

Mathematical Modeling of Cable Sag, Kinematics, Statics, and Optimization of a Cable  
Robot

A thesis presented to  
the faculty of  
the Russ College of Engineering and Technology of Ohio University

In partial fulfillment  
of the requirements for the degree  
Master of Science

Dheerendra M. Sridhar

December 2015

© 2015 Dheerendra M. Sridhar. All Rights Reserved.

This thesis titled  
Mathematical Modeling of Cable Sag, Kinematics, Statics, and Optimization of a Cable  
Robot

by  
DHEERENDRA M. SRIDHAR

has been approved for  
the Department of Mechanical Engineering  
and the Russ College of Engineering and Technology by

Robert L. Williams II  
Professor of Mechanical Engineering

Dennis Irwin  
Dean, Russ College of Engineering and Technology

**ABSTRACT**

SRIDHAR, DHEERENDRA M., M.S., December 2015, Mechanical Engineering

Mathematical Modeling of Cable Sag, Kinematics, Statics, and Optimization of a Cable Robot

Director of Thesis: Robert L. Williams II

Cable sag can have significant effects on the cable length computation in a cable robot and this is more pronounced in large scale cable robots, such as the Algae Harvesting Cable Robot System. This requires modeling the cable as a catenary instead of an approximated straight line model. Furthermore, when there is actuation redundancy involved, the modeling and simulation of the system becomes much more complex, requiring optimizing routines to solve the problem.

The cable sag compensated or the catenary model was used for the Algae Harvesting Cable Robot System and simulated to solve the Kinematics and Statics problems. This involved optimization of cable tensions and finding the errors involved in the cable length. A relative comparative analysis between the straight line and cable sag model is presented. Finally based on the qualitative and quantitative results obtained, recommendations were made on the choice of model and solution methodologies.

## ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Robert L. Williams II (Dr. Bob), for all the support and encouragement he has given me throughout my graduate studies. His constant effort towards one's self-improvement is an inspiration. I would like to thank my committee members, Dr. Hajrudin Pasic, Dr. Greg Kremer, and Dr. Vardges Melkonian for their extended support, valuable suggestions, critique, and candor. My sincere thanks also goes to Mr. Jesus Pagan for his immense help and guidance. I would like to acknowledge the financial assistance provided by the department of Mechanical Engineering (ME) and Engineering Technology and Management (ETM) for my graduate studies at Ohio University. Lastly, I would like to thank my family and friends for their help and support.

## TABLE OF CONTENTS

	Page
Abstract.....	3
Acknowledgments.....	4
List of Tables .....	8
List of Figures.....	9
Chapter 1: Introduction.....	11
Cable Robots.....	12
Kinematics of Cable Robots .....	13
Statics and Pseudostatics.....	15
Redundancy and Optimization.....	15
Cable Sag .....	16
Project Information and Motivation .....	19
Thesis Objectives.....	20
Objective 1 – Kinematics and Statics .....	21
Objective 2 – Optimization.....	21
Objective 3 – Computational Consideration.....	22
Objective 4 – Cable Suggestions .....	22
Chapter 2: Methods.....	23

	6
Elastic Catenary .....	23
Inverse Pose Kinematics and Statics .....	24
Forward Pose Kinematics and Statics.....	26
Inverse Problem of the Algae Harvesting Cable Robot.....	26
Step 1 - Computation of Initial Value.....	28
Step 2 – Cable Tension Optimization .....	29
Step 3 – Cable Length Computation.....	31
Forward Problem of the Algae Harvesting Cable Robot.....	32
Step 1 – Computation of Initial Values.....	32
Step 2 – Calculation of Pose .....	32
Chapter 3: Results and Discussion.....	34
Snapshot Example.....	35
Trajectory Example.....	40
Variation of Parameters .....	47
Effects of Cable Parameters and End-Effector Mass.....	47
Effects of Algae Pond Dimensions .....	52
Computational Considerations.....	54
Cable Suggestions.....	57
Chapter 4: Conclusion and Recommendation.....	60

References..... 63

Appendix: Matlab Code..... 69

**LIST OF TABLES**

	Page
Table 1: Input variables .....	28
Table 2: Values of the variables used for simulation.....	34
Table 3: Cartesian coordinates of snapshot points.....	36
Table 4: Circular check for snapshot points .....	37
Table 5: Comparison between linear programming and pseudoinverse method.....	47
Table 6: Comparison between straight line and cable sag model.....	56
Table 7: Types of cables and their properties .....	58
Table 8: Snapshot examples for different cable materials .....	58
Table 9: Cable OEM details.....	58

## LIST OF FIGURES

	Page
Figure 1: Serial and parallel robot .....	11
Figure 2: Skycam [6], contour crafting robot [8], and tendon driven robot [7].....	12
Figure 3: End-effector attachment point and cable drawing point .....	13
Figure 4: Circular check.....	14
Figure 5: CAD model of the algae harvesting cable robot system .....	19
Figure 6: Cable suspended between two points.....	23
Figure 7: Static equilibrium of a redundant cable robot .....	25
Figure 8: Schematic of the algae harvesting cable robot [9] .....	27
Figure 9: Steps involved in the solution of inverse problem .....	28
Figure 10: Steps involved in the solution for forward problem.....	32
Figure 11: Snapshot points.....	35
Figure 12: Difference in cable lengths vs position .....	38
Figure 13: Percentage error in cable lengths vs position .....	38
Figure 14: Cable tensions vs position.....	39
Figure 15: Harvesting trajectory example.....	40
Figure 16: Cartesian coordinates vs steps.....	41
Figure 17: Cable lengths vs steps.....	42
Figure 18: Difference in cable lengths vs steps .....	43
Figure 19: Percentage error in cable lengths vs steps.....	43
Figure 20: Cable tensions vs steps .....	44

	10
Figure 21: Difference in cable tensions vs steps.....	45
Figure 22: Difference in cable length vs cable diameter for nominal position.....	48
Figure 23: Difference in cable length vs cable diameter for arbitrary position .....	48
Figure 24: Difference in cable length vs density for nominal position.....	49
Figure 25: Difference in cable length vs density for arbitrary position .....	50
Figure 26: Difference in cable length vs end-effector mass for nominal position.....	51
Figure 27: Difference in cable length vs end-effector mass for arbitrary position.....	51
Figure 28: Difference in cable length vs area of the algae pond .....	52
Figure 29: Difference in length vs (PL/PW).....	53
Figure 30: Number of iterations vs position .....	55
Figure 31: Number of iterations vs steps .....	56

## CHAPTER 1: INTRODUCTION

Robotics is the study of devices with multiple degrees of freedom (dof) which are programmable to achieve various tasks [1,2]. It is a highly interdisciplinary area, spanning decades of research that has led to extensive applications, encompassing various fields, even outside of science and engineering. Robots can be classified in many ways. One distinction used to classify is based on the manner by which the links of a manipulator are arranged. Based on this distinction, robots can be classified as serial and parallel robots. In serial robots, links are arranged in a serial manner from the base to the end effector, and in parallel robots, the links are arranged in a parallel manner from the base to the end effector [2]. Figure 1 shows an example each for serial and parallel robot [2]. One particular subclass of parallel robot is the Cable Robot.



Figure 1: Serial and parallel robot

## Cable Robots

Cable robots are robotic manipulators in which the rigid links of conventional robots are replaced by flexible cables. These robots are also referred to as cable suspended robot, wire driven robot, cable driven parallel robot, tendon driven robot, and flexible link parallel robot in the literature [3–12]. In this class of robots, the rigid links are replaced by flexible cables. By varying the lengths of the cables, whilst maintaining tension, the desired end effector pose (position and orientation) can be achieved. Cable robots have high load bearing capacities, lower mass, and unlike other parallel robots, can have very large reachable workspaces. Extensive research has been done in the field of cable robotics, which has led to various prominent applications in the area of manufacturing, assembly, communication, navigation, and haptics [11]. A few examples of cable robots are shown in Figure 2.

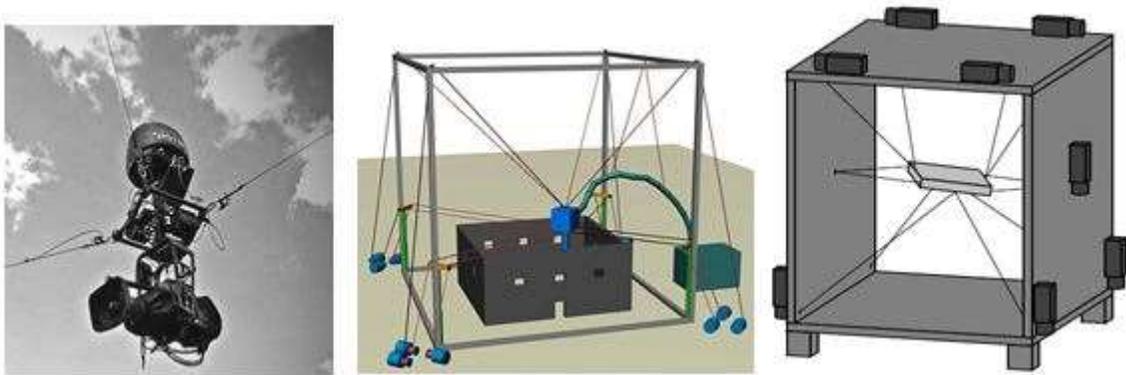


Figure 2: Skycam [6], contour crafting robot [8], and tendon driven robot [7]

### ***Kinematics of Cable Robots***

The basic principle of operation of a cable robot consists of an end-effector controlled by winches or motors via cables, which help in changing the cable lengths and maintain tension. The necessary condition for the operation is that the cables have to always maintain positive tensions, as cables can only pull and not push.

The first problem that has to be addressed when manipulating a cable robot is the Inverse Pose Kinematics (IPK). IPK problem is solving for the active cable lengths when the desired end-effector pose is known. The converse problem is the Forward Pose Kinematics (FPK), which is finding the end effector pose, when the active cable lengths are known.

The IPK when compared to the FPK problem is easier to solve owing to a strong engineering assumption, which is to consider the cables to be massless and in tension. Because of this all cables are straight and the IPK problem is reduced to finding the Euclidean norm between the cable drawing point and the end-effector attachment point after transforming these two points to the same local (or global) frame (Figure 3).

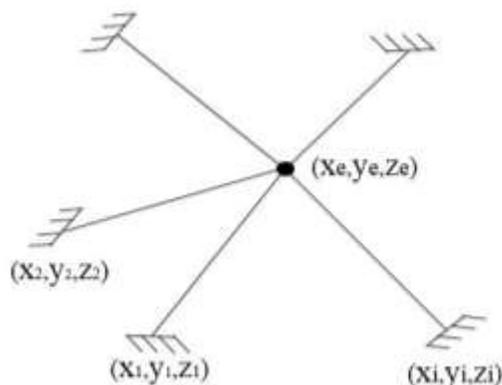


Figure 3: End-effector attachment point and cable drawing point

$$L_{ei} = \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2 + (z_i - z_e)^2} \quad (1.1)$$

Where,

$L_{ei}$  – Euclidean norm cable length,  $(x_i, y_i, z_i)$  – Cable drawing point,  $(x_e, y_e, z_e)$  – End-effector attachment point

The FPK problem on the other hand is relatively hard to solve. The three spheres intersection method discussed by Williams in [4] is an efficient analytical (and graphical) method to solve this problem. Solving the IPK problem is a prime requirement to control and manipulate a cable robot, whereas the FPK problem is useful for simulation and verification purposes.

Solving the IPK and FPK problems together constitute a verification process called the circular check. This is done by taking the output of the IPK problem and inputting it to the FPK problem and verifying if the output of the FPK is same as the input of the IPK problem [2] (Figure 4).

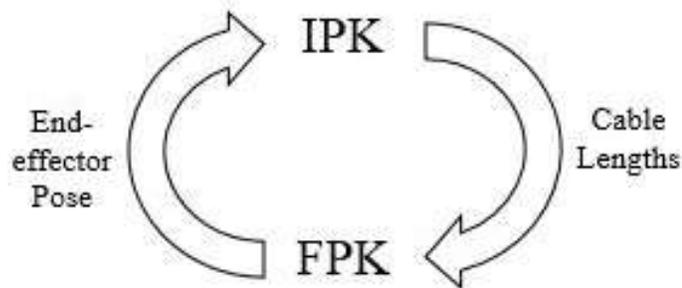


Figure 4: Circular check

### ***Statics and Pseudostatics***

The statics problem of cable robots deals with finding the cable tensions under equilibrium condition of the system. Pseudostatics is using static equilibrium conditions for systems where the velocity and acceleration is small enough to be ignored [5].

The Inverse Pseudostatics problem is solving for the active cable tensions, when the end-effector pose, external force, and end-effector mass is given and the need to solve this problem is twofold. First, to ensure there is a positive cable tension distribution and the second reason is to ensure that a particular set of cable tension is valid and achievable for a particular configuration (or design specification) [5,13]. This problem is much more complex and there is no general solution unlike the kinematics problem. However, the basic idea is to equate the vector sum of cable tensions, external forces, end-effector weight, and external forces to zero, thus yielding the cable tensions.

### ***Redundancy and Optimization***

Based on the number of cables ( $m$ ) and the degrees of freedom ( $n$ ), cable robots can be classified into three categories; underconstrained ( $m < n$ ), perfectly constrained ( $m = n$ ), and overconstrained ( $m > n$ ). The overconstrained cable robots have an actuation redundancy which provides larger workspace and flexibility in control. However, in the overconstrained case, the number of variables (typically cable tensions) outnumber the number of constraints (static equilibrium conditions), thus having infinite valid solutions. The physical interpretation of this scenario is that for a particular pose there can be infinite combination of cable pulling forces and maintain static equilibrium conditions.

The obvious approach to this problem is to optimize, i.e. to choose the best solution amongst various possible solutions such that a desired response or effect is achieved by using mathematical optimization techniques. This problem is of high importance and highly researched in the field of cable robotics. Relevant methods include the use of Linear Programming, Pseudoinverse (especially Moore–Penrose inverse), Quadratic Programming, and Nonlinear Optimization routines [5,11,13–16]. The choice of the method for optimizing mainly depends on the nature of the robot and controller.

### ***Cable Sag***

As mentioned previously, a distinct attribute of cable robots is the possibility of achieving very large workspaces which is extremely difficult or impossible to achieve using rigid link manipulators. In the past two decades major progress has been made in the design and implementation of large scale robots throughout the world.

The Five hundred meter Aperture Spherical radio Telescope (FAST) is large scale cable robot being built in China for communication and astronomical study [17]. Another example is the Skycam [6], which is an aerial camera system that is widely used in the United States and in many places around the world. Other examples include the Project CoGiRo – Control of Giant Robots used for industrial purposes [18] and the Large Cable Mechanism (LCM) used for Radio Telescope Application [12].

The mathematical modeling, kinematics, and statics discussed earlier, are derived from first principals for an ideal case, but they show excellent correspondence to practical results for small scale cable robots. In the case of large scale robots, however, there are

significant deviations due to deviations from the ideal case assumptions. One such significant deviation is the assumption of an ideal massless cable (or straight line) model.

In large scale cable robots, the length and diameter of the cable is dimensionally larger than small scale cable robots. Additionally, material of the cable needs to be stronger. Both these requirements are necessary to sustain heavy loads and bear high cable tension. An immediate effect of this is that the cable sags and the straight line model is no longer valid. This has led to considerable amount of research in the last decade to address the effects of considering the cable mass in the modeling of cable robots.

Kozak [19] addressed the issue of cable sag by studying the effects of considering mass in the statics and stiffness analysis of the FAST robot. This research used the “elastic catenary” discussed by Irvine [20], to model the cable lengths and subsequently address the IPK problem. Kozak also provided experimental validation and showed that the equations of the elastic catenary are in good agreement with experimental results. Additionally, Russell [21] provided experimental validation of the elastic catenary model and quantified the difference between theoretical cable tensions and the corresponding experimental values. Both these studies showed that the catenary equations accurately describe the profile of a sagging cable.

This was followed by researching the accuracy and error compensation study of the 6 dof FAST robot by Yao [22] and force distribution in the cables by Li [23]. The results from these researches showed that cable sag have a considerable effect on the overall accuracy and control of the robot.

Concurrently, research on the effects of sag on the workspace and cable characteristics was performed by Riehl [24,25]. The findings, based on simulations for a 3 cable – 3 dof robot showed that the workspace and the cable tension distribution for straight line and elastic catenary (or cable sag) models differ. This research also showed that the cable tension under cable sag, unlike the cable tension for the straight line model, is not constant throughout the cable for any given pose.

Irvine [20] also presented a simplified model for cable sag based on perturbation analysis. This was used by Gouttefarde [26] to model and simulate a 6 cable – 6 dof robot. Although this model is still nonlinear and does not give an analytical solution, it is simpler compared to the elastic catenary. Also, the relationship between the components of the cable tension is linear in this model. This model was further researched by Nguyen [27] to find the range of validity of the simplified model and also the limitation of the model, which is that the straight-line model is not necessarily applicable throughout the workspace of the robot, unlike the catenary model. This model also lacks sufficient experimental validation, whereas the catenary model is time-tested and has been experimentally verified and the findings have been published.

Another noteworthy work done in these lines was by Dallej [28], which was vision based control of the cable robot. This method used cameras in 3D space to instantaneously compute inverse kinematics, thereby attempting to compensate for cable sag. But this approach is expensive and requires further research to make it viable for field operations and also to mitigate the iterative steps involved.

The literature survey done for this research led to a better understanding of the current state of research on large scale cable robots. This showed that cable sag may or may not be a major impediment on the design and implementation of a cable robot. There was no definitive methodology or technique that was able to quantify the errors involved in computing the cable length when its mass was non-negligible. However, the relevant governing equations were already developed, but this had to be applied and solved based on the specification and design of a particular cable robot. This also included selecting appropriate methods for solving and understanding the practical implications.

### **Project Information and Motivation**

The Algae Harvesting Cable Robot System is a large scale cable robot designed and developed by Dr. Robert L. Williams II, Jesus Pagan, Dr. David J. Bayless, and Noah J. Needler at Ohio University [9]. This robot is a 4 cable – 3 dof (X,Y,Z translation), mobile tower type cable robot which is used for automated algae harvesting.

