Design and Implementation of a High Speed Cable-Based Planar Parallel Manipulator

by

Edmon Hok-Man Chan

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

Waterloo, Ontario, Canada, 2005

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

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

Acknowledgements

I would like to thank my supervisor Dr. Amir Khajepour for all the help and guidance on this project. I have worked for Dr. Khajepour since my coop days. He has such a genius mind and a unique way to approach a problem, and he is always supportive in my research.

I would also like to thank my colleagues, Saeed Behzadipour for his insight in the theoretical development; Robert Dekker for his help in the mechanical and electrical design; and Hang Tran for her assistant in numerous problems that I have encountered during my study.

With all of the mechanical fabrication required in the prototype, I cannot stress how grateful I am for the assistance and training on machining and general mechanical knowledge that were given to me by our technician, Robert Wagner and our student shop supervisor John Potzold. Thanks to them, I have become a better engineer who is capable of not only making an equation that works on paper but also designing a machine that works in real-life.

I also need to thanks my parents and friends for encouraging me throughout my Masters. Joe Tam and Sam Tam, my best University classmate for making my time in the Waterloo such a great experience. Eric Wong, my old friend and mentor back in Britain, who had taught me how to face problem in difficult time. Jacky Chan, my long time childhood friend who is always by my side though the difficult time. Without them, it would not be possible for me to endure all the stressful moments in my study.

I would also like to express thanks to NSERC and OGS for their financial support on my graduate study. Last but not least, I need to thank MMO for their financial support that makes this project possible. Thank you so much.

Abstract

Robotic automation has been the major driving force in modern industrial developments. High speed pick-and-place operations find their place in many manufacturing applications. The goal of this project is to develop a class of high speed robots that has a planar workspace. The presented robots are intended for pick-and-place applications that have a relatively large workspace.

In order to achieve this goal, the robots must be both stiff and light. The design strategies adapted in this study were expanded from the research work by Prof Khajepour and Dr. Behzadipour. The fundamental principles are to utilize a parallel mechanism to enhance robot stiffness and cable construction to reduce moving inertia. A required condition for using cable construction is the ability to hold all cables under tension. This can only be achieved under certain conditions.

The design phase of the study includes a static analysis on the robot manipulator that ensures certain mechanical components are always held under tension. This idea is extended to address dynamic situations where the manipulator velocity and acceleration are bounded. Two concept robot configurations, 2D-Deltabot, and 2D-Betabot are presented. Through a series of analyses from the robot inverse kinematic model, the dynamic properties of a robot can be computed in an effective manner. It was determined that the presented robots can achieve 4g acceleration and 4m/s maximum speed within their 700mm by 100mm workspace with a pair of 890W rotary actuators controlling two degrees of freedom.

The 2D-Deltabot was chosen for prototype development. A kinematics calibration algorithm was developed to enhance the robot accuracy. Experimental test results had shown that the 2D-Deltabot was capable of running at 81 cycles per minute on a 730mm long pick-and-place path. Further experiments showed that the robot had a position accuracy of 0.62mm and a position repeatability of 0.15mm, despite a few manufacturing errors from the prototype fabrication.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Industry always thrives for higher quality and more economical processes. Robotic automation has been the primary driving force for improving modern manufacturing processes. A fast and accurate pick-and-place operation is a desired robotic application for many industrial sectors. Typical pick-and-place applications such as packaging, assembly, and part sorting require manipulation of an object on a flat surface. It is a common practice to stack multiple planar manipulators together while using a conveyor to feed a matrix of objects in a direction normal to the manipulator workspace. This approach usually saves precious manufacturing space in an industrial environment. In this work, a new type of 2D cable-based parallel manipulator is introduced that is intended for high speed pick-and-place applications that require manipulation of light objects.

The primary requirements for pick-and-place operations are high accuracy, high speed, and high repeatability. High accuracy can only be attained if the robot construction is stiff enough to suppress deformation; high speed, on the other hand is limited by the actuator power and the robot inertia. Since high stiffness usually comes at the expense of increasing inertia, a designer must find the right balance for his/her particular solution. The most common industrial robotic manipulators nowadays are Cartesian tables and Articulated manipulators. These types of configurations append independent motions from one link to another in a serial fashion, hence they are classified into the serial robot family. Due to the nature of the design, the robot actuator, or at least the power chain that is connected to the actuator must move with the manipulator. As a result, a serial robot trends to have a large moving inertia to payload ratio regardless of the size of the actuator.

A new type of robotic design, parallel mechanisms has emerged from recent robot developments. This type of robot is constructed by attaching multiple independently actuated kinematic chains to a mobile platform. Since the kinematic chains do not stack from one to another, any force that is applied to the mobile platform is distributed amongst multiple linkages. This effectively increases the stiffness of the robot structure. Moreover, the actuators of each chain can be fixed on to a base, and they do not become part of the moving inertia. These two properties provide parallel mechanisms with an inherent advantage on stiffness and inertia over their serial counterparts. With the advantage of stiffness and low inertia, parallel mechanisms have quickly found their way into many high speed pick-and-place applications. Among which, the Delta configuration is arguably the most successful design in the last decade. Most of the recent parallel robot designs can achieve 150 cycles per minute [1].

If the underlying principle of the success in high speed pick-and-place robots is light and stiff, there must be other methods to improve robot performance by further reduction in moving inertia without compromising too much on structural stiffness. This thesis document expands the design principles from the research work by Prof. Khajepour and Dr. Behzadipour. The fundamental design strategy is to improve the stiffness to inertia ratio by replacing rigid linkages of the kinematic chain with flexible cables. Mechanical cable, which is virtually massless, possesses a relatively high mechanical strength under tension. The fundamental principle of this design approach is to replace the heavy linkages with cables. An additional advantage of using cables is that they can replace revolute joints, which are relatively costly and unreliable due to their limited life expectancy. Furthermore, there are several design issues that must be addressed when using cables in a mechanism.

The layout of this thesis is as follows: Chapter 2 presents the literature review of some parallel mechanisms and design considerations that are specific to cable-based mechanisms. Chapter 3 introduces the conceptual design and the theoretical development for this new class of cable based robots. It also addresses the primary design requirements that satisfy the design constraints. Chapter 4 discusses detailed design strategies and optimization processes. It also offers a systematic methodology that requires only the inverse kinematic equations for the

[1] for more information, refer to http://www.abb.com/robotics

development of this class of robots. Chapter 5 provides a more detailed design implementation of two robots that use the design approach introduced in Chapter 4. The inverse kinematic equations of both robots are developed along with the design procedure. Chapter 6 then shows the kinematic calibration and the experimental evaluations of a prototype that was built during the course of this project. Chapter 7 gives concluding remarks of this work and highlights future research directions.

Chapter 2

Literature Review

2.1 Introduction of Parallel Manipulators

This thesis work focuses on two aspects - Parallel Mechanisms and Cable-Based Manipulators. Before introducing these design concepts, it is appropriate to introduce parallel mechanisms in this chapter. Parallel Mechanisms, or sometimes referred as closed chain mechanisms is a relatively new concept in industry. However, the actual theoretical work can be dated centuries ago[1]. Tsai, a pioneer in recent parallel manipulator developments has once compared different types of parallel mechanisms, and described them to be "a mechanism that is typically consists of a moving platform that is connected to a fixed base by several limbs" [30]. For many parallel manipulators, all the limbs (kinematic chains) are of the same kind and the orientational and translational degrees of freedom are usually coupled.

Early parallel mechanisms were usually targeted for 6 degrees of freedom. The simplest implementation is the Gough Platform designed by Gough and Whitehall in 1947 [33]. In that design, the moving platform is actuated by 6 linear actuators that connect the base and the moving platform through spherical joints. The Stewart platform, a later version of the Gough Platform is shown in Figure 2-1. Similar configurations have been used in flight simulators [5] and Ingersoll's milling machines [13]. One of the most successful implementation of parallel mechanisms is the Delta invented by Clavel in 1990s [28]. In the Delta, the moving platform is

[1] Interested reader are referred to http://www.parallemic.org/Reviews/Review007.html for more details

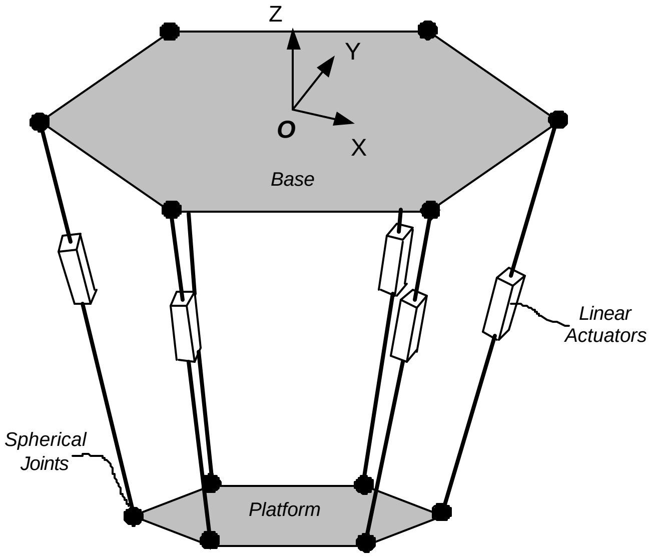

Figure 2-1: Schematic of Stewart Platform

actuated by three pairs of synchronized elementary chains. Each pair is driven independently by a rotary actuator as illustrated in Figure 2-2. This configuration eliminates any rotational degree of freedom and hence produces a three translational DOF manipulator.

Ever since Clavel published his invention, a significant amount of research activities have been conducted to improve the design of Delta and many other similar parallel configurations [4], [9], [10], and [26]. Some of the research works are directed towards a 2 translational degrees of freedom (DOF) version of the parallel mechanism. Ghorbel developed Rice Planar Delta Robot (RPDR) in 1990s to demonstrate various control and analysis techniques for parallel robots [8]. RPDR is kinematically similar to Delta as shown in Figure 2-3, and it was originally designed for experimental purposes. The orientational DOF in RPDR is coupled with the translational DOF, which limits its applications. This shortcoming has been overcome by another design by Huang et al [12]. In that design, a pair of parallel links is used to connect the moving platform to the first link, much like the original Delta design. This planar Delta robot with only translational DOF was intended for high speed pick-and-place operations.

From a development point of view, some of these research projects introduced analytical methods to analyze the complex forward kinematics of parallel robots [14] [24]; others addressed

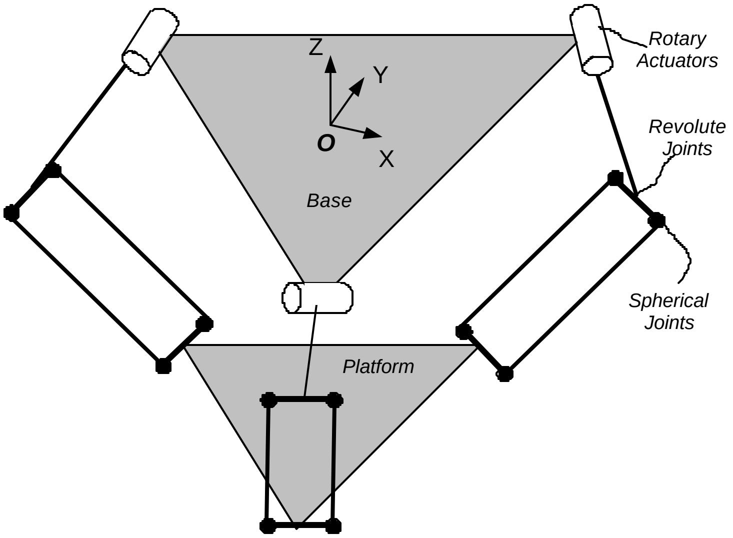

Figure 2-2: Schematic of Delta

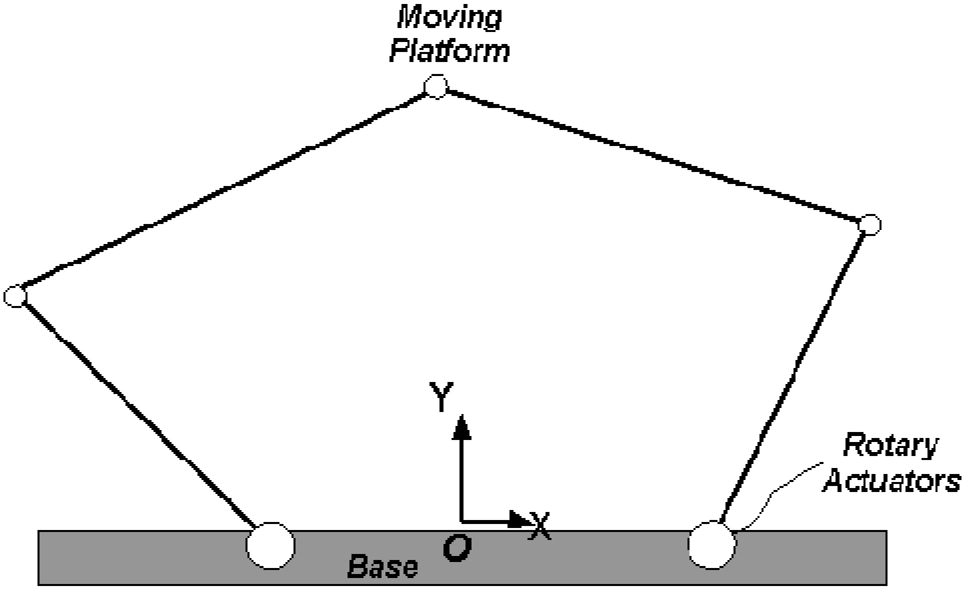

Figure 2-3: Schematic of Rice Planar Delta Robot

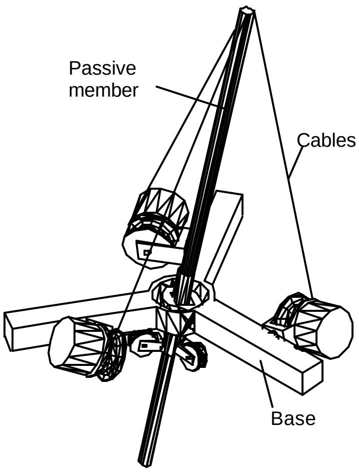

Figure 2-4: Pictorial View of Landsberger Robot

more subtle issues such as internal singular configurations [23]. However, these analytical tools are subject to certain specific conditions, and there is still no universal systematic method to analyze parallel mechanisms.

2.2 Cable-Based Mechanisms

While there are already a significant number of parallel manipulator designs in recent literature, one class of parallel mechanisms that uses cables in its kinematic chains is of particular interest in this thesis work. A cable-based parallel manipulator usually has a lower moving inertia and lower manufacturing cost when compared to its rigid link counterpart. Moreover, the flexibility of a cable allows the replacement of mechanical joints in the design. However, a cable-based robot design has a few shortcomings. Cable is only useful if it is held under tension. Some of the research has been directed towards the tensioning of cables in cable-based robots. Such research includes work from Ming and Higuchi who performed an earlier study on the tension distribution in high DOF cable-based manipulators [2]. Behzadipour et al performed a study on the tensionable cable robot configurations by geometric interpretation [29]. Another design is from Tadokoro et al, who designed cable wire robots using 8 cables, all under tension (with 2 redundant cables) for 6 DOF motion [32].

Early cable-based parallel robots were designed mainly for speed enhancement. Landsberger developed a 3 DOF cable-based manipulator that is somewhat an extension to the Stewart-Gough platform. In Landsberger's design, three cables were attached to a telescopic compression member as shown in Figure 2-4. The lengths of the cables were independently controlled by winches for positioning the end-e¤ector in 3D space [31]. Early implementations of the Landsberger robot were capable of reaching a speed of 3.5m/s.

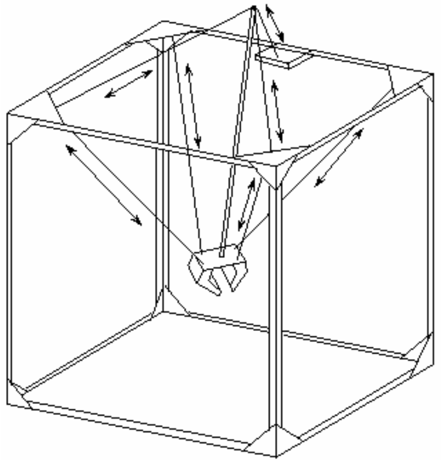

Figure 2-5: Schematic of Falcon

Another cable-based manipulator that features a high DOF and a large workspace is the Falcon. Falcon is designed for positioning applications [18]. There are 7 cables (6+1 for redundant kinematic chain that induces the tension to other cables) as shown in Figure 2-5. Due to the low inertia design, Falcon is capable of achieving 43g of acceleration and a peak velocity of 13m/s. This is a cable-based manipulator that demonstrates high mobility and fast dynamic characteristic.

Over the past two decades, cable-based manipulators have proven to be a suitable choice for high speed applications. This thesis work focuses on the development of a class of planar cable-based parallel mechanisms that are compact and are capable of achieving an even higher dynamic operational speed for industrial applications.

Chapter 3

Robot Design Overview

The essence of a high speed manipulator design is to reduce the moving inertia while maintaining the stiffness. Effective stiffness can be increased if the extraneous forces are transmitted through multiple paths. Parallel mechanisms apply this principle effectively, which produces an inherent advantage on stiffness and allows it to be constructed with lighter components. In order to further reduce the moving inertia, this work proposes to construct some of the linkages with cable wires. This method can be applied to most common parallel robot configurations with simple kinematic chains. However, there are a few issues that need to be addressed to avoid mechanical failure.

The key issue that requires attention is maintaining tension in the cable. This chapter first introduces the proposed cable-robot design in Section 3.1 and analyses the cable tension properties as a static problem in Section 3.2. The result of this analysis is a set of linear equations relating tension (T) to loads (L), i.e. $T = JL$. However, this analysis is not useful until it examines the dynamic behaviour of the cable-robot. Section 3.3 addresses this problem by conservatively estimating the load vector as a function of the end-effector kinetics ($L = L(x; \dot{x}; \ddot{x})$). This allows analysis the cable tension under different dynamic environments.

3.1 Conceptual Design

The proposed cable-based design is a planar manipulator. The basic design consists of a fixed base, a pair of parallel cables, a third cable, a telescopic spine (central pose), and an end-effector as illustrated in Figure 3-1. The focus of this design was on the mobile end of the manipulator where a pick-and-place tool and the cable linkages are to be mounted on the moving platform.

The spine is attached to the end-effector by a revolute joint with its pivoting axis perpendicular to the plane of action. The objective of the spine is to induce a tensile force on the cables. There is also a pair of cables attached to the end-effector. These paired cables are separated by a fixed distance and they form a parallelogram when they are held under tension. Due to the geometric constraint on this parallelogram, the orientation DOF is controlled. In this work, this pair of parallel cables is defined as a parallel cable joint.

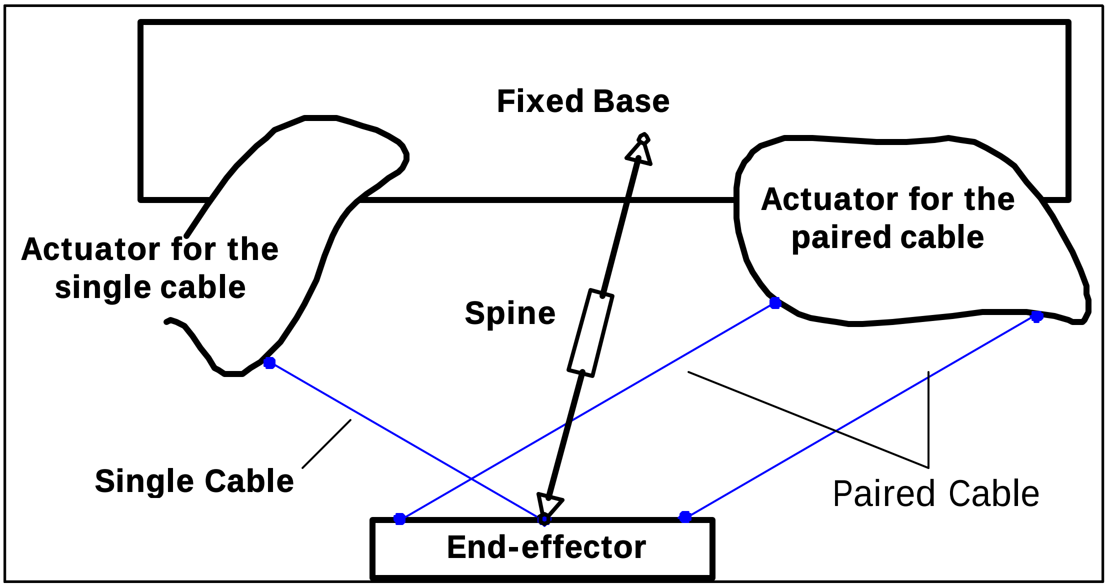

Figure 3-1: Schematic of the Proposed Cable-Based 2DOF Planar Mechanism

Definition 1 *A parallel cable joint* is a joint made of two parallel cables attached to a rigid element E. The centerline direction of the cables is OO' and can vary in time, see Figure 3-2. The rigid element E represents the end-effector in the context of this work. The cables are assumed to be held under tension (positive T_1 and T_2).

There are two distinct types of assembly in the proposed robot design. The cable is part of the kinematic chain that connects these two assemblies together. A natural classification would be based on their function:

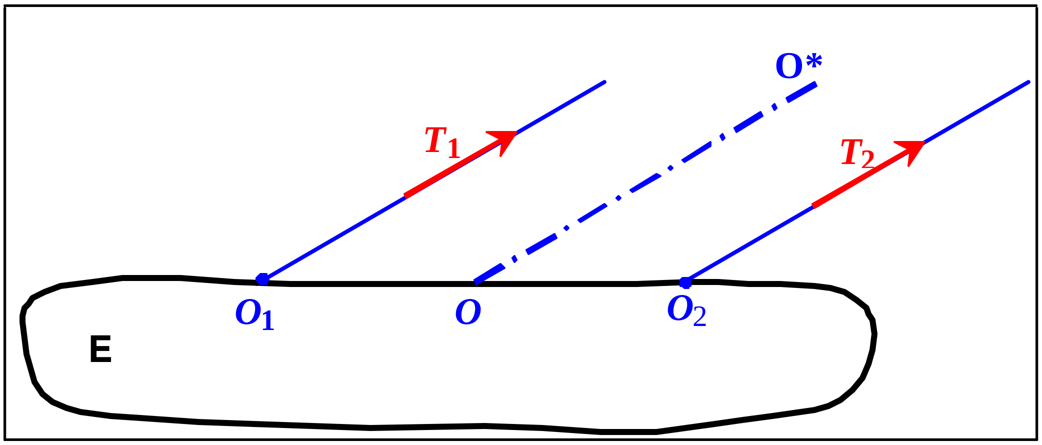

Figure 3-2: Schematic of a Parallel Cable Joint

Definition 2 The **upper robot assembly** is the portion of the robot that is responsible for the actuation of the manipulator. This portion of the robot includes the fixed base, the actuators, and the first part of the kinematic trains.

Definition 3 The **lower robot assembly** is the portion of the robot that is responsible for the proper manipulation of the end-effector. Since it has a relatively large amount of motion, it usually dominates the dynamic characteristic of the manipulator. This portion of the robot is composed of the end-effector and the spine.

While the proposed robot could have many different upper assembly designs to suit the particular application, the lower assembly design remains virtually unchanged. Some of the essential aspects on the lower assembly design are discussed in Section 3.2.

3.2 Theoretical Development of 2D Cable-Based Robot

Parts of the robot kinematic chain introduced in Section 3.1 are constructed from cable instead of rigid linkage. A solid linkage can withstand tension, compression and bending. Depending on the length, slender linkages usually have a low compressive strength even with high mass. Similarly, bending stiffness is usually poor for slender linkages. Therefore, it is highly beneficial to operate linkages under tension only. Under the assumption that a mechanical component operates under tension, there is little difference whether that component is a piece of cable or a rigid slender linkage. As a result, there is an element to provide tension to the cables in cable-based manipulators. This element is called the spine, and it can be another cable, a pneumatic cylinder, or a spring. This section discuss the position of the spine to ensure the cable can be held under tension. Before investigating the detailed aspect of the necessary operation conditions for these cable-based robots, it is appropriate to discuss a few concepts that allow a cable-robot to behave as if it were constructed with rigid elements.

Definition 4 Rigid pose A cable-based manipulator is considered to be in a rigid pose, if the manipulator is not at a singular pose and all of the cables are held in tension under a given set of external loads. Therefore, a cable-robot that is in a rigid pose is considered as a structure.

Definition 5 Tensionability A non-singular pose is tensionable if and only if for any arbitrary external load, there exists a finite spine force to make the manipulator rigid. Tensionability, the ability to apply tension to a mechanical component (in this case, the cable) is a necessary condition to maintain a cable-based manipulator in a rigid pose regardless of any other external load.

Since a useful manipulator must be able to guarantee its rigidity everywhere within the workspace, it is necessary to seek some insight into the tensionability of a manipulator. This chapter discusses a design strategy that always satisfies the tensionability criterion of a 2D cable-based manipulator.

When a manipulator is in a rigid pose, it does not translate nor rotate. Assuming the cable axes are independent and two of the cables form a parallel cable joint, the manipulator will not be in a singular position. In order for a manipulator to be considered as a structure, the tensionability is the only remaining condition that needs to be satisfied. Since the cable tensions are heavily dependent on the spine, the very first condition that shall be examined is the spine location to ensure tensionability. The investigation begins with a simple non-translating end-effector setup. This setup leads to a necessary spine position that produces a tensionable configuration under no external load. The analysis is then extended for the tensionability condition under which the manipulator is tensionable for any finite load (and torque) and a generalized end-effector configuration.

3.2.1 Tensionable Configuration

A) Zero Translational Requirement (with a specific external load)

Suppose two cables are attached to point O of the end-effector under an external load P as seen in Figure 3-3, the end-effector will not translate if the tension in cables T_A and T_B can balance the external load P. This in fact indicates that if the manipulator is tensionable, there exists a set of positive T_A and T_B to balance the external force P. If P lies between the cables bases, (OL_1^0 and OL_2^0), it can be projected onto the bases axis with a positive component. As a result, there exists a set of positive cable tensions to cancel out the component of force resolved in each of the cable axis. Therefore, the manipulator is tensionable as long as P lies inside the OL_1^0 ¡ OL_2^0 envelope. A corollary is that the negation of the force vector P that lies between the cable axes must also lies inside the positively linearly dependent space spanned from T_A and T_B ($T_A > 0$, $T_B > 0$). Also note that an increasing P monotonically increases T_A and T_B as the magnitude of the force component increases.

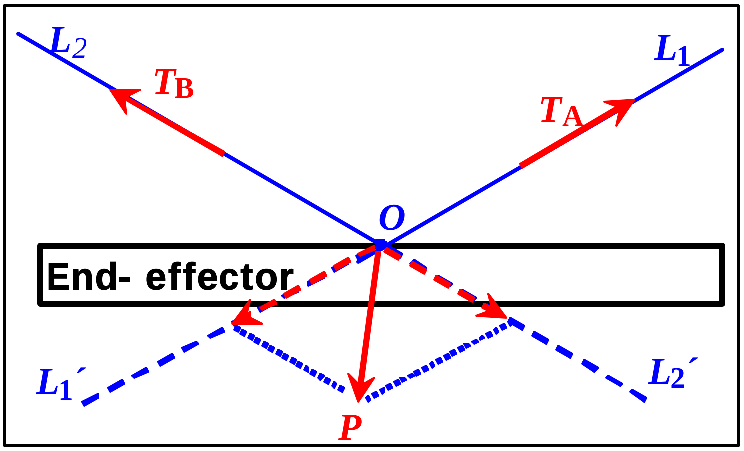

Figure 3-3: Zero Translational Platform

B) Translational and Rotational Requirement (In the absence of external load and torque)

Suppose a parallel cable joint is attached to an end-effector such that its centerline coincided with the third cable and a spine as shown in Figure 3-4, the point where all components coincide is defined to be point O. The assembly is completely in a rigid pose (all cables are in tension, zero translation and zero rotation) if and only if the spine axis, OL_3 is between the cable axes, OL_1 and OL_2 and the spine exerts a compressive force F_C onto the rigid body at point O.

This configuration is an extension to Case A. Let $T_A = T_{A1} + T_{A2}$, there exists positive T_A and T_B for any value of F_C that constrains the translational DOF as shown in Case A. To satisfy the torque equilibrium condition, $\sum M = 0$ (in the absence of any external torque), T_{A1} and T_{A2} are selected to be:

$$T_{A1} = T_{A2} = \frac{T_A}{2} \tag{3.1}$$

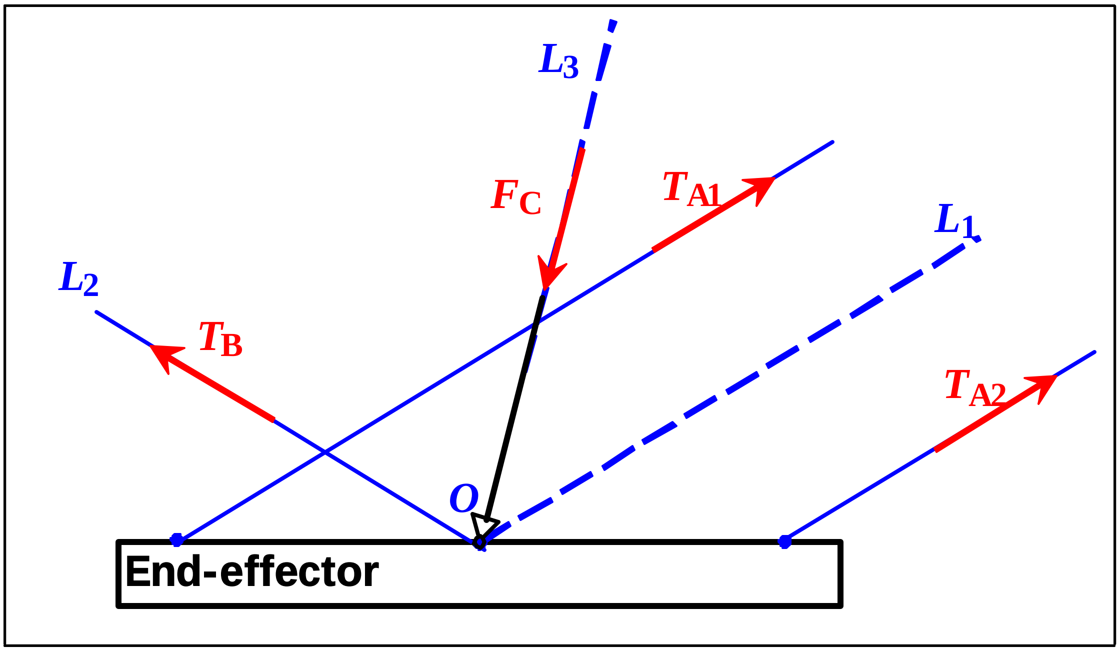

Figure 3-4: Zero Translational and Rotational End Effector Under No Load

C) Zero Translational Requirement with any arbitrary external load

Suppose two cables with axes OL_1 and OL_2, and a spine with axis OL_3 orientated between the cable axes are connected to the end-effector as shown in Figure 3-5, this configuration can balance any finite external force P, in any arbitrary direction. Therefore, it is a tensionable configuration.

