



All Faculty Publications

2000-09-01

A dynamic optimal trajectory generator for Cartesian Path following

Edward Red

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>

 Part of the [Mechanical Engineering Commons](#)

Original Publication Citation

Robotica 18.5(Sep 2): 451-458.

BYU ScholarsArchive Citation

Red, Edward, "A dynamic optimal trajectory generator for Cartesian Path following" (2000). *All Faculty Publications*. 589.
<https://scholarsarchive.byu.edu/facpub/589>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu.

A dynamic optimal trajectory generator for Cartesian Path following

Edward Red

Department of Mechanical Engineering, Brigham Young University, 435 CTB, P.O. Box 24201, Provo, Utah 84602-4201 (USA)

(Received in Final Form: January 12, 2000)

SUMMARY

This paper considers a dynamic and adaptive trajectory generator for negotiating paths using S-curves. Applying constant jerk transitions between the constant acceleration and deceleration periods of the trajectory, the trajectory will optimally transition to the desired speed setting. Optimal is defined to be the minimum time to transition from the current speed to the set speed for the move segment when jerk and acceleration are limited. The S-curve equations will adapt to instantaneous changes in speed setting and path length. An integrated motion planner will determine allowable speeds and transitional profiles based on the remaining move distance.

KEYWORDS: Trajectory generator; Cartesian path; S-curves; Optimal time; Integrated motion planner.

INTRODUCTION

Motion planning and trajectory generation lie at the heart of robotic and machine tool path control. New digital drive technologies and computational speeds are changing motion control paradigms that, in the past, relied on pre-processors to discretize, shape and buffer moves into a series of *micro* move segments passed to the controller. The path speeds were limited by how fast the discretized move segments could be processed by the controller, i.e. the block processing speed.

Free-form shapes (NURBS, B-splines, etc.) now form the core of most modern CAD/CAM systems. The current movement is to develop dynamic interpolators that can process object shapes directly, rather than decomposing them into small linear segments.

The challenge of modern motion planning and trajectory control is to process a series of *macro* moves dynamically and smoothly, while applying a set of limiting conditions such as speed, acceleration and jerk. This paper will show how this might be accomplished by proper motion planning and trajectory generation. An optimal approach is applied that adapts to instantaneous speed changes and increases in path length.

1. Historical perspective

In the past industrial trajectory generators (TG) used constant acceleration trapezoidal profiles to move to a desired speed setting. The simplicity of the profile shape

minimized the calculations for blending consecutive moves. Unfortunately, as the speeds and accelerations of modern robots and machine tools continued to increase, these simple shapes begin to stress the mechanism drives because of inherent acceleration discontinuities at the beginning and end of the profile.

The industrial term *S-curve* or *S-profile* describes the modern profile used to smooth the motion transitions, Figure 1. This curve can simply be a second or higher order polynomial in speed, with polynomial coefficients determined by transitional motion conditions. These profile types are described in robotic texts by Craig¹ and Parkin².

To reduce mechanism jerkiness and noise, more sophisticated polynomial profiles evolved. For example, Lloyd³ used a 5th order polynomial to blend two consecutive moves that are not necessarily tangent and that may change dynamically. This motion environment is representative of telerobotics or sensor driven control moves. Rather than examine speed optimality, Lloyd's algorithms were directed towards blending continuity and smoothness in the face of motion uncertainty.

2. Dynamic considerations

The advent of free form surfaces has spawned the development of *curve interpolators*. Wang⁴ describes a real-time quintic spline interpolator that generates a set of spline segments where $du/ds = 1$ over certain parameter ranges. This means that quintic curves can be parameterized and

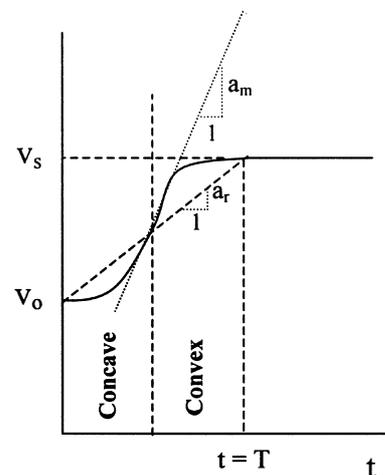


Fig. 1. S-curve profile.

fitted to these segments based on arc length. Because the curve parameterization methods apply numerical pre-processing methods such as Newton's method, they cannot adapt to dynamic changes in speed or path length.

In contrast, Yang⁵ developed an on-line TG capable of dynamically interpolating NURBS paths without shape pre-processing. An efficient and bounded predictor-corrector method uses finite difference theory to predict and correct the parametric changes necessary to generate the desired curvilinear distances along the trajectory.