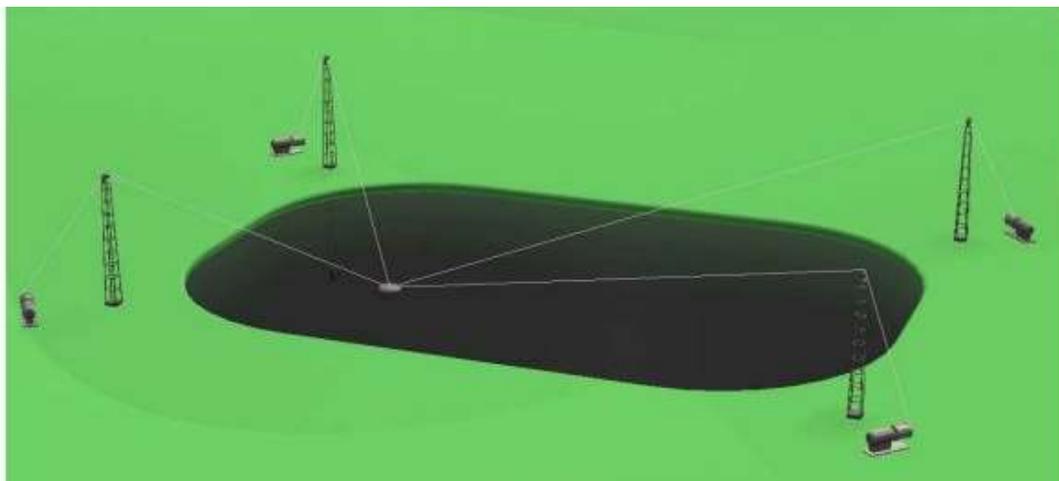


Figure 5: CAD model of the algae harvesting cable robot system

Needler in [9] presented the Kinematics, Statics, Dynamics, and Simulations of the system, assuming the straight line model. The need to study the effects of considering cable mass and quantification of cable sag induced errors, which were mentioned in the future recommendation section of the thesis [9], is the genesis of the current research.

As summarized in the earlier chapter, the only way to understand and quantify the effects of cable sag is to apply the necessary constitutive equations for the robot architecture in question and solve it by selecting appropriate methods. This will be the first step in quantifying the errors in the cable length computation. The second step, proposed for future work after this thesis, is to experimentally verify the same by prototype development and testing. This will further strengthen the validation of the results obtained from the first step.

### **Thesis Objectives**

Based on the literature survey and having an overview of the project, research in the areas of cable sag and optimization, in the context of cable robots is meaningful and relevant. Keeping this in mind, the objectives of this thesis have been formulated.

Considering the framework of this research, certain engineering assumptions are made to simplify the calculations involved. The robot is assumed to be airborne throughout the analysis and the buoyant and drag forces are neglected. All the towers / poles will be assumed to be of the same height and placed on the vertices of a rectangle. The four cables are assumed to be attached at a common point on the end-effector, which is going to be treated as a point mass. For modeling the cable sag, only the diameter of the cable (geometric property) and the cable material density (material property) will be considered.

All other properties such as flexural rigidity, cable strand stacking etc. are ignored. A pseudostatic assumption is also made, i.e. the end-effector moves slowly enough to ignore the effects of acceleration so that conditions of statics can be applied.

Owing to technological, monetary, and time constraints, the current research will be restricted to mathematical modeling and simulations without any experimental validation. Although, this is a limitation, the methods and results obtained will be non-experimentally validated using previously published (peer reviewed) research work and other non-experimental methods. The implications of non-experimental methods are understood and serve the purpose of verifying the methods and results obtained within the framework of this research.

SI units will be used throughout.

### ***Objective 1 – Kinematics and Statics***

Mathematically model the Algae Harvesting Cable Robot, considering the effects of cable sag and find the IPK and FPK solutions. The elastic catenary equations will be applied for modeling the cables of the Algae Harvesting Cable Robot and the IPK and FPK will be found. Using this, the difference in cable lengths between the straight line and cable sag model will be estimated.

### ***Objective 2 – Optimization***

Optimizing cable tension values by minimizing the sum of all cable tensions for the below mentioned cases:

- i. Straight line model
- ii. Cable sag model

By applying the techniques of mathematical optimization, the tension in each cable will be found for a given pose, such that the summation of all the cable tensions takes the lowest possible value.

***Objective 3 – Computational Consideration***

Estimating the computational cost and complexity for the mathematical models and solution procedures developed. For solving the kinematics and statics problem, the computational difficulties involved and their implications will be presented.

***Objective 4 – Cable Suggestions***

Standard cables that can be used in the Algae Harvesting Cable Robot System will be suggested. This includes specifying the type of cables and the Original Equipment Manufacturer (OEM) details.

## CHAPTER 2: METHODS

In this chapter, the methodology used to achieve the thesis objectives is presented. The methods used by Kozak in [19] and subsequently used in [24–27] will be followed in this research. The necessary equations and methods mentioned in the aforementioned works will be used in this thesis after performing the necessary coordinate system transformations.

### Elastic Catenary

The equations of the elastic catenary have been known for more than 80 years and they have been applied in various contexts of engineering. So, the derivation of these equations are not presented here (and it can be found in [19,20] for reference).

Consider a cable suspended between two points A and B (Figure 6),

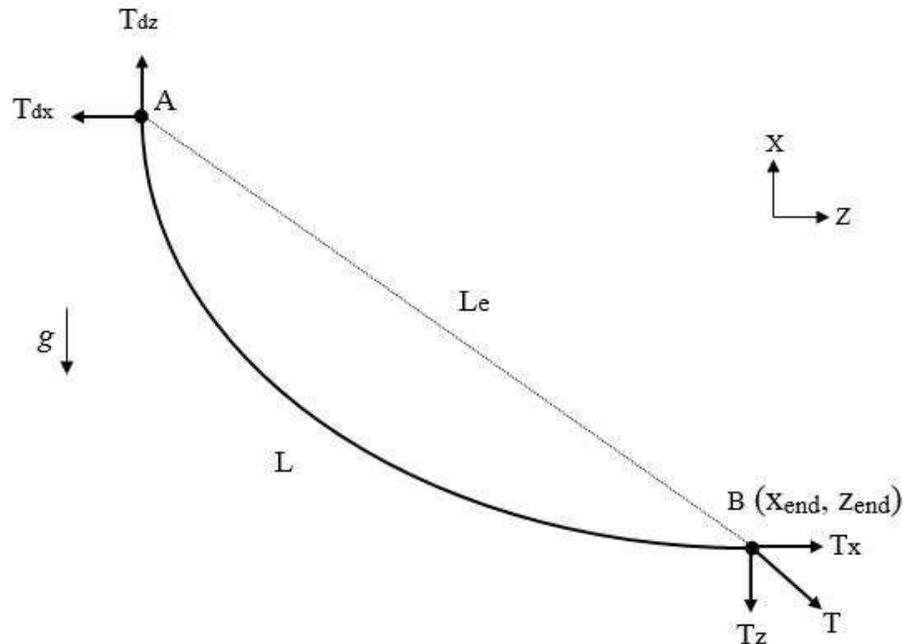


Figure 6: Cable suspended between two points

Where,

A – Cable drawing point, B – End-effector attachment point,  $L_e$  – Straight line (Euclidean norm) distance between A and B, L – Catenary or actual length between A and B,  $g$  – acceleration due to gravity, T – Tension in the cable,  $T_x$  and  $T_z$  – X and Z components of the cable tension at the end effector side,  $T_{px}$  and  $T_{pz}$  – X and Z components of the cable tension at the cable drawing point,  $(x_{end}, z_{end})$  – coordinates of the cable at the end-effector attachment point.

For this cable, the static displacement equations for the inextensible case after simplification are:

$$x_{end} = \frac{|T_x|}{\rho_L g} \left[ \sinh^{-1} \left( \frac{T_z}{T_x} \right) - \sinh^{-1} \left( \frac{T_z - \rho_L g L}{T_x} \right) \right] \quad (2.1)$$

$$z_{end} = \frac{1}{\rho_L g} \left[ \sqrt{T_x^2 + T_z^2} - \sqrt{T_x^2 + (T_z - \rho_L g L)^2} \right] \quad (2.2)$$

Where,  $\rho_L$  is the linear density of the cable material.

### **Inverse Pose Kinematics and Statics**

The IPK problem consists of finding the active cable lengths for a given pose. When considering the effects of cable sag (i.e., the mass of the cables) in modeling, cable tension is involved in finding the cable length, unlike the traditional IPK problem. Hence, kinematics and statics (or pseudostatics) problems are coupled and have to be solved simultaneously, as evident from equations 2.1 and 2.2. In other words, only kinematic equations in the straight-line model are kineto-static equations when cable sag model is considered (i.e. both kinematics and statics are linked together now). This is a system of nonlinear implicit equations, hence there are no analytical solutions, thus forcing the use of numerical methods.

As shown in [19] by Kozak and in [24] by Riehl, for a minimally or perfectly constrained case, the catenary equations (2.1 and 2.2) are solved along with the static equilibrium equations (2.3, 2.4, and 2.5) of the entire system.

$$\sum F_x = 0 \quad (2.3)$$

$$\sum F_y = 0 \quad (2.4)$$

$$\sum F_z = 0 \quad (2.5)$$

Where  $F_x$ ,  $F_y$ , and  $F_z$ , are forces in the X, Y, and Z directions respectively.

For a redundant or overconstrained case, an additional impediment is that the static problem does not have a unique solution. Since the number of variables outnumber the equations available, there are infinite valid solutions. Consider a 4-cable 3-dof (XYZ translation) cable robot as shown below (Figure 7):

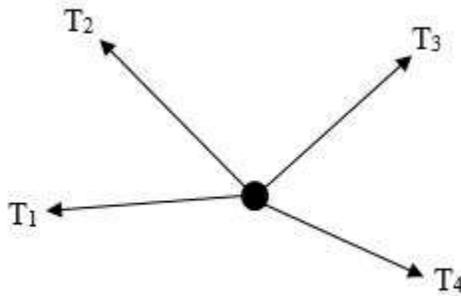


Figure 7: Static equilibrium of a redundant cable robot

Solving only the static equations, for a given valid pose, can have infinite solutions i.e. infinite combinations for  $T = \{T_1, T_2, T_3, T_4\}^T$ . The physical interpretation of this scenario is that at a given pose there are multiple valid ways of tensing the cables to

maintain static equilibrium. To get one desired solution out of the many feasible solutions, techniques of mathematical optimization are used.

There are various methods available for mathematical optimization based on the nature of the problem. One popular approach used in field of robotics, is that of the pseudoinverse (also referred as Moore–Penrose pseudoinverse) of the statics Jacobian matrix, which minimizes the Euclidean norm of the cable tensions. Another useful technique is Linear Programming, which helps to find a solution, to the above problem, provided the objective function and constraints are linear.

As Kozak points out [19], when using the catenary equations for finding the cable lengths of a redundant cable robot, one feasible approach is to solve it as constrained optimization problem or specify the  $(m-n)$  number of forces prior to solving.

### **Forward Pose Kinematics and Statics**

The FPK problem consists of finding the pose of the robot when the cable lengths are given. There are analytical methods to solve this problem such as the 3-sphere intersection algorithm presented in [4] by Williams, which is valid only for the straight line model. When cable sag is considered, FPK suffers the same hindrances that the IPK problem faces, i.e. kinematic and statics problems are coupled, highly nonlinear, and have to be solved iteratively. The methodology here involves finding components of cable tensions using cable lengths and tension and subsequently finding the pose of the robot.

### **Inverse Problem of the Algae Harvesting Cable Robot**

The schematic of the algae harvesting cable robot by Needler [9] is as shown in Figure 8. The base frame –  $\{A\}$ , is fixed to the center of the surface of the algae pond. The

end-effector is the point  $P$  with  $h_i$  being the height of the towers. Points  $B_i$  and  $P_i$  are the base and top points of the towers / poles respectively and points  $A_i$  are the points where winches / motors are located.  $L_i$  (or  $L_{ei}$  according to the notation of this research) is the Euclidean norm cable lengths. In all cases  $i = 1,2,3,4$ .

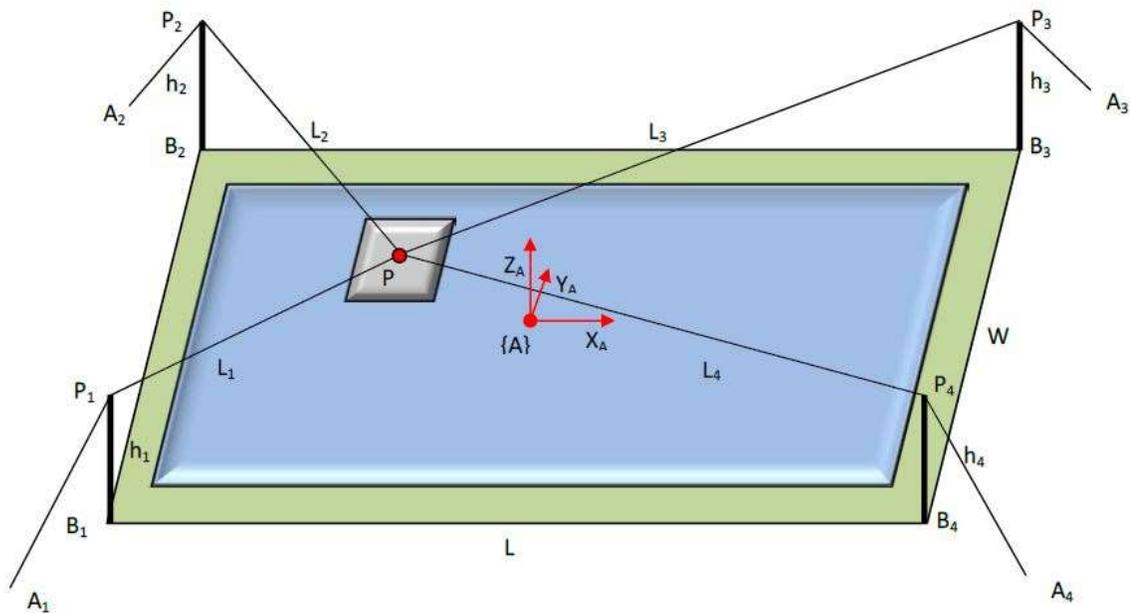


Figure 8: Schematic of the algae harvesting cable robot [9]

The methodology used to address the Inverse Pose Kinematics and Statics problem is as described in [19,22,23,27]. The same methodology is adapted here with suitable transformations made. The details of the method adapted and coded (Appendix) in MATLAB® is as described below:

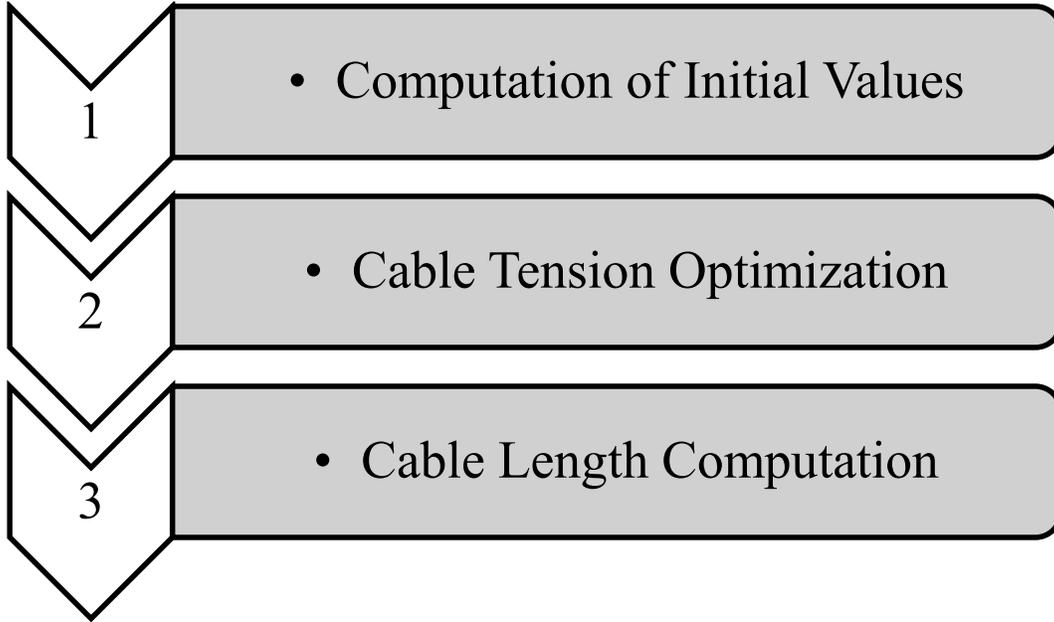


Figure 9: Steps involved in the solution of inverse problem

### *Step 1 - Computation of Initial Value*

In this step, all the required inputs for solving the IPK problem, along with necessary parameters such dimensional details of the algae pond, robot variables, and properties of the cable etc. are entered. Then necessary coordinate transformations are made, which includes transforming global coordinates to local cable coordinates and vice versa. Subsequently, the Euclidean norm lengths of the cable and statics Jacobian matrix are calculated. The following table shows the input variables required:

Table 1: Input variables

<b>Input Variable</b>	<b>Symbol</b>	<b>Unit</b>
Pond Length	PL	m
Pond Width	PW	m

Pole Height	Ph	m
Pond offset	$\Delta X$ and $\Delta Y$	m
End-effector mass	m	kg
Cable Diameter	d	mm
Density of the Cable	$\rho$	$\text{kg/m}^3$
End-effector location	(x,y,z)	m

The Euclidean norm length of the cable is calculated using,

$$L_{ei} = \sqrt{(P_{ix} - x)^2 + (P_{iy} - y)^2 + (P_{iz} - z)^2} \quad (2.6)$$

Where,

$${}^A P_1 = \{-PL/2 - \Delta X, -PW/2 - \Delta Y, Ph\}^T$$

$${}^A P_2 = \{-PL/2 - \Delta X, PW/2 + \Delta Y, Ph\}^T$$

$${}^A P_3 = \{PL/2 + \Delta X, PW/2 + \Delta Y, Ph\}^T$$

$${}^A P_4 = \{PL/2 + \Delta X, -PW/2 - \Delta Y, Ph\}^T$$

The static Jacobian matrix  $[{}^A A]$  is given by,

$$[{}^A A] = \begin{bmatrix} \left(\frac{P_{1x}-x}{L_{e1}}\right) & \left(\frac{P_{2x}-x}{L_{e2}}\right) & \left(\frac{P_{3x}-x}{L_{e3}}\right) & \left(\frac{P_{4x}-x}{L_{e4}}\right) \\ \left(\frac{P_{1y}-y}{L_{e1}}\right) & \left(\frac{P_{2y}-y}{L_{e2}}\right) & \left(\frac{P_{3y}-y}{L_{e3}}\right) & \left(\frac{P_{4y}-y}{L_{e4}}\right) \\ \left(\frac{P_{1z}-z}{L_{e1}}\right) & \left(\frac{P_{2z}-z}{L_{e2}}\right) & \left(\frac{P_{3z}-z}{L_{e3}}\right) & \left(\frac{P_{4z}-z}{L_{e4}}\right) \end{bmatrix} \quad (2.7)$$

### ***Step 2 – Cable Tension Optimization***

In this step, the cable tensions for a given pose are calculated. As mentioned previously, this is a case with multiple valid solutions. To find a unique solution, this

problem is solved as a constrained minimization problem. So, the statics problem is treated as a linear programming problem with an aim of minimizing the cable tensions. The problem is formulated as shown below:

Objective function:-

$$\text{Minimize } (T_1 + T_2 + T_3 + T_4)$$

Subject to,

Constraints:-

$$[AA]\{T\} + \{AF\} + m \{Ag\} = 0 \quad (2.8)$$

$$T_{\min} \leq T \leq T_{\max}$$

Where,

$\{T\} = \{T_1 \ T_2 \ T_3 \ T_4\}^T$ ,  $\{AF\}$  – external force in the A frame,  $T_{\min}$  and  $T_{\max}$  – minimum and maximum allowable cable tensions.