Although this result can be directly observed if one realizes the axes OL_1, OL_2, and OL_3 form a positively linearly dependent basis on a 2D plane, it is worthwhile to quantitatively

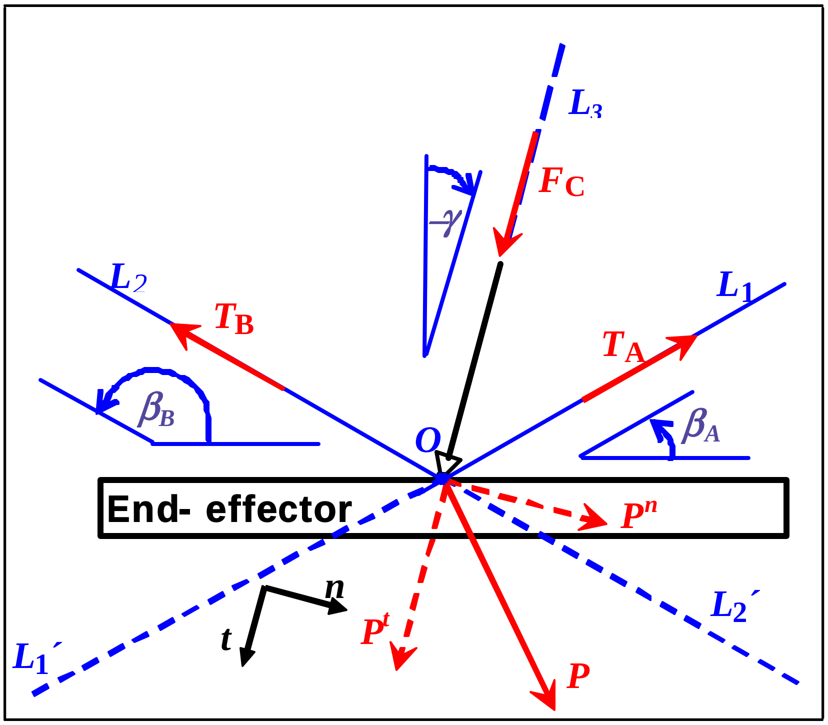

Figure 3-5: Zero Translational End Effector under any arbitrary load. Note the the spine angle, ° is shown in the negative direction.

analyze the problem. The external force can be resolved into an orthogonal basis that is aligned to axis OL_3. The component that is parallel to the OL_3 is P^t, and the component that is normal to OL_3 is P^n. Using the superposition of forces, the system is in force equilibrium (translational rigidity) if both components of the external force can be balanced by the applied load F_C, T_A, and T_B, all with a positive magnitude (tensionable).

The tangential component, P^t produces a tensionable configuration if it is positive (pointing away from the cables) as explained in Case A. If the force component is negative, a contribution of F_C is needed to force the sum of the forces to become positive. Therefore, the tangential component always forms a tensionable configuration.

As for the normal component, P^n, it can either lie between OL_1^0 and OL_2^0 or outside the OL_1^0-OL_2^0 envelope. If it were the former case, the situation is identical to Case A and the system is tensionable. If it were the latter case, a positive component of F_C is needed to bring P^n onto OL_2^0 if P^n is positive, and OL_1^0 if P^n is negative (see Figure 3-5). Therefore, the normal component also forms a tensionable configuration.

Note that it is possible to compute the cable tensions at this point. Let F_C' be the summation of F_C and P^t, it can be shown through geometry and force equilibrium that the cable tensions T_A and T_B can be computed by a set of linear equations. The force equilibrium in a static case shows:

$$\sum F_x = T_A \cos(\beta_A) + T_B \cos(\beta_B) + F_C' \sin(\gamma) + P^n \cos(\gamma) = 0$$

$$\sum F_y = T_A \sin(\beta_A) + T_B \sin(\beta_B) - F_C' \cos(\gamma) + P^n \sin(\gamma) = 0$$

In matrix form:

$$\begin{bmatrix} \cos(\beta_A) & \cos(\beta_B) \\ \sin(\beta_A) & \sin(\beta_B) \end{bmatrix} \begin{Bmatrix} T_A \\ T_B \end{Bmatrix} = \begin{bmatrix} -\sin(\gamma) & -\cos(\gamma) \\ \cos(\gamma) & -\sin(\gamma) \end{bmatrix} \begin{Bmatrix} F_C' \\ P^n \end{Bmatrix}$$

Rearrange and simplify the equilibrium equations using trigonometric identities produce:

$$\begin{Bmatrix} T_A \\ T_B \end{Bmatrix} = \frac{1}{\sin(\beta_B - \beta_A)} \begin{bmatrix} -\cos(\gamma - \beta_B) & \sin(\gamma - \beta_B) \\ \cos(\gamma - \beta_A) & -\sin(\gamma - \beta_A) \end{bmatrix} \begin{Bmatrix} F_C' \\ P^n \end{Bmatrix} \tag{3.2}$$

Also note that the static assumption does not restrict the dynamic analysis, as the applied forces can be replaced by the inertia force from the robot component. A more comprehensive analysis is shown in Section 4.

D) Zero Translational and Rotational Requirement with arbitrary load

The cable-robot in Figure 3-6 is shown to be in a rigid configuration for any external force, P and torque, M with a finite amount of spine force F_C.

The end-effector forms a rigid configuration if there exists a set of positive T_{A1}, T_{A2}, T_B, and F_C for any given applied loads that satisfies the force equilibrium equations:

$$\sum F = 0$$

$$\sum M_O = 0$$

Suppose the tension forces T_{A1} and T_{A2} are replaced with an equivalent force T_A and a torque M_A at point O. Note that T_A is also aligned to the cable axis A. By applying the superposition principle on the translational system (i.e. ignoring the torque), it can be observed that the translational sub-system is identical to Case C. Therefore there exists a spine force, F_C that induces a tensile force on the cables.

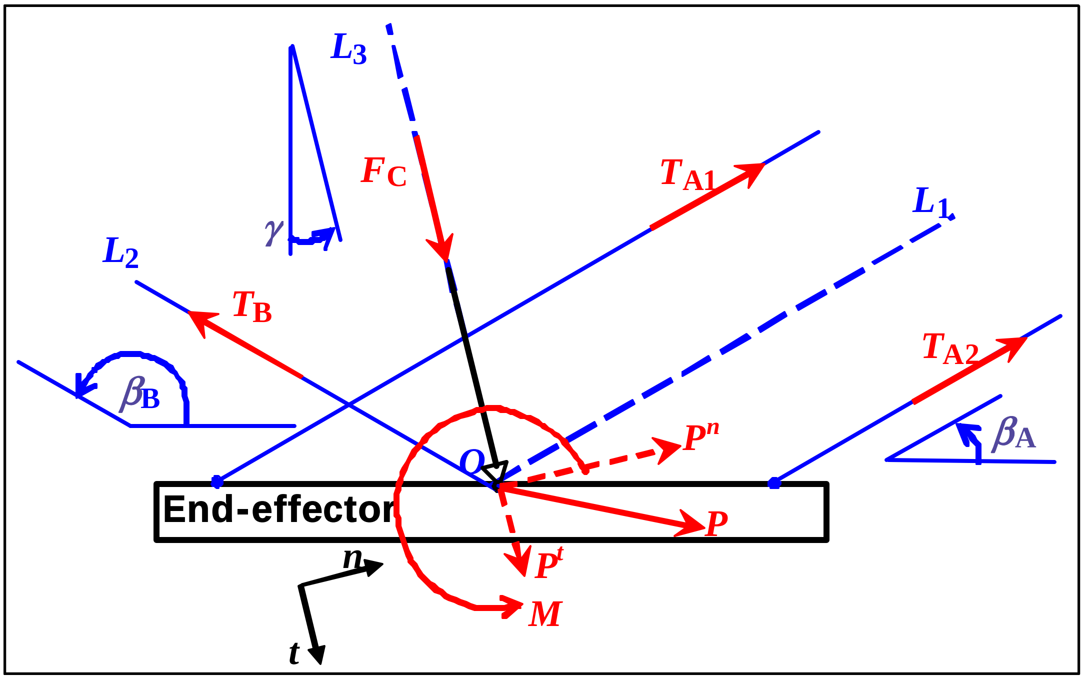

Figure 3-6: Zero Translational and Rotational End Effector under Arbitrary Load and Moment

As for the rotational system, consider the parallel cable joint in Figure 3-7. The system is tensionable for a finite amount of applied torque, M. The torque capacity of a parallel cable joint - the amount of applied torque that a parallel cable joint can balance is dependent on the amount of the total tension on both cables. The equations of the force / torque equilibrium about point O are:

$$\sum F_y = 0 : T_A = T_{A1} + T_{A2} \tag{3.3}$$

$$\sum M_O = 0 : M = T_{A1} d \sin(\beta_A) - T_{A2} d \sin(\beta_A) \tag{3.4}$$

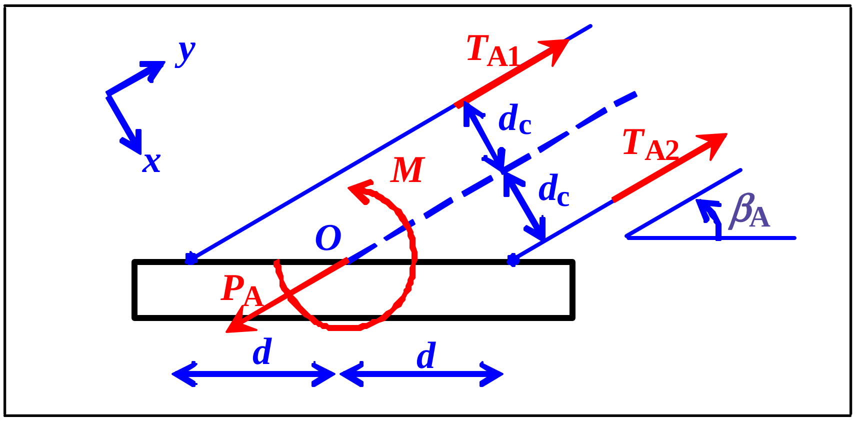

Figure 3-7: Torque capacity of a parallel cable joint. P_A is an applied force that is equal in magnitude, but opposite in direction when compared to the equivalent cable tension, T_A

If both of T_{A1} and T_{A2} are positive, and knowing neither T_{A1} nor T_{A2} is bigger than T_A, it can be concluded that the maximum torque that can be applied to the parallel cable joint without losing the cable tension must satisfy the following inequality:

$$-T_A d \sin(\beta_A) \le M \le T_A d \sin(\beta_A) \tag{3.5}$$

Note that the torque capacity is heavily dependent on T_A, which in turn holds a linear relationship with F_C from Case B. Therefore, an increase in F_C would generally result in a larger torque capacity on the parallel cable joint. It can be concluded that there exists a finite spine force to generate a large enough torque capacity to form a rigid pose for the rotational system. Therefore, the entire system is tensionable.

It is important to compute the minimum necessary spine force and the cable tension quantitatively in the design process. Solving Equations (3.3) and (3.4) simultaneously and appending T_B in the system produces:

$$\begin{Bmatrix} T_{A1} \\ T_{A2} \\ T_B \end{Bmatrix} = \begin{bmatrix} 0.5 & 0 & \frac{1}{2d\sin(\beta_A)} \\ 0.5 & 0 & -\frac{1}{2d\sin(\beta_A)} \\ 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} T_A \\ T_B \\ M \end{Bmatrix} \tag{3.6}$$

Here, two sets of force that are important to the dynamic behaviour of the robot should be defined:

$$\{T\} = \left\{ \begin{array}{c} T_{A1} \\ T_{A2} \\ T_B \end{array} \right\} \quad (3.7)$$

$$\{L\} = \left\{ \begin{array}{c} F^0_C \\ P^n \\ M \end{array} \right\} \quad (3.8)$$

The cable tension vector (3.7), which consists of the tension of the individual cable, and the load vector (3.8), which is a set of external load and moment applied to the end-effector typically from the inertia force of a moving body.

Appending the applied moment, M to Equation (3.2) produces a mapping function that involves L:

$$\left\{ \begin{array}{c} T_A \\ T_B \\ M \end{array} \right\} = \begin{bmatrix} \frac{-\cos(\gamma-\beta_B)}{\sin(\beta_B-\beta_A)} & \frac{\sin(\gamma-\beta_B)}{\sin(\beta_B-\beta_A)} & 0 \\ \frac{\cos(\gamma-\beta_A)}{\sin(\beta_B-\beta_A)} & -\frac{\sin(\gamma-\beta_A)}{\sin(\beta_B-\beta_A)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \left\{ \begin{array}{c} F^0_C \\ P^n \\ M \end{array} \right\}$$

which is also compatible with Equation (3.6). Once it is substituted back into Equation (3.2), it can be shown that T is also linearly related to L by a set of linear equations. The linear mapping matrix will be referred as the J matrix:

$$\left\{ \begin{array}{c} T_{A1} \\ T_{A2} \\ T_B \end{array} \right\} = \begin{bmatrix} 0.5 & 0 & \frac{1}{2d\sin(\beta_A)} \\ 0.5 & 0 & -\frac{1}{2d\sin(\beta_A)} \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\cos(\gamma-\beta_B)}{\sin(\beta_B-\beta_A)} & \frac{\sin(\gamma-\beta_B)}{\sin(\beta_B-\beta_A)} & 0 \\ \frac{\cos(\gamma-\beta_A)}{\sin(\beta_B-\beta_A)} & -\frac{\sin(\gamma-\beta_A)}{\sin(\beta_B-\beta_A)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \left\{ \begin{array}{c} F^0_C \\ P^n \\ M \end{array} \right\} \quad (3.9)$$

$$\{T\} = [J]\{L\}$$

E) General tensionability condition with arbitrary load

All of the configurations that have been investigated so far are restricted to a triple intersecting point between the cable axes and the spine axis. There are cases when these axes may not be able to intersect all the time. The cable-robot shown in Figure 3-8 is the most general configuration for this type of robot. For the general case, let point O be the pivot point of the spine, ±x be the horizontal eccentric distance measured from the axis of the parallel cable joint to point O (positive if the intersection is between the spine axis and cable B; note that the ±x is negative as shown in the Figure 3-8), and ±y to be the vertical eccentric distance measured from the axis of cable B and point O (positive if the intersection is below point O). This generalized setup is tensionable for any external force and torque by a finite value of spine force if the following conditions are met:

$$C1 : {}^{-}{}_{A} \ \mathrm{i}\ {}^{-}{}_{B} < 180^{\pm} \qquad (3.10)$$

$$C2 : 180^{\pm} > {}^{-}{}_{B} > {}^{\circ} + 90^{\pm} > {}^{-}{}_{A} > 0^{\pm} \qquad (3.11)$$

$$C3 : \left| \pm x + \pm y \frac{\cos({}^{\circ}\ \mathrm{i}\ {}^{-}{}_{A})}{\cos({}^{\circ}\ \mathrm{i}\ {}^{-}{}_{B})} \frac{\cos({}^{-}{}_{B})}{\cos({}^{-}{}_{A})} \right| < d \qquad (3.12)$$

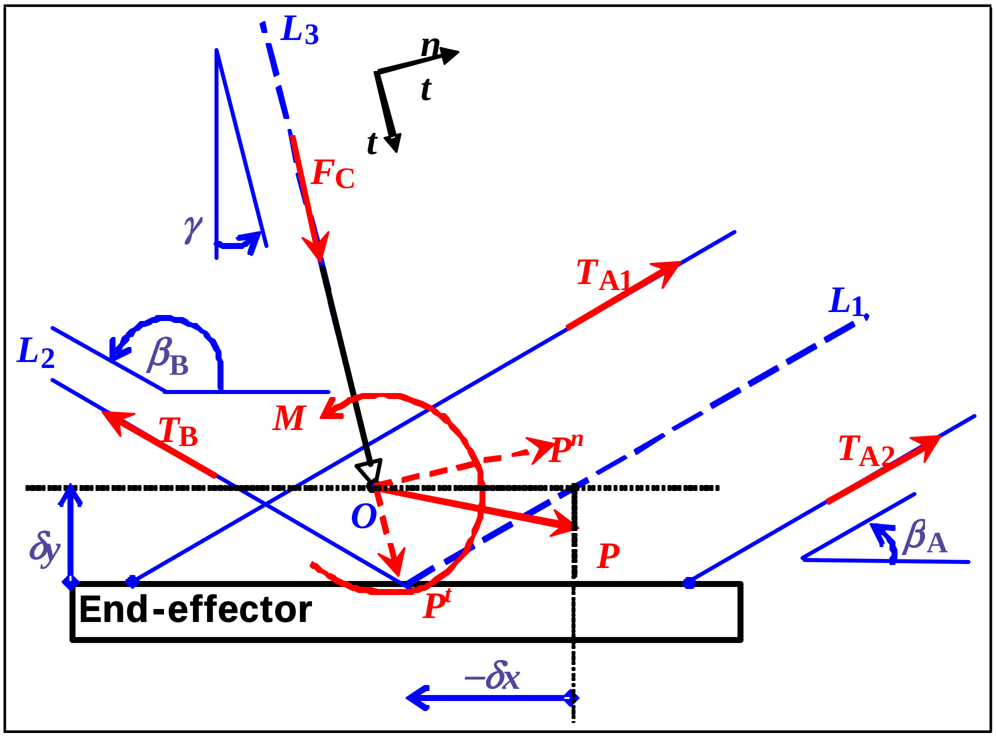

Figure 3-8: General cable-robot configuration with eccentric distance. Note that ±x is negative and ±y is positive as shown in the picture.

These conditions are necessary to ensure a monotonic relationship between the cable tensions and the spine force. The equilibrium equations of the end-effector show that the cable tensions are linearly related to the applied loads. Of all three components in the load vector, the effective spine force, F'_C is the only directly controllable load. Since the objective is to obtain a positive cable tension with a positive spine force, one must ensure all cable tensions monotonically increase with the spine force. Mathematically speaking, the derivatives of the cable tensions with respect to the spine force must be positive.

The introduction of the eccentric distance generates a bias torque about point O. The effect is two additional terms introduced in Equations (3:4). Therefore, the load to tension mapping function can be re-derived with the necessary update from Equations (3:6) and (3:9) accordingly. Incorporating the eccentricity terms in Equation (3:4) produces:

$$M = (T_{A1} - T_{A2})d\sin(\beta_A) + (T_{A1} + T_{A2})(\delta y\cos(\beta_A) - \delta x\sin(\beta_A)) + T_B\delta y\cos(\beta_B) \qquad (3.13)$$

With the new moment equilibrium equation, Equation (3:6) becomes:

$$\begin{Bmatrix} T_{A1} \\ T_{A2} \\ T_B \end{Bmatrix} = \begin{bmatrix} \frac{1}{2}\left(1 - \frac{\delta x}{d}\right) & \frac{1}{2}\frac{\delta y}{d}\frac{\cos(\beta_B)}{\sin(\beta_A)} & \frac{1}{2d\sin(\beta_A)} \\ \frac{1}{2}\left(1 + \frac{\delta x}{d}\right) & -\frac{1}{2}\frac{\delta y}{d}\frac{\cos(\beta_B)}{\sin(\beta_A)} & -\frac{1}{2d\sin(\beta_A)} \\ 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} T_A \\ T_B \\ M \end{Bmatrix} \qquad (3.14)$$

Finally Equation (3:9) can be modified with the update from Equation (3:14) to obtained:

$$\begin{Bmatrix} T_{A1} \\ T_{A2} \\ T_B \end{Bmatrix} = \begin{bmatrix} \frac{1}{2}\left(1 - \frac{\delta x}{d}\right) & \frac{1}{2}\frac{\delta y}{d}\frac{\cos(\beta_B)}{\sin(\beta_A)} & \frac{1}{2d\sin(\beta_A)} \\ \frac{1}{2}\left(1 + \frac{\delta x}{d}\right) & -\frac{1}{2}\frac{\delta y}{d}\frac{\cos(\beta_B)}{\sin(\beta_A)} & -\frac{1}{2d\sin(\beta_A)} \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\cos(\gamma - \beta_B)}{\sin(\beta_B - \beta_A)} & \frac{\sin(\gamma - \beta_B)}{\sin(\beta_B - \beta_A)} & 0 \\ \frac{\cos(\gamma - \beta_A)}{\sin(\beta_B - \beta_A)} & -\frac{\sin(\gamma - \beta_A)}{\sin(\beta_B - \beta_A)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} F'_C \\ P^n \\ M \end{Bmatrix}$$
$$(3.15)$$

The product of the middle matrix in Equation (3:15) forms the new J matrix. After some simplification, each column of the J matrix becomes:

$$
\begin{aligned}
J &= \begin{bmatrix} J_{i1} & J_{i2} & J_{i3} \end{bmatrix} \\
J_{i1} &= \frac{1}{2d}\frac{\cos(\gamma-\beta_B)}{\sin(\beta_B-\beta_A)}\begin{bmatrix} -\left[d-\delta x-\delta y\frac{\cos(\beta_B)}{\sin(\beta_A)}\frac{\cos(\gamma-\beta_A)}{\cos(\gamma-\beta_B)}\right] \\ -\left[d+\delta x+\delta y\frac{\cos(\beta_B)}{\sin(\beta_A)}\frac{\cos(\gamma-\beta_A)}{\cos(\gamma-\beta_B)}\right] \\ 2d\frac{\cos(\gamma-\beta_A)}{\cos(\gamma-\beta_B)} \end{bmatrix} \\
J_{i2} &= \frac{1}{2d}\frac{\sin(\gamma-\beta_B)}{\sin(\beta_B-\beta_A)}\begin{bmatrix} \left[d-\delta x-\delta y\frac{\cos(\beta_B)}{\sin(\beta_A)}\frac{\sin(\gamma-\beta_A)}{\sin(\gamma-\beta_B)}\right] \\ \left[d+\delta x+\delta y\frac{\cos(\beta_B)}{\sin(\beta_A)}\frac{\sin(\gamma-\beta_A)}{\sin(\gamma-\beta_B)}\right] \\ -2d\frac{\sin(\gamma-\beta_A)}{\sin(\gamma-\beta_B)} \end{bmatrix} \\
J_{i3} &= \frac{1}{2d}\frac{1}{\sin(\beta_A)}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}
\end{aligned}
\qquad (3.16)
$$

Due to the linear relationship between the cable tensions and the spine force, the derivatives of the cable tension with respect to the spine force are embedded in the first column of the J matrix in Equation (3:16). The tensionability condition ensured when J_{i1} are positive for $i = 1 \ldots 3$ is synonymous with satisfying the inequalities in Equations (3:10) to (3:12). Conditions C1 and C2 in Equations (3:10) and (3:11) are geometric constraints that must be met in setting up the end-effector configuration. They are also conditions that were inspired by Case C in this subsection. Satisfying these conditions automatically guarantees $\frac{\partial T_{A1}}{\partial F_C'}$, $\frac{\partial T_{A2}}{\partial F_C'}$ to be positive in the absence of any eccentric distance, and they also set $\frac{\partial T_B}{\partial F_C'}$ to be positive without any further restriction. The eccentric distance terms in Equation (3:16) can be interpreted as biases that shift the tension distribution amongst the parallel cable joint. Inequality Equation (3:12) in Condition C3 is designed to establish a minimum bound on the horizontal cable separation distance, d so that it can tolerate this shift in tension.

3.3 Application of the Tensionability Conditions and the Equations of Motion

The tensionability criteria investigated above emphasizes one common feature; there exists a minimum, finite spine force, F_C that generates rigid manipulator configurations for a given bound on the external load. The design strategy is to find that minimum necessary spine force and apply it to the entire workspace of the manipulator. The exact analysis would include the kinematic model of the entire robot. The cable tensions can be determined by Equation (3:15). A design optimization procedure usually includes finding some mechanical parameters such as the maximum and minimum cable tensions, actuator torque powers given some constraints on the desired speed and acceleration. This can be accomplished by evaluating the equations of motion and the equation of tensions using the minimum spine force at discrete points on the workspace. Once all of the relevant mechanical properties have been identified, one can compute the design cost (or merit) of the configuration and apply standard optimization procedure to find the optimal robot configuration. A detailed design procedure is discussed in Chapter 4.

3.3.1 Insight into Tensionability Equations

There are only three (end-effector) kinematical independent parameters in these equations, F_C^0, P^n, and M; these parameters form the load vector. The cable tensions equation can be considered as a 3 by 3 transformation matrix that maps the dynamic force to the cable tensions. There are two important insights to the cable tensions equation. Firstly, the eccentric distances, ±x and ±y provide bias in the cable tensions of the parallel cable joint. The eccentric distances will always increase the range of the cable tensions under different dynamic and operating conditions. Therefore, eccentric distances should be avoided at all times. Secondly, the horizontal cable separation distance, d reduces the effect of any applied torque and all other eccentric loads. This effectively reduces the asymmetric effect on the system. One major drawback of a large d is its association to the end-effector size, which usually causes a larger inertia load. Therefore, the optimal design must be a balance between the cable separation distance and other constraints. Thirdly, the effect of the applied torque is magnified with a small cable axes angle, ¯$_A$. Another geometric interpretation of a small ¯$_A$ is that the distance between the cables in the parallel cable joint approaches to zero. This is effectively a region that is close to the singular configuration of the system. The minimum cable tension condition is expected to occur when the end-effector is close to this critical point.

3.3.2 Rewriting the Load Vector in terms of the End-Effector Kinematics

Although tensionability is a feature that is associated with the statics loading on the end-effector, it is important to ensure that all cables in the cable-robot are held under tension under a given velocity and acceleration. Moreover, there should be a minimum cable tension specification for safety purposes. Under the conditions listed in Section 3.2.1, there exists a minimum spine force to put the manipulator in a rigid pose. It is necessary to find this spine force before evaluating other mechanical performance indicators.

The static equations stated in Equation (3:15) would be a good start. However, the load vector does not constitute a good choice of independent variable as there is no a priori knowledge regarding the P^n and the M acting on the end-effector. Better choices of independent variables are the kinematic parameters such as maximum velocity and acceleration, as they are the natural measures of the robot performance. Therefore, it would be more convenient to express the cable tension vector from these kinematic parameters. In anticipation that a pneumatic cylinder is to be employed for the spine, this work derives the load vector from the inertia force that is produced by cylinders. This analysis is equally valid for any other revolute-prismatic-revolute (RPR) types of spine that produces a constant spine force. The friction and dampening effect of the pneumatic cylinder are ignored for simplicity. The lower robot would now be composed of three rigid bodies: the cylinder body, the cylinder piston, and the end-effector (the cables are considered as massless).

When a pneumatic cylinder is used as a spine, it can be considered as a RPR linkage. The equations of motion can be derived from the Newton's equations. Figure 3-9 shows the free body diagram of the two rigid bodies. In the FBD, F_C^o denotes the force from the pressurized air, F_C is the force that the piston delivers to the end-effector, and it is the equivalent to the spine force discussed in Section 3.2.1. M_c is the moment that is transmitted through the piston. P_{cy} is the normal force that the cylinder exerts on the piston interface, and P_p^n is the normal force that the piston exerts on the end-effector. Note that P_p^n does not represent the

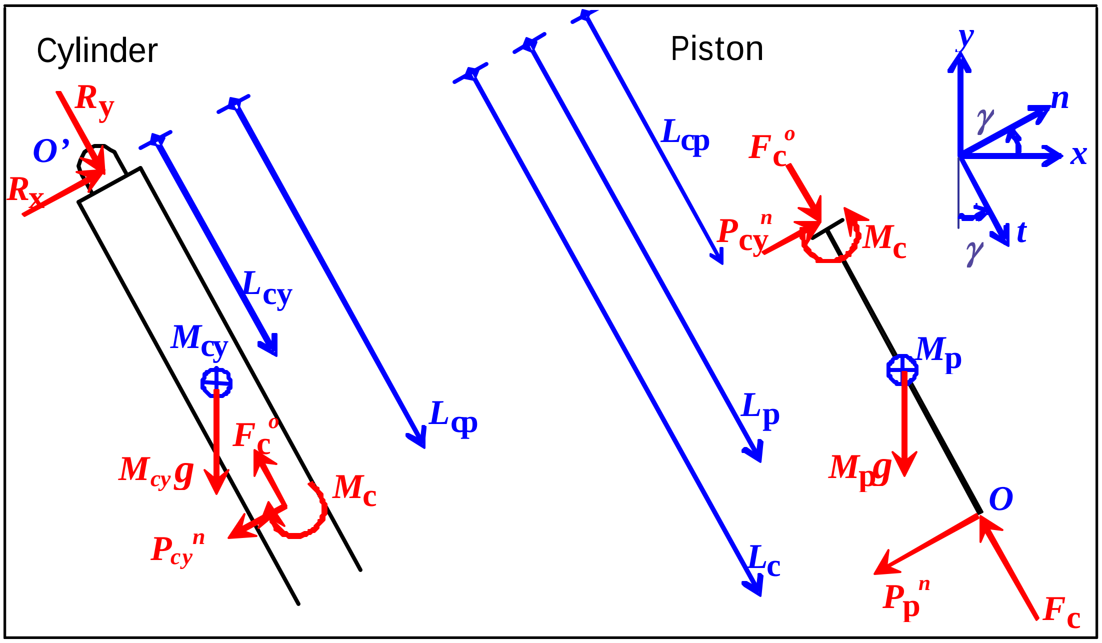

Figure 3-9: Free body diagram of the pneumatic cylinder as the spine element. All linear dimensions are measured from the pivot point O'

P^n in the load vector, it is merely a contribution of P^n.

