Randomized Kinodynamic Planning for Cable-suspended Parallel Robots

Ricard Bordalba, Josep M. Porta, and Lluís Ros

Abstract This paper proposes the use of a randomized kinodynamic planning technique to synthesize dynamic motions for cable-suspended parallel robots. Given two mechanical states of the robot, both with a prescribed position and velocity, the method attempts to connect them by a collision-free trajectory that respects the joint and force limits of the actuators, keeps the cables in tension, and takes the robot dynamics into account. The method is based on the construction of a bidirectional rapidly-exploring random tree over the state space. Remarkably, the technique can be used to cross forward singularities of the robot in a predictable manner, which extends the motion capabilities beyond those demonstrated in previous work. The paper describes experiments that show the performance of the method in point-to-point operations with specific cable-driven robots, but the overall strategy remains applicable to other mechanism designs.

1 Introduction

Cable-suspended parallel robots consist of a moving load hanging from a fixed base by means of cables. The load configuration can be changed by varying the cable lengths or anchor point locations, and gravity is typically used to maintain the cables under tension. As opposed to fully-constrained parallel cable-driven robots, cable-suspended robots are not redundantly actuated, and generally employ as many actuators as the number of degrees of freedom to be governed.

A fundamental issue in such robots is to guarantee that the cable tensions remain positive at all times. In this way, cable slackness is avoided, which allows the control of the load using proper tension adjustments. Traditionally, these robots have been

Ricard Bordalba, Josep M. Porta, and Lluís Ros
Institut de Robòtica i Informàtica Industrial, CSIC-UPC
E-mails: {`rbordalba,porta,lros`}`@iri.upc.edu`

used as robotic cranes, setting them to operate in static or quasi-static conditions, in which gravity is the sole source of cable tension [1, 5]. While this simplifies the planning and control of the motions, it also confines them to the static workspace, which is a region limited by the footprint of the robot. More recently, inertia has also been proposed as another source of tension [11], extending the movement capacity to the dynamic workspace, i.e., the region that can be attained when load accelerations are allowed [3]. Using pendulum-like motions, for example, pick-and-place tasks between points well beyond the robot footprint can be planned [10, 14].

In this new context, there is a strong need for an efficient planning technique that determines the force inputs required to move the robot between two mechanical states (both with a prescribed position and velocity). Such planner must avoid collisions of the robot with itself or with the environment while obeying the physical laws imposed by the motion equations, and the force and joint limits of the actuators. This problem, known as kinodynamic planning in the literature [16], is gaining attention in cable-driven robotics [9]. Early work in this regard includes planning methods for a remarkable mechanism like the Winch-bot, which can follow prescribed paths on a vertical plane with just a single actuator [8], or evolved architectures with additional manipulation abilities [18, 22, 23]. Because of their underactuation, however, these robots cannot control their pose exactly along the motion, which motivated the development of newer methods for fully-actuated designs. For instance, in [11] cyclic trajectories leaving the static workspace were given, and the approach was later extended to synthesize point-to-point trajectories for pick-and-place tasks [10, 14]. Optimal control methods for Robocrane-type platforms were also provided in [2]. While such methods are remarkable, none of them were designed to avoid collisions. The methods in [10, 14], moreover, rely on predefined trajectories, and they need some guidance to define intermediate waypoints when the start and goal configurations fail to be connected by such trajectories.

Particular solutions for specific robots are valuable, but it is the authors' belief that existing randomized techniques can solve the kinodynamic planning problem with great generality in cable-driven robots. The purpose of this paper is, precisely, to show that a recent method of this kind [7] can successfully cope with the kinematic, collision, and positive-tension constraints arising in such robots. The method is based on deploying an exploration tree over the state space and it is probabilistically complete, i.e., it finds a connecting trajectory whenever one exists and enough computing time is available. Remarkably, we show that the method can also be used to cross forward singularities in a predictable manner, which further extends the motion capabilities beyond those envisaged in earlier work [10, 14]. The method is a generalization of a classic planning method [17]. Whereas the approach in [17] was suitable for mechanisms described by means of independent generalized coordinates, the one in [7] can also handle dependent coordinates coupled by kinematic constraints, which often arise in parallel mechanisms. After reviewing the contributions of [17] and [7] (Section 2), we show how the resulting technique can be applied to cable-driven robots (Section 3), and illustrate its performance on challenging problems (Section 4). We finally summarize the main strengths of the approach, and points deserving further attention (Section 5).

