Design and Implementation of NIMS3D, a 3-D Cabled Robot for Actuated Sensing Applications

Per Henrik Borgstrom, Nils Peter Borgstrom, Michael J. Stealey, Brett Jordan, Gaurav S. Sukhatme, *Senior Member, IEEE*, Maxim A. Batalin, *Member, IEEE*, and William J. Kaiser, *Senior Member, IEEE*

***Abstract*—We present NIMS3D, a novel 3-D cabled robot for actuated sensing applications. We provide a brief overview of the main hardware components. Next, we describe installation procedures, including novel calibration methods, that enable rapid in-field deployability for nonexpert end users, and provide simulations and experimental results to highlight their effectiveness. Kinematic and dynamic analysis of the system are provided, followed by a description of control methods. We provide experimental results that illustrate tracking of linear and nonlinear paths by NIMS3D. Thereafter, we briefly present an example of an actuated sensing task performed by the system. Finally, we describe methods of improving energy efficiency by leveraging nonlinear trajectories and energy-optimal tension distributions. Experimental and simulated results show that energy efficiency can be improved significantly by using optimized parabolic trajectories. Furthermore, we provide simulation results that demonstrate improved efficiency enabled by optimal, least norm tension distributions.**

***Index Terms*—Cabled robots, environmental robots, field robots, parallel robots, robotics in hazardous fields.**

I. Introduction

CABLE-DRIVEN robots consist of computer-driven actuators that enable controlled release of cables. These cables, in turn, may support a wide range of end-effector systems. The actuators can be stationary or mobile and are positioned in the extremities of the robot workspace. The range of the end-effector is limited to the volume or plane defined by these actuators, although, in general, stability concerns further limit the range of operation.

The authors of [4] describe several advantages of cabled robots, including the following:

1) remote location of motors and controls;
2) rapid deployability;
3) potentially large workspaces;
4) high load capacity;
5) Reliability.

Because of these characteristics, cabled robots are ideal for many tasks, such as the handling of hazardous materials and disaster search and rescue efforts [5]. A balloon-cable-driven robot for use in search and rescue is developed in [6], but suffers from poor stability in windy conditions [7]. Additionally, several cabled robotic systems such as the SkyCam [8] and Cablecam [9] have found success in the fields of sports and entertainment. Similar platforms have been implemented for use as air vehicle simulators in development of sensing and control strategies [10]. Recently, cabled robotic systems have demonstrated critical capabilities for monitoring terrestrial ecosystems [11] and water resources with a focus on the characterization of contaminants [12]–[14]. The new cabled robotic systems reported here will provide significant advances in capability over these previous systems

There is much prior work in kinematic, static, and dynamic analysis of cabled robotic systems. Williams *et al.* provide analysis and simulations for planar cable-driven robots in [15]–[17], and detail hardware implementation and provide experimental results in [16]. Dynamic analysis of cable array robotic cranes is presented in [18] in the case of rigid cables and in [19] for flexible cables. Other work in design and control of fully constrained cable-driven robots includes the WARP [20] and FALCON [21] systems.

Prior art in trajectory control of underconstrained cable robots is somewhat limited. The authors of [22] employ inverse dynamics and feedforward and feedback control methods to provide trajectory control of an incompletely constrained wrench-type cable robot with mobile actuators. Control is achieved through a proportional-derivative (PD) controller and a precompensator. The authors of [23] and [24] provide simulation and experimental results of two closed-loop asymptotic control mechanisms based on Lyapunov design techniques and feedback linearization, respectively.

In this paper, we detail the design and implementation of NIMS3D, an underconstrained 3-D cabled robot for actuated sensing applications. NIMS3D can assume a four-cabled form, as in Fig. 1, or a three-cabled form, as in Fig. 2. We have developed algorithms for both deployment configurations and provide extensive simulation results. Additionally, we have implemented and deployed a three-cabled system, and provide experimental results to highlight the performance of our methods for this configuration. Specifically, the contributions of this paper are as follows.

Manuscript received May 20, 2008; revised September 15, 2008. First published March 10, 2009; current version published April 3, 2009. This paper was recommended for publication by Associate Editor I.-M. Chen and Editor F. Park upon evaluation of the reviewers' comments. This paper was presented in part at the IEEE/RSJ International Conference on Intelligent Robots and Systems [1], [2] and the IEEE International Conference on Robotics and Automation [3].

P. H. Borgstrom, N. P. Borgstrom, M. J. Stealey, and W. J. Kaiser are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: henrik@ee.ucla.edu; nborgstr@ucla.edu; stealey@ucla.edu; kaiser@ee.ucla.edu).

B. Jordan and M. A. Batalin are with the Center for Embedded Networked Sensing, University of California, Los Angeles, CA 90095 USA (e-mail: bjordan5@gmail.com; maxim.batalin@cens.ucla.edu).

G. S. Sukhatme is with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: gaurav@usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2009.2012339

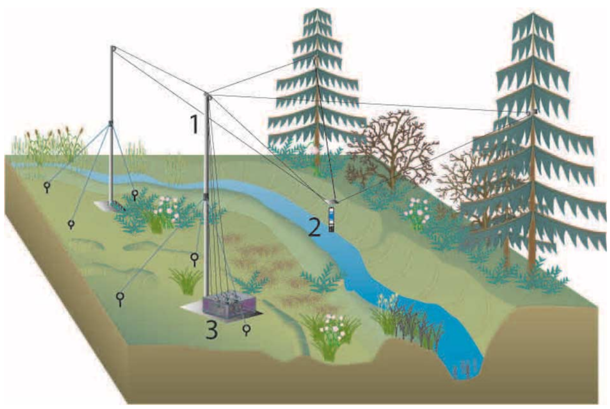


Fig. 1. Schematic of a four-cabled NIMS3D deployment showing: 1) infrastructure, consisting of poles and pulleys; 2) a generic sensor package; and 3) the motor control box.

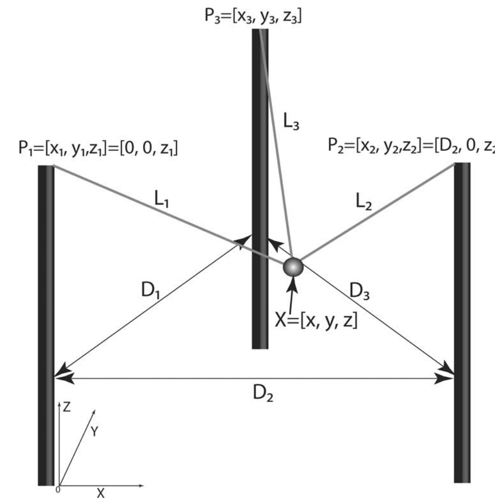


Fig. 2. Schematic of a three-cabled NIMS3D deployment showing the reference frame, end-effector location $\vec{X} = [x,\ y,\ z]^T$, the cable origins, P_i, the horizontal distances D_i, and cable lengths $L_i,\ i = 1, 2, 3$.

1) Description of a novel cabled robotic system.
2) Installation and calibration procedures to enable rapid in-field deployments.
3) Example 3-D actuated sampling experiments.
4) Novel energy-efficient trajectory-generation methods to reduce power requirements and prolong deployment lifetime.
5) Efficient computation of least norm tension distributions for systems wherein the number of cables exceeds the number of DOFs by one.

The remainder of this paper is as follows: In Section II, we provide a hardware overview of NIMS3D. In Section III, we describe rapid deployment procedures, including a novel calibration scheme. In Section IV, the forward and inverse position and velocity kinematics of NIMS3D are derived, while the dynamics are considered in Section V. In Section VI, trajectory control methods are presented and trajectory tracking results are provided. In Section VII, we provide results from an example of an actuated sensing application. In Section VIII, we present novel methods geared toward improving energy efficiency of NIMS3D by leveraging nonlinear trajectories and energy-optimal tension distributions. In Section IX, we conclude the paper and briefly describe current research objectives.

II. System Overview

The NIMS3D system architecture and the methods described here enable rapid deployment in diverse environments ranging from indoor to remote natural environment applications where sensors and robotic end-effectors must be precisely controlled in a large volume. Fig. 1 shows a schematic diagram of a four-cabled NIMS3D system deployed in an environmental monitoring application. The system comprises three components: 1) the infrastructure, consisting of poles and pulleys; 2) a generic node platform, on which a variety of sensors can be mounted; 3) the motor control box, which controls the spooling of the suspension cables. The cables, which are nonelastic, low-mass, high tensile strength fishing lines, all originate from a single motor control box and connect to the node platform via pulleys. By virtue of this common origin, all cables, controllers, motors, and power requirements are isolated to a single area of the deployment site. This allows for easy access to all wired components and enables flexibility in deployment configurations. Furthermore, Fig. 2 shows a geometric schematic of a three-cabled NIMS3D arrangement showing the reference frame, end-effector location $\vec{X} = [x,\ y,\ z]^T$, cable origins P_i, horizontal distances D_i, and cable lengths $L_i,\ i = 1, 2, 3$.

While the reader is referred to [1] for a detailed hardware implementation of NIMS3D, one aspect of its design is particularly pertinent to the control algorithms presented in this paper: precise encoder-enabled control of cable length. Traditionally, motor position can be controlled by mounting an encoder on the motor output shaft and employing feedback to ensure correct positioning. However, the critical issue in this system is not controlling motor position but rather cable length. Because reeling cable in and out changes the spooling radius, the relationship between motor position and line release is nonlinear and becomes very difficult to predict. Previous work [25] has employed specially designed winch systems in which the cable is wound onto a threaded drum that moves laterally as the motor rotates. This solution requires significant hardware customization and also is severely limited in the length of cable that can be stored on the winch. The solution developed for the NIMS3D system reported here is to introduce precise cable tracking enabled by optical encoders mounted on idler wheels that rotate freely as the cable passes over them. Since the diameter of the idler wheels is constant, the amount of released cable is directly proportional to the encoder output value and is thereby directly controllable.

As described in Section I, NIMS3D can be deployed in three- or four-cabled configurations. Four-cabled configurations increase the size of the workspace and enable quadrilateral deploymentsthat are desirable for some target applications. Although the two configurations are fundamentally similar, the addition of a fourth cable results in the existence of an infinite set of feasible tension distributions, as discussed in Section VIII-B.1.

III. Deployment Procedure

NIMS3D has been designed to enable rapid in-field deployment by small teams with limited training. The deployment procedure consists of 1) erecting the infrastructure and 2) calibration. In this section, we describe the deployment process.