Starting from D'Alembert's principle, the acceleration of the cylinder can be calculated from the applied force and the inertia property. Then, isolate the forces, F_C and P_p^t that the piston exerts onto the end-effector. The mathematical details are shown below. The mass component, M_{cy}, M_p, are the mass of the cylinder and piston respectively; similarly the inertia component I_{cy}^o, I_p^o, are the second moment of inertia of the cylinder and the piston with respect to the pivot point O'. The dynamic force associated with the pneumatic cylinder is discussed below:

For the tangential force, apply the force equilibrium on the piston:

$$F_C^o - F_C + M_p g\cos(\gamma_c) - M_p(\ddot{L}_c - L_p\dot{\gamma}^2) = 0 \tag{3.17}$$

$$F_C = F_C^o + M_p\left[L_p\dot{\gamma}^2 - \ddot{L}_c + g\cos(\gamma_c)\right] \tag{3.18}$$

For the normal force, apply the moment equilibrium on cylinder and the piston about pivot point O':

$$-M_c - P^n_{cy} L_{cp} - M_{cy} g \sin(\gamma) L_{cy} - I^o_{cy} \ddot{\theta} = 0 \tag{3.19}$$

$$M_c + P^n_{cy} L_{cp} - M_p g \sin(\gamma) L_p - P^n_p L_c - I^o_p \ddot{\theta} - 2 M_p L_p \dot{L}_p \dot{\gamma} = 0 \tag{3.20}$$

Adding Equation (3:19) and Equation (3:20) together produces:

$$-(M_{cy} L_{cy} + M_p L_p) g \sin(\gamma) - P^n_p L_c - \left(I^o_{cy} + I^o_p\right) \ddot{\theta} - 2 M_p L_p \dot{L}_p \dot{\gamma} = 0 \tag{3.21}$$

The normal force can be isolated from Equation (3:21):

$$P^n_p = -\frac{1}{L_c} \left[\left(I^o_{cy} + I^o_p\right) \ddot{\theta} + 2 M_p L_p \dot{L}_p \dot{\gamma} + (M_{cy} L_{cy} + M_p L_p) g \sin(\gamma) \right] \tag{3.22}$$

The other contribution of the load vector comes from the end-effector. Consider Δx and Δy to be the eccentric distances between the center of gravity of the end-effector that mass M_e and the pivot point O (see Figure 3-10), one can compute the gravitational (static) and the inertia (dynamic) force that is associated with the end-effector. The magnitude and direction of the end-effector acceleration is denoted by a and ϕ_a; the magnitude and the direction of the end-effector velocity is denoted by v and ϕ_v. This inertia force can be translated to point O and resolved into the tangential component P^t, and normal component P^n_e. The rotary effect can be compensated by an equivalent torque, which turns out to be the applied moment, M in the load vector:

$$P^t = M_e[a \sin(\phi_a - \gamma) + g \cos(\gamma)] \tag{3.23}$$

$$P^n_e = -M_e[a \cos(\phi_a - \gamma) + g \sin(\gamma)] \tag{3.24}$$

$$M = -M_e[a[\cos(\phi_a)\Delta y + \sin(\phi_a)\Delta x] + g\Delta x] \tag{3.25}$$

By the principle of superposition, the total applied load vector (F'_C, P^n, M) is the summation of the force contributions from the cylinder in Equations (3:18) and (3:22) and the dynamic force from the end-effector in Equations (3:23) to (3:25). The following equations summarizes

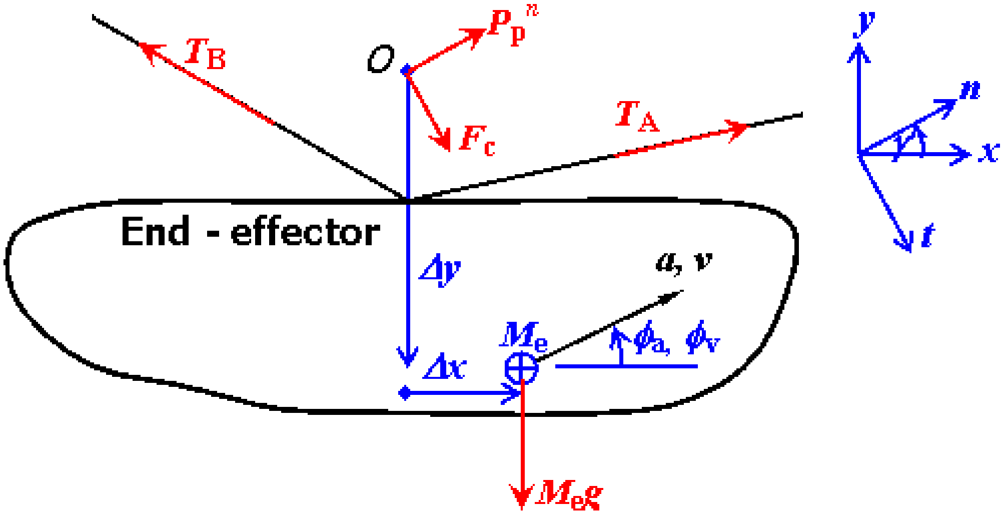

Figure 3-10: Free Body Diagram of the End-Effector

the computation of the load vector:

$$F_C^0 = P^t + F_c$$

$$F_C^0 = M_e a \sin(Á_a \text{ ¡ } °) + M_p \,\mathbf{h}\, L_c \underset{\_}{°}^2 \text{ ¡ } \overset{Ä}{L}_c \,\mathbf{i} + [M_e + M_p] g \cos(°_c) + F_C^o \qquad (3.26)$$

$$P^n = P_e^n + P_p^n$$

$$P^n = \text{¡ } M_e a \cos(Á_a \text{ ¡ } °) \text{ ¡ } \mathbf{"} \frac{\mathbf{i} I_{cy}^o + I_p^o \,\mathbf{¢}}{L_c} \ddot{Ä} + 2M \frac{L_p}{L_c} L_p \underset{\_}{°} \,\mathbf{\#} \text{ ¡ } (M_e + M_{cy} \frac{L_{cy}}{L_c} + M_p \frac{L_p}{L_c}) g \sin(°)$$

(3.27)

$$M = \text{ ¡ } M_e a [\cos(Á_a) ¢ y + \sin(Á_a) ¢ x] + M_e g ¢ x \qquad (3.28)$$

3.3.3 Identifying the Minimum Spine Force and the Maximum Cable Tension

The objective of this analysis is to identify the minimum cylinder force that should be applied to the cylinder and the maximum cable tension that can occur at the cables. It is necessary to identify the worst case parameters that minimizes the cable tensions. Since the cylinder is kinetically dependent on the end-effector, the kinetics of the cylinder should be expressed as the velocity and acceleration of the end-effector. Unfortunately, the complexity of the kinematic equations makes this problem a computationally intensive optimization process. Hence, it is not feasible to compute the exact tension minimizer. However, one can calculate the upper and lower limit from the maximum allowable end-effector velocity, V_{max} and acceleration, A_{max} on a term by term basis. The idea is to identify the range of the cylinder acceleration terms ($\ddot{\gamma}$, $\ddot{L}_c$, $\dot{\gamma}^2$, and $\dot{L}_c\dot{\gamma}$) in Equations (3.18) and (3.22). The enveloping load values can be computed using the extreme values of each acceleration component.

Let the origin of the cylinder be point O' to track the other end of the cylinder at point O located at $(x; y)$ relative to O'. The velocity and the acceleration of the end-effector is denoted according to Figure 3-10. Expressions for L_c and γ can be derived from trigonometry:

$$L_c = \sqrt{x^2 + y^2} \tag{3.29}$$

$$\gamma = \arctan\left(-\frac{y}{x}\right) \tag{3.30}$$

For the time derivatives of $\dot{L}_c$, apply implicit differentiation on $L^2 = x^2 + y^2$, and substitute $L_c \sin(\gamma)$, $-L_c \cos(\gamma)$, $v \cos(\phi_v)$, and $v \sin(\phi_v)$ for x, y, $\dot{x}$, and $\dot{y}$ respectively. A simple expression of $\dot{L}_c$ is:

$$\begin{aligned} \dot{L}_c &= \frac{x}{L_c}\dot{x} + \frac{y}{L_c}\dot{y} \\ &= \frac{v}{L_c}\left[x \cos(\phi_v) + y \sin(\phi_v)\right] \\ &= v\left[\sin(\gamma)\cos(\phi_v) - \sin(\gamma)\cos(\phi_v)\right] \\ &= -v\left[\sin(\phi_v - \gamma)\right] \end{aligned} \tag{3.31}$$

For $\ddot{L}_c$, take the derivative on the first step of Equation (3.31), and use the same substitution on x and y, but substitute $a\cos(\phi_a)$ and $a\sin(\phi_a)$ for $\ddot{x}$ and $\ddot{y}$. A simple expression of $\ddot{L}_c$ is:

$$\begin{aligned}
\ddot{L}_c &= \frac{\dot{x}^2}{L_c} + \frac{x}{L_c}\ddot{x} + \frac{\dot{y}^2}{L_c} + \frac{y}{L_c}\ddot{y} - \frac{\dot{L}_c^2}{L_c} \\
&= \frac{a}{L_c}\left[x\cos(\phi_a) + y\sin(\phi_a)\right] + \frac{v^2}{L_c} - \frac{v^2\sin^2(\phi_v - \gamma)}{L_c} \\
&= a\left[\sin(\gamma)\cos(\phi_a) - \cos(\gamma)\sin(\phi_a)\right] + \frac{v^2}{L_c}\left[1 - \sin^2(\phi_v - \gamma)\right] \\
&= -a\sin(\phi_a - \gamma) + \frac{v^2}{L_c}\cos^2(\phi_v - \gamma) \qquad (3.32)
\end{aligned}$$

Similarly, for $\dot{\gamma}$, take the derivative using the chain rule, and use above substitutions. A simple expression of $\dot{\gamma}$ is:

$$\begin{aligned}
\dot{\gamma} &= \frac{1}{1 + \left(\frac{y}{x}\right)^2}\left[-\frac{\dot{y}}{x} + \frac{y}{x^2}\dot{x}\right] \\
&= \frac{1}{L_c^2}\left[y\dot{x} - x\dot{y}\right] \\
&= -\frac{v}{L_c}\left[\cos(\gamma)\cos(\phi_v) + \sin(\gamma)\sin(\phi_v)\right] \\
&= -\frac{v}{L_c}\left[\cos(\phi_v - \gamma)\right] \qquad (3.33)
\end{aligned}$$

For $\ddot{\gamma}$, apply derivative on Equation (3:33) before the substitution. Then, use the same substitutions described above. A simple expression of $\ddot{\gamma}$ is:

$$\begin{aligned}
\ddot{\gamma} &= -\frac{2\dot{L}_c}{L_c^3}\left[y\dot{x} - x\dot{y}\right] + \frac{1}{L_c^2}\left[y\ddot{x} - x\ddot{y}\right] \\
&= \frac{2\dot{L}_c}{L_c^2}v\left[\cos(\gamma)\cos(\phi_v) + \sin(\gamma)\sin(\phi_v)\right] - \frac{a}{L_c^2}\left[\cos(\gamma)\cos(\phi_a) + \sin(\gamma)\sin(\phi_a)\right] \\
&= \frac{2\dot{L}_c}{L_c^2}v\left[\cos(\gamma - \phi_v)\right] - \frac{a}{L_c^2}\left[\cos(\gamma - \phi_a)\right] \\
&= -\frac{2v^2}{L_c^2}\sin(\phi_v - \gamma)\cos(\gamma - \phi_v) - \frac{a}{L_c^2}\left[\cos(\gamma - \phi_a)\right] \qquad (3.34)
\end{aligned}$$

It is convenient to have the expressions for $\dot{\gamma}^2$ and $\dot{L}_c\dot{\gamma}$ because these terms appear frequently in Equations (3:18) and (3:22) as the centrifugal and Coriolis accelerations. From Equations

Table 3.1: Parameter to calculate the Extreme Load Vector

Load	Eq'n	Value	Centrifugal $\dot{\gamma}^2$	Coriolis $2L_c\dot{\gamma}$	Tangential $\ddot{L}_c$	Angular $\ddot{\gamma}$	End-Effector (dynamic)
F_c^0	3.26	Min	Lower	N/A	Upper	N/A	$-M_eA_{max}$
		Max	Upper	N/A	Lower	N/A	M_eA_{max}
P^n	3.27	Min	N/A	Upper	N/A	Upper	M_eA_{max}
		Max	N/A	Lower	N/A	Lower	$-M_eA_{max}$
M_o	3.28	Min	N/A	N/A	N/A	N/A	$-M_eA_{max}\begin{vmatrix}\Delta x\\ \Delta y\end{vmatrix}$
		Max	N/A	N/A	N/A	N/A	$-M_eA_{max}\begin{vmatrix}\Delta x\\ \Delta y\end{vmatrix}$

(3:31) and (3:33), these expressions are:

$$\dot{\gamma}^2 = \frac{v^2}{L_c^2}\cos^2(\phi_v - \gamma) \tag{3.35}$$

$$L_c\dot{\gamma} = \frac{v^2}{L_c}\frac{\sin(2(\phi_v - \gamma))}{2} \tag{3.36}$$

Since $\sin(\theta) \in [-1;1]$ and $\cos^2(\theta) \in [0;1]$, the extreme accelerations can be computed from Equations (3:32) to (3:36). The ranges of these acceleration terms are:

$$0 \le \dot{\gamma}^2 \le \frac{V_{max}^2}{L_c^2} \tag{3.37}$$

$$-\frac{V_{max}^2}{L_c} \le 2L_c\dot{\gamma} \le \frac{V_{max}^2}{L_c} \tag{3.38}$$

$$-A_{max} \le \ddot{L}_c \le A_{max} + \frac{V_{max}^2}{L_c} \tag{3.39}$$

$$-\left(\frac{V_{max}^2}{L_c^2} + \frac{A_{max}}{L_c}\right) \le \ddot{\gamma} \le \frac{V_{max}^2}{L_c^2} + \frac{A_{max}}{L_c} \tag{3.40}$$

The enveloping load value can be computed by applying the upper (lower) bound of the acceleration on Equations (3:26), (3:27), and (3:28). Table 3.1 summarizes the combination of these acceleration terms in computing the extremum of the load vector. The overall mechanical quantity can be computed using the values in the J matrix and the extremum value found in Table 3.1.

The linearity properties of the dynamic equations guarantees that the minimum spine force exists at the extremum of the load parameters. By rearranging Equations (3:15) and (3:26), one can compute a set of linear equations to find the minimum necessary cylinder force, F_C^{min} (highest lower bound of the cylinder force). The following procedure [1] can be used to identify the F_C^{min} of a particular robot pose given a certain value of minimum cable tension, T^{min}.

Step 1: Calculate $F_C^{*} = (M_p + M_e)(g\cos(\gamma) - A_{max}) - M_p \frac{V_{max}^2}{L_c}$

Step 2: for k=1..3 for each cable, set

$$P_k^n = \left\{ \begin{array}{ll} \max P^n & \text{if } J_{k2} \le 0 \\ \min P^n & \text{otherwise} \end{array} \right\}$$

$$M_{ok} = \left\{ \begin{array}{ll} \max M & \text{if } J_{k3} \le 0 \\ \min M & \text{otherwise} \end{array} \right\}$$

Step 3: Calculate $F_C^{min} = \min\limits_{workspace} \left(\max_k \frac{T_k^{min} - J_{k;2}P^n - J_{k;3}M_k}{J_{k;1}} - F_C^{*} \right)$

Proof. The minimum necessary cylinder (spine) force is the force that set the cable tension to a minimum tension T^{min}. From Equation (3:15), $T_k = \lfloor J_k \rfloor \{L\}$, or $F_C^0 = \frac{T_k^{min} - J_{k;2}P^n - J_{k;3}M_k}{J_{k;1}}$. The cylinder force, F_C^o can be isolated from Equation (3:26), which produces:

$$F_C^o = \frac{T_k^{min} - J_{k;2}P^n - J_{k;3}M_{ok}}{J_{k;1}} - F_C^{*} \qquad (3.41)$$

where F_C^{*} is computed in Step 1, and Equation (3:41) is similar to the expression in Step 3. A tensionable cable system implies $J_{k;1} > 0$. Therefore, the minimum necessary cylinder force occurs when the normal force and the moment terms take the most negative value as illustrated in Step 2. However, Equation (3:41) computes the minimum cylinder force for just one cable. It is necessary to find the biggest F_C^o amongst the three cables to ensure all three cables meet the minimum cable tension requirement. Therefore, the minimum necessary cylinder force is the maximum of all F_C^o. ■

Another important quantity is the maximum cable tension T^{max} when the minimum necessary cylinder force is applied to the end-effector. Unlike the computation of the minimum necessary cylinder force, the maximum cable tension analysis requires the load vector to maximize its value. Therefore, its process is almost opposite to the computation of T^{min}. Equation (3:15) suggests the following procedure to calculate T^{max}:

Step 1: Calculate $F_c^0 = \max F_c^0$

[1] This procedure assumes the tensionablilty condition described in previous sections has been met.

Step 2: for k=1..3 for each cable, set

$$P_k^n = \left\{ \begin{array}{ll} \min P^n & \text{if } J_{k2} \le 0 \\ \max P^n & \text{otherwise} \end{array} \right\}$$

$$M_{ok} = \left\{ \begin{array}{ll} \min M & \text{if } J_{k3} \le 0 \\ \max M & \text{otherwise} \end{array} \right\}$$

Step 3: Calculate $T_k^{max} \le \left(\lfloor J_k \rfloor \left\{ \begin{array}{c} F_C^0 \\ P_k^n \\ M_k \end{array} \right\} \right)$

Chapter 4

Design Considerations and Optimization Method

Since a general purpose programmable manipulator generally does not have any preference in operating orientation, it is reasonable to expect the optimal design will have some symmetry about the Y-axis. From a functional point of view, the robot must be both stiff and light enough to satisfy the cycle rate and repeatability requirements. Moreover, the robot should not take up too much space, yet its workspace should be relatively large for different operating environments. All of these factors pose different conflicting design factors, which a designer must compromise. This chapter explores the process of obtaining optimal robot design parameters X, that mostly consists of robot dimensions. Section 4.1 presents typical design considerations and performance indicators (PI) for the proposed planar manipulator. Section 4.2 introduces the design optimization as a process involving a feasibility test to ensure the resulting design is tensionable. This feasibility test is performed across the robot workspace, and it is implemented as an inner loop nested within an outer loop that optimizes the robot design parameters. Section 4.3 and Section 4.4 detail these inner and outer loops respectively.

4.1 Design Constraints and Criteria

Two design constraints are the usable workspace and the tensionability within the workspace. The former constraint is purely a geometric issue, and it involves the overall robot topology. The latter constraint is local to the end-effector and is addressed in Chapter 3. A typical workspace is specified as a rectangle by its width and height. The actual dimension of the workspace is heavily dependent on the application. In this work, the dimension of the workspace is set to be rectangular in shape. Another design constraint is the robot cycle period; the manipulator should be driven at high speed with a minimal amount of power. As far as cycle period is concerned, the ability to accelerate would be a better indicator since the acceleration and deceleration phases are likely to dominate a cycle period for high speed operations. Another advantage of using acceleration over speed as an indicator is that acceleration provides other mechanical conditions that are useful for evaluating the cost function. A set of robot design parameters are considered to be infeasible if they do not meet either the design constraints or the tensionability constraint.

As for the performance, spatial consideration such as machine footprint should also be integrated in the decision process. This is because space is usually a precious resource in a typical industrial environment. The relevant geometrical measurements include the robot width and the robot height. Other performance indicators (PI) of interest are mostly mechanical properties. The performance indicators are listed as follow:

- The footprint (width) of the design, w
- The height of the design, h
- The minimum cylinder force, F_c^{min}
- The maximum cable tension, T^{max}
- The maximum actuator power, W^{max}
- The maximum actuator torque, M^{max}
- The maximum position error factor, $\pm P^{max}$

The cost of a particular robot design (a set of dimension to define a robot) would be a function of these performance indexes. The computations of the mechanical parameters are discussed in Section 4.3.

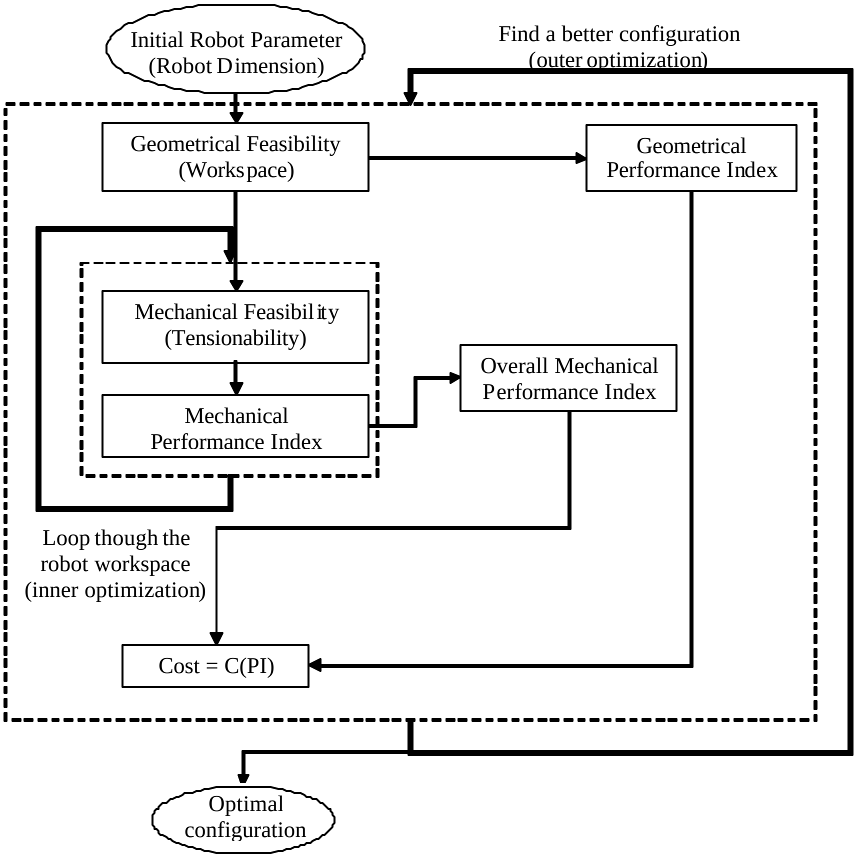

Figure 4-1: Generalized Block Diagram for the Optimization Process

4.2 Optimization Methods Overview

The proposed optimization algorithm consists of two loops as indicated in Figure 4-1; the inner loop concerns the tensionability issue and mechanical properties, while the outer loop optimizes the robot design parameters X. A cost function that is a function of the design criteria ($C = C(PI)$) is constructed to compare two different sets of design parameters. Section 4.3 and Section 4.4 give details on this general optimization structure and the computation of the performance indexes from a robot design parameter ($PI = PI(X)$).

Since tensionability is a system constraint, the inner optimization loop must be able to identify the global extremum for its targeted parameters. While modern optimization algorithms are relatively efficient on global convergence, they do not guarantee to converge to the global extremum [3]. Unless a particular robot has a convex feasibility function, modern optimization algorithm may converges to a local extremum and does not produces the most conservative result. Therefore, a sequential search method would be a more appropriate choice for the inner optimization loop.

As for the outer optimization loop, geometric parameters are the parameters of interest. Since they are the criteria for comparison purpose only, an effective optimization algorithm should be used to find the optimal or a near optimal solution. In order to increase the possibility of obtaining the global optimal design parameters, the outer optimization loop has to be repeated multiple times at different starting points for a complete optimization search.

A set of mechanical indicators are evaluated for each set of robot design parameters, the overall cost function can be a weighted sum of these cost indicators. Some of them, namely the mechanical indicators described in Section 4.3 can be measured by the extreme value and the mean of the extreme value. The extreme values are of particular importance as they dictate the specification of the mechanical components. The mean extreme values should also be considered as they are better representations of the mechanical resource requirements for the maximum robot performance.

In this work, the proposed optimization scheme is performed in two phases. The first phase computes a set of the local optimal design parameters from a random set of starting design parameters, the cost function uses the extreme indicator values only. In the second phase, the inner optimization loop performs a higher resolution search. Furthermore, the mean extreme indicator values as well as the absolute extreme indicator values are used for computing the design cost. The starting points of the second phase optimization are the local optimal design parameters obtained in the previous phase.

4.3 Computation of the Necessary Mechanical Indicators - The Inner Optimization Loop

The purpose of the inner optimization loop is to produce the feasibility information and the mechanical performance indexes listed in Section 4.1. This can only be accomplished by calculating various properties across the workspace.

The first computation is the robot feasibility, which can be ensured if the manipulator is tensionable across its entire workspace. Under certain conditions enlisted in Section 3, the manipulator is always tensionable regardless of the external load. Hence, they will be in a rigid pose as long as there is a sufficient amount of cylinder force to balance the external load. As explained in Subsection 3.2.1, this work assumes a particular design parameter fails to meet the tensionability criterion if there is a negative entry in the first column of the J matrix at anywhere within the designed workspace. When such a violation occurs, the inner loop should either terminate the whole optimization process (hard constraint), or output an exponentially large value as the search approaches to the boundary of the infeasible region of the robot configuration space (barrier function). The implementation of the actual inner optimization loop needs to match the designs of the outer optimization loop.

As mentioned above, it is critical to find the global optimizer on several mechanical properties (F_c^{min}, T^{max}, W^{max}) in the inner optimization loop. It is more appropriate to solve this problem by a sequential search. The idea is to discretize the workspace into a lattice of vertices, and evaluate the mechanical properties at each and every vertex until the extreme values are found. This particular implementation calls for a pneumatic cylinder as the spine element. It is assumed that the cylinder force is held constant across the entire workspace. To implement the algorithm in a computationally effective manner, it is advantageous to have an initial pass to compute J matrix by Equation (3.15) at each vertex. Tensionability and other mechanical indicators can be computed easily from J matrix afterward.

4.3.1 Inverse Kinematics and the Computation of the Basic Mechanical Properties

Before executing the inner optimization loop, one should derive the inverse kinematic model and the cable axis expressions, which are needed when computing J matrix. Both of these expressions map the workspace coordinate to its respective space coordinate:

$$\theta = \theta(x,y); \quad \theta \subseteq f : R^2 \rightarrow R^2 \tag{4.1}$$

$$\beta = \beta(x,y); \quad \beta \subseteq f : R^2 \rightarrow R^2 \tag{4.2}$$

The inverse kinematics equation, µ in Equation (4:1), and the cable axes equation ¯ in Equation (4:2) are dependent on the upper robot assembly and the robot design parameters. Moreover, the computation of the equation of motions also require the cylinder length L_c in Equation (3:29) and the spine axis angle ° in Equation (3:30), which are independent from the upper robot assembly assuming a RPR type of spine is used. In the initial pass of the inner loop, one should calculate J matrix at each vertex of the grid. If the design parameters were not inherently tensionable (as described in Case E of Section 3.2.1), the inner optimization loop should store the minimum value of J_{i1}. This is needed later for a global convergence. At the same time, the minimum cylinder force, F_c^{min} can be computed by the procedure shown in Section 3.3.3.

Once J matrix has been obtained and the F_c^{min} has been determined, other basic mechanical properties can be computed hereafter. It is essential to compute the global F_c^{min} before computing other mechanical indicators such as the maximum cable tension (using the biggest minimum necessary cylinder force of all poses) as most of these indicators monotonically increases with the cylinder force. A typical scenario is to find the worst case mechanical indicators at a bounded end-effector acceleration. In this case, the dynamic effect of the cylinder (central assembly) could be significant. A possible procedure to compute the minimum required cylinder force and the maximum cable tension at a given pose is shown in Section 3.3.3.

4.3.2 Computation of the Position Error Factor

Another important performance index is the position error factor, ±P. The position error factor is defined to be the maximum amount of the position error ($\tilde{x}$, $\tilde{y}$) in the workspace coordinate due to a slight deviation of actuator angle ($\tilde{\mu}_1; \tilde{\mu}_2$) in the robot actuator space. In this work, a first order approximation technique is used to approximate the error. The mathematical formulation is:

$$\pm P = \left\| \frac{1}{\tilde{\mu}_1\ \tilde{\mu}_2} \right\| \left(\sup_{\| \lfloor \tilde{\mu}_1\ \tilde{\mu}_2 \rfloor \| < "} \left\{ \begin{matrix} \tilde{x} \\ \tilde{y} \end{matrix} \right\} \right) \qquad (4.3)$$

where b:c denotes a row matrix, f:g denotes a column matrix, and k:k denotes a general vector norm.

The position perturbation term in Equation (4:3) can be estimated by first order on approximation:

$$\begin{Bmatrix}\tilde{x}\\ \tilde{y}\end{Bmatrix} = \begin{Bmatrix}x(\theta_1+\tilde{\theta}_1,\theta_2+\tilde{\theta}_2)\\ y(\theta_1+\tilde{\theta}_1,\theta_2+\tilde{\theta}_2)\end{Bmatrix} - \begin{Bmatrix}x(\theta_1,\theta_2)\\ y(\theta_1,\theta_2)\end{Bmatrix}$$

$$= \begin{Bmatrix}x(\theta_1,\theta_2)\\ y(\theta_1,\theta_2)\end{Bmatrix} + 5\begin{Bmatrix}(\theta_1,\theta_2)\\ (\theta_1,\theta_2)\end{Bmatrix}\begin{Bmatrix}\tilde{\theta}_1\\ \tilde{\theta}_2\end{Bmatrix} - \begin{Bmatrix}x(\theta_1,\theta_2)\\ y(\theta_1,\theta_2)\end{Bmatrix} + O(2)$$

$$\approx \begin{bmatrix}\frac{dx}{d\theta_1} & \frac{dx}{d\theta_2}\\ \frac{dy}{d\theta_1} & \frac{dy}{d\theta_2}\end{bmatrix}\begin{Bmatrix}\tilde{\theta}_1\\ \tilde{\theta}_2\end{Bmatrix}$$

Note that the mapping matrix is the velocity Jacobian matrix of the manipulator, J_v. Therefore, δP simplifies to:

$$\delta P \approx \frac{\sup\left\| J_v\begin{Bmatrix}\tilde{\theta}_1\\ \tilde{\theta}_2\end{Bmatrix}\right\|}{\left\|\,|\tilde{\theta}_1\;\tilde{\theta}_2|\,\right\|} = \|J_v\| \tag{4.4}$$

in this case $\|J_v\|$ is the matrix norm of J_v

One small problem is that the forward kinematics equations for parallel mechanism are usually derived as a set of implicit equations, or a set of relatively complex explicit expressions. This may raise runtime issues when they are computed many times in the sequential search, and the computation of the Jacobian matrix is likely going to be more time consuming. Fortunately, it is possible to derive the Jacobian matrix from the inverse kinematics using inverse velocity analysis [16]. Moreover, it turns out that part of this computation is also required in computing the maximum actuator torque and power in Section 4.3.3. Therefore, this indirect approach does not add too much computation burden in the overall analysis.

