Guaranteed Reachable Domain and Control Design for a Cable Robot Subject to Input Constraints.

So-Ryeok Oh, Grad. Student
Mechanical System Lab
Dept. of Mechanical Engineering
University of Delaware,
Newark DE 19716.
Email: oh@me.udel.edu

Sunil K. Agrawal, Ph.D, Professor
Mechanical System Lab
Dept. of Mechanical Engineering
University of Delaware,
Newark DE 19716.
Email: agrawal@me.udel.edu

Abstract— **Cable-suspended robots are structurally similar to parallel actuated robots but with the fundamental difference that cables can only pull the end-effector but not push it. From a scientific point of view, this feature makes feedback control of cable-suspended robots a lot more challenging than their counterpart parallel-actuated robots. In this paper, we present a computationally efficient control design procedure for a fully actuated cable robot with positive input constraints. The basic idea is to calculate a set of reachable domain analytically, where a neighboring domain possesses common points. This allows to expand the region of feasible reference signals by simply connecting adjacent feasible domains. Finally, the effectiveness of the proposed method is illustrated by numerical simulations and laboratory experiments on a six degree-of-freedom cable suspended robot.**

Index Terms— **Cable suspended robot, input constraint, reachable domain.**

I. Introduction

In the last decades, robots have made tremendous inroads into industries for manufacturing and assembly. However, for long reach robotics such as inspection and repair in shipyards and airplane hangars, application of robotics is still in its infancy. Conventional robots with serial or parallel structures are impractical for these applications since the workspace requirements are higher by orders of magnitude than what the conventional robots can provide.

However, cables have the unique property - they can not provide compression force on an end-effector. This constraint leads to performance deterioration and even instability, if not properly accounted for in the design procedure.

In this paper, we reconsider a technique recently presented by the authors [1], which deals with the problem of characterizing the reachable domain of a cable robot analytically in a conservative way. One of the drawbacks of the proposed method was that the reachable domain did not cover the entire workspace. To overcome this shortcoming, here we propose extensions of the algorithm to expand the reachable doamin to the boundary of the actual workspace.

The key idea in this paper is to generate sequentially the reachable domain by iteratively using a computational framework. The important advantages to use such a procedure are as follows: (i) it does not require solving nonlinear optimization problems to calcuate a set of admissible reference signals as common in Model Predictive Control ([3],[4]) and Reference Governor based control schemes ([5],[6]), (ii) a family of admissible regions are determined analytically, (iii) on-line characterization of feasible domains is possible. By simply connecting neighboring reachable domains, one can generate piecewise constant reference signals to extend the operation range of the end effector, while keeping the cable from becoming slack.

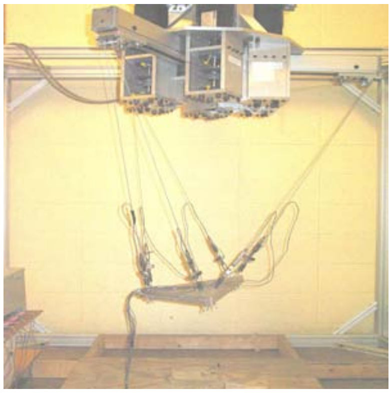

Fig. 1. Detailed camera images of a cable-suspended robot designed and assembled at University of Delaware

The rest of the paper is organized as follows : Section II shows the dynamic equations of the robot. The structure of the controller is described in Section III. In Section IV, the algorithm to generate reachable domains is addressed. This technique is implemented in simulation and in hardware in Sections IV and V, respectively.

II. System Dynamic Model

On using Newton-Euler's laws, the equations of motion can be written in the following form. Please refer to [8] for the detailed kinematics and dynamics of the cable robot.

$$\begin{bmatrix} m\ddot{x}_m \\ m\ddot{y}_m \\ m(\ddot{z}_m - g) \\ I\begin{pmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix} + \begin{pmatrix}\omega_1\\ \omega_2\\ \omega_3\end{pmatrix} \times I\begin{pmatrix}\omega_1\\ \omega_2\\ \omega_3\end{pmatrix}\end{bmatrix} = -J^T(\mathbf{q})\mathbf{u}, \quad (1)$$

0-7803-9098-9/05/$25.00 ©2005 AACC

where m is the mass and I is the moment of inertia of the end-effector about its center of mass with respect to the basis vectors $\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}$.