A. Infrastructure

The design of NIMS3D enables flexibility in selecting infrastructure components. All that is required are pulleys mounted in the perimeter of the desired workspace and means to stabilize the motor control box. We have performed several deployments in which we have used various forms of infrastructure to support the pulleys. For example, in one deployment, pulleys were attached to tree limbs, as depicted in Fig. 1. In other deployments, we have used poles supported by guy-wires. In each of these cases, the time required for setting up the infrastructure was less than 1 h. In our trials, the horizontal distances between cable origins were of the order of 10 m. For larger deployments, infrastructure installation remains entirely practical, although this installation may require additional deployment time. Furthermore, although the use of naturally occurring infrastructure, such as terrain features or trees, may expedite the deployment process, the compliance of these structures may introduce error in calibration and during system operation. Thus, it is left to the discretion of deployment teams to determine whether the potential loss of accuracy is an acceptable tradeoff for reducing installation time.

B. Calibration

Once the infrastructure has been erected, the cables are spooled through the pulleys and connected in the middle of the workspace. Thereafter, the only remaining task is calibration. In order to enable rapid deployability for NIMS3D, it is critical that in-field deployment teams be able to quickly and accurately calibrate the system for a given configuration. Calibration consists of measurement of the configuration-specific parameters, namely the locations of the points of origin of all N cables, $\{P_i = [x_i \; y_i \; z_i]^T \; : \; i = 1 \dots N\}$. Ideally, calibration would not require expensive measurement or surveying equipment and would be simple enough for nonexpert end users. It should be noted that the calibration procedures developed in this section are applicable to both three-cabled and four-cabled configurations.

1) Calibration Methods: While there is little previous work aimed at expediting rapid calibration of in-field parallel mechanisms, the authors of [5] propose the following solution for calibrating a concept rescue robot driven by three cranes: given that there are three locations in the workspace whose positions relative to one another are accurately known, move the end-effector to each of these calibration points and record the length of each cable to that point. The cables are measured using rotary encoders that track their release, which requires no extra equipment. It is evident that, upon completing the measurements, the distance between each cable origin and three points is known, with some measurement noise. The location of each cable origin is then given by the intersection of the three spheres centered at the three measurement points and the, respective, measured distances. Because the authors suggest the use of a global positioning system (GPS) to determine the locations of the three calibration points, we hereafter refer to this method as the GPS method.

This proposed calibration scheme is problematic for a number of reasons. First, accurately measuring the relative locations of three points in 3-D space requires some form of precise ranging equipment or GPS, and becomes more difficult in the absence of a reliable reference plane. Furthermore, deployments might occur in areas with poor GPS connectivity, making reliable localization of the three calibration points more difficult. Second, maneuvering the end-effector to the three calibration points is difficult and must be done manually. Until calibration is complete, it is not possible to define the kinematic mapping from joint space to tip space, and the operator is required to manually steer the node by releasing or reeling in cables. This may require substantial operating time and necessitates some training for the user. Finally, as will be shown in Section III-B.2, the GPS method is highly susceptible to measurement noise and yields poor accuracy.

Consider the example three-cabled configuration shown in Fig. 2. The calibration method proposed in this paper is as follows: a plumb line is suspended from each of the cable origins. Using a laser rangefinder, the horizontal distance D_1, D_2, D_3 between the plumb lines can be measured quickly and precisely. Without loss of generality, x_1 and y_1 can be set to zero, and the horizontal vector from this point toward the plumb line suspended from P_2 can be taken as the x axis. Thus, $P_1 = [0 \, 0 \, z_1]^T$ and $P_2 = [D_2 \; 0 \; z_2]^T$, as illustrated in Fig. 2. The x_3 and y_3 can be found easily and are given by

$$x_3 = \frac{D_1^2 + D_2^2 - D_3^2}{2D_2} \tag{1}$$

$$y_3 = \sqrt{D_1^2 - x_3^2}. \tag{2}$$

In the case of a four-cabled deployment, the x and y coordinates of the fourth point can be easily found through similar range measurements.

In order to determine the heights of the cable origins, a reference plane must be determined. In many cases, there is an obvious and readily available ground plane. For indoor deployments, the floor is taken as the ground plane. In this case, if there is no obstruction between the cable origins and the floor, the heights can be measured directly by releasing enough cable to just touch the floor and then measuring their length via the optical encoders that track cable release. In other cases, there is an obvious reference plane, but it is not immediately accessible, as there may be obstructions between the floor and the points P_i. Alternatively, there are many outdoor applications where similar problems arise. For example, one of the potential applications for NIMS3D is in water monitoring. In such deployments, the cable origins might be above land that may not be level, while a majority of the workspace is above water, which provides a potential reference plane. In absence of a reference plane, one can be created by using a laser level or by stringing a taut horizontal cable across the workspace. The question then becomes how to leverage this ground plane for calibration without having to define and measure any particular calibration points in it.

Consider an arbitrary set of M calibration points $\{P_{c_j} = [x_{c_j}\ y_{c_j}\ 0]^T\ :\ j = 1 \ldots M\}$ that lie in the $z = 0$ plane somewhere within the workspace. If the node is moved to this set of points, and the length of each cable is recorded, then we have that

$$L_{ij} = \|P_i - P_{c_j}\| \quad i = 1 \ldots N,\ j = 1 \ldots M \tag{3}$$

where L_{ij} is the length of the ith cable to the jth calibration point. Each calibration point introduces N nonlinear equations and two unknown variables, x_{c_j} and y_{c_j}. Moreover, $z_{1 \ldots N}$, the heights of the N cable origins, are also unknown. Thus, we have $M \times N$ equations in $2M + N$ variables. For a three-cable system, $N = 3$, and the system of equations is adequately constrained for $M \geq 3$. For a four-cable configuration, $N = 4$, and the system of equations is adequately constrained for $M \geq 2$.

The problem of finding $z_{1 \ldots N}$ that are most consistent with the measured cable lengths can be expressed as the minimization of the following cost function:

$$\sum_{j=1}^{M} \sum_{i=1}^{N} (L_{ij}^2 - \|P_i - \hat{P_{c_j}}\|^2)^2 \tag{4}$$

where $\hat{P_{c_j}}$ is the estimated location of the jth calibration point for a given set of $z_{1 \ldots N}$. Computation of $\hat{P_{c_j}}$ cannot be done by the normal forward kinematic method, which computes the intersection of the three spheres centered at the cable origins with radius equal to the respective cable lengths. This is because P_{c_j} is constrained to the $z = 0$ plane, and applying forward kinematics will generally give solutions that are not in this plane. Instead, the location of the calibration points is found using a least-squares estimator, which is described in, for example, [26]. The least-square estimator computes the point in the $z = 0$ plane that is most consistent with the set of measured cable lengths.

We compute z^*, the optimal set of $z_{1 \ldots N}$ as follows:

$$z^*_{1 \ldots N} = \left| \underset{z_{1 \ldots N}}{\operatorname{argmin}} \sum_{j=1}^{M} \sum_{i=1}^{N} (L_{ij}^2 - \|P_i - \hat{P_{c_j}}\|^2)^2 \right| \tag{5}$$

This optimization can be performed very quickly using any available numerical optimization software, such as MATLAB's $fminsearch$. Taking the absolute value in (5) is necessary because the cost function is symmetric about the $z = 0$ plane, and reflections may occur for some datasets.

There are a number of distinct advantages associated with this calibration method as compared to the GPS method of [5]. First, the set of calibration points is completely arbitrary and the relative positions of the calibration points do not have to be measured. Second, the user who is performing the calibration is not required to maneuver the node to specific locations, as any position in the ground plane is a valid calibration point. This is a critical improvement because of the difficulty involved in manually positioning the node. Whereas it is quite challenging to move the node to a specific location, it is straightforward to move it to a specific height: releasing any cable causes the node to move downward, and shortening any cable causes the node to rise. Finally, as will be discussed in the following section, our proposed calibration scheme produces more accurate results than the GPS method.

TABLE I
EXPERIMENTAL THREE-CABLED NIMS3D CONFIGURATION

Cable Origin	1	2	3
$x(m)$	0	8.46	3.96
$y(m)$	0	0	5.49
$z(m)$	2.36	2.77	3.10

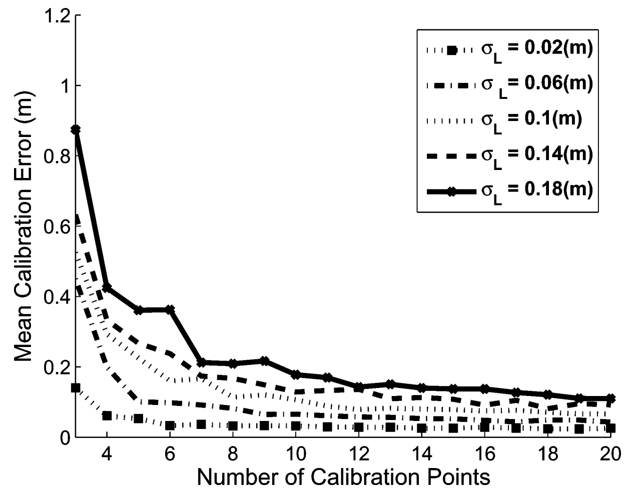


Fig. 3. Mean calibration error shown against the number of calibration points for various σ_L.

2) Calibration Simulations: In order to experimentally quantify the performance of our calibration scheme, we ran a set of simulations on a three-cabled system. A virtual indoor deployment was considered with parameters as shown in Table I. Cable length measurements were corrupted with zero-mean Gaussian noise with standard error σ_L ranging from 0.02 m to 0.20 m. The number of calibration points M ranged from 3 to 20. For all tests, the standard measurement error of $D_{1,2,3}$ was taken to be 0.02 m. Plots of the resulting data are shown in Fig. 3.

In order to compare our calibration scheme with the GPS method, we ran a similar set of calibration experiments using both methods. A completely fair comparison is not possible because the error in measuring the relative positions of the calibration points in the GPS method is application-specific. In our tests, we set this error to zero, but corrupted cable length measurements in the same manner as earlier. Fig. 4 shows plots of mean calibration error for both methods for varying σ_L. It is apparent that as M increases beyond 5 toward 10, the performance of our method begins to surpass that of the GPS method. Thus, even in the case of perfect localization of the three calibration points required for the GPS method, our calibration scheme outperforms the GPS method for reasonably large M. The improvement in performance becomes more substantial when the error in measuring the relative position of calibration points in the previous method is considered. It is important to recognize that, even for large M, the time associated with our calibration method, which is dominated by the time spent in visiting M

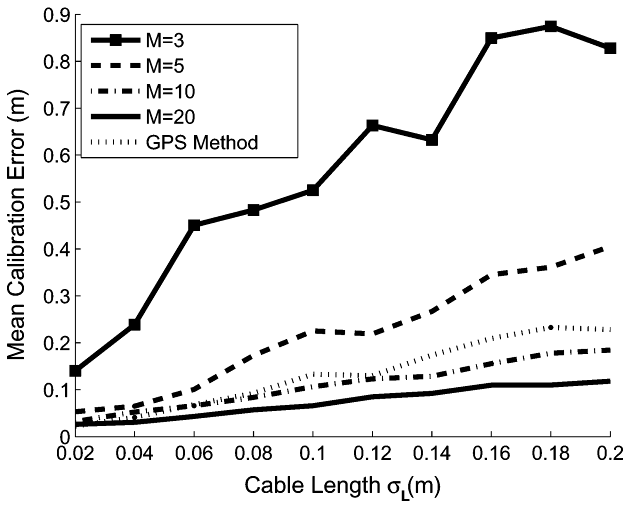


Fig. 4. Mean calibration error shown against σ_L for various M.