Paper Function

This paper will demonstrate how motion planning can adaptively apply an S-curve to smooth the motion when both speed and move distance are changing dynamically, and subject to the following constraints:

- acceleration and deceleration along the trajectory cannot exceed max allowables;
- jerk is specified and limited;
- set speed can be attained if the remaining move distance is sufficient;
- move distance can be dynamically increased during the move when the move buffer detects a new tangent move.

This trajectory generator (TG) is quite low level since it is not concerned with the type of curve (linear, circular, NURBS, etc.) or space (Cartesian or joint) that defines the move. Yang⁶ references a broad survey of papers that consider task-level trajectory generation and motion planning, which is not the focus of this paper.

Rather, the focus is moving smoothly along a sequence of path segments subject to limiting motion conditions. The motion planner should determine the move length, speed setting, and the length of contiguous tangent moves.

The approach outlined minimizes the time to change speeds under constant jerk transitions and linear accel/decel periods, i.e. it is *time optimally smooth*. It is adaptive because it responds immediately to dynamic changes in speeds and distance. And the TG can be applied in both Cartesian space and joint space.

S-CURVE REVIEW

The motion conditions that determine the polynomial coefficients for a typical S-curve are zero acceleration at the transitional endpoints, the desired starting and ending speeds, and transitional time and/or distance. Other limiting conditions may be applied such as maximum jerk and acceleration.

For comparison Figure 1 includes a trapezoidal ramp with $a_r < a_m$ that will accomplish the same speed change in time T. Obviously then, polynomial transitions require more time than a trapezoidal transition when both use the same limiting acceleration. This is the price that you pay for smoothness.

Referencing Figure 1, we consider a pure S-curve (no linear transition) applied to the rise period from initial speed v_o to the set speed v_s . The form assumed for the S-curve speed profile is

$$v(t) = c_0 + c_1 t + c_2 t^2 \quad (1)$$

giving the acceleration and *constant jerk* equations:

$$a(t) = c_1 + 2c_2 t \quad (2)$$

$$j(t) = 2c_2 = \text{const} \quad (3)$$

The rise motion can be divided into 2 periods — a concave period followed by a convex period.

1. Concave period

The concave motion conditions are $v(0) = v_o$; $a(0) = 0$; $a(T/2) = a_m$; and $j(0) = j_m$, where j_m is the maximum jerk allowed for the mechanism, and a_m is the maximum acceleration encountered at the S-curve inflection point.

Applying these conditions to solve for the unknown constants in (1) and (2), we get $c_0 = v_o$; $c_1 = 0$; and $c_2 = j_m/2 = a_m/T$. We note the important relation

$$j_m = 2a_m/T \quad (4)$$

Equation (4) is important because it shows that the profile time, maximum acceleration, and maximum jerk are not independent of each other. If jerk is limited to the value j_m , then a_m and T must be chosen such that $2a_m/T \leq j_m$.

The equations for s (position), v (speed), and a (acceleration) along the concave portion of the S-curve become

$$s(t) = v_o t + j_m t^3/6 \quad (5)$$

$$v(t) = v_o + j_m t^2/2 \quad (6)$$

$$a(t) = j_m t \quad (7)$$

These equations apply for $0 \leq t \leq T/2$. Note also that at $t = T/2$,

$$v(T/2) = (v_s + v_o)/2 \quad (8)$$

$$s(T/2) = s_h = [v_o + a_m^2/(6j_m)]a_m/j_m \quad (9)$$

where s_h is the distance to the inflection point. This distance is expressed in terms of v_o , a_m and j_m , and not T.

2. Convex period

The convex period applies for $T/2 \leq t \leq T$. Letting time be zero as measured from the beginning of the convex period ($0 \leq t \leq T/2$), the pertinent motion conditions are $v(0) = v_h = (v_m + v_o)/2$; $a(0) = a_m$; $a(T/2) = 0$; and $j(0) = -j_m$.