III. Lyapunov Controller

A controller plays an important role in characterizing the reachable domainw since the cable tensions are functions of the motion of the system. However, the controller itself is not the main concern of this paper. Given a controller designed to meet a given set of performance criteria, our emphasis is to guarantee point to point motion of the moving platform. Since the approach is not limited by the type of controller, controllers such as *Computed torque* and *Feedback linearization*, etc., can be applied [2].

In this section, we use a Lyapunov based controller for set-point control of the end-effector to achieve uniform exponential stability [7]. Even though we do not consider uncertainty, exponential convergence is important in applications because it can be shown to be robust to perturbations. For this controller, we define a control surface and a Lyapunov function given below

$$\mathbf{s} = \dot{\mathbf{x}} + \lambda(\mathbf{x} - \mathbf{x}_d) \ , \tag{2}$$

$$V = \frac{1}{2}\mathbf{s}^T\mathbf{s} \ , \tag{3}$$

where $\mathbf{x}_d$ is the desired final states.

Selecting the control law as

$$\begin{aligned}\mathbf{u} = -(J^T)^{-1}[\ & C(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{g}(\mathbf{x}) \\ & - D(\mathbf{x})\lambda\dot{\mathbf{x}} - D(\mathbf{x})\eta\mathbf{s}\ ] \ ,\end{aligned} \tag{4}$$

we can impose

$$\dot{\mathbf{s}} = -\eta\mathbf{s}. \tag{5}$$

and show that

$$\dot{\mathbf{V}} = -\mathbf{s}^T\eta\mathbf{s} \ . \tag{6}$$

The equilibrium at the $\mathbf{x}_d$ is globally exponentially stable since Eq. (6) is negative definite and $\mathbf{s} = 0$ provides the sliding surface.

Eq. (5) can be rewritten in the form of the second-order transfer function by

$$x(s) = \frac{\eta_o\lambda_o}{s^2 + (\eta_o + \lambda_o)s + \eta_o\lambda_o}x_d(s) \ , \tag{7}$$

where x_d stands for the set-point of the x motion, $\lambda = \lambda_o \ \mathtt{diag}(1,1,1,1,1,1)$ and $\eta = \eta_o \ \mathtt{diag}(1,1,1,1,1,1)$. For the second order linear system of Eq. (7), we can find the damping ratio ξ and the natural frequency ω_n in terms of η_o and λ_o. Their expressions are

$$\xi = \frac{\eta_o + \lambda_o}{2\sqrt{\eta_o\lambda_o}} \ , \tag{8}$$

$$\omega_n = \sqrt{\eta_o\lambda_o} \ . \tag{9}$$

Here, ξ is positive since η_o and λ_o should be positve to realize a stabilizing Lyapunov controller. We solve Eq. (8) for η_o, which gives

$$\eta_o = \left[(2\xi^2 - 1) \pm \sqrt{(2\xi^2 - 1) - 1}\right]\lambda_o \ . \tag{10}$$

From the condition that η_o is positive and real, we find the inequality condition for ξ as follows:

$$\begin{aligned} & (2\xi^2 - 1)^2 - 1 \geq 0 \ , \\ \iff \quad & 2\xi^2(\xi^2 - 1) \geq 0 \ , \\ \iff \quad & \xi^2 - 1 \geq 0 \ , \\ \iff \quad & \xi \geq 1 \ . \end{aligned} \tag{11}$$

This result tells us that the above condition ensures an overdamped or a critically damped system. Note that if $\lambda = \eta$, the system is guaranteed to be critically damped, while for $\lambda \neq \eta$, the system becomes overdamped. Given the initial conditions x_0 and $\dot{x}_0 = 0$, the corresponding time-domain solutions are given in terms of ω_n and ξ as follows:

critically damped :

$$\begin{aligned} x(t) &= (x_0 - x_d)e^{-\omega_n t}(1 + \omega_n t) + x_d \ , \\ \dot{x}(t) &= -\omega_n^2(x_0 - x_d)e^{-\omega_n t}t \ , \\ \ddot{x}(t) &= (x_0 - x_d)\omega_n^2 e^{-\omega_n t}(\omega_n t - 1) \ . \end{aligned} \tag{12}$$

overdamped :

