages*. McGraw-Hill New York, 1964.
- [43] M.J.D. Hayes, T. Luu, and X.-W. Chang. Kinematic mapping application to approximate type and dimension synthesis of planar mechanisms. In J. Lenarcic and C. Galletti, editors, *Advances in Robot Kinematics*, pages 41–48, Sestri Levante, Italy, June 28 - July 1, 2004. Kluwer Academic Publishers.
- [44] Vincent Hayward, Jehangir Choksi, Gonzalo Lanvin, and Christophe Ramstein. Design and multi-objective optimization of a linkage for a haptic interface. In *Advances in Robot Kinematics and Computational Geometry*, pages 359–368. Springer, 1994.
- [45] J. M. Herve. The lie group of rigid body displacements, a fundamental tool for mechanism design. *Mechanism and Machine Theory*, 34:717–730, 1999.
- [46] M.E. Hobson and L.E. Torfason. Computer optimization of polycentric prosthetic knee mechanisms. *Bulletin of Prosthetic Research*, 10:187–201, 1975.
- [47] Hong and Erdman. A method for adjustable planar and spherical four-bar linkage synthesis. *ASME Journal of Mechanical Design*, 127 (3):456463, 2005.

- [48] C. Huang. On the finite screw system of the third order associated with a revolute-revolute chain. *ASME Journal of Mechanical Design*, 116:875–883, 1994.
- [49] C. Huang. The cylindroid associated with finite motions of the bennett mechanism. In *Proceedings of the ASME Design Engineering Technical Conferences*, 1996, Irvine, CA, USA, 1996.
- [50] C. Huang. Linear property of the screw surface of the spatial rprp linkage. *ASME Journal of Mechanical Design*, 128:581–586, 2006.
- [51] T. Huang, M. Li, X.M. Zhao, J.P. Mei, D.G. Chetwynd, and S.J. Hu. Conceptual design and dimensional synthesis for a 3-dof module of the trivariant - a novel 5-dof reconfigurable hybrid robot. *IEEE Transactions on Robotics*, 21(3):449–456, 2005.
- [52] T. Huang, L. Zhanxian, M. Li, D.G. Chetwynd, and C.M. Gosselin. Conceptual design and dimensional synthesis of a novel 2-dof translational parallel robot for pick-and-place operations. *AMSE Journal of Mechanical Design*, 126:449–455, 2004.
- [53] Z. Huang and Q. C. Li. Type synthesis of symmetrical lower-mobility parallel mechanisms using the constraint-synthesis method. *The International Journal of Robotics Research*, 22(1):59–79, 2003.
- [54] B. Hudgins, P. Parker, and R.N. Scott. A new strategy for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, 40(1):82–94, Jan 1993.
- [55] KH Hunt. Structural kinematics of in-parallel-actuated robot-arms. *Journal of mechanisms, transmissions and automation in design*, 105(4):705–712, 1983.
- [56] M.L. Husty, M. Pfurner, H.-P. Schrocker, and K. Brunthaler. Algebraic methods in mechanism analysis and synthesis. *Robotica*, 25:661–675, 2007.

- [57] Jensen and Hansen. Dimensional synthesis of spatial mechanisms and the problem of non-assembly. *Multibody System Dynamics*, 15:107–133, 2006.
- [58] Yang Jin-tang, Kong Jian-yi, and Li Gong-fa. Optimization synthesis of realizing track of rrrpr type five-bar mechanism based on genetic algorithm. In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pages 6596–6599, July 2010.
- [59] Takahashi. K. Remarks on robust extraction of hand’s silhouette for 3d hand motion capture system. In *Industrial Electronics, 2009 (IECON 09), Annual Conference of IEEE*, pages 326–2331, 2009.
- [60] G. L. Kinzel K. J. Waldron. *Kinematics, Dynamics, and Design of Machinery, 2nd Edition*.
- [61] M. Yasuda K. Watanabe K. Kiguchi, K. Iwami and T. Fukuda. An exoskeletal robot for human shoulder joint motion assist. *IEEE/ASME Trans. on Mechatronics*, 8:125–135, 2003.
- [62] Sankai Y Kawamoto H. Power assist system hal-3 for gait disorder person. In *Proceedings of the International Conference on Computers Helping People with Special Needs (ICCHP), Berlin, Germany*, volume 2398, 2002.
- [63] H. Kazerooni and S. L. Mahoney. Dynamics and control of robotic systems worn by humans. *ASME Journal of Dynamics Systems, Measurement and Control*, 113:379387, 1991.
- [64] Ben Kenwright. A beginners guide to dual-quaternions: What they are, how they work, and how to use them for 3d. In *Character Hierarchies, The 20th International Conference on Computer Graphics, Visualization and Computer Vision, WSCG 2012 Communication Proceedings*, pp. 1-13. Citeseer, 2012.
- [65] John S Ketchel and Pierre M Larochelle. Self-collision detection in spatial closed chains. *Journal of Mechanical Design*, 130(9):092305, 2008.

- [66] W. Khan and J. Angeles. The kinetostatic optimization of robotic manipulators: The inverse and the direct problems. *ASME Journal of Mechanical Design*, 128:168–178, 2006.
- [67] H.S. Kim and L.-W. Tsai. Design optimization of a cartesian parallel manipulator. *ASME Journal of Mechanical Design*, 125:43–51, 2003.
- [68] H.S. Kim and L.-W. Tsai. Kinematic synthesis of a spatial 3-rps parallel manipulator. *ASME Journal of Mechanical Design*, 125:92–97, 2003.
- [69] E.C. Kinzel, J.P. Schmiedeler, and G.R. Pennock. Kinematic synthesis for finitely separated positions using geometric constraint programming. *ASME Journal of Mechanical Design*, 128:1070–1079, 2006.
- [70] Daniel Klawitter. *Toolbox-Kinematics*. <http://www.mathworks.com/matlabcentral/fileexchange/27114-toolbox-kinematics>.
- [71] M. Kolsch and M. Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *Computer Vision and Pattern Recognition Workshop, May 27-June 2004*, 2004.
- [72] Xianwen Kong and Clment M. Gosselin. Type synthesis of three-degree-of-freedom spherical parallel manipulators. *The International Journal of Robotics Research*, 23(3):237–245, 2004.
- [73] Xianwen Kong and C.M. Gosselin. Type synthesis of 3t1r 4-dof parallel manipulators based on screw theory. *Robotics and Automation, IEEE Transactions on*, 20(2):181–190, April 2004.
- [74] A. Kosinska, M. Galicki, and K. Kedzior. Design and optimization of parameters of delta-4 parallel manipulator for a given workspace. *Journal of Robotic Systems*, 20(9):539–548, 2003.
- [75] H. I. Krebs, S. Mernoff, S. E. Fasoli, R. Hughes, J. Stein, and N. Hogan. A comparison of functional and impairment-based robotic training in severe to moderate chronic stroke: a pilot study. *NeuroRehabilitation*, 23:81–87, 2008.

- [76] A. Aristidou Lasenby. Motion capture with constrained inverse kinematics for real-time hand tracking. In *Communications, control and Signal processing (ISCCSP)*, pages 1–5, 2010.
- [77] E. Lee and C. Mavroidis. Geometric design of 3r manipulators for reaching four end-effector spatial poses. *The International Journal of Robotics Research*, 23(3):247–254, 2004.
- [78] G.F. Liu, J. Meng, J. J. Xu, and Z. X. Li. Kinematic synthesis of parallel manipulators: a lie theoretic approach. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2096–2100 vol.3, Oct 2003.
- [79] M.J.H. Lum, J. Rosen, M.N. Sinanan, and Hannaford B. Kinematic optimization of a spherical mechanism for a minimally invasive surgical robot. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 829–834 Vol.1, 2004.
- [80] Lu-Zhong Ma, An-Xin Liu, Hui ping Shen, and Ting li Yang. A method for structure synthesis of reconfigurable mechanisms based on genetic optimization algorithm. In *Reconfigurable Mechanisms and Robots, 2009. ReMAR 2009. ASME/IFTOMM International Conference on*, pages 148–152, June 2009.
- [81] Akhtar N Malik and DR Kerr. A new approach to type synthesis of spatial robotic mechanisms. *ASME Robotics, Spatial Mechanisms, and Mechanical Syst., Design Eng. Div*, 45:387–395, 1992.
- [82] Elisabeth Andr Matthias Rehm, Nikolaus Bee. Wave like an egyptian - accelerometer based gesture recognition for culture specific interactions. In *British Computer Society*, 2007.
- [83] J. M. McCarthy and G.S. Soh. *Geometric Design of Linkages*. Springer-Verlag, New York, second edition, 2010.

