Time-Optimal Parabolic Interpolation with Velocity, Acceleration, and Minimum-Switch-Time Constraints

Puttichai Lertkultanon and Quang-Cuong Pham
School of Mechanical and Aerospace Engineering
Nanyang Technological University
Singapore

***Abstract*—Time-optimal trajectories with bounded velocities and accelerations are known to be parabolic, *i.e.*, piecewise constant in acceleration. An important characteristic of this class of trajectories is the distribution of the switch points, *i.e.*, the time instants when the acceleration of any robot joint changes. When integrating parabolic trajectory generation into a motion planning pipeline, especially the one which involves a shortcutting procedure, the resulting trajectory usually contains a large number of switch points with a dense distribution. This high frequency acceleration switching intensifies joint motor wear as well as hamper the robot performance. In this paper we propose an algorithm for planning parabolic trajectories subject to both physical bounds, *i.e.*, joint velocity and acceleration limits, and the minimum-switch-time constraint. The latter constraint ensures that the time duration between any two consecutive switch points is always greater than a given minimum value. Analytic derivations are given as well as comparison with other methods.**

***Note to Practitioners*—A large number of industrial robots accept piecewise constant acceleration trajectories as inputs. Currently available procedures for planning such trajectories usually cause trajectories to have many switch points, the time instants when a robot joint changes its acceleration, concentrated in short time intervals. Switch points that are too close to each other might hamper the robot performance, in terms of execution time or tracking accuracy. To address this problem we develop an algorithm which plans piecewise constant acceleration trajectories by taking into account not only joint velocity and acceleration bounds but also the *minimum-switch-time* constraint. The latter constraint ensures that the time duration between any two consecutive switch points is always greater than a given minimum value. Although the constraint might result in longer trajectory durations, it significantly reduces the number of switch points. Comparisons with other methods show that our method produces higher quality trajectories. We also provide an open-source implementation in Python.**

***Index Terms*—parabolic trajectories, shortcutting, switch time.**

I. Introduction

As the execution time of robot movements is a determining factor in industrial productivity, planning fast robot trajectories is an important topic in industrial robotics. A large body of work has been devoted to the planning of time-optimal trajectories for robot manipulators, subject to various types of constraints, such as torque bounds [1], [2], gripper and payload constraint [3], or velocity and acceleration bounds [4], [5], [6].

In trajectory generation for industrial robots two families of trajectories, namely splines and polynomials, are often used. B-splines were used in [7] while fifth-degree polynomials were used in [8] for trajectory generation. However, in both cases the authors only considered the case when terminal velocities were zero. [9] considered polynomial-like trajectory with arbitrary terminal velocities. Although it was claimed to be computationally light, no physical bounds such as velocity limits were taken into account. [10] and [11] extended the framework to handle others classes of trajectories. However, neither of the above mentioned work addressed the time-optimality issue in the trajectory classes used.

While many trajectory classes can be utilized, a significant proportion of industrial robots are controlled in acceleration, *i.e.*, their inputs are time-series of the joint accelerations, resulting in piecewise parabolic (or second-order) trajectories. Hauser and Ng-Thow-Hing [5] and Kröger *et al.* [12] investigated the problem of planning time-optimal piecewise parabolic trajectories subject to velocity and acceleration bounds (note that in this case the parabolicity of the optimal trajectories is *a posteriori*). However, the trajectories computed by these authors tend to have a large number of *switch points*, *i.e.*, time instants when the acceleration of any joint changes. Furthermore, these switch points are sometimes concentrated in a short time interval, see Fig. 1, especially after a shortcutting procedure [5], [13] has been applied.

Yet, in many applications, switch points that are too close to each other might hamper the performance of the system. For instance, in applications where human operators must implement the planned trajectory, time durations between two switches cannot be smaller than the cognitive processing time of the operator, which can be of the order of seconds.

One possibility to take into account the minimum-switch-time constraint could be to include this constraint, along with velocity and acceleration bounds, into an *optimization program*. However, this program is non-convex, and therefore cannot be efficiently solved (see Section V-A).

In this paper we propose an algorithm for planning piecewise parabolic trajectories that takes into account the minimum-switch-time constraint in an efficient way. We first start with the algorithm of [5], [12], which computes time-optimal interpolations subject to velocity and acceleration bounds (recalled in Section II). Then we examine all the possibilities of violation of the minimum-switch-time constraint, and propose a solution to address the violation for each case (Sections III and IV). A proof of time-optimality is provided, as well as an open-source implementation (see `https://copy.com/jQtScjfYJ6voQsFT`). The new interpolation routine is integrated into a full-fledged

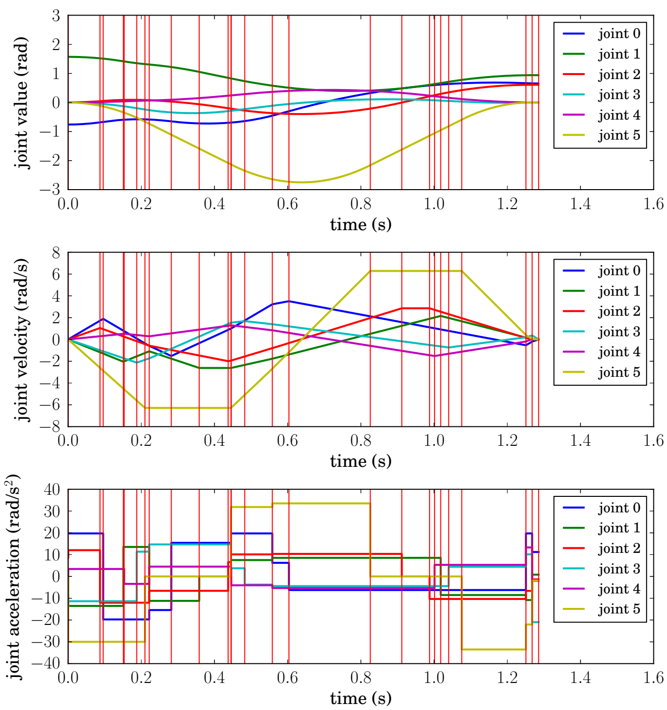

Fig. 1. An example trajectory planned and shortcut using the algorithm in [5]. Red vertical lines indicate time instants when at least one DOF switches its acceleration (switch points). The minimum-switch-time of this trajectory is $0.49\,\mathrm{ms}$, which is attained near $t = 0.1\,\mathrm{s}$.

motion planner and its efficiency is demonstrated through simulations (Section V).

II. Background: time-optimal parabolic interpolation without minimum-switch-time constraint

In the sequel, bold lower-case letters will denote vectors, normal lower-case letters will denote scalars, and bold upper-case letters will denote matrices.

A. Motion planning pipeline

Before recalling the time-optimal parabolic interpolation algorithm of [5], [12], this section presents the global motion planning pipeline that is based on this algorithm. Although there exists a large number of motion planning methods, this *plan-and-shortcut* pipeline is one of the most robust [14] and widely used in industrial robotics; it is the default pipeline in the robot programming environment OpenRAVE [15], which in turn is used by a large number of research groups and robotics companies worldwide.