$$\begin{aligned} x(t) = \ & \tfrac{(x_0 - x_d)}{2\sqrt{\xi^2 - 1}}[(-\xi + \sqrt{\xi^2 - 1})e^{-(\xi + \sqrt{\xi^2 - 1})\omega_n t} \\ & +(\xi + \sqrt{\xi^2 - 1})e^{(-\xi + \sqrt{\xi^2 - 1})\omega_n t}] + x_d \ , \\ \dot{x}(t) = \ & \tfrac{\omega_n(x_0 - x_d)}{2\sqrt{\xi^2 - 1}}[e^{-(\xi + \sqrt{\xi^2 - 1})\omega_n t} \\ & -e^{(-\xi + \sqrt{\xi^2 - 1})\omega_n t}] \ , \\ \ddot{x}(t) = \ & \tfrac{\omega_n^2(x_0 - x_d)}{2\sqrt{\xi^2 - 1}}[-(\xi + \sqrt{\xi^2 - 1})e^{-(\xi + \sqrt{\xi^2 - 1})\omega_n t} \\ & +(\xi - \sqrt{\xi^2 - 1})e^{(-\xi + \sqrt{\xi^2 - 1})\omega_n t}] \ . \end{aligned} \tag{13}$$

For the overdamped case, the bounds on states are given by

$$\begin{aligned} x(t) &= [x_0, \quad x_d] \\ \ddot{x}(t) &= (x_0 - x_d)\omega_n^2[\mathtt{max}(x_s), \quad -1] \\ &= (x_0 - x_d)\omega_n^2[Z_M, \quad -1] \end{aligned} \tag{14}$$

where $x_d - x_0 \geq 0$ and $x_s = \frac{1}{2\sqrt{\xi^2 - 1}}[-(\xi + \sqrt{\xi^2 - 1})e^{-(\xi + \sqrt{\xi^2 - 1})\omega_n t} +(\xi - \sqrt{\xi^2 - 1})e^{(-\xi + \sqrt{\xi^2 - 1})\omega_n t}]$. $\mathtt{max}(x_s)$ w.r.t. ξ is shown in Fig. 2. The upper bound, denoted by Z_M, varies according to ξ.

IV. Reachable Domain

Given a controller and a geometrically feasible initial point, we define the region in which the system can move without violating the positive tension criteria as the reachable domain. The points that belong to this region are called feasible points. The goal of this section is to find a conservative estimate of the reachable domain as a set of inequalities. The procedure to search for such a domain can be summarized as follows: (i) the states of the end-effector

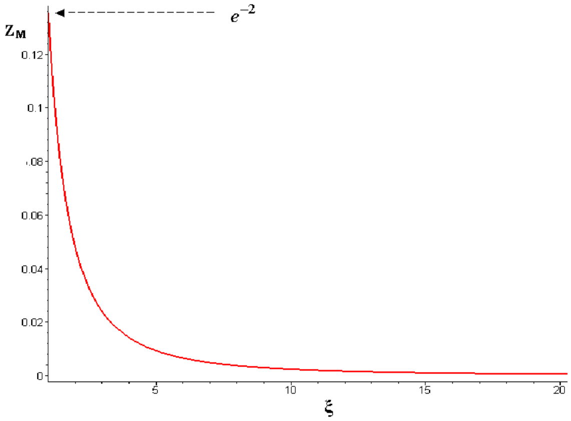

Fig. 2. The plot of x_s with respect to ξ

are first expressed in terms of set points (see Eq. (12), (13)). (ii) we transform the input constraints into those of set points, by making conservative bound on the terms so that the positive input constraints are satisfied during motion. Hence, this method always provides a *conservative bound* on the reachable domain.

There is little difference in the overall approach between the overdamped and the critically damped case in calculating the reachable domain. Hence, we consider only the overdamped case for a translation motion in a xy plane. The reachable domain of the xy motion is given as follows (see [2]):

$$\begin{bmatrix} \frac{\sqrt{3}z_c\omega_n^2 Z_M}{z_c\omega_n^2+g} & 1 \\ 0 & 1 \\ 0 & 1 \\ \sqrt{3} & 1 \\ \sqrt{3} & 1 \\ \sqrt{3}\frac{z_c\omega_n^2+g}{z_c\omega_n^2 Z_M} & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \end{bmatrix} \leq \begin{bmatrix} \frac{\sqrt{3}(z_c\omega_n^2 Z_M+g)}{z_c\omega_n^2+g} & \frac{z_c\omega_n^2}{z_c\omega_n^2+g} \\ 0 & \frac{z_c\omega_n^2}{z_c\omega_n^2+g} \\ 0 & \frac{z_c\omega_n^2 Z_M+g}{z_c\omega_n^2 Z_M} \\ \frac{\sqrt{3}(z_c\omega_n^2 Z_M+g)}{z_c\omega_n^2 Z_M} & \frac{z_c\omega_n^2 Z_M+g}{z_c\omega_n^2 Z_M} \\ \sqrt{3}\frac{z_c\omega_n^2}{z_c\omega_n^2+g} & \frac{z_c\omega_n^2}{z_c\omega_n^2+g} \\ \frac{\sqrt{3}}{Z_M} & \frac{z_c\omega_n^2 Z_M+g}{z_c\omega_n^2 Z_M} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} +$$