2 A kinodynamic motion planner

The planning of dynamic motions typically takes place in the state space of the robot, i.e., the set $\mathscr{X}$ of kinematically-valid states $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$, where $\boldsymbol{q}$ is a vector of n_q generalized coordinates describing the configuration of the robot, and $\dot{\boldsymbol{q}}$ is the time derivative of $\boldsymbol{q}$, which describes its velocity. The coordinates in $\boldsymbol{q}$ may be independent or not. In the former case, any pair $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{2n_q}$ is kinematically valid, and $\mathscr{X}$ becomes parametrically defined. The latter case is more complex. The configuration space (C-space) of the robot is the set $\mathscr{C}$ of points $\boldsymbol{q}$ that satisfy a system of n_e nonlinear equations

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0} \tag{1}$$

encoding, e.g., loop-closure constraints, or geometric constraints due to nonminimal representations of $SO(3)$. As a result, the valid values of $\dot{\boldsymbol{q}}$ are those that fulfill

$$\boldsymbol{\Phi_q}\, \dot{\boldsymbol{q}} = \boldsymbol{0}, \tag{2}$$

where $\boldsymbol{\Phi_q} = \partial \boldsymbol{\Phi}/\partial \boldsymbol{q}$. Then, $\mathscr{X}$ becomes a nonlinear manifold of dimension $d_{\mathscr{X}} = 2(n_q - n_e)$ generically, defined implicitly by Eqs. (1) and (2).

Irrespective of the form of $\mathscr{X}$, the motions must always be confined to a feasibility region $\mathscr{X}_{\text{feas}} \subseteq \mathscr{X}$ of collision-free states respecting joint and constraint force limits (such as tension positivities in cable-driven robots). Finally, the motions must also obey the dynamic equations of the robot, which can be written in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}). \tag{3}$$

In this equation, $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$ is an appropriate differentiable function, and $\boldsymbol{u}$ is a d-vector of actuator forces subject to lie in a bounded subset $\mathscr{U} \subset \mathbb{R}^d$. Then, given start and goal states, $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$, the kinodynamic planning problem consists in finding a time function $\boldsymbol{u}(t)$ such that the system trajectory $\boldsymbol{x}(t)$ determined by Eqs. (1)-(3) for $\boldsymbol{x}(0) = \boldsymbol{x}_s$, fulfills $\boldsymbol{x}(t_f) = \boldsymbol{x}_g$ for some time $t_f > 0$, and $\boldsymbol{u}(t) \in \mathscr{U}$, $\boldsymbol{x}(t) \in \mathscr{X}_{\text{feas}}$ for all $t \in [0, t_f]$.

The solution proposed in [17] assumes that the $\boldsymbol{q}$ coordinates are independent, so that Eqs. (1) and (2) need not be considered. The resulting planner looks for a solution by constructing a rapidly-exploring random tree (RRT) over $\mathscr{X}$. The RRT is rooted at $\boldsymbol{x}_s$ and it is grown incrementally towards $\boldsymbol{x}_g$ while staying inside $\mathscr{X}_{\text{feas}}$. Every tree node stores a feasible state $\boldsymbol{x} \in \mathscr{X}_{\text{feas}}$, and every edge stores the action $\boldsymbol{u} \in \mathscr{U}$ needed to move between the connected states. This action is assumed to be constant during the move. The expansion of the RRT proceeds by applying three steps repeatedly (Fig. 1, top-left). First, a state $\boldsymbol{x}_{rand} \in \mathscr{X}$ is randomly selected; then, the RRT state $\boldsymbol{x}_{near}$ that is closest to $\boldsymbol{x}_{rand}$ is computed according to some metric; finally, a movement from $\boldsymbol{x}_{near}$ towards $\boldsymbol{x}_{rand}$ is performed by applying an action $\boldsymbol{u} \in \mathscr{U}$ during a fixed time Δt. The movement from $\boldsymbol{x}_{near}$ towards $\boldsymbol{x}_{rand}$ is simulated by integrating Eq. (3) numerically, which yields a new state $\boldsymbol{x}_{new}$ that may or may not be in $\mathscr{X}_{\text{feas}}$. In the former case $\boldsymbol{x}_{new}$ is added to the RRT, and in the latter it
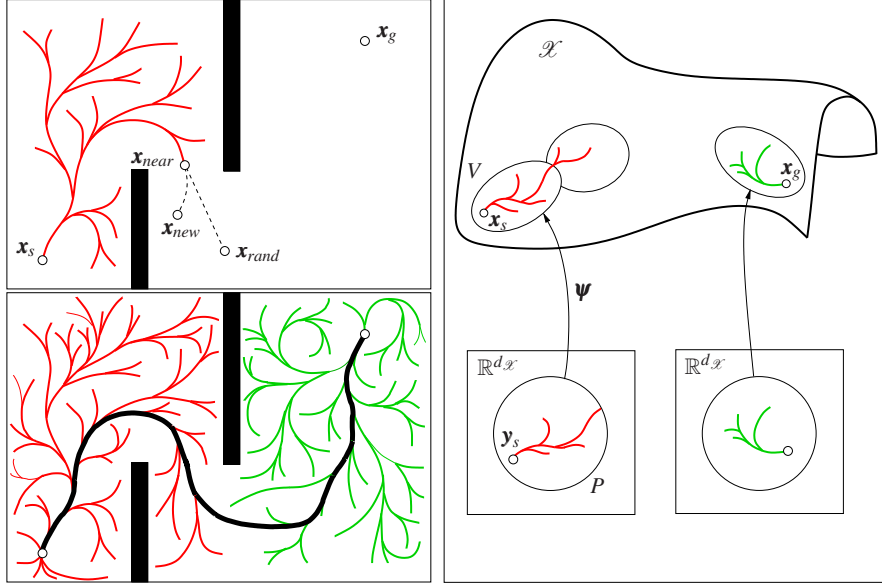
Fig. 1: Left-Top: Extension process of an RRT. Left-Bottom: A kinodynamic planning problem is often solved faster with a bidirectional RRT. Right: Construction of an RRT on an implicitly-defined state space manifold.

