Time-Optimal Trajectory Planning of Cable-Driven Parallel Mechanisms for Fully Specified Paths With G1-Discontinuities

Eric Barnett*, **Clément Gosselin**
Départment de génie mécanique, Université Laval
Québec, QC, Canada, G1V 0A6
Email: eric.barnett.1@ulaval.ca, gosselin@gmc.ulaval.ca

Time-optimal trajectory planning (TOTP) is a well-studied problem in robotics and manufacturing, which involves the minimization of the time required for the operation point of a mechanism to follow a path, subject to a set of constraints. A TOTP technique, designed for fully specified paths that include abrupt changes in direction, was previously introduced by the first author of this paper: an incremental approach called minimum-time trajectory shaping (MTTS) was used. In the current paper, MTTS is converted to a dynamic technique and adapted for use with cable-driven parallel robots, which exhibit cable tension and motor torque constraints. For many applications, cable tensions along a path are verified after trajectory generation, rather than imposed during trajectory generation. For the technique proposed in this paper, the cable-tension constraints are imposed directly and fully integrated with MTTS, during trajectory generation, thus maintaining a time-optimal solution. MTTS is applied to a test system and path, and compared to the bang-bang technique. With the same constraints, the results obtained with both techniques are found to be very close. However, MTTS can be applied to a wider variety of paths, and accepts constraints on jerk and total acceleration that would be difficult to apply using the bang-bang approach.

1 Introduction

The time-optimal trajectory planning (TOTP) problem involves the minimization of the time required for the operation point of a mechanism to follow a path, subject to constraints on velocity, acceleration, jerk, force, torque, and other parameters. The problem has received extensive treatment in the literature; both dynamic and kinematic techniques have been disclosed. With the former, the torque and/or force required for a manipulator to follow a path are specified directly. For the latter, kinematic constraints must dominate, and a torque and/or force controller is assumed to be available to produce specified kinematic quantities.

A second distinction exists between *online* and *offline* methods. For online methods, trajectory generation takes place while the positioning system is completing the task at hand, while for offline methods, trajectories are generated before the system starts the task. Obviously, online implementation places a strict limit on computational complexity. Therefore, if path geometry is known before the task is to be executed, an offline technique is normally preferable. Further distinctions among TOTP techniques include the ability to handle partially specified or fully specified paths, and the ability to handle G^1 path discontinuities, which consist of abrupt changes in direction. These distinctions will be discussed in more detail as the related work is introduced.

Time-optimal trajectory planning for cable-driven parallel mechanisms is also constrained by the need for cables to remain in positive tension at all times. Negative tension in any cable causes a loss of manipulator positioning precision and unpredictable behavior. Maximum cable tension constraints are also needed, in some cases. A spatial three-degree-of-freedom cable-suspended robot is shown in Fig. 1. It consists of a point-mass end-effector, suspended by three cables, whose lengths are controlled by fixed, actuated spools.

Positioning system hardware constraints are normally expressed most directly as maximum forces and torques. Therefore, dynamic techniques will usually result in a faster TOTP solution, compared to kinematic techniques. Several dynamic techniques for time-optimal control have been proposed in the field of robotics [1–4]. Most of these techniques use bang-bang control, whereby acceleration switching points are identified along a path, at which actuators switch from maximum torque to minimum or vice-versa.

Expressing positioning system constraints as kinematic quantities is often approximate. Therefore, kinematic TOTP solutions will usually be slower than dynamic techniques. However, many applications involve kinematic *process* constraints. For example, a maximum speed constraint is common for dispensing applications. The dynamic techniques

Preprint submitted to the ASME Journal of Dynamic Systems, Measurement, and Control

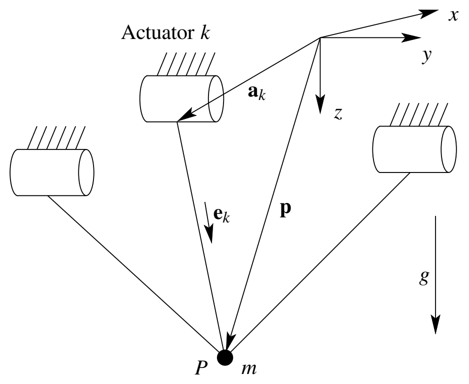


Fig. 1. Spatial three-DOF cable-suspended robot.

listed above all have limited applicability to TOTP problems with kinematic constraints. While velocity constraints and joint acceleration constraints are readily included with the bang-bang method, total linear acceleration and jerk constraints are not.

Several kinematic approaches have been proposed for solving the TOTP problem. In [5], [6], and [7] robot paths are represented using cubic splines, quintic polynomials, and B-splines, respectively. In all of these cases, velocity, acceleration, and jerk constraints can be imposed, and all are applied to *partially specified* paths, i.e., paths with sparsely distributed way points. It is unlikely that they would be suitable for fully specified paths, composed of hundreds or even thousands of closely spaced points, or for paths that contain abrupt changes in direction.

In the manufacturing sector, methods for solving the TOTP problem are typically designed for Cartesian three-axis positioning systems. Since tool path geometry is often fully known beforehand, many methods are offline, designed for fully specified paths.

In [8] and [9], the TOTP problem was solved with the additive manufacturing (AM) application in mind. Elsewhere, paths were parameterized in terms of arc length, and closed-form expressions, such as Bézier curves and NURBs, were used to interpolate over path points [10, 11]. A common problem with all four of these techniques is the inability to handle G^1 discontinuities. Dong et al. [12] used a procedural approach, where paths are sampled into equal-length segments, the problem being solved in the trajectory position-velocity-acceleration phase space, with G^1 discontinuities handled indirectly by limiting the jerk.

A common feature of all of the manufacturing sector techniques is path sampling into equal-length segments, which is suitable for certain algorithm steps, but suboptimal for others, particularly for the final output data. Variable spatial sampling, dependent mainly on path curvature, is often more suitable, since it can be used to provide extra detail when necessary, in highly curved path regions, and less detail in straight regions. Here, it is important to note that a path segment is simply the line joining two adjacent path points, while a path region is composed of multiple adjacent path segments. Uniform space sampling also provides insufficient *time resolution* at the start and end of a path, where the speed is close to zero.

For cable-driven parallel mechanisms, the dynamic constraint of minimum cable tension is often dominant, especially when the end-effector undergoes high acceleration or approaches the boundaries of its workspace. Satisfying this constraint can be accomplished by adjusting constraints on parameters such as acceleration or torque and verifying *a posteriori* that a minimum tension is exhibited by all cables, at all points along the trajectory. This technique is incompatible with the minimum-time objective because it affects the entire trajectory, rather than the specific regions where the tension constraint is violated. Additionally, the tuning process must be performed *separately* for each path to be followed. A technique for directly imposing the tension constraint *during* the solution of the TOTP problem is preferable, since it can ensure that the minimum-time solution is preserved.

In [13] and [14], a series of trajectories for two-DOF and three-DOF cable-suspended parallel robots are defined parametrically as analytic functions of time. Minimum-tension constraints are satisfied by substituting the trajectories into a set of algebraic inequalities that represent constraints on cable tensions, thereby establishing bounds on the trajectory parameters. In [15], redundant cables are used to satisfy positive tension constraints for a three-DOF planar cable-suspended robot.

Trevisani developed a TOTP technique that ensures positive and bounded cable tensions for straight-line and circular paths for a two-DOF cable suspended robot [16, 17]. This technique translates cable tension constraints into velocity and acceleration constraints for the end-effector; the trajectory planning technique used is based on quintic polynomials.

Based on the TOTP techniques presented in [1–3], Behzadipour and Khajepour developed a TOTP technique for cable-based manipulators [18]: minimum cable tension constraints are expressed in the position-velocity phase plane for a trajectory. This technique is suitable for applications where paths and constraints are compatible with the bang-bang approach.

Based on the advantages and drawbacks of the techniques listed above for solving the TOTP problem, a new technique called minimum-time trajectory shaping (MTTS) was developed [19]. MTTS was originally developed as a kinematic, offline technique, which can handle arbitrary, fully specified, three-dimensional paths, with abrupt changes in direction, and can impose velocity, acceleration, and jerk constraints. In this paper, we present a modified version of MTTS, which is a *dynamic* technique, since minimum and maximum cable tension constraints are included. Additionally, motor torque constraints are readily converted to tension constraints for many cable-driven mechanisms. MTTS is applied to a three-DOF cable-driven parallel mechanism for a 3D test path and it is then compared directly to the bang-bang technique. Finally, conclusions and plans for future work are discussed.

2 Minimum-Time Trajectory Shaping (MTTS), For a Three-DOF Cable-Suspended Robot

For a path specified as a series of points in *xyz*-Euclidean space, we define a time vector $\mathbf{t}$ as an array of instants at which the positioning system operation point P—the point at which the task is specified—passes through each path point. The point-to-point duration vector, or array of first-order differences $\Delta\mathbf{t}$, and the second-order difference vector $\Delta^2\mathbf{t}$, then become

$$\Delta\mathbf{t} = \begin{bmatrix} t_2 - t_1 \\ t_3 - t_2 \\ \vdots \\ t_n - t_{n-1} \end{bmatrix}, \quad \Delta^2\mathbf{t} = \begin{bmatrix} t_3 - 2t_2 + t_1 \\ t_4 - 2t_3 + t_2 \\ \vdots \\ t_n - 2t_{n-1} + t_{n-2} \end{bmatrix} \tag{1}$$

with higher-order difference vector arrays defined likewise. Linear interpolation is used to transform $\Delta^2\mathbf{t}$ and $\Delta^3\mathbf{t}$ to vectors of the same dimension as $\Delta\mathbf{t}$. The MTTS optimization problem is defined as

$$\min_{\Delta\mathbf{t}} t_n, \quad t_n = \sum_{i}^{n-1} \Delta t_i \tag{2}$$

subject to constraints on velocity, acceleration, jerk, and cable tension.

MTTS is composed of several subalgorithms. The first step of MTTS is the PATHSPLIT subalgorithm, which is used to split paths into subpaths at abrupt changes in direction, and then re-sample subpath data uniformly in Euclidean space. An initial maximum-speed trajectory is then established for each subpath, using the VCON subalgorithm, based on the system constraints. Discontinuities in acceleration are corrected in the next step, the ACON subalgorithm. Then, acceleration curves from and to rest are integrated with the AREST subalgorithm. Next, jerk discontinuities are identified and eliminated with JCON. Finally, control data for the subtrajectories are assembled to form a single data set for the original path. Cable tension constraints are integrated with MTTS at the VCON, ACON, and AREST steps. A graphical summary of the steps of MTTS is shown in Fig. 11.

