**Proceedings of the ASME 2016 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2016
August 21-24, 2016, Charlotte, NC, USA**

DETC2016-60495

KINEMATICS AND STATICS INCLUDING CABLE SAG FOR LARGE CABLE-SUSPENDED ROBOTS

Dheerendra Sridhar
Mechanical Engineering Department
Ohio University
Athens, Ohio, USA
ds843012@ohio.edu

Robert L. Williams II, Ph.D.
Mechanical Engineering Department
Ohio University
Athens, Ohio, USA
williar4@ohio.edu

ABSTRACT

Cable sag can have significant effects on the cable length computation in a cable-suspended robot and this is more pronounced in large-scale outdoor systems. 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.

A cable-sag-compensated (catenary) model was implemented in simulation for an example large outdoor cable-suspended robot system to solve the coupled kinematics and statics problems. This involved optimization of cable tensions and finding the errors involved in the cable length. A comparative analysis between the straight-line and cable sag model is presented, the main contribution of this paper. Based on the qualitative and quantitative results obtained, recommendations were made on the choice of model and solution methodologies.

KEYWORDS

Cable-suspended robot, cable sag, non-negligible cable mass, catenary model, forward and inverse position kinematics, pseudostatics

1. INTRODUCTION

A distinct attribute of cable-suspended robots is the possibility of achieving very large workspaces which is 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-suspended robot under development in China for astronomical study [1]. Another example is the Skycam [2], which is an aerial camera system that is widely used in sporting arenas. Other examples include CoGiRo (Control of Giant Robots) used for industrial purposes [3] and the Large Cable Mechanism (LCM) used for Radio Telescope Application [4].

Kozak et al. [5] 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 catenary discussed by Irvine [6], to model the cable lengths and subsequently address the inverse pose kinematics problem. Kozak et al. [5] also provided experimental validation and showed that the equations of the catenary are in good agreement with experimental results. Additionally, Russell and Lardner [7] provided experimental validation of the catenary model and quantified the difference between theoretical and experimental cable tensions.

An accuracy and error compensation study of the 6-dof FAST robot was presented by Yao et al. [8] and force distribution in the cables by Li et al. [9]. These results showed that cable sag has a considerable effect on the overall accuracy and control of the robot.

Research on the effects of sag on the workspace and cable characteristics was performed by Riehl et al. [10]. 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 catenary (cable sag) models differ. Cable tension under cable sag, unlike the cable tension for the straight-line model, is not constant throughout the cable.

Irvine [6] presented a simplified model for cable sag based on perturbation analysis. This was used by Gouttefarde et al. [11] 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 catenary. Also, the relationship between the components of the cable tension is linear in this model. This model was further researched by Nguyen et al. [12] to find the limitation of the simplified 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 has been experimentally verified.

Another noteworthy work was by Dallej et al. [13], which was vision-based control of a cable-suspended 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 mathematical modeling of kinematics and pseudostatics for small scale cable-suspended robots generally works well with the assumption of ideal massless cables (straight-line model). However, for large-scale cable-suspended robots, significant errors may arise when assuming the straight-line model for all cables. The main purpose of this paper is to investigate the differences in cable length errors and computation, comparing the straight-line cables assumption vs. a cable-sag model. dit Sandretto et al. [14] test the hypothesis that ignoring cable mass and cable sag is sufficient, with regard to their CoGiRo project. This hypothesis was confirmed for their current prototype hardware, but it was rejected for a planned larger robot. Riehl et al. [10] simply conclude that the cable catenaries must be accounted for, in large workspace cable robots, "in order to achieve good positioning and accuracy." Yaun et al. [15] develop static and dynamic stiffness models for large cable-suspended robots; they conclude that the cable catenary is "important" for stiffness studies.

This paper first presents the methods, followed by results and discussion.

2. METHODS

The methods used by Kozak et al. [5] and subsequently used in [10-12] will be followed in this research.

Cable Sag Catenary

The equations of the cable catenary have been known for more than 80 years and they have been applied in various contexts of engineering, and their derivations are not presented (see [5, 6]). Consider a cable suspended between two points A and B as in Figure 1.

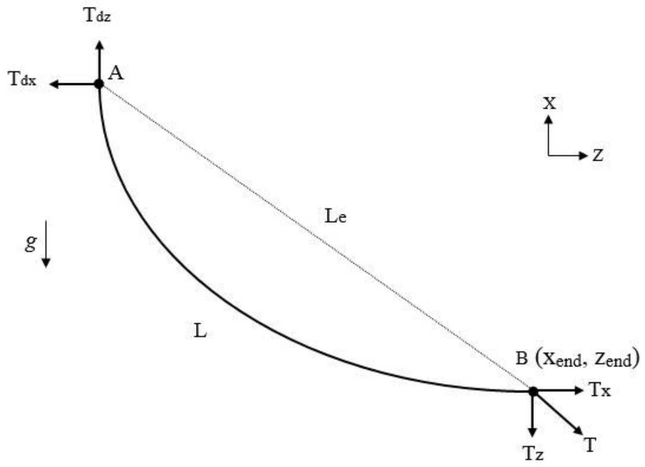

Figure 1: Cable suspended between two points XZ Axes shown for clarity (origin is point A)

Where A is the cable drawing point, B is the end-effector attachment point, L_e is the straight-line (Euclidean norm) distance between A and B, L is the catenary (actual) length between A and B, *g* is the acceleration due to gravity, T is the cable tension with X and Z components T_x and T_z at the end effector side, T_{dx} and T_{dz} are the X and Z components of the cable tension at the cable drawing point, and (x_{end}, z_{end}) are the coordinates of the cable at the end-effector attachment point. For this cable, the static catenary displacement equations for the inextensible case after simplification are (we ignore the axial elasticity since the cable mass dominates the sag):

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

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

Where ρ_L is the linear density of the cable material.