is discarded. To test whether $\boldsymbol{x}_{new} \in \mathscr{X}_{\text{feas}}$, $\boldsymbol{x}_{new}$ is checked for collisions by using standard algorithms [15], and the joint positions and constraint forces are computed to check whether they stay within bounds. The action $\boldsymbol{u}$ applied is typically chosen as the one from $\mathscr{U}$ that brings the robot closer to $\boldsymbol{x}_{rand}$. One can either try all possible values in $\mathscr{U}$ (if it is a discrete set) or only those of n_s random points on $\mathscr{U}$ (if it is continuous). To force the RRT to extend towards $\boldsymbol{x}_g$, $\boldsymbol{x}_{rand}$ is set to $\boldsymbol{x}_g$ once in a while, stopping the whole process when a RRT leaf is close enough to $\boldsymbol{x}_g$. Usually, however, a solution trajectory can be found more rapidly if two RRTs respectively rooted at $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$ are grown simultaneously towards each other (Fig. 1, left-bottom). The expansion of the tree rooted at $\boldsymbol{x}_g$ is based on the integration of Eq. (3) backward in time.

The previous strategy has proved to be effective in many situations, but in parallel robots the coordinates in $\boldsymbol{q}$ are often dependent. This fact complicates the generation of RRTs over $\mathscr{X}$, because there is no straightforward way to randomly select points $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$ satisfying Eqs. (1) and (2), and the numerical integration of Eq. (3) easily drifts away from $\mathscr{X}$ when standard methods for ordinary differential equations are used. These two issues have been recently circumvented in [7] by constructing an atlas of $\mathscr{X}$ in parallel to the RRT.

An atlas is a collection of charts mapping $\mathscr{X}$ entirely, where each chart is a local diffeomorphism $\boldsymbol{\psi}$ from an open set $P \subseteq \mathbb{R}^{d_{\mathscr{X}}}$ of parameters to an open set $V \subset \mathscr{X}$ (Fig. 1, right). The V sets can be thought of as partially-overlapping tiles

covering $\mathscr{X}$, in such a way that every $\boldsymbol{x} \in \mathscr{X}$ lies in at least one set V. Assuming that an atlas is available, the problem of sampling $\mathscr{X}$ boils down to generating random values $\boldsymbol{y}$ in the P sets, since these values can always be projected to $\mathscr{X}$ using $\boldsymbol{x} = \boldsymbol{\psi}(\boldsymbol{y})$. Also, the atlas allows the conversion of the vector field defined on $\mathscr{X}$ by Eq. (3) into one in the coordinate spaces P, which permits the integration of Eq. (3) using local coordinates [21]. As a result, the RRT motions satisfy Eqs. (1) and (2) by construction, eliminating any drift from $\mathscr{X}$ to machine precision.

As explained in [7], the construction of the atlas is incremental. The atlas is initialized with two charts covering $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$, respectively (Fig. 1, right). Then, these charts are used to pull the expansion of the RRT, which in turn adds new charts to the atlas as needed, until $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$ become connected. To be able to construct the charts, the method requires $\mathscr{X}$ to be smooth, which implies that the robot cannot exhibit C-space singularities, i.e., points $\boldsymbol{q}$ for which $\boldsymbol{\Phi_q}$ is rank deficient [4, 6]. In practice, the exclusion of such singularities can be achieved by choosing appropriate mechanism dimensions, since generically $\boldsymbol{\Phi_q}$ will be full rank. Another choice is to set joint limits excluding the presence of such singularities.

3 Application to a cable-suspended robot

To apply the previous method to a specific cable-suspended robot we need to obtain Eqs. (1)-(3) and verify that $\boldsymbol{\Phi_q}$ is full rank over the C-space. Moreover, to determine whether a given $\boldsymbol{x}$ belongs to $\mathscr{X}_{\mathrm{feas}}$, note that the joint limits can be trivially checked, and we can use the methods in [15] to detect the collisions. Thus, we only need to provide a means to compute the cable tensions for each $\boldsymbol{x} \in \mathscr{X}$. We next illustrate these points in the particular robot of Fig. 2.