2.1 Path Splitting (PATHSPLIT)

The PATHSPLIT subalgorithm is used to break input paths into subpaths at abrupt changes in direction. We define β_i as the segment-to-segment angle at path point i, as shown in Fig. 2. An abrupt change in direction is defined as a path location where the segment-to-segment angle exceeds a threshold value $\beta_{\max}$, which is equivalent to satisfying the inequality

$$\beta_i = \cos^{-1}\left(\frac{\mathbf{d}_i \cdot \mathbf{d}_{i+1}}{\|\mathbf{d}_i\|\,\|\mathbf{d}_i + 1\|}\right) < \beta_{\max}, \tag{3}$$
$$\mathbf{d}_i = (x_{i+1} - x_i)\hat{\mathbf{i}} + (y_{i+1} - y_i)\hat{\mathbf{j}} + (z_{i+1} - z_i)\hat{\mathbf{k}}.$$

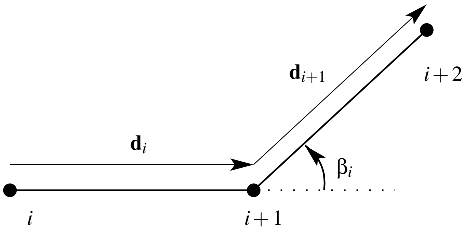


Fig. 2. The segment-to-segment angle, β_i.

Depending on the application, it may or may not be acceptable to smooth path data, before and/or after PATHSPLIT. MTTS was developed primarily for a large-scale cable-driven robot for 3D printing, where the principal objective is to construct a part as accurately as possible, based on the CAD file geometry. If this geometry exhibits sharp edges, then boundary paths with abrupt changes in direction must be followed. Linear interpolation should be used in this case, because smoother interpolation techniques would produce significant error at the abrupt changes in path direction. On the other hand, fill paths can be smoothed because they do not define the part boundary. Smoothing can be accomplished by techniques such as a simple moving average filter or cubic smoothing spline interpolation.

After path splitting has been performed, each subpath is re-sampled into equal-length segments. If the path must be strictly followed, linear interpolation is recommended; otherwise, smoother interpolation methods can be used. The subsequent subalgorithms VCON, ACON, JCON, and AREST are applied *separately* to each subpath, though subpaths will be referred to as paths from now on for conciseness.

2.2 Maximum-Speed Boundary Formation (VCON)

For the VCON and ACON steps of MTTS, paths consist of n supporting points, which are not modified; instead, the point-to-point duration vector $\Delta\mathbf{t}$ is formed and its entries are adjusted. The path position vector $\mathbf{p}_i$ at each path point i, is defined according to

$$\mathbf{p}_i = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}} \tag{4}$$

The unit vector $\mathbf{e}_k$ pointing along cable k from the spool to the end-effector attachment point, is given by

$$\mathbf{e}_k = \frac{\mathbf{p} - \mathbf{a}_k}{\|\mathbf{p} - \mathbf{a}_k\|} = \frac{1}{\rho_k}(\mathbf{p} - \mathbf{a}_k), \quad k = 1,2,3 \tag{5}$$

with all quantities defined in Fig. 1. For cable k, we have velocity $\dot{\rho}_k$, acceleration $\ddot{\rho}_k$, and jerk $\dddot{\rho}_k$. The total linear Cartesian velocity, acceleration, and jerk, are given by

$$v = |\dot{\mathbf{p}}| = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{6a}$$
$$\alpha = |\ddot{\mathbf{p}}| = \sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2} \tag{6b}$$
$$j = |\dddot{\mathbf{p}}| = \sqrt{\dddot{x}^2 + \dddot{y}^2 + \dddot{z}^2} \tag{6c}$$

where α is used rather than a for acceleration because a is used to refer to path points below.

It is important to note that, except for path regions that are straight in Euclidean space, $\alpha \neq \ddot{s}$ and $j \neq \dddot{s}$. A simple trajectory to illustrate the difference between these quantities is a circle of radius r in the xy-plane, traversed at constant speed $\dot{s}$, such that

$$x = r\cos t,\ \dot{x} = -r\sin t,\ \ddot{x} = -r\cos t,\ \dddot{x} = \ \ r\sin t \tag{7a}$$

$$y = r\sin t,\ \dot{y} = \ \ r\cos t,\ \ddot{y} = -r\sin t,\ \dddot{y} = -r\cos t \tag{7b}$$

Here, $\ddot{s}$ and $\dddot{s}$ are both zero, while α and j are both equal to r.

Typical kinematic constraints include maximum magnitudes for velocity, acceleration, and jerk for each cable $\dot{\rho}_{k,\max}$, $\ddot{\rho}_{k,\max}$, $\dddot{\rho}_{k,\max}$, and for the Cartesian parameters in Eq. (6), $v_{\max}$, $\alpha_{\max}$, and $j_{\max}$. In some cases, constraints could also be needed for individual Cartesian directions, and/or for angular jerk.

The objective of VCON is to establish a boundary curve in s-$\dot{s}$, space, as is done initially when applying the bang-bang technique. To produce this curve, we make the important assumption that the end-effector does not need to accelerate to reach the maximum speed, permitted by each constraint, at each path point. Equivalently, we account for the *spatial* interdependence among path points, while neglecting the *temporal* interdependence. Mathematically, we assume Δt_i is independent from Δt_{i+1} and Δt_{i-1}. Under this assumption, the kinematic cable constraints are given by

$$\left|\Delta\rho_{k,i}\right| / \Delta t_i \leq \dot{\rho}_{k,\max} \tag{8a}$$

$$\left|\Delta^2\rho_{k,i}\right| / \Delta t_i^2 \leq \ddot{\rho}_{k,\max} \tag{8b}$$

$$\left|\Delta^3\rho_{k,i}\right| / \Delta t_i^3 \leq \dddot{\rho}_{k,\max} \tag{8c}$$

where the only unknown is Δt_i. For the Cartesian constraints, we discretize Eqs. (6) according to

$$v_i = \sqrt{\left(\frac{\Delta x_i}{\Delta t_i}\right)^2 + \left(\frac{\Delta y_i}{\Delta t_i}\right)^2 + \left(\frac{\Delta z_i}{\Delta t_i}\right)^2} \tag{9a}$$

$$\alpha_i = \sqrt{\left(\frac{\Delta^2 x_i}{\Delta t_i^2}\right)^2 + \left(\frac{\Delta^2 y_i}{\Delta t_i^2}\right)^2 + \left(\frac{\Delta^2 z_i}{\Delta t_i^2}\right)^2} \tag{9b}$$

$$j_i = \sqrt{\left(\frac{\Delta^3 x_i}{\Delta t_i^3}\right)^2 + \left(\frac{\Delta^3 y_i}{\Delta t_i^3}\right)^2 + \left(\frac{\Delta^3 z_i}{\Delta t_i^3}\right)^2} \tag{9c}$$

$$i = 1 \ldots n-1$$

leading to the constraints

$$v_i \leq v_{\max}, \quad \alpha_i \leq \alpha_{\max}, \quad j_i \leq j_{\max}. \tag{10}$$

The path position s_i is given by

$$s_i = \sum_{i=1}^{i} \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2 + (\Delta z_i)^2}, \tag{11}$$

where it is important to note that

$$v_i = \Delta s_i / \Delta t_i, \quad \alpha_i \neq \Delta^2 s_i / \Delta t_i^2, \quad j_i \neq \Delta^3 s_i / \Delta t_i^3 \tag{12}$$

because quantities α_i and j_i represent the total linear acceleration and jerk, respectively, while $\Delta^2 s_i/\Delta t_i^2$ and $\Delta^3 s_i/\Delta t_i^3$ represent *tangential* acceleration and jerk. A similar development could be used to establish constraints on total angular kinematic quantities.

The *geometric* significance of this process is to be highlighted. The total linear acceleration constraint only limits acceleration due to path curvature, or equivalently, acceleration due to change in the direction of the velocity vector. Similarly, the jerk constraint does not affect straight-line motion at this stage, but does have an impact in path locations where there is high jerk due to path geometry, such as the transition from a circular round to a straight line.

In many cases, dynamic constraints, such as restrictions on actuator torque and cable tension, are also needed for cable-driven robots. For simple setups where actuators drive spools of cable, the torque T_k is given by $T_k = r_k F_k$, where r_k is the radius of spool k and F_k is the tension force in cable k. Therefore, in many cases, *all* of the dynamic constraints can be imposed using the equations developed below.

As shown in [13], Newton's second law of motion for a three-DOF spatial cable-suspended robot is given by

$$\mathbf{M}\boldsymbol{\tau} = \mathbf{g} - \boldsymbol{\alpha} \tag{13}$$

where

$$\mathbf{M} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & m_9 \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix},$$

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T = \frac{1}{m}\begin{bmatrix} F_1 & F_2 & F_3 \end{bmatrix}^T,$$

$$\mathbf{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T, \quad \boldsymbol{\alpha} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T,$$

with $\mathbf{e}_k$ being the unit vector pointing along cable k from the spool to the end-effector attachment point, defined in Eq. (5) and shown in Fig. 1. The variable g denotes acceleration due to gravity and $\boldsymbol{\tau}$ is the vector of cable tensions per unit mass m of the end-effector.

Using the assumption that was applied for the kinematic constraints listed above, namely, that the end-effector does not need to accelerate to reach the maximum speed, Eq. (13) is discretized according to

$$\mathbf{M}_i \tau_i = \begin{bmatrix} -\Delta^2 x_i / \Delta t_i^2 \\ -\Delta^2 y_i / \Delta t_i^2 \\ g - \Delta^2 z_i / \Delta t_i^2 \end{bmatrix}. \tag{14}$$

Dropping the i subscript, we obtain

$$\begin{aligned} m_1\tau_1 + m_2\tau_2 + m_3\tau_3 &= -\Delta^2 x/\Delta t^2 \\ m_4\tau_1 + m_5\tau_2 + m_6\tau_3 &= -\Delta^2 y/\Delta t^2 \\ m_7\tau_1 + m_8\tau_2 + m_9\tau_3 &= -\Delta^2 z/\Delta t^2 + g \end{aligned} \tag{15}$$

though it is important to note that all quantities of Eq. (15), except for g, are different for every path point i.