Cable-suspended Robot Model

The kinematic diagram of the cable-suspended robot considered is shown in Figure 2. The base frame {A} is fixed to the center of the robot footprint. The end-effector control point is 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 on the ground. L_i (or L_{ei} according to the notation in Figure 1 and used later in this paper) are the Euclidean norm (straight-line) cable lengths. In all cases *i* = 1,2,3,4. The length and width of the cable-suspended robot footprint are *L* and *W*, respectively.

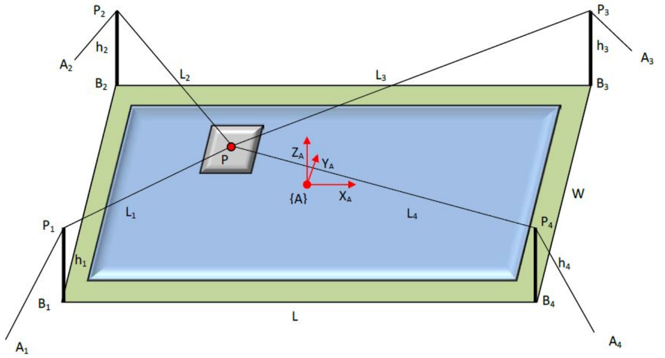

Figure 2: Cable-suspended robot kinematic diagram (1 acre)

Inverse Position Kinematics (IPK) and Statics

The IPK problem consists of finding the active cable lengths for a given position. 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 straight-line IPK problem. Hence, the kinematics and pseudostatics problems are coupled and have to be solved simultaneously, as evident from equations (1) and (2). This is a system of nonlinear implicit equations, hence there are no analytical solutions, thus forcing the use of numerical methods.

As shown in [5] and [10], for a minimally or perfectly constrained case, the catenary equations (1) and (2) are solved along with the static equilibrium equations (3).

$$\begin{aligned} \sum F_x &= 0 \\ \sum F_y &= 0 \\ \sum F_z &= 0 \end{aligned} \qquad (3)$$

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 outnumbers the equations available, there are infinite valid solutions. Consider a 4-cable 3-dof (*XYZ* translation) cable-suspended robot as shown in Figures 2 and 3.

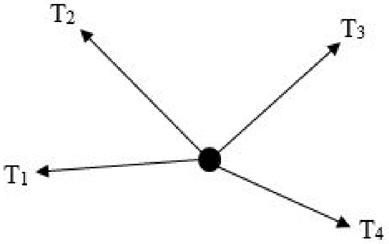

Figure 3: Free-Body Diagram of a four-cable robot

Solving only the static equations, for a given valid position, can have infinite solutions i.e. infinite valid combinations for $T = \begin{bmatrix} T_1 & T_2 & T_3 & T_4 \end{bmatrix}^T$. The physical interpretation of this scenario is that at a given position 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 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 pointed out in [5], when using the catenary equations for finding the cable lengths of a redundant cable-suspended robot, one feasible approach is to solve it as constrained optimization problem or specify the (m-n) number of forces prior to solving.

The methodology adapted here to address the Inverse Position Kinematics and Statics problem is as described in [5,8,9,12]. The details of the method adapted and coded in MATLAB are shown in Figure 4 and described below.

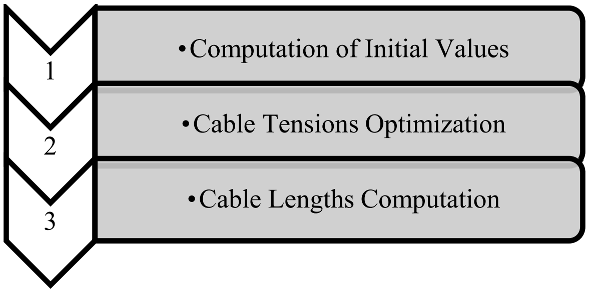
1
•Computation of Initial Values
2
•Cable Tensions Optimization
3
•Cable Lengths Computation

Figure 4: Inverse position problem solution steps

Step 1 - Computation of Initial Values

In this step, all the required inputs are entered for solving the IPK problem, along with necessary parameters such dimensional details of the robot footprint, robot variables, and properties of the cable. 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. Table 1 shows the input variables required.

Table 1: Input variables

Input Variable	Symbol	Unit
Length	L	m
Width	W	m
Pole Height	h	m
End-effector mass	m	kg
Cable Diameter	d	mm
Density of the Cable	ρ	kg/m^3
End-effector location	(*x*,*y*,*z*)	m

The Euclidean norm length of the straight-line cable is calculated using:

$$L_{ei} = \sqrt{(P_{xi}-x)^2 + (P_{yi}-y)^2 + (P_{zi}-z)^2} \qquad (4)$$

The straight-line static Jacobian matrix expressed in {A} coordinates is given by:

$$\left[{}^{A}A\right] = \begin{bmatrix} \frac{P_{x1}-x}{L_{e1}} & \frac{P_{x2}-x}{L_{e2}} & \frac{P_{x3}-x}{L_{e3}} & \frac{P_{x4}-x}{L_{e4}} \\ \frac{P_{y1}-y}{L_{e1}} & \frac{P_{y2}-y}{L_{e2}} & \frac{P_{y3}-y}{L_{e3}} & \frac{P_{y4}-y}{L_{e4}} \\ \frac{P_{z1}-z}{L_{e1}} & \frac{P_{z2}-z}{L_{e2}} & \frac{P_{z3}-z}{L_{e3}} & \frac{P_{z4}-z}{L_{e4}} \end{bmatrix} \qquad (5)$$

where *x*, *y*, *z* is the end-effector location (point B in Figure 1) in {*A*}, P_{xi}, P_{yi}, P_{zi} are the fixed cable connection points (point A in Figure 1), and L_{ei} are the straight-line cable lengths (Figure 1).