3.1 Kinematic model

Consider a point mass suspended from three cables of fixed length ρ_i, $i = 1,2,3$. Each cable is connected to an actuated slider that can move along a guide line defined by a point $\boldsymbol{A}_i$, with position vector $\boldsymbol{a}_i$, and a unit vector $\boldsymbol{w}_i$, both given in a fixed frame O_{xyz} (Fig. 2). Although the guides are horizontal and parallel in this figure, they could take any direction in general. By changing the displacements $\boldsymbol{d} = [d_1, d_2, d_3]^\mathsf{T}$ of the sliders, the robot can control the position $\boldsymbol{P}$ of the point mass, with position vector $\boldsymbol{p} = [x, y, z]^\mathsf{T}$ in the mentioned frame. In this robot, it is natural to choose $\boldsymbol{q} = [x, y, z, d_1, d_2, d_3]^\mathsf{T}$ so that Eq. (1) becomes the system formed by

$$\rho_i^2 - \boldsymbol{c}_i^\mathsf{T} \boldsymbol{c}_i = 0, \tag{4}$$

for $i = 1,2,3$, where $\boldsymbol{c}_i = (\boldsymbol{a}_i + d_i \boldsymbol{w}_i) - \boldsymbol{p}$ is the vector from $\boldsymbol{P}$ to the ith slider position. By taking the partial derivatives of Eqs. (4) with respect to $\boldsymbol{p}$ and $\boldsymbol{d}$ we obtain
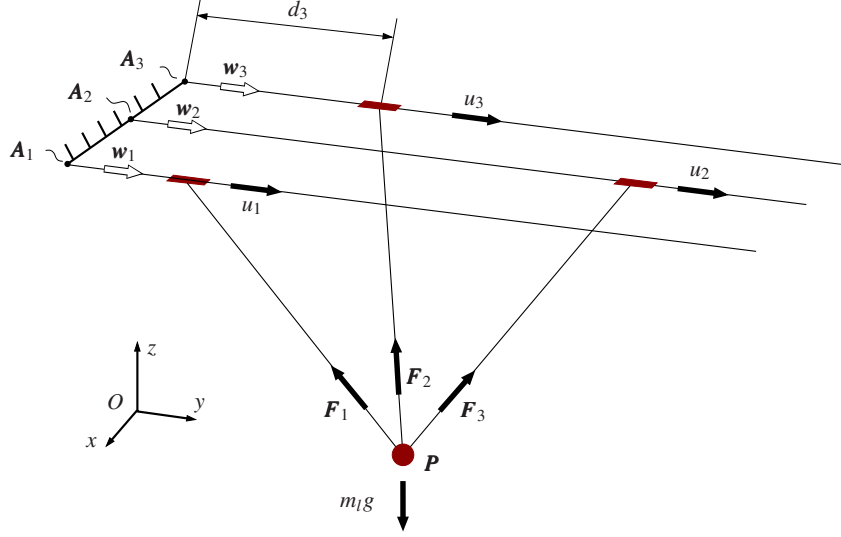
Fig. 2: A spatial 3-DOF cable-driven robot.

$$\boldsymbol{\Phi_q} = \underbrace{\begin{bmatrix} 2\boldsymbol{c}_1{}^\mathsf{T} \\ 2\boldsymbol{c}_2{}^\mathsf{T} \\ 2\boldsymbol{c}_3{}^\mathsf{T} \end{bmatrix}}_{\boldsymbol{\Phi_p}} \underbrace{\begin{matrix} -2\boldsymbol{c}_1{}^\mathsf{T}\boldsymbol{w}_1 & 0 & 0 \\ 0 & -2\boldsymbol{c}_2{}^\mathsf{T}\boldsymbol{w}_2 & 0 \\ 0 & 0 & -2\boldsymbol{c}_3{}^\mathsf{T}\boldsymbol{w}_3 \end{matrix}}_{\boldsymbol{\Phi_d}}, \tag{5}$$

which readily provides Eq. (2).

By inspection of the previous Jacobian, it is easy to see that a configuration $\boldsymbol{q}$ is a C-space singularity if, and only if, the three cables lie on a plane π orthogonal to the three guides. Certainly, if for a given $\boldsymbol{q}$ all cables lie on such a plane π, the subjacobian $\boldsymbol{\Phi_d}$ is null, and the subjacobian $\boldsymbol{\Phi_p}$ is rank deficient. This implies that all 3×3 minors of $\boldsymbol{\Phi_q}$ will vanish, so that $\boldsymbol{q}$ is a C-space singularity. Conversely, if not all cables lie on such a plane π, but they are still on a plane not orthogonal to the guides, $\boldsymbol{\Phi_p}$ is rank deficient but $\boldsymbol{\Phi_d}$ will be full rank. If the cables are not coplanar, $\boldsymbol{\Phi_p}$ is full rank. In any of the two situations, therefore, $\boldsymbol{\Phi_q}$ will be full rank.