The minimum tension constraint τ_{min} is imposed by setting $\tau_k = \tau_{k,\text{min}}$ for each point along a path. Then, Eq. (15) is solved analytically for the other unknowns, which consist of the tensions in the other cables and the quantity Δt. The maximum tension constraint is imposed in the same manner, thus producing six tension constraints on Δt for each path point i. To render the solutions more compact, the following intermediate variables are introduced

$$\begin{aligned} D_1 &= m_2 m_6 - m_5 m_3, \quad & N_1 &= m_6 \Delta^2 x - m_3 \Delta^2 y \\ D_2 &= m_1 m_6 - m_3 m_4, \quad & N_2 &= m_2 \Delta^2 y - m_5 \Delta^2 x \\ D_3 &= m_1 m_5 - m_4 m_2, \quad & N_3 &= m_1 \Delta^2 y - m_4 \Delta^2 x \\ & R = m_7 D_1 - m_8 D_2 + m_9 D_3. \end{aligned} \tag{16}$$

The minimum tension in Cable 1 is produced by setting $\tau_1 = \tau_{\text{min}}$ and solving Eq. (15) for the vector of unknowns $\begin{bmatrix} \tau_2 & \tau_3 & \Delta t \end{bmatrix}^T$, producing

$$\Delta t = \sqrt{\frac{\Delta^2 z D_1 - m_8 N_1 - m_9 N_2}{g D_1 + R\tau_{\text{min}}}} \tag{17a}$$

$$\tau_2 = \frac{N_1/\Delta t^2 - D_2 \tau_{\text{min}}}{D_1} \tag{17b}$$

$$\tau_3 = \frac{N_2/\Delta t^2 + D_3 \tau_{\text{min}}}{D_1}. \tag{17c}$$

The minimum tensions for the other cables are produced similarly, leading to the three equations

$$\Delta t = \sqrt{\frac{\Delta^2 z D_1 - m_8 N_1 - m_9 N_2}{g D_1 + R\tau_{\text{min}}}} \tag{18a}$$

$$\Delta t = \sqrt{\frac{\Delta^2 z D_2 - m_7 N_1 - m_9 N_3}{g D_2 - R\tau_{\text{min}}}} \tag{18b}$$

$$\Delta t = \sqrt{\frac{\Delta^2 z D_3 + m_7 N_2 - m_8 N_3}{g D_3 + R\tau_{\text{min}}}} \tag{18c}$$

which are applied to each point i of the path. Maximum cable tension constraints are imposed by replacing τ_{min} by τ_{max} in Eq. (18).

It is important to correctly interpret the restrictions τ_{min} and τ_{max} place on Δt. For every path point, either the maximum *or* the minimum tension constraint will produce a *minimum* restriction on Δt, with the other constraint producing an imaginary value for Δt, which can be ignored. This conclusion is now justified, based on the equations used to produce the tension constraints.

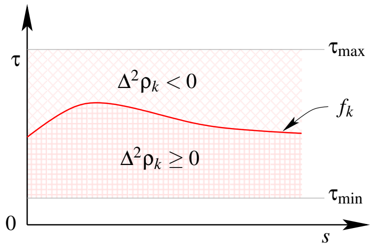


Fig. 3. Valid tension ranges for joint k along one path, during VCON

In, Eq. (14), we can separate the tension due to gravitational and inertial effects by writing

$$\boldsymbol{\tau}_i = \mathbf{M}_i^{-1}\mathbf{g} - \mathbf{M}_i^{-1}\boldsymbol{\alpha}_i \tag{19}$$

where $\mathbf{M}_i^{-1}\boldsymbol{\alpha}_i$ is equal to $\begin{bmatrix} \ddot{\rho}_{i,1} & \ddot{\rho}_{i,2} & \ddot{\rho}_{i,3} \end{bmatrix}^T$, the vector of cable accelerations. For path point i and cable k, we can therefore write

$$\tau_{i,k} = f_{i,k} - \ddot{\rho}_{i,k} \approx f_{i,k} - \frac{\Delta^2 \rho_{i,k}}{\Delta t_i^2} \tag{20}$$

where f_k is the gravitational contribution to the force in each cable, shown in Fig. 3 along one path. Solving for Δt_i, we have

$$\Delta t_i = \sqrt{\frac{\Delta^2 \rho_{i,k}}{f_{i,k} - \tau_{i,k}}}. \tag{21}$$

We assume that the path is chosen such that

$$\tau_{\text{min}} \leq f_{i,k} \leq \tau_{\text{max}} \tag{22}$$

as shown in Fig. 3. Therefore, to produce a positive real Δt_i, the tension constraints for a particular point i on a trajectory are restricted as follows

$$\tau_{\text{min}} \leq \tau_{i,k} \leq f_{i,k} \quad \text{if} \quad \Delta^2 \rho_{i,k} \geq 0 \tag{23a}$$

$$f_{i,k} < \tau_{i,k} \leq \tau_{\text{max}} \quad \text{if} \quad \Delta^2 \rho_{i,k} < 0. \tag{23b}$$

At each point, either the maximum *or* the minimum tension constraint produce a valid restriction on Δt_i; the other constraint produces an imaginary Δt_i.

As a simple example, we consider the case of a single cable raising and lowering a weight. If the mass is accelerating upward at point i, i.e., if $\Delta^2 \rho$ is negative, the tension force in the cable must be greater than the weight force.

The minimization problem for VCON is given by Eq. (2), subject to constraints based on Eqs. (8), (10), and

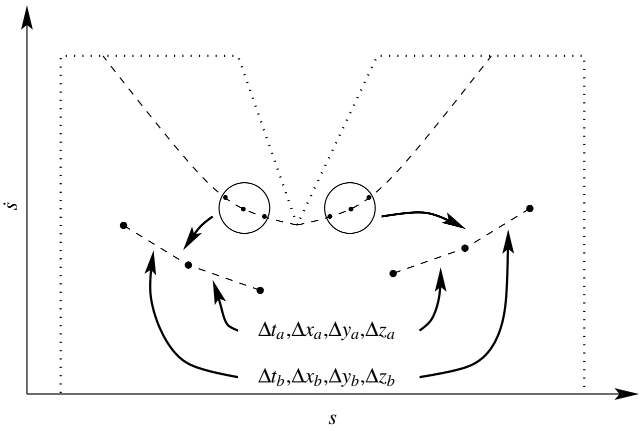

$\dot{s}$
$\Delta t_a,\Delta x_a,\Delta y_a,\Delta z_a$
$\Delta t_b,\Delta x_b,\Delta y_b,\Delta z_b$
s

Fig. 4. Illustration of the ACON subalgorithm.

(18). The calculations needed to produce $\mathbf{\Delta t}$ are all explicit and straightforward. As mentioned above, at this stage, $\mathbf{\Delta t}$ represents a maximum-speed envelope for the path, which can be expressed in s-$\dot{s}$ space. Subsequent steps of MTTS are constrained to shape the trajectory *within* this envelope. For the test path used in the case study, the blue dotted line of Fig. 11 corresponds to this envelope.

2.3 Elimination of Acceleration Discontinuities (ACON)

The ACON subalgorithm is used to apply the same constraints as in Subsec. 2.2, except we now consider that the end-effector *does* need to accelerate to reach the maximum allowed speed at each path point. Equivalently, the maximum allowed speed at path point i is assumed to depend on the speed at points $i-1$ and $i+1$.

ACON operates using the following procedure. All path points in s-$\dot{s}$ are initially assumed to be constraint-violating. The local minimum in s-$\dot{s}$, among the constraint-violating points, is identified, and constraint-satisfying curves are swept leftward and rightward from this point, until they intersect the maximum-speed boundary curve. The points along these constraint-satisfying curves are removed from the list of constraint-violating points, and the procedure continues until this list is empty. It is important to note that, while speed-increase jerk constraints are imposed using this procedure, speed-decrease jerk is dealt with using JCON. Figure 4 shows the curve-sweeping procedure for one local minimum in s-$\dot{s}$.

All constraints imposed during VCON used the approximation $dt \approx \Delta t_i$ for point i. Under the new assumption, we use the approximation $dt \approx (\Delta t_i + \Delta t_{i+1})/2$. Applying this assumption to Eq. (8a), we obtain

$$\frac{\left|(\Delta\rho_{k,i}+\Delta\rho_{k,i+1})/2\right|}{(\Delta t_i+\Delta t_{i+1})/2} \le \dot{\rho}_{k,\max}. \tag{24}$$

It becomes clear that, if this constraint is satisfied at points i and $i+1$, then it is also satisfied for Eq. (24). The same conclusion can be drawn for all constraints at the velocity level.

The acceleration constraint, Eq. (8b), becomes

$$\frac{\left|\Delta\rho_{k,b}/\Delta t_b - \Delta\rho_{k,a}/\Delta t_a\right|}{(\Delta t_a+\Delta t_b)/2} \le \ddot{\rho}_{k,\max}. \tag{25}$$

where we set $a=i$ and $b=i+1$ for conciseness. Assuming that Δt_a is known, we can produce the constraint relations

$$\begin{aligned} 2(\Delta\rho_{k,b}\Delta t_a - \Delta\rho_{k,a}\Delta t_b) &\le \\ &\Delta t_a\Delta t_b(\Delta t_a+\Delta t_b)\ddot{\rho}_{k,\max} \qquad (26a) \\ 2(\Delta\rho_{k,b}\Delta t_a - \Delta\rho_{k,a}\Delta t_b) &\ge \\ &-\Delta t_a\Delta t_b(\Delta t_a+\Delta t_b)\ddot{\rho}_{k,\max} \qquad (26b) \end{aligned}$$

where Δt_b is the only unkown. The minimum, real, positive solution for Δt_b is chosen among the four solutions to these two quadratic equations. This solution is guaranteed to be the minimum solution for Δt_b that respects Eq. (25). A proof of this assertion is now provided.

Since a is a point on an acceleration sweeping curve, the speed at a is below the s-$\dot{s}$ curve, shown in Fig. 4. Equivalently, if we let $a=i$ in Eq. 8b, then $\Delta t_a > \Delta t_i$. Therefore, the solution $\Delta t_b = \Delta t_a$ will always satisfy Eqs. (26), and there must be at least one positive real solution that respects these constraints.

We first consider positive $\Delta\rho_{k,b}$. In Eq. (26b), if $\Delta\rho_{k,a}$ is also positive, then positive Δt_b is only produced if

$$\Delta t_b \ge \frac{\Delta\rho_{k,b}}{\Delta\rho_{k,a}}\Delta t_a. \tag{27}$$

Therefore, solutions to Eq. (26b) can only cause a *decrease* in joint speed under these conditions, and they can therefore be discarded. If $\Delta\rho_{k,a}$ is negative, no positive real solution for Δt_b exists.