Suppose Equation (4:1) has been derived, the first order approximation on some small perturbation in the workspace coordinate suggests a set of equivalent equations that is similar to J_v:

$$\begin{Bmatrix} \theta_1(x+\delta x, y+\delta y) \\ \theta_2(x+\delta x, y+\delta y) \end{Bmatrix} = \begin{Bmatrix} \theta_1(x,y) \\ \theta_2(x,y) \end{Bmatrix} + \nabla\theta \begin{Bmatrix} \delta x \\ \delta y \end{Bmatrix}$$
$$\begin{Bmatrix} \delta\theta_1 \\ \delta\theta_2 \end{Bmatrix} = \begin{bmatrix} \frac{d\theta_1}{dx} & \frac{d\theta_1}{dy} \\ \frac{d\theta_2}{dx} & \frac{d\theta_2}{dy} \end{bmatrix} \begin{Bmatrix} \delta x \\ \delta y \end{Bmatrix} \tag{4.5}$$

It follows that the manipulator Jacobian matrix, J_v is the inverse of the inverse kinematics gradient function, $[\nabla\theta]^{-1}$:

$$J_v = [\nabla\theta]^{-1} \tag{4.6}$$

From Equation (4:4), it follows that the position error factor is some form of the norm of $[\nabla\theta]^{-1}$. This work assumes the position deviation is caused by an independent normally distributed random error in the actuators. As a result, the norm to be employed is the (matrix) euclidean norm, which makes δP become a proportional constant between the position deviation and the standard deviation of a normally distributed actuator errors:

$$\delta P = \left\| [\nabla\theta]^{-1} \right\|_2 \tag{4.7}$$

In some applications, a designer may be required to design a manipulator against the worst case scenario. In those cases, a 1-norm would be a more appropriate choice as it makes δP become a measurement of position deviation due to a uniformly distributed random actuator errors.

Since the position error factor is to be evaluated at every node of the inner optimization loop, it is suggested to simplify the computation as much as possible. Mathematically speaking, the euclidean norm is the same as the largest singular value of the $[\nabla\theta]^{-1}$, which is also equal to the square root of the largest eigenvalue of $\left[\nabla\theta^{-1}\right]^T \left[\nabla\theta^{-1}\right]$. Since $\left[\nabla\theta^{-1}\right]^T \left[\nabla\theta^{-1}\right] = \left[\nabla\theta^T\ \nabla\theta\right]^{-1}$ is a positive definite matrix providing the manipulator is not in a singular pose, it can be shown that the matrix inversion step in Equation (4:7) is not necessary. An equivalent equation is:

$$\pm P = \frac{1}{\sqrt{\min\left(\mathrm{eig}\left(5\mu\ 5\ \mu^T\right)\right)}} \qquad (4.8)$$

where eig is the function that calculates the eigenvalue of a square matrix.

4.3.3 Computation of the Maximum Actuator E¤ort and Power

Another mechanical indicator to be explored is the maximum actuator e¤ort and power based on the desired maximum velocity. The following analysis is based on a rotary actuator and the rotary e¤ort - Torque. The same analysis is equally valid with linear actuator and linear e¤ort - force. Unlike other mechanical parameters, the actuator e¤orts and powers involve a significant amount of dynamic force from the upper robot assembly. Consider a cable attached to a rotary actuator i through a rigid linkage, the torque that the actuator delivers to the system, M_i and the power that the actuator delivers to the system W_i can be calculated by the following equations:

$$M_i = T_i \pounds L_i + I_i\ddot{\mu}_i + m_i\bar{L}_i\cos(\mu)g \qquad (4.9)$$

$$W_i = M_i\mu_i \qquad (4.10)$$

where, μ_i is the direction of rotation measured from horizontal

T_i is the cable tension

L_i is the length of the moment arm of the rigid linkage

$\bar{L}_i$ is the distance between the center of mass of the rigid linkage and the pivot point

I_i is the mass moment of inertia of the rigid linkage

m_i is the mass of the rigid linkage

This torque is the summation of the applied torque from the cable, the inertia torque from the rigid linkage, and the gravitational force from the mass of the linkage. The actual maximum joint torque is difficult to compute in general, however it is possible to establish an upper bound on the joint power by assuming the maximum applied torque coexists with the maximum inertia torque. Since there is no limitation imposed on the joint speed and acceleration, it is necessary to relate the joint kinetics to the workspace kinetics. Expressions can be obtained by taking the time derivative of the inverse kinematic equation (4.1) as shown below:

$$\begin{aligned}\dot{\theta}_i &= \nabla\theta_i(x,y)\begin{Bmatrix}\dot{x}\\ \dot{y}\end{Bmatrix}\\ &= \begin{bmatrix}\frac{\partial\theta_i}{\partial x} & \frac{\partial\theta_i}{\partial y}\end{bmatrix}\begin{Bmatrix}\dot{x}\\ \dot{y}\end{Bmatrix} \qquad (4.11)\\ \ddot{\theta}_i &= \nabla\theta_i(x,y)\begin{Bmatrix}\ddot{x}\\ \ddot{y}\end{Bmatrix} + \begin{bmatrix}\dot{x} & \dot{y}\end{bmatrix} H(\theta_i(x,y))\begin{Bmatrix}\dot{x}\\ \dot{y}\end{Bmatrix}\\ &= \begin{bmatrix}\frac{\partial\theta_i}{\partial x} & \frac{\partial\theta_i}{\partial y}\end{bmatrix}\begin{Bmatrix}\ddot{x}\\ \ddot{y}\end{Bmatrix} + \begin{bmatrix}\dot{x} & \dot{y}\end{bmatrix}\begin{bmatrix}\frac{\partial^2\theta_i}{\partial x^2} & \frac{\partial^2\theta_i}{\partial x\partial y}\\ \frac{\partial^2\theta_i}{\partial x\partial y} & \frac{\partial^2\theta_i}{\partial y^2}\end{bmatrix}\begin{Bmatrix}\dot{x}\\ \dot{y}\end{Bmatrix} \qquad (4.12)\end{aligned}$$

Since the magnitude of the end-effector velocity and acceleration are bounded by V_{max} and A_{max}, the maximum actuator velocity and acceleration can be obtained by solving the constrained optimization equations. First consider the joint velocity, it is in a linear form with respect to the end-effector velocity. From linear algebra, the least upper bound (supremum) and the biggest lower bound (infimum) of the expression is $\pm\|\nabla\theta_i\|_2 V_{max}$.

As for the joint acceleration expression in Equation (4.12), it consists of two linearly independent expressions. The extremum of $\ddot{\theta}_i$ exists at the extreme value of these expressions. The first expression is related to the direct acceleration, and it is in a linear form with respect to the end-effector acceleration; therefore, the magnitude of the first term is bounded by $\|\nabla\theta_i\|_2 A_{max}$. The second joint acceleration expression is related to Christoffel symbols, and it is in a quadratic form with respect to the end-effector velocity. Since the Hessian of a smooth function is always symmetrical, the extreme values from the second expression are related to the eigenvalues, λ of the Hessian matrix. There are several possibilities on the resulting extremum values in a quadratic form listed in Table 4.1:

An observation from Table 4.1 shows that the extremum values of the second expression are $V_{max}^2\max(\lambda, 0)$ and $V_{max}^2\min(\lambda, 0)$. The following equations concisely compute the joint

Table 4.1: The extremum value from the second joint acceleration expression

Case	Positive Definiteness	Eigenvalue	Least Upper Bound	Biggest Lower Bound
1	Positive Definite	$¸_1 > 0$ $¸_2 > 0$	$V^2_{max}¸_1$	0
2	Positive Semidefinite	$¸_1 > 0$ $¸_2 = 0$	$V^2_{max}¸_1$	0
3	Indefinite	$¸_1 > 0$ $¸_2 < 0$	$V^2_{max}¸_1$	$V^2_{max}¸_2$
4	Negative Semidefinite	$¸_1 = 0$ $¸_2 < 0$	0	$V^2_{max}¸_2$
5	Negative Definite	$¸_1 < 0$ $¸_2 < 0$	0	$V^2_{max}¸_2$

kinetics extremum:

$$µ_i^+ = V_{max}\, k\, 5 µ_i k_2 \tag{4.13}$$

$$\ddot{µ}_i^+ = A_{max}\, k\, 5 µ_i k_2 + V^2_{max} \max(¸_{max}; 0) \tag{4.14}$$

$$µ_i^{¡} = ¡\, V_{max}\, k\, 5 µ_i k_2 \tag{4.15}$$

$$\ddot{µ}_i^{¡} = ¡\, A_{max}\, k 5 µ_i k_2 + V^2_{max} \min(¸_{min}; 0) \tag{4.16}$$

where $µ_i^+$, $\ddot{µ}_i^+$ represent the upper bound of the joint velocity and acceleration respectively, and

$µ_i^{¡}$, $\ddot{µ}_i^{¡}$ represent the lower bound of the joint velocity and acceleration respectively.

Typical actuators do not have a preference on the operating direction, except for double acting cylinders, which provide slightly more force in the pushing direction than in the pulling direction. Therefore, it is reasonable to assume that the torque capacity of the actuator is the bigger of the forward torque and the reverse torque. From the computational point of view, it is slightly more efficient to compute the reverse torque to be a positive value in the reverse direction. Since the tension component always provides a forward torque, a maximum reverse torque must occur with a minimal tension and the most negative acceleration. The above arguments result in the following equation that computes the maximum actuator torque requirement that always produces a positive torque:

$$M_i^{max} = \max\left\{T_i^{max} \times L + I_i\ddot{\theta}_i^{+} + m_i\bar{L}_i\cos(\theta)\,g;\ -T_i^{min} \times L - I_i\ddot{\theta}_i^{-} - m_i\bar{L}_i\cos(\theta)\,g\right\} \qquad (4.17)$$

If it is further assumed the amount of the sinking power be the same as the amount of the sourcing power in the actuator, the maximum power requirement of the actuator i would be the higher of the sinking power and the sourcing power. In this case, the reverse torque should be considered as a negative torque in the forward direction as indicated in the following equation:

$$W_i^{max} = \max\left\{\left(T_i^{max} \times L + I_i\ddot{\theta}_i^{+} + m_i\bar{L}_i\cos(\theta)\,g\right)\theta_i^{+};\ \left(T_i^{min} \times L + I_i\ddot{\theta}_i^{-} + m_i\bar{L}_i\cos(\theta)\,g\right)\theta_i^{-}\right\} \qquad (4.18)$$

Note that the 2nd term in Equation (4:18) produces a positive power by multiplying a negative torque with a negative velocity; it is needed for comparison purposes.

4.4 Computation of the Desired Geometrical and Performance Indicators - The Outer Optimization Loop

The objective of the outer optimization loop is to find a set of robot design parameters that has a low design cost. Since this optimization does not require the global optimal result, a simple and effective direct search optimization algorithm is used to identify a set of optimal or near optimal robot design parameters. The simplex method was chosen to implement in this work. This step requires the kinematic model of the entire robot, hence, the topology of the upper robot must be known. A simplex algorithm and other direct search methods are described in [21].

Direct search algorithms typically do not produce the global optimal solution. Thus, this outer optimization is run multiple times to increase the probability of obtaining a set of the global optimal design parameters. The n-parameters that are required to constrain a robot design is denoted to be $X \subset R^n$. In this work, a variation of the simplex algorithm is chosen for optimization purposes. To set up a simplex algorithm, an initial robot design parameter X_0 should be randomly chosen in the first phase of optimization. Feasibility should be checked for the initial design parameters. If the design parameters are feasible, a set of n + 1 robot design parameter that forms a convex-hull near to X_0 is chosen as the initial simplex. One way to choose such a set of design parameter is to randomly add a small normalized deviation to X_0. The small deviation can be generated by a Gaussian distribution routine (and normalized afterward) for an unbiased result. The resulting simplex does not always inscribe X_0, but its centroid will be closed to the feasible design parameter. Regardless of how these robot parameters were generated, it is essential to check for feasibility at each and every single design parameter. This is because there is no guarantee that the newly generated design parameters are also laying inside the feasible region of the design configuration space. Once the initial simplex is chosen, the simplex algorithm can begin to execute. Upon the convergent of the simplex, the best design parameter amongst the $n + 1$ points in the configuration space is saved for the second phase of analysis.

4.4.1 Modified Dynamic Simplex Algorithm

Simplex algorithm, just like any other direct search method can guarantee a global convergence to a stationary point only if there is a sufficiently rich set of feasible search directions available in the algorithm [22]. This is particularly challenging if there is a non-linear and/or if there is no a priori knowledge on the constraint function. Although modern direct search algorithms that utilize stochastic techniques provide a stronger convergent characteristic [15], they are relatively difficult to implement and often require several expert parameter selections for the finest performance. Therefore, this work implements a simple dynamic simplex algorithm with L1 logarithmic penalty function to achieve a near optimal result.

Instead of establishing the entire constrained optimization method, it is more straightforward to introduce the unconstrained simplex search algorithm. Let $F(X)$ be the design cost function to minimize and Y^i be the cost value of the i^{th} robot design parameter in the simplex set X^i, the Nelder and Meld simplex algorithm calls for successive reflection on the most costly set of design parameters [17]. For the purpose of completeness, the algorithm is reproduced with specific elaborations on this design problem.

The Nelder and Meld Simplex:

Set $K = 0$

Step 1 Reflection Step:

Let X^{min} and X^{max} be the least and most costly design parameter and $\bar{X}$ be the centroid (mean) of all $n + 1$ design parameters, compute

$X_{ref} = 2\bar{X} - X^{max}$.

Step 2: Contraction / Expansion Step:

case 1: If $F(X_{ref}) < F(X^{min})$, attempt expansion :

$$\begin{pmatrix} X_{new} \\ K \end{pmatrix} = \begin{cases} \begin{pmatrix} 2X_{ref} - \bar{X} \\ 0 \end{pmatrix} & \text{if} \quad F\left(2X_{ref} - \bar{X}\right) < F(X_{ref}) \\ \begin{pmatrix} X_{ref} \\ 0 \end{pmatrix} & \text{otherwise} \end{cases}$$

case 2: if $F(X_{ref}) > F(X^{max})$, attempt contraction

$$\begin{pmatrix} X_{new} \\ K \end{pmatrix} = \begin{cases} \begin{pmatrix} \frac{X_{ref}+\bar{X}}{2} \\ 0 \end{pmatrix} & \text{if} \quad F\left(\frac{X_{ref}+\bar{X}}{2}\right) < F(X^{min}) \\ \begin{pmatrix} \frac{X_{ref}+\bar{X}}{2} \\ K+1 \end{pmatrix} & \text{if} \quad F(X^{min}) \le F\left(\frac{X_{ref}+\bar{X}}{2}\right) < F(X^{max}) \\ \begin{pmatrix} X_{ref} \\ n+1 \end{pmatrix} & \text{otherwise} \end{cases}$$

Step 3: Replace the vertex

If $K < n + 1$, replace X^{max} with X_{new}

otherwise contract the entire simplex: set $X_i = \frac{(X_i+\bar{X})}{2}$

Step 4: Check for convergence

Here the characteristic norm of the simplex is defined to be the largest Euclidean distance between the simplex centroid and its vertices. If the characteristic norm is below a certain predefined threshold σ^{max}, the simplex algorithm is said to be converged and the result of the search is X^{min}.

Otherwise, go back to Step 1

However, the design problem requires a constrained optimization algorithm. In order to avoid hitting the infeasible region, a modified objective function that takes the degree of the admissibility into consideration is employed. This approach is known as the L1 penalty method [21]. The underlying principle is to add a penalty cost as the searching sequence approaches to the boundary of the admissible region, but the penalty is de-weighted as the sequence converges [21]. Let $F(X)$ and $C(X)$ be the cost function to be minimized and the constraint function for which a feasible robot design parameter would produce a positive value, the effective objective function can be written as a weighted sum between $F(X)$ and $C(X)$:

$$G(x) = F(X) + {}^{1} \log(C(X)) \tag{4.19}$$

The weighting parameter ^{1}, is designed such that it will asymptotically approach to zero as the search sequence converges. In this case, the weighting parameter can be the characteristic norm of the simplex described in Step 4 of the presented simplex algorithm. This penalty approach prevents the search sequence from stepping over the constraint boundary, while introducing minimal amount of disturbance to the cost function during the search. The natural choice of the constraint function for the tensionability requirement would be the smallest value of the column of J matrix as the system becomes untensionable when the first column of J matrix becomes zero. The simplex algorithm enlisted above can be adopted by simply replacing $F(X)$ with $G(X)$.

4.5 Summary of the Optimization Process

There are two phases in the optimization process. In the first phase, the optimization outlined in Figure 4-2 is run multiple times with different randomly selected starting design parameters, X^{0}. It is recommended to perform at least 200 iterations in the first phase of the optimization process. Within each call of this outer optimization loop, an inner optimization loop is called several times as indicated by the shaded blocks in Figure 4-2. The inner loop is shown in Figure 4-3. In this inner loop, a sequential search technique is employed to calculate the mechanical parameters within the workspace. The local optimal design parameters $X^{¤}$ that are computed in the first phase are saved for the 2nd phase optimization.

The second phase of the optimization process is similar to the first phase with two key changes:

1. The inner loop of the second phase uses a denser mesh and looks for the mean of the mechanical performance indicators to compute a design cost.

2. The second phase analysis begins with the promising local optimal design parameters that are obtained from the previous phase. Since the second phase of the optimization process begins with relatively good starting points, it does not require as many iteration as in the first phase of the optimization process. It is recommended reprocess the best 25% of the local optimal design parameters.

The overall optimal design is the design parameter that has the lowest cost value in the second phase.

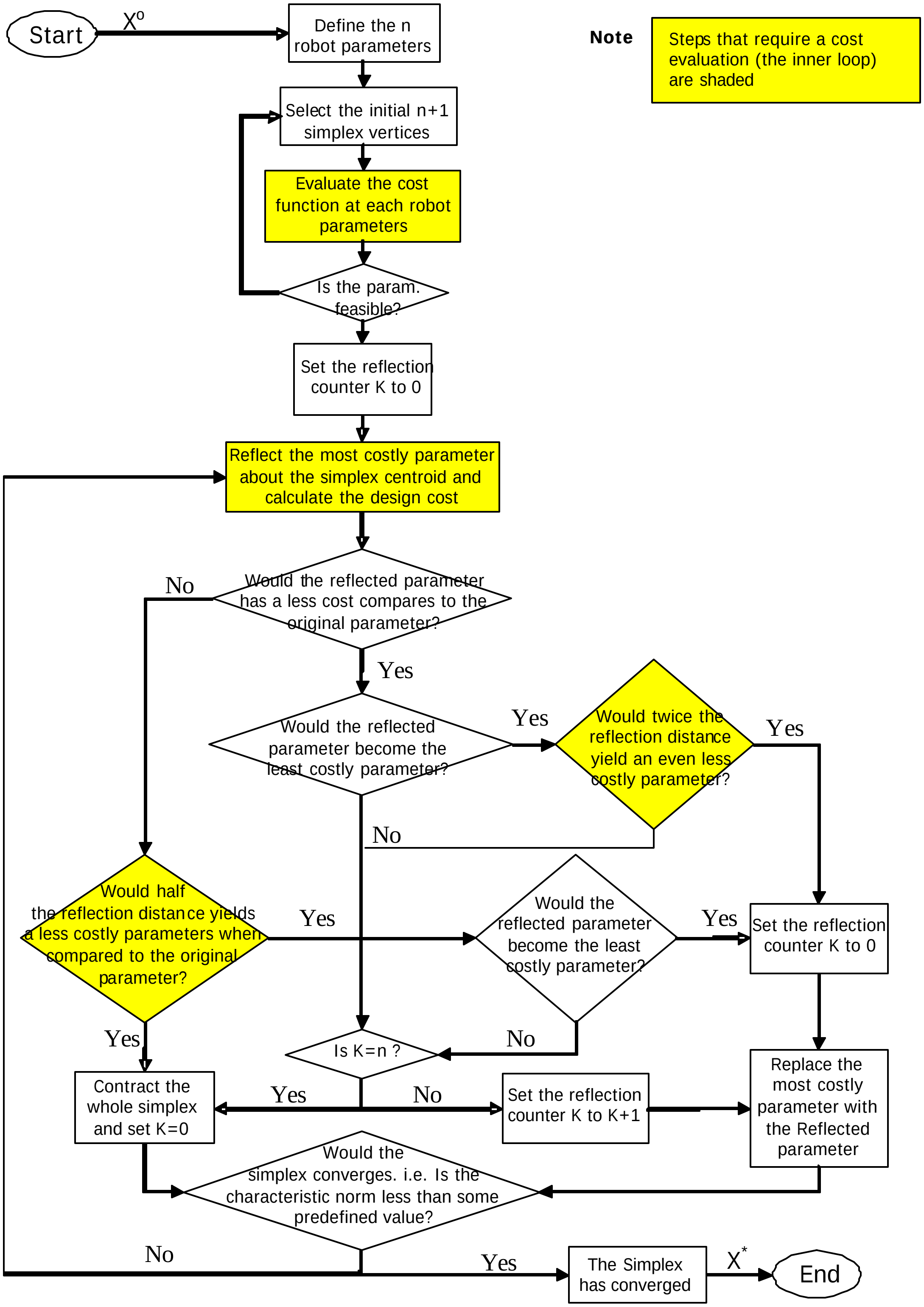
Start
X^o
Define the n robot parameters
Note
Steps that require a cost evaluation (the inner loop) are shaded
Select the initial n+1 simplex vertices
Evaluate the cost function at each robot parameters
Is the param. feasible?
Set the reflection counter K to 0
Reflect the most costly parameter about the simplex centroid and calculate the design cost
No
Would the reflected parameter has a less cost compares to the original parameter?
Yes
Would the reflected parameter become the least costly parameter?
Yes
Would twice the reflection distance yield an even less costly parameter?
Yes
No
Would half the reflection distance yields a less costly parameters when compared to the original parameter?
Yes
Would the reflected parameter become the least costly parameter?
Yes
Set the reflection counter K to 0
Yes
Is K=n ?
No
Replace the most costly parameter with the Reflected parameter
Contract the whole simplex and set K=0
Yes
No
Set the reflection counter K to K+1
Would the simplex converges. i.e. Is the characteristic norm less than some predefined value?
No
Yes
The Simplex has converged
X^*
End

Figure 4-2: Outer Optimization of Geometric Parameters

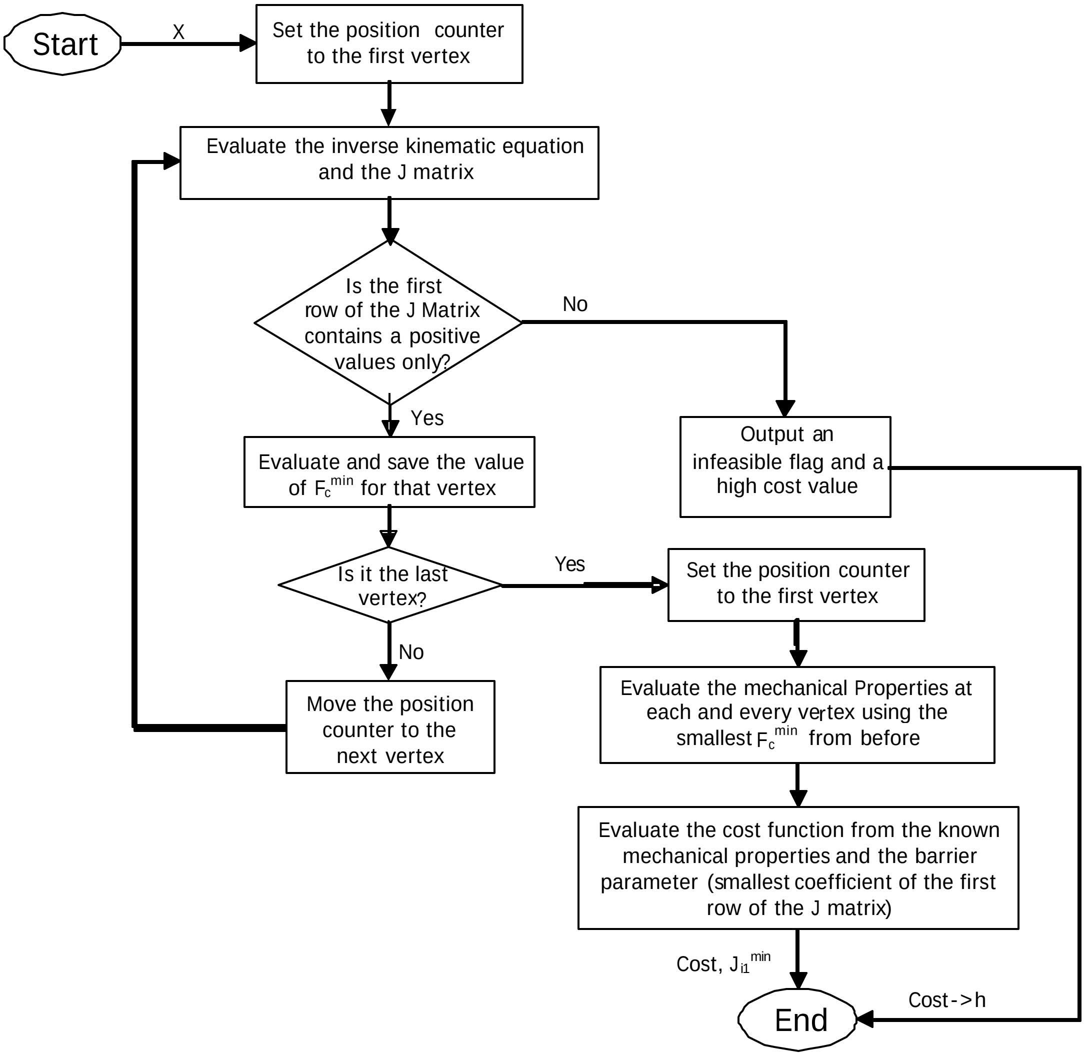
Start
X
Set the position counter to the first vertex
Evaluate the inverse kinematic equation and the J matrix
Is the first row of the J Matrix contains a positive values only?
No
Yes
Output an infeasible flag and a high cost value
Evaluate and save the value of F_c^{min} for that vertex
Is it the last vertex?
Yes
Set the position counter to the first vertex
No
Move the position counter to the next vertex
Evaluate the mechanical Properties at each and every vertex using the smallest F_c^{min} from before
Evaluate the cost function from the known mechanical properties and the barrier parameter (smallest coefficient of the first row of the J matrix)
Cost, J_{i1}^{min}
Cost->h
End

Figure 4-3: Inner Optimization of the Mechanical Parameters

Chapter 5

Mechanical Design of Cable-Based Planar Robot

Even with a good conceptual design and a sound implementation strategy, this work would not be completed without showing some typical optimal design results. The purpose of this chapter is twofold, firstly it gives comprehensive examples of the proposed parallel cable-robot implementation; secondly, it provides the design parameter (dimension) of the optimal robot so that it can be built and evaluated experimentally. Section 5.1 and 5.2 explore two completed implementations of the proposed cable-based robot design. Both of these designs utilize a pneumatic cylinder as the spine and two rotary actuators to manipulate the end-effector. In each design, there is a comprehensive study on the robot kinematics, the design weighting, and the optimal robot performance documented in their respective sections. Due to some historical background of this project, the actual prototype does not have the optimal design parameters presented in this chapter.

This analysis assumes that the spine force is held constant across the entire workspace. The inverse kinematic equations for these designs are discussed for deriving the actuator torque and power equations. The workspace considered in this study is a 700mm by 100mm rectangle and the maximum velocity and acceleration of the end-effector are set to 4:0m=s and 4g respectively. Also, the operating cable tension is set to be no less than 1N. The performance indexes that are used in this design are the same as those listed in Section 4.1

There are some standard mechanical components, which are heavy enough to have a considerable amount of effect on the dynamic behaviour of the robot. In both designs, it is assumed that a 25:4mm diameter, 2mm wall thickness aluminium cylinder tube is used. The length of the cylinder tube is set to be the same as the distance measured from the top of the workspace to the cylinder upper pivot. The center of mass is assumed to be at its geometric center. The cylinder piston is to be constructed by two 6:35mm diameter steel rods, and they should be long enough to cover the necessary stroke. The end-effector is assumed to be a 40mm wide by 5mm thick aluminium plate, and its length is set to be equal to twice the horizontal cable separation distance, d. The horizontal and vertical distance measured from the pivot point O to the combined center of mass of the end-effector and 500g payload should be within §1cm and §5cm respectively.

One important tool that has been neglected in this work is a general strategy to come up with the inverse kinematic equations for this kind of parallel mechanism, which has been partially addressed in the recent literature. A general workspace analysis of a broad class of 2D parallel manipulators can be found in [14] and [24].

5.1 Design 1: 2D-Deltabot

The first design uses a pair of rotational linkages to actuate the end-effector. In this design, the cables do not change in length, and we will refer to them as passive cables [19],[20]. Figure 5-1 is a pictorial view of the proposed design. This configuration is kinematically similar to Delta, and this document will refer to it as 2D-Deltabot [20].

