Georgia Institute of Technology, Technical Report, GT-GOLEM-2011-006, 2011

Turning Paths Into Trajectories Using Parabolic Blends

Tobias Kunz and Mike Stilman

***Abstract*— We present an approach for converting a path of multiple continuous linear segments into a trajectory that satisfies velocity and acceleration constraints and closely follows the given path without coming to a complete stop at every waypoint. Our method applies parabolic blends around waypoints to improve speed. In contrast to established methods that smooth trajectories with parabolic blends, our method does not require the timing of waypoints or durations of blend phases. This makes our approach particularly useful for robots that must follow kinematic paths that are not explicitly parametrized by time. Our method chooses timing automatically to achieve high performance while satisfying the velocity and acceleration constraints of a given robot.**

I. Introduction

The output of typical robot motion planners is a path in configuration space consisting of continuous straight line segments between waypoints. Such a path is not differentiable at the waypoints. Since the robot has nonzero mass and the forces that can be applied are finite, the robot cannot instantaneously change its direction of motion. In order to follow the path precisely, it would have to come to a complete stop at each of the waypoints leading to slow execution. One standard method of avoiding a complete stop at the waypoints is that of adding parabolic blends. However, the treatment of parabolic blends in existing literature such as [1] and [2] is not directly applicable to kinematic paths. [1] and [2] assume that we know the time durations of the linear and blend phases. In contrast to trajectories, paths do not explicitly refer to time. An automated solution to finding the durations that satisfy the velocity and acceleration constraints of a given robot is crucial for turning paths into trajectories.

We present an algorithm that automatically chooses durations to generate a smooth, valid trajectory. Our algorithm attempts to generate the fastest trajectory possible given the constraints on joint accelerations and velocities. But in order for the trajectory to be valid, the algorithm might in certain cases resort to highly suboptimal trajectories with respect to time.

In Section II we define the problem of turning paths into trajectories more formally. Section III presents linear polynomials with parabolic blends as described in [1] and [2]. We cover these topics more thoroughly with greater attention to detail. In Section IV we introduce a new method to automatically choose the durations of the linear and blend phases such that velocity and acceleration constraints are met. Finally, Section V discusses the proposed method, its strengths and limitations.

The authors are with the Center for Robotics and Intelligent Machines (RIM) at the Georgia Institute of Technology, Atlanta 30332, USA. Email: tobias@gatech.edu, mstilman@cc.gatech.edu

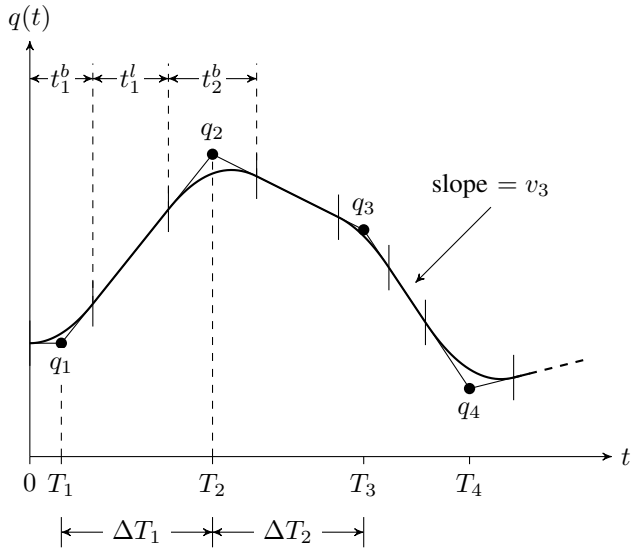

Fig. 1. One-dimensional example of a linear function with parabolic blends. The first four of at least five waypoints are shown.