In what follows, we shall assume that our robot does not exhibit C-space singularities. To ensure so, note that it suffices to choose cable lengths for which it is impossible to assemble the mechanism with all of its cables stretched and lying on the plane π just described.

The configurations in which $\boldsymbol{\Phi_p}$ is rank deficient are the so-called forward singularities of the mechanism [6]. In these singularities, the velocities of the actuators do not determine the velocity of the end effector if only Eq. (2) is considered. However, in the next section we shall see that, dynamically, the evolution of the mechanism is perfectly predictable across such singularities.

3.2 Dynamic model

To formulate Eq. (3), we use the Euler-Lagrange equations with multipliers [12] which lead to a compact treatment of dynamics and are easily applicable to other cable-driven architectures. These equations take the form

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{q}} - \frac{\partial K}{\partial q} + \frac{\partial U}{\partial q} + \boldsymbol{\Phi_q}^\mathsf{T}\boldsymbol{\lambda} = \boldsymbol{\tau}, \tag{6}$$

where K and U are the expressions of the kinetic and potential energies of the robot, $\boldsymbol{\lambda}$ is a vector of n_e Lagrange multipliers, and $\boldsymbol{\tau}$ is the generalized force corresponding to the non-conservative forces applied on the system.

In the robot of Fig. 2, we assume that the cables have negligible mass, and let m_l and m_s refer to the mass of the moving load and the mass of each slider, respectively. By defining $\boldsymbol{M}_l = m_l\boldsymbol{I}_3$ and $\boldsymbol{M}_s = m_s\boldsymbol{I}_3$, where $\boldsymbol{I}_3$ is the 3×3 identity matrix, the kinetic energy of the robot is given by

$$K = \frac{1}{2}\begin{bmatrix} \dot{\boldsymbol{p}}^\mathsf{T} & \dot{\boldsymbol{d}}^\mathsf{T} \end{bmatrix}\begin{bmatrix} \boldsymbol{M}_l & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_s \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{d}} \end{bmatrix} = \frac{1}{2}\dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{M}\dot{\boldsymbol{q}}, \tag{7}$$

where $\boldsymbol{M}$ is the so-called mass matrix, which is always symmetric and positive definite. The potential energy of the robot, on the other hand, is given by

$$U = m_l g z, \tag{8}$$

where g is the gravitational acceleration. By substituting Eq. (7) into Eq. (6), the Euler-Lagrange equations of our robot reduce to

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + U_{\boldsymbol{q}} + \boldsymbol{\Phi_q}^\mathsf{T}\boldsymbol{\lambda} = \boldsymbol{\tau}. \tag{9}$$

where the term $U_{\boldsymbol{q}}$ is given by the partial derivatives of Eq. (8), i.e.,

$$U_{\boldsymbol{q}} = [0,0,m_l g,0,0,0]^\mathsf{T}. \tag{10}$$

Also, assuming for simplicity that all contacts are frictionless, and letting u_i denote the force exerted by the ith actuator, we have

$$\boldsymbol{\tau} = [0,0,0,u_1,u_2,u_3]^\mathsf{T}. \tag{11}$$

Since Eq. (9) is a system of n_q equations in $n_q + n_e$ unknowns (the values of $\ddot{\boldsymbol{q}}$ and $\boldsymbol{\lambda}$), we need extra equations to be able to solve for $\ddot{\boldsymbol{q}}$. These can be obtained by differentiating Eq. (2), which yields

$$\boldsymbol{\Phi_q}\ddot{\boldsymbol{q}} - \boldsymbol{\xi} = \boldsymbol{0}, \tag{12}$$

where $\boldsymbol{\xi} = -(\boldsymbol{\Phi_{qq}}\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$. Eqs. (9) and (12) can then be written as

$$\begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{\Phi}_{\boldsymbol{q}}^{\top} \\ \boldsymbol{\Phi}_{\boldsymbol{q}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - U_{\boldsymbol{q}} \\ \boldsymbol{\xi} \end{bmatrix}. \tag{13}$$

Clearly, if $\boldsymbol{\Phi}_{\boldsymbol{q}}$ is full rank, i.e. there are no C-space singularities, the matrix on the left-hand side of Eq. (13) is invertible, even at forward singularities, and thus we can write

$$\ddot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{I}_{n_q} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{\Phi}_{\boldsymbol{q}}^{\top} \\ \boldsymbol{\Phi}_{\boldsymbol{q}} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\tau} - U_{\boldsymbol{q}} \\ \boldsymbol{\xi} \end{bmatrix}. \tag{14}$$

To finally obtain Eq. (3), we transform Eq. (14) into a first-order ordinary differential equation using the change of variables $\dot{\boldsymbol{q}} = \boldsymbol{v}$, yielding

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},\boldsymbol{u}) \end{bmatrix} = \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}). \tag{15}$$