$$+ \begin{bmatrix} \frac{ag}{z_c\omega_n^2+g} \\ \frac{ag}{2(z_c\omega_n^2+g)} \\ \frac{ag}{2z_c\omega_n^2 Z_M} \\ \frac{ag}{z_c\omega_n^2 Z_M} \\ \frac{ag}{z_c\omega_n^2+g} \\ \frac{ag}{\sqrt{3}z_c\omega_n^2 Z_M} \end{bmatrix} . \qquad (15)$$

The feasible region can be sketched by the six linear inequalities. In the following sections, Eq. (15) is used for (i) comparison between a numerical method and the proposed analytical method, (ii) describing an algorithm to extend the feasible region. Results for other submotions such as x, y, xyz, and ϕ can be found similarly.

V. Algorithm to Expand the Reachable Domain

A. Feasible initial points

In the previous section, we have presented a set of inequalities, which guarantee reachable points from a given initial point (x_0, y_0). However, the obtained reachable domain may not describe the entire region of the actual workspace, since the technique obtains only a subspace of the entire workspace. To overcome this shortcoming, it is important to investigate initial points which have non-empty reachable domains, since such initial points allow us to generate a family of reachable domains by utilizing the analytical framework of Eqs. (15) repeatedly. As a result, the repetitive calculation of reachable domains, based on the feasible initial values, provides us with an useful tool to extend a workspace. The computation associated with this problem in not expensive since the solutions Eqs. (15) do not change in form except the dependence on initial values.

TABLE I

System parameters in MKS unit, unless specified in the table

Sys. Parameter	Value	RD Parameter	Value
a	0.45	$O1$	(0,0)
b	0.25	$O2$	(0.08,0.08)
I_1, I_2	0.018	$O3$	(0.12,0.11)
I_3	0.075	$O4$	(0.14,0.15)
m	6	$O5$	(0.15,0.17)
λ	3	z_c	1
η	3		

To find a set of initial points with a non-empty reachable domain, we put $x_d = x_0$ and $y_d = y_0$ as the worst case into Eqs. (15). This procedure comes up with a set of inequalities as follows

$$\begin{aligned} y_0 &\leq \sqrt{3}x_0 + a \\ y_0 &\leq \tfrac{a}{2} \\ y_0 &\leq -\tfrac{a}{2} \\ y_0 &\geq -\sqrt{3}x_0 - a \\ y_0 &\leq -\sqrt{3}x_0 + a \\ y_0 &\leq \sqrt{3}x_0 - a \ . \end{aligned} \qquad (16)$$

The feasible domain of these initial values is the same as the static workspace, which can be easily verified by replacing acceleration terms in Eq. (13) by $\ddot{x} = \ddot{y} = 0$. This result implies that the analytical workspace shown in Eqs. (15) can be used repeatedly for any point within the static workspace. Fig. 3 shows a family of reachable domains generated from different initial points $(O, O1, O2, O3, O4)$. The procedure to generate sequential reachable domains is as follows: After abtaining the first reachable domain ($RD1$) evaluated at O, find $RD2$ from $O1$ located within $RD1$ to ensure non-empty intersection. In a similar fashion, one can get $RD3$, $RD4$, and so on. Finally, the procedure results in a chain of reachable domains where neighboring domains possess common intersection points such as $O1, O2, \cdots$, from which new reachable workspaces were recalculated. This extends the reachable domain of the end effector as shown in Fig. 3.

The system parameters and control gains are listed in Table I. Note that $O1, O2, \cdots$ in Fig. 3 were chosen arbitrarily

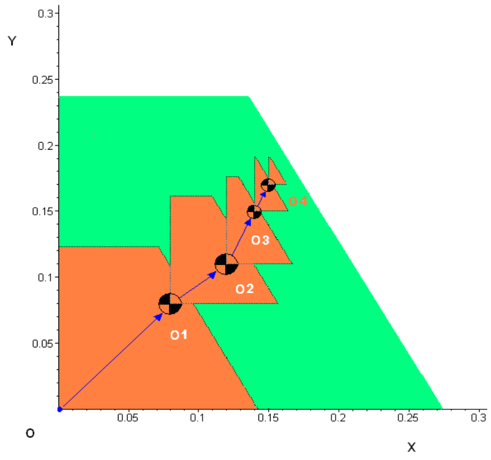

Fig. 3. The generation of a consecutive reachable domains in a $x-y$ plane

to show the feasibility of the technique to generate a family of reachable domains. In the following section, we introduce a systematic way to select evalution points $O1, O2$, and so on.