- [84] J Michael McCarthy and Gim Song Soh. *Geometric design of linkages*, volume 11. Springer, 2010.
- [85] J.M. McCarthy. *Introduction to Theoretical Kinematics*. The MIT Press, first edition, 1990.
- [86] J.Michael McCarthy. *21st Century Kinematics*. Springer, New York, first edition, 2012.
- [87] J. Mehrholz, T. Platz, J. Kugler, and M. Pohl. Electromechanical and robot-assisted arm training for improving arm function and activities of daily living after stroke. *Cochrane Database Syst. Rev.*, 4:CD006876, 2008.
- [88] J.-P. Merlet. Optimal design of robots. In *Proceedings of Robotics: Science and Systems*, June, 2005, Cambridge, USA, 2005.
- [89] J.P. Merlet. Jacobian, manipulability, condition number, and accuracy of parallel robots. *ASME Journal of Mechanical Design*, 128:199–206, 2006.
- [90] Minaar, Tortorelli, and Snyman. On nonassembly in the optimal dimensional synthesis of planar mechanisms. *Structural Multidisciplinary Optimization*, 21:345–354, 2001.
- [91] R. S. Mosher. Handyman to hardiman. *Society of Automotive Engineers*, MS670088, 1967.
- [92] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, 1994.
- [93] James Nielsen and Bernard Roth. *Formulation and solution for the direct and inverse kinematics problems for mechanisms and mechatronics systems*. Springer, 1998.
- [94] Saeed B Niku. *Introduction to robotics: analysis, systems, applications*, volume 7. Prentice Hall New Jersey, 2001.

- [95] S.B. Nokleby and R.P. Podhorodeski. Optimization-based synthesis of grashof geared five-bar mechanisms. *ASME Journal of Mechanical Design*, 138:529–534, 2006.
- [96] I.A. Parkin. A third conformation with the screw systems: Finite twist displacements of a directed line and point. *Mechanism and Machine Theory*, 27:177–188, 1992.
- [97] A. Perez and J. M. McCarthy. Dimensional synthesis of bennett linkages. *ASME Journal of Mechanical Design*, 125(1):98–104, 2003.
- [98] ALBA Perez and J MICHAEL McCarthy. Dimensional synthesis of spatial rr robots. In *Advances in Robot Kinematics*, pages 93–102. Springer, 2000.
- [99] A. Perez Gracia. Synthesis of spatial rprp loops for a given screw system. In Pisla, Ceccarelli, Husty, and Corves, editors, *New Trends in Mechanism Science. Proc. of the EuCoMeS, 3rd European Conference on Mechanism Science*, Cluj-Napoca, Romania, September 14-18, 2010. Springer.
- [100] A. Perez Gracia and J. M. McCarthy. The kinematic synthesis of spatial serial chains using clifford algebra exponentials. *Proceedings of the Institution of Mechanical Engineers, Part C, Journal of Mechanical Engineering Science*, 220(7):953–968, 2006.
- [101] P.Parker, K. Englehart, and B. Hudgins. Myoelectric signal processing for control of powered limb prostheses. *Journal of Electromyography and Kinesiology*, 16:541–548, 2006.
- [102] N.M. Rao and K.M. Rao. Dimensional synthesis of a 3-rps parallel manipulator for a prescribed range of motion of spherical joints. *Mechanism and Machine Theory*, 44:477–486, 2009.
- [103] B. Ravani and B. Roth. Motion synthesis using kinematic mappings. *ASME Journal of Mechanism, Transmissions, and Automation in Design*, 105:460467, 1983.

- [104] D. J. Reinkensmeyer, J. A. Galvez, L. Marchal, E. T. Wolbrecht, and J. E. Bobrow. Some key problems for robot-assisted movement therapy research: A perspective from the university of california at irvine. In ASME, editor, *Proc. 10th IEEE Intl. Conf. on Rehabilitation Robotics*, pages 1009–1015, Noordwijk, The Netherlands, 2007.
- [105] Ben Zion Sandler. *Robotics: Designing the Mechanisms for Automated Machinery*. Academic Press, 1999.
- [106] George N. Saridis and Thomas P. Gootee. Emg pattern analysis and classification for a prosthetic arm. *Biomedical Engineering, IEEE Transactions on*, BME-29(6):403–412, June 1982.
- [107] G. Schmeisser and W. Seamone. An upper limb prosthesis-orthosis power and control system with multi-level potential. *Bone Joint Surg*, 55:14931501, 1973.
- [108] H.-P. Schrocker, M. L. Husty, and J. M. McCarthy. Kinematic mapping based assembly mode evaluation of planar four-bar mechanisms. *AMSE Journal of Mechanical Design*, 129:924–929, 2007.
- [109] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2024–2030, 2006.
- [110] F. Sergi, D. Accoto, N.L. Tagliamonte, G. Carpino, S. Galzerano, and E. Guglielmelli. Kinematic synthesis, optimization and analysis of a non-anthropomorphic 2-dofs wearable orthosis for gait assistance. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4303–4308, Oct 2012.
- [111] Q. Shen, Y.M. Al-Smadi, P.J. Martin, K. Russell, and R.S. Sodhi. An extension of mechanism design optimization for motion generation. *Mechanism and Machine Theory*, 44:17591767, 2009.

- [112] L.W. Shieh, W.-B. and Tsai and S. Azarm. Design and optimization of a one-degree-of-freedom six-bar leg mechanism for a walking machine. *Journal of Robotic Systems*, 14(12):871–880, 1997.
- [113] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown University TR*, 120, 2006.
- [114] Zhibin Song and Shuxiang Guo. Force sensor-based platform for upper-limb motor function for stroke assessment. In *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pages 58–63, Dec 2010.
- [115] C. Harris Stephens and M. A combined corner and edge detector. In *the 4th Alvy Vision conference*, pages 147–151, 1988.
- [116] G. Stillfried and P. van der Smagt. Movement model of a human hand based on magnetic resonance imaging (mri). In *Proceedings of the 1st Int. Conf. on Applied Bionics and Biomechanics*, Venice, Italy, October 14-16,, 2010.
- [117] M. Stock and K. Miller. Optimal kinematic design of spatial parallel manipulators: Application to linear delta robot. *ASME Journal of Mechanical Design*, 125:292–301, 2003.
- [118] Hai-Jun Su, Peter Dietmaier, and J Michael McCarthy. Trajectory planning for constrained parallel manipulators. *Journal of mechanical design*, 125(4):709–716, 2003.
- [119] L.-W. Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Interscience, New York, 1999.
- [120] L.-W. Tsai. *Enumeration of Kinematic Structures According to Function*. CRC Press, 2000.
- [121] Irfan Ullah and Sridhar Kota. Optimal synthesis of mechanisms for path generation using fourier descriptors and global search methods. *ASME Journal of Mechanical Design*, 119(4):504–510, 1997.

- [122] David G Ullman. *The mechanical design process*, volume 2. McGraw-Hill New York, 1992.
- [123] Steve Verrill. Optimization java package, 2000.
- [124] K.J. Waldron. A study of overconstrained linkage geometry by solution of closure equations - part ii- four-bar linkages with lower pair joints other than screw joints. *Mechanism and Machine Theory*, 8:233–247, 1973.
- [125] Eric W Weisstein. Quadratic surface. *From MathWorld—A Wolfram Web Resource*. <http://mathworld.wolfram.com/QuadraticSurface.html>, 2011.
- [126] E. Wolbrecht, D.J. Reinkensmeyer, and A. Perez-Gracia. Single degree-of-freedom exoskeleton mechanism design for finger rehabilitation. In *Proceedings of the ICORR 2011: Int. Conference on Rehabilitation Robotics*, June 29–July 1, 2011, Zurich, Switzerland, 2011.
- [127] E. Wolbrecht, H.-J. Su, A. Perez, and J.M. McCarthy. Geometric design of symmetric 3-rrs constrained parallel platforms. In ASME, editor, *Proceedings of the 2004 ASME International Mechanical Engineering Congress and Exposition*, November 13–19, 2004, Anaheim, California, USA, 2004.
- [128] E. T. Wolbrecht, D. J. Reinkensmeyer, and J. E. Bobrow. Optimizing compliant, model-based robotic assistance to promote neurorehabilitation. *IEEE Transactions Neural Systems and Rehabilitation Engineering*, 16:286–297, 2008.
- [129] Ch. Wu, X-J. Liu, L. Wang, and J. Wang. Optimal design of spherical 5r parallel manipulators considering the motion/force transmissibility. *ASME Journal of Mechanical Design*, 132:0310021–03100210, 2010.
- [130] J. Wu, A. Purwar, and Q. J. Ge. Interactive dimensional synthesis and motion design of planar 6r single-loop closed chains via constraint manifold modification. *ASME Journal of Mechanisms and Robotics*, 2(3), 2010.