This problem is a standard linear programming problem in four variables, with the static equilibrium equations used as constraints and bounds on the cable tensions based on necessary conditions ( $T > 0$ ). Bounds not only help in obtaining non-negative solutions, (negative solutions for cable tensions means cable is pushing, which is an unacceptable solution), but also restrict the solution to be within practical limitations, such as extremely high cable tensions, which might break the cable or cannot be supported by the winch / motor. This problem is solved using the linear programming solver called `linprog ( )` in MATLAB® [29].

Additionally, the pseudoinverse method was also implemented using the ‘`pinv( )`’ [29] command in MATLAB® to bring about a relative comparison. The cable tensions

obtained using this method are denoted by  $T_{pi}$ . Thus, at the end of this step, four active cable tensions;  $\{T\} = \{T_1 T_2 T_3 T_4\}^T$  are obtained.

### ***Step 3 – Cable Length Computation***

In this final step, cable lengths are computed using the catenary equations, by numerically solving a system of equations. This system of equations is shown below:

$$x_{iend} = \frac{|T_{xi}|}{\rho_L g} \left[ \sinh^{-1} \left( \frac{T_{zi}}{T_{xi}} \right) - \sinh^{-1} \left( \frac{T_{zi} - \rho_L g L_i}{T_{xi}} \right) \right] \quad (2.9)$$

$$z_{iend} = \frac{1}{\rho_L g} \left[ \sqrt{T_{xi}^2 + T_{zi}^2} + \sqrt{T_x^2 + (T_{zi} - \rho_L g L_i)^2} \right] \quad (2.10)$$

$$T_i = \sqrt{T_{xi}^2 + T_{zi}^2} \quad (2.11)$$

Where  $i = 1,2,3,4$ .

For each cable this a system of three equations with three variables ( $T_x$ ,  $T_y$ , and  $L$ ). To solve this system of equation the `fsolve` [29] command in MATLAB® is used, which is an iterative solver used to solve a system of nonlinear equations with real variables. In view of achieving Objective – 3 (Computational Considerations), the number of iterations is tracked. Finally, this solver returns the components of the cable tensions along with the cable lengths.

To summarize, the methodology consists of finding the initial variables and subsequent coordinate transformation. Followed by this, an optimization routine is performed to get a valid set of cable tensions  $\{T\}$ , such that the sum of cable tensions ( $\Sigma T$ ) is minimized. Finally, these cable tensions are used in the catenary equations to obtain the cable lengths. The code combines all the three steps to solve the Inverse Pose Kinematics and Statics Problem comprehensively, such that when the user enters a valid pose, the program returns the cable tensions and lengths.

## Forward Problem of the Algae Harvesting Cable Robot

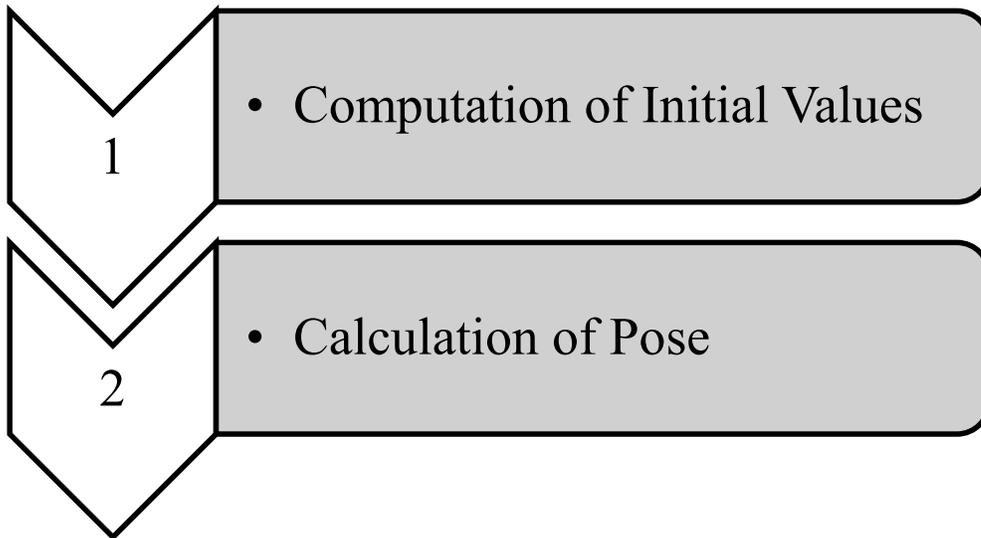


Figure 10: Steps involved in the solution for forward problem

### ***Step 1 – Computation of Initial Values***

Similar to IPK problem, in this step all the necessary input values and coordinate transformations are entered. The active cable lengths and their respective tensions, dimensional details of the algae pond, and the geometrical and material properties of the cables are entered.

### ***Step 2 – Calculation of Pose***

In this step, the static displacement equations of the catenary (2.9 – 2.11) along with the static equilibrium equations are solved numerically along with necessary transformations of coordinate system when required. This system of equations is solved (similar to its inverse counterpart) using the `fsolve` command in MATLAB® [29] and its solution yields the pose of the robot.

In summary, the method consists of finding the initial values and necessary transformations. This is followed by solving a system of nonlinear equations whose solution gives the pose. A major difference in this problem, when compared to the inverse problem, is the absence of optimization step, thus making it considerably faster to solve. However, both problems must be solved numerically (i.e. iteratively), when the effect of cable sag has to be considered.

### CHAPTER 3: RESULTS AND DISCUSSION

Using the code written, based on the methods described in the previous chapter, simulations were performed. This included simulating snapshot examples, a trajectory, and parameter variations. The results obtained and their interpretations are discussed in this chapter.

Although the created program works for any valid dimensions and cable properties, the simulation results presented here use the following values for the variables:

Table 2: Values of the variables used for simulation

Variable	Value	Notes
Pond Length (PL)	50 m	1 acre pond
Pond Width (PW)	80.9 m	
Pole Height (Ph)	7.6 m	All poles are of same height
Pond offset ( $\Delta X$ and $\Delta Y$ )	6.1 m	-
End-effector mass (m)	258.6 kg	Mass of platform, algae, and water collected
Cable Diameter (d)	20 mm	-
Density of the Cable ( $\rho$ )	7860 kg/m <sup>3</sup>	Density of a steel cable
External Force ( $^A F$ )	0	-
Tension Lower Limit ( $T_{\min}$ )	2536.866	Weight of the end-effector
Tension Higher Limit ( $T_{\max}$ )	$+\infty$	To find the maximum force that might be required

### Snapshot Example

Both the Inverse and Forward Problems were solved for five random poses including a nominal position  $(0, 0, 0)$ . The five poses are graphically shown in Figure 11.

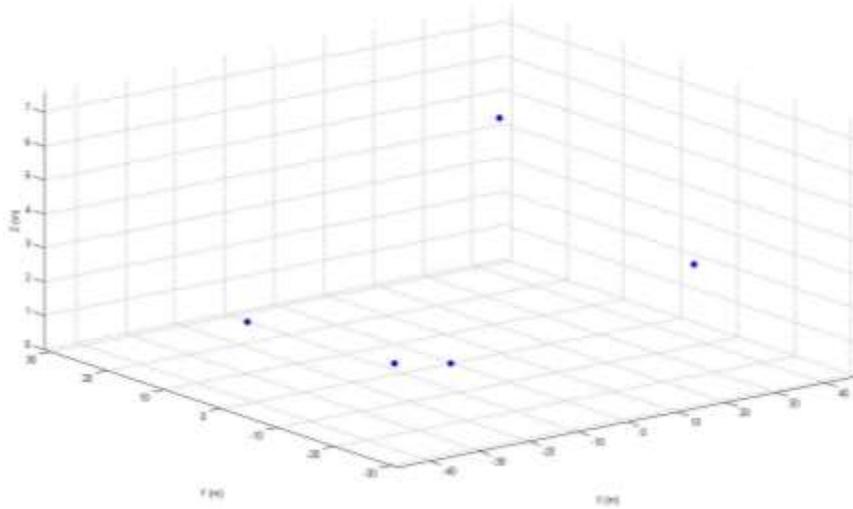


Figure 11: Snapshot points

When the code for the inverse problem is executed with these snapshot points as inputs, the program outputs cable lengths and tensions.

Table 3: Cartesian coordinates of snapshot points

Point No.	End-effector position (m)		
	X	Y	Z
1	0	0	0
2	-29.4	10.2	1.5
3	-33	-18.8	2
4	28.5	-18	3.1
5	35	22	5

First, the circular check is performed to verify and partly validate the results obtained. To serve this purpose, both the inverse and forward problems were solved for all the five snapshot points. The results are summarized in Table 4 and the circular check is verified (highlighted columns have equal corresponding values).

Table 4: Circular check for snapshot points

Point No.	<u>INVERSE PROBLEM</u>					<u>FORWARD PROBLEM</u>				
	Input	Output				Input				Output
	Pose	L <sub>1</sub>	L <sub>2</sub>	L <sub>3</sub>	L <sub>4</sub>	L <sub>1</sub>	L <sub>2</sub>	L <sub>3</sub>	L <sub>4</sub>	Pose
1	0, 0, 0	56.7	56.7	56.7	56.7	56.7	56.7	56.7	56.7	0, 0, 0
2	-29.4, 10.2, 1.5	45.25	27.72	81.38	87.87	45.25	27.72	81.38	87.87	-29.4, 10.2, 1.5
3	-33, -18.8, 2	19.14	52.46	96.41	83.22	19.14	52.46	96.41	83.22	-33, -18.8, 2
4	28.5, -18, 3.1	77.75	90.26	53.11	22.75	77.75	90.26	53.11	22.75	28.5, -18, 3.1
5	35, 22, 5	97.64	84.7	14.93	54.92	97.64	84.7	14.93	54.92	35, 22, 5

The cable length difference between the cable sag and straight model is calculated, followed by cable length error computation. Corresponding equations used to accomplish this are as follows:

$$D_i = L_i - L_{ei} \quad (3.1)$$

$$ER_i = \frac{L_i - L_{ei}}{L_i} \times 100 = \frac{D_i}{L_i} \times 100 \quad (3.2)$$

The results of difference in cable lengths and their percentage error are plotted as shown in Figure 12 and 13.

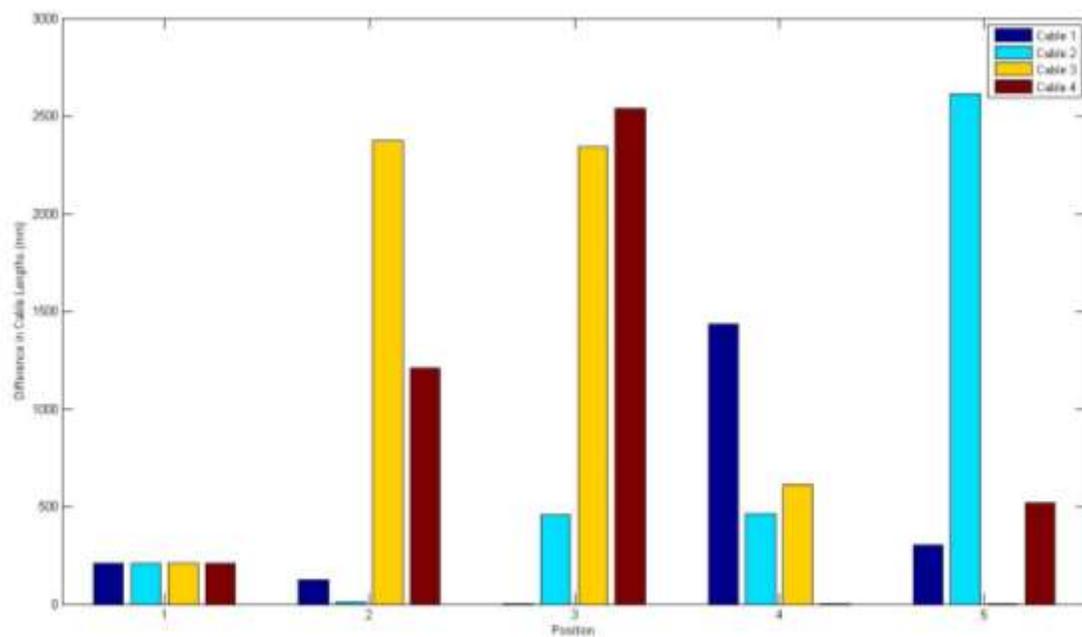


Figure 12: Difference in cable lengths vs position

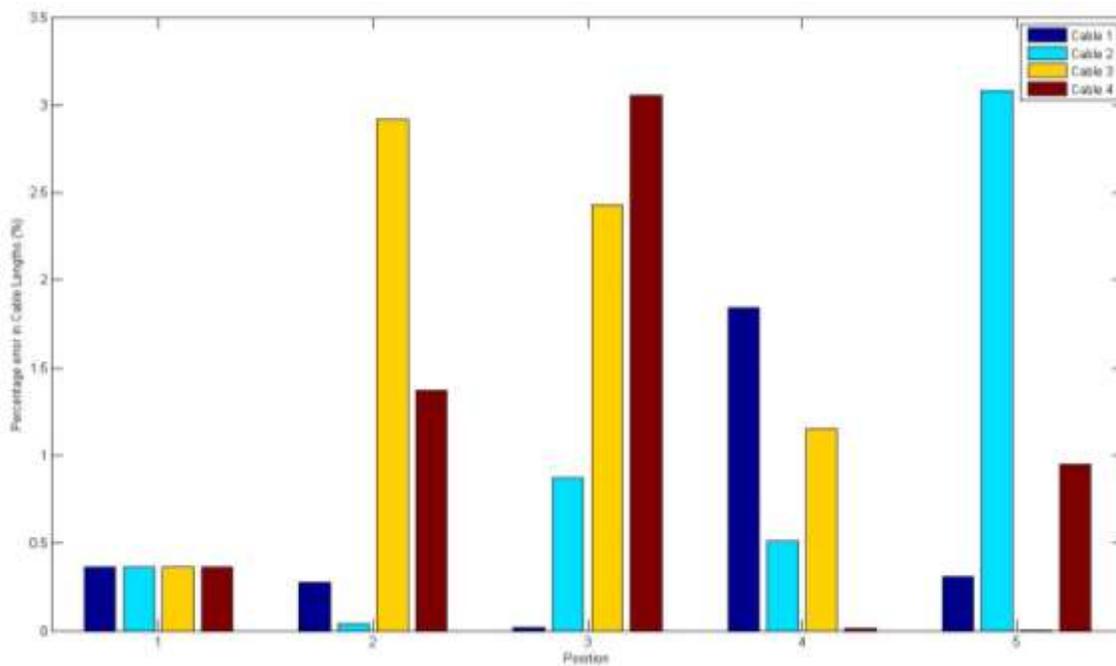


Figure 13: Percentage error in cable lengths vs position

Along with computation of cable lengths, the cable tensions were also calculated using two methods; Linear Programming (LP) and Pseudoinverse Method (PI). These two methods give different values for cable tensions as the objective functions in both cases are different. The resulting graph is shown in Figure 14.