Section 3.3 discusses the negative impacts of the end-effector eccentric distances. In this design, these eccentric distances are eliminated by attaching the cables to the end-effector through revolute joints. The design parameters of this mechanism include the motor separation distance (L_0), the length of the first linkage (L_1), the length of the cable (L_2), the minimum operation height (Y^{min}), the top cylinder operating offset (dH_c), and the cable separation distance (d). In anticipation that the optimal design is symmetrical about its center axis, the linkage on both actuators are assumed to be the same size. From the kinematics point of view, the synchronized linkage can be replaced with an equivalent linkage midway between

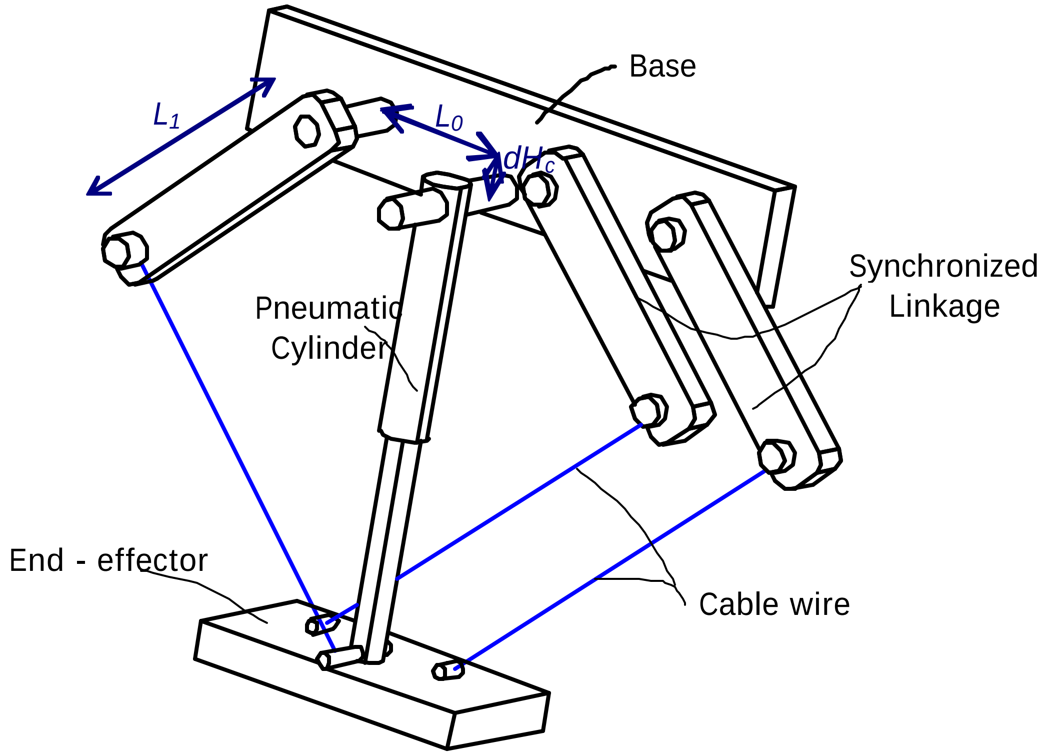

Figure 5-1: Pictorial View of 2D-Deltabot

the paired linkages. The reachable workspace of this design is shown in Figure 5-2. Note that 2D-Deltabot has a convex workspace boundary at the bottom and a concave boundary at the top [12]. A concave work space boundary is disadvantageous as it is inefficient to allocate a rectangular workspace inside the concave space.

The kinematic model for 2D-Deltabot is similar to the Rice Planar Delta Robot [14] [8]. Derived from geometry, the inverse kinematic equation for the kinematic chain of side A (the parallel cable joint side) is:

$$\theta_A(X;Y) = \arccos\left(\frac{L_1^2 + Y^2 + (L_0 - X)^2}{2L_1\sqrt{Y^2 + (L_0 - X)^2}}\right) + \arctan\left(\frac{Y}{L_0 - X}\right) - \pi \tag{5.1}$$

Due to the symmetry about the Y-axis, the second inverse kinematic equation, $\theta_B(X;Y)$ is $\theta_A(-X;Y)$.

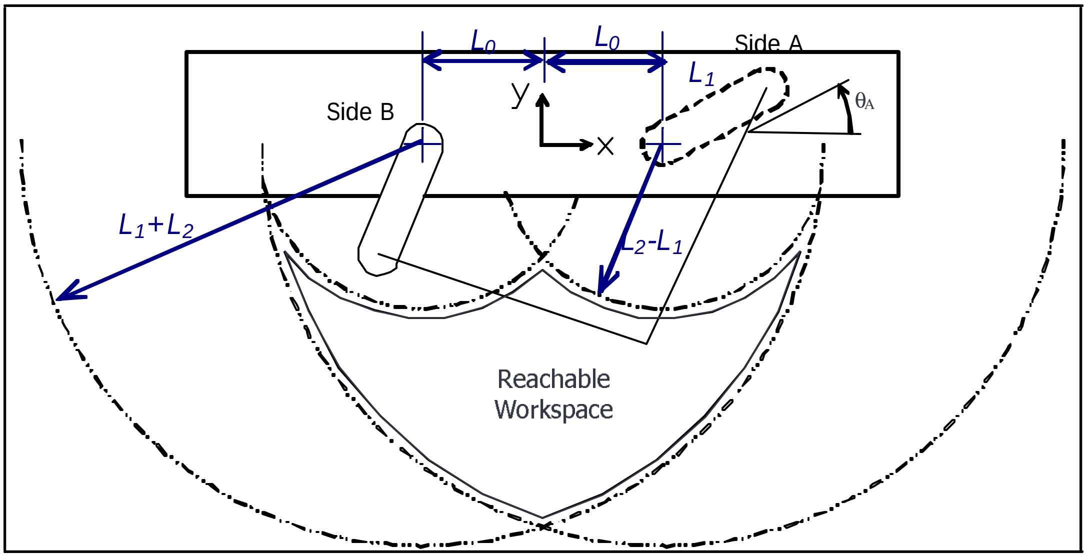

Figure 5-2: Workspace Analysis of 2D-Deltabot

Section 4.3 discussed a method to calculate the limiting bound of the mechanical power using the Gradient and the Hessian of the inverse kinematic equations. Let $A = \sqrt{X^2 + Y^2}$, and $Q_a = \frac{L_1^2 - L_2^2 - A^2}{\sqrt{4L_1A^2 - (L_1^2 - L_2^2 + A^2)^2}}$, the corresponding equations for side A of 2D-Deltabot are:

$$\nabla\theta_A = -\frac{1}{A^2}\left(Q_a\begin{Bmatrix} L_0 - X \\ -Y \end{Bmatrix} + \begin{Bmatrix} L_0 - X \\ Y \end{Bmatrix}\right)^T \tag{5.2}$$

$$\begin{aligned} H_{11} &= 2\frac{L_0 - X}{A^2}\left[\frac{\partial\theta_A}{\partial X} - \frac{Q_a(L_0 - X)}{L_1^2 - L_2^2 - A^2}\left(1 + Q_a^2\frac{L_1^2 + L_2^2 - A^2}{L_1^2 - L_2^2 - A^2}\right)\right] + \frac{Q_a}{A^2} \\ H_{12} &= -2\frac{Y}{A^2}\left[\frac{\partial\theta_A}{\partial X} - \frac{Q_a(L_0 - X)}{L_1^2 - L_2^2 - A^2}\left(1 + Q_a^2\frac{L_1^2 + L_2^2 - A^2}{L_1^2 - L_2^2 - A^2}\right) - \frac{1}{2Y}\right] \\ H_{22} &= -2\frac{Y}{A^2}\left[\frac{\partial\theta_A}{\partial Y} - \frac{Q_aY}{L_1^2 - L_2^2 - A^2}\left(1 + Q_a^2\frac{L_1^2 + L_2^2 - A^2}{L_1^2 - L_2^2 - A^2}\right)\right] + \frac{Q_a}{A^2} \end{aligned} \tag{5.3}$$

Due to symmetry, the corresponding Gradient and Hessian equations for side B can be derived

Table 5.1: Weighting on various perfomance indexes of 2D-Deltabot

	Geometric		Mechanical Properties			
	w	h	F_c^{min}	T^{max}	W^{max}	E^{max}
unit	/mm	/mm	/N	/N	/W	r/mm
Abs	0.5	0.1	5	1	1	50
Mean	N/A	N/A	N/A	10	10	20

the corresponding equations from side A as:

$$\nabla\theta_B(X,Y) = \begin{bmatrix} -\frac{\partial\theta_A}{\partial X}(-X,Y) & -\frac{\partial\theta_A}{\partial X}(-X,Y) \end{bmatrix} \tag{5.4}$$

$$H_B(X,Y) = \begin{bmatrix} \frac{\partial^2\theta_A}{\partial X^2}(-X,Y) & -\frac{\partial^2\theta_A}{\partial X\partial Y}(-X,Y) \\ -\frac{\partial^2\theta_A}{\partial X\partial Y}(-X,Y) & \frac{\partial^2\theta_A}{\partial Y^2}(-X,Y) \end{bmatrix} \tag{5.5}$$

5.1.1 Design Weighting and Optimization Results for 2D-Deltabot

The weighting for the design merit in the initial study is tabulated in Table 5.1. Typical mid-size servo DC brushless motor delivers 3Nm to 8Nm of torque. This is not enough to drive this robot setup. Therefore, a gear-reducer is a necessary component in this design. As a result, the maximum actuator torque is not critical. The best 60 (distinct) set of robot parameters from the initial optimization process were reprocessed with a finer inner mesh resolution. Table 5.2 shows the top 6 distinct set of robot parameters based on the established cost function. Since most of these optimal results are very close to one to another, it is reasonable to assume that the global optimal result has been reached. The optimal 2D-Betabot design requires 127N of central cylinder force. Using a 25.4mm pneumatic cylinder as the spine element, the robot operates roughly on 2.5bar of compressed air. Furthermore, the upper bound of the cable tension is 127N. A Ø2mm steel cable wire can easily withstand this level of force [1]. As for spatial performance, the footprint of the robot is 983mm for a 700mm wide workspace. This is more than 75% spatial efficiency.

Although the design procedure is targeted to minimize the absolute as well as the average cable tension and actuator power, a study in the critical tension distribution over the workspace

Table 5.2: The top 10 configurations and the associated performance indexes for 2D-Deltabot

	Configuration						Performance					
	L_0	L_1	L_2	dH_c	Y^{min}	d	w	h	F_c^{min}	T^{max}	W^{max}	E^{max}
	mm	mm	mm	mm	mm	mm	mm	mm	N	N	W	mm=r
1	202	239	688	29	-574	101	983	574	127	127	648	93
2	195	238	683	28	-571	98	964	571	119	128	648	95
3	177	235	666	23	-599	89	913	559	124	130	651	99
4	190	235	672	25	-560	95	945	560	121	289	653	94
5	162	233	659	19	-557	81	871	557	128	132	652	105
6	196	236	674	27	-559	98	962	559	119	128	657	92

provides some insight into the design. Figure 5-3 and 5-4 shows the contour plot of the maximum tension on the cable and the maximum actuator power of side B. Due to the symmetry about the Y-axis, the cable tension on side B is the same as the equivalent cable tension on the parallel cable joint when the end-effector is on the opposite side of the workspace. Since there is only one cable on side B (the single cable size), the maximum cable tension will always be carried by Cable B; hence it carries the critical tension. A simulation of the optimal actuator power shows that the worst case actuator power occurs at the top of the workspace near to the actuator and the opposite lower corner of the workspace. The actuator power is profoundly dependent on cable tension; as a result, the maximum cable tension occurs near to the region where maximum actuator power occurs.

5.2 Design 2: 2D-Betabot

The second design utilizes a winch to actuate the end-effector. In this design, the cable lengths are controlled by the winch, and we will refer to them as active cables [19],[20]. Figure 5-5 is a pictorial view of the proposed design. This document will refer this design as 2D-Betabot [20].

Moreover, there will be a round bending surface at the cable attachment points. If there are two different cable bending radii on the upper and lower attachment points, there are 6 design parameters in this design (see Figure 5-7): the motor separation distance (L_0), the upper bending radius (R_u), the lower bending radius (R_l), the minimum operation height (Y^{min}), the top cylinder operating offset (dH_c), and the cable separation distance (d). The bending surface

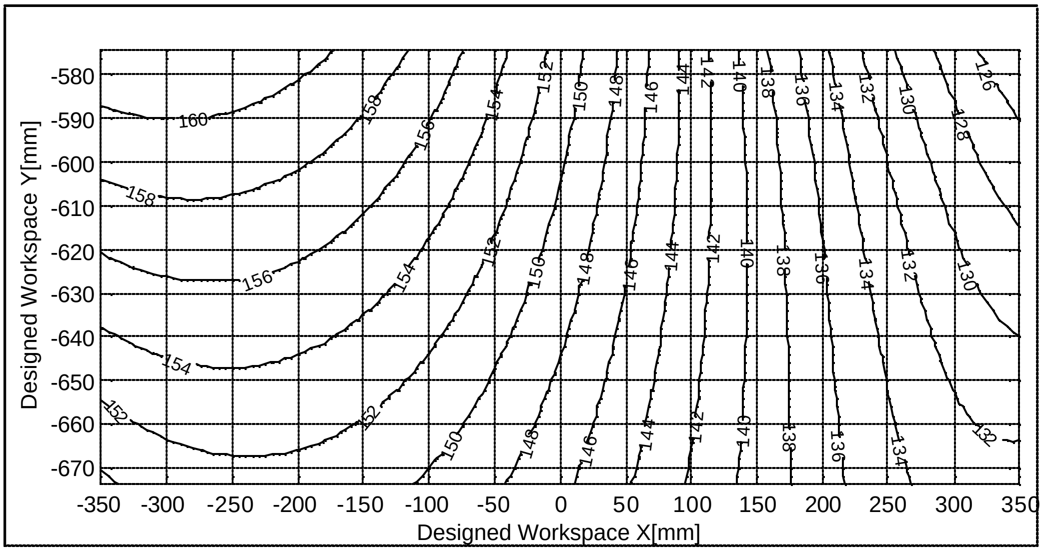

Figure 5-3: Maximum cable tension (occurs on the Side B) [N] for the optimal 2D-Deltabot configuration.

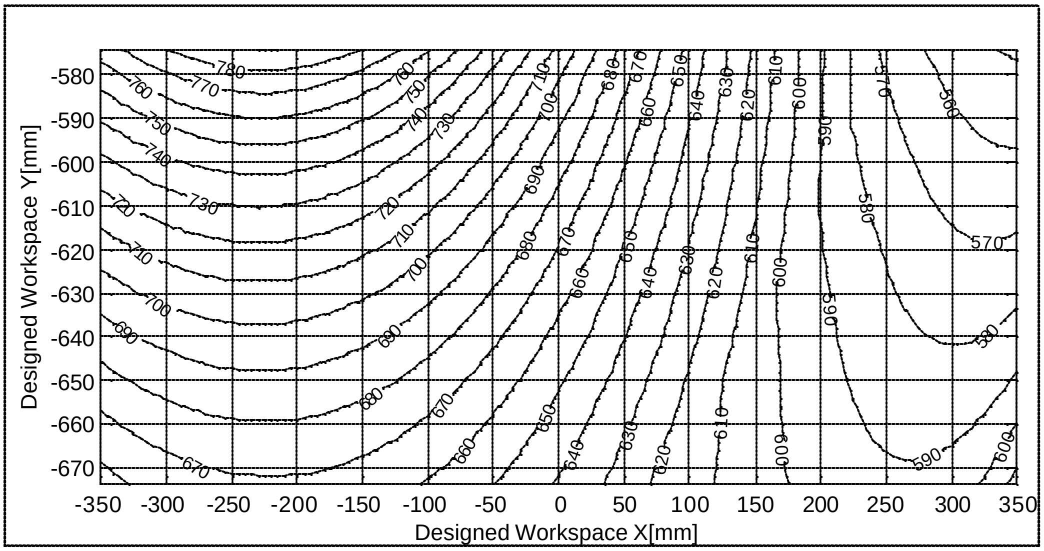

Figure 5-4: Maximum actuator power [W] of side B for the optimal 2D-Deltabot configuration.

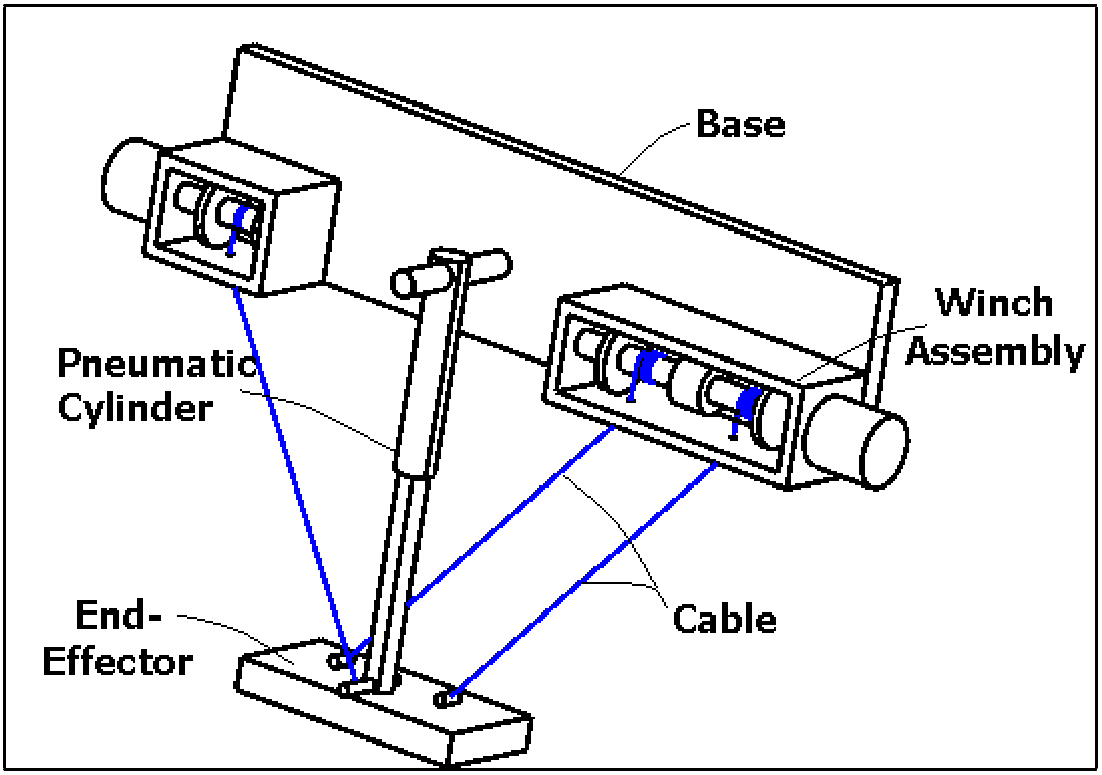

Figure 5-5: Pictorial View of 2D-Betabot. Note the bending radius is not shown for simplicity.

can be constructed from a cable pulley. This feature eliminates the need for revolute joints and increases the reliability of the mechanism. The radius of curvature for the bend should be at least 45 times bigger than the radius of the steel cable to avoid bending fatigue [1]. However, this bending radius introduces eccentric distances at the end-effector. From Section 3.3, this feature is going to compromise the cable joint torque capacity. As a result, there is a competing factor between a thicker cable and a smaller bending radius. The problem is that the cable thickness is determined by maximum cable tension, which in turn is affected by the kinematics of the robot. Instead of solving this implicit constraint analytically, one alternative approach is to force a logarithmic barrier of the minimum bending radius constraint on the cost function. This approach is similar to the penalty method implemented in the optimization process, which prevents the search sequence from stepping over a constraint boundary without adding a significant impact on the cost function.

Since the kinematics of this robot is new to the best of the author's knowledge, a detailed

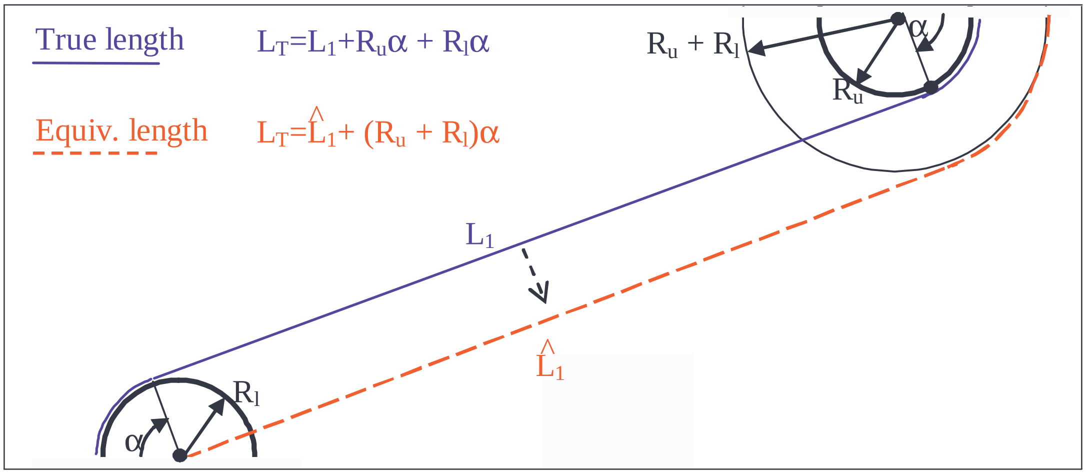

Figure 5-6: Equivalent cable length can be obtained by translating the bending radius.

derivation of the kinematic equations is needed in this work. Analogous to the workspace analysis of 2D-Deltabot, the parallel cable joint is replaced by an equivalent linkage in this analysis. The two bending radii can be simplified by perpendicularly translating the cable by R_l (from L_1 to $\hat{L}_1$). This equivalent cable wraps around an equivalent pulley with radius R_T. Hence, it has the same overall length as the original cable. Due to the geometric constraints, this equivalent cable always passes through the center of the lower bending radius, see Figure 5-6 for illustration.

Figure 5-7 shows the reachable workspace of the robot. Unlike 2D-Deltabot, 2D-Betabot has a convex workspace. There are three limiting boundaries on the reachable workspace. The top boundary is established by a horizontal line that is R_T below the center of the upper bending radius; similarly, the left and right boundaries are established by the two vertical lines that are R_T away from the upper bending radii. The robot violates condition C1 in Equation (3:10) if the end-effector is on or above the top boundary and the end-effector disengages from the pulleys if it is not between the left and right boundaries. The lower bound of the workspace is limited by the amount of cable in the winch or the stroke length of the pneumatic cylinder only.

The total cable length is determined from two sections, the straight cable $L_{x1}$and the wrap-around cable L_{x2}, x µ fA; Bg. On side A, consider the triangle formed by the joint center,

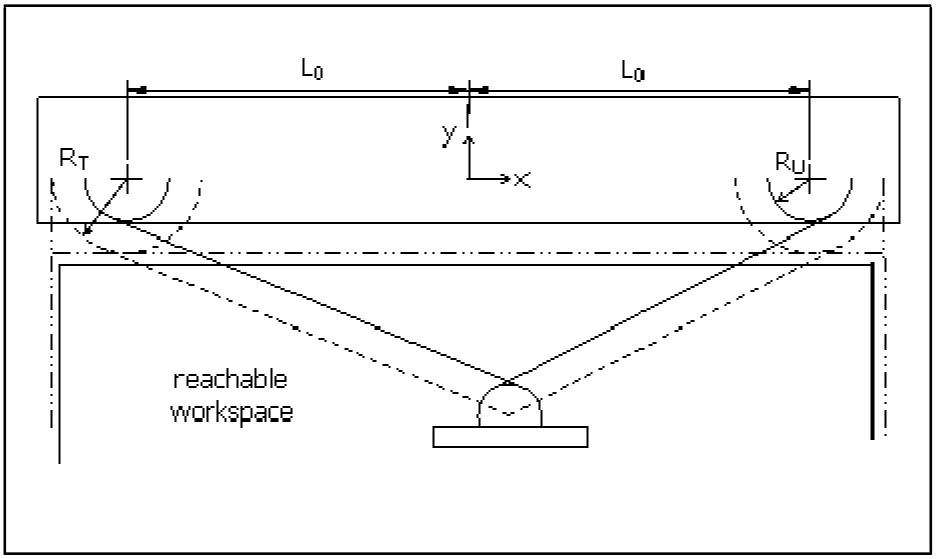

Figure 5-7: Workspace Analysis of 2D-Betabot

the cable take off point, and the center of the end-effector, the straight section of the cable can then be derived from the distance equation as:

$$L_{A1} = \sqrt{(L_0 - X)^2 + Y^2 - R_T^2} \tag{5.6}$$

The calculation of the wrap-around section requires computing the angle that the cable is in contact with the pulley. Using the same triangle, the wrap-around cable length on side A is:

$$L_{A2} = R_T \left[\pi - \arctan\left(-\frac{Y}{L_0 - X} \right) - \arctan\left(\frac{L_{A1}}{R_T} \right) \right] \tag{5.7}$$

Due to the symmetry about the Y-axis, the equivalent cable length of side B, L_B is $L_A(-X; Y)$: The Gradient and the Hessian matrix of the cable length can be computed from the sum of Equations (5:6) and (5:7). Let A be the distance between the center of a pulley pair, the Gradient and the Hessian of 2D-Betabot can be computed by the following equations:

$$A = \sqrt{(L_0 - X)^2 + Y^2}$$

$$\nabla L_A = \frac{1}{A}\left(\partial L_{A1}\begin{Bmatrix} X - L_0 \\ Y \end{Bmatrix} + R_T\begin{Bmatrix} Y \\ L_0 - X \end{Bmatrix}\right)^T \tag{5.8}$$

$$H_A = \begin{bmatrix} \frac{L_0 - X}{A^2}\left(2\frac{\partial L_A}{\partial X} + \frac{L_0 - X}{L_{A1}} + \frac{L_{A1}}{L_0 - X}\right) & -\frac{Y}{A^2}\left(2\frac{\partial L_A}{\partial X} + \frac{L_0 - X}{L_{A1}} - \frac{R_T}{Y}\right) \\ -\frac{Y}{A^2}\left(2\frac{\partial L_A}{\partial X} + \frac{L_0 - X}{L_{A1}} - \frac{R_T}{Y}\right) & \frac{Y}{A^2}\left(-2\frac{\partial L_A}{\partial Y} + \frac{Y}{L_{A1}} + \frac{L_{A1}}{L_0 - X}\right) \end{bmatrix} \tag{5.9}$$

In this design, pivot point O is misaligned with the cable axes as shown in Figure 3-8. Assuming the pivot point coincides with the center of the lower pulley of side B, the horizontal cable eccentric distance, δx can be computed by extending the equivalent cable A to the horizontal line that passes through the pulley center axis. Using the sign convention from Section 3.2.1, this eccentric distance is always positive. Similarly, the vertical cable eccentric distance, δy is the intersection of Cable B and the vertical line that passes though the pulley center axis. This distance is also positive in sign. The numerical calculation of these eccentric distances are:

$$\delta x = \frac{R_l}{\sin(\beta_A)} \tag{5.10}$$

$$\delta y = -\frac{R_l}{\cos(\beta_B)} \tag{5.11}$$

Since the effect of eccentric distances shows up as several residual terms on J matrix, substituting Equations (5:10) and (5:11) to Equation (3:15) shows that the magnitude of the cable axis angles, β has a significant contribution to the torque capacity. Therefore, it is expected that the operating workspace should be further below the upper pulley's level to maximize the torque capacity. Moreover, it is possible to reduce this side effect by placing the pivot point closer to the cable take off points of the lower pulleys. However, this approach will introduce two additional design parameters in the optimization process and increase the search magnitude by two full orders. As an initial study, it was decided that no eccentric distance compensation should be made on the end-effector design.

5.2.1 Design Weighting and Optimization Result for 2D-Betabot

Table 5.3 tabulates the design weighting for 2D-Beta design parameters and Table 5.4 shows the top 6 (distinct) set of robot parameter based on the established cost function. The optimal 2D-Betabot design requires 157N of cylinder force. Assuming a 25:4mm pneumatic cylinder is to be used, the mechanism operates on approximately 45psi of compressed air. While the 2D-Betabot design has compatible mechanical indicators such as maximum cable tension and actuator power when compared to the 2D-Deltabot counterpart, the 2D-Betabot performs poorly from the spatial performance point of view. The optimization results show that the 2D-Betabot design requires up to 50% more footprint and a 300% wider end-effector when compared to the optimal 2D-Deltabot design. Furthermore, the pulley radius is concentrated at the lower end of the allowable limits. It is clear that the wide foot print is primarily due to a significantly larger end-effector, which in turn is caused by a lower torque capacity.

Table 5.3: Weighting on various perfomance indexes of 2D-Betabot

	Geometric		Mechanical Properties			
	w	h	F_c^{min}	T^{max}	W^{max}	E^{max}
unit	/mm	/mm	/N	/N	/W	mm=mm
Abs	0.5	0.1	5	1	1	500
Mean	N/A	N/A	N/A	10	10	200

Table 5.4: The top 10 configurations and the associated performance indexes for 2D-Betabot

	Configuration						Performance					
	L_0	R_U	R_L	dH_c	Y^{min}	d	w	h	F_c^{min}	T^{max}	W^{max}	E^{max}
	mm	mm	mm	mm	mm	mm	mm	mm	N	N	W	mm=mm
1	610	34	33	302	-669	285	1593	669	157	177	890	1.27
2	630	33	34	320	-680	270	1641	680	159	178	894	1.24
3	584	34	34	279	-654	274	1529	654	161	180	900	1.30
4	571	34	34	266	-644	305	1495	644	149	180	904	1.32
5	559	34	33	257	-638	315	1456	638	149	180	905	1.35
6	549	34	34	242	-627	292	1440	627	155	182	912	1.35

Note that the bending radius always converges to the lower end of the tolerance limit. This indicates that the bending radius should be kept at the minimum level. A contour plot

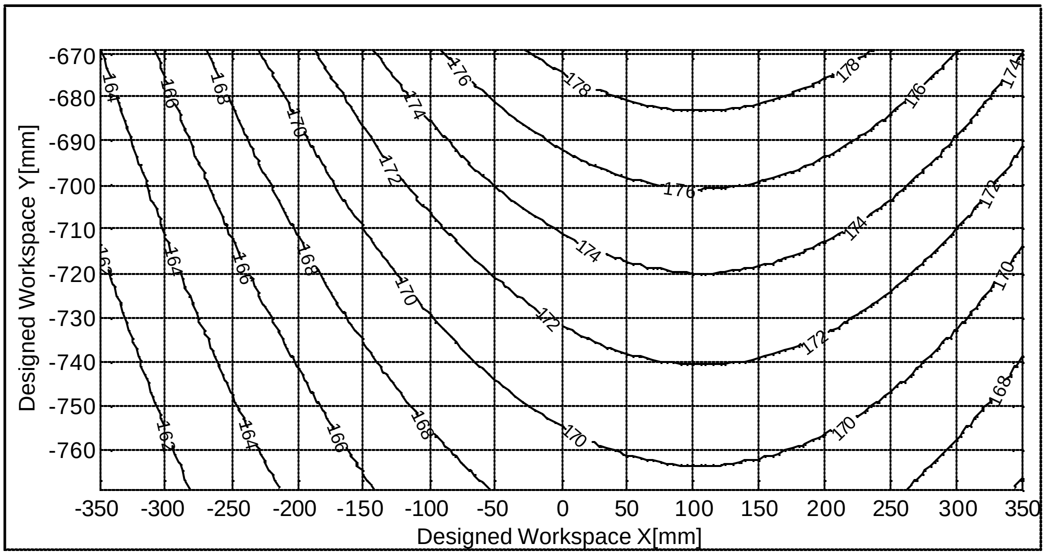

Figure 5-8: Maximum cable tension [N] (occurs on side B) for the optimal 2D-Betabot configuration.

for the maximum cable tension and actuator power for the optimal 2D-Betabot design are shown in Figure 5-8 and Figure 5-9 respectively. Unlike the optimal 2D-Deltabot design, the maximum cable tension in 2D-Betabot occurs in one location only. More importantly, there is no stationary point ($\nabla T = 0$) inside the workspace. As a result, the maximum cable tension could have been identified easily by conventional optimization algorithms, and the result would still be the global maximum. This would be a major improvement on the computational efficiency for the design optimization.