- [131] Yuanqing Wu, Han Ding, Jian Meng, and Zexiang Li. Lie theoretical approach to synthesizing $t(3)$ parallel kinematic manipulators. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 741–746, Aug 2005.
- [132] Zhiyong Wu, Haijun Niu, Deyu Li, Fang Pu, and Yubo Fan. A virtual semg prosthetic hand development system based on labview and pci-1710hg a/d card. In *Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on*, volume 4, pages 1743–1745, Oct 2010.
- [133] Charles W. Wampler. Solving the kinematics of planar mechanisms. In *1998 ASME Design Engineering Technical Conferences, Sept 13-16, 1998, Atlanta, Georgia, USA, 1998*.
- [134] Jianhua Xiao, Jin Xu, Zehui Shao, Congfeng Jiang, and Linqiang Pan. A genetic algorithm for solving multi-constrained function optimization problems based on ks function. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 4497–4501, Sept 2007.
- [135] Hong-Sen Yan. A methodology for creative mechanism design. *Mechanism and machine theory*, 27(3):235–242, 1992.
- [136] Yao and Angeles. Computation of all optimum dyads in the approximate synthesis of planar linkages for rigid-body guidance. *Mechanism and Machine Theory*, 35:1065–1078, 2000.
- [137] Y. Yihun, K. Bosworth, and Perez Gracia. Link-based performance optimization of spatial mechanisms. *submitted to the ASME Journal of Mechanical Design*,, 2013.
- [138] Y. Yihun, R. Miklos, A. Perez-Gracia, D.J. Reinkensmeyer, K. Denney, and E.T. Wolbrecht. Single degree-of-freedom exoskeleton mechanism design for thumb rehabilitation. In *Proc. of the 34th Annual International Conference of the IEEE EMBS*, San Diego, California USA,, 28 August - 1 September, 2012.

- [139] Yimesker Yihun, Md Shamim N Rahman, and Alba Perez-Gracia. Design of an exoskeleton as a finger-joint angular sensor. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 4176–4180. IEEE, 2012.
- [140] Ken Bosworth Yimesker Yihun and Alba Perez-Gracia. Link-based performance optimization of spatial mechanisms. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference Volume 6A: 37th Mechanisms and Robotics Conference Portland, Oregon, USA, August 47, 2013*, 2013.
- [141] H.. Zhou. Dimensional synthesis of adjustable path generation linkages using the optimal slider adjustment. *Mechanism and Machine Theory*, 2009.

Appendix

(*The optimization considers manufacturability and compactness issues by setting minimum and maximum link lengths and offsets along the axis. In addition to these constraints force transmission and obstacle avoidance constraints are included.

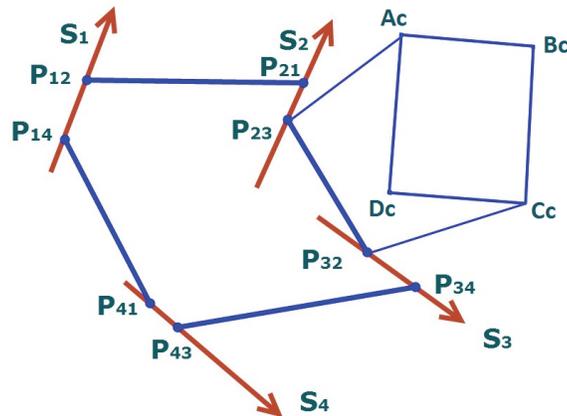
The following files are important for the operation of the algorithm. “ToMatlab.m” can be downloaded from <http://library.wolfram.com/infocenter/MathSource/577/>

<< “F:\\Matrix.m”

<< “F:\\ToMatlab.m”

The methodology is general enough to be applied for any spatial mechanism. This particular example is focused on a Bennett mechanism used as a cabinet hinge and door application.

For the optimization, the inputs can be in the form of the screw axes of the joints expressed in plücker coordinate system and their connectivity or two arbitrary points on the joint axis and their connectivity. It is always easy to go from one form of the input to the other. In this example the inputs are points on the joint axes expressed as P_{ij} where i represents the joint and j represents the preceding or the next joint.*)



$$P_{14} = \{-0.959686, -2.4095, 4.802\};$$

$$P_{12} = \{-0.658693, -1.582623, 5.277056\};$$

$$P_{21} = \{-3.97987, 2.85262, 3.70129\};$$

$$P_{23} = \{-3.392076, 3.661586, 3.70984722\};$$

$$P_{32} = \{2.00351, -0.848523, -4.02768\};$$

$$P34 = \{2.801034, -1.220379, -3.552624\};$$

$$P43 = \{0.788636, 4.83263, -2.70136\};$$

$$P41 = \{1.776294, 4.676238, -2.69280278\};$$

$$(*\text{End-effector(the door dimensions)*}) \text{Ac} = \{-4.50, 4.5, 0\};$$

$$\text{Bc} = \{4.5, 4.5, 0\};$$

$$\text{Cc} = \{4.5, -4.5, 0\};$$

$$\text{Dc} = \{-4.5, -4.5, 0\};$$

(*From the points the direction of the axis (si) and then the plücker expression of the axes (Si) can be calculated as follow:*)

$$s1 = N[(P12 - P14)/\text{Sqrt}[(P12 - P14).(P12 - P14)]];$$

$$s2 = N[(P23 - P21)/\text{Sqrt}[(P23 - P21).(P23 - P21)]];$$

$$s3 = N[(P34 - P32)/\text{Sqrt}[(P34 - P32).(P34 - P32)]];$$

$$s4 = N[(P43 - P41)/\text{Sqrt}[(P43 - P41).(P43 - P41)]]; \text{S1} = \{s1, P14 \times s1\};$$

$$\text{S2} = \{s2, P21 \times s2\};$$

$$\text{S3} = \{s3, P32 \times s3\};$$

$$\text{S4} = \{s4, P41 \times s4\};$$

(*Here, the norm of the axes is checked, i.e, si.si=1 and si.s0=0*) S1[[1]].S1[[1]];

$$\text{S2}[[1]].\text{S2}[[1]];$$

$$\text{S3}[[1]].\text{S3}[[1]];$$

$$\text{S4}[[1]].\text{S4}[[1]];$$

$$\text{S1}[[1]].\text{S1}[[2]];$$

$$\text{S2}[[1]].\text{S2}[[2]];$$

$$\text{S3}[[1]].\text{S3}[[2]];$$

$$\text{S4}[[1]].\text{S4}[[2]];$$

(*CommonNormalGen, is a function in the matrix.m file, which gives the {{Xline, a, Alpha}, {q1, q2}}; where a= the common normal length, α = the angle of the common normal with the axis, q1 a point on the first axis and q2 is a point on the second axis.*)

$$\text{P12C} = \text{CommonNormalGen}[\text{S1}, \text{S2}][[2, \{1, 2\}]] [[1]];$$