TABLE II
CALIBRATION RESULTS

Cable Number	1	2	3
$x(m)$	0	4.03	8.63
$y(m)$	0	5.90	0
$z^*(m)$	2.44	3.07	2.73
$z_{meas}(m)$	2.46	3.08	2.80
Relative Error(%)	.8	0.3	2.5

calibration points in the ground plane, is much shorter than the time associated with accurately localizing three points within the workspace and moving the node precisely to these locations.

3) Experimental Calibration Results: In order to experimentally verify the accuracy of the proposed calibration method, we performed a calibration for an indoor deployment of a three-cabled system. Because the system was deployed indoors, the floor could be used as a ground plane. The number M of calibration points was 43. We considered such a large number of points in order to enable statistical evaluation of the convergence of (5) with increasing M, as is discussed next. Furthermore, there were no obstructions between the cable origins and the floor, so the heights of the origins could be measured directly by means of a laser rangefinder. Thus, the calculated values of z^* can be compared to directly measured values. However, we have no equipment to enable more precise measurement of the x and y coordinates of the points of origin. Therefore, we assume that the values calculated using (1) and (2) are valid, as the laser rangefinder used to measure $D_{1,2,3}$ is accurate within ± 1.5 mm [27]. As shown in Table II, the resulting values of z^* are within a few centimeters of the directly measured values, z_{meas}.

In order to evaluate the convergence properties of (5) with respect to increasing M, we randomly selected m of the M data points for $m = 1 \ldots M-1$ and calculated the corresponding cable origin heights, $z^*(m)$. This was repeated 100 times for each m, and the standard deviation $\sigma_{\|z^*\|}$ of the distribution of $\|z^*(m)\|$ was calculated. The resulting data is plotted in Fig. 5. Clearly, (5) converges rapidly with increasing M. For values of $M \geq 9$, $\sigma_{\|z^*\|}$ is less than 0.01 m. Thus, M does not need to be large, which substantially reduces the calibration time.

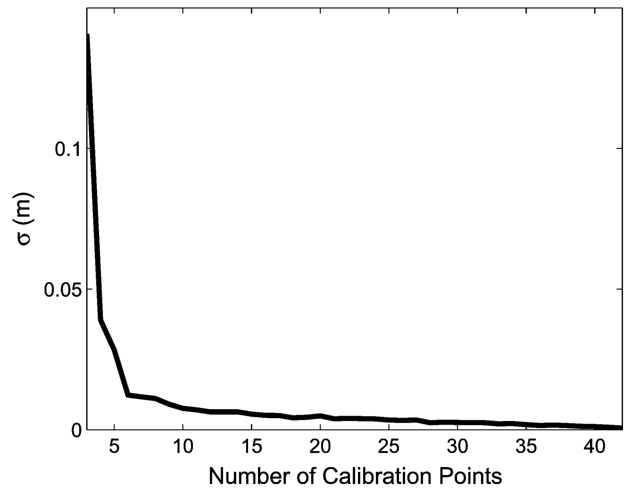


Fig. 5. Standard deviation ($\sigma_{\|z^*\|}$) of the distribution of cable origin heights ($\|z^*(m)\|$) shown against the number of calibration points.

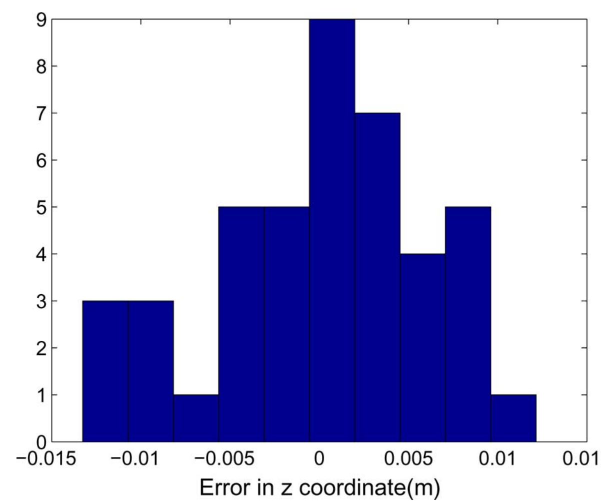


Fig. 6. Distribution of the errors in the z coordinates of calibration points.

Once z^* has been computed, the locations of the calibration points can be found via forward position kinematics routines described in Section IV. While their x and y coordinates were never directly measured, the points should all reside in the $z = 0$ plane. Thus, one way to verify the accuracy of the calibration is to examine their computed z coordinates. For this experiment, the distribution of z coordinates for the calibration points has a mean of $-38\,\mu$m and a standard deviation of 6.1 mm. This distribution is plotted in Fig. 6. The close agreement with the expected values indicates that the results of the calibration are accurate.

IV. KINEMATICS

For the sake of completeness, we derive kinematic properties of NIMS3D in this section. These kinematics are applicable to both three-cabled and four-cabled configurations.

A. Position Kinematics

As is usual for parallel manipulators, treatment of reverse kinematics for NIMS3D is less complex than that for forward kinematics. The inverse position kinematics, which yield the set of cable lengths $L_i,\ i = 1 \ldots N$ compatible with a given node position $\vec{X} = [x\ y\ z]^T$, are given by

$$L_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2},\ \ i = 1 \ldots N. \quad (6)$$

The forward position kinematics compute the end-effector position $\vec{X}$ corresponding to a set of cable lengths. This consists of computing the intersection points of three spheres centered at the cable origins, P_i, with radii equal to the corresponding cable lengths, $L_i,\ i = 1, 2, 3$. To solve the forward kinematics, we begin by temporarily shifting our origin upward to P_1. The three spheres can now be defined as follows:

$$x^2 + y^2 + z^2 = L_1^2 \quad (7)$$

$$(x - x_2)^2 + y^2 + (z - z_2)^2 = L_2^2 \quad (8)$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = L_3^2. \quad (9)$$

Subtracting (7) from (8) and solving for x in terms of z yields

$$x = \alpha z + \beta \quad (10)$$

where $\alpha = -\frac{z_2}{x_2}$ and $\beta = \frac{z_2^2 + x_2^2 + L_1^2 - L_2^2}{2x_2}$. Subtracting (7) from (9) and substituting (10) results in

$$y = \lambda z + \gamma \quad (11)$$

where

$$\lambda = -\frac{\alpha x_3 + z_3}{y_3}$$

$$\gamma = \frac{L_1^2 - L_3^2 + x_3^2 + y_3^2 + z_3^2 - 2\beta x_3}{2y_3}.$$

Substituting (10) and (11) into (7) results in the following quadratic expression for z:

$$(\alpha^2 + \lambda^2 + 1)z^2 + 2(\alpha\beta + \gamma\lambda)z + (\beta^2 + \gamma^2 - L_1^2) = 0. \quad (12)$$

This can readily be solved, and solutions for x and y follow. Thereafter, the initial reference shift is reversed.

B. Velocity Kinematics

The inverse velocity kinematics for the ith cable are found by taking the partial derivatives $\frac{\partial L_i}{\partial \vec{X}_j},\ \ j = 1, 2, 3$. This yields

$$\frac{\partial L_i}{\partial x} = \frac{x - x_i}{L_i}$$

$$\frac{\partial L_i}{\partial y} = \frac{y - y_i}{L_i}$$

$$\frac{\partial L_i}{\partial z} = \frac{z - z_i}{L_i}. \quad (13)$$

Thus, the inverse velocity kinematics can be expressed as follows:

$$\begin{bmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \end{bmatrix} = \begin{bmatrix} \vec{l}_1^T \\ \vec{l}_2^T \\ \vec{l}_3^T \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (14)$$

where $\vec{l_i}$ is a unit vector directed from the ith cable origin toward the cable insertion. Equation (14) can be written as $\dot{L} = J\dot{X}$, where J is the inverse Jacobian matrix. Since J is a 3×3 invertible matrix, it follows directly that $\dot{X} = J^{-1}\dot{L} = B\dot{L}$, where $B = J^{-1}$ is the forward Jacobian matrix.

V. Dynamics

Although the dynamics of three-cabled NIMS3D configurations are straightforward, we provide them in this section for the sake of completeness. Addition of a fourth cable results in a nonsquare Jacobian matrix, which complicates matters slightly. This is discussed in Section VIII-B.1.

Using a point-mass approximation, we have

$$m(\ddot{X} + \vec{g}) = \sum_{i=1}^{3} T_i = \Lambda\vec{T} \quad (15)$$

where $\vec{T} = [T_1 \quad T_2 \quad T_3]^T$ is a vector of cable tensions, $\vec{g} = [0 \quad 0 \quad g]^T$ is the gravity vector, and Λ is a pose-dependent structure matrix whose ith column is given by $-\vec{l_i}$. Thus, as expected, $\Lambda = -J^T$. The dynamics then become

$$\ddot{X} = \frac{1}{m}\Lambda\vec{T} - \vec{g} \quad (16)$$

where m is the mass of the end-effector. Note that the end-effector is approximated as a point-mass.

One of the primary characteristics that differentiate cable-driven parallel mechanisms from those with rigid arms is the inability of cables to exert push forces. This dramatically impacts the range of operation of cabled systems. Previous work [28] has suggested the use of a force feasible workspace (FFW) concept. The FFW is the range of operation within which the end-effector can apply a given desired force set, which is taken to be a sphere. The authors of [29] extend notions of manipulability ellipsoids to the wire-driven case. However, NIMS3D is not intended to exert forces on external objects, so these notions of determining range of operation are not very useful. Instead, we ensure that the end-effector never assumes a pose such that any cable tension falls below $T_{\min}$, the lower tension limit, or above $T_{\max}$, the upper tension limit. In order to ensure that cable tensions remain within these bounds, (16) is used to check desired trajectories for potential tension violations before execution.

VI. Trajectory Control

In our model of NIMS3D, the following purely kinematical representation is adopted:

$$\vec{X}(n+1) \approx \vec{X}(n) + B(\vec{X}(n))\vec{V}(n)T_S \quad (17)$$

where $\vec{V(n)}$ is a vector of motor velocities, T_S is the period of the discrete system, and B is the forward Jacobian matrix described in Section V. The removal of dynamic considerations in this model is valid as long as all cable tensions are positive. Thus, as long as appropriate trajectories are used, the purely kinematical representation holds.