II. Problem Statement

We are given a path in configuration space $\mathcal{C}$ as a list of waypoints $q_1, \ldots, q_n$ with $q_i \in \mathcal{C}$. The path consists of straight line segments connecting these waypoints. We are looking for a fast trajectory that follows this path closely and satisfies velocity and acceleration constraints. A trajectory is defined by a duration t_f and a function $q : [0, t_f] \to \mathcal{C}$, which returns the robot's configuration at any continuous point in time. We are looking for a trajectory that satisfies the following constraints:

1) It must start and end at the first and last waypoint respectively with zero velocity, $q(0) = w_1$, $q(t_f) = w_n$ and $\dot{q}(0) = \dot{q}(t_f) = 0$.
2) The trajectory must satisfy velocity and acceleration constraints $\forall t : |\dot{q}(t)| \leq v_{\max} \wedge |\ddot{q}(t)| \leq a_{\max}$

We do not formally define the requirement that the trajectory stay close to the original path. We only guarantee to move along each of the straight line segments of the path once and in the correct order. In-between following parts of the straight line segments, the trajectory deviates from the original path.

III. Linear Polynomials with Parabolic Blends

This section explains the generation of trajectories that consist of two alternating phases, a linear phase and a parabolic blend phase. Following [1] and [2] we present a detailed method for creating such a trajectory from waypoint locations and timing information. Note that during the linear

phase acceleration is zero, velocity is constant and position is linear in time. During the parabolic blend phase acceleration is constant, velocity is linear and position is quadratic, and thus parabolic, in time. Furthermore, during the parabolic blend phase, the trajectory follows a parabola in configuration space.

For the moment, we assume that the timing of the waypoints is known. Hence, for each pair of neighboring waypoints q_i and q_{i+1} we are given a time ΔT_i, that it takes to move in between the two. We also assume that we are given the duration t_i^b of the blend phase at waypoint i. If we move on a straight line at constant velocity in-between waypoints then the velocity between waypoint i and $i+1$ is given by Eq. 1.

$$v_i = \frac{q_{i+1} - q_i}{\Delta T_i} \tag{1}$$

Alternatively, we could have assumed that we are given velocities and calculated the timing.

Moving from one waypoint to the next at constant velocity, leads to discontinuous velocities at the waypoints. A nondifferentiable trajectory that moves from one waypoint to the next at constant velocity is given by Eq. 2.

$$q_l(t) = T_i + v_i(t - T_i) \quad \text{if } T_i \leq t \leq T_{i+1} \quad i \in [1, ..., n-1] \tag{2}$$

In order to create a continuous trajectory, we insert blend phases at the waypoints. The overall timing of the trajectory is not affected. The blend phase replaces part of neighboring straight line segments with a parabola that is followed at a constant acceleration. Around waypoint i, acceleration is given by Eq. 3

$$a_i = \frac{v_i - v_{i-1}}{t_i^b} \tag{3}$$

The blend phase is centered around the waypoint with respect to time. In other words, it overlaps with both neighboring linear phases for the same amount of time. In order for the trajectory to be valid, two blend phases cannot overlap with each other. Thus, the timing of the waypoints and the durations of the blend phases need to be given such that

$$t_i^b + t_{i+1}^b \leq 2\Delta T_i. \tag{4}$$

So far, we have defined the acceleration during the blend phase such that we move from velocity v_{i-1} to v_i in the given blend time t_i^b. We must also make sure that replacing part of the straight line segments with parabolic blends results in a continuous trajectory such that at the start and end time of the blend the robot is at the same configuration it would be following the straight-line trajectory $q_l(t)$. This condition is satisfied as shown in Lemma 1.

Lemma 1: Let T_{li} be the time at which waypoint, i, is reached when following $q_l(t)$. If we start the blend around waypoint i at time $T_{li} - \frac{t_i^b}{2}$, configuration $q_l(T_{li} - \frac{t_i^b}{2})$ and velocity $\dot{q}_l(T_{li} - \frac{t_i^b}{2})$, and apply constant acceleration a_i during the blend, then at the end of the blend phase at time $T_{li} + \frac{t_i^b}{2}$ the robot is at configuration $q_l(T_{li} + \frac{t_i^b}{2})$.