Applying the motion conditions to solve for the unknown coefficients in (1)–(3), we get the equations for s, v, and a along the convex portion of the S-curve:

$$s(t) = v_h t + a_m t^2/2 - j_m t^3/6 \quad (10)$$

$$v(t) = v_h + a_m t - j_m t^2/2 \quad (11)$$

$$a(t) = a_s - j_m t \quad (12)$$

Over the time period T/2 the distance change measured from the halfway point to the ending point of the S-profile is

$$s(T/2) = [v_h + a_m^2/(3j_m)]a_m/j_m \quad (13)$$

Adding in the distance at the halfway point gives the total distance traversed in the S-curve, including both concave and convex sections. This simplifies to the total distance

$$S = (v_s^2 - v_o^2)/a_m \quad (14)$$

3. Max jerk considerations

The S-curve cannot be ideally fitted between all speed changes if the acceleration and jerk are maintained at their allowable limits. Given jerk j_m , starting speed v_o , and ending speed v_s , we determine speeds that respectively end the concave transition and begin the convex transition for an ideal S-curve:

$$v_1 = v_o + a_m^2 / (2j_m) \quad (15)$$

$$v_2 = v_s - a_m^2 / (2j_m) \quad (16)$$

By setting $v_1 = v_2$, we calculate the jerk that would fit the S-curve between the speed change $\Delta v = v_s - v_o$:

$$j_f = a_s^2 / \Delta v \quad (17)$$

The jerk computed in (17) may be less than allowable and thus sub-optimal, or exceed the allowable.

If we use the max allowable jerk, then it is possible for some Δv 's that the ending and beginning speeds overlap ($v_1 > v_2$). This will call for an *acceleration transition* to be made as described later in the paper.

If $v_1 < v_2$, then to minimize the time for a speed change we incorporate a linear transition at constant acceleration a_m . In the max jerk formulation the length of the linear transition is determined by the speed changes spent in the max jerk concave and convex periods. Once these changes in speed have been subtracted from the total Δv , the remaining speed change is left for a linear transition.

The equations that describe the concave and convex profiles are the same as (5)–(7) and (10)–(12), except that (5)–(7) applies for $t \leq t_1$, where t_1 is the time to complete the concave S-rise. In (10)–(12) v_h is replaced by v_2 .

The linear period is described by:

$$s(t) = v_1 t + a_m t^2 / 2 \quad (18)$$

$$v(t) = v_1 + a_m t \quad (19)$$

where jerk is zero and $0 < t \leq T - 2t_1$.

ADAPTIVE S-CURVE

The adaptive S-curve must respond immediately to speed changes or to changes in path length. For example, a machine operator may increase or decrease the feedrate during a face milling operation using a thumbwheel. The path length may increase because a motion buffer frees a slot for a move that is tangent to the move currently being processed.

To adapt to these changes the algorithms must allow for non-zero initial speed, jerk, and acceleration and also for the possibility of a set speed change during the move, Figure 2. Continuity in position, speed, and acceleration must be maintained, but discontinuity in jerk is allowed. Note that jerk discontinuity is built into the S-curve profile when transitioning to or from any period spent in constant acceleration. This includes the linear ramps and the constant speed periods.

1. Adaptive algorithm

To make the ideal S-curve both adaptive and responsive when the set speed changes during a move, we apply the

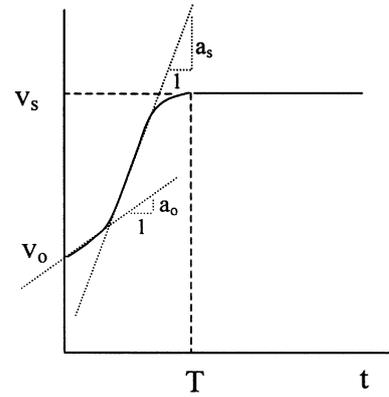


Fig. 2. Adaptive S-curve profile.

motion conditions $v(0) = v_o$; $v(T_c) = v_s$; $a(0) = a_o$; and $a(T_c) = 0$. Given the entry conditions, T_c becomes the time to complete the transition to the new set speed. To evaluate the transition requirements, we *force a convex profile* and determine the jerk required to complete the transition without considering any bounds on the jerk value. We then compare this jerk value to the maximum jerk as follows.

Applying the motion conditions leads to these equations:

$$v(t) = v_o + a_o t - a_o t^2 / (2T_c) \quad (20)$$

$$a(t) = a_o - a_o t / T_c \quad (21)$$

Applying the condition $v(T_c) = v_s$ leads to the following equation for T_c :

$$T_c = 2 \Delta v / a_o \quad (22)$$

where $\Delta v = v_s - v_o$.

If $\Delta v > 0$, there will be a real positive solution for T_c , but if $\Delta v = 0$ and $a_o \neq 0$, then $T_c = 0$ would specify an infinite jerk. Thus, a speed change can cause an undesirable jerk situation. The response to the speed change must not exceed the maximum acceleration or maximum jerk.

For $\Delta v > 0$ the jerk is calculated from

$$j(T_c) = -a_o / T_c \quad (23)$$

If the absolute value of (23) exceeds the maximum jerk allowed ($a_o / T_c > j_m$), then a transition must be made to an intermediate speed using the maximum jerk.