Step 2 – Cable Tensions Optimization

In this step, the cable tensions for a given position 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: Minimize $(T_1 + T_2 + T_3 + T_4)$

Subject to Constraints: $\left[{}^{A}A\right]\{T\} + \left\{{}^{A}F\right\} + m\,{}^{A}g = \{0\}$

$T_{\min} \le T_i \le T_{\max}$

Where the cable tensions are $T = \begin{bmatrix} T_1 & T_2 & T_3 & T_4 \end{bmatrix}^T$, $\left\{{}^{A}F\right\}$ is the external force, $m\,{}^{A}g$ is the end-effector weight (both expressed in {*A*} coordinates), and $T_{\min}$ and $T_{\max}$ are the 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 ($T_i > 0$). Bounds not only help in obtaining non-negative solutions (a negative solution for cable tension means a cable must push, which is unacceptable), but also restrict the solution to be within practical limitations, avoiding extremely high cable tensions, which might break the cable or which cannot be supported by the winch / motor. This problem is solved using the linear programming solver in MATLAB, **linprog( )**. Additionally, the pseudoinverse method was also implemented using the **pinv( )** command in MATLAB for comparison purposes.

Step 3 – Cable Lengths 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:

$$x_{iend} = \frac{|T_{xi}|}{\rho_L g}\left[\sinh^{-1}\left(\frac{T_{zi}}{T_{xi}}\right) - \sinh^{-1}\left(\frac{T_{zi} - \rho_L g L_i}{T_{xi}}\right)\right] \qquad (6)$$

$$z_{iend} = \frac{1}{\rho_L g}\left[\sqrt{T_{xi}^2 + T_{zi}^2} - \sqrt{T_{xi}^2 + (T_{zi} - \rho_L g L_i)^2}\right] \qquad (7)$$

$$T_i = \sqrt{T_{xi}^2 + T_{zi}^2} \qquad (8)$$

where *i* = 1,2,3,4. For each cable this a system of three equations with three variables (T_{xi}, T_{zi}, and L_i). To solve this system of equations the **fsolve()** command in MATLAB is used, which is an iterative solver used to solve a system of nonlinear equations with real variables. Also, the number of iterations is recorded. 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. An optimization routine is then performed to get a valid set of cable tensions {*T*}, such that the sum of cable tensions ($\sum T_i$) 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 position kinematics and statics problem comprehensively, such that when the user enters a valid position, the program returns the cable tensions and lengths.

Forward Position Kinematics (FPK) and Statics

The FPK problem consists of finding the position 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 [16], 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. the kinematics 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 tensions and subsequently finding the position of the robot.

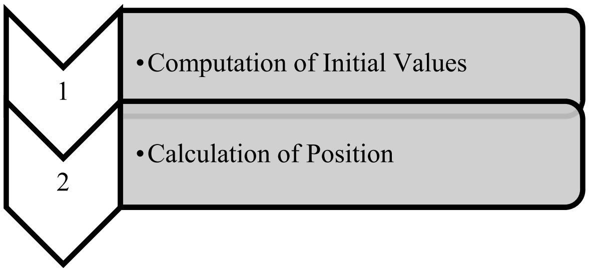

Figure 5: Forward position problem solution steps

Step 1 – Computation of Initial Values

Similar to the 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 robot footprint, and the geometrical and material properties of the cables are entered.

Step 2 – Calculation of Position

In this step, the static displacement equations of the catenary (6-8) along with the static equilibrium equations (3) are solved numerically along with necessary transformations of coordinate system. This system of equations is also solved using the **fsolve()** command in MATLAB and its solution yields the *XYZ* position 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 position. A major difference in this FPK problem, when compared to the inverse position 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.

3. RESULTS AND DISCUSSION

Based on the methods described in the previous section, simulations were performed. This included simulating snapshot examples, a trajectory, and parameter variations. The results obtained and their interpretations are discussed in this section. The simulation results presented here use the values in Table 2.

Table 2: Values of the variables used for simulation

Variable	Value	Notes
Length (L)	50 m	1 acre footprint
Width (W)	80.9 m	
Pole Height (h)	7.6 m	All poles are same height
End-effector mass (*m*)	258.6 kg	-
Cable Diameter (d)	20 mm	-
Density of the Cable (ρ)	7860 kg/m^3	Density of a steel cable
External Force (^AF)	{0}	0 *xyz* vector
Tension Lower Limit (T_{min})	2537 N	-
Tension Higher Limit (T_{max})	+ ∞	To find the maximum force

Snapshot Example

Both the Inverse and Forward Position Problems were solved for four random positions and a nominal position $\{0 \quad 0 \quad 0\}^T$. The five positions are graphically shown in Figure 6 and stated numerically in Table 3.