Proof: The configuration of the robot during a parabolic blend at time t after the start of the blend is given by

$$b(t) = b_0 + \dot{b}_0 t + \frac{1}{2} a t^2, \tag{5}$$

where b_0 is the configuration at which the blend is started, $\dot{b}_0$ is the velocity at which the blend is started and a is the constant acceleration during the blend. For a blend around waypoint i, these quantities are given as

$$b_0 = q_l(T_{li} - \frac{t_i^b}{2}) = q_i - v_{i-1}\frac{t_i^b}{2}, \tag{6}$$

$$\dot{b}_0 = \dot{q}_l(T_{li} - \frac{t_i^b}{2}) = v_{i-1}. \tag{7}$$

$$a = a_i \tag{8}$$

Using these quantities we show that $b(t_i^b) = q_l(T_i + \frac{t_i^b}{2})$.

$$b(t_i^b) = b_0 + \dot{b}_0 t_i^b + \frac{1}{2} a (t_i^b)^2 \tag{9}$$

$$= \left(q_i - v_{i-1}\frac{t_i^b}{2}\right) + v_{i-1} t_i^b + \frac{1}{2} a_i (t_i^b)^2 \tag{10}$$

$$= q_i + v_{i-1}\frac{t_i^b}{2} + \frac{1}{2} a_i (t_i^b)^2 \tag{11}$$

$$= q_i + v_{i-1}\frac{t_i^b}{2} + \frac{1}{2}\frac{v_i - v_{i-1}}{t_i^b}(t_i^b)^2 \tag{12}$$

$$= q_i + v_i \frac{t_i^b}{2} \tag{13}$$

$$= q_l(T_i + \frac{t_i^b}{2}) \tag{14}$$

■

Since the blend phases replace part of the linear phases, the parts of the linear phases that actually gets executed before and after waypoint i are shorter than ΔT_{i-1}, ΔT_i respectively. During the linear phase that is not overlapped by some blend we move along a part of the line segment connecting waypoints q_i and q_{i+1} at velocity v_i. Between waypoints i and $i+1$ the trajectory starts to follow the straight line at time $T_i + \frac{t_i^b}{2}$ and stops at $T_{i+1} - \frac{t_{i+1}^b}{2}$. Thus, the duration of the actually executed linear phase between q_i and q_{i+1} is given by Eq. 15.

$$t_i^l = \Delta T_i - \frac{t_i^b}{2} - \frac{t_{i+1}^b}{2} \tag{15}$$

Special care needs to be taken with the first and last waypoints. The blend phases at the first and last waypoints only have one neighboring linear phase. Because we start and end at rest, we can pretend that there is a linear segment before the first and after the last waypoint with zero velocity and undetermined duration. We then treat the first and last blend phase like all the others. Remember that the blend phase is centered in time around the waypoint. As the first and last blend phase need to be part of the trajectory completely, the trajectory starts $\frac{t_0^b}{2}$ before the first waypoint and ends $\frac{t_n^b}{2}$ after the last waypoint. The time of a waypoint i is redefined as

follows:

$$T_i = \frac{t_1^b}{2} + \sum_{j=1}^{i-1} \Delta T_j \tag{16}$$

The total duration of the trajectory is

$$t_f = \frac{t_1^b}{2} + \sum_{i=1}^{n-1} \Delta T_i + \frac{t_n^b}{2}. \tag{17}$$

The resulting trajectory is given by $q : [0, t_f] \to \mathcal{C}$ with

$$q(t) = \begin{cases} q_i + v_{i-1}(t - T_i) + \frac{1}{2} a_i (t - T_i + \frac{t_i^b}{2})^2 \\ \qquad \text{if } T_i - \frac{t_i^b}{2} \le t \le T_i + \frac{t_i^b}{2} \quad i \in \{1, ..., n\} \\ q_i + v_i (t - T_i) \\ \qquad \text{if } T_i + \frac{t_i^b}{2} \le t \le T_{i+1} - \frac{t_{i+1}^b}{2} \quad i \in \{1, ..., n-1\} \end{cases} \tag{18}$$

where $v_0 = v_{n+1} = 0$. The first line represents blend phases and the second line represents linear phases. We can also explicitly calculate the time derivatives of the trajectory:

$$\dot{q}(t) = \begin{cases} v_{i-1} + a_i (t - T_i + \frac{t_i^b}{2}) \\ \qquad \text{if } T_i - \frac{t_i^b}{2} \le t \le T_i + \frac{t_i^b}{2} \quad i \in \{1, ..., n\} \\ v_i \\ \qquad \text{if } T_i + \frac{t_i^b}{2} \le t \le T_{i+1} - \frac{t_{i+1}^b}{2} \quad i \in \{1, ..., n-1\} \end{cases} \tag{19}$$

$$\ddot{q}(t) = \begin{cases} a_i \\ \qquad \text{if } T_i - \frac{t_i^b}{2} \le t \le T_i + \frac{t_i^b}{2} \quad i \in \{1, ..., n\} \\ 0 \\ \qquad \text{if } T_i + \frac{t_i^b}{2} \le t \le T_{i+1} - \frac{t_{i+1}^b}{2} \quad i \in \{1, ..., n-1\} \end{cases} \tag{20}$$

IV. Turning Paths into Trajectories

Section III applied [1] and [2] to formulate trajectories that could be used to turn a path consisting of linear path segments into a smooth trajectory satisfying velocity and acceleration constraints. It assumed that we know the timing of the waypoints and the lengths of the blend phases. However, this information is not typically given by a path. Thus, we have to choose the timing of waypoints and the duration of the blend phases that satisfy the velocity and acceleration constraints of the robot. In this section we provide an automated solution to the complete problem.

We present an iterative approach to choose the timings of the waypoints and the duration of the blend phases. First, we choose the timing of the waypoints such that the velocities are maximized. The vector of all joint velocities during a linear segment is maximized if at least one joint is moving at its maximum velocity.

$$\Delta T_i = \max_j \frac{|q_{i+1}[j] - q_i[j]|}{v_{\max}[j]} \tag{21}$$

$q_i[j]$ denotes the j^{th} component of vector q_i.

The durations of the blend phases are chosen such that the accelerations are maximized.

$$t_i^b = \max_j \frac{|v_i[j] - v_{i-1}[j]|}{a_{\max}[j]} \tag{22}$$

Eq. 4 must be satisfied for the trajectory to be valid. Eq. 4 makes sure that there is no overlap between two blend phases. If two blend phases overlap, the resulting trajectory would not be differentiable and might not even be continuous. In the rest of this section we describe how we handle overlapping blend phases in order to ensure a valid trajectory.

If the blend phase around a single waypoint i overlaps with more than half of one of the neighboring linear phases and overlaps with the other blend phase neighboring that linear phase, then we are reducing the velocity of the two neighboring linear phases by the same factor $0 < f_i < 1$ such that the blend phase overlaps with at most half of each of the two neighboring linear phases.

$$f_i = \sqrt{\frac{\min\{\Delta T_{i-1}, \Delta T_i\}}{t_i^b}} \tag{23}$$

We are multiplying the velocity of the linear phase by the reciprocal of the factor the blend phase is overlapping too much. The square root is applied because linear velocities affect the overlap quadratically. Multiplying the neighboring velocities by some factor f_i reduces the blend duration by the same factor f_i. It also increases the duration of the neighboring linear phases by factor $\frac{1}{f_i}$. Both lead to a reduced overlap. The following lemma and proof show the correctness of Eq. 23 more formally.

Lemma 2: If the velocities v_{i-1} and v_i of the two neighboring linear phases of waypoint i are multiplied by factor f_i defined by Eq. 23 and the blend duration t_i^b is updated according to Eq. 22 using the new velocities, then the blend phase around waypoint i overlaps with exactly half of at least one of the two neighboring linear segments and overlaps with less than half of the other neighboring linear segment. More formally, if

$$v_{i-1,\text{new}} = f_i v_{i-1} \quad \wedge \tag{24}$$

$$v_{i,\text{new}} = f_i v_i \quad \wedge \tag{25}$$

$$t_{i,\text{new}}^b = \max_j \frac{|v_{i,\text{new}}[j] - v_{i-1,\text{new}}[j]|}{a_{\max}[j]} \tag{26}$$

then $$t_{i,\text{new}} = \min\{\Delta T_{i,\text{new}}, \Delta T_{i-1,\text{new}}\} \tag{27}$$