Considering Eq. (26a), the quadratic roots are given by

$$\begin{aligned} \Delta t_b = \frac{-B \pm \sqrt{B^2-4AC}}{2A}, \quad & A = \Delta t_a\ddot{\rho}_{k,\max}, \qquad (28) \\ B = \Delta t_a^2\ddot{\rho}_{k,\max} + 2\Delta\rho_{k,a}, \quad & C = -2\Delta t_a\Delta\rho_{k,b}. \end{aligned}$$

We see that both roots are positive only if B is negative and $0 \le 4AC \le B^2$. However, $\Delta\rho_{k,b}$ is positive, so $4AC$ is negative and there will be one positive real solution for Δt_b. Therefore, the optimal value that satisfies Eqs. (26) lies in the range

$$0 \le \Delta t_b \le \frac{\Delta\rho_{k,b}}{\Delta\rho_{k,a}}\Delta t_a \tag{29}$$

if both $\Delta\rho_{k,a}$ and $\Delta\rho_{k,b}$ are positive. If $\Delta\rho_{k,a}$ is negative, the term $4AC$ remains the same, so there is still only one positive real solution for Δt_b. Additionally, we know that the optimal Δt_b will satisfy $\Delta t_b \le \Delta t_a$, due to constraint application during VCON. A graphical interpretation of the application of Eqs. (26) is shown in Fig. 5.

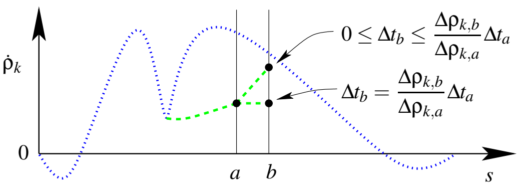


Fig. 5. Sweeping an acceleration curve for joint k

A similar line of reasoning can be used to show that, for negative $\Delta\rho_{k,b}$, Eq. (26b) yields the only positive real solution for Δt_b, within the permitted conditions.

The total linear Cartesian acceleration, Eq. (6b), is approximated by

$$\ddot{x} \approx \frac{\Delta x_b/\Delta t_b - \Delta x_a/\Delta t_a}{(\Delta t_a + \Delta t_b)/2}, \quad \ddot{y} \approx \frac{\Delta y_b/\Delta t_b - \Delta y_a/\Delta t_a}{(\Delta t_a + \Delta t_b)/2}$$
$$\ddot{z} \approx \frac{\Delta z_b/\Delta t_b - \Delta z_a/\Delta t_a}{(\Delta t_a + \Delta t_b)/2}, \quad \sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2} \leq \alpha_{\max}. \tag{30}$$

Equations (30) are combined to produce a quartic equation in Δt_b. The roots of this equation are found using Ferrari's solution, reported by Cardano in [20]. This method was found to be about twice as fast computationally as the ROOTS function in MATLAB. Again, the minimum, real, positive solution for Δt_b is chosen, which is guaranteed to exist because we know $\Delta t_b = \Delta t_a$ satisfies Eq. (30).

The cable tension constraints are formed similarly, by approximating Eq. (13) as

$$m_1\tau_1 + m_2\tau_2 + m_3\tau_3 = \frac{\Delta x_a/\Delta t_a - \Delta x_b/\Delta t_b}{(\Delta t_a + \Delta t_b)/2} \tag{31a}$$
$$m_4\tau_1 + m_5\tau_2 + m_6\tau_3 = \frac{\Delta y_a/\Delta t_a - \Delta y_b/\Delta t_b}{(\Delta t_a + \Delta t_b)/2} \tag{31b}$$
$$m_7\tau_1 + m_8\tau_2 + m_9\tau_3 = \frac{\Delta z_a/\Delta t_a - \Delta z_b/\Delta t_b}{(\Delta t_a + \Delta t_b)/2} + g. \tag{31c}$$

The additional intermediate variables

$$\begin{aligned} N_1 &= m_2\Delta y_b - m_5\Delta x_b, \quad & N_4 &= m_3\Delta y_a - m_6\Delta x_a \\ N_2 &= m_6\Delta x_b - m_3\Delta y_b, \quad & N_5 &= m_5\Delta x_a - m_2\Delta y_a \\ N_3 &= m_1\Delta y_b - m_4\Delta x_b, \quad & N_6 &= m_1\Delta y_a - m_4\Delta x_a \\ S_k &= gD_k - (-1)^k\tau_k R, \quad & k &= 1,2,3 \end{aligned} \tag{32}$$

are introduced; variables shown in Eq. (16) are still valid, except for N_1, N_2, and N_3.

Equations (31) are solved *analytically* six times: each time, the tension τ_k in one cable is set to $\tau_{\min}$ or $\tau_{\max}$, the unknowns being the tensions in the other two cables and Δt_b. This leads to six quadratic equations, two for each cable, defined by

$$A\Delta t_b^2 + B\Delta t_b + C \tag{33a}$$
$$1\begin{cases} A = \Delta t_a S_1 \\ B = \Delta t_a^2 S_1 + 2(\Delta z_a D_1 + m_8 N_4 + m_9 N_5) \\ C = 2\Delta t_a (m_9 N_2 + m_8 N_1 - \Delta z_b D_1) \end{cases} \tag{33b}$$
$$2\begin{cases} A = \Delta t_a S_2 \\ B = \Delta t_a^2 S_2 + 2(\Delta z_a D_2 + m_7 N_4 - m_9 N_6) \\ C = 2\Delta t_a (m_9 N_3 + m_7 N_1 - \Delta z_b D_2) \end{cases} \tag{33c}$$
$$3\begin{cases} A = \Delta t_a S_3 \\ B = \Delta t_a^2 S_3 + 2(\Delta z_a D_3 - m_7 N_5 - m_8 N_6) \\ C = 2\Delta t_a (m_8 N_3 - m_7 N_2 - \Delta z_b D_3). \end{cases} \tag{33d}$$

The two quadratic equations for each cable define a valid range for Δt_b, within which the tension constraints for that cable are satisfied. This is in contrast to VCON, where only a minimum value for Δt was found. A similar line of reasoning to that used for Eqs. (26) can be applied to Eqs. (33) to interpret the solutions to these equations.

Firstly, we can claim that $\Delta t_b = \Delta t_a$ is a solution that respects all tension constraints, because this equality represents a point below the s-$\dot{s}$ curve, and thus a point in between the f_k and the valid tension constraint for a particular point, as shown by the valid ranges for τ in Fig. 3.

Applying Eq. (20) to two adjacent points a and b, we have

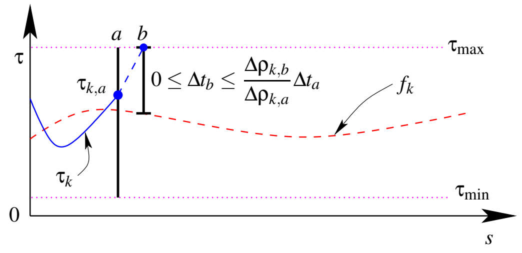
$$\frac{\Delta\rho_{k,b}/\Delta t_b - \Delta\rho_{k,a}/\Delta t_a}{(\Delta t_a + \Delta t_b)/2} = f_{k,ab} - \tau_{k,ab}, \tag{34}$$

where the subscript ab indicates the mean value of a property between points a and b. Eq. (34) is rearranged to produce the quadratic equation of Eq. (33a), with coefficients

$$\begin{aligned} A =& \Delta t_a (f_{k,ab} - \tau_{k,ab}), \\ B =& \Delta t_a^2 (f_{k,ab} - \tau_{k,ab}) + 2\Delta\rho_{k,a}, \quad C = -2\Delta\rho_{k,b}\Delta t_a. \end{aligned} \tag{35}$$

We see that Eq. (35) has the same form as Eq. (28). Therefore, a minimum restriction on Δt_b is always produced: for positive $\Delta\rho_{k,b}$, the minimum tension constraint will produce this restriction; for negative $\Delta\rho_{k,b}$, the maximum tension constraint will produce this restriction. The analysis of the four roots to the two quadratic equations in Δt_b is essentially the same as that for Eq. (28). The tension constraint application is shown in Fig. 6.

The remaining constraints to be imposed with ACON are the jerk constraints. A similar procedure to that used to develop the acceleration constraints would entail the consideration of three trajectory points, a, b, and c, leading to three- and six-degree polynomial equations for the cable jerk and the total linear jerk, respectively. Additionally, we would need to guarantee that two of the three points under consideration are constraint-respecting. This formulation is possible, but the jerk constraints rarely need to be strictly followed, so an approximate, faster method is called for.



τ
a
b
$\tau_{\max}$
$\tau_{k,a}$
$0 \leq \Delta t_b \leq \frac{\Delta\rho_{k,b}}{\Delta\rho_{k,a}}\Delta t_a$
f_k
τ_k
$\tau_{\min}$
0
s

Fig. 6. Tension constraints for joint k, imposed during ACON, for negative $\Delta\rho_{k,b}$

For each cable k, we can approximate the jerk equations as

$$\frac{1}{6}\dddot{\rho}_{k,\max}(\Delta t_b)^3 + \frac{1}{2}\ddot{\rho}_{k,a}(\Delta t_b)^2 + \dot{\rho}_{k,a}(\Delta t_b) - \Delta\rho_{k,b} = 0 \tag{36a}$$

$$-\frac{1}{6}\dddot{\rho}_{k,\max}(\Delta t_b)^3 + \frac{1}{2}\ddot{\rho}_{k,a}(\Delta t_b)^2 + \dot{\rho}_{k,a}(\Delta t_b) - \Delta\rho_{k,b} = 0 \tag{36b}$$

which are solved to find six values of Δt_b; the minimum positive real solution is again kept. The values of $\ddot{\rho}_{k,a}$ and $\dot{\rho}_{k,a}$ are updated following the selection of each point on the constraint-satisfying curve in s-$\dot{s}$. If the curve starts at a local minimum, $\ddot{\rho}_{k,a}$ is set to zero at this point. Otherwise, it is computed based on the adjacent segment in the opposite direction to the sweeping curve, since this point must be constraint-satisfying.