A trajectory planning problem consists in finding a fast trajectory connecting two robot configurations $\boldsymbol{x}_{\text{init}}$ and $\boldsymbol{x}_{\text{goal}}$ in the robot joint space $\mathcal{C} \subset \mathbf{R}^n$, where n is the number of degrees of freedom (DOFs) of the robot, subject to velocity and acceleration bounds of each joint, as well as collision avoidance. This pipeline is based on the path-velocity decomposition approach [16] which decompose the problem into planning a *path* and a *velocity profile*. In the first stage a path planner such as a Rapidly-Exploring Random Tree (RRT) planner [17] is used to search for a collision-free *path* connecting $\boldsymbol{x}_{\text{init}}$ and $\boldsymbol{x}_{\text{goal}}$. The solution path is piecewise linear since it is formed by concatenating a linear path segment together. Each linear path segment P connecting $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ can be represented by a parameterization $\boldsymbol{x}(s) = \boldsymbol{x}_a + s(\boldsymbol{x}_b - \boldsymbol{x}_a)$, where $s \in [0, 1]$.

The second stage, called time-parameterization, assigns to a path a *velocity profile*, which is the time-derivative of the path parameterization function $s : [0, T] \to [0, 1]$, where T is the total duration of the velocity profile and thus of the trajectory. The problem can be alternatively viewed as a velocity profile interpolation problem. To respect the path geometry (avoiding thereby new collisions at this stage), we must time-parameterize paths of different joints *simultaneously* through the path parameterization function s. Furthermore, to avoid discontinuities in the velocity vector at the junctions of the linear segments, we need to ensure that the velocities at the beginning and the end of each segment are zero. In the end we will have a trajectory $\boldsymbol{x} : [0, T] \to \mathcal{C}$, where $\boldsymbol{x}(0) = \boldsymbol{x}_{\text{init}}$ and $\boldsymbol{x}(T) = \boldsymbol{x}_{\text{goal}}$.

In the third stage, a randomized shortcutting procedure is applied to improve the execution time of the trajectory, which is initially high because of the nature of a randomized path planner[1] and of the start-stop behavior at the junctions between linear segments. In each shortcutting iteration two random time instants t_a and t_b, are selected. Then a time-optimal trajectory is interpolated between $(\boldsymbol{x}(t_a), \dot{\boldsymbol{x}}(t_a))$ and $(\boldsymbol{x}(t_b), \dot{\boldsymbol{x}}(t_b))$. To effectively *shortcut* the trajectory, different DOFs may be interpolated *independently*, provided that all the interpolants have the same duration. If the shortcut has shorter duration than $t_b - t_a$ and is collision-free, then we replace the original trajectory segment by the shortcut.

In the aforementioned procedure two interpolation primitives are required: (i) simultaneous interpolation with zero terminal velocities (in the planning stage) (ii) independent interpolation with arbitrary terminal velocities subject to the condition that the interpolants have the same duration (in the shortcutting stage). The following sections recall the algorithms proposed in [5] for each case.

B. Simultaneous interpolation with zero terminal velocities

Definition 1. *A ramp is a constant-acceleration velocity profile. A ramp with non-zero acceleration is called a parabolic ramp or P-ramp. A ramp with zero acceleration is called a linear ramp or L-ramp.* □

Definition 2. *A P-ramp with maximal (respectively minimal) acceleration is noted P^+ (respectively P^-). An L-ramp with maximal (respectively minimal) velocity is noted L^+ (respectively L^-).* □

Consider a linear path segment P connecting $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ which is parameterized by a path parameterization function $s : [0, T] \to [0, 1]$. A valid velocity profile must be subject to boundary conditions $\boldsymbol{v}(0) = \boldsymbol{v}(T) = \mathbf{0}$, where $\boldsymbol{v}(t) = \dot{\boldsymbol{x}}(t)$. Let $\boldsymbol{v}_m$ and $\boldsymbol{a}_m$ be the vectors of velocity and acceleration

[1]Some planners, such as RRT* [18] optimize path quality during the planning itself, as opposed to the post-processing approach advocated here. However, such planners have large execution overheads and are seldom used in practical industrial settings.

bounds respectively. The velocity and acceleration bounds for the function s are then given by $\dot{s}_m \stackrel{\text{def}}{=} \min_i(\boldsymbol{v}_{m,i}/|\boldsymbol{x}_{b,i} - \boldsymbol{x}_{a,i}|)$ and $\ddot{s}_m \stackrel{\text{def}}{=} \min_i(\boldsymbol{a}_{m,i}/|\boldsymbol{x}_{b,i} - \boldsymbol{x}_{a,i}|)$, where the subscript i denotes the i^{th} component of the vector.

The n-DOF path-parameterization problem of the segment has been transformed to a single-DOF problem in the variable s subject to the velocity bound $\dot{s}_m$ and the acceleration bound $\ddot{s}_m$. The time-optimal velocity profile can be shown to be either $\mathrm{P}^+\mathrm{P}^-$ or $\mathrm{P}^+\mathrm{L}^+\mathrm{P}^-$. In the former case, the switch point is given by $t_s = \sqrt{1/\ddot{s}_m}$. In the latter case, the first switch point is given by $t_0 = \dot{s}_m/\ddot{s}_m$ and the switch time between the two switch points is $t_1 = 1/\dot{s}_m - 1/\ddot{s}_m$.

C. Independent interpolation with arbitrary terminal velocities

1) Single DOF: Given a straight-line path from x_0 to x_1, the time-optimal velocity profile with the initial and final velocities v_0 and v_1, subject to velocity and acceleration bounds v_m and a_m can be shown to have only four possible types: $\mathrm{P}^+\mathrm{P}^-$, $\mathrm{P}^-\mathrm{P}^+$, $\mathrm{P}^+\mathrm{L}^+\mathrm{P}^-$, and $\mathrm{P}^-\mathrm{L}^-\mathrm{P}^+$. For given boundary conditions and bounds we can compute (or *interpolate*) the time-optimal velocity profile by exploring each of the four cases above. The detailed calculations of the switch points in each case can be found, for example, in [5], [6] and [12].

2) Multiple DOFs: Here the authors of [5] first interpolate (the velocity profile of) each joint independently. Suppose the k^{th} joint has the longest duration of T. Then they *re-interpolate* the velocity profiles of the remaining DOFs such that all new velocity profiles have the same duration T. However, this interpolation is not always possible, and determining whether the re-interpolation is possible for a given set of boundary conditions and bounds is actually a difficult problem [6], [12]. The authors of [5] suggest to explore four possibilities similar to $\mathrm{P}^+\mathrm{P}^-$, $\mathrm{P}^-\mathrm{P}^+$, $\mathrm{P}^+\mathrm{L}^+\mathrm{P}^-$, and $\mathrm{P}^-\mathrm{L}^-\mathrm{P}^+$. In each case they constrain both P-ramps to have the same magnitude of acceleration $|a|$. Doing so allows then to subsequently solve for $|a|$. This extra constraint, however, considerably decreases the success rate of the re-interpolation, which in turn decreases the performance of the shortcutting method. A new re-interpolation method – with or without considering the minimum-switch-time constraint – which achieves a higher success rate is proposed in this paper, see Section IV-B.