During each iteration of the control loop, the following steps are executed.

1) Estimate current cable lengths based on previous cable lengths and motor velocities.
2) Perform forward kinematics.
3) Compute $B(\vec{X}(n))$.
4) Use an appropriate controller to select $\vec{V}(n+1)_{\text{desired}}$.
5) Transmit $\vec{V}(n+1)_{\text{desired}}$ to the motors, which, as a form of handshaking, return their current encoder counts.

In Section VI-A, the motor model used to estimate cable lengths in step 1 is presented, and in Section VI-B, the optimal receding horizon control law used in step 4 is presented.

A. Motor Modeling

Ideally, all motors would instantly transition from $\vec{V}(n)$ to $\vec{V}(n+1)_{\text{desired}}$. However, this would require an impractical zero delay and corresponding infinite acceleration. There are delays associated with RS232 communication between control computing systems and motor systems and with decoding by the motor control systems. Additionally, the motors accelerate with a finite acceleration. Therefore, in step 1 of the algorithm shown earlier, the first-order estimate that the motors release cable at exactly the desired rate is unrealistic.

In order to enable more accurate estimation of the current cable lengths, the actual velocity at which a motor releases cable during an iteration, $\hat{v}(n)$, is modeled as follows:

$$\hat{v}(n+1) = \phi_1 v(n+1)_{\text{desired}} + \phi_2 \hat{v}(n). \tag{18}$$

That is, the actual velocity of cable release during an iteration is a function of the desired velocity and the actual velocity during the previous iteration. These two factors are weighted by $\phi_{1,2}$. The task now becomes to determine these weightings. An online training algorithm has been created to this end by which the following matrices are populated for each motor:

$$\bar{V} = \begin{bmatrix} v_{n_{\text{desired}}} & \hat{v}_{n-1} \\ \vdots & \vdots \\ v_{n-\eta_{\text{desired}}} & \hat{v}_{n-\eta-1} \end{bmatrix}, \text{ for some } \eta \geq 1$$

$$\bar{\Psi} = \begin{bmatrix} \hat{v}_n \\ \vdots \\ \hat{v}_{n-\eta} \end{bmatrix}.$$

The problem of finding the least-squares optimal weightings $\Phi^* = [\phi_1 \ \phi_2]^T$ then becomes

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \left[\bar{V}\Phi - \bar{\Psi}\right]^T \left[\bar{V}\Phi - \bar{\Psi}\right]. \tag{19}$$

The solution to (19) is well known and is given by

$$\Phi^* = \left[\bar{V}^T \bar{V}\right]^{-1} \bar{V}^T \bar{\Psi}. \tag{20}$$

TABLE III
AVERAGE CABLE LENGTH ESTIMATION ERRORS

	Average Error(cm)	STD(cm)
$\Phi = [1\ 0]^T$	1.50	1.23
$\Phi_{Optimal}$	1.23	1.10

Once the system has been in operation sufficiently long to populate these matrices, the estimate of cable length is made based on the current optimal weighting factors $\phi_{1,2}$. $\bar{V}$ and $\bar{\Psi}$ are continuously updated in a sliding window manner such that only the η most recent values are considered in determining $\phi_{1,2}$. Thus, the weighting factors are sensitive to changes in the system and adjust accordingly within a period of η iterations. Table III shows average errors in cable length estimation during a typical run for the case where $\Phi = [1\ 0]^T$ and for the least-squares optimal weightings. The optimal case reduces error by approximately 18%.

B. Feedback Control Law

The controller used to calculate the feedback matrix in NIMS3D is an optimal fixed-endpoint fixed-time control law derived in [30] that minimizes the cost functional given by $\sum_{k=n}^{n+N-1} x_k^T Q_k x_k + u_k^T R_k u_k$, where Q and R are positive-definite symmetric weighting matrices. A slight adaptation is made to this control law, in that the desired state is taken to be $\vec{X}(n+\Delta)_{\text{desired}} - \vec{X}(n)$ for some integer $\Delta \geq 1$. Thus, the control law yields the optimal feedback gain to bring the node to the point on the trajectory corresponding to the $n+\Delta\text{th}$ iteration in Δ iterations. Selection of Δ is a tradeoff between stability and responsiveness in tracking trajectories. Responsiveness improves with small Δ, whereas stability improves with large Δ.

C. Trajectory Tracking Results

We performed a series of trajectory control experiments on a physical three-cabled NIMS3D system tracking a piecewise linear trajectory. Because of the difficulty associated with precisely measuring end-effector location, we use forward kinematics to compute position. A trace of the x-coordinate of the end-effector is shown in Fig. 7. In addition to linear trajectories, a more complex trajectory was used to demonstrate path tracking capabilities. This consisted of a spiral that rose 1.125 m while traversing circles with radius increasing constantly from 0 to 0.60 m. A 3-D plot of the resulting trajectory is shown in Fig. 8, and a time-lapse image of the node executing the trajectory is shown in Fig. 9. Further trajectory tracking results are available in [2].

Because these experiments rely on forward kinematics to determine end-effector position, the effects of pole deflection and cable extension are not considered. However, experimental observation of these effects was done. Here, a series of maneuvers by the NIMS3D system was performed where the end-effector followed a commanded trajectory immediately adjacent to the reference ground plane. The close proximity of this trajectory to the ground plane afforded high sensitivity for observation of the effects of errors. Both pole deflection and cable extension would result in downward positioning error, manifested in the

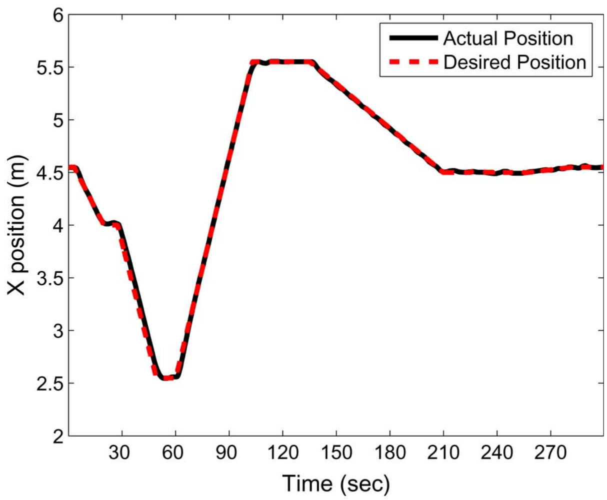


Fig. 7. Trace of x-coordinate for tracking piecewise linear trajectory.

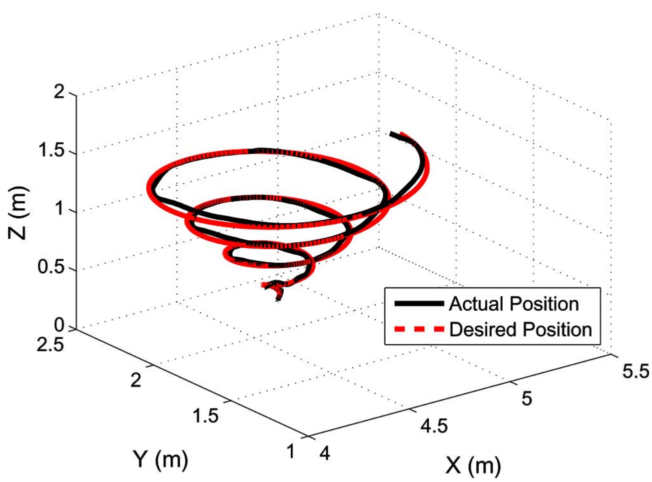


Fig. 8. Spiral trajectory tracking experiment.

form of a downward deflection of the end-effector that our experimental system was developed to detect. These downward deflections could not be observed, indicating that, if cables and infrastructure are sufficiently stiff, the effects of cable stretch and pole deflection are relatively small.

VII. Actuated Sensing Applications

The intended purpose of NIMS3D is to maneuver a generic sensor node throughout its span. The system has been shown to enable accurate positioning within its range of operation, and a variety of sensors have been deployed indoors. In this section, we briefly present results from topographical mapping experiments performed with a three-cabled system. The reader is directed to [1] for more examples of actuated sensing enabled by NIMS3D. In the topographical mapping experiment, an artificial topography was created and mapped using a downward facing rangefinder. An idealized contour plot enables qualitative evaluation of these topographical plots. The scanned area contained several object forms intended to verify mapping and reconstruction capability. This included cylindrical objects and rectangular prisms, one of which had a rectangular cavity at one corner. Parts of these objects were outside of the system's span and are therefore clipped in the scans. Plots of the experimental and idealized data are shown in Fig. 10.

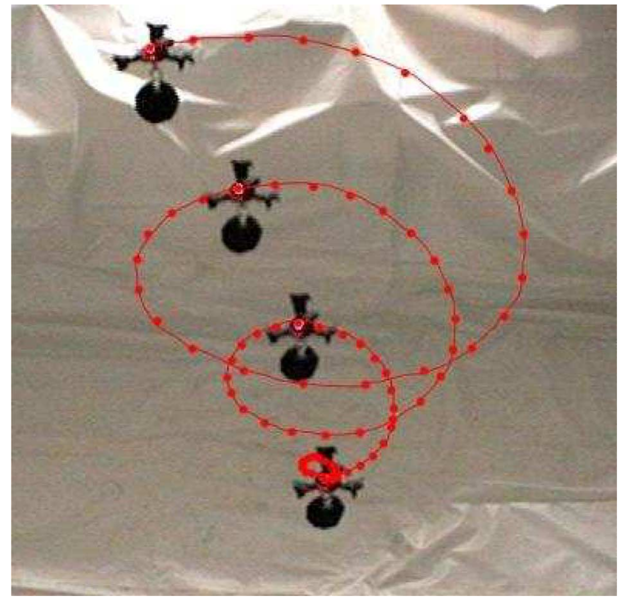


Fig. 9. Time-lapse image of NIMS3D executing a spiral trajectory. Dots show node position at evenly spaced time intervals. In addition, the node is pictured four times along the trajectory.