5.3 Prototype: 2D-Deltabot

5.3.1 Prototype Design and Specifications

Due to the past success in the development of the Deltabot, a cable-based version of the Delta, the 2D-Deltabot was implemented for prototyping. While the design specification is reasonable for most industrial applications, the production of the prototype was limited by the existing equipment, which includes three 0:47kW Kollmorgen servo motor and three Thompson Industries 12:1 right angle gear-reducer. The rated torque for this drive system is 36N-m

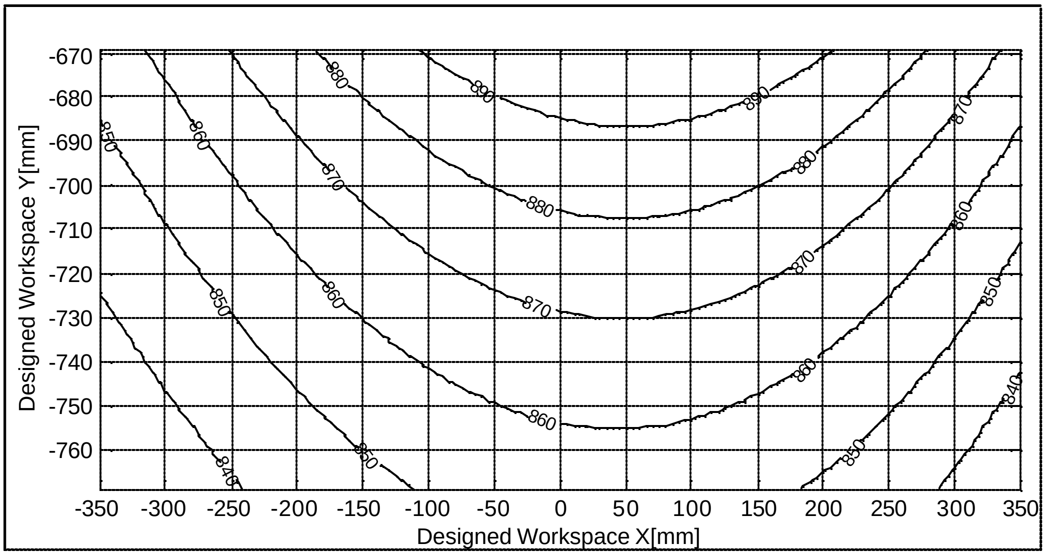

Figure 5-9: Maximum actuator power [W] of side B for the optimal 2D-Betabot configuration.

[7]. With the limited actuator power, none of the optimal designs could satisfy the power requirements. As a result, the maximum end-effector speed was lowered to $2.0 m/s$ and the optimal set of the design parameters is listed below:

$L_0 = 150mm$

$L_1 = 240mm$

$L_2 = 525mm$

$dH_c = 0mm$

$Y^{min} = 392mm$

$dPl = 75mm$

With a reduction in the required maximum end-effector speed, the maximum required actuator power is reduced to $0.35kW$ and the maximum cable tension is reduced to $149N$ as shown in Figure 5-10 and Figure 5-11. Since the maximum end-effector velocity affects both the dynamic cable force and the actuator speed, this is reasonable to have more than 50% reduction in actuator power when the required end-effector velocity is halved.

Another limiting factor in the prototype design is the limited amount of the available actuator torque. The prototype requires just under $36N\,m$ to operate at the rated specification.

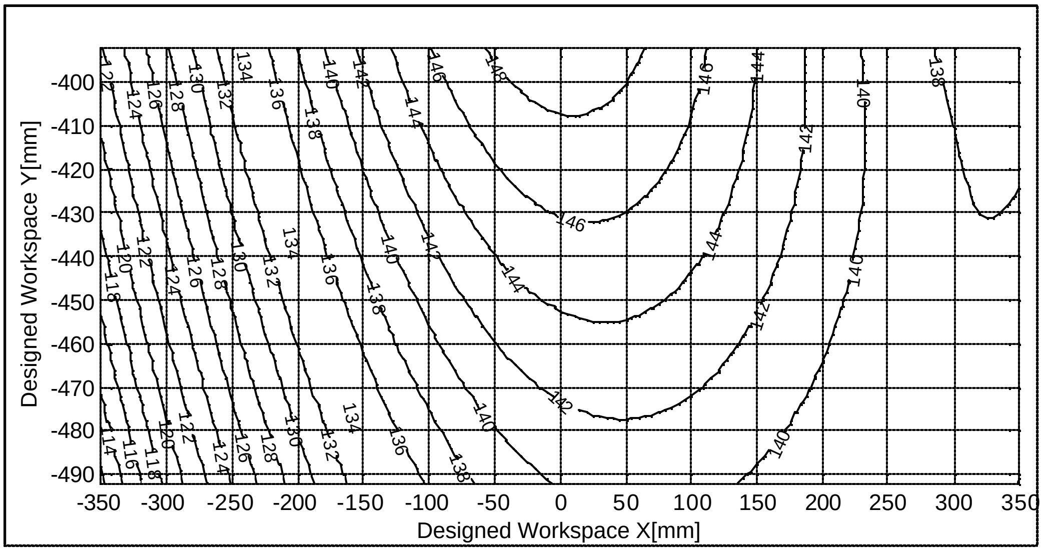

Figure 5-10: Maximum cable tension (occurs on the Side B) [N] for the Prototype 2D-Deltabot
.

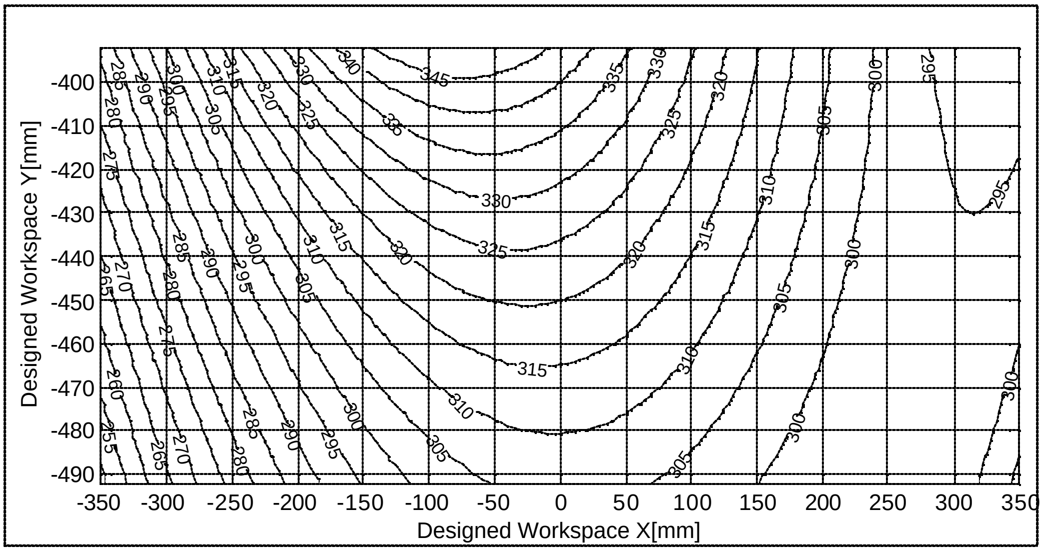

Figure 5-11: Maximum actuator power [W] of side B for the prototype 2D-Deltabot

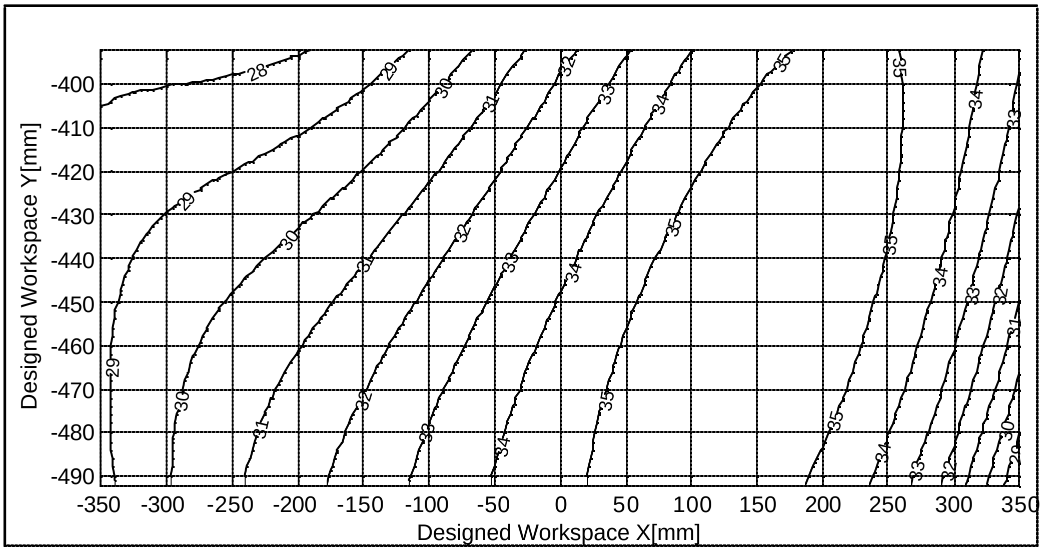

Figure 5-12: Maximum actuator torque [W] of side B for the prototype 2D-Deltabot

Figure 5-12 shows the maximum actuator torque distribution of the prototype. A comparison between the maximum actuator torque (Figure 5-12) and the maximum cable tension (Figure 5-10) shows that the actuator torque is more sensitive to the cable angle than to the cable tension. More importantly, the maximum actuator torque occurs in a different location from the maximum torque. This indicates that there will be different limiting resources at different locations of the workspace. Nonetheless, it is worthwhile to investigate the region where the manipulator can provide performance that is better than the design specifications.

5.3.2 Prototype Theoretical Performance Analysis

Since the cylinder force is held constant throughout the workspace to ensure minimum tension at the rated performance, there are regions in the workspace which may lose rigidity when it is driven beyond its designed specifications. A study on the minimum cable tension distribution would be necessary to determine the critical region where the manipulator loses it rigidity. Figure 5-13 is a contour plot of the minimum cable tension in the robot workspace. Note that the minimum cable tension occurs in side A and its trend is virtually a mirror image from the maximum cable tension contour in Figure 5-10.

Since the minimum tension occurs in the upper left corner (when the effective cable sepa-

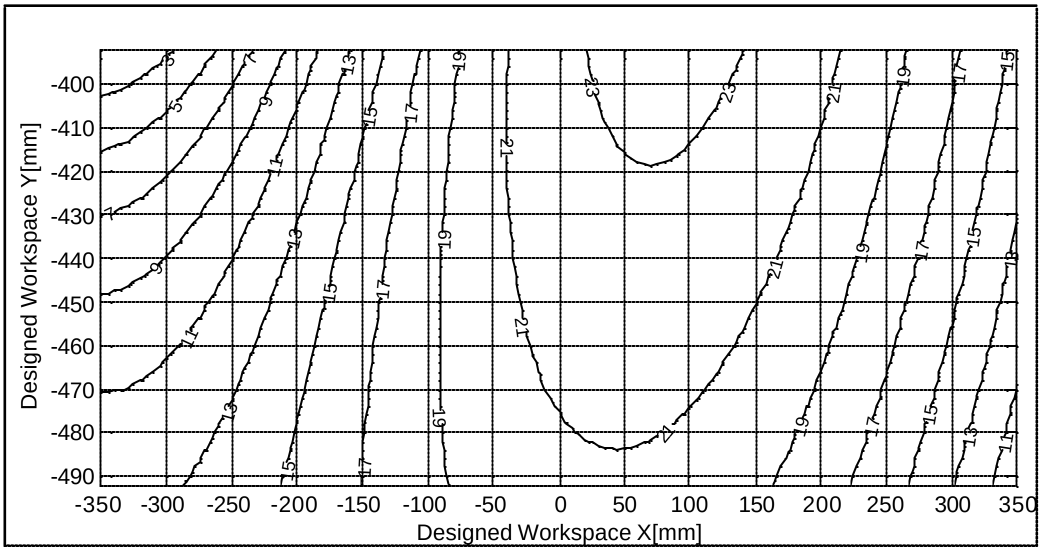

Figure 5-13: Minimum cable tension [T] of side A for the prototype 2D-Deltabot

rating distance, $d \sin(¯_A)$ is the smallest), it is expected that the manipulator cannot perform any better than the specifications at that location. Figure 5-14 shows the maximum velocity of the end-effector under the worst case condition in the robot workspace. As expected from previous observations, the limiting speed of the prototype robot is significantly lower in the upper left (opposite to the parallel cable joint) corner of the workspace, where the cable tension reaches minimum. Also, the speed is also limited towards the center of the workspace where the maximum actuator torque occurs.

In the most optimal pick-and-place trajectory, the end-effector should reach its maximum speed at the center of the path. Therefore, it is reasonable to expect that the limiting speed is the maximum allowable speed near the upper center of the workspace. In the prototype design, it is approximately 2:5m=s. At this limiting speed, the robot can complete a 1:8m pick-and-place cycle (up 0:1m, across 0:7m, down 0:1m and goes back to the starting position) in 0:84s. This translates to approximately 71 cycles / minute.

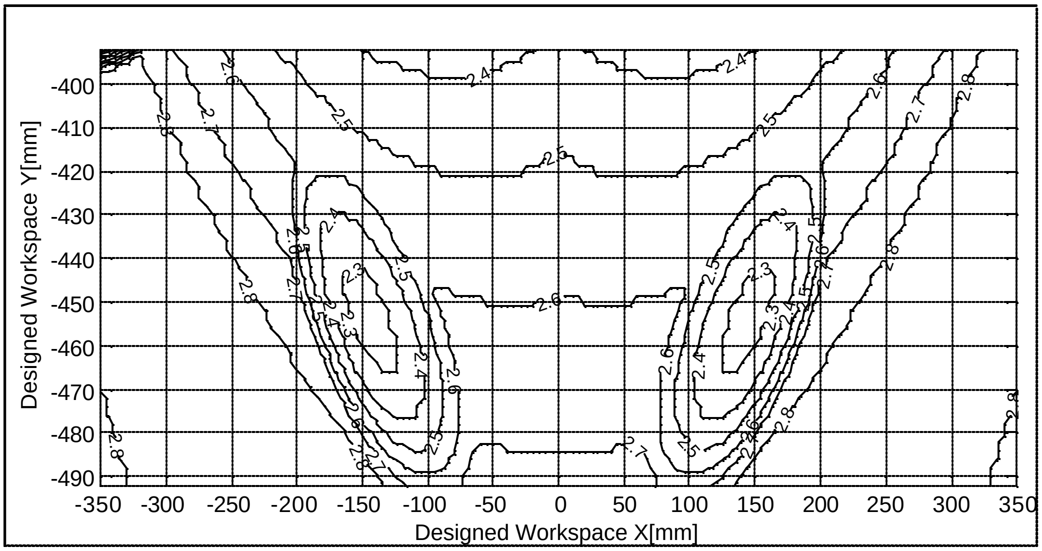

Figure 5-14: Maximum Velocity [m/s] for the prototype 2D-Deltabot

Chapter 6

Robot Experimental Evaluations

There were numerous tests and calibrations performed on the prototype 2D-Deltabot. Most of these tests were designed to measure the robots operating performance, others were designed to address the reliability of individual mechanical components. This chapter is focused on the experiments that were aimed to evaluate the kinematic performance of 2D-Deltabot. From the kinematic point of view, the two important metrics to gauge the robot quality were the accuracy and the repeatability. While repeatability is almost a built-in trait of a robot that cannot be altered substantially, the robot accuracy can be significantly improved by means of careful calibration. Section 6.1 of this chapter introduces a computationally efficient algorithm to calibrate 2D-Deltabot. Section 6.2 then presents the kinematic accuracy experiment that evaluates the prototype robot accuracy after the kinematic calibration. Section 6.3 presents the repeatability experiment that evaluates the repeatability of the robot in a typical high speed pick-and-place motion.

6.1 Kinematic Calibration

Kinematic calibration of a parallel mechanisms has been addressed in recent literature [11], [25]. The fundamental objectives of existing techniques are to identify the design parameters including all possible deviations. These existing methods apply optimization techniques to find the optimal set of design parameters that minimizes the position error in world coordinates [25]. While this generalized approach works well theoretically in a general robot configuration, it could be computationally intensive to solve the kinematic chains of a parallel manipulator that has 2 or 3 degrees of freedom. Consider the Delta configuration as an example, it has 3DOF and requires 24 design parameters after several assumptions. The number of measurements involved in a calibration process increases exponentially with the number of DOF in the manipulator. If there are 10 measurements taken for each dimension of Delta, a complete calibration process could require 1000 measurements. This is equivalent to solving a set of 24 parameter equations 1000 times per each cost function evaluation. Therefore, it is of a great interest to utilize a computationally efficient method to solve this optimization problem. This work presents a relatively straight forward and versatile algorithm to calibrate a wide class of parallel mechanisms. The calibration algorithm is based on the least square error techniques, and it requires the first order information from the full robot kinematic model. Subsection 6.1.1 presents the formulation of the proposed calibration algorithm and some implementation considerations including the measurement errors and numerical singularity issues. A numerical simulation was performed to evaluate the robustness of the algorithm in Subsection 6.1.2.

6.1.1 Formulation of the Calibration Algorithm

The fundamental principle of kinematic calibration is to reduce the errors between the measured end-effector position and the modeled position. This falls into an optimization problem. Modern optimization algorithms show superior convergent properties if the objective function is at least twice differentiable. Most of the efficient algorithms require knowledges of the first derivative information to increase the algorithm convergent rates. Due to the nature of parallel mechanisms, a complete analysis (including the 1st and 2nd order information) of the forward kinematic model is virtually impractical. Therefore, it is advantageous to develop a new strategy to solve the optimization problem. The proposed calibration method is based on the Least Square Error (LSE) algorithm with two distinct characteristics. Instead of minimizing the position error of the manipulator, the algorithm minimizes the error on one of the robot geometric parameters (i.e. a dimension of the robot). Moreover, the problem can be formulated to obtain a closed form solution. This allows linearization on the error function at each measurement point to solve the overall error model analytically. However, the calculation must be repeated multiple times to account for this approximation.

Linearization and Least Square Error Method

The difficulty of a conventional parallel manipulator calibration routine arises when one wants to compute the Jacobian matrix of the forward kinematic equations. In a calibration routine, all manipulator parameters must be taken into consideration. Depending on the manipulator configuration and assumptions made in the analysis, the size of the resulting Jacobian matrix may range from 20 entries to a few 100 entries [25]. Furthermore, parallel robots usually have several unsensed joints in its kinematic chains. The states in these unsensed joints can only be solved as a system of implicit equation. This complicates the forward kinematic equations, which in turn increases the level of complexity in obtaining the first order information for optimization purpose. In the proposed algorithm, this problem is avoided by solving the distance equation that has the unsensed joints. Consider 2D-Deltabot as an example, there is one pair of unsensed joints in each of the kinematic chain as shown in Figure 6-1. The position of both ends of the joint can be computed by conventional forward kinematic analysis techniques. These points will be represented by a set of vector $\overrightarrow{P_1}$ (Base Side) and $\overrightarrow{P_2}$ (End-effector side). The distance between them, L_2 is the length of the cable, and it can be calculated by the distance equation:

$$L_2^2 = \left(\overrightarrow{P_2} - \overrightarrow{P_1}\right)^T \left(\overrightarrow{P_2} - \overrightarrow{P_1}\right) \qquad (6.1)$$

The rest of the analysis requires estimating and measuring several robot parameters. For the remaining analysis, the notation convention outlined in Table 6.1 will be used to distinguish various interpretations to a dimension quantity (e.g. true length, estimated length, a non-deterministic measured length).

In a calibration process, the position of $\overrightarrow{P_1}$ relative to the robot origin can be computed using 7 parameters. These parameters are the robot parameters to be calibrated. Since $\overrightarrow{P_2}$ is rigidly attached to the end-effector, it can be measured directly by the external measurement system. Let $\overline{L_2}$ be the estimated cable length and $\tilde{L}_2$ be the distance between the cable joints as computed by the kinematic parameters and the measured end-effector position, the different

Table 6.1: The Notation Convention of Different Quantities

Modifier		Representation	eg
None	X	A True Deterministic Scalar Quantity or Function	L_2 represents the true cable length
Vector	$\vec{X}$	A True Deterministic Vector Quantity or Function	$\vec{P}_1$ represents the true position of the base side cable attachment point
Bar	$\overline{X}$	An Estimated Deterministic Quantitiy (scalar or vector)	$\overline{L}_2$ represents the estimated cable length in the kinematic model $\overline{P}_1$ represents the estimated position of the base side cable attachment point based on the kinematic model
tilde	$\tilde{X}$	A Non-Deterministic Quantitiy or a quantity that is estimated by a non-deterministic quantity (scalar or vector)	$\tilde{L}_2$ represents the cable length measured indirectly through the measurment system $\tilde{P}_2$ represents the position of the end -effector side cable attachment point with the measurment errors

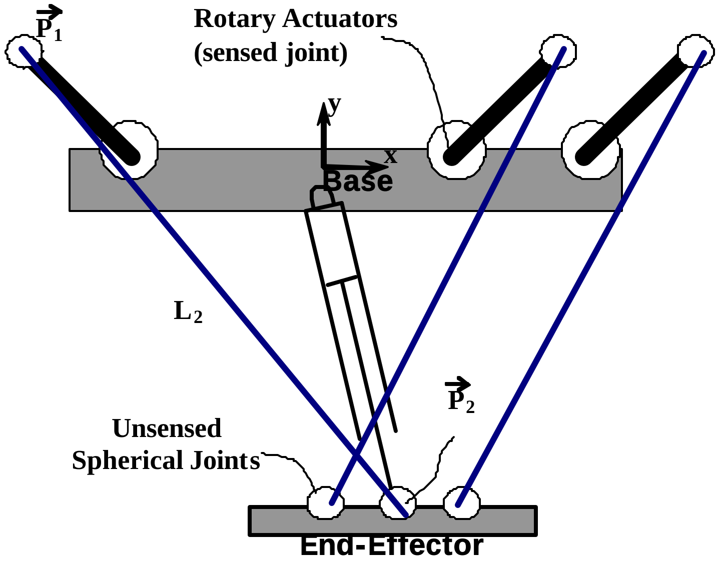

Figure 6-1: The Unsensed Joint and the Cable Points of 2D-Deltabot

between these values is the residual of the kinematic equations R:

$$R = \overline{L_2} - \hat{L}_2 \tag{6.2}$$

Given ρ is the list of the kinematic parameters, and $\hat{\chi}$ is the measured position. The value of R can be estimated by the first order approximation method:

$$R(\bar{\rho};\hat{\chi}) \approx R(\rho;\chi) + \frac{\partial R}{\partial \rho}(\bar{\rho} - \rho) + \frac{\partial R}{\partial \chi}(\hat{\chi} - \chi) \tag{6.3}$$

It is reasonable to assume that the bias in the measurement device is negligible; hence, the third term of Equation (6.3) is equal to zero. Furthermore, the residual of a perfectly calibrated robot (i.e. the residual are computed using the true parameter values) should also be zero. Therefore, Equation (6.3) can be simplified to:

$$\overline{R}(\bar{\rho}) = \nabla_{\rho} R(\bar{\rho})\,\delta\rho \tag{6.4}$$

where $\overline{R}$ is the function that estimates the kinematic residual, $\nabla_{\rho} R$ is the gradient function of R with respect to the parameter only, and the parameter errors $\delta\rho$ is the different between the the estimated parameter $\bar{\rho}$ and the true parameter ρ ($\delta\rho = p - \bar{\rho}$).

Equation (6.4) would be a close approximation if the estimated parameters are close to the true parameters. In practice, the manufacturing tolerance of the robot is usually tight enough for a fair approximation. If measurements were taken at multiple locations on the same robot setup, the data can be grouped together in matrix form and Equation (6.4) becomes:

$$\begin{aligned} \left\{ \begin{array}{c} \overline{R}_1 \\ \overline{R}_2 \\ \vdots \\ \overline{R}_n \end{array} \right\} &= \begin{bmatrix} \nabla_{\rho} R_1 \\ \nabla_{\rho} R_2 \\ \vdots \\ \nabla_{\rho} R_n \end{bmatrix} \delta\rho \\ R &= \nabla R\,\delta\rho \end{aligned} \tag{6.5}$$

note that the subscript ρ is dropped in the residual gradient matrix for simplicity.

One can use the LSE technique to find a set of robot parameters that minimizes the residual.

This is the essence of the calibration process. The closed form solution for ±½ is:

$$\pm½ = \mathbf{i}\,5R^{T}\,5R^{\mathbf{¢}\,i\,1}\,5R^{T}R \tag{6.6}$$

Once the parameter errors, ±p are computed, the estimated robot parameter values, þ can be updated for a better approximation. This process is to be repeated until the norm of the residual converges to a satisfactory value.

Full Robot Kinematic Analysis

At this point, it is appropriate to discuss the full kinematic model of 2D-Deltabot. This will be used for the computation of R and 5R. The full forward kinematic model of any parallel mechanism can be calculated by many different ways. Depending on the analysis, the forward kinematic equations can be based on different robot parameter sets. This work classifies three types of parameter in the kinematic equation:

- the basic parameter that defines the position of the robot regardless of its pose
- the sensed joint parameter that can be measured directly by the actuator
- the unsensed joint state (typically in a spherical joint) that can only be solved in conjunction with other kinematic chains in the robot.

The architecture of the calibration algorithm has been designed to avoid requiring any information from the unsensed joint state, which significantly simplifies the analysis. The first step of computing R requires the computation of the cable point locations. These locations can be described by 7 consecutive elementary transformations that map the position of the cable points relative to the robot base frame. It takes 3 elementary transformations (hence 3 robot parameters) to identify the motor origin with respect to the robot base. Two additional orientation transformations are needed to describe the motor axis with respect to the robot coordinate system. The last two transformations represent the amount of rotation on the motor and the amount of translation from the motor origin. This seven step transformation can be divided in to two almost identical transformations. Figure 6-2 shows the first four elementary transformations that partially align the motor axis to the robot coordinate system.

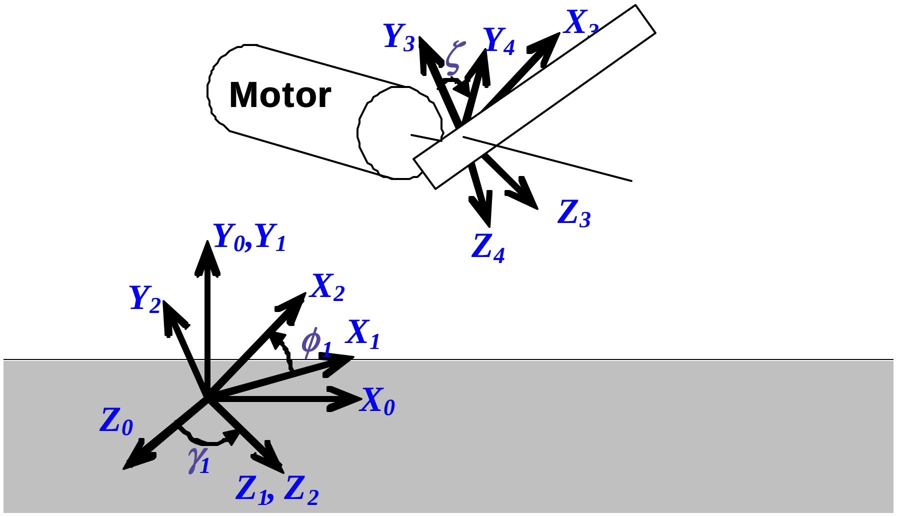

Figure 6-2: Transformation of Robot Base Frame to the Motor Axis

Note that the analysis does not follow the Denavit-Hartenberg convention. The method shows several symmetries that simplify the analysis.

The first step of the transformation is to line up the X-axis so that it points to (but is not aligned with) the motor revolute joint. From the parameterization point of view, this can be achieved by two successive rotation about the Y-axis and the Z-axis.

$$T_0^2 = \mathrm{Rot}\,(Y_0;\, °_1)\,\mathrm{Rot}\,(Z_1;\, Á_1)$$

Now that the X-axis points to the motor joint, the next step is to translate the frame by the motor separation distance, L_0 along the X-axis. A secondary rotation about the X-axis is needed for lining up the resulting Z-axis to the motor revolute joint:

$$T_2^4 = \mathrm{Trans}\,(X_2;\, L_0)\,\mathrm{Rot}\,(X_3;\, ³)$$

The overall transformation up to this point can be represented by a transformation operation:

$$\begin{aligned} T_0^4(\gamma_1;\phi_1;\zeta;L_0) &= \mathrm{Rot}(Y_0;\gamma_1)\,\mathrm{Rot}(Z_1;\phi_1)\,\mathrm{Trans}(X_2;L_0)\,\mathrm{Rot}(X_3;\zeta) \\ &= \begin{bmatrix} C_{\gamma_1}C_{\phi_1} & -C_{\gamma_1}S_{\phi_1}+S_{\gamma_1}S_\zeta & C_{\gamma_1}S_{\phi_1}S_\zeta+S_{\gamma_1}C_\zeta & L\,C_{\gamma_1}C_{\phi_1} \\ S_{\phi_1} & C_{\phi_1}C_\zeta & -C_{\phi_1}S_\zeta & L\,S_{\phi_1} \\ -S_{\gamma_1}C_{\phi_1} & S_{\gamma_1}S_{\phi_1}C_\zeta+C_\gamma S_\zeta & -S_{\gamma_1}S_{\phi_1}S_\zeta+C_{\gamma_1}S_\zeta & -L\,S_{\gamma_1}C_{\phi_1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{6.7}$$

where C_θ and S_θ are the short hands for $\cos(\theta)$ and $\sin(\theta)$ respectively.