III. Simultaneous interpolation with zero terminal velocities subject to the minimum-switch-time constraint

Let the minimum allowed switch time be $\delta \geq 0$. Consider first the case when the velocity profile $\dot{s}(t)$ computed in Section II-B is $\mathrm{P}^+\mathrm{P}^-$. Assume that the duration of each ramp (both ramps have equal duration by construction) is $t_s < \delta$. To address this constraint violation, we construct a new velocity profile, $\dot{s}'(t)$, which has two ramps of equal duration. To make the velocity profile time-optimal while not violating the constraint, each ramp must have a duration of $t'_s = \delta$. Thus, the new peak velocity becomes $\dot{s}'_p = 1/\delta$ and the new acceleration of each ramp has the magnitude $|\ddot{s}'| = \dot{s}'_p/\delta$. It can be easily verified from direct calculations that $\dot{s}'_p < \dot{s}_m$ and $|\ddot{s}'| < \ddot{s}_m$.

	t_0	t_1	t_2
PP1	$< \delta$	$\geq \delta$	-
PP2	$\geq \delta$	$< \delta$	-
PP3	$< \delta$	$< \delta$	-
PLP1	$< \delta$	$\geq \delta$	$\geq \delta$
PLP2	$\geq \delta$	$\geq \delta$	$< \delta$
PLP3	$\geq \delta$	$< \delta$	$\geq \delta$
PLP4	$< \delta$	$< \delta$	$\geq \delta$
PLP5	$\geq \delta$	$< \delta$	$< \delta$
PLP6	$< \delta$	$\geq \delta$	$< \delta$
PLP7	$< \delta$	$< \delta$	$< \delta$

TABLE I
All possibilities of minimum-switch-time constraint violation.

Next consider the case when the velocity profile $\dot{s}(t)$ computed in Section II-B is $\mathrm{P}^+\mathrm{L}^+\mathrm{P}^-$ and violates the minimum-switch-time constraint. There are two possible subcases depending on the duration of P-ramps, $t_0 = \dot{s}_m/\ddot{s}_m$.

1) $t_0 \geq \delta$: Here only the L-ramp violates the constraint. We have two possible ways to address this violation: (a) we stretch the L-ramp to δ. The accelerations of the two P-ramp remain unchanged but their durations are shortened so as to maintain the total displacement of 1; (b) we re-interpolate the velocity profile such that it has two P-ramps of equal duration $t'_s = 1/\dot{s}_m$. The peak velocity still saturates the bound while the magnitude of acceleration of each ramp is reduced to $|\ddot{s}'_m| = \dot{s}_m^2$. By comparing the two velocity profiles above, we then choose the valid and smallest one.

2) $t_0 < \delta$: Here both P-ramps violate the constraint. There are three subcases; the condition for each subcase follows directly from direct calculations: (a) $\dot{s}_m\delta \geq 1$. The new time-optimal velocity profile has two ramps of equal duration δ. (b) $2\dot{s}_m\delta \leq 1$. The new time-optimal velocity profile has three ramps; both P-ramps have durations of δ while the L-ramp saturates the velocity bound and has a duration of $1/\dot{s}_m - \delta$. The condition of this case guarantees that the duration of the new L-ramp is not shorter than δ. (c) Otherwise, we proceed in the same way as in the case $\dot{s}_m/\ddot{s}_m \geq \delta$ stated above.

IV. Independent interpolation with arbitrary terminal velocities subject to the minimum-switch-time constraint

A. Single DOF

The task here is to interpolate a velocity profile for each DOF subject to boundary conditions (x_0, v_0) and (x_1, v_1) while respecting the velocity and acceleration bounds v_m and a_m, and the minimum-switch-time constraint δ.

We use a two-step method. In the first step we compute a time-optimal velocity profile subject to only velocity and acceleration constraints. Then in case the minimum-switch-time constraint is violated, we re-interpolate the profile by also taking into account the minimum-switch-time constraint. Let t_0, t_1, and t_2 be the durations of the first ramp, the second ramp, and the third ramp (for PLP) of the original velocity profile. Since the original profile can be either PP or PLP, there are 10 exhaustive cases of possible minimum-switch-time constraint violation which are summarized in Table I.

1) PP1: In the first step we stretch the first ramp to δ. We constrain the second ramp to remain at the same acceleration

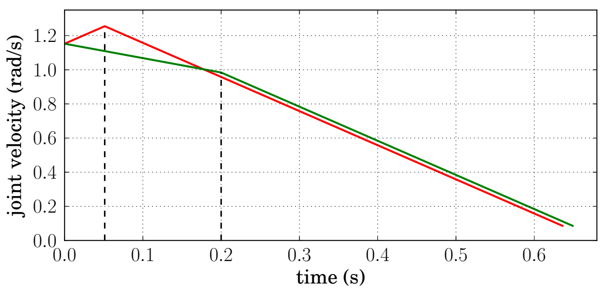

Fig. 2. This figure illustrates our re-interpolation procedure for the case PP1. The original velocity profile is shown in red. The re-interpolated velocity profile is shown in green. The first ramp of the new velocity profile is stretched such that it has the duration of δ.

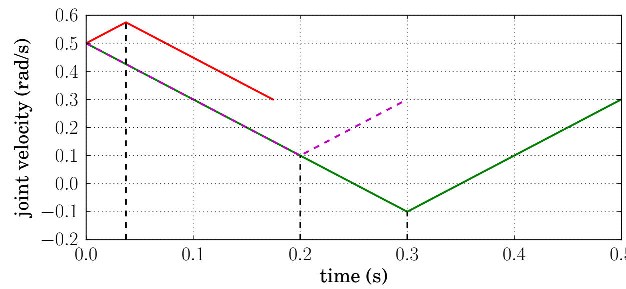

Fig. 3. The original velocity profile is shown in red. The minimum-switch-time constraint δ is set to be 0.2. By using v_p' computed from (2), two velocity profiles can be constructed (shown in dashed magenta and solid green). In this case both profiles are valid.

so it will be shortened. The new peak velocity at the new switch point $t_0' = \delta$ can be computed as

$$v_p' = \frac{1}{2}\left(k_1 - \mathrm{sgn}(k_1)\sqrt{k_1^2 + 4k_1 v_0 + 4v_1^2 - 8a_1 d}\right), \quad (1)$$

where $d = x_1 - x_0$ is the total displacement of the trajectory, $k_1 = a_1\delta$, and a_1 is the acceleration of the last ramp. The new acceleration of the first ramp a_0' can be computed accordingly from $a_0' = (v_p' - v_0)/\delta$. Fig. 2 shows an example of our re-interpolation according to the case PP1. A proof in Appendix A shows that the terms inside the square root in (1) is always non-negative. After following the first step if the new duration of the last ramp, $t_1' = (v_1 - v_p')/a_1$, is shorter than δ, we need to re-interpolate the profile again. This can be done in two ways : (a) the new velocity profile has two ramps of equal duration δ ; (b) the new velocity profile has one single ramp. In both cases we can compute explicitly the total durations as well as the accelerations and peak velocities. Finally we choose the time-optimal one (note that at least (b) is guaranteed to be valid ; see Appendix A).

2) PP2: This case is symmetric with PP1 and the procedure is similar to that of PP1.