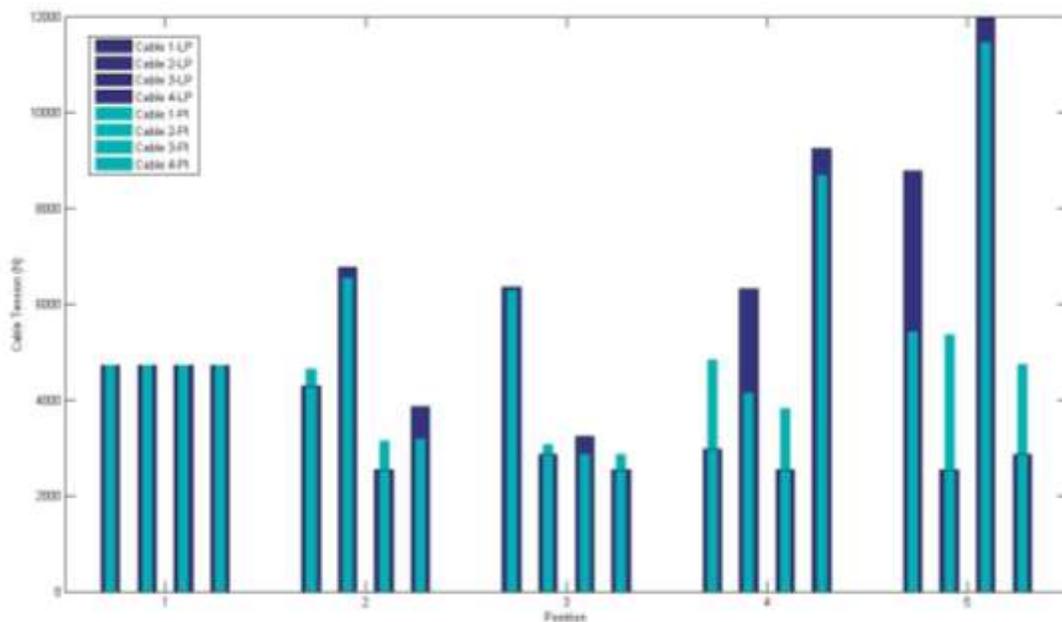


Figure 14: Cable tensions vs position

From the graphs, it can be observed that the difference in cable lengths obtained from the straight line model and cable sag model ranges from (0 – 2600) mm, which appears to be significantly higher. However, when the relative error is computed, the range is a narrow 0-3 %. The Algae Harvesting Cable Robot System, unlike the FAST [17] or LCM [12], is not meant for accurate positioning of the end-effector, hence from the snapshot examples the effects of cable sag appears to be tolerable. But the five examples

are a small sample size and are random points; this necessitates running the program to simulate a trajectory.

### Trajectory Example

The harvesting trajectory of the proposed system customarily involves a trajectory similar to a pick-and-place path. Therefore, a sample trajectory of the robot was simulated with a step size of 0.5 m as shown in Figure 15.

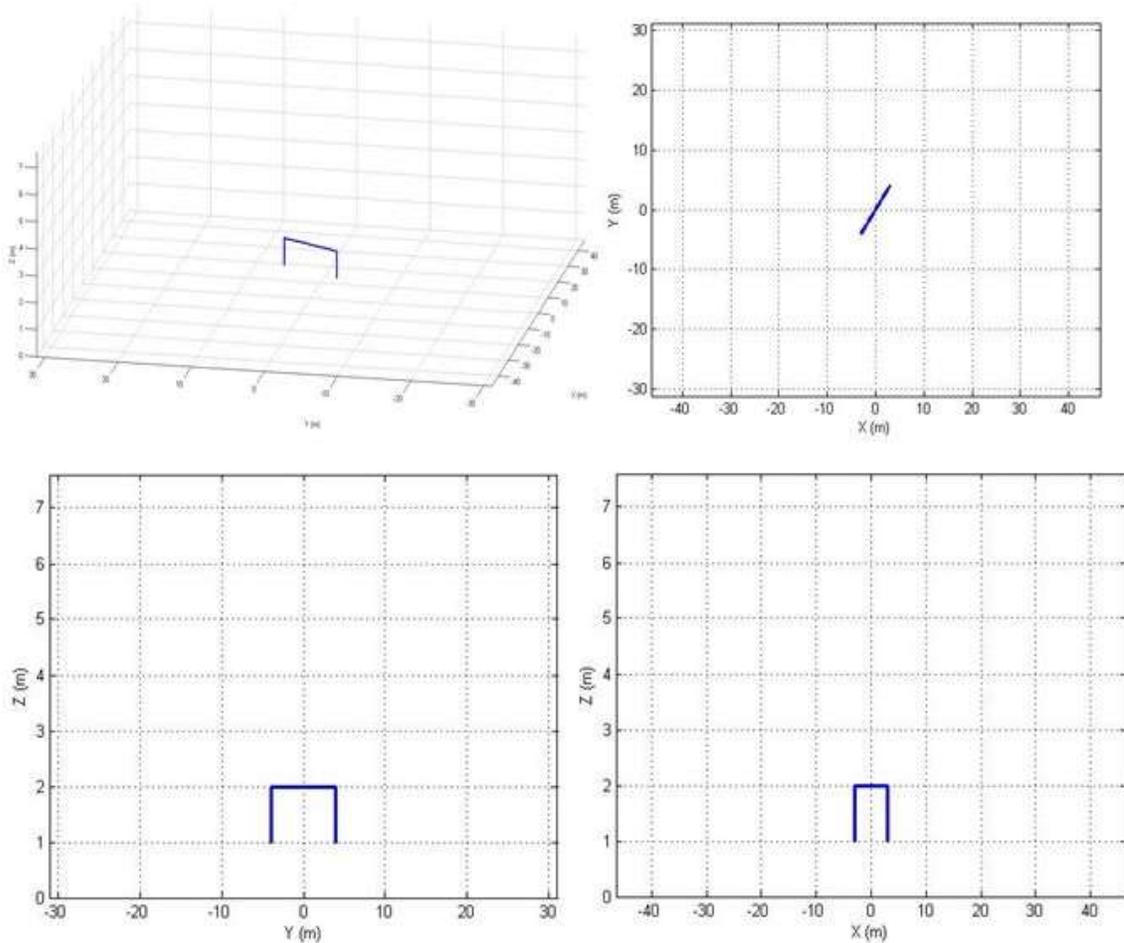


Figure 15: Harvesting trajectory example

The variation of Cartesian coordinates and cable lengths for the trajectory is shown in Figure 16 and 17.

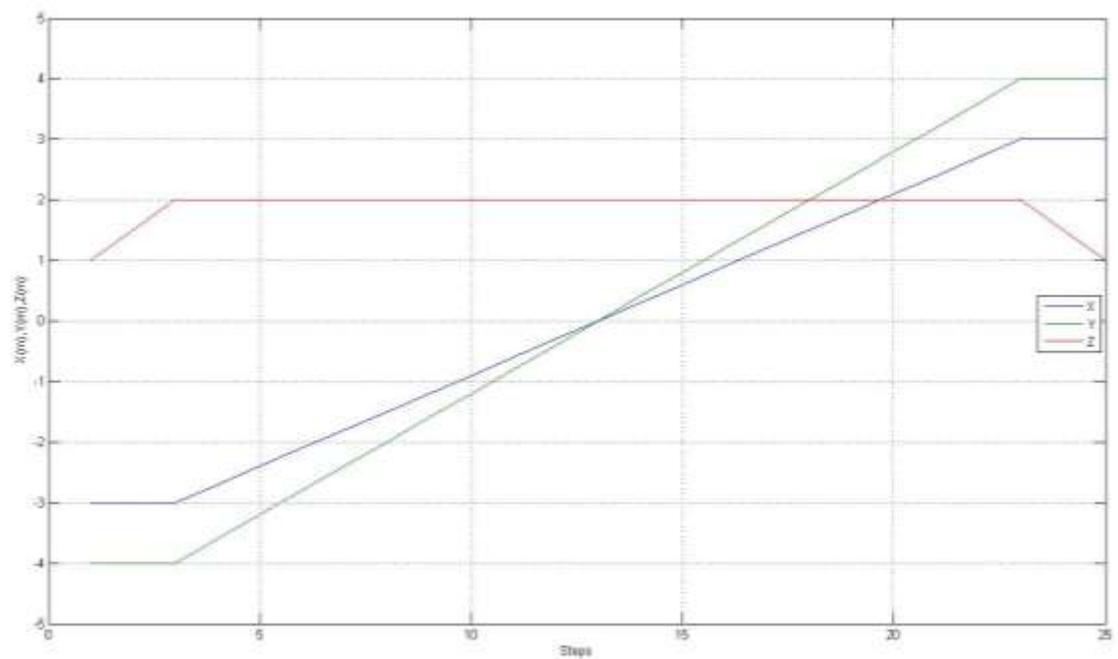


Figure 16: Cartesian coordinates vs steps

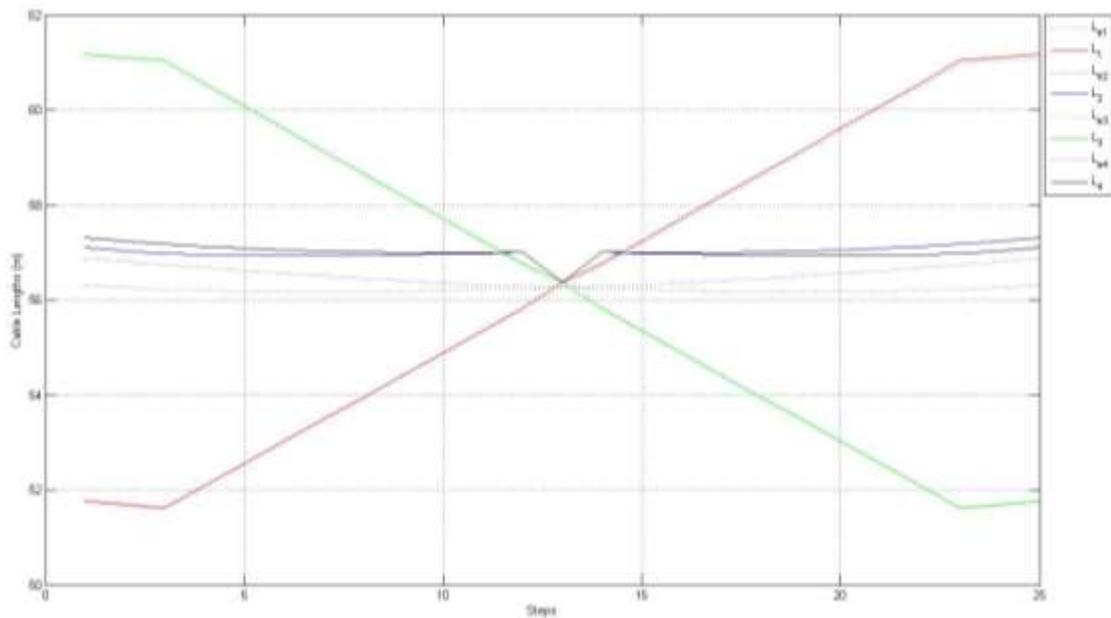


Figure 17: Cable lengths vs steps

Similar to the snapshot example, the cable length differences between the cable sag and straight models are calculated, followed by cable length error computation. This is shown in Figure 18 and 19. As observed from the graphs, the difference in cable lengths obtained from the straight line model and cable sag model ranges from (0-800) mm and the relative error ranges from (0-1.4) %. These values further indicate that, although cable sag contributes to erroneous cable length computation, the error is low enough for purposes such as the one in consideration, where high accuracy and precision is not a prime requirement.

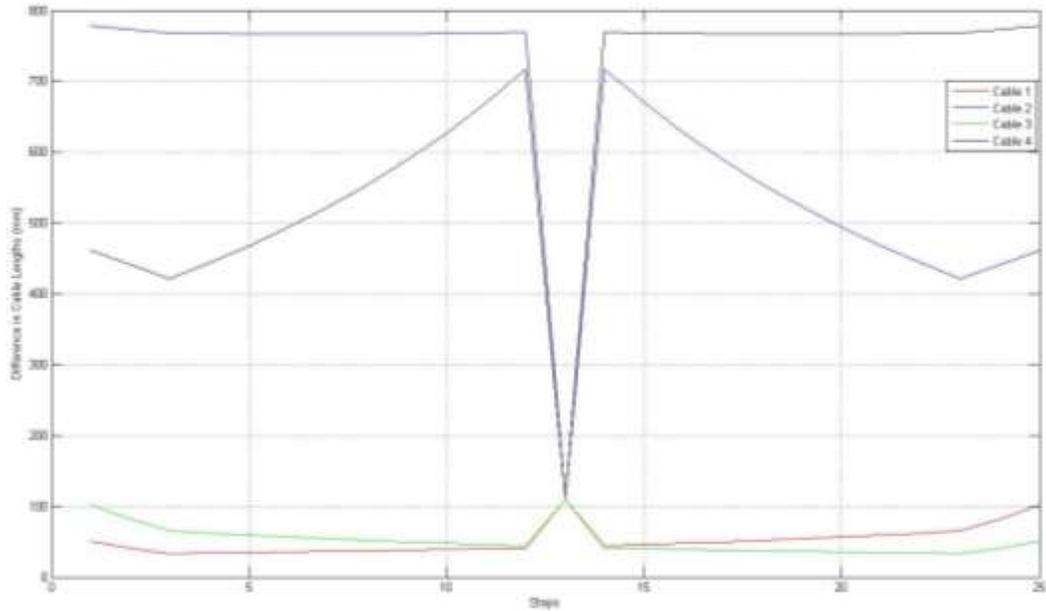


Figure 18: Difference in cable lengths vs steps

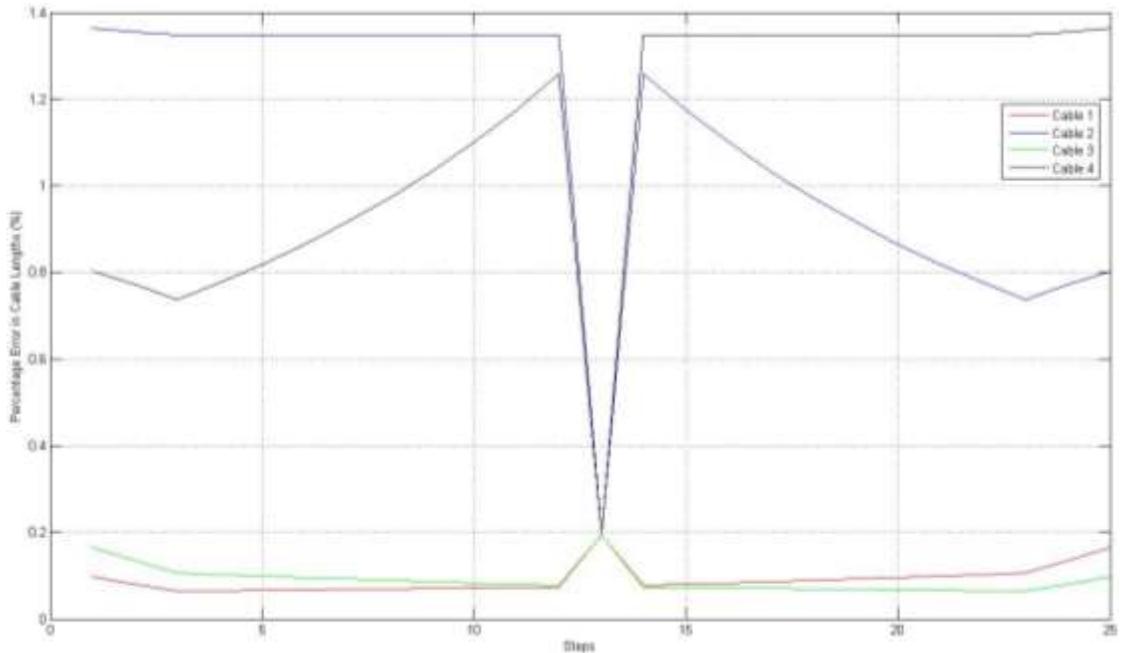


Figure 19: Percentage error in cable lengths vs steps

The cable tensions were calculated for all the steps in the trajectory by both methods. This was followed by finding the difference between the summation of cable tensions obtained from linear programming and pseudoinverse methods ( $\Sigma T_{pi} - \Sigma T_i$ ). The results are presented in Figures 20 and 21.

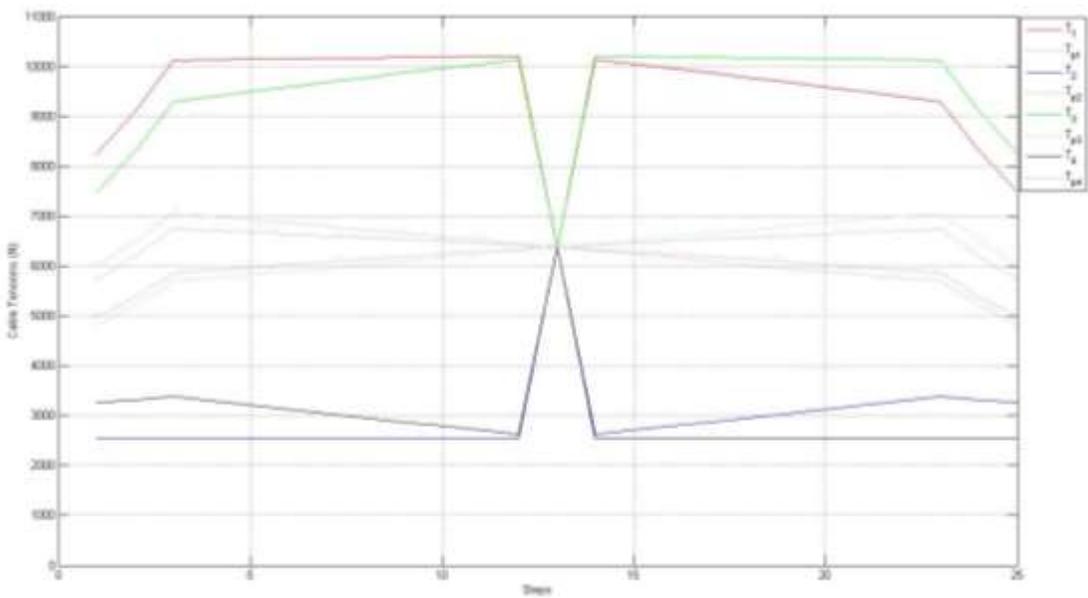


Figure 20: Cable tensions vs steps

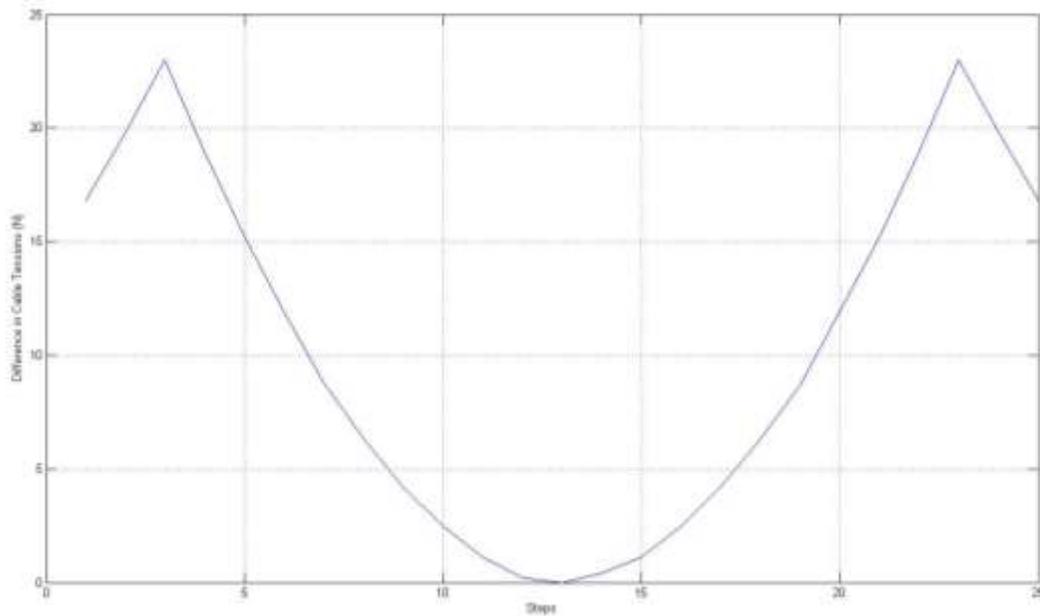


Figure 21: Difference in cable tensions vs steps

From Figure 12, 18, and 19, a straightforward observation is that the two methods give different solutions for cable tensions except when the cable lengths are equal. When all the cable lengths are equal ( $x=0, y=0$ ), both the methods yield the same solution. Except for this case, the linear programming gives a solution such that the overall cable tensions are less, when compared to the corresponding pseudoinverse solution. But the independent solutions vary significantly.