$$\text{P21C} = \text{CommonNormalGen}[\text{S1}, \text{S2}][[2, \{1, 2\}]] [[2]];$$

```

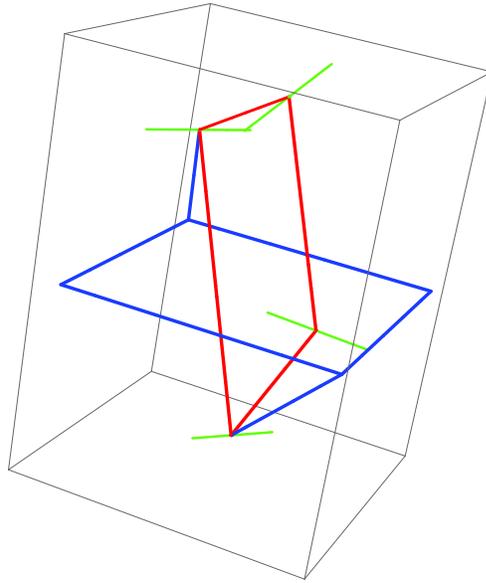
P23C = CommonNormalGen[S2, S3][[2, {1, 2}]][[1]];
P32C = CommonNormalGen[S2, S3][[2, {1, 2}]][[2]];
P34C = CommonNormalGen[S3, S4][[2, {1, 2}]][[1]];
P43C = CommonNormalGen[S3, S4][[2, {1, 2}]][[2]];
P41C = CommonNormalGen[S4, S1][[2, {1, 2}]][[1]];
P14C = CommonNormalGen[S4, S1][[2, {1, 2}]][[2]];
(*Plot the mechanism based on the common normal points obtained, for later comparisons:*)
VertexColor1 = {{Black, Black}, {Blue, Blue}, {Red, Red}, {Green, Green},
{Blue, Blue}, {Gray, Gray}, {Magenta, Magenta}, {Yellow, Yellow},
{Magenta, Magenta}, {Black, Black}, {Blue, Blue}, {Gray, Red}}; linkagemcom =
Graphics3D[List[Line[{P12C, P14C}, VertexColors → VertexColor1[[4]],
Thickness[0.006], Line[{P12C, P21C}, VertexColors → VertexColor1[[3]],
Thickness[0.005], Line[{P21C, P23C}, VertexColors → VertexColor1[[4]],
Thickness[0.005],
Line[{P12C, P12C + 2 * S1[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P12C, P12C - 2 * S1[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P21C, P21C + 2 * S2[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P21C, P21C - 2 * S2[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P32C, P32C + 2 * S3[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P32C, P32C - 2 * S3[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P43C, P43C + 2 * S4[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P43C, P43C - 2 * S4[[1]]}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P32C, P34C}, VertexColors → VertexColor1[[4]], Thickness[0.005],
Line[{P43C, P41C}, VertexColors → VertexColor1[[4]], Thickness[0.007],
Line[{P23C, P32C}, VertexColors → VertexColor1[[3]], Thickness[0.007],
Line[{P34C, P43C}, VertexColors → VertexColor1[[3]], Thickness[0.007],
Line[{P41C, P14C}, VertexColors → VertexColor1[[3]], Thickness[0.007],
Line[{Ac, Bc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{Bc, Cc}, VertexColors → VertexColor1[[5]], Thickness[0.007],

```

```

Line[{Cc, Dc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{Dc, Ac}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{P23C, Ac}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{P32C, Cc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
AxesLabel → {X, Y, Z}] comntotlength =

```



```

Chop[(P12C - P14C).(P12C - P14C) + (P21C - P12C).(P21C - P12C)+
(P23C - P21C).(P23C - P21C) + (P32C - P23C).(P32C - P23C)+
(P34C - P32C).(P34C - P32C) + (P43C - P34C).(P43C - P34C)+
(P41C - P43C).(P41C - P43C) + (P14C - P41C).(P14C - P41C)]
Sqrt[(P12C - P14C).(P12C - P14C)];
Sqrt[(P23C - P21C).(P23C - P21C)];
Sqrt[(P34C - P32C).(P34C - P32C)];
Sqrt[(P41C - P43C).(P41C - P43C)];
Sqrt[(P21C - P12C).(P21C - P12C)];
Sqrt[(P32C - P23C).(P32C - P23C)];
Sqrt[(P43C - P34C).(P43C - P34C)];
Sqrt[(P14C - P41C).(P14C - P41C)];

```

(*In this particular example, the mechanism formed by connecting the common normal is more compact, however, the dimensions are too small for manufacturing

and assembling. For instance the placement of the joints along the axis is almost zero. So we need to consider minimum and maximum lengths, in this case we may end up with a less compact but applicable mechanism.

Let S_i be screw axis and C_i be the point on S_i at which the line from the origin and S_i be perpendicular, where $i = 1 \dots n$,

P_{ij} and $P_{i(j+1)}$ are two arbitrary points on axis S_i , and t_{ij} and $t_{i(j+1)}$ are corresponding distances of P_{ij} and $P_{i(j+1)}$ from point C_i respectively.

The procedure is as follow: express $S_i = s_1 + \epsilon s_0$ in dual Quaternional form and find $C_i = (s_1 x s_0) / (s_1 . s_1)$. Then write P_{ij} and $P_{i(j+1)}$ as a function of C_i , t_{ij} and $t_{i(j+1)}$ like $P_{ij} = C_i + t_{ij} S_i$ and $P_{i(j+1)} = C_i + t_{i(j+1)} S_i$, then write the cost function as a norm to minimize length

Here the Points ($C_1, C_2, C_3 \dots C_6$) are calculated for the first configuration, then based on these points, the mechanism will be defined as a function of t_{12}, t_{23} etc. These variables will be optimized based on the subsequence positions of points (P_{12}, p_{23} etc.)*)

$$C_1 = (S_1[[1]] \times S_1[[2]]) / (S_1[[1]].S_1[[1]]);$$

$$C_2 = (S_2[[1]] \times S_2[[2]]) / (S_2[[1]].S_2[[1]]);$$

$$C_3 = (S_3[[1]] \times S_3[[2]]) / (S_3[[1]].S_3[[1]]);$$

$$C_4 = (S_4[[1]] \times S_4[[2]]) / (S_4[[1]].S_4[[1]]);$$

$$P_{12n} = C_1 + t_{12} * S_1[[1]];$$

$$P_{14n} = C_1 + t_{14} * S_1[[1]];$$

$$P_{21n} = C_2 + t_{21} * S_2[[1]];$$

$$P_{23n} = C_2 + t_{23} * S_2[[1]];$$

$$P_{32n} = C_3 + t_{32} * S_3[[1]];$$

$$P_{34n} = C_3 + t_{34} * S_3[[1]];$$

$$P_{43n} = C_4 + t_{43} * S_4[[1]];$$

$$P_{41n} = C_4 + t_{41} * S_4[[1]];$$

$$\text{ObjFunc} = \text{Chop}[(P_{12n} - P_{14n}).(P_{12n} - P_{14n}) + (P_{21n} - P_{12n}).(P_{21n} - P_{12n}) +$$

```
(P23n - P21n).(P23n - P21n) + (P32n - P23n).(P32n - P23n)+
(P34n - P32n).(P34n - P32n) + (P43n - P34n).(P43n - P34n)+
(P41n - P43n).(P41n - P43n) + (P14n - P41n).(P14n - P41n)+
(P23n - Ac).(P23n - Ac) + (P32n - Cc).(P32n - Cc)];
```

```
ObjectiveFunfile = FileNameJoin[{"F:\\CabinetDesign", "mycabfun.m"}];
RRRROb = OpenWrite[ObjectiveFunfile, PageWidth -> 300]
OutputStream[F:\\CabinetDesign\\mycabfun.m, 79] WriteString[RRRROb,
"% The objective function for the optimization problem. It is formed as a square
of link lengths and offset length of the spatial mechanism \\n\\n"]
WriteString[RRRROb, "function f = mycabfun(x) \\n\\n"]
(*WriteString[sRR, "K := 40 \\n\\n"]*)
WriteString[RRRROb,
"t14=x(1);t12=x(2); t21=x(3);t23=x(4);t32=x(5); t34=x(6);t43=x(7);t41=x(8);
\\n\\n"]
WriteMatlab[ObjFunc, RRRROb, f]
Close[RRRROb] F:\\CabinetDesign\\mycabfun.m (*To minimize the distance be-
tween points in the same axis as well as the links between each axis:*) lequ1 = t12 - t14 == b1;
lequ2 = t23 - t21 == b2;
lequ3 = t34 - t32 == b3;
lequ4 = t41 - t43 == b4; Aeqn =
(Normal[CoefficientArrays[{lequ1, lequ2, lequ3, lequ4}, {t12, t14, t23, t21, t34, t32, t41, t43}]] [[
2]]); b = 5;
beqn = {b, b, b, b}; Lmin = 20;
Lmax = 45; noncon1 = Chop[-((P21n - P12n).(P21n - P12n) - Lmin^2)];
noncon2 = Chop[-((P32n - P23n).(P32n - P23n) - Lmin^2)];
noncon3 = Chop[-((P43n - P34n).(P43n - P34n) - Lmin^2)];
noncon4 = Chop[-((P14n - P41n).(P14n - P41n) - Lmin^2)];
noncon5 = Chop[-((P21n - P12n).(P21n - P12n) - Lmax^2)];
noncon6 = Chop[-((P32n - P23n).(P32n - P23n) - Lmax^2)];
```

```

noncon7 = Chop[-((P43n - P34n).(P43n - P34n) - Lmax^2)];
noncon8 = Chop[-((P14n - P41n).(P14n - P41n) - Lmax^2)];
(*Force Transmission optimization*)

acR = N[Pi] * 30/180;
acP = N[Pi] * 75/180;

fc1 = -(((P21n - P12n)/Sqrt[(P21n - P12n).(P21n - P12n)]).S2[[1]]) - Cos[acR];
fc2 = -(((P32n - P23n)/Sqrt[(P32n - P23n).(P32n - P23n)]).S3[[1]]) - Cos[acR];
fc3 = -(((P43n - P34n)/Sqrt[(P43n - P34n).(P43n - P34n)]).S4[[1]]) - Cos[acR];

hhh = Table[conx[i], {i, 1, 50}]

```

(*In the obstacle avoidance, it is required to consider more positions along the trajectory. If we don't have enough points or we don't have the trajectory, it is required to interpolate the task positions to get more intermediate positions. Or we need to solve the closed form solution to get the joint variables at each position.