Figure 3 shows how the transition must be made to new set speeds, when the jerk is limited and where v_s' is the old set speed and v_s is the new set speed.

In 3a) the jerk is not exceeded in moving to the new set speed. In 3b) and 3c) the maximum jerk is exceeded and the transition to the new set speed will require a transition to an intermediate speed v_i . At this intermediate speed the acceleration is zero. This profile can then be followed by a normal S-curve transition to the new set speed indicated by v_s .

The transition to the new set speed is thus broken into two phases:

- jerk transition to speed v_i where the acceleration is zero.
- normal S-curve transition from v_i to v_s .

The next few sections will consider the transitional cases that must be handled by the adaptive generator in more detail.

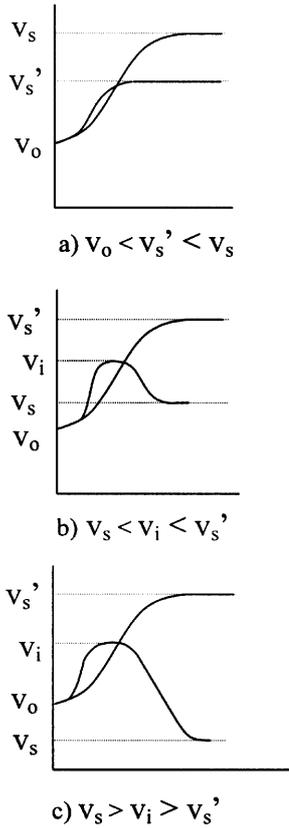


Fig. 3. Adapting to changing speeds.

2. Transitions where $\Delta v > 0$ and $j(T_c) < j_m$

Figure 3 presents a number of subtleties. If $\Delta v_s > 0$, we apply the check $a_0/T_c < j_m$ to see if the maximum jerk is exceeded. If so, we transition to v_i and call this case a *speed transition*.

If the maximum jerk is not exceeded, then we must determine how to transition to the new set speed. The transition possibilities vary, depending on whether we are initially in a concave period or convex period and the relative difference in the set speed.

In the optimal sense we choose a set of moves that always uses the maximum jerk. To determine the proper transition, we note that equation (16) determines the speed that begins a max jerk convex transition to the new set speed. We let this initial speed be represented as v_c and thus repeat an earlier equation as $v_c = v_s - a_m^2 / (2j_m)$.

We also note that the time to complete the convex transition is $T_c = a_m / j_m$. At T_c the distance change required for the convex transition is $s(T_c) = [v_c + a_m^2 / (3j_m)] a_m / j_m$.

3. Transitions where $v_0 < v_s < v_s'$ and v_0 on concave profile

The problem is to make a constant jerk transition from a concave profile when the set speed is reduced below the old set speed but exceeds the current speed v_0 . To resolve this problem, we compare v_c to the speed that ends the concave portion of the current transition, given the current entry conditions.

The time, speed, and distance required to complete the concave transition are

$$t_1 = (a_m - a_0) / j_m \tag{24}$$

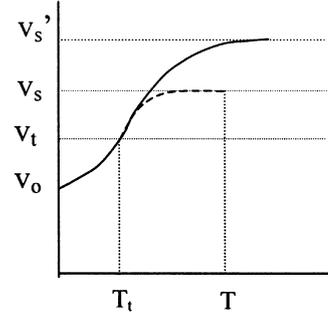


Fig. 4. Acceleration transition.

$$v_1 = v_0 + (a_m^2 - a_0^2) / (2j_m) \tag{25}$$

$$S_1 = v_0 t_1 + a_0 t_1^2 / 2 + j_m t_1^3 / 6 \tag{26}$$

If $v_c \geq v_1$, then the transition to the new set speed occurs in one of two ways:

- If $v_c = v_1$, we complete the current concave transition to speed v_1 , then follow it with a convex transition from v_1 to the new set speed. This is referred to as a *jerk transition*.
- If $v_c > v_1$, then we complete the current concave transition, follow it with a linear transition between v_1 and v_c , and then, beginning at v_c , complete a new convex transition to the new set speed. This is referred to as a *linear transition*.

If $v_c < v_1$, then the Figure 4 case emerges. This case, referred to as an *acceleration transition*, can be solved by noting that the speed and acceleration for the previous concave curve and the new convex curve must be equal at T_t where the speed is v_t .

In this case the speed change is sufficiently small that we cannot reach the maximum acceleration. Nevertheless, there exists a point where the concave profile from the previous condition will be tangent to the convex profile from the new transition. This point will lie between v_0 and v_s . At this point the acceleration and speed of both profiles are the same, although there will be a sign change in the jerk.