3) PP3: There are two possibilities of re-interpolation : (a) the new velocity profile has two ramps of equal duration δ ; (b) the new velocity profile has one single ramp. However, unlike the previous two cases where one-ramp velocity profile is guaranteed to be valid, *i.e.*, not shorter than δ, there is a possibility that one-ramp profile is still shorter than δ. To handle such cases, we shall "flip" the original velocity profile. If the original profile is $\mathrm{P}^+\mathrm{P}^-$, then the new profile will be $\mathrm{P}^-\mathrm{P}^+$ and so on. The idea of flipping the profile is illustrated in Fig. 3. The new peak velocity can then be computed from

$$v_p'^2 = (v_0^2 + v_1^2)/2 + a_0'(x_1 - x_0), \quad (2)$$

provided that the right-hand term is non-negative. Of the two solutions, we shall select the one which makes the new velocity profile valid and shortest possible.

In case $(v_0^2 + v_1^2)/2 + a_0'(x_1 - x_0)$ is negative, or is positive but both of the resulting velocity profiles are not valid, this means both ramps cannot saturate the acceleration bounds at the same time. Here we modify the velocity profile following the idea for PP1 and PP2. For example, consider the case when $t_0 > t_1$, *i.e.*, the first ramp of the original velocity profile is longer than the last. The "flipped" velocity profile will have the last ramp longer than the first. To obtain this flipped profile we follow the routine for PP1. The first ramp of the new profile has a duration of δ while the last ramp saturates the acceleration bound.

4) PLP1: Here our re-interpolation routine explores two possibilities : PLP1A) the re-interpolated profile has three ramps ; PLP1B) the re-interpolated profile has two ramps. Then we choose the case which gives the shorter duration. In PLP1A we stretch the duration of the first ramp of the original profile to δ while the final velocity, v_p, remains at the velocity bound. The acceleration of the new first ramp is then $a_0' = (v_p - v_0)/\delta$. The velocity profile for the remaining portion of the trajectory is re-interpolated using the routine for PP1. In PLP1B the first two ramps of the original velocity profile are merged into a single ramp by retaining the boundary conditions. The third ramp of the original profile remains unchanged. Finally we choose the case which gives the shorter duration. Note that at least the case PLP1B is valid since the merged ramp will always be longer than δ. Here the case where the re-interpolated profile has one ramp is not explored since it cannot be shorter than the velocity profile from PLP1B.

5) PLP2: This case is symmetric with PLP1 and the procedure is similar to that of PLP1.

6) PLP3: In this case we have to stretch the duration of the middle ramp of the original velocity profile to δ. There are two possibilities depending on the acceleration of the new middle ramp, a_1' : PLP3A) $a_1' \neq 0$; PLP3B) $a_1' = 0$. In PLP3A we leave either the first or the last ramps unchanged. The remaining two ramps, which can be viewed as a two-ramp velocity profile that violates the minimum-switch-time constraint, are re-interpolated with routines for PP. To choose which two to be re-interpolated, we explore both cases and choose the case which gives the shorter profile. In PLP3B the first and the last ramp will still have the same accelerations after re-interpolation. The velocity of the new middle ramp, v_p', can then be calculated as

$$v_p' = \frac{1}{2}\left(-k + \mathrm{sgn}(a_0)\sqrt{k^2 + 4a_0 d + 2(v_0^2 + v_1^2)}\right), \quad (3)$$

where $k = a_0\delta$ and a_0 is the acceleration of the first ramp. Finally we choose the case which gives the shorter duration.

7) PLP4: This case can be divided into two subcases : PLP4A) the re-interpolated profile has three ramps ; PLP4B) the re-interpolated profile has two ramps. In PLP4A and PLP4B we proceed in the same way as in PLP1A and PLP1B respectively. However, unlike PLP1 where PLP1B will always give a valid velocity profile, the merged ramp can sometimes be shorter than δ. In that case we re-interpolated the velocity profile again using the routines for PP1, *i.e.*, the duration of the first ramp (the merged ramp) is stretched to δ ; the new peak velocity is then calculated from (1) ; the acceleration of the remaining ramp remains unchanged. Finally we choose the case which gives the shorter duration.

8) PLP5: This case is symmetric with PLP4 and the procedure is similar to that of PLP4.

9) PLP6: We examine 4 exhaustive ways to modify the original velocity profile : PLP6A) the durations of both the first and the last ramps are stretched to δ. The new middle ramp is still at the velocity bound and hence shortened ; PLP6B) we stretch only the duration of the first ramp to δ. There are two subcases depending on the final velocity of the first ramp. In the first subcase the final velocity v_p' is set to the velocity bound. The remaining portion is re-interpolated as a two-ramp velocity profile using the routines for PP. In the second subcase the final velocity v_p' is calculated as in PP1 (Eq.(1)) and the remaining portion is re-interpolated accordingly ; PLP6C) we stretch only the duration of the last ramp to δ. There are two subcases similar to PLP6B) ; PLP6D) we re-interpolate the profile such that it has only one ramp. Finally we choose the case which gives the shorter duration. Note that at least the case PLP6D is valid since the total duration after re-interpolation is always greater than the duration of the original middle ramp which in turn greater than δ.

10) PLP7: We examine 3 cases of re-interpolate profiles : PLP7A the new velocity profile has three ramps. Since all ramps of the original velocity profile are shorter than δ, the duration of each ramp of the new velocity profile must be equal to δ. To re-interpolate a velocity profile to be three-ramp with specified durations for all ramps, we use the procedure described in Appendix C ; PLP7B the new velocity profile has two ramps ; and PLP7C the new velocity profile has one ramp. To re-interpolate a velocity profile into two- or one-ramp (PLP7B or PLP7C), we can use the routine for the case PP3. After exploring all three cases, we choose a valid profile (if any) which gives the shortest duration. The case which all PLP7A, PLP7B, and PLP7C do not give any valid velocity profile can occur, however, when the velocity limit is very low and one of the terminal velocities is zero. In this case we proceed as follows. Let v_p be the peak velocity of the original trajectory. If $|v_p - v_0| > |v_p - v_1|$, we stretch the duration of the new first ramp to δ while the final velocity remains the same. Then we re-interpolate the remaining portion to be one ramp. On the other hand, if $|v_p - v_0| < |v_p - v_1|$, we stretch the duration of the last ramp to δ while the initial velocity of the last ramp remains the same. Then we re-interpolate the remaining portion to be one ramp.

B. Multiple DOFs

After independently interpolating velocity profiles for different joints for a shortcut path, each joint may have different time duration. However, in order to be a valid shortcut, velocity profiles of all the joints for the shortcut path must have the same duration. Let the m^{th} joint, $1 \leq m \leq n$, have the slowest velocity profile of the duration T. Since the shortcut cannot have its duration less than T, the purpose here is to re-interpolate all the remaining $n-1$ joints to have the same duration T.

Given a trajectory and a fixed time T, a velocity profile re-interpolation problem has either infinitely many solutions or no solution [6]. Moreover, even when the minimum-switch-time constraint is not taken into account deciding whether the problem has a solution is difficult [12]. Therefore, to simplify the problem so that we can solve it analytically, the class of the re-interpolated velocity profile (two- or three-ramp) and some of free variables might be specified beforehand. The authors of [5] suggested to explore both classes and to constrain the accelerations of the first and the last ramps to have equal magnitude. However, constraining the acceleration magnitude is restrictive and therefore results in relatively low re-interpolation success rate, as can be seen in Section V-C.