Once we have these more positions, inverse kinematics will be applied to obtain the lists of the subsequent joint angles.)* $\theta z2 = 32 * \text{Pi}/180$;

$\theta z3 = 45 * \text{Pi}/180$;

$\beta x3 = \text{Pi}/2$;

$N[\text{RotationZ}[\theta z2]]$;

$\text{Po1} = \{\{1, 0, 0, 0\}, \{0, 1, 0, 0\}, \{0, 0, 1, 0\}, \{0, 0, 0, 1\}\}$;

$\text{Po2} = N[\{\{\text{Cos}[(8\pi)/45], -\text{Sin}[(8\pi)/45], 0, 0\}, \{\text{Sin}[(8\pi)/45], \text{Cos}[(8\pi)/45], 0, 0\}, \{0, 0, 1, 5\}, \{0, 0, 0, 1\}\}]$;

$\text{Po3} = N[\{\{\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}, 5\}, \{\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}, 5\}, \{0, 1, 0, 1\}, \{0, 0, 0, 1\}\}]$;

$\text{homPose} = \{\text{Po1}, \text{Po2}, \text{Po3}\}$;

$\text{dqPoseIn} = \text{Table}[\text{HM2dq}[\text{homPose}[[i]]], \{i, 1, \text{Length}[\text{homPose}]\}]$;

$\text{p3in} = \{\{-1, -1, 1\}, \{-0.3181, -1.3780, 6.0000\}, \{5.0000, 3.5858, 0.0000\},$

$\{-1.0806, -1.0622, 1.4950\}, \{-1.1503, -1.1206, 2.0877\}, \{-1.1903, -1.1747, 2.7996\},$

$\{-1.1618, -1.2267, 3.6508\}, \{-0.9858, -1.2847, 4.6466\}, \{-0.5105, -1.3583, 5.7337\}\}$;

```

p6in = {{-1, 1, -1}, {-1.3780, 0.3181, 4.0000}, {3.5858, 5.0000, 2.0000},
{-1.1296, 0.8735, -0.5668}, {-1.2684, 0.7476, -0.0326}, {-1.4086, 0.6270, 0.6302},
{-1.5298, 0.5169, 1.4543}, {-1.5827, 0.4203, 2.4702}, {-1.4585, 0.3365, 3.6772}};
p7in = {{-1, -1, -1}, {-0.3181, -1.3780, 4.0000}, {3.5858, 5.0000, 0.0000},
{-1.0535, -1.1240, -0.5039}, {-1.0942, -1.2409, 0.0921}, {-1.1055, -1.3419, 0.8084},
{-1.0556, -1.4149, 1.6625}, {-0.8830, -1.4441, 2.6556}, {-0.4755, -1.4043, 3.7345}};
p8in = {{1, -1, -1}, {1.3780, -0.3181, 4.0000}, {5.0000, 6.4142, 0.0000},
{0.9448, -1.0472, -0.4792}, {0.8974, -1.0635, 0.1375}, {0.8696, -1.0325, 0.8666},
{0.8845, -0.9324, 1.7205}, {0.9878, -0.7380, 2.6956}, {1.2659, -0.4208, 3.7423}};
Pmi = {{-0.1329, -0.0679, 0.6099}, {-0.2650, -0.1115, 1.3806}, {-0.3654, -0.1178, 2.3575},
{-0.3489, -0.0791, 3.5784}, {1.0828, 0.1712, 6.2500}, {2.9853, 0.7959, 6.3658},
{4.5889, 2.2245, 4.8947}, {5.1136, 3.8541, 2.8452}};
xm = Table[(p8in[[i]] - p7in[[i]])/Sqrt[(p8in[[i]] - p7in[[i]])*(p8in[[i]] - p7in[[i]])],
{i, 1, Length[Pmi]}];
ym = Table[(p6in[[i]] - p7in[[i]])/Sqrt[(p6in[[i]] - p7in[[i]])*(p6in[[i]] - p7in[[i]])],
{i, 1, Length[Pmi]}];
zm = Table[(p3in[[i]] - p7in[[i]])/Sqrt[(p3in[[i]] - p7in[[i]])*(p3in[[i]] - p7in[[i]])],
{i, 1, Length[Pmi]}];
homeint =
Table[{{xm[[i, 1]], ym[[i, 1]], zm[[i, 1]], Pmi[[i, 1]]},
{xm[[i, 2]], ym[[i, 2]], zm[[i, 2]], Pmi[[i, 2]]}, {xm[[i, 3]], ym[[i, 3]], zm[[i, 3]], Pmi[[i, 3]]},
{0, 0, 0, 1}}, {i, 1, Length[xm]}];
xplot =
Table[
{Thickness[0.01],
Line[{homPose[[i, {1, 2, 3}, 4]], homPose[[i, {1, 2, 3}, 4]] + 2 * homPose[[i, {1, 2, 3}, 1]]},
VertexColors -> {Blue}}], {i, 1, Length[homPose]}];

```

```

yplot =
Table[
{Thickness[0.01],
Line[{homPose[[i, {1, 2, 3}, 4]], homPose[[i, {1, 2, 3}, 4]] + 2 * homPose[[i, {1, 2, 3}, 2]]},
VertexColors → {Blue → Placeholder}], {i, 1, Length[homPose]};
zplot =
Table[
{Thickness[0.01],
Line[{homPose[[i, {1, 2, 3}, 4]], homPose[[i, {1, 2, 3}, 4]] + 2 * homPose[[i, {1, 2, 3}, 3]]},
VertexColors → {Blue}], {i, 1, Length[homPose]};
plotIndexA =
Table[Graphics3D[{xplot[[j]], yplot[[j]], zplot[[j]]}, AspectRatio → Automatic,
Axes → True, AxesLabel → z], {j, 1, Length[xplot]};
Show[plotIndexA]

xmplot =
Table[
{Thickness[0.01],
Line[{homeint[[i, {1, 2, 3}, 4]], homeint[[i, {1, 2, 3}, 4]] + 2 * homeint[[i, {1, 2, 3}, 1]]},
VertexColors → {Red}], {i, 1, Length[homeint]};
ymplot =
Table[
{Thickness[0.01],
Line[{homeint[[i, {1, 2, 3}, 4]], homeint[[i, {1, 2, 3}, 4]] + 2 * homeint[[i, {1, 2, 3}, 2]]},
VertexColors → {Red → Placeholder}], {i, 1, Length[homeint]};
zmplot =
Table[

```

```

{Thickness[0.01],
Line[{homeint[[i, {1, 2, 3}, 4]], homeint[[i, {1, 2, 3}, 4]] + 2 * homeint[[i, {1, 2, 3}, 3]]},
VertexColors → {Red}}], {i, 1, Length[homeint]}};
plotIndexAm =
Table[Graphics3D[{xmp[plot[[j]], ymp[plot[[j]], zmp[plot[[j]]], AspectRatio → Automatic,
Axes → True, AxesLabel → z], {j, 1, Length[xmp]}}];
Show[plotIndexAm, plotIndexA]