The pertinent equations are:

$$v_0 + a_0 T_t + j_m T_t^2 / 2 = v_s - j_m (T - T_t)^2 / 2 \tag{27}$$

$$a_0 + j_m T_t = j_m (T - T_t) \tag{28}$$

Solving these equations, we get:

$$T = [-a_0 + \text{sqrt}(2a_0^2 + 4\Delta v j_m)] / j_m \tag{29}$$

$$T_t = (j_m T - a_0) / (2j_m) \tag{30}$$

4. Transitions where $v_0 < v_s' < v_s$ and v_0 on convex profile

We take the initial conditions and determine the speed v_1 that would conclude a concave transition. We again calculate v_c , and then compare v_c to v_1 . If $v_c \geq v_1$, then the transition to the new set speed occurs as either a jerk or linear transition.

5. Transitions where $v_0 < v_s' < v_s$ and v_0 on linear profile.

We again calculate v_c then compare v_c to v_0 . If $v_c > v_0$, then the transition to the new set speed is attained by a linear transition from v_0 to v_s . If $v_c = v_0$, then we apply a jerk transition to v_s .

6. Speed transition

In a speed transition any entry state is immediately converted to a convex profile with jerk $-j_m$. This minimizes the time for the transition.

The motion conditions are $v(0) = v_o$; $v(T_i) = v_i$; $a(0) = a_o$; $a(T_i) = 0$; and $j(0) = -j_m$. These conditions lead to the equations:

$$s(t) = v_o t + a_o t^2/2 - j_m t^3/6 \tag{31}$$

$$v(t) = v_o + a_o t - j_m t^2/2 \tag{32}$$

$$a(t) = a_o - j_m t \tag{33}$$

where s is interpreted as a position change. Applying $a(T_i) = 0$ leads to the solution for the transition time:

$$T_i = a_o/j_m \tag{34}$$

Substituting T_i into (31) and (32) gives the solutions for S_i and v_i at T_i . The transitional modes to get from v_i to the new set speed now depend on the speed difference.

7. Deceleration transitions

We have not examined all possible dynamic move situations. In the cases considered thus far, the set speed is either greater than the entry speed and/or the entry acceleration is positive.

When the new set speed is less than the entry speed and/or the entry acceleration is negative, transitional requirements will be applied that are similar to those already considered. For brevity we will not examine these cases directly, although the next section indirectly considers deceleration situations when the move is to be ended with non-zero speeds and accelerations. This is the case where a tangent move enters the buffer and the motion type is continuous path, i.e. it is desired to maintain the speed across contiguous moves.

8. Non-zero terminal conditions

The function of the Motion Planner (MP) is to accumulate the length of the tangent moves in the buffer that follow the current move. Yang⁶ shows that length computations are not computationally intensive and can be made dynamically, providing accurate estimates even for moves along parametric curves. As the planner updates the tangent distance they are made available to the TG.

When the current move is followed by one or more tangent moves, the MP must provide continuity across the boundary between the current move and the next move. This means that the move will terminate with non-zero speed. The ending acceleration and profile type will depend on the desired speed and the remaining distance of the contiguous tangent moves.

MOTION PLANNING

It is the purpose of the motion-planning algorithm to determine if an increased speed can be reached in the time and distance available. If not, the MP must set a speed that can be attained. The Motion Planner (MP) then supplies the time step to the TG and returns the state at the completion of this step.

Two observations are useful here:

- A set speed that is reduced from the previous value will result in the TG attaining satisfactory terminal conditions. Thus, speed can be reduced without a speed check.
- A set speed that is increased from the previous setting may result in unsatisfactory terminal conditions. Thus, the set speed will have an upper bounds based on the current motion state.

1. Planning approach

The normal approach for a Motion Planner (MP) is to determine a feasible upper bound to the set speed, given the remaining move distance, and other terminal motion conditions. Unfortunately, a closed-form solution proves intractable, given the practical constraints of a real-time TG.

The complexity arises because we are never sure what form the S-curve will take as the set speed is changed. Without knowing whether the S-curve can attain its max acceleration, trapezoidal profiles cannot be used to estimate the top set speed that can be reached within the remaining move distance.

The solution is to make a max jerk step to increase the entry speed towards the new set speed. In effect, we incrementally adjust the speed up, but within the bounds of remaining distance and other motion conditions. We only increase the speed if we can meet the terminal conditions.

We do not predict analytically the max speed attainable at each step, but instead predict a set speed that is attainable under the current state, given the remaining distance and other motion conditions. This approach is *incrementally optimal*.