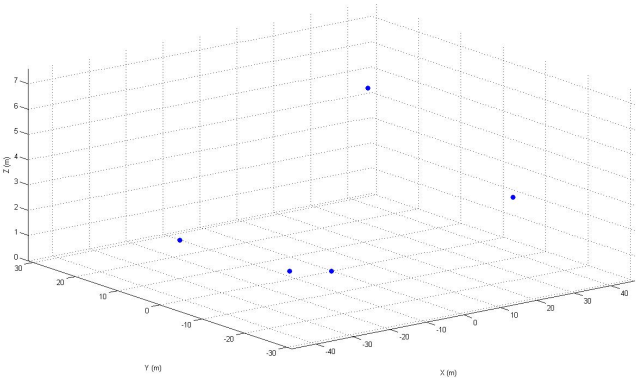

Figure 6: Snapshot points

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

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 circular check was verified for all 5 snapshot points (not shown; see [18]).

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	-33.0	-18.8	2.0
4	28.5	-18.0	3.1
5	35.0	22.0	5.0

The cable length difference between the cable sag and straight model is calculated, followed by cable length error computation:

$$D_i = L_i - L_{ei} \tag{9}$$

$$ER_i = \frac{L_i - L_{ei}}{L_i} \times 100\% \tag{10}$$

The results of difference in cable lengths and their percentage error are plotted in Figure 7 and 8.

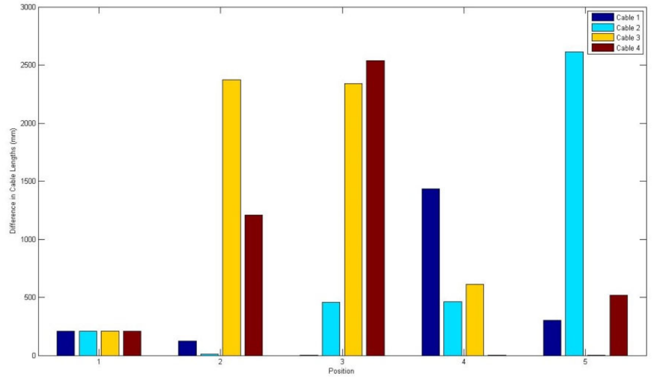

Figure 7: Difference in cable lengths vs. position

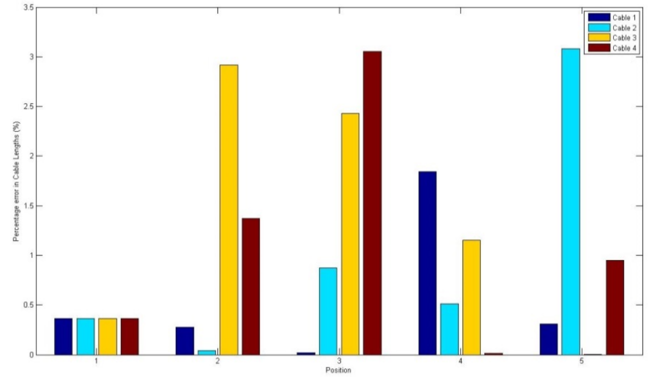

Figure 8: Percentage error in cable lengths vs. position

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 9.

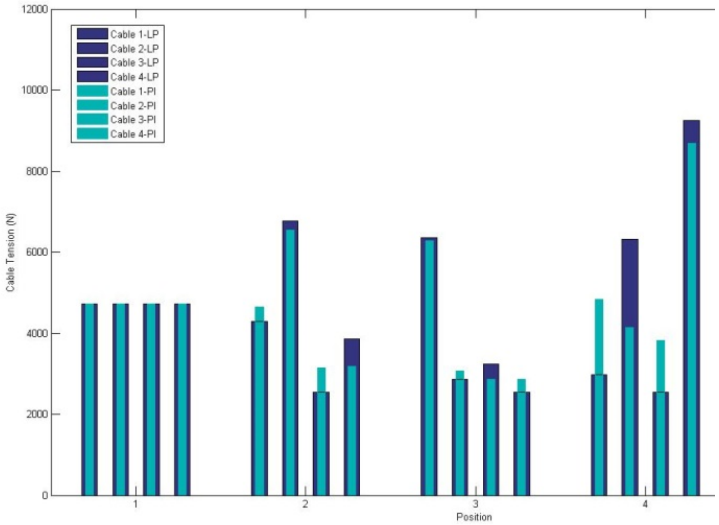

Figure 9: LP and PI Cable tensions vs. position

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 high. However, when the relative error is computed, the range is 0-3 %. The current cable-suspended Robot System, unlike the FAST [1] or LCM [4], 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 of random points; this necessitates running the program to simulate a trajectory.

Trajectory Example

A pick-and-place robot trajectory was simulated with a step size of 0.5 m as shown in Figure 10. The ideal Cartesian coordinates and straight-line cable lengths for this trajectory are shown in Figures 11 and 12.