We propose here a routine which also explores both classes of possible velocity profiles but instead of constraining the accelerations, it makes some initial guesses on the duration of each ramp of the re-interpolated velocity profile. In addition, our routine can also take into account the minimum-switch-time constraint for multiple-DOFs by requiring that

$$\min_{i,k,j,l} |t^{sw}(i,j) - t^{sw}(k,l)| \geq \delta, \tag{4}$$

where $t^{sw}(i,j)$ denotes the j^{th} switch point of the i^{th} DOF. This implies that any pair of switch points will be separated by a duration of at least a time duration of δ.

Suppose velocity profiles are $\gamma_i, i = \{1, 2, \ldots, n\}$ and γ_m is the slowest with the duration T. We need to re-interpolate all other velocity profiles, $\gamma_i, i = \{1, 2, \ldots, n\}, i \neq m$, such that they all have the new durations equal to that of γ_m. We start by dividing each ramp of γ_m into equally long segments whose durations are the smallest possible but still larger than δ (in our implementation the total number of such segments is capped to 50 to avoid long computation time when δ is small or null, *i.e.*, no minimum-switch-time constraint). After constructing a grid we explore the two classes of velocity profiles as follows.

1) Two-ramp velocity profile: We try to locate the unique switch point at each of grid lines. The acceleration of each ramp can be analytically determined from the location of the switch point (see Appendix B).

2) Three-ramp velocity profile: There are $\binom{N}{2}$ possible ways to choose two locations of switch points out of N grid lines. To reduce computational cost, instead of trying all possibilities, we make further guess on the locations of the switch points. In particular, let t_0', t_1', and t_2' be durations of the first, the middle, and the last ramps of the new velocity profile. Further guesses are made on the ratios $t_0'/T, t_1'/T$, and t_2'/T. Choices of these three ratios are arbitrary. However, with some trial-and-error testings, we suggest the values $t_0'/T = 1/4$, $t_1'/T = 1/2$, and $t_2'/T = 1/4$ which give fairly high success rate of re-interpolation. Then we examine 4 ways of snapping those switch points to their nearest grid lines. Given the duration of each ramp, we can formulate the problem of finding accelerations, a_0', a_1', and a_2', for all ramps as a feasibility problem which can be solved analytically (see Appendix C).

If neither the two- nor the three-ramp attempt succeeds, then the re-interpolation is considered as failed and the corresponding shortcut path is discarded.

V. Implementation and use in motion planning

A. Implementation and comparison with optimization method

Finding the time-optimal parabolic interpolation subject to velocity, acceleration, and minimum-switch-time constraints can also be formulated as an optimization problem. The optimization variables are $\boldsymbol{y} = [t'_0, t'_1, v'_p]^T$ for two-ramp profiles, where v'_p is the new peak velocity and $\boldsymbol{z} = [t'_0, t'_1, t'_2, v'_{p0}, v'_{p1}]^T$ for three-ramp profiles, where v'_{p0} is the initial velocity of the new second ramp and v'_{p1} is the initial velocity of the new third ramp.

Let the boundary conditions be (x_0, v_0) and (x_1, v_1), and $d = x_1 - x_0$. For two-ramp velocity profiles the problem can be formulated as

$$\begin{aligned} \underset{\boldsymbol{y}}{\text{minimize}} \quad & \boldsymbol{c}_0^T \boldsymbol{y} \\ \text{subject to} \quad & \boldsymbol{G}_0 \boldsymbol{y} \preceq \boldsymbol{h}_0 \\ & \frac{1}{2}\boldsymbol{y}^T \boldsymbol{A}_0 \boldsymbol{y} + \boldsymbol{b}_0^T \boldsymbol{y} - 2d = 0, \end{aligned} \tag{5}$$

where $\boldsymbol{A}_0 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $\boldsymbol{b}_0 = \begin{bmatrix} v_0 \\ v_1 \\ 0 \end{bmatrix}$, $\boldsymbol{c}_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$,

$$\boldsymbol{G}_0 = \begin{bmatrix} -a_m & 0 & 1 \\ -a_m & 0 & -1 \\ 0 & -a_m & 1 \\ 0 & -a_m & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \text{ and } \boldsymbol{h}_0 = \begin{bmatrix} v_0 \\ -v_0 \\ v_1 \\ -v_1 \\ -\delta \\ -\delta \\ v_m \\ v_m \end{bmatrix}.$$

Similarly, an optimization problem for the three-ramp case can be formulated as

$$\begin{aligned} \underset{\boldsymbol{y}}{\text{minimize}} \quad & \boldsymbol{c}_1^T \boldsymbol{z} \\ \text{subject to} \quad & \boldsymbol{G}_1 \boldsymbol{z} \preceq \boldsymbol{h}_1 \\ & \frac{1}{2}\boldsymbol{z}^T \boldsymbol{A}_1 \boldsymbol{z} + \boldsymbol{b}_1^T \boldsymbol{z} - 2d = 0, \end{aligned} \tag{6}$$

where $\boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$, $\boldsymbol{b}_1 = \begin{bmatrix} v_0 \\ 0 \\ v_1 \\ 0 \\ 0 \end{bmatrix}$, $\boldsymbol{c}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$,

$$\boldsymbol{G}_1 = \begin{bmatrix} -a_m & 0 & 0 & 1 & 0 \\ -a_m & 0 & 0 & -1 & 0 \\ 0 & -a_m & 0 & 1 & -1 \\ 0 & -a_m & 0 & -1 & 1 \\ 0 & 0 & -a_m & 0 & 1 \\ 0 & 0 & -a_m & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \text{ and } \boldsymbol{h}_1 = \begin{bmatrix} v_0 \\ -v_0 \\ 0 \\ 0 \\ v_1 \\ -v_1 \\ -\delta \\ -\delta \\ -\delta \\ v_m \\ v_m \\ v_m \\ v_m \end{bmatrix}.$$

Algorithms	PP		PLP	
	Optim.	Exact	Optim.	Exact
Comp. time (s.)	0.0181	0.000107	0.0418	0.000295
% stuck	13.8%	–	14.4%	–
% failed	2.2%	–	8.6%	–

TABLE II
A comparison between using optimization and the proposed methods

Inequality constraints represent constraints on velocity, acceleration, and minimum-switch-time while equality constraints represent constraints on the total displacement of trajectories. From the above formulations we can obviously see that both problems are non-convex since the matrices $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ are not positive-semidefinite, *i.e.*, some of their eigenvalues are negative. The non-convexity of the problems makes them difficult and more computationally expensive to solve. In addition, a solver can also be stuck in local minima.

For comparison, we created 500 random problem instances for each class of velocity profile. We then used both proposed re-interpolation routines (Section IV) and PyIpopt[2] to solve the same problems. Average computation time, the number of times the optimizer was stuck in a local minimum, and the number of times the optimizer failed (the maximum number of iterations exceeded) were recorded. The statistics are reported in Table II. We found that the solution velocity profiles returned from the optimizer always had either the same or longer duration (the latter case implies that the optimizer was stuck in a local minima).

B. Trajectory planning and shortcutting with minimum-switch-time constraint

Here we include the minimum-switch-time constraint into the motion planning pipeline described in Section II-A. In the trajectory planning stage we use the routines described in Section III to time-parameterize the path taking into account the minimum-switch-time constraint. The parameterization is done through the path parameter s and therefore does not affect the path geometry.