Another major advantage of using linear programming is that we can restrict the solution space by using the bounds ( $T_{\min}$  and  $T_{\max}$ ). For example, in this simulation  $T_{\min}$  was set to be equal to the weight of end-effector, which can be increased if the cable tensions are found to be insufficient to keep it taut and decreased if the cable tension is high enough to break the cable or infeasible because of winch / motor torque limitations.

A similar argument can be made for  $T_{\max}$ . In this simulation,  $T_{\max}$  was set to be  $+\infty$  to get an idea of the maximum tension that a particular configuration might reach.

The pseudoinverse method on the other hand does not give this flexibility, at least on both the ends. But a major merit of the pseudoinverse approach is that, it has a closed-form analytical, unlike the iterative linear programming method.

There are a few issues associated with the use of LP method that require attention. The LP approach at times gives an abrupt increase or decrease in the solution, thus not giving a smooth curve for trajectories as observed from Figure 18. Another issue is that the LP solution at times tends to give a solution that is the lower bounds or upper bounds ( $T_{\min}$  or  $T_{\max}$ ) for the problem. These issues can partly be alleviated by changing the bounds or choosing an alternate solution if a particular point has multiple solutions. Regardless, a valid solution can be obtained by this method and research is continuously being done in this field to get smoother results with less iterations.

Borgstrom [14] shows that linear programming can be suitably modified and, with the assistance of suitable control systems, make it more efficient and computationally less expensive. Considering all of these factors, use of linear programming for cable tension calculation is highly advisable, at least for simulation purposes, if not for field implementation. A summary of this discussion is provided in the form of a comparison chart below:

Table 5: Comparison between linear programming and pseudoinverse method

<b>Linear Programming</b>	<b>Moore Penrose Pseudoinverse</b>
In the current case minimize the sum of cable tensions (Min $\Sigma T$ )	Minimizes the second norm of the cable tensions (Min $\sqrt{\Sigma(T^2)}$ )
Can be applied for other objective functions.	Only one objective function possible
Iterative method	Closed form analytical solution possible
In this particular case, overall cable tension is relatively small	Overall cable tension when compared to LP method, at the least can be equal to LP solution
More flexible method	Less flexible method
Multiple solutions possible	Single solution
MATLAB® command – linprog ( )	MATLAB® command – pinv( )

### **Variation of Parameters**

As mentioned in the thesis objectives, the current research is validated by literature and non-experimental methods. In view of this and also to understand the effects of cable robot's input variables, further simulations were performed using different cable and algae pond parameters.

#### ***Effects of Cable Parameters and End-Effector Mass***

The input parameters of the cable are diameter (geometric property) and density (material property). For a nominal (0, 0, 0) and an arbitrary (8, -5, 2) position, these parameters were varied independently and the results are graphically presented below.

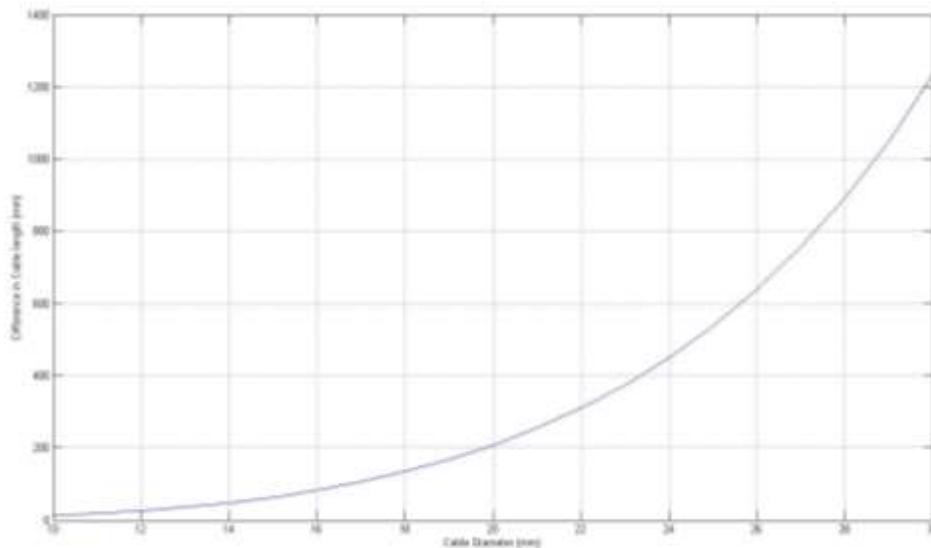


Figure 22: Difference in cable length vs cable diameter for nominal position

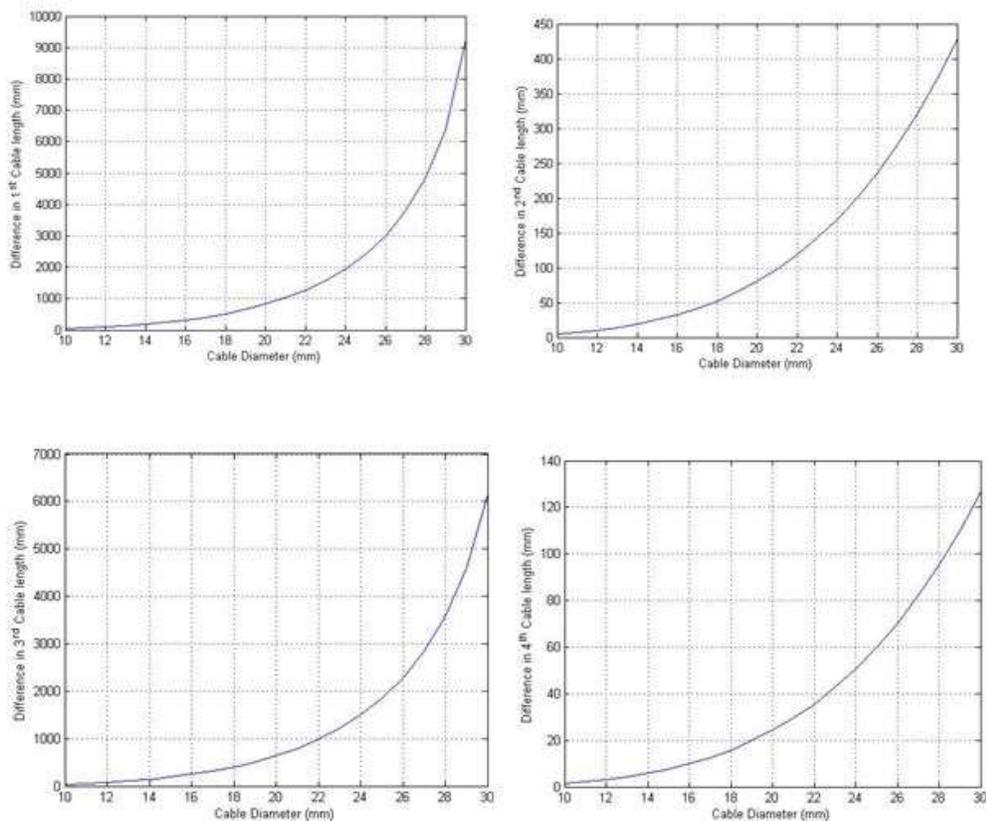


Figure 23: Difference in cable length vs cable diameter for arbitrary position

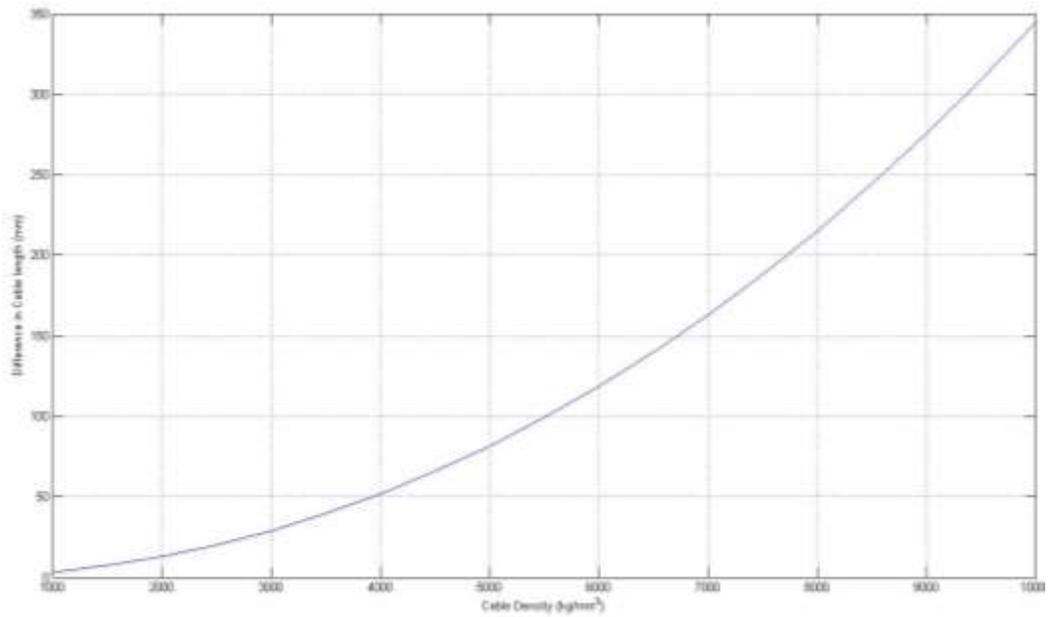


Figure 24: Difference in cable length vs density for nominal position

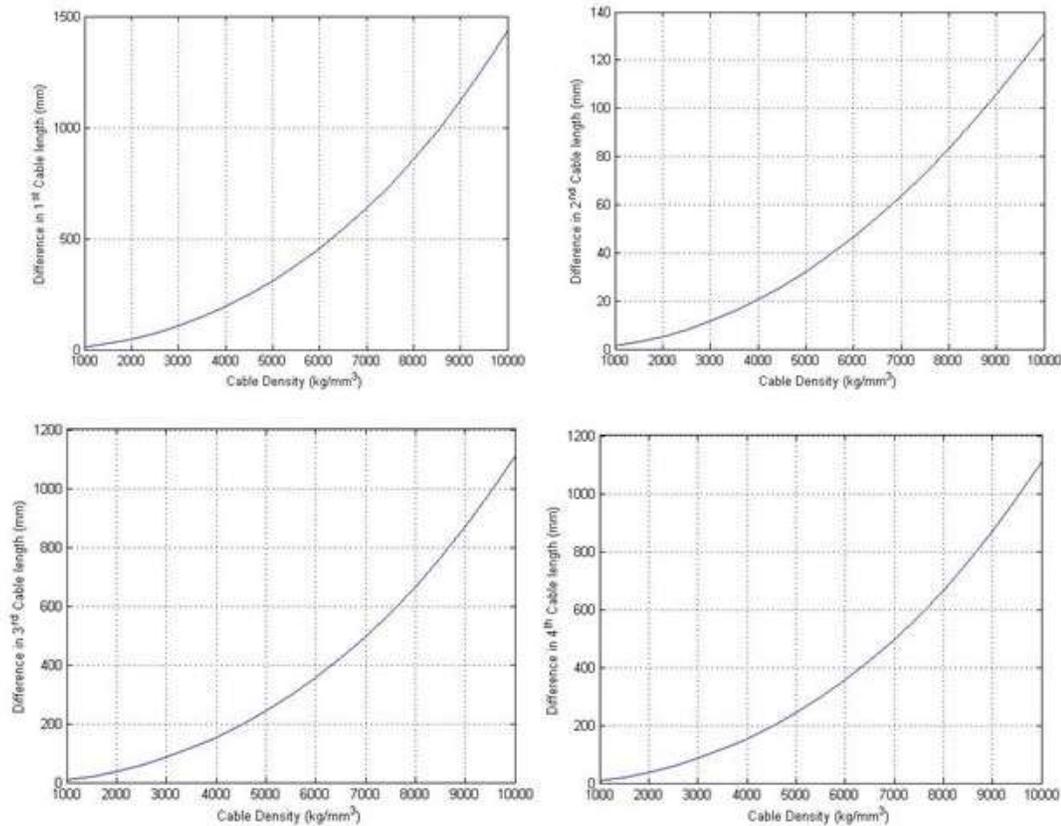


Figure 25: Difference in cable length vs density for arbitrary position

From the physical understanding of the effects of cable sag, it is evident that if the cable weight increases, then cable sag increases, which in turn increases the error or cable length difference between the cable sag and straight line models. Increasing cable diameter and / or cable material density increases cable weight. Based on the physical interpretation of cable sag and understanding the nature of the catenary equations, we expect a nonlinear increase in the difference in cable lengths when cable diameter and density is increased. This is verified by the simulations as shown in Figures 22 - 25.

Another important parameter is the end-effector mass. This is of special importance since it varies continuously during the operation of the cable robot. The variation of

difference in cable length with an increase in end-effector mass is as shown in Figure 26 and 27.

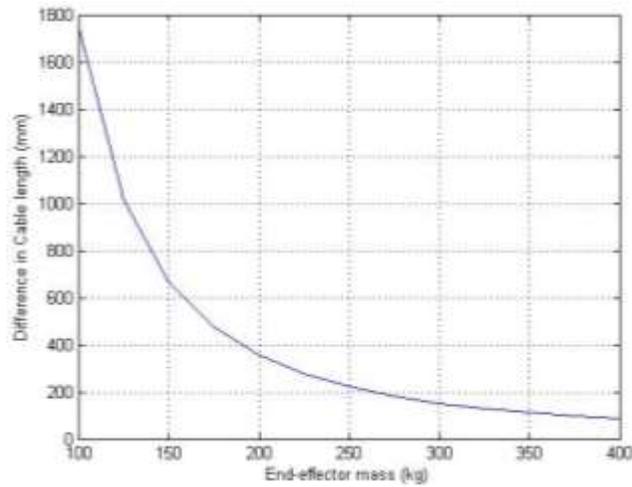


Figure 26: Difference in cable length vs end-effector mass for nominal position

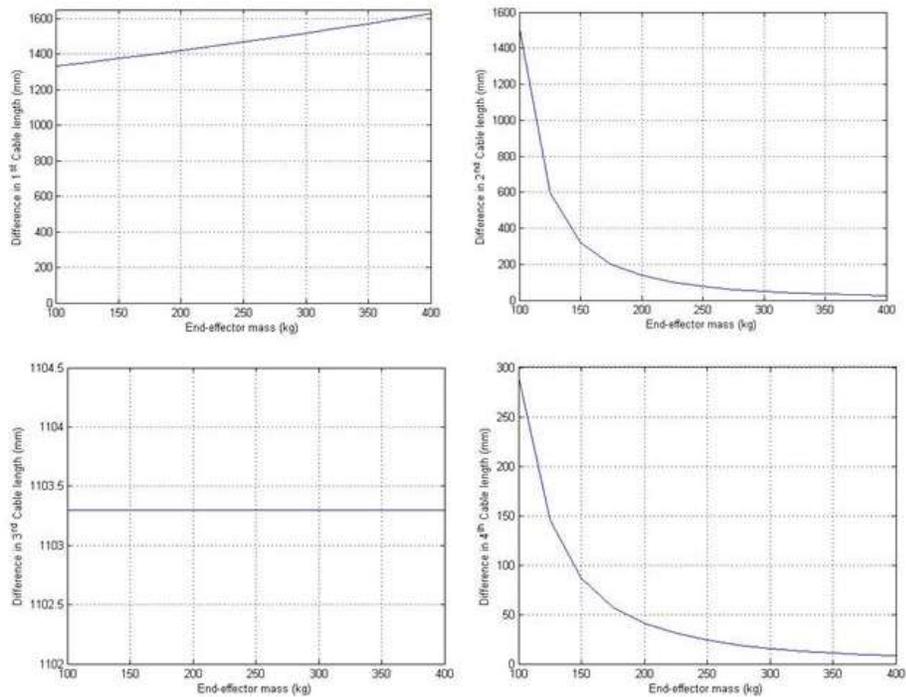


Figure 27: Difference in cable length vs end-effector mass for arbitrary position

An increase in end-effector mass has different effects on different cables as observed in Figures 26 and 27. The reason for this is one of the limitations of the LP method to yield solutions that tend to fall on the bounds. In case of the arbitrary position (Figure 27), the 3<sup>rd</sup> cable solution falls on the lower bound, hence the variation in cable length is constant. However, all solutions are still valid for both nominal and arbitrary positions.