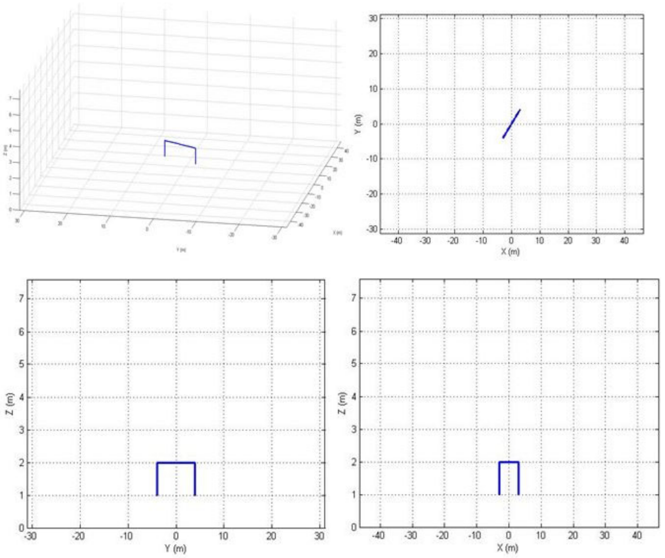

Figure 10: Pick-and-Place Trajectory Example

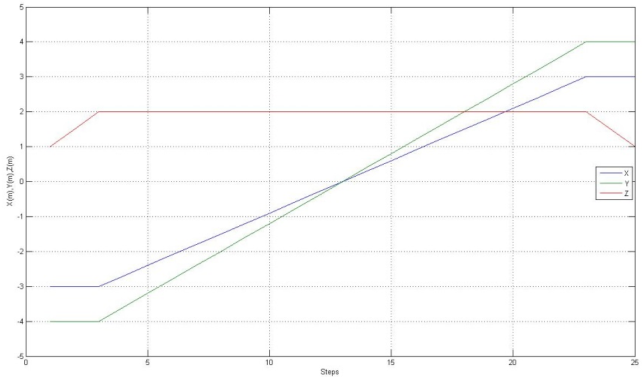

Figure 11: Cartesian coordinates vs. steps

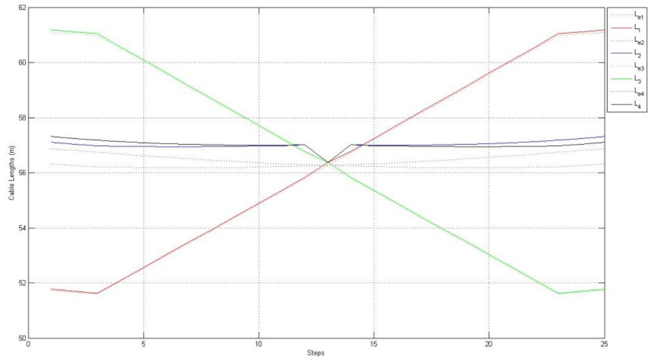

Figure 12: Straight-line cable lengths vs. steps

Similar to the snapshot example, the cable length differences between the cable sag and straight-line models are calculated, followed by cable length error computation. This is shown in Figure 13 and 14. 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 where high accuracy is not a prime requirement.

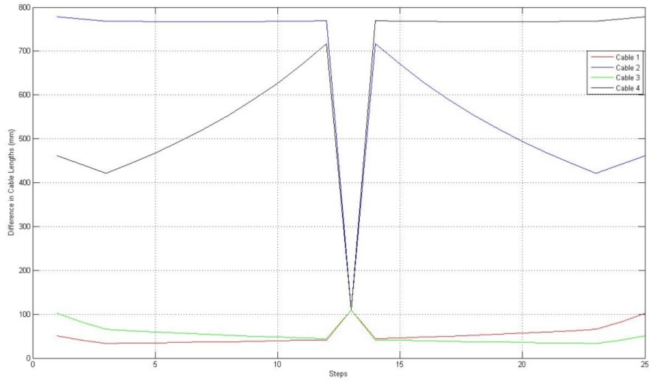

Figure 13: Difference in cable lengths vs. steps

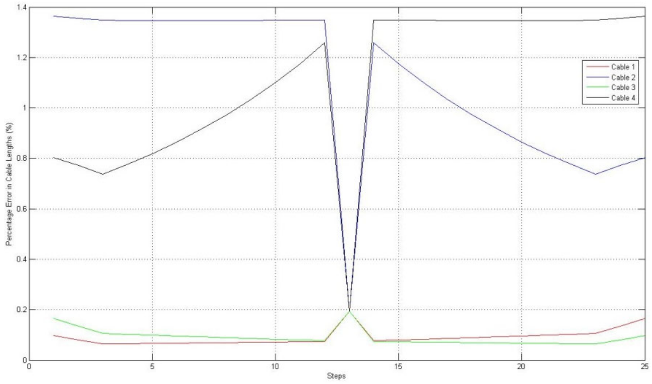

Figure 14: Percentage error in cable lengths vs. steps

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 (LP) and pseudoinverse (PI) methods $\sum T_{LPi} - \sum T_{PIi}$. The results are presented in Figures 15 and 16.