```

```

dqPoseInN = Table[HM2dq[homeint[[i]], {i, 1, Length[homeint]}}];
homPoseRel = Table[Flatten[quatmultSep[dqPoseInN[[i]], dqPoseIn[[1]]],
{i, 1, Length[dqPoseInN]}}];
S1q = makequate[S1,  $\theta_1$ , 0];
S2q = makequate[S2,  $\theta_2$ , 0];
S3q = makequate[S3,  $\theta_3$ , 0];
S4q = makequate[S4,  $\theta_4$ , 0];
Vvector2Ra = Chop[Collect[Expand[quatmult[S1q, S2q]],  $\epsilon$ ]];
Vvector2R = sepQ[Vvector2Ra];
Vvector2R2 = Chop[Collect[Expand[quatmult[S4q, S3q]],  $\epsilon$ ]];
Vvector2R2 = sepQ[Vvector2R2];
col1 = Coefficient[Vvector2R, Cos[ $\theta_2/2$ ]Sin[ $\theta_1/2$ ]];
col2 = Coefficient [Vvector2R, Cos [ $\frac{\theta_1}{2}$ ] Sin [ $\frac{\theta_2}{2}$ ]] ;
col3 = Coefficient [Vvector2R, Sin [ $\frac{\theta_1}{2}$ ] Sin [ $\frac{\theta_2}{2}$ ]] ;
col4 = Coefficient [Vvector2R, Cos [ $\frac{\theta_1}{2}$ ] Cos [ $\frac{\theta_2}{2}$ ]] ;
col1n = Coefficient[Vvector2R2, Cos[ $\theta_3/2$ ]Sin[ $\theta_4/2$ ]];
col2n = Coefficient [Vvector2R2, Cos [ $\frac{\theta_4}{2}$ ] Sin [ $\frac{\theta_3}{2}$ ]] ;
col3n = Coefficient [Vvector2R2, Sin [ $\frac{\theta_4}{2}$ ] Sin [ $\frac{\theta_3}{2}$ ]] ;

```

```

col4n = Coefficient [Vvector2R2, Cos [ $\frac{\theta4}{2}$ ] Cos [ $\frac{\theta3}{2}$ ]];
mat = Transpose[{col1, col2, col3, col4}];
matn = Transpose[{col1n, col2n, col3n, col4n}];
sol1 = Table[PseudoInverse[mat].homPoseRel[[i]], {i, 1, 8}];
sol2 = Table[PseudoInverse[matn].homPoseRel[[i]], {i, 1, 8}];
theta1 = Table[ArcTan[sol1[[i, 1]]/sol1[[i, 1]]], {i, 1, 8}];
theta2 = Table[ArcTan[sol1[[i, 3]]/sol1[[i, 1]]], {i, 1, 8}];
theta3 = Table[ArcTan[sol2[[i, 1]]/sol2[[i, 1]]], {i, 1, 8}];
theta4 = Table[ArcTan[sol2[[i, 3]]/sol2[[i, 1]]], {i, 1, 8}];
myanglesCab1 = Table[{2 * theta1[[i]], 0}, {2 * theta2[[i]], 0}, {i, 1, Length[theta1]}];
myanglesCab2 = Table[{2 * theta3[[i]], 0}, {2 * theta4[[i]], 0}, {i, 1, Length[theta2]}];
myaxescab1 = {S1, S2};
myaxescab2 = {S3, S4};
S1nn = Table[makequat[S1, theta1[[i]], 0], {i, 1, 8}];
S2nn = Table[makequat[S2, theta2[[i]], 0], {i, 1, 8}];
quatmult[S1nn, S2nn]
myref = homPose[[1]];
chainPoses = AnalyzeC[myaxescab1, myanglesCab1, myref]
xplot =
Table[
Line[{chainPoses[[i, {1, 2, 3}, 4]],
chainPoses[[i, {1, 2, 3}, 4]] + 2 * chainPoses[[i, {1, 2, 3}, 1]]}, VertexColors  $\rightarrow$  {Red}],
{i, 1, Length[chainPoses]}];
yplot =
Table[
Line[{chainPoses[[i, {1, 2, 3}, 4]],
chainPoses[[i, {1, 2, 3}, 4]] + 2 * chainPoses[[i, {1, 2, 3}, 2]]}, VertexColors  $\rightarrow$  {Green}],

```

```

{i, 1, Length[chainPoses]};
zplot =
Table[
Line[{chainPoses[[i, {1, 2, 3}, 4]],
chainPoses[[i, {1, 2, 3}, 4]] + 2 * chainPoses[[i, {1, 2, 3}, 3]]}, VertexColors → {Blue}],
{i, 1, Length[chainPoses]};
plotIndexChain =
Table[Graphics3D[{Thick, xplot[[j]], yplot[[j]], zplot[[j]]}, AspectRatio → Automatic],
{j, 1, Length[xplot]};
Show[plotIndexChain]

```

```

 $\theta_{1s} = \text{theta1};$ 

```

```

 $\theta_{2s} = \text{theta2};$ 

```

```

 $\theta_{3s} = \text{theta3};$ 

```

```

 $\theta_{4s} = \text{theta4};$ 

```

```

S1q = Table[sepQ2[makeequate[S1,  $\theta_{1s}[[i]]$ , 0]], {i, 1, Length[theta1]};

```

```

S2q = Table[sepQ2[makeequate[S2,  $\theta_{2s}[[i]]$ , 0]], {i, 1, Length[theta1]};

```

```

S3q = Table[sepQ2[makeequate[S3,  $\theta_{3s}[[i]]$ , 0]], {i, 1, Length[theta1]};

```

```

S4q = Table[sepQ2[makeequate[S4,  $\theta_{4s}[[i]]$ , 0]], {i, 1, Length[theta1]};

```

```

S2l = {Append[S2[[1]], 0], Append[S2[[2]], 0]};

```

```

S3l = {Append[S3[[1]], 0], Append[S3[[2]], 0]};

```

```

S4l = {Append[S4[[1]], 0], Append[S4[[2]], 0]};

```

```

S2m = Chop[Table[quatmultSep[quatmultSep[S1q[[i]], S2l], quatconj[S1q[[i]]]][{1, 2}, {1, 2, 3}],
{i, 1, Length[theta1]};

```

```

S3m = Chop[Table[quatmultSep[quatmultSep[S4q[[i]], S3l], quatconj[S4q[[i]]]][{1, 2}, {1, 2, 3}],
{i, 1, Length[theta1]};

```

```

Pq12 = {{0, 0, 0, 1}, Append[P12n, 0]};
Pq21 = {{0, 0, 0, 1}, Append[P21n, 1]};
Pq23 = {{0, 0, 0, 1}, Append[P23n, 1]};
Pq32 = {{0, 0, 0, 1}, Append[P32n, 1]};
Pq34 = {{0, 0, 0, 1}, Append[P34n, 1]};
Pq43 = {{0, 0, 0, 1}, Append[P43n, 1]};

P12m = Table[quatmultSep[quatmultSep[S1q[[i]], Pq12], quatconj4[S1q[[i]]][[{1, 2}, {1, 2, 3}]],
{i, 1, Length[theta1]};
P21m =
Chop[Table[quatmultSep[quatmultSep[S1q[[i]], Pq21], quatconj4[S1q[[i]]][[{1, 2}, {1, 2, 3}]],
{i, 1, Length[theta1]};
P23m =
Chop[Table[quatmultSep[quatmultSep[S1q[[i]], Pq23], quatconj4[S1q[[i]]][[{1, 2}, {1, 2, 3}]],
{i, 1, Length[theta1]};
P32m =
Chop[Table[quatmultSep[quatmultSep[S4q[[i]], Pq32], quatconj[S4q[[i]]][[{1, 2}, {1, 2, 3}]],
{i, 1, Length[theta1]};
P34m =
Chop[Table[quatmultSep[quatmultSep[S4q[[i]], Pq34], quatconj[S4q[[i]]][[{1, 2}, {1, 2, 3}]],
{i, 1, Length[theta1]};
P43m = Table[P43n, {i, 1, Length[theta1]};
P41m = Table[P41n, {i, 1, Length[theta1]};
P14m = Table[P14n, {i, 1, Length[theta1]};
pointD = {{0, 0, 5}, {5, 5, 1}};
Sc = {0, 0, -5.25};
Rc = 6.304;
surfcon1 = Table[-((P12m[[i, 2]] - Sc).(P12m[[i, 2]] - Sc)) + Rc^2, {i, 1, Length[theta1]};