2. Planning algorithm

The MP considers the entry state (v_o, a_o, L_o) and the desired set speed (v_s). The other important parameters that are used to make transitional decisions are:

L_m = total distance of the current move

L_R = remaining distance to complete the move = $L_m - L_o$
(calculated at entry)

L_f = distance of following tangent moves. For point to point moves this parameter will be passed into the TG with a 0 value.

L_a = available distance to complete decel to 0 speed from current motion state = $L_R + L_f$

There are numerous transition possibilities depending on the entry state and the set speed (see Table I). States 1–7 are self-explanatory, while 8–13 need further explanation. CONVEX_INT and CONCAVE_INT refer to an entry state where the new set speed calls for a *speed transition* to an intermediate speed (v_i).

The defined constants that include PARTIAL in their name refer to an entry state and speed change that require an

Table I. Profile numbers for each transitional state

Transitional State	State Number	Transitional State	State Number
S_RISE_CONCAVE	1	CONVEX_INT	8
S_RISE_LIN	2	CONCAVE_INT	9
S_RISE_CONVEX	3	PARTIAL_CONCAVE_RISE	10
CONST_SPD	4	PARTIAL_CONVEX_RISE	11
S_FALL_CONVEX	5	PARTIAL_CONVEX_FALL	12
S_FALL_LIN	6	PARTIAL_CONCAVE_FALL	13
S_FALL_CONCAVE	7		

acceleration transition because we cannot reach the maximum acceleration a_m .

To accommodate the transition through the various states, we define

n_p = number of profiles traversed

\mathbf{P} = vector of profile types

\mathbf{L}_p = distance vector

\mathbf{V}_p = speed vector

\mathbf{A}_p = acceleration vector

\mathbf{T}_p = time vector

where each vector is of size 7. There are never more than 7 transitional states for any entry state.

\mathbf{P} is a vector that stores the profile type for each segment in sequence. The remaining vectors store the boundary values necessary to transition each intermediate profile given the remaining move distance and desired set speed. For example, we could enter on a CONST_SPD transition, then have the speed increased, requiring an order $\mathbf{P} = [1, 2, 3, 4, 5, 6, 7]$.

One trajectory step in time is then used to step through the profiles, occasionally moving through more than one profile type in the \mathbf{P} sequence.

3. MP transitions

The algorithms presented here step into Case 1 or 2 when the set speed is non-zero, depending on the entry state. Case 3 considers the special case where the set speed is zero (abort, stop mechanism in middle of move):

Case 1: If the entry state is CONST_SPD ($a_0 = 0$) and $v_0 = v_s$, then determine whether immediate decel is to begin or a time step at constant speed can be made, given the remaining distance. Enter TG and take a time step based on distance left and transition as needed through profiles. Exit the TG and MP.

Case 2: For entry rise or fall states take the desired set speed and other entry conditions, and determine whether the set speed can be attained, given the available distance L_a . If so, take a normal time step and exit the TG and MP. If the conditions are such that you cannot reach the desired set speed, then apply a max jerk step to increase the speed and exit the TG and MP.

Case 3: If set speed is zero and current speed is non-zero, then decelerate to zero speed. In these cases zero speed may be attained before the end of the move is reached.

Case 4: If the entry conditions reflect a *quiet* state (previously aborted or attained zero speed before move distance reached), simply return.

Case 2 incorporates the *incrementally optimal* move described earlier when the desired set speed cannot be attained. By applying the transitional steps properly in each step, Case 2 conditions will always permit a max jerk step to increase the speed. But how well we approach the max possible set speed depends on the time steps being much less than the time to make speed changes. It is recommended that these methods only be applied at trajectory rates that exceed 100 Hz. Note that the higher the rate, the closer the incremental optimal speed will approach the max possible speed.

To transition the profiles, the entry time step is compared to the ending and beginning times for the sequence of profiles stored in \mathbf{P} . We thus establish the profile that bounds this step in time. Once established we apply the appropriate equations presented earlier to determine the terminal conditions and then return.

4. Cubic solution

When speed is to be maintained across contiguous tangent moves, the TG encounters the problem of determining the time to complete the current move given the remaining distance L_R . This occurs because it is possible that the current move may complete within a profile rather than at the profile endpoint. And it is critical that the time step to complete the current move never be less than the entry value.

The profile transitions must compare the distance that ends the profile to the distance remaining for the current move. The remaining distance can be used to calculate the time to complete the move. The MP then makes a terminal time step that is never less than the entry step.

Reviewing the previous equations it is evident that many of the profile types will require a cubic solution for time in terms of the remaining current move distance. This time can then be compared to the trajectory step to determine how and when to complete the current move. The cubic solutions implemented in the MP can be found at <http://www2.hawaii.edu/suremath/jrootsCubic.html>. Picking the correct solution is simply selecting the minimum real time from the three possible solutions.