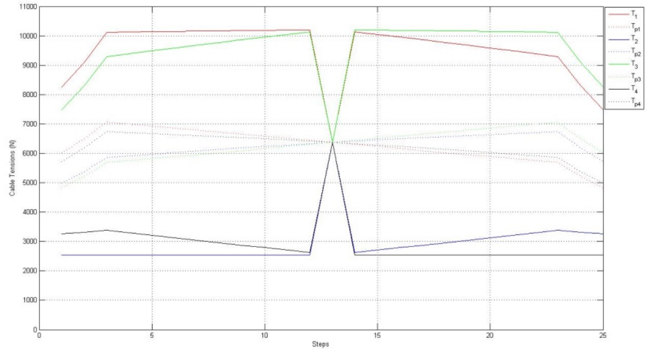

Figure 15: LP and PI Cable tensions vs. steps

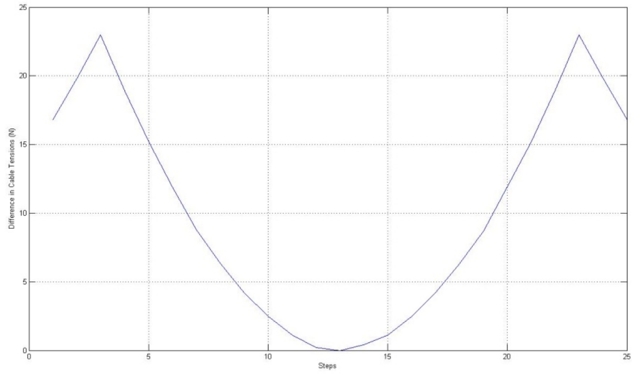

Figure 16: Difference in cable tensions between LP and PI

From Figures 9, 15, and 16, a straightforward observation is that the two methods (LP and PI) give different solutions for cable tensions except when the cable lengths are equal. Except for this case, the linear programming method gives a solution such that the overall cable tensions are less, when compared to the corresponding pseudoinverse solution.

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 feasible. A similar argument can be made for T_{max}. In this simulation, T_{max} was set to be +∞ to get an idea of the maximum tension that a particular configuration reaches.

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

There are a few issues associated with the use of the LP method that require attention. The LP approach at times gives an abrupt increase or decrease in the tension solutions, thus not resulting in smooth curves for trajectories (see Figure 15). 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 one of the cables. Regardless, a valid solution can be obtained by this method and research is being done in this field to get smoother results with less iterations. Borgstrom et al. [17] show 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 advisable. A summary of this discussion is provided in the form of a comparison chart in Table 5.

Table 5: Linear programming and pseudoinverse comparison

Linear Programming (LP)	Moore Penrose Pseudoinverse (PI)
Minimize the sum of the cable tensions; Min $(T_1+T_2+T_3+T_4)$	Minimizes the second norm of the cable tensions; Min ($\sqrt{T_1^2+T_2^2+T_3^2+T_4^2}$)
Can be applied for other objective functions.	Only one objective function possible
Iterative method	Closed form analytical solution possible
The overall cable tensions are relatively small	Cable tensions at the least can be equal to the LP solution
More flexible method	Less flexible method
Multiple solutions possible	Single solution results
MATLAB command – **linprog( )**	MATLAB command – **pinv( )**

Variation of 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 position $\{0 \quad 0 \quad 0\}^T$ and an arbitrary position $\{8 \quad -5 \quad 2\}^T$, these parameters were varied independently and the results are graphically presented in Figures 17 and 18; all results compare the straight-line to the sag cable models.

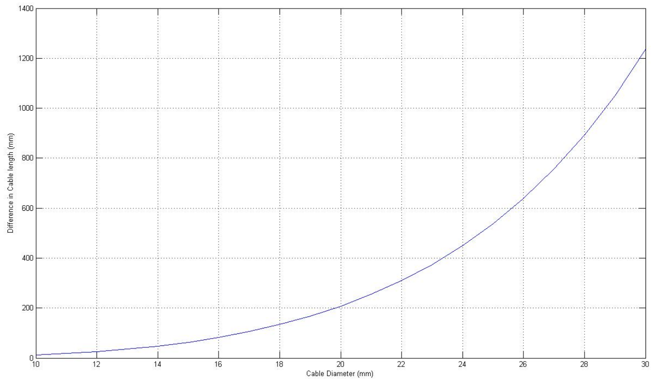

Figure 17: Difference in cable length vs. cable diameter

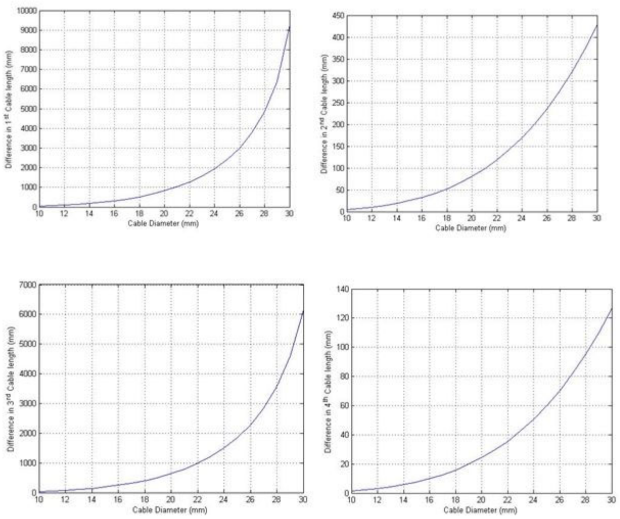

Figure 18: Difference in cable length vs. cable diameter

From 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 nature of the catenary equations, we expect a nonlinear increase in the difference in cable lengths when cable diameter and density is increased, as verified by the simulations of Figures 17 and 18. The trends for increasing cable density are very similar to increasing cable diameter and hence are not shown [18].

Another important parameter is the end-effector mass. This is of special importance since it may vary during the operation of a cable-suspended robot. The variation of difference in cable length between the cable sag and straight-line models with an increase in end-effector mass is shown in Figures 19 and 20.