```

```

surfcon2 = Table[-((P21m[[i, 2]] - Sc).(P21m[[i, 2]] - Sc)) + Rc^2, {i, 1, Length[theta1]};
surfcon3 = Table[-((P23m[[i, 2]] - Sc).(P23m[[i, 2]] - Sc)) + Rc^2, {i, 1, Length[theta1]};
surfcon4 = Table[-((P32m[[i, 2]] - Sc).(P32m[[i, 2]] - Sc)) + Rc^2, {i, 1, Length[theta1]};
surfcon5 = Table[-((P34m[[i, 2]] - Sc).(P34m[[i, 2]] - Sc)) + Rc^2, {i, 1, Length[theta1]};
surfcon6 = Table[-((P43m[[i]] - Sc).(P43m[[i]] - Sc)) + Rc^2, {i, 1, Length[theta1]};
surfcon7 = Table[-((P41m[[i]] - Sc).(P41m[[i]] - Sc)) + Rc^2, {i, 1, Length[theta1]};
surfcon8 = Table[-((P14m[[i]] - Sc).(P14m[[i]] - Sc)) + Rc^2, {i, 1, Length[theta1]};

surfcon9 =
Table[
-(Sqrt[(Sc - P21m[[i, 2]]) × (Sc - P12m[[i, 2]])].(Sc - P21m[[i, 2]]) × (Sc - P12m[[i, 2]])/
Sqrt[(P21m[[i, 2]] - P12m[[i, 2]])].(P21m[[i, 2]] - P12m[[i, 2]]) + Rc^2,
{i, 1, Length[theta1]};

surfcon10 =
Table[
-(Sqrt[(Sc - P23m[[i, 2]]) × (Sc - P32m[[i, 2]])].(Sc - P23m[[i, 2]]) × (Sc - P32m[[i, 2]])/
Sqrt[(P23m[[i, 2]] - P32m[[i, 2]])].(P23m[[i, 2]] - P32m[[i, 2]]) + Rc^2,
{i, 1, Length[theta1]};

surfcon11 =
Table[
-(Sqrt[(Sc - P43m[[i, 2]]) × (Sc - P34m[[i, 2]])].(Sc - P43m[[i, 2]]) × (Sc - P34m[[i, 2]])/
Sqrt[(P43m[[i, 2]] - P34m[[i, 2]])].(P43m[[i, 2]] - P34m[[i, 2]]) + Rc^2,
{i, 1, Length[theta1]};

surfcon12 =
Table[
-(Sqrt[(Sc - P41m[[i, 2]]) × (Sc - P14m[[i, 2]])].(Sc - P41m[[i, 2]]) × (Sc - P14m[[i, 2]])/
Sqrt[(P41m[[i]] - P14m[[i]])].(P41m[[i]] - P14m[[i]]) + Rc^2, {i, 1, Length[theta1]};

nonconset = {noncon1, noncon2, noncon3, noncon4, noncon5, noncon6, noncon7,

```

```

noncon8, fc1, fc2, fc3, surfcon1[[1]], surfcon1[[2]], surfcon2[[1]], surfcon2[[2]],
surfcon3[[1]], surfcon3[[2]], surfcon4[[1]], surfcon4[[2]], surfcon5[[1]],
surfcon5[[2]], surfcon6[[1]], surfcon6[[2]], surfcon7[[1]], surfcon7[[2]],
surfcon8[[1]], surfcon8[[2]], surfcon9[[1]], surfcon9[[2]], surfcon10[[1]],
surfcon10[[2]], surfcon11[[1]], surfcon11[[2]], surfcon12[[1]], surfcon12[[2]]};
cset = Table[hhh[[i]], {i, 1, Length[nonconset]}];
NonlConstfile = FileNameJoin[{“F:\\CabinetDesign”, “myconcab.m”}];
RRRR = OpenWrite[NonlConstfile, PageWidth → 300]
OutputStream[F:\\CabinetDesign\\myconcab.m, 88]
WriteString[RRRR,
“%The set of nonlinear constraints applied to the optimization of a four bar spatial
mechanism \n\n”]
WriteString[RRRR, “function[c,ceq]=myconcab(x) \n\n”]
(*WriteString[sRRR, “K := 40 \n\n”]*)
WriteString[RRRR,
“t14=x(1);t12=x(2); t21=x(3);t23=x(4);t32=x(5); t34=x(6);t43=x(7);t41=x(8);
\n\n”]
Table[WriteMatlab[nonconset[[i]], RRRR, hhh[[i]], {i, 1, Length[nonconset]}];
WriteMatlab[cset, RRRR, c]
WriteString[RRRR, “ceq=[];”]
Close[RRRR] F:\\CabinetDesign\\myconcab.m

```

(*Writing the linear constraint equations and the genetic algorithm executable file into the Matlab file. “mycabrun.m” is the one we need to run it in matlab to get the solution.*)

```

xinitial = RandomReal[{-4000, 4000}, 8]
{1058.23, 847.305, 1034.41, -1359.03, 808.926, 3277.65, -609.296, 2319.75}
linConstfile = FileNameJoin[{“F:\\CabinetDesign”, “mycabrun.m”}];
RRRRlin = OpenWrite[linConstfile, PageWidth → 300]
OutputStream[F:\\CabinetDesign\\mycabrun.m, 90]

```

```

WriteMatlab[Aeqn, RRRRlin, Aeq]
WriteMatlab[beqn, RRRRlin, beq]
WriteMatlab[xinitial, RRRRlin, x0]
WriteString[RRRRlin,
"options=gaoptimset('PlotFcns',{@gaplotbestf,@gaplotmaxconstr},'Display','iter','HybridFcn',
@fmincon);\n\n"];
WriteString[RRRRlin,
"[x,fval,exitflag,output,lambda]=ga(@mycabfun,8,[],[],Aeq,beq,[],[],@myconcab,options)\n\n"]
WriteString[RRRRlin, "save('datax','x')\n\n"]
Close[RRRRlin]

F:\CabinetDesign\mycabrun.m

```

(*The solution obtained from matlab is imported in the following steps and the mechanism is plotted*)

```

x = Flatten[Import["F:\CabinetDesign\datax.mat"]]
C1 = (S1[[1]] × S1[[2]])/(S1[[1]].S1[[1]]);
C2 = (S2[[1]] × S2[[2]])/(S2[[1]].S2[[1]]);
C3 = (S3[[1]] × S3[[2]])/(S3[[1]].S3[[1]]);
C4 = (S4[[1]] × S4[[2]])/(S4[[1]].S4[[1]]);
P12nn = C1 + t12 * S1[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],
t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]]};
P21nn = C2 + t21 * S2[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],
t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]]};
P23nn = C2 + t23 * S2[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],
t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]]};
P32nn = C3 + t32 * S3[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],
t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]]};
P34nn = C3 + t34 * S3[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],

```

```

t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]];
P43nn = C4 + t43 * S4[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],
t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]];
P41nn = C4 + t41 * S4[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],
t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]];
P14nn = C1 + t14 * S1[[1]]/.{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]],
t32 → x[[5]], t34 → x[[6]], t43 → x[[7]], t41 → x[[8]];
linkagemcom =
Graphics3D[List[Line[{P12nn, P14nn}, VertexColors → VertexColor1[[4]],
Thickness[0.006], Line[{P12nn, P21nn}, VertexColors → VertexColor1[[1]],
Thickness[0.006], Line[{P21nn, P23nn}, VertexColors → VertexColor1[[4]],
Thickness[0.006],
Line[{P32nn, P34nn}, VertexColors → VertexColor1[[4]], Thickness[0.006],
Line[{P43nn, P41nn}, VertexColors → VertexColor1[[4]], Thickness[0.006],
Line[{P23nn, P32nn}, VertexColors → VertexColor1[[1]], Thickness[0.006],
Line[{P34nn, P43nn}, VertexColors → VertexColor1[[1]], Thickness[0.006],
Line[{P41nn, P14nn}, VertexColors → VertexColor1[[1]], Thickness[0.006],
Line[{Ac, Bc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{Bc, Cc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{Cc, Dc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{Dc, Ac}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{P23nn, Bc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
Line[{P32nn, Dc}, VertexColors → VertexColor1[[5]], Thickness[0.007],
AxesLabel → {X, Y, Z}]
Points = {P14nn, P12nn, P21nn, P23nn, P32nn, P34nn, P43nn, P41nn}/.
{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]], t32 → x[[5]], t34 → x[[6]],
t43 → x[[7]], t41 → x[[8]]}

```

Func =

$$\begin{aligned} & \text{Chop}[(P12nn - P14nn).(P12nn - P14nn) + (P21nn - P12nn).(P21nn - P12nn) + \\ & (P23nn - P21nn).(P23nn - P21nn) + (P32nn - P23nn).(P32nn - P23nn) + \\ & (P34nn - P32nn).(P34nn - P32nn) + (P43nn - P34nn).(P43nn - P34nn) + \\ & (P41nn - P43nn).(P41nn - P43nn) + (P14nn - P41nn).(P14nn - P41nn)]/. \end{aligned}$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

$$\text{Sqrt}[(P12nn - P14nn).(P12nn - P14nn)]/.$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

$$\text{Sqrt}[(P23nn - P21nn).(P23nn - P21nn)]/.$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

$$\text{Sqrt}[(P34nn - P32nn).(P34nn - P32nn)]/.$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

$$\text{Sqrt}[(P41nn - P43nn).(P41nn - P43nn)]/.$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

$$\text{Sqrt}[(P21nn - P12nn).(P21nn - P12nn)]/.$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

$$\text{Sqrt}[(P32nn - P23nn).(P32nn - P23nn)]/.$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

$$\text{Sqrt}[(P43nn - P34nn).(P43nn - P34nn)]/.$$

$$\{t14 \rightarrow x[[1]], t12 \rightarrow x[[2]], t21 \rightarrow x[[3]], t23 \rightarrow x[[4]], t32 \rightarrow x[[5]], t34 \rightarrow x[[6]], \\ t43 \rightarrow x[[7]], t41 \rightarrow x[[8]]\}$$

t43 → x[[7]], t41 → x[[8]]}

Sqrt[(P14nn - P41nn).(P14nn - P41nn)]/.