In the shortcutting stage the condition (4) also needs to be satisfied. In each shortcutting iteration we begin by randomly selecting two time instants t_a and t_b. Let t_a^{sw} be the nearest switch point preceding t_a and t_b^{sw} the nearest consecutive switch point of t_b. If $t_a - t_a^{sw} < \delta$, then we re-assign t_a to be t_a^{sw}. Similarly, if $t_b^{sw} - t_b < \delta$, then we re-assign t_b to be t_b^{sw}. Next we continue in a usual manner by first interpolating a velocity profile for each DOF independently and determining the duration T of the slowest DOF. In case $T > t_b - t_a$ the shortcut is discarded right away. Otherwise, we use the routines described in Section IV to re-interpolate all the remaining velocity profiles again. And if the resulting trajectory is collision-free, then we replace the original trajectory segment with the shortcut.

C. Simulation results and comparisons with an existing method

For all the simulations we used the joint velocity and acceleration bounds of Denso VS-060 which is a 6-DOF

[2]PyIpopt is a freely available Python optimization module; see `https://github.com/xuy/pyipopt`

industrial manipulator.

First we made a comparison between a re-interpolation routine proposed in [5] and the routine proposed in Section IV with three minimum-switch-time constraints: $\delta = 0$ (no minimum-switch-time constraint), $\delta = 0.008$ s., and $\delta = 0.1$ s., in terms of success rate of independent interpolation. Comparison were made with two cases of generated pairs of boundary conditions $\{(\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0), (\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1)\}$. In the first case we randomly generated $1,000$ pairs of boundary conditions while in the other case $1,000$ pairs of boundary conditions were randomly picked from existing trajectories. For the latter case we started with two (random) pairs of $\boldsymbol{x}_{\text{init}}$ and $\boldsymbol{x}_{\text{goal}}$. Then we planned 50 trajectories connecting each pair using the aforementioned planning procedure (without shortcutting). Then for each of 100 trajectories we randomly picked 10 pairs of boundary conditions. The statistics in Table III show that the re-interpolation routines proposed in [5] give relatively low success rate due to restrictive constraints on ramp accelerations.

For the next comparison a piecewise linear path was generated for each of two pairs of the initial and goal configurations. Then for each of path segments we simultaneously interpolated velocity profiles with zero terminal velocities, both with and without the minimum-switch-time constraint Finally we gave 1 s. to both proposed method and the one proposed in [5] to shortcut each trajectory. This process was repeated $1,000$ times. Initial trajectory durations, average final durations, and numbers of shortcutting iterations executed from each method are reported in Table IV.

Despite the fact that the proposed method is more computationally expensive as can be seen from Table III, when the same amount of time are given to both methods, the proposed method still performs better in terms of resulting trajectory duration after shortcutting.

VI. Conclusion

In this paper we introduce a new constraint, minimum-switch-time constraint, which is a constraint on time intervals between consecutive switch points. Integrating this constraint into interpolation routines can prevent velocity profiles from having high-frequency switching in accelerations. For industrial manipulators this therefore reduces joint motor wear.

We then present a new algorithm for planning piecewise parabolic trajectories subject to velocity, acceleration, and minimum-switch-time constraints. The algorithm starts by computing time-optimal velocity profiles for straight line paths subject to only velocity and acceleration constraints as in [5], [6], and [12]. Then we address minimum-switch-time constraint violations case by case. We also propose a new strategy for re-interpolation of velocity profiles given a fixed time duration. Simulation results show that our method achieves a higher interpolation success rate and gives higher quality trajectories than the previous method [5].

References

[1] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[2] Q.-C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *Robotics, IEEE Transactions on*, vol. 6, pp. 1533–1540, 2014.

[3] Z. Shiller and S. Dubowsky, "Robot path planning with obstacles, actuator, gripper, and payload constraints," *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 3–18, 1989.

[4] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM (JACM)*, vol. 40, pp. 1048–1066, 1993.

[5] K. Hauser and V. Ng-Thow-Hing, "Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts," in *Robotics and Automation, 2010. Proceedings. ICRA '10. IEEE International Conference on*, 2010, pp. 2493–2498.

[6] T. Kunz and M. Stilman, "Probabilisitically complete kinodynamic planning for robot manipulators with acceleration limits," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, 2014.

[7] S. E. Thompson and R. V. Patel, "Formulation of joint trajectories for industrial robots using B-splines," *Industrial Electronics, IEEE Transactions on*, vol. IE-34, no. 2, pp. 192–199, 1987.

[8] J. S. Gyorfi and C.-H. Wu, "A minimum-jerk speed-planning algorithm for coordinated planning and control of automated assembly manufacturing," *Automation Science and Engineering, IEEE Transactions on*, vol. 3, no. 4, pp. 454–462, 2006.

[9] K. Ahn, W. K. Chung, and Y. Youm, "Arbitrary states polynomial-like trajectory (ASPOT) generation," in *Industrial Electronics Society (IECON), 2004. 30th Annual Conference of IEEE*, vol. 1, 2004, pp. 123–128.

[10] J. W. Jeon, "An efficient acceleration for fast motion of industrial robots," in *Industrial Electronics, Control, and Instrumentation, 1995., Proceedings of the 1995 IEEE IECON 21st International Conference on*, vol. 2, 1995, pp. 1336–1341.

[11] J. W. Jeon and Y. Y. Ha, "A generalized approach for the acceleration and deceleration of industrial robots and CNC machine tools," *Industrial Electronics, IEEE Transactions on*, vol. 47, no. 1, pp. 133–139, 2000.

[12] T. Kröger, A. Tomiczek, and F. M. Wahl, "Towards on-line trajectory computation," in *Intelligent Robots and Systems (IROS), 2006 IEEE/RSJ International Conference on*, 2006, pp. 736–741.

[13] R. Geraerts and M. Overmars, "Creating high-quality paths for motion planning," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 845–863, 2007.

[14] J. Luo and K. Hauser, "An empirical study of optimal motion planning," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 2014, pp. 1761–1768.

[15] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, August 2010. [Online]. Available: http://www.programmingvision.com/rosen_diankov_thesis.pdf

[16] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[17] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, Tech. Rep., 1998.

[18] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

Appendix A
Proof of time-optimality

We present here detailed proofs for only the cases PP1 and PLP1. Proofs for cases PP2 and PLP2 can be easily adapted from PP1 and PLP1, respectively, while for the remaining cases, we have already explored all possibilities of re-interpolation.

Consider first the case PP1. Let T_{PP1} be the total duration of the newly re-interpolated velocity profile and t'_0 the duration of the new first ramp. Thus, $T_{\text{PP1}} = t'_0 + (v_1 - v'_p)/a_1$, where v_1 and a_1 are the final velocity and the acceleration of the second ramp, and v'_p is the new peak velocity. The following proposition holds.