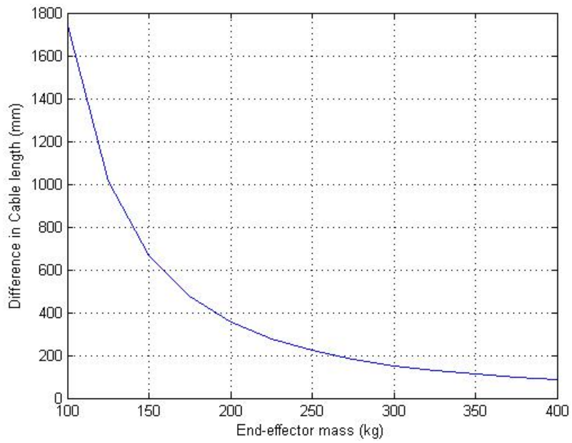

Figure 19: Difference in cable length vs. end-effector mass

For this case there is an inverse relationship, i.e. the cable lengths differences decrease as the end-effector mass increases. This makes sense since, for a given cable size, larger end-effector mass will dominate more and more relative to the cable mass, meaning the straight-line model becomes more and more accurate.

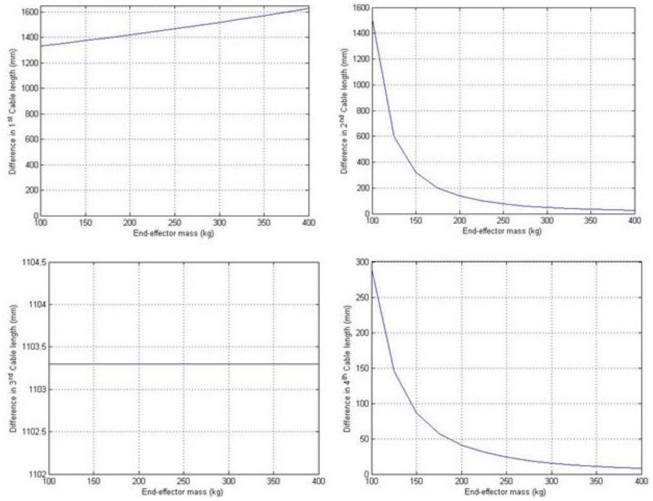

Figure 20: Difference in cable length vs. end-effector mass

An increase in end-effector mass has different effects on different cables for the arbitrary position. The reason for this is one of the limitations of the LP method: solutions tend to fall on the tension bounds. In case of the arbitrary position (Figure 20), the third cable solution falls on the lower bound, hence the variation in cable length 3 is constant. Cables 2 and 4 follow the same inverse trend of Figure 19, and the cable 1 length difference actually increases with increasing end-effector mass.

Effects of Footprint Dimensions

As the size of the robot footprint 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 footprint length to width constant (L/W = constant), the area was increased in steps from 1 to 6 acres and the difference in cable lengths was computed. As expected, the cable length difference increases with an increase in area as shown in Figure 21.

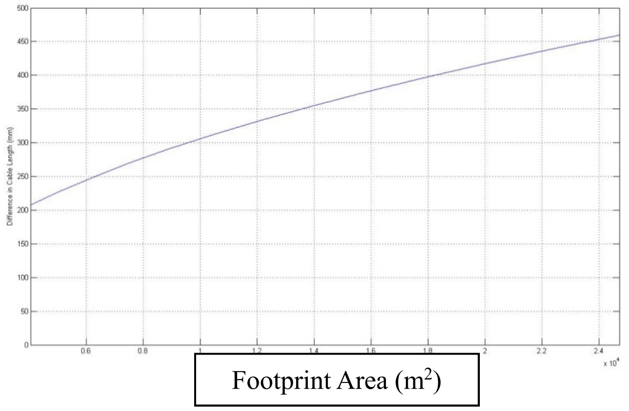

Figure 21: Difference in cable length vs. footprint area

Complimentary to the previous case, we next study the effects of variation of length to width (L/W) ratio, keeping the area constant. For the nominal position, at a constant tower height, the variation of the Euclidean norm length depends on the footprint length L and width W. By the Pythagorean Theorem, this is dependent on the term $\sqrt{L^2+W^2}$. 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 22.

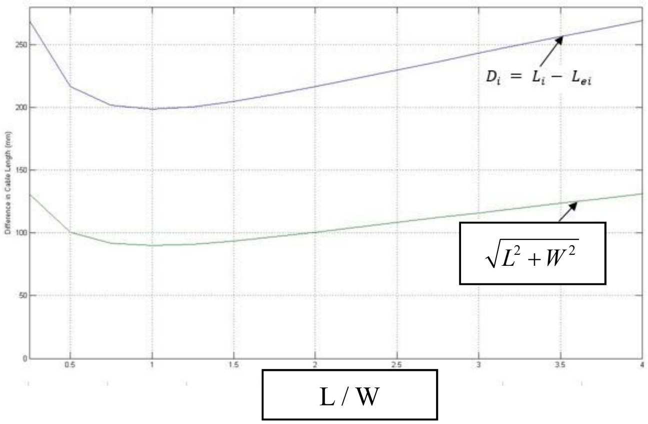

Figure 22: Difference in length (mm) vs. L/W (unitless)

Computational Considerations

The straight-line model has been used in most cable-suspended robot systems when compared to the cable sag model. One of the main reasons for this is its simplicity and an analytical model which is 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, 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-suspended robot system. Hence, understanding the computational complexities involved is important.

The catenary equations by themselves are highly nonlinear and are implicit. 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 investigate this issue, during the computation of cable lengths the number of iterations for both snapshot points and trajectory was recorded for the cable sag model. This information is presented in Figures 23 and 24. For comparison, the straight-line model requires no iteration, so the number of iterations for that case is always 1.