3.3 Tension computation

Let $\boldsymbol{F}_i$ denote the force applied by the ith cable on the moving load (Fig. 2). Such a force can be written as $\boldsymbol{F}_i = \boldsymbol{c}_i F_i / \rho_i$, where F_i is the tension of the ith cable. We next see that the tensions F_i can be obtained from the Lagrange multipliers $\boldsymbol{\lambda}$. Note that Eq. (9) can be decomposed into

$$\boldsymbol{M}_l \, \ddot{\boldsymbol{p}} = [0,0,-m_l g]^{\mathsf{T}} - \boldsymbol{\Phi}_{\boldsymbol{p}}^{\mathsf{T}} \boldsymbol{\lambda}, \tag{16}$$

$$\boldsymbol{M}_s \, \ddot{\boldsymbol{d}} = [u_1,u_2,u_3]^{\mathsf{T}} - \boldsymbol{\Phi}_{\boldsymbol{d}}^{\mathsf{T}} \boldsymbol{\lambda}, \tag{17}$$

which correspond, respectively, to Newton's 2nd law applied to the load and the sliders. Using Eq. (16), for instance, we see that the term $-\boldsymbol{\Phi}_{\boldsymbol{p}}^{\mathsf{T}} \boldsymbol{\lambda}$ must be the resultant force applied by the cables $\boldsymbol{F}_c = \boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3$, because the other two terms are the weight of the load and the time derivative of its linear momentum. Thus we can say that $\boldsymbol{F}_c = -\boldsymbol{\Phi}_{\boldsymbol{p}}^{\mathsf{T}} \boldsymbol{\lambda}$, or, using the value of $\boldsymbol{\Phi}_{\boldsymbol{p}}$ in Eq. (5),

$$\boldsymbol{F}_c = 2\boldsymbol{c}_1 \lambda_1 + 2\boldsymbol{c}_2 \lambda_2 + 2\boldsymbol{c}_3 \lambda_3. \tag{18}$$

On the other hand, $\boldsymbol{F}_c$ can also be written as

$$\boldsymbol{F}_c = \frac{\boldsymbol{c}_1}{\rho_1} F_1 + \frac{\boldsymbol{c}_2}{\rho_2} F_2 + \frac{\boldsymbol{c}_3}{\rho_3} F_3, \tag{19}$$

and comparing Eqs. (18) and (19) we obtain $F_i = 2\rho_i \lambda_i$. We note this expression for F_i could also have been obtained by departing from Eq. (17) instead. Moreover, since the robot is assumed to be free from C-space singularities, the $\boldsymbol{\lambda}$ values are always determined by Eq. (13), implying that the tensions F_i will be determined too, even at forward singularities.

4 Experiments

The planner has been implemented in C, and it has been integrated into the CUIK Suite [20]. To illustrate its performance, we next show three experiments of increasing complexity. The first two experiments involve a planar version of the robot of Fig. 2, whereas the last experiment is three-dimensional. In all cases the mass of the load is 1 [kg], the mass of each slider is 0.1 [kg], and the force applied by the sliders is limited to the range $[-8, 8]$ [N], so that $\mathscr{U} = [-8, 8]^d$, with $d \in \{2, 3\}$. A bidirectional RRT is always constructed, and each time it is extended, $n_s = 25$ actions are randomly sampled from $\mathscr{U}$. Each of these actions is applied during $\Delta t = 0.5$ [s].

For each experiment, Table 1 summarizes the values of n_q, n_e, and $d_\mathscr{X}$, as well as the performance statistics on an iMac with an Intel i7 processor with 8 CPU cores running at 2.93 Ghz. The statistics include the number of samples and charts generated, and the planning time in seconds, all averaged over ten runs. The planner successfully connected the start and goal states in all runs. Finally, the table also indicates the execution time, t_f, for the trajectories of Fig. 3.

Experiment 1: Moving in the dynamic workspace

In this example, the load is suspended from two cables, and the sliders move along vertical guides [Fig. 3(a)]. The cables and the sliders move on different planes and, thus, their collisions need not be checked. The distance between the two guides is 2 [m], and the cables' length is 8 [m]. The goal here is to move the load from a low position to a higher position, both in rest and outside the static workspace. The load has to oscillate along the trajectory in order to gain momentum and finally reach the goal. The smaller the allowable force on the motors, the larger the number of oscillations and the harder the planning problem. The bidirectional RRT created encompasses two trees rooted at the start and goal states, shown in red and green respectively. Note that although the robot has a limited static workspace (the region between the guides), including dynamics in the planning has increased the usable workspace substantially.