### *Effects of Algae Pond Dimensions*

As the size of the algae pond increases, the cable length and its overall weight increases, thus increasing the cable sag and increasing the difference in cable length. Keeping the ratio of algae pond length to width constant ( $PL/PW = \text{constant}$ ), its area was increased and the difference in cable lengths was computed. As expected, the cable length difference increases with an increase in area as shown in Figure 28.

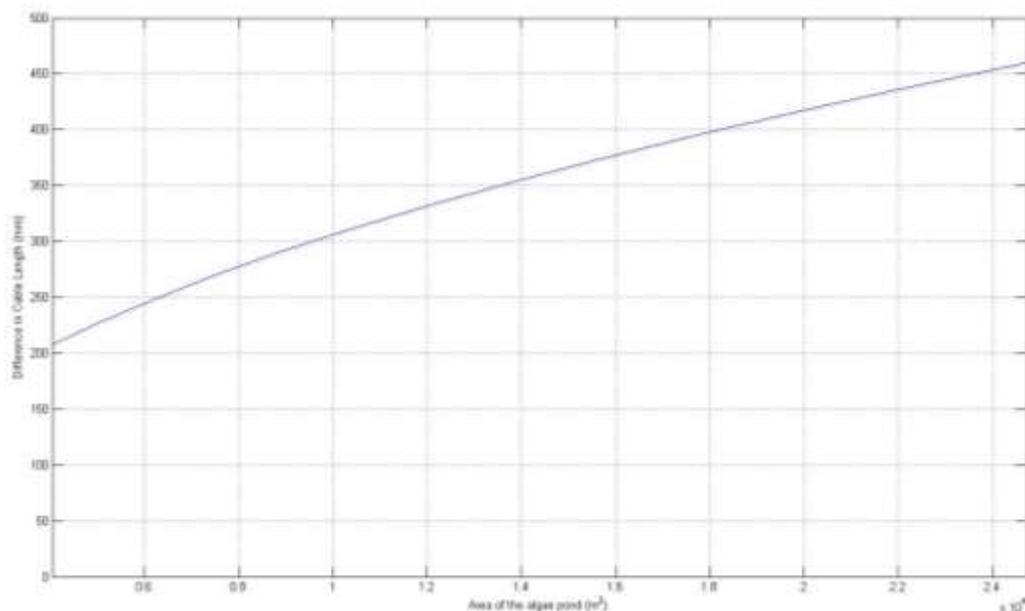


Figure 28: Difference in cable length vs area of the algae pond

Complimentary to the previous case, the study of the effects of variation of pond length to width (PL/PW) ratio, keeping the area constant seemed relevant. For the nominal position, at a constant tower height, the variation of the Euclidean norm length depends on the pond length (PL) and pond width (PW). By the Pythagorean theorem, this is dependent on the term:

$$\sqrt{(PL)^2 + (PW)^2}$$

Also, the point where the length and width interchange their values, we expect the difference in cable lengths to remain the same. All these facts are verified by simulation results as shown in Figure 29. The difference in cable length closely follows the expected trend and the extreme values have the same difference in cable lengths (as the values of the length and width interchange) thus serving as additional validation of the results.

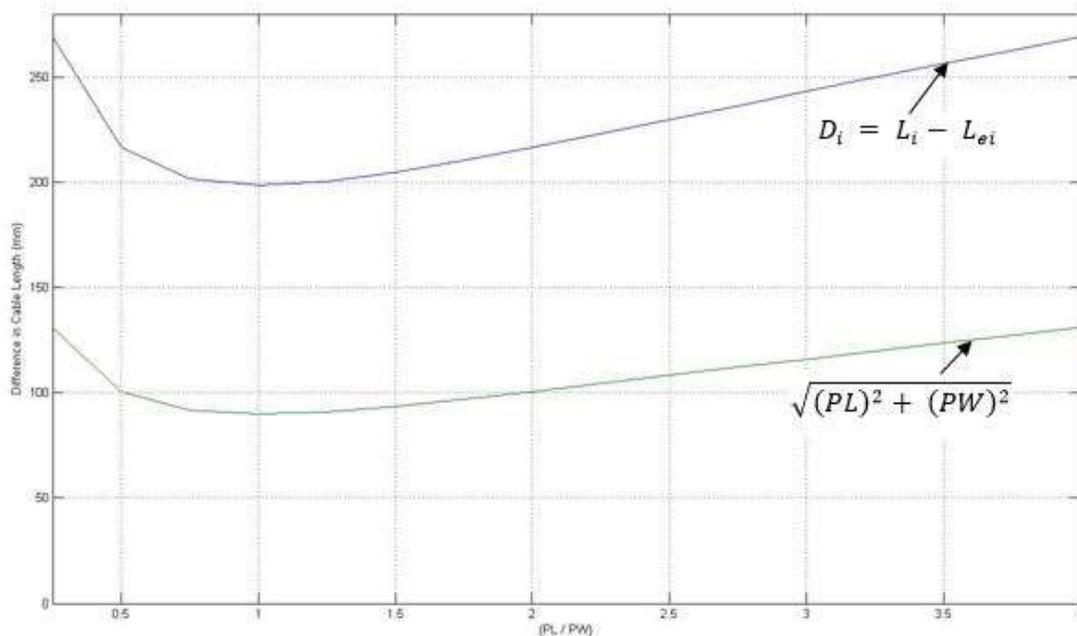


Figure 29: Difference in length vs (PL/PW)

## **Computational Considerations**

The straight line model has been used in most the cable robot systems when compared to the cable sag model (which is used in a small fraction of cable robotic systems). One of the main reasons for this is its simplicity and an analytical model which is extremely easy to use, manipulate, and implement in control systems.

The cable sag model which uses the catenary equations describes the profile of the cable more accurately when compared to the straight line model. However, as described in the previous chapter, the methods required to handle this are highly complicated. Ultimately, any model has to be implemented in a real-time control system to manipulate the cable robot system. All the calculations that are discussed in this research have to be implemented in a control architecture that is suitable for the Algae Harvesting Cable Robot System. Hence, understanding the computational complexities involved is of the utmost importance.

The catenary equations by themselves are highly nonlinear and are implicit in nature. These equations have to be solved simultaneously with other equations by numerical methods iteratively, which is not only time consuming, but may also involve iteration errors. This is a major drawback to the cable sag model.

Another major impediment in using iterative methods is the truncation errors involved. These are especially dominant when exponential and hyperbolic terms are approximated using truncated infinite series, thus reducing the accuracy of the solution. To improve the accuracy of the solution, such approximations have to be made with more

terms in a series expansion, hence requiring more data storage and ultimately increasing the computational cost.

To further investigate this issue, during the computation of cable lengths, the number of iterations for both snapshot points and trajectory were recorded. This information is presented in Figures 30 and 31.

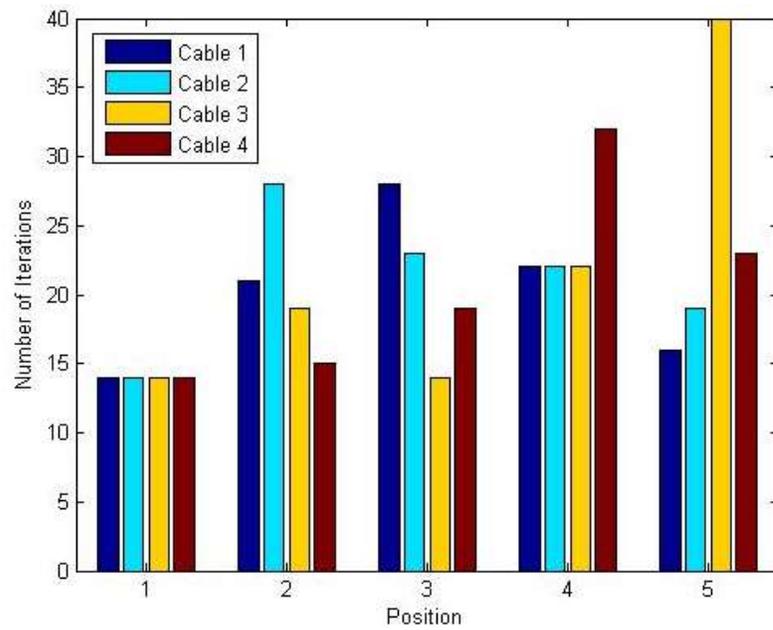


Figure 30: Number of iterations vs position

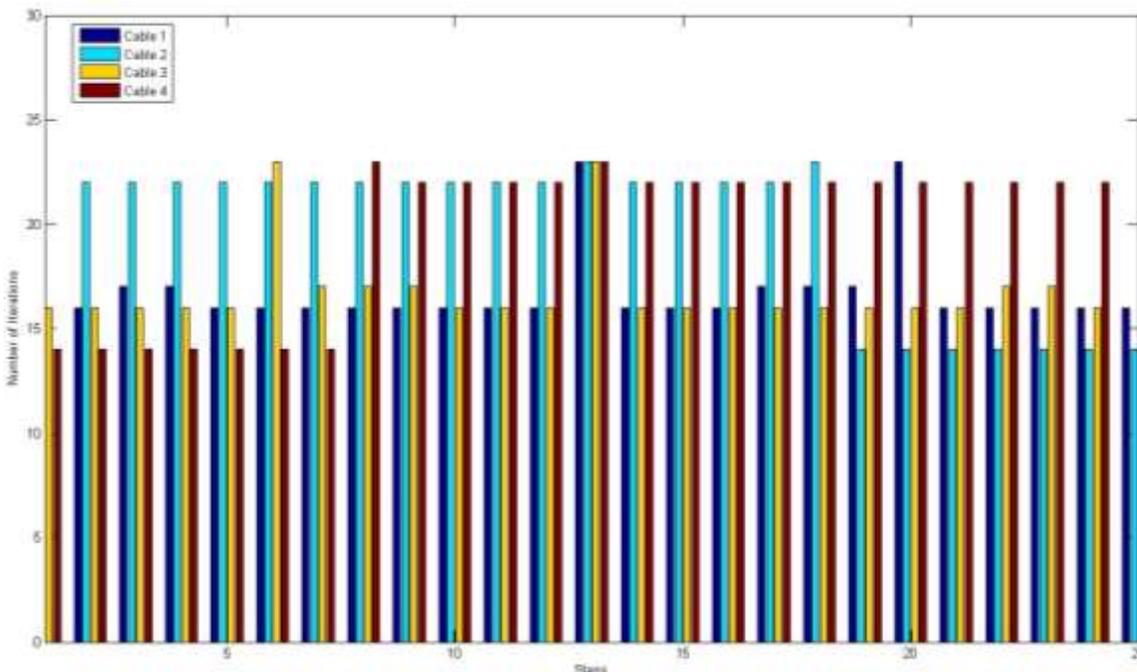


Figure 31: Number of iterations vs steps

There is no definitive prediction that can be made on the number of iterations for a different trajectory or snapshot example. However, the examples shown above help in approximately predicting the iterations for similar trajectories. More importantly, they show that even for the simplest trajectories or snapshot points, each cable length computation requires a considerable number of iterations, ranging from (10-40). When implemented in real time control systems, this elevates the computational complexity steeply. Thus, the cable sag model despite being an accurate model, suffers from serious computational efficiency, considering practical implementation. The relative comparison between the straight line model and cable sag model is as shown below.

Table 6: Comparison between straight line and cable sag model

Criteria	Straight line model	Cable sag model

Governing equations	Euclidean norm between two points	Catenary equations
Type of the model	Approximate model	Accurate model
Kinematics and Statics	Has closed form analytical solution for both. Both problems can be solved independently.	Both problems are coupled and there is no analytical solution.
Nature of solution methods	Analytical and graphical	Numerical
Mass of the cable	Neglected	Included
Areas of application	Small scale robots and where accuracy is not a prime concern.	Any cable robot system
Errors involved	Cable length computation errors	Iterative errors, truncation errors
Control system application	Easy	Difficult

### **Cable Suggestions**

From the results obtained earlier, it is clear that cable density and diameter have a significant effect on cable sag. The appropriate choice of cable must not only have lower density and diameter, but also offer sufficient mechanical strength and be available in different sizes for more flexibility to cater different pond sizes. Research was done, keeping these things in perspective, to find standard cables and their respective OEM's information. Traditionally, metal (especially steel) cables are preferred for high load and large span applications involving cable actuation. But these cables suffer from corrosion and changes in their mechanical properties with temperature and time [9]. Hence non-metallic cables, such as synthetic cables were also researched. Non-metallic cables offer relatively high strength to weight ratio, corrosion resistant properties, and lower variation of mechanical

properties over time. But non-metallic cables are generally expensive compared to metallic cables; however in some cases this margin is narrow.

The tables below shows a relative comparison between metallic, non-metallic, and synthetic cables and also a snapshot example from each category.

Table 7: Types of cables and their properties

Type	Strength	Corrosion Resistance	Cost	Density	Strength / Weight	Temperature Strain
Metallic	Medium	Low	Low	High	Medium	High
Non-metallic	High	High	High	Low	High	Medium
Synthetic	High	High	High	Low	High	Low

Table 8: Snapshot examples for different cable materials

Type	Material	Density (kg/m <sup>3</sup> )	Difference in Cable length (mm)
Metallic	Steel	7860	207.5
Non-metallic	Polythene	1150	4.2
Synthetic	Kevlar	1440	6.6

The following table provides some OEM details for a few cable choices including metallic and non-metallic cables. These manufacturers offer different cable diameters, so that they can cater to various pond sizes and end-effector masses.

Table 9: Cable OEM details

OEM Details	Cable Type

DuPont™ Kevlar® [30]	Composite
VER Sales, Inc. [31]	Metallic and Non-metallic
Hanes Supply Inc. (HSI) [32]	Metallic and Non-metallic
Novabraid Spectra® [33]	Synthetic
Cortland Company [34]	Synthetic
Marlow Ropes [35]	Metallic and Non-metallic
Yale Cordage [36]	Metallic and Non-metallic
Cancord Inc. [37]	Non-metallic

## CHAPTER 4: CONCLUSION AND RECOMMENDATION

The current research was conducted primarily with an intention of studying and understanding the qualitative and quantitative effects of cable sag on the calculation of cable lengths in cable robots. The research also involved finding the effects of a few key parameters, such as cable density, diameter, pond size, and the computational expenses involved in the methods required to find the results presented.

The Algae Harvesting Cable Robot System is not a manipulator that requires high accuracy and speed. Furthermore, simplicity in design and operation could be an extremely attractive trait for its field implementation and commercialization of this idea. This requires certain tradeoffs in design and modeling of the system.

Based on the results of the snapshot and trajectory examples, the relative error in cable lengths does not exceed 3%. The cable sag model suffers from computational complexities. On the other hand, the straight line model is simple to manipulate, control, and implement practically. Considering all these factors and results presented earlier, the straight line model is preferred over the cable sag model for this particular application.

Cable tension distribution is an extremely important aspect of cable suspended robots and, based on the results of this research, linear programming serves as an efficient tool for computing and ensuring appropriate cable tensions in cables. An additional benefit of this method is to help in finding if a given cable tension range is acceptable for motion control of a cable robot system and is within the torque limitations of a winch / motor. Conversely, the simulation results could be used for appropriate choice of winch / motor.

The results of this research were based on simulations and its validation was done based on literature and non-experimental methods. The lack of experimental validation is a limitation of this research and could be a potential vulnerability during field applications. However, various verifications were performed to ensure the simulations were performed accurately. Despite these limitations, the simulations still serve as powerful tools for understanding a broad range of aspects surrounding the cable robot system. Besides, it might be impractical and expensive to test all possible parameters experimentally. Hence the results of simulations do have an importance that cannot be overlooked or compensated by experimental methods.

This research has focused primarily on addressing the cable sag and tension distribution issues of the Algae Harvesting Cable Robot System pointed out by Needler in [9] and with the help of simulations key decisions were recommended, thus furthering the project towards field implementation and commercialization.

Although certain important issues were addressed in this research, there are other important areas which require attention. One of them is the prototype development and testing of the system. This is one of major steps towards field implementation and automation of the Algae Harvesting Cable Robot System.

Another important area of work is to explore the possibilities of using a Global Positioning System (GPS) to manipulate and control the system. This basically involves using GPS and related technology to control the motion of the end-effector. This idea, if successful, could be of paramount importance not only for the Algae Harvesting Cable Robot system, but cable robots in general. Also, cable sag compensation could be achieved

using this technology without cumbersome calculations or complex modeling and solution procedures.

Significant research has been done in the field of GPS implementation to autonomous robots [38,39]. This includes path or trajectory planning, obstacle detection, motion control, and other related areas. However, very little or none of this research is focused on addressing issues related to cable robots. But, the concept might be extended to cable robots with adequate research in this field. Considering all these factors, GPS implementation for the Algae Harvesting Cable Robot System is promising.

## REFERENCES

- [1] Craig, J. J., 2005, Introduction to robotics: mechanics and control, Pearson/Prentice Hall, Upper Saddle River, N.J.
- [2] Williams II, R. L., 2013, ME 4290/5290 Mechanics & Control of Robotic Manipulators: Notesbook and online supplement, Dr. Bob Productions.
- [3] Albus, J. S., Bostelman, R. V., and Dagalakis, N. G., 1992, “The NIST ROBOCRANE,” *J. Robot. Syst.*, **10**, No. 5, pp. 709 – 724.
- [4] Williams II, R. L., Albus, J. S., and Bostelman, R. V., 2004, “3D Cable-Based Cartesian Metrology System,” *J. Robot. Syst.*, **21**(5), pp. 237–257.
- [5] Williams II, R. L., and Gallina, P., 2003, “Translational planar cable-direct-driven robots,” *J. Intell. Robot. Syst.*, **37**(1), pp. 69–96.
- [6] “SkyCam” [Online]. Available: <http://www.skycam.tv/>. [Accessed: 23-Mar-2015].
- [7] Verhoeven, R., 2004, “Analysis of the workspace of tendon-based Stewart platforms,” Universität Duisburg-Essen, Fakultät für Ingenieurwissenschaften» Ingenieurwissenschaften-Campus Duisburg» Abteilung Maschinenbau.
- [8] Bosscher, P., Williams II, R. L., Bryson, L. S., and Castro-Lacouture, D., 2007, “Cable-suspended robotic contour crafting system,” *Autom. Constr.*, **17**(1), pp. 45–55.
- [9] Needler, N. J., 2013, “Design of an Algae Harvesting Cable Robot, Including a Novel Solution to the Forward Pose Kinematics Problem,” Ohio University.