The Cartesian tangential jerk constraint is imposed according to

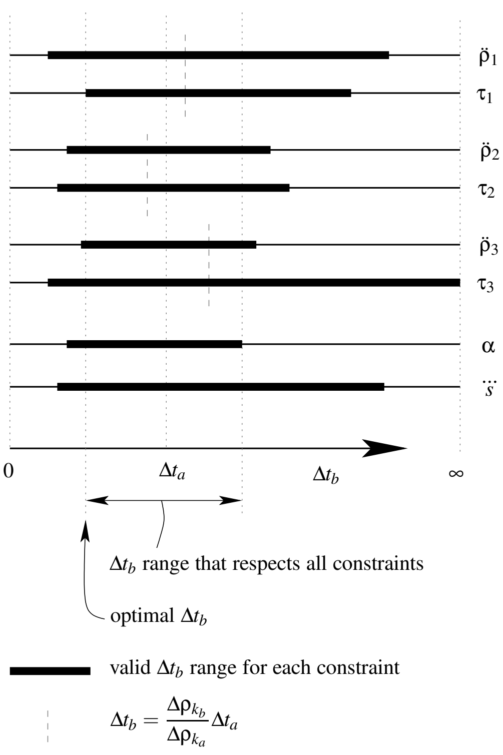
$$\frac{1}{6}\dddot{s}_{\max}(\Delta t_b)^3 + \frac{1}{2}\ddot{s}_a(\Delta t_b)^2 + \dot{s}_a(\Delta t_b) - \Delta s_b = 0. \tag{37}$$

The purpose of the jerk constraints presented here is to prevent abrupt changes in acceleration, which are characteristic of the bang-bang method and can cause many undesirable effects. Based on the results presented in the paper, MTTS is successful in this respect, though it is not recommended for problems where jerk constraints are dominant or when they need to be strictly adhered to.

When $\dddot{s}_{\max}$ is the only jerk constraint, the allowed Δt_b ranges for all constraints are guaranteed to overlap, as shown in Fig. 7. As explained in Subsec. 2.2, when the $\dddot{s}_{\max}$ constraint is imposed, only increases in speed $\dot{s}$ are allowed when computing Δt_b, so we know that $0 \leq \Delta t_b \leq \Delta t_a$, which must overlap with the acceleration constraints.

On the other hand, when joint jerk constraints $\dddot{\rho}_{k,\max}$ are included, decreases in speed could be called for, with $\Delta t_b > \Delta t_a$. Therefore, there is no guarantee that the allowed ranges for Δt_b will overlap, and constraint violations can occur. Such violations can be eliminated through iteration of ACON.

It is important to specify which constraints depend on the direction of travel along the path. The acceleration and jerk constraints are symmetric, because their application is identical, whether the path is traversed one way or the other, as shown in Fig. 4. Tension constraint application depends on the traversal direction, however. Therefore, again consulting Fig. 4, for Eqs. (31) to be valid, variables with a subscripts must always apply to the left path segment, and b variables to the right segment. As a result, when generating leftward acceleration curves—which will produce *deceleration* when the path is followed—the a and b quantities are interchanged for application of the tension constraint. Equations (31) are thus solved for Δt_a, which applies to the same segment as the Δt_b values found through application of the acceleration and jerk constraints.



$\ddot{\rho}_1$
τ_1
$\ddot{\rho}_2$
τ_2
$\ddot{\rho}_3$
τ_3
α
$\dddot{s}$
0
Δt_a
Δt_b
∞
Δt_b range that respects all constraints
optimal Δt_b
valid Δt_b range for each constraint
$\Delta t_b = \frac{\Delta\rho_{k_b}}{\Delta\rho_{k_a}}\Delta t_a$

Fig. 7. The application of constraints to determine Δt_b, during ACON

In summary, for all constraints at the acceleration level, including tension constraints, a minimum value for Δt_b is found. In fact, for each constrained quantity, a maximum value for Δt_b will also exist, though in some cases it will be infinite. We know that $\Delta t_b = \Delta t_a$ will satisfy all constraints at the acceleration level, so the allowed ranges for Δt_b for each constrained quantity must overlap. This is shown graphically in Fig. 7.

For each point of a constraint-satisfying acceleration curve, the highest value of Δt_b, computed using Eqs. (26), (30), (33), (36), and (37), is used to replace the corresponding point on the s-$\dot{s}$ curve, if it is larger. Then, point b replaces point a and the procedure is repeated for the next trajectory segment. These steps continue until the constraint-generated Δt_b is smaller than the corresponding point on s-$\dot{s}$. The green dashed line of Fig. 11 shows the s-$\dot{s}$ curve for the test path used in the case study, after ACON is applied.

2.4 Acceleration From and to Rest Curves (AREST)

Acceleration from and to rest should be considered separately, because the evenly spaced position data used up to this point have insufficient resolution to produce smooth curves at low speeds in the time domain. AREST is similar to ACON, except that it operates via even time-sampling, rather than even position-sampling. Therefore, AREST is the first step of MTTS that modifies path-point locations: evenly time-sampled points are added at the start and end of the trajectory.

Equations (30) and (33) still apply, except Δx_b, Δy_b, and Δz_b are now unknown and Δt_a and Δt_b are both equal to a chosen time-sampling rate. However, the relationship between x, y, and z is known everywhere on the path. To avoid singularities in the solution procedure and ensure numerical stability, the coordinate axis that exhibits the largest rate of change for the current path point is chosen as the independent variable, to be included in the vector of unknowns. For example, if $|\Delta y_b| > |\Delta x_b|$ and $|\Delta y_b| > |\Delta z_b|$, we can replace the vector $\begin{bmatrix}\Delta x_b & \Delta y_b & \Delta z_b\end{bmatrix}^T$ with $\Delta y_b \begin{bmatrix}C_1 & 1 & C_2\end{bmatrix}^T$, where C_1 and C_2 are known constants. The acceleration constraint then becomes

$$\frac{1}{(\Delta t)^2}\sqrt{\begin{array}{c}(C_1\Delta y_b - \Delta x_a)^2 + (\Delta y_b - \Delta y_a)^2 \\ + (C_2\Delta y_b - \Delta z_a)^2\end{array}} \leq \alpha^*_{\max} \qquad (38)$$

where $\alpha^*_{\max}$ is used to denote a *local* maximum acceleration, which must not increase by more than $\dddot{s}_{\max}\Delta t$ from one point to the next, to ensure that the jerk constraint is also satisfied. Cable acceleration and jerk constraints are simpler, and are imposed similarly, solving for $\Delta\rho_{k,b}$ in Eq. (26).

The cable tension equations become

$$m_1\tau_1 + m_2\tau_2 + m_3\tau_3 = \frac{\Delta x_a - C_1\Delta y_b}{(\Delta t)^2} \qquad (39a)$$

$$m_4\tau_1 + m_5\tau_2 + m_6\tau_3 = \frac{\Delta y_a - \Delta y_b}{(\Delta t)^2} \qquad (39b)$$

$$m_7\tau_1 + m_8\tau_2 + m_9\tau_3 = \frac{\Delta z_a - C_2\Delta y_b}{(\Delta t)^2} + g. \qquad (39c)$$

which are solved for Δy_b six times, setting one of the cable tensions to $\tau_{\min}$ or $\tau_{\max}$ each time, with the vector of unknowns being Δy_b and the tensions in the other two cables. The solution procedure is similar to that used for ACON.

Next, the minimum value of the magnitude of Δy_b found using Eqs. (38), (39), and any other constraint equation needed, is selected, and Δs_b is computed to generate the next point on the acceleration curve. The process iterates until the s-$\dot{s}$ curve is intersected.

AREST is applied at the start of the path, and at the end of the path in reverse. As was the case for ACON, the tension constraints depend on the direction traveled along the path but the acceleration and jerk constraints do not. After AREST completes, linear interpolation along the s-$\dot{s}$ curve is used to produce new positioning data, equally sampled in time. The magenta dash-dot line of Fig. 11 shows the s-$\dot{s}$ curve for the test path used in the case study, after AREST is applied.

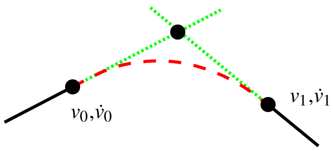


Fig. 8. The JCON subalgorithm

2.5 Elimination of Jerk Discontinuities (JCON)

As mentioned above, ACON cannot be applied to address speed-decrease jerk violations. Therefore, a different technique is used to eliminate this type of jerk: cubic interpolating polynomials at the velocity level are used to *distribute* jerk where the maximum jerk is exceeded; this is achieved by updating the $\mathbf{\Delta t}$ vector. In each path region where the jerk is to be smoothed, the *sign* of the tangential acceleration is not allowed to change. Equivalently, no inflection point is permitted in the polynomial curve. Without this restriction, uneven jerk distribution and unwanted speed variations could result. The JCON subalgorithm is illustrated in Fig. 8.

The locations of jerk violations are first identified along a path. Subsets of adjacent points are then formed among the identified points. Each subset is gradually expanded by incrementing the leftmost and rightmost points left and right, respectively, until the maximum jerk magnitude produced by the interpolating polynomial respects the jerk constraint. Leftward expansion of the interval is halted if a change in acceleration sign is encountered; rightward expansion is halted on the same condition, but separately. Once the interpolation interval Δt^* has been found, the speed over this interval is given by

$$\begin{aligned} &v = D\lambda^3 + C\lambda^2 + B\lambda + A, \qquad 0 \leq \lambda \leq 1 \qquad (40) \\ &A = v_0, \qquad C = 3v_1 - \dot{v}_1\Delta t^* - 3A - 2B, \\ &B = \dot{v}_0\Delta t^*, \qquad D = v_1 - A - B - C \end{aligned}$$

where v_0, v_1, $\dot{v}_0$, and $\dot{v}_1$ are the velocity and tangential acceleration at the start and end of the interval, and λ is normalized time. This procedure is approximate, since the change in acceleration is computed based on the point-to-point durations, and these durations will change after application of the polynomial interpolant. However, the durations can only increase, so the jerk estimate is slightly higher than the final jerk exhibited. Since the $\mathbf{\Delta t}$ vector is modified using this procedure, linear interpolation among $\mathbf{\Delta t}$ and $\mathbf{\Delta s}$ is again used to restore equal-time sampling. The solid red line of Fig. 11 shows the s-$\dot{s}$ curve for the test path used in the case study, after JCON is applied.