Proposition 1. *T_{PP1} is an increasing function in t'_0.*

	$(\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0)$ and $(\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1)$ generated randomly		$(\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0)$ and $(\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1)$ randomly picked from trajectories			
			Trajectories connecting $\boldsymbol{x}_{\text{init},0}$ and $\boldsymbol{x}_{\text{goal},0}$		Trajectories connecting $\boldsymbol{x}_{\text{init},1}$ and $\boldsymbol{x}_{\text{goal},1}$	
	Success rate	Avg. comp. time (ms.)	Success rate	Avg. comp. time (ms.)	Success rate	Avg. comp. time (ms.)
From [5]	36.5%	0.364	61.0%	0.324	51.8%	0.381
$\delta = 0.0$	78.1%	1.394	90.8%	1.512	85.8%	1.606
$\delta = 0.008$	77.0%	1.431	89.8%	1.581	85.2%	1.666
$\delta = 0.1$	75.7%	1.420	75.4%	2.806	74.0%	2.859

TABLE III
INDEPENDENT INTERPOLATION WITH ARBITRARY TERMINAL VELOCITIES (MULTIPLE DOFS): SUCCESS RATE AND AVERAGE COMPUTATION TIME

	A trajectory connecting $\boldsymbol{x}_{\text{init},0}$ and $\boldsymbol{x}_{\text{goal},0}$			A trajectory connecting $\boldsymbol{x}_{\text{init},1}$ and $\boldsymbol{x}_{\text{goal},1}$		
	Initial dur. (s.)	Avg. final dur. (s.)	Avg. # iterations	Initial dur. (s.)	Avg. final dur. (s.)	Avg. # iterations
From [5]	2.831	1.802	413.1	7.000	3.002	237.3
$\delta = 0.0$	2.831	1.360	171.71	7.000	2.275	131.6
$\delta = 0.008$	2.831	1.372	178.4	7.000	2.272	127.8
$\delta = 0.1$	2.903	1.710	228.7	7.088	2.235	153.1

TABLE IV
BOTH PROPOSED METHOD AND THE ONE FROM [5] WERE GIVEN 1 S. TO SHORTCUT GIVEN TRAJECTORIES. THE EXPERIMENT WAS REPEATED $1,000$ TIMES FOR EACH TRAJECTORY.

Proof: We first consider the case $\mathrm{P}^+\mathrm{P}^-$. Substituting the expression of v'_p from (1) and $a_1 = -a_m$ into the expression of T_{PP1} yields

$$T_{\text{PP1}} = t'_0 + \frac{1}{2a_m}\left(k_1 + \sqrt{k_1^2 + 4k_1 v_0 + 4v_1^2 + 8a_m d}\right) - \frac{v_1}{a_m}$$

$$T_{\text{PP1}} = \frac{1}{2}\left(t'_0 - \frac{2v_1}{a_m}\right) + \frac{1}{2}\sqrt{\Delta},$$

where $\Delta = \left(t'_0 - 2v_0/a_m\right)^2 + 4\left(v_1^2 - v_0^2 + 2a_m d\right)/a_m^2$. The first derivative of T_{PP1} with respect to t'_0 is then given by

$$\frac{d}{dt'_0}T_{\text{PP1}} = \frac{1}{2} + \frac{1}{2}\frac{t'_0 - \frac{2v_0}{a_m}}{\sqrt{\left(t'_0 - \frac{2v_0}{a_m}\right)^2 + \frac{4}{a_m^2}\left(v_1^2 - v_0^2 + 2a_m d\right)}}. \tag{7}$$

If $t'_0 > 2v_0/a_m$, it follows directly that $dT_{\text{PP1}}/dt'_0 > 0$. Otherwise, we examine the second term in the square root.

Before proceeding further, notice that for all $\mathrm{P}^+\mathrm{P}^-1$-velocity profile, we always have that $v_0 > v_1$. Assume, for contradiction, that a velocity profile falls in the case $\mathrm{P}^+\mathrm{P}^-1$ with $v_0 < v_1$. Since $|a_0| = |a_1| = a_m$, we then have that $(v_p - v_0)/a_m > (v_p - v_1)/a_m$ which results in $t_0 > t_1$. This contradicts with the assumption that the trajectory is in the case $\mathrm{P}^+\mathrm{P}^-1$.

Now consider when $v_0 > v_1$. The time-optimal velocity profile connecting v_0 and v_1 subject to velocity and acceleration bounds has one ramp if and only if the total displacement is $d^* = (v_0^2 - v_1^2)/(2a_m)$. Therefore, for any $\mathrm{P}^+\mathrm{P}^-$-velocity profile, the total displacement must be greater than d^*, *i.e.*,

$$v_1^2 - v_0^2 + 2a_m d > 0 \tag{8}$$

By applying (8) to (7),we can see that dT_{PP1}/dt'_0 is always positive. This completes the proof for the case $\mathrm{P}^+\mathrm{P}^-$. For the case $\mathrm{P}^-\mathrm{P}^+$, the same argument which leads to $dT_{\text{PP1}}/dt'_0 > 0$ can be made. This completes the proof for the case PP1. ■

Therefore, following Proposition 1, to time-optimally re-interpolate a velocity profile subject to the minimum-switch-time constraint δ, the resulting duration of the first ramp must be the shortest possible, $t'_0 = \delta$.

Next consider the case PLP1. Let t'_0 be the duration of the first ramp of the re-interpolated velocity profile. We claim that in order for the re-interpolated velocity profile to be time-optimal as well as satisfying all the constraints, t'_0 must be only in $[t_0^A, t_0^B]$, where t_0^A is the duration of the new first ramp in case the velocity profile is re-interpolated according to the case PLP1A, *i.e.*, $t_0^A = \delta$, and t_0^B is the duration of the new first ramp in case the velocity profile is re-interpolated according to the case PLP1B. Fig. 4 shows velocity profiles of the original PLP1-velocity profile as well as resulting velocity profiles from the cases PLP1A and PLP1B.

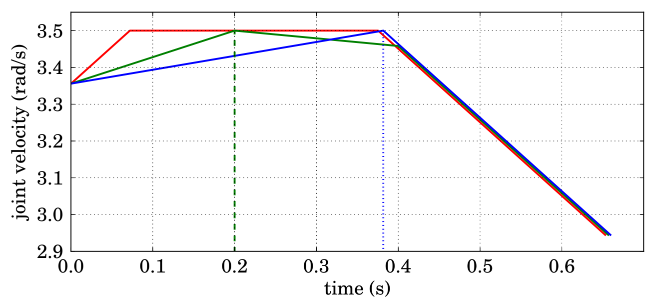

Fig. 4. The original PLP1-velocity profile which violates the minimum-switch-time constraint $\delta = 0.2$ is shown in red. The green and blue velocity profiles result from re-interpolating according to PLP1A and PLP1B respectively. The time instants t_0^A and t_0^B are marked with a dashed green line and a dotted blue line respectively. In this particular example, re-interpolation with PLP1A gives a shorter velocity profile.

The re-interpolation procedure of a PLP1-velocity profile can be thought of as first re-interpolating the first ramp with a duration $t'_0 \in [t_0^A, t_0^B]$ and then re-interpolating the rest of the profile using routines for PP1. Let $d'_0 = (v'_p - v_0)t'_0/2$ be the displacement covered by the first ramp of the newly re-interpolated velocity profile. The displacement covered by the remaining part of the velocity profile is then $d_{\text{rem}} = d - d'_0$. Let the total duration of the new velocity profile be T_{PLP1}. The following proposition holds.