DEMONSTRATIONS

These algorithms have been programmed and tested in C and are being used in university and commercial trajectory motion applications.

Figures 5–7 show the adaptive characteristics of the MP/TG combination for a case where the length of the original move is 400 mm. The set speed is changed several times throughout the move. At 300 mm into the move, another tangent move enters the move buffer of length 100 mm. The MP immediately knows to generate a continuous path profile transition.

The maximum acceleration is set to 4000 mm/s² and the maximum deceleration is set to 3000 mm/s² in each case, while the jerk maximum is increased by ×10 in each case.

The acceleration values are comparable to those of commercial robots.

Given a low jerk value of 5000 mm/s³, Figure 5 shows an expected sloppy response to the dynamic speed changes. A jerk value of 50000 mm/s³ improves the velocity response markedly in Figure 6. And this value of jerk should be reasonable for many mechanisms.

When the jerk is set high to simulate infinite jerk, the S-curve in Figure 7 approximates a trapezoidal response profile. Thus, the max jerk TG can closely approximate a trapezoidal profile generator, if necessary.

Obviously, the jerk value becomes critical in the trade-off between mechanism responsiveness and smoothness. Ultimately, the mechanisms’s design characteristics will decide

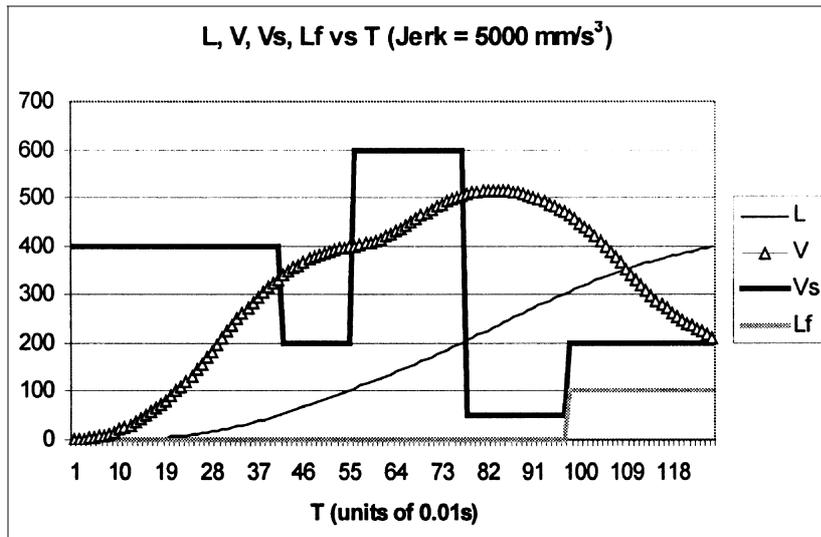


Fig. 5. S-curve response to changing V_s and L_f for jerk = 5000 mm/s³.

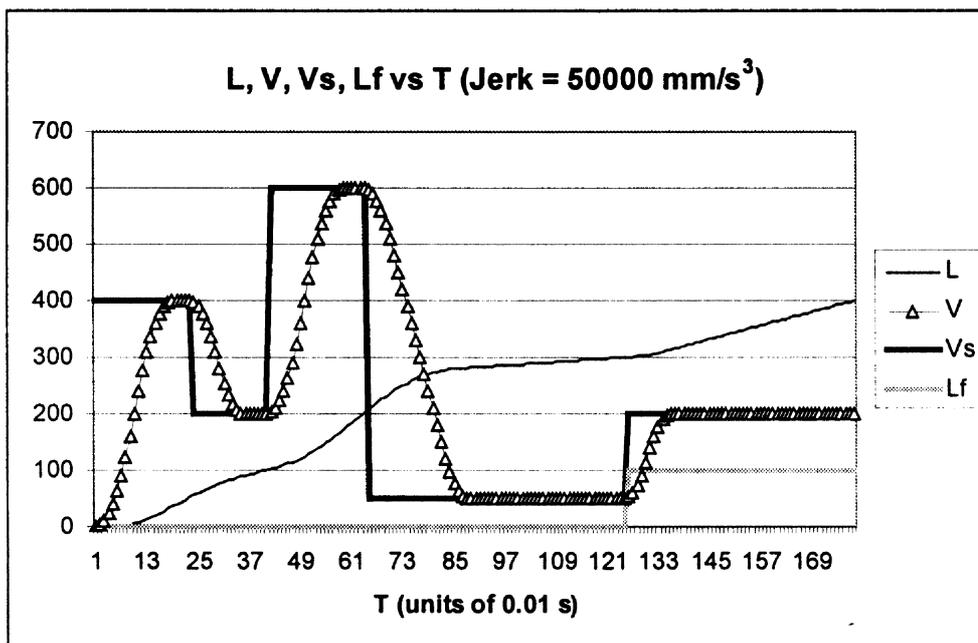


Fig. 6. S-curve response to changing V_s and L_f for jerk = 50000 mm/s³.