The next phase of the analysis is to bring the frame to the cable point. The idea is to line up frame 4 with the motor axis, rotate the frame by the required joint angle, ϕ_2 and translate the frame by the length of the arm in the kinematic chain. If Z_4 were to be aligned with the motor axis, this would have been the same transformation operation described by Equation (6:7) with only the change in the parameter values. In this case:

$$T_4^7 = \mathrm{Rot}(Y_4;\gamma_2)\,\mathrm{Rot}(Z_5;\phi_2)\,\mathrm{Trans}(X_6;L_1)\,\mathrm{Rot}(X_7;0)$$

The overall frame transformation, T_0^7 is the product of T_0^4 and T_4^7. It coincides with the top cable points, P_1 and the position of the frame is embedded in its translational submatrix:

$$P_1 = \left\{ \begin{array}{c} L_0C_{\gamma_1}C_{\phi_1} + L_1(C_{\gamma_1}C_{\phi_1}C_\zeta C_{\gamma_2}C_{\phi_2} - C_{\gamma_1}S_{\phi_1}C_\zeta C_{\phi_2} \\ +S_{\gamma_1}S_\zeta S_{\phi_2} - C_{\gamma_1}S_{\phi_1}S_\zeta S_{\gamma_2}C_{\phi_2} - S_{\gamma_1}C_\zeta S_{\gamma_2}C_{\phi_2}) \\ \\ L_0S_{\phi_1} + L_1\left(S_{\phi_1}C_{\gamma_2}C_{\phi_2} - C_{\phi_1}C_\zeta S_{\phi_2} + C_{\phi_1}C_\zeta S_{\gamma_2}C_{\phi_2}\right) \\ \\ -L_0S_{\gamma_1}C_{\phi_1} + L_1(-S_{\gamma_1}C_{\phi_1}C_{\gamma_2}C_{\phi_2} + S_{\gamma_1}S_{\phi_1}C_\zeta S_{\phi_2} \\ +C_{\gamma_1}S_\zeta S_{\phi_2} + S_{\gamma_1}S_{\phi_1}S_\zeta S_{\gamma_2}C_{\phi_2} - C_{\gamma_1}C_\zeta S_{\gamma_2}C_{\phi_2}) \end{array} \right\} \tag{6.8}$$

Although this analysis uses 7 parameters to compute the kinematic chains, the cable length, L_2 (distance between two unsensed joints in general) is never used. This is generally true for this type of analysis. While it may be computationally more efficient and less susceptible to numerical issues if the cable length were treated as a separate parameter, this document work includes the cable length as the 8th kinematic parameter to maintain consistency throughout the analysis. The next step in the full kinematic analysis is the computation of the residual gradient. From Equation (6:1), the implicit partial derivative of the cable distance is:

$$\frac{\partial L_2}{\partial \rho} = \frac{\left(\overrightarrow{P_2} - \overrightarrow{P_1}\right)^T}{L_2} \frac{\partial}{\partial \rho} \left(\overrightarrow{P_2} - \overrightarrow{P_1}\right) \tag{6.9}$$

Since the true value of L_2 is unknown, the best approximation that can be used is $\overline{L_2}$. Similarly, $\overrightarrow{P_1}$ and $\overrightarrow{P_2}$ can only be computed using an estimated value or measured from noisy data. Therefore, their value can only be approximated by $\overline{P_1}$ and $\tilde{P}_2$. Moreover, $\overrightarrow{P_2}$ is obtained through measurements that are independently of calculations involving the kinematic parameters. Therefore, Equation (6:9) leads to:

$$\frac{\partial \tilde{L}_2}{\partial \rho} = - \frac{\left(\tilde{P}_2 - \overline{P_1}\right)^T}{\overline{L_2}} \frac{\partial P_1}{\partial \rho} \tag{6.10}$$

The estimation of the residual gradient in Equation (6:2) becomes:

$$\overline{R}(\rho) = \begin{cases} \frac{(\overline{P_1} - \tilde{P}_2)^T}{\overline{L_2}} \nabla_\rho P_1 & \text{if } \rho \neq L_2 \\ 1 & \text{if } \rho = L_2 \end{cases} \tag{6.11}$$

The remaining work is to take the partial derivative of Equation (6:8), and back-substitute into Equation (6:11). Although it involves a fair amount of algebra to calculate the gradient from Equation (6:8), such a closed form expression can be manipulated easily with a symbolic solver. Appendix B consists of a sample Matlab^{TM} code that performs symbolic partial derivative by the symbolic toolbox that has internal call to a Maple^{TM} symbolic solver.

Measurement Error Consideration

One minor item that has been ignored in the above analysis is the noise in the measurement system. This affects the value of $\overrightarrow{P_2}$ and leads to uncertainty in the calibration process.

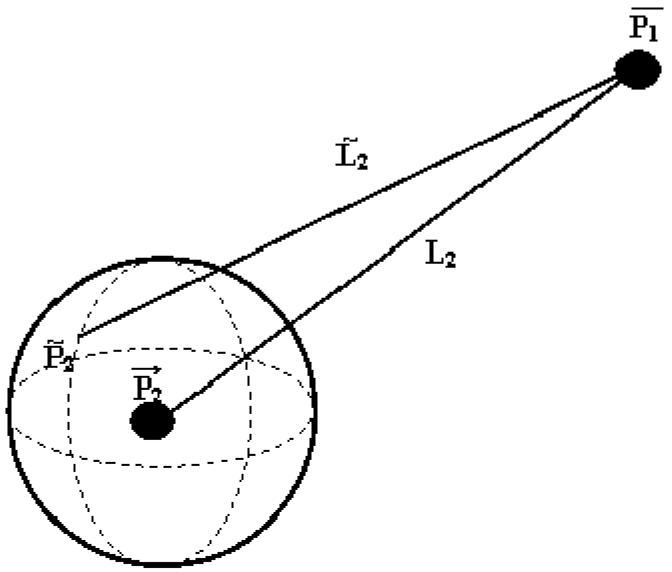

Figure 6-3: A graphical representation of measurement error effect on the kinematics chain

The effect of the noise can be compensated if one knows the characteristic distribution of the measurement noise. This subsection describes a compensation method for a Gaussian white measurement noise that is characterized by a measurement variant of $¾^2$. This analysis would be more complex if the measurement noise is correlated. The sphere in Figure 6-3 represents an isoprobability surface of the measured end-effector position, $\tilde{P}_2$. The figure has shown qualitatively that Gaussian white noise imposes a positive bias on the measured cable length $\tilde{L}_2$ as there is a larger region on the isoprobability sphere; this would make $\tilde{L}_2$ appear to be further away from the true dimension.

A quantitative estimate of this bias is possible by taking the expectation of $\tilde{L}_2$. Let ϵ_r, ϵ_n, and ϵ, be three orthogonal random deviation with a $¾^2$ variant, and ϵ_r be the deviation that is parallel to L_2. From Equation (6:1), the measured cable length for $\tilde{L}_2$ is expected to be:

$$E\left[\tilde{L}_2\right] = E\left[\sqrt{\left(\tilde{P}_2 - \overline{P_1}\right)^T \left(\tilde{P}_2 - \overline{P_1}\right)}\right] \qquad (6.12)$$

Equation 6.12 is a non-linear equation, that cannot be solved analytically. The following analysis uses a first order approximation to estimate the numerical value of $\tilde{L}_2$:

$$
\begin{aligned}
E\left[\tilde{L}_2\right] &= E\left[\sqrt{\left(\tilde{P}_2 - \overline{P_1}\right)^T\left(\tilde{P}_2 - \overline{P_1}\right)}\right] \\
&= \sqrt{\left(\overrightarrow{P}_2 + \begin{Bmatrix}\epsilon_x \\ \epsilon_y \\ \epsilon_z\end{Bmatrix} - \overline{P_1}\right)^T \left(\overrightarrow{P}_2 + \begin{Bmatrix}\epsilon_x \\ \epsilon_y \\ \epsilon_z\end{Bmatrix} - \overline{P_1}\right)} \\
&= E\left[\sqrt{\begin{bmatrix}\epsilon_x & \epsilon_y & \epsilon_z\end{bmatrix}\begin{Bmatrix}\epsilon_x \\ \epsilon_y \\ \epsilon_z\end{Bmatrix} + 2\left(\overrightarrow{P}_2 - \overline{P_1}\right)\begin{Bmatrix}\epsilon_x \\ \epsilon_y \\ \epsilon_z\end{Bmatrix} + \left(\overrightarrow{P}_2 - \overline{P_1}\right)^T\left(\overrightarrow{P}_2 - \overline{P_1}\right)}\right] \\
&= E\left[\sqrt{\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2 + 2\begin{bmatrix}\epsilon_x & \epsilon_y & \epsilon_z\end{bmatrix}\left(\overrightarrow{P}_2 - \overline{P_1}\right)^T + \overline{L_2}^2}\right] \\
&= \overline{L_2}E\left[\sqrt{1 + \frac{\left(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2\right)}{\overline{L_2}^2} + \frac{\begin{bmatrix}\epsilon_x & \epsilon_y & \epsilon_z\end{bmatrix}\left(\overrightarrow{P}_2 - \overline{P_1}\right)^T}{\overline{L_2}^2}}\right] \\
&= \overline{L_2}E\left[1 + \frac{\left(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2\right)}{2\overline{L_2}^2} + \frac{\begin{bmatrix}\epsilon_x & \epsilon_y & \epsilon_z\end{bmatrix}\left(\overrightarrow{P}_2 - \overline{P_1}\right)^T}{2\overline{L_2}^2} + O(2)\right] \\
&\approx \overline{L_2} + \frac{1}{2\overline{L_2}}\left(E\left[\left(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2\right)\right] + E\left\{\begin{bmatrix}\epsilon_x & \epsilon_y & \epsilon_z\end{bmatrix}\left(\overrightarrow{P}_2 - \overline{P_1}\right)^T\right\}\right) \\
&\approx \overline{L_2} + \frac{1}{2\overline{L_2}}\left(E\left[\left(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2\right)\right] + E\left\{\begin{bmatrix}\epsilon_x & \epsilon_y & \epsilon_z\end{bmatrix}\left(\overrightarrow{P}_2 - \overline{P_1}\right)^T\right\}\right) \\
&\approx \overline{L_2} + \frac{3}{2}\frac{\sigma^2}{\overline{L_2}}
\end{aligned}
$$

As an approximation, the zero bias estimation of the cable length is:

$$
\overline{L_2} \approx \tilde{L}_2\left(\frac{1}{2} + \sqrt{\frac{1}{4} - \frac{3}{2}\frac{\sigma^2}{\tilde{L}_2^{\,2}}}\right) \tag{6.13}
$$

Singularity Consideration

While the LSE approach is simple to implement, it has some drawbacks in its robustness. As the calibration routine executes, there are situations when the columns of the residual gradient matrix, ∇R in Equation (6:6) start to align with one to another and the gradient matrix gradually loses its rank. Although it is rare to have a complete loss of rank in ∇R, it does pose numerical issues when solving the system iteratively. To avoid those problems from occurring, it is vital to avoid updating any set of kinematic parameters that is close to the null space of ∇R. A singular value decomposition (SVD) technique would be a numerical technique to address the problem. Every $N \times M$ matrix Q can be transformed into:

$$Q = USV^T$$

with two orthonormal (unitary if Q contains imaginary element) matrices U and V, and $S \in R^{M,N}$ is a diagonal matrix whose elements, σ are the singular values of Q.

The singular values of Q is analogous to the eigenvalues of a square matrix, and they are a reliable quantitative measure of the rank of Q. Singular values are usually sorted in descending order. As a general rule of thumb, a matrix is said to be ill-conditioned if the magnitude of the condition number (the ratio of the largest and the smallest singular value) is larger than or approximately equal to the precision of the computation. An ill-conditioned matrix is susceptible to numerical error and they should be avoided. This leads to solving the system of equations only on the directions that are well defined in the range of Q [27]. Suppose the maximum allowable condition number is 10^8 [1] and σ_m is smallest singular value that satisfies the condition, the parameter update of Equation (6:6) can be computed by:

$$\delta\rho = V \begin{bmatrix} \frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_m} \\ & & 0_{M-m} & \end{bmatrix} \begin{bmatrix} I_m \\ 0_{M-m} \end{bmatrix} U^T R \tag{6.14}$$

[1] It is more convenient to implement this condition as the $\min(\sigma)/\max(\sigma) > 10^{-8}$

where λ_i are the singular values of ∇R, I_m is the identity matrix of size m and 0_{M-m} is the zero matrix of size $M - m$.

The proof of Equation (6.14) requires two additional mathematical facts about the SVD of a matrix $\nabla R \in \mathbb{R}^{N,M}$ that has more row than column ($N > M$):

– The i^{th} column of U is a basis that is associated with λ_i. If λ_i is non-zero, it represents a basis of the range space of ∇R.

– The SVD can be written in a "reduced form" with S begin a diagonal matrix containing the singular value of ∇R, and a $U_r \in \mathbb{R}^{N,M}$, which composes the first M columns of R. Note that the orthonormal property of U implies that $U_r^T U_r = I_M$

Applying the reduced form of SVD to ∇R, Equation (6.5) becomes:

$$R = U_r \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_M \end{bmatrix} V^T \delta\rho$$

it can be rearranged to:

$$U_r^T R = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_M \end{bmatrix} V^T \delta\rho$$

Since it is assumed that the computer cannot distinguish between zero and all singular values that is smaller than λ_m, the useful range space of ∇R consists of the first m columns of U_r. Therefore, anything beyond the m^{th} column is meaningless. Equivalently, all elements of $U_r^T R$ that is below the m^{th} row can be rounded to zero without losing any useful information. The product of $\begin{bmatrix} I_m & \\ & 0_{M-m} \end{bmatrix} U^T R$ in Equation (6.14) served that purpose.

The solution of this system of equations can be obtained by pre-multiplying V and the inverse of S. However the singular values that are beyond λ_m are small, which leads to a 1/0 error. This can be avoided if one takes only the inverse of the non-singular portion of S and appends a 0_{M-m} matrix to maintain the dimension. Note that the elements that are artificially appended to the matrix inverse do not affect the final results as these numbers would be multiplied by a zero in the lower part of $U_r^T R$.

The following algorithm summarizes the LSE calibration algorithm:

Step 1: Set k = 0, and the initial parameter estimate ½[0] to be the designed robot parameter. Then obtain a list of the measured end-effector positions Â and the corresponding sensed joint angles, $Á_2$.

Step 2: Calculate R by Equation (6:2), compensate the residual by the expectation of the measurement error if possible. Also compute 5 R by Equation (6:11). Evaluate both expressions by the most updated parameter estimates at each of the measurement point.

Step 3: If the norm of R is smaller than a predetermined value, the calibration goal is considered to be reached. Return the estimated robot parameters and exit the algorithm.

Step 4: Computed the singular value decomposition 5 R and calculate the parameter error, ±p by Equation (6:14)

Step 5: Update the parameter estimate ½[k + 1] = ½[k] + ±p, then update k = k + 1. go to Step 2

6.1.2 Convergence Study

In an iterative algorithm, the convergence rate is an important measure of the system efficiency. A good optimization algorithm must be able to converge to the minimum point quickly. While the convergent properties are difficult to measure experimentally, this work studies the convergent properties of the presented algorithm through numerical simulation. The first study imposes random errors on the presented 2D-Deltabot. A set of 800 measurement points were computed using the true kinematic parameters and randomly generated joint angle. A normally distributed random error is used to contaminate the first 400 true measurement points. The LSE calibration algorithm was applied on the contaminated measurements. The effectiveness of the algorithm is based on comparison between the predicted position computed from the resultant parameters and the other 400 true measurement points. This study requires the full forward kinematic model of 2D-Deltabot, and it can be found in Appendix A of this

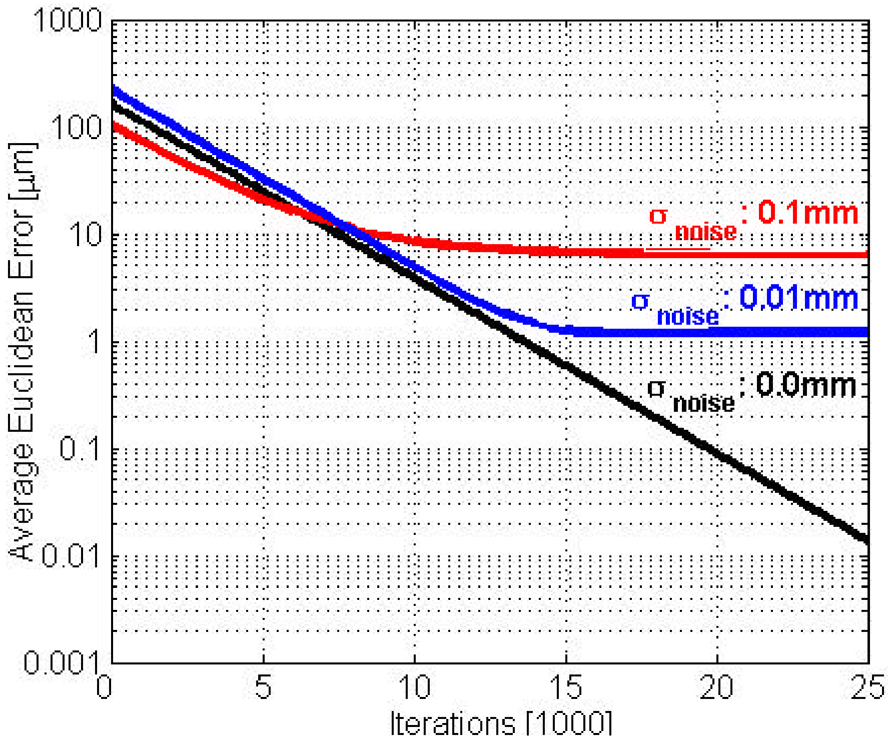

Figure 6-4: Convergence Study at Various Measurment Noise Level

thesis work. Figure 6-4 shows a typical convergent graph for the LSE algorithm with different noise variance. While the initial prediction error varies depending on actual random deviations being applied to the model, the convergent rate and the convergent point of the algorithm is repeatable. If there is no measurement noise, the presented algorithm appears to converge to the true robot parameters. For the Contaminated measurement data, the presented algorithm is capable of converging to a set of robot parameters with an accuracy approximately equal to 10% of the standard deviation of the measurement noise. This is considered acceptable as the calibration algorithm is capable of rejecting more than 90% of the noise to produce a solution that is even more accurate than the measurement scheme. Since the average prediction error declines linearly on a semi-log scale until the system converges, it is hypothesized that the calibration algorithm makes the position error declines exponentially with respect to the iteration. Although it takes relatively large amount of iterations to achieve a good calibration result, the error is acceptable if it is below 0:01mm in practice.

Convergence alone does not justify the effort of implementing a new optimization algorithm, the other factor that needs to be taken into consideration is the time required to obtain a solution. As mentioned early in this section, kinematic calibration is usually performed with direct kinematic model and standard optimization algorithm. Matlab^{TM} optimization toolbox box is one of the most popular optimization tools available in the market. It has features that automatically select the most suitable algorithm for the objective function. The second study is to benchmark the presented algorithm against the Matlab^{TM} optimization toolbox. This study compares both the time and the convergent performances of the two methods in 100 test cases. The same test cases were minimized by both methods on the same computer. This minimizes the uncertainty due to the computer performance. The test procedure is similar to the first study except the number of iterations has been capped and the amount of random deviation has been reduced to minimize the computational time. Figure 6-5 shows the resulting average prediction error between the two methods. It has been shown that the LSE algorithm produces a better approximation of the robot parameter than the Matlab^{TM} optimization toolbox. Furthermore, the LSE approach produces a result every 2 minutes, while Matlab^{TM} optimization toolbox needs between 17 and 18.25 minutes for each computation. Note that this is not to say the LSE method is superior than the Matlab^{TM} toolbox in general. The large difference in performance is mainly due to the availability of the first order information in the LSE approach, whereas the Matlab^{TM} toolbox is constrained with a direct line-search algorithm. Therefore, the LSE approach is more suitable for this analysis.

6.2 Accuracy Test

With a reliable calibration algorithm, it is now more meaningful to perform the accuracy test to check if the robot is capable of reaching its intended position. An optical measurement system, Krypton Rodym6D coordinate measuring machine was used for an accurate measurement of the end-effector position. The Krypton system consists of 3 cameras that track the position of high frequency LEDs. It provides a reference for comparison between the actual robot position and the modeled position. As per the OEM specifications, the accuracy and the repeatability

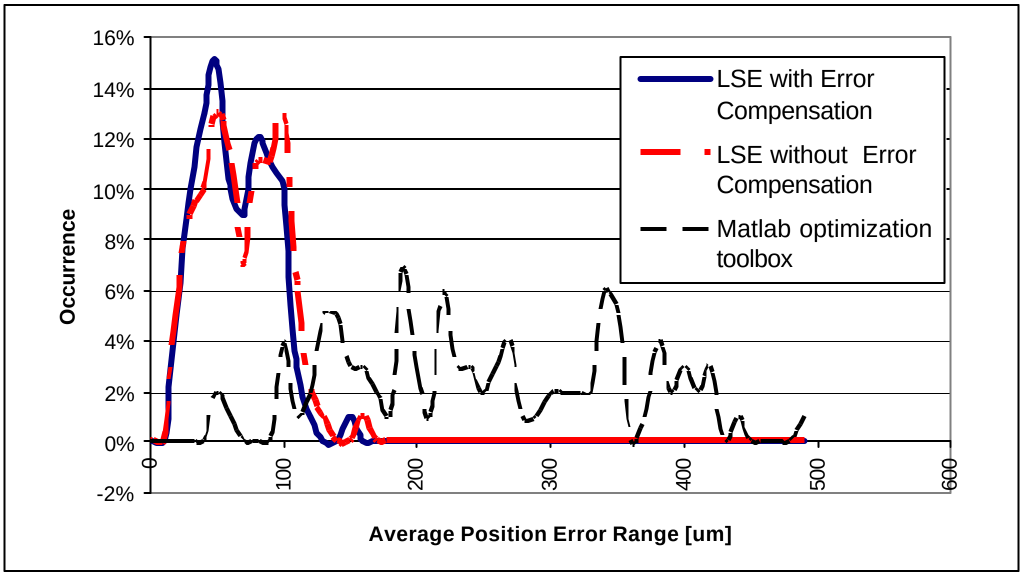

Figure 6-5: Comparison Between Different Calibration Methods

of the Krypton Rodym6D is 0.1mm and 0.01mm respectively [2]. While it was not possible to verify the accuracy of the measuring system, a separate test that used the camera to track a stationary object showed the standard deviation of the camera signal noise was approximately 7µm.

One problem in setting up the camera system is the difficulty in lining up the camera to the robot coordinate frame. Instead of accurately positioning the camera system in the laboratory, the orientation of the camera relative to the robot coordinate frame was measured using statistical methods. The idea is to jog the robot within its workspace while measuring the end-effector position in an arbitrary camera frame. Since the 2D planar robot has a planar rectangular workspace, the measurement data has a distinct spread along three orthogonal directions. Since the spread of a set of data is characterized by the covariant matrix of the data, one can determine the three characteristic directions (principal directions) by analyzing the covariant matrix of the measured data. It turns out that these directions are the eigenvectors of the covariant matrix; and the amounts of the data scatter (spread) along these eigenvectors are proportional to the associating eigenvalues. This establishes the spatial relationship between

[2] Refer to www.krypton.be. for more details

Table 6.2: Root Mean Square Position Error of 2D-Deltabot

	Before Calibration	After Calibration
In-plane	4.54mm	0.304mm
Out-of-Plane	0.435mm	0.439mm
Overall	4.56mm	0.540mm

the camera coordinate frame and the robot coordinate frame.

6.2.1 Experimental Procedure and Results

In the accuracy test, the manipulator is jogged to different locations in the workspace. The measurement points were taken inside a 700mm by 65mm rectangle, and they were separated by 14mm in the horizontal direction and 8.125mm in the vertical direction. There were 459 points used in the experiment. A motion program was written to move the end-effector amongst these points in a spiral-in pattern. There was a 200ms dwell period allocated for the camera to capture the end-effector position. The camera was setup to collect data at 50Hz. This resulted in about 6-8 measurements during the stoppage phase of the end-effector. The entire experiment was performed at a relatively low speed and the motor following error was negligible.

In the first part (calibration phase) of the testing, half of the measurement points were randomly selected and used for robot calibration, the remaining points were not used. In the second part (verification phase) of testing, the entire robot path was recalculated with the full kinematic parameters obtained from the calibration phase.

Figure 6-6 shows the end-effector position error of 2D-Deltabot before the calibration . There is a significant amount of position error in the manipulator. Table 6.2 tabulates the characteristics of the robot position error. The overall RMS robot position error was 4.56mm before the calibration. The in-plane inaccuracy accounted for most of the position error. The robot was only accurate within 6.3mm in the horizontal direction, 3.5mm in the vertical direction, and 1.1mm in the out-of-plane direction. The large amount of inaccuracy of the robot was believed to be due to the poor tolerance in the cable length.

Figure 6-7 shows the end-effector position error of 2D-Deltabot in the verification phase (after the calibration). The overall RMS accuracy for the robot was reduced to 0.540mm.

Table 6.3: The Calibrated Robot Parameters

	Designed Values	Side A	Side B
L_0	150mm	147:0mm	150:9mm
L_1	240mm	238:0mm	239:7mm
L_2	525mm	519:7mm	530:3mm
³	$0^{\pm}$	¡ $0{:}66^{\pm}$	¡ $1{:}20^{\pm}$
$°_1$	$0^{\pm}$ (Side A) $180^{\pm}$ (Side B)	¡ $0{:}36^{\pm}$	$181{:}25^{\pm}$
$°_2$	$0^{\pm}$	$1{:}56^{\pm}$	$0{:}11^{\pm}$
$Á_1$	$0^{\pm}$	$4{:}69^{\pm}$	$3{:}24^{\pm}$
$Á_2$	$0^{\pm}$	¡ $0{:}07^{\pm}$	$1{:}60^{\pm}$

While the in-plane position error had been reduced by almost 95%, the out-of-plane error remained virtually unchanged. This is not surprising as the out-of plane motion is constrained solely by a passive RPR (cylinder) link. Its error is dictated by the quality of the hardware used in the robot and cannot be adjusted by any software calibration. This result shows that the calibration method is effective, but it is limited by the out-of plane accuracy of the robot.

Table 6.3 tabulates the calibrated parameters of the robot after the calibration process. The large amount of in-plane rotations, $Á_1$ suggest that the robot coordinate frame used in the analysis may be misaligned with the designed robot coordinate frame. This is possible as the camera reference frame (analysis frame) was established by the uncalibrated robot measurements. However, this is not a concern as long as the final robot kinematics are consistent with the reference frame. When the robot is mounted on an assembly line, the calibration process should be preformed using measurements that are accurate with respect to the rest of the operating environment. The calibration process would then adjust the robot parameter to compensate any installation error that may have occurred.

As a pick-and-place tool, the confidence level of the position error is sometimes a better performance metric to gauge robot accuracy. The in-plane position errors of these measurements (in the second test) are plotted in Figure 6-8. The maximum standard deviation of the position error is 0.24mm. In order to provide a better representation of the manipulator accuracy, a probability density function of the position error is constructed using the Parizen method with a Gaussian window function. The numbers on the isoprobability curves in Figure 6-8 show

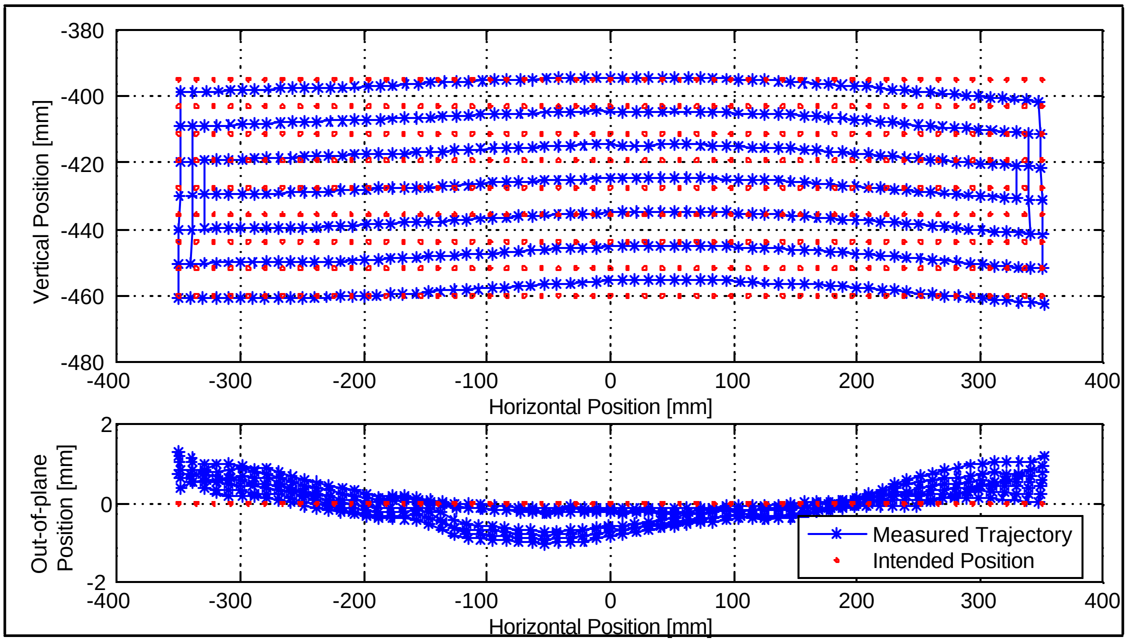

Figure 6-6: The Accuracy of the Prototype 2D-Deltabot before Calibration.