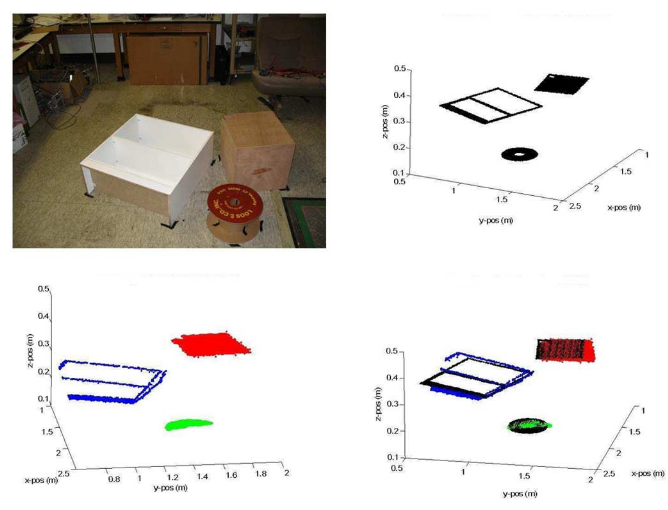


Fig. 10. Multiscale topographical mapping experiment.

VIII. Methods for improving Energy Efficiency

A. Parabolic Trajectories

Cabled robotic platforms such as NIMS3D may require large cable tensions to support an end-effector, particularly when the weight of the end-effector must be supported near the upper limit of the workspace by cables whose angle relative to the horizontal plane is small. Because actuator torque and armature current are linearly related [31], high cable tensions result in high currents, which cause large ohmic I^2R losses, and consequently, excessive power dissipation in the motors. For remote deployments where no power grid is available, this can significantly reduce deployment lifetime. Additionally, industrial robots might incur high operating costs due to poor efficiency.

There is much previous work in trajectory generation for parallel manipulators. The authors of [32] describe generation of time-minimal trajectories for six DOF parallel configurations, whereas [33] describes trajectory generation for cable-based parallel manipulators. Much prior work aims to minimize time of execution of a desired trajectory, which often adversely affects energy efficiency. The problem that we address is as follows: given a starting location and a desired destination in an unobstructed NIMS3D workspace, how might nonlinear trajectories be utilized to improve overall efficiency and reduce I^2R loss in the actuators? It should be noted that the methods we present in this section apply to both three- and four-cabled configurations.

1) Actuator Current Response Modeling: In order to enable improved energy efficiency in a robotic system, it is important to generate a valid model for its actuators. The actuators in our robot are brushed dc gearmotors that are driven by means of pulsewidth modulation (PWM) controlled H-bridges [34]. The dc motors are governed by the equations given in (21) and (22) [31]

$$V(t) = L_{\text{arm}}\frac{\partial I(t)}{\partial t} + R_{\text{arm}}I(t) + K_E\omega(t) \tag{21}$$

$$\tau(t) = K_T I(t) \tag{22}$$

where $V(t)$ is the applied voltage, $I(t)$ is actuator current, L_{arm} and R_{arm} are the armature inductance and resistance, respectively, τ is torque, ω is motor speed, and K_T and K_E are motor constants, which, in SI units, are equal. In our study, a quasi-static approximation is made, and time derivatives are taken to be zero. Thus, after dropping the $L_{\text{arm}}\frac{\partial I(t)}{\partial t}$ term, slight algebraic manipulation yields

$$VI = \tau\omega + I^2R_{\text{arm}} = \tau\omega + IR_{\text{arm}}\frac{\tau}{K_T} \tag{23}$$

where VI is electrical power, $\tau\omega$ is power delivered to the load, and I^2R_{arm} is loss in the armature resistance. Motor efficiency is therefore given by

$$\frac{\omega}{\omega + \frac{IR_{\text{arm}}}{K_T}}. \tag{24}$$

It is apparent from (24) that motor efficiency increases with motor speed, ω.

While this formulation does yield an important result, the motors considered are ideal and are not affected by nonlinearities such as friction. The motors used in NIMS3D are heavily geared to enable improved weight capacity, and these gearboxes cause a substantial amount of friction. To enable characterization of the current response of the actuators to various cable velocities and tensions, current measuring devices were installed for each actuator in NIMS3D. Armature current was recorded while the motors raised and lowered loads with varying masses and velocities. For these experiments, each cable supports its own load and raises it vertically. Thus, if velocity is constant, the tension

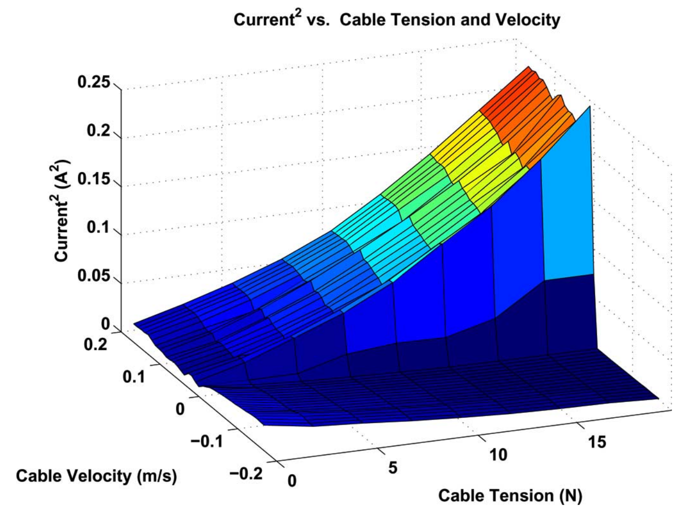


Fig. 11. Square of motor current shown as a function of cable velocity and torque. Positive cable velocities indicate raising a load, while negative velocities indicate lowering a load.

in the cable is equal to the weight of the load and is thereby directly related to motor torque. Transient current spikes associated with initial acceleration were ignored, as is consistent with our quasistatic assumption regarding the NIMS3D trajectories.

The current value associated with each velocity and cable tension for each motor are stored in lookup tables. A plot of the square of these current values, shown in Fig. 11, reveals 1) a quadratic relationship between power loss and cable tension in raising a load and 2) negligible levels of power loss in lowering a load.

2) Parabolic Trajectory Generation: Operating in the upper regions of a workspace results in high cable tensions and low efficiency. Thus, it may be desirable to exploit the lower tensions and reduced Ohmic loss associated with the nether regions of a workspace. The heuristic we use to enable this is to employ downward parabolic trajectories. In generating a parabolic trajectory, an appropriate shift and rotation of coordinates is employed such that the starting point is at the origin and the destination lies at some point $(x_f, 0, z_f)$, in a temporary coordinate system defined as $\hat{x}, \hat{y}, \hat{z}$. A trajectory can be defined as

$$\hat{z} = \mu\hat{x}^2 + \zeta\hat{x} + \chi \tag{25}$$

where μ, ζ, and χ are parameters determining the shape of the parabola. At the origin of the temporary coordinate system, $\hat{x} = 0$ and $\hat{z} = 0$, so χ must also be 0. We also have that $z_f = \mu x_f^2 + \zeta x_f$, which yields

$$\zeta = \frac{z_f - \mu x_f^2}{x_f}. \tag{26}$$

Thus, the trajectory can be completely defined by a single parameter μ, which is advantageous in subsequent optimizations. This is one of the primary motivations in selecting parabolic trajectories. More complex trajectories, which require optimization across several variables, were found to yield negligible improvements in performance.

TABLE IV
EXPERIMENTAL TRAJECTORIES

Traj.	Start(m)	Dest(m)
1	$[6.00\ 1.00\ 1.70]^T$	$[4.25\ 4.00\ 2.10]^T$
2	$[4.25\ 4.00\ 2.10]^T$	$[2.50\ 1.00\ 1.70]^T$
3	$[2.50\ 1.00\ 1.70]^T$	$[6.00\ 1.00\ 1.70]^T$

Clearly, a parabolic path between two points is longer than the corresponding linear path. Because node velocity cannot exceed $V_{\max}$, the velocity limit of the system, the parabolic path incurs delays in completing the desired move. The relative increase in distance is small for moderate values of μ, but as μ grows, the distance penalty increases, and it is this distance penalty that ultimately limits the extent to which optimized trajectories exploit lower regions of the workspace.

The expected current in the $i\text{th}$ motor at any point along the trajectory can be found by looking up in the actuator current response model the current value corresponding to the appropriate cable velocity and tension. Thus, the total energy associated with a trajectory can be approximated by computing the integral of the square of this current over the duration of the trajectory and multiplying the result by R_{arm}. In our system, this is accomplished by means of a second-order Simpson's rule approximation that we have found to provide good accuracy even for a small number of function evaluations. The reader is referred to [3] for more in-depth discussion.

3) Parabolic Trajectory Optimization: Large values of μ result in large downward departures from a linear path and thereby low-power operation, while small values remain close to the linear trajectory and are prone to the high power of operation in upper regions of a workspace. Thus, a reduction in trajectory cost is expected with increasing μ. However, the increase in trajectory length corresponding to growing μ causes the time to perform the trajectories to grow, as node velocity cannot exceed $V_{\max}$, the velocity limit of the system. Thus, the lower average power of operation is ultimately offset by the longer integration period. The task then becomes to find $\mu_{\text{opt}} \geq 0$, the energy optimal value of μ. This bounded optimization is performed by means of an interior-reflective Newton method as described in [35] and [36]. It should be noted that, while the metric that we minimize in this paper is I^2R loss, we can also choose to minimize a weighted sum of I^2R loss and time, which allows for flexibility based on the time and energy constraints of a deployment.

4) Three-Cabled Experimental Results on a Physical System: In order to provide meaningful experimental results, NIMS3D was deployed and a set of representative start and endpoints was selected. The configuration parameters of the deployment are given in Table I, and the start and endpoints of the experimental trajectories are shown in Table IV. The set of trajectories maps out a roughly equilateral triangle with slight variations in height.

An experiment was performed in which straight line and optimized parabolic trajectories through the selected points were executed, and motor currents were sampled throughout the execution time. The experimental NIMS3D configuration and the resulting trajectories are shown in Fig. 12.

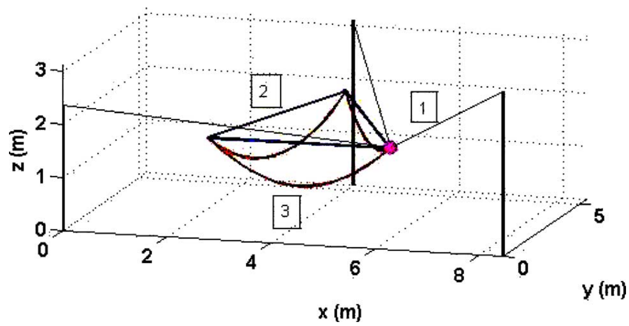


Fig. 12. Straight line and optimized parabolic trajectories through a set of test points.