B. Algorithm to select set points ($\mathbf{x}_d^i$)

Let us define $\mathbf{x}_d^i$ as the set point of the i *th* reachable domain, or an inital value of the $(i+1)$ *th* reachable domain. One constructs the following linear programing problem to generate i *th* desired value $\mathbf{x}_d^i$, where A and $\mathbf{b}$ are the matrix representation of Eqs. (15). In addition, $\mathbf{x}_f$ is defined as the goal point.

$$\begin{array}{rl} & min(|\mathbf{x_d^i}(\beta) - \mathbf{x}_f|) \\ & \mathbf{x}_d^i(\beta) = \mathbf{x}_f + \beta(\mathbf{x}_d^{i-1} - \mathbf{x}_f) \\ s.t. & A\mathbf{x_d^i}(\beta) \geq \mathbf{b}, \end{array} \qquad (17)$$

where i is the index of iterations. In the above equation, β varies over $[0\ \ 1]$. The following bisection algorithm provides a systematic way to search for β closest to 0.

1) If $\beta = 0$ is admissible, set $\beta = 0$ and stop.
2) Otherwise, set $\beta^- = 0$, $\beta^+ = 1$.
3) Set $\beta = \frac{1}{2}(\beta^- + \beta^+)$.
4) If β is admissible, set $\beta^+ = \beta$. Otherwise, $\beta^- = \beta$.
5) Execute steps (3), (4), and (5) $(N-1)$ times.
6) Set $\beta = \beta^+$ and stop.

In summary, the linear programming problem posed above produces the set point $\mathbf{x}_d^i$ of the reachable domain RDi, closest to the goal point $\mathbf{x}_f$. Then, $\mathbf{x}_d^i$ is used as initial point for $RD(i+1)$. $\mathbf{x}_d^{i+1}$, $\mathbf{x}_d^{i+2}, \cdots$ are determined in a similar way. Using a piece-wise step signal to connect these set points, one can increase the range of motion of the end-effector, while satisfying the system constraints.

VI. Simulation

Fig. 4 shows a flow chart to calculate a piece-wise step signal used in the simulation and a following experiment. First of all, one needs to check if the system enters a steady state, since Eqs. (15) were built under the assumption $\dot{\mathbf{x}}_0 = 0$. Then, the bisection algorithm updates the value of β, N times over $\beta = [0,\ 1]$ and each time the input constraints are checked.

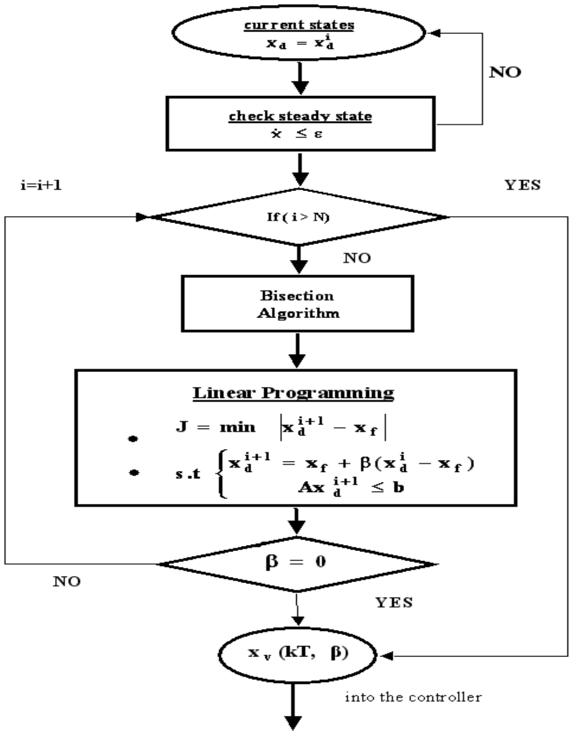
current states
$\mathbf{x_d} = \mathbf{x_d^i}$
NO
check steady state
$\dot{\mathbf{x}} \leq \varepsilon$
i=i+1
YES
If (i > N)
NO
Bisection Algorithm
Linear Programming
$J = \min \left|\mathbf{x}_d^{i+1} - \mathbf{x}_f\right|$
s.t $\mathbf{x}_d^{i+1} = \mathbf{x}_f + \beta(\mathbf{x}_d^i - \mathbf{x}_f)$
$A\mathbf{x}_d^{i+1} \leq \mathbf{b}$
β = 0
NO
YES
$\mathbf{x}_v$ (kT, β)
into the controller

Fig. 4. The flow chart shows the proposed algorithm to create a new reference signal

As shown in the flow chart, the analytical solution of Eqs. (15) to estimate the system's reachable domain makes the procedure straight forward and *on-line implementable*.