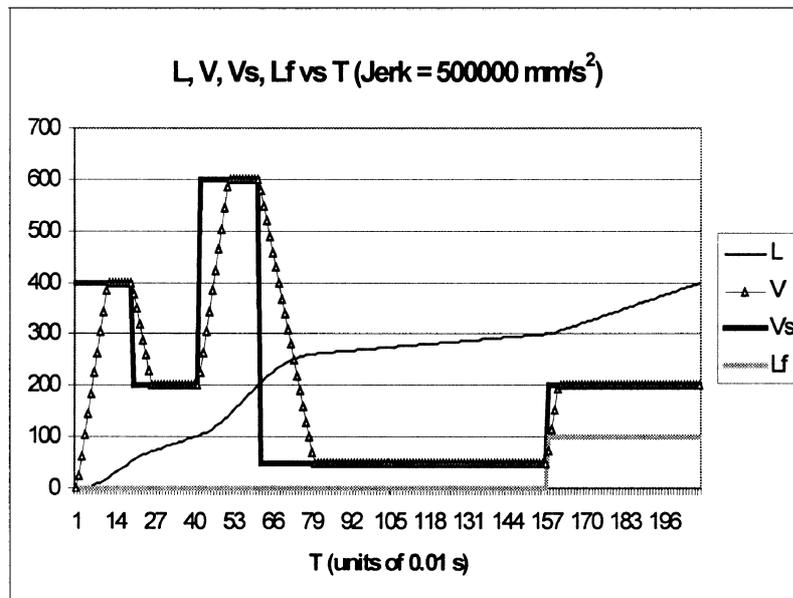


Fig. 7. S-curve response to changing V_s and L_f for jerk = 500000 mm/s³.

what is acceptable. Smoothness of motion versus responsiveness is a trade-off that can be easily made using these algorithms. Designers can always test the mechanism with different jerk values to get the correct blend of responsiveness and smoothness.

Although this example illustrates Cartesian path following, these methods extend just as well to joint moves by ensuring the parameters are in radians.

In some Cartesian moves, the tool control frame may experience more orientation change than position change, e.g. when a screw vector is used to represent tool orientation change. The MP/TG will accommodate orientation change if the motion parameters are entered in radians.

1. Computational speed

Placed into a large computational loop and tested over a variety of profile shapes, these algorithms required less than 40 μ s per trajectory step. These algorithms were run under Windows NT 4.0 on an IBM 300 MHz PC.

CONCLUSIONS

Applying constant jerk transitions between the constant acceleration and deceleration periods of the trajectory, the trajectory generator will transition to the desired speed setting in the minimum time, when jerk and acceleration are limited.

Equations are derived that make the S-curve adaptive to dynamic changes in speed setting and to increases in path length. If the desired set speed cannot be attained, then a max jerk acceleration is used to increase the set speed by an incrementally optimal amount.

A dynamically integrated motion planner determines allowable speeds and transitional profiles based on the remaining move distance. In some cases the transitions require an intermediate constant acceleration period.

The methods apply to Cartesian moves and joint moves equally well. In addition, the algorithms are very fast, averaging less than 40 μ s per trajectory step.

References

1. J.J. Craig, *Introduction to Robotics: Mechanics and Control* (Addison-Wesley, 2nd edition, 1989).
2. R.E. Parkin, *Applied Robotic Analysis* (Prentice-Hall, 1991).
3. J. Lloyd and V. Hayward, "Trajectory Generation for Sensor-Driven and Time-Varying Tasks" *Int. J. Robotics Research* **12**, 380–393 (August, 1993).
4. F.C. Wang and P.K. Wright, "Open Architecture Controllers for Machine Tools, Part 2: A Real-Time Quintic Spline Interpolator" *ASME Journal of Manufacturing Science and Engineering* **120**, 425–432 (May, 1998).
5. Z. Yang and E. Red, "On-line Cartesian trajectory control of mechanisms along complex curves" *Robotica* **15**, Part 3, 263–274 (1997).
6. Z. Yang, "On-line Cartesian Trajectory Control of Mechanisms along Complex Curves" *PhD Dissertation* (Brigham Young University, September, 1995).