{t14 → x[[1]], t12 → x[[2]], t21 → x[[3]], t23 → x[[4]], t32 → x[[5]], t34 → x[[6]],

t43 → x[[7]], t41 → x[[8]]}

(*Once the mechanism fulfills the requirement the next step is to model the mechanism in solidworks. In this case we need the points on the axes, and exporting the points to excel will simplify for latter use in solidworks.*)

Export["F:\\CabinetDesign\\fourbarpoints.xls", Points]

F:\\CabinetDesign\\fourbarpoints.xls

Curriculum Vitae

Yimesker Siraw Yihun

Education

- 05/2014** **Ph.D.** in Engineering and Applied Science.
Mechanical Engineering Department, Idaho State University (ISU).
Pocatello, ID
Advisor: Dr.Alba Perez.
- 08/2007** **M-Tech.** in Mechanical Engineering Design.
Mechanical Engineering Department, Indian Institute of Technology
Bombay (IIT Bombay).
Mumbai, India
Advisor: Prof.C.Amarnath.
- 06/2004** **B.Sc.** in Mechanical Engineering.
Mechanical Engineering Department, Bahir Dar University(BDU).
Bahir Dar, Ethiopia

Experience

- 08/2010-05/2014** Measurement and Control Research Center, ISU, Ph.D. advisor:
Dr.Alba Perez: Research Assistant- Design of spatial mechanisms and Exoskeleton devices.
- 08/2012-05/2013** Dept. of Mechanical Engineering, ISU, Teaching Assistant.
- 11/2009-08/2010** University of Salerno, Salerno, Italy: Research Assistant- Characterization of a new generation of lubricants incorporating inorganic fullerenes and other nano- materials.
- 07/2007-10/2010** Dept. of Mechanical Engineering, BDU, Lecturer, Courses taught:
Theory of Mechanisms and Machinery, Introduction to Vibration, Engineering Mechanics, Machine Drawing, Instrumentation and Process Control

Awards and Scholarships

- Selected for the NSF RI research grant at Idaho State University, Pocatello, Idaho 2013.
- Selected for the graduate teaching assistant position, Idaho State University, Pocatello, Idaho 2012.
- Selected for the prosthetic hand grant at Idaho State University, Pocatello, Idaho 2010.

- Selected for the Add Nano research grant, at University of Salerno, Italy 2009.
- Selected by Ethiopian Government and IIT Bombay, India for a scholarship grant in 2005.

Certificate Training

- Screw-Theory Based Methods in Robotics, Florianopolis, Brazil (2013)
- ISO9001:2000 (Quality Management System) from the German consultant company, Handwerkskammer Rhein-Main, (2008)
- Pedagogical skills that mainly focused on Learner Centered Methods of Instruction, Measurement and Evaluation in Higher learning Institutions, Bahir Dar, University, Ethiopia (2005)

Publications

1. Yimesker Yihun, Eric T. Wolbrecht, Alba Perez-Gracia "Methodology for the Design of Exoskeletons ", to be submitted to IEEE Transactions on Robotics.
2. Yimesker Yihun, Ken Bosworth, Alba Perez-Gracia "Link-based performance optimization of spatial mechanisms", ASME Journal of Mechanical Design, 2013 (under review).
3. Batchimeg Batbold, Yimesker Yihun, James S. Wolper and Alba Perez-Gracia"Exact Workspace Synthesis for RCCR linkages", Computational Kinematics 2013: 6th International Workshop on Computational Kinematics, Barcelona, Spain, May 12-15, 2013 Barcelona.
4. Yimesker Yihun, Ken Bosworth, Alba Perez-Gracia "Link-based performance optimization of spatial mechanisms" the ASME 2013 International Design Engineering Technical Conferences Computers and Information in Engineering Conference, August 4-7, 2013, Portland, Oregon, USA.
5. PavanKumar Y, Yimesker Yihun, Alba Perez Gracia, Marco P. Schoen "Real time finger joint angle control for prosthetics using sEMG sensor and system identification" Proceedings of the 11th International Symposium, Computer Methods in Biomechanics and Biomedical Engineering April 3 - 7, 2013, Salt Lake City, Utah, USA.
6. Yimesker Yihun, Robert Miklos, Alba Perez-Gracia, David J. Reinkensmeyer, Keith Denney and Eric T. Wolbrecht "Single Degree-of-Freedom Exoskeleton Mechanism Design for Thumb Rehabilitation " Proceedings of the 34th Annual International IEEE EMBS Conference, San Diego, CA, USA, August 28-September 1, 2012.

7. Yimesker Yihun, Md Shamim N.Rahman, Alba Perez-Gracia “Design of an Exoskeleton as a Finger-Joint angular Sensor ” Proceedings of the 34th Annual International IEEE EMBS Conference, San Diego, CA, USA , August 28-September 1, 2012.
8. C. Potluri, Y.Yihun, M. Anugolu, P. Kumar, S. Chiu, M. P. Schoen, and D. S. Naidu Implementation of sEMG-Based Real-Time Embedded Adaptive Finger Force Control for a Prosthetic Hand, 50th IEEE Conference on Decision and Control and European Control Conference, Orlando (FL), USA, December 12-15, 2011.
9. P. Kumar, A. Sebastian, C. Potluri, Yimesker Yihun, A. Ilyas, M. Anugolu, R. Sharma, J. Creelman, A. Urfer, D. S. Naidu, and M. P. Schoen, Spectral Analysis of sEMG Signals to Investigate Skeletal Muscle Fatigue, 50th IEEE Conference on Decision and Control and European Control Conference, Orlando (FL), USA, December 12-15, 2011.
10. C. Potluri, M. Anugolu, Y.Yihun, A. Jensen, S. Chiu, M. P. Schoen, D. S. Naidu, Optimal Tracking of a sEMG based Force Model for a Prosthetic Hand, Proceedings of the 33rd Annual International IEEE EMBS Conference, Boston, MA, August 30- September 3, 2011.
11. P. Kumar, C. Potluri, A. Sebastian, Y.Yihun, A. Ilyas, M. Anugolu, R. Sharma, S. Chiu, J. Creelman, A. Urfer, D. S. Naidu, and M. P. Schoen, A Hybrid Adaptive Multi Sensor Data Fusion for Estimation of Skeletal Muscle Force for Prosthetic Hand Control, ICAI11 - The 2011 International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, July 18-21, 2011.
12. C. Potluri, M. Anugolu, A. Fassih, Y. Yihun, P. Kumar, S. Chiu, and D. S. Naidu, Real-Time sEMG Acquisition and Processing Using a PIC 32 Microcontroller, ESA11 - 9th Int’l Conference on Embedded Systems and Applications, Las Vegas, Nevada, USA, July 18-21, 2011.
13. P. Kumar, C. H. Chen, A. Sebastian, M. Anugolu, C. Potluri, A. Fassih, Y. Yihun, A. Jensen, Y. Tang, S. Chiu, K. Bosworth, D. S. Naidu, Marco P. Schoen, J. Creelman and A. Urfer, An Adaptive Hybrid Data Fusion Based Identification of Skeletal Muscle Force with ANFIS and Smoothing Spline Curve Fitting, 2011 International Conference on Fuzzy Systems, Taipei, Taiwan, June 27 - June 30, 2011.
14. C. Potluri, Yimesker Yihun, P. Kumar, J. Molitor, S. Chiu, H. S. Mousavinezhad and D. S. Naidu, sEMG Based Real- Time Embedded Force Control Strategy For a Prosthetic Hand Prototype, 2011 IEEE International Conference on Electro/Information Technology, Minnesota State University, Mankato, MN, USA, May 15-17, 2011.