The primary JCON technique can fail to satisfy the negative jerk constraint in some situations. However, a secondary technique can be invoked to guarantee that the jerk

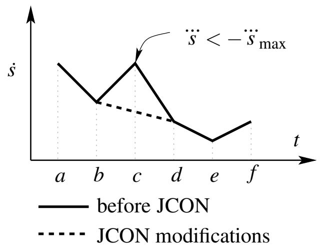


Fig. 9. The secondary technique applied during JCON

constraint is satisfied. Figure 9 shows an example where the cubic polynomial curve technique would fail. In this case, JCON would begin with the jerk violation detected at point c, and proceed leftward to point b and rightward to point e, both of which are points where the acceleration sign changes. The primary JCON method would then attempt to construct a concave-down curve from point c to point d. Such a curve, which maintains velocity and acceleration continuity, and satisfies the jerk constraint, is not possible in this case.

This failure is easily detected, at which point JCON would revert to the secondary method. Between points b and e, subranges of points where the jerk is negative are first identified, with one such range, b to d, being identified in this case. Linear interpolation in t-$\dot{s}$ is used across each range identified, as shown by the dashed line of Fig. 9.

Following the linear interpolation, the subrange b to e is verified for jerk violations. If any are detected, JCON first attempts to use the primary method; if this does not work, the secondary method is reapplied. This iterative process continues until the negative jerk constraint is respected everywhere in the range. This technique is guaranteed to eliminate all negative jerk violations because the secondary method is capable of removing *all* negative jerk.

In most cases, the primary JCON method, the cubic polynomial curve, will work and the secondary method is unnecessary. For example, the secondary method is never used for the test path shown in this paper.

As was the case for positive jerk in ACON, tangential jerk can be dealt with using JCON, as presented. However, the technique would need to be modified to include joint jerk constraints. The main problem that can occur here relates to path regions where $\dot{\rho}_k$ crosses zero. The application of the jerk constraint as presented would result in suboptimal path modifications in these path regions. In some cases, the jerk constraint could call for the sign of $\dot{\rho}_k$ to change at a particular path location, which is not allowed. Therefore, to use JCON for constraints on $\dddot{\rho}_k$, a subalgorithm would be needed to treat cases where modifications are called for in path regions where $\dot{\rho}_k = 0$.

Following JCON, data for all subpaths produced by PATHSPLIT are assembled and converted to the desired output format. For example, Cartesian-to-joint-space conversion can be performed, or feedrates can be easily computed, since positioning data are equally sampled in time. Reassembly is accomplished trivially via array concatenation, which is also possible because data are equally sampled in time. Deceleration to zero speed is thus automatically imposed at all abrupt changes in direction that were detected with PATHSPLIT.

Table 1. Spool attachment points for a 3-DOF cable-driven robot

Cable #	x (m)	y (m)	z (m)
1	5.0655	-1.9978	-0.0652
2	-5.1958	-2.3085	-0.0096
3	0.1302	4.3064	0.0747

3 Case Study: A Three-DOF Cable-Suspended Robot

The MTTS algorithm, adapted to include tension constraints, is applied to a test path in the workspace of a three-DOF spatial cable-suspended robot available in the Laboratoire de robotique at Université Laval. Analytic solutions to the cable tension equations, Eqs. (15), (31), and (39), are found using Maple 14. All other programming is performed with MATLAB r2011b.

The robot has an architecture similar to that shown in Fig. 1, with the spools forming a close-to-equilateral triangle, mounted about 6 m above the floor on the walls of a room 7 m high, 12 m long, and 10 m wide. A coordinate frame is placed at the centroid of the three attachment points, with the z-axis pointing downward. The spool attachment point positions in this frame are listed in Tab. 1.

The test path is the 3D curve shown in Fig. 10(a), 5.143 m long, with the y-coordinates produced according to

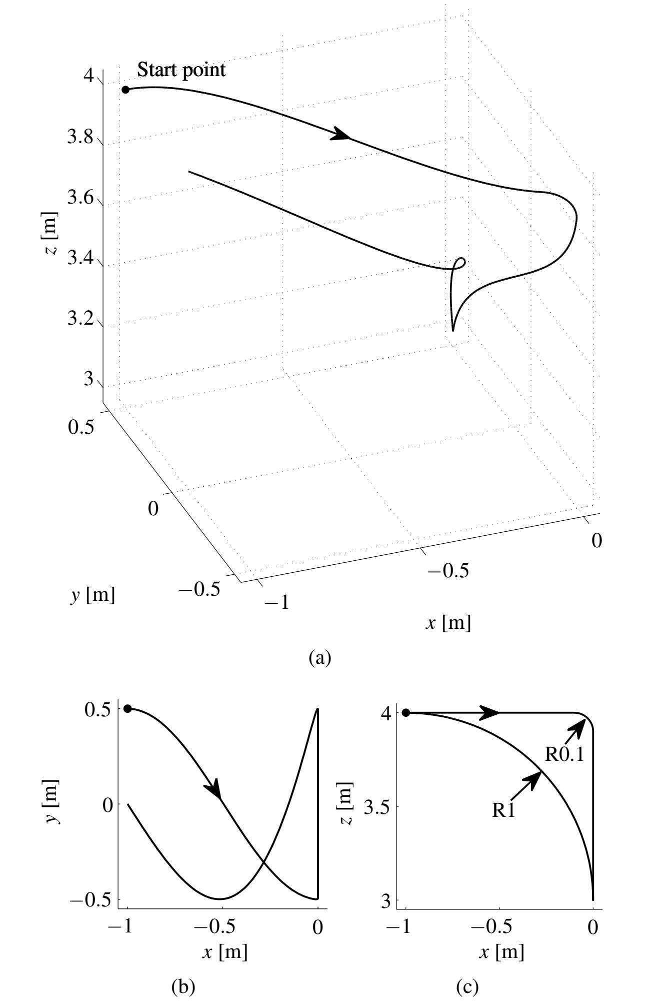
$$y = \frac{1}{2}\cos\left(\frac{7\pi\sigma}{2}\right), \qquad 0 \le \sigma \le 1 \tag{41}$$

where σ is the normalized arc length for the curve in Fig. 10(c). The path is supplied to MTTS as arrays of x, y and z data, with the constraints

$$\begin{aligned} v_{\max} &= 2 \text{ m/s} & \alpha_{\max} &= 4 \text{ m/s}^2 & \dddot{s}_{\max} &= 50 \text{ m/s}^3 \\ \tau_{\min} &= 2 \text{ N/kg} & \tau_{\max} &= 8 \text{ N/kg} & \beta_{\max} &= \pi/6 \end{aligned} \tag{42}$$

where $\beta_{\max}$ is the maximum allowed angle between two adjacent line segments before path splitting occurs, as defined in Fig. 2 and Eq. 3. The cable tension constraints are equivalent to an allowed tension range of 2–8 N, for each cable, with an end-effector of mass 1 kg.

Path data are first resampled into 1 mm-long segments, using linear interpolation. One abrupt change in direction is then detected using the PATHSPLIT subalgorithm; VCON, ACON, AREST, and JCON are applied *separately* to each subpath, as described Subsec. 2.1. Figure 11 shows the s-$\dot{s}$ curve following each step of MTTS, for *both* subtrajectories. Output data are arrays of x, y and z coordinates, equally sampled in time, in this case with a resolution of 0.016 s. Figure 12 shows the 5.104 s trajectory produced using the constraints of Eq. 42.



(a)

(b)

(c)

Fig. 10. Test path: (a) 3D plot; (b) Projection on the xy-plane; (c) Projection on the xz-plane.

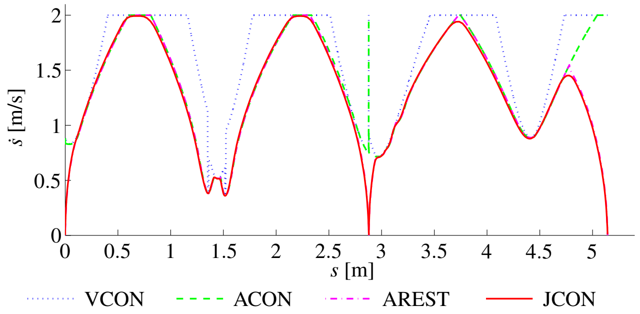


Fig. 11. Steps of the MTTS algorithm, for the path shown in Fig. 10, with all constraints listed in Eq. (42) active.

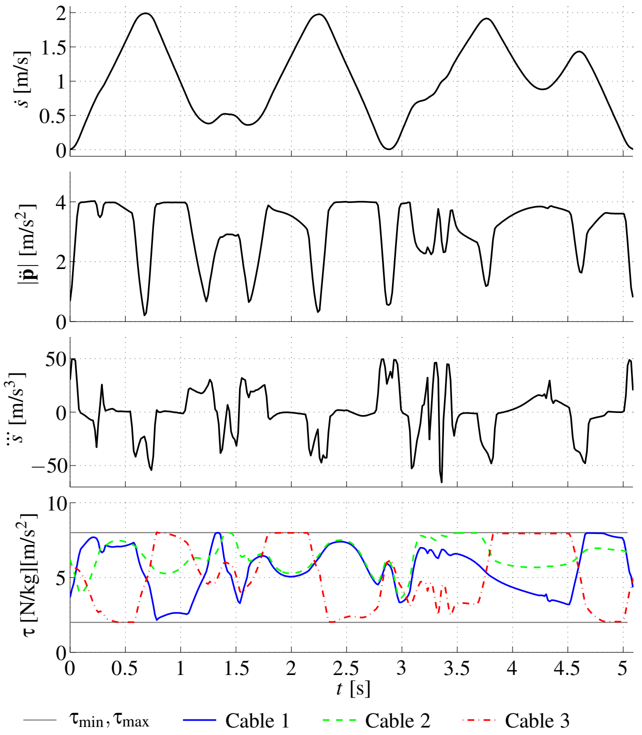


Fig. 12. Trajectory parameters for the path shown in Fig. 10, using MTTS with all constraints listed in Eq. (42) active.

To establish the optimality and highlight the advantages of MTTS, the bang-bang method is also used to solve the TOTP problem for the path shown in Fig. 10. The set of constraints is reduced by eliminating $\alpha_{\max}$ and $\dddot{s}_{\max}$, because the bang-bang technique cannot impose these constraints. This leaves the tension and linear velocity constraints.

A brief overview of the application of the bang-bang method to a 3-DOF cable-suspended positioning mechanism is provided here. More detail on this method, including its application to a different cable-driven robot, can be found in [1, 18].