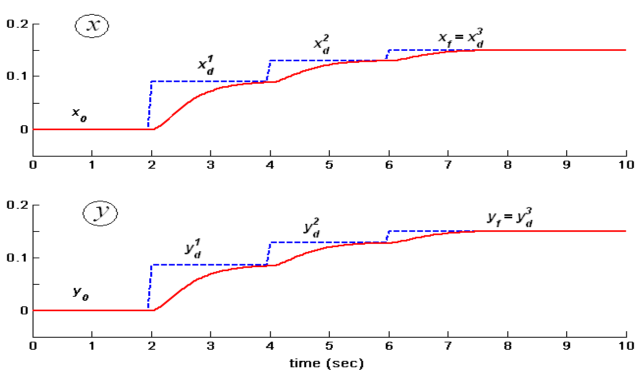

Fig. 5. Simulation plots for an $x-y$ maneuver of the end effector.

We consider a specific move of the end-effector from $\mathbf{x}_0 = [0, 0,\ \ 1.0, 0^o, 0^o, 0^o]^T$ to

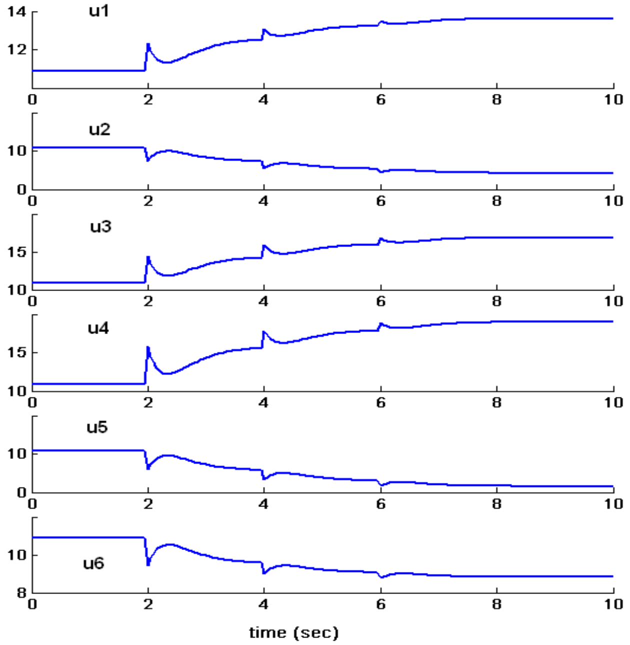

Fig. 6. Simulation plots of tensions for the $x-y$ maneuver shown in Fig. 5: Desired(dashed line), actual(solid line)

$\mathbf{x}_f = [0.15, 0.15, 1.0, 0^o, 0^o, 0^o]^T$. All signals are in MKS unit. We are assured that $\mathbf{x}_0$ is an admissible reference signal since initial tension values at $t = 0$ are positive. Fig. 5 shows the trajectories of the states $\mathbf{x}$ and the desired signals $\mathbf{x}_d^i$, $i = 1, 2, 3$. The set point $\mathbf{x}_d^i$ is propagated as follows:

$$\mathbf{x}_d = \underbrace{\begin{bmatrix}0\\0\end{bmatrix}}_{\mathbf{x}_0} \mapsto \overbrace{\begin{bmatrix}0.09\\0.085\end{bmatrix}}^{\mathbf{x}_d^1} \mapsto \underbrace{\begin{bmatrix}0.13\\0.128\end{bmatrix}}_{\mathbf{x}_d^2} \mapsto \overbrace{\begin{bmatrix}0.15\\0.15\end{bmatrix}}^{\mathbf{x}_d^3=\mathbf{x}_f}, \tag{18}$$

Here, Lyapunov controller is in charge of the stability and the performance of the control system, while a trajectory planner based on reachable domains generates an admissible piecewise step signals ($\mathbf{x}_d^i$, $1 = 1, 2, 3$) by solving the linear programming of Eqs. (17). Cable tension graphs in Fig. 6 show that positive input constraints are satisfied by u_1 through u_6, where u_5 approaches zero. In addition, we sketched guaranteed reachable domains to observe if those values $x_d^i, i = 1, 2, 3$ are within the corresponding reachable domains. As shown in Fig. 7, $\mathbf{x}_d^i, i = 1, 2, 3$ are not only inside the corresponding reachable domains ($RDi, i = 1, 2, 3$), but also close to $\mathbf{x}_f$ between its initial value $\mathbf{x}_d^{i-1}$ and final value $\mathbf{x}_f$. These results are consistent with the problem statement of Linear programming posed in Eqs. (17). In addition, x_d^3 is set to x_f since x_f is within $RD3$.