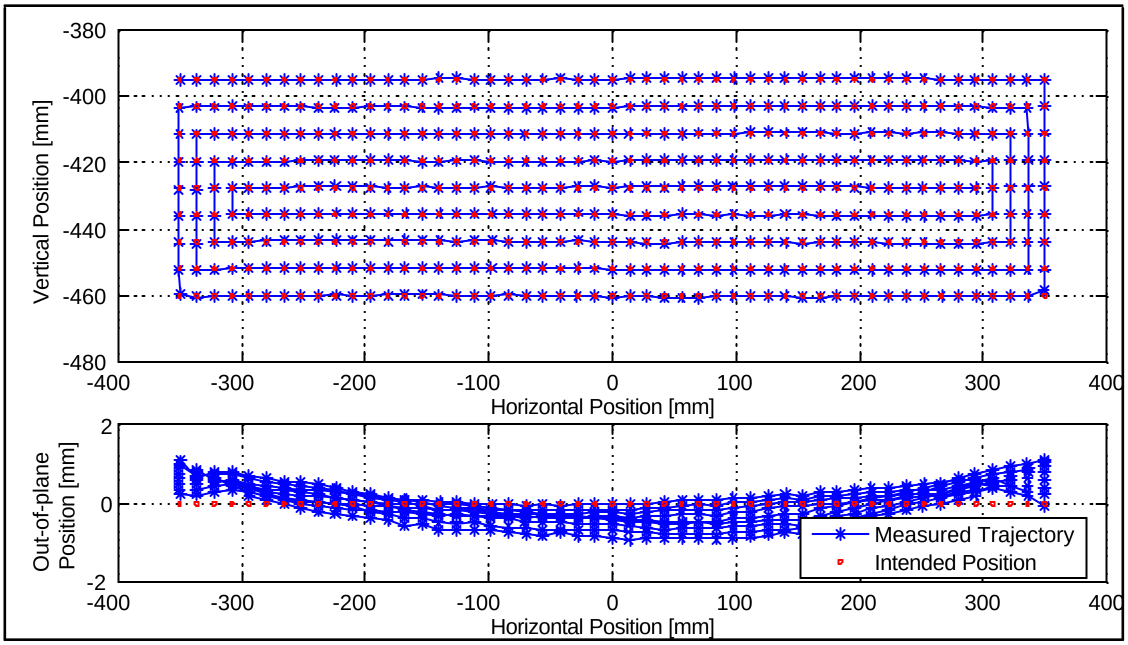

Figure 6-7: The Accuracy of the Prototype 2D-Deltabot after Calibration

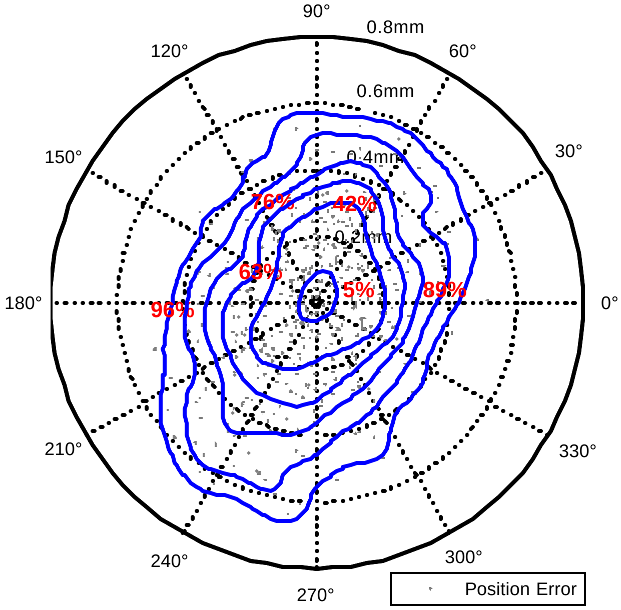

Figure 6-8: Position Error of the Prototype 2D-Deltabot after Calibration

the confident levels of the position error that falls inside the respective encirclements. The covariant of the Gaussian window used in this study is set to be 0:004mm^2 (0.063mm standard deviation or roughly one quarter of the maximum standard deviation of the overall position error). Since the probability density function is constructed by a relatively conservative window function, the confident level is also conservative. For instance, approximately 62% of the measurements fall inside the 43% confident level curve. Based on the experimental results, it has concluded that the accuracy of the prototype is §0:62mm (0:024").

6.2.2 Observations

After further examinations on the prototype and the measurement data, it is believed that the accuracy problem is most likely caused by an out-of-plane motion due to a pair of imperfect clevis bushings on the upper cylinder revolute joint. The original bushings were loose and misaligned. This results in a free-play on the revolute joint. The defect caused up to 1.2mm out-of-plane motion on the end-effector, which was never modelled in the kinematic analysis. Further enhancement on the prototype includes replacing the clevis joint with an Ø10mm bearing to improve the performance of the robot.

6.3 Repeatability test

Similar to the robot accuracy, the robot repeatability is also an important metric to gauge the quality of a robot. The objective of the repeatability test is to check if the robot is capable of reaching the same point repeatedly under the normal operating conditions. It is important to have the robot operate at a high speed in this test to produce meaningful results. In this test, the manipulator was commanded to move along a path that was 76.2mm up, 635mm across, and 76.2mm down. One repetition consisted of commanding the end-effector to go from one end of the path to the opposite end and then return to its starting point. There was a 100ms dwell time allocated at the end of the path for the camera to capture the end-effector position. The overall motion time including the dwell period was 0.902s. If a typical of 20ms dwell period were used in the trajectory, the manipulator would have operated at 81 cycles per minute. The test was preformed with a properly tuned motor and a well designed trajectory profile as illustrated in Subsection 6.3.1. The experiment result is presented in Subsection 6.3.2.

6.3.1 PMAC and Trajectory Generation

Unlike the accuracy test where the manipulator moves slowly over a short distance, the large displacement in the repeatability test could lead to serious structural damage if the actuators were not tuned properly. The tuning of the actuators was performed through the Programmable Multiple Axis Controller (PMAC), a powerful industrial motion controller for robotic applications. PMAC has a PID and feed forward controls structure that controls the motors to perform coordinated motions. It has several options to generate the command motion profile. The command generator chosen for this experiment is Spline1, which uses cubic spline to interpolate successive command points in the motion profile.

In addition to the actuator control, it is vital to plan the trajectory (time associated path of motion) ahead to avoid excess acceleration or jerk during the motion. A trajectory planning program has been developed in Matlab^{TM} to generate the command points for PMAC. The core of the program is to generate a non-dimensional trajectory for a generalized time-space mapping.

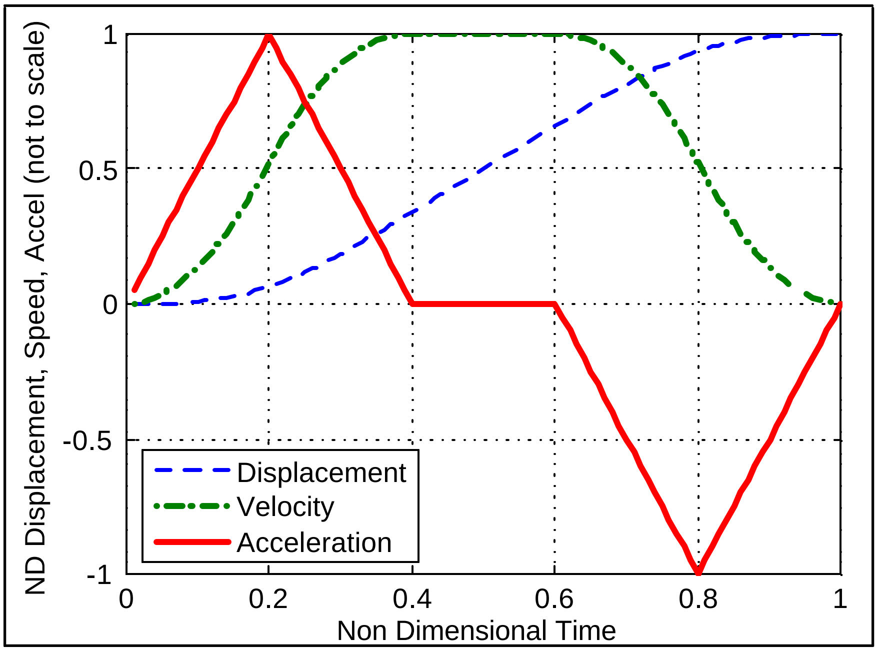

Figure 6-9: Trajectory Profile for High Speed Motion

The trajectory has a triangular acceleration profile and the subsequent kinetics (velocity and displacement) were derived assuming zero initial conditions. A triangular acceleration profile ensures a finite jerk and smooth velocity profile on the end-effector, which is easier to track by the motors. Figure 6-9 shows the trajectory profile for the manipulator. The vertices in the acceleration profile are the tunable setting that were determined experimentally. Once the trajectory is generated, eleven characteristic points were picked from the displacement profile evenly in the time domain. These points would be mapped to the actual motion path for the PMAC motion program. Figure 6-10 shows how the control points were selected from the displacement profile at every 10% interval of the motion time. The path shown in the figure represents a motion from one end of the path (point A) to the opposite end (point B); the return trajectory assumes a symmetrical profile.

Figure 6-11 shows two typical cycles of the commanded joint trajectory and the resulting joint error after some fine tuning on the motors. There are still considerable amount of following error during the middle of the motion, but the error settled down near the dwell points [3]. This

[3] Dwell point is a PMAC terms that represents the stall portion of a trajectory [6].

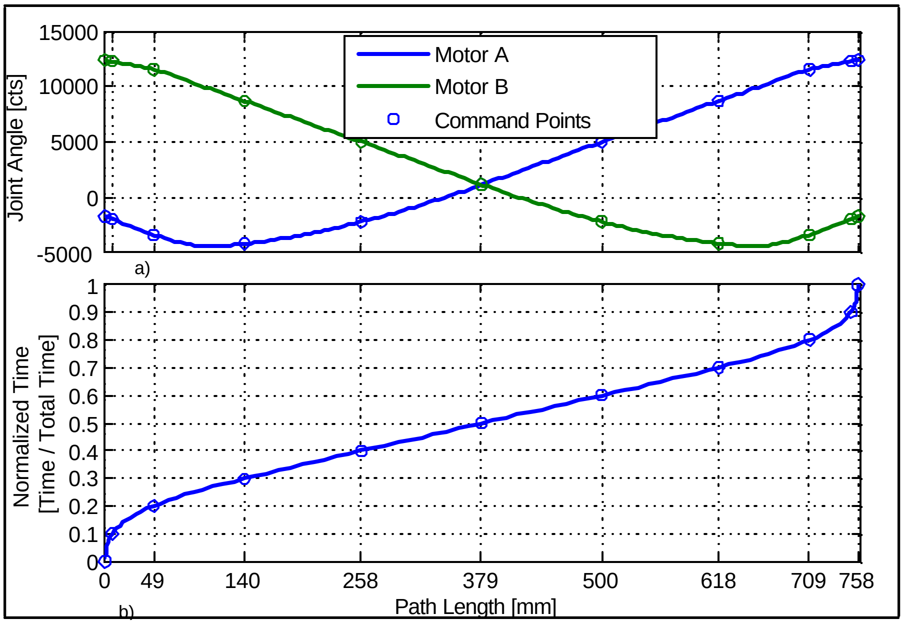

Figure 6-10: Selection of the control point for PMAC motion program. a) The joint angle expressed in the path space. b) The time expressed in the path space

indicates a stable condition for the robot, at least from the joint space point of view. This was considered safe to run the repeatability test for a long period of time.

6.3.2 Experiment Procedure and Results

In the repeatability test, the manipulator was run unsupervised along the testing trajectory for approximately 20 minutes. While the camera measured the end-e¤ector position at 100Hz, it was stopped once every 2 minutes to prevent data over‡ows. The end-e¤ector was considered to be 'stalled' if the measured position is less than 0.1mm away from its previously measured position. Once a stalled point is detected, the subsequent measurements would be placed in an array of stalled position until the end-e¤ector started to move again. At this point, the length of the array would be checked, and the array is considered to represent the robot dwell position if and only if the array contains between 8 and 12 measurements. The measurement that is at the center of the array is registered as the dwelled position. These measured dwelled points are placed in two groups (A and B) for either end of the path. The position repeatability is

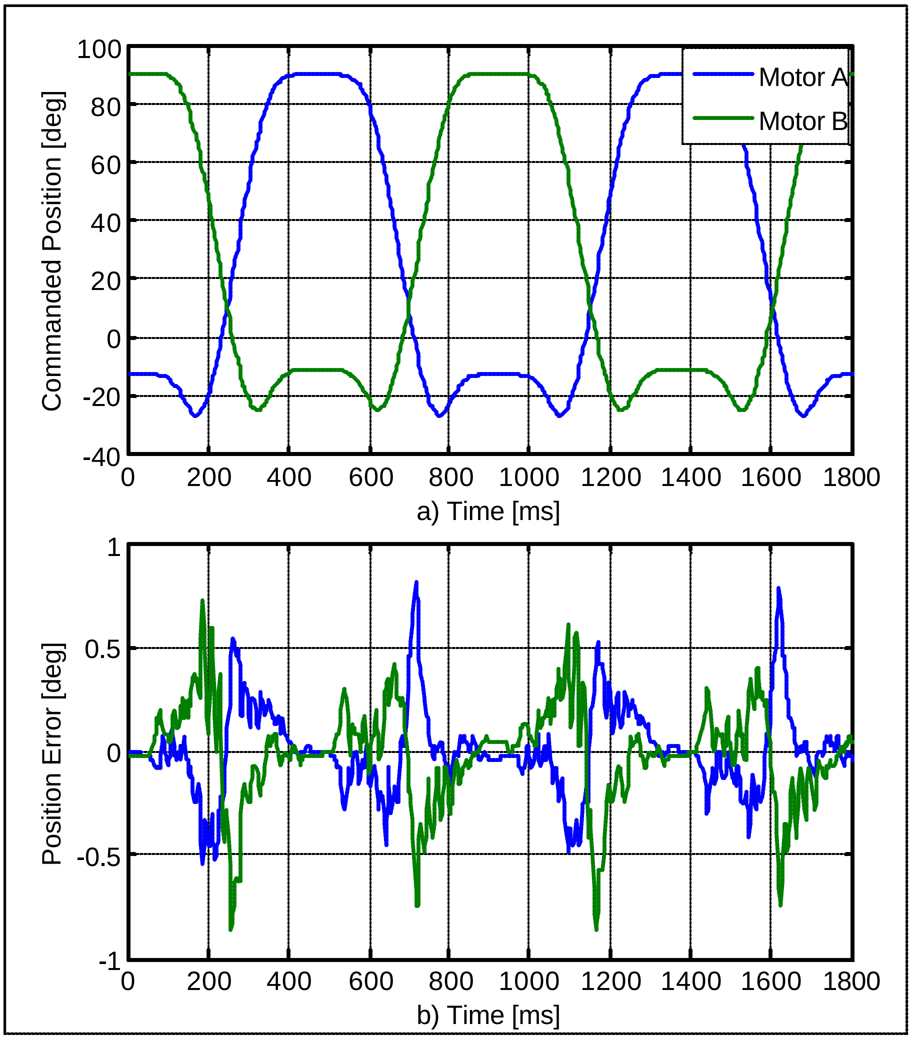

Figure 6-11: a) Joint (arm) commanded position b) Resulting joint (arm) position following error

Table 6.4: The Repeatability of the Prototype 2D-Deltabot

	Group A	Group B	Total
Number of Measurements	805	802	1607
Principle Variants	397 ¹m^2 1406 ¹m^2 2266 ¹m^2	432 ¹m^2 1622 ¹m^2 2658 ¹m^2	415 ¹m^2 1516 ¹m^2 2456 ¹m^2
Repeatability	0:143mm	0:155mm	0:149mm

calculated as follows:

$$C_X = \frac{\left(X - \bar{X}\right)^T \left(X - \bar{X}\right)}{N_X - 1}$$

$$v_X = \mathrm{eig}(C_X)$$

$$RP = 3\sqrt{\max(v_x)}$$

where C_X is the covariance matrix of the measurement.

$\bar{X}$ is the mean position of the measurement group.

N_X is the number of measurement in the group.

eig computes the eigenvalues of a matrix. The eigenvalues of the covariance matrix represent the principal variance of the data.

RP is the repeatability of the robot.

Table 6.4 summarizes the results of the repeatability test. This experiment found that repeatability of the robot is 0:15mm. This is about one …fth of the accuracy of the robot. It was observed that the robot supporting pillars were not sti¤ enough to hold the structure stationary when the robot is running at the operating speed. The end-e¤ector is likely to vibrate considerably when the structural vibration is coupled with the free-play in the clevis bushing. A sti¤er supporting pillar should replace the existing units before further repeatability tests are conducted on the robot.

Chapter 7

Conclusions

The goal of this thesis work was to investigate and validate the design and implementation of a class of 2D cable-based parallel manipulators. This thesis work expands the research work by Prof. Khajepour and Dr. Behzadipour that uses cable as a mechanical component in a parallel manipulator design. The fundamental design strategy was to constrain the orientation of the end-effector by cables so that the robot had only two pure translational degrees of freedom in a plane. With a light weight cable construction, the resulting robot had little inertia and was able move rapidly within its workspace. Two new robot designs, 2D-Deltabot and 2D-Betabot were introduced to illustrated this concept. This work proved that certain manipulator configurations were inherently tensionable. Those tensionable members could be constructed by cable wire to reduce moving inertia. This work also showed that some features in the end-effector had direct impact on the overall performance of the robot.

Although it was useful to know if certain robot configurations were inherently tensionable, the robot implementation would not be successful if the design parameters were not optimized properly. This work illustrated a straight forward and computationally feasible method to calculate various limits on the robot dynamic characteristics. The analysis was based on inverse kinematic model and it does not require the complicated forward kinematic analysis. The resulting information provided the building blocks for constructing the cost function of a given set of robot parameters. The rest of the optimization process was simply to combine the cost function with a direct search method. This work showed that the proposed 2D-Betabot was able to achieve 4g acceleration everywhere within its workspace with a pair of 890W rotary actuators, and the 2D-Deltabot could do the same with a pair of 648W rotary actuators. The 2D-Deltabot had been chosen for prototyping and it was designed and fabricated over the course of this study.

Upon completing the robot fabrication, the 2D-Deltabot went through a series of tests to evaluate its accuracy and repeatability. In light of the possibility that numerous robot designs could be derived from this work, a general calibration algorithm was developed for a quick calibration procedure. The presented calibration algorithm converged to the true robot model in simulation, and it was an order of magnitude faster than the conventional standard optimization methods. The experimental evaluations showed that:

1. The robot matched fairly accurately to its commanded position that was predicted by the kinematic model. The 95% confident level of the robot accuracy was determined to be within 0.62mm. There were considerable amounts of out-of-plane motion in the end-effector that limited the accuracy of the robot. This problem was purely due to fabrication error, and it should be addressed through hardware replacement.

2. The robot arms were capable of tracking the commanded path at high speed after tuning the motors. The arms tracked a 730mm long path within $0.7^{\pm}$ in the joint space at 67 cycles per minute (81 cycles per minute for a shorter dwell cycle). The majority of the region where a large tracking error occurred was concentrated at the middle of the trajectory. Those were the regions in the pick-and-place path where tracking ability was not a primary concern.

3. The position of the robot end-effector is marginally repeatable. The repeatability of the robot was determined to be 0:15mm. The error was likely due to the vibration on the robot and the free-play in the cylinder revolute joint. The repeatability was expected to be improved once the mechanical problems were resolved.

With a few mechanical design changes, the 2D-Deltabot could become a very accurate and reliable pick-and-place robot. The inherently low inertia design in the presented cable based robots give them a major advantage on speed enhancement and cost reduction in the design. The potential for these manipulators to improve the speed and increase productivity for many industrial applications is substantial.

Bibliography

[1] Machinery Handbook 25. Industrial Press Inc, 1996.

[2] Ming A. and Higuchi T. Study on multiple degree-of-freedom positioning mechanism using wires (part1). International Journal of Japan Society of Precision Engineering, 28(2):131–138, 1994.

[3] Mason F. Andreatta G. and Serafini P. Advanced School on Stochastics in Combinatorial Optimization. World Scientific, Singapore, 1986 edition, 1986.

[4] Khajepour A. Behzadipour S., Dekker R. and Chan E. Deltabot a new cable-based ultra high speed robots. International Mechanical Engineering Congress and Expo, ASME, IMECE2003-41470, 2003.

[5] Stewart D. A platform with six degrees of freedom. Proceedings of the Institute of Mechanical Engineering, 180:371–386, 1965.

[6] DeltaTau. PMAC Software Manual. 1992.

[7] Chan E. The mechanical components of the ultra high speed pick and place robots. Technical report, 2002.

[8] Ghorbel F. and Chetelat O. A reduced model for constrained rigid bodies with application of parallel robots. Proc. of the IFAC Symposium on Robot Control, SYROCO, pages 57–62, Sept 1994.

[9] Kim H. and Tsai L.W. Design optimization of a cartesian parallel manipulator. Journal of Mechanical Design, transaction of the ASME, v125(n1):43–51, 2003.

[10] Kim H. and Tsai L.W. Kinematic synthesis of a spatial 3-rps parallel manipulator. Journal of Mechanical Design, transaction of the ASME, v125(n1):92–97, 2003.

[11] Zhuang H. and Roth Z. A closed form solution to the kinematic parameter identification of robot manipulators. Proceedings of International conference on Robotics Automation, pages 2682–2688, 1991.

[12] Meng Li. Chetwynd G. Huang T., Zhanxian Li. and Cosselin M. Conceptual design and dimensional synthesis of a novel 2-dof translational parallel robot for pick and place operations. Journal of Mechanical Design, (7):449–455, 2004.

[13] Ingersoll. Octahedral machine tool frame. Technical report, 1995.

[14] Cervantes-Sanchez J. and Rendon-Sanchez J. Simplified approach for obtaining the workspace of a class of 2-dof planar parallel manipulators. Mechanism and Machine Theory, v34(7):1057–1073, 1999.

[15] Dennis J. and Audet C. A pattern search filter method for nonlinear programming without derivative. SIAM Journal on Optimization, v 14(n4):980–1010, 2004.

[16] Duffy J. Statics and Kinematics with Applications to Robots. Cambridge University Press, 1996 edition, 1996.

[17] Mckinnon K. Convergence of the nelder-mead simplex method to a nonstationary point. SIAM Journal on Optimization, v 9(n1):148–158, 1998.

[18] Choe W. Kawamura S. and Tanak S. Development of an ultrahigh speed robot falcon using wire driven systems. Proceedings of IEEE International conference on Robotics and Automation, pages 215–220, 1995.

[19] Behzadipour S. Khajepour A. and Dekker R. Light weight parallel manipulators using active/passive cables. Technical report, 2002.

[20] Dekker R. Khajepour A., Behzadipour S. and Chan E. New cable-based ultra high speed robots, a review. International Conference on, Intelligent Manipulation and Grasping, IMG04, 2004.

[21] Torczon V. Lewis M. and Trosset M. Direct search methods: Then and now. Journal of Computational and Applied Mathematics, v124(n1-2):191–207, 2000.

[22] Lewis M. and Torczon V. A globally convergent augmented lagrangian pattern search algorithm for otpimization with general constraintes and simle bounds. SIAM Journal on Optimization, v 12(n4):1075–1089, 2002.

[23] J-P. Merlet. Parallel Robots. Kluwer Academic Publishers, 1st edition, 2000.

[24] Gosselin C. Merlet J. and Mouly N. Workspace of planar parallel manipulators. Mechanism and Machine Theory, v33(1-2):7–20, 1998.

[25] Vischer P. and Clavel R. Kinematic calibration of the parallel delta robot. Robotica, 16:207–218, 1998.

[26] Fournier A. Pierrot F. and Dauchez P. Towards a fully-parallel 6 dof robot for high-speed applications. Proc. of IEEE International Conference on Robotics and Automation, pages 1288–1293, 1991.

[27] Teukolsky S. Press W., Flannery B. and Vetterling W. Numerical Recipes (Fortran). Cambridge University Press, 1989.

[28] Clavel R. Device for the movement and positioning of an element in space. Technical report, 1990.

[29] Behzadipour S. and Khajepour A. Design of reduced dof parallel cable-based robots. Journal of Mechanism and Machine Theory, 39(10):1051–1065, 2004.

[30] Joshi S. and Tsai L.W. A comparison study of two 3-dof parallel manipulators: One with three and the other with four supporting legs. Proc. of IEEE International Conference on Robotics and Automation, pages 1288–1293, 1991.

[31] Landsberger S. and Sheridan T. A new design for parallel manipulators. Proceedings International Conference on Cybernetics and Society, pages 812–814, 1985.

[32] Kimura T. Hattori M. Takamori T. Tadokoro S., Nishioka S. and Maeda K. On fundamental design of wire configurations of wire-driven parallel manipulators with redundancy. Proceedings of Japan-U.S.A. Symposium on Flexible Automation, 1:151–158, 1996.

[33] Gough V.E. and Whitehall S.G. Univeral tyre test machine. Proceedings of 9th International Technical Conference F.I.S.I.T.A, pages 177–135, 1962.

Appendix A

Full Direct Kinematic Model of the 2D-Deltabot

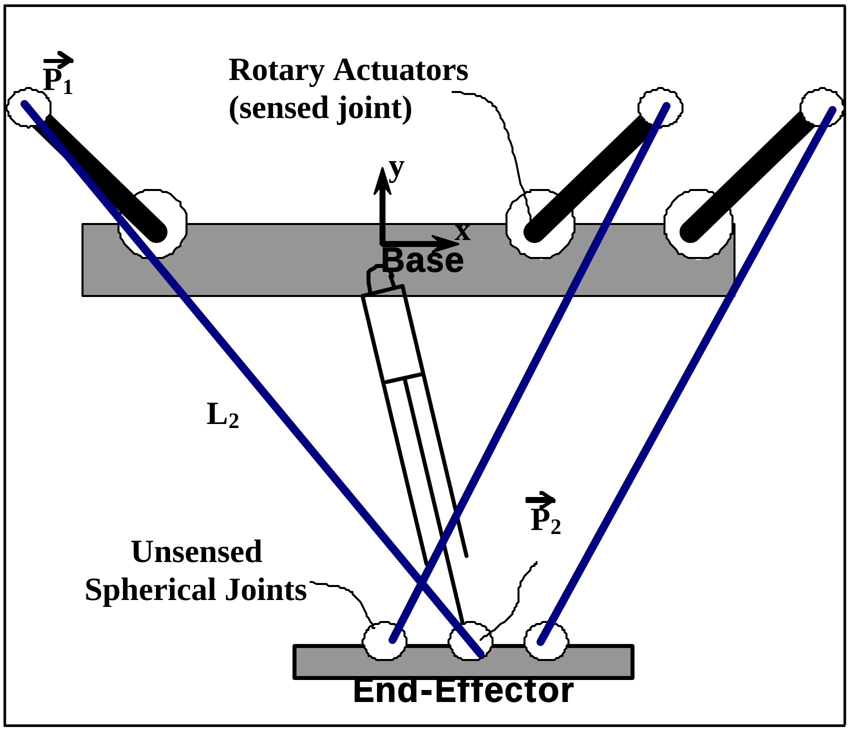

Figure 7-1: The Cable Points Needed for the Full Direct Kinematic Model

Each kinematics chain of the 2D Deltabot has 8 parameters; therefore, the full direct kinematics equation of the 2D-Deltabot consists of 16 parameters. The first step in the kinematics equation is to locate the cable arm-side attachment points, $\vec{P}_1$. as shown in 7-1. This can be done by conventional direct kinematic analysis method.

The full kinematic equations can be constructed by intersecting the Z plane with two spheres centered in these cable points with their respective cable length as their radius. The intersection of two sphere is a circle, this can be computed by forming a triangle with the a line that connects the cable points, and the two cable length. The circle of intersection is formed by revolving the corner of cable intersection by its opposite side. The triple intersection is the intersection of the circle and the Z plane. The following Matlab code computes the triple intersection.

%Inputs:

```
%P: 3X2 matrix that contains the coordinate of the cable point
%L: 1X2 matrix that contains the cable length
%Output:
%Pos: the coordinate of the end-effector
function Pos=IntersectPt(P,L)
%Calc The Center(c_int12), Radius^2(r_int12sq), and Normal(dP12) of Sphere A intersect Sphere B
dP12=P(:,1)-P(:,2);
LdP12sq=dP12'*dP12;
%cos of the angle on the intersection 12, oppositite to side 3

cqint12_o3=(LdP12sq+L(2)^2-L(1)^2)/(2*sqrt(LdP12sq)*L(2));
rsq=L(2)^2*(1-cqint12_o3^2); %Radius Sq
c=P(:,2)+L(2)*cqint12_o3*dP12/sqrt(LdP12sq); %Center
k=dP12/sqrt(LdP12sq);
l=cross(k,[0;0;1]);
l=l/norm(l);
r=sqrt(rsq);
C=r*(l(1)*k(1)+l(2)*k(2))*k(3)+c(3);
B=r*l(1)*k(2)-r*l(2)*k(1);
sq=C/B;
cq=sqrt(1-sq^2);
vq=1-cq;
M=[[k(1)^2*vq+cq k(1)*k(2)*vq-k(3)*sq k(1)*k(3)*vq+k(2)*sq];
[k(1)*k(2)*vq+k(3)*sq k(2)^2*vq+cq k(2)*k(3)*vq-k(1)*sq];
[k(1)*k(3)*vq-k(2)*sq k(2)*k(3)*vq+k(1)*sq k(3)^2*vq+cq];];
Pos=sqrt(rsq)*M *l+c;
```

Appendix B

Getting 1st order information by symbolic solver

This sample code illustrates how to use Matlab to solve complex but close form partial derivative equations, such as the one required for the LSE calibration process. It requires the Matlab symbolic toolbox, which invokes a Maple symbolic engine internally. Let there be n homogenous transformation matrix, T_i, each requires 1 free parameters x_i. The overall transformation is:

$$T = \prod T_i$$

The first step in the setting up the solver is to define the parameter symbolically. This can be done by a Matlab command:

```
syms x1x2 x3 x4 ... xn
```

Then, the individual transformation matrix can be defined symbolically by the usual command. For instance, a rotation about the Z-axis by x1 radian is:

```
T1=   [[cos(x1) -sin(x1) 0 0];
      [sin(x1)  cos(x1) 0 0];
      [0        0       1 0];
      [0        0       0 1]];
```

The primary interest in this work is the translational sub-matrix. Therefore, it should be extracted from the total frame transformation:

```
P1=T1*T2*T3...T4
P1=P1(1:3,4)
```

The partial derivative of the total translational sub-matrix can be obtained by:

```
diP1diX1= diff(P,x1);
diP1diX2= diff(P,x2);
...
```

The inline and the char function are sometimes convenient to setup the problem. However, be aware of the maple command when using the char function. For matrix equations, filter out the first 7 (“Matrix(”) and the last (“)”) characters returned by the char function:

```
diP1diX1_txt= char(diff(P,X1));
```

```
diP1diX1= inline(diP1_diX1_txt(8:end-1));
```