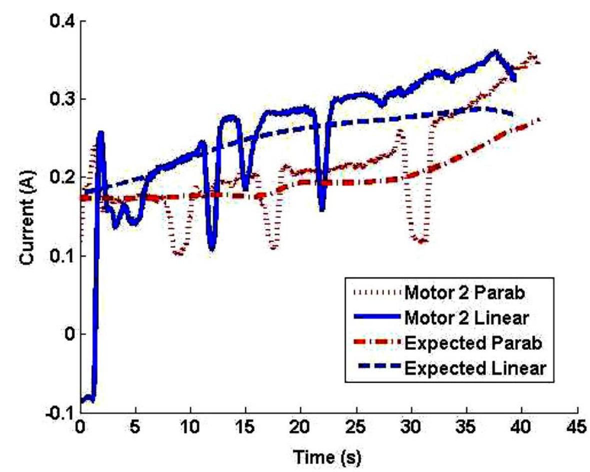


Fig. 13. Motor 2 current shown against expected values.

TABLE V
ENERGY EFFICIENCY OF PARABOLIC TRAJECTORIES

Traj.	E_{Expect}	$E_{Expmt.}$	T_{Ratio}	ρ_{Expect}	$\rho_{Expmt.}$
1	.97	.95	1.03	.94	.92
2	.79	.69	1.15	.69	.60
3	.83	.78	1.13	.73	.69
Avg.	.86	.81	1.10	.79	.74

Fig. 13 shows currents for motor 2 against the expected values. It is apparent that the experimental current values are in fair agreement with expected values and that there is a substantial decrease in average power. The intermittent dips in motor 2 current are most likely due to changes in the control effort resulting from slight trajectory overshoots.

The expected and experimentally observed reductions in I^2R loss for the three paths are shown in Table V, where E indicates the amount of I^2R energy loss in a parabolic trajectory relative to that in a linear path, T_{Ratio} indicates the relative time penalty, and ρ indicates the reduction of average power relative to a linear path. Average power is reduced more than total energy due to the longer execution time of parabolic paths. It should be noted that the relative improvement in energy efficiency that an optimized parabolic trajectory presents is highly dependent on system configuration and region of operation in the workspace. In upper regions of a workspace characterized by very high cable tension and motor currents, slight downward deviations from linearity result in tremendous decreases in required actuator torque, whereas this effect is significantly reduced in lower regions of the workspace.

B. Optimizations for Four-Cabled NIMS3D Configurations

1) Energy-Optimal Tension Distribution: In cabled systems like NIMS3D, four-cable configurations are often preferable over three-cable configurations, as they increase the workspace of a deployment. Inclusion of a fourth cable results in an infinite set of feasible tension solutions, whereas a three-cabled configuration defined by (16) has a unique tension distribution. There exists considerable prior work in determining tension distributions for cabled robots wherein the number of cables exceeds the number of DOFs by one. The authors of [37] derive analytical expressions for optimal tension distributions in a fully constrained six DOF robot. Their method reduces the order of the computation to a one-variable linear programming (LP) problem with the sum of all tensions used as the objective function. The optimization is then solved by exhaustively checking all extreme points. Other previous work [18], [19], [25] has suggested LP approaches to the problem of determining tension distributions in cabled robots. In particular, [18] and [19] consider a cable array crane similar to a four-cable NIMS3D arrangement. Various objective functions have been considered, including minimizing or maximizing the sum of all tensions, to reduce effort or increase stability, respectively. Another method is to maximize the sum of the two lowest tensions, which also is intended to improve stability.

A major shortcoming of all these methods is that they aim to optimize linear sums of cable tensions. Thus, the least effort LP distributions are optimal in the one-norm sense but do not consider the two-norm of the cable tension vector, which is a more significant measure, as it is directly related to I^2R loss. We have developed an alternate method of determining tension distribution that is less computationally intensive and is energy-optimal in ideal actuators governed by linear torque–current relationships.

The problem of finding the energy-optimal tension distribution is a least norm problem with linear inequality constraints, as shown in

$$\begin{aligned}&\min\|\vec{T}\|_2^2\\ &\text{s.t. } \Lambda_{n\times m}\vec{T} = M(\vec{a}+\vec{g}) = b\\ &\qquad T_{\min} \le \vec{T} \le T_{\max}\end{aligned} \tag{27}$$

where n is the number of degrees of freedom, $m = n+1$ is the number of cables, M is the robot's inertia matrix, $\vec{a}$ is the acceleration vector, and where we have introduced the new variable b. $\Lambda_{n\times m}$ is the pose-dependent structure matrix of the system. For a four-cable NIMS3D configuration, for example, it is a 3×4 matrix whose ith column is given by $-\vec{l}_i$. We begin in a fashion similar to that presented in [37] and partition $\Lambda_{n\times m}$ and $\vec{T}$ as follows:

$$\begin{aligned}\Lambda_{n\times m} &= [F \mid H]\\ \vec{T} &= \begin{bmatrix}\vec{T}_{1:n}\\ T_m\end{bmatrix}\end{aligned} \tag{28}$$

where $F \subseteq \Re^{n\times n}$, $H \subseteq \Re^n$, and $\vec{T}_{1:n}$ is a vector containing the first n cable tensions. Thus, we have

$$[F \mid H]\begin{bmatrix}\vec{T}_{1:n}\\ T_m\end{bmatrix} = b. \tag{29}$$

If we perform the component-wise matrix multiplication, subtract $T_m H$ from both sides, and premultiply by F^{-1}, we are left with

$$\vec{T}_{1:n} = F^{-1}(b - T_m H) \tag{30}$$

and thus

$$\vec{T} = \begin{bmatrix}F^{-1}(b-T_m H)\\ T_m\end{bmatrix}. \tag{31}$$

From (31), it is clear that a tension distribution can be uniquely defined by assigning a tension to one of the cables. In this paper, we have arbitrarily chosen cable m, but any cable can be selected.

Returning to the optimization problem in (27), we see that the cost function can be written in terms of T_m. Defining $G = F^{-1}$, we have

$$\begin{aligned}\min\|\vec{T}\|_2^2 &= T_m^2 + \|G(b-T_m H)\|_2^2\\ &= b^T G^T G b - 2b^T G^T G H T_m + (H^T G^T G H + 1)T_m^2\end{aligned} \tag{32}$$

Similarly, the constrains in (27) can be expressed in terms of T_m

$$T_{\min} \le \begin{bmatrix}G(b-T_m H)\\ T_m\end{bmatrix} \le T_{\max} \tag{33}$$

which, after subtracting Gb from the first n expressions and multiplying them by -1 yields

$$\begin{bmatrix}Gb - T_{\max}\\ T_{\min}\end{bmatrix} \le \begin{bmatrix}T_m GH\\ T_m\end{bmatrix} \le \begin{bmatrix}Gb - T_{\min}\\ T_{\max}\end{bmatrix}. \tag{34}$$

If these bounds are infeasible, then there is no solution to the constrained optimization. If they are feasible, then finding T_m^*, the optimal value of T_m, has been reduced to a bounded minimization of a quadratic function. By taking the derivative of (32) with respect to T_m and setting it equal to zero, we solve for the optimal point $\hat{T}_m$ of the unconstrained problem

$$\hat{T}_m = \frac{b^T G^T G H}{1 + H^T G^T G H}. \tag{35}$$

We show in the Appendix that $\hat{T}_m$ is equal to the unconstrained least norm solution given by $\Lambda_{n\times m}^T(\Lambda_{n\times m}\Lambda_{n\times m}^T)^{-1}b$. If $\hat{T}_m$ lies within the feasible range as determined by (34), then it is the optimal point. If it lies below that range, then the optimal point is equal to the highest lower bound. If $\hat{T}_m$ lies above the feasible range, then the optimal operating point is the lowest upper bound. In other words

$$T_m^* = \min(\epsilon_+, \max(\hat{T}_m, \epsilon_-)) \tag{36}$$

where ϵ_- is the greatest lower bound and ϵ_+ is the lowest upper bound.

TABLE VI
EXPERIMENTAL NIMS3D CONFIGURATION

Cable Origin	1	2	3	4
$x(m)$	0	0	10	10
$y(m)$	0	10	10	0
$z(m)$	4	4	4	4

TABLE VII
SIMULATED TRAJECTORIES

Traj. Number	Start(m)	Dest(m)
1	$[3.00\ 3.00\ 2.70]^T$	$[3.00\ 7.00\ 2.70]^T$
2	$[3.00\ 7.00\ 2.70]^T$	$[7.00\ 7.00\ 2.70]^T$
3	$[7.00\ 7.00\ 2.70]^T$	$[7.00\ 3.00\ 2.70]^T$
4	$[7.00\ 3.00\ 2.70]^T$	$[3.00\ 3.00\ 2.70]^T$

TABLE VIII
RELATIVE I^2R LOSS FOR LINEAR AND PARABOLIC TRAJECTORIES WITH LEAST NORM AND LP TENSION DISTRIBUTIONS

	Linear LN	Parabolic LN	Parabolic LP
Ideal	.67	.35	.54
Nonideal	.86	.71	.87

This method of determining tension distribution yields substantial improvements in efficiency and computational intensity over the LP methods discussed in [18] and [19]. The reduction in computation time is critical, as the tension distribution must be solved several times in each iteration of the trajectory optimization. The simplex method [38] used to generate the LP tension distributions results in significant delays, whereas the least-norm method is fast enough to enable real-time generation of optimized trajectories.

2) Tension Distribution Simulations: In order to verify the effect of least-norm tension distributions in reducing I^2R loss, we simulated a hypothetical four-cable NIMS3D system with the configuration parameters shown in Table VI. A representative set of start and endpoints was chosen. These trajectories, shown in Table VII, map a square at constant height. Optimized parabolic trajectories were generated for each set of start and end points. Tension distributions were given by: 1) the least effort LP method described in [18], wherein the sum of all tensions is minimized and 2) our least norm (LN) tension distribution. Table VIII shows relative I^2R losses for linear and parabolic trajectories with LN tension distributions, as well as for parabolic trajectories with LP tension distributions. The values shown are losses relative to the case in [18], where linear paths are used and tension distributions are found using the LP method. Because the relative efficiency of our methods is heavily dependent on actuator models, we show data for ideal actuators with linear current–torque relationships and for those in our system. It is evident that, for both actuator models, using LN tension distributions and parabolic trajectories significantly reduces I^2R loss as compared to linear paths with LP tension distributions. The nonidealities of the motors in our system reduce the improvement in efficiency, although significant improvements relative to the linear, LP methods are still observed.

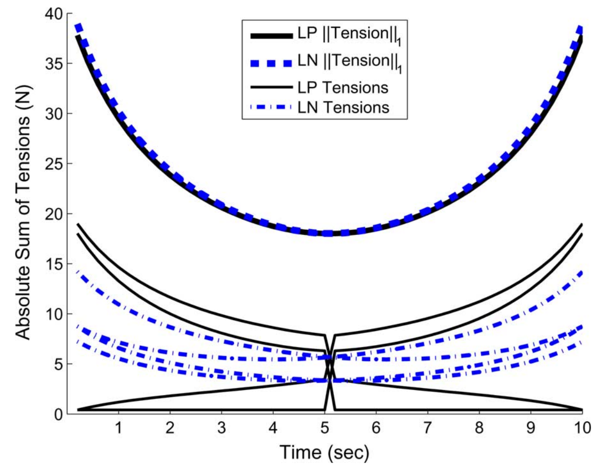


Fig. 14. Absolute sum of tensions for a parabolic trajectory with LN and LP tension distributions. Traces of individual cable tensions are shown as well.