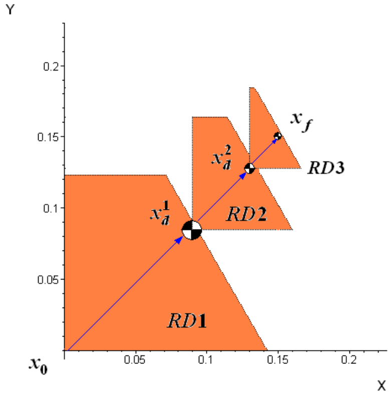

Fig. 7. The generation of a family of reachable domains in a $x-y$ plane

TABLE II
SYSTEM(RG) PARAMETERS IN MKS UNIT, UNLESS SPECIFIED IN THE TABLE

Sys. Parameter	Value	RD Parameter	Value
a	0.45	N	8
b	0.25	T_s	50 msec
I_1, I_2	0.018	η	3
I_3	0.075	λ	3
m	6		

VII. EXPERIMENT

The control of the cable-driven robot is through a dSpace 1103 board which does the data acquisition and control of the system. A dSpace ControlDesk acts as a front end interface. The control code is written in MATLAB Simulink and is downloaded on to the dSpace board using Real-Time Workshop. The mechanical design consists of an end-effector, suspended by six cables, and driven by pulleys (see Fig. 1). Direct-drive servomotors from Kollmorgen, fitted with encoders, drive each cable. Each cable has a force sensor in series to measure the tension being transmitted to the end-effector at any time during motion.

All required numerical calculations in the algorithms, including inversion of a 6×6 Jacobian matrix and the linear programming algorithm, are coded in C to achieve real time implementations. In addition, a force control loop using six loadcell sensors, shown in Fig. 1, is used to compensate for potential sources of error in the experiment due to cable friction in the drive train.

The initial and final configurations are given by $\mathbf{x}_0 = [0, 0, 1.0, 0^o, 0^o, 0^o]^T$ and $\mathbf{x}_d = [0.15, 0.15, 1.0, 0^o, 0^o, 0^o]^T$. The sysem parameters used in the experiment are the same as in simulation and are listed in Table I. The results of implementation using the guaranteed reachable domains based feedback controller are given in Figs. 8 and 9. As shown in Fig. 8, the system response of the experiment shows a good match

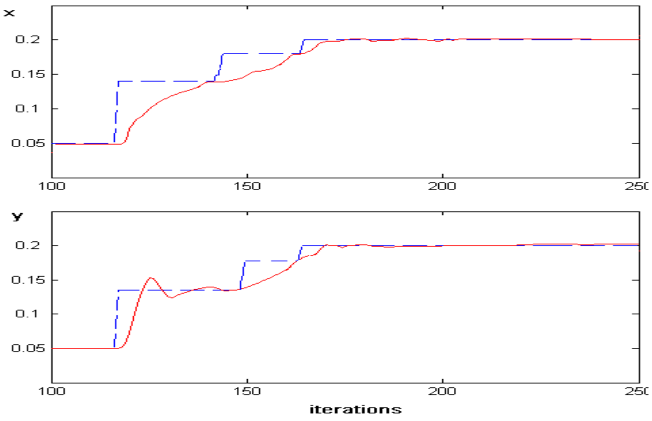

Fig. 8. Experiment results showing the plots of actual positions x, y of the end-effector.

with simulation results in Fig. 5. The dotted lines in Fig. 8 are reference signals obtained by the linear programming problem in Eq. (17), which are similar to those of the simulation (see Fig. 5). The position and velocity of the end-effector are calculated precisely, based on encoder readings and the forward kinematics. Overall, the controller achieves the desired steady-state values with a small error, which may be due to system uncertainties such as friction in the pulley. Fig. 9 show that all tensions are positive and the fifth cable tension approaches zero, as predicted in the Simulation.

VIII. Conclusions

This paper addressed the issue of control design for a non-redundant six degree-of-freedom cable robot with positive input constraints. The design is based on a Lyapunov controller augmented with a trajectory planner built upon guaranteed reachable domains. We formulated an *on-line implementable* linear programming problem to generate sequential reachable domains and searched for feasible set points which are closest to the goal point.