The central idea to the bang-bang technique is that each constraint equation can be expressed as a function of the distance traveled along the path s and its derivatives. For our 3-DOF cable-suspended mechanism, we have the inequality relations

$$\tau_{k,\min} \leq c_{1k}(s)\ddot{s} + c_{2k}(s,\dot{s}) \leq \tau_{k,\max}. \tag{43}$$

To determine the functions c_{1k} and c_{2k}, we express x, y, and z parametrically as functions of s in Eq. 13 to obtain

$$m_1\tau_1 + m_2\tau_2 + m_3\tau_3 = -\frac{d^2x}{ds^2}\dot{s}^2 - \frac{dx}{ds}\ddot{s} \tag{44a}$$

$$m_4\tau_1 + m_5\tau_2 + m_6\tau_3 = -\frac{d^2y}{ds^2}\dot{s}^2 - \frac{dy}{ds}\ddot{s} \tag{44b}$$

$$m_7\tau_1 + m_8\tau_2 + m_9\tau_3 = -\frac{d^2z}{ds^2}\dot{s}^2 - \frac{dz}{ds}\ddot{s} - g \tag{44c}$$

An analytical solution to Eq. (44) for τ can be found without numerically specifying the entries of $\mathbf{M}$. The tension in cable k can then be represented by

$$\tau_k = C_{1k}\ddot{s} + C_{2k}\dot{s}^2 + C_{3k}g. \tag{45}$$

where the constants C_{1k}, C_{2k}, and C_{3k} depend only on the entries of $\mathbf{M}$ and the first- and second-order derivatives of

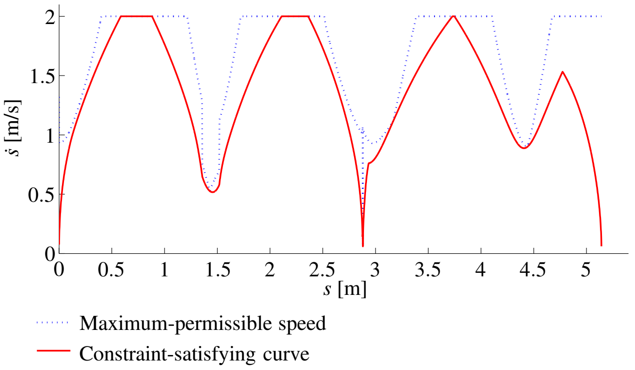


Fig. 13. The bang-bang method, for the path shown in Fig. 10, with no constraints on jerk or acceleration.

x, y, and z with respect to s, shown in Eq. (44), which are known at every point along the path.

Combining Eqs. (43) and (45) yields

$$\tau_{k,\min} \le C_{1k}\ddot{s} + C_{2k}\dot{s}^2 + C_{3k}g \le \tau_{k,\max}, \tag{46}$$

which can be expressed in the form

$$f_k(s,\dot{s}) \le \ddot{s} \le g_k(s,\dot{s}) \tag{47}$$

where

$$f_k(s,\dot{s}) = \begin{cases} \dfrac{\tau_{k,\min} - C_{2k}\dot{s}^2 - C_{3k}g}{C_{1k}}, & C_{1k} > 0 \\ \dfrac{\tau_{k,\max} - C_{2k}\dot{s}^2 - C_{3k}g}{C_{1k}}, & C_{1k} < 0 \end{cases} \tag{48a}$$

$$g_k(s,\dot{s}) = \begin{cases} \dfrac{\tau_{k,\max} - C_{2k}\dot{s}^2 - C_{3k}g}{C_{1k}}, & C_{1k} > 0 \\ \dfrac{\tau_{k,\min} - C_{2k}\dot{s}^2 - C_{3k}g}{C_{1k}}, & C_{1k} < 0 \end{cases}. \tag{48b}$$

The maximum permissible speed at each path point can be found by setting

$$f_k(s,\dot{s}) = g_l(s,\dot{s}) \tag{49}$$

for cables k and l, with $k \neq l$. For each path point, six equations are found and solved, producing maximum allowed values for $\dot{s}$. The lowest allowed $\dot{s}$ for each path point then becomes a point on the maximum-permissible speed curve. Applying this procedure to the path shown in Fig. 10 produces the s-$\dot{s}$ curve shown in Fig. 13.

With the initial s-$\dot{s}$ curve identified, the next step is to identify switching points along this curve, from which acceleration curves will be swept by integrating $\ddot{s}$, while respecting the six constraint relations. In [1], the identification of these switching points is described as the most difficult step of the bang-bang method, and the difficulty increases with the number of switching points needed. There, a trial-and-error approach is proposed, with alternating acceleration and deceleration curves being produced when moving from left to right in s-$\dot{s}$ space. Here, we use an alternate approach, similar to that described in ACON, where acceleration curves are swept rightward *and* leftward from local minima in s-$\dot{s}$, producing the constraint-satisfying curve of Fig. 13. Kinematic parameters and cable tensions for this curve are plotted in Fig. 14. We can see that a constraint violation occurs at one location along the path, which corresponds to the abrupt change in direction.

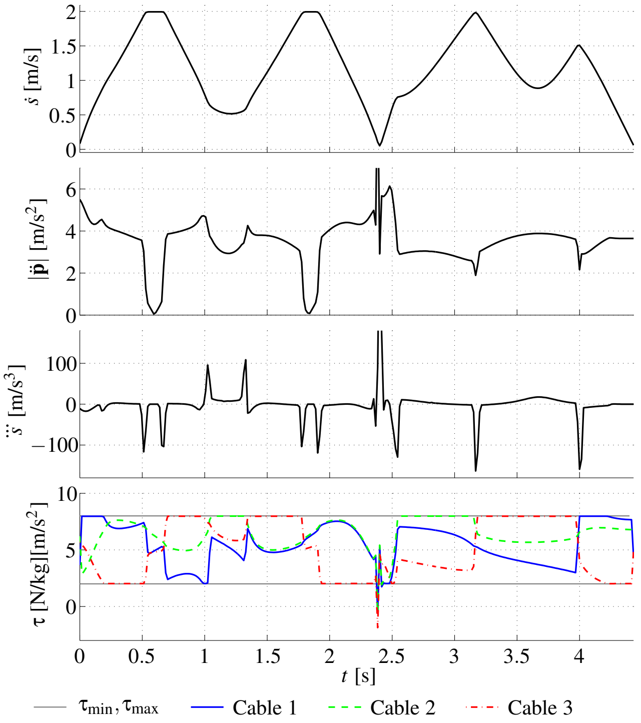
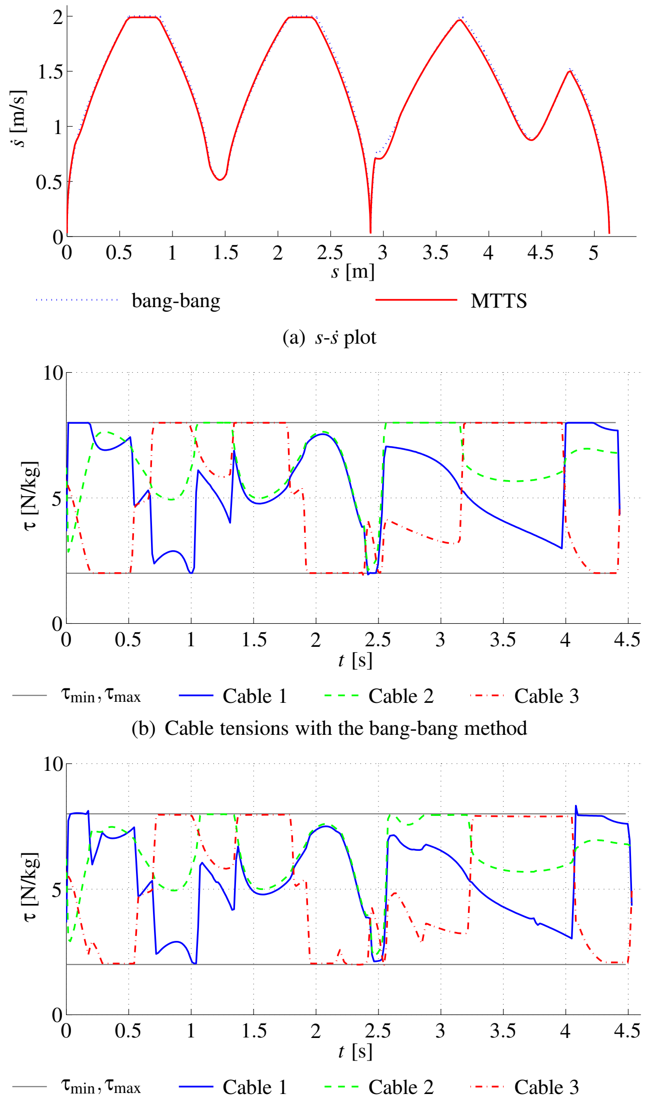


Fig. 14. Trajectory parameters for the path shown in Fig. 10, using the bang-bang method, with no jerk or acceleration constraints.

To obtain a trajectory for direct comparison to the bang-bang method, MTTS is applied to the test path a second time, neglecting jerk and total acceleration constraints. Additionally, the bang-bang method is modified by using PATHSPLIT to prevent the constraint violation seen in Fig. 14: the test path is split into two subpaths, which are submitted *separately* to the bang-bang algorithm, after which the trajectory is reassembled, in much the same way that path splitting works for MTTS.

These modifications are applied to produce the trajectories shown in Fig. 15, where a direct comparison between the two techniques can be made. We can see that the two trajectories are quite similar, with the times of travel being 4.432 s for the bang-bang technique, and 4.544 s for MTTS. Therefore, it is reasonable to claim that MTTS produces trajectories that are nearly time-optimal. The discrepancy between the two techniques is likely caused by the discretization and interpolation used in MTTS. For example, MTTS requires a discrete number of path points at all stages. During certain steps, the trajectory is re-sampled, and the number of points



(a) s-$\dot{s}$ plot

(b) Cable tensions with the bang-bang method

(c) Cable tensions with MTTS

Fig. 15. The bang-bang method compared to MTTS, for the path shown in Fig. 10, with no jerk or acceleration constraints.

is always rounded up, to prevent new constraint violations from occurring. This effect can be seen in Fig. 15(a), where the MTTS curve is nearly always slightly below the bang-bang curve.

One might expect the acceleration shown in Figs. 12 and 14 to be *negative* when the speed is decreasing. However, a decrease in speed only contributes to negative *tangential* acceleration $\ddot{s}$, whereas the *total* acceleration α is always positive.