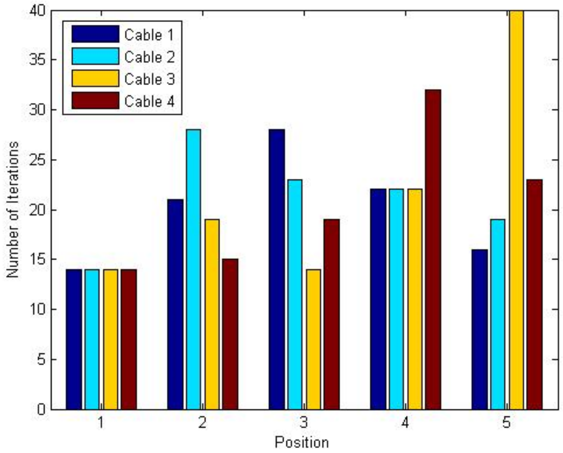

Figure 23: Number of iterations vs. position, snapshots

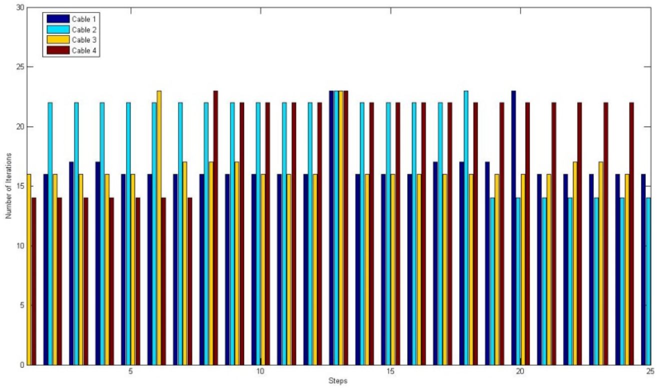

Figure 24: Number of iterations vs. steps, trajectory

There is no definitive prediction that can be made on the number of iterations for a different trajectory or snapshot example, but the examples shown above are representative. 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. Thus, the cable sag model, despite being an accurate model, suffers from increased computational requirement. A relative comparison between the straight-line model and cable sag model is shown in Table 6.

Solving the cable tension and length problems independently in separate steps (i.e. using the straight-line model) offers significant practical benefits. Firstly, it offers easy control system implementation, since ensuring positive cable tension is a necessary condition and cable sag computation can be circumvented if the corresponding error is within limits. Secondly, solving the steps separately greatly reduces the computational time. Additionally, if the steps are combined (i.e. using the cable-sag model) the problem becomes a constrained non-linear optimization problem (instead of a robust linear programming problem) which needs more sophisticated optimization routines and is not practical to implement in simple, cost-effective, real-time control system architectures.

Table 6: Straight-line and cable sag model comparison

Criteria	Straight-line model	Cable sag model
Governing equations	Euclidean norm between two points	Catenary equations
Type of model	Approximate model	Accurate model
Kinematics and Statics	Analytical solution for both; problems solved independently.	Both problems are coupled and there is no analytical solution.
Nature of solutions	Analytical	Numerical
Mass of the cable	Neglected	Included
Areas of application	Small scale robots and where accuracy is not a prime concern.	Any cable-suspended robot system, especially large outdoor systems.
Errors involved	Cable length computation errors	Iterative errors, truncation errors
Control system application	Straight-forward	Difficult

4. CONCLUSIONS AND RECOMMENDATIONS

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-suspended robots. The research also involved studying the effects of cable density, cable diameter, robot footprint size, and computational requirements involved.

Based on the results of the snapshot and trajectory examples (for a 1-acre footprint robot), 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 quantitative comparison results presented earlier, the straight-line model is preferred over the cable sag model at this particular large scale (1-acre footprint, 4047 m^2). In cases where the cable sag and errors are greater, the use of a Cartesian servo controller based on GPS sensing of the end-effector location is recommended.

Cable tension distribution is an 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 (the pseudoinverse-based method is much more common). An additional benefit of the LP method is to help in finding if a given cable tension range is acceptable for motion control of a cable-suspended robot system and is within the torque limitations of a winch / motor. Conversely, the simulation results could be used for appropriate choices in winch / motor design.

REFERENCES

[1] 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.

[2] L.L. Cone, 1985, "Skycam: An aerial robotic system", BYTE, October.

[3] 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.

[4] 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.

[5] 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.

[6] Irvine, H. M., 1992, Cable Structures, Dover Publications, Mineola, NY, ISBN 0486671275.

[7] Russell, J. C., and Lardner, T. J., 1997, "Statics experiments on an elastic catenary," J. Eng. Mech., 123(12), pp. 1322–1324.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] dit Sandretto, J.A., Trombettoni, G. and Daney, D., "Confirmation of hypothesis on cable properties for cable-driven robots", In Mechanisms and Machine Science, Vol. 7, pp. 85{93, Kluwer Academic Publishers, Santander, Spain, ISSN 22110984, 2013.

[15] Yuan, H., Courteille, E., and Deblaise, D., 2015, "Static and dynamic stiffness analyses of cable-driven parallel robots with non-negligible cable mass and elasticity", Mechanism and Machine Theory, 85: 64–81.

[16] 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.

[17] 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.

[18] D. Sridhar, 2015, "Mathematical Modeling of Cable Sag, Kinematics, Statics, and Optimization of a Cable Robot", April 29, advisor R.L. Williams II.