Proof: From Eq. 1, 24 and 25 follows

$$\Delta T_{i-1,\text{new}} = \frac{1}{f_i} \Delta T_{i-1} \tag{28}$$

$$\Delta T_{i,\text{new}} = \frac{1}{f_i} \Delta T_i \tag{29}$$

The following equations prove Eq. 27

$$t^b_{i,\text{new}} = \max_j \frac{|v_{i,\text{new}}[j] - v_{i-1,\text{new}}[j]|}{a_{\max}[j]} \tag{30}$$

$$= \max_j \frac{|f_i v_i[j] - f_i v_{i-1}[j]|}{a_{\max}[j]} \tag{31}$$

$$= f_i \max_j \frac{|v_i[j] - v_{i-1}[j]|}{a_{\max}[j]} \tag{32}$$

$$= f_i t^b_i \tag{33}$$

$$= \sqrt{\frac{\min\{\Delta T_{i-1}, \Delta T_i\}}{t^b_i}}\, t^b_i \tag{34}$$

$$= \sqrt{\min\{\Delta T_{i-1}, \Delta T_i\}\, t^b_i} \tag{35}$$

$$= \sqrt{\frac{t^b_i}{\min\{\Delta T_{i-1}, \Delta T_i\}}}\, \min\{\Delta T_{i-1}, \Delta T_i\} \tag{36}$$

$$= \frac{1}{f_i} \min\{\Delta T_{i-1}, \Delta T_i\} \tag{37}$$

$$= \min\left\{\frac{1}{f_i}\Delta T_{i-1}, \frac{1}{f_i}\Delta T_i\right\} \tag{38}$$

$$= \min\{\Delta T_{i-1,\text{new}}, \Delta T_{i,\text{new}}\} \tag{39}$$

■

So far we have only considered a single blend phase and its two neighboring linear phases. But we have to do the described slow-down procedure for all blend phases that are overlapping too much. A linear phase might have both of its neighboring blend phases overlap more than half. In this case the linear phase gets ascribed two slow-down factors, f_i and f_{i+1}. Now we have a conflict, because only for the specific slow-down factor f_i we can guarantee that the blend phase i is not going to overlap with more than half of the linear phase after the adjustment.

We are solving this conflict by applying the minimum of the two slow-down factors $\min\{f_i, f_{i+1}\}$. Note that this solution does not guarantee that the resulting blend phases are free of overlaps with each other. For one of the two neighboring blend phases the velocity of the linear phase gets reduced more than what was calculated to guarantee an overlap with less than half of the linear phase. The reason why this might lead to an invalid overlap is because one of the neighboring linear phases of the blend is slowed down unilaterally without slowing down the other neighboring linear phase to the same extent.

Although it is possible for this arbitration method to produce invalid overlaps, experimentally we found this unlikely. This is because we only reduce velocities. Reducing velocities always leads to a longer linear phase which makes it more likely for two blend phases not to overlap. If after the slow-down two blend phases are still overlapping, we repeat the slow-down procedure until no more blend phases overlap with each other. This reduces the optimality of the trajectory, but guarantees termination with a valid trajectory without overlapping blend phases.

V. Analysis

In this paper we presented a method for generating fast, valid trajectories that closely follow a path while satisfying velocity and acceleration constraints of robot joints. Experimentally, we have found this method to work well on a number of real and simulated test cases. At this time, we do not guarantee that the resulting trajectory is optimal. Furthermore, we do not take obstacles into account. Thus, because the generated trajectories deviate from the paths arbitrarily far, a valid trajectory may still collide with obstacles. However, since the trajectory is built largely from linear segments, it is likely to stay close to the path, and unlikely to hit an obstacle if the original path has sufficient obstacle clearance.

VI. Acknowledgements

This work is supported by Toyota Motor Engineering & Manufacturing North America (TEMA). We appreciate the great contribution to our robotics research and education.

References

[1] John J. Craig. *Introduction to Robotics: Mechanics and Control (3rd Edition)*. Prentice Hall, 3 edition, August 2004.

[2] B. Siciliano, L. Sciavicco, and L. Villani. *Robotics: modelling, planning and control*. Advanced textbooks in control and signal processing. Springer, 2009.