- [10] Lambert, C., Nahon, M., and Chalmers, D., 2007, "Implementation of an Aerostat Positioning System With Cable Control," *IEEEASME Trans. Mechatron.*, **12**(1), pp. 32–40.
- [11] Bruckmann, T., and Pott, A., eds., 2013, *Cable-Driven Parallel Robots*, Springer Berlin Heidelberg, Berlin, Heidelberg.
- [12] Meunier, G., Boulet, B., and Nahon, M., 2009, "Control of an Overactuated Cable-Driven Parallel Mechanism for a Radio Telescope Application," *IEEE Trans. Control Syst. Technol.*, **17**(5), pp. 1043–1054.
- [13] Mikelsons, L., Bruckmann, T., Hiller, M., and Schramm, D., 2008, "A real-time capable force calculation algorithm for redundant tendon-based parallel manipulators," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, IEEE, pp. 3869–3874.
- [14] Borgstrom, P. H., Jordan, B. L., Sukhatme, G. S., Batalin, M. A., and Kaiser, W. J., 2009, "Rapid Computation of Optimally Safe Tension Distributions for Parallel Cable-Driven Robots," *IEEE Trans. Robot.*, **25**(6), pp. 1271–1281.
- [15] Oh, S.-R., and Agrawal, S. K., 2005, "Cable suspended planar robots with redundant cables: controllers with positive tensions," *Robot. IEEE Trans. On*, **21**(3), pp. 457–465.
- [16] Fang, S., Franitza, D., Torlo, M., Bekes, F., and Hiller, M., 2004, "Motion Control of a Tendon-Based Parallel Manipulator Using Optimal Tension Distribution," *IEEEASME Trans. Mechatron.*, **9**(3), pp. 561–568.

- [17] Nan, R., Li, D., Jin, C., Wang, Q., Zhu, L., Zhu, W., Zhang, H., Yue, Y., and Qian, L., 2011, “The five-hundred-meter aperture spherical radio telescope (FAST) project,” *Int. J. Mod. Phys. D*, **20**(06), pp. 989–1024.
- [18] Michelin, M., Baradat, C., Nguyen, D. Q., and Gouttefarde, M., 2015, “Simulation and Control with XDE and Matlab/Simulink of a Cable-Driven Parallel Robot (CoGiRo),” *Cable-Driven Parallel Robots*, A. Pott, and T. Bruckmann, eds., Springer International Publishing, Cham, pp. 71–83.
- [19] Kozak, K., Qian, Z., and Jinsong, W., 2006, “Static analysis of cable-driven manipulators with non-negligible cable mass,” *IEEE Trans. Robot.*, **22**(3), pp. 425–433.
- [20] Irvine, H. M., 1981, *Cable structures*, MIT Press, Cambridge, Mass.
- [21] Russell, J. C., and Lardner, T. J., 1997, “Statics experiments on an elastic catenary,” *J. Eng. Mech.*, **123**(12), pp. 1322–1324.
- [22] Yao, R., Li, H., and Zhang, X., 2013, “A Modeling Method of the Cable Driven Parallel Manipulator for FAST,” *Cable-Driven Parallel Robots*, T. Bruckmann, and A. Pott, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 423–436.
- [23] Li, H., Zhang, X., Yao, R., Sun, J., Pan, G., and Zhu, W., 2013, “Optimal Force Distribution Based on Slack Rope Model in the Incompletely Constrained Cable-Driven Parallel Mechanism of FAST Telescope,” *Cable-Driven Parallel Robots*, T. Bruckmann, and A. Pott, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 87–102.

- [24] Riehl, N., Gouttefarde, M., Krut, S., Baradat, C., and Pierrot, F., 2009, “Effects of non-negligible cable mass on the static behavior of large workspace cable-driven parallel mechanisms,” *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*, IEEE, pp. 2193–2198.
- [25] Riehl, N., Gouttefarde, M., Baradat, C., and Pierrot, F., 2010, “On the determination of cable characteristics for large dimension cable-driven parallel mechanisms,” *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, pp. 4709–4714.
- [26] Gouttefarde, M., Collard, J.-F., Riehl, N., and Baradat, C., 2012, “Simplified static analysis of large-dimension parallel cable-driven robots,” *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, pp. 2299–2305.
- [27] Nguyen, D. Q., Gouttefarde, M., Company, O., and Pierrot, F., 2013, “On the simplifications of cable model in static analysis of large-dimension cable-driven parallel robots,” *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, pp. 928–934.
- [28] Dallej, T., Gouttefarde, M., Andreff, N., Dahmouche, R., and Martinet, P., 2012, “Vision-based modeling and control of large-dimension cable-driven parallel robots,” *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1581–1586.
- [29] “MATLAB Documentation” [Online]. Available: <http://www.mathworks.com/help/matlab/>. [Accessed: 06-Apr-2015].

- [30] “Kevlar® Ropes and Cables | DuPont | DuPont USA” [Online]. Available:  
<http://www.dupont.com/products-and-services/fabrics-fibers-nonwovens/fibers/uses-and-applications/ropes-cables.html>. [Accessed: 08-Apr-2015].
- [31] “VER SALES, INC. - Wire Rope” [Online]. Available:  
[http://versales.com/ns/wire\\_ropes/wirerope.html](http://versales.com/ns/wire_ropes/wirerope.html). [Accessed: 08-Apr-2015].
- [32] “HSI: Hanes Supply Contractor and Industrial Supplier - Home” [Online]. Available: <http://www.hanessupply.com/content/default.aspx>. [Accessed: 08-Apr-2015].
- [33] “Spectra® Rope | Novabraid” [Online]. Available:  
<http://www.novabraid.com/rope/material/spectra.html>. [Accessed: 08-Apr-2015].
- [34] “Cortland Engineered synthetic ropes, heavy lift slings, electro-optical-mechanical cables, and umbilicals | Cortland” [Online]. Available:  
<http://www.cortlandcompany.com/>. [Accessed: 08-Apr-2015].
- [35] “Welcome to Marlow Ropes Ltd” [Online]. Available:  
<http://www.marlowropes.com/>. [Accessed: 08-Apr-2015].
- [36] “Custom Cordage & Braided Rope | Yale Cordage” [Online]. Available:  
<http://www.yalecordage.com/>. [Accessed: 08-Apr-2015].
- [37] “Rope Manufacturing - Industrial Cord and Quaility Twine - Cancord” [Online]. Available: <http://www.cancord.com/pagefiles/aboutus>. [Accessed: 08-Apr-2015].

- [38] Maier, D., and Kleiner, A., 2010, “Improved GPS sensor model for mobile robots in urban terrain,” *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, pp. 4385–4390.
- [39] Saito, T., and Kuroda, Y., 2013, “Mobile robot localization using multiple observations based on place recognition and gps,” *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, pp. 1548–1553.

## APPENDIX: MATLAB CODE

Below mentioned are the codes for simulating the inverse and forward pose kinematics-statics problem and optimization of cable tensions. This also includes the effects of cable parameters, algae pond areas, and computational complexities with relevant graphs.

```
% Inverse Pose Kinematics and Statics Problem for Catenary Cable Model
% Date : 2/25/2015, Author : Dheerendra Sridhar, Revision No : 00

clear;clc;

%----- Start - Input Values -----
%-----%
PL = 80.9; PW = 50; Ph = 7.6; % Dimensions of the algae pond in 'm'
delx = 6.1; dely = 6.1; % Pond offsets in 'm'
x = 0; % Position %
y = 0; % of the end %
z = 0; % effector %
m = 258.6; % Mass of the end effector in 'kg'
d = 20; % Diameter of the cable in 'mm'
rho = 7860; % Density of the cable material in 'kg/m^3'
rho_l = rho*(pi/4)*(d^2)*(1/(10^6)); % Linear Density of the cable in
'kg/m'
g = -9.81; % Acceleration due to gravity in 'm/s^2'
%----- End - Input Values -----
%-----%

P1x = -PL/2-delx; P1y = -PW/2-dely; P1z = Ph; % Coordinates %
P2x = -PL/2-delx; P2y = PW/2+dely; P2z = Ph; % of the tower %
P3x = PL/2+delx; P3y = PW/2+dely; P3z = Ph; % in the global %
P4x = PL/2+delx; P4y = -PW/2-dely; P4z = Ph; % or 'A' frame %

Le1 = sqrt((P1x-x)^2 + (P1y-y)^2 + (P1z-z)^2) % Euclidean %
Le2 = sqrt((P2x-x)^2 + (P2y-y)^2 + (P2z-z)^2) % norm lengths %
Le3 = sqrt((P3x-x)^2 + (P3y-y)^2 + (P3z-z)^2) % of the %
Le4 = sqrt((P4x-x)^2 + (P4y-y)^2 + (P4z-z)^2) % cables %

Aeq = [(P1x-x)/Le1 (P2x-x)/Le2 (P3x-x)/Le3 (P4x-x)/Le4; % Statics
Jacobian Matrix; %
(P1y-y)/Le1 (P2y-y)/Le2 (P3y-y)/Le3 (P4y-y)/Le4; % Also the
coefficient matrix %
(P1z-z)/Le1 (P2z-z)/Le2 (P3z-z)/Le3 (P4z-z)/Le4]; % of the
constraints %

%----- Start - Optimization of cable tensions using
Linear Programming -----%
f = [1; 1; 1; 1]; % Co-efficients of the objective functions i.e. cable
tensions (T1+T2+T3+T4)
```

```

beq = [0; 0; -m*g]; % RHS of the constraint equations i.e. -[0;0;mg]
lb = [2536.866;2536.866;2536.866;2536.866]; % Lower bounds of cable
tensions in 'N'
ub = [Inf;Inf;Inf;Inf]; % Upper bounds of cable tensions in 'N'
A = []; % No inequality constraints
b = []; % No inequality constraints
T = linprog(f,A,b,Aeq,beq,lb,ub)
%----- End - Optimization of cable tensions using
Linear Programming -----%

Tin1=T(1); Tin2=T(2); Tin3=T(3); Tin4=T(4); % Optimum cable tension
values from 'linprog' which is used in 'fsolve'

T_pi = pinv(Aeq)*beq % Moore-Penrose pseudoinverse solution for cable
tensions
sigma_T = sum(T); % Summation of cable tension values got from Linear
Optimization
sigma_T_pi = sum(T_pi); % Summation of cable tension values got from
using Pseudoinverse
Difference_in_total_cable_tension = sigma_T_pi - sigma_T

%----- Start - Coordinates of the cable
calculated in "local cable frame" -----%
z1_end = -Ph + z; z2_end = -Ph + z; z3_end = -Ph + z; z4_end = -Ph + z;
x1_end = sqrt(Le1^2-z1_end^2); x2_end = sqrt(Le2^2-z2_end^2); x3_end =
sqrt(Le3^2-z3_end^2); x4_end = sqrt(Le4^2-z4_end^2);
%----- End - Coordinates of the cable
calculated in "local cable frame" -----%

%----- Start - Elastic Catenary model for cable length
computation -----%
% Cable 1
x0 = [10;-10;Le1]; % 'Initial guess' to solve the system of equations
f = @(x) cable_catenary_1(x,g,rho_l,Tin1,x1_end,z1_end);
options = optimoptions('fsolve','Display','iter');
[x] = fsolve(f,x0,options) % Call solver
% Cable 2
x0 = [10;-10;Le2]; % 'Initial guess' to solve the system of equations
f = @(x) cable_catenary_2(x,g,rho_l,Tin2,x2_end,z2_end);
options = optimoptions('fsolve','Display','iter');
[x] = fsolve(f,x0,options) % Call solver
% Cable 3
x0 = [10;-10;Le3]; % 'Initial guess' to solve the system of equations
f = @(x) cable_catenary_3(x,g,rho_l,Tin3,x3_end,z3_end);
options = optimoptions('fsolve','Display','iter');
[x] = fsolve(f,x0,options) % Call solver
% Cable 4
x0 = [10;-10;Le4]; % 'Initial guess' to solve the system of equations
f = @(x) cable_catenary_4(x,g,rho_l,Tin4,x4_end,z4_end);
options = optimoptions('fsolve','Display','iter');
[x] = fsolve(f,x0,options) % Call solver
%----- End - Elastic Catenary model for cable length
computation -----%

```

```

function F = cable_catenary_1(x,g,rho_l,Tin1,x1_end,z1_end)
F = [(x(1)/(rho_l*g))*(asinh((x(2)/x(1)))-asinh((x(2)-
(rho_l*g)*x(3))/x(1))) - x1_end;
      (1/(rho_l*g))*( Tin1 - sqrt(x(1)^2 + (x(2) - (rho_l*g)*x(3))^2)) -
z1_end;
      sqrt(x(1)^2 + x(2)^2) - Tin1];
function F = cable_catenary_2(x,g,rho_l,Tin2,x2_end,z2_end)
F = [(x(1)/(rho_l*g))*(asinh((x(2)/x(1)))-asinh((x(2)-
(rho_l*g)*x(3))/x(1))) - x2_end;
      (1/(rho_l*g))*( Tin2 - sqrt(x(1)^2 + (x(2) - (rho_l*g)*x(3))^2)) -
z2_end;
      sqrt(x(1)^2 + x(2)^2) - Tin2];
function F = cable_catenary_3(x,g,rho_l,Tin3,x3_end,z3_end)
F = [(x(1)/(rho_l*g))*(asinh((x(2)/x(1)))-asinh((x(2)-
(rho_l*g)*x(3))/x(1))) - x3_end;
      (1/(rho_l*g))*( Tin3 - sqrt(x(1)^2 + (x(2) - (rho_l*g)*x(3))^2)) -
z3_end;
      sqrt(x(1)^2 + x(2)^2) - Tin3];
function F = cable_catenary_4(x,g,rho_l,Tin4,x4_end,z4_end)
F = [(x(1)/(rho_l*g))*(asinh((x(2)/x(1)))-asinh((x(2)-
(rho_l*g)*x(3))/x(1))) - x4_end;
      (1/(rho_l*g))*( Tin4 - sqrt(x(1)^2 + (x(2) - (rho_l*g)*x(3))^2)) -
z4_end;
      sqrt(x(1)^2 + x(2)^2) - Tin4];
% Cable length differences
clear;clc;
P = [1,2,3,4,5];
DL = [207.4 207.4 207.4 207.4; 125.8 11.1 2373.4 1206.7; 4 459.3 2340.5
2539.6; 1433.2 464.2 613.3 3.4; 300.6 2611.5 0.6 520.6];

bar(P,DL);
xlabel('Position');
ylabel('Difference in Cable Lengths (mm)');
legend('Cable 1','Cable 2','Cable 3','Cable 4');
% Cable length errors
clear;clc;
P = [1,2,3,4,5];
EL = [0.365759029 0.365759029 0.365759029 0.365759029; 0.277954537
0.040033903 2.916355379 1.373208382; 0.020896786 0.875380704
2.427587266 3.051325613; 1.843327456 0.514283499 1.154575285
0.014941443; 0.307836308 3.082914644 0.004018001 0.947850041];

bar(P,EL);
xlabel('Position');
ylabel('Percentage error in Cable Lengths (%)');
legend('Cable 1','Cable 2','Cable 3','Cable 4');
% Cable tension differences
clear;clc;
P = [1,2,3,4,5]; % place bars at these points along x-axis
T = [4714.6 4714.6 4714.6 4714.6; 4298.5 6763.3 2536.8
3856.7; 6353.5 2865.1 3244.8 2536.8; 2967.4 6323.1 2536.8
9240.4; 8775.6 2536.8 11976.6 2861.7];

```

```

Tp = [4714.6    4714.6    4714.6    4714.6;4645.3    6550.3    3144.1
3190.6;6277.5    3071.5    2871.6    2856.9;4827.8    4134.1    3816.8
8685.8;5417.6    5368.7    11461.5    4738.4];

figure
width1 = 0.5;
bar(P,T,width1, 'FaceColor', [0.2,0.2,0.5])

hold on
width2 = width1/2;
bar(P,Tp,width2, 'FaceColor', [0,0.7,0.7],...
      'EdgeColor', [0,0.7,0.7])

hold off
xlabel('Position');
ylabel('Cable Tension (N)');
legend('Cable 1-LP','Cable 2-LP','Cable 3-LP','Cable 4-LP','Cable 1-
PI','Cable 2-PI','Cable 3-PI','Cable 4-PI');

% Points for which the IPK&S will be calculated
clear;clc;
X = [0,-29.4,-33,28.5,35];
Y = [0,10.2,-18.8,-18,22];
Z = [0,1.5,2,3.1,5];
length(X)
length(Y)
length(Z)
scatter3(X,Y,Z, 'fill');
xlabel('{X} (m)');
ylabel('{Y} (m)');
zlabel('{Z} (m)');
axis([-46.5 46.5 -31.1 31.1 0 7.6]);
% Computational Efficiency
clear;clc;
P = [1,2,3,4,5];
I = [14 14 14 14; 21 28 19 15; 28 23 14 19; 22 22 22 32; 16
19 40 23];

bar(P,I);
xlabel('Position');
ylabel('Number of Iterations');
legend('Cable 1','Cable 2','Cable 3','Cable 4');
% Cable Length vs Steps
clear;clc;
x =
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
Ce1 =
[51.7162,51.6548,51.5981,52.0618,52.5261,52.991,53.4566,53.9228,54.3896
,54.857,55.3249,55.7934,56.2625,56.7321,57.2022,57.6728,58.1438,58.6154
,59.0874,59.5599,60.0329,60.5063,60.98,61.028,61.08];
Ce2 =
[56.322,56.2656,56.2136,56.1985,56.1878,56.1816,56.1798,56.1824,56.1896
,56.2011,56.2171,56.2376,56.2625,56.2918,56.3255,56.3637,56.4063,56.453
2,56.5046,56.5603,56.6204,56.6848,56.7536,56.8051,56.8609];