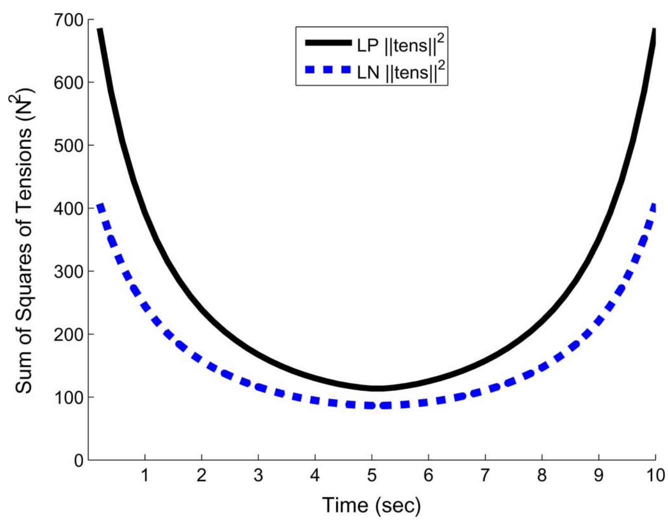


Fig. 15. Sum of squares of tensions for a parabolic trajectory with LN and LP tension distributions.

To better visualize the effect of LN and LP tension distributions, the one-norm and two-norm of tension distributions during execution of a parabolic trajectory are shown in Figs. 14 and 15, respectively. From Fig. 14, it is apparent that the one-norm of tensions for the LN method is very close to that of the LP method throughout. In other words, the cost in the one-norm sense of optimizing for the two-norm is minimal. Additionally, traces of all cable tensions are shown for both the LN (dotted lines) and the LP (solid lines) methods. These traces reveal a number of things: First, the cable tensions found by LP methods are not continuous. Halfway through the trajectory, the tension distribution encounters a discontinuity. This is an undesirable property, as rapid switching from one tension to another may excite high-frequency modes in the cables.

The LP method of determining tension distributions can be reduced to a single-variable optimization in the same way as shown in (31). In this case, the LP problem is to minimize the sum of tensions, subject to the same constraints as in (34). The

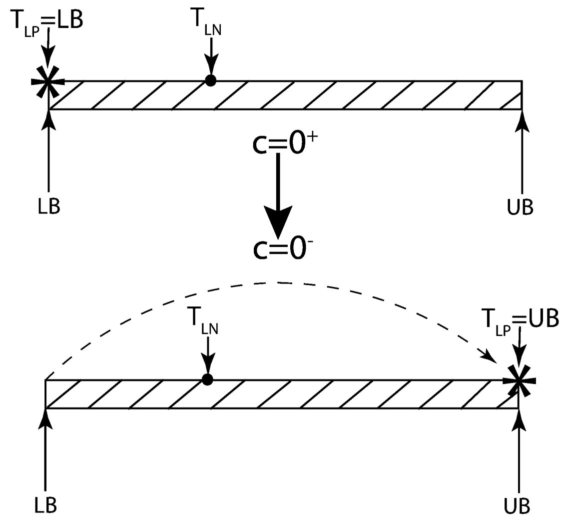


Fig. 16. Example of discontinuity in LP tension distributions. As c transitions from 0^+ to 0^-, the LP-optimal point jumps from the lower bound to the upper bound, while the LN-optimal point moves only an infinitesimal amount.

sum of tensions is given by

$$\begin{aligned}\sum T &= \mathbf{1}^T \vec{T}_{1:n} + T_m \\ &= \mathbf{1}^T F^{-1} b + T_m (1 - \mathbf{1}^T F^{-1} H) \\ &= \mathbf{1}^T F^{-1} b + T_m c \end{aligned} \tag{37}$$

where $c = 1 - \mathbf{1}^T F^{-1} H$. In LP theory [39], an optimal point always occurs at a vertex of the feasible polyhedron. If the constraints in (34) are feasible, the feasible polyhedron is a line segment, and the optimal operating point is the endpoint that has the smallest value for (37). If the sign of c changes, the optimal solution jumps from an upper boundary to a lower boundary or vice versa. The LN method does not exhibit such behavior, as the desired operating point given by (36) moves gradually across the feasible region between the boundaries. This is shown in Fig. 16.

Furthermore, it is evident from Fig. 14 that the LP method results in tensions that are either very low or very high, while the LN distribution tends to produce intermediate cable tensions. Thus, the LP method results in operating points that are closer to $T_{\min}$ and $T_{\max}$. Approaching the lower limit may produce near-slack conditions, while approaching the upper limit incurs increased I^2R losses. The square of the two-norms of the tension distributions for both methods are shown in Fig. 15. The LN method shows significant reductions in cost as compared to the LP method. This is due to the fact that the LN method avoids excessively high cable tensions, which results in significant reductions in I^2R loss.

IX. Conclusion

In this paper, we have detailed the design and implementation of NIMS3D, an underconstrained, 3-D cabled robot for actuated sensing applications. The system is intended for rapid in-field deployments. Therefore, rapid calibration methods have been developed to reduce the deployment time. Kinematic and dynamic analysis of the system have been provided, and results from trajectory control experiments have been shown. We have briefly detailed an example actuated sensing application and provided qualitative results. Finally, we have described methods of generating energy-efficient trajectories and tension distributions, and have provided experimental and simulation results that show significant reductions in energy costs as compared to previous methods.

NIMS3D systems are now under development for a wide range of environmental sensing applications including direct mapping of carbon dioxide flux, meteorological phenomena, and solar radiation distribution under forest canopies. Here, the capability for precise and autonomous manipulation of sensors will provide the first direct mapping of these phenomena.

An additional set of new applications is also associated with direct mapping of contaminant distribution and flux in river, lake, and reservoir systems. Here, an NIMS3D architecture includes NIMS-AQ, a four-cabled planar cable-driven robot for aquatic applications. The end-effector is neutrally buoyant and carries a sensing module containing many water quality sensor devices. This can be translated to sampling locations and lowered to the desired depth according to appropriate sampling schedules. We have developed optimal methods of tension distribution that are sufficiently fast for real-time operation. We plan to deploy these systems across large transects that may span entire rivers or lakes, thus enabling autonomous monitoring and regulation of important waterways.

Appendix

The unconstrained least norm solution $\tilde{T}$ is given by

$$\tilde{T} = \Lambda_{n\times m}^T (\Lambda_{n\times m} \Lambda_{n\times m}^T)^{-1} b. \tag{38}$$

Upon partitioning $\Lambda_{n\times m}$ as in (28), we have

$$\tilde{T} = \begin{bmatrix} F^T \\ H^T \end{bmatrix} \left([F\,|H] \begin{bmatrix} F^T \\ H^T \end{bmatrix} \right)^{-1} b. \tag{39}$$

The m^{th} entry of $\tilde{T}$ is given by

$$\tilde{T}_m = H^T \left(FF^T + HH^T \right)^{-1} b. \tag{40}$$

Application of the Sherman–Morrison formula for the matrix inverse yields

$$\begin{aligned}\tilde{T}_m &= H^T \left(G^T G - \frac{G^T G H H^T G^T G}{1 + H^T G^T G H} \right) b \\ &= b^T \left(G^T G - \frac{G^T G H H^T G^T G}{1 + H^T G^T G H} \right) H \end{aligned} \tag{41}$$

where we have used $G = F^{-1}$. Continuing

$$\begin{aligned}\tilde{T}_m &= \frac{b^T (G^T G + (H^T G^T G H) G^T G - G^T G H H^T G^T G) H}{1 + H^T G^T G H} \\ &= \frac{b^T (G^T G H + H^T G^T G H (G^T G H - G^T G H))}{1 + H^T G^T G H} \\ &= \frac{b^T G^T G H}{1 + H^T G^T G H} = \hat{T}_m. \end{aligned} \tag{42}$$

The tension in the remaining cables is uniquely defined by assigning one tension, so we conclude that

$$\tilde{T} = \hat{T} = \begin{bmatrix} F^{-1}(b - \hat{T}_m H) \\ \hat{T}_m \end{bmatrix}. \tag{43}$$

REFERENCES

[1] P. H. Borgstrom, M. Stealey, M. Batalin, and W. Kaiser, "Nims3d: A novel rapidly deployable robot for 3-dimensional applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3628–3635.

[2] P. H. Borgstrom, N. P. Borgstrom, M. Stealey, B. Jordan, G. S. Sukhatme, M. Batalin, and W. Kaiser, "Discrete trajectory control algorithms for nims3d, an autonomous underconstrained three-dimensional cabled robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct./Nov. 2007, pp. 253–260.

[3] P. H. Borgstrom, N. P. Borgstrom, M. J. Stealey, B. Jordan, G. S. Sukhatme, M. Batalin, and W. Kaiser, "Generation of energy-efficient trajectories for nims3d, a three-dimensional cabled robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 2222–2227.

[4] A. Riechel, P. Bosscher, H. Lipkin, and I. Ebert-Uphoff, "Concept paper: Cable-driven robots for use in hazardous environments," presented at the 10th Int. Top. Meeting Robot. Remote Syst. Hazardous Environ., Gainesville, FL, Mar. 2004.

[5] P. Bosscher, R. L. Williams, and M. Tummino, "A concept for rapidly-deployable cable robot search and rescue systems," presented at the ASME IDETC/CIE, Long Beach, CA, Sep. 2005.

[6] F. Takemura, M. Enomoto, T. Tanaka, K. Denou, Y. Kobayashi, and S. Tadokoro, "Development of the balloon-cable driven robot for information collection from sky and proposal of the search strategy at a major disaster," in *Proc. 2005 IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul., pp. 658–663.

[7] F. Takemura, K. Maeda, and S. Tadokoro, "Attitude stability of a cable driven balloon robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 3504–3509.

[8] SkyCam. (2007). [Online]. Available: www.skycam.tv

[9] CableCam. (2007). [Online]. Available: www.cablecam.com

[10] K. Usher, G. Winstanley, P. Corke, D. Stauffacher, and R. Carnie, "Air vehicle simulator: An application for a cable array robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 2241–2246.

[11] B. Jordan, M. Batalin, and W. Kaiser, "Nims rd: A rapidly deployable cable based robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 144–150.