Experiment	n_q	n_e	$d_\mathscr{X}$	No. of samples	No. of charts	Planning Time [s]	t_f [s]
1	4	2	4	1928	283	12.6	10.1
2	4	2	4	20946	2622	140	7.1
3	6	3	6	24398	2244	234	8.4

Table 1: Problem dimensions and performance statistics for the shown experiments.
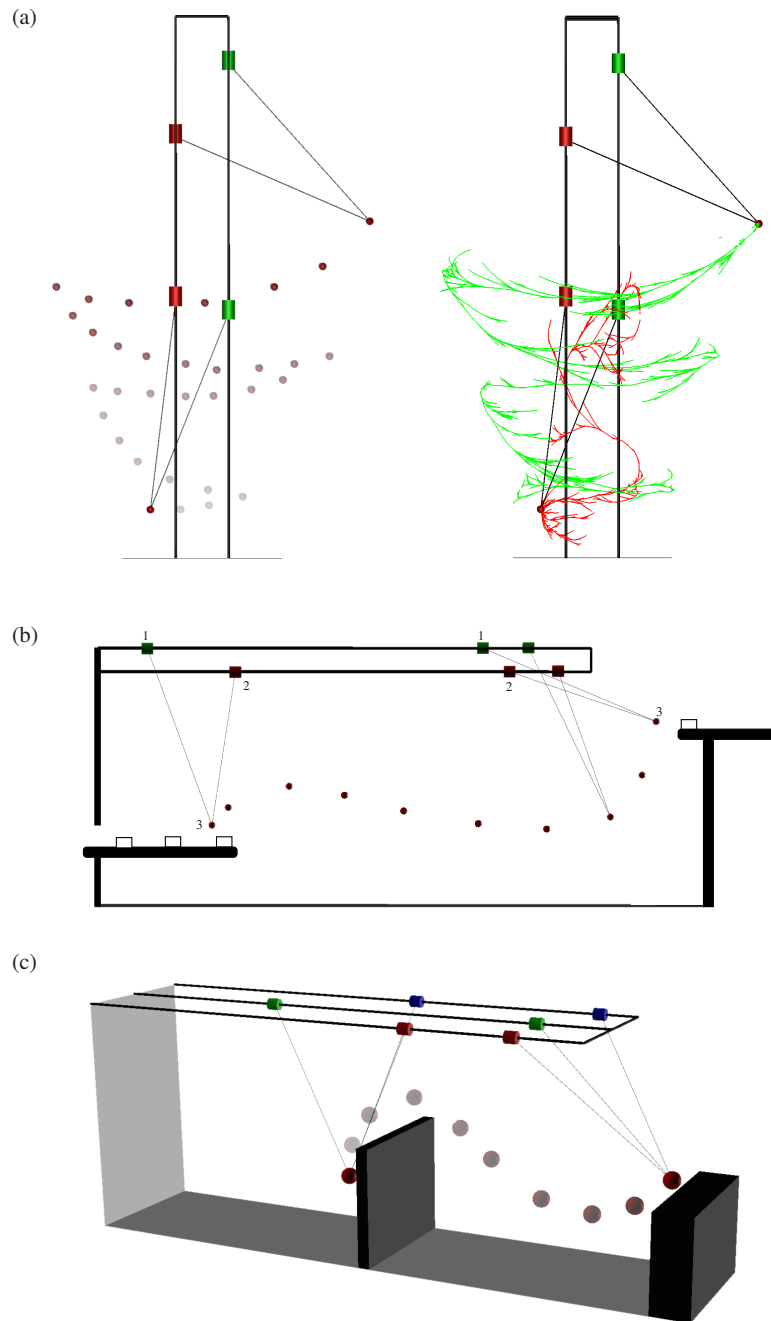
(a)

(b)

(c)

Fig. 3: Three planning problems on planar and spatial versions of the robot.

Experiment 2: Singularity crossing

We now consider the robot of Fig. 3(b), in which the load is suspended from two horizontal guides separated 1 [m] from each other. The lengths of the cables have been set to 6.6 and 8 [m], which allows them to align at 45° relative to the guides. The planning problem consists in finding a trajectory to move the load between the left and rightmost positions in the figure, assuming that the cables cannot collide with the guides nor with themselves. Note that the triangle 1-2-3 has a different orientation at the start and goal positions, so that the robot will have to cross a forward singularity to connect them. Although the inverse static problem is indeterminate in such a singularity, we have shown how both the tensions and the evolution of the robot remain dynamically determined (Sections 3.2 and 3.3). The planner, as a result, has no trouble in computing the shown trajectory, which certainly crosses the singularity somewhere between the two configurations depicted on the right.

Experiment 3: Obstacle avoidance

Finally, a 3D cable-driven robot with three horizontal guides is used to demonstrate how obstacles are avoided. The distance between two consecutive guides is 3 [m], and the cables' length is 8 [m]. The robot moves from a rest position inside the static workspace to another position outside of it. Both positions are separated by a wall in the middle of the workspace, which has to be avoided during the move. The robot is able to overpass the obstacle and manages to reach the goal as seen in Fig. 3(c).

5 Conclusions

This paper has shown how a recent randomized kinodynamic planning technique can be applied to generate dynamic trajectories for cable-suspended parallel robots. Taking into account the system dynamics enlarges the robot workspace substantially, allowing to reach points further apart from the footprint of the supporting structure. Moreover, the joint consideration of obstacle avoidance, force and joint limits, positive tension constraints, and singularity crossings makes the planner applicable to challenging scenarios. The approach has been validated with experiments on particular architectures, but it remains applicable to other robot designs.