The effect of the jerk constraint imposed by MTTS is visible in Figs. 11 and 12, where we see that this constraint is active along several regions of the path. It is interesting to note the geometric significance of the double local minimum in $\dot{s}$, visible in Fig. 11 near $s = 1.5$. This feature is caused by the jerk constraint, as the end effector passes from a straight-line path in projected xz-space, to the 0.1 m round, and back to a straight-line path in projected xz-space, as seen in Fig. 10(c). At each of these two transitions, there is a curvature discontinuity on the path, leading to an abrupt change in acceleration, whose magnitude is limited by imposing the tangential jerk constraint.

Another important factor to consider when comparing the two methods is computational time. Required computational times were 2.4 s and 2.3 s, for the bang-bang method and MTTS, respectively, for the trajectories shown in Fig. 15, using an Intel Core i7-2720QM processor (2.2 GHz, 3.3 GHz turbo). Therefore, the computational burden for the two techniques is about the same. For MTTS, acceleration and jerk constraints were rendered inactive by simply raising them, with all constraint-verification computations still taking place. Indeed, when the acceleration and jerk constraints of Eq. (42) are also included in MTTS, producing the trajectory of Fig. 12, the computational time is still 2.3 s. The most computationally intensive constraints for MTTS are those on cable tension, because analytic solutions to 3 × 3 systems of equations are repeatedly evaluated. The tension constraints *can* be turned off in MTTS, and when this is done, the computational time reduces to 1.0 s for the test path.

Each technique should be applied to a large data set of paths to further investigate the computational cost and establish robustness. Both techniques are readily adaptable to parallel computation, where parallelization could be applied by submitting different paths, or even different subpaths produced using PATHSPLIT, to different processors.

For this case study, the output data are sampled at 0.016 s, though other formats and sampling rates can easily be produced. For example, for the cable-driven robot being studied, in some cases it is preferable to provide cable-length data sampled at a higher rate. These data can be readily produced by solving the inverse displacement problem and resampling using cubic spline interpolation. Linear interpolation should never be used when resampling at a higher rate, since large oscillations in acceleration, jerk, and tension would be produced. At this stage, cubic spline interpolation in the time domain will not cause significant deviation from the path, because points are densely spaced around abrupt changes in direction.

4 Discussion

Among the techniques discussed in the Introduction, only MTTS and the bang-bang technique, proposed in [18], are capable of imposing a tension constraint for a cable-driven mechanism, for a path of *arbitrary* geometry. MTTS was shown to produce a close-to-optimal trajectory when compared to the this method, subject to the same constraints. A major advantage of MTTS is the ability to handle path data with abrupt changes in direction. However, we established that the bang-bang method can also handle these data, if the PATHSPLIT subalgorithm is used in a pre-treatment step.

For many applications, a major deficiency of the bang-bang technique is the absence of jerk control, which can lead to vibrations and high following error, particularly in path regions of high curvature. Jerk control, such as that implemented with MTTS, dampens these vibrations and results in a lower following error, because a trajectory that requires accelerations to change smoothly is typically much easier to track.

The original formulation of the bang-bang technique, introduced in [1], cannot handle total acceleration constraints or jerk constraints. If we cast the total acceleration, defined in Eq. (6), using the bang-bang form of Eq. (43), we obtain

$$C_1\dot{s}^4 + C_2\dot{s}^2\ddot{s} + C_3\ddot{s}^2 \leq \alpha_{\max}^2 \tag{50}$$

where C_1, C_2, and C_3 can be computed at each path point, according to the first and second order derivatives of x, y, and z with respect to s. For the cable k jerk, we have

$$\frac{d^3\rho_k}{ds^3}\dot{s}^3 + 3\frac{d^2\rho_k}{ds^2}\dot{s}\ddot{s} + \frac{d\rho_k}{ds}\dddot{s} \leq \dddot{\rho}_{k,\max}. \tag{51}$$

These equations cannot be solved using the bang-bang method, which relies on isolating $\ddot{s}$, and then solving for $\dot{s}$ in the constraint equations. However, Dong et al. introduced a modified, kinematic version of the bang-bang algorithm in [12], which does allow for the inclusion of jerk constraints, as described in Sec. 1. Their technique requires the solution of a series of nonlinear single-variable optimization problems, which is claimed to be computationally efficient, though the computational time is not listed for the test path used. Additionally, it is unclear if the technique can be applied to systems with *coupled* constraint equations; the experimental setup used to demonstrate the method was a 2-DOF system consisting of two prismatic joints positioning a stage. Coupled constraint equations would greatly increase the complexity of the optimization problems to be solved.

Both MTTS and the regular bang-bang technique do not rely on solving optimization problems, but instead use explicit, computationally-efficient steps. They are also both applicable to systems with coupled axes. The 3-DOF cable-suspended system studied in this paper is such a system, as the solution to a 3×3 equation must be found each time the tension constraint is applied.

MTTS is also more suitable for complex paths, which contain highly variable curvature, many abrupt changes in direction, and as a result, many changes in acceleration sign. Such paths would require the identification of many, in some cases hundreds, of switching points with the bang-bang-method. This would likely render the computational burden of the method prohibitive, especially when we consider that in [18], it was reported that the standard pick-and-place path required the identification of five switching points, and the computational time for producing the time-optimal trajectory was nearly four minutes.

5 Conclusions

In this paper, minimum time trajectory shaping (MTTS), a time-optimal trajectory planning technique, was adapted for use with three-DOF spatial cable-driven robots. MTTS admits velocity, acceleration, jerk, and dynamic constraints that can be expressed as minimum and maximum cable tensions. The tension constraint equations are imposed using analytic solutions to 3×3 non-linear systems. The trajectory-shaping steps of MTTS are all simple and explicit, making the algorithm both fast and robust. MTTS was applied to a test path for large-scale, three-DOF cable-suspended robot, and the associated performance was compared to that of the bang-bang method.

In the future, an experimental validation of the technique is planned. Additionally, the technique will be modified to use numerical solutions to the non-linear tension-constraint equations, which will permit the generalization of the technique to higher-DOF cable-driven robots.

6 Acknowledgment

The authors gratefully acknowledge the support of The Social Sciences and Humanities Research Council of Canada (SSHRC), under Strategic Research Grant 890-2011-0039, led by Prof. Aaron Sprecher, McGill University. This work was also supported by the Le Fonds de Recherche du Québec –Nature et Technologies (FRQ-NT), in the form of a post-doctoral fellowship for E. Barnett.

References

[1] Bobrow, J. E., Dubowsky, S., and Gibson, J. S., 1985. “Time-optimal control of robotic manipulators along specified paths”. *Int. J. Robot. Res.,* **4**(3), pp. 3–17.

[2] Shin, K. G., and McKay, N. D., 1986. “A dynamic programming approach to trajectory planning of robotic manipulators”. *IEEE Trans. Autom. Control,* **31**(6), Jun., pp. 491–500.

[3] Pfeiffer, F., and Johanni, R., 1987. “A concept for manipulator trajectory planning”. *IEEE J. Robot. Autom.,* **RA-3**(2), Apr., pp. 115–123.

[4] Obradović, A., Vuković, J., Mladenović, N., and Mitrović, Z., 2011. “Time optimal motions of mechanical system with a prescribed trajectory”. *Meccanica,* **46**, pp. 803–816.

[5] Gasparetto, A., and Zanotto, V., 2008. “A technique for time-jerk optimal planning of robot trajectories”. *Robot. Comp. Integ. Manuf.,* **24**, pp. 415–426.

[6] Macfarlane, S., and Croft, E. A., 2003. “Jerk-bounded manipulator trajectory planning: Design for real-time applications”. *IEEE Trans. Robot. Autom.,* **19**(1), pp. 42–52.

[7] Wang, C. H., and Horng, J. G., 1990. “Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-spline functions”. *IEEE Trans. Autom. Control,* **35**(5), pp. 573–577.

[8] Wu, Y. J. E., and Beaman, J. J., 1990. “Contour following for scanning control in SFF application: Control trajectory planning”. In Proc. Solid Freeform Fab. Symp., pp. 126–134.

[9] Fadel, G. M., and Ganti, R., 1998. “Parametric based controller for rapid prototyping applications”. In Proc. Solid Freeform Fab. Symp., pp. 236–243.

[10] Timar, S. D., and Farouki, R. T., 2007. "Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds". *Robot. Comp. Int. Manuf.,* **23**, pp. 563–579.

[11] Zhang, K., Yuan, C. H., Gao, X. S., and Li, H., 2012. "A greedy algorithm for feedrate planning of CNC machines along curved tool paths with confined jerk". *Robot. Comp. Int. Manuf.,* **28**, pp. 472–483.

[12] Dong, J., Ferreira, P. M., and Stori, J. A., 2007. "Feed-rate optimization with jerk constraints for generating minimum-time trajectories". *Int. J. Mach. Tools Manuf.,* **47**, pp. 1941–1955.

[13] Gosselin, C., 2012. "Global planning of dynamically feasible trajectories for three-dof spatial cable-suspended parallel robots". In Proc. First Int. Conf. on Cable-Driven Parallel Robots, pp. 3–22. (Springer Mechanisms and Machine Science, T. Bruckmann and A. Pott, eds., Vol. 12, Springer-Verlag, Berlin).

[14] Gosselin, C., Ren, P., and Foucault, S., 2012. "Dynamic trajectory planning of a two-dof cable-suspended parallel robot". In Proc. IEEE Int. Conf. Robot. Autom., pp. 1476–1481.

[15] Oh, S. R., and Agrawal, S. K., 2003. "Cable-suspended planar parallel robots with redundant cables: controllers with positive cable tensions". In Proc. IEEE Int. Conf. Robot. Autom., pp. 3023–3028.

[16] Trevisani, A., 2010. "Underconstrained planar cable-direct-driven robots: a trajectory planning method ensuring positive and bounded cable tensions". *Mechatronics,* **20**, pp. 113–127.

[17] Trevisani, A., 2012. "Experimental validation of a trajectory planning approach avoiding cable slackness and excessive tension in underconstrained translational planar cable-driven robots". In Proc. First Int. Conf. on Cable-Driven Parallel Robots, pp. 23–39. (Springer Mechanisms and Machine Science, T. Bruckmann and A. Pott, eds., Vol. 12, Springer-Verlag, Berlin).

[18] Behzadipour, S., and Khajepour, A., 2006. "Time-optimal trajectory planning in cable-based manipulators". *IEEE Trans. Robot.,* **22**(3), pp. 559–563.

[19] Barnett, E., 2012. "The design of an integrated system for rapid prototyping with ice". PhD Thesis, McGill, Montreal, QC, Canada.

[20] Cardano, G., 1545. *Ars magna (The Rules of Algebra).* Dover, Mineola, NY.