Prior to characterizing a family of reachable domains, the framework described in Eqs. (15) proved to have non-empty reachable domain within the entire static workspace. This offered to us the rationale to use the formulation iteratively to enlarge the motion range of the end-effector. Through computer simulations, the effectiveness of this controller was shown. Finally, using the reachable domain obtained by the proposed method, we were able to experimentally verify set-point control on an in-house developed cable-suspended robot. The results showed a good match between the theory and the experiment.

Acknowledgment

The authors appreciate financial supports of NSF Award No. IIS-0117733, NIST MEL Award No. 60NANB-2D0137, PTI/NIST Award No. AGR20020506, and NIST Award No. SB 1341-03-W-0338.

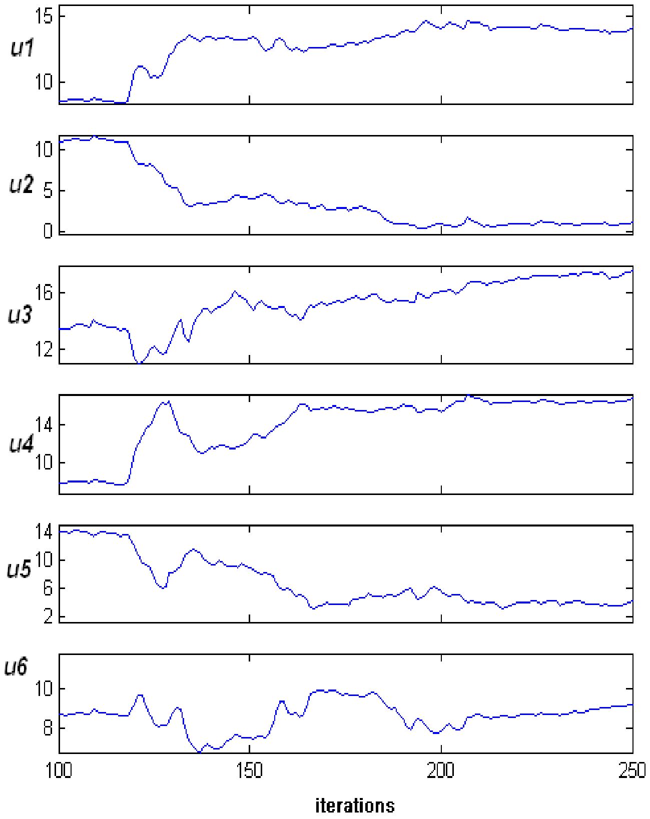

Fig. 9. Experiment results for $x - y$ move in Fig. 8 showing the plots of cable tensions.

References

[1] Oh, S. R. and Agrawal, S. K., "Sliding Mode Control and Feasible Workspace Analysis for a Cable Suspened Robot", *2004 American Control Conference*, paper no. FrM19.1, Boston, 2004.

[2] Oh, S. R. and Agrawal, S. K., "Feasible Set Points with Positive Tensions for a Cable Suspended Robot with a Lyapunov Controller", *2004 ASME International Mechanical Engineering Congress and R/D Expo*, IMECE2004-61539, Anaheim.

[3] Rawlings, J. B., " Tutoral Overview of Model Predictive Control", *IEEE Control Systems Magazine: SPECIAL SECTION Industrial Process Control*, 38-52, June, 2002.

[4] Findeisen, R. and Allgower, F., " An Introduction to Nonlinear Model Predictive Control", *21st Benelux Meeting on System and Control*, Veldhoven, 1-23, 2002.

[5] Gilbert, E. G., Kolmanovsky, I., and Tan, K. T., "Nonlinear Control of Discrete Time Linear Systems with State and Control Constraints: A Reference Governor with Global Convergence Properties", *Proceedings of the IEEE Conference on Decision and Control*, Orlando, FL, 1994.

[6] Bemporad A., Casavola A., and Mosca E., "Nonlinear Control of Constrained Linear Systems via Predictive Reference Management", *IEEE Transactions on Automatic Control*, vol. 42, pp. 340-349, 1997.

[7] Isidori, A. "Nonlinear Control Systems: an Introduction", *Berlin: Springer Verlag*, 1995.

[8] Alp, A. B. and Agrawal, S. K., "Cable Suspended Robots: Design, Planning and Control", *Proceedings of International Conference on Robotics and Automation*, Washington, DC, 4275-4280, 2002.