[12] T. Harmon, R. Ambrose, R. Gilbert, J. Fisher, M. Stealey, and W. Kaiser, "High-resolution river hydraulic and water quality characterization using rapidly deployable networked infomechanical systems (nims rd)," *Envion. Eng. Sci.*, vol. 24, no. 2, pp. 151–159, 2006.

[13] A. Singh, M. Batalin, M. Stealey, V. Chen, M. Hansen, T. Harmon, G. Sukhatme, and W. Kaiser, "Mobile robot sensing for environmental applications," in *Proc. Field Serv. Robot.*, 2007, pp. 125–135.

[14] A. Singh, M. Batalin, V. Chen, M. Stealey, B. Jordan, J. Fisher, T. Harmon, M. Hansen, and W. Kaiser, "Autonomous robotic sensing experiments at San Joaquin river," in *Proc. 2007 IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 4987–4993.

[15] R. L. Williams and P. Gallina, "Planar cable-direct-driven robots, part i: Kinematics and statics," presented at the 2001 ASME Des. Tech. Conf., Pittsburgh, PA, Sep. 2001.

[16] R. L. Williams, P. Gallina, and A. Rossi, "Planar cable-direct-driven robots, part ii: Dynamics and control," presented at the 2001 ASME Des. Tech. Conf., Pittsburgh, PA, Sep. 2001.

[17] R. L. Williams, P. GAllina, and J. Vadia, "Planar translational cable-driven robots," *J. Robot. Syst.*, vol. 20, no. 3, pp. 107–120, 2003.

[18] W. Shiang, D. Cannon, and J. Gorman, "Dynamic analysis of the cable array robotic crane," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1999, vol. 4, pp. 2495–2500.

[19] W. Shiang, D. Cannon, and J. Gorman, "Optimal force distribution applied to a robotic crane with flexible cables," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2000, vol. 2, pp. 1948–1954.

[20] K. Maeda, S. Tadokoro, T. Takamori, M. Hiller, and R. Verhoeven, "On design of a redundant wire-driven parallel robot warp manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1999, vol. 2, pp. 895–900.

[21] S. Kawamura, W. Choe, S. Tanaka, and S. Pandian, "Development of an ultrahigh speed robot falcon using wire drive system," in *Proc. 1995 IEEE Int. Conf. Robot. Autom.*, pp. 215–220.

[22] M. Yamamoto, N. Yanai, and A. Mohri, "Trajectory control of incompletely restrained parallel-wire-suspended mechanism based on inverse dynamics," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 840–850, Oct. 2004.

[23] A. Basar and S. Agrawal, "Cable suspended robots: Feedback controllers with positive inputs," in *Proc. Amer. Control Conf.*, May 2002, pp. 840–850.

[24] S. Oh and S. Agrawal, "Cable suspended planar robots with redundant cables: Controllers with positive inputs," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 457–465, Jun. 2005.

[25] C. Pham, G. Yang, and S. Yeo, "Dynamic analysis of cable-driven parallel mechanisms," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 24–28, 2005, pp. 612–617.

[26] K. Cheung, H. So, W. Ma, and Y. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1121–1127, Apr. 2004.

[27] L. Geosystems. (2007). [Online]. Available: http://www.leica-geosystems.com/ cpd/en/support/lgs_6598.htm?cid=9467

[28] A. Riechel and I. Ebert Uphoff, "Force-feasible workspace analysis for underconstrained point-mass cable robots," in *Proc. 2004 IEEE Int. Conf. Robot. Autom.*, Apr., pp. 4956–4962.

[29] Y. Shen, H. Osumi, and T. Arai, "Manipulability measures for multi-wire driven parallel mechanisms," in *Proc. 1994 IEEE Int. Conf. Ind. Technol.*, Dec., pp. 550–554.

[30] W. Kwon and A. Pearson, "On feedback stabilization of time-varying discrete linear systems," *IEEE Trans. Automat. Contr.*, vol. 23, no. 3, pp. 479–481, Jun. 1978.

[31] S. Cetinkunt, *Mechatronics*. New York: Wiley, 2007.

[32] H. Abdellatif and B. Heimann, "Adapted time-optimal trajectory planning for parallel manipulators with full dynamic modelling," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 411–416.

[33] M. Hiller, S. Fang, S. Mielczarek, R. Verhoeven, and D. Franitza, "Design, analysis and realization of tendon-based parallel manipulators," *Mech. Mach. Theory*, vol. 40, pp. 429–445, Aug. 2004.

[34] S. Cubed. 2007. [Online]. Available: http://www.solutionscubed.com

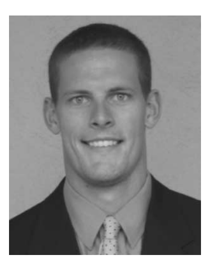
[35] T. Coleman and Y. Li, "On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds," *Math. Program.*, vol. 67, no. 2, pp. 189–224, 1994.

[36] T. Coleman and Y. Li, "An interior, trust region approach for nonlinear minimization subject to bounds," *SIAM J. Optim.*, vol. 6, pp. 418–445, 1996.

[37] S. Fang, D. Franitza, M. Torlo, F. Bekes, and M. Hiller, "Motion control of a tendon-based parallel manipulator using optimal tension distribution," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 3, pp. 561–568, Sep. 2004.

[38] G. Dantzig, A. Orden, and P. Wolfe, "Generalized simplex method for minimizing a linear form under linear inequality constraints," *Pacific J. Math.*, vol. 5, no. 2, pp. 183–195, 1955.

[39] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Programming*. Belmost, MA: Athena Scientific, 1997.



Per Henrik Borgstrom received the B.S. and M.S. degrees in electrical engineering from the University of California, Los Angeles (UCLA), in 2005 and 2007, respectively.

He is currently working toward the Ph.D. degree in electrical engineering at UCLA, where his thesis is on the design of cabled robotic platforms for use in actuated sensing applications. His current research interests include algorithms and cabled robotic platforms for actuated sensing applications, with emphasis on calibration, and tension distribution methods.

Nils Peter Borgstrom received the B.S. degree in electrical engineering in 2008 from the University of California, Los Angeles (UCLA).

He is currently working toward the M.S. degree in biomedical engineering at UCLA. His current research interests include acquisition, processing, and analysis of biomedical signals and images, including digital modalities such as X-ray, CT, MR, PET, and SPECT.

Michael J. Stealey received the B.S. degree in computer science and the M.S. degree in electrical engineering from the University of California, Los Angeles (UCLA).

He is currently a Senior Research Software Developer for the Renaissance Computing Institute (RENCI), Chapel Hill, NC.

Brett Jordan received the B.S. degree and the M.S. degree in mechanical engineering from the University of California, Los Angeles (UCLA), in 2006 and 2008, respectively.

He is currently a Staff Research Engineer at UCLA and is involved in mechanical design and product development.

His current research interests include mechanical design and development of robotic systems used in environmental sensing applications as well as systems developed for medical applications.

Gaurav S. Sukhatme (S'94–M'97–SM'05) received the M.S. and Ph.D. degrees in computer science from the University of Southern California (USC), Los Angeles.

He is currently an Associate Professor of Computer Science (joint appointment in Electrical Engineering) with USC. He is the Co-Director of the USC Robotics Research Laboratory and the Director of the USC Robotic Embedded Systems Laboratory, which he founded in 2000. He has served as a Project Investigator (PI) on several National Science Foundation (NSF), Defense Advanced Research Projects Agency (DARPA), and National Aeronautics and Space Administration (NASA) grants. He is a Co-PI at the Center for Embedded Networked Sensing (CENS), an NSF Science and Technology Center. His current research interests include multirobot systems, sensor/actuator networks, and robotic sensor networks. He is the author or coauthor of several papers published extensively in these and related areas.

Dr. Sukhatme is a member of the Association for the Advancement of Artificial Intelligence (AAAI) and the Association of Computing Machinery (ACM) and is one of the founders of the Robotics: Science and Systems Conference. He is Program Chair of the 2008 IEEE International Conference on Robotics and Automation and the Editor-in-Chief of *Autonomous Robots*. He has served as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, and on the Editorial Board of the IEEE PERVASIVE COMPUTING. He was a recipient of the NSF CAREER Award and the Okawa Foundation Research Award.

Maxim A. Batalin (S'03–M'02) received the M.S. degree and the Ph.D. degree in computer science from the University of Southern California (USC), Los Angeles, in 2002 and 2005, respectively. He is currently a Researcher at the Center for Embedded Networked Sensing, University of California, Los Angeles (UCLA).

He joined the Wireless Health Program at the UCLA in 2005 in the Actuated, Sensing, Coordinated, and Embedded Networked Technologies (ASCENT) Laboratory. While at ASCENT, among other projects, he has managed the development of the first of its kind incremental diagnosis method for biomedical applications that allows accurate patient state identification with minimum sensor usage. He has also managed the development of the first end-to-end wireless health monitoring system, MEDIC, which offers a wireless, wearable architecture for biomedical data acquisition, local processing, and transmittal to a remote enterprise server. He is the author of over 35 conference and journal publications in the areas of wireless health, mobile robotics, sensor networking, actuated sensor systems, biomedical systems, environmental sampling, and others. He has expertise also in the development of software systems for mobile wireless platforms. He specializes in the research, design, and implementation of systems in areas of robotics, sensors, general biomedical computing, and system integration.

Dr. Batalin is a member of Sigma Xi.

William J. Kaiser (M'95–SM'00) received the Ph.D. degree in solid state physics from Wayne State University, Detroit, MI, in 1984.

From 1977 through 1986, as a member of Ford Motor Company Research Staff, his development of automotive sensor and embedded system technology resulted in large-volume commercial sensor production. At Ford, he also developed the first spectroscopies directed to microelectronics systems based on scanning tunneling microscopy. From 1986 through 1994, at the Jet Propulsion Laboratory, he initiated the NASA Microinstrument Program for distributed sensing. In 1994, he joined the Faculty of the University of California, Los Angeles (UCLA)'s Electrical Engineering Department. Along with Prof. Pottie, he initiated the first wireless networked microsensor program with a vision of linking the Internet to the physical world through distributed monitoring. This continued research includes the topics of low-power embedded computing for wireless networked sensing, biomedical embedded computing, robotic sensor systems for environmental monitoring, and distributed sensing for energy and water resource management. He served as Electrical Engineering Department Chairman from 1996 through 2000. He is the author or coauthor of more than 170 publications and is the holder of 23 patents.

Prof. Kaiser received the Peter Mark Award of the American Vacuum Society, the NASA Medal for Exceptional Scientific Achievement, the Arch Colwell Best Paper Award of the Society of Automotive Engineers, the Best Paper Award at BodyNets 2008, two R&D 100 Awards, the Allied Signal Faculty Research Award, the Brian P. Copenhaver Award for Innovation in Teaching with Technology, and the UCLA Gold Shield Faculty Award.