Proposition 2. *T_{PLP1}, as a function of t'_0, attains its minimum only when $t'_0 = t_0^A$ or $t'_0 = t_0^B$.*

Proof: We first consider the case when $v'_p = v_m$. The total duration T_{PLP1} can be expressed as

$$T_{\text{PLP1}} = t'_0 + T_{\text{PP1}}$$
$$T_{\text{PLP1}} = t'_0 + \frac{1}{2}\left(\delta - \frac{2v_1}{a_m}\right) + \frac{1}{2}\sqrt{\left(\delta - \frac{2v'_p}{a_m}\right)^2 + \frac{4}{a_m^2}\left(v_1^2 - {v'_p}^2 + 2a_m d_{rem}\right)}.$$

From the expression of T_{PLP1}, we can compute $d^2T_{\text{PLP1}}/{dt'_0}^2$ which is always negative. This implies that the minimum of T_{PLP1} can only occur when either $t'_0 = t_0^A$ or $t'_0 = t_0^B$.

For the case when $v'_p = -v_m$, the similar argument which leads to $d^2T_{\text{PLP1}}/{dt'_0}^2$ can be made. This completes the proof for the case PLP1. ∎

Therefore, following Proposition 2, we can conclude that the time-optimal velocity profile subject to the minimum-switch-time constraint δ must be obtained from either PLP1A or PLP1B.

Appendix B
Velocity profile re-interpolation to be two-ramp

Let T be the given total duration and τ_0 the given switch point. Since the total displacement must remain the same after re-interpolation, we have

$$d = \left(v_0\tau_0 + \frac{1}{2}a'_0{\tau_0}^2\right) + \left((v_0 + a'_0\tau_0)\tau_1 + \frac{1}{2}a'_1{\tau_1}^2\right), \quad (9)$$

where a'_0 and a'_1 are accelerations of the first and the last ramp of the new velocity profile respectively, and $\tau_1 = T - \tau_0$. We also have a relationship between v_0 and v_1 as

$$v_1 = v_0 + a'_0\tau_0 + a'_1\tau_1. \quad (10)$$

We can rewrite Eq. (9) and (10) as a system of linear equations in a'_0 and a'_1 as

$$\begin{bmatrix} \frac{1}{2}\tau_0^2 + \tau_0\tau_1 & \frac{1}{2}\tau_1^2 \\ \tau_0 & \tau_1 \end{bmatrix} \begin{bmatrix} a'_0 \\ a'_1 \end{bmatrix} = \begin{bmatrix} d - v_0T \\ v_1 - v_0 \end{bmatrix}. \quad (11)$$

We can then solve for a'_0 and a'_1. Since the determinant of the square matrix is $\tau_0\tau_1(\tau_0 + \tau_1)/2 \neq 0$, the system (11) is always solvable.

Appendix C
Velocity profile re-interpolation to be three-ramp

Let T be the given total duration; τ_0, τ_1, τ_2 the desired durations of the first, the middle and the last ramps of the new velocity profile respectively. Since the total displacement must remain the same after re-interpolation, we have

$$d = \left(v_0\tau_0 + \frac{1}{2}a'_0{\tau_0}^2\right) + \left((v_0 + a'_0\tau_0)\tau_1 + \frac{1}{2}a'_1{\tau_1}^2\right) + \left((v_0 + a'_0\tau_0 + a'_1\tau_1)\tau_2 + \frac{1}{2}a'_2{\tau_2}^2\right), \quad (12)$$

where v_0 is the initial velocity; v'_{p0} and v'_{p1} are the first and the second peak velocities respectively; a'_0, a'_1, and a'_2 are accelerations of the first, middle, and last ramps of the new velocity profiles respectively. Let $\alpha = \tau_0^2/2 + \tau_0\tau_1 + \tau_0\tau_2$, $\beta = \tau_1^2/2 + \tau_1\tau_2$, and $\sigma = \tau_2^2/2$. Then Eq. (12) can be more compactly rewritten as

$$\alpha a'_0 + \beta a'_1 + \sigma a'_2 = d - v_0T. \quad (13)$$

Furthermore, we can derive a relationship between v_0 and v_1, the final velocity, as

$$v_1 = v_0 + \tau_0a'_0 + \tau_1a'_1 + \tau_2a'_2. \quad (14)$$

We can write Eq. (13) and (14) as a system of linear equations in $\boldsymbol{a} = [a'_0, a'_1, a'_2]^T$ as

$$\boldsymbol{A}\boldsymbol{a} = \boldsymbol{b}, \quad (15)$$

where $A = \begin{bmatrix} \alpha & \beta & \sigma \\ \tau_0 & \tau_1 & \tau_2 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} d - v_0T \\ v_1 - v_0 \end{bmatrix}$.

The system (15) is underdetermined. Therefore, a general solution, $\boldsymbol{a}_g$, to the system can be expressed as $\boldsymbol{a}_g = \boldsymbol{a}_p + \boldsymbol{a}_h$, where $\boldsymbol{a}_p$ is a particular solution and $\boldsymbol{a}_h$ is a non-trivial element of the null space of $\boldsymbol{A}$, *i.e.*, satisfying $\boldsymbol{A}\boldsymbol{a}_h = 0$. Here we take the minimum-norm solution to (15) as a particular solution, *i.e.*, $\boldsymbol{a}_p = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b}$. For a homogeneous solution, we perform Gaussian elimination on $\boldsymbol{A}$ to obtain

$$\boldsymbol{a}_h = k\begin{bmatrix} (\beta\tau_2 - \sigma\tau_1)/(\alpha\tau_1 - \beta\tau_0) \\ (\alpha\tau_2 - \sigma\tau_0)/(\alpha\tau_1 - \beta\tau_0) \\ 1 \end{bmatrix},$$

where $k \in \mathbf{R}$.

Next consider the velocity bound v_m. The peak velocities of the re-interpolated velocity profiles are $v'_{p0} = v_0 + a'_0\tau_0$ and $v'_{p1} = v_1 - a'_2\tau_2$. Therefore, we get two sets of constraints on the peak velocities, in terms of the accelerations, as

$$\begin{array}{ccccc} (-v_m - v_0)/\tau_0 & \leq & a'_0 & \leq & (v_m - v_0)/\tau_0 \quad \text{and} \\ (-v_m + v_1)/\tau_0 & \leq & a'_2 & \leq & (v_m + v_1)/\tau_0. \end{array}$$

Combining velocity and acceleration bounds together yields constraints on accelerations as

$$\boldsymbol{l} \preceq \boldsymbol{a}_p + k\boldsymbol{a}_h \preceq \boldsymbol{u}, \quad (16)$$

where

$$\boldsymbol{l} = \begin{bmatrix} \min\left(\frac{v_m + v_0}{\tau_0}, a_m\right) \\ -a_m \\ \min\left(\frac{v_m - v_1}{\tau_0}, a_m\right) \end{bmatrix} \text{ and } \boldsymbol{u} = \begin{bmatrix} \min\left(\frac{v_m - v_0}{\tau_0}, a_m\right) \\ a_m \\ \min\left(\frac{v_m + v_1}{\tau_0}, a_m\right) \end{bmatrix}.$$

Finally, we can solve for the feasible interval k from (16). If there is no such k satisfying (16), then no feasible solution for acceleration exists. Otherwise, any k in the interval can be used to construct a feasible solution for accelerations.