The trajectory directly returned by the planner is smooth in position, but not in velocity and acceleration. A point deserving further attention, thus, is the application of local optimization techniques to obtain twice-differentiable trajectories. Also, global optimization methods should be developed to obtain trajectories involving minimum-time or energy consumption [13, 19]. Finally, efforts should be devoted to enhance the metric used to measure the distance between states, which is known to be a challenging task in all sampling-based kinodynamic planners.

References

1. Albus, J., Bostelman, R.V., Dagalakis, N.: The NIST Robocrane. Journal of Robotic Systems **10**(5), 709–724 (1993)
2. Bamdad, M.: Time-energy optimal trajectory planning of cable-suspended manipulators. In: Cable-Driven Parallel Robots, pp. 41–51. Springer (2013)
3. Barrette, G., Gosselin, C.: Determination of the dynamic workspace of cable-driven planar parallel mechanisms. Transactions of the ASME-R-Journal of Mechanical Design **127**(2), 242–248 (2005)
4. Bohigas, O., Manubens, M., Ros, L.: Singularities of non-redundant manipulators: A short account and a method for their computation in the planar case. Mechanism and Machine Theory **68**, 1–17 (2013)
5. Bohigas, O., Manubens, M., Ros, L.: Planning wrench-feasible motions for cable-driven hexapods. IEEE Transactions on Robotics **32**(2), 442–451 (2016)
6. Bohigas, O., Manubens, M., Ros, L.: Singularities of robot mechanisms: Numerical computation and avoidance path planning. Springer (2016)
7. Bordalba, R., Ros, L., Porta, J.M.: Kinodynamic planning on constraint manifolds (2017). Institut de Robòtica i Infomàtica. CSIC-UPC. Technical Report
8. Cunningham, D., Asada, H.H.: The Winch-bot: A cable-suspended, under-actuated robot utilizing parametric self-excitation. In: IEEE International Conference on Robotics and Automation, pp. 1844–1850 (2009)
9. Gosselin, C.: Cable-driven parallel mechanisms: State of the art and perspectives. Mechanical Engineering Reviews **1**(1), DSM0004–DSM0004 (2014)
10. Gosselin, C., Foucault, S.: Dynamic point-to-point trajectory planning of a two-DOF cable–suspended parallel robot. IEEE Transactions on Robotics **30**(3), 728–736 (2014)
11. Gosselin, C., Ren, P., Foucault, S.: Dynamic trajectory planning of a two-DOF cable-suspended parallel robot. In: IEEE International Conference on Robotics and Automation, pp. 1476–1481 (2012)
12. Greenwood, D.T.: Advanced dynamics. Cambridge University Press (2006)
13. Hauser, K., Zhou, Y.: Asymptotically optimal planning by feasible kinodynamic planning in a state-cost space. IEEE Transactions on Robotics **32**(6), 1431–1443 (2016)
14. Jiang, X., Gosselin, C.: Dynamic point-to-point trajectory planning of a three-DOF cable-suspended parallel robot. IEEE Transactions on Robotics **32**(6), 1550–1557 (2016)
15. Jiménez, P., Thomas, F., Torras, C.: 3D collision detection: A survey. Computers & Graphics **25**(2), 269–285 (2001)
16. LaValle, S.M.: Planning Algorithms. Cambridge University Press, New York (2006)
17. LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. International Journal of Robotics Research **20**(5), 378–400 (2001)
18. Lefrançois, S., Gosselin, C.: Point-to-point motion control of a pendulum-like 3-DOF underactuated cable-driven robot. In: IEEE International Conference on Robotics and Automation, pp. 5187–5193 (2010)
19. Li, Y., Littlefield, Z., Bekris, K.E.: Asymptotically optimal sampling-based kinodynamic planning. The International Journal of Robotics Research **35**(5), 528–564 (2016)
20. Porta, J.M., Ros, L., Bohigas, O., Manubens, M., Rosales, C., Jaillet, L.: The Cuik Suite: Analyzing the motion of closed-chain multibody systems. IEEE Robotics and Automation Magazine **21**(3), 105–114 (2014)
21. Potra, F.A., Yen, J.: Implicit numerical integration for Euler-Lagrange equations via tangent space parametrization. Journal of Structural Mechanics **19**(1), 77–98 (1991)
22. Zanotto, D., Rosati, G., Agrawal, S.K.: Modeling and control of a 3-DOF pendulum-like manipulator. In: IEEE International Conference on Robotics and Automation, pp. 3964–3969 (2011)
23. Zoso, N., Gosselin, C.: Point-to-point motion planning of a parallel 3-DOF underactuated cable-suspended robot. In: IEEE International Conference on Robotics and Automation, pp. 2325–2330 (2012)