```

```

Ce3 =
[61.08, 61.028, 60.98, 60.5063, 60.0329, 59.5599, 59.0874, 58.6154, 58.1438, 57.
6728, 57.2022, 56.7321, 56.2625, 55.7934, 55.3249, 54.857, 54.3896, 53.9228, 53.
4566, 52.991, 52.5261, 52.0618, 51.5981, 51.6548, 51.7162];
Ce4 =
[56.8609, 56.8051, 56.7536, 56.6848, 56.6204, 56.5603, 56.5046, 56.4532, 56.406
3, 56.3637, 56.3255, 56.2918, 56.2625, 56.2376, 56.2171, 56.2011, 56.1896, 56.18
24, 56.1798, 56.1816, 56.1878, 56.1985, 56.2136, 56.2656, 56.322];
C1 =
[51.7664, 51.6958, 51.6312, 52.0957, 52.5608, 53.0266, 53.493, 53.9601, 54.4278
, 54.8962, 55.3651, 55.8346, 56.3723, 56.776, 57.248, 57.7205, 58.1937, 58.6675,
59.1419, 59.6168, 60.0924, 60.5685, 61.0453, 61.11, 61.1817];
C2 =
[57.1002, 57.0387, 56.9818, 56.966, 56.9548, 56.9483, 56.9465, 56.9493, 56.9567
, 56.9688, 56.9854, 57.0068, 56.3723, 57.009, 56.9953, 56.9909, 56.9951, 57.0073
, 57.027, 57.054, 57.0878, 57.1281, 57.1747, 57.2468, 57.3217];
C3 =
[61.1817, 61.1099, 61.0453, 60.5685, 60.0923, 59.6168, 59.1419, 58.6675, 58.193
7, 57.7206, 57.248, 56.776, 56.3723, 55.8346, 55.3651, 54.8962, 54.4278, 53.9601
, 53.493, 53.0266, 52.5608, 52.0957, 51.6312, 51.6958, 51.7664];
C4 =
[57.3217, 57.2468, 57.1747, 57.1281, 57.0878, 57.054, 57.027, 57.0073, 56.995, 5
6.9909, 56.9953, 57.009, 56.3723, 57.0068, 56.9854, 56.9688, 56.9567, 56.9493, 5
6.9465, 56.9483, 56.9548, 56.966, 56.9818, 57.0387, 57.1002];

plot(x, Ce1, ':r', x, C1, 'r', x, Ce2, ':b', x, C2, 'b', x, Ce3, ':g', x, C3, 'g', x, Ce4,
':k', x, C4, 'k');
grid;
xlabel('Steps');
ylabel('Cable Lengths (m)');
legend('L_e_1', 'L_1', 'L_e_2', 'L_2', 'L_e_3', 'L_3', 'L_e_4', 'L_4');
% Cable length difference vs step
clear;clc;
t =
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25];
D1 =
[50.2, 41, 33.1, 33.9, 34.7, 35.6, 36.4, 37.3, 38.2, 39.2, 40.2, 41.2, 109.8, 43.9, 4
5.8, 47.7, 49.9, 52.1, 54.5, 56.9, 59.5, 62.2, 65.3, 82, 101.7];
D2 =
[778.2, 773.1, 768.2, 767.5, 767, 766.7, 766.7, 766.9, 767.1, 767.7, 768.3, 769.2,
109.8, 717.2, 669.8, 627.2, 588.8, 554.1, 522.4, 493.7, 467.4, 443.3, 421.1, 441.7
, 460.8];
D3 =
[101.7, 81.9, 65.3, 62.2, 59.4, 56.9, 54.5, 52.1, 49.9, 47.8, 45.8, 43.9, 109.8, 41.
2, 40.2, 39.2, 38.2, 37.3, 36.4, 35.6, 34.7, 33.9, 33.1, 41, 50.2];
D4 =
[460.8, 441.7, 421.1, 443.3, 467.4, 493.7, 522.4, 554.1, 588.7, 627.2, 669.8, 717.
2, 109.8, 769.2, 768.3, 767.7, 767.1, 766.9, 766.7, 766.7, 767, 767.5, 768.2, 773.1
, 778.2];

plot(t, D1, 'r', t, D2, 'b', t, D3, 'g', t, D4, 'k');
grid;
xlabel('Steps');
ylabel('Difference in Cable Lengths (mm)');

```

```

legend('Cable 1','Cable 2','Cable 3','Cable 4');

% Cable length errors vs step
clear;clc;
t =
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
EC1 =
[0.096974099,0.079310118,0.064108524,0.065072549,0.066018782,0.06713611
7,0.068046286,0.06912515,0.070184722,0.071407493,0.072608918,0.07378937
1,0.194776513,0.077321403,0.080002795,0.082639617,0.085748114,0.0888055
57,0.09215125,0.095442895,0.099014185,0.102693644,0.106969742,0.1341842
58,0.166226175];
EC2 =
[1.36286738,1.355395547,1.34814976,1.347294878,1.34668193,1.346308845,1
.3463514,1.346636394,1.346812579,1.347579728,1.348240076,1.349312714,0.
194776513,1.258046975,1.17518462,1.100526575,1.033071264,0.971980781,0.
916057306,0.865320573,0.818738855,0.775975396,0.736514577,0.771571511,0.
803884044];
EC3 =
[0.166226175,0.134020838,0.106969742,0.102693644,0.098847939,0.09544289
5,0.09215125,0.088805557,0.085748114,0.082812722,0.080002795,0.07732140
3,0.194776513,0.073789371,0.072608918,0.071407493,0.070184722,0.0691251
5,0.068046286,0.067136117,0.066018782,0.065072549,0.064108524,0.0793101
18,0.096974099];
EC4 =
[0.803884044,0.771571511,0.736514577,0.775975396,0.818738855,0.86532057
3,0.916057306,0.971980781,1.032897623,1.100526575,1.17518462,1.25804697
5,0.194776513,1.349312714,1.348240076,1.347579728,1.346812579,1.3466363
94,1.3463514,1.346308845,1.34668193,1.347294878,1.34814976,1.355395547,
1.36286738];

plot(t,EC1,'r',t,EC2,'b',t,EC3,'g',t,EC4,'k');
grid;
xlabel('Steps');
ylabel('Percentage Error in Cable Lengths (%)');
legend('Cable 1','Cable 2','Cable 3','Cable 4');
% Cable length error vs P/L ratio
clear;clc;
R = [0.25;0.5;0.75;1;1.25;1.5;1.75;2;2.25;2.5;2.75;3;3.25;3.5;3.75;4];
EL =
[269.3;216.6;201.9;198.8;200.6;205;210.5;216.6;223.2;229.8;236.6;243.3;
249.9;256.4;262.9;269.3];
SQ =
[131.1154072;100.5609268;;91.79914669;89.94442729;91.06179221;93.617128
06;96.90293376;100.5609268;104.3984089;108.3074328;112.2259288;116.1177
563;119.9619331;123.7465728;127.4653548;131.1154072];

plot(R,EL,R,SQ);
axis([0.25 4 0 280]);
grid;
xlabel('(PL / PW)');
ylabel('Difference in Cable Length (mm)');

```

```

% Differenc in Cable Tensions
clear;clc;
x =
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
T =
[16.8,19.8,23,18.9,15.2,11.9,8.8,6.3,4.2,2.5,1.1,0.2,0,0.4,1.1,2.5,4.2,
6.3,8.7,11.9,15.2,19,23,19.8,16.8];
plot(x,T);
grid
xlabel('Steps');
ylabel('Difference in Cable Tensions (N)');

% Cable Tensions vs Steps
clear;clc;
x =
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
T1 =
[8250.3,9104.3,10111.9,10126.1,10139.3,10151.4,10162.5,10172.5,10181.5,
10189.4,10196.2,10202.1,6371.9,10125.6,10041.9,9955.8,9867.2,9776.2,968
2.6,9586.6,9488.2,9387.2,9283.8,8306.3,7477.7];
T2 =
[2536.8,2536.8,2536.8,2536.8,2536.8,2536.8,2536.8,2536.8,2536.8,2536.8,
2536.9,2536.9,6371.9,2621.1,2705.5,2789.9,2874.5,2959.3,3044.3,3129.4,3
214.7,3300.3,3386.1,3319.1,3262.3];
T3 =
[7477.7,8306.3,9283.8,9387.2,9488.2,9586.6,9682.6,9776.2,9867.2,9955.8,
10041.9,10125.6,6371.9,10202.1,10196.2,10189.4,10181.5,10172.5,10162.5,
10151.4,10139.3,10126,10111.9,9104.3,8250.3];
T4 =
[3262.3,3319.1,3386.1,3300.3,3214.7,3129.4,3044.3,2959.3,2874.5,2789.9,
2705.5,2621.1,6371.9,2536.8,2536.9,2536.8,2536.8,2536.8,2536.8,2536.8,2
536.8,2536.8,2536.8,2536.8,2536.8];
Tp1 =
[6009.6,6494.7,7067,6998.5,6929.7,6860.5,6791,6721.4,6651.6,6581.7,6511
.7,6441.8,6371.9,6302.1,6232.4,6162.9,6093.7,6024.7,5956,5887.7,5819.8,
5752.3,5685.3,5223.2,4831.4];
Tp2 =
[4977,5379.4,5854.1,5912.9,5970.2,6025.9,6080,6132.6,6183.6,6233,6280.8
,6327.1,6371.9,6415,6456.7,6496.8,6535.3,6572.4,6607.9,6642,6674.5,6705
.6,6735.2,6189,5725.9];
Tp3 =
[4831.4,5223.2,5685.3,5752.3,5819.8,5887.7,5956.1,6024.7,6093.7,6162.9,
6232.4,6302,6371.9,6441.8,6511.7,6581.7,6651.6,6721.4,6791,6860.5,6929.
7,6998.5,7067,6494.7,6009.6];
Tp4 =
[5725.9,6189,6735.2,6705.6,6674.5,6642,6607.9,6572.4,6535.3,6496.8,6456
.7,6415,6371.9,6327.1,6280.8,6233,6183.6,6132.6,6080,6025.9,5970.2,5912
.9,5854.1,5379.4,4977];

plot(x,T1,'r',x,Tp1,':r',x,T2,'b',x,Tp2,':b',x,T3,'g',x,Tp3,':g',x,T4,'
k',x,Tp4,':k');
grid
axis([0 25 0 11000]);
xlabel('Steps');

```

```

ylabel('Cable Tensions (N)');
legend('T_1','T_p_1','T_2','T_p_2','T_3','T_p_3','T_4','T_p_4');

% Cartesian Positions vs step
clear;clc;
t =
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
x = [-3,-3,-3,-2.7,-2.4,-2.1,-1.8,-1.5,-1.2,-0.9,-0.6,-
0.3,0,0.3,0.6,0.9,1.2,1.5,1.8,2.1,2.4,2.7,3,3,3];
y = [-4,-4,-4,-3.6,-3.2,-2.8,-2.4,-2,-1.6,-1.2,-0.8,-
0.4,0,0.4,0.8,1.2,1.6,2,2.4,2.8,3.2,3.6,4,4,4];
z = [1,1.5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1.5,1];
plot(t,x,t,y,t,z);
grid;
axis([0 25 -5 5]);
xlabel('Steps');
ylabel('X(m),Y(m),Z(m)');
legend('X','Y','Z');
% Computational Efficiency Trajectory
clear;clc;
P =
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
I = [16 19 16 14; 16 22 16 14; 17 22 16 14; 17 22 16 14; 16
22 16 14;16 22 23 14;16 22 17 14;16 22 17 23;16 22 17
22;16 22 16 22;16 22 16 22;16 22 16 22;23 23 23;16
22 16 22;16 22 16 22;16 22 16 22;17 22 16 22;17 23 16
22;17 14 16 22;23 14 16 22;16 14 16 22;16 14 17 22;16
14 17 22;16 14 16 22;16 14 16 19];

bar(P,I);
axis([1 25 0 30]);
xlabel('Steps');
ylabel('Number of Iterations');
legend('Cable 1','Cable 2','Cable 3','Cable 4');
% Effects of Area of the pond
clear;clc;
A =
[4045,5005.8,6180,7477.8,8899.2,10444.2,12112.8,13906.5,15822.4,17861.9
,20025,22311.7,24722];
L =
[207.4,226.7,248,269.3,290.5,311.6,332.8,354,375.2,396.3,417.5,438.6,45
9.8,];
plot(A,L);
grid
axis([4045 24722 0 500]);
xlabel('Area of the algae pond (m^2)');
ylabel('Difference in Cable Length (mm)');
% Effects of Cable density on cable length
clear;clc;
x =
[1000,1500,2000,2500,3000,3500,4000,4500,5000,5500,6000,6500,7000,7500,
8000,8500,9000,9500,10000];

```



```

y1 =
[45.7, 67.3, 95.8, 132.9, 180.2, 239.8, 314.1, 405.7, 518.1, 655.3, 822.3, 1025.4,
1272.9, 1575.7, 1948.8, 2414.2, 3004.9, 3776, 4829, 6392, 9248.8];

plot(x, y1)
grid;
xlabel('Cable Diameter (mm)');
ylabel('Difference in 1 ^s^t Cable length (mm)');
clear;clc;
x = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30];
y2 =
[5, 7.3, 10.3, 14.2, 19.1, 25.2, 32.7, 41.7, 52.5, 65.3, 80.5, 98.1, 118.6, 142.2, 16
9.4, 200.3, 235.6, 275.6, 320.8, 371.7, 428.9];

plot(x, y2);
grid;
xlabel('Cable Diameter (mm)');
ylabel('Difference in 2 ^n^d Cable length (mm)');
clear;clc;
x = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30];
y3 =
[35.8, 52.7, 75, 104, 141.1, 187.7, 245.7, 317.2, 404.7, 511.3, 640.7, 797.6, 987.7
, 1218.9, 1501.5, 1850, 2285.6, 2841.5, 3573.6, 4590.5, 6156.7];

plot(x, y3);
grid;
xlabel('Cable Diameter (mm)');
ylabel('Difference in 3 ^r^d Cable length (mm)');
clear;clc;
x = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30];
y4 =
[1.5, 2.2, 3.1, 4.3, 5.8, 7.6, 9.9, 12.6, 15.8, 19.7, 24.2, 29.5, 35.6, 42.7, 50.7, 59
.9, 70.3, 82.1, 95.3, 110, 126.6];

plot(x, y4);
grid;
xlabel('Cable Diameter (mm)');
ylabel('Difference in 4 ^t^h Cable length (mm)');
clear;clc;
x =
[1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500,
8000, 8500, 9000, 9500, 10000];
y1 =
[11.7, 26.5, 47.4, 74.5, 107.9, 148, 194.9, 248.8, 310.1, 379.1, 456.1, 541.8, 636.
5, 740.8, 855.4, 981.1, 1118.8, 1269.5, 1434.4];

plot(x, y1)
grid;
xlabel('Cable Density (kg/mm^3)');
ylabel('Difference in 1 ^s^t Cable length (mm)');
clear;clc;

```

```

x =
[1000,1500,2000,2500,3000,3500,4000,4500,5000,5500,6000,6500,7000,7500,
8000,8500,9000,9500,10000];
y2 =
[1.3,2.9,5.1,8,11.6,15.7,20.6,26.1,32.2,39.1,46.6,54.8,63.6,73.2,83.4,9
4.3,106,118.3,131.4];

plot(x,y2)
grid;
xlabel('Cable Density (kg/mm^3)');
ylabel('Difference in 2 ^n^d Cable length (mm)');
clear;clc;
x =
[1000,1500,2000,2500,3000,3500,4000,4500,5000,5500,6000,6500,7000,7500,
8000,8500,9000,9500,10000];
y3 =
[9.2,20.7,37.1,58.3,84.5,115.9,152.5,194.7,242.5,296.4,356.5,423.1,496.
7,577.6,666.4,763.4,869.4,985.1,1111.2];

plot(x,y3)
grid;
xlabel('Cable Density (kg/mm^3)');
ylabel('Difference in 3 ^r^d Cable length (mm)');
clear;clc;
x =
[1000,1500,2000,2500,3000,3500,4000,4500,5000,5500,6000,6500,7000,7500,
8000,8500,9000,9500,10000];
y4 =
[9.2,20.7,37.1,58.3,84.5,115.9,152.5,194.7,242.5,296.4,356.5,423.1,496.
7,577.6,666.4,763.4,869.4,985.1,1111.2];

plot(x,y4)
grid;
xlabel('Cable Density (kg/mm^3)');
ylabel('Difference in 4 ^t^h Cable length (mm)');
% Effects of End-effector mass on cable length
clear;clc;
x = [100,125,150,175,200,225,250,275,300,325,350,375,400];
y1 =
[1332.3,1353.5,1375.3,1397.6,1420.5,1444,1468.1,1492.8,1518.2,1544.4,15
71.2,1598.8,1627.1];

plot(x,y1)
grid;
axis([100 400 0 1650]);
xlabel('End-effector mass (kg)');
ylabel('Difference in 1 ^s^t Cable length (mm)');
% Effects of End-effector mass on cable length
clear;clc;
x = [100,125,150,175,200,225,250,275,300,325,350,375,400];
y2 =
[1515.3,596.1,320.8,200.8,137.6,100.2,76.3,60,48.4,39.9,33.6,28.5,24.5]
;

```

```

plot(x,y2)
grid;
% axis([100 400 0 1650]);
xlabel('End-effector mass (kg)');
ylabel('Difference in 2 ^n^d Cable length (mm)');
% Effects of End-effector mass on cable length
clear;clc;
x = [100,125,150,175,200,225,250,275,300,325,350,375,400];
y3 =
[1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3,1103.3];

plot(x,y3)
grid;
% axis([100 400 0 1650]);
xlabel('End-effector mass (kg)');
ylabel('Difference in 3 ^r^d Cable length (mm)');

% Effects of End-effector mass on cable length
clear;clc;
x = [100,125,150,175,200,225,250,275,300,325,350,375,400];
y4 = [291.1,144.8,86.6,57.6,41.1,30.8,23.9,19.1,15.6,13,11,9.4,8.2];

plot(x,y4)
grid;
% axis([100 400 0 1650]);
xlabel('End-effector mass (kg)');
ylabel('Difference in 4 ^t^h Cable length (mm)');

```



**OHIO**  
UNIVERSITY

Thesis and Dissertation Services